Neural Network Based Terminal Iterative Learning Control for Tracking Run-Varying Reference Point

Tianqi Liu, Danwei Wang
EXQUISITUS, Centre for E-City,
School of Electrical and
Electronic Engineering,
Nanyang Technological University,
639798, Singapore

Ronghu chi
School of Automation
& Electrical Engineering,
Qingdao University of
Science & Technology,
Qingdao, 266042, China

Qiang Shen
EXQUISITUS, Centre for E-City,
School of Electrical and
Electronic Engineering,
Nanyang Technological University,
639798, Singapore

***Abstract*—In this paper, a neural network based terminal iterative learning Control (NNTILC) method is proposed for a class of discrete time linear run-to-run systems to track run-varying reference point. An iterative training radial basis function (RBF) neural network is developed to estimate the effect of initial condition on terminal output and to learn the changes in initial condition iteratively at the same time. By involving these information in the control scheme, the proposed NNTILC can drive the system to track run-varying reference point fast and precisely beyond the initial disturbance and reference change. Stability and convergence of this NNTILC method is proved and computer simulation result is provided to confirm its effectiveness further.**

I. INTRODUCTION

Iterative learning control (ILC), initially proposed by Arimoto [1], is a control scheme that updates and refines the control sequence by errors in the last trial for repetitive operation systems. It has been successfully used into repetitive process in industry [2], [3], [4]. The development and methodology of ILC can be found in [5], [6], [7].

However, the ILC approach works based on the measurement signal in the entire trajectory which may not be accessible in some real industry cases. In order to overcome this problem, Chen et al. [8] developed a terminal iterative learning control (TILC) method. TILC adjusts the set point of repetitive systems based only on the terminal errors in the previous trials instead of the tracking error in the whole trajectory. By doing so, the idea of ILC is successfully applied into systems focus solely on terminal output. Investigations [9], [10] have shown that TILC can achieve convergence in iteration domain.

In previous works, most ILC and TILC approaches consider only the cases for tracking run-invariant fixed reference. Then ILC and TILC can update the controller based on errors run by run to track the fixed reference or trajectory. However, this makes the control scheme quite depend on the reference, that is, once the reference changes, the controller has to learn again with another learning process.

Moreover, TILC approach requires to set the initial condition to exactly the same value in every run[11], [12], which also limits the application of TILC in industry. Previous works in ILC solve the problem through the following methods. 1) ILC with initial state learning scheme[13], in which the controller learns the initial condition firstly in every run to make the initial condition the same in every run; 2) Multirate ILC schemes [14], in which the input update rate is different from the sampling rate of feedback system or the input update rates of ILC are different at low and high frequency bands; 3) Cutoff frequency phase-in profile[15], in which the cutoff frequency of the filter for tracking error is time-varying and follows a predefined profile. However, the above scheme is complicated for realization and the performance is not quite good.

In this paper, a neural network based terminal iterative learning control (NNTILC) method is proposed to solve the problems of initial condition disturbance and tracking run-varying reference point. Neural network has been proved to be efficient in function approximation and parameter estimation[16]. So in this paper the effect of initial condition on the terminal output is estimated by a neural network, and at the same time the pattern change of initial condition is also learned iteratively. Then by conducting the control law involving the effect of initial condition as well as the reference information, the system can track the run-varying reference fast and precisely beyond the initial disturbance and reference change. Considering that initial condition may not be accessible for parameter estimation in the same run, a RBF neural network is introduced to estimate the effect using the signal in the last trail taking the advantage of repetitive operation systems and repetitive disturbance. Convergence analysis of the proposed method is derived mathematically, and simulation results confirm the effectiveness of the proposed NNTILC method further.

The remainder of this paper is organized as follows. In section II, the structure of the problem is introduced and at the same time a new NNTILC controller is designed. In section III, convergence analysis of the proposed NNTILC method is derived. Section IV presents computer simulation results to illustrate the effectiveness of NNTILC method and also compares its performance with conventional TILC method. Section V draws some final conclusions.

II. Problem Formulation and Controller Design

Consider a class of discrete time dynamical linear systems as follows:

$$\begin{aligned} x_k(t+1) &= Ax_k(t) + Bu_k(t) \\ y_k(t+1) &= Cx_k(t+1) \end{aligned} \quad (1)$$

where $t = 0, 1, 2, ..., T$ is the sampling index and k is the iteration index. Matrices A, B and C are time invariant with appropriate dimensions; $x_k(t) \in R^p$ is the state vector, $y_k(t) \in R^n$ is the output vector, and $u_k(t) \in R^m$ is the control vector at the $t-th$ sampling time in the $k-th$ run. The system runs on time interval $[0, T]$. In run-to-run control, the control input is required to be a constant at all sampling times in the same run, i.e., $u_k(t) = u_k$ for all $t = 0, 1, 2, ..., T$.

The proposed controller in this paper aims to track a single terminal point r_k at T in the $k-th$ run. The system is both controllable and observable.

From (1), the relationship between the system terminal output $y_k(T)$ and initial state $x_k(0)$ can be developed as

$$y_k(T) = F_k + B^* u_k \quad (2)$$

where $B^* = \sum_{m=0}^{T-1} CA^{T-1-m}B$ and $F_k = CA^T x_k(0)$ is the effect of $x_k(0)$ on the terminal output. It should be noted that B^* is full rank, because the system is both controllable and observable. What is more, the initial condition in this paper is run-varying with repetitive bounded noise; F_k is unknown because $x_k(0)$ is not accessible in the $k-th$ run.

The tracking error of the system is defined as

$$e_k = r_k - y_k \quad (3)$$

where $r_k = f(k)$ is the terminal single reference point at time instant T, which is run-dependent with k. $f(\cdot)$ can be any function of k.

Substitute (2) into (3), the tracking error dynamic of the system becomes

$$e_k = r_k - F_k - B^* u_k \quad (4)$$

In this proposed NNTILC, a RBF neural network is introduced to estimate F_k by using $x_{k-1}(0)$.

It is assumed that there exist an ideal function of $x_{k-1}(0)$ and neural network which make

$$F_k = CA^T x_k(0) = F(x_{k-1}(0)) = W_D^T \phi(x_{k-1}(0)) \quad (5)$$

where $F(\cdot)$ denotes the ideal function, $W_D^T \in R^{n\times L}$ is an unknown optimal NN weights matrix, L denotes the number of neurons in the hidden layer, $\phi(\cdot)$ denotes a known vector of basis activation function and $\phi(x_{k-1}(0)) \in R^L$ is the output of the neural network's hidden layer.

Remark 1: ILC and TILC are usually used into repetitive operation systems. The resetting pattern and disturbance of initial state is also repetitive. So it is reasonable to assume that this repetitiveness can help to estimate $x_k(0)$ by $x_{k-1}(0)$.

The basis activation function is chosen as

$$\phi_i(x) = \exp\left(-\frac{\|x-\mu_i\|^2}{2\sigma_i^2}\right) \quad (6)$$

where $\mu_i \in R^p$ and σ_i are the centre and width of the $i-th$ hidden neuron, receptively, $i = 1, 2, ..., L$. In this paper, only the weight matrix is updated iteratively. The centre and width of the hidden neurons can be initialized by several history data or chosen randomly around the reference trajectory.

The approximation of $F(x_{k-1}(0))$ in the $k-th$ run is

$$\hat{F}(x_{k-1}(0)) = \hat{W}_k^T \phi(x_{k-1}(0)) \quad (7)$$

where $\hat{W}_k^T$ is the estimation of NN weight matrix in the $k-th$ run.

Since B^* is full rank, the control law then can be constructed as

$$u_k = {B^*}^{-1}[r_k - \hat{W}_k^T \phi(x_{k-1}(0))] \quad (8)$$

Substitute (8) into the system dynamic (2),

$$y_k = F_k + r_k - \hat{W}_k^T \phi(x_{k-1}(0)) \quad (9)$$

From (9), it is obvious that as long as our neural network can approximate F_k precisely, the output of the system will track exactly the reference.

In order to train the neural network, another neural network updating law is introduced as

$$\hat{W}_{k+1}^T = \hat{W}_k^T - \alpha e_k \phi^T(x_{k-1}(0)) \quad (10)$$

where α is a learning gain, which affects the convergence of the proposed method.

III. Convergence Analysis

Theorem 1: For MIMO discrete-time linear system (1), if the α in the (10) satisfy $0 < \alpha < \frac{2}{L}$, where L is the number of neurons in hidden layer of the neural network, then the control law (8), along with the neural network updating law (10), can guarantee that

1) the weight matrix W_k^T of the RBF neural network is convergent in the sense of Lyapunov function $(\tilde{W}_{k+1}^T \tilde{W}_{k+1} - \tilde{W}_k^T \tilde{W}_k) \leq 0$.

2) the terminal tracking error e_k converges to zero asymptotically as k approaches infinity.

Proof: From (5), it is obvious that there exist an ideal weight matrix W_D^T, which makes

$$F_k = W_D^T \phi(x_{k-1}(0)) \quad (11)$$

Substitute the control law (8) and (11) into the error dynamic (4), yields

$$\begin{aligned} e_k &= r_k - W_D^T \phi(x_{k-1}(0)) - r_k + \hat{W}_k^T \phi(x_{k-1}(0)) \\ &= \tilde{W}_k^T \phi(x_{k-1}(0)) \end{aligned} \quad (12)$$

where $\tilde{W}_k^T = \hat{W}_k^T - W_D$ is the weight estimation error of the neural network.

Subtract W_D^T from both side of the NN updating law (10), it is easy to get

$$\tilde{W}_{k+1}^T = \tilde{W}_k^T - \alpha e_k \phi^T(x_{k-1}(0)) \tag{13}$$

From (6), it can be derived that $\phi(\cdot)$ satisfies

$$0 \le \phi_i(\cdot) \le 1, i = 1, 2, ..., L \tag{14}$$

where $\phi_i(\cdot)$ is the $i-th$ entry of $\phi(\cdot)$.

Since there are L neurons in the hidden layer, i.e., $\phi(\cdot) \in R^{L\times 1}$, together with (14), it is obvious that

$$0 \le \phi^T(x_{k-1}(0))\phi(x_{k-1}(0)) \le L \tag{15}$$

for any k.

Define energy function as

$$J = trace(\tilde{W}_{k+1}^T \tilde{W}_{k+1}) \tag{16}$$

then

$$\Delta J = trace(\tilde{W}_{k+1}^T \tilde{W}_{k+1} - \tilde{W}_k^T \tilde{W}_k) \tag{17}$$

From (13), the Lyapunov function can be constructed as in (17) which is shown at the end of this page.

Simplify (17), together with (12), the the Lyapunov function becomes

$$\begin{aligned}
&\Delta J = trace\left[-2\alpha e_k^T e_k + \alpha^2 e_k \phi^T(x_{k-1}(0))\phi(x_{k-1}(0)) e_k^T\right] \\
&\le trace\left[-2\alpha e_k^T e_k + \alpha^2 L e_k^T e_k\right] \\
&= \alpha\left(\alpha L - 2\right) \cdot trace\left(e_k^T e_k\right) \\
&= \alpha\left(\alpha L - 2\right) e_k^T e_k
\end{aligned} \tag{18}$$

By choosing

$$0 < \alpha < \frac{2}{L} \tag{19}$$

Together with (18), it can be derived that

$$\Delta J \le \alpha\left(\alpha L - 2\right) e_k^T e_k < 0 \tag{20}$$

By using (18) repetitively, (18) can be rewritten into

$$\begin{aligned}
&trace(\tilde{W}_{k+1}^T \tilde{W}_{k+1}) \\
&\le tarce(\tilde{W}_0^T \tilde{W}_0) - \sum_{i=0}^{k} \alpha\left(\alpha L - 2\right) e_i^T e_i
\end{aligned} \tag{21}$$

Since $\tilde{W}_{k+1}^T \tilde{W}_{k+1}$ is non-negative and bounded, from (21), together with (19)

$$\lim_{k\to\infty} \alpha\left(\alpha L - 2\right) e_k^T e_k = 0 \tag{22}$$

(15) implies that $\alpha\left(\alpha L - 2\right)$ is bounded and non-zero, so from (22)

$$\lim_{k\to\infty} e_k^T e_k = 0 \tag{23}$$

■

IV. SIMULATION

In order to illustrate the effectiveness of the proposed NNTILC scheme thoroughly, simulations on SISO system and MIMO system are done separately in this section.

What is more, the performance of the proposed NNTILC method and the conventional TILC method, whose control updating law is constructed as equation (24), are compared.

$$u_{k+1} = u_k + l \cdot e_k \tag{24}$$

where u_{k+1} and u_k are control input in the $(k+1)-th$ and $k-th$ run respectively, e_k is the terminal tracking error in the $k-th$ run, and l is the learning gain.

In this section, both of the parameters α and l are chosen by trail and error. In NNTILC, α affects the convergence according to theorem 1. In order to guarantee a more flexible choice of α, the number of hidden neurons should be as small as possible. Usually, a larger α comes with a faster convergence rate. But α should not be too large since overshoot may be introduced as a result.

Besides, the activation function as shown in (6) is adopted in this simulation. The neural network is initialized randomly with 3 hidden neurons and the centres being chosen around the reference trajectory.

A. Simulation on SISO System

Firstly consider the following discrete time SISO system

$$\begin{aligned}
x_k(t+1) &= \begin{pmatrix} 0.5 & 0.035 & 0.025 \\ 0.255 & 0.6 & -0.99 \\ 0.75 & 0.03 & 0.025 \end{pmatrix} x_k(t) \\
&+ \begin{pmatrix} 0.2 & 0.2 & 0.0 \end{pmatrix}^T u_k(t) && (25) \\
y_k(t) &= \begin{pmatrix} 1.0 & 0.0 & 1.0 \end{pmatrix} x_k(t) && (26)
\end{aligned}$$

where the system operates on time interval $[0, 5]s$ for every run.

Case 1: Tracking Run-invariant Terminal Reference Point

In this simulation, the NNTILC approach and the conventional TILC method are used for tracking run-invariant terminal reference point $y_k^d = r_k = 10$ at $T = 5$ for every k, with $\alpha = 0.6$ and $l = 1.0$. Here, a random noise is added to the initial condition x_0 as shown in Fig. 1. The output curve in the $30th$ run of NNTILC and tracking error in iteration domain and are shown in Fig. 2. The simulation result shows that in iteration domain NNTILC performs a faster convergence rate than conventional TILC, and the tracking error converge to zero homogeneously. It is also shown that NNTILC approach can suppress the effect of changes and disturbance in initial condition effectively.

$$\begin{aligned}
&\Delta J = trace(\tilde{W}_{k+1}^T \tilde{W}_{k+1} - \tilde{W}_k^T \tilde{W}_k) \\
&= trace\left[\tilde{W}_k^T \tilde{W}_k - \alpha \tilde{W}_k^T \phi(x_{k-1}(0)) e_k^T - \alpha e_k \phi^T(x_{k-1}(0)) \tilde{W}_k + \alpha^2 e_k \phi^T(x_{k-1}(0))\phi(x_{k-1}(0)) e_k^T - \tilde{W}_k^T \tilde{W}_k\right]
\end{aligned} \tag{17}$$

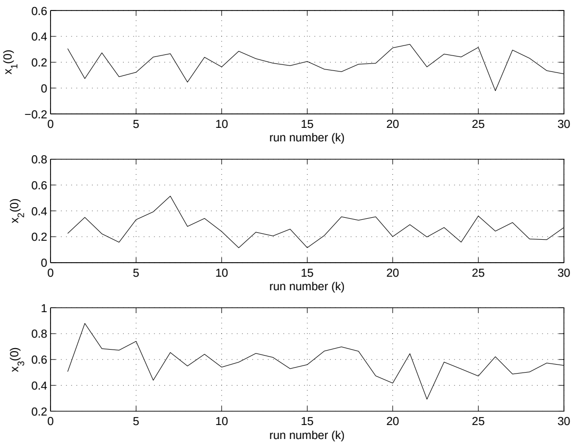

Fig. 1. Initial condition in different run

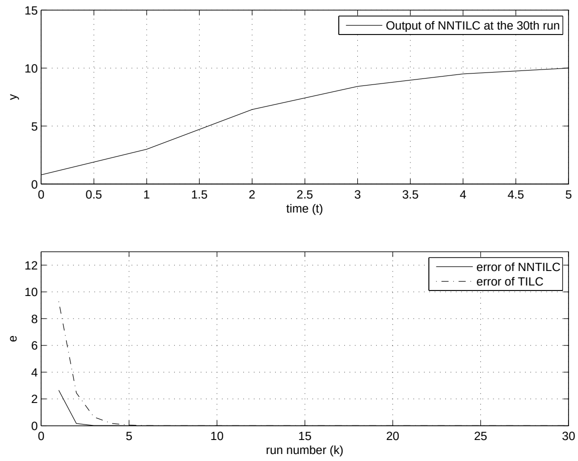

Fig. 2. Performance for Tracking run-invariant terminal reference

Case 2: Tracking Run-varying Terminal Reference Point

In this simulation, the NNTILC approach and conventional TILC method are used for tracking run-varying reference points as shown in Fig. 3. Random noise as shown in Fig. 1 is also adopted here. The controller works with $\alpha = 0.65$ and $l = 1.5$. Fig. 3 illustrates the terminal output curve and tracking error in iteration domain. The figure shows the superior performance of NNTILC in tracking run-varying reference point. As we can see, the tracking error of NNTILC converge to zero very fast within 5 runs and can track it quite precisely afterwards, however, the tracking error of conventional TILC performs period property without converging. On the other hand, the convergence rate of NNTILC in this case is also better than conventional TILC.

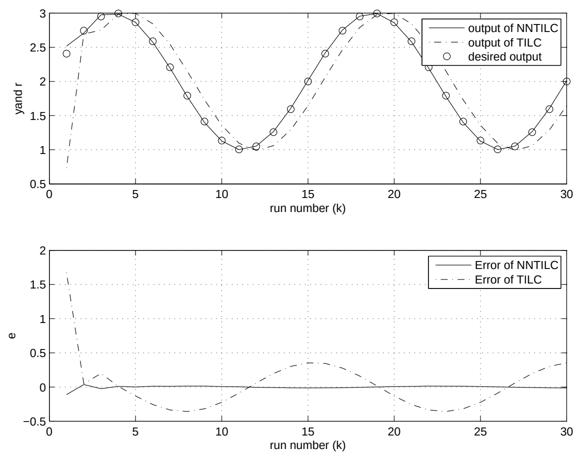

Fig. 3. Performance for Tracking run-varying reference

B. Simulation on MIMO system

Next, consider a MIMO system, which can be represented by

$$x_k(t+1) = \begin{pmatrix} 0.5 & 0.035 & 0.025 \\ 0.255 & 0.6 & -0.99 \\ 0.75 & 0.03 & 0.025 \end{pmatrix} x_k(t) + \begin{pmatrix} 0.2 & 0.03 & 0.025 \\ 0.2 & 0.2 & 0 \end{pmatrix}^T u_k(t) \quad (27)$$

$$y_k(t) = \begin{pmatrix} 1.0 & 0 & 1.0 \\ 0 & 0 & 1.0 \end{pmatrix}^T x_k(t) \quad (28)$$

the system also operates on time interval $[0, 5]s$ for every run.

In this simulation, the NNTILC method and the conventional TILC method are used into the above MIMO system for tracking run-varying reference points as shown in Fig. 5. Random noise as shown in Fig.4 is added to the initial condition. The tracking error of both NNTILC method and conventional TILC method are given in Fig. 6. The controllers work with $\alpha = 0.6$ and $l = 0.8$. The simulation results show that our proposed NNTILC method can be successfully used into MIMO systems. As shown in Fig. 6, for each output, the NNTILC converge quite fast. Whereas the error of conventional TILC method is much larger than NNTILC method.

V. CONCLUSION

For discrete time linear systems with run-varying initial state and reference point, a new neural network based terminal iterative learning control (NNTILC) method is proposed. This method uses a RBF neural network to estimate the effect of initial state on the terminal output and learn the change in initial state iteratively. By involving this information into the control scheme, NNTILC can converge very fast and track

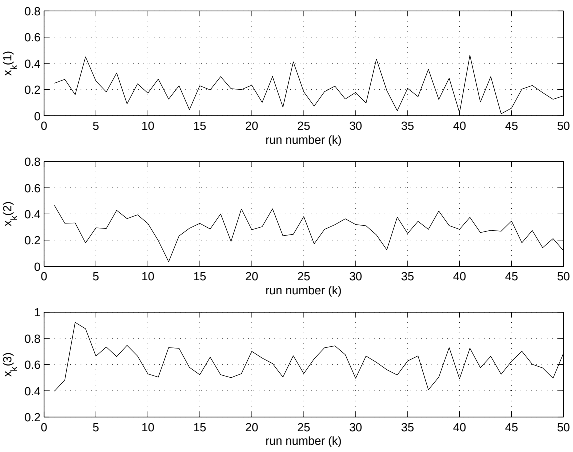

Fig. 4. Random noise added into initial condition

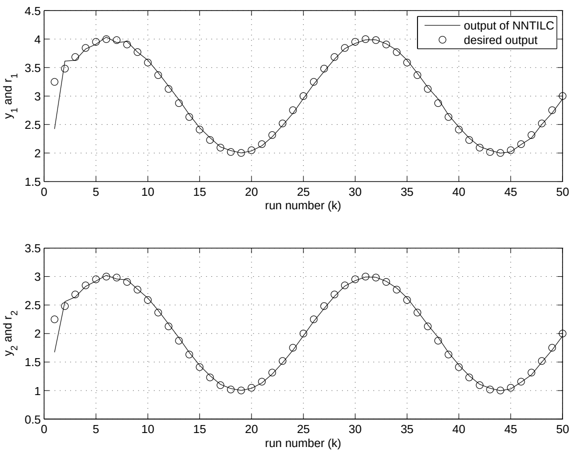

Fig. 5. Performance for Tracking run-varying reference in MIMO system

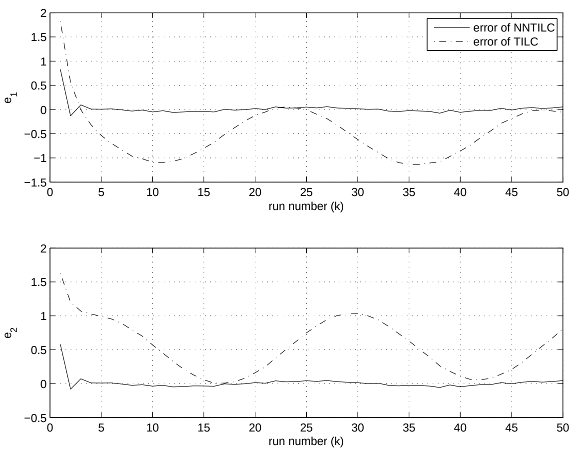

Fig. 6. Tracking error for MIMO system

run-varying reference point precisely beyond initial noise and reference change. The convergence of the method is derived, and the simulation results confirm the effectiveness of the proposed NNTILC method further.

REFERENCES

[1] S. Arimoto, S. Kawamura, and F. Miyazaki, “Bettering operation of robots by learning,” *Journal of robotic systems*, vol. 1, no. 2, pp. 123–140, 1984.

[2] M. Norrlof, “An adaptive iterative learning control algorithm with experiments on an industrial robot,” *Robotics and Automation, IEEE Transactions on*, vol. 18, no. 2, pp. 245–251, 2002.

[3] D. De Roover and O. Bosgra, “Synthesis of robust multivariable iterative learning controllers with application to a wafer stage motion system,” *International Journal of Control*, vol. 73, no. 10, pp. 968–979, 2000.

[4] W. Hoffmann, K. Peterson, and A. Stefanopoulou, “Iterative learning control for soft landing of electromechanical valve actuator in camless engines,” *Control Systems Technology, IEEE Transactions on*, vol. 11, no. 2, pp. 174–184, 2003.

[5] D. Bristow, M. Tharayil, and A. Alleyne, “A survey of iterative learning control,” *Control Systems, IEEE*, vol. 26, no. 3, pp. 96–114, 2006.

[6] D. Wang, “Convergence and robustness of discrete time nonlinear systems with iterative learning control,” *Automatica*, vol. 34, no. 11, pp. 1445–1448, 1998.

[7] G. Casalino and G. Bartolini, “A learning procedure for the control of movements of robotic manipulators,” in *IASTED symposium on robotics and automation*, 1984, pp. 108–111.

[8] Y. Chen, J. Xu, and C. Wen, “A high-order terminal iterative learning control scheme [rtp-cvd application],” in *Decision and Control, 1997., Proceedings of the 36th IEEE Conference on*, vol. 4. IEEE, 1997, pp. 3771–3772.

[9] Z. Xiong and J. Zhang, “Batch-to-batch optimal control of nonlinear batch processes based on incrementally updated models,” in *Control Theory and Applications, IEE Proceedings-*, vol. 151, no. 2. IET, 2004, pp. 158–165.

[10] J. Flores-Cerrillo and J. MacGregor, “Iterative learning control for final batch product quality using partial least squares models,” *Industrial & engineering chemistry research*, vol. 44, no. 24, pp. 9146–9155, 2005.

[11] R. Longman, “Iterative learning control and repetitive control for engineering practice,” *International Journal of Control*, vol. 73, no. 10, pp. 930–954, 2000.

[12] J. Xu and R. Yan, “On initial conditions in iterative learning control,” *Automatic Control, IEEE Transactions on*, vol. 50, no. 9, pp. 1349–1354, 2005.

[13] Y. Chen, C. Wen, Z. Gong, and M. Sun, “An iterative learning controller with initial state learning,” *Automatic Control, IEEE Transactions on*, vol. 44, no. 2, pp. 371–376, 1999.

[14] B. Zhang, D. Wang, Y. Ye, Y. Wang, and K. Zhou, “Multirate iterative learning control schemes,” in *Control, Automation, Robotics and Vision, 2008. ICARCV 2008. 10th International Conference on*. IEEE, 2008, pp. 769–774.

[15] B. Zhang, D. Wang, and Y. Ye, “Cutoff-frequency phase-in iterative learning control,” *Control Systems Technology, IEEE Transactions on*, vol. 17, no. 3, pp. 681–687, 2009.

[16] J. Sjöberg, Q. Zhang, L. Ljung, A. Benveniste, B. Delyon, P. Glorennec, H. Hjalmarsson, and A. Juditsky, “Nonlinear black-box modeling in system identification: a unified overview,” *Automatica*, vol. 31, no. 12, pp. 1691–1724, 1995.