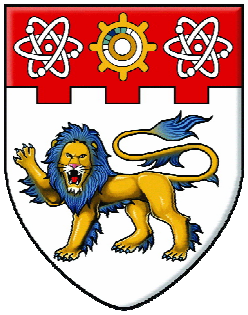


NANYANG TECHNOLOGICAL UNIVERSITY

ON PARALLEL STRUCTURE RC FOR BETTER PERFORMANCE AND NN BASED TILC FOR A CLASS OF NONLINEAR SYSTEMS

LIU TIANQI

SCHOOL OF ELECTRICAL & ELECTRONIC ENGINEERING

2016

ON PARALLEL STRUCTURE RC FOR BETTER PERFORMANCE AND NN BASED TILC FOR A CLASS OF NONLINEAR SYSTEMS

LIU TIANQI

2016

ON PARALLEL STRUCTURE RC FOR BETTER PERFORMANCE AND NN BASED TILC FOR A CLASS OF NONLINEAR SYSTEMS

LIU TIANQI

LIU TIANQI

School of Electrical & Electronic Engineering

A thesis submitted to the Nanyang Technological University
In partial fulfillment of the requirements for the degree of
Doctor of Philosophy

2016

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research and has not been submitted for a higher degree to any other University or Institution.

Date	LIU TIANQI

Acknowledgement

I would like to express my sincere appreciation to my supervisor, Professor Wang Danwei, for his invaluable guidance, support and suggestions. I like to thank Prof. Chi Ronghu and Prof. Zhou Keliang for the discussion on research and for their kind help on experiments. I also want to thank my colleagues in the Intelligent Robotics Lab of NTU, for their generous help. Last but not least, I want to thank my family, especially my parents and HB, for their constant love and encouragement.

Abstract

Repetitive control and iterative learning control are both learning control schemes which update and refine control sequence using tracking errors in past trials. With the superior ability to handle system uncertainty and repetitive disturbance, repetitive control and iterative learning control have been applied to industry successfully in the past decades. The basic ideas of these two control schemes are similar to each other, but their analysis methods and applications are quite different.This thesis is divided into two parts , each of which focuses on the developments of repetitive control and iterative learning control, respectively.

In PART I, a new parallel structure repetitive control scheme is proposed. This new repetitive controller realizes selective compensation on the targeted harmonics. Compared with the conventional methods, it achieves better tracking performance with less delay and computational burden. Furthermore, correction factors which modify the poles of conventional repetitive control in fractional cases are introduced. As a result, robust response is achieved in fractional cases where sampling frequency is not integer multiple of fundamental frequency. As an application, a high-performance grid simulator controlled by parallel structure fractional repetitive controller is designed and implemented. Experimental results testify the effectiveness of the proposed control scheme.

In PART II, combination of neural network and iterative learning control is exploited to address system uncertainties and zero-error initial condition in nonlinear

non-affine iterative learning control systems. In this new control scheme, a radial basis function neural network is adopted to estimate the effect of initial state on terminal output. With this estimation included in the control law, the proposed control method can drive nonlinear non-affine systems to track run-varying reference point in the presence of initial state variance. Stability and convergence of this method are proved and simulation results are provided to testify its effectiveness.

Contents

List of Figures

List of Tables

Symbols and Acronyms

f_0	Fundamental frequency.
ω_0	Fundamental angular frequency.
f_s	Sampling frequency.
T_0	Fundamental period.
T_s	Sample period.
$N = \frac{f_s}{f_0}$	Ratio between sampling frequency and fundamental frequency.
N^*	$N^* = round(N)$.
k_i	Control gain in different branches.
k_r	Control gain of repetitive controller.
$R(z)$	Reference.
$y(z)$	Output.
$d(z)$	Disturbance.
$u_r(z)$	Control signal from repetitive controller.
$u(z)$	Overall control signal.
δ	Correction factor.

E	DC bus voltage.
L_f	Output filter inductance.
C_f	Output filter capacitance.
R	Resistor load.
L_r, C_r, R_r	Resistance, inductance, capacitance of rectifier load.
i_0	Output current.
v_0	Output voltage.
v_{ref}	Reference voltage.
$I_{a,b,c}$	Three-phase current.
$u_{ab,bc,ca}$	Three-phase voltage.
$v_{\alpha,\beta}$	Three-phase voltages in $\alpha\beta$ frame.
ϕ, ξ	Voltage drop factor.
V_h	Amplitude of the harmonic components.
$x_k(t) \in \mathbb{R}^p$	State vector.
$y_k(t) \in \mathbb{R}^n$	Output vector.
$u_k(t) \in \mathbb{R}^m$	Control vector.
u_k^D	Ideal input in the kth iteration.
$x_k(0)$	Initial state vector.
$\Gamma \in \mathbb{R}^{m \times n}$	Self-define control gain matrix.
θ_k	System parameter in the kth iteration.

θ_k^D	Ideal value of θ_k.
$\hat{\theta}_k^D$	Estimation of θ_k^D.
$\tilde{\theta}_k^D$	Estimation error of θ_k^D.
$g(\cdot)$	Ideal nonlinear function mapping initial state and reference to system parameters.
$\hat{g}_k(\cdot)$	The approximation of the ideal function $g(\cdot)$.
$W_D^T \in \mathbb{R}^{n\times L}$	An unknown ideal weight matrix.
$\hat{W}_k^T \in \mathbb{R}^{n\times L}$	Estimation of the ideal weight matrix.
$\tilde{W}_k^T \in \mathbb{R}^{n\times L}$	Estimation error of the ideal weight matrix.
L	The number of neurons in the hidden layer.
$\phi(\cdot)$	A known vector of basis activation function.
$\nu(k)$	Modelling noise.
$\mu_i \in \mathbb{R}^{p+n}$	Centre of the ith hidden neuron.
σ_i	Width of the ith hidden neuron.
$e_a(k)$	A priori error.
J	Lyapunov energy function.
η	Control constant in NNTILC.
α	Learning gain in NNTILC.
RC	Repetitive Control.
CRC	Conventional Repetitive Control.

ORC	Odd-harmonic Repetitive Control.
ILC	Iterative Learning Control.
TILC	Terminal Iterative Learning Control.
CTILC	Conventional Terminal Iterative Learning Control.
PWM	Pulse Width Modulation.
SPWM	Sinusoidal Pulse Width Modulation.
IMP	Internal Model Principle.
SSSRC	Serial Structure Selective Repetitive Control.
PSFRC	Parallel Structure Fractional Repetitive Control.
NNTILC	Neural Network based Terminal Iterative Learning Control.
RBF	Radial Basis Function.
THD	Total Harmonic Distortion.
SRC	Spatial Repetitive Control.
SFB	State Feed Back.
RMS	Root Mean Square.
PQ	Power Quality.
PI	Proportional-Integral.
WR	Waveform Relaxation.
IGBT	Insulated Gate Bipolar Transistor.
SISO	Single Input Single Output.

MIMO Multiple Input Multiple Output.

AS Asymptotically Stable.

Chapter 1

Introduction

Repetitive control (RC), iterative learning control (ILC) and terminal iterative learning control (TILC) are all learning control schemes which refine and update control sequence using errors from previous trials. They take fully advantage of the repetitive feature and achieve outstanding control performance.

RC is applied to stable periodic systems without the requirement of identical initial state. While ILC is applied to a system with repeated operations and requires identical starting point to ensure convergence. When the initial states are not identical, the tracking performance is affected.

This chapter is constructed as follows. Section 1.1 gives introduction on RC and ILC. Literature review is provided in section 1.2. Section 1.3 discusses the motivation of this research work. The contributions of this thesis are stated in section 1.4. Section 1.5 describes the organization of the thesis.

1.1 Basics of repetitive control and iterative learning control

1.1.1 Repetitive control

Repetitive control is proposed based on the internal model principle (IMP) [1]. It can be applied to closed-loop systems directly in plug-in style. Structure of a conventional RC controller is shown in Fig. 1.1, where $e(z)$ is the tracking error, k_r is the learning gain, $u_r(z)$ is the control signal, $G_{rc}(z)$ denotes the transfer function of RC, $N = \frac{f_s}{f_0}$ with f_s being the sampling rate and f_0 being the fundamental frequency.

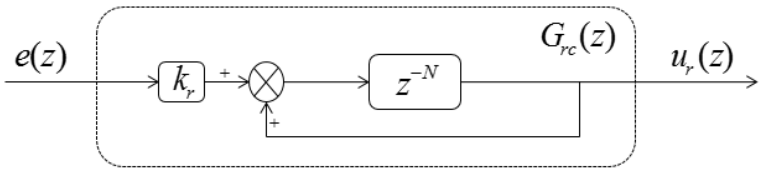


Figure 1.1: Conventional RC controller

From Fig. 1.1, the transfer function of the conventional RC can be derived as

$$G_{rc}(z) = \frac{u_r(z)}{e(z)} = \frac{k_r z^{-N}}{1 - z^{-N}}. \tag{1.1}$$

It can be calculated from (1.1) that RC controller locates poles at $j\omega = kj\omega_0$, where $\omega_0 = 2\pi f_0$ and $k = 0, 1, 2....$ From the view of IMP, these poles of RC guarantee zero-error tracking performance in repetitive operational systems.

Fig. 1.1 shows an ideal RC controller, but high frequency components may lead the system to instability due to the infinite poles located on the unit circle. Therefore, filters are adopted after the delay units to achieve better robustness in real

applications. A typical closed-loop RC system is shown in Fig. 1.2, where $R(z)$ is the reference, $G_c(z)$ is a state-feedback controller, $G_p(z)$ is the control plant, $u(z)$ is the final control signal, $d(z)$ is disturbance, $y(z)$ is the output, $Q(z)$ and $G_f(z)$ are both filters.

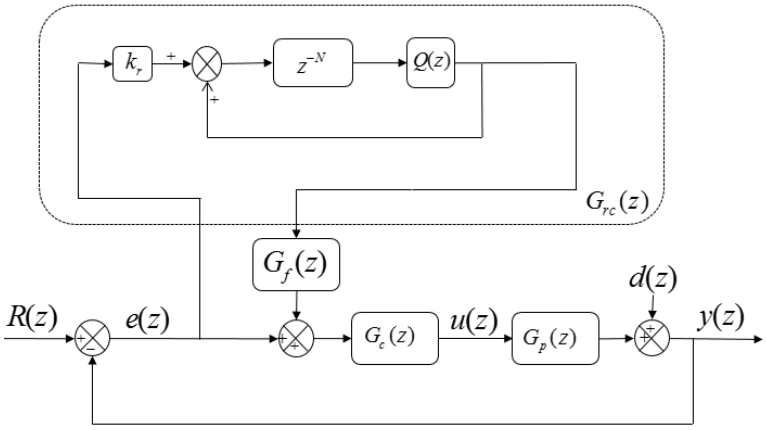


Figure 1.2: Typical closed-loop RC system

As shown in Fig. 1.2, RC controller is plugged into a closed-loop feedback system. The transfer function of the closed-loop system without RC can be calculated as

$$G(z) = \frac{G_c(z)G_p(z)}{1 + G_c(z)G_p(z)} \tag{1.2}$$

Derived from Fig. 1.2, along with equation (1.2), the transfer function from $R(z)$ to $y(z)$ with RC is:

$$\begin{aligned}
\frac{y(z)}{R(z)} &= \frac{\left[1+k_rG_f(z)\frac{Q(z)z^{-N}}{1-Q(z)z^{-N}}\right]G_c(z)G_p(z)}{1+\left[1+k_rG_f(z)\frac{Q(z)z^{-N}}{1-Q(z)z^{-N}}\right]G_c(z)G_p(z)} \\
&= \frac{\left(1-Q(z)z^{-N}+k_rG_f(z)Q(z)z^{-N}\right)G_c(z)G_p(z)}{1-Q(z)z^{-N}+\left(1-Q(z)z^{-N}+k_rG_f(z)Q(z)z^{-N}\right)G_c(z)G_p(z)} \\
&= \frac{G_c(z)G_p(z)-\left[1-k_rG_f(z)\right]Q(z)z^{-N}G_c(z)G_p(z)}{1+G_c(z)G_p(z)-\left[1+G_c(z)G_p(z)-k_rG_f(z)G_c(z)G_p(z)\right]Q(z)z^{-N}} \\
&= \frac{\frac{G_c(z)G_p(z)}{1+G_c(z)G_p(z)}-\left[1-k_rG_f(z)\right]Q(z)z^{-N}\frac{G_c(z)G_p(z)}{1+G_c(z)G_p(z)}}{1-\left[1-k_rG_f(z)\frac{G_c(z)G_p(z)}{1+G_c(z)G_p(z)}\right]Q(z)z^{-N}} \\
&= \frac{\left[1-Q(z)z^{-N}\left[1-k_rG_f(z)\right]\right]G(z)}{1-Q(z)z^{-N}\left[1-k_rG_f(z)G(z)\right]}
\end{aligned} \tag{1.3}$$

, and the transfer function from $d(z)$ to $y(z)$ can be derived by the same way as

$$\frac{y(z)}{d(z)} = \frac{1-Q(z)z^{-N}}{1+G_c(z)G_p(z)}\frac{1}{1-Q(z)z^{-N}[1-k_rG_f(z)G(z)]} \tag{1.4}$$

From (1.3) and (1.4), error transfer function of the overall system can be derived as

$$\frac{e(z)}{(R(z)-d(z))} = \frac{(1-Q(z)z^{-N})(1-G(z))}{1-Q(z)z^{-N}(1-k_rG_f(z)G(z))} \tag{1.5}$$

Then stability criterion for systems in Fig. 1.2 can be derived from (1.5) [2]:

(1) The closed-loop feedback system $G(z)$ is stable, i.e., all the poles of $G(z)$ are located inside the unit circle;

(2) The following inequality holds:

$$\mid Q(z)(1-k_rG_f(z)G(z)) \mid \leq 1. \tag{1.6}$$

1.1.2 Iterative learning control

In ILC systems, analysis in iteration domain is much more convenient than that in time domain or frequency domain. A typical ILC system in iteration domain can be described by Fig. 1.3 [3].

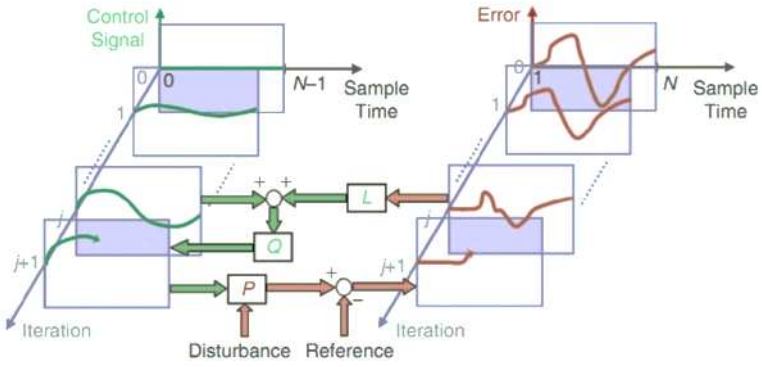


Figure 1.3: Two-dimensional first-order ILC system

Consider a discrete time-invariant linear system as follows,

$$\begin{aligned} \boldsymbol{x}_k(t+1) &= A\boldsymbol{x}_k(t) + B\boldsymbol{u}_k(t) \\ \boldsymbol{y}_k(t+1) &= C\boldsymbol{x}_k(t+1) \end{aligned} \tag{1.7}$$

where $t = 0, 1, 2, ..., N$ is the sampling index and k is the iteration index, matrices A, B and C are time invariant with appropriate dimensions; $\boldsymbol{x}_k(t) \in \mathbb{R}^p$ is the state vector, $\boldsymbol{y}_k(t) \in \mathbb{R}^n$ is the output vector, and $\boldsymbol{u}_k(t) \in \mathbb{R}^m$ is the control vector at the tth sampling time in the kth iteration.

Expand (1.7) as Markov parameter form and put it into matrix, the system becomes

$$\begin{bmatrix} y_k(1) \\ \vdots \\ y_k(N) \end{bmatrix} = \begin{bmatrix} p_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ p_N & \cdots & p_1 \end{bmatrix} \begin{bmatrix} u_k(1) \\ \vdots \\ u_k(N) \end{bmatrix} + \begin{bmatrix} d(1) \\ \vdots \\ d(N) \end{bmatrix} \tag{1.8}$$

or in a simplified form as

$$y_k = Pu_k + d \tag{1.9}$$

where $p_n = CA^{n-1}B$ and $d(n) = CA^n x(0)$.

Define error vector E_k as

$$\begin{bmatrix} e_k(1) \\ e_k(2) \\ \vdots \\ e_k(N) \end{bmatrix} = \begin{bmatrix} y_d(1) \\ y_d(2) \\ \vdots \\ y_d(N) \end{bmatrix} - \begin{bmatrix} y_k(1) \\ y_k(2) \\ \vdots \\ y_k(N) \end{bmatrix} \tag{1.10}$$

From Fig.1.3, ILC controller updates control sequence using error vector in previous iterations. Let u_{k+1} and u_k be the control vector in two adjacent iterations. Then a typical iterative P-type learning control law can be constructed for system (1.9) as

$$\begin{bmatrix} u_{k+1}(0) \\ u_{k+1}(1) \\ \vdots \\ u_{k+1}(N-1) \end{bmatrix} = \begin{bmatrix} q_0 & q_{-1} & \cdots & q_{-(N-1)} \\ q_1 & q_0 & \cdots & q_{-(N-2)} \\ \vdots & \vdots & \ddots & \vdots \\ q_{N-1} & q_{N-2} & \cdots & q_0 \end{bmatrix} \left(\begin{bmatrix} u_k(0) \\ u_k(1) \\ \vdots \\ u_k(N-1) \end{bmatrix} + \begin{bmatrix} l_0 & l_{-1} & \cdots & l_{-(N-1)} \\ l_1 & l_0 & \cdots & l_{-(N-2)} \\ \vdots & \vdots & \ddots & \vdots \\ l_{N-1} & l_{N-2} & \cdots & l_0 \end{bmatrix} \begin{bmatrix} e_k(1) \\ e_k(2) \\ \vdots \\ e_k(N) \end{bmatrix} \right) \tag{1.11}$$

or in a simplified form as

$$U_{k+1} = Q(U_k + LE_k) \tag{1.12}$$

where the Q and L are defined as the Q-filter and learning function, respectively.

Combining (1.9), (1.10) and (1.12) yields

$$u_{k+1} = Q(I - LP)u_k + QL(y_d - d) \tag{1.13}$$

To guarantee that (1.13) is asymptotically stable (AS), all the eigenvalues of $Q(I - LP)$ should be within the unit cycle, such that

$$\rho\{Q(I - LP)\} < 1 \tag{1.14}$$

It should be noted that when the Q-filter and the learning function are causal, the matrix $Q(I - LP)$ is lower triangular and Toeplitz. As a result, the repeated eigenvalue will be $\lambda = q_0(1 - l_0 p_1)$.

In this case, (1.14) is equivalent to the following scalar condition

$$|q_0(1 - l_0 p_1)| < 1 \tag{1.15}$$

which is equivalent to

$$|q_0(1 - l_0 CB)| < 1 \tag{1.16}$$

1.1.3 Relationship between RC and ILC

Although analysis methods for RC and ILC are different, the underlying ideas of these two control methods are the same. The relationship between RC and ILC is discussed in this section.

The transfer function of a typical repetitive controller has been derived in section 1.1.1 as follows:

$$G_{rc}(z) = \frac{u_r(z)}{e(z)} = \frac{k_r z^{-N}}{1 - z^{-N}}. \tag{1.17}$$

Rearranging the transfer function in (1.17) yields

$$u_r(z) = u_r(z)z^{-N} + k_r e(z)z^{-N} \tag{1.18}$$

In time domain, the updating law in (1.18) is

$$u(kT_s) = u(kT_s - NT_s) + k_r e(kT_s - NT_s) \tag{1.19}$$

where $T_s = 1/f_s$ is the sample period, NT_s is the period of the disturbance and $k = 0, 1, 2....$

The time domain expression of RC controllers illustrates that RC controllers refine the control sequence $u(kT_s)$ using the tracking error of last period $e(kT_s - NT_s)$ by P-type learning. If the time axis is divided into several identical time interval NT_s and denote every time interval as one iteration, the control law can be rewritten to

$$u_j = u_{j-1} + k_r e_{j-1} \tag{1.20}$$

where $j = 0, 1, 2...$ is the iteration index, u_j, u_{j-1} and e_{j-1} are all vectors contain N elements.

Equation (1.20) represents a P-type iterative learning control law. Therefore, RC and ILC share the same basic principle but are applied to different applications. Detailed comparison between repetitive control and iterative learning control can be found in [4].

1.2 Literature Review

1.2.1 Repetitive control

Repetitive control is firstly developed in [5] for periodic disturbance suppression. The stability condition of RC system was derived in [6] using small gain theorem, and it was shown that the stability could be guaranteed only when the plants are proper but not strictly proper. To improve the robustness of repetitive control, [7] modified repetitive controller introduced a low-pass filter in series with the time-delay units. Modified repetitive controller is mainly analyzed in frequency domain and [5,8–10] provide some general design rules. Other synthesis procedures, such as the state-space approach, factorization approach, and H_∞ optimal design approach, can be found in [11–15]. For systems with uncertain parameters, [16,17] proposed a graphical frequency domain robust design method in interval plants.

The direction of discrete-time RC is set in [18]. Some successful design method for discrete time repetitive control systems can be found in [19–22]. In discrete time repetitive controller, the sampling rate should be chosen high to avoid inter-sample ripple. But if it is too large, the order of the time-delay model will increase. Prototype repetitive controller (PRC) based on zero phase error tracking controller is proposed to address this problem [18]. Based on PRC, [23,24] replace the the control gain with a real proper function to handle unstable zeros and obtain faster convergence. In [25], PRC is extended to MIMO systems. Disturbance observer and adaptive law are adopted in [26–29].

Digital repetitive controller requires large number of delay units and memory cells. These delay units increase the computational burden and slow down repetitive controller inevitably. Many faster repetitive controllers are proposed in the past decades to improve transient performance. One of the solutions is selective repetitive control. This type of repetitive control is mostly developed for single-phase or three-

phase AC systems, because the total harmonic distortion (THD) of such systems are dominated by specific harmonic groups. For single-phase systems, odd harmonic repetitive control schemes [30] are proposed. Investigations have shown that odd-harmonic RC is twice faster than conventional RC. Following this idea, $6k \pm 1$ repetitive control is developed for three-phase AC systems [31–34]. The realization of $6k \pm 1$ repetitive control is usually done in dual parallel synchronous frame (positive and negative). Besides, parallel structure repetitive control [35] separates repetitive controller into several parallel branches and eliminates those unnecessary branches. As a result, the compensation and tuning for different harmonic groups become flexible and selective [36,37].

Moreover, performance of digital repetitive controllers is sensitive to changes of the system's fundamental frequency. So fractional and adaptive digital repetitive controller is required especially when the period of fundamental reference or disturbance cannot be divided by sampling period completely. FIR filter based repetitive control is one of the most straightforward ways to address this issue [38–45]. The basic idea is to approximate fractional order of delay using filters. Spatial repetitive control and multi-rate repetitive control provide other approaches to solve this problem [46–49]. These kinds of repetitive controllers are multiple-sampling-rate controller, in which, the sample rate of RC is adaptive to the changes of fundamental frequency. In these systems, the number of control actions in one cycle is fixed for different fundamental frequencies. Position-domain modelling and analysis of the plant are required. High-order repetitive control is another scheme to improve the robustness of RC in fractional cases [50–55].

Repetitive controllers was motivated by a power supply regulation problem in proton synchrotron accelerator and has been widely applied to various other fields [56–59]. Recently a variety of successful applications of repetitive control have been investigated, including harmonic distortion elimination in PWM inverters or UPS

[60–68], current compensation in active filters [69–73], speed control of motors [74–81], friction compensation [82–88], torque vibration compensation [76,89,90], current control of photovoltaic generation systems [91–96], position control of piezoelectric actuators [97–100] and so on.

1.2.2 Iterative learning control

Iterative learning control was firstly proposed by Uchiyama [101] for repetitive operational systems. It achieves outstanding tracking performance for repetitive reference trajectory [102]. Besides, system uncertainty and repetitive disturbance can be handled by ILC [103–105]. In [106–108], the equivalence between ILC and feedback control is discussed. It is proved that ILC finds the best feedforward gain in time domain and the best feedback controller in iteration domain. In [109], uniform convergence of ILC is proved by Arimoto and Naniwa. Discussions on convergence and robustness of ILC can also be found in [110,111].

ILC updates and refines the control sequence using errors in previous trials by various updating rules, among which PID-type ILC is the most widely used scheme [112–116]. Besides, fractional ILC method is also investigated in [117–120]. In [121–129], adaptive and self-tuning ILC are reported. [130] updates ILC in Hilbert space. The combination of ILC with intelligent algorithm is investigated in [131–138].

In some special cases such as the control of a oven's temperature for wafer deposition thickness, it is usual that only the end-point error is measurable. In order to apply ILC to such cases, Chen et al. [139] developed a terminal iterative learning control (TILC) method, which adjusts the set point of repetitive system based only on the terminal errors in previous trials. This method focuses sorely on tracking the desired terminal reference point and has been widely applied to different fields such as the control of thermoforming ovens [140], mobile robots [141], station stop control of a train [142], multi-agent systems [143] and so on. Investigation-

s [144,145] have shown that TILC could achieve convergence in iteration domain. In addition, [146,147] also investigate the TILC for tracking multiple intermediate pass points.

In both ILC and TILC, repetition of the initial state should be satisfied. In previous works, some solutions to the problem of zero error initial condition have been proposed as follows. 1) ILC with initial state learning scheme [148], in which the controller learns the initial state firstly to make it identical in every iteration; 2) Multi-rate ILC schemes [149], in which the input updating rate is different from the sampling rate of the feedback system or the input updating rates at low and high frequency bands are different; 3) Cut-off frequency phase-in profile [150], in which the cut-off frequency of the filter is time-varying and follows a predefined profile. [151–159] also provide solutions to initial shift problem.

Besides, [160,161] and [162] proposed a class of controllers for repetitive operational systems to track iteration-varying trajectory. All these controllers are based on zero error initial condition. In [163], adaptive ILC was introduced to track iteration-varying reference beyond initial state variance. To further improve the robustness of ILC against uncertainty and disturbance, stochastic ILC [164], H_∞ approach [165], model-based method [166], backstepping based ILC [167], optimal ILC [168–172] are proposed. Till then, ILC and TILC are never limited to systems with repetitive initial state and reference trajectory.

ILC has found its applications successfully in repetitive industrial processes. The application of ILC in general robotic such as rigid and flexible manipulators can be found in [173–181]. Besides, ILC is applied to under water robot [182–184], acrobat robots [185,186], mobile robots [187], gantry robot [188,189] and so on. ILC is also widely adopted in the control of permanent magnet synchronous motors, linear motors, induction motors and AC servo motors [190–197]. Examples of ILC applications to chemical and biomedical applications can be found in [198–209].

1.3 Motivation

Repetitive control and iterative learning control schemes have been applied to industrial applications successfully. However, some underlying assumptions limit their application. Based on the existing challenges in RC and ILC, this section introduces the motivation of this research work.

1.3.1 Repetitive control

Repetitive control is widely used in the control of power systems due to its outstanding performance on high-precise tracking and harmonic elimination. But tracking performance of RC is quite sensitive to the change of fundamental frequency. The lack of adaptiveness limits the application of repetitive control. For example, in renewable energy systems, gird connected converters are required to generate high quality output voltages adaptive to the grid changes. Besides, because digital repetitive controller assumes that the ratio between sampling frequency and fundamental frequency ($N = f_s/f_0$) is an integer, conventional digital repetitive controller cannot be implemented when the sampling frequency cannot be divided by fundamental frequency without remainder. Therefore, it is important to develop adaptive and fractional repetitive controller to overcome these limitations.

Moreover, conventional repetitive controllers compensate all harmonics of the system, thus introducing a large number of delay units. But this is reductant in some special cases, because not all the harmonics exist in particular systems. For example, the THD of AC systems are dominated by odd harmonics only. In these cases, selective or reduced repetitive controller is enough for static tracking performance. By doing so, the number of delay units required can be decreased largely and transient response can also be improved. Since repetitive controllers are actually periodic signal generators which combine infinite resonant controllers in parallel,

it is straightforward to rearrange these resonant controllers into different parallel branches and tune them independently.

1.3.2 Iterative learning control

Conventional iterative learning control assumes that initial state in each iteration should be identical. The reference trajectory is also required to be fixed. However, these underlying assumptions limit the application of iterative learning control in real industry. Terminal iterative learning control suffers the same problem in industrial applications. Compared with ILC, TILC aims to track a terminal reference point and the control input is set to be a constant in the whole iteration. So it is easier for TILC to include input-output dynamics of the system in the control scheme thus eliminating the zero-error initial condition. Considering the high uncertainties in some complicated processes, neural network is quite promising to estimate the system's dynamics. Therefore, a Radial Basis Function (RBF) neural network is adopted in this thesis to improve the performance of terminal iterative learning control.

1.4 Contribution

This thesis has developed novel repetitive control and terminal iterative learning control schemes to improve their performance in various applications. Parallel structure repetitive control is proposed for better selectivity and adaptiveness. Besides, terminal iterative leaning control scheme is combined with neural network to expand its application beyond the challenges of zero-error initial condition and tracking iteration-varying references. The main contributions of this thesis are summarized as follows:

1. Performance comparison between repetitive controllers in parallel structure and

serial structure: an improved parallel structure selective repetitive control scheme is proposed for PWM converters to compensate any $nk \pm m$th order harmonics selectively. This new control scheme divides conventional repetitive control into two branches in parallel. This parallel structure avoids connecting filters in series and achieves higher gains at fundamental and targeted harmonic frequencies. Meanwhile, requirement on memory and computational time are reduced. Stability and convergence of this approach are proved mathematically. Experiments on a PWM converter system verify the effectiveness of the proposed control scheme.

2. Development of a novel parallel structure fractional repetitive control (PSFRC) scheme: a parallel structure fractional repetitive control scheme is proposed to improve the performance of PWM inverters in fractional cases where sampling rate of the digital control system is not an integer multiple of the fundamental frequency. By introducing a correction factor, the new control scheme increases the control gains for all harmonics and locates poles accurately at targeted harmonic frequencies. Besides, parallel structure fractional repetitive controller requires less data memory. Dynamic response is also improved. Stability and convergence of this method are proved. Experimental results on a single phase PWM inverter illustrate the advantages of this control scheme.

3. Design and implementation of a high-performance grid simulator using parallel structure fractional repetitive control: a high-performance grid simulator is proposed based on parallel structure fractional repetitive control scheme to simulate various distortions and perturbations in real power distribution systems. PSFRC realizes the independent tunings and compensations for different harmonic groups. As a result, this grid simulator is capable to add selective harmonics to the main output voltages accurately and eliminate unwanted harmonic components. Moreover, faster transient performance is achieved for simulating sudden changes in grid. Design process and stability criteria are presented in details.

4. Development of neural network based terminal iterative learning control for tracking iteration-varying reference with initial error: a neural network based terminal iterative learning control method is proposed for a class of uncertain nonlinear non-affine systems to track iteration-varying reference point with initial state variance. The non-affine terminal dynamics are converted affine in this control scheme. Besides, the unrealisable recurrent network is simplified to realisable static network. By neural network, the effect of initial state and control signal on terminal output can be estimated and included in the control signal. As a result, the proposed control scheme can drive nonlinear non-affine systems to track iteration-varying reference point in the presence of initial state variance. Stability and convergence of this approach are analyzed and simulation results are provided to illustrated the effectiveness of the proposed method.

1.5 Organization of the thesis

The outline of this thesis is shown in Fig. 1.4. The thesis is divided into two parts: part I and part II for repetitive control and iterative learning control, respectively. The remainder of the thesis is organized as follows:

Chapter 2 presents an improved parallel structure selective repetitive control (PSSRC) scheme and compares this parallel-structure scheme with serial-structure RC schemes. An experimental setup is introduced, and a set of experimental results are provided to illustrate the advantages of the proposed approach.

Chapter 3 extends PSSRC to a multi-branches parallel-structure RC. Parallel structure fractional repetitive control (PSFRC) scheme is proposed to overcome the limitations on RC in fractional applications.

Chapter 4 introduces a high-performance grid simulator using parallel structure fractional repetitive control to emulate various operation scenarios of power grids

for testing power products.

Chapter 5 presents a neural network based terminal iterative learning control scheme for linear systems with iteration-varying initial states. A RBF neural network is adopted to approximate the dynamics of the systems.

Chapter 6 extends the NNTILC to nonlinear non-affine systems, and case studies are done on a wafer fabrication system to testify the proposed approach.

Chapter 7 summarizes the contributions of the thesis, and provides the outlines scope for future work.

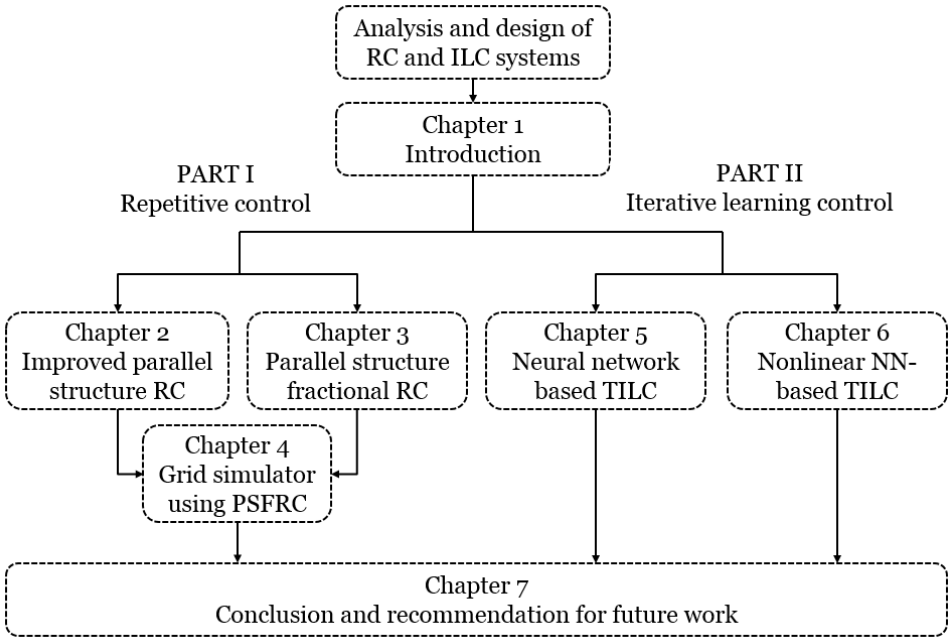


Figure 1.4: Outline of the thesis

Part I

REPETITIVE CONTROL

Chapter 2

Improved Parallel Structure Selective Repetitive Control

2.1 Introduction

Repetitive control is an effective control scheme for repetitive operation systems to suppress periodic disturbance. Conventional repetitive control (CRC) uses an internal signal generator consisting of N delay units (N is the number of samples in one period of the repetitive signal) in a plug-in style. Such signal generator can locate infinite poles at each harmonic frequency of the system and derive infinitely large gains there. Based on the internal model principle (IMP) [1], RC can achieve zero-error tracking and harmonic rejection for any signals whose frequencies equal to integer multiple of the fundamental frequency. Recently, RC has been widely applied to different industrial applications such as AC power source, servo system of optical disk drive, three phase active filter, industrial robotic manipulator and so on. Especially in PWM converters, harmonic distortion caused by switches, non-linear loads and parameter uncertainties can be compensated effectively by RC. Therefore, designs and applications of RC in PWM converters have been widely investigated

in the past decades.

However, RC usually requires a large number of delay units and the dynamic response is relatively poor. Several new RCs, such as dual-mode-structure plug-in repetitive controller [35], odd-harmonic digital repetitive controller (ORC) [30,210], synchronous-frame repetitive controller [33], are proposed in previous works to solve this problem. These RC controllers emphasize their compensations on particular dominant harmonics and reduce the gains for non-dominant components. As a result, the number of delay units is reduced , computation burden is decreased and transient performance is improved. This is possible because it is not necessary to compensate all harmonics in PWM converter systems. For example, single phase converters contain only odd harmonics in their Fourier series expansions, and $6k \pm 1th$ order harmonics dominate the THD in three-phase AC systems. Therefore, a selective repetitive controller is preferred to provide optimal trade-off among steady state tracking performance, memory requirement and transient response in different applications.

In [211], a generic $nk \pm m$ RC scheme is proposed and a general structure of RCs compensating different harmonic groups is constructed. Because this control scheme is formulated in a serial structure, it is called serial structure selective repetitive control (SSSRC) here. However, SSSRC needs more delay units than other conventional RC controllers with similar functionalities. For instance, a $4k \pm 1$ SSSRC controller in [211] needs $3N/4$ delay units, while a conventional odd-harmonic RC [210] which also compensates $4k \pm 1$th order harmonics only needs $N/2$.

Besides, it is known that all the poles generated by RC are located on the unit circle, so RC systems are easily affected by high-frequency components and led to instability. In order to improve the robustness of RC systems, filters are introduced after the delay units. But filters can attenuate the control gains of RC from infinity to 50 dB. Because SSSRC connects two groups of delay units with filters in series,

the control gains is attenuated further [211,212].

To address the above problems, an improved parallel structure selective repetitive control (PSSRC) scheme is proposed in this chapter. In this new controller, selective repetitive controller is constructed in parallel. In PSSRC, harmonics to be compensated can be selected at will. Besides, the number of delay units required is reduced to 2/3 of that in SSSRC. Computational burden is also decreased. Moreover, PSSRC adopts parallel structure and avoids connecting filters in series, thus improving control gains at fundamental and harmonic frequencies.

Stability criteria for PSSRC are derived in this chapter. Experimental results of applying a $4k \pm 1$ PSSRC to a single-phase PWM inverter system are presented to illustrate the advantages of the proposed approach.

The remainder of this chapter is organized as follows. In section 2.2, an improved parallel structure selective repetitive controller is proposed and stability criteria are derived; Section 2.3 provides experimental results of a $4k \pm 1$ PSSRC applied to a single-phase inverter; Section 2.4 provides some conclusions.

2.2 Parallel structure selective repetitive control

2.2.1 Serial structure selective repetitive control

Selective repetitive control in previous work [211] is realized in serial structure as shown in Fig. 2.1, where $e(z)$ is the tracking error, k_r is the learning gain, $N = \frac{f_s}{f_0}$ with f_s being the sampling rate and f_0 being the fundamental frequency, $G^s_{RC}(z)$ is the transfer function of the overall controller, m and n are integers which can be arbitrarily assigned to target different $nk \pm m$th order harmonic groups.

The signal flow graph can be drawn according to Fig. 2.1 as in Fig. 2.2.

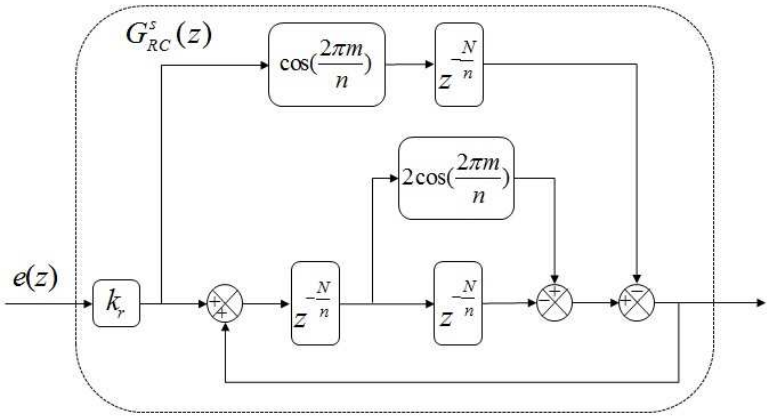


Figure 2.1: Serial structure selective repetitive controller (SSSRC)

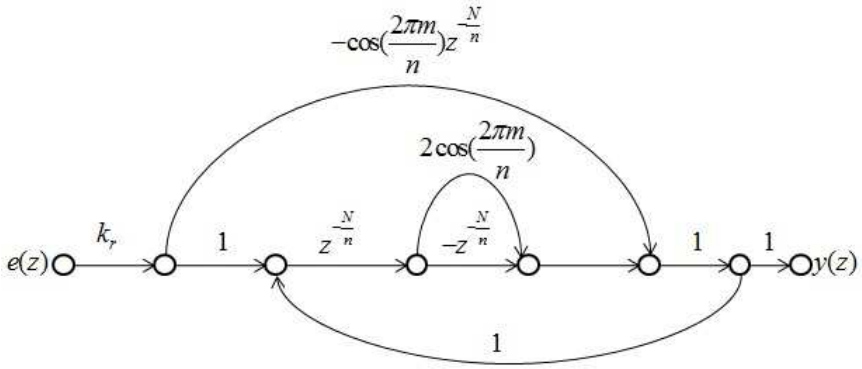


Figure 2.2: Signal flow graph

According to Fig. 2.2, the forward path gains and determinants can be calculated as:

$$\begin{aligned} P_1 &= -k_r \cos(\tfrac{2\pi m}{n}) z^{-\frac{N}{n}} \quad , \quad \Delta_1 = 1 \\ P_2 &= -k_r z^{-\frac{2N}{n}} \quad , \quad \Delta_2 = 1 \\ P_3 &= 2k_r \cos(\tfrac{2\pi m}{n}) z^{-\frac{N}{n}} \quad , \quad \Delta_3 = 1 \end{aligned} \tag{2.1}$$

and the loop gains can be calculated as:

$$\begin{aligned} L_1 &= -z^{-\frac{2N}{n}} \\ L_2 &= 2\cos(\tfrac{2\pi m}{n}) z^{-\frac{N}{n}} \\ \Delta &= 1 - L_1 - L_2 \end{aligned} \tag{2.2}$$

According to the Mason's Gain Formula, $G^s_{RC}(z)$ can be derived as

$$\begin{aligned} G^s_{RC}(z) &= \frac{1}{\Delta}\sum_{i=1}^{3} P_i\Delta_i \\ &= k_r \frac{\cos(\frac{2m\pi}{n})z^{-\frac{N}{n}} - z^{-\frac{2N}{n}}}{\left(1 - z^{-\frac{N}{n}}e^{-\frac{2m\pi j}{n}}\right)\left(1 - z^{-\frac{N}{n}}e^{\frac{2m\pi j}{n}}\right)}. \end{aligned} \tag{2.3}$$

From (2.3), the poles of SSSRC are located at $(nk \pm m)j\omega_0$. From the viewpoint of the internal model principle (IMP), these poles guarantee zero-error tracking and rejection for targeted $nk \pm m$th order harmonics.

2.2.2 Improved parallel structure selective repetitive control

To improve the performance of selective repetitive control, parallel structure selective repetitive control scheme is proposed as shown in Fig. 2.3, where $e(z)$ is the tracking error; k_r is the learning gain; $\omega_0 = 2\pi f_0 = 2\pi/T_0$ is the fundamental frequency; $N = T_0/T_s$ with T_s being the sample period of the discrete system; $G_{rc}(z)$ is the transfer function of Block-1 and Block-2; $G^P_{RC}(z)$ is the transfer function of the overall PSSRC controller; δ_s and δ_c are

$$\delta_s = \frac{j(e^{-jm\omega_0 t} - e^{jm\omega_0 t})}{2} = sin(\omega_0 t) \tag{2.4}$$

$$\delta_c = \frac{e^{jm\omega_0 t} + e^{-jm\omega_0 t}}{2} = cos(\omega_0 t), \tag{2.5}$$

where j is the imaginary unit and $t = \kappa T_s$ with $\kappa \in \mathbb{N}$ being the time index.

Comparing Fig. 2.1 and Fig. 2.3 shows that the number of delay units (or memory cells) required by PSSRC is reduced to 2/3 of that in SSSRC.

In order to investigate the properties of PSSRC controller, its transfer function in z-domain is derived in the following development.

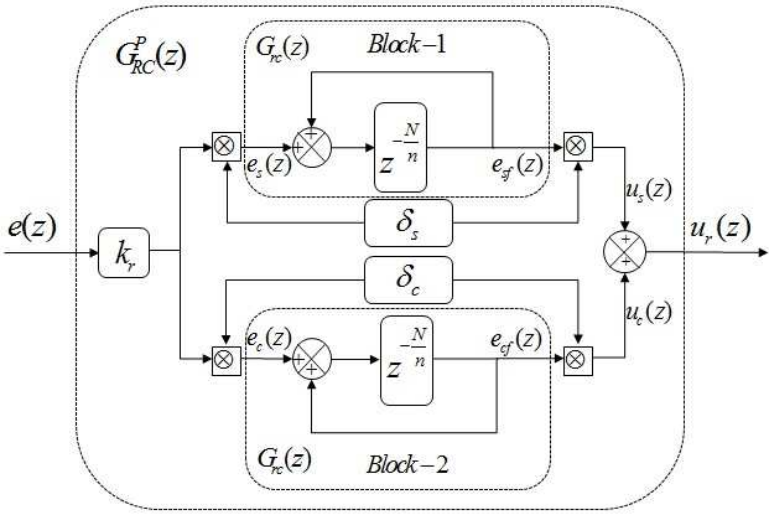


Figure 2.3: Parallel structure selective repetitive controller (PSSRC)

Suppose $e(t)$ and $e_c(t)$ are the time-domain expressions of $e(z)$ and $e_c(z)$ in the lower branch in Fig. 2.3, then $e_c(t)$ can be calculated in time-domain as

$$e_c(t) = k_r \delta_c e(t) = k_r \frac{e^{jm\omega_0 t} + e^{-jm\omega_0 t}}{2} e(t), \tag{2.6}$$

According to the rules of z-transformation, z-domain expression of (2.6), i.e. $e_c(z)$, is

$$e_c(z) = \frac{1}{2} k_r e(ze^{-jm\omega_0 T_s}) + \frac{1}{2} k_r e(ze^{jm\omega_0 T_s}). \tag{2.7}$$

where T_s is the sample period of the discrete system.

The output of Block-2 in the lower branch is

$$e_{cf}(z) = \frac{1}{2} k_r G_{rc}(z) e(ze^{-jm\omega_0 T_s}) + \frac{1}{2} k_r G_{rc}(z) e(ze^{jm\omega_0 T_s}). \tag{2.8}$$

Similar to the derivation from (2.6) to (2.7), the output of the lower branch, $u_c(z)$,

is derived as

$$\begin{aligned} u_c(z) &= \frac{1}{2}e_{cf}(ze^{-jm\omega_0 T_s}) + \frac{1}{2}e_{cf}(ze^{jm\omega_0 T_s}) \\ &= \frac{1}{2}k_r[\frac{1}{2}G_{rc}(ze^{-jm\omega_0 T_s})e(ze^{-2jm\omega_0 T_s}) \\ &\quad +\frac{1}{2}G_{rc}(ze^{-jm\omega_0 T_s})e(z)] \\ &\quad +\frac{1}{2}k_r[\frac{1}{2}G_{rc}(ze^{jm\omega_0 T_s})e(z) \\ &\quad +\frac{1}{2}G_{rc}(ze^{jm\omega_0 T_s})e(ze^{2jm\omega_0 T_s})]. \end{aligned} \tag{2.9}$$

The output of the upper branch, $u_s(z)$, can be derived similarly as $u_c(z)$ by replacing δ_c with $\delta_s = \frac{j(e^{-jm\omega_0 t} - e^{jm\omega_0 t})}{2}$,

$$\begin{aligned} u_s(z) &= \frac{j}{2}k_r[\frac{j}{2}G_{rc}(ze^{jm\omega_0 T_s})e(ze^{2jm\omega_0 T_s}) \\ &\quad -\frac{j}{2}G_{rc}(ze^{jm\omega_0 T_s})e(z)] \\ &\quad -\frac{j}{2}k_r[\frac{j}{2}G_{rc}(ze^{-jm\omega_0 T_s})e(z) \\ &\quad -\frac{j}{2}G_{rc}(ze^{-jm\omega_0 T_s})e(ze^{-2jm\omega_0 T_s})]. \end{aligned} \tag{2.10}$$

Combining (2.9) and (2.10) leads to

$$\begin{aligned} u_r(z) &= u_c(z) + u_s(z) \\ &= \frac{1}{2}k_r G_{rc}(ze^{-jm\omega_0 T_s})e(z) + \frac{1}{2}k_r G_{rc}(ze^{jm\omega_0 T_s})e(z), \end{aligned} \tag{2.11}$$

and (2.11) leads to the overall PSSRC as

$$G_{RC}^P(z) = \frac{u_r(z)}{e(z)} = \frac{k_r}{2}[G_{rc}(ze^{-jm\omega_0 T_s}) + G_{rc}(ze^{jm\omega_0 T_s})]. \tag{2.12}$$

From Fig.2.3, the transfer function $G_{rc}(z)$ can be obtained as

$$G_{rc}(z) = \frac{z^{-\frac{N}{n}}}{1 - z^{-\frac{N}{n}}}. \tag{2.13}$$

Substituting (2.13) into (2.12) leads to the transfer function of the proposed PSSRC as

$$\begin{aligned} G^P_{RC}(z) &= \frac{k_r}{2}\left(\frac{z^{-\frac{N}{n}} e^{\frac{-2m\pi j}{n}}}{1 - z^{-\frac{N}{n}} e^{\frac{-2m\pi j}{n}}} + \frac{z^{-\frac{N}{n}} e^{\frac{2m\pi j}{n}}}{1 - z^{-\frac{N}{n}} e^{\frac{2m\pi j}{n}}}\right) \\ &= k_r \frac{cos(\frac{2m\pi}{n}) z^{-\frac{N}{n}} - z^{-\frac{2N}{n}}}{(1 - z^{-\frac{N}{n}} e^{\frac{-2m\pi j}{n}})(1 - z^{-\frac{N}{n}} e^{\frac{2m\pi j}{n}})}. \end{aligned} \tag{2.14}$$

Comparing (2.3) and (2.14) shows that SSSRC and PSSRC have equivalent transfer function, i.e. PSSRC achieves the same selectivity with less delay units and simpler structure.

But Fig. 2.1 and Fig. 2.3 show ideal SSSRC and PSSRC controllers. In practical applications, high-frequency components and system uncertainty may lead the system to instability due to the infinite poles on the unit circle. In order to solve this problem, low-pass filters are usually introduced to improve robustness of the whole system. Typical closed-loop systems with plug-in modified PSSRC or SSSRC are shown in Fig. 2.4, where $R(z)$ is the reference signal; $y(z)$ is the output; $d(z)$ is the disturbance; $u(z)$ is the final control signal; $G_p(z)$ is the control plant; $G_c(z)$ is a feedback controller used to compensate the control plant; $Q(z)$ and $G_f(z)$ are filters to be designed.

$Q(z)$ is a zero phase moving average filter with the form in (2.15) and it is introduced to suppress the unexpected high frequency repetitive disturbance. The design of $Q(z)$ is consistent in different types of RC controllers [59].

$$Q(z) = \frac{\left(\sum_{l=0}^{h} a_l z^l + \sum_{l=1}^{h} a_l z^{-l}\right)}{2\sum_{l=1}^{h} a_l + a_0} \tag{2.15}$$

Suppose $z = e^{j\omega}$, then it can be obtained from (2.15) that $|Q(e^{j\omega})| \leq 1$ for all ω and $Q(e^{j\omega}) = 1$ only when $\omega = 0$. So low pass filter $Q(z)$ attenuates RC's gains at harmonic frequencies. Benefiting from the parallel structure, however, PSSRC achieves higher gains than SSSRC.

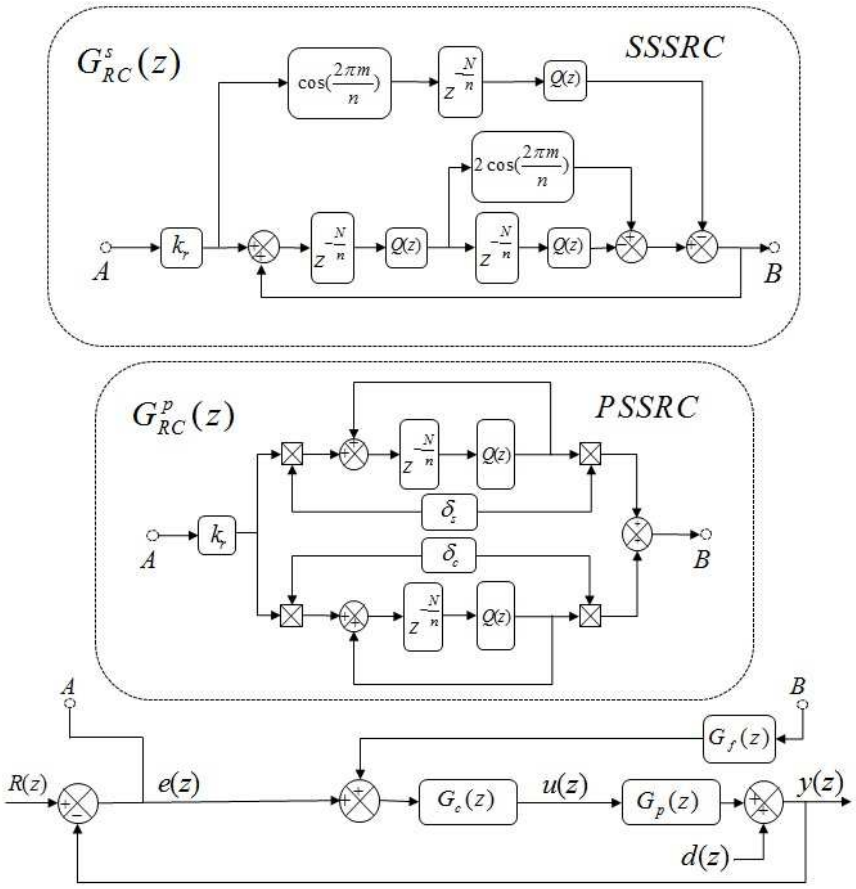


Figure 2.4: Plug-in PSSRC and SSSRC systems

From Fig. 2.4, the transfer function of the modified PSSRC and SSSRC can be derived as

$$G^P_{RC} = \frac{k_r}{2}\left(G^{P+}_{RC} + G^{P-}_{RC}\right), \tag{2.16}$$

where

$$G_{RC}^{P+} = \frac{Q(ze^{jm\omega_0 T})z^{-\frac{N}{n}}e^{-\frac{2m\pi j}{n}}}{1 - Q(ze^{jm\omega_0 T})z^{-\frac{N}{n}}e^{-\frac{2m\pi j}{n}}}, \tag{2.17}$$

$$G_{RC}^{P-} = \frac{Q(ze^{-jm\omega_0 T})z^{-\frac{N}{n}}e^{\frac{2m\pi j}{n}}}{1 - Q(ze^{-jm\omega_0 T})z^{-\frac{N}{n}}e^{\frac{2m\pi j}{n}}}, \tag{2.18}$$

and

$$G_{RC}^{S} = \frac{k_r}{2}\left(G_{RC}^{S+} + G_{RC}^{S-}\right), \tag{2.19}$$

where

$$G_{RC}^{S+} = \frac{Q(z)z^{-\frac{N}{n}}e^{-\frac{2m\pi j}{n}}}{1 - Q(z)z^{-\frac{N}{n}}e^{-\frac{2m\pi j}{n}}}, \tag{2.20}$$

$$G_{RC}^{S-} = \frac{Q(z)z^{-\frac{N}{n}}e^{\frac{2m\pi j}{n}}}{1 - Q(z)z^{-\frac{N}{n}}e^{\frac{2m\pi j}{n}}}. \tag{2.21}$$

Combining (2.15), (2.16), (2.17) and (2.18) yields,

$$\left|G_{RC}^{P}\right|_{z=e^{jm\omega_0 T}} = \frac{Q(1)}{1 - Q(1)} + \frac{Q(e^{2jm\omega_0 T})e^{-\frac{4m\pi j}{n}}}{1 - Q(e^{2jm\omega_0 T})e^{-\frac{4m\pi j}{n}}} = \inf \tag{2.22}$$

which means that PSSRC achieves infinite gain at $j\omega = jm\omega_0$.

Similarly, SSSRC's gain at $j\omega = jm\omega_0$ can be derived from (2.19) as

$$\left|G_{RC}^{S}\right|_{z=e^{jm\omega_0 T}} = \frac{Q(e^{jm\omega_0 T})}{1 - Q(e^{jm\omega_0 T})} + \frac{Q(e^{jm\omega_0 T})e^{-\frac{4m\pi j}{n}}}{1 - Q(e^{jm\omega_0 T})e^{-\frac{4m\pi j}{n}}} \tag{2.23}$$

Comparing (2.22) and (2.23) shows that PSSRC's control gain at $j\omega = jm\omega_0$ is largely improved to infinity with the same low pass filter $Q(z)$ as in SSSRC. When $m = 1$, this gain is for fundamental frequency which guarantees better tracking performance, while this gain is for particular harmonic frequency when $m > 1$. It is noted that in single phase or three-phase converter systems, m usually equals to 1 because $4k \pm 1$th and $6k \pm 1$th order harmonics dominate the THD in these two kinds

of systems. Besides, instead of connecting two low-pass filters in series as in Fig. 2.4, PSSRC adopts parallel structure to improve the control gains at harmonics.

The calculated gains by Matlab from (2.16) and (2.19) when $n = 4$, $m = 1$, $h = 1$, $a_0 = 0.5$ and $a_1 = 0.25$ are shown in Fig. 2.5. The circles and triangles mark the control gains at fundamental and odd-harmonic frequencies. Fig. 2.5 shows that both PSSRC and SSSRC achieves high gains at the system's fundamental and odd-harmonic frequencies. But comparing the control gains of these two schemes, PSSRC offers higher gains especially at fundamental frequency.

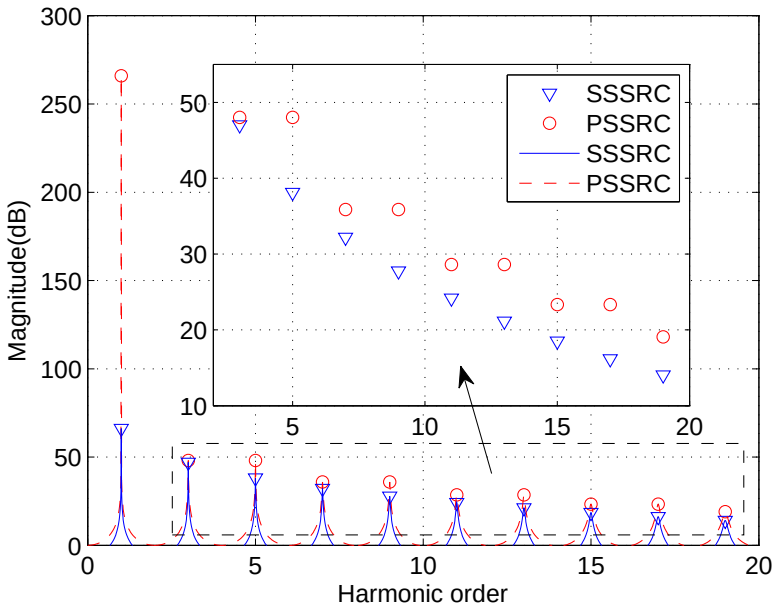


Figure 2.5: Comparison of control gains of different RCs with the same $Q(z)$

Remark 2.1. *PSSRC constructs a selective repetitive controller in a relatively simpler and more effective way. The delay units (or memory cells) required is reduced to 2/3 of that in SSSRC. Besides, PSSRC improves the control gains at the system's fundamental and harmonic frequencies, thus offering better static tracking performance.*

2.2.3 Stability analysis

In this section, stability criteria are developed for the proposed digital PSSRC systems.

In Fig. 2.4, $G_c(z)$ is designed to stabilize the closed-loop system without the plug-in PSSRC controller. This means that all the poles of the following transfer function must locate inside the unit circle:

$$H(z) = \frac{G_c(z)G_p(z)}{1 + G_c(z)G_p(z)}. \tag{2.24}$$

After plugging the PSSRC controller into the closed-loop system, output of the overall system can be derived from Fig. 2.4 as

$$\frac{y(z)}{R(z)} = \frac{[1 + G_{RC}^P(z)]H(z)}{1 + G_{RC}^P(z)H(z)}, \tag{2.25}$$

and

$$\frac{y(z)}{d(z)} = \frac{1}{[1 + G_{RC}^P(z)H(z)][1 + G_c(z)G_p(z)]} \tag{2.26}$$

In order to guarantee the stability of the overall plug-in PSSRC system, all the poles of (2.25) and (2.26) must be inside the unit circle. For (2.26), the roots of $1 + G_c(z)G_p(z) = 0$ can be located inside the unit circle by choosing appropriate $G_c(z)$ from (2.24). Therefore, in order to ensure stability, it must hold that

$$1 + G_{RC}^P(z)H(z) \neq 0 \quad for \quad \forall |z| \ > 1. \tag{2.27}$$

Substituting (2.17) and (2.18) into (2.27) leads to that

$$\frac{k_r}{2}[G_{RC}^{P+}(z) + G_{RC}^{P-}(z)] \neq -\frac{1}{G_f(z)H(z)} \quad for \quad \forall |z| \ > 1. \tag{2.28}$$

Therefore, $G_f(z)$ must be carefully chosen. Without losing generality, $H(z)$ is

rewritten to

$$H(z) = \frac{z^{-d}B(z)}{A(z)} \tag{2.29}$$

where d represents the delay of the system, then $G_f(z)$ is designed as [59]

$$G_f(z) = \frac{z^d A(z) B^-(z^{-1})}{B^+(z)b} \tag{2.30}$$

where B^- represents the noncancellable portion of B, B^+ represents the cancellable portion, and $b \geq \mid B^-(z) \mid^2$ for all z. It should be noted that the delay $z^{-N/n}$ introduced by PSSRC with $N/n > d$ can help to implement the noncausal part z^d in $G_f(z)$.

From (2.30), it is straightforward to derive that

$$0 < G_f(z)H(z) = \frac{B^-(z)B^-(z^{-1})}{b} \leq 1. \tag{2.31}$$

With further manipulation on (2.31), it is easy to get

$$-\frac{1}{G_f(z)H(z)} \leq -1. \tag{2.32}$$

It has been proved in [211] that the following inequalities hold

$$Re[G_{rc}^{P+}(z)] \geq -\frac{1}{2} \quad for \quad \forall \mid z \mid > 1 \tag{2.33}$$

$$Re[G_{RC}^{P-}(z)] \geq -\frac{1}{2} \quad for \quad \forall \mid z \mid > 1. \tag{2.34}$$

Combine (2.33), (2.34) and if k_r satisfies $0 < k_r < 2$, then

$$Re[\frac{k_r}{2}[G_{RC}^{P+}(z) + G_{RC}^{P-}(z)]] > -1. \tag{2.35}$$

Equation (2.32), along with (2.35), guarantees that (2.28) holds, which means that

no poles of the overall system are outside the unit circle.

From the above analysis, the closed-loop PSSRC systems as shown in Fig. 2.4 are asymptotically stable if the following conditions are satisfied:

1. The closed-loop system without the plug-in PSSRC controller is stable, or all the roots of $1 + G_c(z)G_p(z) = 0$ are inside the unit circle;

2. $G_f(z)$ is chosen as in the form of (2.30) and satisfies (2.31);

3. k_r is in the range of (0,2).

2.3 Case study

In this section, the proposed PSSRC is applied to a single-phase DC/AC PWM inverter to illustrate its effectiveness. Fig. 2.6 shows the plug-in PSSRC-controlled system. In Fig. 2.6, E is the DC bus voltage; L and C are LC filter; i_o is the output current; v_o is the output voltage. The load bank consists of two kinds of loads: (a) resister load with R and (b) rectifier load with L_r, C_r and R_r. The detailed system parameters are listed in Table 2.1. The controller is programmed in Matlab/Simulink and realized in realtime by a Dspace 1104.

2.3.1 System modelling and controller design

In the system shown in Fig. 2.6, v_{in} is the input PWM voltage defined by the following pattern:

$$v_{in} = \begin{cases} E \quad when\, S1\, and\, S4\, are\, on,\ S2\, and\, S3\, are\, off; \\ -E \quad when\, S2\, and\, S3\, are\, on,\ S1\, and\, S4\, are\, off. \end{cases} \tag{2.36}$$

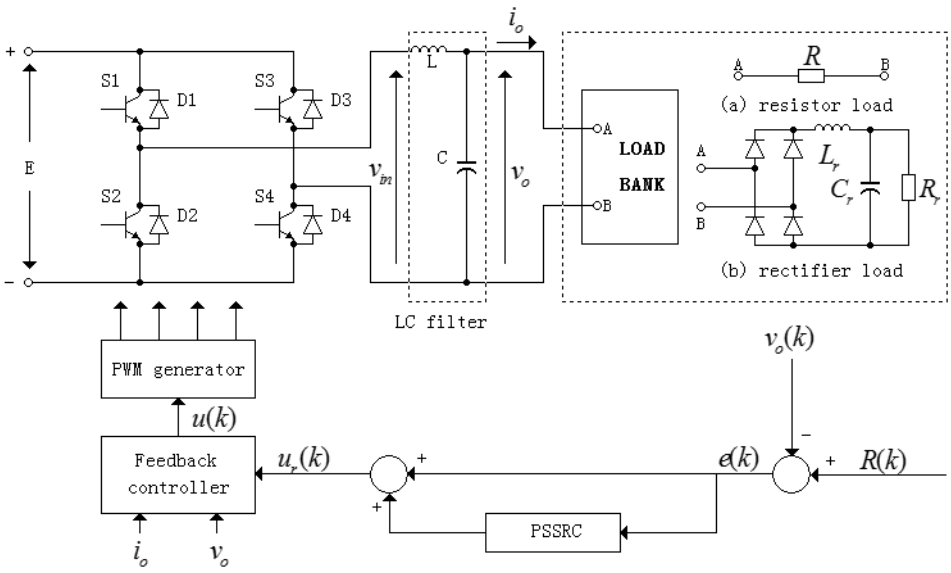


Figure 2.6: Plug-in PSSRC controlled single phase inverter system

Table 2.1: System parameters of single phase PWM inverter (PSSRC)

Meaning and symbol	*Value*
DC bus voltage E	400 V
Output filter inductance L_f	7 mH
Output filter capacitance C_f	50 uF
Resistor load R	20 Ω
Rectifier load L_r	5 mH
Rectifier load C_r	1100 uF
Rectifier load R_r	30 Ω
Fundamental frequency f_0	50 Hz
Sampling frequency f_s	10 kHz
Switching frequency	10 kHz
Amplitude of reference R	270 V

Then the discrete single phase PWM DC/AC inverter system can be expressed by

$$\begin{cases} x(k+1) = Ax(k) + Bu(k) \\ y(k) = Cx(k) \end{cases} \tag{2.37}$$

where $x(k) = \begin{bmatrix} v_o(k) \\ \dot{v}_0(k) \end{bmatrix}$, $u(k) = v_{in}(k)$, $y(k) = v_o(k)$, $A = \begin{bmatrix} \varphi_{11} & \varphi_{12} \\ \varphi_{21} & \varphi_{22} \end{bmatrix}$, $B = \begin{bmatrix} g_1 \\ g_2 \end{bmatrix}$ with

$\varphi_{11} = 1 - (1/2LCf_s^2)$,

$\varphi_{12} = 1/f_s - (1/2CRf_s^2)$,

$\varphi_{21} = -(1/LCf_s) + (1/2LC^2Rf_s^2)$,

$\varphi_{22} = 1 - (1/CRf_s) - (1/2LCf_s^2) + (1/2C^2R^2f_s^2)$,

$g_1 = 1/2LCf_s^2$ and $g_2 = (1/LCf_s)(1 - (1/2CRf_s))$.

Following the analysis in section 2.2.3 and stability criterion 1, $G_c(z)$ should be designed to make (2.24) stable. Suppose the following state feedback controller is adopted in discrete domain

$$u(k) = \begin{bmatrix} -k_1 & -k_2 \end{bmatrix} x(k) + hv_{ref}(k), \tag{2.38}$$

and substitute (2.38) into (2.37), then state equation of the system with only the state feedback controller can be derived as

$$\begin{aligned} x(k+1) = & \begin{bmatrix} \varphi_{11} - k_1g_1 & \varphi_{12} - k_1g_2 \\ \varphi_{21} - k_2g_1 & \varphi_{22} - k_2g_2 \end{bmatrix} x(k) \\ & + \begin{bmatrix} g_1h \\ g_2h \end{bmatrix} v_{ref}(k) \end{aligned} . \tag{2.39}$$

Finding the transfer function of (2.39), the system without the plug-in PSSRC can be expressed by

$$H(z) = \frac{m_1 z + m_2}{z^2 + p_1 z + p_2} \tag{2.40}$$

where

$p_1 = -(\varphi_{11} - g_1 k_1 + \varphi_{22} - g_2 k_2)$,

$p_2 = (\varphi_{11} - g_1 k_1)(\varphi_{22} - g_2 k_2) - (\varphi_{12} - g_1 k_2)(\varphi_{21} - g_2 k_1)$,

$m_1 = g_1 h$ and $m_2 = (\varphi_{12} - g_1 k_2) g_2 h - (\varphi_{22} - g_2 k_2) g_1 h$.

From (2.40), it is clear that the poles can be arbitrarily assigned by designing the state feedback controller (2.38).

Suppose a state feedback controller is designed as

$$u(k) = \begin{bmatrix} -1.6255 & -1.0224 \times 10^{-3} \end{bmatrix} x(k) + 2.0 v_{ref}(k), \tag{2.41}$$

then (2.40) can be calculated as

$$H(z) = \frac{0.0029z - 0.187}{z^2 - 1.048z + 0.2022}. \tag{2.42}$$

It is quite straightforward to verify that all the poles of $H(z)$ are inside the unit circle.

From (2.30), filter $G_f(z)$ is chosen as

$$G_f(z) = \frac{z^6}{H(z)} \tag{2.43}$$

where z^6 is used to compensate the phase lag and unmodelled delay of the system. The number of delays used is determined from the experiment by trial and error.

Since odd harmonics dominate the THD in a single phase AC system, n is set to be 4 and m is set to be 1. Detailed parameters of the controller are listed in Table 2.2.

Table 2.2: Controller design for single phase system (parameters)

$k_1 = 1.6255$	$k_2 = 1.0224 \times 10^{-3}$
$h = 2.0$	$k_r = 0.0015$
$Q(z) = 0.25z^{-1} + 0.5 + 0.25z$	$m = 1, n = 4$

2.3.2 Experimental results

The capability of the proposed PSSRC to reject harmonics under different loads is verified by experimental results shown in Fig. 2.7. Fig. 2.7(a) and Fig. 2.7(c) show the output voltage v_o, reference voltage v_{ref} and current i_o under two different loads. Fig. 2.7(b) and Fig. 2.7(d) are the corresponding harmonic spectrums of the output voltage v_o. Fig. 2.7 shows that PSSRC can drive the output to track the reference with very low THD of 0.55% under resistor load and 1.39% under rectifier load.

Besides, in order to illustrate the advantages of the proposed PSSRC, SSSRC and CRC controlled experimental results are provided in Fig. 2.8 and Fig. 2.9. The same parameters and filters as in Table 2.2 are applied in these two cases. Comparing Fig. 2.7 and Fig. 2.8, PSSRC gets better static tracking and rejection performance than SSSRC under both linear and nonlinear loads. This is because $4k \pm 1$ PSSRC achieves higher gains at fundamental frequency and harmonic frequencies than $4k \pm 1$ SSSRC (refer to Fig. 2.5). Meanwhile, the performance of PSSRC is slightly worse than CRC because PSSRC rejects only odd-harmonics while CRC works on all the harmonics. But PSSRC gets much better convergence rate and transient performance than CRC. This can be illustrated by the experimental results in Fig. 2.10 and Fig. 2.11.

Fig. 2.10 shows the tracking error of PSSRC, SSSRC and CRC after plugging in the RCs at $t = 1s$. It is shown that CRC takes nearly $0.4s$ to converge while PSSRC converges within $0.2s$. The convergence rate is largely improved compared with CRC. Besides, Fig. 2.10 shows that PSSRC and SSSRC offer similar convergence

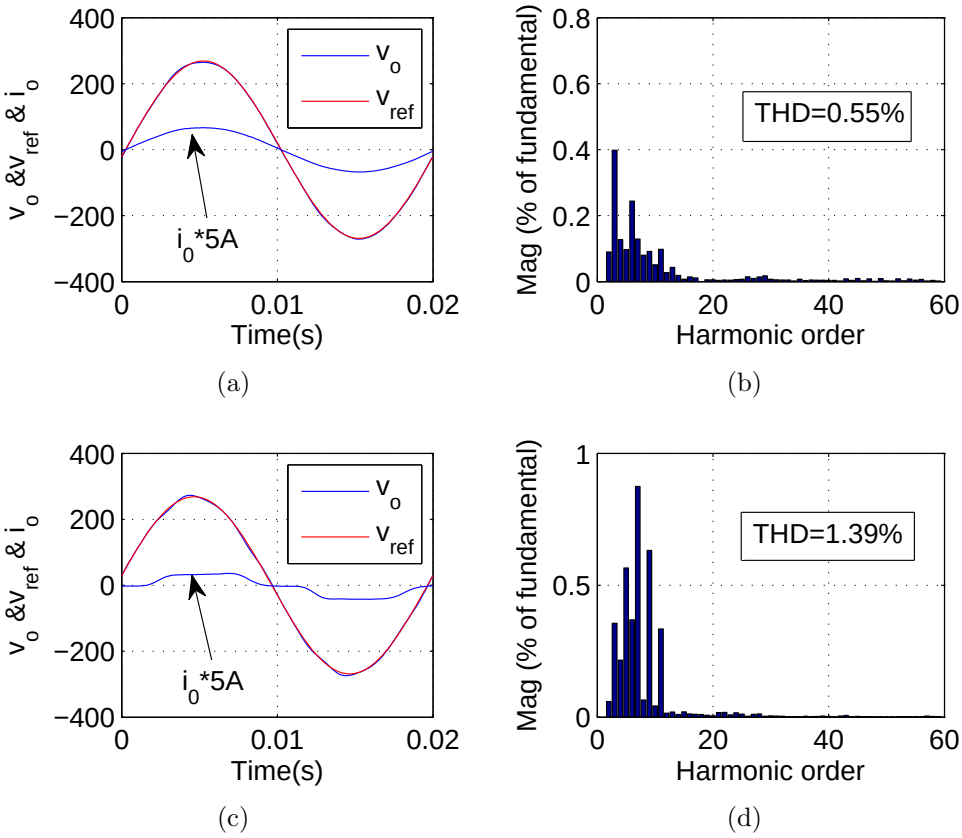


Figure 2.7: Steady performance of PSSRC. Output voltage & current under: (a) resistor load; (c) rectifier load. Harmonic spectrum of the output voltage under: (b) resistor load; (d) rectifier load.

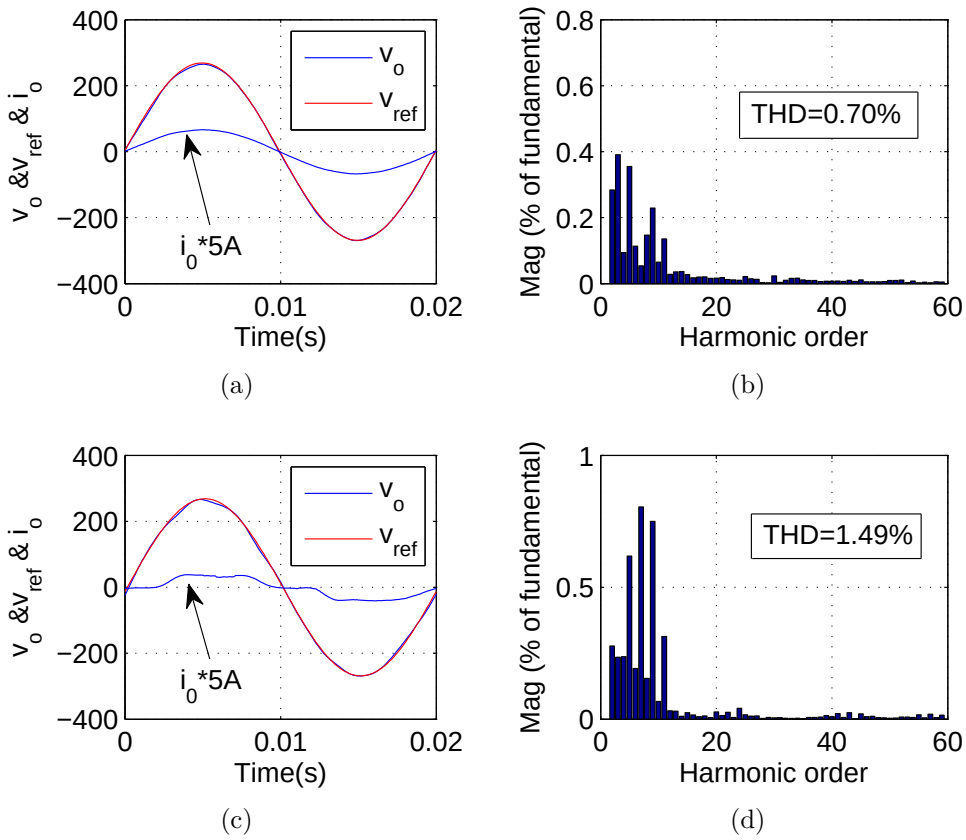


Figure 2.8: Steady performance of SSSRC. Output voltage & current under: (a) resistor load; (c) rectifier load. Harmonic spectrum of the output voltage under: (b) resistor load; (d) rectifier load.

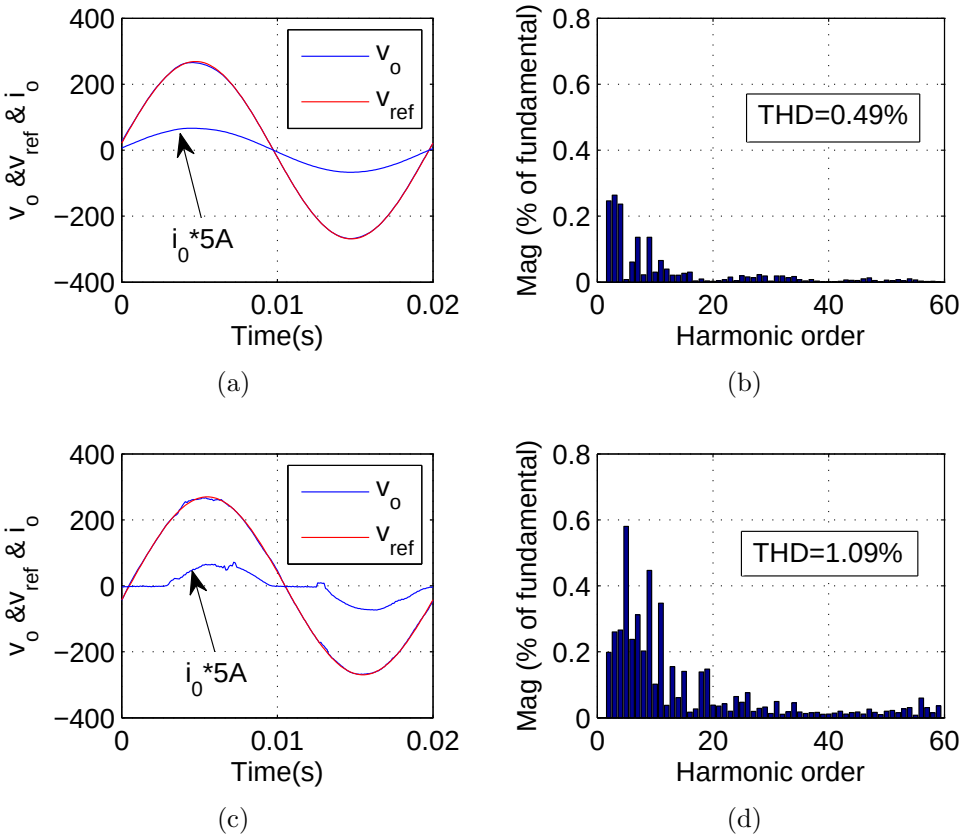
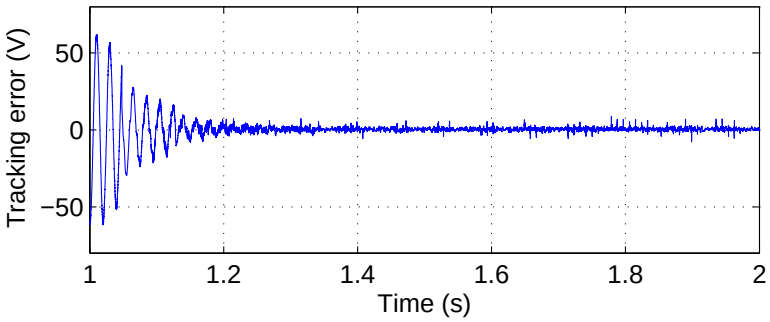


Figure 2.9: Steady performance of CRC. Output voltage & current under: (a) resistor load; (c) rectifier load. Harmonic spectrum of the output voltage under: (b) resistor load; (d) rectifier load.

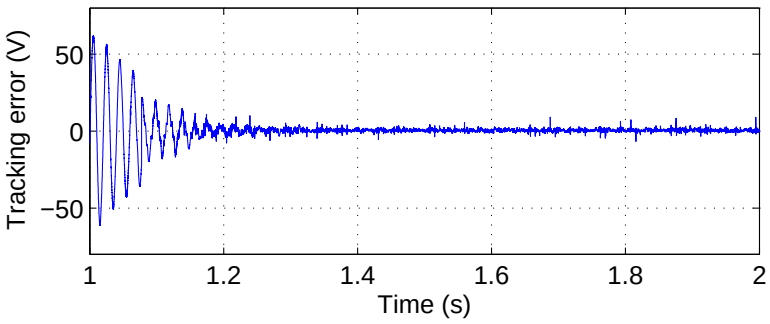
rates.

Comparison of transient performance between PSSRC and CRC when load changes occur is made in Fig. 2.11. It shows that step load change between rectifier load and no load can be handled well by PSSRC. The PSSRC controlled output voltage recovers within three cycles, while it takes much more time for the CRC controlled output voltage to recover.

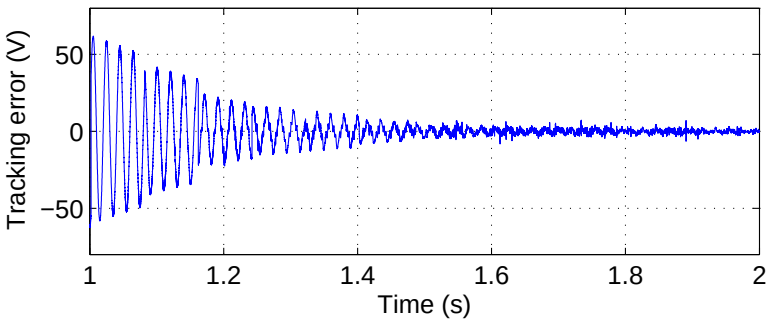
Detailed comparisons of performance between PSSRC and other RCs are listed in Table 2.3. It is noted that the number of delay units (or memory cells) required in PSSRC is the least among the three RC controllers. Besides, the computational time of PSSRC in real-time is decreased by 10% compared to SSSRC while the steady tracking performance is improved against SSSRC. It should be noted that



(a)



(b)



(c)

Figure 2.10: Convergence rate with different RC controllers plugged into the system at $t = 1.0s$. (a) PSSRC; (b) SSSRC; (c) CRC.

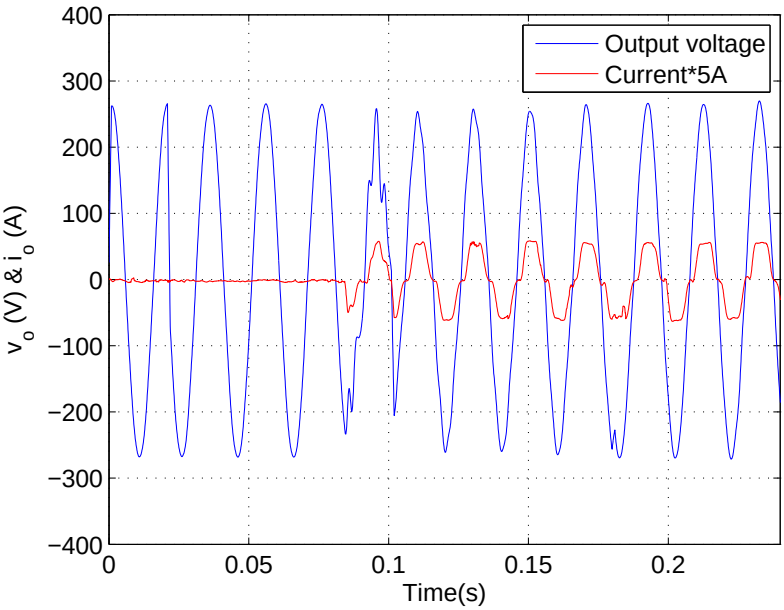
Table 2.3: Performance comparison between PSSRC and other RCs

	PSSRC	*SSSRC*	*CRC*
THD under resistor load	0.55%	0.70%	0.49%
THD under rectifier load	1.39%	1.49%	1.09%
Convergence time	About 0.2 s	About 0.2 s	About 0.4 s
Flexibility	Selective	Selective	None
Delay units required	2×50	3×50	200
Turnaround time	36.55 μs	41.75 μs	38.94 μs

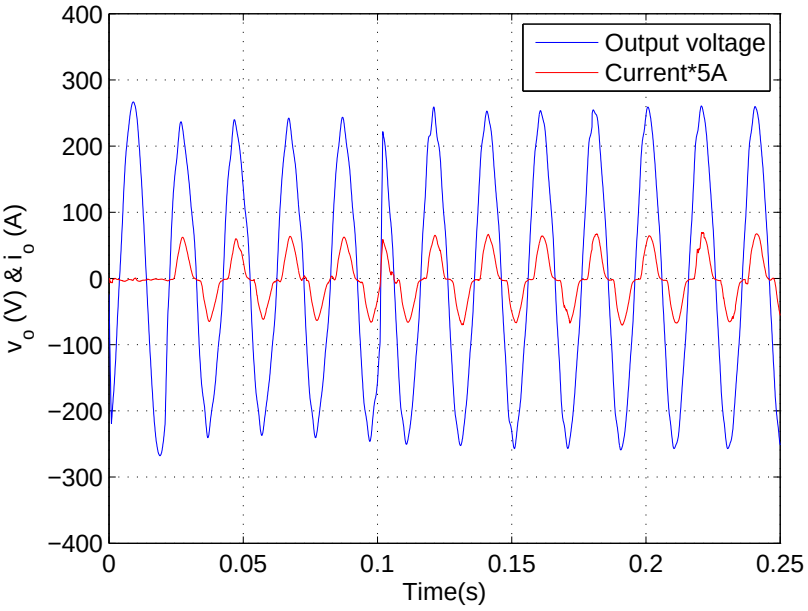
δ_c and δ_s are generated by software in the controller in this experiment, so the computational burden can be further decreased if they are generated by hardware circuits. Moreover, PSSRC achieves flexible selectivity by assigning appropriate values to m and n.

2.4 Summary

This chapter proposes a parallel structure selective repetitive control scheme for PWM converters. It is proved in this chapter that PSSRC achieves zero-error tracking and rejection at any $nk \pm m$th order harmonics by selecting m and n at will. Besides, a simpler parallel structure is adopted in PSSRC to avoid connecting filters in series. As a result, the control gains are improved against SSSRC and better static tracking performance is guaranteed. Moreover, both the number of delay units required and computational burden are decreased in PSSRC. Experiments on a single-phase PWM DC/AC inverter system are carried out and the experimental results verify the advantages of PSSRC successfully.



(a)



(b)

Figure 2.11: Transient performance: (a) From no load to rectifier load (PSSRC); (b) From no load to rectifier load (CRC)

Chapter 3

Parallel Structure Fractional Repetitive Control

3.1 Introduction

In the last chapter, digital selective repetitive controller is realized in parallel structure and achieves better flexibility and transient response. Digital repetitive controller is mostly realized by N delay units as shown in Fig. 3.1. This configuration works perfectly when N is an integer, i.e., the sampling frequency can be divided by fundamental frequency completely without remainder. However, N is not always an integer in real applications. For example, conventional repetitive control (CRC) requires 166.67 delay units to accurately compensate a $60Hz$ output voltage when the sampling rate of the system $f_s = 10kHz$. In these cases, CRC usually rounds N to its nearest integer. But because the poles of CRC are no longer located accurately at harmonic frequencies, the gains of CRC can be attenuated from $250dB$ to $40dB$. Recently, several RC controllers have been proposed to improve the performance of RC in fractional cases. In [51,52] , high-order RC decreases the partial derivative of RC's control gains with respect to the fundamental frequency to

achieve better robustness. [45,213] solve this problem by approximating fractional order of delay using filters. [46] proposes a spatial repetitive controller (SRC) to address this problem in phase sample position domain.

In this chapter, THD of CRC controlled output is analyzed to investigate the functionalities and limitations of CRC in fractional cases. It is found that CRC works well for high-order harmonics. Low-order odd components dominate the THD, although all the gains of CRC are attenuated substantially. Therefore, a new parallel structure fractional repetitive control (PSFRC) scheme is proposed to enhance the compensation on low-order odd harmonics. This control scheme locates poles accurately at every targeted low-order harmonic frequency and derives infinite high gains. Meanwhile, control gains for higher-order components are improved to guarantee better rejection performance. As a result, PSFRC suppresses all dominant harmonics in fractional cases effectively. Besides, transient performance is improved by focusing PSFRC's compensation on odd harmonics. Furthermore, the number of required delay units in PSFRC is reduced to half of that in CRC.

Stability criteria are derived for plug-in PSFRC systems in this chapter. Experimental results of applying PSFRC to a single-phase DC/AC inverter are presented to verify the advantages of the proposed method.

The remainder of this chapter is organized as follows. In section 3.2, parallel structure fractional repetitive control scheme is proposed with stability analysis. Section 3.3 provides experimental results applying PSFRC to a single-phase PWM inverter to illustrate the effectiveness of this scheme. Section 3.4 summarizes this chapter.

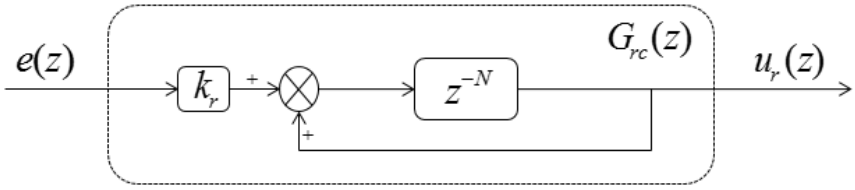


Figure 3.1: Conventional plug-in RC controller

3.2 Parallel structure fractional repetitive control

3.2.1 Conventional RC and its limitations in fractional cases

A typical closed-loop system controlled by plug-in CRC is shown in Fig. 3.2, where $R(z)$ is the reference signal; $y(z)$ is the output; $e(z)$ is the tracking error; $d(z)$ is the disturbance; k_r is the learning gain of CRC; $u_r(z)$ is the control signal from CRC; $u(z)$ is the overall control signal; $G_c(z)$ is a state-feedback (SFB) controller; $G_p(z)$ is the objective plant; $N = f_s/f_0$ with f_s being the sampling rate and $f_0 = \omega_0/2\pi$ being the fundamental frequency; $G_{crc}(z)$ is the transfer function of CRC; $Q(z)$ and $G_f(z)$ are filters to improve robustness of the whole system.

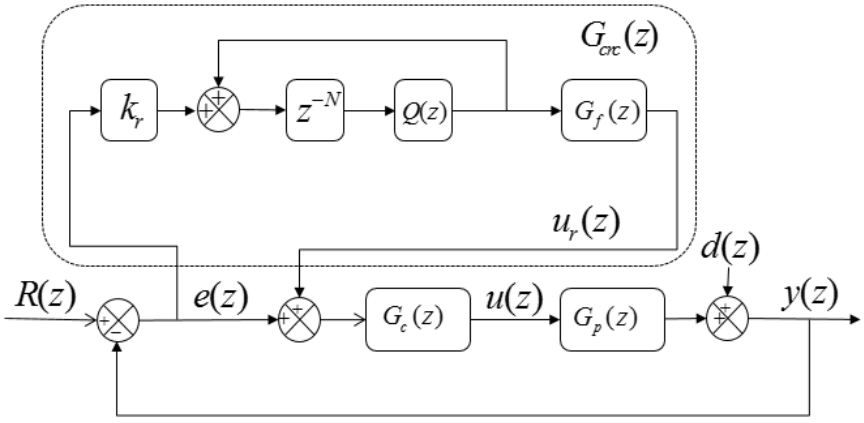


Figure 3.2: CRC controlled closed-loop system

From Fig. 3.2, transfer function of CRC can be obtained as

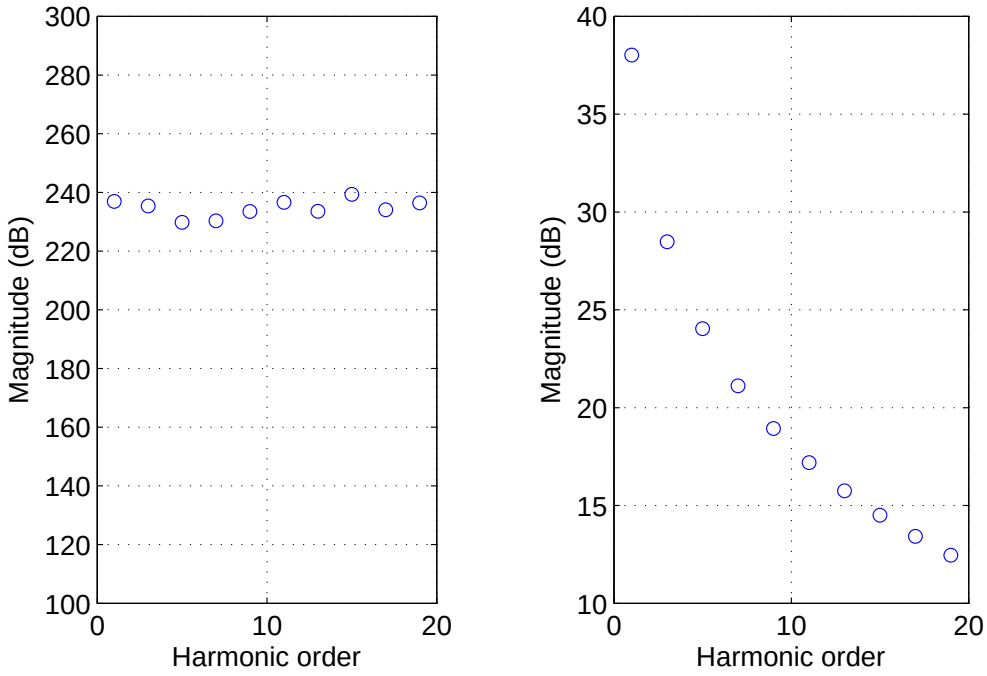
$$G_{crc}(z) = G_f(z)\frac{k_r z^{-N} Q(z)}{1 - z^{-N} Q(z)}. \tag{3.1}$$

Equation (3.1) shows that CRC has poles located at fundamental frequency and all harmonics whose frequencies equal to $k\omega_0$ with $k = 0, 1, 2, 3....$ However, CRC is realizable only in discrete systems whose sampling rates f_s can be divided by fundamental frequency f_0 without remainder. In fractional cases, N has to be rounded to its nearest integer $N^* = round(N)$, however, the gains for all frequencies are attenuated. For example, $N_1 = 150$ in a discrete system with $f_s = 9kHz$ and $f_0 = 60Hz$. In this case, CRC controller can be implemented directly as in Fig. 3.1. However, $N_2 = 166.67$ if the system's sampling rate $f_s = 10kHz$. In order to implement CRC, N_2 has to be rounded to $N_2^* = 167$. The control gains of CRC for odd harmonics in these two scenarios are shown in Fig. 3.3(a) and Fig. 3.3(b), respectively. It is noted here that $Q(z)$ is omitted in this comparison to show the difference because it affects CRC in the same way in both normal and fractional cases. The following transfer function is used to calculate the gains by Matlab:

$$G_{crc}^*(z) = \frac{z^{-N}}{1 - z^{-N}}. \tag{3.2}$$

Fig.3.3 shows that the control gains of CRC are very large around $250dB$ when N is an integer, while the control gains reduce to lower than $40dB$ when N is rounded to its nearest integer in fractional cases. This is because that all the poles of CRC are shifted from $j\omega = kj\omega_0$ (fundamental harmonics) to $j\omega = kj\omega_0^*$ after rounding N to N^*, where $\omega_0 = 2\pi f_0 = 2\pi f_s/N$, $\omega_0^* = 2\pi f_s/N^*$ and $k = 0, 1, 2...$

Next the functionalities and limitations of CRC in fractional cases are analyzed using real experimental data. Fig. 3.4 shows the output voltage of SFB controlled



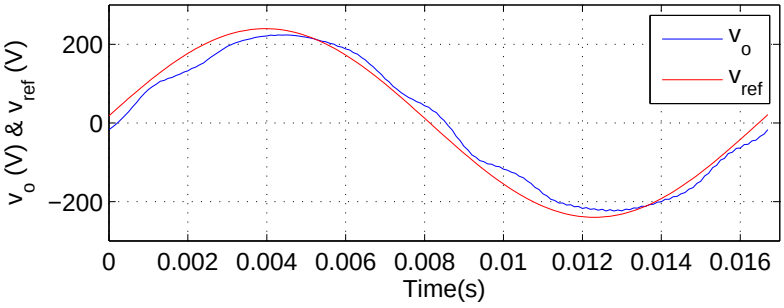
(a) (b)

Figure 3.3: Control gains of CRC with different f_s (a) $f_s = 9kHz$, $N_1 = 150$, $f_0 = 60Hz$; (b) $f_s = 10kHz$, $N_2^* = round(166.67) = 167$, $f_0 = 60Hz$.

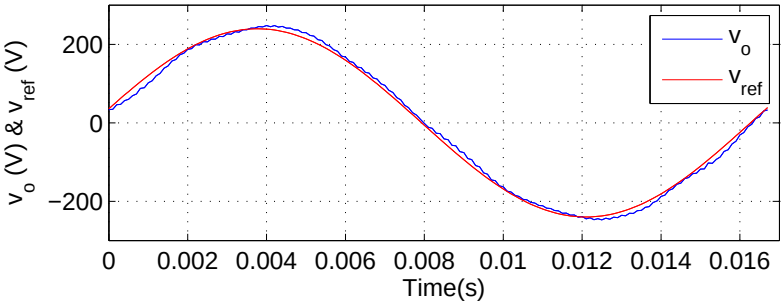
and plug-in CRC controlled single-phase PWM inverters. Fig. 3.5 and Fig. 3.6 are the spectrums of the corresponding output voltages. The sampling rate of the system $f_s = 10kHz$ and the fundamental frequency $f_0 = 60Hz$. It is straightforward to calculate that $N = 166.7$ and the controller as shown in Fig. 3.2 is applied with $N^* = 167$.

Fig. 3.4 shows that plug-in CRC achieves better tracking performance than SFB control. The THD of SFB controlled output and plug-in CRC controlled output are 4.95% and 2.48%, respectively.

Comparing Fig. 3.5 and Fig. 3.6, it can be seen that harmonics higher than 10th order are effectively rejected by CRC in fractional case. From Fig. 3.6(b), the magnitude of high-order harmonics are suppressed to lower than 0.02% which is very small and only make minor contributions to the THD. Meanwhile, Fig. 3.6(a)

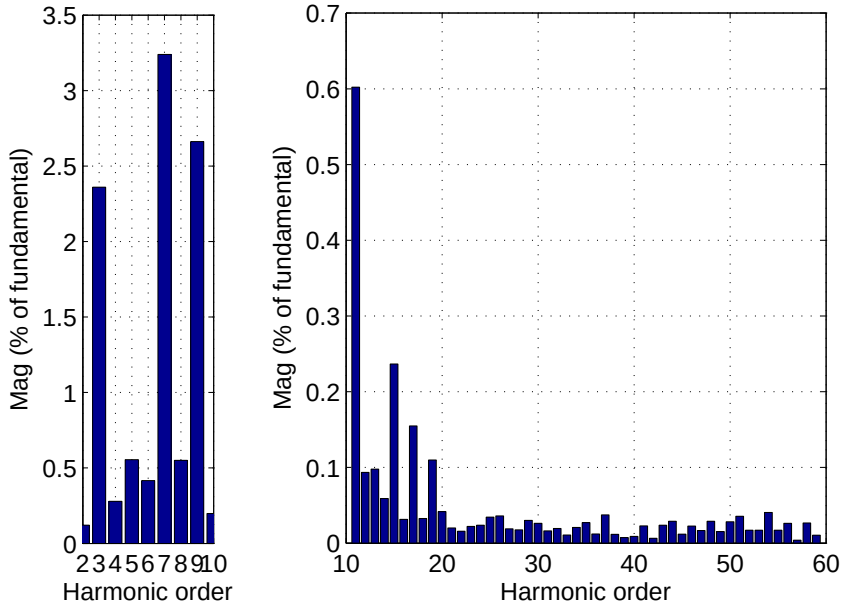


(a)



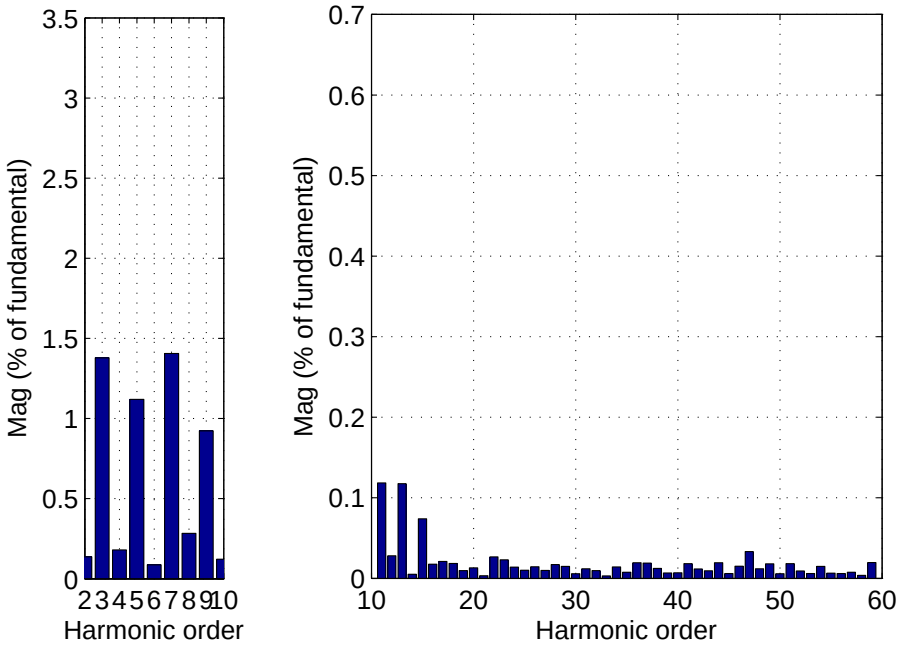
(b)

Figure 3.4: Tracking performance of different control schemes in single phase PWM inverter. (a) with SFB control only; (b) with plug-in CRC control.



(a) (b)

Figure 3.5: Spectrum of output voltage controlled by SFB only. (a) spectrum under 10th order; (b) spectrum over 10th order.



(a) (b)

Figure 3.6: Spectrum of output voltage controlled by plug-in CRC. (a) spectrum under 10th order; (b) spectrum over 10th order.

shows that low-order odd harmonics dominate the THD. In other words, unpleasant distortions in CRC controlled output are caused by the weak compensation on low-order odd harmonics, and the compensation from CRC is satisfiable for high-order harmonic rejection. Therefore, the key point to improve RC's performance in fractional cases is to improve the control gains for dominant low-order harmonics.

3.2.2 Digital PSFRC controller

Based on the analysis in section 3.2.1, a novel parallel structure fractional repetitive control (PSFRC) scheme is proposed as shown in Fig. 3.7, where $e(z)$ is the tracking error; k_i is the learning gain with $i = 1, 3, 5...$ and $i \leq n$; $f_0 = \omega_0/2\pi$ is the fundamental frequency; f_s is the sampling rate; δ is a correction factor that will be defined later; $u_r(z)$ is the control signal from PSFRC; $G_i(z)$ is the transfer function of the ith branch and $G_{RC}(z)$ is the transfer function of the overall PSFRC.

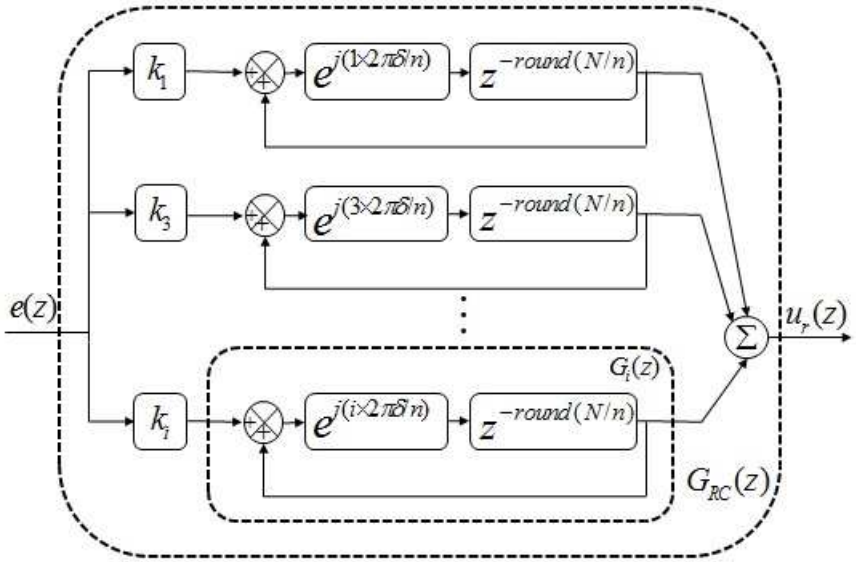


Figure 3.7: Parallel structure fractional repetitive controller (PSFRC)

From Fig. 3.7, $G_i(z)$ can be obtained as

$$G_i(z) = \frac{e^{j(i\times 2\pi\delta/n)}z^{-N^*}}{1 - e^{j(i\times 2\pi\delta/n)}z^{-N^*}} \tag{3.3}$$

where $N^* = round(N/n)$ with $N = f_s/f_0$, and the poles of (3.3) are located at

$$j\omega = (nk + i \times \delta)j\omega_0^* \tag{3.4}$$

with $k = 0, 1, 2...$ and $\omega_0^* = \frac{2\pi f_s}{nN^*}$.

Set the correction factor δ as

$$\delta = \frac{\omega_0 n N^*}{2\pi f_s} = \frac{nN^*}{N}, \tag{3.5}$$

and substitute (3.5) into (3.4), it can be obtained that (3.3) has a pole at $j\omega = ij\omega_0$ when $k = 0$. Adding the output of all the branches in Fig. 3.7 together, PSFRC locates poles accurately at fundamental frequency and each dominant odd-harmonic

frequency under nth order at $ij\omega_0$. Besides, PSFRC improves the gains for higher-order harmonics to further guarantee tracking performance. This is realized by shifting all the poles at $(nk+i)j\omega_0^*$ $(k \geq 1)$ to $(nk+i\times\delta)j\omega_0^*$ which are closer to fundamental harmonic frequencies.

The control gains of CRC and PSFRC with $n = 10$ are shown in Fig. 3.8. Fig. 3.8 shows that PSFRC achieves very high gains at fundamental frequency and dominant low-order harmonics under 10th order. The gains of PSFRC for high-order harmonics are also larger than that of CRC.

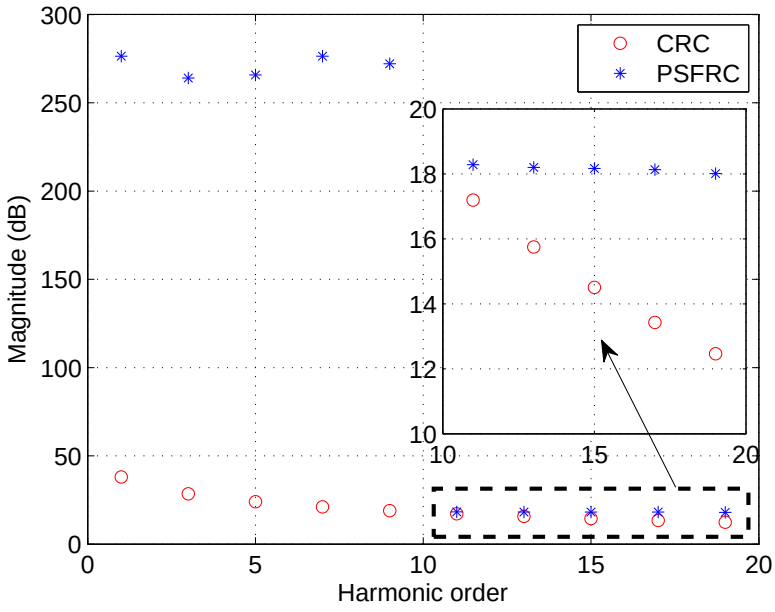


Figure 3.8: Gain comparison between CRC and PSFRC (n=10) at odd harmonics ($f_0 = 60Hz$, $f_s = 10kHz$)

Remark 3.1. *The PSFRC in Fig. 3.7 provides a universal model for different RCs. For example, PSFRC becomes conventional repetitive control by setting $\delta = 1$, $n = 1$; it can also be used as an odd harmonic repetitive controller when $\delta = 1$, $n = 2$; and PSFRC provides a parallel structure odd harmonic repetitive control scheme with $\delta = 1$, $n = 4$.*

Remark 3.2. *The main difference between PSFRC and existing high-order RC or FIR based fractional RC is that PSFRC does not approximate fractional order of delay units for low order harmonics but locates poles accurately for them. Because it has been shown in section 3.2.1 that low-order harmonics dominate the THD of CRC controlled output in fractional cases, it is better to emphasize the improvement of compensations on them. Besides, PSFRC offers a very simple way to address this adaptive problem without the complex interpolations in FIR based fractional RCs. On the other hand, different from the position-domain design as in SRC, PSFRC is based on fixed sampling frequency which is consistent with CRC. But it should be noted that since more than one branches are needed in PSFRC, the computational burden increases. To reduce computational burden and improve transient performance, unnecessary branches in PSFRC can be eliminated in particular applications. For example, in three-phase AC systems, the branches which compensate harmonics other than* $6k \pm 1$*th order can be abandoned.*

Fig. 3.7 shows an ideal PSFRC controller, but high frequency components and system uncertainty may lead the system to instability due to the infinite poles on the unit circle. So filters are usually introduced to improve the robustness of the whole systems in real applications. Fig. 3.9 shows a modified PSFRC controlled system, where $R(z)$ is the reference signal; $e(z)$ is the tracking error; $y(z)$ is the output; $d(z)$ is the disturbance; $u_r(z)$ is the control signal from PSFRC; $u(z)$ is the overall control signal; $G_c(z)$ is a state feedback controller and $G_p(z)$ is the control plant. $Q(z)$ and $G_f(z)$ are filters.

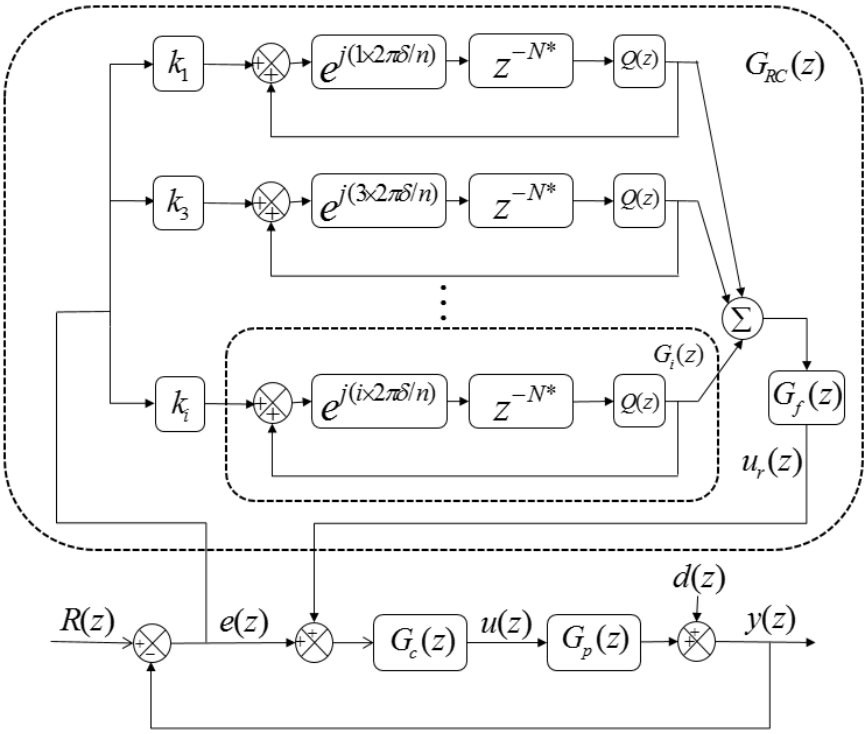


Figure 3.9: Digital modified PSFRC system

3.2.3 Stability analysis

In Fig. 3.9, $G_c(z)$ is designed to stabilize the closed-loop system without the plug-in PSFRC controller. This means that all the poles of the following transfer function must fall inside the unit circle:

$$H(z) = \frac{G_c(z)G_p(z)}{1 + G_c(z)G_p(z)}. \tag{3.6}$$

After plugging the PSFRC controller into the system, output of the overall system can be derived from Fig. 3.9 as

$$\frac{y(z)}{R(z)} = \frac{[1 + G_{RC}(z)]H(z)}{1 + G_{RC}(z)H(z)} \tag{3.7}$$

and

$$\frac{y(z)}{d(z)} = \frac{1}{[1 + G_{RC}(z)H(z)][1 + G_c(z)G_p(z)]} \tag{3.8}$$

where $G_{RC}(z) = \sum k_i G_f(z) G_i(z)$ with

$$G_i(z) = \frac{e^{j(i\times 2\pi\delta/n)} Q(z) z^{-N^*}}{1 - e^{j(i\times 2\pi\delta/n)} Q(z) z^{-N^*}}. \tag{3.9}$$

$Q(z)$ is a zero phase moving average filter with the form in (3.10). It is introduced to suppress the unexpected high-frequency repetitive disturbance.

$$Q(z) = \frac{\left(\sum_{i=0}^{j} a_i z^i + \sum_{i=1}^{j} a_i z^{-i}\right)}{2\sum_{i=1}^{j} a_i + a_0} \tag{3.10}$$

In order to guarantee the stability of the overall plug-in PSFRC system, all the poles of (3.7) and (3.8) must be inside the unit circle. From (3.6), the roots of $1 + G_c(z)G_p(z) = 0$ can be located inside the unit circle by choosing appropriate $G_c(z)$. Therefore, it must be true that

$$1 + G_{RC}(z)H(z) \neq 0 \quad for \quad \forall\,|z| \; > 1 \tag{3.11}$$

or

$$\sum k_i G_i(z) \neq -\frac{1}{G_f(z)H(z)} \quad for \quad \forall\,|z| \; > 1. \tag{3.12}$$

Without loss of generality, $H(z)$ is rewritten to

$$H(z) = \frac{z^{-d}B(z)}{A(z)} \tag{3.13}$$

where d represents the delay of the system, then $G_f(z)$ is designed as [59]

$$G_f(z) = \frac{z^d A(z) B^-(z^{-1})}{B^+(z) b} \tag{3.14}$$

where B^- represents the noncancellable portion of B; B^+ represents the cancellable portion, and $b \geq \mid B^-(z) \mid^2$ for all z.

From (3.14), it can be derived that

$$0 < G_f(z)H(z) = \frac{B^-(z^{-1})B^-(z)}{b} \leq 1, \tag{3.15}$$

and (3.15) leads to that

$$-\frac{1}{G_f(z)H(z)} \leq -1. \tag{3.16}$$

It has been proved in [211] that

$$Re[G_i(z)] \geq -\frac{1}{2} \quad for \quad \forall |z| \ > 1. \tag{3.17}$$

From (3.17), and if k_i satisfy $0 < \sum k_i < 2$, then

$$Re[\sum k_i G_i(z)] > -1 \tag{3.18}$$

Inequality (3.16), along with (3.18), guarantees that (3.12) holds, which means that no poles of the overall system are outside the unit circle.

From the above analysis, the closed-loop PSFRC system as shown in Fig. 3.9 is asymptotically stable if the following conditions are satisfied.

1. the closed-loop system without plug-in PSFRC controller is stable, or all the roots of $1 + G_c(z)G_p(z) = 0$ are inside the unit circle;

2. $G_f(z)$ is chosen as in the format of (3.14) and satisfies (3.16);

3. $0 < \sum k_i < 2$.

Table 3.1: System parameters of single-phase PWM inverter (PSFRC)

Meaning and symbol	*Value*
DC bus voltage E	400V
Output filter inductance L_f	7mH
Output filter capacitance C_f	50uF
Resistor load R	20Ω
Rectifier load L_r	5mH
Rectifier load C_r	1100uF
Rectifier load R_r	30Ω
Fundamental frequency f_0	60Hz
Sampling frequency f_s	10kHz
Switching frequency	10kHz
Amplitude of reference R	240V

3.3 Case study: PSFRC on a single-phase PWM inverter system

Fig. 3.10 shows a plug-in PSFRC controlled single-phase DC/AC inverter system where E is the DC bus voltage; L and C are inductor-capacitor (LC) filter; i_o is the output current; v_o is the output voltage; The load bank consists of two kinds of load: (a) resister load with R and (b) rectifier load with L_r, C_r and R_r. The detailed system parameters are listed in Table 3.1. The output LC filter is designed to eliminate high-order harmonics by circuit. It is modelled in $G_p(z)$ and can be compensated by the design of feedback controller $G_c(z)$ from (3.6). But an appropriate cut-off frequency, which is usually chosen lower than $0.1f_s$, should be selected. Because n is set to be 10 for PSFRC in the following experiments, the control gains for harmonics higher than 10th order is relatively low. Thus, it is better to design the cut-off frequency of the LC filter around $10f_0$. In this case, the cut-off frequency 850 Hz of output LC filter allows the inverter to efficiently produce output voltages of frequency 60Hz and eliminate high-order harmonics.

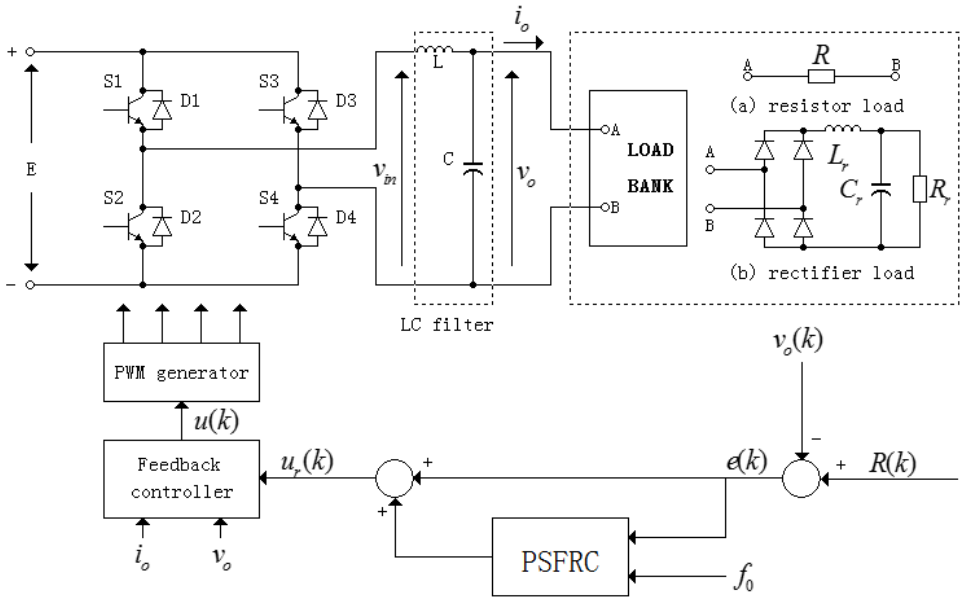


Figure 3.10: Plug-in PSFRC controlled single-phase inverter system

Experiments in this section are carried out on the PWM DC/AC experimental system as shown in Fig. 3.10. The controller is programmed in Matlab/Simulink and realized in real time by a Dspase1104 at 10kHz.

3.3.1 System modelling and controller design

The discrete single-phase PWM DC/AC inverter system used to testify PSFRC in this section is the same as in section 2.3.1. Based on the former modelling process, the transfer function of the state-feedback closed-loop system without plug-in PSSRC can be derived as

$$H(z) = \frac{m_1 z + m_2}{z^2 + p_1 z + p_2} \tag{3.19}$$

where

$p_1 = -(\varphi_{11} - g_1 k_1 + \varphi_{22} - g_2 k_2)$,

$p_2 = (\varphi_{11} - g_1 k_1)(\varphi_{22} - g_2 k_2) - (\varphi_{12} - g_1 k_2)(\varphi_{21} - g_2 k_1)$,

$m_1 = g_1 h$ and $m_2 = (\varphi_{12} - g_1 k_2) g_2 h - (\varphi_{22} - g_2 k_2) g_1 h$, and the poles of (3.19) can

be arbitrarily assigned by designing the state-feedback controller $G_c(z)$.

To stabilize the closed-loop system without PSFRC, the state feedback controller in section 2.3.1 is adopted.

$$u(k) = \begin{bmatrix} -1.6255 & -1.0224 \times 10^{-3} \end{bmatrix} x(k) + 2.0 v_{ref}(k), \tag{3.20}$$

and $H(z)$ as in (3.19) can be calculated as

$$H(z) = \frac{0.0029z - 0.187}{z^2 - 1.048z + 0.2022}. \tag{3.21}$$

It is quite straightforward to verify that all the poles of $H(z)$ are inside the unit circle, which satisfies stability criterion 1.

From (3.14), filter $G_f(z)$ is chosen as

$$G_f(z) = \frac{z^6}{H(z)} \tag{3.22}$$

where z^6 is used to compensate the phase lag and unmodelled delay of the system. The number of delays used is determined from the experiment by trial and error.

From Table 3.1, it can be calculated that $N = f_s/f_0 = 166.67$, so the CRC is implemented with rounding N to 167 and the learning gain k_r for CRC is set as 0.0014. PSFRC locates poles at odd-harmonics under 10th order by setting $n = 10$. Because the learning gain k_r and k_i affect the convergence rate of repetitive control, the learning gain of PSFRC is set as $k_i = 2k_r/n$ which makes $\sum k_i = k_r$. It should be noted that $N^* = round(f_s/f_0/n)$ and δ in (4.2) for PSFRC controller is calculated in real time using the information from the reference f_0. Detailed parameters of the controllers are listed in Table 3.2.

Table 3.2: Parameters of SFB, PSFRC and CRC controller

Controller	*Parameters*
SFB	$k_1 = 1.6255$, $k_2 = 1.0224 \times 10^{-3}$, $h = 2.0$
PSFRC	$n = 10$, $k_i = 0.0014/5$, $Q(z) = 0.25z^{-1} + 0.5 + 0.25z$
CRC	$k_r = 0.0014$, $Q(z) = 0.25z^{-1} + 0.5 + 0.25z$

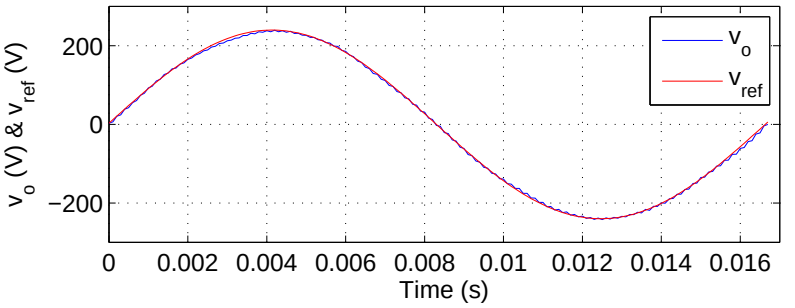
3.3.2 Experimental results

The capabilities of the proposed PSFRC to track periodic reference and to reject harmonics in fractional case are verified in this section. Fig. 3.11, Fig. 3.12 and Fig. 3.13 show the tracking performance of PSFRC and CRC plugging into a closed-loop single-phase inverter system under different loads.

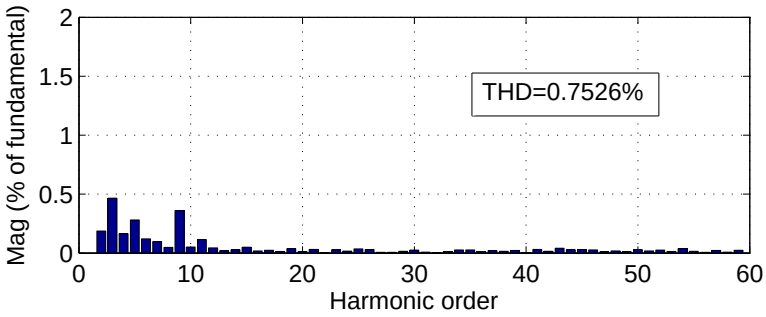
Fig. 3.11(a) and Fig. 3.11(c) show the output voltage v_o and reference v_{ref} of PSFRC and CRC under no load. Fig. 3.11(b) and Fig. 3.11(d) are the spectrums of corresponding v_o. From Fig. 3.11(a), PSFRC achieves outstanding tracking performance with very low THD of 0.7526% under no load. Comparing Fig. 3.11(b) with Fig. 3.11(d), it can be seen that the dominant harmonics in CRC controlled output are effectively suppressed by PSFRC. In Fig. 3.11(d), the magnitude of all the odd-harmonics under 10th order are higher than 1%, while these components are suppressed to under 0.5% by PSFRC as shown in Fig. 3.11(a).

Moreover, PSFRC is testified under resistor load. Fig. 3.12 are the experimental results of both PSFRC and CRC. Fig. 3.12(d) shows that the THD of CRC controlled output voltage is dominated by low-order odd harmonics and THD= 2.6585%. Compared with Fig. 3.12(d), Fig. 3.12(b) shows that all harmonics in PSFRC controlled output voltage are suppressed to lower than 0.5% with THD= 1.0830%.

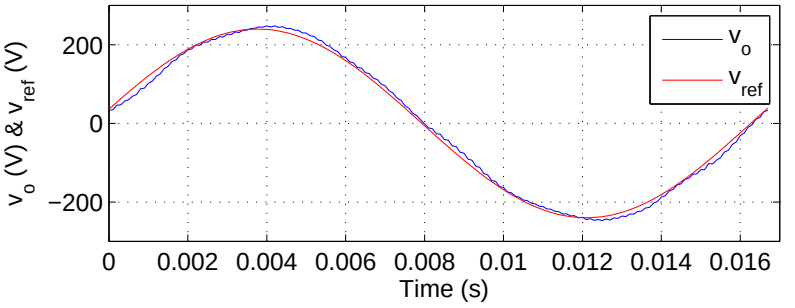
Similar results are also shown in Fig. 3.13 to illustrate the effectiveness of PSFRC under rectifier load. Fig. 3.13 shows that PSFRC achieves very low THD= 1.5249%



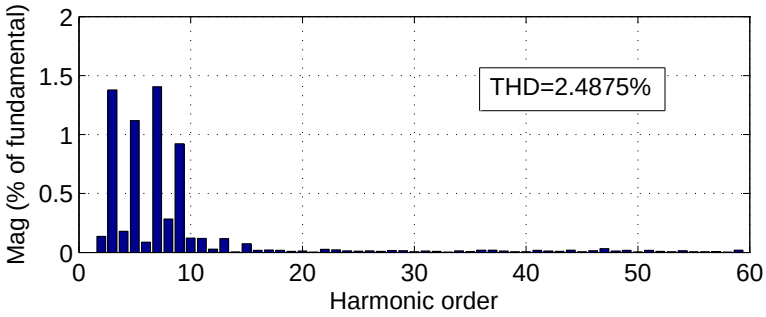
(a)



(b)

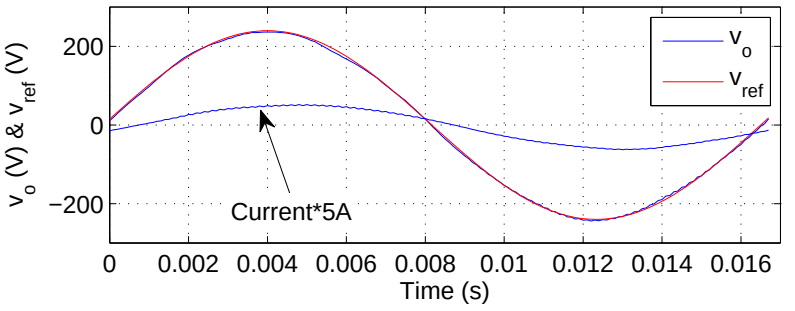


(c)

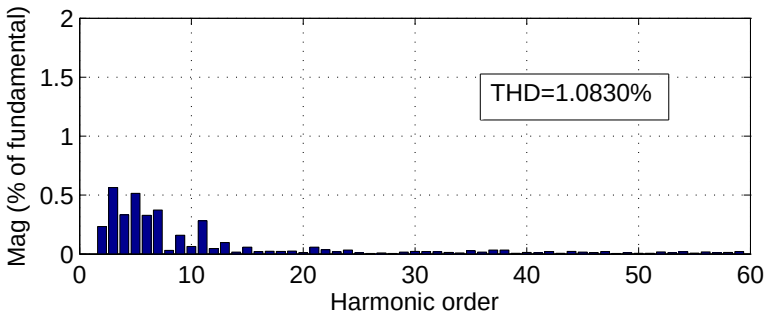


(d)

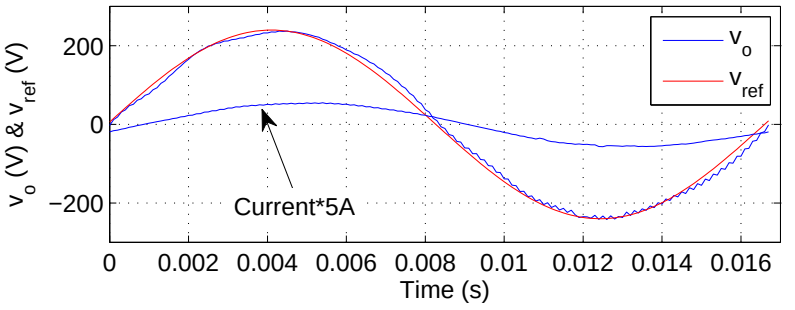
Figure 3.11: Static tracking performance of PSFRC and CRC under no load: (a) PSFRC controlled output and reference; (b) spectrum of the output in (a); (c) CRC controlled output and reference; (d) spectrum of the output in (c).



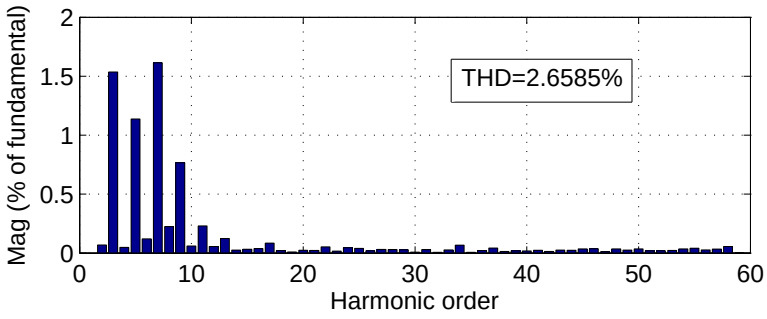
(a)



(b)



(c)



(d)

Figure 3.12: Static tracking performance of PSFRC and CRC under resistor load: (a) PSFRC controlled output and reference; (b) spectrum of the output in (a); (c) CRC controlled output and reference; (d) spectrum of the output in (c).

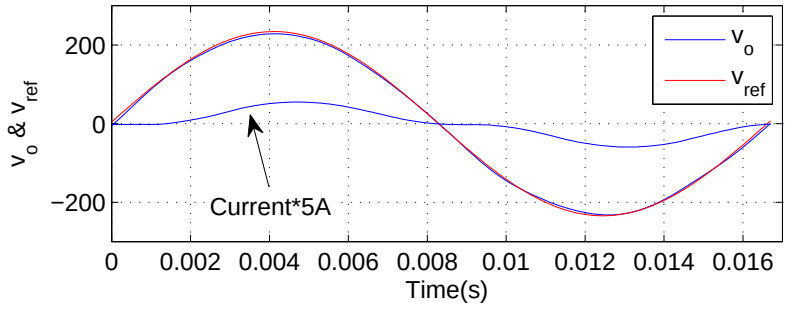
under rectifier load, while THD of CRC controlled output is much higher with THD= 3.3463%.

Besides, Fig. 3.14(a) and Fig. 3.14(b) show the convergence of tracking error with PSFRC and CRC, respectively. Both PSFRC and CRC controllers are plugged into the SFB closed-loop system at $t = 1.0s$. From Fig. 3.14(a), PSFRC converges fast within $0.1s$ (within 6 cycles), while CRC takes about $0.2s$ to converge which is twice slower than PSFRC. But compared with PI controller, the convergence time of PSFRC is still relatively longer. Moreover, it is shown in Fig. 3.14 that PSFRC offers better static tracking performance than CRC. After convergence, the root mean square (RMS) value of tracking error in PSFRC is $1.6033V$, while the RMS static tracking error in CRC is $5.6325V$.

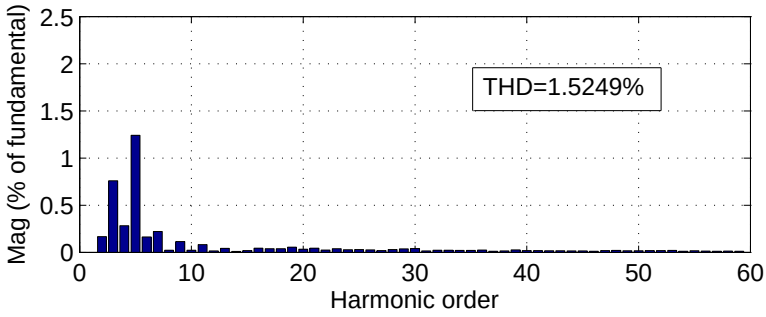
Detailed comparisons of tracking performance between PSFRC and CRC are listed in Table 3.3. It should be noted that the turnaround time of the proposed PSFRC is longer than that of CRC. This is because multiple branches are added. But the number of delay units required in PSFRC is reduced as many as possible to release computational burden. Besides, unnecessary branches in PSFRC can be eliminated to further improve transient performance and reduce turnaround time as stated in Remark 3.2.

The transient performance of PSFRC when step load changes occur is shown in Fig. 3.15. Step load change between resistor load and no load are applied to the system. It can be seen from Fig. 3.15 that PSFRC can handle step load change well and drive the output voltage back within 6 cycles.

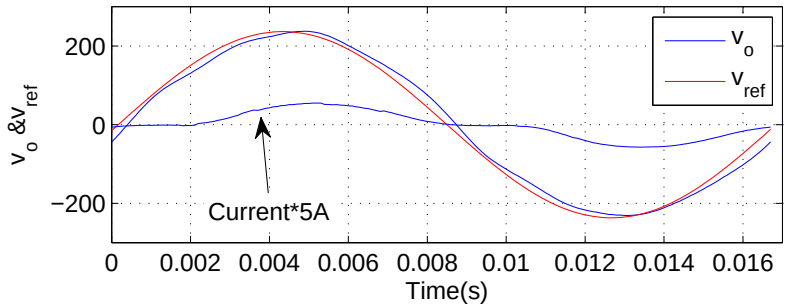
Moreover, the transient performance of PSFRC, when the frequency of the reference (or fundamental frequency) f_0 changes, is investigated. In this case, f_0 is changed from $60Hz$ to $40Hz$. The experimental results are shown in Fig. 3.16. Fig. 3.16(a) shows the output voltage v_0 and current i_0, and Fig. 3.16(b) shows the tracking error of v_0. From Fig. 3.16(b), PSFRC can drive the output back to the



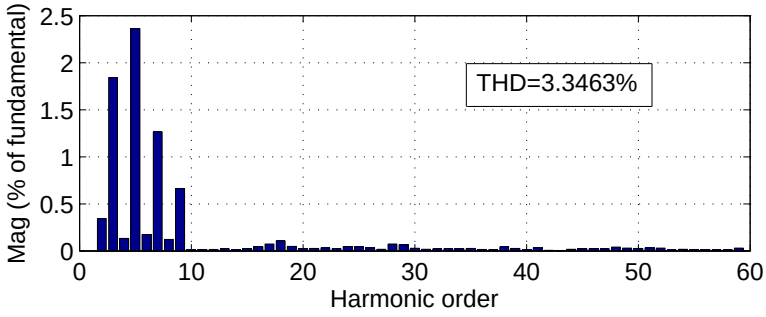
(a)



(b)

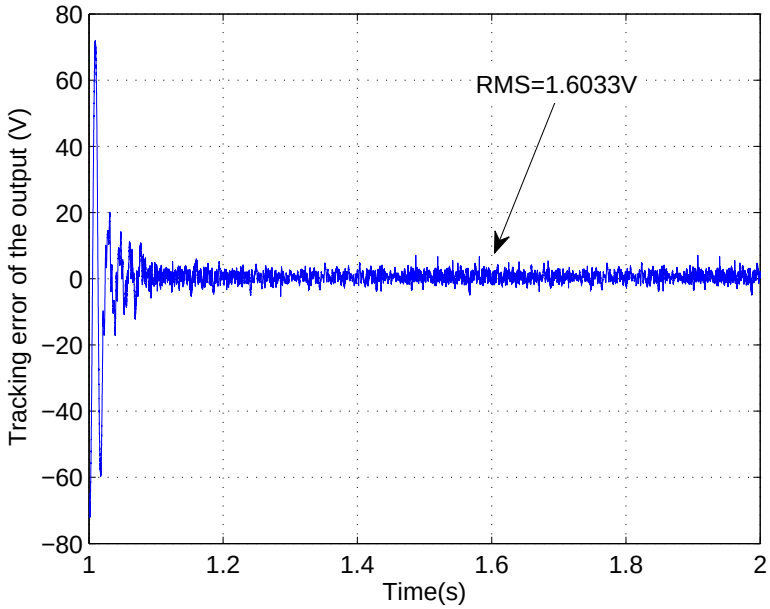


(c)

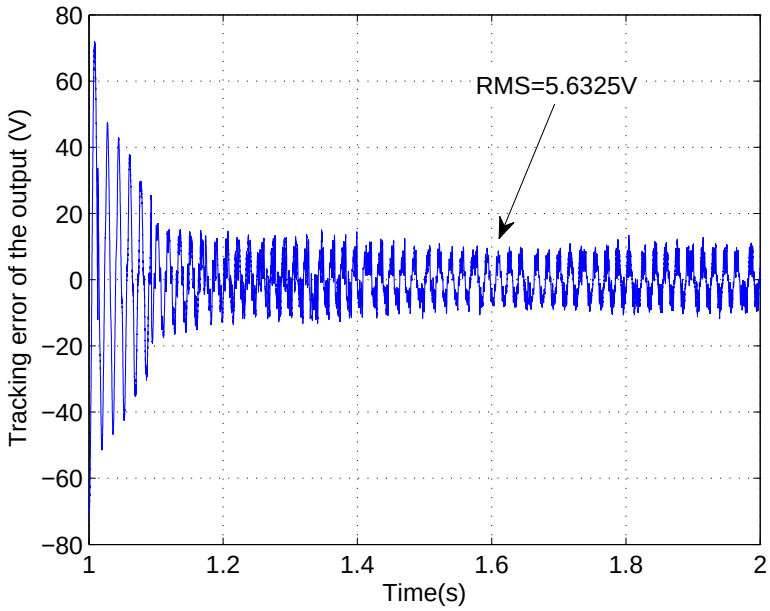


(d)

Figure 3.13: Static tracking performance of PSFRC and CRC under rectifier load: (a) PSFRC controlled output and reference; (b) spectrum of the output in (a); (c) CRC controlled output and reference; (d) spectrum of the output in (c).

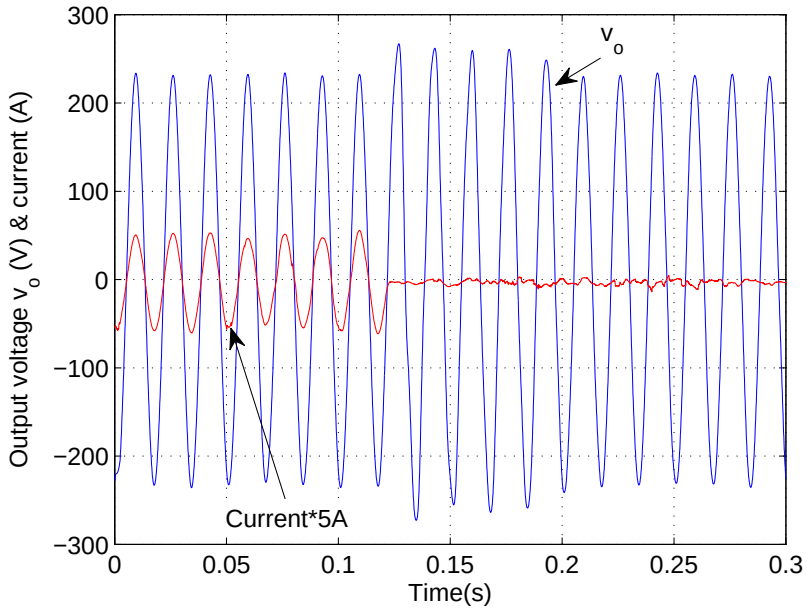


(a)

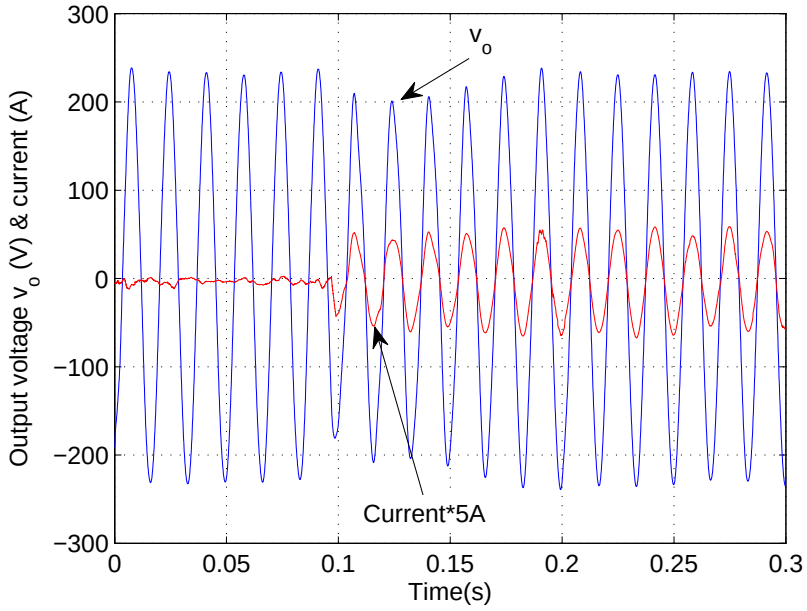


(b)

Figure 3.14: Convergence of error in PSFRC and CRC: (a) PSFRC; (b) CRC.



(a)



(b)

Figure 3.15: Transient performance of PSFRC under load change: (a) from resistor load to no load; (b) from no load to resistor load.

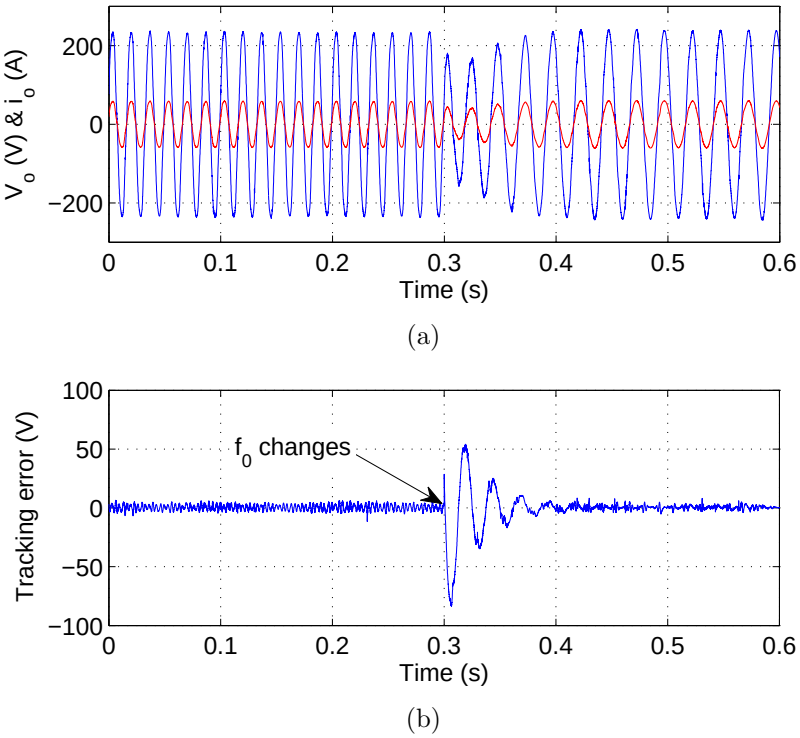


Figure 3.16: Transient performance of PSFRC under frequency change: (a) output voltage and current; (b) tracking error.

new reference fast within $0.1s$ when the fundamental frequency changes. On the other hand, CRC's response to this change of f_0 is shown in Fig. 3.17. It shows that CRC needs much more time to track the new reference.

3.4 Summary

This chapter proposes a novel parallel structure fractional repetitive control scheme for PWM inverters. Using a correction factor and adopting parallel structure, PSFRC moves all poles of repetitive controller closer to harmonic frequencies in fractional cases. Most importantly, it locates poles precisely at targeted harmonic frequencies. As a result, the proposed control scheme achieves infinite high gains for targeted harmonics and improves the control gains for others. Compared with CRC, PSFRC offers both better tracking performance and faster transient performance in fraction-

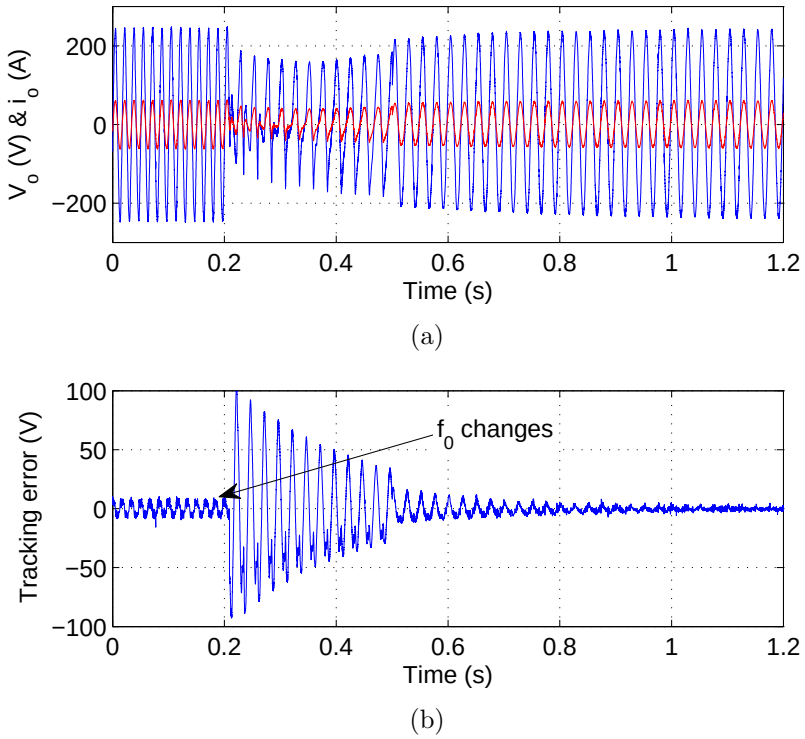


Figure 3.17: Transient performance of CRC under frequency change: (a) output voltage and current; (b) tracking error.

Table 3.3: Performance comparison between PSFRC and CRC

Items	*PSFRC*	*CRC*
THD under no load	0.7526%	2.4875%
THD under resistor load	1.0830%	2.6585%
THD under rectifier load	1.5249%	3.3463%
Tracking error (RMS)	1.6033V	5.6325V
Convergence time	About 0.1s	About 0.2s
Number of delay units used	85	167
Turnaround time in DSP	62.5 μ s	45.3 μ s

al cases. Furthermore, the number of delay units required is reduced to half of that in CRC. Stability of this control scheme is proved. Experiments on a single-phase PWM inverter are carried out and the experimental results verify the advantages of PSFRC.

Chapter 4

Grid Simulator using PSFRC

4.1 Introduction

Chapter 3 introduces the basic idea of PSFRC and proves its stability. This chapter will further investigate the advantages of PSFRC compared with other types of RCs. A grid simulator based on PSFRC to address power quality problem is developed in this chapter.

Power quality (PQ) issues have attracted a lot attention from researchers in recent years to provide high-quality source voltages for different electrical equipment [214–221]. But some PQ uncertainties, especially those caused by short circuit, earth faults, transformer energizing, motor starting and so on, are hardly predictable in real power distribution systems. So grid simulators are needed to emulate various operation scenarios to test electrical equipment. Several grid simulators have been proposed so far: [45] proposes a programmable AC source to generate AC voltages with various frequencies; [222] simulates voltage sag, swell and flicker with series injected inverter; [223] presents a single phase grid simulator controlled by proportional-integral (PI) controllers; [224] presents an algorithm for transient simulation of power grids using waveform relaxation (WR) technique; [225] designs

a grid simulator using resonant controllers; [226] designs grid-simulator controlled by basic PI controller for a transient analysis of grid-connected renewable energy system; [227] proposes a grid simulator to evaluate control performance of grid-connected inverters; [228] presents a high-performance controller handling nonlinear load; and [229] proposes a multiple-loop control system in 400Hz inverter power supply based on unipolar sinusoidal pulse width modulation(SPWM). However, most of these grid simulators are designed for particular perturbations only. Besides, grid simulators based on three-phase four-leg converters are also proposed [230]. Four-leg inverters avoid bulky capacitors or transformers for three-phase four-wire systems and have been investigated widely recently [231–233]. But the control of four-leg converters are more complex compared with conventional three-leg converters. Detailed comparisons between the control of four-leg and three-leg inverters are made in [234].

Repetitive control is a control scheme designed for tracking repetitive references. Besides, digital RC controllers have simple structures formulated by N delay units (N is the number of samples in one period of the repetitive reference), and can be applied in a plug-in style. Therefore, RC is one of the most promising control schemes for grid simulators.

However, limitations on transient response, selectivity and flexibility limit the performance of conventional repetitive control (CRC) in grid simulators. Besides, conventional repetitive controllers compensate all the system's harmonics equivalently. But it is better for grid simulators to achieve independent tunings and controls for different harmonic components.

In this case, parallel structure fractional repetitive control (PSFRC) scheme provides a solution to address the above problems faced by conventional repetitive control. Remind the analysis in chapter 3, PSFRC locates poles at fundamental and dominant harmonic frequencies in fractional case. Besides, it adopts parallel

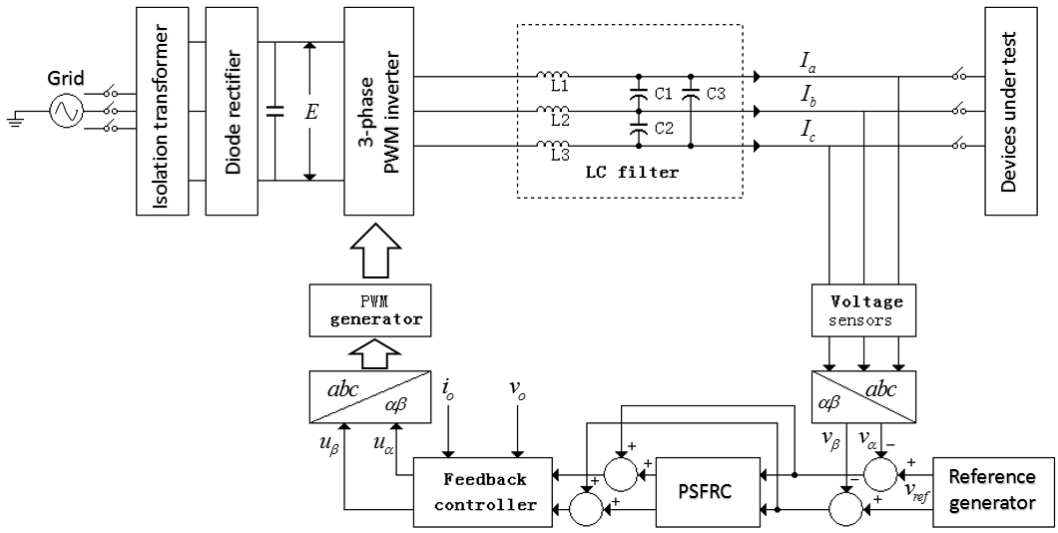


Figure 4.1: PSFRC based grid simulator

structure to separate the tunings and controls of different harmonics. Therefore, PSFRC-controlled grid simulator offers better capability to generate accurate outputs with wanted harmonic distortions. Moreover, the number of delay units required in PSFRC can be reduced to at least half of that in CRC. Transient response is also improved for the simulations of sudden voltage drops and voltage flickers.

Stability analysis and detailed design of controllers are provided in this chapter. A reference generator is designed to provide various references. Moreover, experiments are carried out to verify the effectiveness of the proposed grid simulator.

The remainder of this chapter is organized as follows. In section 4.2, a new design of grid simulator is proposed based on PSFRC. Its topology structure, controller design, stability analysis and reference generator, are described in detail. Section 4.3 provides experimental results to illustrate the effectiveness of this scheme. Section 4.4 concludes this chapter.

4.2 Grid simulator system

The proposed grid simulator is shown in Fig. 4.1, where a three-phase H-bridge PWM inverter with LC type output filters is used to generate voltages with different types of disturbances and distortions. The control scheme of the grid simulator consists of two parts: a state feedback controller and a parallel structure fractional repetitive controller (PSFRC). Besides, a reference generator is designed to provide reference for the whole system. Details of the proposed grid simulator are presented in this section.

4.2.1 System modeling and analysis

The proposed grid simulator can also be described by Fig. 4.2, where $R(z)$ is the reference from the reference generator; $e(z)$ is the tracking error; $u(z)$ is the control signal; $d(z)$ is the disturbance; $y(z)$ is the output; $G_{RC}(z)$, $G_c(z)$ and $G_p(z)$ are the transfer function of the PSFRC controller, the state feedback controller and the control plant, respectively.

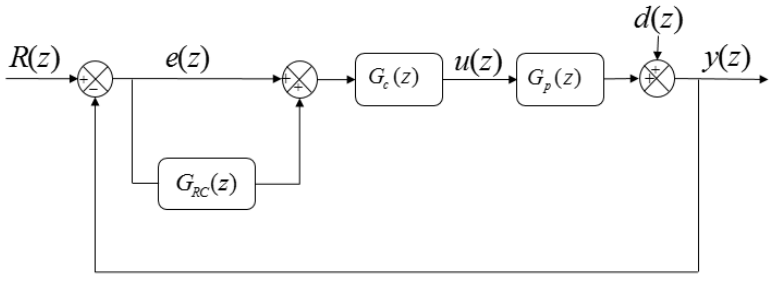


Figure 4.2: Digital PSFRC based grid simulator

In this grid simulator, the three-phase PWM inverter, i.e. $G_p(z)$, is decoupled into two identical single-phase systems as follows:

$$\begin{cases} x(k+1) = Ax(k) + Bu(k) \\ y(k) = Cx(k) \end{cases} \tag{4.1}$$

where $x(k) = \begin{bmatrix} v_\alpha(k) \\ i_\alpha(k) \end{bmatrix}$ or $\begin{bmatrix} v_\beta(k) \\ i_\beta(k) \end{bmatrix}$, $u(k) = u_\alpha(k)$ or $u_\beta(k)$, $y(k) = v_o(k)$,

$A = \begin{bmatrix} \varphi_{11} & \varphi_{12} \\ \varphi_{21} & \varphi_{22} \end{bmatrix}$, $B = \begin{bmatrix} g_1 \\ g_2 \end{bmatrix}$ with

$\varphi_{11} = 1 - (1/RCf_s) + (1/2R^2C^2f_s^2) - (1/6LCf_s^2)$,

$\varphi_{12} = (1/3Cf_s) - (1/6RC^2f_s^2)$,

$\varphi_{21} = -(1/Lf_s) + (1/2LCRf_s^2)$,

$\varphi_{22} = 1 - (1/6LCf_s^2)$,

$g_1 = E/6LCf_s^2$ and $g_2 = E/Lf_s$.

Remark 4.1. *It should be noted here that the three-phase inverter is reduced to two identical single-phase ones. Therefore, all the designs in this section are also applicable to single phase grid simulators.*

The structure of the proposed PSFRC controller is shown in Fig. 4.3, where $e(z)$ is the tracking error; k_i is the learning gain with $i = 1, 3, 5...$ and $i \leq n$; $N = f_s/f_0$ with $f_0 = \omega_0/2\pi$ being the fundamental frequency and f_s being the sampling rate; $u_r(z)$ is the control signal from PSFRC; $G_i(z)$ is the transfer function of the ith branch; $G_{RC}(z)$ is the transfer function of the overall PSFRC; $Q(z)$, $G_f(z)$ are both filters to improve the robustness of the whole system and δ is a correction factor defined by

$$\delta = \frac{\omega_0 n N^*}{2\pi f_s} = \frac{nN^*}{N}, \tag{4.2}$$

where $N^* = round(N/n)$.

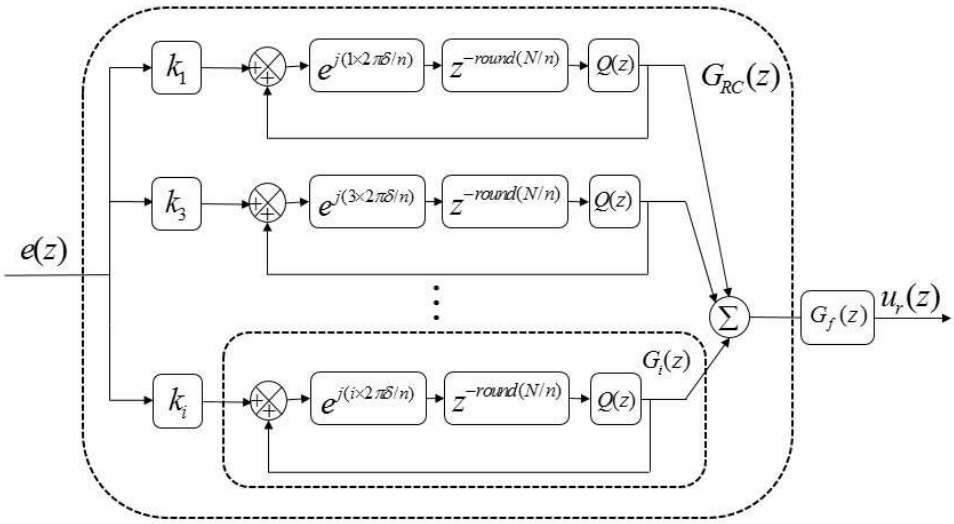


Figure 4.3: Parallel structure fractional repetitive controller (PSFRC)

From Fig. 4.3, $G_i(z)$ can be obtained as

$$G_i(z) = \frac{e^{j(i\times 2\pi\delta/n)}Q(z)z^{-N^*}}{1 - e^{j(i\times 2\pi\delta/n)}Q(z)z^{-N^*}}. \tag{4.3}$$

In order to explore the properties of PSFRC controller in grid simulator, $Q(z)$ is regarded as an ideal zero-phase filter with uniform gain, then it can be derived from (4.3) that the poles of $G_i(z)$ are located at

$$j\omega = (nk + i \times \delta)j\omega_0^*, \tag{4.4}$$

where $k = 0, 1, 2...$ and $\omega_0^* = \frac{2\pi f_s}{nN^*}$.

The properties of PSFRC controller in grid simulator application under different scenarios can be analyzed based on (4.4) as follows:

Normal cases where N^* is an integer

In this case, $\delta = 1$, $N^* = N$ and $\omega_0^* = \omega_0$. From (4.4), the poles of PSFRC's ith branch are located at $j\omega = (nk + i)j\omega_0$. Adding the output of all the branches in Fig. 4.3 together, PSFRC locates poles at $j\omega = (2k + 1)j\omega_0$, which correspond to the system's fundamental frequency and all the odd-harmonic frequencies. Because odd harmonics dominate the THD in PWM AC systems, these poles guarantee zero-error tracking of periodic reference with its fundamental frequency being f_0.

Fractional cases where N^* is not an integer

In this case, $\delta \neq 1$ and $\omega_0^* \neq \omega_0$. Substituting (4.2) into (4.4), it can be obtained that (4.3) has a pole at $j\omega = ij\omega_0$ when $k = 0$. Adding the output of all the branches in Fig. 4.3 together, PSFRC locates poles accurately at fundamental frequency and each dominant odd-harmonic under nth order at $ij\omega_0$. Besides, since $\delta \neq 1$, (4.4) indicates that PSFRC shifts all the poles at $(nk + i)j\omega_0^*$ (poles of CRC) to $(nk + i \times \delta)j\omega_0^*$ $(k \geq 1)$ which are closer to harmonic frequencies. It means that the gains for higher order harmonics are improved. Therefore, PSFRC proposed in this chapter achieves higher gains compared with CRC in fractional cases.

Remark 4.2. *PSFRC controller shown in Fig. 4.3 provides a flexible and selective repetitive control scheme for grid simulators. The correction factor δ offers a very simple and adaptive way to provide high-accuracy tracking performance in both normal and fractional cases from the above analysis. Each branch of PSFRC compensates one harmonic group at $j\omega = (nk + i)j\omega_0$. So it is convenient to tune different harmonics independently. Moreover, the unnecessary branches can be removed to further improve transient performance and to reduce computational burden. For example, the branches with k_3 and k_9 are not needed for three-phase AC systems.*

4.2.2 Controller design

As shown in Fig. 4.2, a plug-in PSFRC controller combined with a state-feedback controller is adopted in this grid simulator. The state feedback controller, $G_c(z)$, is designed to stabilize and compensate the PWM inverter. The plug-in PSFRC controller, $G_{RC}(z)$, offers precise tracking performance and harmonic elimination.

In this chapter, a state-feedback controller as in (4.5) is adopted,

$$u(k) = \begin{bmatrix} -k_1 & -k_2 \end{bmatrix} x(k) + hv_{ref}(k). \tag{4.5}$$

Substituting (4.5) into (4.1), state equation of the system with state feedback controller only can be derived as

$$\begin{aligned} x(k+1) = &\begin{bmatrix} \varphi_{11} - k_1 g_1 & \varphi_{12} - k_1 g_2 \\ \varphi_{21} - k_2 g_1 & \varphi_{22} - k_2 g_2 \end{bmatrix} x(k) \\ &+ \begin{bmatrix} g_1 h \\ g_2 h \end{bmatrix} v_{ref}(k) \end{aligned} . \tag{4.6}$$

Finding the transfer function of state equation (4.6) in z-domain, the system without the plug-in PSFRC can be expressed by

$$H(z) = \frac{G_c(z)G_p(z)}{1 + G_c(z)G_p(z)} = \frac{m_1 z + m_2}{z^2 + p_1 z + p_2} \tag{4.7}$$

where

$p_1 = -(\varphi_{11} - g_1 k_1 + \varphi_{22} - g_2 k_2)$,

$p_2 = (\varphi_{11} - g_1 k_1)(\varphi_{22} - g_2 k_2) - (\varphi_{12} - g_1 k_2)(\varphi_{21} - g_2 k_1)$,

$m_1 = g_1 h$ and $m_2 = (\varphi_{12} - g_1 k_2) g_2 h - (\varphi_{22} - g_2 k_2) g_1 h$.

From (4.7), poles of $H(z)$ can be arbitrarily assigned by designing appropriate state-feedback controller.

Table 4.1: System parameters of three phase PWM inverter

Meaning and symbol	*Value*
DC bus voltage E	$500V$
Output filter inductance $L_{1,2,3}$	$7mH$
Output filter capacitance $C_{1,2,3}$	$50uF$
Load resistance R	50Ω
Sampling frequency f_s	$6kHz$
Switching frequency	$6kHz$

Choose $u(k)$ as

$$u(k) = \begin{bmatrix} -0.7 & -0.1 \end{bmatrix} x(k) + 2.0 v_{ref}(k), \tag{4.8}$$

then $H(z)$ can be derived from the detailed parameters listed in Table 4.1 as,

$$H(z) = \frac{1.323z + 1.316}{z^2 - 0.3373z + 0.2713}. \tag{4.9}$$

It is straightforward to verify that all the poles of $H(z)$ are inside the unit circle, which means that the system is stable without plug-in PSFRC controller.

As analyzed in Chapter 3, the sum of k_i, i.e. $\sum k_i$, should fall in the range of $(0, 2)$. It should be noted that k_i can be tuned independently to improve the compensation on dominant harmonics. For three-phase grid simulators, k_5 and k_7 should be larger to guarantee better tracking performance since 5th and 7th order harmonic groups dominant the THD. In this case, the settings of PSFRC are as follows: $k_1 = k_9 = 0.02$, $k_3 = 0.01$, $k_5 = 0.05$, $k_7 = 0.3$, $n = 10$, and the values are determined by trial and error.

4.2.3 Hardware description

The hardware equipments and prototype of the grid simulator are shown in Fig. 4.4 and the system is depicted in Fig. 4.5.

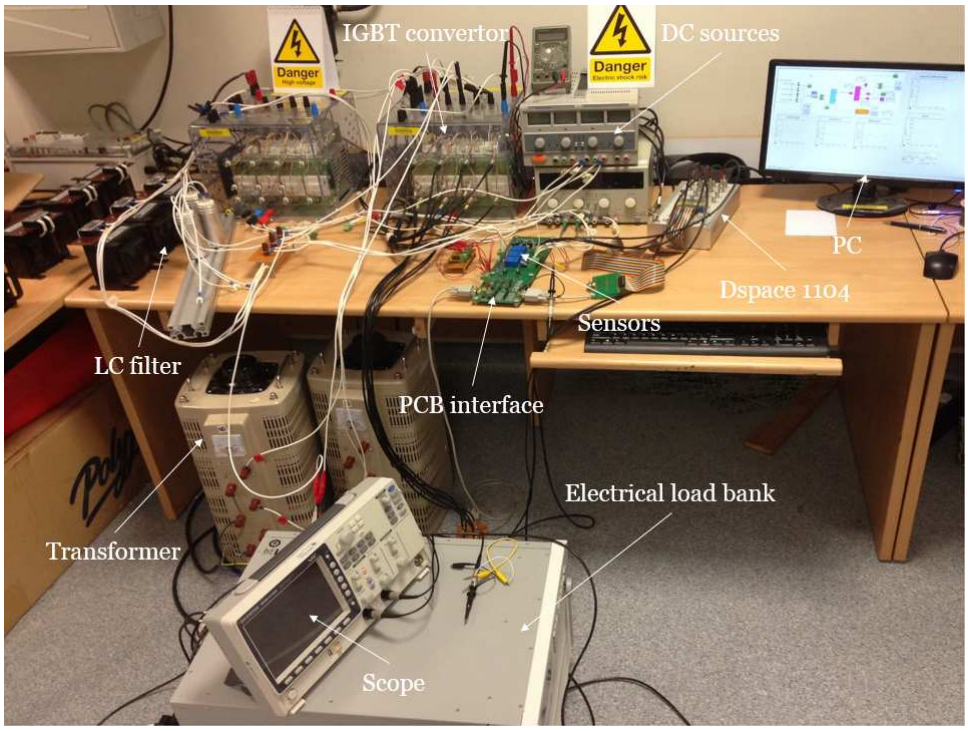


Figure 4.4: Experimental setups

In this grid simulator, the diode rectifier is a SKD 51 rectifier module as shown in Fig. 4.6. This is a three-phase uncontrolled diode rectifier module. This rectifier can be used in single phase configuration also, by leaving one branch of the rectifier disconnected. The filtering capacitors after the rectifier are electrolytic capacitors, their individual value is 2200 μF/400 V, 2 connected in series, 2 in parallel. The equivalent capacitor of the complete DC bus is 1100 μF/800 V.

As shown in Fig. 4.7 , a two-level PWM inverter is connected after the DC bus. The PWM inverter is implemented by three IGBT modules SKM 50 GB 123 D as in Fig. 4.8. Each of these modules is an inverter leg, and is made of 2 IGBTs with

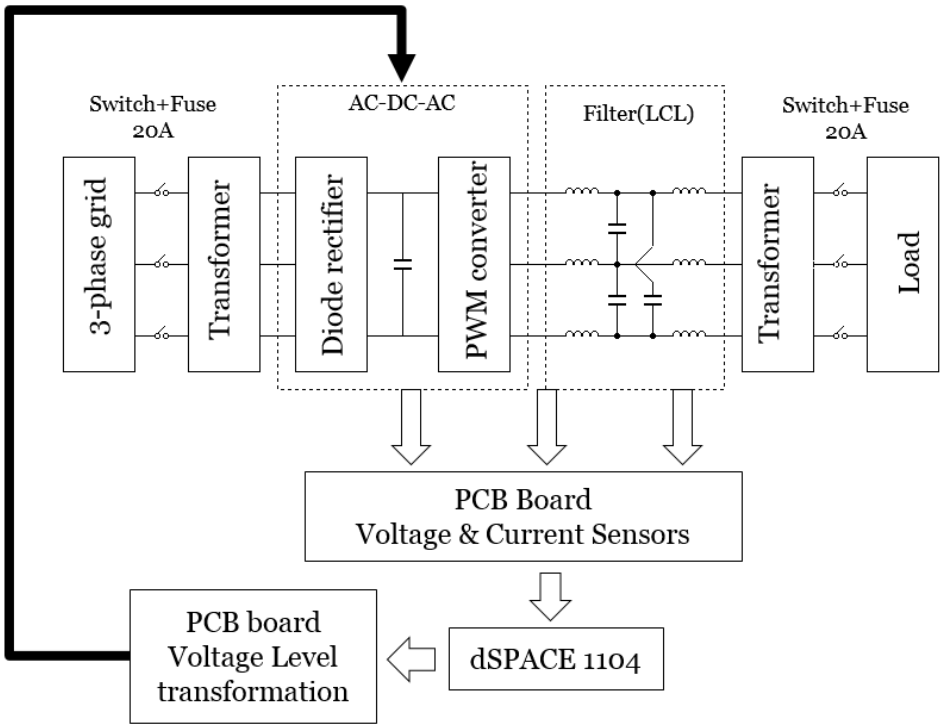


Figure 4.5: Grid simulator system

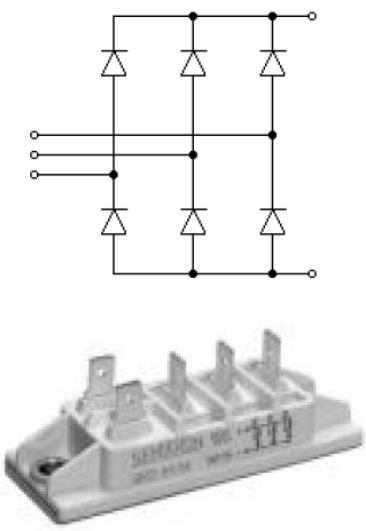


Figure 4.6: Rectifier module

an anti-parallel diode, connected in series, of which the middle point is at terminal 1. The top IGBT collector is at terminal 3 and the bottom IGBT emitter is at terminal 2. The IGBTs are forced closed by a 15V signal between the gate and the

associated emitter and forced open by a −15 V order on the same terminals.

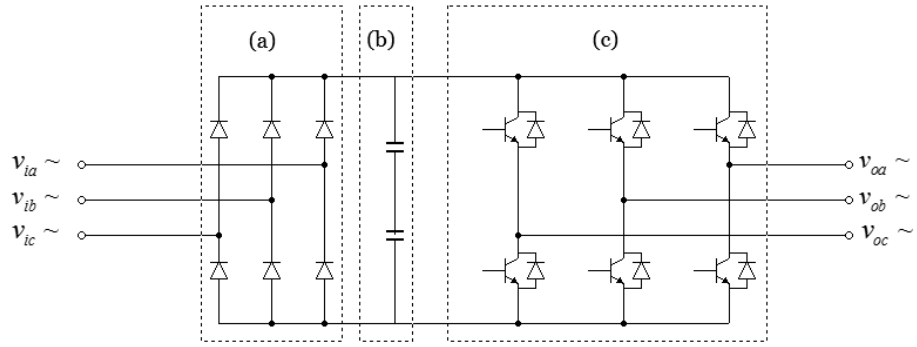


Figure 4.7: AC-DC-AC converter: (a) Diode rectifier; (b) Capacitor bank; (c) PWM inverter

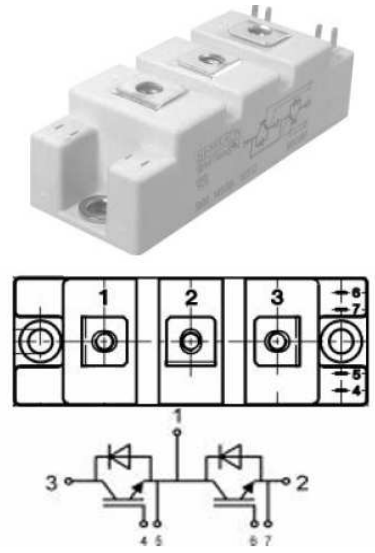


Figure 4.8: IGBT module

The IGBT drivers are based on three SKHI 22 A-R cores as in Fig. 4.9, each one controlling an inverter leg (both Top and Bottom IGBTs of one single module). The SKHI 22 A-R must be powered with 0/15 V, and the power supply must be

able to deliver a current above the drivers' consumption (160 mA / driver max). It is equipped with an internal 10 $k\Omega$ pull-up resistor versus Vs. The error signal is negative logic, i.e. it gives a +15 V signal when no error has occurred.

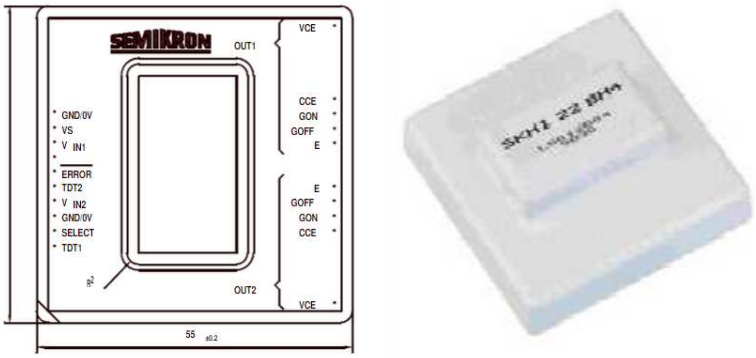


Figure 4.9: IGBT driver

The controller is programmed in Matlab/Simulink and realized in real time by a Dspase1104 at $6kHz$. The slave DSP in the DS1104 are used to generate PWM signals for IGBT drivers. It provides 3 output channels (phases) for 3-phase PWM signal generation (PWM3) in the frequency range 1.25 Hz to 5 MHz. For PWM3, the DS1104 (and the optional connector panels CP1104/ CLP1104) provides the signals for both the non-inverted and the inverted PWM3 phases.

The output filters are LC type with parameters listed in Table 4.1. Voltage sensors are LEM 25-P and LEM current transducer HX-50P is used for current measurements. The DC voltage at the capacitor bank of the rectifier is always set to be $500V$ by adjusting the variable transformer.

4.2.4 Reference generator

Grid simulator generates output voltages with various perturbations by tracking different references from the reference generator. A fast and high-organized reference

generator improves the performance of grid simulators. In this section, a general reference generator is designed.

Voltage dips

Voltage dips in three-phase systems can be classified into three typical categories: [235]

- type A: an equal drop in all the three phases;
- type B: a drop in two phases only;
- type C: a large drop in one phase with small drops in the other two phases.

and all these three types of voltage dips can be depicted in Fig. 4.10.

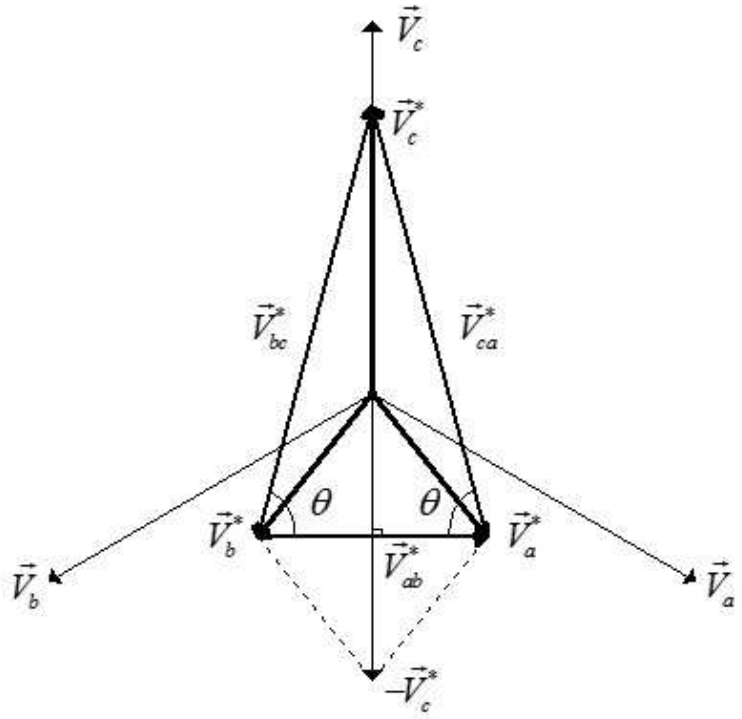


Figure 4.10: Voltage phaser representation of phase-to-phase voltage dips

Suppose the amplitude of balanced three-phase voltages is V_{ref}, i.e. $\left|\vec{V}_a\right| = \left|\vec{V}_b\right| = \left|\vec{V}_c\right| = V_{ref}$, then drops of AC voltage's amplitude in each phase in Fig. 4.10 can be represented by

$$\left|\vec{V}_c^*\right| = \xi V_{ref} \tag{4.10}$$

$$\left|\vec{V}_a^*\right| = \left|\vec{V}_b^*\right| = \Phi V_{ref} \tag{4.11}$$

where Φ, ξ are real constant and $0 < \Phi \leq 1$, $0 < \xi \leq 1$.

It can be derived from Fig. 4.10 that

$$\tan\theta = \frac{2\xi}{\sqrt{4\Phi^2 - \xi^2}}, \tag{4.12}$$

$$\left|\vec{V}_{ab}^*\right| = V_{ref}\sqrt{4\Phi^2 - \xi^2}, \tag{4.13}$$

$$\left|\vec{V}_{bc}^*\right| = \left|\vec{V}_{ca}^*\right| = V_{ref}\sqrt{\Phi^2 + 2\xi^2}, \tag{4.14}$$

then the three types of voltage dips can be obtained by assigning appropriate values to Φ and ξ: type A can be obtained when $0 < \Phi = \xi < 1$; type B can be obtained when $\Phi = 1$ and $0 < \xi < 1$; type C can be obtained when $0 < \Phi \neq \xi < 1$. θ can be calculated according to (4.12).

Given Φ and ξ, reference for three-phase voltages with voltage dip can be generated by the following equations:

$$V_{ab}^* = \left|\vec{V}_{ab}^*\right| \sin(2\pi f_0 t), \tag{4.15}$$

$$V_{bc}^* = \left|\vec{V}_{bc}^*\right| \sin(2\pi f_0 t - \pi + \theta), \tag{4.16}$$

$$V_{ca}^* = \left|\vec{V}_{ca}^*\right| \sin(2\pi f_0 t + \pi - \theta). \tag{4.17}$$

Harmonic distortion and voltage flicker

Harmonic distortion are generated from the following functions and added to the main reference voltages,

$$\begin{aligned} v_{ab}^{h} &= V_h sin(2h\pi f_0 t), \\ v_{bc}^{h} &= V_h sin(2h\pi f_0 t + \frac{2\pi}{3}), \\ v_{ca}^{h} &= V_h sin(2h\pi f_0 t + \frac{4\pi}{3}), \end{aligned} \tag{4.18}$$

where V_h is the amplitude of the harmonic component and h is the harmonic order. Different order harmonics are generated independently from (4.18), thus they can be added simultaneously.

Voltage flicker is realized by modulating the amplitude of main reference voltage as

$$V_{ref}^{f} = \sigma V_{ref} \tag{4.19}$$

where σ is a time-varying modulation factor that can be either square wave or sinusoidal wave. The amplitude variation is limited to 10%.

4.3 Experimental results

4.3.1 High-quality three-phase AC voltages with various frequencies

The PSFRC-controlled grid simulator proposed in this chapter is capable to generate three-phase AC voltages with various frequencies. In this section, 50Hz and 60Hz AC voltages are simulated to verify the effectiveness of the grid simulator.

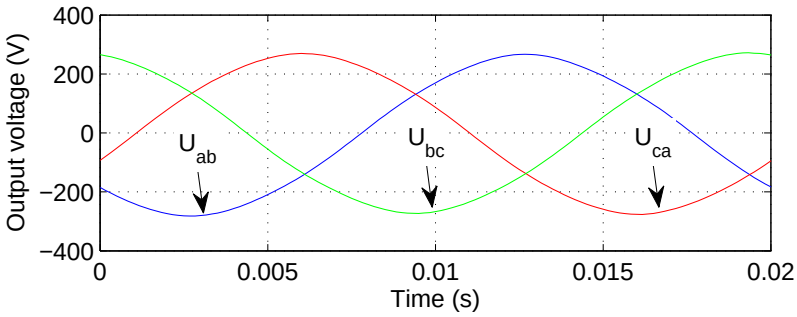
Fig. 4.11 shows the experimental results of generating 200V, 50Hz three-phase voltages feeding a balanced Δ-connected resistor load. Fig. 4.11(a) shows the output

line-to-line voltages, Fig. 4.11(b) shows the output currents and Fig. 4.11(c) are the spectrums of U_{ab}. The experimental results show that the grid simulator can generate high-quality three-phase voltages with very low THD= 0.61%.

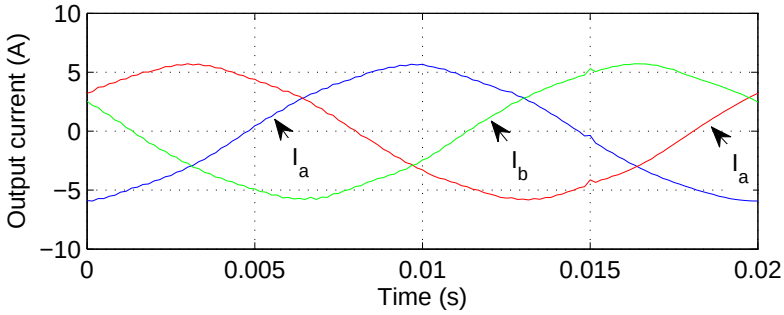
Besides, the PSFRC-controlled grid simulator is also testified to generate 200V, 60Hz output voltages and is connected to a rectifier load as shown in Fig. 4.12. The experimental results are shown in Fig. 4.13. Fig. 4.13 shows that nonlinearities in load side do not affect the output of the proposed grid simulator a lot. It can be seen that a very low THD of 1.53% is achieved.

The capability of the grid simulator to generate high-quality output voltages in fractional cases is also verified by the following experiments. In this case, 200V, 49.5Hz three-phase reference voltages, which correspond to a reasonable frequency disturbance range in E.ON grid code, are set in the reference generator. In this case, $N = f_s/f_0 = 121.2$ and it can be calculated that $\delta = 0.99$. The output of the PSFRC-controlled grid simulator feeding nonlinear rectifier load is shown in Fig. 4.14. Fig. 4.14(a), Fig. 4.14(b) and Fig. 4.14(c) are the line-to-line output voltages, output currents and spectrums of output U_{ab}, respectively. From the experimental results, a very low THD= 1.54% is achieved in fractional cases.

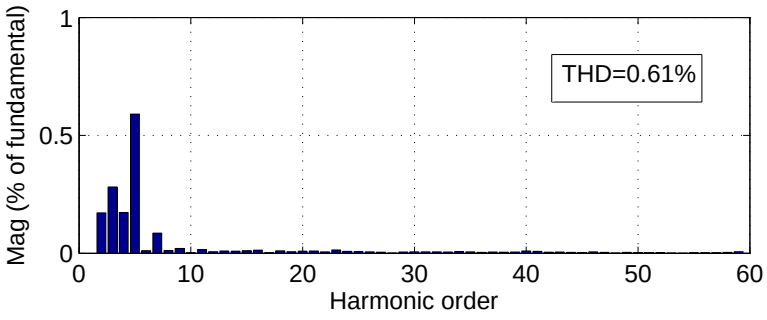
Moreover, a CRC-controlled grid simulator is also implemented, and Fig. 4.15 shows the structure of the CRC used. It should be noted that the learning gain of CRC k_r has an identical stable range with $\sum k_i$ in PSFRC [31]. CRC's k_r is set to be $\sum k_i = 0.39$ in this experiment because the choice of k_r affects the tracking performance of RCs. Fig. 4.16 shows the experimental results, and it shows that the proposed PSFRC-controlled grid simulator achieves better tracking performance than CRC-controlled one whose THD is 2.28%.



(a)



(b)



(c)

Figure 4.11: 50Hz three-phase voltages with linear load (proposed grid simulator). (a) Output voltage; (b) output current; (c) spectrum of U_{ab}.

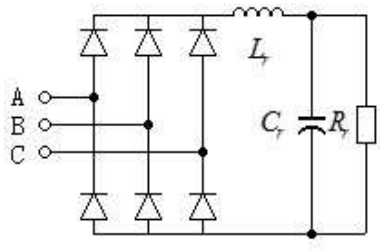


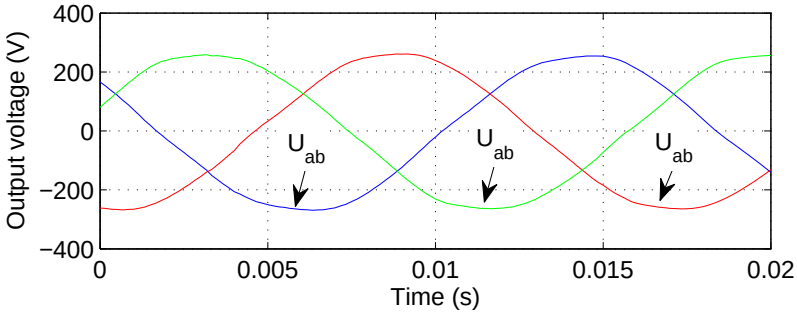
Figure 4.12: Rectifier load

The convergence of PSFRC and second-order odd-harmonic RC (ORC) are shown in Fig. 4.17. Second-order odd-harmonic RC is an improved robust and fast RC controller designed for fractional applications [52]. Both the PSFRC and the ORC are plugged into the state-feedback system at t=1.0s. Fig. 4.17 shows that PSFRC converges within 0.2s and faster than ORC. Besides, the grid simulator's transient response to a frequency change of reference from 50Hz to 60Hz are shown in Fig. 4.18. It shows that PSFRC controlled grid simulator tracks the new reference within $0.12s$.

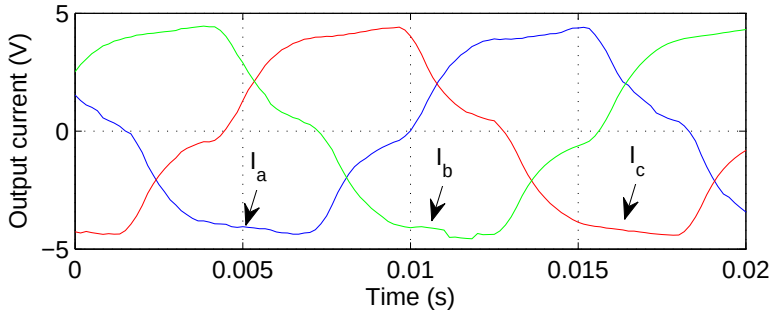
4.3.2 Harmonic distortion

In this section, the proposed grid simulator is testified to generate three-phase voltages with specific harmonic distortions. 200V, 50Hz three-phase voltages are generated with 10% of 5th and 5% of 7th order harmonic components. The proposed grid simulator is connected to a rectifier load as in Fig. 4.12. The line-to-line output voltages, output currents and the spectrum of U_{ab} are shown in Fig. 4.19. Spectrum analysis in Fig. 4.19(c) shows that the 5th order and the 7th order harmonic components takes up 9.91% and 5.21% of fundamental voltage, respectively.

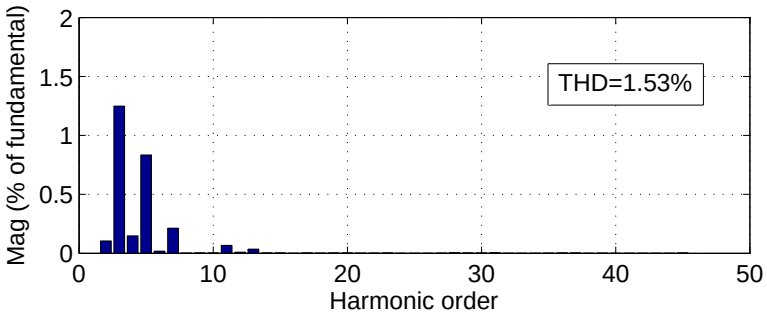
Comparing with the proposed grid simulator, a CRC-controlled grid simulator is also used to generate the same output voltages. k_r is set to be 0.39 which equals to $\sum k_i$ in this case, and the experimental results are shown in Fig. 4.20. The spectrum analysis in Fig. 4.20(c) shows that big error exists in the tracking of



(a)

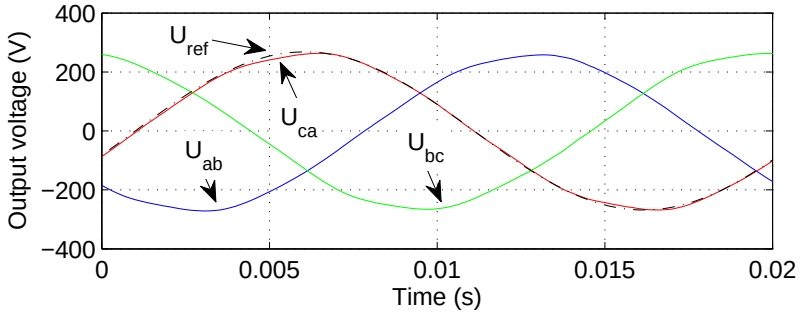


(b)

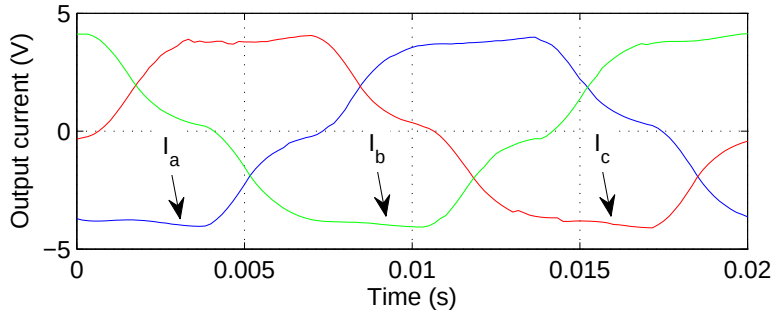


(c)

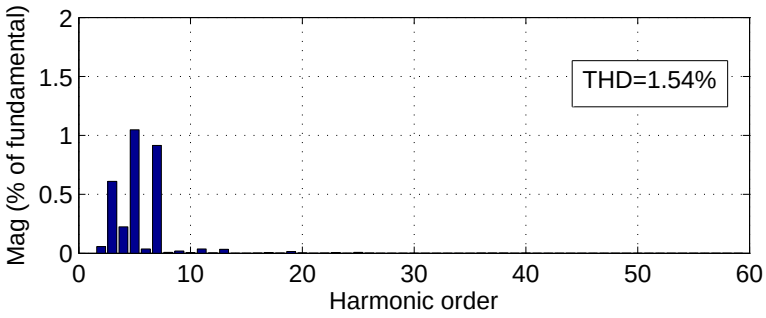
Figure 4.13: 60Hz three-phase voltages with nonlinear load (proposed grid simulator). (a) Output voltage; (b) output current; (c) spectrum of U_{ab}.



(a)



(b)



(c)

Figure 4.14: 49.5Hz three-phase voltages with nonlinear load (proposed grid simulator). (a) Output voltage; (b) output current; (c) spectrum of U_{ab}.

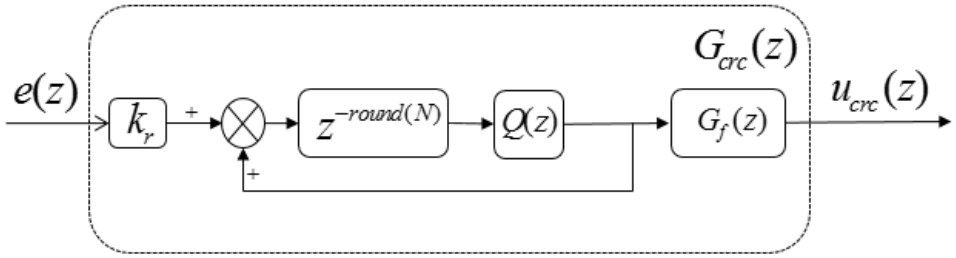


Figure 4.15: Conventional repetitive controller

harmonic components. The 5th and 7th order harmonics takes up only 3.26% and 1.23%, respectively.

4.3.3 Voltage dips

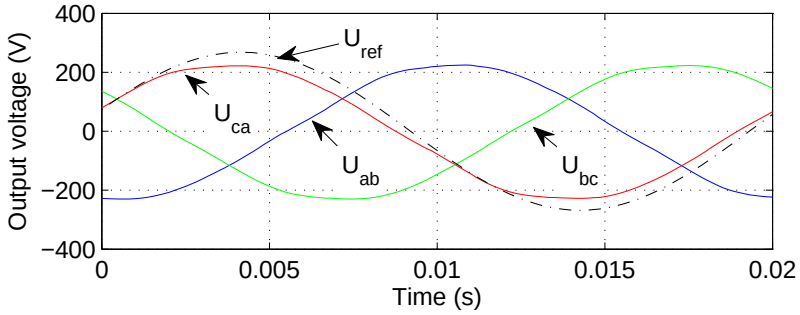
In this section, different types of voltage dips are simulated by the proposed grid simulator.

Firstly, a 10% voltage drop is added to each phase of 200V, 50Hz output voltages simultaneously. Stable voltage drop last for $1s$ and then go back to 200V. This experiment simulates normal grid voltage dips correspond to voltage dip type A in section 4.2. Fig. 4.21(a) shows that the proposed grid simulator can simulate such voltage dips successfully.

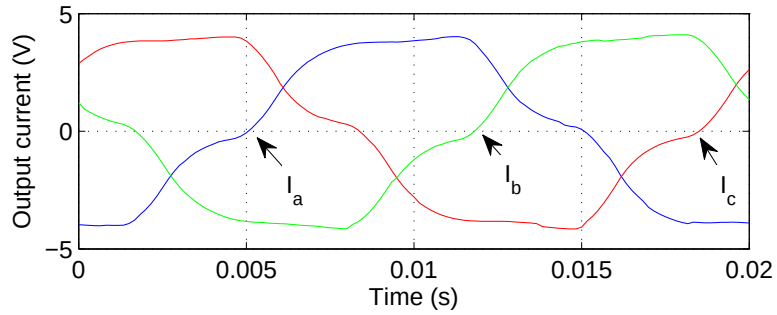
Besides, the tracking error of line-to-line voltage U_{ab} is shown in Fig. 4.21(b). Fig. 4.21(b) shows that only 0.05s (less than 3 cycles) are needed to finish the transient progress.

Secondly, the grid simulator is testified to simulate sudden short interruption of grid with a magnitude drop to its 1%. The experimental results are shown in Fig. 4.22.

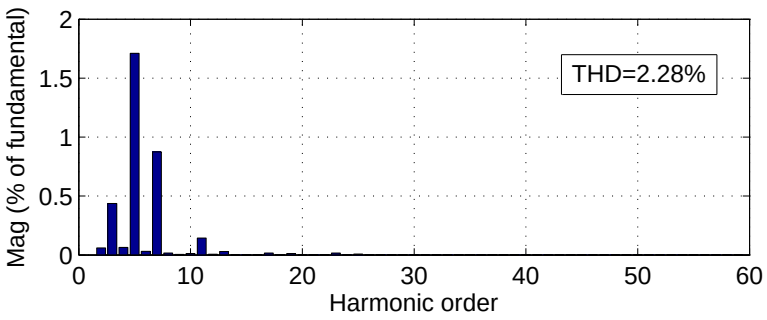
Furthermore, voltage dips of type B and type C are simulated by the proposed grid simulator. Fig. 4.23 shows a voltage dip type B generated by the grid simulator feeding nonlinear rectifier load. In this case, $\Phi = 1$ and $\xi = 0.8$ in the reference generator. Fig. 4.24 shows the experimental results for voltage dip type C with



(a)

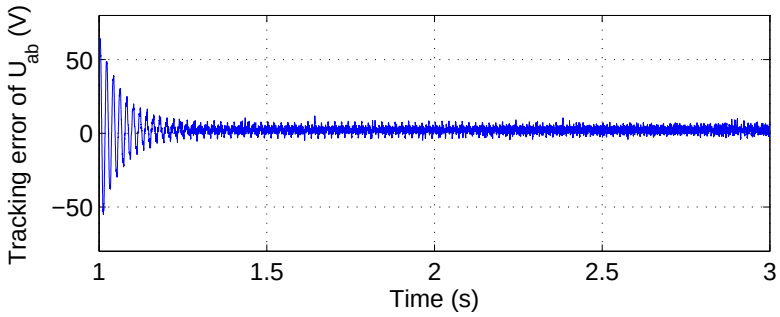


(b)

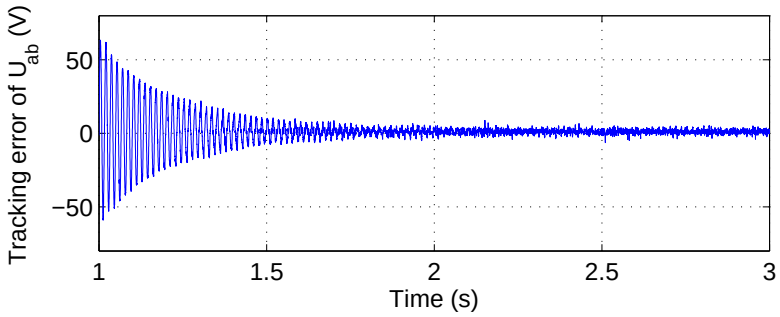


(c)

Figure 4.16: 49.5Hz three-phase voltages with nonlinear load (CRC-controlled grid simulator). (a) Output voltage; (b) output current; (c) spectrum of U_{ab}.



(a)



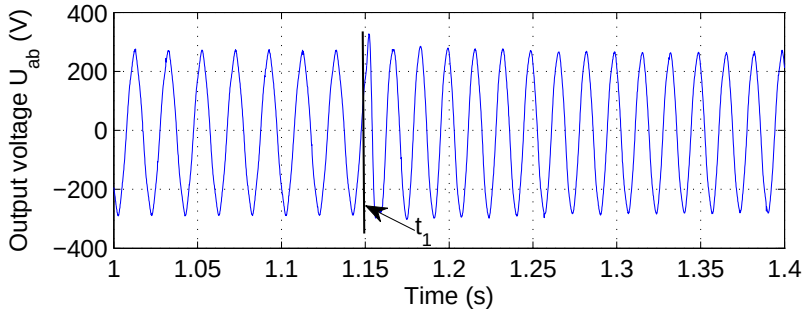
(b)

Figure 4.17: Convergence of RCs. (a) PSFRC; (b) ORC.

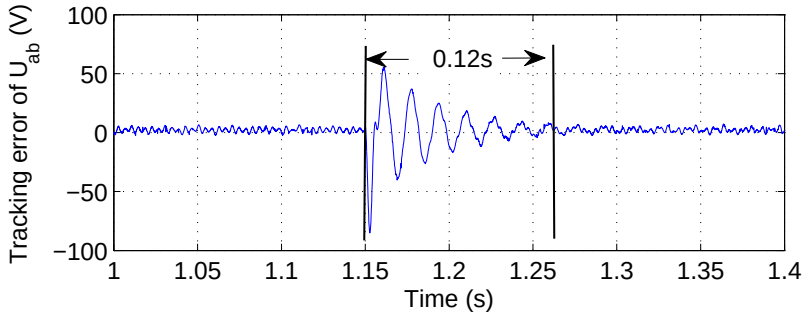
$\Phi = 0.8$ and $\xi = 1$. The grid simulator is connected with a nonlinear load as in Fig. 4.12 too. The experimental results verify the capability of the proposed grid simulator to simulate different types of voltage dips.

4.3.4 Voltage flicker

In this section, the capability of the proposed grid simulator to generate low-frequency oscillations in voltage amplitude is testified. 200V, 50Hz three-phase voltages with sinusoidal flicker is generated. The amplitude variance is 5% of main voltage and the frequency of flicker is 4Hz. Fig. 4.25 shows the generated output voltage and verifies the effectiveness of the proposed grid simulator to generate voltage flickers.

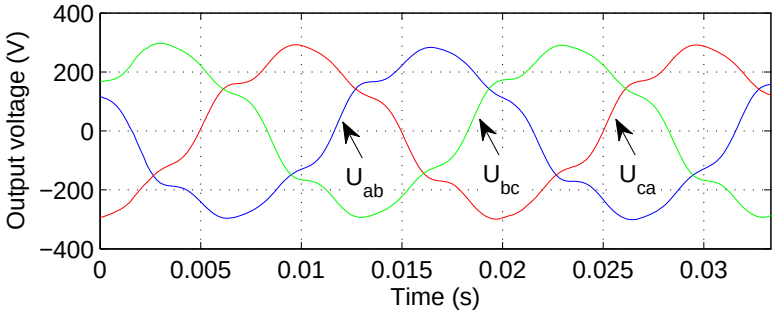


(a)

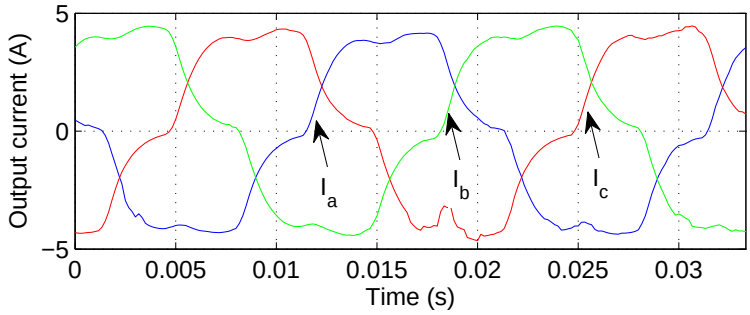


(b)

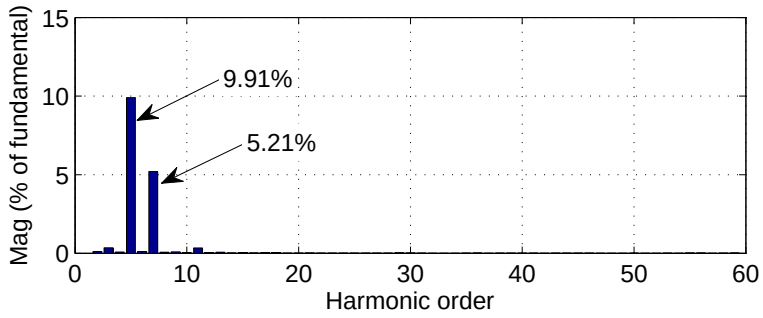
Figure 4.18: Transient performance when frequency changes from 50Hz to 60Hz. (a) Output voltage; (b) Tracking error.



(a)

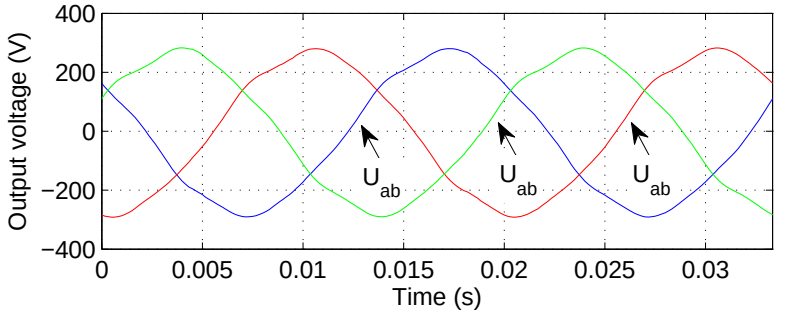


(b)

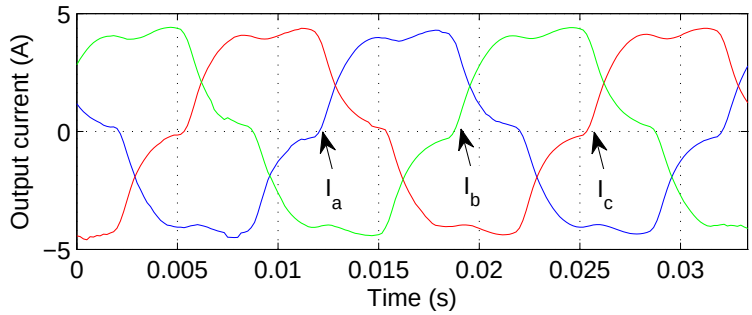


(c)

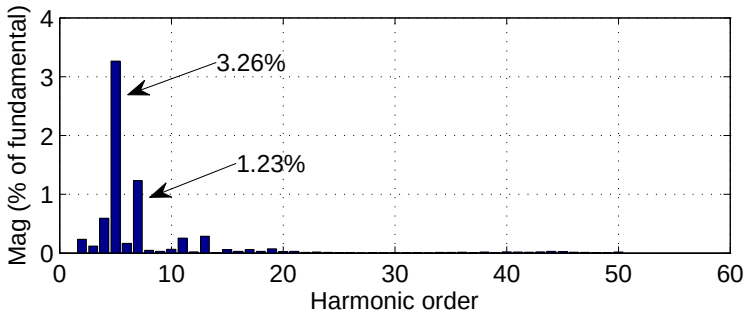
Figure 4.19: Three-phase voltages with harmonic distortion (proposed grid simulator). (a) Output voltage; (b) output currents; (c) spectrum of U_{ab}.



(a)

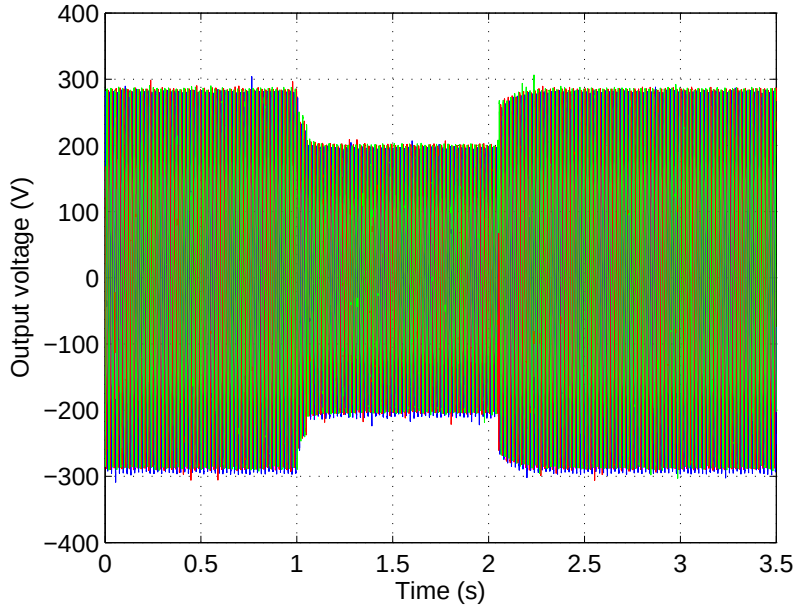


(b)

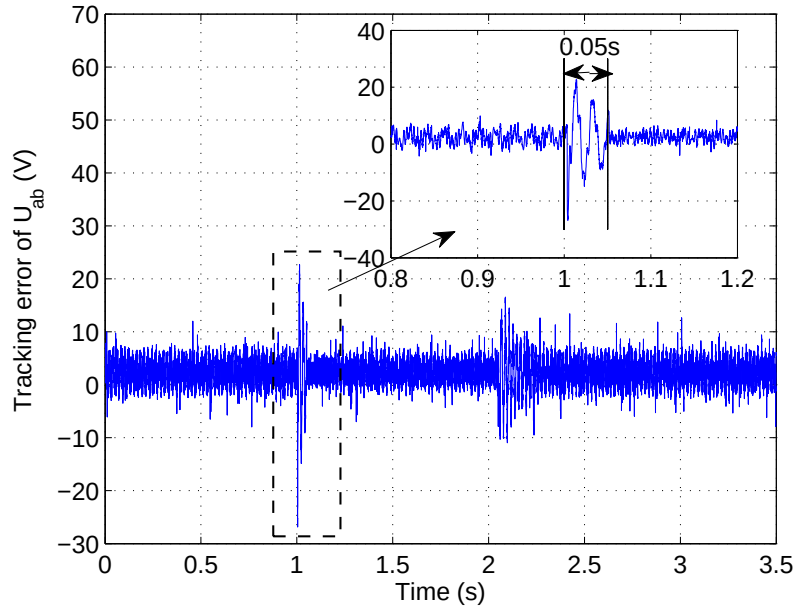


(c)

Figure 4.20: Three-phase voltage with harmonic distortion (CRC-controlled grid simulator). (a) Output voltage; (b) output currents; (c) spectrum of U_{ab}.

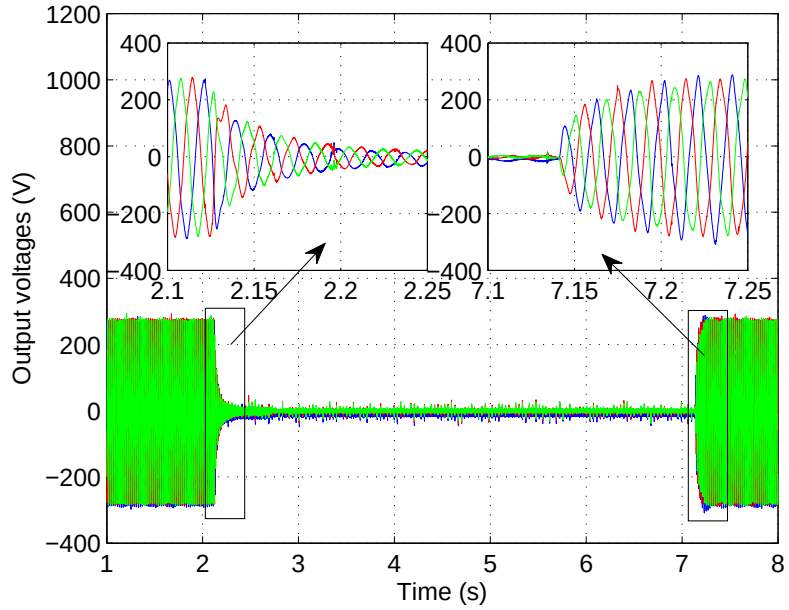


(a)

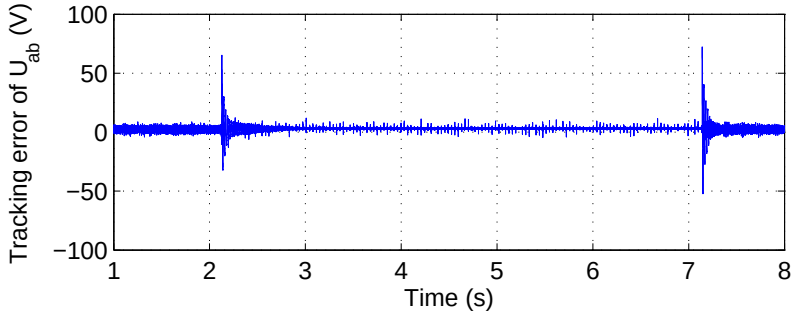


(b)

Figure 4.21: Balanced three-phase voltage dip (Type A) . (a) Output voltage; (b) Tracking error of U_{ab}.

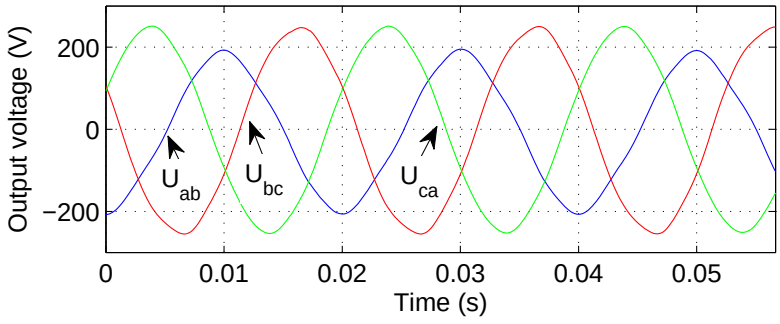


(a)

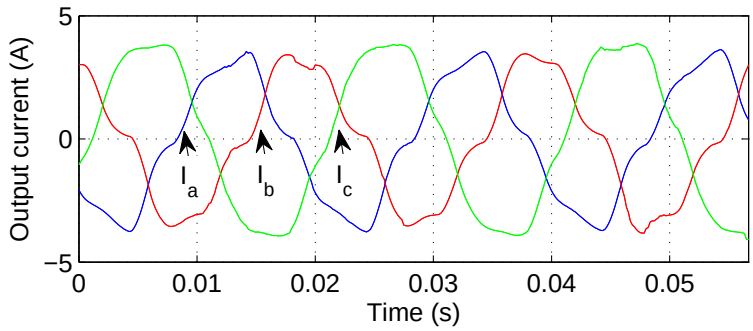


(b)

Figure 4.22: Simulation of short interruption in grid . (a) Output voltage; (b) Tracking error of U_{ab}.

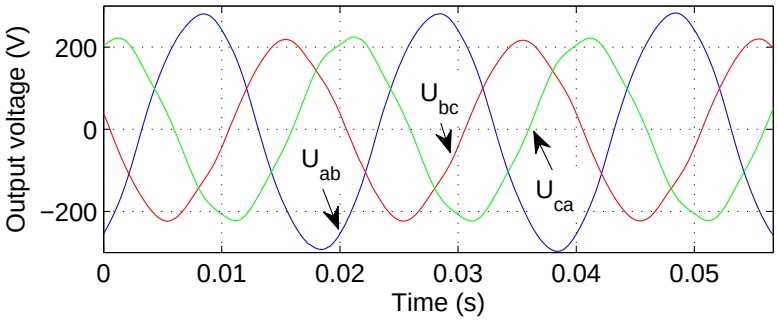


(a)

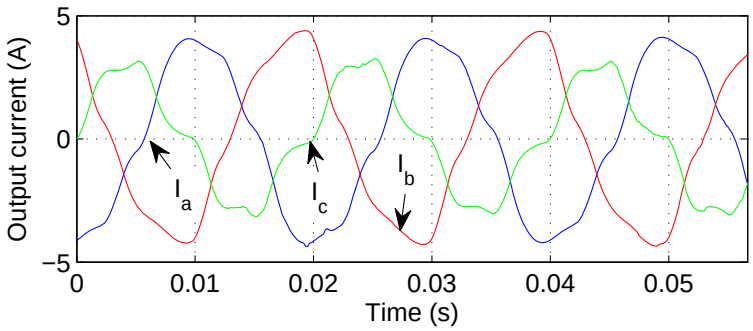


(b)

Figure 4.23: Unbalanced three-phase voltage dip (Type B) . (a) Output voltage; (b) Output current.



(a)



(b)

Figure 4.24: Unbalanced three-phase voltage dip (Type C) . (a) Output voltage; (b) Output current.

4.3.5 Hybrid disturbance

In order to further illustrate the effectiveness of the proposed grid simulator, hybrid disturbance contains multiple types of disturbance is generated in this section. In this case, both voltage dips (type B) and harmonic distortion are added to the main voltage. As shown in Fig. 4.26, the grid simulator tracks a 200V, 50Hz voltage reference suffers from voltage dip type B with $\Phi = 1$, $\xi = 0.8$. Besides, a 10% 5th order and a 5% 7th order harmonic distortion are required in U_{ab}; a 7% of 5th order harmonic component is required in U_{bc}; and U_{ca} is set to contain 5% of 5th order harmonics. The grid simulator is connected to an nonlinear rectifier load. Fig. 4.26 shows the output voltages and currents.

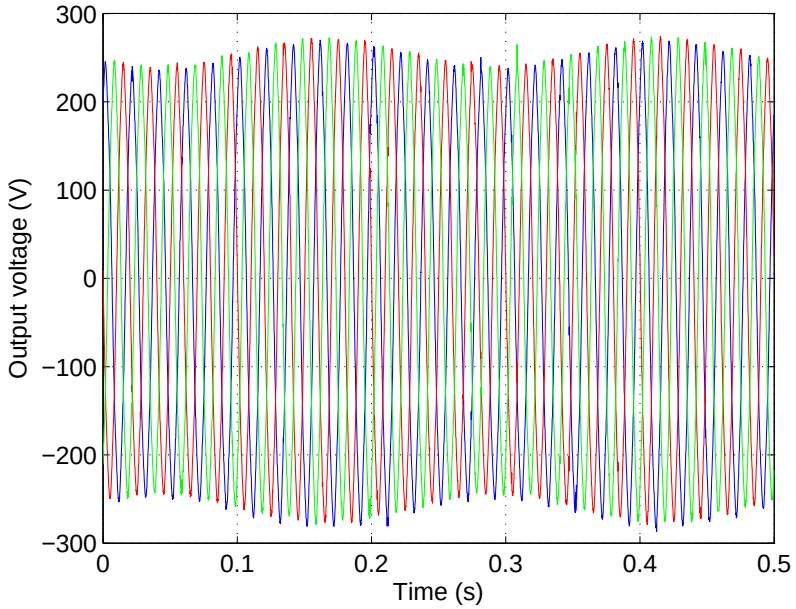
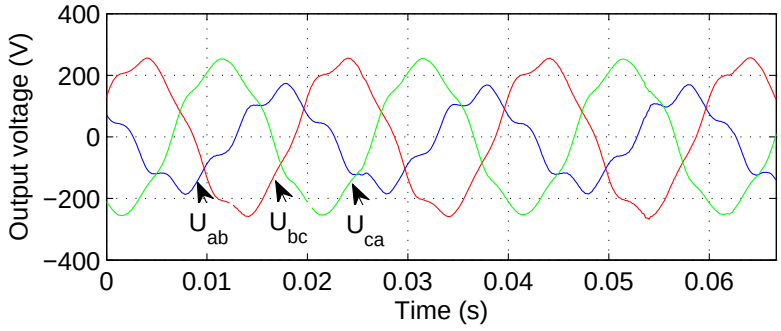
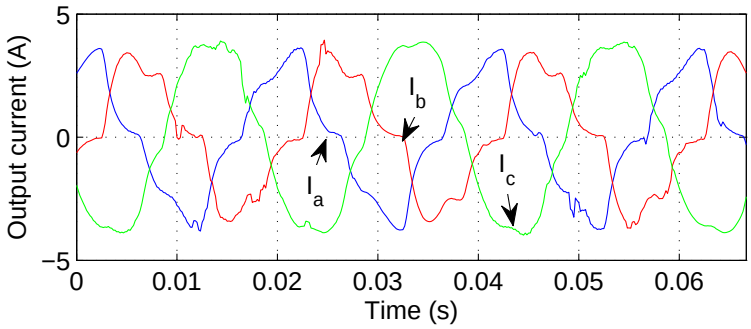


Figure 4.25: Voltage flicker



(a)



(b)

Figure 4.26: Hybrid disturbance. (a) Output voltage; (b) Output current.

4.4 Summary

This chapter applies the parallel structure fractional repetitive control scheme to grid simulators. The new control scheme provides independent tunings for different harmonic groups, thus offering more flexible compensations. Besides, correction factors are introduced in each branch of PSFRC to improve the control gains at dominant harmonics in fractional cases. As a result, PSFRC largely improves the tracking accuracy of the grid simulator in both normal and fractional cases. Transient response is also improved. Therefore, the proposed PSFRC-controlled grid simulator is flexible, fast and accurate. Experimental results are provided to verify the effectiveness of the proposed scheme.

Part II

ITERATIVE LEARNING CONTROL

Chapter 5

Neural Network Based Terminal Iterative Learning Control (Linear Case)

5.1 Introduction

Terminal iterative learning control focuses on tracking the desired terminal reference point. Previous research works mostly consider tracking iteration-invariant reference point with identical initial state [236]. These assumptions simplify the control designs but make the tracking performance depend on the invariance of reference and initial state. In other words, once the reference or initial state changes, a fresh learning process is required. Thus, it is desirable to develop new ILC and TILC controllers to handle iteration-varying reference and initial state.

Several methods to handle non-identical initial state have been proposed in previous works. ILC in [148] solves this problem by learning the initial state before each iteration. Multi-rate ILC scheme in [149] sets the input updating rate different from the sampling rate. Cut-off frequency phase-in profile [150] adopts filter with

time-varying cut-off frequency following a predefined profile for better performance.

As for the difficulty faced by conventional ILC to track iteration-varying reference, [160,161] and [162] propose a class of controllers but are all based on the assumption that zero-error initial condition is satisfied. In [163], adaptive ILC is introduced to track iteration-varying reference beyond initial state variance.

In this chapter, a neural network based terminal iterative learning control (NNTILC) method is proposed to solve the above problems. It is known that neural network is efficient in function approximation and parameter estimation [237]. Using this feature, NNTILC approximates the effect of initial state and reference on the terminal output by a RBF neural network. Then by conducting the control law involving the estimation and the reference signal, it can drive the system to track iteration-varying reference point fast and precisely beyond initial state variance. Besides, since the system's dynamics are included in the control scheme, NNTILC works without any extra learning process even when the reference changes. Convergence analysis of the proposed method is derived mathematically. Simulation results confirm the effectiveness of this method.

The remainder of this chapter is organized as follows. In section 2, the formulation of the problem is introduced. In section 3, a new NNTILC controller is designed for a class of uncertain discrete-time linear systems. Convergence analysis is also provided in section 3. Section 4 provides computer simulation results to illustrate the effectiveness of NNTILC. Section 5 draws some conclusions.

5.2 Problem formulation

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

$$\begin{aligned} x_k(t+1) &= Ax_k(t) + Bu_k(t) \\ y_k(t+1) &= Cx_k(t+1) \end{aligned} \tag{5.1}$$

where $t = 0, 1, 2, ..., T$ is the sampling index and k is the iteration index. Matrices A, B and C are unknown and time invariant with appropriate dimensions; $x_k(t) \in \mathbb{R}^p$ is the state vector, $y_k(t) \in \mathbb{R}^n$ is the output vector, and $u_k(t) \in \mathbb{R}^m$ is the control vector. In terminal iterative learning control, the control input is set as a constant at all sampling times in the same iteration, i.e., $u_k(t) = u_k$ for all $t = 0, 1, 2, ..., T$.

The controller proposed in this chapter aims to track iteration-varying terminal point r_k at time instant T in the kth iteration when initial state variance exists.

From (5.1), the relationship between the system's terminal output $y_k(T)$ and initial state $x_k(0)$ can be developed as

$$y_k(T) = F_k + B^* u_k \tag{5.2}$$

where $F_k = CA^T x_k(0)$ is the effect of initial state on the terminal output and $B^* = \sum_{m=0}^{T-1} CA^{T-1-m}B$.

The tracking error of the system is defined as

$$e_k = r_k - y_k \tag{5.3}$$

where $r_k = f(k)$ is the iteration-varying reference. $f(k)$ can be any value that the system can reach.

Substitute (5.2) into (5.3), the tracking error of the system becomes

$$e_k = r_k - F_k - B^* u_k \tag{5.4}$$

The following assumptions are imposed on the MIMO system (5.1):

Assumption 1: The system is both controllable and observable, $n \leq m$, and $B^* \in \mathbb{R}^{n \times m}$ is of full rank.

Assumption 2: The initial state is accessible in every iteration.

Remark 5.1. *The assumption that B^* is of full rank indicates that the output of the system at terminal time instant T is controllable. It is commonly used in control theory; As for assumption 2, TILC is applied to systems, in which the initial state will be reset before every iteration, so it is reasonable to assume that the initial state is accessible.*

5.3 NNTILC controller design with convergence analysis

In this proposed NNTILC, the controller is designed to adjust the input such that the terminal output in different iterations approach different desired reference values. From (5.4), it can be calculated that the desired input in the kth iteration, which makes the tracking error 0, is

$$u_k^D = B^{*-1}(r_k - F_k) \tag{5.5}$$

However, neither B^* nor F_k is known. Besides, B^* is a system parameter while F_k is a function of initial state. In order to make the estimation simpler, they are

estimated together in a new parameter θ_k, and the controller is constructed as:

$$u_k = \Gamma(r_k - \theta_k) \tag{5.6}$$

where $\Gamma \in \mathbb{R}^{m \times n}$ is a self-define control gain matrix and θ_k is a new parameter which combines both the information of B^* and F_k.

Next, the ideal value of the new parameter θ_k in (5.6), which is denoted as θ_k^D, is identified.

Substitute θ_k^D into (5.6), we have

$$u_k^D = \Gamma(r_k - \theta_k^D) \tag{5.7}$$

Combine (5.5) and (5.7), it can be derived that

$$\theta_k^D = (B^* \cdot \Gamma)^{-1}[F_k - (I - B^* \cdot \Gamma)r_k] \tag{5.8}$$

Remark 5.2. *From assumption* 1*, it can be calculated that* $rank(B^*) = n$*. If* Γ *is of full rank, i.e.* $rank(\Gamma) = n$*, then* $rank(B^* \cdot \Gamma) \leq min[rank(B^*), rank(\Gamma)] = n$*. So there exists such a* Γ *which guarantees that* $B^* \cdot \Gamma$ *is of full rank. Therefore, the ideal* θ_k^D *in equation (5.8) exists.*

To avoid estimating every parameter, a RBF neural network is introduced to model the relationship in (5.8). Then all the estimations are converted to estimating the weight matrix of the neural network.

It is assumed that there exists an ideal function of $x_k(0)$ and r_k and an ideal RBF

neural network which make

$$\begin{aligned}\theta_k^D &= g\left(\begin{bmatrix} x_k(0) \\ r_k \end{bmatrix}\right) \\ &= (B^* \cdot \Gamma)^{-1}[F_k - (I - B^* \cdot \Gamma)r_k] \\ &= (B^* \cdot \Gamma)^{-1}[CA^T x_k(0) - (I - B^* \cdot \Gamma)r_k] \\ &= W_D^T \phi\left(\begin{bmatrix} x_k(0) \\ r_k \end{bmatrix}\right) + \nu(k)\end{aligned} \tag{5.9}$$

where $g(\cdot)$ denotes the ideal function, $W_D^T \in \mathbb{R}^{n \times L}$ is an unknown ideal weight matrix, L denotes the number of neurons in the hidden layer, $\phi(\cdot)$ denotes a known vector of basis activation function, $\begin{bmatrix} x_k(0) \\ r_k \end{bmatrix} \in \mathbb{R}^{p+n}$ is the input vector of the neural network and $\phi\left(\begin{bmatrix} x_k(0) \\ r_k \end{bmatrix}\right) \in \mathbb{R}^L$ is the output of the neural network's hidden layer, $\nu(k)$ is the modelling noise.

The basis activation function is chosen as

$$\phi_i(x) = \exp\left(-\frac{\|x - \mu_i\|^2}{2\sigma_i^2}\right) \tag{5.10}$$

where $\mu_i \in \mathbb{R}^{p+n}$ and σ_i are the centre and width of the ith hidden neuron, $i = 1, 2, ..., L$. In this chapter, only the weight matrix is updated iteratively.

The approximation of θ_k^D in the kth iteration can be represented as

$$\theta_k = \hat{g}_k\left(\begin{bmatrix} x_k(0) \\ r_k \end{bmatrix}\right) = \hat{W}_k^T \phi\left(\begin{bmatrix} x_k(0) \\ r_k \end{bmatrix}\right) \tag{5.11}$$

where $\hat{g}_k(\cdot)$ denotes the approximation of the ideal function $g(\cdot)$ and $\hat{W}_k^T$ is the estimation of the ideal weight matrix in the kth iteration.

From equation (5.10), it can be obtained that

$$F_k = B^* \cdot \Gamma\theta_k^D + (I - B^* \cdot \Gamma)r_k \tag{5.12}$$

Combining (5.2), (5.4), (5.6), (5.11) and (5.12), the tracking error becomes

$$\begin{aligned}
e_k &= r_k - [B^* \cdot \Gamma\theta_k^D + (I - B^* \cdot \Gamma)r_k] - B^* \cdot \Gamma(r_k - \theta_k) \\
&= r_k - [B^* \cdot \Gamma W_D^T \phi \left(\begin{bmatrix} x_k(0) \\ r_k \end{bmatrix} \right) + \nu(k) + (I - B^* \cdot \Gamma)r_k] \\
&- B^* \cdot \Gamma[r_k - \hat{W}_k^T \phi \left(\begin{bmatrix} x_k(0) \\ r_k \end{bmatrix} \right)] \\
&= r_k - B^* \cdot \Gamma W_D^T \phi \left(\begin{bmatrix} x_k(0) \\ r_k \end{bmatrix} \right) - \nu(k) - (I - B^* \cdot \Gamma)r_k \\
&- B^* \cdot \Gamma r_k + B^* \cdot \Gamma \hat{W}_k^T \phi \left(\begin{bmatrix} x_k(0) \\ r_k \end{bmatrix} \right) \\
&= B^* \cdot \Gamma \tilde{W}_k^T \phi \left(\begin{bmatrix} x_k(0) \\ r_k \end{bmatrix} \right) - \nu(k)
\end{aligned} \tag{5.13}$$

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

Define the noise-free *a priori* error in the kth iteration as

$$e_a(k) = B^* \cdot \Gamma \tilde{W}_k^T \phi \left(\begin{bmatrix} x_k(0) \\ r_k \end{bmatrix} \right) \tag{5.14}$$

The tracking error becomes

$$e_k = e_a(k) - \nu(k) \tag{5.15}$$

Assumption 3: The noise $\nu(k)$ is zero-mean, independent, identically distributed and independent of the *a priori* error $e_a(k)$.

Remark 5.3. *Assumption 3 is commonly used in convergence analysis of most adaptive or kernel algorithms [238].*

To train the neural network, a neural network updating law is introduced as

$$\hat{W}_{k+1}^T = \hat{W}_k^T - \alpha e_k \phi^T \left(\begin{bmatrix} x_k(0) \\ r_k \end{bmatrix} \right) \tag{5.16}$$

where $\alpha \in \mathbb{R}$ is a learning gain, which affects the convergence of the proposed method.

The NNTILC controller is proposed based on the control law (5.6) and the neural network updating law (5.16). The overall scheme of the proposed NNTILC controller is shown in Fig. 5.1.

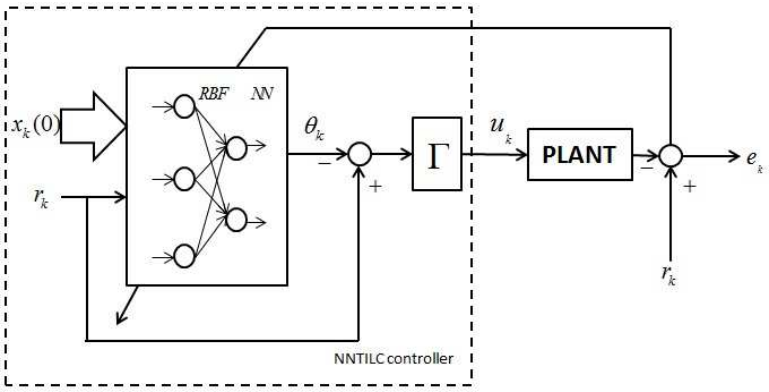


Figure 5.1: Overall scheme of the proposed NNTILC

Theorem 5.1. *For MIMO uncertain discrete-time linear system (5.1), suppose that the control gain* Γ *in (5.6) and neural network updating gain* α *in (5.16) satisfy that*

(1) $\alpha > 0$*;*

(2) $B^* \cdot \Gamma$ *is of full rank;*

(3) $B^* \cdot \Gamma$ *and* $(2I - \alpha LB^* \cdot \Gamma)$ *are positive definite, where* L *is the number of neurons in the hidden layer of the neural network and* I *is an identity matrix.*

Then NNTILC can guarantee that

(1) the weight matrix W_k *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. Subtract W_D^T from both sides of the neural network updating law (5.16), it can be derived that

$$\tilde{W}_{k+1}^T = \tilde{W}_k^T - \alpha e_k \phi^T \left(\begin{bmatrix} x_k(0) \\ r_k \end{bmatrix} \right) \tag{5.17}$$

From (5.10), $\phi(\cdot)$ satisfies

$$0 \leq \phi_i(\cdot) \leq 1, i = 1, 2, ..., L \tag{5.18}$$

where $\phi_i(\cdot)$ is the ith entry of $\phi(\cdot)$.

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

(5.18), it can be derived that

$$0 \leq \phi^T \left(\begin{bmatrix} x_k(0) \\ r_k \end{bmatrix} \right) \phi \left(\begin{bmatrix} x_k(0) \\ r_k \end{bmatrix} \right) \leq L \tag{5.19}$$

.

Choose a candidate Lyapunov function as follows:

$$\begin{aligned} J &= trace(\tilde{W}_{k+1}^T (B^* \cdot \Gamma) \tilde{W}_{k+1}) \\ &= trace(B^* \cdot \Gamma \tilde{W}_{k+1}^T \tilde{W}_{k+1}), \end{aligned} \tag{5.20}$$

then the incremental change of the candidate Lyapunov function is

$$\Delta J = trace[B^* \cdot \Gamma (\tilde{W}_{k+1}^T \tilde{W}_{k+1} - \tilde{W}_k^T \tilde{W}_k)] \tag{5.21}$$

From (5.17), along with the property of trace calculation $trace(X^T Y) = trace(XY^T)$, (5.21) can be rewritten to

$$
\begin{aligned}
&\Delta J = trace[B^* \cdot \Gamma(\tilde{W}_{k+1}^T \tilde{W}_{k+1} - \tilde{W}_k^T \tilde{W}_k)] \\
&= trace\{B^* \cdot \Gamma \left[\tilde{W}_k^T - a e_k \Phi^T \left(\begin{bmatrix} x_k(0) \\ r_k \end{bmatrix}\right)\right]^T \left[\tilde{W}_k^T - a e_k \Phi^T \left(\begin{bmatrix} x_k(0) \\ r_k \end{bmatrix}\right)\right] \\
&- B^* \cdot \Gamma \tilde{W}_k^T \tilde{W}_k\} \\
&= trace\{B^* \cdot \Gamma \left[\tilde{W}_k - a\Phi \left(\begin{bmatrix} x_k(0) \\ r_k \end{bmatrix}\right) e_k^T\right] \left[\tilde{W}_k^T - a e_k \Phi^T \left(\begin{bmatrix} x_k(0) \\ r_k \end{bmatrix}\right)\right] \\
&- B^* \cdot \Gamma \tilde{W}_k^T \tilde{W}_k\} \\
&= trace\{B^* \cdot \Gamma[\tilde{W}_k^T \tilde{W}_k - \alpha \tilde{W}_k^T \Phi \left(\begin{bmatrix} x_k(0) \\ r_k \end{bmatrix}\right) e_k^T - \alpha e_k \Phi^T \left(\begin{bmatrix} x_k(0) \\ r_k \end{bmatrix}\right) \tilde{W}_k \\
&+ \alpha^2 e_k \Phi^T \left(\begin{bmatrix} x_k(0) \\ r_k \end{bmatrix}\right) \Phi \left(\begin{bmatrix} x_k(0) \\ r_k \end{bmatrix}\right) e_k^T - \tilde{W}_k^T \tilde{W}_k]\}
\end{aligned}
\tag{5.22}
$$

Simplify (5.22), we get

$$
\begin{aligned}
&\Delta J = trace[B^* \cdot \Gamma(\tilde{W}_{k+1}^T \tilde{W}_{k+1} - \tilde{W}_k^T \tilde{W}_k)] \\
&= trace\{B^* \cdot \Gamma[-2\alpha \tilde{W}_k^T \phi \left(\begin{bmatrix} x_k(0) \\ r_k \end{bmatrix}\right) e_k^T \\
&+ \alpha^2 e_k \phi^T \left(\begin{bmatrix} x_k(0) \\ r_k \end{bmatrix}\right) \phi \left(\begin{bmatrix} x_k(0) \\ r_k \end{bmatrix}\right) e_k^T]\}
\end{aligned}
\tag{5.23}
$$

From (5.19), it can be obtained that

$$
\Delta J \leq trace[-2\alpha B^* \cdot \Gamma \tilde{W}_k^T \phi \left(\begin{bmatrix} x_k(0) \\ r_k \end{bmatrix}\right) e_k^T + \alpha^2 L B^* \cdot \Gamma e_k e_k^T] \tag{5.24}
$$

Substitute (5.14) and (5.15) into (5.24), it yields

$$\begin{aligned}
&\Delta J \leq trace(-2\alpha e_k e_k^T + \alpha^2 LB^* \Gamma e_k e_k^T - 2\alpha v(k) v(k)^T - 2\alpha v(k) e_a^T(k)) \\
&= trace[\alpha(\alpha LB^* \cdot \Gamma - 2I) e_k e_k^T - 2\alpha v(k) v(k)^T - 2\alpha v(k) e_a^T(k)] \\
&\leq trace[\alpha e_k^T (\alpha LB^* \cdot \Gamma - 2I)^T e_k - 2\alpha v(k) e_a^T(k)] \\
&= -\alpha e_k^T (2I - \alpha LB^* \cdot \Gamma)^T e_k - 2\alpha v^T(k) e_a(k)
\end{aligned} \tag{5.25}$$

We are interested in the mean square behaviour of the NNTILC. Take expectation of both side of (5.25), it yields

$$E(\Delta J) \leq E(\alpha e_k^T (2I - \alpha LB^* \cdot \Gamma)^T e_k - 2\alpha v^T(k) e_a(k)) \tag{5.26}$$

Combine assumption 3 and (5.26), we have

$$E(\Delta J) \leq E(\alpha e_k^T (2I - \alpha LB^* \cdot \Gamma)^T e_k) \tag{5.27}$$

From (5.27), if $\alpha > 0$ and Γ satisfy that $(2I - \alpha L \Gamma B^*)$ is positive definite, then

$$E(\Delta J) < 0 \tag{5.28}$$

Using (5.27) repetitively, it can be derived that

$$\begin{aligned}
&E\{trace[\tilde{W}_{k+1}^T (B^* \cdot \Gamma) \tilde{W}_{k+1}]\} \\
&\leq tarce[\tilde{W}_0^T (B^* \cdot \Gamma) \tilde{W}_0] - \sum_{i=0}^{k} E[e_i^T (2I - \alpha LB^* \cdot \Gamma)^T e_i]
\end{aligned} \tag{5.29}$$

Because the factor $B^* \cdot \Gamma$ is positive definite, along with equation (5.21), J is non-negative and bounded. Together with (5.28), it leads that

$$\lim_{k \to \infty} e_k^T (2I - \alpha LB^* \cdot \Gamma)^T e_k = 0 \tag{5.30}$$

Since the factor $(2I - \alpha LB^* \cdot \Gamma)^T$ is positive definite, we have

$$\lim_{k\to\infty} e_k^T e_k = 0 \tag{5.31}$$

□

Remark 5.4. *In this NNTILC scheme, two control gains* Γ *and* α *are introduced.* Γ *affects the control effort of the controller, while the neural network updating gain* α *decides the convergence rate of the system, i.e., larger* α *leads to faster convergence rate, but overshoot may be introduced. From Theorem 5.1,* Γ *can help to loosen the constraint on* α*.*

Remark 5.5. *In NNTILC, zero error initial condition is not required. Besides, it can be derived from the proof that once the neural network converges, no extra learning is required when the initial state or the reference changes.*

5.4 Simulation studies

To illustrate the effectiveness of the proposed NNTILC scheme, simulations on SISO system and MIMO system are done in this section. The performance of the proposed NNTILC method is compared with conventional TILC method. Conventional TILC is constructed as,

$$u_{k+1} = u_k + l \cdot e_k \tag{5.32}$$

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

5.4.1 Case study on SISO system

Consider the following discrete time SISO system

$$\begin{aligned} x_k(t+1) &= \begin{pmatrix} 0.5 & 0.035 & 0.025 \\ 0.255 & 0.6 & -0.99 \\ 0.75 & 0.03 & 0.025 \end{pmatrix} x_k(t) \\ &+ \begin{pmatrix} 0.2 & 0.2 & 0.0 \end{pmatrix}^T u_k(t) \\ y_k(t) &= \begin{pmatrix} 1.0 & 0.0 & 1.0 \end{pmatrix} x_k(t), \end{aligned} \tag{5.33}$$

and the system operates on time interval $[0, 20]$ in every iteration.

In this simulation, the NNTILC scheme and the conventional TILC method are used to track iteration-varying reference points as shown in Fig. 5.3. The reference points in different iterations are given as

$$r_k = f(k) = 2 + sin(\frac{4\pi}{k}) \tag{5.34}$$

In SISO systems, α and Γ are both real numbers. α affects the convergence rate of the neural network's weight matrix, while Γ decides the control effort. A larger α leads to faster convergence but may introduce overshoot. Usually α is set in the range of $[0, 1]$. Γ should be adjusted to guarantee the stability of the controller, i.e. to satisfy condition (3) in theorem 5.1. In this simulation, the neural network is initialized randomly with 3 neurons in the hidden layer, the centres are chosen properly around the reference trajectory. The activation function in (5.10) is used. The neural network updating gain α is set as 0.4 and the control gain Γ is chosen as 1.43 by trial and error.

The learning gain of conventional TILC l is 1.2. Initial state in this simulation is shown in Fig. 5.2.

Fig. 5.3 shows the terminal output and the tracking error in iteration domain. It shows that the tracking error of NNTILC converges to zero very fast within five iterations. However, the tracking error of conventional TILC performs periodic property without convergence.

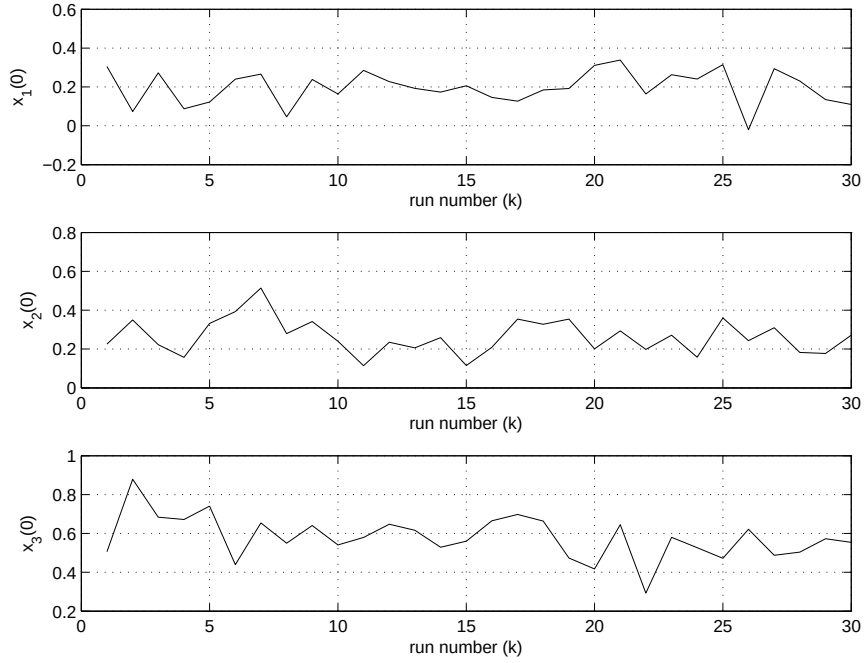


Figure 5.2: Initial state in different runs (SISO case).

5.4.2 Case study on MIMO system

Consider a MIMO system, which can be represented as

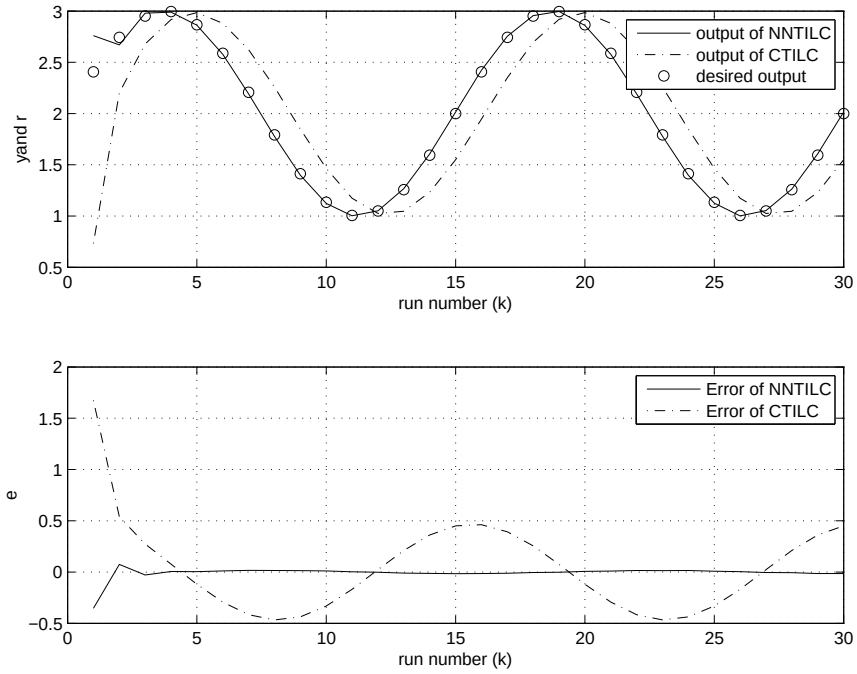


Figure 5.3: Performance for tracking run-varying reference (on SISO system) in iteration domain.

$$
\begin{aligned}
x_k(t+1) &= \begin{pmatrix} 0.5 & 0.035 & 0.025 \\ 0.255 & 0.6 & -0.99 \\ 0.75 & 0.03 & 0.025 \end{pmatrix} x_k(t) \\
&+ \begin{pmatrix} 0.2 & 0.03 & 0.025 \\ 0.2 & 0.2 & 0 \end{pmatrix}^T u_k(t) \\
y_k(t) &= \begin{pmatrix} 1.0 & 0 & 1.0 \\ 0 & 2.0 & 1.0 \end{pmatrix}^T x_k(t),
\end{aligned}
\tag{5.35}
$$

the system operates on time interval $[0, 20]$ in every iteration.

For MIMO systems, the selection of control factors α and Γ is more complicated. In general, the elements on the diagonal of Γ affect the corresponding output. So these elements can be adjusted firstly with all the other elements as 0. In practice,

an upper triangular matrix is enough. In this simulation, the NNTILC controller works with $\alpha = 0.8$ and $\Gamma = \begin{bmatrix} 0.1250 & -0.4167 \\ 0 & 0.8333 \end{bmatrix}$. The RBF neuron network works with 3 hidden neurons.

The reference points in different iterations are shown in Fig. 5.5 and the mathematical representation is as below,

$$\begin{cases} r_{1,k} = f_1(k) = 2 + 3\sin(\frac{4\pi}{k}) \\ r_{2,k} = f_2(k) = 1 + 4\sin(\frac{4\pi}{k}) \end{cases} \tag{5.36}$$

Initial states in every iteration are shown in Fig. 5.4.

The terminal output and the tracking error of NNTILC are shown in Fig. 5.5 and Fig. 5.6, respectively. It shows that the NNTILC converges quite fast for every output.

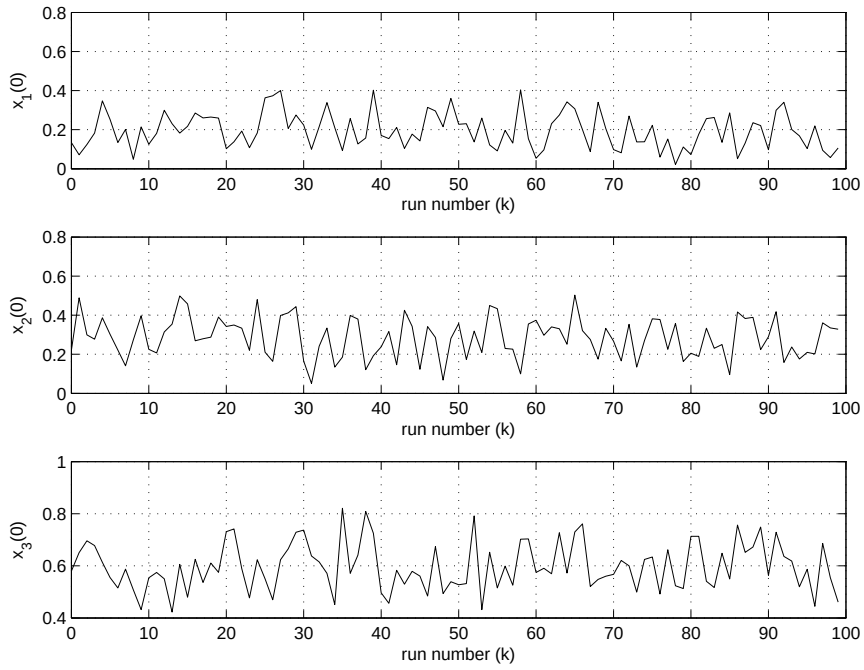


Figure 5.4: Initial state in different runs (MIMO case).

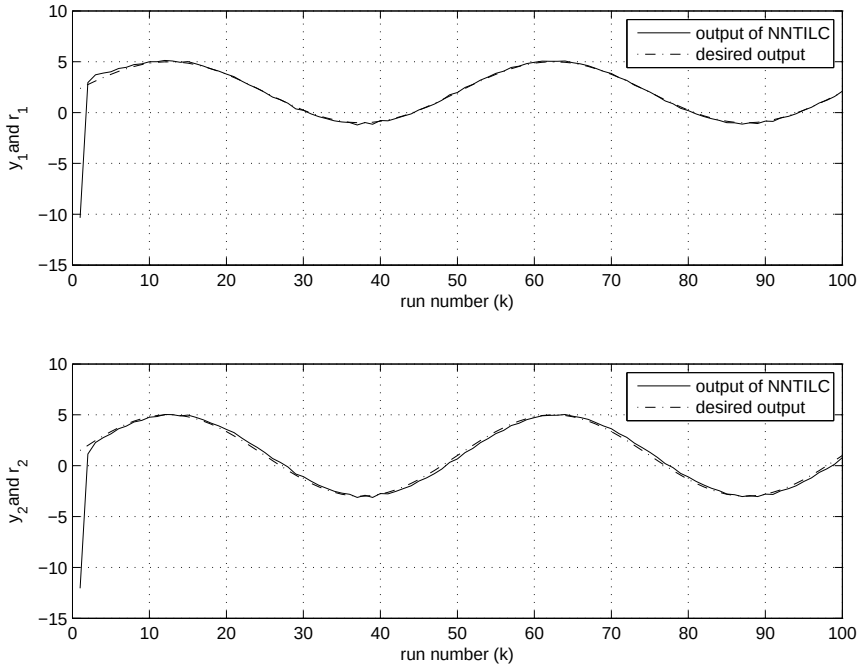


Figure 5.5: Performance for tracking run-varying reference (on MIMO system) in iteration domain.

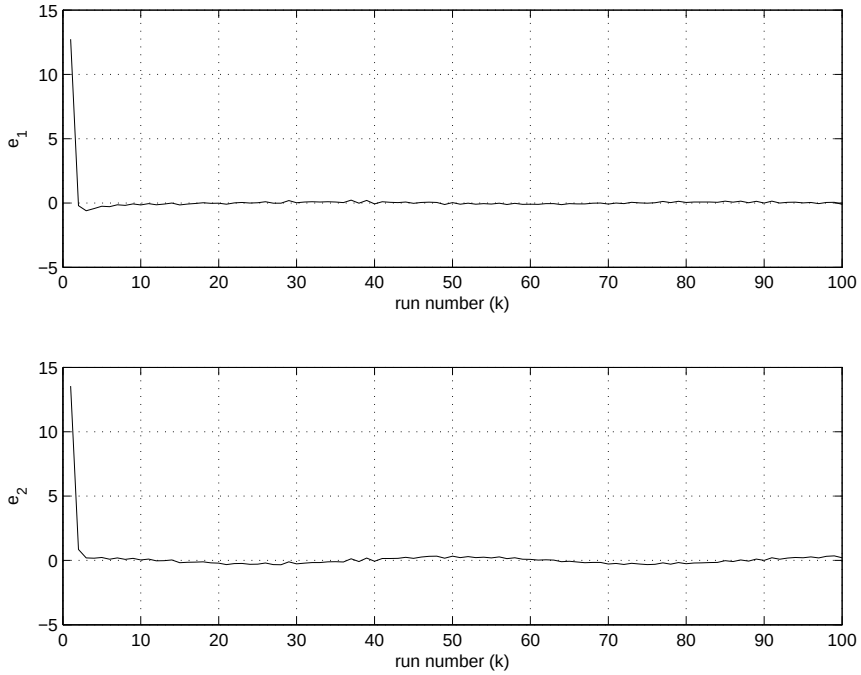


Figure 5.6: Tracking error of NNTILC in iteration domain.

5.5 Summary

For uncertain linear systems, a new neural network based terminal iterative learning control (NNTILC) method is proposed to track iteration-varying reference point with initial state variance. This method uses a RBF neural network to approximate the effect of initial state and reference on the terminal output. It involves both this estimation and the reference signal in the control law. As a result, NNTILC can converge very fast, and no extra learning is required when the initial state and reference change. Convergence of the method is proved, and simulation results confirm the effectiveness of the proposed NNTILC method.

Chapter 6

Neural Network Based Terminal Iterative Learning Control (Nonlinear Case)

6.1 Introduction

In this chapter, the novel neural network based terminal iterative learning control (NNTILC) method is extended to nonlinear non-affine systems. In order to overcome the difficulties faced in the control of non-affine systems, the non-affine terminal output dynamics are converted affine. Besides, an unrealisable recurrent network is simplified to a static realizable one. Because neural network has been proved to be efficient in function approximation and parameter estimation [237], the simplified network is approximated by a RBF neural network iteratively. The proposed NNTILC can track the reference point fast and precisely in the presence of initial state variance. Besides, no fresh learning process is required when the reference changes after convergence.

Convergence of the proposed method is derived mathematically. Simulations on a

thickness control system for wafer fabrication with quartz window effect verify the effectiveness of this method.

The remainder of this chapter is organized as follows. In section 6.2, the formulation of the problem is introduced. In section 6.3, a new NNTILC controller is designed for a class of uncertain nonlinear non-affine systems. Section 6.4 proves the convergence and stability of NNTILC in nonlinear cases. Numerical simulation results compared with conventional TILC are presented in section 6.5. Section 6.6 summarizes this chapter.

6.2 Problem formulation

For a bunch of single input single output (SISO) nonlinear non-affine systems as follows:

$$\begin{cases} \dot{\boldsymbol{x}}(t) = \boldsymbol{f}\{\boldsymbol{x}(t), u(t)\} \\ y(t) = h\{\boldsymbol{x}(t)\} \end{cases} \tag{6.1}$$

where $\boldsymbol{x}(t) \in \mathbb{R}^P$ represents the state vector, $y(t) \in \mathbb{R}$ is the output of the system, $u(t) \in \mathbb{R}$ is the control vector, $\boldsymbol{f}(\cdot)$ and $h(\cdot)$ are both unknown nonlinear functions. The system runs on time interval $[0, T]$ in every iteration. In this chapter, the proposed controller aims to drive the system's terminal output to track a given iteration-varying reference.

In TILC, the control signal is usually set as a constant in each iteration, such that

$$u(t) = u_k \qquad for \quad all \quad t \in [0, T] \tag{6.2}$$

where k is the iteration index. So the system's dynamics can be represented as

$$\begin{cases} \dot{\boldsymbol{x}}_k(t) = \boldsymbol{f}\{\boldsymbol{x}_k(t), u_k\} \\ y_k(t) = h\{\boldsymbol{x}_k(t)\} \end{cases}. \tag{6.3}$$

The relationship of terminal output to initial state and control signal is expressed by

$$y_k(T) = y_k = F\{\boldsymbol{x}_k(0), u_k\} \tag{6.4}$$

where $F(\cdot)$ denotes a function that maps the initial state and control signal to the terminal output of the system. In nonlinear systems, $F(\cdot)$ is complex but the exact format of $F(\cdot)$ is not required. For simplicity, $y_k(T)$ is denoted by y_k in the rest of this chapter.

The tracking error of the system is defined as

$$e_k = r_k - y_k \tag{6.5}$$

where r_k is the iteration-varying reference and it is iteration-dependent and bounded.

The following assumptions are imposed on the SISO non-linear non-affine system (6.1).

Assumption 1: The partial derivative of (6.4) with respect to u_k satisfies

$$\frac{\partial F(\boldsymbol{x}_k(0), u_k)}{\partial u_k} > 0 \tag{6.6}$$

Remark 6.1. *Assumption 1 states that the terminal output of the system is controllable and the control direction is known. Without losing generality, the partial derivative is assumed positive in this chapter.*

Assumption 2: The initial states are measurable and can be used to construct the control signal in every iteration.

Remark 6.2. *Both ILC and TILC are used in repetitive operation systems, in which the initial states will be reset before every iteration. The reset values are usually fixed or follows some fixed pattern in practice. So it is reasonable to assume that the initial states are known in ILC and TILC.*

6.3 NNTILC controller

In this section, a novel NNTILC controller is designed for uncertain nonlinear non-affine systems to track iteration-varying reference with initial state variance.

Equation (6.4) is not affine to u_k, and is rewritten to

$$y_k = F\{\boldsymbol{x}_k(0), u_k\} = F\{\boldsymbol{x}_k(0), u_k\} - \eta u_k + \eta u_k \tag{6.7}$$

where $\eta \in \mathbb{R}$ is a constant to be defined later.

Define

$$\theta_k = \Delta(\boldsymbol{x}_k(0), u_k) = F\{\boldsymbol{x}_k(0), u_k\} - \eta u_k \tag{6.8}$$

where $\Delta(\cdot)$ is a function to represent the relationship between $x_k(0)$, u_k and θ_k, then the system's terminal output becomes

$$y_k = \theta_k + \eta u_k. \tag{6.9}$$

The control law is constructed as

$$u_k = \frac{1}{\eta}(r_k - \hat{\theta}_k^D) \tag{6.10}$$

where $r_k \in \mathbb{R}$ is the iteration-dependent terminal tracking reference, θ_k^D corresponds to the desired input u_k^D which makes the output of the system track the reference, and $\hat{\theta}_k^D$ is the estimation of θ_k^D in the kth iteration.

Submitting (6.10) into (6.9) yields

$$y_k = r_k + \theta_k - \hat{\theta}_k^D, \tag{6.11}$$

and (6.11) leads the terminal tracking error, $e_k \in \mathbb{R}$, to

$$\begin{aligned} e_k &= r_k - y_k = r_k - (r_k + \theta_k - \hat{\theta}_k^D) = \hat{\theta}_k^D - \theta_k \\ &= \hat{\theta}_k^D - \Delta(\mathbf{x}_k(0), u_k) \\ &= \hat{\theta}_k^D - \Delta\left[\mathbf{x}_k(0), \tfrac{1}{\eta}(r_k - \hat{\theta}_k^D)\right] \end{aligned} . \tag{6.12}$$

From (6.10) and (6.12), the desired control signal, which makes the tracking error in (6.12) to be 0, should be

$$u_k^D = \frac{1}{\eta}(r_k - \theta_k^D) \tag{6.13}$$

where θ_k^D is a solution to the following function

$$h(\boldsymbol{x}_k(0), r_k, \hat{\theta}_k^D) \triangleq \Delta\left[\boldsymbol{x}_k(0), \frac{1}{\eta}(r_k - \hat{\theta}_k^D)\right] - \hat{\theta}_k^D = 0 \tag{6.14}$$

or

$$\hat{\theta}_k^D = \Delta\left[\boldsymbol{x}_k(0), \frac{1}{\eta}(r_k - \hat{\theta}_k^D)\right]. \tag{6.15}$$

It should be noted that (6.15) is a recurrent network with a feedback of $\hat{\theta}_k^D$. This makes the approximation of θ_k^D unrealisable in iteration domain because $\hat{\theta}_k^D$ is unavailable in the kth iteration to estimate itself. In order to overcome this problem, Lemma 1 helps to reduce the recurrent network to a realisable static one.

Lemma 1: If *Assumption 1* holds and the constant η in (6.10) satisfies the condition

$$\eta > \frac{\partial F(\boldsymbol{x}_k(0), u_k)}{\partial u_k}, \tag{6.16}$$

then there exists a set $\boldsymbol{\Omega}_{\boldsymbol{x}} \subset \mathbb{R}^n$ and unique θ_k^D, which is a function of r_k and $\boldsymbol{x}_k(0)$

such that $\theta_k^D = g(\boldsymbol{x}_k(0), r_k)$, satisfies (6.14) for all $(\boldsymbol{x}_k(0), r_k) \in \boldsymbol{\Omega}_{\boldsymbol{x}} \times \mathbb{R}$.

Proof. Firstly, it is shown here that the solution θ_k^D to (6.14) exists. The sufficient condition for the existence is that the mapping $\Delta(\cdot)$ is a contraction over the entire input domain, that is, the following inequality must hold [239]

$$\left| \frac{\partial \Delta}{\partial \hat{\theta}_k^D} \right| < 1 \tag{6.17}$$

which can be rewritten to

$$\begin{aligned} \left| \tfrac{\partial \Delta}{\partial \hat{\theta}_k^D} \right| &= \left| \tfrac{\partial [F(\boldsymbol{x}_k(0), u_k) - \eta u_k]}{\partial u_k} \cdot \tfrac{\partial u_k}{\partial \hat{\theta}_k^D} \right| \\ &= \left| \left(\tfrac{\partial F(\boldsymbol{x}_k(0), u_k)}{\partial u_k} - \eta \right) \cdot \left(-\tfrac{1}{\eta} \right) \right| \quad . \\ &= \left| 1 - \tfrac{1}{\eta} \cdot \tfrac{\partial F(\boldsymbol{x}_k(0), u_k)}{\partial u_k} \right| \end{aligned} \tag{6.18}$$

It can be seen from (6.18) that if (6.16) is satisfied, inequality (6.17) holds under assumption 1.

Secondly, the function $h(\cdot)$ is proved nonsingular around θ_k^D. Differentiating the left hand side of (6.14) with respect to $\hat{\theta}_k^D$ yields

$$\begin{aligned} &\tfrac{\partial}{\partial \hat{\theta}_k^D} h(\boldsymbol{x}_k(0), r_k, \hat{\theta}_k^D) \Big|_{\hat{\theta}_k^D = \theta_k^D} \\ &= \tfrac{\partial}{\partial \hat{\theta}_k^D} \left\{ \Delta[\boldsymbol{x}_k(0), u_k] - \hat{\theta}_k^D \right\} \Big|_{\hat{\theta}_k^D = \theta_k^D} = \tfrac{\partial}{\partial \hat{\theta}_k^D} \Delta[\boldsymbol{x}_k(0), u_k] \Big|_{\hat{\theta}_k^D = \theta_k^D} - 1 \\ &= \tfrac{\partial}{\partial \hat{\theta}_k^D} \left\{ F(\boldsymbol{x}_k(0), u_k) - \eta u_k \right\} \Big|_{\hat{\theta}_k^D = \theta_k^D} - 1 = \tfrac{\partial}{\partial u_k} \left\{ F(\boldsymbol{x}_k(0), u_k) - \eta u_k \right\} \tfrac{\partial u_k}{\partial \hat{\theta}_k^D} \Big|_{\hat{\theta}_k^D = \theta_k^D} - 1 \\ &= \left\{ \tfrac{\partial}{\partial u_k} F(\boldsymbol{x}_k(0), u_k) \Big|_{\hat{\theta}_k^D = \theta_k^D} - \eta \right\} \cdot \left(-\tfrac{1}{\eta} \right) - 1 = -\tfrac{1}{\eta} \tfrac{\partial}{\partial u_k} F(\boldsymbol{x}_k(0), u_k) \Big|_{\hat{\theta}_k^D = \theta_k^D} \end{aligned} \tag{6.19}$$

Under assumption 1, equation (6.19) is nonzero. Thus, there exists a unique $g(x_k(0), r_k)$ satisfying (6.14) for all $(x_k(0), r_k)$ according to the implicit function theory. □

Lemma 1 leads to an ideal function $g(\cdot)$ such that

$$\theta_k^D = g(\boldsymbol{x}_k(0), r_k). \tag{6.20}$$

Assume there exists an ideal neural network which makes

$$\theta_k^D = g\left(\begin{bmatrix} \boldsymbol{x}_k(0) \\ r_k \end{bmatrix}\right) = W_D^T \boldsymbol{\phi}\left(\begin{bmatrix} \boldsymbol{x}_k(0) \\ r_k \end{bmatrix}\right) + \nu(k) \tag{6.21}$$

where $W_D^T \in \mathbb{R}^L$ is an ideal weight matrix, L is the number of neurons in the hidden layer of the neural network, $\boldsymbol{\phi}(\cdot)$ is a known basis function, and $\nu(k) \in \mathbb{R}$ stands for the modelling noise.

The basis activation function is chosen as

$$\phi_i(x) = \exp\left(-\frac{\|x - \mu_i\|^2}{2\sigma_i^2}\right) \tag{6.22}$$

where $\mu_i \in \mathbb{R}^{p+1}$ and σ_i are the centre and width of the ith hidden neuron with $i = 1, 2, ..., L$. Only the weight matrix is updated iteratively. The center and width of the hidden neurons are chosen from the input domain properly.

In this chapter, a RBF neural network is introduced to approximate θ_k^D in (6.21) as

$$\hat{\theta}_k^D = \hat{g}_k(\boldsymbol{x}_k(0), r_k) = \hat{W}_k^T \boldsymbol{\phi}\left(\begin{bmatrix} \boldsymbol{x}_k(0) \\ r_k \end{bmatrix}\right) \tag{6.23}$$

where $\hat{\theta}_k^D$, $\hat{g}_k(\cdot)$ and $\hat{W}_k^T$ are the estimations of θ_k^D, $g(\cdot)$ and W_D^T in the kth iteration, respectively.

Subtract (6.21) from (6.23), the estimation error of θ_k^D can be obtained as

$$\begin{aligned}\tilde{\theta}_k^D &= \hat{W}_k^T \boldsymbol{\phi}\left(\begin{bmatrix} \boldsymbol{x}_k(0) \\ r_k \end{bmatrix}\right) - W_D^T \phi\left(\begin{bmatrix} \boldsymbol{x}_k(0) \\ r_k \end{bmatrix}\right) - \nu(k) \\ &= \tilde{W}_D^T \phi\left(\begin{bmatrix} \boldsymbol{x}_k(0) \\ r_k \end{bmatrix}\right) - \nu(k) \\ &= e_a(k) - \nu(k) \end{aligned} \tag{6.24}$$

where $\tilde{W}_D^T = \hat{W}_k^T - W_D^T$ is the estimation error of W_D^T and $e_a(k) = \tilde{W}_D^T \phi\left(\begin{bmatrix} \boldsymbol{x}_k(0) \\ r_k \end{bmatrix}\right)$ is the noise-free *a priori* error.

Assumption 3: The noise $\nu(k)$ is bounded, zero-mean, identically distributed and independent of the *a priori* error $e_a(k)$.

Remark 6.3. *Assumption 3 is commonly used in convergence analysis of most adaptive or kernel algorithms [238]. Equation (6.12) and (6.24) imply that both e_k and $e_a(k)$ are functions of $\hat{\theta}_k^D$, so e_k and $e_a(k)$ are dependent on each other. Under assumption 3, $\nu(k)$ and e_k are independent.*

A neural network updating law is introduced to train the RBF neural network as

$$\hat{W}_{k+1}^T = \hat{W}_k^T - \alpha e_k \boldsymbol{\phi}^T\left(\begin{bmatrix} \boldsymbol{x}_k(0) \\ r_k \end{bmatrix}\right) \tag{6.25}$$

where $\alpha \in \mathbb{R}$ is a learning gain.

The overall scheme of the proposed NNTILC controller is shown in Fig. 6.1.

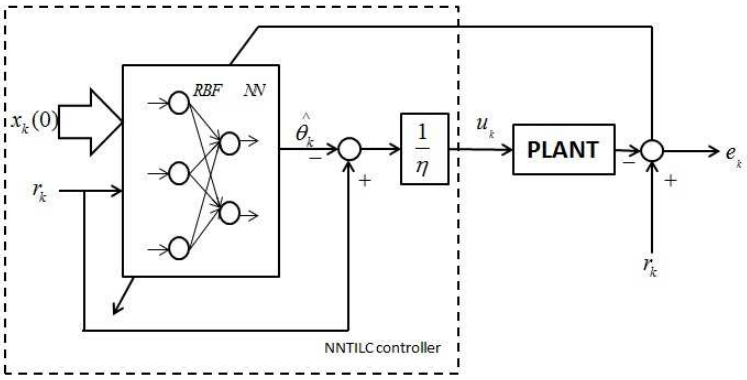


Figure 6.1: Overall scheme of the proposed NNTILC

6.4 Convergence analysis

In this section, the convergence of NNTILC systems is proved.

Theorem 6.1. *For SISO nonlinear non-affine system (6.1), suppose the neural network learning gain α in (6.25) satisfies that*

$$0 \leq \alpha \leq \frac{2}{L} \tag{6.26}$$

where L is the number of neurons in the hidden layer, then the NNTILC controller (6.10) with neural network updating law (6.25) can achieve that

1. the weight matrix W_k 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 increases.

Proof. Subtracting W_D^T from both sides of (6.25) yields

$$\tilde{W}_{k+1}^T = \tilde{W}_k^T - \alpha e_k \boldsymbol{\phi}^T \left(\begin{bmatrix} \boldsymbol{x}_k(0) \\ r_k \end{bmatrix} \right) \tag{6.27}$$

where $\tilde{W}_k^T = \hat{W}_k^T - W_D^T$ is the estimation error of W_D^T in the kth iteration.

Choose a candidate Lyapunov function as follows:

$$J = trace(\tilde{W}_k^T \tilde{W}_k), \tag{6.28}$$

then the incremental change of the candidate lyapunov function is

$$\Delta J = trace(\tilde{W}_{k+1}^T \tilde{W}_{k+1} - \tilde{W}_k^T \tilde{W}_k). \tag{6.29}$$

Substituting (6.27) into (6.29) yields

$$\begin{aligned}
&\Delta J = trace[(\tilde{W}_{k+1}^T \tilde{W}_{k+1} - \tilde{W}_k^T \tilde{W}_k)] \\
&= trace\{ \left[\tilde{W}_k^T - a e_k \Phi^T \left(\begin{bmatrix} x_k(0) \\ r_k \end{bmatrix} \right) \right] \times \left[\tilde{W}_k^T - a e_k \Phi^T \left(\begin{bmatrix} x_k(0) \\ r_k \end{bmatrix} \right) \right]^T \\
&- \tilde{W}_k^T \tilde{W}_k \} \\
&= trace\{ \tilde{W}_k^T \tilde{W}_k - \alpha e_k \tilde{W}_k \Phi^T \left(\begin{bmatrix} x_k(0) \\ r_k \end{bmatrix} \right) \tilde{W}_k^T - \alpha e_k \tilde{W}_k^T \Phi \left(\begin{bmatrix} x_k(0) \\ r_k \end{bmatrix} \right) \\
&+ \alpha^2 e_k^2 \Phi^T \left(\begin{bmatrix} x_k(0) \\ r_k \end{bmatrix} \right) \Phi \left(\begin{bmatrix} x_k(0) \\ r_k \end{bmatrix} \right) - \tilde{W}_k^T \tilde{W}_k] \} \\
&= \alpha^2 e_k^2 \Phi^T \left(\begin{bmatrix} x_k(0) \\ r_k \end{bmatrix} \right) \Phi \left(\begin{bmatrix} x_k(0) \\ r_k \end{bmatrix} \right) - 2\alpha e_k \tilde{W}_k^T \Phi \left(\begin{bmatrix} x_k(0) \\ r_k \end{bmatrix} \right)
\end{aligned} \tag{6.30}$$

It is clear that the basis function $\phi_i(\cdot)$, defined in (6.22), is bounded by $[0, 1]$. Since $\boldsymbol{\phi}(\cdot) \in R^L$, it can be derived that

$$0 \leq \boldsymbol{\phi}^T(\cdot)\boldsymbol{\phi}(\cdot) \leq L \tag{6.31}$$

where L is the number of neurons in the hidden layer.

From (6.24), (6.30) and (6.31), it can be derived that

$$\begin{aligned}\Delta J &\leq \alpha^2 L e_k^2 - 2\alpha e_k \tilde{W}_k^T \phi \left(\begin{bmatrix} \mathbf{x}_k(0) \\ r_k \end{bmatrix} \right) \\ &= \alpha^2 L e_k^2 - 2\alpha e_k [\tilde{\theta}_k^D + \nu(k)]\end{aligned} \tag{6.32}$$

where $\tilde{\theta}_k^D = \hat{\theta}_k^D - \theta_k^D$ is the estimation error of θ_k^D.

Taking partial differentiation of (6.12) with respect to $\hat{\theta}_k^D$ yields

$$\begin{aligned}\frac{\partial e_k}{\partial \hat{\theta}_k^D} &= 1 - \frac{\partial \Delta\left(x_k(0), \frac{1}{\eta}(r_k - \hat{\theta}_k^D)\right)}{\partial \hat{\theta}_k^D} \\ &= 1 - \frac{\partial [F(x_k(0), u_k) - \eta u_k]}{\partial u_k} \cdot \frac{\partial u_k}{\partial \hat{\theta}_k^D} \\ &= 1 - \left(\frac{\partial F(x_k(0), u_k)}{\partial u_k} - \eta\right) \cdot \left(-\frac{1}{\eta}\right) \\ &= \frac{1}{\eta} \cdot \frac{\partial F(x_k(0), u_k)}{\partial u_k}\end{aligned} . \tag{6.33}$$

It can be calculated that

$$\frac{\partial e_k}{\partial \tilde{\theta}_k^D} = \frac{\partial e_k}{\partial (\hat{\theta}_k^D - \theta_k^D)} = \frac{\partial e_k}{\partial \hat{\theta}_k^D} \cdot \frac{\partial \hat{\theta}_k^D}{\partial (\hat{\theta}_k^D - \theta_k^D)} = \frac{\partial e_k}{\partial \hat{\theta}_k^D}, \tag{6.34}$$

so

$$\frac{\partial e_k}{\partial \tilde{\theta}_k^D} = \frac{1}{\eta} \cdot \frac{\partial F\left(x_k(0), u_k\right)}{\partial u_k} \tag{6.35}$$

From (6.16) and (6.35), it can be derived that

$$0 < \frac{\partial e_k}{\partial \tilde{\theta}_k^D} < 1. \tag{6.36}$$

Equation (6.12) and (6.14) show that if $\hat{\theta}_k^D \to \theta_k^D$ or $\tilde{\theta}_k^D \to 0$, then $e_k \to 0$. So the curve e_k versus $\tilde{\theta}_k^D$ passes through the point $(0, 0)$. Therefore, (6.36) leads to

$$|e_k| = \left| \int \frac{\partial e_k}{\tilde{\theta}_k^D} d\tilde{\theta}_k^D \right| < \left| \tilde{\theta}_k^D \right| \tag{6.37}$$

and

$$sgn(e_k) = sgn(\tilde{\theta}_k^D). \tag{6.38}$$

Substituting (6.37) and (6.38) into (6.32) yields

$$\begin{aligned} \Delta J &\leq \alpha^2 L e_k^2 - 2\alpha e_k \tilde{\theta}_k^D - 2\alpha e_k \nu(k) \\ &\leq \alpha^2 L e_k^2 - 2\alpha e_k^2 - 2\alpha e_k \nu(k) \\ &= \alpha(\alpha L - 2) e_k^2 - 2\alpha e_k \nu(k). \end{aligned} \tag{6.39}$$

Taking expectation of (6.39) yields

$$E(\Delta J) \leq E[\alpha(\alpha L - 2) e_k^2 - 2\alpha e_k \nu(k)]. \tag{6.40}$$

Noting **Assumption 3** and **Remark 6.3**, (6.40) can be simplified to

$$E(\Delta J) \leq E[\alpha(\alpha L - 2) e_k^2] \tag{6.41}$$

From (6.41), if α satisfies

$$0 \leq \alpha \leq \frac{2}{L}, \tag{6.42}$$

then

$$E(\Delta J) \leq 0, \tag{6.43}$$

and statement 1 in theorem 6.1 is proved.

By using (6.41) repetitively, it can be obtained that

$$\begin{aligned} &E[trace(\tilde{W}_{k+1}^T \tilde{W}_{k+1})] \\ &\leq E[trace(\tilde{W}_0^T \tilde{W}_0)] - \sum_{i=0}^{k} E(\alpha(\alpha L - 2) e_i^2). \end{aligned} \tag{6.44}$$

Because (6.42) indicates that the factor $\alpha(\alpha L - 2)$ is positive, together with the

fact that $\tilde{W}_{k+1}^T \tilde{W}_{k+1}$ is positive,

$$\lim_{k\to\infty} e_k^T e_k = 0, \tag{6.45}$$

and statement 2 in theorem 6.1 is proved.

□

Remark 6.4. *In Theorem 6.1, L denotes the number of neurons in the hidden layer. Smaller L leads to more flexible choices of learning gain α, so it is better to diminish the number of neurons in the hidden layer to its minimum in real applications.*

6.5 Simulation studies

The effectiveness of the proposed NNTILC approach is illustrated by simulations on a nonlinear system in this section. A thickness control system in the wafer fabrication with quartz window effect as shown in (6.46) is used [240],

$$\begin{cases} \frac{dT_w}{dt} = \frac{\sigma A_w E_w \left(T_q^4 - T_w^4\right) + f E_w Q P}{M_w} \\ \frac{dT_q}{dt} = \frac{QP + hA_q(T_{amb} - T_q)}{M_w} \\ \frac{dS}{dt} = k_0 \exp\left(\frac{\gamma}{RT_w}\right) \end{cases} \tag{6.46}$$

where $hA_q \triangleq 1.84 cal/s/K$ and the other related parameters are listed in Table 1. The system runs on time interval $[0, 200]s$.

The iteration-varying reference is given as

$$r_k = 0.5 + 0.1 \cdot \left(\frac{2k\pi}{50}\right) \tag{6.47}$$

and the initial states are set as (6.48) in different iterations as shown in Fig. 6.2.

$$\begin{cases} x_{1,k}(0) = 300 + 100 \cdot \frac{\sin(20k\pi)}{50} \\ x_{1,k}(0) = 300 + 100 \cdot \frac{\sin(16k\pi)}{50} \\ x_{3,k}(0) = 0 \end{cases} \tag{6.48}$$

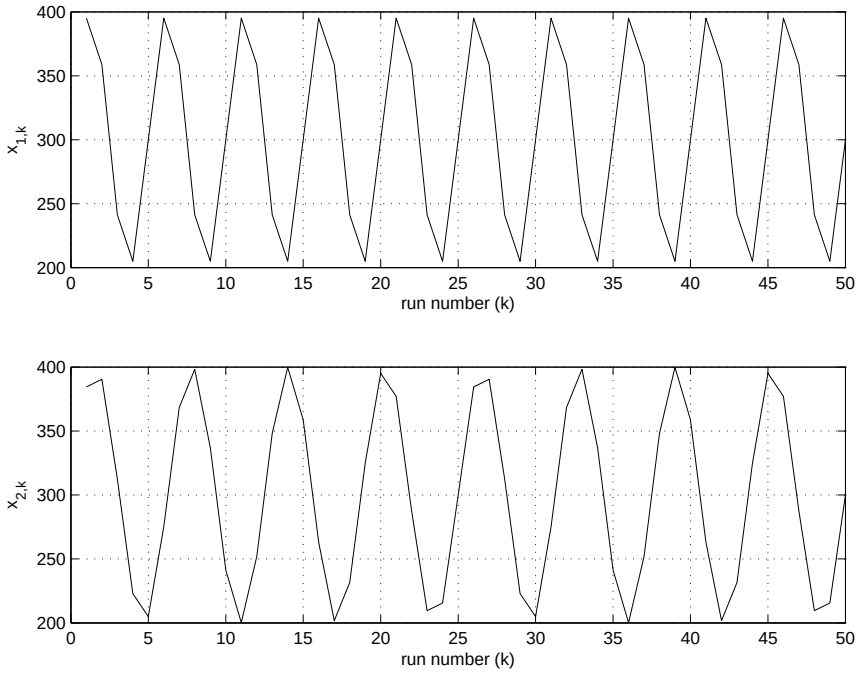


Figure 6.2: Initial state in different iterations

In NNTILC, α and η are two important factors that require careful adjusting. α affects the convergence time of the controller, that is, larger α leads to faster convergence. But too large a α may introduce overshoot to the system. From Theorem 1, α can be selected in $[0, \frac{2}{L}]$. That means the number of neurons in the hidden layer should be set as small as possible to guarantee more flexible choices of α. In this simulation, 6 hidden neurons are used, so α should be chosen from

$[0, 0.33]$. The centre of the neuron network is set randomly around the reference trajectory.

η is very important because it decides the existence of the the function in (6.20). It can be set large firstly to guarantee the condition specified in Lemma 1, and then adjusted accordingly by trial and error. In this simulation, $\eta = 6.5$ and $\alpha = 0.3$.

Besides, the conventional terminal iterative learning control (CTILC) method is simulated on the same system. Its control law is constructed as

$$u_{k+1} = u_k + l \cdot e_k \tag{6.49}$$

where u_{k+1} and u_k represent the control signal in the $(k+1)$th and kth iteration, e_k represents the terminal tracking error and l is a learning gain. In this section, l is set as 0.37.

Fig. 6.3 and Fig. 6.4 show that NNTILC method can be applied to nonlinear non-affine systems successfully. It can be seen from Fig. 6.4 that the tracking error converges within 5 iterations. However, it can be seen from Fig. 6.4 that the tracking error of the conventional TILC exhibits an oscillation and the average error is much larger than that of NNTILC.

Moreover, a trade-off between overshoot oscillation and average error exists in conventional TILC. Fig. 6.5 shows that the overshoot oscillation is evident with $l = 0.5$, although the final average tracking error with $l = 0.5$ is smaller than that when $l = 0.27$. By contrast, NNTILC offers better tracking performance and convergence rate compared with CTILC. It should be noted that initial state variance and reference change have no effect on the tracking performance of NNTILC. The detailed comparison between different controllers' performance can be found in Table 6.1.

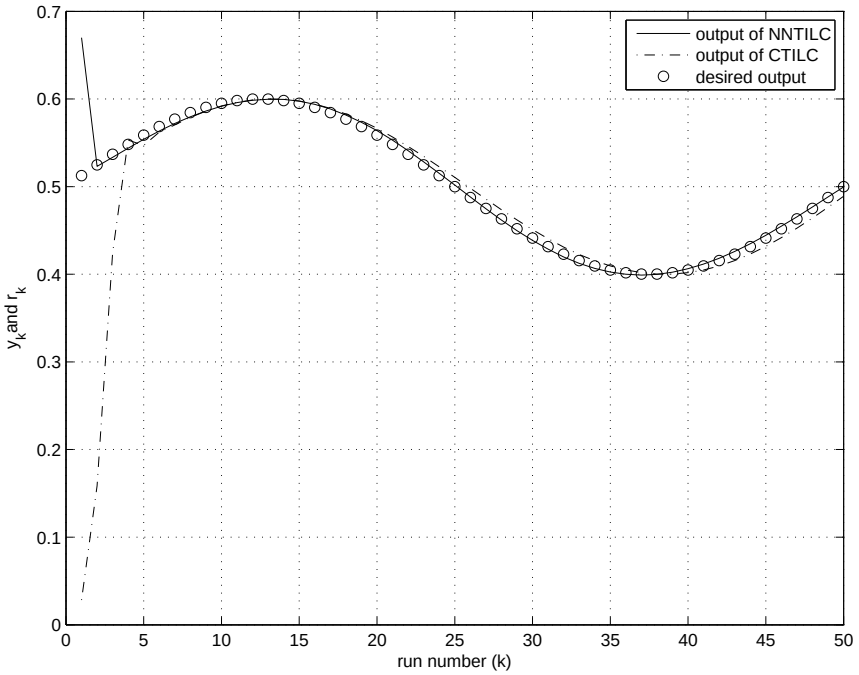


Figure 6.3: Terminal output and reference in iteration domain

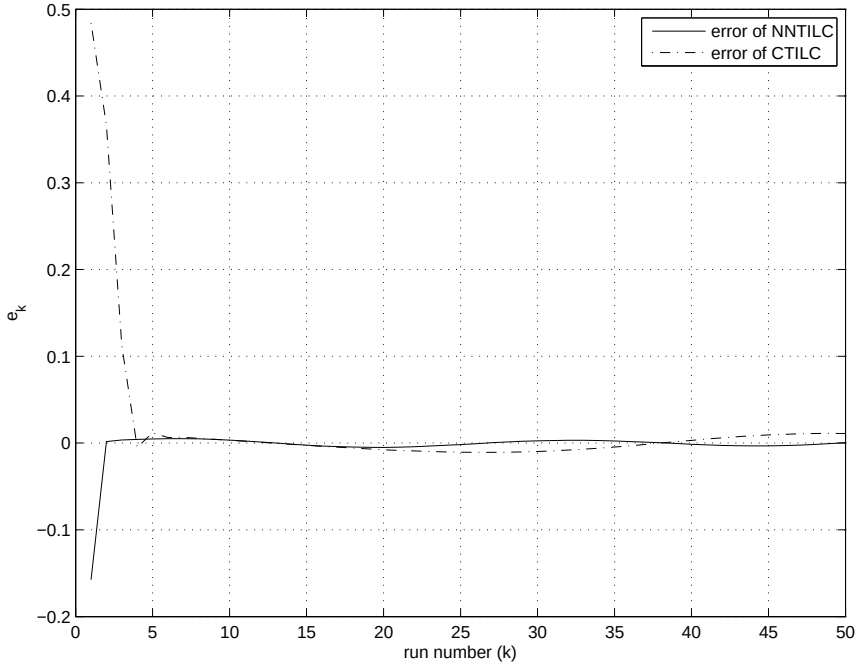


Figure 6.4: Tracking error

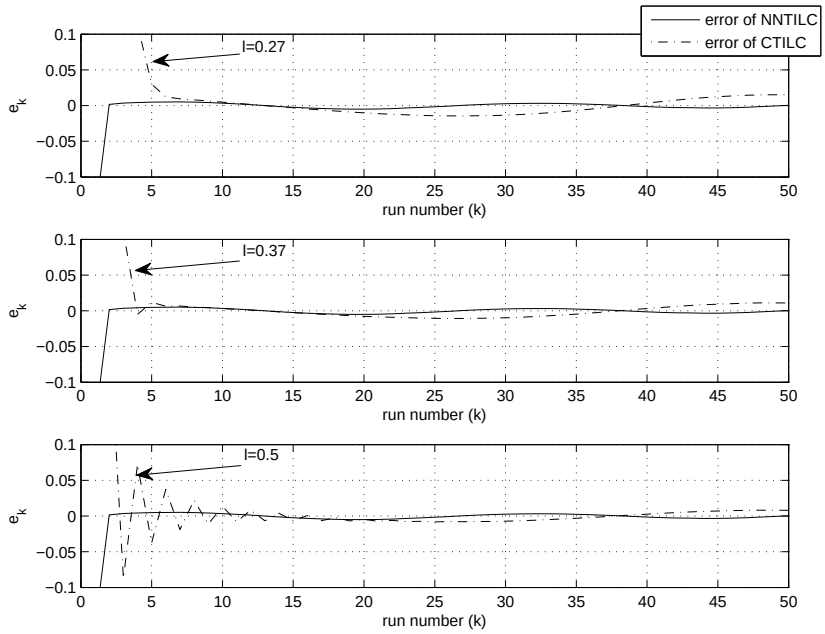


Figure 6.5: Tracking error with different l

Table 6.1: Performance comparison of different controllers

		CTILC		
Item	NNTILC	$l = 0.27$	$l = 0.37$	$l = 0.5$
Convergence time (iterations)	3	9	7	15
Average tracking error$\times 10^{-3}(\mu m)$	6.3	17	14	10
Highest overshoot$\times 10^{-3}(\mu m)$	-	-	-	90

6.6 Summary

This chapter extends the application of neural network based terminal iterative learning control scheme to nonlinear non-affine systems. The new control scheme estimates the system's dynamics by a simplified static RBF neuron network. As a result, NNTILC releases the requirement of identical initial condition. Besides, it handles iteration-varying reference effectively. Compared with conventional TILC, NNTILC achieves both better convergence rate and static tracking performance in the presence of initial states variance and reference changes. The convergence of NNTILC is proved and numerical simulation results confirm the effectiveness of the

proposed method.

Table 6.2: Parameters in the thickness control system

Symbol	*Value*	*Meaning*	*Unit*
A_q		Quartz window area	cm^2
A_w	400	Wafer area	cm^2
E_w	0.8	wafer emissivity	no unit
f	0.5	Lamp power absorbed by wafer	no unit
ϵ		Heat transfer coefficient for forced convection	$cal/cm^2/s/K$
P	$\in [0, 1]$	Lamp power control factor	no unit
Q	1076	Lamp power constant	cal/s
R	1.98	Gas constant	$cal/(gmolK)$
S		Polysilicon deposition thickness	μm
T_w		Wafer temperature	K
T_q		Quartz window temperature	K
T_{abm}		Ambient temperature	K
M_w	1	Wafer thermal mass	cal/K
M_q	100	Quartz window thermal mass	cal/K
k_0	591000	Pre-exponential constant of polysilicon deposition	$\mu m/s$
γ	39200	Activation energy of polysilicon deposition	$cal/g/mol$
σ	1.356×10^{-12}	Boltzmann constant	$cal/(s \cdot cm^2 \cdot K^2)$

Chapter 7

Conclusion and Future Research

7.1 Conclusion

Repetitive control and iterative learning control are both powerful control schemes for repetitive operational systems. This thesis aims to improve the performance of RC and TILC in the following aspects: better adaptiveness, better robustness, more flexible selectivity and faster transient response. Some conclusions of this thesis are summarized as follows:

(i) Parallel structure selective repetitive control scheme is developed. RC controller is implemented in two parallel branches. PSFRC realizes the selective compensation of any $nk \pm m$th harmonics with less memory and faster transient response. Besides, because the two groups of delay units are structured in parallel, connecting filters in series is avoided.

(ii) The parallel-structure scheme is further extended to any n branches. This extension further improves the flexibility of repetitive control and realizes the independent tuning and compensation for different harmonic groups. Besides, the correction factor introduced in each branch corrects the location of poles in fractional cases. As a result, the new parallel structure fractional repetitive control

approach achieves outstanding tracking performance even when the sampling frequency cannot be divided by fundamental frequency without remainder.

(iii) The proposed parallel structure fractional repetitive control is used to control a grid simulator to emulate various grid scenarios. An AC-DC-AC IGBT based setup is designed and implemented. Controlled by PSFRC, the proposed grid simulator emulate different types of voltage dips, voltages with wanted harmonics, voltage flickers and so on successfully.

(iv) Neural network is combined with terminal iterative learning control to overcome the problem of zero-error initial condition and tracking iteration-varying reference in ILC systems. The neural network approximates the dynamics of repetitive operational systems and estimates the output. Using the estimation from neural network, the propose NNTILC can drive the system to track iteration-varying reference with the presence of initial error.

(v) The NNTILC scheme is extended to nonlinear non-affine systems. The non-affine output dynamics of the system are converted affine and approximated by a static RBF neural network. The existence of the unique ideal input is proved and the convergence is analyzed mathematically. Simulation results are provided to illustrate the effectiveness of NNTILC in nonlinear non-affine systems.

7.2 Future research

Some future research works and possible extensions are identified as follows.

(i) Parallel structure fractional repetitive control can be adaptive to the changes of fundamental frequency without the limitation on N. It can be applied to converters connected to the grid in renewable power systems like wind turbines and PV systems. By tuning the correction factor according to the grid's fundamental frequency in real-time, PSFRC is capable to guarantee high-quality power.

(ii) The development of new topologies of converters is blooming in recent decades. The harmonic spectrum may be different from each other. Parallel structure selective repetitive control scheme provides more flexible solutions to the control problem in such applications. By analyzing the behavior of different types of converters, optimal parallel structure selective repetitive control schemes can be designed to offer faster and more robust control scheme.

(iii) In this thesis, the neural network based terminal iterative learning control scheme is discussed for SISO nonlinear non-affine systems only. As for MIMO systems, the dynamics will be more complex. In these cases, more powerful neural network may be required and this is another possible further research work.

AUTHOR'S PUBLICATIONS

1. Tianqi Liu, Danwei Wang, "Parallel structure fractional repetitive control for PWM inverters", *IEEE Trans. Industrial Electronics*, vol. 62, no. 8, pp. 5045-5054, 2015.

2. Tianqi Liu, Danwei Wang,"High-performance grid simulator using parallel structure fractional repetitive control", *IEEE Trans. Power Electronics*, vol. 31, no. 3, pp. 2669-2679, 2016.

3. Tianqi Liu, Danwei Wang and Ronghu Chi, "Neural network based terminal iterative learning control for uncertain nonlinear non-affine systems", *International Journal of Adaptive Control and Signal Processing*, vol. 29, no. 10, pp. 1274-1286, 2015.

4. Tianqi Liu, Danwei Wang, "Improved parallel structure selective repetitive control for PWM converters", submitted to *IEEE Trans. Industrial Electronics*.

5. Tianqi Liu, Danwei Wang, "Parallel Structure 6k $\pm$ 1 Repetitive Control for Three-phase PWM Inverters", The 10th IEEE Conference on Industrial Electronics and Applications (ICIEA 2015), in print.

6. Tianqi Liu, Danwei Wang, "Robust Repetitive Controller for Compensation of Odd-harmonic Components in PWM converters", Control Automation Robotic-

s & Vision (ICARCV), 2014 13th International Conference on, pp. 1444-1448. IEEE, 2014.

7. Tianqi Liu, Danwei Wang, and Ronghu Chi. "Neural network based terminal iterative learning control for tracking run-varying reference point with initial state variance", In Control, Automation and Systems (ICCAS), 2013 13th International Conference on, pp. 213-218. IEEE, 2013.

8. Tianqi Liu, Danwei Wang, Ronghu Chi, and Qiang Shen. "Neural network based terminal iterative learning control for tracking run-varying reference point", In Control Conference (ASCC), 2013 9th Asian, pp. 1-5. IEEE, 2013.

Bibliography

[1] B. Francis and W. Wonham, "The internal model principle of control theory," *Automatica*, vol. 12, no. 5, pp. 457–465, 1976.

[2] K. Zhou, K. S. Low, Y. Wang, F. L. Luo, B. Zhang, and Y. Wang, "Zero-phase odd-harmonic repetitive controller for a single-phase pwm inverter," *IEEE Transactions on Power Electronics*, vol. 21, no. 1, pp. 193–201, 2006.

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

[4] Y. Wang, F. Gao, and F. J. D. Iii, "Survey on iterative learning control, repetitive control, and run-to-run control," *Journal of Process Control*, vol. 19, no. 10, pp. 1589–1600, 2009.

[5] T. Nakano, M. Kubo, T. Matsumoto, and H. Baba, "High accuracy control of a proton synchrotron magnet power supply," *Proceedings of the 8th IFAC World Congress*, pp. 216–221, 1981.

[6] S. Hara, T. Omata, and M. Nakano, "Synthesis of repetitive control systems and its application," in *IEEE Conference on Decision Control*, 1986, pp. 1387–1392.

[7] S. Hara, Y. Yamamoto, T. Omata, and M. Nakano, "Repetitive control system: a new type servo system for periodic exogenous signals," *IEEE Transactions on Automatic Control*, vol. 33, no. 7, pp. 659–668, 1988.

[8] T. Inoue, "Practical repetitive control system design," in *Proceedings of the 29th IEEE Conference on Decision and Control.*, 1991, pp. vol.3, 1673–1678.

[9] M. C. Tsai and W. S. Yao, "Design of a plug-in type repetitive controller for periodic inputs," *IEEE Transactions on Control Systems Technology*, vol. 10, no. 4, pp. 547–555, 2002.

[10] L. Guvenc, "Repetitive controller design in parameter space," in *Proceedings of the American Control Conference*, 2001, pp. 2749–2754 vol.4.

[11] S. Hara, T. Omata, and M. Nakano, "Stability condition and synthesis methods for repetitive control systems," *Transactions of the Society of Instrument Control Engineers*, vol. 22, no. 1, pp. 36–42, 1986.

[12] T. E. Peery and H. Ozbay, "On h_∞ optimal repetitive controllers," in *Proceedings of the 32nd IEEE Conference on Decision and Control*, 1993, pp. 1146–1151 vol.2.

[13] T. Hornik and Q. C. Zhong, "A current-control strategy for voltage-source inverters in microgrids based on h_∞ and repetitive control," *IEEE Transactions on Power Electronics*, vol. 26, no. 3, pp. 943–952, 2011.

[14] G. Weiss, Q. C. Zhong, T. C. Green, and J. Liang, "H_∞ repetitive control of dc-ac converters in microgrids," *IEEE Transactions on Power Electronics*, vol. 19, no. 1, pp. 219–230, 2004.

[15] T. Hornik and Q. C. Zhong, "H_∞ repetitive voltage control of gridconnected inverters with a frequency adaptive mechanism," *IET Power Electronics*, vol. 3, no. 6, pp. 925–935, 2010.

[16] J. H. Moon, M. N. Lee, and M. J. Chung, "Repetitive control for the track-following servo system of an optical disk drive," *IEEE Transactions on Control Systems Technology*, vol. 6, no. 5, pp. 663–670, 1998.

[17] T. Y. Doh, J. R. Ryoo, and M. J. Chung, "Repetitive controller design for track-following servo system of an optical disk drive," in *7th International Workshop on Advanced Motion Control*, 2002, pp. 176–181.

[18] M. Tomizuka, T. C. Tsao, and K. K. Chew, "Analysis and synthesis of discrete-time repetitive controllers," *Journal of Dynamic Systems Measurement Control*, vol. 111, no. 3, pp. 353–358, 1989.

[19] G. F. Ledwich and A. Bolton, "Repetitive and periodic controller design," *Iee Proceedings D*, vol. 140, no. 1, pp. 19–24, 1993.

[20] F. R. Shaw and K. Srinivasan, "Discrete-time repetitive control system design using the regeneration spectrum," *Journal of Dynamic Systems Measurement Control*, vol. 115, no. 2A, pp. 228–237, 1993.

[21] G. Hillerstrom and J. Sternby, "Application of repetitive control to a peristaltic pump," *Journal of Dynamic Systems Measurement Control*, vol. 116, no. 4, pp. 136–141, 1993.

[22] N. Sadegh, "Synthesis of a stable discrete-time repetitive controller for mimo systems," *Journal of Dynamic Systems Measurement Control*, vol. 117, no. 1, pp. 92–98, 1995.

[23] M. Yamada, Y. Funahashi, S. Ishihara, and M. Matsushita, "Extended discrete-time prototype repetitive controllers and its application," in *Proceedings of the IEEE Conference on Decision and Control*, 1996, pp. 3606–3611 vol.4.

[24] M. Yamada, Z. Riadh, and Y. Funahashi, "Design of discrete-time repetitive control system for pole placement and application," *IEEE/ASME Transactions on Mechatronics*, vol. 4, no. 2, pp. 110 – 118, 1999.

[25] D. Jeong and B. C. Fabien, "A discrete time repetitive control system for mimo plants," *Proceedings of the American Control Conference*, vol. 6, pp. 4295–4299 vol.6, 1999.

[26] C. Smith and M. Tomizuka, "Shock rejection for repetitive control using a disturbance observer," in *Proceedings of the 35th IEEE Conference on Decision and Control*, 1996, pp. 2503 – 2504.

[27] L. Zhou, M. Wu, and J. H. She, "Design of observer-based robust repetitive-control system," in *Proceedings of the 2009 international conference on Robotics and biomimetics*, 2009, pp. 1452 – 1457.

[28] S. Jinhua, Y. O. D. of Mechatronics, and J. M. Nakano, "Rejection of non-periodic disturbances in repetitive control systems," *Kongzhi Lilun Yu Yingyong/control Theory Applications*, vol. 18, 2001.

[29] M. Yamada and Y. Funahashi, "Sensitivity reduction of adaptive repetitive control system with asymptotical rejection against uncertain periodic disturbance." in *IEEE International Conference on Control Applications (CCA)*, 2010, pp. 1672 – 1677.

[30] R. Costa-Castelló, R. Grinó, and E. Fossas, "Odd-harmonic digital repetitive control of a single-phase current active filter," *IEEE Transactions on Power Electronics*, vol. 19, no. 4, pp. 1060–1068, 2004.

[31] K. Zhou and D. Wang, "Digital repetitive learning controller for three-phase cvcf pwm inverter," *IEEE Transactions on Industrial Electronics*, vol. 48, no. 4, pp. 820–830, 2001.

[32] P. Mattavelli and F. P. Marafao, "Repetitive-based control for selective harmonic compensation in active power filters," *IEEE Transactions on Industrial Electronics*, vol. 51, no. 5, pp. 1018–1024, 2004.

[33] S. Jiang, D. Cao, Y. Li, J. Liu, and F. Z. Peng, "Low-thd, fast-transient, and cost-effective synchronous-frame repetitive controller for three-phase ups inverters," *IEEE Transactions on Power Electronics*, vol. 27, no. 6, pp. 2994–3005, 2012.

[34] P. Loh, Y. Tang, F. Blaabjerg, and P. Wang, "Mixed-frame and stationary-frame repetitive control schemes for compensating typical load and grid harmonics," *IET power electronics*, vol. 4, no. 2, pp. 218–226, 2011.

[35] K. Zhou, D. Wang, B. Zhang, and Y. Wang, "Plug-in dual-mode-structure repetitive controller for cvcf pwm inverters," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 3, pp. 784–791, 2009.

[36] W. Lu, K. Zhou, D. Wang, and M. Cheng, "A general parallel structure repetitive control scheme for multiphase dc–ac pwm converters," *IEEE Transactions on Power Electronics*, vol. 28, no. 8, pp. 3980–3987, 2013.

[37] W. Lu, K. Zhou, and D. Wang, "General parallel structure digital repetitive control," *International Journal of Control*, vol. 86, no. 1, pp. 70–83, 2013.

[38] D. Chen, J. Zhang, and Z. Qian, "An improved repetitive control scheme for grid-connected inverter with frequency-adaptive capability," *IEEE Transactions on Industrial Electronics*, vol. 60, no. 2, pp. 814–823, 2013.

[39] Y. Ye, K. Zhou, B. Zhang, D. Wang, and J. Wang, "High-performance repetitive control of pwm dc-ac converters with real-time phase-lead fir filter," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 53, no. 8, pp. 768–772, 2006.

[40] Y. R. Teo and A. J. Fleming, "A new repetitive control scheme based on non-causal fir filters," in *American Control Conference (ACC), 2014*. IEEE, 2014, pp. 991–996.

[41] J. Bao and R. W. Longman, "Enhancements of repetitive control using specialized fir zero-phase filter designs (aas 07-340)," *Advances in the astronautical sciences*, vol. 129, no. 2, p. 1413, 2008.

[42] G. Escobar, M. Hernandez-Gomez, G. A. Catzin, P. R. Martinez-Rodriguez, and A. A. Valdez-Fernandez, "Implementation of repetitive controllers subject to fractional delays," *IEEE Transactions on Industrial Electronics*, vol. 20, pp. 5985–5990, 2013.

[43] S. Jiang, D. Cao, and F. Z. Peng, "High performance repetitive control for three-phase cvcf pwm inverter using a 4th-order linear phase iir filter," in *Twenty-Seventh Annual IEEEApplied Power Electronics Conference and Exposition (APEC)*. IEEE, 2012, pp. 225–231.

[44] Z.-X. Zou, K. Zhou, Z. Wang, and M. Cheng, "Frequency-adaptive fractional-order repetitive control of shunt active power filters," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 3, pp. 1659–1668, 2015.

[45] Z. Zou, K. Zhou, Z. Wang, and M. Cheng, "Fractional-order repetitive control of programmable ac power sources," *IET Power Electronics*, vol. 7, no. 2, pp. 431–438, 2014.

[46] S. Dasgupta, S. K. Sahoo, and S. K. Panda, "Single-phase inverter control techniques for interfacing renewable energy sources with microgridpart i: Parallel-connected inverter topology with active and reactive power flow control along with grid current shaping," *IEEE Transactions on Power Electronics*, vol. 26, no. 3, pp. 717–731, 2011.

[47] S. Dasgupta, S. K. Sahoo, S. K. Panda, and G. Amaratunga, "Single-phase inverter-control techniques for interfacing renewable energy sources with microgridpart ii: Series-connected inverter topology to mitigate voltage-related problems along with active power flow control," *IEEE Transactions on Power Electronics*, vol. 26, no. 3, pp. 732–746, 2011.

[48] F. Gonzalez-Espin, P. Mattavelli, E. Figueres, G. Garcera, and R. Foley, "A variable multi-rate plug in repetitive controller for single phase inverters operation in the islanding mode," in *Proc. 15th International Power Electronics and Motion Control Conference,(EPE-PEMC2012)*, 2012, p. 6.

[49] M. Stefanello, H. Pinheiro, and H. Grundling, "Multi-rate repetitive controller for ups applications," in *Industry Applications Conference, 2007. 42nd IAS Annual Meeting. Conference Record of the 2007 IEEE.* IEEE, 2007, pp. 2340–2345.

[50] G. Pipeleers, B. Demeulenaere, J. De Schutter, and J. Swevers, "Robust high-order repetitive control: optimal performance trade-offs," *Automatica*, vol. 44, no. 10, pp. 2628–2634, 2008.

[51] M. Steinbuch, S. Weiland, and T. Singh, "Design of noise and period-time robust high-order repetitive control, with application to optical storage," *Automatica*, vol. 43, no. 12, pp. 2086–2095, 2007.

[52] G. A. Ramos and R. Costa-Castello?, "Power factor correction and harmonic compensation using second-order odd-harmonic repetitive control," *IET Control Theory & Applications*, vol. 6, no. 11, pp. 1633–1644, 2012.

[53] R. Costa-Castelló, G. A. Ramos, J. M. Olm, and M. Steinbuch, "Second-order odd-harmonic repetitive control and its application to active filter control," in *49th IEEE Conference on Decision and Control (CDC)*. IEEE, 2010, pp. 6967–6972.

[54] Z. Chen, K. Yamada, and T. Sakanushi, "A new design method of high-order modified repetitive control systems for reference inputs with uncertain period-time," *Mathematical Problems in Engineering*, vol. 2013, 2013.

[55] E. Kurniawan, Z. Cao, and Z. Man, "Design of robust repetitive control with time-varying sampling periods," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 6, pp. 2834–2841, 2014.

[56] S. Dasgupta, S. N. Mohan, S. K. Sahoo, and S. K. Panda, "Application of four-switch-based three-phase grid-connected inverter to connect renewable energy source to a generalized unbalanced microgrid system," *IEEE Transactions on Industrial Electronics*, vol. 60, no. 3, pp. 1204–1215, 2013.

[57] Y. Tzou, R. Ou, S. Jung, and M. Chang, "High-performance programmable ac power source with low harmonic distortion using dsp-based repetitive control technique," *IEEE Transactions on Power Electronics*, vol. 12, no. 4, pp. 715–725, 1997.

[58] R. Griñó, R. Cardoner, R. Costa-Castelló, and E. Fossas, "Digital repetitive control of a three-phase four-wire shunt active filter," *IEEE Transactions on Industrial Electronics*, vol. 54, no. 3, pp. 1495–1503, 2007.

[59] C. Cosner, G. Anwar, and M. Tomizuka, "Plug in repetitive control for industrial robotic manipulators," in *Robotics and Automation, 1990. Proceedings., 1990 IEEE International Conference on.* IEEE, 1990, pp. 1970–1975.

[60] K. Shabaani and M. Jalili-Kharaajoo, "Application of adaptive lqr with repetitive control for ups systems," *proceedings of IEEE Conference on Control Applications*, vol. 2, pp. 1124–1129 vol.2, 2003.

[61] H. A. Grundling, E. G. Carati, and J. R. Pinheiro, "Analysis and implementation of a modified robust model reference adaptive control with repetitive controller for ups applications," in *Proceedings of the 24th Annual Conference of the IEEE Industrial Electronics Society*, 1998, pp. 391 – 395.

[62] Q. C. Zhong, T. Green, J. Liang, and G. Weiss, "Robust repetitive control of grid-connected dc-ac converters," in *IEEE Conference on Decision Control*, 2002, pp. 2468 – 2473.

[63] K. Zhou and D. Wang, "Unified robust zero-error tracking control of cvcf pwm converters," *IEEE Transactions on Circuits Systems I Fundamental Theory Applications*, vol. 49, no. 4, pp. 492–501, 2002.

[64] X. Lin, S. Duan, Y. Kang, and J. Chen, "Waveform compensation of pwm inverter in ups," in *Proceedings of 2001 4th IEEE International Conference on Power Electronics and Drive Systems*, 2001, pp. 292 – 295.

[65] K. Zhou, D. Wang, and K. S. Low, "Periodic errors elimination in cvcf pwm dc/ac converter systems: repetitive control approach," *IEE Proceedings - Control Theory and Applications*, vol. 147, no. 6, pp. 694–700, 2000.

[66] T. Haneyoshi, A. Kawamura, and R. G. Hoft, "Waveform compensation of pwm inverter with cyclic fluctuating loads," *IEEE Transactions on Industry Applications*, vol. 24, no. 4, pp. 582–589, 1988.

[67] Y. Y. Tzou, S. L. Jung, and H. C. Yeh, "Adaptive repetitive control of pwm inverters for very low thd ac-voltage regulation with unknown loads," *IEEE Transactions on Power Electronics*, vol. 14, no. 5, pp. 973–981, 1999.

[68] Y. Y. Tzou and H. C. Yeh, "Dsp-based adaptive repetitive control of a pwm inverter for ups with very low harmonic distortion," in *Proceedings of the 1996 IEEE IECON 22nd International Conference on Industrial Electronics, Control, and Instrumentation*, 1996, pp. 1122–1127 vol.2.

[69] R. Grino, R. Costa-Castello, and E. Fossas, "Digital control of a single-phase shunt active filter," in *PESC Record - IEEE Annual Power Electronics Specialists Conference*, 2003, pp. 1038–1042 vol.3.

[70] K. Amanuma, M. Fuwa, and Y. Sakaki, "High accurate ripple reducing method based on the repetitive control," in *Power Electronics Specialists Conference, PESC '94 Record., 25th Annual IEEE*, 1994, pp. 571–576 vol.1.

[71] R. Grino, R. Cardoner, R. Costa-Castello?, and E. Fossas, "Digital repetitive control of a three-phase four-wire shunt active filter," *IEEE Transactions on Industrial Electronics*, vol. 54, no. 3, pp. 1495–1503, 2007.

[72] R. Costa-Castello, R. Grino, and E. Fossas, "Odd-harmonic digital repetitive control of a single-phase current active filter," *IEEE Transactions on Power Electronics*, vol. 19, no. 4, pp. 1060 – 1068, 2004.

[73] A. D. L. Roux, H. D. T. Mouton, and H. Akagi, "Dft-based repetitive control of a series active filter integrated with a 12-pulse diode rectifier," in *Power*

Electronics Specialists Conference, 2007. PESC 2007. IEEE, 2007, pp. 2997–3002.

[74] P. Mattavelli, L. Tubiana, and M. Zigliotto, "Torque-ripple reduction in pm synchronous motor drives using repetitive current control," *IEEE Transactions on Power Electronics*, vol. 20, no. 6, pp. 1423–1431, 2005.

[75] S. L. Chen and T. H. Hsieh, "Repetitive control design and implementation for linear motor machine tool," *International Journal of Machine Tools Manufacture*, vol. 47, no. 12-13, pp. 1807–1816, 2007.

[76] T. Su, S. Hattori, M. Ishida, and T. Hori, "Suppression control method for torque vibration of brushless dc motor utilizing repetitive control with fourier transform," in *Proceedings of 6th International Workshop on Advanced Motion Control*, 2000, pp. 427 – 432.

[77] F. Kobayashi, S. Hara, and H. Tanaka, "Reduction of motor speed fluctuation using repetitive control," in *Proceedings of the 29th IEEE Conference on Decision and Control*, 1990, pp. 1697–1702 vol.3.

[78] M. Vijayakarthick, S. Sathishbabu, and P. K. Bhaba, "Real time implementation of modified repetitive control strategy in a dc motor." in *Control Automation Robotics amp; Vision (ICARCV), 2010 11th International Conference on*, 2010, pp. 109–113.

[79] M. Vijayakarthick and P. K. Bhaba, "Position tracking performance of ac servomotor based on new modified repetitive control strategy," *International Journal of Research Reviews in Applied Sciences*, vol. 10, pp. 119–128, 2012.

[80] M. Ishida, S. Higuchi, and T. Hori, "Reduction control of mechanical vibration of an induction motor with fluctuated torque load using repetitive controller,"

in *Proceedings of the IEEE International Conference on Industrial Technology*, 1995, pp. 533 – 537.

[81] J. Na, X. Ren, R. Costa-Castell, R. Gri?, and Y. Guo, "Repetitive control for systems with time-delays and application to robotic servo motor," *Lecture Notes in Computer Science*, vol. 7429, pp. 377–389, 2012.

[82] E. Tung, G. Anwar, and M. Tomizuka, "Low velocity friction compensation and feedforward solution based on repetitive control," in *American Control Conference, 1991*, 1991, pp. 2615 – 2620.

[83] W. Huang, L. Cai, and X. Tang, "Adaptive repetitive output feedback control for friction and backlash compensation of a positioning table," in *Proceedings of the 37th IEEE Conference on Decision and Control*, 1998, pp. 1250 – 1251.

[84] H. Fujimoto and T. Takemura, "High-precision control of ball-screw-driven stage based on repetitive control using -times learning filter," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 7, pp. 3694–3703, 2014.

[85] S. Yang, M. A. Yan, and S. U. Bao ku, "Friction compensation in mechanical bearing control system," *Journal of Chinese Inertial Technology*, vol. 19, no. 6, pp. 749–753, 2011.

[86] M. Tomizuka, "On the compensation of friction forces in precision motion control," in *Motion Control Proceedings, 1993., Asia-Pacific Workshop on Advances in*, 1993, pp. 69–74.

[87] Z. Jamaludin, H. V. Brussel, G. Pipeleers, and J. Swevers, "Accurate motion control of xy high-speed linear drives using friction model feedforward and cutting forces estimation," *CIRP Annals - Manufacturing Technology*, vol. 57, no. 1, pp. 403–406, 2008.

[88] C. L. Chen and T. C. Chiu, "Position-dependent disturbance rejection using spatial-sampling robust repetitive control with actuator saturation and load uncertainty," in *ASME 2002 International Mechanical Engineering Congress and Exposition*, 2002, pp. 83–88.

[89] S. Hattori, M. Ishida, and T. Hori, "Reduction control method of torque vibration for brushless dc motor utilizing repetitive control with fourier transform," *Transactions of the Society of Instrument Control Engineers*, vol. 36, no. 5, pp. 427–432, 2000.

[90] M. Nakamura, H. Hata, Y. Nakamura, T. Endo, and K. Iizuka, "Vibration reduction in rolling piston-type compressors through motor torque control : Experimental study on control effects," *Jsme International Journal.ser Vibration Control Engineering Engineering for Industry*, vol. 34, no. 3, pp. 438–447, 1991.

[91] L. V. Chirayath and R. Narcissstarbell, "A pv micro-inverter system using repetitive current control," *International Journal of Engineering Advanced Technology*, no. 2, pp. 313–316, 2012.

[92] K. Zhov, Y. Yang, and F. Blaabjerg, "Frequency adaptive repetitive control of grid-tied single-phase pv inverters," in *IEEE Energy Conversion Congress and Exposition (ECCE)*, 2015.

[93] L. Liang, S. Su, and J. Zhu, "Study of three-phase photovoltaic grid system based on the proportional resonance repetitive control," *Electric Power Science Engineering*, 2014.

[94] C. L. Hao, L. L. Xie, X. Li, T. Wang, and J. Zhang, "Research on repetitive control strategy of photovoltaic grid-connected inverter," *Advanced Materials Research*, vol. 933, pp. 510–515, 2014.

[95] X. Chen, "Research of control strategy on grid-connected photovohaic inverter based on the repetitive control," *Application of Electronic Technique*, vol. 34, no. 7, pp. 72–75, 2008.

[96] T. Liu, X. Hao, X. Yang, and M. Zhao, "A novel repetitive control scheme for three-phase grid-connected inverter with lcl filter," in *7th International Power Electronics and Motion Control Conference (IPEMC)*, 2012, pp. 335–339.

[97] H. S. Ahn, "Design of a repetitive control system for a piezoelectric actuator based on the inverse hysteresis model," in *Proceedings of International Conference on Control and Automation, Icca*, 2003, pp. 128 – 132.

[98] D. Jin, D. Sun, W. Chen, D. Wang, and L. Tong, "Static shape control of repetitive structures integrated with piezoelectric actuators," *Smart Materials Structures*, vol. 14, no. 6, pp. 1410–1420(11), 2005.

[99] B. S. Kim, J. Li, and T. C. Tsao, "Two-parameter robust repetitive control with application to a novel dual-stage actuator for noncircular machining," *IEEE/ASME Transactions on Mechatronics*, vol. 9, no. 4, pp. 644–652, 2004.

[100] C. X. Li, G. Y. Gu, M. J. Yang, and L. M. Zhu, "High-speed tracking of a nanopositioning stage using modified repetitive control," *IEEE Transactions on Automation Science Engineering*, pp. 1–11, 2015.

[101] M. Uchiyama, "Formation of high-speed motion pattern of a mechanical arm by trial," *Transactions of the Society of Instrument Control Engineers*, vol. 14, pp. 706–712, 1978.

[102] 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.

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

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

[105] 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.

[106] P. B. Goldsmith, "On the equivalence of causal lti iterative learning control and feedback control," *Automatica*, vol. 38, no. 4, pp. 703–708, 2002.

[107] D. H. Owens and E. Rogers, "Comments on on the equivalence of causal lti iterative learning control and feedback control," *Automatica*, vol. 40, no. 5, pp. 895–898, 2004.

[108] P. B. Goldsmith, "Author's reply to: "comments on: 'on the equivalence of causal lti iterative learning control and feedback control' " [automatica j. ifac 40]," *Automatica J Ifac*, no. 5, pp. 895–898, 2004.

[109] S. Arimoto and T. Naniwa, "Learnability and adaptability from the viewpoint of passivity analysis," *Intelligent Automation Soft Computing*, vol. 8, no. 2, pp. 71–94, 2002.

[110] H.-S. Lee and Z. Bien, "Design issues on robustness and convergence of iterative learning controller," *Intelligent Automation Soft Computing*, vol. 8, no. 2, pp. 95–106, 2002.

[111] M. Norrlöf and S. Gunnarsson, "Time and frequency domain convergence properties in iterative learning control," *International Journal of Control*, vol. 75, no. 14, pp. 1114–1126, 2002.

[112] Y. Q. Chen and K. L. Moore, "Pi-type iterative learning control revisited," in *American Control Conference*, vol. 3, 2002, pp. 2138–2143.

[113] J. J. Hatonen, T. J. Harte, D. H. Owens, J. D. Ratcliffe, P. L. Lewin, and E. Rogers, "Discrete-time arimoto ilc-algorithm revisited," *IFAC Workshop on Adaptation and Learning in Control and Signal Processing (ALCOSP 04) and IFAC Workshop on Periodic Control Systems (PSYCO 04), Yokohama, Japan*, pp. 541–546, 2004.

[114] J. xin Xu and Y. Tan, "Brief on the p-type and newton-type ilc schemes for dynamic systems with non-affine-in-input factors," *Automatica*, vol. 38, no. 7, pp. 1237–1242, 2002.

[115] X. G. Yan, I. M. Chen, and J. Lam, "D-type learning control for nonlinear time-varying systems with unknown initial states and inputs," *Transactions of the Institute of Measurement Control*, vol. 23, no. 2, pp. 69–82, 2001.

[116] S. J. Yu, J. H. Wu, and X. W. Yan, "A pd-type open-closed-loop iterative learning control and its convergence for discrete systems," in *Proceedings of 2002 International Conference on Machine Learning and Cybernetics*, 2002, pp. 659–662 vol.2.

[117] Y. Q. Chen and K. L. Moore, "On d"-type iterative learning control," in *Proceedings of the 40th IEEE Conference on Decision and Control*, 2001, pp. 4451–4456 vol.5.

[118] Y. Li, Y. Q. Chen, and H. Ahn, "Fractionaliterative learning control for fractionallinear systems," *Asian Journal of Control*, vol. 13, no. 1, pp. 54–63, 2011.

[119] Y. Li, Y. Q. Chen, and H. S. Ahn, "Fractional order iterative learning control for fractional order system with unknown initialization," in *American Control Conference (ACC), 2014*, 2014, pp. 5712–5717.

[120] Y. Li, Y. Q. Chen, H. S. Ahn, and G. Tian, "A survey on fractional-order iterative learning control," *Journal of Optimization Theory Applications*, vol. 156, no. 1, pp. 127–140, 2013.

[121] R. W. Longman and S. L. Wirkander, "Automated tuning concepts for iterative learning and repetitive control laws," in *Proceedings of the 37th IEEE Conference on Decision and Control*, vol. 1.

[122] A. Madady, "Self tuning iterative learning control systems," in *Control Conference, 2004. 5th Asian*, vol. 3.

[123] ——, "Self-tuning iterative learning control for time variant systems," in *44th IEEE Conference on Decision and Control*, 2006, pp. 2445–2450.

[124] Y. Ming, G. Yang, Y. U. Yong, and X. D. Guo, "On-line self-tuning of pi controller for pmsm drives based on the iterative learning control," *Electric Machines Control*, vol. 3, pp. 1889–1893, 2005.

[125] D. H. Owens and G. Munde, "Adaptive iterative learning control," in *IEE Colloquium on Adaptive Control*, 1996, pp. 6–6.

[126] A. Tayebi, "Adaptive iterative learning control for robot manipulators ," *Automatica*, vol. 40, no. 7, pp. 1195–1203, 2004.

[127] A. Tayebi and S. Islam, "Adaptive iterative learning control for robot manipulators: Experimental results ," *Control Engineering Practice*, vol. 14, no. 7, pp. 843–851, 2006.

[128] C. J. Chien, C. T. Hsu, and C. Y. Yao, "Fuzzy system-based adaptive iterative learning control for nonlinear plants with initial state errors," *IEEE Transactions on Fuzzy Systems*, vol. 12, no. 5, pp. 724–732, 2004.

[129] J. Y. Choi and J. S. Lee, "Adaptive iterative learning control of uncertain robotic systems," *IEE Proceedings - Control Theory and Applications*, vol. 147, no. 2, pp. 217–223, 2000.

[130] K. E. Avrachenkov, H. S. Beigi, and R. W. Longman, "Updating procedures for iterative learning control in hilbert space," *Intelligent Automation Soft Computing*, vol. 1, no. 2, pp. 183–189, 2002.

[131] W. Ying-Chung, C. Chiang-Ju, and T. Ching-Cheng, "Direct adaptive iterative learning control of nonlinear systems using an output-recurrent fuzzy neural network." *IEEE Transactions on Systems Man Cybernetics Part B Cybernetics A Publication of the IEEE Systems Man Cybernetics Society*, vol. 34, no. 3, pp. 1348–1359, 2004.

[132] P. Jiang and R. Unbehauen, "Iterative learning neural network control for nonlinear system trajectory tracking ," *Neurocomputing*, vol. 48, no. s 1−4, pp. 141–153, 2002.

[133] C.-J. Chien and L.-C. Fu, "An iterative learning control of nonlinear systems using neural network design," *Asian Journal of Control*, vol. 4, no. 1, pp. 21−−29, 2002.

[134] M. Mailah, "Intelligent active force control of a rigid robot arm using neural network and iterative learning algorithms," *University of Dundee*, 1998.

[135] R. Precup, S. Preitl, J. K. Tar, M. L. Tomescu, M. Takacs, P. Korondi, and P. Baranyi, "Fuzzy control system performance enhancement by iterative learning control," *IEEE Transactions on Industrial Electronics*, vol. 55, no. 9, pp. 3461–3475, 2008.

[136] C. J. Chien, "A combined adaptive law for fuzzy iterative learning control of nonlinear systems with varying control tasks," *IEEE Transactions on Fuzzy Systems*, vol. 16, no. 1, pp. 40–51, 2008.

[137] C.-J. Chien, "A sampled-data iterative learning control using fuzzy network design," *International Journal of Control*, vol. 73, no. 10, pp. 902–913, 2000.

[138] Y. M. Pok, K. H. Liew, and J. X. Xu, "Fuzzy pd iterative learning control algorithm for improving tracking accuracy," in *IEEE International Conference on Systems, Man, and Cybernetics*, vol. 2.

[139] 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.

[140] G. Gauthier and B. Boulet, "Terminal iterative learning control design with singular value decomposition decoupling for thermoforming ovens," in *American Control Conference, 2009. ACC'09.* IEEE, 2009, pp. 1640–1645.

[141] S. Mondal, Y. Yun, and W. Chung, "Terminal iterative learning control for calibrating systematic odometry errors in mobile robots," in *Advanced Intelligent Mechatronics (AIM), 2010 IEEE/ASME International Conference on.* IEEE, 2010, pp. 311–316.

[142] Z. Hou, Y. Wang, C. Yin, and T. Tang, "Terminal iterative learning control based station stop control of a train," *International Journal of Control*, vol. 84, no. 7, pp. 1263–1274, 2011.

[143] D. Meng and Y. Jia, "Finite-time consensus for multi-agent systems via terminal feedback iterative learning," *Control Theory & Applications, IET*, vol. 5, no. 18, pp. 2098–2110, 2011.

[144] 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.

[145] 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.

[146] T. Son and H. Ahn, "Terminal iterative learning control with multiple intermediate pass points," in *American Control Conference (ACC), 2011.* IEEE, 2011, pp. 3651–3656.

[147] T. Son, D. Nguyen, and H. Ahn, "An interpolation method of multiple terminal iterative learning control," in *Intelligent Control (ISIC), 2011 IEEE International Symposium on.* IEEE, 2011, pp. 1528–1533.

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

[149] 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.

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

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

[152] M. Sun and D. Wang, "Initial condition issues on iterative learning control for non-linear systems with time delay," *International Journal of Systems Science*, vol. 32, no. 11, pp. 1365–1375, 2001.

[153] K.-H. Park and Z. Bien, "A generalized iterative learning controller against initial state error," *International Journal of Control*, vol. 73, no. 10, pp. 871–881, 2000.

[154] M. Sun and D. Wang, "Closed-loop iterative learning control for non-linear systems with initial shifts," *International Journal of Adaptive Control Signal Processing*, vol. 16, no. 7, pp. 515–538, 2002.

[155] F. Yong, Y. C. Soh, and G. G. Feng, "Convergence analysis of iterative learning control with uncertain initial conditions," in *Proceedings of the 4th World Congress on Intelligent Control and Automation*, vol. 2, 2002, pp. 960–963.

[156] M. Sun and D. Wang, "Iterative learning control with initial rectifying action," *Automatica*, vol. 38, no. 7, pp. 1177–1182, 2002.

[157] K.-H. Park, "A study on the robustness of a pid-type iterative learning controller against initial state error," *International Journal of Systems Science*, vol. 30, no. 1, pp. 49–59, 1999.

[158] M. Sun and D. Wang, "Initial shift issues on discrete-time iterative learning control with system relative degree," *IEEE Transactions on Automatic Control*, vol. 48, no. 1, pp. 144–148, 2003.

[159] S. Y. Yang, X. P. Fan, and A. Luo, "Experience based acquisition of the initial value for the iterative learning control inputs," *Control Decision*, vol. 19, no. 1, pp. 27–26, 2004.

[160] S. Saab, W. Vogt, and M. Mickle, "Learning control algorithms for tracking slowly varying trajectories," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 27, no. 4, pp. 657–670, 1997.

[161] X. Ruan, Z. Bien, and K. H. Park, "Decentralized iterative learning control to large-scale industrial processes for nonrepetitive trajectory tracking," *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, vol. 38, no. 1, pp. 238–252, 2008.

[162] J. Xu and J. Xu, "On iterative learning from different tracking tasks in the presence of time-varying uncertainties," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 34, no. 1, pp. 589–597, 2004.

[163] R. Chi, Z. Hou, and J. Xu, "Adaptive ilc for a class of discrete-time systems with iteration-varying trajectory and random initial condition," *Automatica*, vol. 44, no. 8, pp. 2207–2213, 2008.

[164] S. S. Saab, "On a discrete-time stochastic learning control algorithm," *IEEE Transactions on Automatic Control*, vol. 46, no. 8, pp. 1333–1336, 2001.

[165] W. Paszke, K. Galkowski, E. Rogers, and D. H. Owens, "H control of discrete linear repetitive processes," in *Proceedings of the IEEE Conference on Decision and Control*, vol. 1.

[166] Z. Qu and J. Xu, "Model-based learning controls and their comparisons using lyapunov direct method," *Asian Journal of Control*, vol. 4, no. 1, pp. 99–110, 2002.

[167] Y. P. Tian and X. Yu, "Robust learning control for a class of uncertain non-linear systems," *World Congress*, no. 6, pp. 912–914, 2002.

[168] S. Gunnarsson and M. Norrlof, "Some aspects of an optimization approach to iterative learning control," in *Proceedings of the 38th IEEE Conference on Decision and Control*, 1999, pp. 1581 – 1586.

[169] J. Hätönen and D. H. Owens, "Convex modifications to an iterative learning control law," *Automatica*, vol. 40, no. 7, pp. 1213 – 1220, 2004.

[170] V. Hatzikos, J. Hätönen, and D. H. Owens, "Genetic algorithms in norm-optimal linear and non-linear iterative learning control," *International Journal of Control*, vol. 77, no. 2, pp. 188–197, 2004.

[171] V. E. Hatzikos, J. Hatonen, T. Harte, and D. H. Owens, "Robust analysis of a genetic algorithm based optimization method for real-time iterative learning control applications," in *Proceedings of IEEE Conference Emerging Technologies and Factory Automation*, vol. 2.

[172] M. Rzewuski, E. Rogers, and D. H. Owens, "A comparison of optimal iterative learning control schemes," *Rzewuski M*, pp. 77–82, 2001.

[173] C. Zhu, Y. Aiyama, and T. Arai, "Releasing manipulation with learning control," in *Proceedings of IEEE International Conference on Robotics and Automation, 1999*, vol. 4.

[174] Z. W. H. Y. C. Yao, "A sort of iterative learning control algorithm for tracking of robot trajectory," *Acta Armamentarii*, vol. 25, no. 3, pp. 330–334, 2004.

[175] Y. Ye and D. Wang, "Multi-channel design for ilc with robot experiments," in *7th International Conference on Control, Automation, Robotics and Vision*, vol. 2.

[176] M. Yamada, X. Li, and O. Saito, "Iterative learning control of a robot manipulator using nd practical tracking approach," in *The 2004 47th Midwest Symposium on Circuits and Systems*, vol. 4.

[177] D. Sun and J. K. Mills, "Performance improvement of industrial robot trajectory tracking using adaptive-learning scheme," *Journal of Dynamic Systems Measurement Control*, vol. 121, no. 2, pp. 285–292, 1999.

[178] M. Norrlöf and S. Gunnarsson, "Experimental comparison of some classical iterative learning control algorithms," *IEEE Transactions on Robotics Automation*, vol. 18, no. 4, pp. 636–641, 2002.

[179] D. Liu, A. Konno, and M. Uchiyama, "Flexible manipulator trajectory learning control with input preshaping method," in *38th Annual Conference Proceedings of the SICE Annual*, 1999, pp. 967–972.

[180] P. Jiang and R. Unbehauen, "Robot visual servoing with iterative learning control," *IEEE Transactions on Systems Man and Cybernetics - Part A Systems and Humans*, vol. 32, no. 2, pp. 281–287, 2002.

[181] K. Hamamoto and T. Sugie, "Iterative learning control for robot manipulators using the finite dimensional input subspace," in *Proceedings of the 40th IEEE Conference on Decision and Control*, 2001, pp. 632–635.

[182] N. Sakagami, M. Inoue, and S. Kawamura, "Theoretical and experimental studies on iterative learning control for underwater robots," *International Journal of Offshore Polar Engineering*, vol. 13, no. 2, pp. 120–127, 2003.

[183] N. Sakagami and S. Kawamura, "Time optimal control for underwater robot manipulators based on iterative learning control and time-scale transformation," in *OCEANS 2003. Proceedings*, 2003, pp. 1180–1186 Vol.3.

[184] S. Kawamura and N. Sakagami, "Analysis on dynamics of underwater robot manipulators based on iterative learning control and time-scale transformation," in *Proceedings of IEEE International Conference on Robotics and Automation*, 2002, pp. 1088–1094.

[185] T. Watabe, M. Yamakita, T. Mita, and M. Ohta, "Output zeroing and iterative learning control for 3 link acrobat robot," in *Proceedings of the 41st SICE Annual Conference*, 2002, pp. 2579–2584 vol.4.

[186] M. Yamakita, T. Yonemura, Y. Michitsuji, and Z. Luo, "Stabilization of 3 link acrobat robot in upright position," in *Proceedings of the 41st SICE Annual Conference*, 2002, pp. 2996–3001 vol.5.

[187] M. Norrlöf, "Disturbance rejection using an ilc algorithm with iteration varying filters," *Asian Journal of Control*, vol. 6, no. 3, p. 432C438, 2004.

[188] J. D. Ratcliffe, T. J. Harte, J. J. Hatonen, P. L. Lewin, E. Rogers, and D. H. Owens, "Practical implementation of a model inverse optimal iterative learning controller on a gantry robot," *IFAC Workshop on Adaptation and Learning in Control and Signal Processing (ALCOSP 04) and IFAC Workshop on Periodic Control Systems (PSYCO 04), Yokohama, Japan*, pp. 687–692, 2004.

[189] J. J. Hatonen, T. J. Harte, D. H. Owens, J. D. Ratcliffe, P. L. Lewin, and E. Rogers, "A new robust iterative learning control algorithm for application on a gantry robot," in *Proceedings of IEEE Conference Emerging Technologies and Factory Automation*, 2003, pp. 305–312 vol.2.

[190] J. H. Lee, K. S. Lee, and W. C. Kim, "Model-based iterative learning control with a quadratic criterion for time-varying linear systems," *Automatica*, vol. 36, no. 5, pp. 641–657, 2000.

[191] K. S. Lee and J. H. Lee, "Convergence of constrained model-based predictive control for batch processes," *IEEE Transactions on Automatic Control*, vol. 45, no. 10, pp. 1928 – 1932, 2000.

[192] S. C. Lee, R. W. Longman, and M. Q. Phan, "Direct model reference learning and repetitive control," *Intelligent Automation Soft Computing*, vol. 8, no. 2, pp. 143–161, 2002.

[193] T. H. Lee, K. K. Tan, S. Y. Lim, and H. F. Dou, "Iterative learning control of permanent magnet linear motor with relay automatic tuning," *Mechatronics*, vol. 10, no. 1-2, pp. 169–190, 2000.

[194] K. K. Tan, H. Dou, Y. Q. Chen, and H. L. Tong, "High precision linear motor control via relay-tuning and iterative learning based on zero-phase filtering," *IEEE Transactions on Control Systems Technology*, vol. 9, no. 2, pp. 244–253, 2001.

[195] K. Mainali, S. K. Panda, J. X. Xu, and T. Senjyu, "Position tracking performance enhancement of linear ultrasonic motor using iterative learning control," in *PESC Record - IEEE Annual Power Electronics Specialists Conference*, 2004, pp. 4844–4849 Vol.6.

[196] S. Ss., "A stochastic iterative learning control algorithm with application to an induction motor," *International Journal of Control*, vol. 77, no. 77, pp. 144–163, 2004.

[197] Z. K. Shi, "Real-time learning control method and its application to acservomotor control," in *Proceedings of International Conference on Machine Learning and Cybernetics*, 2002, pp. 900–905 vol.2.

[198] J. X. Xu, T. H. Lee, and Y. Tan, "Enhancing trajectory tracking for a class of process control problems using iterative learning," *Engineering Applications of Artificial Intelligence*, vol. 15, no. 1, pp. 53–64, 2002.

[199] C. H. Tsai and C. J. Chen, "Application of iterative path revision technique for laser cutting with controlled fracture," *Optics Lasers in Engineering*, vol. 41, no. 1, pp. 189–204, 2004.

[200] Y. Tan, H. Sibarani, and Y. Samyudia, "Iterative learning strategy for a class of nonlinear controllers applied to constrained batch processes," in *5th Asian Control Conference*, 2004, pp. 1282–1289 Vol.2.

[201] Z. Xiong and J. Zhang, "Product quality trajectory tracking in batch processes using iterative learning control based on time-varying perturbation models," *Industrial Engineering Chemistry Research*, vol. 42, no. 26, pp. 6802–6814, 2003.

[202] J. X. Xu, Q. Hu, H. L. Tong, and S. Yamamoto, "Iterative learning control with smith time delay compensator for batch processes," in *Proceedings of the 2001 American Control Conference*, 2001, pp. 1972–1977 vol.3.

[203] M. Grundelius, "Iterative optimal control of liquid slosh in an industrial packaging machine," in *Proceedings of the IEEE Conference on Decision and Control*, 2000, pp. 3427–3432 vol.4.

[204] K. K. Tan, S. N. Huang, and S. Zhao, "A novel predictive and iterative learning control algorithm," *Control Intelligent Systems*, no. 1, pp. 1–9, 2003.

[205] Y. C. Huang, M. Chan, Y. P. Hsin, and C. C. Ko, "Use of iterative learning control on improving intra-oral hydraulic loading system of dental implants," in *IEEE International Symposium on Intelligent Control*, 2003, pp. 63–68.

[206] H. Dou, K. K. Tan, H. L. Tong, and Z. Zhou, "Iterative learning feedback control of human limbs via functional electrical stimulation," *Control Engineering Practice*, vol. 7, no. 3, pp. 315–325, 1999.

[207] H. Wu, Z. Zhou, S. Xiong, and W. Zhang, "Adaptive iteration learning control and its applications for fns multi," in *Proceedings of the 17th IEEE Instrumentation and Measurement Technology Conference*, 2000, pp. 983–987 vol.2.

[208] M. Hu, H. Du, S. F. Ling, Z. Zhou, and Y. Li, "Motion control of an electrostrictive actuator," *Mechatronics*, vol. 14, no. 03, pp. 153–161, 2001.

[209] J. Waissman, V, C. B. Youssef, and G. Vazquez, R, "Iterative learning control for a fedbatch lactic acid reactor," in *IEEE International Conference on Systems, Man and Cybernetics*, 2002.

[210] R. Griñó and R. Costa-Castelló, "Digital repetitive plug-in controller for odd-harmonic periodic references and disturbances," *Automatica*, vol. 41, no. 1, pp. 153–157, 2005.

[211] W. Lu, K. Zhou, D. Wang, and M. Cheng, "A generic digital nk ± m order harmonic repetitive control scheme for pwm converters," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 3, pp. 1516–1527, 2014.

[212] G. Escobar, P. G. Hernandez-Briones, P. R. Martinez, M. Hernandez-Gomez, and R. E. Torres-Olguin, "A repetitive-based controller for the compensation of $6l \pm 1$ harmonic components," *IEEE Transactions on Industrial Electronics*, vol. 55, no. 8, pp. 3150–3158, 2008.

[213] Y. Wang, D. Wang, B. Zhang, and K. Zhou, "Fractional delay based repetitive control with application to pwm dc/ac converters," in *IEEE International Conference on Control Applications*. IEEE, 2007, pp. 928–933.

[214] B. Singh, K. Al-Haddad, and A. Chandra, "A review of active filters for power quality improvement," *IEEE Transactions on Industrial Electronics*, vol. 46, no. 5, pp. 960–971, 1999.

[215] B. Singh, B. N. Singh, A. Chandra, K. Al-Haddad, A. Pandey, and D. P. Kothari, "A review of three-phase improved power quality ac-dc converters," *IEEE Transactions on Industrial Electronics*, vol. 51, no. 3, pp. 641–660, 2004.

[216] M. Prodanovic and T. C. Green, "High-quality power generation through distributed control of a power park microgrid," *IEEE Transactions on Industrial Electronics*, vol. 53, no. 5, pp. 1471–1482, 2006.

[217] I. W. Lee and P. K. Dash, "S-transform-based intelligent system for classification of power quality disturbance signals," *IEEE Transactions on Industrial Electronics*, vol. 50, no. 4, pp. 800–805, 2003.

[218] M. Prodanovic and T. C. Green, "Control and filter design of three-phase inverters for high power quality grid connection," *IEEE Transactions on Power Electronics*, vol. 18, no. 1, pp. 373–380, 2003.

[219] E. W. Gunther and H. Mebta, "A survey of distribution system power quality-preliminary results," *IEEE Transactions on Power Delivery*, vol. 10, no. 1, pp. 322–329, 1995.

[220] C. R. Baier, J. A. Munoz, J. R. Espinoza, P. E. Melín, J. I. Guzman, and L. A. Morán, "Improving power quality in cascade multilevel converters based on single-phase non-regenerative power cells," in *IECON 2011-37th Annual Conference on IEEE Industrial Electronics Society*. IEEE, 2011, pp. 4192–4197.

[221] J. He, Y. Li, and F. Blaabjerg, "Flexible microgrid power quality enhancement using adaptive hybrid voltage and current controller," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 6, pp. 2784–2794, 2014.

[222] Y. Chung, G. Kwon, T. Park, H. Kim, and J. Moon, "Voltage sag, swell and flicker generator with series injected inverter," in *Power Engineering Society General Meeting, 2005. IEEE.* IEEE, 2005, pp. 1308–1313.

[223] R. Zhang, M. Cardinal, P. Szczesny, and M. Dame, "A grid simulator with control of single-phase power converters in dq rotating frame," in *IEEE 33rd Annual Power Electronics Specialists Conference*, vol. 3. IEEE, 2002, pp. 1431–1436.

[224] R. Achar, M. S. Nakhla, H. S. Dhindsa, A. R. Sridhar, D. Paul, and N. M. Nakhla, "Parallel and scalable transient simulator for power grids via waveform relaxation (pts-pwr)," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 19, no. 2, pp. 319–332, 2011.

[225] J. Eloy-García, J. C. Vasquez, and J. M. Guerrero, "Grid simulator for power quality assessment of micro-grids," *IET Power Electronics*, vol. 6, no. 4, pp. 700–709, 2013.

[226] N. Kim, S.-Y. Kim, H.-G. Lee, C. Hwang, G.-H. Kim, H.-R. Seo, M. Park, and I.-K. Yu, "Design of a grid-simulator for a transient analysis of grid-connected renewable energy system," in *International Conference on Electrical Machines and Systems (ICEMS).* IEEE, 2010, pp. 633–637.

[227] S. Gulur, V. John *et al.*, "A grid simulator to evaluate control performance of grid-connected inverters," in *IEEE International Conference on Power Electronics, Drives and Energy Systems (PEDES).* IEEE, 2014, pp. 1–6.

[228] B. Tamyurek, "A high-performance spwm controller for three-phase ups systems operating under highly nonlinear loads," *IEEE Transactions on Power Electronics*, vol. 28, no. 8, pp. 3689–3701, 2013.

[229] C. Li, S.-m. Ji, and D.-p. Tan, "Multiple-loop digital control method for a 400-hz inverter system based on phase feedback," *IEEE Transactions on Power Electronics*, vol. 28, no. 1, pp. 408–417, 2013.

[230] F. Li, X. Wang, Z. Chen, and X. Zhang, "A laboratory grid simulator based on three-phase four-leg inverter: Design and implementation," in *International Conference on Electrical Machines and Systems (ICEMS)*. IEEE, 2011, pp. 1–5.

[231] C. Garcia, M. Rivera, M. López, J. Rodriguez, R. Pena, P. Wheeler, and J. Espinoza, "A simple current control strategy for a four-leg indirect matrix converter," *IEEE Transactions on Power Electronics*, vol. 30, no. 4, pp. 2275–2287, 2015.

[232] M. Zhang, D. J. Atkinson, B. Ji, M. Armstrong, and M. Ma, "A near-state three-dimensional space vector modulation for a three-phase four-leg voltage source inverter," *IEEE Transactions on Power Electronics*, vol. 29, no. 11, pp. 5715–5726, 2014.

[233] R. Zhang, V. H. Prasad, D. Boroyevich, and F. C. Lee, "Three-dimensional space vector modulation for four-leg voltage-source converters," *IEEE Transactions on Power Electronics*, vol. 17, no. 3, pp. 314–326, 2002.

[234] N.-Y. Dai, M.-C. Wong, F. Ng, and Y.-D. Han, "A fpga-based generalized pulse width modulator for three-leg center-split and four-leg voltage source inverters," *IEEE Transactions on Power Electronics*, vol. 23, no. 3, pp. 1472–1484, 2008.

[235] M. H. Bollen, "Algorithms for characterizing measured three-phase unbalanced voltage dips," *IEEE Transactions on Power Delivery*, vol. 18, no. 3, pp. 937–944, 2003.

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

[237] V. Elanayar, Y. Shin *et al.*, "Radial basis function neural network for approximation and estimation of nonlinear stochastic dynamic systems," *IEEE Transactions on Neural Networks*, vol. 5, no. 4, pp. 594–603, 1994.

[238] S. Haykin, *Adaptive Filetering Theory*, 3rd ed. New York: Prentice-Hall, 1996.

[239] A. Calise, N. Hovakimyan, and M. Idan, "Adaptive output feedback control of nonlinear systems using neural networks," *Automatica*, vol. 37, no. 8, pp. 1201–1211, 2001.

[240] J. Xu, Y. Chen, T. Lee, and S. Yamamoto, "Terminal iterative learning control with an application to rtpcvd thickness control," *Automatica*, vol. 35, no. 9, pp. 1535–1542, 1999.