Title
High Sampling Rate Dynamic Inversion - Digital Signal Processing, Filter Realizations and Applications in Digital Control

Author
Chang, Herrick

Publication Date
2012

Peer reviewed|Thesis/dissertation

University of California

Los Angeles

High Sampling Rate Dynamic Inversion - Digital Signal Processing, Filter Realizations and Applications in Digital Control

A dissertation submitted in partial satisfaction

of the requirements for the degree

Doctor of Philosophy in Mechanical Engineering

by

Herrick Lin Chang

2012

ABSTRACT OF THE DISSERTATION

High Sampling Rate Dynamic Inversion - Digital Signal Processing, Filter Realizations and Applications in Digital Control

by

Herrick Lin Chang

Doctor of Philosophy in Mechanical Engineering

University of California, Los Angeles, 2012

Professor Tsu-Chin Tsao, Chair

In the past few decades, computational power and speed has made it such that the Controls literature has moved away from addressing issues of finite-word-length (FWL) issues, quantization, and limited computational resources. On the other hand, the signal processing community has studied this issue extensively [Mit04, PB87]. In recent years, the introduction of Nano and Microelectromechanical systems (MEMS) with large bandwidth systems requires the use of high-sampling rate controllers. To satisfy such high sampling-rates, fixed-point based platforms such as Field Programmable Gate Arrays (FPGAs) and fixed-point micro-controllers are needed. This trend results in a need for high-sampling rate controllers that are more sophisticated than simple loop shaping while addressing the issues of FWL effects and limited computational resources. The aim of this dissertation is to introduce novel controllers that incorporate signal processing techniques and address these issues with controller design/realizations to control high bandwidth electro-mechanical systems.

The dissertation of Herrick Lin Chang is approved.

Alan N. Willson, Jr.

Robert T. M'Closkey

James S. Gibson

Tsu-Chin Tsao, Committee Chair

University of California, Los Angeles

2012

To my wife, for being infinitely patient and supportive.

Table of Contents

List of Figures

List of Tables

Acknowledgments

UCLA has served as my home for almost a decade where it has provided with invaluable engineering resources from both the Electrical Engineering and Mechanical Engineering and Aerospace Department. Furthermore, I never would have made it without the support of many key individuals.

First of all, I'd like to thank my advisor Professor Tsu-Chin "T-C" Tsao for being an unbelievable source of guidance and inspiration academically. His willingness to take a chance on me when I was an undergraduate is the main reason I continued my graduate studies as a member of the Mechatronics and Controls Laboratory. His wealth of knowledge, creativity and engineering intuition never ceases to amaze me. He is without a doubt my academic role model in which I aspire to emulate.

I am eternally grateful to all members of the Mechatronics and Controls Laboratory at UCLA. Yigang, Kevin, Jason, Chris L., Stephen, David, Chris K., Shalom, Yen-Chi, and Kuo-Tai have been great people to be around day in and day out. The family type environment allowed for many fruitful brainstorming sessions where we was able to bounce ideas off one another. Yigang and Kevin were extremely helpful in being my unofficial mentors and assisting me through graduate student life. Jason, thank you for keeping me in check and making sure I didn't sweat about the small stuff.

I'd also like to thank Alan Willson, Jr. , Robert M'Closkey, and Steve Gibson for being great committee members. To Professor Willson, I can't express how thankful I am for our discussion and being able to publish with you.

I need to thank my parents for instilling within me the necessity and the value of education. They are responsible for teaching me the value and the rewards of a good work ethic.

Lastly, I would like to thank my wife, Tina, for her unwavering support these past years. She has been so understanding about all the late nights in the lab. She is my best friend and the love of my life. Without her, I'm not sure where I would be today.

VITA

2007	B.S. (Electrical Engineering), University of California, Los Angeles (UCLA), Los Angeles, CA.
2009	M.S. (Mechanical Engineering), University of California, Los Angeles (UCLA), Los Angeles, CA.
2009	Best Paper Award (Theory), International Symposium of Flexible Automation, 2010
2010	Intern, Western Digital
2007-2011	Teaching Assistant, Mechanical and Aerospace Engineering Department, University of California, Los Angeles (UCLA), Los Angeles, CA
2007-2012	Graduate Student Researcher, Mechanical and Aerospace Engineering Department, University of California, Los Angeles (UCLA), Los Angeles, CA
2011-2012	National Science Foundation GK-12 Fellow

Publications

H. L. Chang and T.-C. Tsao, "Efficient fixed-point realization of approximate dynamic inversion compensators for nonminimum phase systems," in *Proceedings of the 2010 American Control Conference*, Jun. 2010, pp. 4193-4198.

H. L. Chang and T.-C. Tsao, "Rep/etitive control of a levitated shaft - FPGA implementation based on Powell-Chau filters," in *Proceedings of the 2010 International Symposium on Flexible Automation*, Jul. 2010.

Y. Wang, K. C. Chu, H. L. Chang, and T.-C. Tsao, "Laguerre based adaptive control of piezoelectric actuator for nanopositioning," in *Proceedings of the 2010 Conference on Decision and Control*, Dec. 2010.

H. L. Chang and T.-C. Tsao, "High sampling rate dynamic inversion - filter realization and applications in digital control," *IEEE/ASME Transactions on Mechatronics*, Accepted for Publication, 2012.

CHAPTER 1

Introduction

The investigation of advanced digital controllers for improving performance in high-bandwidth systems has gained much attention for performance critical applications such as precision non-circular machining [TCR87, HT98, RTH94, TT94, TSH08], scanning probe microscopy [PBA07, LD07, KL12], computer data storage [KT10, LKC10, LDW11, WWC11], and power electronics [CKK08, MZE04, WPX10, EML07]. These advanced controllers improve performance through improved trajectory planning, reference tracking, and disturbance rejection. However, the computational power necessary for these controllers tends to constrain the sampling rate of the digital control algorithm and in turn limits the control performance. Platforms capable of such computational power include digital signal processors (DSPs), micro-controllers, and personal computers. On one extreme, systems that require high servo rates could be implemented with analog circuits or hybrid platforms such as field programmable analog arrays (FPAAs) [SP08, YBR12] but with far less sophisticated control structures, which limits the control performance. On the other hand, the use of traditional micro-controllers, DSPs, and personal computers may not provide sufficient parallel processing threads to complete control computations in a small number of processor clock cycles for high order control algorithms that require fast sampling rates.

In recent years there has been a rise in popularity of controller implementations on Field Programmable Gate Arrays (FPGAs), a computational device with programmable logic gates, have a limited number of adders and multipliers for arithmetic operations [HRE09, MD11, MC07, JPK08]. Its popularity stems from its parallel processing

ability, low level interface to sensor and actuator electronics, and inherent configurability which allows for the customization and optimization for specific applications. Furthermore, the algorithms on FPGAs can be ported to application-specific integrated circuits (ASICs). The amount of resources used on an FPGA correlate to size and cost of the layout of an ASIC.

FPGAs can be configured to emulate DSP cores where the FPGA reuses computational resources repeatedly in many clock cycles to complete computation but with substantially reduced servo rates. However the attractiveness of a FPGA for control applications, from a computational standpoint, stems from its parallel processing capabilities. Examples of these applications are those that require high sampling rates, such as in scanning probe microscopy, and multiple parallel loops, such as large channel count micro actuator array control loops.

Since FPGAs have limited computational resources, realization of the controllers as filters must take these constraints into account. This is particularly a challenge to high performance controllers, which often involve high-order filters. Thus, special filter structures and realizations that use computation resources very efficiently can enable the use of FPGAs for high performance control at high sampling rates. To reach such high processing speeds and to maintain the parallel processing properties, computations are typically restricted to fixed-point computations.

With high-speed fixed-point controllers, the issues of quantization errors and the of number of multipliers become significant factors on the feasibility of implementing and even stability of certain controllers. Finite-word-length (FWL) effects contribute to quantization noise, and inaccurate pole/zero locations of your desired controllers/filters [FPW97]. In digital control, large sampling rates with respect to a system's bandwidth will cause its discrete-time transfer function to have poles/zeros that "migrate" towards the unit circle. This "migration" effect coupled with the FWL effects, only exacerbates the inaccuracy of the pole/zero locations and provides wildly different frequency responses.

In this dissertation, we consider accurate inversion of non-minimum phase dynamics with high order filters and their efficient realization. We use this for inversion based feedforward tracking (FF) and repetitive controllers (RC) and realize them on a FPGA. As a demonstrative example, we implemented these controllers to control a multi-input multi-output (MIMO) system at a high sampling rate.

The proposed stable inversion compensator is based on linear phase real-time infinite impulse response (IIR) filters first introduced by Powell & Chau [PC91]. Powell & Chau's realization involves L-length localized time reversals, overlap-add sectioned-convolutions, and another time reversal. The result is a linear phase IIR filter with phase equal to that of z^{-4L}. The "Reset", from [PC91], used to truncate the impulse response introduces parasitic sinusoidal phase disturbances resulting in only an *approximately* linear phase IIR filter. To solve the nonlinearity issue, Kurosu [KMT03] exploits the fact that any finite impulse response (FIR) filter can be represented as the difference of two IIR filters. He replaces the "Reset" with the difference of two IIR filters which results in a perfectly linear phase FIR filter, using IIR realizations, without phase distortions. For a long impulse response, a FIR filter realizing the same input-output relation would require more multipliers and adders, up to one or two orders of magnitude, than that of Kurosu's modified Powell-Chau filter. In other words, our proposed filter structure is most efficient when impulse responses are long and/or when sampling rates are high. In this dissertation, we show that with some modifications, Kurosu's filter can be used to efficiently invert non-minimum phase zeros (NMPZs). This filter is useful for feedforward tracking control where a finite length of the desired output can be previewed [Tsa94, AT95]. Furthermore, it is also useful for inversion based discrete-time repetitive control, where both linear phase and inversion filters have been employed in the repetitive control loop to achieve asymptotic tracking and regulation subject to periodic disturbances [TT94, TQN00]. In Iterative Learning Control (ILC), similar time reversal techniques for discrete and continuous time off-line inversion of nonlinear non-minimum phase plants have been presented [MHN01, GP01]. This off-line technique

for ILC which was previously not applicable to real-time RC algorithms. Using techniques from Kurosu's filter, this dissertation will illustrate a real-time reversal technique for RC.

Repetitive Control has been found to be useful in applications such as power electronics [ZW01] due to the periodic nature of the error signals. A version of RC uses the Zero Phase Error Tracking Controller (ZPETC) [Tom87] to approximate a plant inversion. ZPETC performs stable pole-zero cancelation and conjugates of the NMPZs. Depending on the location of the NMPZs, ZPETC which may result in a large dynamic range and become numerically unstable. In this dissertation we give a complete formulation of a numerically stable and efficient linear phase inversion filtering based on our preliminary work in [CT10a, CT10b]. We use the proposed inversion in feedforward and repetitive control for electromechanical systems at high sampling rate, in which the ZPETC based approach cannot be effective.

The remainder of this dissertation is structured as follows: Chapter 2 describes a basic understanding of the Powell-Chau linear phase IIR filter and modified structure known as the Kurosu filter. The chapter also shows that Kurosu's filter can be even more efficient, in terms of number of multipliers and equivalent delay, through multirate techniques. Chapter 3 illustrates the proposed inversion of NMPZs through a modification of Kurosu's filter. In addition, it describes how the proposed inversion can be extended into a new RC structure appropriate for high sampling rate fixed-point controllers. ILC, under the correct conditions, is introduced to serve as a double-precision equivalent of RC. Chapter 4 provides experimental results showing how the proposed inversion and new RC can be used in the controller design for a magnetically levitated shaft. Note that even though 100 kHz sampling rate in this dissertation is merely a limitation of the ADCs and DACs used. In actuality, the proposed compensators were designed to operate at or near the clock-rate of the FPGA. Chapter 5 contains experimental results of the proposed fixed-point controllers on a piezoelectric actuator. This experimental example highlights finite word length effects and demonstrates how the

Delta Operator [MG86] can be used to ameliorate these effects with minor increase in computational cost. Lastly, Chapter 6 contains experimental results of double-precision ILC on the same piezoelectric actuator. In addition to seeing the performance of ILC, it also serves as a equivalent of double-precision RC in which we can compare the performance of the fixed-point results of Chapter 5.

CHAPTER 2

Enhancement of Powell-Chau/Kurosu Filters

For many current applications, the computational speed and capability provided by modern electronics exceeds real-time application requirements. A popular platform, the field programmable gate array (FPGA), which has the ability to emulate digital circuits yet remain programmable, can be in this position. One FPGA drawback is its limited number of general purpose multipliers. Fortunately, multirate processing can serve as a vehicle to permit more efficient (higher data rate) use of these limited resources. We shall show how such multirate techniques can improve systems that employ approximately-linear-phase infinite impulse response (IIR) Powell-Chau filters and their variants. Using multirate processing, a few such modified Powell-Chau filter examples demonstrate how to reduce the demand for excessively numerous general purpose FPGA multipliers. In addition, it is shown that one of these structures can also reduce the long real-time delay typically associated with Powell-Chau filters and their variants.

Linear-phase filters have the valuable property that all frequencies pass through the filter with equal time delays. Traditionally, linear-phase filters are created using finite impulse response (FIR) filters whose filter-tap coefficent sequences exhibit special symmetry [Vai93]. Usually, to obtain high quality FIR filters, a large number of taps are required, which dictates many simultaneous multiplications. By contrast, IIR filters can provide transfer functions $H(z)$ with high quality magnitude characteristics while using significantly fewer tap multipliers. They do not, however, possess the linear-phase property. The Powell & Chau real-time linear-phase IIR filters [PC91] employ an

approach long known to be useful in "off-line" (non-real-time) digital filtering, wherein data sequences are processed by a nonlinear-phase filter, multiple times, in a way that results in overall linear-phase behavior. Often such off-line systems process a data sequence, in both a forward and in a reverse direction–hence the traditional non-real-time restriction.

The Powell & Chau linear-phase IIR realization employs length-L localized time reversals, overlap-add sectioned-convolutions, and another set of time reversals, where L is the length of the crucial last-in first-out (LIFO) elements. A typical choice of L is the approximate length of the first part of the filter's (infinite) impulse response $h(n)$ up to a point where the impulse response sample magnitudes reduce to signal quantization levels. The combination produces a linear-phase IIR filter with phase (delay) equal to that of z^{-4L}. The "Reset" used to truncate the IIR impulse response results in certain unwanted parasitic sinusoidal phase distortions; thus, yielding only an approximate linear-phase IIR filter. Others [WO94, DPL98, KMT03] have modified the transfer function and/or the structure to obtain improved frequency responses. Kurosu [KMT03] modifies and improves on Powell-Chau's original structure such that the phase imperfections of Powell-Chau's structure are eliminated. The modified filter is proven to have *no* phase disturbances. Kurosu exploits the fact that any FIR filter can be represented as a subtraction of two IIR filters. By combining that feature with the Powell-Chau filter, a perfectly linear-phase filter can be obtained. The resulting filter has an FIR input-output relation, but it requires three IIR filters, which could represent a reasonably high multiplier/adder hardware penalty when the impulse response is not very long. For feedback control applications, Chang and Tsao [CT10a, CT10b] have observed that Kurosu's modified Powell-Chau filter could be used to approximate the frequency response of unstable filters. They then use Kurosu's filter to approximately invert non-minimum phase zeros, thereby providing an improvement upon Kurosu's exact linear-phase structure. The present chapter uses P/I multirate theory from [JW97, YW99] to reduce the number of multipliers required to achieve the same input-output characteristics. Also,

using multirate theory, a structure is exhibited that can reduce the (long) $4L$ real-time delay typically associated with the Powell-Chau filter. Such reductions in both the delays and the number of multipliers are beneficial to real-time applications running on platforms having limited resources such as filters implemented as ASICs or those using FPGAs.

Even though both the Powell & Chau and the Kurosu filter use fewer multipliers than an equivalent FIR filter, there are still opportunities to further reduce the number of multipliers. From an implementation standpoint, the Kurosu filter has slightly more multipliers, which is due to the presence of the $H_L(z)$ filter. Furthermore, the largest hinderance for both structures is the $4L$ delay. This delay, can be undesirable in certain applications such as control system designs or time-sensitive filtering [CT10a, CT10b] while it may be acceptable in some other applications such as wireless communications and image processing.

Fig. 2.4 shows that the same filter $H(z)$ is employed three times, which may appear to be excessive. In principle, the multipliers can be reused through some data management techniques thereby saving on the number of multipliers. However, the implementation will vary from one case to another and the analysis of the intermediate signals can be awkward and complicated.

Concerning the delay problem, if the impulse response of $H(z)$ is long, the $4L$ delay can pose serious problems for time sensitive signal processing such as that in feedback controls [CT10a, CT10b]. We will now address both of these issues: systematically reducing multiplier hardware and reducing real-time delay.

2.1 Linear Phase Filtering Background

Definition 2.1.1. *Let FIR filters be denoted with a lower-case letter (e.g. $d(z)$) and IIR filters with a upper-case letter (e.g. $D(z)$).*

Definition 2.1.2. *Let $d^\star(z)$ and $D^\star(z)$ be defined as the time-reversal filter*

$$d^\star(z) = d(z)|_{z=z^{-1}} \tag{2.1}$$

$$D^\star(z) = D(z)|_{z=z^{-1}} . \tag{2.2}$$

Notice that this is also the definition for the complex conjugate of the frequency response when evaluated along the unit circle.

Linear phase filters are typically finite impulse response (FIR) filters with symmetric taps (coefficients) [Mit04]. Given some FIR filter $d(z)$, denote $e(z) = d(z) \times d^\star(z)$ where $d^\star(z)$ is the time reversal filter of $d(z)$. The resulting $e(z)$ will have linear phase with magnitude response of $|D(z)|^2$ [Tom87].

Definition 2.1.3. $|\cdot|$ *is the magnitude of the transfer function evaluated along the unit circle (i.e., the magnitude of filter frequency response).*

$e(z)$ is noncausal but can be made causal by cascading it with the delay element z^{-1}. Notice that $d^\star(z)$'s zeros are mirrored images of $d(z)$'s zeros. Thus, stable linear phase filters have mirrored pairs for both their poles and zeros. "Mirrored" in this context is with respect to the unit circle. ZPETC uses this method of conjugating the NMPZ along with inverting the stable portions of the plant. As for linear phase IIR filters, it would mean their poles and zeros would have to come in mirrored pairs. Linear phase IIR filters are typically not realizable since their conjugate poles are unstable (i.e. the stable pole's mirror image is an unstable pole outside the unit circle).

For finite-length sequences (and/or offline calculations) there are linear phase filtering techniques known as forward-backward filtering [KJ74, Gus96]. Simply put:

- A finite-length input is passed through an IIR or FIR filter.
- The output sequence is time-reversed and fed through the filter once again.
- Output of offline filter has linear phase.

The phase lag/lead introduced by the filter are canceled out when the time-reversed sequence is passed through the filter once again. Such applications of forward-backward filtering have been seen in Iterative Learning Control [MHN01], which use time-reversals to cancel out the phase but are not applicable in real-time RC.

2.2 Powell-Chau Filter

Powell & Chau [PC91] took the forward-backward filtering concept and produced an approximately linear phase IIR filter in real-time. To keep the filter stable and to implement a finite-length time reversal, the impulse response of the desired filter is truncated by a "Reset" seen in Fig. 2.1. Linear phase is achieved through the time reversals of the L-length localized time reversals through the use of Last-In-First-Out (LIFO) structures. L is the approximately the length of impulse response at which the impulse response enters the quantization region. Let us denote the filter within the dotted line in Fig. 2.1 as the time-reversal filter. The "time-reversal filter" performs a batch L-length time reversal. Like the forward-backward filter, the "*time-reversal filter*" will cancel out any phase lags/leads introduced by $H(z)$. To truncate the impulse response of an IIR filter, a "Reset" is used to clear the filter states of the previous L-length batch but also introduces phase non-linearities. This makes the entire filter to be only *approximately* linear phase. Fig. 2.2 shows a variant which is limit-cycle free [PC91]. The input-output relation of both Fig. 2.1 and Fig. 2.2 can be approximated by

$$\frac{Y(z)}{R(z)} \approx |H(z)|^2 \cdot z^{-4L}. \tag{2.3}$$

Notice that the $H_1(z)$ and $H_2(z)$ can be different to produce different frequency response characteristics [DPL98, WO94]. For this dissertation, however, we assume that $H_1(z)$ and $H_2(z)$ are equal (e.g. $H_1(z) = H_2(z) = H(z)$) and that they are causal filters.

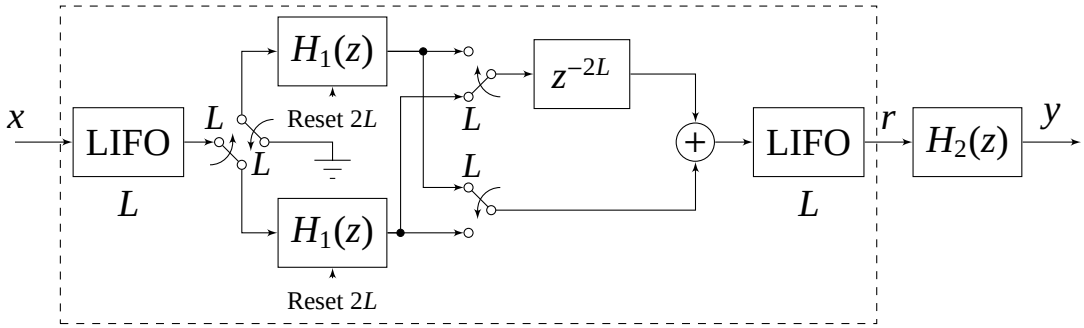

Figure 2.1: Powell-Chau approximately-linear-phase filter.

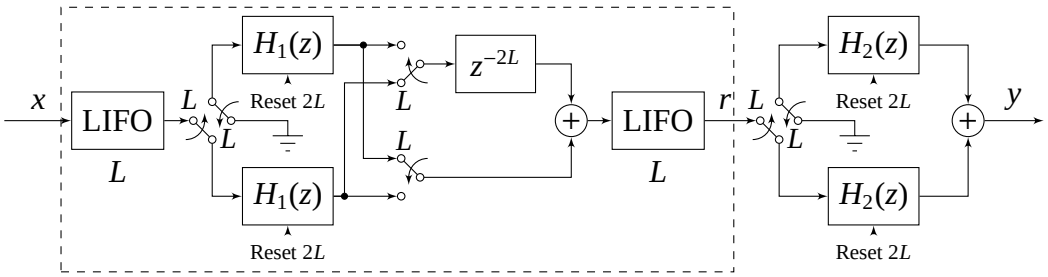

Figure 2.2: Powell-Chau limit-cycle free linear-phase filter

2.3 Kurosu Filter

Fig. 2.4 illustrates Kurosu's modified Powell-Chau filter. We also require that $H(z) = H_1(z) = H_2(z)$ and are causal. Kurosu [KMT03] replaces the "Reset" in the time-reversal filter with a difference between $H(z)$ and $H_L(z)$. Kurosu utilizes the fact that that any FIR filter can be represented as the subtraction of two IIR filters. $H(z)$ and $H_L(z)$ are both IIR filters of same order, where $H_L(z)$ or $[\cdot]_L$ is the truncated or unwanted portion of the impulse response. $H(z)$ can be described as

$$H(z) = \frac{b_0 + b_1 z^{-1} + \ldots + b_{K-1} z^{-(K-1)} + b_K z^{-K}}{1 + a_1 z^{-1} + \ldots + a_{K-1} z^{-(K-1)} + a_K z^{-K}}. \tag{2.4}$$

in its filter form. Also $q \geq r$, $b_i, a_i \in \mathbb{R}$. From [KMT03],

$$H_L(z) = \frac{-[c_0 + c_1 z^{-1} + \ldots + c_{K-1} z^{-(K-1)}]}{1 + a_1 z^{-1} + \ldots + a_{K-1} z^{-(K-1)} + a_K z^{-K}} \tag{2.5}$$

where

$$
\begin{aligned}
c_0 &= a_1 h(L-1) + a_2 h(L-2) + \ldots + a_K h(L-K)) \\
c_1 &= a_2 h(L-1) + a_3 h(L-2) + \ldots + a_K h(L-K+1) \\
c_2 &= a_3 h(L-1) + a_4 h(L-2) + \ldots + a_K h(L-K+2) \\
&\vdots \\
c_{K-2} &= a_{K-1} h(L-1) + a_K h(L-2) \\
c_{K-1} &= a_K h(L-1).
\end{aligned}
$$

An alternative way to view $H_L(z)$ is

$$H_L(z) = z^{-L} \cdot \left(H(z) \cdot z^{L} \right)_{+} \tag{2.6}$$

where $(\cdot)_+$ is only the *causal* portion of the impulse response.

Definition 2.3.1. *Let the* truncated (FIR) filter *be*

$$h_T(z) = H(z) - H_L(z) \tag{2.7}$$

$$= z^{-L} \cdot \left(H(z) \cdot z^{L} \right)_{-} \tag{2.8}$$

where $(\cdot)_-$ *is only the* noncausal *portion of the impulse response.*

Definition 2.3.2. *Let* $f^{-}(z)$ *be the* "time-reversal filter"

$$f^{-}(z) = h_T^{\star}(z) \cdot z^{-4L}. \tag{2.9}$$

From Fig. 2.4, it follows that

$$\frac{Y(z)}{R(z)} = h_T^{\star}(z) \cdot h_T(z) z^{-4L} \tag{2.10}$$

$$= |h_T(z)|^2 z^{-4L}. \tag{2.11}$$

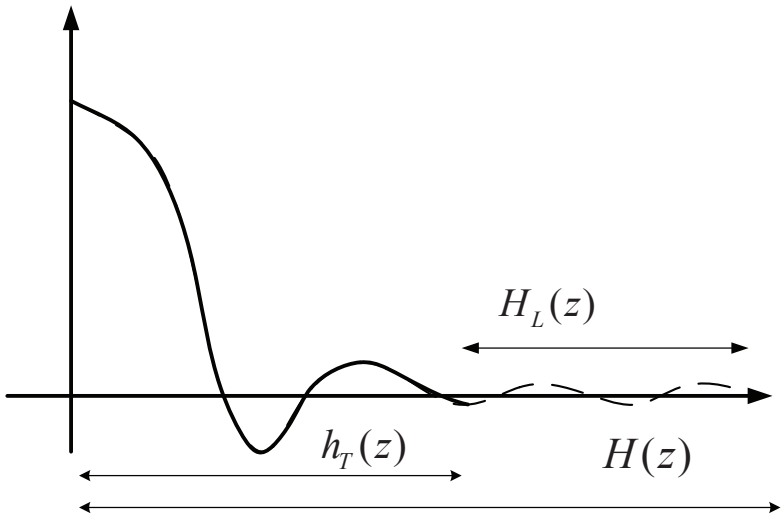

Figure 2.3: Truncation of infinite impulse response at length L.

Although this implementation is actually an FIR filter, it is termed as a *perfectly linear phase* IIR filter since it uses IIR realizations and also to remain consistent with [KMT03]. It is perfectly linear phase because it solves the phase non-linearity problem introduced by the "Reset" in Fig. 2.1. Fig. 2.3 shows that L is typically chosen such that the truncated portion of the impulse response sits near quantization level. Notice that as $L \to \infty$, then $h_T(z) \to H(z)$ and (2.11) $\to$ (2.3). The length of L will change the magnitude characteristics of the frequency response. $H_L(z) \cdot z^L$ is on the order of $H(z)$ (i.e. an increase in multipliers). To reduce the number of multipliers, $H_L(z)$ can be represented using a lower-order model approximation using some system identification methods such as Prony's method [PB87]. Throughout this dissertation $f^-(z)$ will also be referred to as the "time-reversal filter", and when appropriate it will be mentioned as the "inversion filter" in Section 3.1.

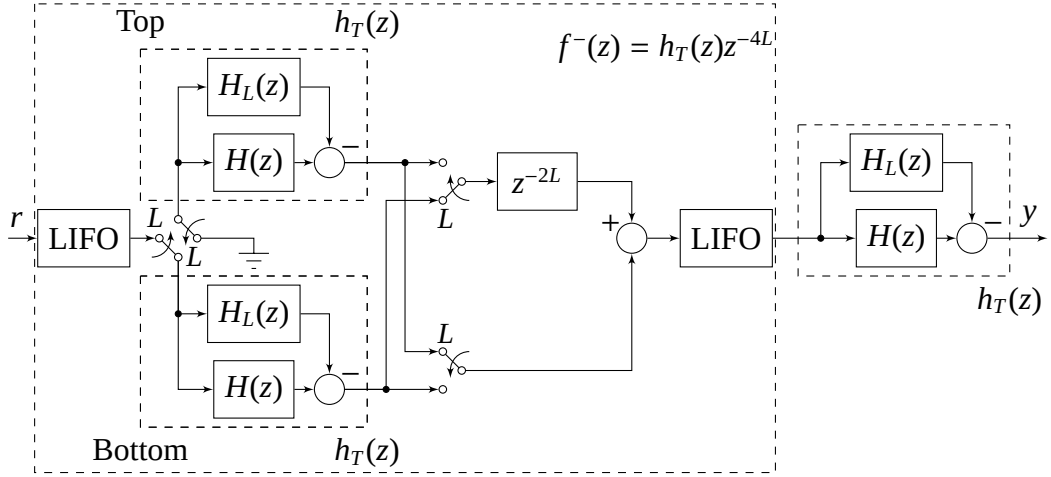

Figure 2.4: Kurosu's exact linear-phase filter.

2.4 Multiplier Reduction

To reduce the number of copies of $H(z)$ that are required by Kurosu's filter, a pipelining/interleaving (P/I) multirate technique [JW97, YW99], illustrated in Fig. 2.5, can be employed. In Fig. 2.5 the $x_1(n)$ and $x_2(n)$ signals are two independent inputs into $H(z)$ while $y_1(n)$ and $y_2(n)$ are the outputs of $H(z)z^{-1}$ with $x_1(n)$ and $x_2(n)$, respectively, as inputs.

The Fig. 2.5 technique requires operating $H(z)$ at twice the clock rate, using twice as many delays (registers or memory), but using just one set of filter-tap multipliers, not two. For VLSI and FPGA applications, the operation of these multipliers at a higher frequency is typically not a challenge. The maximum sampling frequency that the digital filter experiences is usually dictated by the analog-to-digital or digital-to-analog converters and not by the maximum clock rate for the digital logic.

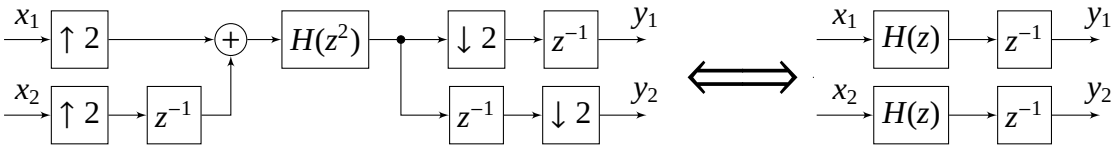

Figure 2.5: P/I technique to reuse multipliers.

2.4.1 Multirate Solution for Kurosu

Kurosu's Filter of Fig. 2.4 requires multiple copies of $H(z)$ and $H_L(z)$. Applying the P/I technique of Fig. 2.5 to the top and bottom $H(z)$ and $H_L(z)$ of Fig. 2.4 produces the dotted box in Fig. 2.6. The two instances of $H(z)$ and $H_L(z)$ of $f^-(z)$ result in an upsampling by two.

Notice however there are actually three instances of $H(z)$ and $H_L(z)$ in Fig. 2.4, all three of which can be handled by the P/I technique if we employ an upsampling by three. (See, e.g., Fig. 4 of [JW97].) Fig. 2.7 shows a way to do this.

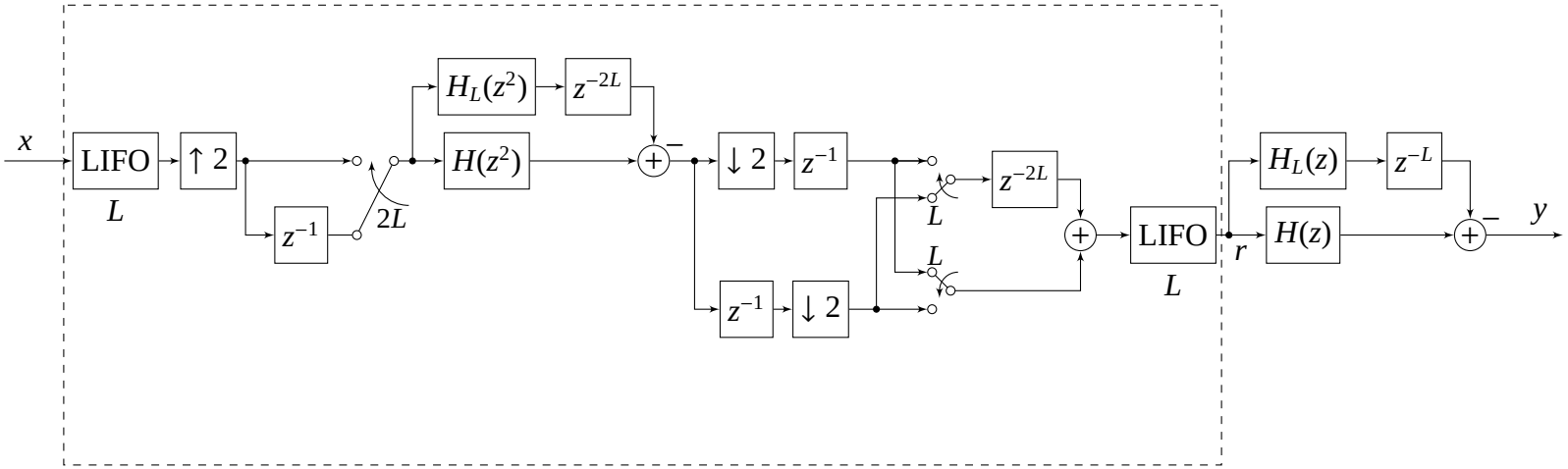

Figure 2.6: Proposed multirate modification of time reversed filter.

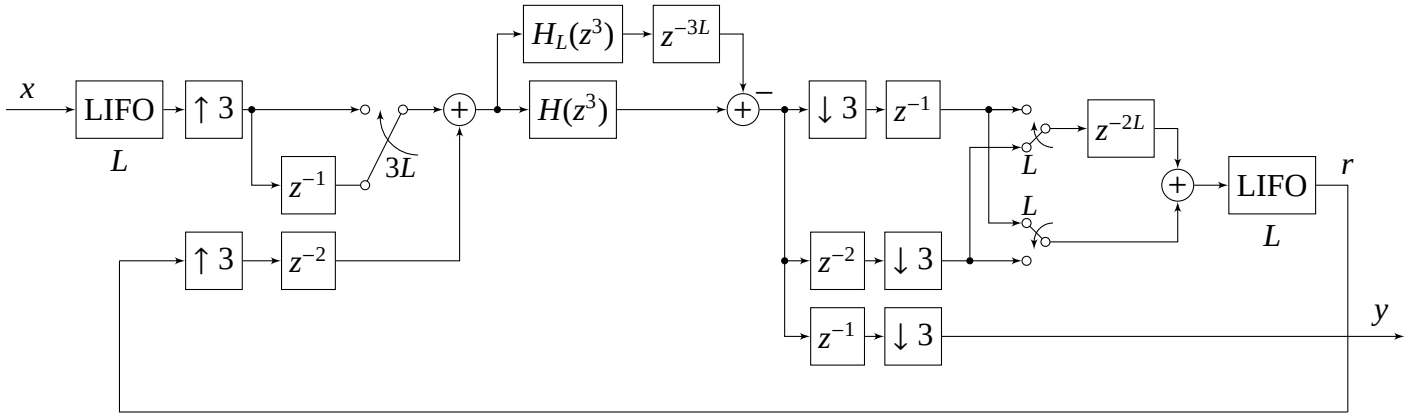

Figure 2.7: Proposed multirate modification.

Table 2.1: Number of Multipliers Comparison

Linear-Phase Filter Type	No. of Mult $H(z)$	No. of Mult $H_L(z)$
Powell-Chau (Fig. 2.1)	$3m$	0
Limit Cycle-Free Powell-Chau (Fig. 2.2)	$4m$	0
Kurosu (Fig. 2.4)	$3m$	$2m$
multirate Kurosu (Fig. 2.6)	$2m$	$2m$
multirate Kurosu (Fig. 2.6) - w/o H_L	$2m$	0
multirate Kurosu (Fig. 2.7) - w/o H_L	$1m$	0

If L is chosen sufficiently large, $H_L(z) \approx 0$ due to quantization noise. The number of general purpose multipliers can be reduced by excluding $H_L(z)$ provided that a longer delay is acceptable. Table 2.1 reflects both the inclusion and exclusion of $H_L(z)$. We introduce the parameter m in Table 2.1 to represent the number of coefficients in the filter. By using Fig. 2.7, a savings of up to 66% is possible.

The P/I multirate technique is useful for reducing multipliers for systems implemented on platforms in which multipliers are used in parallel or used independently, such as VLSI and FPGA implementations. This multirate structure serializes the general purpose multipliers of the filters and reuses them at a rate higher than the analog-to-digital and digital-to-analog sampling rate. Clearly, however, for applications on *serial* processors such as Digital Signal Processors (DSP) and single-core computers, the proposed technique brings no multiplier savings.

2.5 Real-time Delay Reduction

As mentioned earlier, one of the largest drawbacks of the Powell-Chau Filter or the Kurosu filter is the large delay associated with the filter. This problem has been addressed by Miyase [MTT00]. However, the Miyase approach adds additional filters (hence, more multipliers) to achieve the smaller delay. Typically, most filter designers are not willing to pay such a price; general-purpose multipliers are expensive and/or limited. By combining Miyase's solution with the P/I technique, we will be able to reduce the delay while using fewer multipliers.

2.5.1 Miyase's Filter Structure

It is shown in [MTT00] that by exploiting the overlap-add technique of the time-reversed filter, $f^-(z)$ can be expanded to more than just the two copies of $H(z)$ that Powell-Chau and Kurosu both used. Miyase observed that by using $N+1$ filters the delay can be reduced. The relation between equivalent delay and the number $N+1$ of filters is

$$\text{Equiv. Delay} = 2L/N + 2L. \tag{2.12}$$

Fig. 2.8 shows $f^-(z)$ with $N = 2$ (i.e., three $H(z)$ filters). In the Powell-Chau and Kurosu case where there are two $H(z)$ filters ($N = 1$), we find that (2.12) yields the expected $4L$ delay. For large enough N, the delay can be reduced to approximately $2L$. Let $M = L/N$, where L and N must be chosen such that M is an integer. L is still the approximate length of the impulse response and $N+1$ is the number of filters used in the overlap technique in the time-reversed filter $f^-(z)$. As shown in Fig. 2.9, the result of Fig. 2.8 can easily be extended to Kurosu's Filter.

Accepting an increase in multipliers to reduce the delay can be an unattractive compromise. As shown in Table 2.2, while the delay can be reduced to $2L$, the required

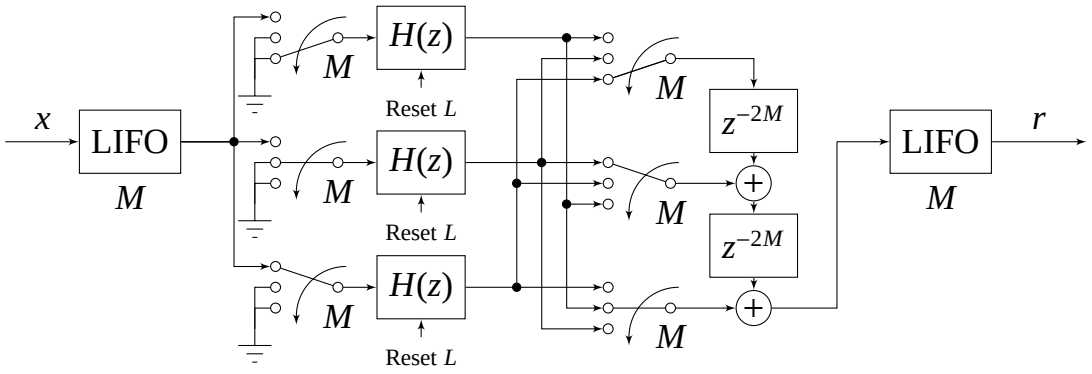

Figure 2.8: Miyase's time-reversed filter $f^{-}(z)$ with reduced delay ($N = 2$).

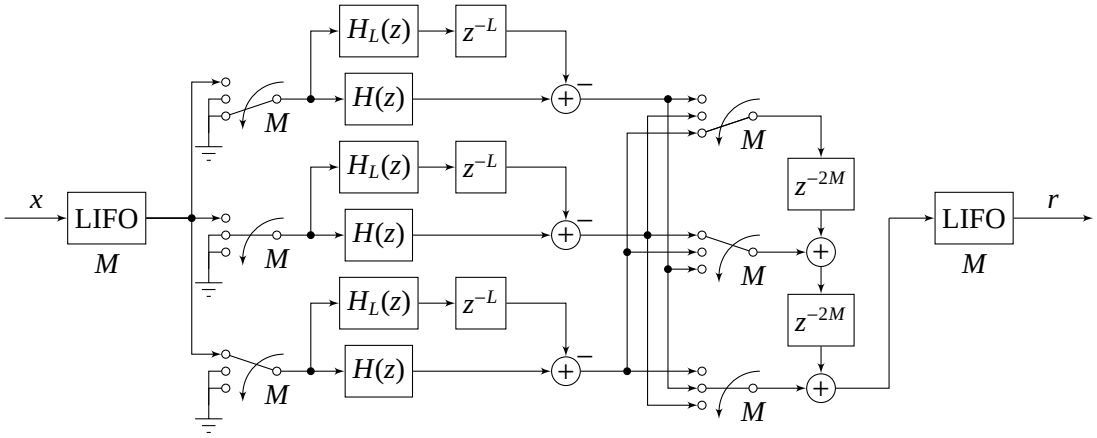

Figure 2.9: Miyase's time-reversed filter $f^{-}(z)$ with reduced delay on Kurosu's filter ($N = 2$).

number of multipliers now comprises $N + 1$ filters worth of computations. In Table 2.2, m denotes the number of multipliers used for the filter $H(z)$ or $H_L(z)$.

2.5.2 Multirate Structure for Delay Reduction of Time-Reversed Filter

Notice that Miyase's technique requires $N + 1$ copies of the same filter. The P/I multirate technique shown in Fig. 2.5 can reduce the redundant multipliers introduced by Miyase's delay reducing technique. This gives rise to a new technique, one that reduces both the delay *and* the number of multipliers of Kurosu's filter, assuming that the multipliers are able to perform at the higher sampling rate. The three $H(z)$ and three $H_L(z)$

Table 2.2: Delay and Multiplier Costs using Miyase's Delay Reduction

Case	Delay	No. of Multipliers
$N = 1$	$4L$	$2m$
$N = 2$	$3L$	$3m$
$N = 3$	$\frac{2}{3}L + 2L$	$4m$
$\vdots$	$\vdots$	$\vdots$
$N = n$	$\frac{2}{n}L + 2L$	$(n+1)m$
$\vdots$	$\vdots$	$\vdots$
$N = large$	$\approx 2L$	$(N+1)m$

of just the time-reversed filter in Fig. 2.9 can be converted into $H(z^3)$ and $H_L(z^3)$, respectively, using the P/I technique. The resulting filter is seen in Fig. 2.10 where $N = 2$, and where an upsampling of 3 is now employed.

If N is chosen large enough, the delay can be reduced to approximately $2L$ without increasing the number of multipliers. Of course, the choice of N is limited from above by the maximum sampling rate of the platform's (e.g., the FPGA's) general purpose multipliers. The advantage of adding the P/I technique to Miyase's method is made clear by comparing Table 2.3 with Table 2.2. The size of N is determined by the physical limitations on how fast the multipliers can operate. Unlike the multirate multiplier reduction structure, the multirate delay reduction structure can be beneficial in both serial and parallel processing platforms.

2.5.3 Combining Delay Reduction and Multiplier Reduction

The P/I Miyase technique of this section can easily be combined with the P/I multiplier savings technique of Section 2.4. Given that the multipliers used are fast enough, delays and multipliers can be reduced simultaneously. For example, combining the multiplier-reduced Fig. 2.7 system and the delay-reduced Fig. 2.10 system, we obtain the Fig. 2.11

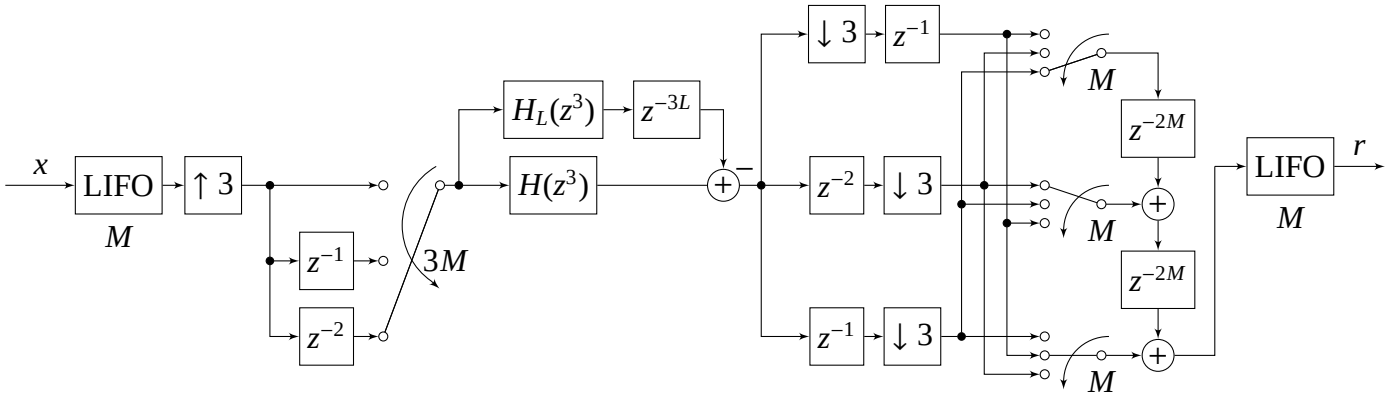

Figure 2.10: P/I version of Kurosu's filter with Miyase's delay reduction ($N = 2$).

Table 2.3: Delay and multiplier costs using P/I technique with Miyase's delay reduction

Case	Delay	No. of Multipliers
$N = 1$	$4L$	$1m$
$N = 2$	$3L$	$1m$
$N = 3$	$\frac{2}{3}L + 2L$	$1m$
$\vdots$	$\vdots$	$\vdots$
$N = n$	$\frac{2}{n}L + 2L$	$1m$
$\vdots$	$\vdots$	$\vdots$
$N = large$	$\approx 2L$	$1m$

system. Three $H(z)$ and three $H_L(z)$ filters are introduced by Miyase's technique in Fig. 2.9. A fourth $H(z)$ and $H_L(z)$ comes from the rightmost dotted box of Fig. 2.4. The four required $H(z)$ and $H_L(z)$ blocks are obtained by the use of the single $H(z^4)$ block in Fig. 2.11.

2.6 Example with Reduced Multipliers and Reduced Delays

Applying the proposed multirate multiplier and delay reduction technique to a lowpass filter example used by Powell & Chau and Kurosu will illustrate the benefits of the proposed structure. The following example was implemented on, a Xilinx Virtex-5 LX50 FPGA, accessed via the National Instruments PCIe-7852R board and LabVIEW graphical development tool. For this filter, the impulse response length is chosen to be $L = 500$, where the sampling frequency is chosen to be 100 kHz.

The lowpass filter is constructed using a combination of many all-pass functions. This structure is advantageous in that it facilitates yet another reduction in the number of multipliers through use of special all-pass function structures [SL98, VMN86]. The

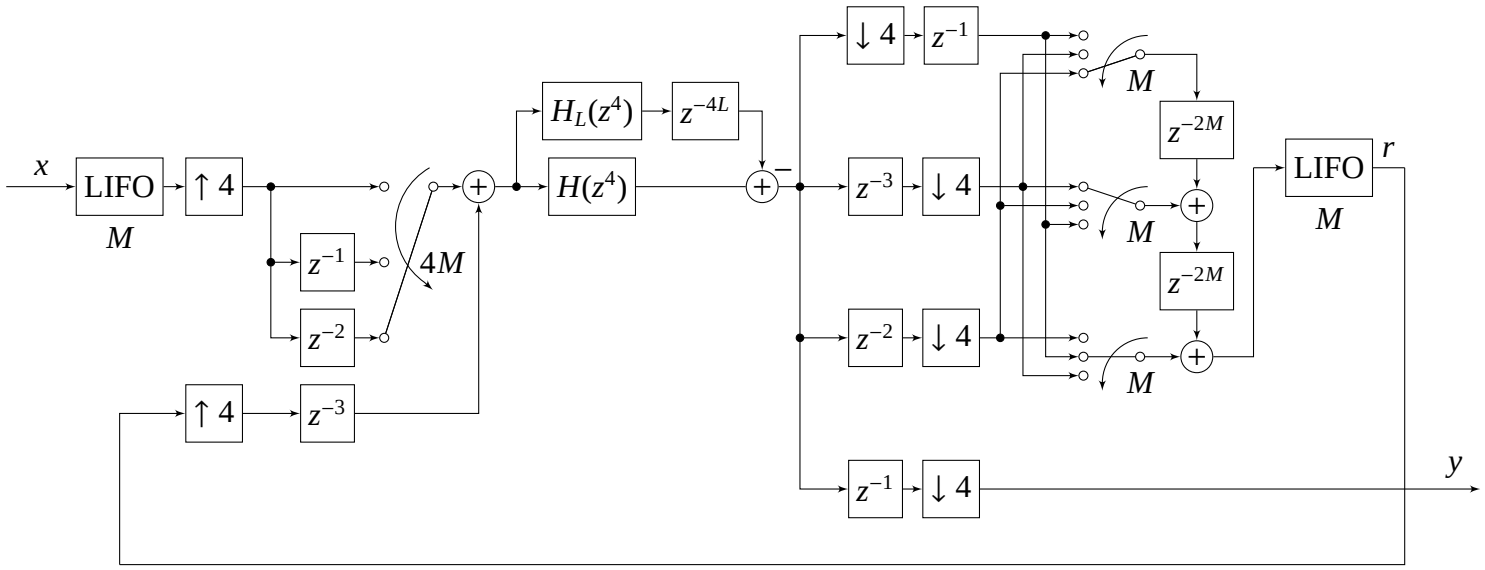

Figure 2.11: Multirate reduction of multipliers and delays ($N = 2$).

filter transfer function $F(z)$ is given [PC91, KMT03] as

$$F(z) = \frac{1}{2}\left[\frac{a_0 + z^{-1}}{1 + a_0 z^{-1}} \cdot \frac{b_0 + b_1 z^{-1} + z^{-2}}{1 + b_1 z^{-1} + b_0 z^{-2}} + \frac{c_0 + c_1 z^{-1} + z^{-2}}{1 + c_1 z^{-1} + c_0 z^{-2}} \cdot \frac{d_0 + d_1 z^{-1} + z^{-2}}{1 + d_1 z^{-1} + d_0 z^{-2}}\right]$$

where

$$\begin{aligned} a_0 &= 0.1404000 \\ b_0 &= 0.6832507 \\ b_1 &= 0.6008522 \\ c_0 &= 0.2868453 \\ c_1 &= 0.4101568 \\ d_0 &= 0.9175521 \\ d_1 &= 0.7085589. \end{aligned}$$

Table 2.4: Example illustrating comparison of Kurosu Filter with P/I multiplier and delay reduced Kurosu Filter.

Linear-Phase Filter Type	Delay	No. of Multipliers
Kurosu w/o $H_L(z)$(Fig. 2.4)	4000	21
P/I Mult. Reduction w/o $H_L(z)$(Fig. 2.7)	4000	7
P/I Delay Reduction w/o $H_L(z)$(Fig. 2.10)	3000	21
P/I Delay and Mult. Reduction w/o $H_L(z)$ (Fig. 2.11)	3000	7

Clearly, this all-pass implementation of $F(z)$ requires just seven multipliers. To illustrate the potential multiplier and delay savings Table 2.4 shows how the original Kurosu filter (Fig. 2.4) compares with the P/I multiplier reduced structure (Fig. 2.7), the P/I delay reduced structure (Fig. 2.10), and the P/I combined (multiplier and delay)

reduction structure (Fig. 2.11). The numbers in Table 2.4 were also confirmed in the HDL Synthesis Compile report provided by the Xilinx Compiler during FPGA compilation. Notice that P/I techniques for reducing the multipliers and delays do not affect the magnitude response. Fig. 2.12 shows, using $H(z) = F(z)$, how, when using different upsampled Kurosu filters, $f^{-}(z)$ has the same magnitude response as $H_T(z)$. The nonlinear portions of the phase of $f^{-}(z)$ cancel the nonlinear-phase portions of $H_T(z)$ which makes $f^{-}(z) \times H_T(z)$ exactly linear-phase. As mentioned previously, if L is sufficiently long, then $H_L(z)$ can be left out making $f^{-}(z) \times H(z)$ approximately-linear-phase. Fig. 2.13 illustrates how even upsampling at five times the sampling period ($N = 4$), can produce significant delay reduction.

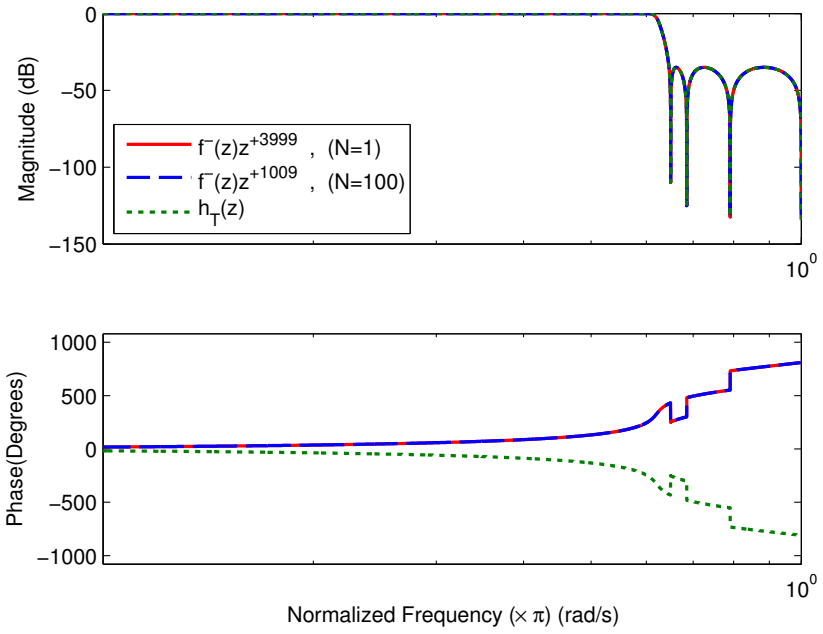

Figure 2.12: Frequency Response Comparison of Different Upsampling/Downsampling Rates

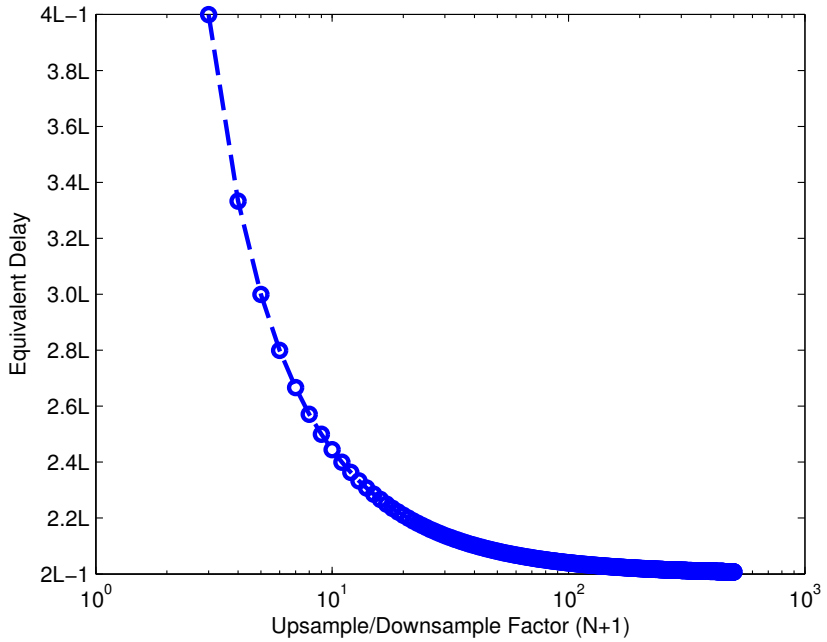

Figure 2.13: Equivalent Filter Delay vs. Upsample/Downsample Factor (N+1)

2.7 Conclusion

Both the Powell-Chau filter and the Kurosu filter bring about the advantages of linear-phase IIR filtering. Although they are already computationally efficient, we have shown how a reduction in multipliers can be achieved since many copies of the same filter are employed in these structures. Using the proposed P/I multirate techniques, Kurosu's modified Powell-Chau filter can potentially reduce the number of required multipliers to one-third the original amount. A single FPGA can benefit from a significant savings of valuable computation resources which could then be used for additional filters or other processes. In addition to multiplier savings, additional P/I techniques applied to Miyase's delay reduction structure helps to reduce the long ($4L$) delay that plagues the Powell-Chau filter. Assuming sufficiently fast multipliers, the delay can be further reduced down to virtually $2L$ by use of our new computationally efficient linear-phase IIR filter with a delay less than that of Kurosu's filter.

CHAPTER 3

Efficient Feedforward, Repetitive Control, and Iterative Learning Control of Nonminimum Phase Systems

3.1 Inversion of Non-Minimum Phase Zeros

For minimum phase systems, direct pole-zero cancelation is easiest. However for non-minimum phase systems, the direct cancelation of NMPZs would result in an unstable controller. The approximate inverse of NMPZs can be performed through a high-order FIR filter through deconvolution or equalization. Long FIR filters are costly in terms of number of multipliers and additions when compared to IIR filters [Mit04]. Reusing FIR filters would significantly reduce the maximum servo rate. IIR filters can reduce the number of multipliers/additions enabling faster sampling/servo frequencies but stability becomes an issue. Our proposed inversion has the stability of an FIR filter with the computational complexity of an IIR filter.

Given some stable (or closed-loop stabilized) linear time-invariant (LTI) system,

$$G(z) = z^{-d} \cdot \frac{b^{+}(z)b^{-}(z)}{a(z)} \tag{3.1}$$

where $a(z)$, $b^{+}(z)$, $b^{-}(z)$, and d are the stable poles, stable zeroes, unstable zeros, and relative order, respectively. Let the proposed inversion filter be $F(z) = f^{-}(z) \cdot F^{+}(z)$ as illustrated in Fig. 3.1. Table 3.1 summarizes the different designs of the filter $H(z)$ and $F^{+}(z)$ for their corresponding overall transfer function $F(z)G(z)$. To better understand Table 3.1, some definitions are introduced.

Definition 3.1.1. *Define deg(·) as the degree or order of the polynomial (i.e. number of roots).*

Definition 3.1.2. *Define the constants* ρ

$$\rho = deg(b^{-}(z)). \tag{3.2}$$

3.1.1 Novel Inversion Filter Structure

For this case, let $F^{+}(z)$ is the inversion of stable poles/zeros where

$$F^{+}(z) = \frac{a(z)}{b^{+}(z)}M \tag{3.3}$$

By letting $H(z)$ inside $f^{-}(z)$ be

$$H(z) = \frac{1}{b^{-\star}(z)}M \tag{3.4}$$

approximate inversions of non-minimum phase zeros is possible. Since most physical systems are bandlimited, $M(z)$ can be chosen to be the reference model, which limit the bandwidth of the inversion to avoid large gains at the high frequency regions. Similarly, $M^{\star}(z)$ needed such that the reference model is linear phase. For simplicity, we will assume $M(z) = M^{\star}(z) = 1$. Ideally, a stable inversion is desired such that $|f^{-}(z)| \approx \left|\frac{1}{b^{-}(z)}\right|$. Using Fig. 2.4, Eqn. (2.8) and the above choice of $H(z)$ results in

$$h_T(z) = \frac{1}{b^{-\star}(z)} - \left[\frac{1}{b^{-\star}(z)}\right]_L. \tag{3.5}$$

Notice if L long enough, then

$$f^{-}(z) = h_T^{\star}(z)z^{-4L} \approx H^{\star}(z)z^{-4L} \tag{3.6}$$

If $H(z) = \frac{1}{b^{-\star}(z)}$

$$H^{\star}(z)z^{-4L} = \left(\frac{1}{b^{-\star}(z)}\right)^{\star} z^{-4L} \tag{3.7}$$

$$= \frac{1}{b^{-}(z)}z^{-4L}. \tag{3.8}$$

Table 3.1: Choices of $H(z)$ inside $f^-(z)$ filter and $F^+(z)$.

Filter	$H(z) =$	$F^+(z) =$	$F(z)G(z)$
Approx. Inv.	$\frac{1}{b^{-\star}(z)}$	$\frac{a(z)}{b^+(z)}$	$\approx z^{-(4L+d)}$
IIR-ZPETC	$\frac{b^+(z)b^-(z)}{a(z)}$	1	$\approx \lvert G(z)\rvert^2 z^{-(4L+d)}$
ZPETC	$\frac{b^-(z)}{b^-(1)^2}$	$\frac{a(z)}{b^+(z)}$	$= \frac{\lvert b^-(z)\rvert^2}{b^-(1)^2} z^{-(4L+d)}$

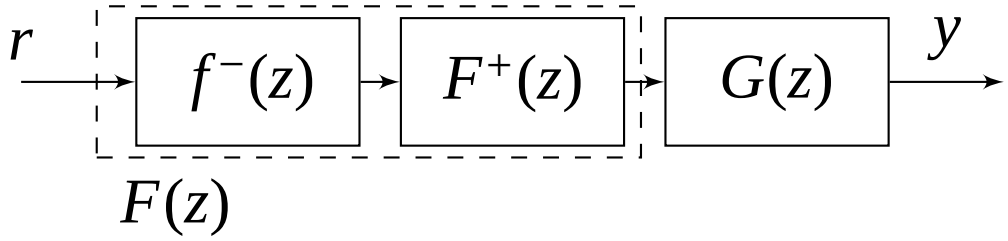

Figure 3.1: Realization of approximate NMPZ inversion.

Thus

$$f^-(z) = \left(\frac{1}{b^-(z)}\right)_T z^{-4L} \approx \frac{1}{b^-(z)} z^{-4L}. \tag{3.9}$$

Recall $f^-(z)$ uses the Kurosu filter (Fig. 2.4) to realize an approximate inverse since in a standard filter form it is unstable. Letting $H(z)$ be the inverse of the mirrored NMPZs will produce an approximate inversion. This means that as $L \to \infty$, $f^-(z) \cdot F^+(z) \to H^{-1}(z) \cdot z^{-4L}$. In summary, Kurosu's filter allows us to implement a L^{th}-order FIR inversion filter for the NMPZs using only a few IIR filters. This results in $F(z)G(z) \approx z^{-(4L+d)}$.

3.1.2 IIR-ZPETC

If $H(z) = \frac{b^+(z)b^-(z)}{a(z)}$ and $F^+(z) = 1$, this would result in $f^-(z) \approx \frac{b^{+\star}(z)b^{-\star}(z)}{a^\star(z)} \cdot z^{-4L}$. We will denote this choice of $H(z)$ and $F^+(z)$ as the IIR-ZPETC where $F(z)G(z)$ is approximately linear phase (i.e. $F(z)G(z) = |G(e^{i\omega})|^2 z^{-(4L+d)}$).

3.1.3 ZPETC

If $H(z) = \frac{b^-(z)}{b^-(1)^2}$ and $F^+(z) = \frac{a(z)}{b^+(z)}$, the resulting compensator is a delayed ZPETC [Tom87]. Since $b^-(z)$ is a FIR filter, $L = \rho$ then $f^-(z) = b^{-\star}(z) \cdot z^{-4L}$. Using Kurosu's filter to implement ZPETC results in $F(z)G(z) = \frac{|b^-(z)|^2}{b^-(1)^2} z^{-(4L+d)}$. From a practical standpoint, implementing the standard ZPETC would probably use less resources on an FPGA and $F(z)G(z) = \frac{|b^-(z)|^2}{b^-(1)^2} z^{-(\rho+d)}$. In principle, one could realize ZPETC with a Powell-Chau/Kurosu filter. This implies that wherever ZPETC can be applied, that IIR-ZPETC and the proposed inversion filter may be possible alternatives.

Comparing the proposed inversion filter against ZPETC and IIR-ZPETC, the proposed filter has more control over the design of the reference model using $M(z)$ and $M^\star(z)$. We expect in the inversion filter to perform better since $f^-(z)$ is the equivalent of a L^{th} order FIR filter whereas the $f^-(z)$ in ZPETC is only of order $b^-(z)$.

3.2 Repetitive Control and FPGA Implementation

3.2.1 Novel Repetitive Control Structure

Fig. 3.2 illustrates a simple RC structure where $F(z)$ is a type of FF inversion of the stabilized plant $G(z)$. $F(z)$ can be the proposed inversion, IIR-ZPETC, Kurosu implementation ZPETC, or the classic ZPETC [Tom87, TTC89]. $F(z)$ inverts the stable poles/zeros cascaded with a linear phase complement of the NMPZs. N_1 is adjusted such that the non-causality, introduced by $b^{-\star}(z)$, is absorbed. Letting $F(z)$ be the proposed inversion, then $FG \approx z^{-N_2}$. To ensure stability, RC must satisfy

$$|(z^{-N_2} - FG)q| < 1 \tag{3.10}$$

Notice that $|\cdot|$ is an abuse of notation and (3.10) is meant that the magnitudes are less than one across all frequencies. Assuming $F(z)$ is a nearly perfect inversion, the

nominal sensitivity function for RC for the lead-stabilized system is

$$S_{RC} = \frac{1 - qz^{-(N_1+N2)}}{1 - qz^{-(N_1+N_2)} + FGqz^{-N_1}} \tag{3.11}$$

Assuming $FG \approx z^{-N_2}$ and (4.15), then

$$S_{RC} \approx 1 - qz^{-(N_1+N_2)} = 1 - (qz^{N_q})z^{-N} \tag{3.12}$$

where qz^{N_q} is approximately zero phase. To ensure performance of RC using any of the FF inversion techniques from Section 3.1, N_2 and N_1 must satisfy

$$N_2 = 4L + d \tag{3.13}$$

$$N_1 = N - N_2 - N_q \tag{3.14}$$

$$N = \frac{f_s}{f}. \tag{3.15}$$

where f_s is the sampling frequency and f is the reference or disturbance frequency. $q(z)$ is typically a linear phase low-pass filter (LPF) to maintain robust stability against model uncertainties and to maintain the zero-phase property [TT94]. Let the actual plant be denoted as $G_a(z)$. Then the stability condition in (3.10) suggests that

$$\left|z^{-N_2} - FG_a\right| = \left|z^{-N_2} - FG + FG - FG_a\right|. \tag{3.16}$$

Since $\left|z^{-N_2} - FG\right| \approx 0$, (i.e. $|FG| \approx 1$), then

$$\left|z^{-N_2} - FG\right| \approx \left|FG\left(\frac{G - G_a}{G}\right)\right| = \left|\frac{G - G_a}{G}\right|. \tag{3.17}$$

Using (3.10) and (3.17), a sufficient condition for stability with respect to $q(z)$ is

$$|q| < \left|\frac{G}{G - G_a}\right| \tag{3.18}$$

The choice of $q(z)$ then becomes a filter design problem with the constraint of the filter being a linear phase FIR filter. N_q is the equivalent linear phase time delay produced by $q(z)$ (i.e. $\angle q(z) = \angle z^{-N_q}$). To obtain a linear phase LPF, a high order may be necessary.

For platforms such as a FPGA, this requires either more general purpose-multipliers or reduction of the maximum sampling rate if these multipliers are reused. Fortunately, the Kurosu filter implementation can resolve these issues. An example of this will be shown in the next section.

Remark 3.2.1. *Notice if* $F(z) = \frac{b^{-\star}(z)a(z)}{b^{-}(1)^2 b^{+}(z)}$, *a ZPETC from [TTC89], then* $N_2 = \rho + d$.

Remark 3.2.2. *FPGA based digital repetitive control has recently been employed at* $100KHz$ *sampling rate for high-speed scanning of atomic force microscopes, where only plant delay was compensated for realization, i.e.* $F(z) = 1$, $q(z) = a/z + b$ *[KL12]. We are arguably the first to perform accurate inversion compensation by the very efficient filter realization, which substantially increases the stability margin and performance in repetitive control.*

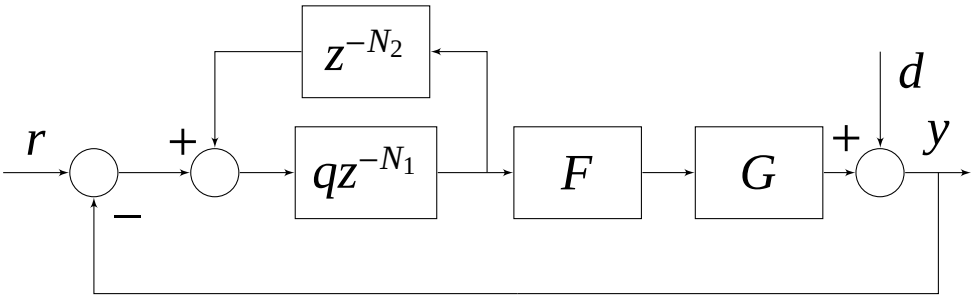

Figure 3.2: RC with feed-forward inversion and lead controller.

3.2.2 FPGA Implementation

In order to make full use of the FPGA's parallel architecture we will be using common filter structures such as the Direct Form II Transposed (Fig. 3.3), [Mit04]. For fixed-point FPGA implementations, it is advantageous to break up a filter into cascaded second-order sections (SOS) $X(z)$ such that

$$F(z) = \prod_{i=0}^{k-1} X_i(z) \tag{3.19}$$

where

$$X_i(z) = \frac{b_{0,i} + b_{1,i}z^{-1} + b_{2,i}z^{-2}}{1 + a_{1,i}z^{-1} + a_{2,i}z^{-2}} \tag{3.20}$$

and k is the number of second-order sections. Reduction to SOS is useful for minimizing quantization noise [Mit04].

Remark 3.2.3. *These second-order sectioned filters are also known as* biquadratic *filters.*

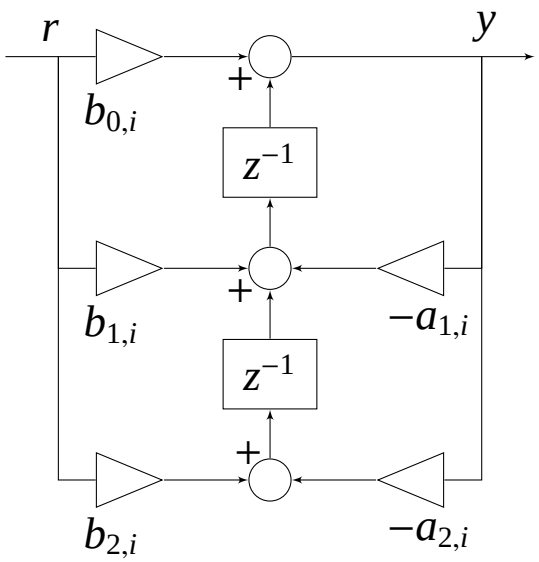

Figure 3.3: Direct form II transposed SOS filter structure.

3.3 Iterative Learning Control

Iterative Learning Control (ILC) has been extremely popular among roboticists as a form of control due to their nature of having repetitive movements. As such, ILC is a control algorithm that learns from iteration to iteration of each completed path or movement. In this context, an iteration is defined as the system be actuated given some precalculated control signal. Since, the learning happens from iteration to iteration, the feedforward control signal can be calculated offline using double precision computation. ILC is useful in that it is robust against model uncertainties and repeating distur-

bances. The goal of ILC is not to learn a inversion model but rather to find the correct control input to obtain the desired output. [BTA06, ACM07] provide an extensive survey of the Iterative Learning Control literature. RC and ILC have some similarities. RC can be considered the online version of ILC, as such, many robust stability criterion derived in RC can be used in ILC as well. [WGI09] provides a broad framework that links the relationship between ILC, RC, and Run-to-Run (R2R) control. Longman [Lon00] showed that if good transient conditions are satisfied, stability conditions between linear ILC and linear RC are equivalent.

3.3.1 Iterative Learning Control Formulation

The system description and ILC formulation is based off of [BTA06]. Consider a stable, discrete-time, single-input single-output, system

$$G(q) = g(0)q^{-1} + g(1)q^{-2} + \cdots, \tag{3.21}$$

where the sequence $g(0), g(1), \ldots,$ is the impulse response and q^{-1} is the time-domain shift operator in this context. Furthermore, its input-output time domain relationship can be described as

$$\mathbf{y}_j(k) = \mathbf{G}(q)\mathbf{u}_j(k) + \mathbf{d}(k). \tag{3.22}$$

Explicitly written, it is

$$\underbrace{\begin{bmatrix} y_j(0) \\ y_j(1) \\ \vdots \\ y_j(N-1) \end{bmatrix}}_{\mathbf{y_j}} = \underbrace{\begin{bmatrix} g(0) & 0 & \cdots & 0 \\ g(1) & g(0) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ g(N-1) & g(N-2) & \cdots & g(0) \end{bmatrix}}_{\mathbf{G}} \underbrace{\begin{bmatrix} u_j(0) \\ u_j(1) \\ \vdots \\ u_j(N-1) \end{bmatrix}}_{\mathbf{u_j}} + \underbrace{\begin{bmatrix} d_j(0) \\ d_j(1) \\ \vdots \\ d_j(N-1) \end{bmatrix}}_{\mathbf{d_j}} \tag{3.23}$$

where k is the time index, j is the iteration index, d is a repeating external disturbance. $\mathbf{y}_j$ is the output of the system $\mathbf{G}$ given some control input $\mathbf{u}_j$ at iteration j. Additionally, $\mathbf{y}_j, \mathbf{u}_j, d$ are $N \times 1$ vectors and $\mathbf{G}$ is a $N \times N$ matrix. Assuming that $G(q)$ is a proper rational function with relative order of 0. Notice that if the system is strictly proper

then it is possible to time-advance the system by its relative order to match this formulation. It follows that **G** can be seen as a convolution matrix, or alternatively known as the "lifted-system" framework in the ILC literature. For ILC, a common control law [BTA06], is

$$\mathbf{u}_{j+1} = \mathbf{Q}(\mathbf{u}_j + \mathbf{L}\mathbf{e}_j). \tag{3.24}$$

In its explicit form, it is

$$\begin{aligned}\underbrace{\begin{bmatrix} u_{j+1}(0) \\ u_{j+1}(1) \\ \vdots \\ u_{j+1}(N-1) \end{bmatrix}}_{\mathbf{u_{j+1}}} = &\underbrace{\begin{bmatrix} q(0) & q(-1) & \cdots & q(-(N-1)) \\ q(1) & q(0) & \cdots & q(-(N-2)) \\ \vdots & \vdots & \ddots & \vdots \\ q(N-1) & q(N-2) & \cdots & q(0) \end{bmatrix}}_{\mathbf{Q}} \Bigg(\underbrace{\begin{bmatrix} u_j(0) \\ u_j(1) \\ \vdots \\ u_j(N-1) \end{bmatrix}}_{\mathbf{u}_j} \\ &+ \underbrace{\begin{bmatrix} l(0) & l(-1) & \cdots & l(-(N-1)) \\ l(1) & l(0) & \cdots & l(-(N-2)) \\ \vdots & \vdots & \ddots & \vdots \\ l(N-1) & l(N-2) & \cdots & l(0) \end{bmatrix}}_{\mathbf{L}} \underbrace{\begin{bmatrix} e_j(0) \\ e_j(1) \\ \vdots \\ e_j(N-1) \end{bmatrix}}_{\mathbf{e}_j} \Bigg).\end{aligned} \tag{3.25}$$

where, **u**, **L**, **Q**, and j are the control signal, inversion filter (in matrix form), zero-phase low-pass-filter (in matrix form), and iteration number, respectively. Also, the learning is based off learning from past errors

$$\begin{aligned}\mathbf{e}_j &= \mathbf{r} - \mathbf{y}_j \\ \underbrace{\begin{bmatrix} e_j(0) \\ e_j(1) \\ \vdots \\ e_j(N-1) \end{bmatrix}}_{\mathbf{e}_j} &= \underbrace{\begin{bmatrix} r(0) \\ r(1) \\ \vdots \\ r(N-1) \end{bmatrix}}_{\mathbf{r}} - \underbrace{\begin{bmatrix} y_j(0) \\ y_j(1) \\ \vdots \\ y_j(N-1) \end{bmatrix}}_{\mathbf{y}_j}\end{aligned} \tag{3.26}$$

where **r**, a $n \times 1$ vector, is the desired output.

Remark 3.3.1. *Notice that since the time signals are finite in time and will be represented through bold fonts to denote a vector (e.g.,* $\mathbf{r} = [r(0) \dots r(n-1)]^T$*). Capital letters denote matrices, and lower case denote vectors.*

Remark 3.3.2. *Notice that* **G** *is lower triangular, indicating causality. However,* **Q** *and* **L** *are not lower triangular, meaning that noncausal solutions are allowed since ILC is an offline controller.*

3.3.2 ILC for Nonminimum Phase System

The problem of inverting nonminimum phase systems in the ILC literature have been tackled through numerous techniques [AO94, CFR07, FLR05, MHN01, GP99, GP01, Sog02]. In many of these techniques they involve noncausal solutions and allowing the use of preactuation. One method by [MHN01] is closely related to the proposed real-time inversion from Section 3.1. Recall, that Kurosu's filter was used to provide the forward-backward filtering [Gus96] for a real-time environment. Assuming that **L**, also known as the "*learning function*" is an approximate inversion of **G**, similar to $f^-(z)$ from Section 3.1.1, then it can be described as

$$\mathbf{L} = \mathbf{I}_R \mathbf{F}^- \mathbf{I}_R \mathbf{F}^+ \tag{3.27}$$

In its expanded form,

$$
\begin{aligned}
&\underbrace{\begin{bmatrix} l(0) & l(-1) & \cdots & l(-(N-1)) \\ l(1) & l(0) & \cdots & l(-(N-2)) \\ \vdots & \vdots & \ddots & \vdots \\ l(N-1) & l(N-2) & \cdots & l(0) \end{bmatrix}}_{\mathbf{L}} \\
&= \underbrace{\begin{bmatrix} 0 & \cdots & 0 & 1 \\ \vdots & \iddots & 1 & 0 \\ 0 & \iddots & \iddots & \vdots \\ 1 & 0 & \cdots & 0 \end{bmatrix}}_{\mathbf{I}_R} \underbrace{\begin{bmatrix} f^-(0) & 0 & \cdots & 0 \\ f^-(1) & f^-(0) & \ddots & \vdots \\ \vdots & \vdots & \ddots & 0 \\ f^-(N-1) & f^-(N-2) & \cdots & f^-(0) \end{bmatrix}}_{\mathbf{F}^-} \\
&\times \underbrace{\begin{bmatrix} 0 & \cdots & \cdots & 0 & 1 \\ \vdots & & \iddots & 1 & 0 \\ \vdots & \iddots & \iddots & \iddots & \vdots \\ 0 & \iddots & \iddots & & \vdots \\ 1 & 0 & \cdots & & 0 \end{bmatrix}}_{\mathbf{I}_R} \underbrace{\begin{bmatrix} f^+(0) & f^+(-1) & \cdots & \cdots & 0 \\ f^+(1) & f^+(0) & \ddots & & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \vdots & \vdots & & \ddots & f^+(-1) \\ f^+(N-1) & f^+(N-2) & & \cdots & f^+(0) \end{bmatrix}}_{\mathbf{F}^+}
\end{aligned}
\tag{3.28}
$$

where $\mathbf{I}_R$, a $N \times N$ matrix, be the time-reversal operation in matrix form. $\mathbf{F}^-$, a $N \times N$ matrix, represents the conjugate inverse of the nonminimum phase zero. $f^-(i)$ is the impulse response of $\frac{1}{b^{-\star}(z)}$. Caution not to confuse $f^-(i)$ by the filter $f^-(z)$. $\mathbf{F}^+$, a $N \times N$ Toeplitz matrix, is the inversion of the stable poles and zeros where $f^+(i)$ is the impulse response of $\frac{a(z)}{b^+(z)} z^{\rho+d}$. Notice that $\frac{a(z)}{b^+(z)} z^{\rho+d}$ could possibly be noncausal and $\mathbf{F}^+$ will have $\rho + d$ super diagonals.

3.3.3 A Robust Stability Condition for ILC

[Lon00] has established an equivalence between ILC and RC, then the following robustness from RC can also be used for ILC. The **Q** is a zero-phase LPF meant to ensure robust stability from iteration to iteration. In the ILC framework, stability means that

the error converges from iteration to iteration. **Q**, is based off of the same $Q(z)$ from Section 3.2. In both ILC and RC, $Q(z)$ and **Q** are commonly referred to as the "Q-Filter" and will be used interchangeably with **Q**. **Q**, a $n \times n$ Toeplitz matrix, represents a zero-phase LPF described as

$$\underbrace{\begin{bmatrix} q(0) & q(-1) & \cdots & q(-(N-1)) \\ q(1) & q(0) & \cdots & q(-(N-2)) \\ \vdots & \vdots & \ddots & \vdots \\ q(N-1) & q(N-2) & \cdots & q(0) \end{bmatrix}}_{\mathbf{Q}}$$

$$= \underbrace{\begin{bmatrix} 0 & \cdots & 0 & 1 \\ \vdots & \iddots & 1 & 0 \\ 0 & \iddots & \iddots & \vdots \\ 1 & 0 & \cdots & 0 \end{bmatrix}}_{\mathbf{I}_R} \cdot \begin{bmatrix} q_0 & 0 & \cdots & 0 \\ q_1 & q_0 & \ddots & \vdots \\ \vdots & \vdots & \ddots & 0 \\ q_{N-1} & q_{N-2} & \cdots & q_0 \end{bmatrix} \cdot \underbrace{\begin{bmatrix} 0 & \cdots & 0 & 1 \\ \vdots & \iddots & 1 & 0 \\ 0 & \iddots & \iddots & \vdots \\ 1 & 0 & \cdots & 0 \end{bmatrix}}_{\mathbf{I}_R} \cdot \begin{bmatrix} q_0 & 0 & \cdots & 0 \\ q_1 & q_0 & \ddots & \vdots \\ \vdots & \vdots & \ddots & 0 \\ q_{N-1} & q_{N-2} & \cdots & q_0 \end{bmatrix} \tag{3.29}$$

where the sequence $q_0, q_1, q_2, \ldots, q_{N-1}$ is the impulse response of a desired IIR low-pass filter $Q(z)$ for n time steps. Process of being filtered through **Q** is the same as using the forward-backward filtering through $Q(z)$, giving the filter zero-phase with magnitude of $|Q^2(z)|$. Although there the one-sided z-transform is meant for signals of length $N = \infty$, it has been shown that if (**Q**), and (**L**) are causal, frequency domain analysis using $Q(z)$ and $L(z)$ for stability is valid for fixed-length ILC [NG02, AOR96, BTA06]. If **Q** and **L** are noncausal, time-delays can be added to compensate for the delays. Impulse response truncation and lengthening of the reference are techniques previously mentioned in the RC case (Section 3.2).

Remark 3.3.3. *Matlab has a built-in function,* `filtfilt(·)`*, for forward-backward zero-phase filtering.*

Remark 3.3.4. *Notice for the* $j = 0$ *case, assuming* $\mathbf{Q} = \mathbf{I}$ *and* $\mathbf{u}_0 = \mathbf{y}_0 = [0 \ldots 0]^T$,

then (3.24) reduces to

$$\mathbf{u}_1 = \mathbf{L}\mathbf{r} \tag{3.30}$$

which is the feedforward control case.

CHAPTER 4

Fixed-Point Control Example of a Levitated Shaft using a FPGA

4.1 Experimental System Description

For the purpose of demonstrating the proposed approach with the FPGA implementation, a Magnetic Moment MBC 500 levitated shaft system [40] was used as the plant for a control experiment. The FPGA used is the National Instruments PXI-7833R FPGA board, which carries a Virtex II 3M gate chip having 96 general purpose 18×18 multipliers. A 16-bit word length was used to represent both filter coefficients and signals because this would produce the most efficient use of resources. Bit-shifting techniques were used to avoid overflow/saturation. Through system identification and dynamic decoupling/transformations, the four-input four-output MIMO system is decoupled into 4 separate single-input single-output (SISO) systems [CWW10]. For simplicity, we observe only the Y-axis translational and rotational systems since the X-axis is very similar. The resulting transfer functions are

$$P_1(s) = \frac{-681.1214(s-1651)(s+1381)}{(s+4045)(s+417.3)(s-387.3)} \tag{4.1}$$

$$P_2(s) = \frac{-589.6263(s-2582)(s+1612)}{(s+4070)(s+428)(s-441.6)} \tag{4.2}$$

where the subscript 1 and 2 represents the translational and rotational axes, respectively.

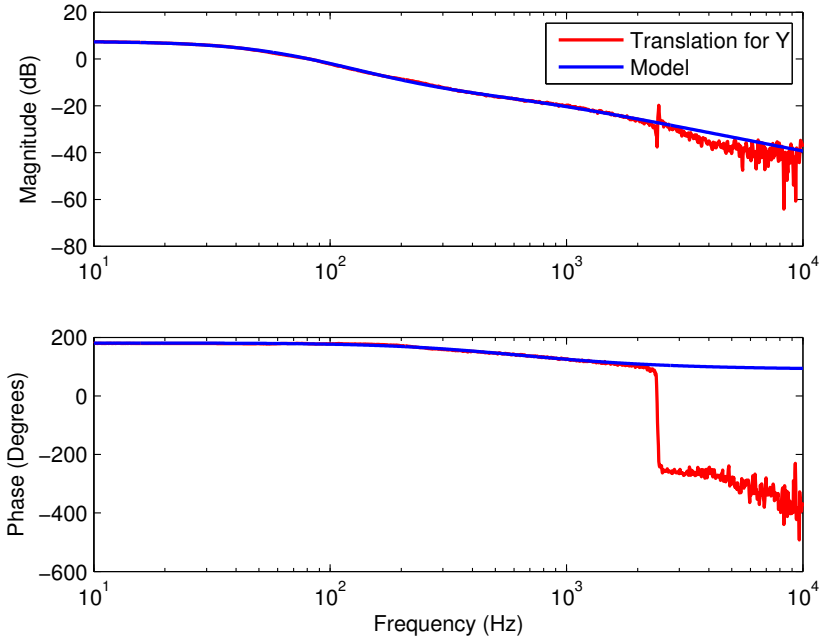

Figure 4.1: Curve fit of translational model data.

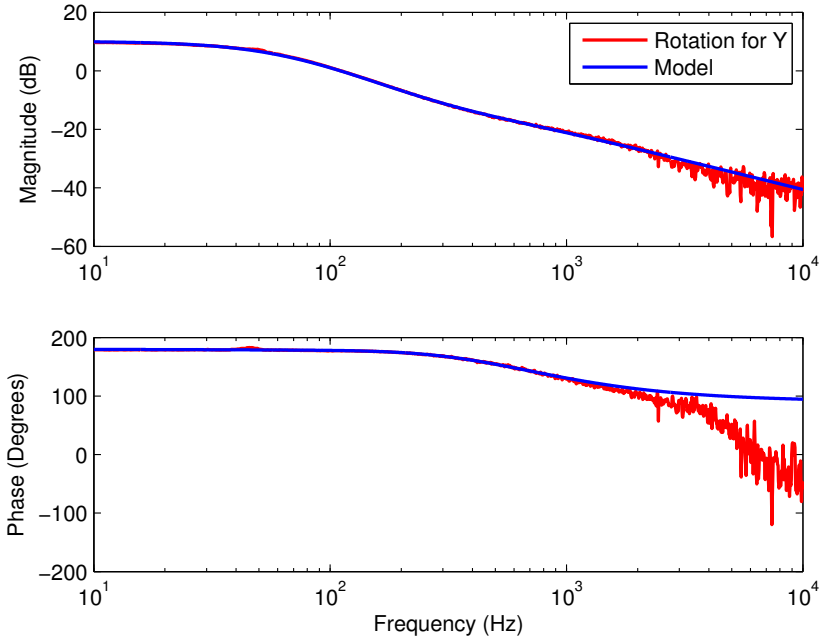

Figure 4.2: Curve fit of rotational model data.

Fig. 4.1 and 4.2 shows the fit of the model compared to actual frequency response

data collected. Both rotational and translational plants exhibit similar pole/zero locations with an unstable pole and a non-minimum phase zero. The Y-axis and X-axis have very little effect one another [CWW10], which justifies our analysis of only the Y-axis. The Y-axis is kinematically decoupled into *translational* and *rotational* models which are good for simple controllers. For the design of more complicated controllers such as RC, coupling effects may still play a role in performance.

Equation (4.1) shows that the decoupled systems are unstable. A simple lead controller with negative feedback is necessary to stabilize the closed-loop system. The controller was designed to reduce low frequency sensitivity. Since both the rotational and translational plants are similar, the stabilizing lead controller can be used for both axes. The continuous-time controller designed is

$$K(s) = \frac{3.0281(s + 430.7)}{(s + 1628)} \tag{4.3}$$

Digital controllers can be designed in the discrete-time domain directly or by approximating analog filters designed in the continuous-time domain. In the latter approach the sampling frequency must be sufficiently high to render desired approximation. Here, the discrete-approximation of the continuous-time lead compensator in (4.3) is approximated by a digital filter at an appropriate sampling rate. By applying the Tustin (Trapezoidal) Transformation to the lead compensator and zero-order-hold to the plant model, Fig. 4.3 illustrates the discrete sensitivity functions with different sampling frequencies and shows that a sampling frequency of $10kHz$ and above would be sufficient for approximating the analog filter. Nonetheless, to demonstrate a FPGA realization of the proposed Kurosu inversion filters at a sampling frequency beyond typical of a real-time target system, the stabilizing lead compensator and subsequent feedforward and repetitive controller will be designed and implemented at 100 KHz.

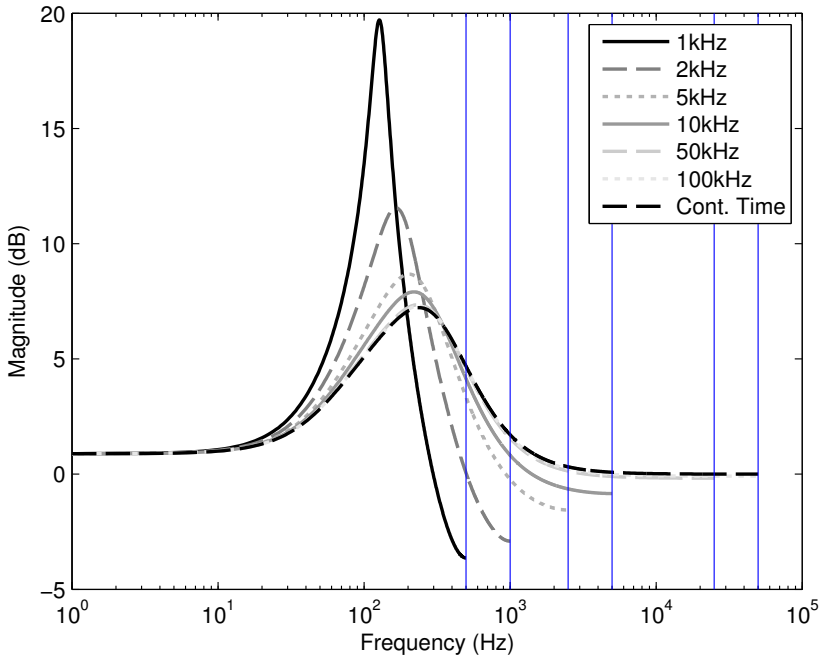

Figure 4.3: Sensitivity function with different sampling rates.

The resulting discrete controller and closed-loop plants are

$$K(z) = \frac{3.0281(1 - 0.9957z^{-1})}{(1 - 0.9839z^{-1})} \tag{4.4}$$

$$\begin{aligned} G_1(z) = &\frac{-0.020062z^{-1}(1 - 1.017z^{-1})}{(1 - 1.989z^{-1} + 0.9892z^{-2}))} \\ &\cdot \frac{(1 - 0.9957z^{-1})(1 - 0.9863z^{-1})}{(1 - 1.975z^{-1} + 0.9751z^{-2})} \end{aligned} \tag{4.5}$$

$$\begin{aligned} G_2(z) = &\frac{-0.017307z^{-1}(1 - 1.026z^{-1})}{(1 - 0.9957z^{-1})(1 - 0.9843z^{-1})} \\ &\cdot \frac{(1 - 0.9957z^{-1})(1 - 0.984z^{-1})}{(1 - 1.981z^{-1} + 0.9817z^{-2})} \end{aligned} \tag{4.6}$$

$G_1(z)$ and $G_2(z)$ represents the discrete-time closed-loop translational plant and rotational plant, respectively. From here we will assume that translational plant and rotational plant are completely decoupled. For brevity, we will address only the translational plant since control design and issues for the rotational plant will be similar.

4.2 Repetitive Control with ZPETC

We first consider realizing the RC using ZPETC for plant inversion in the structure from Fig. 3.2. ZPETC [Tom87] produces the following compensator

$$\begin{aligned} F_{ZPETC}(z) &= \left[\frac{1}{b^{-}(1)^2}\right]\frac{b^{-\star}(z)a(z)z^{-2}}{b^{+}(z)} \\ &= 1.83\times 10^{5}\cdot\frac{(1-0.9836z^{-1})z^{-2}a(z)}{b^{+}(z)}. \end{aligned} \tag{4.7}$$

The Kurosu's filter realization of the ZPETC introduces additional three sample delays. Representing the gain of the ZPETC in (4.7) requires at least 18-bits. Additional bits will be needed for the representation of the input signal. Fig. 4.4 gives an indication of signal dynamic range, where ZPETC is about $120dB$ while the proposed inversion is only about $40dB$. Thus, ZPETC implementation would require significantly larger word length than the 18-bits already required for the filter gain. The ZPETC can be cascaded with a linear phase low-pass filter to lower the large gain at high frequencies but would require too many multipliers and too much computation time to fit within the desired sampling interval. As a result, ZPETC for this system cannot be implemented under the 16-bit constraint.

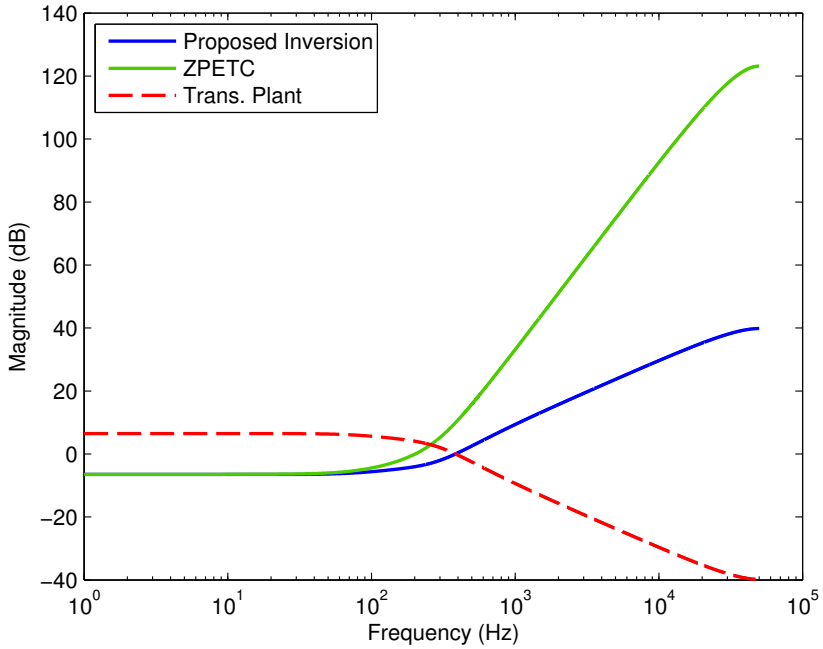

Figure 4.4: Magnitude comparison of ZPETC and proposed inversion filter.

4.3 Repetitive Control with Approximate Inversion

A FF controller using the inversion filter from (3.3) and (3.4) was created to track a delayed reference by inverting the closed-loop stabilized plant $G(z)$. In fact, the numerator of (3.3) need not be $a(z)$ and can be distributed into the numerator of (3.4) while retaining the same $F(z)G(z)$. More specifically, some of the poles of $G(z)$, $a(z)$, are moved to mirrored zeros of $H(z)$ to change the individual dynamic range of $H(z)$ and $F^+(z)$ in order to prevent internal overflow/saturation. $H_1(z)$, $H_2(z)$, $F_1^+(z)$ and $F_2^+(z)$ were chosen to minimize L and adjust the dynamic range and coefficient range to fit within a 16-bit framework. $H_1(z)$ and $H_2(z)$ are the transfer function inside of $f_1^-(z)$ and $f_2^-(z)$, respectively. The $4L$ delay introduced by the inversion filter is absorbed into the delay of the repetitive control loop. For the levitated shaft system, the designed filters are

$$H_1(z) = \frac{6.9641(1 - 2.011z^{-1} + 1.011z^{-2})}{(1 - 0.9836z^{-1})} \tag{4.8}$$

$$H_2(z) = \frac{7.4348(1 - 2.018z^{-1} + 1.019z^{-2})}{(1 - 0.9745z^{-1})} \tag{4.9}$$

$$H_{1,L}(z) = \frac{1.8521 \times 10^{-6}}{(1 - 0.9836z^{-1})} \tag{4.10}$$

$$H_{2,L}(z) = \frac{2.7084 \times 10^{-8}}{(1 - 0.9745z^{-1})} \tag{4.11}$$

$$F_1^+(z) = \frac{6.9641(1 - 1.975z^{-1} + 0.9751z^{-2})}{(1 - 0.9957z^{-1})(1 - 0.9863z^{-1})} \tag{4.12}$$

$$F_2^+(z) = \frac{7.4348(1 - 0.9957z^{-1})(1 - 0.9843z^{-1})}{(1 - 0.9957z^{-1})(1 - 0.984z^{-1})} \tag{4.13}$$

From Fig. 4.5, we chose $L_1 = 500$ for H_1 since it ensures that the impulse response after 500 samples are well below a 16-bit quantization level. Notice that the first few steps are large due to the NMPZs and the slow decay is associated with the pole. Since, the translational and rotational plants are similar, H_1 and H_2 have similar impulse responses. Thus, $L_1 = L_2 = 500$ for H_1 and H_2. When the gains in (4.10) and (4.11) are realized, they are effectively zero since they are below the 16-bit fixed-point quantization level. The inversion filter dynamic range indicated by Fig. 4.4 and the Kurosu filter gains and coefficients in (4.8) to (4.13) suggest that it is feasible to realize it with the FPGA 16-bit fixed-point arithmetics.

The RC was designed for a fundamental frequency of 25 Hz and $f_s = 100$kHz with $N = 4000$, $N_1 = 2001$, and $N_2 = 1999$. Notice that $N_2 \neq 2000$ since a one-step delay was used to make $H_1(z)$ and $H_2(z)$ causal. Having a delay in $H_1(z)$ and $H_2(z)$ acts as a preview when placed in Kurosu's filter which makes $N_2 = 1999$. The RC should track periodic reference signals and reject periodic disturbances of 25 Hz and its harmonics.

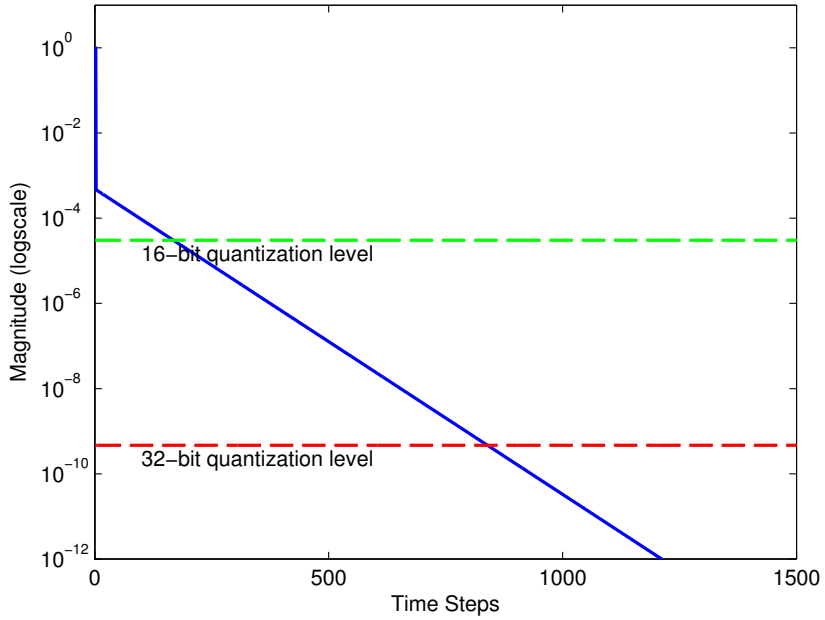

Figure 4.5: Absolute value of impulse response of $H_1(z)$: $|h_1(k)|$ overlaid with 16-bit and 32-bit quantization levels.

To ensure stability and robustness of the RC, a linear phase low-pass filter is often chosen. A sharp gain drop-off enables the closed-loop system to achieve robustness without sacrificing the closed loop bandwidth. Kurosu's linear phase IIR filter (Fig. 2.4) is the compromise between the two. It uses the computational complexity of an IIR filter while retaining the linear phase of an FIR filter. This implementation is ideal for the low-pass q-filter in RC.

Recall, that (3.18) is a sufficient condition for the RC to be robustly stable is by designing a $q(z)$ such that it is lower than the inverse of the multiplicative modeling error. To realize the linear phase low-pass q-filter, an IIR filter $Q_{base}(z)$ can be designed using any traditional filter design techniques [Mit04] to obtain the desired magnitude characteristic. After placing this filter in the Kurosu's Filter from Section 2.3, the resulting filter will become $q(z) \approx Q^{\star}_{base}(z)Q_{base}(z) \cdot z^{-4L_q}$. More specifically, use $H(z) = Q_{base}(z)$ inside Kurosu's filter.

An IIR filter with corner frequency $1500Hz$ was designed to satisfy robustness as

shown in Fig. 4.6 where,

$$Q_{base}(z) = \frac{0.00041651(z+1)^2}{(z^2 - 1.941z + 0.9431)} \tag{4.14}$$

$$q(z) \approx Q^{\star}_{base}(z) Q_{base}(z) z^{-4L_q}. \tag{4.15}$$

In Fig. 4.6, the actual closed loop discrete-time plant G_a in (3.18) is calculated from the zero order hold equivalence of the open loop continuous-time plant data (Fig. 4.1 and 4.2) and the discrete-time lead compensator $K(z)$. It should noted that besides the fitting error of the model with respect to the frequency response, inaccuracies originate from the decoupling and linearization of the inherently nonlinear coupled systems. Consequently, the q-filter was designed with sufficient margins above $2.5kHz$ (Fig. 4.6). Approximate impulse response length was chosen to be $L_q = 200$. This means the equivalent FIR filter would require 200 multiplier taps. This also means $N_q = 4L_q = 800$. Typically,

$$q_{121}(z) = 0.25 + 0.5z^{-1} + 0.25z^{-2} \tag{4.16}$$

is a popular choice [TT94, HT98, RTH94, TSH08] since it is linear phase and can be realized by shift registers instead of multipliers. A large drawback is its fixed roll-off frequency. For systems requiring lower roll-off frequency, cascading multiple instances of $q_{121}(z)$ is possible. To obtain similar roll-off characteristics as $|Q_{base}(z)|^2$, it requires close to 80 cascaded q_{121} filters. Another option is using MATLAB's `fdatool` to generate linear phase filter using the least-squares method [PB87]. One result is 100^{th} order FIR filter $q_{ls}(z)$. Fig. 4.7 compares the magnitude responses of different possible FIR robustness filters. Observe that our Kurosu q-filter is realized by two 2^{nd} order IIR filters, that uses only 6 multipliers, which is a significant reduction in resources when compared to high-order linear phase FIR filters.

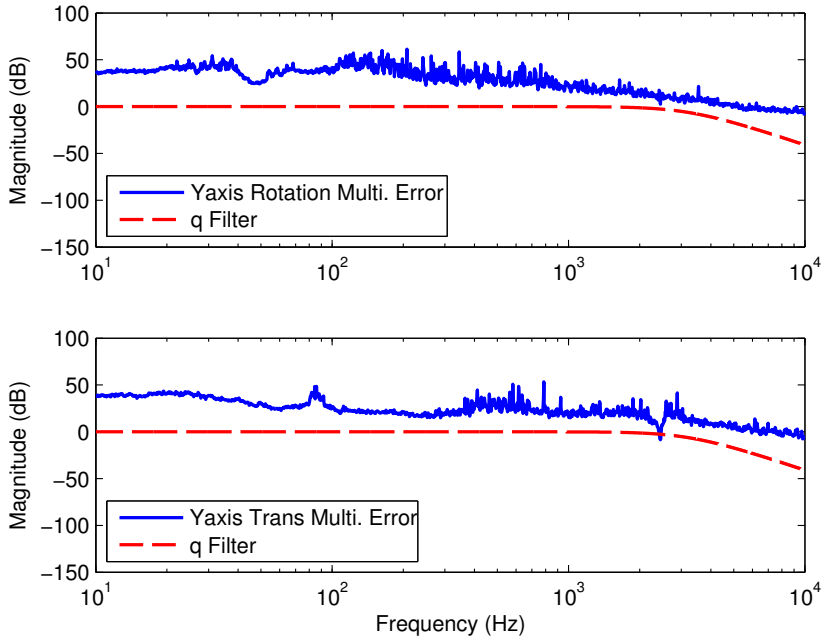

Figure 4.6: Magnitude of q-filter vs. inverse of multiplicative modeling error for robust stability of translational and rotational systems.

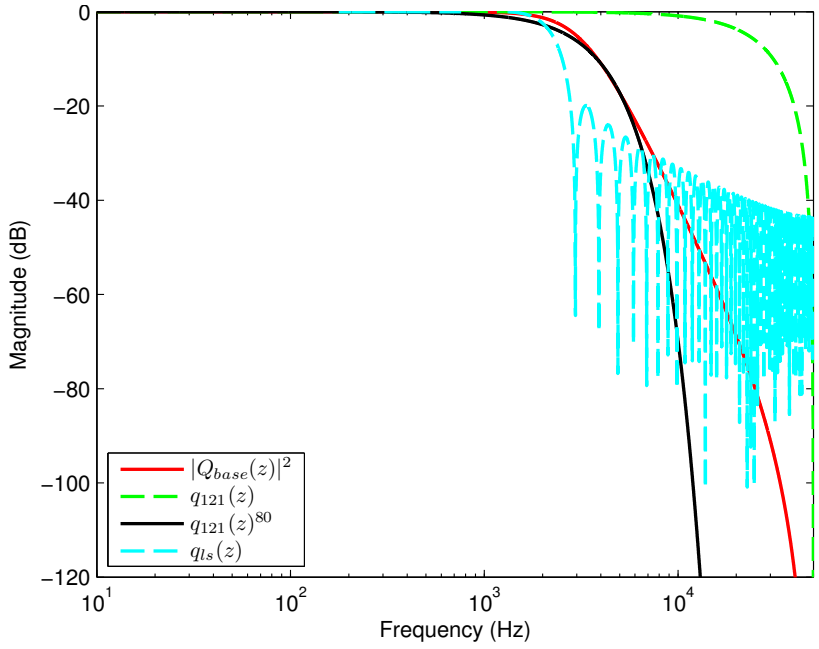

Figure 4.7: Comparing different robustness filters.

4.4 Experimental Results - Levitated Shaft Example

The control experiment was performed using only 56 of the 96 general purpose 18×18 multipliers in parallel. Had we used the FIR implementation for $f^{-}(z)$, approximately a 500^{th} order filter, we would have been forced to reuse multipliers and reduce the maximum servo rate to accommodate for the computation time. Representative experimental results shown in Fig. 4.8, 4.9, 4.10, 4.11, 4.12 and 4.13 are discussed below.

Figs. 4.8, 4.9, 4.10 show tracking and error power spectral density (PSD) of a $50Hz$ $244\mu m$ peak-to-peak triangular wave, respectively. The lead controller for the translational axis attempts to track a periodic reference. However, the presence of the periodic waveform on the rotational axis, which is only under regulation, infers that the lead controller is unable to compensate for the coupling between the translational and rotational axis. Furthermore, the "parabolic" shape seen in the translational output under lead control can be attributed to magnetic nonlinearity and amplifier saturation. The addition of our feedforward inversion provides better performance against the coupling disturbances and nonlinearities than just only the lead controller. RCs were designed for both translational and rotational systems. The translational RC is used for reference tracking and the rotational RC is meant for disturbance rejection. Capable of compensating for Fourier harmonics, the two RCs effectively reduce the tracking errors caused by linear dynamics and nonlinearities. The RMS error values for the lead-only feedforward tracking control and RC are listed in Table 4.1 to better illustrate the performance. Since the rotational movement is under regulation with RC, the error value is also indicative of the noise level introduced by fixed-point quantization.

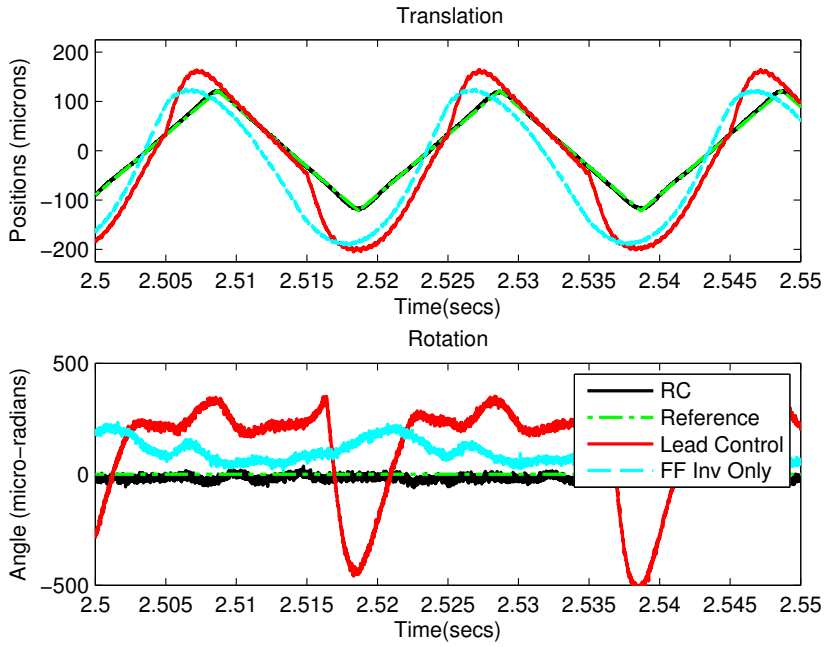

Figure 4.8: Experimental Results - Tracking performance of 50Hz triangular wave. Comparison of lead, FF and proposed RC.

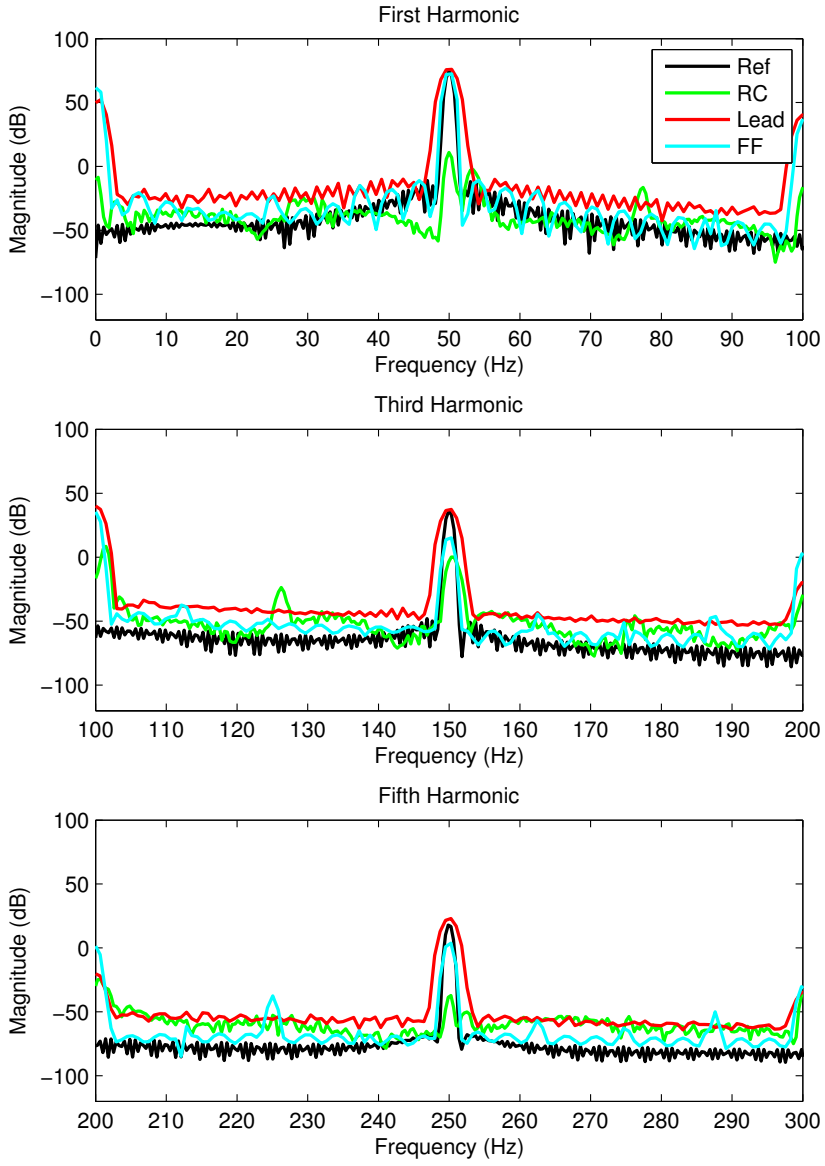

Figure 4.9: Error Power Spectral Density (PSD) of Translational Axis - Tracking performance of $\pm 122 \mu m$ 50Hz triangular wave. Comparison of lead, FF and proposed RC.

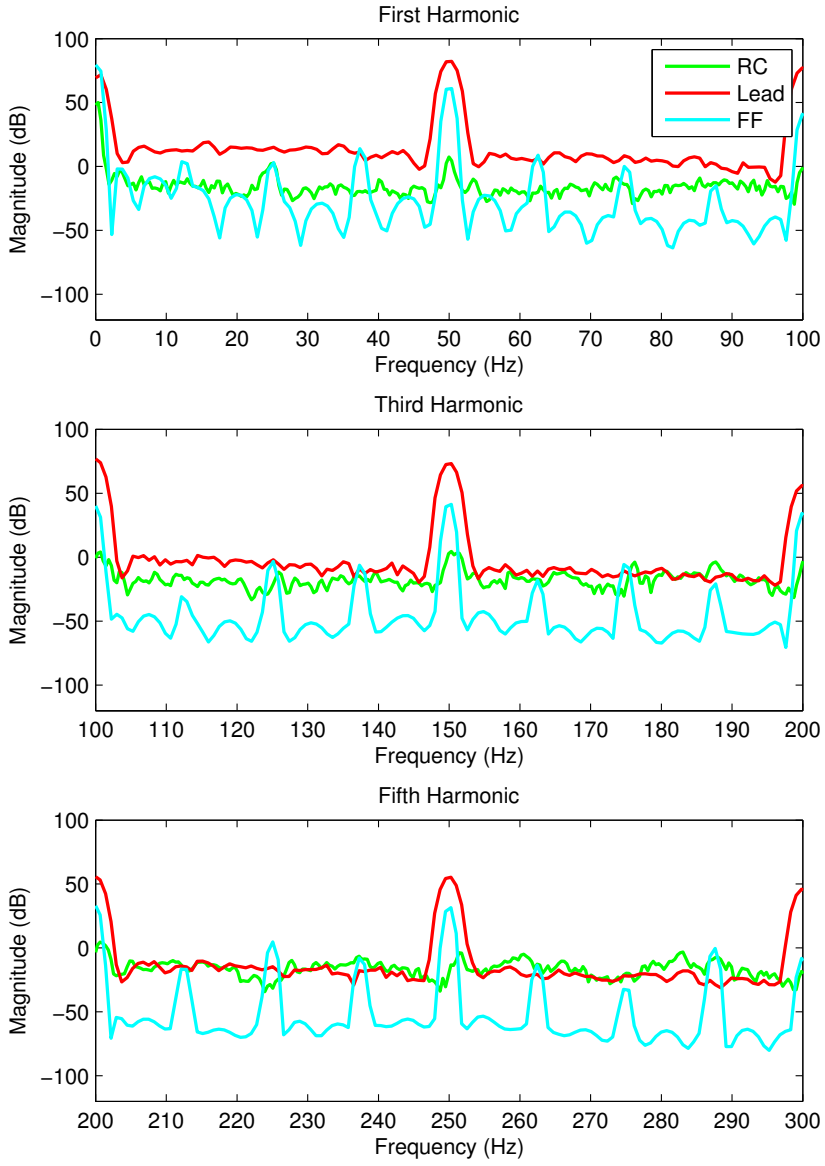

Figure 4.10: Error Power Spectral Density (PSD) of Rotational Axis under regulation at $0\mu rad$. Comparison of lead, FF and proposed RC.

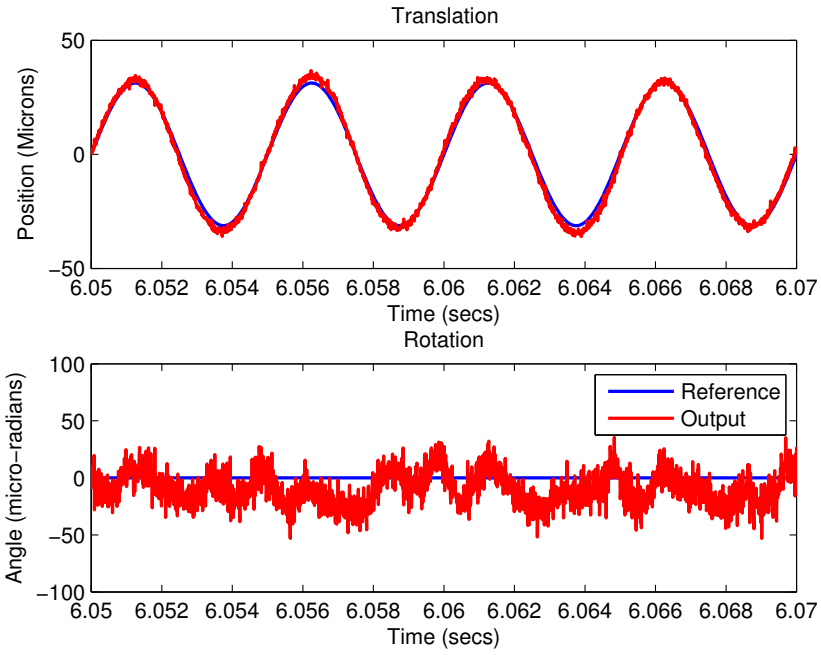

Figure 4.11: Experimental Results - Tracking performance of 200Hz sine wave under RC.

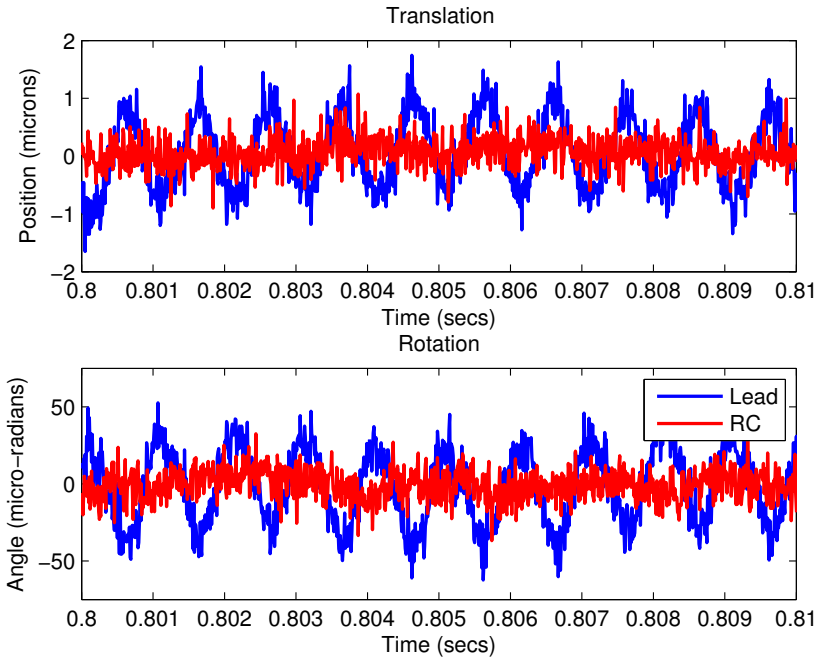

Figure 4.12: Experimental Results - Disturbance Rejection of 1kHz sine wave on rotation axis. Comparison of lead and proposed RC.

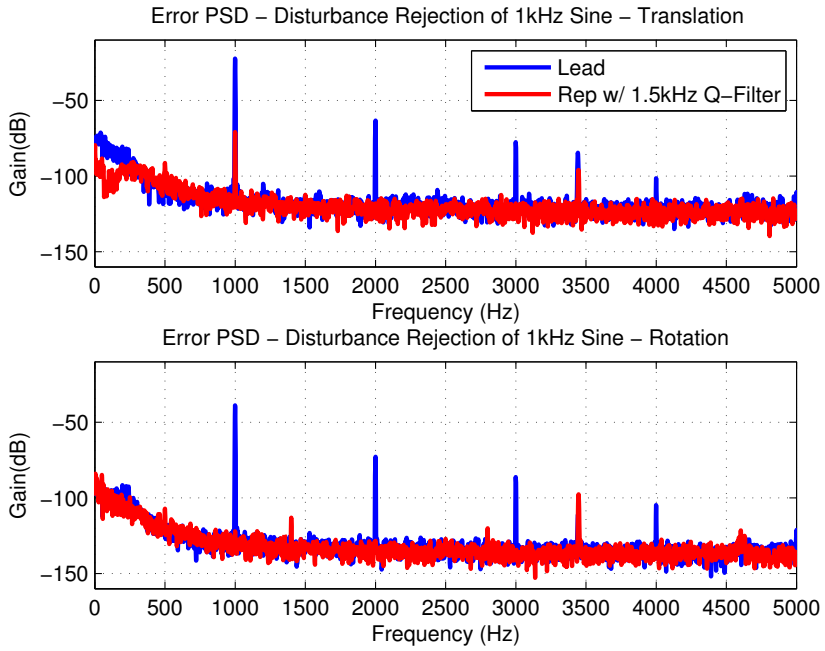

Figure 4.13: Error PSD - Disturbance Rejection of 1kHz sine wave on translational and rotation axis. Comparison of lead and proposed RC.

The significant errors in regulating the rotational axis in the lead control suggest significant coupling between the two axes. The FF control improves the performance somewhat even though Fig. 4.2 suggests that the modeling error is small below $1000Hz$. This is due to the axis coupling. RC is able to achieve substantially smaller error even under the unmodeled coupling effect. Note, that ZPETC Repetitive Control is not compared here because it is not implementable due to the hardware constraints of working in a 16-bit framework. Fig. 4.11 shows the performance of tracking a $200Hz$ sine wave in the translational axis and regulation in the rotational axis. In Figs. 4.12 and 4.13, a $1kHz$ sine wave disturbance is injected at the control input of the rotational axis, while both the rotation and translation axes were under regulation, to demonstrate disturbance rejection performance.

Table 4.1: Error RMS values, translation reference 244 micron peak-to-peak triangle wave and with rotational regulation.

Controller	Translation Error (μm RMS)	Rotation Error (μrad RMS)
Lead	123.42	250.50
Feedforward	89.64	115.98
Repetitive Control	3.47	19.97

CHAPTER 5

Improved Fixed-Point Controller Performance using the Delta Operator

5.1 Background on the Delta Operator

A typical issue that plagues fixed-point realization of controllers is the issue of FWL effects. Given some coefficient truncation, phase and magnitude characteristics can be affected. The obvious fix of increasing the word length is often prohibitive in terms of both limited resources and/or timing restrictions. Given these constraints, the typical solution is to use different filter realizations, such as Direct Form II transposed (DFIIt), to ameliorate quantization noise and FWL effects [Mit04]. Quantization noise becomes a lower bound to which an error control signal can achieve. With high sampling rate, high performance, FWL controllers, the levels of quantization noise can be substantial [CT10a]. The Delta Operator is a filter realization specifically used for high sampling rates and to mitigate these FWL effects. Middleton and Goodwin first introduced the Delta Operator and studied FWL with and without the Delta Operator [MG86]. The advances with the Delta Operator have brought about methods in which the Delta Operator uses slightly more computational resources while providing substantial robustness against FWL effects [KLH98, CC07]. This dissertation investigates the performance benefit of using the Delta Operator filter form versus the DFIIt filter form for the repetitive control of a piezo-electric actuator.

The Delta Operator is defined as

$$\delta^{-1} = \frac{\Delta q^{-1}}{1 - q^{-1}} \tag{5.1}$$

where q^{-1} is the shift operator in the time-domain. The equivalent frequency domain representation is

$$\gamma^{-1} = \frac{\Delta z^{-1}}{1 - z^{-1}}, \tag{5.2}$$

where Δ is the sampling period [MG86]. The block-diagram filter realization for the Delta Operator is shown in Fig. 5.1.

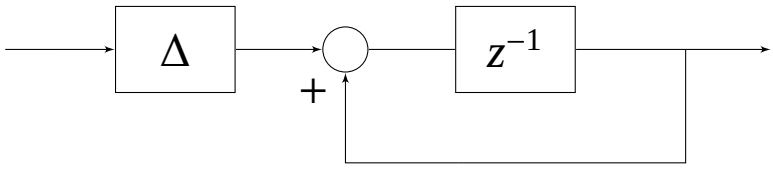

Figure 5.1: Implementation of Delta Operator - γ^{-1}

Fig. 3.3 showed that a controller can be broken into series of second order transfer functions/filters defined as

$$H_k(z) = \frac{b_{0,k} + b_{1,k}z^{-1} + b_{2,k}z^{-2}}{1 + a_{1,k}z^{-1} + a_{2,k}z^{-2}} \tag{5.3}$$

where k denotes the k^{th} second-order section (SOS).

A mapping of $H_k(z)$ into $H_k(\gamma)$, exists as

$$H_k(z)\Big|_{z=1+\Delta\gamma} = H_k(\gamma). \tag{5.4}$$

The resulting Delta Operator SOS would be

$$H_k(\gamma) = \frac{\beta_{0,k} + \beta_{1,k}\gamma^{-1} + \beta_{2,k}\gamma^{-2}}{1 + \alpha_{1,k}\gamma^{-1} + \alpha_{2,k}\gamma^{-2}} \tag{5.5}$$

Figure 5.2 shows how the DFIIt structure is similar where the shift operator, z^{-1}, is replaced by the Delta Operator, γ^{-1}. There exists a relationship between z-domain coefficients and γ-domain coefficients shown in Table 5.1 and 5.2. Δ does not necessarily

have to be the sampling period since the Δ values in γ^{-1} and coefficients β_k, α_k cancel each other out [KLH98]. To save on multipliers, Δ can be chosen to be a power of 2 which can be efficiently realized as simple bit shifts.

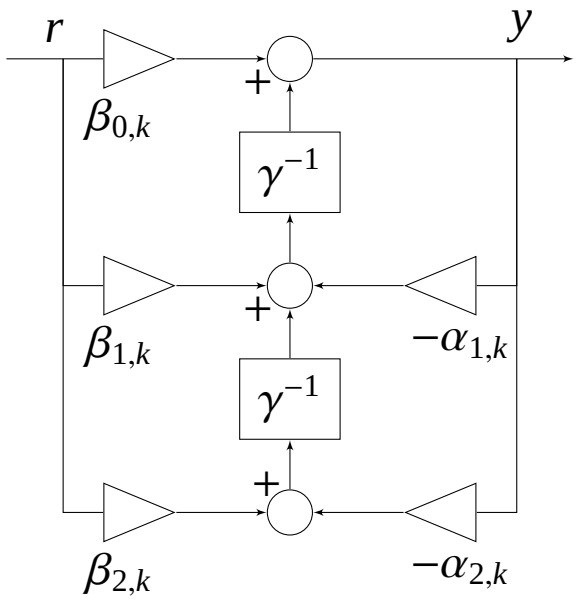

Figure 5.2: Direct Form II Tranposed Implementation - Delta Operator

β_0	$\beta_0 = b_0$	α_0	$\alpha_0 = 1$
β_1	$\beta_1 = \frac{b_0+b_1}{\Delta}$	α_1	$\alpha_1 = \frac{1+a_1}{\Delta}$

Table 5.1: Delta Operator Coefficient Mapping of First Order Transfer Function

β_0	$\beta_0 = b_0$	α_0	$\alpha_0 = 1$
β_1	$\beta_1 = \frac{2b_0+b_1}{\Delta}$	α_1	$\alpha_1 = \frac{2+a_1}{\Delta}$
β_2	$\beta_2 = \frac{b_0+b_1+b_2}{\Delta^2}$	α_2	$\alpha_1 = \frac{1+a_1+a_2}{\Delta^2}$

Table 5.2: Delta Operator Coefficient Mapping of Second Order Transfer Function

The Delta Operator can be viewed as the forward difference mapping of the unit circle, in the z-domain, and mapping it into a pseudo s-domain (continuous time) domain. Results from [MG86] show that with sufficiently high-sampling rate, 12-bit Delta Operator representation is capable of significantly lower quantization noise than a 12-bit shift operator realization. In addition, for every delay in the DFIIt form, the Delta Operator form uses one extra addition which is inexpensive compared to a multiplier. The Delta Operator is attractive due to its potentially substantial decrease of quantization noise comes at a slight increase in computational complexity [FPW97]. The use of the Delta Operator seems to be a very useful in the cases of high-sampling rate control on a FPGA as in Section 5.2.

5.2 A Fixed-Point Delta Operator Repetitive Control Example on a Piezoelectric Device

5.2.1 System Identification

The experimental example used for this dissertation is a piezoelectric cutting tool actuator designed for dynamic variable depth of cut machining [Lin08]. In addition to being mechanically preloaded to reduce hysteresis, a discrete-time PI controller, $C(z)$, was added to prevent position drift of the open loop system, $P(z)$, during regulation and to reduce the magnitude of the resonant peak of the open-loop system.

The identified closed-loop system in Fig. 5.3, $G(z) = CP/(1+CP)$, was excited with a pseudo random binary sequence (PRBS) at $100kHz$ sampling rate.. The following input-output relationship was obtained through Prediction Error Method (PEM) type system identification techniques

$$G(z) = \frac{-0.00020352(z - 7.577)(z - 0.8446)}{(z - 0.9707)(z^2 - 1.849z + 0.8812)} \times \frac{(z^2 - 1.724z + 0.9237)(z^2 - 3.222z + 3.559)}{(z^2 - 1.885z + 0.9829)(z^2 - 1.214z + 0.8176)}. \tag{5.6}$$

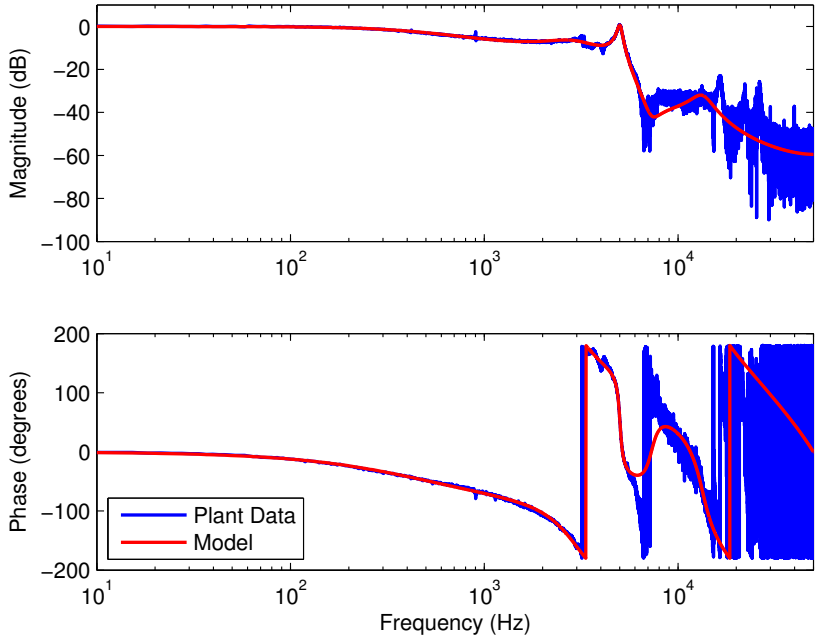

Figure 5.3: Plant Data vs. Model

5.2.2 Repetitive Control of Piezoelectric Actuator

Figure 5.4 illustrates a simple RC plug-in structure where $F(z)$ and $F(\gamma)$ is a type of feedforward inversion of the closed-loop plant $G(z)$ [WWZ05, TTC89]. Other variations of RC include the ZPETC repetitive control structure [Tom87, TTC89]. $F(z)$ is constructed through the methods of Section 3.1.1 with $L = 50$. The corresponding

controllers are

$$F^{+}(z) = \frac{-4913.5972(1 - 0.9707z^{-1})}{(1 - 0.8446z^{-1})} \times \frac{(1 - 1.849z^{-1} + 0.8812z^{-2})(1 - 1.885z^{-1} + 0.9829z^{-2})(1 - 1.214z^{-1} + 0.8176z^{-2})}{(1 - 1.724z^{-1} + 0.9237z^{-2})} \tag{5.7}$$

$$H(z) = \frac{-1}{26.971 - 27.97z^{-1} + 10.8z^{-2} - z^{-3}} \tag{5.8}$$

$$H_L(z) = \frac{-7.518e - 016 + 7.878e - 016z^{-1} - 9.089e - 017z^{-2}}{1 - 1.037z^{-1} + 0.4005z^{-2} - 0.03708z^{-3}} \tag{5.9}$$

For $F(\gamma)$, techniques from [KLH98] were used to choose appropriate power-of-2 values for Δ for every SOS.

$$F^{+}(\gamma) = -\frac{(0.3756 + 0.1878\gamma^{-1})(+0.113851526151328 + 0.1138\gamma^{-1} + 0.0284\gamma^{-2})}{(1 + 0.4340\gamma^{-1})(1 + 0.5473\gamma^{-1} + 0.0939\gamma^{-2})} \tag{5.10}$$

$$H(\gamma) = \frac{-4067.6(6.0212 + 0.0881\gamma^{-1})(4.2894 + 0.3240\gamma^{-1} + 0.0346\gamma^{-2})}{(1 + 0.0777\gamma^{-1})(1 + \gamma^{-1} + 0.2500\gamma^{-2})^2} \times \frac{(0.3008 + 0.1183\gamma^{-1} + 0.0454\gamma^{-2})(1 + 0.0574\gamma^{-1} + 0.02445\gamma^{-2})}{(1 + 0.1378\gamma^{-1} + 0.0498\gamma^{-2})} \tag{5.11}$$

N_1 and N_2 are chosen such that $N_1 + N_2 + N_q = N$ and $N = f/f_s$. f_s is the sampling frequency, f is the fundamental frequency of the periodic reference or disturbance, and N_q is the equivalent linear phase delay introduced by the low pass filter $q(z)$. RC will track and reject the fundamental frequency and all of its harmonics of the reference and disturbance, respectively. For our application, the sampling frequency was chosen such that $f_s = 100kHz$. The high sampling rate is necessary to track a $1kHz$ triangular wave. A base frequency of $250Hz$, $N = 400$, to accommodate the long delay from $F(z)$ or $F(\gamma)$ and since 250 is common factor of 1000.

$q(z)$ was chosen to be $0.25 + 0.5z^{-1} + 0.25z^{-2}$ with an equivalent phase delay of z^{-N_q} with $N_q = 1$. $q(z)$ is a linear phase low pass filter where the coefficients are represented as efficient bit shifts instead of multipliers. The $q(z)$ in the repetitive controller serves

to help stabilize the system and can be seen as a frequency dependent learning gain [TT94]. Given the choice of $q(z)$ and L, then $N_2 = 4L + \rho + d = 200 + 3 + 1 = 204$ and $N_1 = N - N_2 - N_q = 400 - 204 - 1 = 195$.

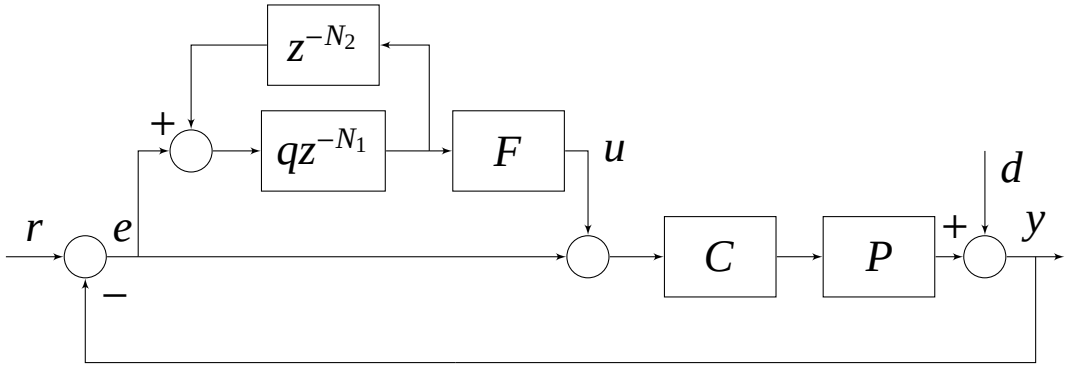

Figure 5.4: Add-on RC with feed-forward inversion and PI controller.

5.2.3 Experimental Results

A Xilinx Virtex 5 based National Instruments FPGA PCIe-7852 was used to implement both, the DFIIt and Delta Operator, forms of a repetitive controller. All filter (controller) coefficients were implemented using a 16-bit representation while signals were represented with 32-bit to ameliorate overflow/saturation effects. A sampling rate of $100kHz$ was used to track high frequency periodic references for both the DFIIt and Delta Operator implementation.

Figures 5.5 and 5.6 compares the experimental results when tracking a $1kHz$ triangular reference. The reference tracking plot in Fig. 5.5 shows adequate performance for both the DFIIt and Delta Operator realization. The error signals in Fig. 5.5 show that both filter forms approach their respective quantization noise floor. As seen in [CT10a], quantization noise floor prevents the error from reaching absolute zero. As predicted, the Delta Operator error is much lower than the DFIIt realization's error. The control signal, u, of Fig. 5.5 refers to only the contribution of the repetitive controller portion (output of $F(z)$). We can see in the "control signal" that the signal is not purely periodic but laced with quantization noise. Figure 5.6 shows the power spectral density (PSD)

of the error signal of the DFIIt and Delta operator realization. Quantization noise can be seen in in both the DFIIt and Delta Operator across all frequencies bands but the Delta Operator has much less.

Quantization noise is typically a function of both the filter (controller) and the reference signal. Table 5.3 lists the max error, $|e_{max}|$, and RMS error, e_{RMS}, for reference signals with varying magnitudes and frequencies. In the Delta Operator realization, the errors grow linearly with respect to the magnitude of the reference. However, this is not the case in the DFIIt realization where the errors have become much larger than expected from the linear growth as the reference magnitude increases beyond some level. At small magnitudes, such as regulation of $0\mu m$ reference, the e_{RMS} of Delta and DFIIt are comparable but $|e_{max}|$ of DFIIt is significantly larger. Overall, the general trend shows that the errors $|e_{max}|$ and e_{RMS} is almost always smaller in the Delta Operator form than the DFIIt for this experiment.

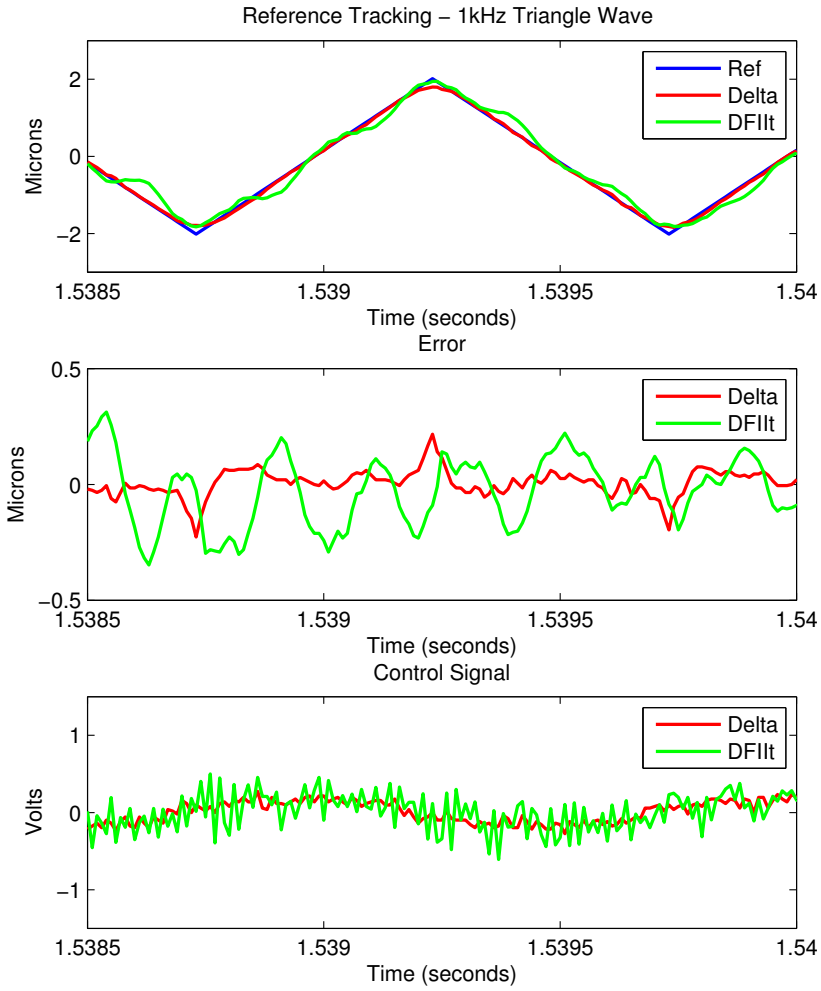

Figure 5.5: Reference tracking, error, and control signal for reference of 1kHz triangular wave.

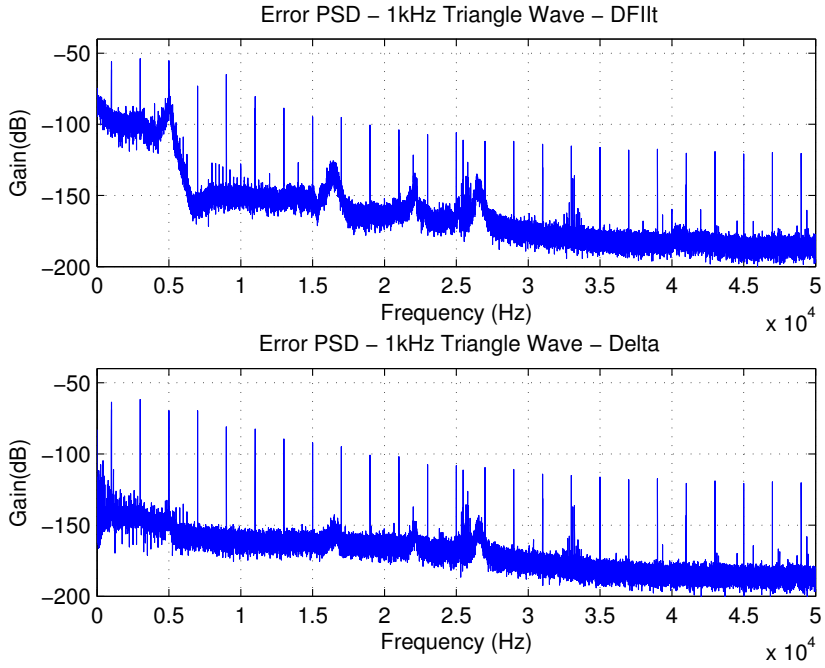

Figure 5.6: Error PSD for reference tracking of 1kHz triangular wave.

Table 5.3: Repetitive Control Performance with Delta Operator and DFIIt.

		e_{RMS} (μm)		$\lvert e_{max}\rvert$ (μm)	
	Triangular Reference	Delta	DFIIt	Delta	DFIIt
	$0\mu m$	0.0247	0.0251	0.1158	1.0675
$1kHz$	$\pm 0.50\mu m$	0.0288	0.0799	0.1460	2.0847
	$\pm 1.01\mu m$	0.0367	0.0989	0.2014	2.2609
	$\pm 2.01\mu m$	0.0589	0.3259	0.3172	3.2025
	$\pm 4.03\mu m$	0.1087	2.4832	0.5338	7.6437
	$\pm 8.06\mu m$	0.2190	4.5914	0.9869	11.0728
$2kHz$	$\pm 0.52\mu m$	0.0596	0.0869	0.1964	2.1350
	$\pm 1.04\mu m$	0.1105	0.2235	0.2870	3.1068
	$\pm 2.09\mu m$	0.2164	1.7736	0.4985	6.4806
	$\pm 4.19\mu m$	0.4260	3.0222	0.8812	8.7616

The efficient NMPZ inversion and RC structure haved been improved through the Delta Operator on a FPGA and established a still efficient repetitive control structure. The experimental results for controlling a piezoelectric actuator show that the improvement of Delta Operator over the DFIIt is significant. Since our FPGA realization does not reuse the computation resources to serialize the digital signal processing during the controller update, the sampling rate of the controller can be increased to near the FPGA clock speed, $40MHz$ in our case, if it is called for in other applications. At such high rate, it would be unlikely to realize high-order controllers, such as the inversion based repetitive control, without exploiting the efficient digital signal processing techniques discussed in this dissertation.

5.2.4 Tradeoff between Accurate Inversion versus Quantization Noise Reduction

Inherently, there are techniques to reduce quantization noise [KLH98], however it comes at the cost of inaccurate inversion. $F(z)$ from this chapter will serve as a demonstrative example to show the benefits of the Delta Operator has over the DFIIt.

Fig. 5.7 demonstrates *when the quantization noise* is ignored, the Delta Operator SOS , $F(\gamma)$, and DFIIt filter realization, $F(z)$, are pretty much the same. One thing to note that with 16-bits there is warping of the frequency response at higher frequencies of $F(z)$.

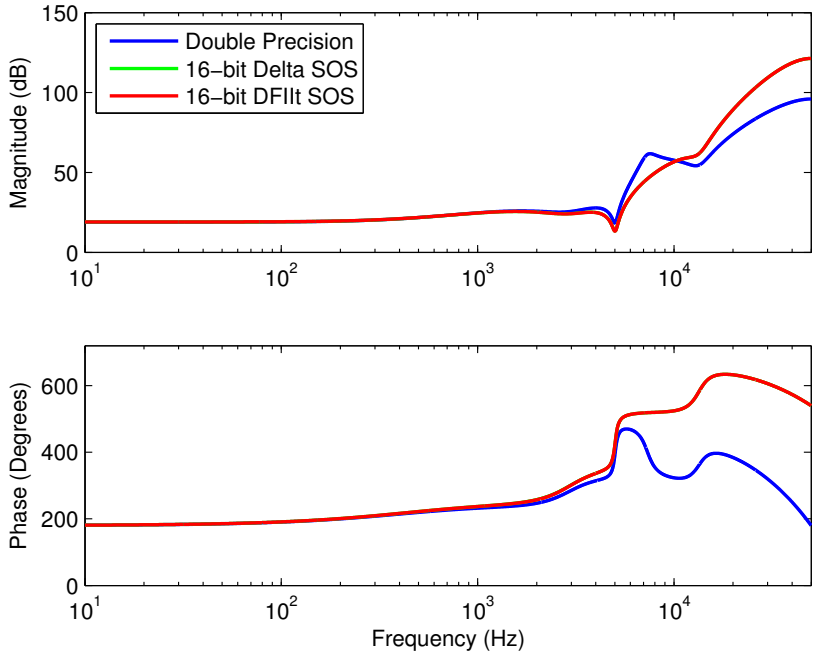

Figure 5.7: Bode plots of $F^{+}(z)$ with coefficient quantization and different filter realizations where $F(z)$ is designed for an accurate filter inversion.

Fig. 5.8 shows the results when some bits for $F(z)$ and $F(\gamma)$ are used to accommodate scaling gains to help reduce the effects of quantization noise. The DFIIt Realization ($F(z)$) and the Delta Operator SOS ($F(\gamma)$) had the same scaling gains applied but their frequency responses varies drastically. $F(\gamma)$ seems to mimic the double-precision version of $F(z)$ the best.

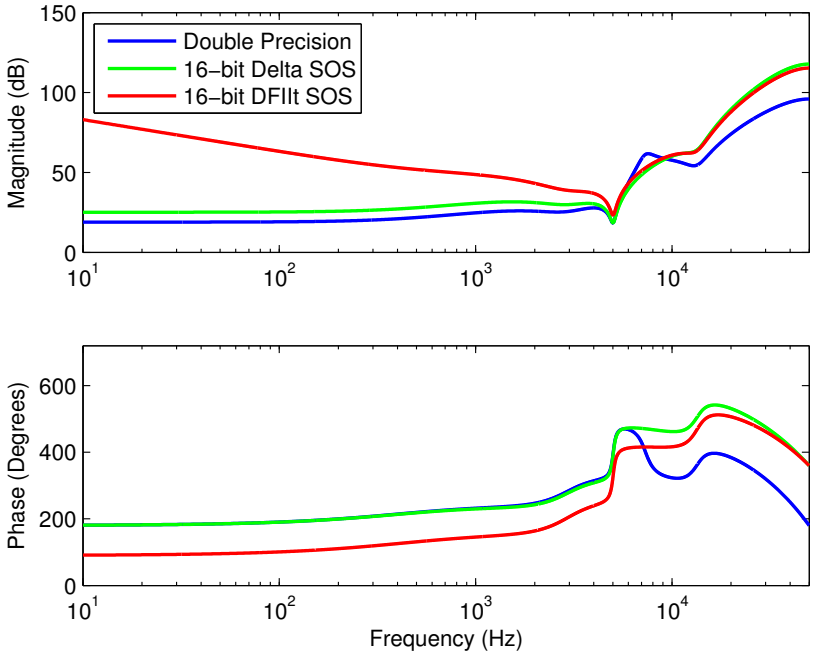

Figure 5.8: Bode plots of $F^+(z)$ with coefficient quantization and different filter realizations where $F(z)$ is designed to also reduce quantization noise.

Knowing that even for the best case, a 16-bit $F(\gamma)$ or $F(z)$ cannot perfectly invert the high frequencies accurately. Previous chapters have already established that with current hardware, double-precision repetitive control for the constraints of this experiment is not possible. This begs the question of whether is an offline equivalent of repetitive control that exists that can help serve as that performance benchmark. The next chapter will address this offline equivalence.

CHAPTER 6

Floating Point Control Example on a Piezoelectric Actuator using Iterative Learning Control

The same piezoelectric actuator, as described by (5.6) used in Section 5.2 is same the one used for this chapter. The sampling rate still remains at 100kHz with a closed-loop PI controller to ameliorate hysteresis effects. The ILC is performed on a Labview Realtime PC which transfers the desired profile to the National Instruments PCIe-7852 FPGA board to send the controller commands for every run of the experiment. The control law used were based off of (3.24), (3.26), (3.27), and (3.29). This section mainly serves to serve as a bench mark for the results of Section 5.2. Since we are unable to perform double-precision linear repetitive control due to hardware limitations, double-precision linear ILC can serve as an equivalent measure as long as initial/final conditions are well-behaved (as discussed in Section 3.3). For this chapter experimental results from the DFIIt RC (Section 5.2), Delta Operator RC (Section 5.2), ILC experimental ILC results, and ILC simulation results are gathered and compared.

To ensure that we are analyzing similar performance criterion between double precision ILC and fixed-point RC, the time-domain data used to calculate the error is important. In Fig. 6.1, illustrates what the RMS or max errors are based off of. The reference signal contains approximately 50 periods of the reference triangular wave. The beginning is zero-padded to allow for noncausal and pre-actuation solutions. The end is also zero-padded to set the same initial and final conditions to ensure accurate results if the forward-backward filtering technique is used to calculate these noncausal solutions. The error is computed from the 1000 points located in the middle of 50 tri-

angular periods. We choose these points in order to produce ILC errors that can be compared to the steady-state errors of the fixed-point RC results from Section 5.2.

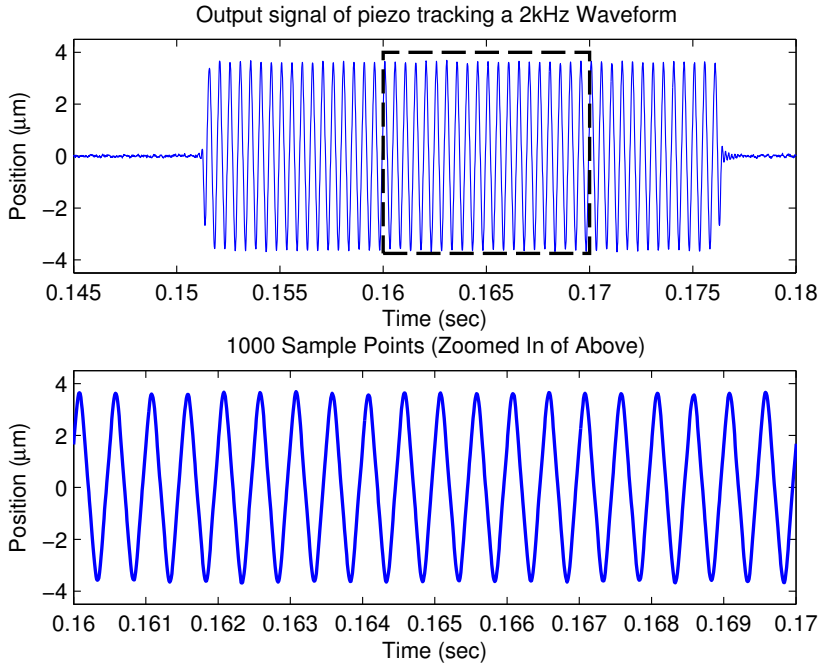

Figure 6.1: Error based on the middle 1000 sample points to establish equivalent error as RC.

6.1 Robust Stability Condition

The robust stability condition mentioned in (3.18), is a sufficient condition for stability. This means that for certain frequencies, it is possible that $|Q^2_{barely}| > \left|\frac{G}{G-G_a}\right|$ and the ILC converges. Fig. 6.4. Fig. 6.5 shows that $|Q_{bad}(z)|$ is only slightly greater than $|Q_{barely}(z)|$, the error of the ILC diverges. This also implies a degree of "necessity" for this robust stability condition for this specific example.

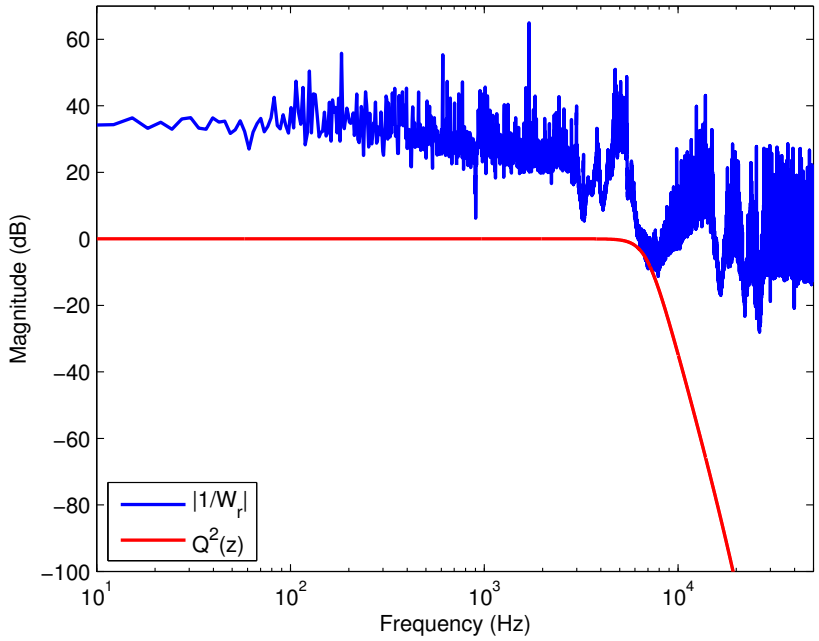

Figure 6.2: Plot of $\left|\frac{G}{G-G_a}\right|$ vs. $|Q^2_{barely}(z)|$, where Q violates robustness condition, but ILC error still converges.

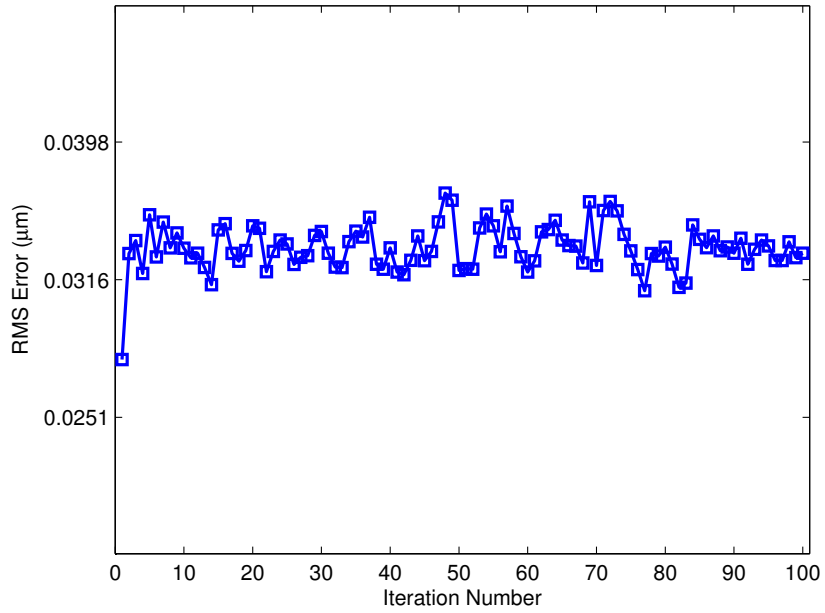

Figure 6.3: Q_{barely} slightly violates robustness condition, ILC still converges.

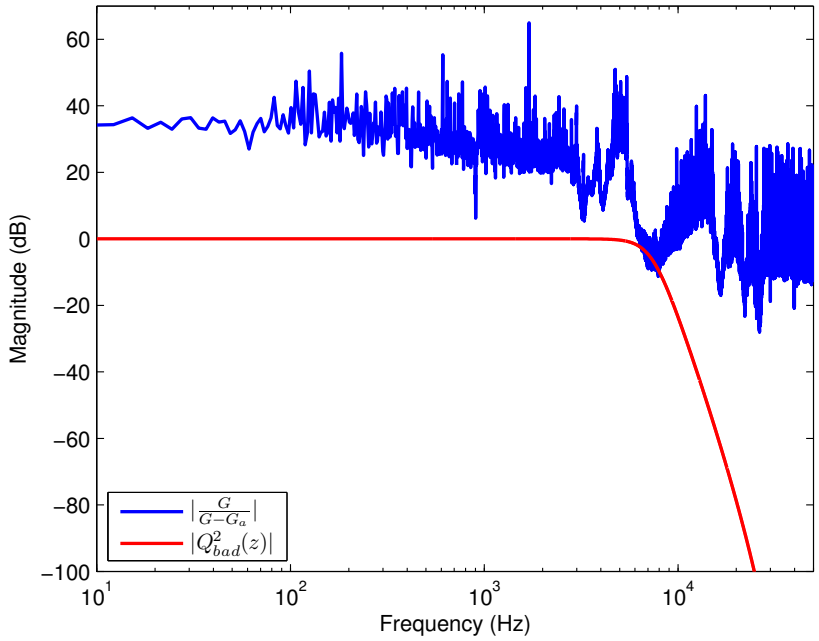

Figure 6.4: Plot of $\left|\frac{G}{G-G_a}\right|$ vs. $|Q^2_{bad}(z)|$, where Q violates robustness condition and then ILC error diverges.

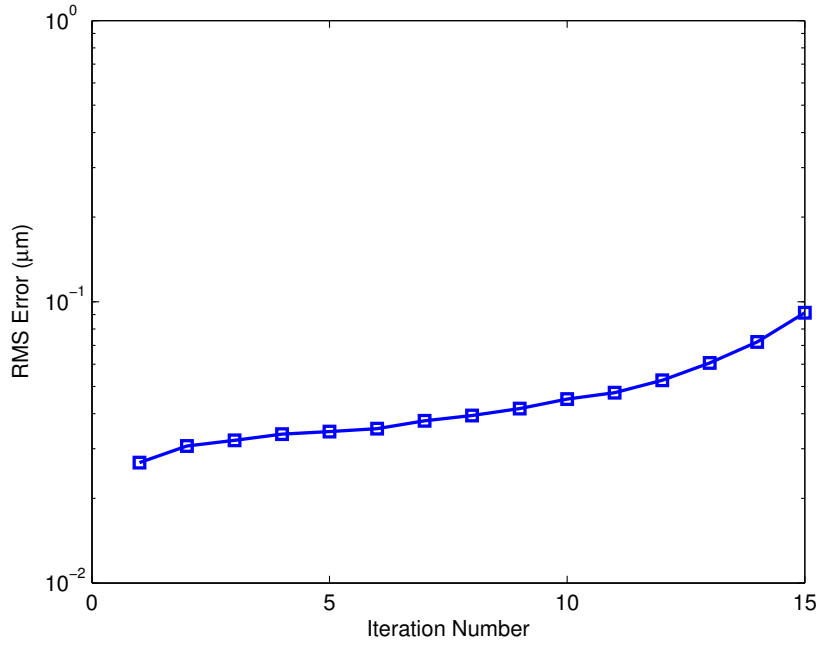

Figure 6.5: Q_1 slightly violates the sufficient stability condition and the ILC diverges.

Unlike the RC, where the *Q*-Filter must be of minimal order to fit on the FPGA,

ILC does not have that restriction due to its off-line computation. Fig. 6.6 shows a 5th order Butterworth low-pass Q-Filter, that satisfies (3.18), results in

$$Q_1(z) = \frac{4.2153 \times 10^{-6} \cdot (1+z^{-1})^7}{(1-0.6763z^{-1})(1-1.39z^{-1}+0.4977z^{-2})} \times \frac{1}{(1-1.507z^{-1}+0.6232z^{-2})(1-1.714z^{-1}+0.847z^{-2})}. \tag{6.1}$$

Notice that $|Q_1(z)|^2$ will dictate the frequency content the ILC will learn and track. As a general rule of thumb, the ILC will not learn any frequencies past the bandwidth of $|Q_1(z)|^2$.

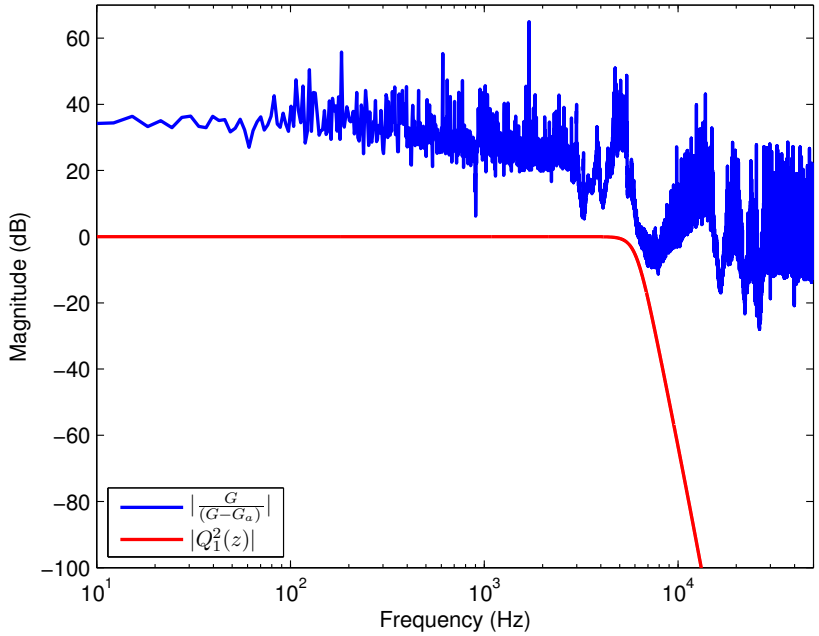

Figure 6.6: Plot of $\left|\frac{G}{G-G_a}\right|$ vs. $|Q_1^2(z)|$, where Q satisfies robustness condition.

6.2 Convergence of ILC

As long as asymptotic convergence is guaranteed in the iteration domain through the Q-Filter, then rate of convergence can be discussed. Convergence rate is highly dependent on the learning function **L**. If **L** is chosen to be an accurate and stable plant inversion, the error can converge in as little as 1 iteration [BTA06]. Since the **L** in this chapter is chosen to be an approximate plant inversion, we can see in Fig. 6.7 how accurate

the model and its inversion are. The results show that the error does not convergence until the 3rd or 4th iteration, meaning that there are still some unmodeled dynamics and plant uncertainties that the ILC adjusts for. For the rest of the paper, we'll use the 5th iteration as the basis of our steady-state error calculations.

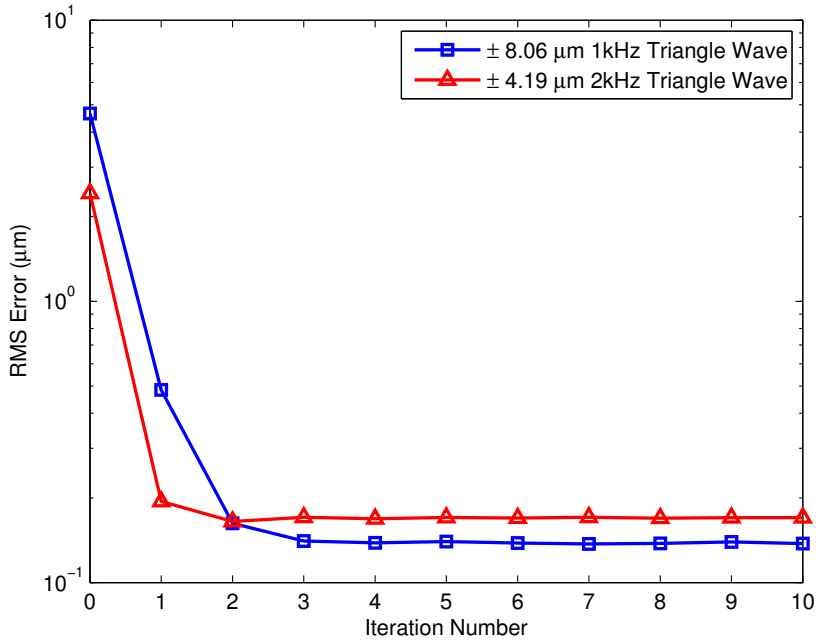

Figure 6.7: RMS error convergence plot for ILC given different reference profiles.

6.3 Example of Iterative Learning Control - Simulation vs. Experimental Results

The example in the previous section showed that ILC converged in less than 5 steps. This quick convergence implies that both the inversion and model of the system is fairly accurate. With a fairly accurate model of the system and inversion, then it means that simulation and experimental results should align fairly closely.

Fig. 6.8 compares simulation and experimental results on the piezoelectric actuator example. The time domain plot shows nearly identical performance for reference tracking performance after the 10^{th} iteration of learning. The power spectral density (PSD)

plot shows that harmonics are learning more or less equally well in both the simulation and experimental case. The major difference is the broadband quantization noise present in the experimental case from the fixed-point PI controller, analog-to-digital convertor, and digital-to-analog converter.

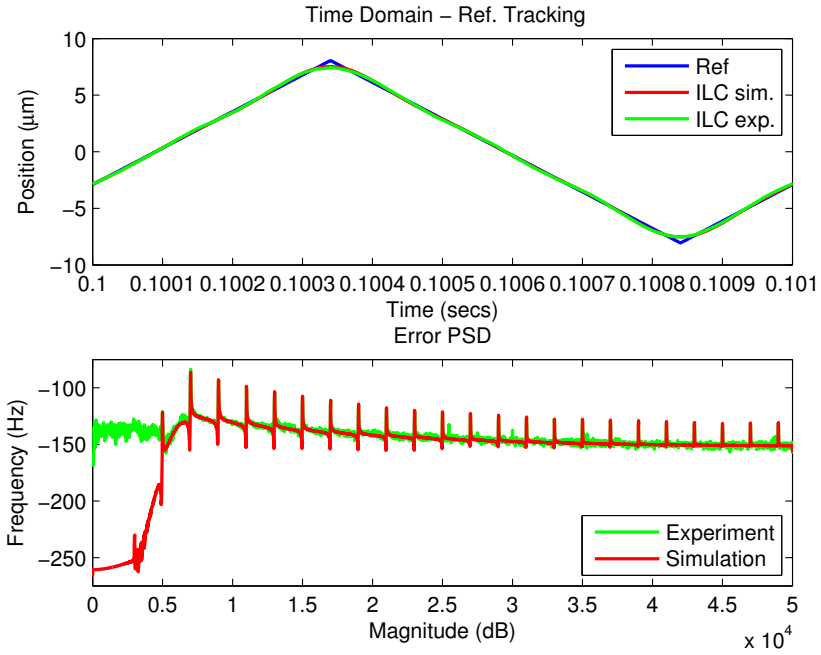

Figure 6.8: Comparing 10^{th} iteration of ILC simulation and experimental results. Time domain and error PSD results used to highlight minor differences in simulation and experimental results.

6.4 Comparison Between Offline Double-Precision FF Control and Double-Precision ILC

Recall that Remark 3.3.4 mentioned that if $\mathbf{Q} = I$ and iteration $j = 1$, that the ILC reduces to the offline double-precision FF case. Table 6.1 compares performance between $\mathbf{Q} = I$ and $\mathbf{Q} = \mathbf{Q_1}$ ($\mathbf{Q_1}$ is based off $Q_1(z)$). $Q_1(z)$ will ignore high frequency content of your desired reference signal. On the other hand, $Q_1(z)$ also serves to reject model uncertainties at high frequencies.

Table 6.1: Comparing of performance between double-precision offline FF and first iteration of the double-precision ILC.

		e_{RMS} (μm)		$\lvert e_{max} \rvert$ (μm)	
	Triangular Magnitude	FF	ILC	FF	ILC
	$0\mu m$	0.0274	0.0358	0.1057	0.1410
$1kHz$	$\pm 0.50\mu m$	0.0298	0.0370	0.1199	0.1505
	$\pm 1.01\mu m$	0.0395	0.0375	0.1156	0.1540
	$\pm 2.01\mu m$	0.0712	0.0480	0.1798	0.1972
	$\pm 4.03\mu m$	0.4654	0.0757	1.1588	0.3339
	$\pm 8.06\mu m$	1.5115	0.1375	3.5562	0.5923
$2kHz$	$\pm 0.52\mu m$	0.0407	0.0395	0.1307	0.1373
	$\pm 1.04\mu m$	0.0774	0.0515	0.2715	0.1890
	$\pm 2.09\mu m$	0.1988	0.0909	0.5078	0.3527
	$\pm 4.19\mu m$	0.3831	0.1702	0.9078	0.6550

6.5 Comparing of Fixed-Point Repetitive Control and Double-Precision ILC

It has been established in Section 3.3 that under certain conditions, ILC and repetitive control are equivalent. This equivalence is useful in determining what type of performance double-precision real-time repetitive control can achieve if there were no hardware limitations. By using the double-precision ILC results, we can determine how close the Delta Operator RC and DFIIt RC can get to double-precision ILC results. Fig. 6.9 and 6.10 serves as an illustrative example of and the performance differences between fixed-point controllers and double-precision floating point controllers. It is interesting to note the performance gains from the Delta Operator realization over the

DFIIt realization. Furthermore, it also illustrates how close the fixed-point Delta Operator is to the floating-point double-precision ILC.

Table 6.2 lists the RMS errors (e_{RMS}), maximum absolute errors ($|e_max|$), and the average of the ten largest absolute errors ($|e_{max,10}|_{avg}$) for multiple controller implementation. $|e_{max,10}|_{avg}$ is used to ensure that $|e_{max}|$ isn't an outlier (e.g. spike from electronic noise). The controllers being compared are the simulated ILC, double-precision experimental ILC, fixed-point experimental Delta RC, and fixed-point experimental DFIIt RC. All three cases were setup for reference tracking of a triangular wave with various amplitudes and frequencies.

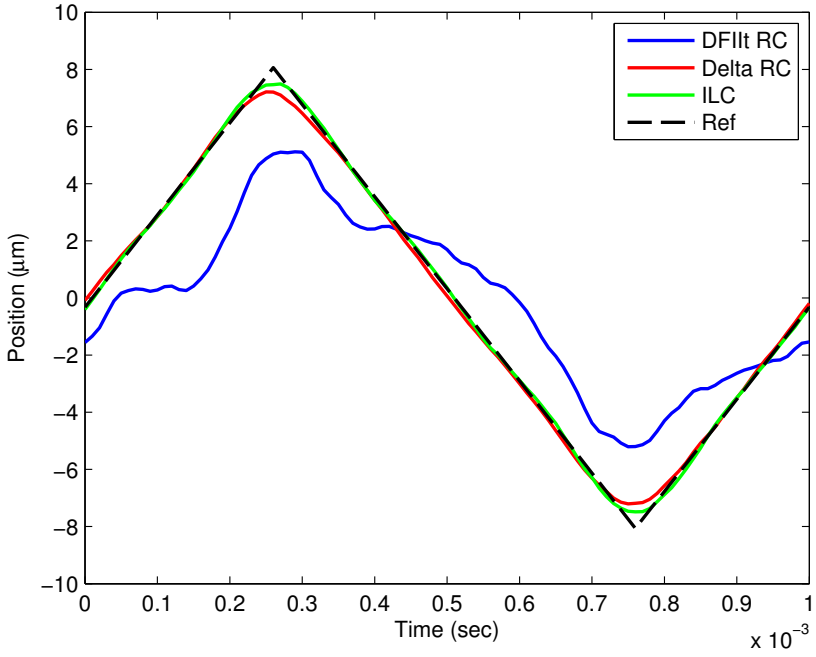

Figure 6.9: Tracking Performance Comparison - ±8.06μm 1kHz Triangular Wave. Results compare performance between double-precision ILC, fixed-point DFIIt RC, and fixed-point Delta-RC.

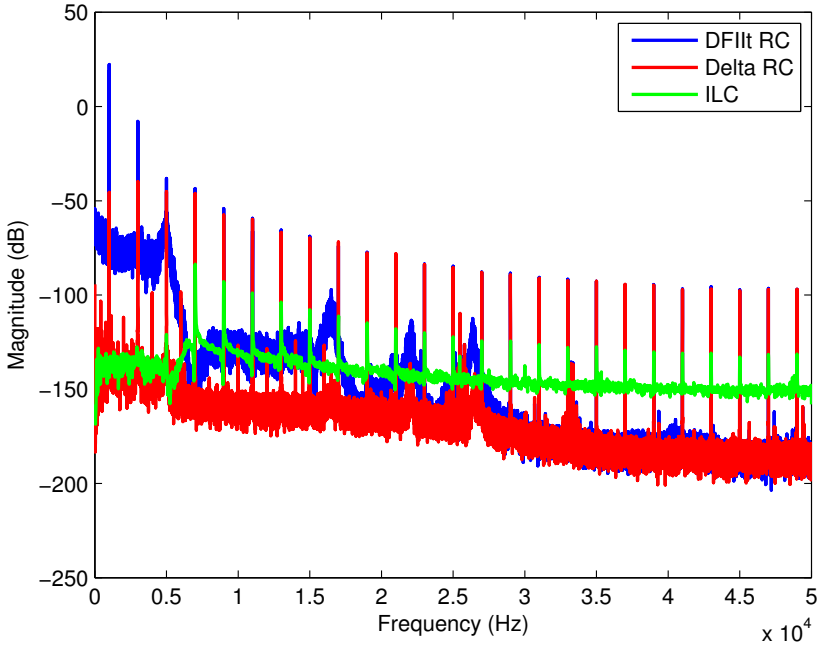

Figure 6.10: Comparison of error PSD for double-precision ILC, fixed-point DFIIt RC, and fixed-point Delta-RC given a $\pm 8.06 \mu m$ 1 kHz triangular reference.

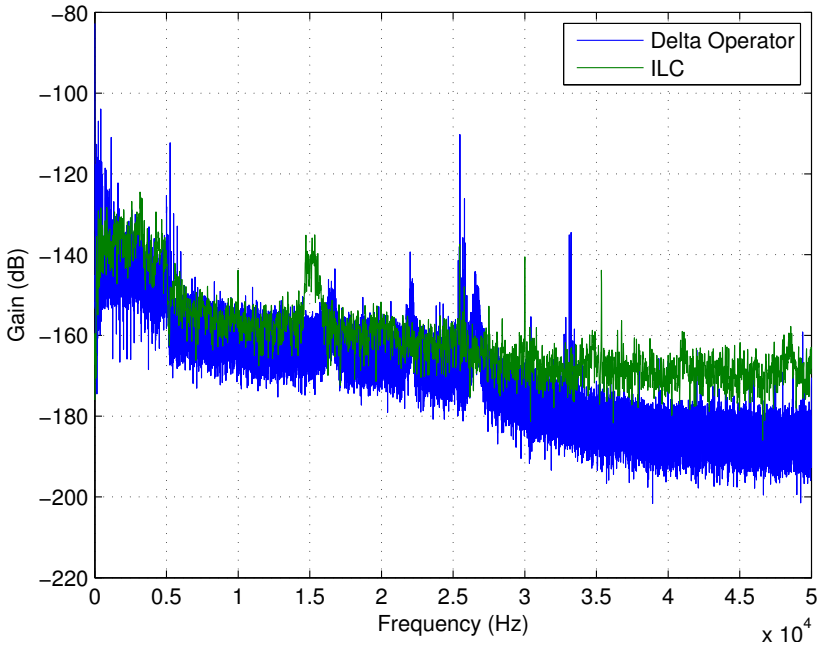

Figure 6.11: Comparison of error PSD for ILC and Delta RC.

Table 6.2: Comparing the simulated ILC (Sim. ILC), the Double-precision ILC (ILC), the fixed-point Delta Operator RC (Delta RC), and the fixed-point DFIIt RC (DFIIt RC) steady-state performance for tracking of a triangular profile.

		e_{RMS} (μm)				$\lvert e_{max}\rvert$ (μm)				$\lvert e_{max,10}\rvert_{avg}$ (μm)			
	Triangular Magnitude	Sim. ILC	ILC	Delta	DFIIt	Sim. ILC	ILC	Delta	DFIIt	Sim. ILC	ILC	Delta	DFIIt
	$0\mu m$	0.0000	0.0358	0.0247	0.0251	0.0000	0.1410	0.1158	1.0675	0.0000	0.1021	0.1089	0.8647
$1kHz$	$\pm 0.50\mu m$	0.0005	0.0370	0.0288	0.0799	0.0020	0.1505	0.1460	2.0847	0.0020	0.1108	0.1328	1.7780
	$\pm 1.01\mu m$	0.0009	0.0375	0.0367	0.0989	0.0040	0.1540	0.2014	2.2609	0.0040	0.1148	0.1872	2.1400
	$\pm 2.01\mu m$	0.0018	0.0480	0.0589	0.3259	0.0081	0.1972	0.3172	3.2025	0.0081	0.1727	0.2980	3.1233
	$\pm 4.03\mu m$	0.0037	0.0757	0.1087	2.4832	0.0161	0.3339	0.5338	7.6437	0.0161	0.3105	0.5223	7.3192
	$\pm 8.06\mu m$	0.0074	0.1375	0.2190	4.5914	0.0323	0.5923	0.9869	11.0728	0.0323	0.5689	0.9723	10.6911
$2kHz$	$\pm 0.52\mu m$	0.0013	0.0395	0.0596	0.0869	0.0044	0.1373	0.1964	2.1350	0.0044	0.1199	0.1872	2.0247
	$\pm 1.04\mu m$	0.0026	0.0909	0.1105	0.2235	0.0088	0.1890	0.2870	3.1068	0.0088	0.1720	0.2806	2.8102
	$\pm 2.09\mu m$	0.0053	0.0515	0.2164	1.7736	0.0176	0.3527	0.4985	6.4806	0.0176	0.3353	0.4820	6.2041
	$\pm 4.19\mu m$	0.0106	0.1702	0.4260	3.0222	0.0353	0.6550	0.8812	8.7616	0.0353	0.6092	0.8652	8.4407

Table 6.2 and Fig. 6.11 show that for small reference magnitudes that the ILC noise floor is actually larger than the Delta Operator RC noise floor. Notice that three types of errors are given to give a complete picture. The simulation-based ILC (Sim. ILC) represents the best case when the model is assumed perfect with no external disturbances present (i.e. error stems only from the ILC). Under larger magnitudes, the experimental results of ILC have smaller error due to the smaller quantization noise of double-precision computation. However, it appears under certain conditions, fixed-point filters under the Delta Operator and DFIIt formulation of RC actually reduce noise at high frequencies. Recall, the LPF q-filter and **Q** used in RC and ILC, respectively. Since the RC was implemented in fixed-point, at a certain magnitude attenuation, any high frequency content in the error signals will enter the quantization floor and will not be amplified by inversion filter $f^{-}(z)$. However for ILC, the quantization floor is significantly lower and thus high frequency content can still be amplified by the inversion filter ("learning function") **L**.

6.6 ILC with Various High Order Q-Filter

Since, the Q-filter in ILC is implemented offline, filter order and the introduction of phase distortion is not an issue since zero-phase techniques are available (e.g. Chapter 2). We can freely design aggressive filter gains without considering phase in order to maximize performance. Instead of the typical LPF, we can use arbitrary magnitudes to get as close to the robustness criterion (3.18) without violating it. Fig. 6.12 shows an aggressive Q-filter, $Q_2(z)$, was designed as a Chebyshev 1 LPF cascaded with a Chebyshev 2 notch filter resulting in

$$Q_2(z) = \left(\frac{0.1913 - 0.02634z^{-1} + 0.2526z^{-2} - 0.02634z^{-3} + 0.1913z^{-4}}{1 - 1.301z^{-1} + 1.132z^{-2} - 0.3415z^{-3} + 0.09317z^{-4}}\right)$$
$$\times \left(\frac{0.7199 - 3.7641z^{-1} + 8.7202z^{-2} - 11.3400z^{-3} + 8.7202z^{-4} - 3.7641z^{-5} + 0.7199z^{-6}}{1.0000 - 4.6715z^{-1} + 9.6639z^{-2} - 11.2295z^{-3} + 7.7132z^{-4} - 2.9668z^{-5} + 0.5030z^{-6}}\right) \tag{6.2}$$

In addition to tracking the bandwidth of $Q_1(z)$, $Q_2(z)$ now has the added advantage of tracking frequencies in the 10-11kHz range. Table 6.3 shows how $Q_1(z)$ and $Q_2(z)$ affects the errors differently given various frequencies and amplitudes. Fig. 6.13 compares the performance differences when $Q_1(z)$ or $Q_2(z)$ is used. For triangular waves, typically the 1^{st}, 3^{rd} and 5^{th} harmonics are the dominant frequency components. For a 1kHz triangular wave, $Q_1(z)$ and $Q_2(z)$ pass the 1, 3 and 5 kHz harmonics which explains the minor improvement for $Q_2(z)$. For the 2kHz triangular wave, $Q_1(z)$ passes the 2kHz and parts of the 6kHz frequency but $Q_2(z)$ passes the 2, 6, and 10kHz frequencies. This explains the substantial error reduction for the 2kHz triangular wave. Similarly, the 3.5kHz triangular wave also sees an error reduction since $Q_1(z)$ passes only the 3.5kHz frequency and $Q_2(z)$ passes the 3.5 and 10.5kHz frequency components.

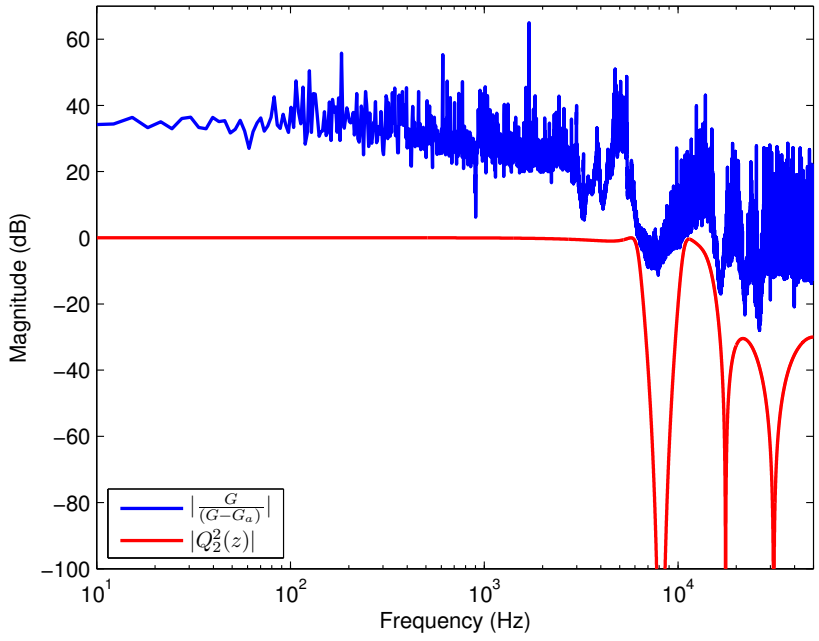

Figure 6.12: Plot of $|\frac{G}{G-G_a}|$ vs. $|Q_2^2(z)|$, where Q satisfies robustness condition.

Table 6.3: Sim. ILC, $Q_1(z)$ ILC, $Q_2(z)$ Delta-RC steady-state performance for tracking of a triangular profile.

		e_{RMS} (μm)			$\lvert e_{max} \rvert$ (μm)			$\lvert e_{max,10} \rvert_{avg}$ (μm)		
	Triangular Magnitude	Sim. ILC	$Q_2(z)$ ILC	$Q_1(z)$ ILC	Sim. ILC	$Q_2(z)$ ILC	$Q_1(z)$ Delta	Sim. ILC	$Q_2(z)$ ILC	$Q_1(z)$ Delta
	$0\mu m$	0.0000	0.0312	0.0358	0.0000	0.0957	0.1410	0.0000	0.0838	0.1021
$1kHz$	$\pm 0.50\mu m$	0.0005	0.0320	0.0370	0.0020	0.1072	0.1505	0.0020	0.0890	0.1108
	$\pm 1.01\mu m$	0.0009	0.0360	0.0375	0.0040	0.1187	0.1540	0.0040	0.1069	0.1148
	$\pm 2.01\mu m$	0.0018	0.0456	0.0480	0.0081	0.2274	0.1972	0.0081	0.1643	0.1727
	$\pm 4.03\mu m$	0.0037	0.0701	0.0757	0.0161	0.3138	0.3339	0.0161	0.2886	0.3105
	$\pm 8.06\mu m$	0.0074	0.1302	0.1375	0.0323	0.5872	0.5923	0.0323	0.5602	0.5689
$2kHz$	$\pm 0.52\mu m$	0.0013	0.0336	0.0395	0.0044	0.1222	0.1373	0.0044	0.1027	0.1199
	$\pm 1.04\mu m$	0.0026	0.0441	0.0909	0.0088	0.2091	0.1890	0.0088	0.1663	0.1720
	$\pm 2.09\mu m$	0.0053	0.0677	0.0515	0.0176	0.3024	0.3527	0.0176	0.2749	0.3353
	$\pm 4.19\mu m$	0.0106	0.1255	0.1702	0.0353	0.5342	0.6550	0.0353	0.4976	0.6092

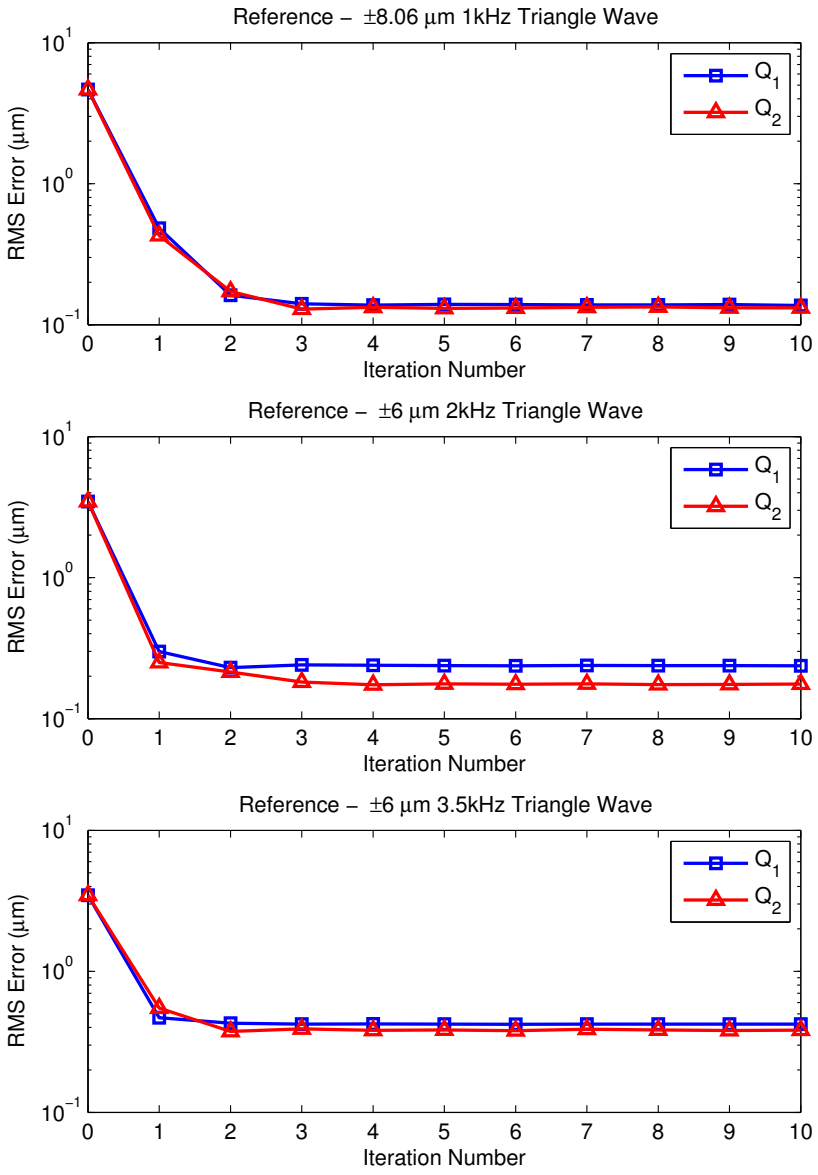

Figure 6.13: Comparison of error convergence using ILC for different Q-filters.

CHAPTER 7

Conclusion

This dissertation has brought together techniques from the signal processing community and controls community to address the need for high performance controllers at high sampling rates. It has addressed practical concerns of limited computational resources by introducing multi-rate signal processing techniques to reduce the multipliers and equivalent delay in the linear-phase Powell-Chau/Kurosu filters. A modified Kurosu filter produced an efficient yet high performing real-time non-minimum phase inversion feedforward controller. This Kurosu filter-based feedforward inversion resulted in a modified repetitive controller structure for high sampling rate fixed-point applications. Experimental results for this new repetitive controller structure were performed on a levitated shaft and piezoelectric actuator to demonstrate the effectiveness of these controllers and to also highlight the properties of fixed-point controllers. Experimental results on the piezoelectric actuators served as an analog for atomic force microscope applications. Delta Operator realizations were incorporated to reduce quantization noise and to ameliorate filter coefficient quantization effects on the frequency response. Experimental results on a piezoelectric actuator showed how the Delta Operator substantially improved performance with minor additional computational costs. To complete the study, double-precision ILC was used to serve as a practical equivalent to high sampling rate double-precision repetitive control if and when the hardware capabilities becomes affordably available. Given practical limitations on present day hardware that controllers can be implemented on, this dissertation has provided possible controller structures that are not only efficient but also high performing.

REFERENCES

[ACM07] Hyo-Sung Ahn, YangQuan Chen, and K.L. Moore. "Iterative Learning Control: Brief Survey and Categorization." *IEEE Trans. Syst., Man, Cybern. C*, **37**(6):1099–1121, Nov. 2007.

[AO94] Notker Amann and David H. Owens. "Non-minimum phase plants in iterative learning control." In *International Conference on Intelligent Systems Engineering, 1994*, pp. 107 –112, Sep. 1994.

[AOR96] Notker Amann, David H. Owens, Eric Rogers, and Anja Wahl. "An H8 approach to linear iterative learning control design." *International Journal of Adaptive Control and Signal Processing*, **10**(6):767–781, 1996.

[AT95] D.M. Alter and Tsu-Chin Tsao. "Optimal feedforward tracking control of linear motors for machine tool drives." In *Proc. of the 1995 American Control Conference*, volume 1, pp. 210–214, Jun. 1995.

[BTA06] Douglas A. Bristow, Marina Tharayil, and Andrew G. Alleyne. "A survey of iterative learning control." *IEEE Control Syst. Mag.*, **26**(3):96–114, Jun. 2006.

[CC07] Hung-Ming Cheng and G.T.-C. Chiu. "Coupling between Sample Rate and Required Wordlength for Finite Precision Controller Implementation with Delta Transform." In *Proc. of the 2007 American Control Conference*, pp. 3588–3593, Jul. 2007.

[CFR07] Zhonglun Cai, C. Freeman, E. Rogers, and P. Lewin. "Reference Shift Iterative Learning Control for a Non-minimum Phase Plant." In *Proc. of the 2007 American Control Conference*, pp. 558–563, Jul. 2007.

[CKK08] P. Cortes, M.P. Kazmierkowski, R.M. Kennel, D.E. Quevedo, and J. Rodriguez. "Predictive Control in Power Electronics and Drives." *IEEE Trans. Ind. Electron.*, **55**(12):4312–4324, Dec. 2008.

[CT10a] Herrick Lin Chang and Tsu-Chin Tsao. "Efficient Fixed-Point Realization of Approximate Dynamic Inversion Compensators for Non-Minimum Phase Systems." In *Proc. of the 2010 American Control Conference*, pp. 4193–4198, Jun. 2010.

[CT10b] Herrick Lin Chang and Tsu-Chin Tsao. "Repetitive Control of a Levitated Shaft - FPGA Implementation Based on Powell-Chau Filters." In *Proc. of 2010 International Symposium on Flexible Automation*, Jul. 2010.

[CWW10] Kevin Chu, Yigang Wang, Jason Wilson, Chi-Ying Lin, and Tsu-Chin Tsao. "Modelling and Control of a Magnetic Bearing System." In *Proc. of the 2010 American Control Conference*, Jun. 2010.

[DPL98] B. Djokic, M. Popovic, and M. Lutovac. "A new improvement to the Powell and Chau linear phase IIR filters." *IEEE Trans. Signal Process.*, **46**(6):1685–1688, Jun. 1998.

[EML07] Gerardo Escobar, P. R. Martinez, and J. Leyva-Ramos. "Analog Circuits to Implement Repetitive Controllers With Feedforward for Harmonic Compensation." *IEEE Trans. Ind. Electron.*, **54**(1):567–573, Feb. 2007.

[FLR05] C. T. Freeman, P. L. Lewin, and E. Rogers. "Experimental evaluation of iterative learning control algorithms for non-minimum phase plants." *International Journal of Control*, **78**(11):826–846, 2005.

[FPW97] Gene F. Franklin, David J. Powell, and Michael L. Workman. *Digital Control of Dynamic Systems (3rd Edition)*. Prentice-Hall, Dec. 1997.

[GP99] Jayati Ghosh and Brad Paden. "Iterative learning control for nonlinear nonminimum phase plants with input disturbances." In *Proc. of the 1999 American Control Conference*, volume 4, pp. 2584–2589, 1999.

[GP01] Jayati Ghosh and Brad Paden. "Iterative Learning Control for Nonlinear Nonminimum Phase Plants." *Journal of Dynamic Systems, Measurement, and Control*, **123**(1):21–30, 2001.

[Gus96] F. Gustafsson. "Determining the initial states in forward-backward filtering." *IEEE Trans. Signal Process.*, **44**(4):988–992, Apr. 1996.

[HRE09] M. Hartmann, S.D. Round, H. Ertl, and J.W. Kolar. "Digital Current Controller for a 1 MHz, 10 kW Three-Phase VIENNA Rectifier." *IEEE Trans. Power Electron.*, **24**(11):2496 –2508, Nov. 2009.

[HT98] R. D. Hanson and Tsu-Chin Tsao. "Reducing Cutting Force Induced Bore Cylindricity Errors by Learning Control and Variable Depth of Cut Machining." *Journal of Manufacturing Science and Engineering*, **120**(3):547–554, 1998.

[JPK08] H. Janocha, D. Pesotski, and K. Kuhnen. "FPGA-Based Compensator of Hysteretic Actuator Nonlinearities for Highly Dynamic Applications." *IEEE/ASME Trans. Mechatronics*, **13**(1):112–116, Feb. 2008.

[JW97] Zhongnong Jiang and Alan N Willson, Jr. "Efficient digital filtering architectures using pipelining/interleaving." *IEEE Trans. Circuits Syst. II*, **44**(2):110–119, Feb. 1997.

[KJ74] J. Kormylo and V. Jain. "Two-pass recursive digital filter with zero phase shift." *IEEE Trans. Acoust., Speech, Signal Process.*, **22**(5):384–387, Oct. 1974.

[KL12] B.J. Kenton and K.K. Leang. "Design and Control of a Three-Axis Serial-Kinematic High-Bandwidth Nanopositioner." *IEEE/ASME Trans. Mechatronics*, **17**(2):356–369, Apr. 2012.

[KLH98] J. Kauraniemi, T.I. Laakso, I. Hartimo, and S.J. Ovaska. "Delta operator realizations of direct-form IIR filters." *IEEE Trans. Circuits Syst. II*, **45**(1):41–52, January 1998.

[KMT03] A. Kurosu, S. Miyase, S. Tomiyama, and T. Takebe. "A technique to truncate IIR filter impulse response and its application to real-time implementation of linear-phase IIR filters." *IEEE Trans. Signal Process.*, **51**(5):1284–1292, May 2003.

[KT10] K. Kalyanam and Tsu-Chin Tsao. "Experimental Study of Adaptive - Control for Disk Drive Track-Following Servo Problem." *IEEE/ASME Trans. Mechatronics*, **15**(3):480–491, Jun. 2010.

[LD07] K.K. Leang and S. Devasia. "Feedback-Linearized Inverse Feedforward for Creep, Hysteresis, and Vibration Compensation in AFM Piezoactuators." *IEEE Trans. Control Syst. Technol.*, **15**(5):927–935, Sept. 2007.

[LDW11] Hui Li, Chunling Du, and Youyi Wang. "Optimal Reset Control for a Dual-Stage Actuator System in HDDs." *IEEE/ASME Trans. Mechatronics*, **16**(3):480–488, Jun. 2011.

[Lin08] Chi-Ying Lin. *Adaptive and Repetitive Control of a Fast Tool Servo for Precision Motion Control*. PhD dissertation, University of California, Los Angeles, Los Angeles, CA, USA, 2008.

[LKC10] Choong Woo Lee, Bong Sik Kwak, Chung Choo Chung, and M. Tomizuka. "Design of the Tracking Controller for Holographic Digital Data Storage." *IEEE/ASME Trans. Mechatronics*, **15**(2):242–252, Apr. 2010.

[Lon00] Richard W. Longman. "Iterative learning control and repetitive control for engineering practice." *International Journal of Control*, **73**(10):930–954, 2000.

[MC07] E. Monmasson and M.N. Cirstea. "FPGA Design Methodology for Industrial Control Systems-A Review." *IEEE Trans. Ind. Electron.*, **54**(4):1824–1842, Aug. 2007.

[MD11] Aung Myaing and V. Dinavahi. "FPGA-Based Real-Time Emulation of Power Electronic Systems With Detailed Representation of Device Characteristics." *IEEE Trans. Ind. Electron.*, **58**(1):358 –368, Jan. 2011.

[MG86] R. Middleton and G. Goodwin. "Improved finite word length characteristics in digital control using delta operators." *IEEE Trans. Autom. Control*, **31**(11):1015–1021, Nov. 1986.

[MHN01] O. Markusson, H. Hjalmarsson, and M. Norrlof. "Iterative learning control of nonlinear non-minimum phase systems and its application to system and model inversion." In *Proceedings of the 2001 IEEE Conference on Decision and Control*, volume 5, pp. 4481–4482, 2001.

[Mit04] Sanjit K Mitra. *Digital Signal Processing: A Computer-Based Approach*. McGraw-Hill, 2004.

[MTT00] Syoichiro Miyase, Shigenori Tomiyama, and Tsuyoshi Takebe. "Sample delay reduction of linear phase IIR filters by shortening section length of signals." *Electronics and Communications in Japan (Part III: Fundamental Electronic Science)*, **83**(3):1–11, 2000.

[MZE04] D. Maksimovic, R. Zane, and R. Erickson. "Impact of digital control in power electronics." In *Proceeding of the 16th International Symposium on Power Semiconductor Devices and ICs*, pp. 13–22, May 2004.

[NG02] Mikael Norrlof and Svante Gunnarsson. "Time and frequency domain convergence properties in iterative learning control." *International Journal of Control*, **75**(14):1114–1126, 2002.

[PB87] TS Parks and CS Burrus. *Digital Filter Design*. Wiley, New York, 1987.

[PBA07] L.Y. Pao, J.A. Butterworth, and D.Y. Abramovitch. "Combined Feedforward/Feedback Control of Atomic Force Microscopes." In *Proc. of the 2007 American Control Conference*, pp. 3509–3515, Jul. 2007.

[PC91] S.R. Powell and P.M. Chau. "A technique for realizing linear phase IIR filters." *IEEE Trans. Signal Process.*, **39**(11):2425–2435, Nov. 1991.

[RTH94] J.D. Rasmussen, T.-C. Tsao, R.D. Hanson, and S.G. Kapoor. "Dynamic variable depth of cut machining using piezoelectric actuators." *International Journal of Machine Tools and Manufacture*, **34**(3):379–392, 1994.

[SL98] A. Simic and M. Lutovac. "New realizations of second-order all-pass transfer functions." In *9th European Signal Processing Conference*, pp. 295–298, Sep. 1998.

[Sog02] Takuya Sogo. "Stable inversion for nonminimum phase sampled-data systems and its relation with the continuous-time counterpart." In *Proceedings of the 2002 IEEE Conference on Decision and Control*, volume 4, pp. 3730–3735, Dec. 2002.

[SP08] G. Schitter and Nghi Phan. "Field Programmable Analog Array (FPAA) based control of an Atomic Force Microscope." In *Proc. of the 2008 American Control Conference*, pp. 2690–2695, Jun. 2008.

[TCR87] M. Tomizuka, M. S. Chen, S. Renn, and T. C. Tsao. "Tool Positioning for Noncircular Cutting With Lathe." *Journal of Dynamic Systems, Measurement, and Control*, **109**(2):176–179, 1987.

[Tom87] Masayoshi Tomizuka. "Zero Phase Error Tracking Algorithm for Digital Control." *Journal of Dynamic Systems, Measurement, and Control*, **109**(1):65–68, 1987.

[TQN00] Tsu-Chin Tsao, Yao-Xin Qian, and Mahadevamurty Nemani. "Repetitive Control for Asymptotic Tracking of Periodic Signals With an Unknown Period." *Journal of Dynamic Systems, Measurement, and Control*, **122**(2):364–369, 2000.

[Tsa94] Tsu-Chin Tsao. "Optimal Feed-Forward Digital Tracking Controller Design." *Journal of Dynamic Systems, Measurement, and Control*, **116**(4):583–592, 1994.

[TSH08] Tsu-Chin Tsao, Zongxuan Sun, Reed D. Hanson, and Alexanda Babinski. "Design, Modeling, and Motion Control of the Noncircular Turning Process for Camshaft Machining." *Journal of Dynamic Systems, Measurement, and Control*, **130**(3):031005, 2008.

[TT94] Tsu-Chin Tsao and Masayoshi Tomizuka. "Robust Adaptive and Repetitive Digital Tracking Control and Application to a Hydraulic Servo for Noncircular Machining." *Journal of Dynamic Systems, Measurement, and Control*, **116**(1):24–32, 1994.

[TTC89] Masayoshi Tomizuka, Tsu-Chin Tsao, and Kok-Kia Chew. "Analysis and Synthesis of Discrete-Time Repetitive Controllers." *Journal of Dynamic Systems, Measurement, and Control*, **111**(3):353–358, 1989.

[Vai93] P.P. Vaidyanathan. *Multirate Systems and Filter Banks*. Englewood Cliffs, NJ: Prentice-Hall, 1993.

[VMN86] P. Vaidyanathan, S. Mitra, and Y. Neuvo. "A new approach to the realization of low-sensitivity IIR digital filters." *IEEE Trans. Acoust., Speech, Signal Process.*, **34**(2):350–361, Apr. 1986.

[WGI09] Youqing Wang, Furong Gao, and Francis J. Doyle III. "Survey on iterative learning control, repetitive control, and run-to-run control." *Journal of Process Control*, **19**(10):1589–1600, 2009.

[WO94] Alan N Willson, Jr. and H J Orchard. "An improvement to the Powell and Chau linear phase IIR filters." *IEEE Trans. Signal Process.*, **42**(10):2842–2848, Oct. 1994.

[WPX10] X.H. Wu, S.K. Panda, and J.X. Xu. "Design of a Plug-In Repetitive Control Scheme for Eliminating Supply-Side Current Harmonics of Three-Phase PWM Boost Rectifiers Under Generalized Supply Voltage Conditions." *IEEE Trans. Power Electron.*, **25**(7):1800 –1810, Jul. 2010.

[WWC11] Jianyi Wang, Youyi Wang, and Shuyu Cao. "Add-On Feedforward Compensation for Vibration Rejection in HDD." *IEEE/ASME Trans. Mechatronics*, **16**(6):1164–1170, Dec. 2011.

[WWZ05] Yigang Wang, Danwei Wang, Bin Zhang, and Yongqiang Ye. "From iterative learning control to robust repetitive learning control." In *Proceedings of 2005 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pp. 969–974, Jul. 2005.

[YBR12] Y. K. Yong, B. Bhikkaji, and S. O. Reza Moheimani. "Design, Modeling, and FPAA-Based Control of a High-Speed Atomic Force Microscope Nanopositioner." *IEEE/ASME Trans. Mechatronics*, **PP**(99):1–12, 2012.

[YW99] Fengqi Yu and Alan N Willson, Jr. "Hardware efficient architectures for coupled-form IIR filters." In *Proc. of the 1999 IEEE Int. Symp. on Circuits and Sys.*, volume 3, pp. 355–358, Jul. 1999.

[ZW01] Keliang Zhou and Danwei Wang. "Digital repetitive learning controller for three-phase CVCF PWM inverter." *IEEE Trans. Ind. Electron.*, **48**(4):820–830, Aug. 2001.