Design, Simulation and Implementation of High Precision Control Algorithms for a Galvanometer Laser Scanner

by

Tomas Torres Bonet

BEng, University of Victoria, 2011

A Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of
Master of Applied Science
in the Department of Electrical and Computer Engineering

Supervisory Committee

Design, Simulation and Implementation of High Precision Control Algorithms for a Galvanometer Laser Scanner

by

Tomas Torres Bonet

BEng, University of Victoria, 2011

Supervisory Committee

Dr. Pan Agathoklis, Supervisor
Department of Electrical and Computer Engineering

Dr. Daler Rachmatov, Departmental Member
Department of Electrical and Computer Engineering

Supervisory Committee

Dr. Pan Agathoklis, Supervisor
Department of Electrical and Computer Engineering

Dr. Daler Rachmatov, Departmental Member
Department of Electrical and Computer Engineering

Abstract

This thesis focuses on the theory, design, simulation and implementation of several digital controllers for periodic signals on a laser scanning galvanometer. A model for the galvanometer was obtained and verified using closed loop identification techniques. Using this model, controllers were designed and simulated using MATLAB and then implemented on a custom FPGA control processor with a focus on tracking performance. The types of controllers used were: an Iterative Learning Controller, an RST pole placement controller, an Adaptive Feed-forward cancellation controller, a combined Iterative Learning and Adaptive Feed-forward cancellation controller and a simple PID controller.

The simulated results were better than the experimental results because of system noise and modelling uncertainties but the relative performance between each of the controllers was similar for both the simulation and experimental setup. The experimental results achieved were very good with one controller reaching errors under 50μrad.

Contents

List of Tables

List of Figures

Table of Notations

C, L, P, Q	LTI Discrete transfer function in terms of z^{-1}
C_r	Crest factor
e_k	Error at current iteration
$e[k], r[k], y[k]$	Discrete signals at sample number k
$\theta(s), i(s)$	LTI Continuous transfer function in terms of s
n_P, n_S, n_R	Order of P, S and R polynomials
r, y	Reference and output of system
u	Control output and into plant
u_k	Control output at current iteration

Table of Acronyms

AFC	Adaptive Feed-forward cancellation controller
CNC	Computer Numerical Control
CPU	Central Processing Unit
FFT	Fast Fourier Transform
FPGA	Field Programmable Gate Array
HDD	Hard Disk Drive
ID	Identification
LTI	Linear Time-Invariant
PRBS	Pseudo Random Binary Sequence
PID	Type of controller with Proportional, Integral and Derivative terms
PD	Discrete controller with Proportional and Derivative terms
RMS	Root Mean Square
RST	Type of discrete controller with R, S and T polynomials in terms of z^{-1}
SPU	Servo Processing Unit. A custom CPU implemented on an FPGA
XCLOE	Extended Closed Loop Output Error identification technique

Acknowledgements

I would like to thank Mile Erlic, Tom Erlic and the whole team at Precision Micro Dynamics for supplying me with an office to work in, the Polaris controller, amplifiers and galvanometers and for their guidance with any problems or questions I had along the way. Without their help, this work would not have been possible.

I would also like to thank my supervisor Dr. Pan Agathoklis for his ongoing support, feedback and guidance while working on my research and writing my thesis.

Chapter 1

Introduction

The high precision Computer Numerical Control (CNC) machining industry is a large sector consisting of many micrometer machining applications such as semi-conductor manufacturing, PCB stencil cutting, and laser ablation. The main requirements of these industries consist of high throughput and tight machining tolerances. The demand for electronic devices has been continuously increasing and their size has been continuously decreasing. Manufacturers have had to adopt machining techniques that allow them to achieve very strict tolerances and very high throughput rates to keep up with demand. One way to achieve this is to use a two axis galvanometer laser system because of their high accuracy and bandwidth [2]. The galvanometer system can then be controlled with high precision motion control algorithms.

A galvanometer scanning system is comprised of two galvanometers with a mirror attached to each one. They can be mounted on a stationary beam or onto another moving actuator depending on the application. A laser beam is then shone at the two mirrors and they work in unison to ablate, cut or burn the material at a specific X,Y location on the cutting surface. Each galvanometer works as an independent control system and the coordination is performed by the trajectory generator. Typically the trajectories used are periodic since many identical parts or cuts must be made. It is the job of the control algorithm to ensure the galvanometers track the trajectory as close as possible.

1.1 Control System Basics

A control system is comprised of a control algorithm plus a system that must be controlled, normally called the plant. In motion control, the plant can be a wide range of actuators such as linear and rotary motors or galvanometers. The control algorithm is responsible for making the plant perform what the user intends, such as following a specified trajectory. One example of a control system used everyday is a car and its driver. The driver (control algorithm) adjusts the pressure on the accelerator or brake to maintain a constant speed.

The control algorithm is responsible for rejecting any disturbances acting on the system, reducing steady state error and providing a good transient response. A good transient response allows the system to respond quickly to any sharp changes in the reference signal and also keep the overshoot to a minimum when this happens.

There are many different types of control systems used in practice today. They range from learning controllers which can improve their performance iteratively, to complex frequency domain based controllers to simple ones which can be hand tuned without any analysis. This thesis covers just a few of the many possible controllers.

1.2 Contribution and Outline of the Thesis

The goal of this thesis is to analyze and compare the performance of several different control algorithms when used with a galvanometer scanner. The results of this thesis also demonstrate the importance of accurate models for certain controllers and the difficulty of getting a model for galvanometers which can be used for μrad error tolerances.

This work can help improve the performance of CNC machining systems by demonstrating the strengths and weaknesses of the different control algorithms researched. Machine manufacturers are then able to select the appropriate control algorithm that suits their application.

Many different algorithms exist but this work will cover four main ones. These are: Iterative Learning Control (ILC), Pole placement using an RST structure, Adaptive Feed-Forward Cancellation (AFC) and finally the most widely used control in industry the PID controller is covered.

Each of these motion control algorithms have different applications and work in different situations. Iterative Learning control can be used when tracking a periodic trajectory. RST control gives the designer lots of flexibility on the desired performances of the system. AFC can be used to remove sinusoidal errors from the system or to improve performance when tracking sinusoidal trajectories and the PID control system can often achieve very good results with minimal design work and modelling.

Chapter 2 introduces the ILC and AFC controllers which specialize in following periodic trajectories, the RST controller which is a polynomial pole placement controller and finally the PID controller is described. The design process and stability requirements of each controller are also outlined in this chapter.

Chapter 3 begins with general modelling and also explains the modelling performed on a galvanometer actuator. This chapter also introduces the trajectories that will be used when simulating the system. Then the design process and simulation results on the generated model will be described.

Chapter 4 covers the experimental setup, implementation and results of the different algorithms when applied to a galvanometer scanner. The results are then compared with the simulations and against each other.

Finally Chapter 5 outlines some conclusions and possibilities for future work.

Notes on Notation Used

In the following Chapters when referring to an IIR or FIR digital filter transfer function in terms of the delay term z^{-1} such as $L(z^{-1})$ or $Q(z^{-1})$ the (z^{-1}), term has been dropped to improve the readability of the figures and equations. In the case of an FIR filter the transfer functions simplify to polynomials.

If a capital letter is used as a gain, it will explicitly mention it, and any continuous time transfer functions will include the (s) term.

Chapter 2

Overview of Algorithms for Control Design

This chapter introduces the theory of the different motion control algorithms used in this work. First an overview of the ILC algorithm is shown, then the RST controller, followed by an overview of Adaptive Feed-forward cancellation and finally PID control is introduced.

2.1 ILC Overview

Iterative Learning Control is a popular control technique used for tracking periodic reference trajectories. It is a widely researched control technique in robotics, semiconductor fabrication and HDD disks, where high accuracy tracking of periodic trajectories is very important [3][4]. Iterative learning control aims to give machines the same general thought process humans have when throwing a ball. For example we are able to adjust our power and aim after each throw of the ball based on where it landed on the previous throw, this process is repeated until we hit our mark.

The first mention of learning control was by Uchiyama in 1978 in Japanese but the first iterative learning control paper published in English was Arimoto et al's paper entitled "Bettering of operations through learning" in 1984 [5]. It showed that the tracking performance of robot manipulators could be improved by learning. The previous operation data could be used to improve the performance of the current operation.

The control action is improved upon on each iteration of the periodic reference and each iteration is independent of the other ones. The system always begins at the same initial state at each iteration. This shares a similar goal with repetitive control which also deals with periodic reference signals. The main difference is that in repetitive control each trial is directly linked to the previous one.

2.1.1 Types of ILC Algorithms

Iterative learning control analysis can be classified into two general categories; Standard and Lifted System ILC. Lifted system ILC was first introduced by Phan in 1988 [6]. The performance of standard versus lifted system ILC was compared by I. Rotariu et al in [7]. In lifted ILC the input and output of a plant are considered finite vectors which span one period and the plant is described as a matrix which relates the input and output in a finite time interval [8]. In standard ILC the plant inputs and outputs are considered to be of infinite time duration and are analyzed using transfer functions. In [7] they implemented a standard ILC and a lifted ILC algorithm on a semiconductor wafer machine and found that either method resulted in similar tracking errors. However the standard ILC algorithm converged in less iterations and was easier to implement since it used a transfer function description of the system; therefore the standard ILC algorithm was used in this work.

2.1.2 Standard ILC: D-Type and PID-Type

In Figure 2.1 the PID type ILC controller is shown. Here the linear time invariant (LTI) system $P(z^{-1})$ is commanded to follow a periodic trajectory, which would reset to the original state after each period. The goal of all the ILC algorithms is to iteratively produce the ideal control signal u to achieve perfect tracking of the reference.

The derivative learning equation shown below in Equation 2.1 was introduced in [5]. The next control term is based on the current trials control plus the derivative of the current error multiplied by the learning gain L.

$$u_{k+1} = u_k + L\nabla e_k \tag{2.1}$$

Arimoto also introduced the more general PID type ILC updating algorithm.

$$u_{k+1} = u_k + \Phi e_k + L\nabla e_k + \Psi \sum e_k, \tag{2.2}$$

where ∇ and $\sum$ correspond to the backwards difference operator and a discrete integration operation respectively.

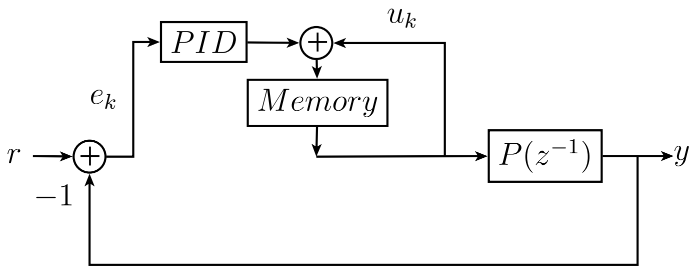

Figure 2.1: PID ILC Control Structure

2.1.3 General Standard ILC

There are many different forms and structures of standard ILC. Two common implementations are shown in Figures 2.2 and 2.3 below. The feedback controller C is an existing controller such as a PID and the ILC is added to the existing system to work in conjunction with the controller C. The parameters L and Q can be implemented as simple gains which can be hand tuned or as complex digital filters designed to minimize the tracking error. The D-Type and PID-Type updating algorithms mentioned in Section 2.1.2 can be easily implemented using the two general structures in Figures 2.2 and 2.3.

The simplest form of an ILC controller is one without a stabilizing controller C. The update law of the control signal for this form of ILC is shown below:

$$u_{k+1} = Qu_k + Le_k \tag{2.3}$$

The goal is to find the two gains or polynomials L and Q such that the control signal u_k converges to a fixed point u^*, that is to say $u^*_{k+1} = u^*_k$. In [9] it was shown that Equation 2.3 converges if

$$\parallel Q - LP \parallel_\infty < 1 \tag{2.4}$$

Where the infinity norm for a discrete time transfer function is given by

$$\parallel H(z) \parallel_\infty = \max_{\omega \in [0, \frac{\omega_s}{2}]} \{|H(e^{j\omega})|\}, \tag{2.5}$$

where ω_s corresponds to the Nyquist frequency.

The proof of convergence is as follows, where r corresponds to the desired input and y_k to the output at that iteration [9]:

$$u_{k+1} = Qu_k + Le_k = Qu_k + L(r - y_k) = Qu_k + L(r - Pu_k) = f(u_k) \tag{2.6}$$

If $f(u_k)$ satisfies the following inequality,

$$\parallel f(x_1) - f(x_2) \parallel = \parallel Q(x_1 - x_2) - LP(x_1 - x_2) \parallel \leq \parallel Q - LP \parallel_\infty \parallel x_1 - x_2 \parallel \tag{2.7}$$

Then $f(u_k)$ is said to be Lipshitz. A Lipschitz function is one where there exists a constant C independent of x and y which satisfies the inequality below:

$$|f(x) - f(y)| \leq C|x - y|$$

For a Lipshitz function the function converges to it's fixed point $u^*_{k+1} = u^*_k$ if the Lipschitz constant is less than 1. In the above equation the Lipschitz constant is $\parallel Q - LP \parallel_\infty$. Therefore the term will converge to the final solution as long as it is less than 1.

To achieve no error, Q must be chosen to be a gain of 1. If this is the case, then 2.3 becomes:

$$u_{k+1} = u_k + Le_k \tag{2.8}$$

Which converges to the fixed point

$$u^* = u^* + Le^* \tag{2.9}$$

This implies that $e^* = 0$ [9]. This is not always possible since it sometimes requires the plant P to be invertible[9].

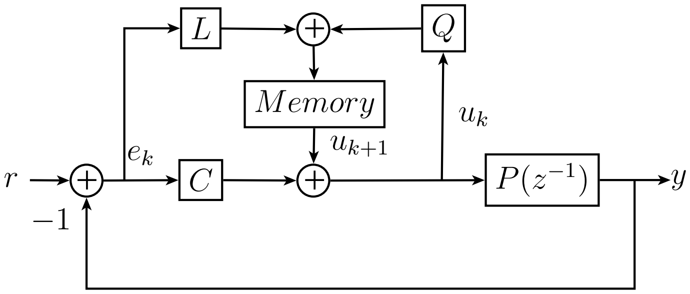

Figure 2.2: ILC Control Structure 1

The control equation corresponding to Figure 2.2 is shown in Equation 2.10 below.

$$u_{k+1} = Qu_k + Le_k + Ce_{k+1} \tag{2.10}$$

For the system in Figure 2.2 with the additional controller C it was shown in [10] that 2.10 converges as long as the parameters Q and L satisfy the inequality 2.11.

$$\| \frac{Q - LP}{1 + PC} \|_\infty < 1 \tag{2.11}$$

Given the inequality 2.11 $L = QP^{-1}$ results in the fastest convergence, where Q is an IIR filter usually picked to have low pass characteristics. Since the minimum error is achieved when Q is a gain of one, the higher the cut off frequency of Q, the lower the error. Ideally Q would always be one which allows the error to converge down to the noise floor of the system but in practice this is often not possible.

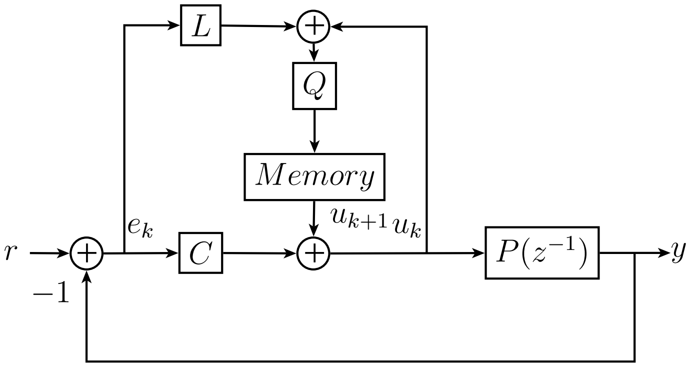

Figure 2.3: ILC Control Structure 2

The control equation corresponding to Figure 2.3 is shown in Equation 2.12.

$$u_{k+1} = Q(u_k + Le_k) + Ce_{k+1} \tag{2.12}$$

The low pass filter Q is in place to decrease the sensitivity to high frequency modelling errors. The Q filter corner frequency should be chosen large enough to cover the bandwidth of the plant and controller combination but low enough to avoid introducing errors caused by high frequency uncertainties. To assure convergence of the ILC algorithm 2.12 the inequality in 2.13 must be valid.

$$\| Q[\frac{1-LP}{1+PC}] \|_{\infty} < 1 \tag{2.13}$$

Using the convergence criteria above, the choice for L that gives the fastest convergence is $L = P^{-1}$.

The previous two block diagrams show the use of memory blocks which are used to store one full period of the control signal to be used in the next iteration. The performance of conventional ILC (with memory blocks) and an ILC structure where the memory blocks were removed was compared in [10]. With the memory removed, the reference trajectory was continually input into the system like it is done with repetitive control. It was shown that there was very little difference between the equivalent feedback implementation and the conventional ILC. This also allows the L and Q filters to be used in the control system in real time. The major constraint this imposes is that the filter L must be causal. In many systems this is not possible since inverting the plant results in a non causal transfer function.

Since the control system in the experiment described in Chapter 4 was implemented on an FPGA, it was not possible to use a memory buffer in the ILC implementation. The FPGA did not have enough memory to store one complete period of the tracking trajectory and therefore, the memory blocks were removed and the system was operated in a repetitive control manner. Figure 2.4 shows the resulting block diagram with the memory block removed.

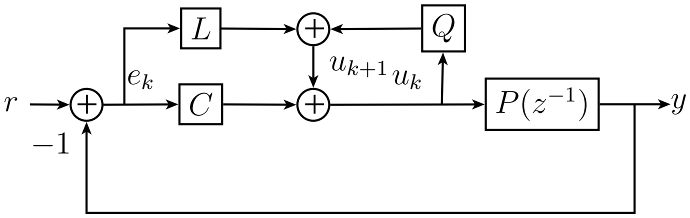

Figure 2.4: ILC control structure with memory block removed

2.2 RST Control

2.2.1 RST Overview

RST controllers are a polynomial pole placement controller consisting of three polynomials: R, S and T. This two degree of freedom controller gives the designer lots of flexibility and allows the user to design for separate tracking or regulation requirements. RST controllers have been used in the power electronics, automotive and robotics industries as well as many other applications [11][12][13][14].

The design of RST controllers is performed mainly in the frequency domain using a digital LTI transfer function model of the system to be controlled. The design is implemented by shaping the different frequency functions such as the input sensitivity function or the output sensitivity function as well as by specifying specific dominant pole locations for the desired system response. This allows the user to design for certain robustness requirements using the sensitivity functions and also achieve the desired system step response by selecting the appropriate dominant poles. For the application in Chapter 4 the desired system response was to achieve very accurate tracking of high frequency trajectories; therefore a fast step response was desired to be able to track as quickly as possible.

The RST control structure is shown in Figure 2.5.

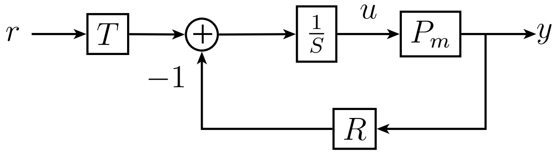

Figure 2.5: RST Control Structure

The equation that corresponds with the RST structure is given by

$$\begin{aligned} Su &= Tr - Ry \\ y &= uP_m(z^{-1}) = u\frac{z^{-d}B(z^{-1})}{A(z^{-1})} \end{aligned} \tag{2.14}$$

Where r is the desired control trajectory for the actuator model $P_m(z^{-1})$, y is the output of the model and d is the model delay. Using (2.14) the closed loop system can then be described by

$$H_{CL}(z^{-1}) = \frac{z^{-d}B(z^{-1})T(z^{-1})}{A(z^{-1})S(z^{-1}) + z^{-d}B(z^{-1})R(z^{-1})} = \frac{z^{-d}B(z^{-1})T(z^{-1})}{P(z^{-1})} \tag{2.15}$$

Where $P(z^{-1})$ corresponds to the equation of the desired closed loop poles. It is shown in [1] how the polynomials R and S can be obtained using the

Bezout identity when the polynomials A, B and P are given.

$$P(z^{-1}) = A(z^{-1})S(z^{-1}) + z^{-d}B(z^{-1})R(z^{-1}) \tag{2.16}$$

In [1] the following are defined:

$$n_A = degA(z^{-1})$$

$$n_B = degB(z^{-1})$$

$$n_P = degP(z^{-1}) \leq n_A + n_B + d - 1$$

$$n_S = degS(z^{-1}) = n_B + d - 1$$

$$n_R = degR(z^{-1}) = n_A - 1$$

Then the polynomials R and S can be expressed as

$$S(z^{-1}) = 1 + s_1 z^{-1} + \cdots + s_{n_S} z^{-n_S} = 1 + z^{-1} S^*(z^{-1}) \tag{2.17}$$

$$R(z^{-1}) = r_0 + r_1 z^{-1} + \cdots + r_{n_R} z^{-n_R} \tag{2.18}$$

The Diophantine equation (2.16) can be solved in matrix form $x = M^{-1}p$ where x, p and M are given by

$$x^T = [1, s_1, \ldots, s_{n_S}, r_0, \ldots, r_{n_R}]$$
$$p^T = [1, p_1, \ldots, p_{n_P}, 0, \ldots, 0]$$

$$M = \begin{bmatrix} 1 & 0 & \cdots & 0 & 0 & \cdots & \cdots & 0 \\ a_1 & 1 & \cdots & \cdot & b'_1 & & & 0 \\ a_2 & & & 0 & b'_2 & & & b'_1 \\ \cdot & & & 1 & \cdot & & & b'_2 \\ \cdot & & & a_1 & \cdot & & & \cdot \\ a_{n_A} & & & a_2 & b'_{n_B} & & & \cdot \\ 0 & & & \cdot & 0 & & & \cdot \\ 0 & \cdots & 0 & a_{n_A} & 0 & 0 & 0 & b'_{n_B} \end{bmatrix}$$

$b'_i = 0$ for $i = 0, 1, \ldots d$ and $b'_i = b_{i-d}$ for $i \geq d + 1$ [1]

Solving the Diophantine equation gives the polynomials R and S to be used in the controller. In some cases the polynomials R and S may have specific desired components. For example if no steady state error is desired then the S polynomial must contain a $(1 - z^{-1})$ term. If rejection of a specific disturbance is necessary then the S term must contain a pair of complex zeros at the desired frequency using Equation 2.19. Therefore the S and R terms may contain a fixed section which corresponds to different design needs. Rewriting the Diophantine equation allows the addition of these fixed terms. The terms H_R and H_S correspond to the known fixed parts of S and R. $R^{'}(z^{-1})$ and $S^{'}(z^{-1})$ must be found using the method shown above.

$$\begin{aligned} &H_S(z^{-1}) = 1 + \alpha z^{-1} + z^{-2} \\ &\text{where } \alpha = -2cos(2\pi f/f_s) \end{aligned} \tag{2.19}$$

$$\begin{aligned} &P(z^{-1}) = A(z^{-1})S(z^{-1}) + z^{-d}B(z^{-1})R(z^{-1}) \\ &\text{where } S(z^{-1}) = H_S(z^{-1})S^{'}(z^{-1}) \\ &\text{and } R(z^{-1}) = H_R(z^{-1})R^{'}(z^{-1}) \end{aligned} \tag{2.20}$$

2.2.2 Control System Design Using an RST Structure

Design of an RST controller can be completed in three steps. First the desired close loop performance such as the settling time, overshoot and rise time are achieved by selecting the appropriate dominant poles. The second step is to set the fixed parts of S and R to achieve zero steady state error or rejection of a periodic disturbance. Then robustness can be improved using sensitivity function shaping. The output sensitivity function can be expressed as:

$$S_y = \frac{1}{1 + H_{OL}} = \frac{A(z^{-1})S(z^{-1})}{A(z^{-1})S(z^{-1}) + z^{-d}B(z^{-1})R(z^{-1})} \tag{2.21}$$

The output sensitivity function describes how sensitive the control and plant model parameters are to disturbances acting on the system. Sensitivity function shaping is based on limiting the modulus margin (ΔM). The modulus margin can be calculated as the inverse of the maximum of the output sensitivity function. This value represents the shortest distance from -1 to the curve of the open loop nyquist plot. A good choice for ΔM is for it to be greater than 0.5 [1]. This requires the maximum of the sensitivity function to be less than 2.0 or 6dB. In Figure 2.6 the modulus margin, inverse of gain margin (ΔG) and phase margin ($\Delta\Phi$) are shown together in one nyquist plot. Choosing an appropriate modulus margin usually assures good values for the phase and gain margins. Typically the gain margin should be greater than 6dB and the phase margin should be between 30°and 60°.

The gain margin is given by the difference between 0dB and the magnitude of the open loop bode plot when the phase crosses the -180°line. The phase margin can be calculated by finding the difference in phase between -180°and the phase when the magnitude crosses the 0dB line.

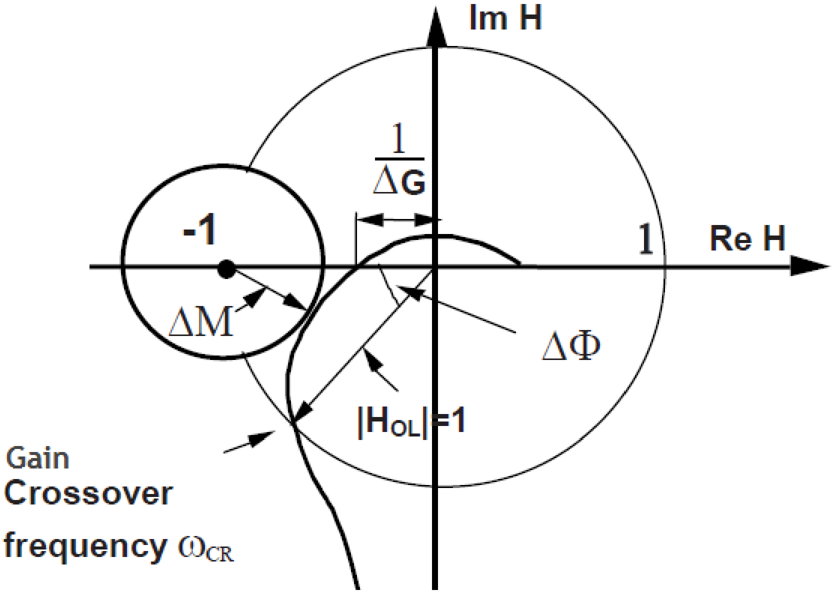

Figure 2.6: Nyquist Plot Showing Modulus, Phase and Gain Margin [1]

When designing an RST controller the initial step is to pick the desired dominant closed loop poles which will dictate the rise time, overshoot and settling time of the system and then calculate the polynomials R and S. Once they have been calculated the output sensitivity function can be plotted and the maximum value can be calculated. If all design requirements such as the modulus, gain, and phase margin are met then the design is complete. If the maximum of the sensitivity function is too high then the value can be decreased by introducing real auxiliary poles or by adding notch filters. Which one gets added depends on three situations as described in [1]:

1. The maximum occurs in a high frequency band far away from the attenuation band of S_y.
2. The maximum occurs in a frequency region close to the attenuation band of S_y.

3. The maximum occurs both at high and low frequencies.

For cases one and three, real auxiliary poles can be added at the high frequencies, doing this will move the maximum towards lower frequencies which is case 2. In case 2 a notch filter can be added by including the zeros of the filter into the fixed part of $S(z^{-1})$ and including the poles into the closed loop polynomial $P(z^{-1})$. A discretization of the notch filter shown in equation (2.22) can be used.

$$F(s) = \frac{s^2 + 2\zeta_{num}\omega_0 s + \omega_0^2}{s^2 + 2\zeta_{den}\omega_0 s + \omega_0^2} \tag{2.22}$$

The auxiliary poles are typically added between .05 and 0.5 on the real axis[1]. The addition of the real poles lowers the value of sensitivity function around the frequency of the pole and improves robustness. Since the auxiliary poles are at a much higher frequency than the dominant poles they do not have an affect on the response of the closed loop system.

Once the R and S polynomials have been computed, the T polynomial can be chosen. Depending on the goal, the T polynomial can be selected in two ways. If the desired tracking dynamics is equivalent to the regulation dynamics then the T polynomial reduces to a simple gain to ensure correct gain matching between the reference and the output [1].

$$T(z^{-1}) = \begin{cases} \frac{P(1)}{B(1)} \text{ if } B(1) \neq 0 \\ 1 \quad\ \text{ if } B(1) = 0 \end{cases} \tag{2.23}$$

If separate tracking and regulation dynamics are desired then the T polynomial is chosen as

$$T(z^{-1}) = GP(z^{-1}) \tag{2.24}$$

$$G = \begin{cases} \frac{1}{B(1)} \text{ if } B(1) \neq 0 \\ 1 \quad \text{ if } B(1) = 0 \end{cases} \tag{2.25}$$

Using Equation 2.24 will result in the closed loop poles being cancelled which leaves the following expression for the closed loop system.

$$H_{CL}(z^{-1}) = \frac{z^{-d}B(z^{-1})}{B(1)} \tag{2.26}$$

The tracking dynamics would then come from the trajectory generator since the closed loop system is not affecting the input.

2.3 Adaptive Feed Forward Cancellation

Adaptive Feed Forward Cancellation (AFC) is one of the techniques used in repetitive control. Repetitive control is based on the internal model principle. The internal model principle is that to completely reject a disturbance, the model of that disturbance must be included in the control loop [15] [16]. AFC filters are used in conjunction with a controller such as a PID or lead-lag compensator to remove sinusoidal disturbances or to track sinusoidal trajectories. AFC filters are commonly used in high precision diamond turning applications because of the sinusoidal nature of the reference trajectories [17][18] but they can be used anywhere sinusoidal disturbances must be rejected. A major advantage of AFC filters is that they can be “plugged” into a controller without any changes to the existing controller structure.

The adaptive feed forward cancellation algorithm is based on generating a control signal such that the disturbance 2.27 is completely rejected.

$$d(t) = a_1 cos(\omega_1 t) + b_1 sin(\omega_1 t) \tag{2.27}$$

Bodson in [16] showed that this can be accomplished by letting the control signal $u(t)$ equal to

$$u(t) = \theta_1 cos(\omega_1 t) + \theta_2 sin(\omega_1 t) \tag{2.28}$$

Such that the disturbance is completely cancelled out when the adaptive parameters reach the solution: $\theta^* = [\theta_1, \theta_2]^* = [-a_1, -b_1]$

One possible update law used by Messner and Bodson in [19] was to let the parameters θ_1 and θ_2 be updated by the continuous update law:

$$\begin{aligned} \dot{\theta}_1(t) &= gycos(\omega_1 t + \phi) \\ \dot{\theta}_2(t) &= gysin(\omega_1 t + \phi) \end{aligned} \tag{2.29}$$

Where g corresponds to the gain of the adaptive algorithm and y the output of the system.

The AFC algorithm can be easily extended to remove multiple harmonics by including more terms in the control equation. The algorithm above was expressed as a continuous time LTI transfer function in [19]:

$$H(s) = g_n \frac{scos(\phi_n) + \omega_n sin(\phi_n)}{s^2 + \omega_n^2}, \tag{2.30}$$

where s corresponds to the Laplace transform parameter $s = \omega j$, the parameters g_n, ω_n, and ϕ_n correspond to the learning gain, frequency to reject and the phase advance parameter. The transfer function in Equation 2.30 was converted to the discrete domain using a zero order hold so that it could be implemented on the real system.

Expressing the AFC algorithm using the transfer function in 2.30 simplifies the stability analysis of the closed loop system when multiple AFC resonators are used in a system. This can be done in the frequency domain by adding more AFC resonators in parallel. Lu in [20] examined having the error pass through and then added the AFC filters in parallel to the inner control loop as shown in Figure 2.7.

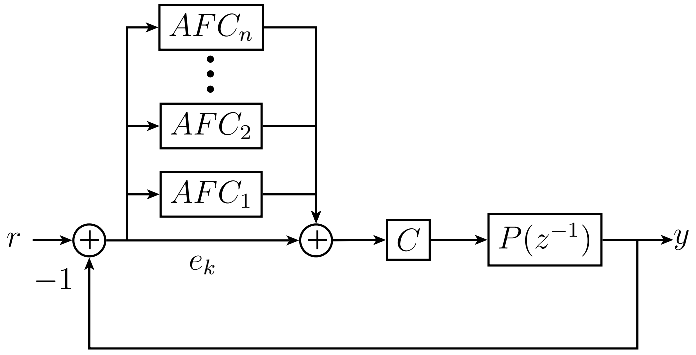

Figure 2.7: AFC control system block diagram

One way of designing a control system using an AFC controller is the loop shaping method explained by Byl et al. in [17] and Lu in [20]. When designing a control system using AFC resonators, the gain and phase advance parameters must be selected to achieve stability of the closed loop and the necessary rejection of the harmonic disturbances. The ω_n parameter of each resonator must be selected to match the desired frequency to be rejected.

The resonator has two complex poles on the imaginary axis corresponding to the frequency of ω_n and one zero on the real axis. The phase advance parameter ϕ_n affects where on the real axis this zero is placed; this in turn affects the shape of the frequency plot of the resonator. It was shown in [17] that the choice of ϕ_n does not have a significant affect on the magnitude of the frequency response close to the resonant peak but it does have a larger affect on the phase response of the resonator.

The phase response of the AFC resonator has a discontinuity of $-\pi$ radians centred at ϕ_n at the resonator frequency. Figure 2.8 shows two separate AFC filters with the same resonant frequency but different phase parameters. As can be seen in the figure, the phase plot is centred on ϕ_n. It is also clear that the phase parameter does not have any noticeable effect on the magnitude response of the filter.

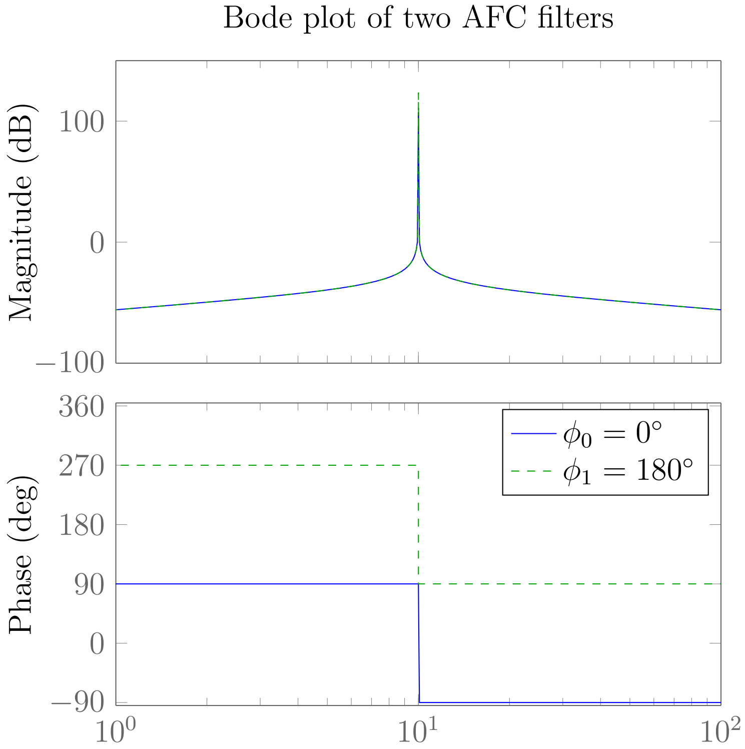

Figure 2.8: Bode plot of two AFC Resonators

It was shown in [20] that the phase advance ϕ_n can be set equal to the phase of the closed loop system (shown below). This ensures that the phase margin of the overall system is unaffected by the addition of the AFC filters.

$$\phi_n = \angle \frac{P(j\omega_n)C(j\omega_n)}{1 + P(j\omega_n)C(j\omega_n)} \tag{2.31}$$

2.4 PID Control

Proportional-Integral-Derivative (PID) control is the most widely used control system in industry. This is because of several reasons: the PID controller is fairly easy to tune and does not need a system model. As the name suggests a PID controller consists of three sections: a proportional term, an integral term and a derivative term. Two common configurations for a PID controller are to have each term added in parallel or in series. The advantage of having the system in parallel is that it is easier to adjust each term independently. The control term $u[k]$ for the parallel PID controller is given by:

$$u[k] = K_p e[k] + K_i \sum e[k] + K_d \nabla e[k], \tag{2.32}$$

where ∇ and $\sum$ correspond to the backwards difference operator and a discrete integration operation respectively.

The following table from [21] shows the affect each parameter has on the overall system. This table can be used to very quickly and effectively perform online tuning of a system until the desired closed-loop requirements are met.

Closed-Loop Response	**Rise Time**	**Overshoot**	**Settling Time**	**Steady-State Error**	**Stability**
Increasing $\mathrm{K_p}$	Decrease	Increase	Small Increase	Decrease	Degrade
Increasing $\mathrm{K_i}$	Small Decrease	Increase	Increase	Large Decrease	Degrade
Increasing $\mathrm{K_d}$	Small Decrease	Decrease	Decrease	Minor Change	Improve

Table 2.1: Affect of PID parameters on step response

The online tuning approach can also be combined with experimental bode plot estimates so that the system gain and phase margins can be observed while tuning the system. These techniques allow very effective tuning without knowledge of a system model ahead of time. The figure below is a typical block diagram of a parallel form PID controller where $P(z^{-1})$ corresponds to the LTI system to be controlled.

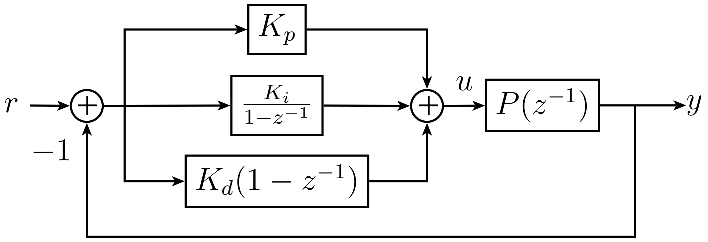

Figure 2.9: Parallel PID block diagram

2.5 Conclusions

This chapter introduced the different control algorithms used for the experiments in Chapters 3 and 4. It was mentioned that the ILC and AFC controllers are specialized controllers for periodic trajectories which are not heavily dependent on an accurate system model. The RST and PID controllers are capable of tracking any trajectory but the RST does require an accurate model of the system to be controlled. The advantage of the PID is that it can be hand tuned to the required performances of the system.

Chapter 3

Modelling, Controller Design and Simulation

The purpose of this chapter is to show the simulation performance of the different algorithms on a galvanometer actuator and to establish a best case scenario of performance which can be used to compare the experimental results performed in Chapter 4. This chapter will also introduce the modelling performed for a galvanometer actuator as well as the controller design process and simulation results of the ILC, RST, AFC, and PID controllers.

3.1 Modelling

Many high precision motion control algorithms are implemented using a mathematical model of the system to be controlled. Such models can be continuous, discrete, linear or nonlinear. The model of the galvanometer used here will be an LTI discrete model and the parameters of this model will be obtained using identification techniques. In system identification, a known excitation signal is inserted into the actuator and its response is recorded. Then recursive identification algorithms can be used to find the coefficients of the actuator to reproduce the measured response. A general view of modelling can be seen in the figure below:

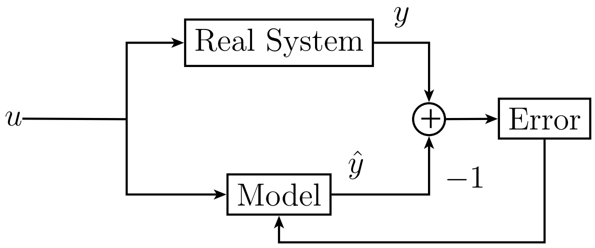

Figure 3.1: General modelling block diagram

In system identification the relationship between the input and the output can be described using the difference equation shown below [22].

$$y(t) + a_1 y(t-1) + ... + a_n y(t-n) = b_1 u(t-1) + ... + b_m u(t-m)$$

Which can be expressed in vector form as:

$$\begin{aligned} &\theta = \begin{bmatrix} a_1 & ... & a_n & b_1 & ... & b_m \end{bmatrix}^T \\ &\phi(t) = \begin{bmatrix} -y(t-1) & ... & -y(t-n) & u(t-1) & ... & u(t-m) \end{bmatrix}^T \\ &\hat{y}(t) = \phi^T(t)\theta \end{aligned} \tag{3.1}$$

The simplest form of identifying the parameters θ using this equation is the least squares method, which attempts to minimize the least squares error of $y(t) - \hat{y}(t)$ where $y(t)$ corresponds to the measured data and $\hat{y}(t)$ to the predictor [22].

Choice of Input Signal

The choice of input signal or otherwise known as the excitation signal is very important in system identification. The input signal must have a large enough bandwidth to excite the system at higher frequencies and it should have enough length to get an accurate low frequency measurement [22]. Another important property is for the signal to have as much input power as possible since a small covariance is desired and the covariance matrix is inversely proportional to the input power [22]. The covariance matrix describes the statistical relationship between the parameters θ and the input spectrum.

For practical reasons the signal must also be bounded; therefore the crest factor is used as a guide when choosing an input signal. The crest factor (C_r) is given by the ratio of the peak value of waveform to its RMS value. A small crest factor is desired which means there are few large peaks in the signal. The minimum crest factor is a value of 1 which is common of binary signals. Because of this fact, a common excitation signal is to use a Pseudo-Random-Binary-Sequence (PRBS). A PRBS can be designed to have any desired bandwidth, has a small crest factor and can be made arbitrarily long.

$$C_r = \frac{|u|_{peak}}{u_{rms}} \tag{3.2}$$

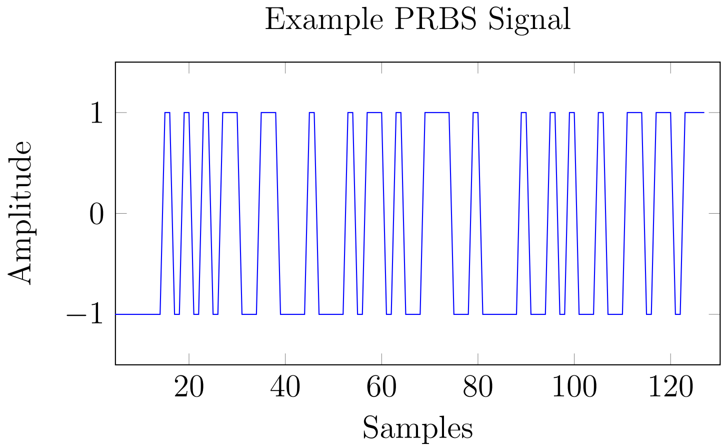

Figure 3.2: PRBS signal example

3.1.1 Closed Loop System Identification

The identification block diagram and method described at the start of this chapter shows an open loop system with no controller. This is the best way to perform identification since the output is directly dependent on the input with no effects from a closed loop or controller present. This allows a model of the plant itself to be created. In many cases open loop identification is not possible because of non-linearities in the system such as drift in the position or the system is unstable in open loop. However if the motor cannot be operated in open loop then it was shown in [23] that it is better to identify the system in closed loop using closed loop ID algorithms.

When the actuator cannot be operated in open loop, it is necessary to perform the identification in closed loop using a controller. Closed loop identification techniques involve using the controller parameters to modify the measured data while the identification is performed. This results in a new measurements vector $\phi(t)$ which includes the effect of the closed loop system. Figure

3.3 from [1] shows a general closed loop identification diagram using an RST controller.

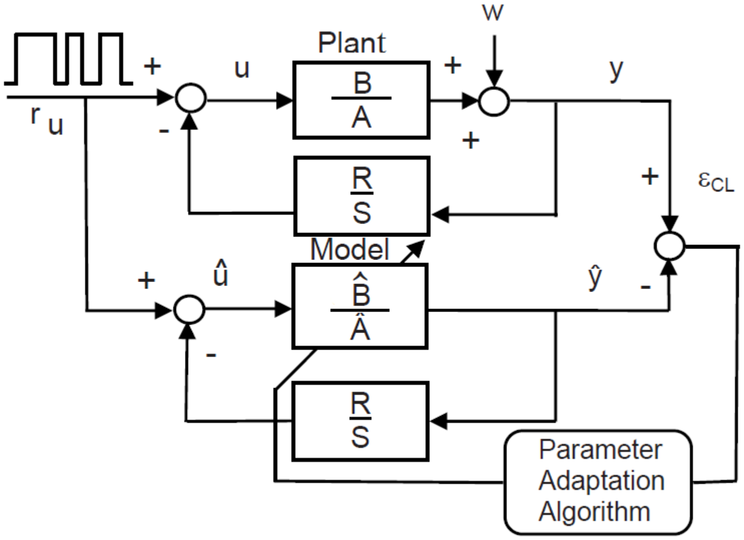

Figure 3.3: Closed Loop Identification Using RST Controller

An algorithm of doing closed loop identification called Extended Closed Loop Output Error (XCLOE) is described in [1] where the closed loop RST controller is taken into account. The XCLOE algorithm uses an ARMAX model to represent the plant and disturbances acting on it. The system equations when using the XCLOE algorithm are shown below [1]. Where d is the delay of the system, $e(t)$ is Gaussian white noise, A, B, and C are the denominator, numerator, and noise filter polynomials of the system respectively.

$$\begin{aligned} y(t) &= -A(z^{-1})y(t-1) + B(z^{-1})u(t-d-1) + C(z^{-1})e(t-1) + e(t) \\ u(t) &= -\frac{R(z^{-1})}{S(z^{-1})}y(t) + r_u(t) \end{aligned} \tag{3.3}$$

Similarly to Equation 3.1 the adjustable closed loop predictor is given by the following more complex equations from [1].

$$\begin{aligned}\hat{y}(t) &= -A(z^{-1}) + B(z^{-1})\hat{u}(t-d-1) + H(z^{-1})\frac{\epsilon_{CL}(t-1)}{S(z^{-1})} \\ &= \hat{\theta}(t-1)^T\phi(t-1),\end{aligned} \tag{3.4}$$

where

$$\begin{aligned}&\hat{u}(t) = -\frac{R(z^{-1})}{S(z^{-1})}\hat{y}(t) + r_u(t) \\ &\hat{\theta}(t)^T = \begin{bmatrix} a_1(t) & ... & a_{n_A}(t) & b_1(t) & ... & b_{n_B}(t) & h_1(t) & ... & h_{n_H}(t)\end{bmatrix} \\ &\phi(t)^T = \begin{bmatrix}\phi(t)^T & \epsilon_{CLf} & ... & \epsilon_{CLf}(t-n_H+1)\end{bmatrix} \\ &\epsilon_{CLf}(t) = \frac{\epsilon_{CL}(t)}{S(z^{-1})} \\ &\epsilon_{CL}(t) = y(t+1) - \hat{\theta}(t+1)^T\phi(t) \\ &\text{and} \\ &H(z^{-1}) = 1 + C(z^{-1})S(z^{-1}) - P(z^{-1}).\end{aligned} \tag{3.5}$$

The parameters θ are then updated recursively using the following equations from [1].

$$\begin{aligned}\hat{\theta}(t) &= \hat{\theta}(t-1) + F(t-1)\phi(t-1)\epsilon(t) \\ F(t)^{-1} &= \lambda_1(t-1)F(t-1)^{-1} + \lambda_2\phi(t-1)\phi(t-1)^T\end{aligned} \tag{3.6}$$

The gain F is called the adaptation gain of the algorithm and influences how fast the algorithm converges. The two other gains λ_1 and λ_2 are called forgetting factors. Depending on the value of the forgetting factor different weights can be assigned to the error data during the identification. Small values will make the identification more sensitive to the current samples. Typically a forgetting factor between 0.98 and 1 is used [1].

3.1.2 Galvanometer Modelling

Laser scanning galvanometers consist of the galvanometer itself plus a mirror connected to the shaft which is used for deflecting a laser beam onto the work surface. A galvanometer works by using a permanent magnet attached to a pivot with a small air gap and a coil surrounding the magnet. When a current is applied to the coil, the magnet will move to try and orient itself. Because of the attachment of the magnet, galvanometers cannot move one full rotation, they usually only pivot between two angles. Figure 3.4 shows two galvanometer scanners with mirrors attached.

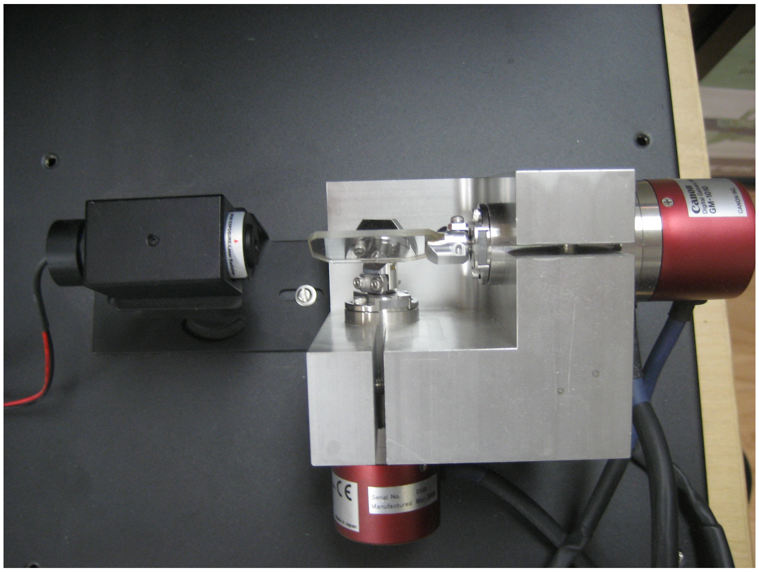

Figure 3.4: Two galvanometers with mirrors attached

As derived in [24] and [25] the continuous time galvanometer transfer function between the output position and the input current is given by:

$$
\begin{aligned}
&\frac{\theta(s)}{i(s)} = \frac{T_i}{J_r}\frac{s^2+a}{(s^2+aJ_m/J_r)(s^2+a)-a^2J_m/J_r}\\
&\text{where}\\
&a = \omega_r^2 J_m/J_r\\
&\omega_r = \sqrt{\frac{K(J_r+J_M)}{J_rJ_M}}
\end{aligned}
\tag{3.7}
$$

The parameters J_m, J_r, T_i, K correspond to the mirror inertia, rotor inertia, motor torque constant and the stiffness of the rotor shaft. The parameters of Equation 3.7 can often be acquired from the actuator data sheets but it was not available for the galvanometer used in this work and therefore Equation 3.7 could not be used, but it serves as a starting point for what the transfer function should resemble.

The Canon galvanometer had a small amount of position drift when operated in open loop since the weight of the mirror would make the shaft rotate into an equilibrium position. It was also not possible to inject a signal with sufficient input power without the motor hitting its end stops; therefore the system identification was performed in closed loop. This allowed a larger amplitude signal to be used and removed all position drifts.

A loosely tuned PD controller was used in the closed loop controller. The excitation signal used for the identification was a pseudo random binary sequence (PRBS). The PRBS signal was inserted at the output of the PD controller as shown in Figure 3.5. The reference was set to 0 to centre the galvanometer during the identification.

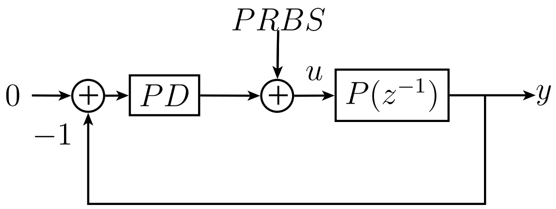

Figure 3.5: Closed Loop Identification

In the XCLOE algorithm the closed loop controller is assumed to be an RST type controller. Which means that the PD controller used in the real system must be converted to the RST controller form. This can simply be done by rearranging the block diagram in Figure 3.5 into the block diagram shown below in Figure 3.6.

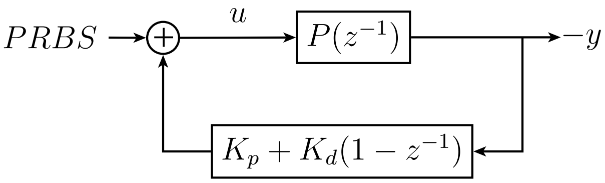

Figure 3.6: Closed Loop Identification

The following equations can then be obtained from the rearranged system model.

$$u = r - y(K_p + K_d(1 - z^{-1})$$

The previous equations can then be compared against the typical RST controller equation shown below:

$$Su = Tr - Ry$$

Which results in the following values for R, S, and T.

$$\begin{aligned} R &= K_p + K_d - K_d z^{-1} \\ T &= 1 \\ S &= 1 \end{aligned}$$

A 4 KHz 0.3 rad amplitude PRBS signal was injected into the output of the controller and the encoder position was measured. Two separate data sets were collected, one was used for the identification and the other for the validation. The XCLOE algorithm with the above RST polynomials was used. Using Equation 3.7 as a guide, the model orders that gave the best validation results were $n_A = 3$, $n_B = 1$, $n_C = 6$ and $d = 2$. Figure 3.7 shows the real and model output using the second validation data with a percent fit of about 98% calculated using the R^2 formula shown below.

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}}, \tag{3.8}$$

where

$$SS_{tot} = \sum_i (y_i - \bar{y})^2,$$

$$SS_{res} = \sum_i (y_i - f_i)^2,$$

where y_i corresponds to the measured data, f_i corresponds to the modelled data and $\bar{y}$ to the mean of the measured data.

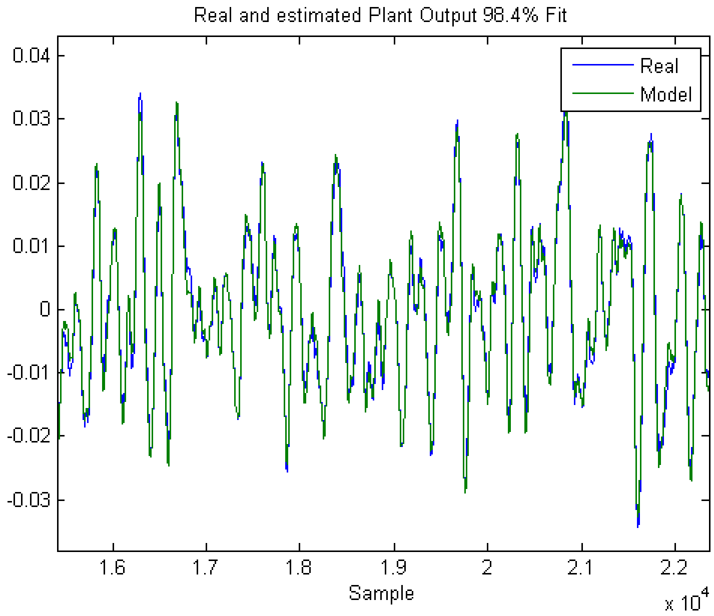

Figure 3.7: Model validation test

As shown in Figure 3.8 the model also passed the uncorrelation test between the residuals and the predicted output. The uncorrelation test describes the correlation between the observations $\phi(t)$ and the closed loop predictor error [1]. If these are uncorrelated then it ensures unbiased parameter estimates which are not dependent on the excitation signal used[1]. This test combined with the good percent fit give confidence that the model is accurate enough for control.

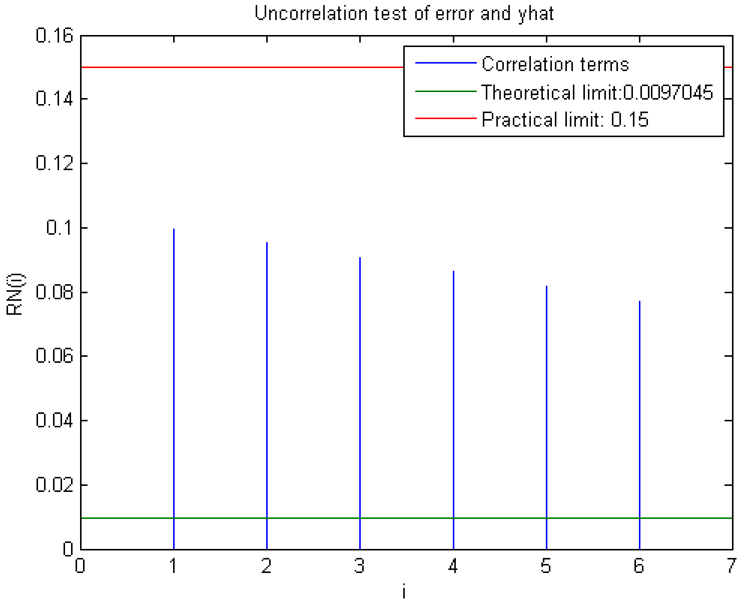

Figure 3.8: Model uncorrelation test

The model coefficients are listed below along with the bode plot. It can be seen from the bode plot that the model has an open loop frequency bandwidth of 100 Hz.

Model Coefficients				
	z^0	z^{-1}	z^{-2}	z^{-3}
Numerator	0	0	0	0.0001326
Denominator	1	-1.586340634	0.184711901	0.402245189

Table 3.1: Digital Model Coefficients

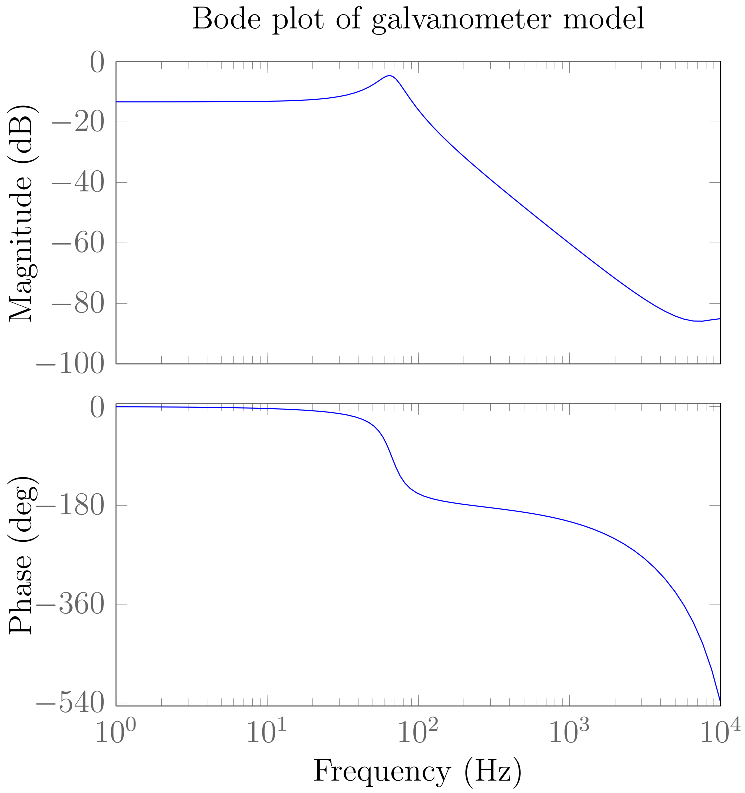

Figure 3.9: Bode plot of galvanometer model

To test the galvanometer model, a slow and simple RST controller with dominant poles at 50 Hz was implemented. The slow poles were chosen to ensure the controller did no saturate. The simulated closed loop frequency response was then compared with the experimental frequency response. Figure 3.10 shows the two bode plots. The model and experimental bode plots are very close but do seem to have a slightly different gain and cut off frequency.

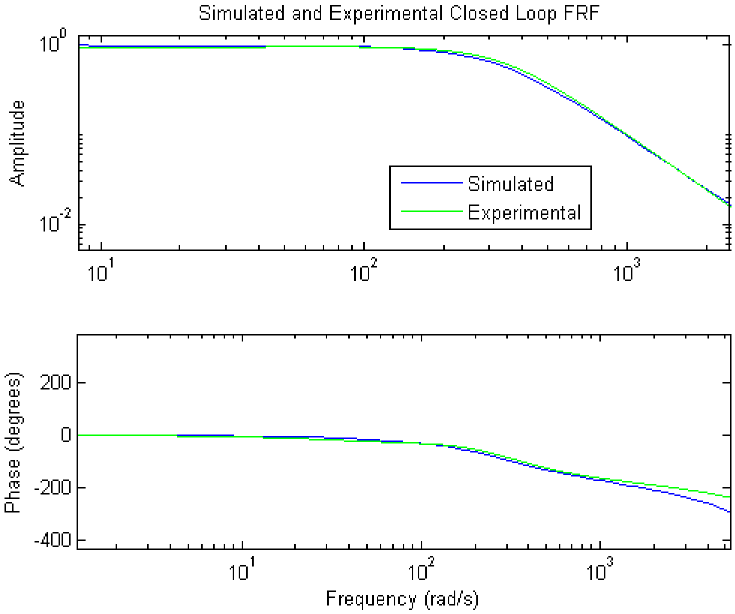

Figure 3.10: Experimental and theoretical closed loop bode plot

3.2 Controller Design and Simulations

Laser scanning galvanometers are used in a variety of different applications and each one requires a different trajectory to be tracked which are commonly periodic. The one thing that is common between each application is that the galvanometer must be fast and accurate while following a reference signal; therefore the desired controller should have as low of a tracking error as possible and be as fast as possible while staying within the physical limitations of the system.

3.2.1 Iterative Learning Control

As was mentioned in Chapter 2 almost all ILC systems are implemented using a memory block which holds one entire period of the control signal to be used for the process. Each trial is then repeated from the start until the error converges to the final value. Since this control system is implemented on an FPGA there is not enough memory on board to hold one entire period of the signal. It was shown in [10] that the memory block could be removed and the reference signal could be continuously passed to the input of the controller as if it were more like a repetitive controller. The modified control structure with the memory removed is shown in Figure 3.11 below. The ILC simulations also do not use a memory block.

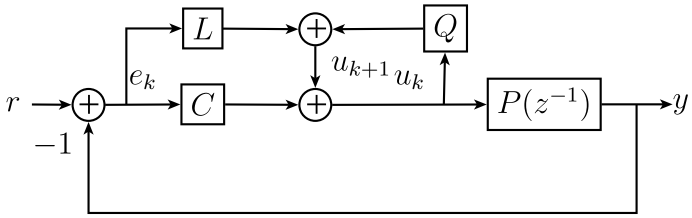

Figure 3.11: ILC control structure with memory block removed

When designing an Iterative Learning Controller there are two major steps: designing the L and Q filters. The L filter can be obtained fairly easily using Equation 2.11. The choice of the Q filter will have a significant effect on the tracking error of the entire system. Using the block diagram shown in Figure 2.2 the error term of the closed loop system can be calculated to be:

$$E(z^{-1}) = \frac{Y(1-Q)}{P(L+C)} \tag{3.9}$$

The best choice of Q for minimum error would be to let it be equal to one. This is what was used by Arimoto in [5]. In many ILC systems a choice of $Q = 1$ is not possible since it makes the overall system unstable or drives the system into saturation; therefore it is common to let Q be a low pass filter. The filter cut off frequency should be made as high as possible to minimize the error while still satisfying Equation 2.11. The Q filter cut off can also be interpreted as the controller being able to deal with frequencies in the input and control signal up to the cut off frequency of Q.

The choice of learning filter will affect the speed of convergence of the algorithm and it depends heavily on the plant model. As was mentioned in Chapter 2, the choice of the L filter which achieves the fastest convergence is to let $L = QP^{-1}$. One issue with using the inverse of the plant model is that any non minimum phase zeros become unstable poles and any system with a higher order numerator than denominator (such as systems with time delays) will result in a non-causal filter. The model used in this work falls under the category of the latter. The following section will explain how a causal inverse can be approximated.

Inverting A System with Time Delay for the Design of the Learning Filter L

When performing system identification in the discrete domain, it is very common for there to be a delay. This results in a numerator which has no constant term; therefore when it is inverted it results in a non-causal filter which cannot be implemented in a real-time system. This presents a problem when designing an ILC controller where the optimal learning (L) filter choice involves inversion of the plant model. To achieve a causal L filter, an approximation to the real inverse can be made. Given the system $P(z^{-1})$ and its inverse $P^{-1}(z^{-1})$:

$$\begin{aligned} P(z^{-1}) &= \frac{b_1 z^{-1} + b_2 z^{-2} + b_3 z^{-3}}{1 + a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3}} \\ P^{-1}(z^{-1}) &= \frac{1 + a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3}}{b_1 z^{-1} + b_2 z^{-2} + b_3 z^{-3}} \end{aligned} \tag{3.10}$$

It is clear that the inverse transfer function is non-causal. A method which can be used to approximate a causal inverse is to add a constant term to the numerator before inverting it as shown below.

$$\begin{aligned} P^{*}(z^{-1}) &= \frac{1 + b_1 z^{-1} + b_2 z^{-2} + b_3 z^{-3}}{1 + a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3}} \\ P^{*-1}(z^{-1}) &= \frac{1 + a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3}}{1 + b_1 z^{-1} + b_2 z^{-2} + b_3 z^{-3}} \end{aligned} \tag{3.11}$$

Looking at the bode plot below, it can be seen that the approximated inverse magnitude plot is identical to the real inverse except for a gain shift. The phase plot also matches very closely up to the cut-off of the plant. This does not create a problem since the stability condition 2.11 involves the infinity norm of the system shown in Equation 2.11.

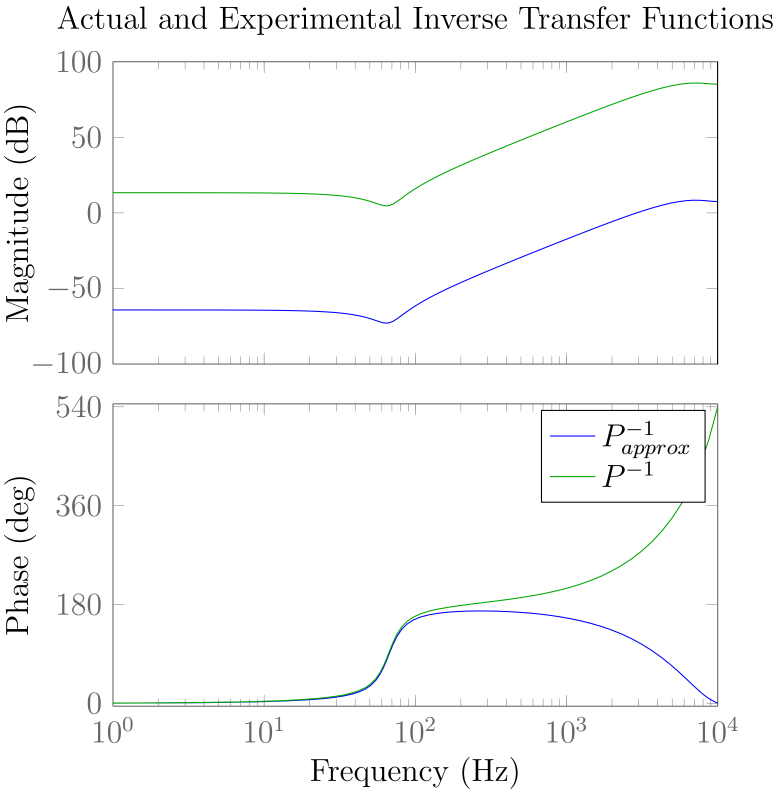

Figure 3.12: Comparison between approximate and real inverse of P

The correct gain K can be calculated by equating the approximate inverse and the exact inverse.

$$K\frac{1+a_1z^{-1}+a_2z^{-2}+a_3z^{-3}}{1+b_1z^{-1}+b_2z^{-2}+b_3z^{-3}} \approx \frac{1+a_1z^{-1}+a_2z^{-2}+a_3z^{-3}}{b_1z^{-1}+b_2z^{-2}+b_3z^{-3}} \tag{3.12}$$

Which results in the following value for K at steady state:

$$K = \frac{1+b_1+b_2+b_3}{b_1+b_2+b_3} \tag{3.13}$$

After adjusting for the correct gain the approximate inverse is very close to the real inverse with a percent error of much less than 1% when comparing the two infinity norms.

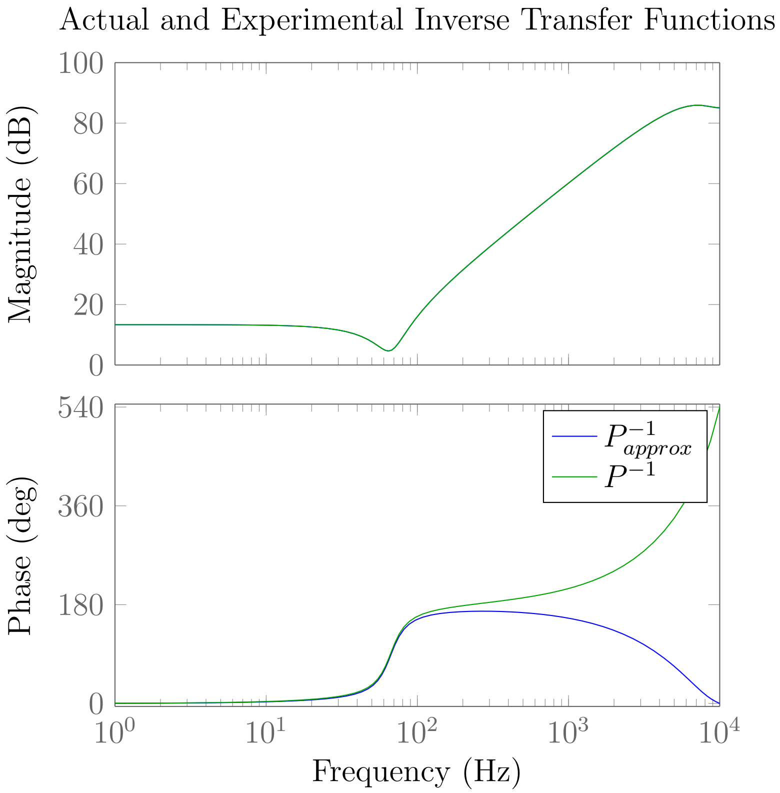

Figure 3.13: Approximate and real inverse of P after gain adjustment

After finding a way to approximate the plant inverse, a causal learning filter can be achieved using $L = QP^{-1}_{approx}$. The next step is to choose a low pass filter for Q. Using Equation 2.11 a cut-off frequency of 300 Hz gives a conservative infinity norm convergence value of 0.20. The convergence value when using the real plant inverse was found to be the exact same as with the approximate inverse. This was expected since the magnitude of both matched very well. The following figure shows the bode plot of the convergence criteria in Equation 2.11. The value of 0.20 occurs around the

470 Hz frequency. The cut off frequency of Q must be chosen to balance the amount of tracking error as well as the gain of the filter L. A higher cut off frequency results in a higher gain for the learning filter.

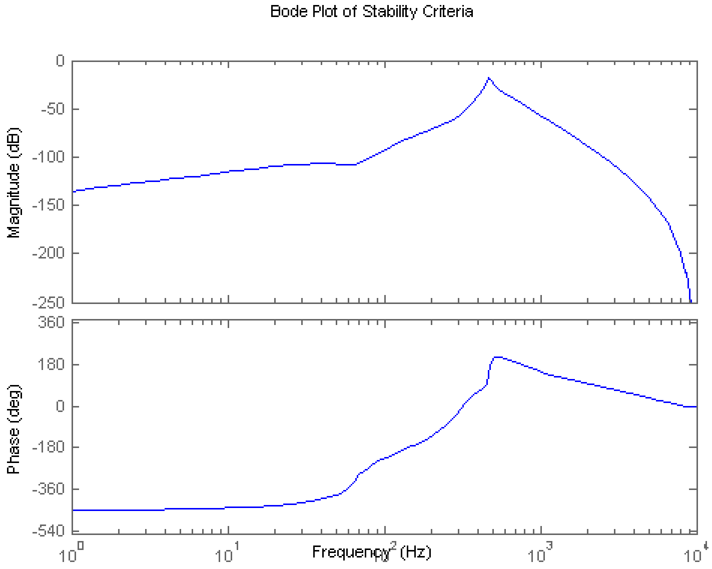

Figure 3.14: Bode plot of stability criteria in eqn. 2.11

ILC Simulations

MATLAB Simulink was used to test the controller and plant model before implementing them on the real system. The simulation was matched to the experimental system as close as possible by including a saturation term at the input of the plant model as well as a delay going to the Q filter. Three sine waves with one degree amplitudes were used as position references for testing. The RMS tracking error was then calculated and compared to the performance when only using a PID controller.

The results of the simulations are displayed in Table 3.2 below. As can be seen the tracking error when using the ILC was considerably reduced compared to only using a PID controller. The error was reduced by 93% when tracking the 5 Hz signal. By looking at the error in Figure 3.16 it's clear that the error initially starts large and then converges to the final value. Using a higher cut off for the Q filter reduced the tracking error further during the simulations, but resulted in an unstable system when applied to the experimental setup. This is most likely a result of modelling errors or non-linear behaviour of the real system.

Frequency	**PID RMS Error** μrad	**PID Peak-to-Peak Error** μrad	**ILC RMS Error** μrad	**ILC Peak-to-Peak Error** μrad
5 Hz	614.72	1813.5	43.05	119.52
10 Hz	775.91	2271.7	105.85	299.26
20 Hz	807.08	2304.8	215.07	609.95

Table 3.2: Tracking error comparison between PID and PID+ILC

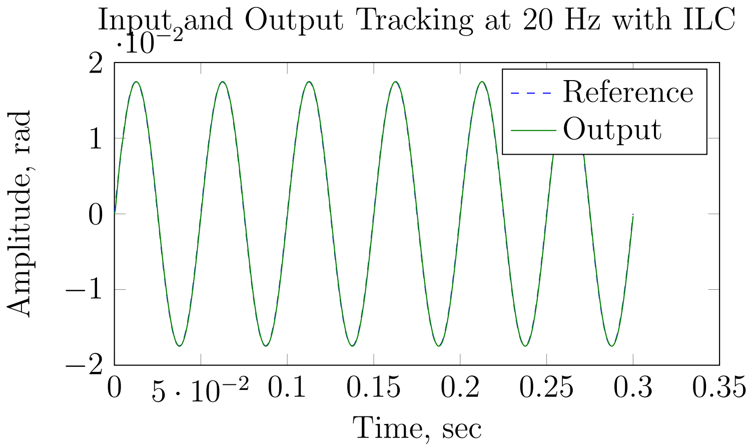

Figure 3.15: Input and Output of ILC simulation

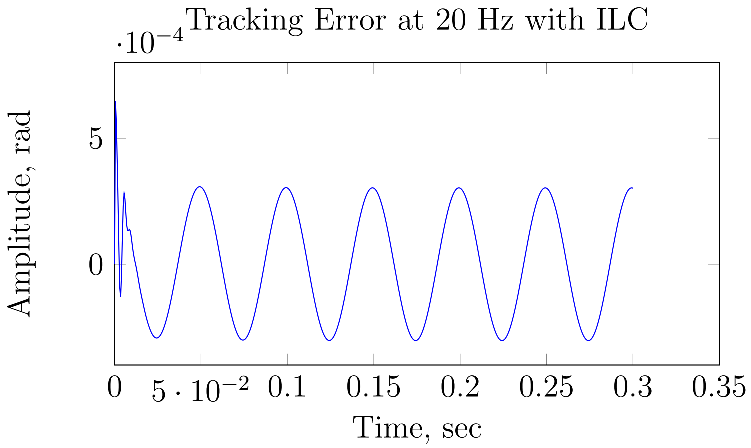

Figure 3.16: Tracking Error of ILC Controller at 20Hz

3.2.2 RST Controller

Design of the RST controller is done in the frequency domain using pole placement techniques. The closed loop poles of the controller are specified to achieve the required modulus, gain and phase margins. The polynomial P in Equation 2.16 corresponds to the desired closed loop denominator. Chapter 2 introduced the theory of designing an RST controller.

The first step is to choose dominant closed loop poles that will set the desired cut off frequency of the controller while also staying within the stability requirements of the system. Several dominant frequencies were tested with the best performance achieved by placing the dominant complex conjugate poles at 300Hz with a damping ratio of 0.8. This allows the system to have a fairly wide bandwidth which is useful for tracking higher frequency signals. This also gives a fairly damped response with little overshoot which helps in tracking as well. The fixed part of the S polynomial (H_S eqn. 2.20) was chosen as an integrator to reduce the steady state error and the T term was selected as a simple gain to ensure the correct steady state value. The resulting maximum of the sensitivity function of this controller was found to be 30.1dB which is far larger than the recommended value of 6dB. The modulus margin is the inverse of the maximum of the output sensitivity function (S_y) and directly affects the phase and gain margins of the overall system, this relationship can be easily seen in Figure 2.6. As expected the corresponding gain and phase margins of 0.27dB and 1.86°are well outside the traditionally recommended values of ≈10dB and ≥30°.

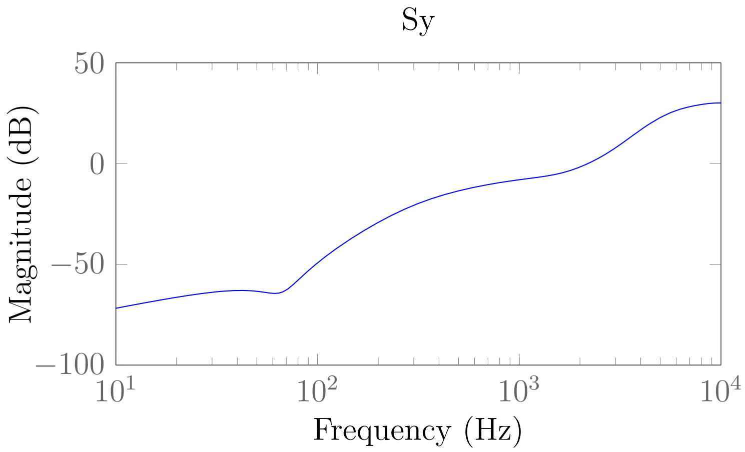

Figure 3.17: Output Sensitivity function of two pole controller

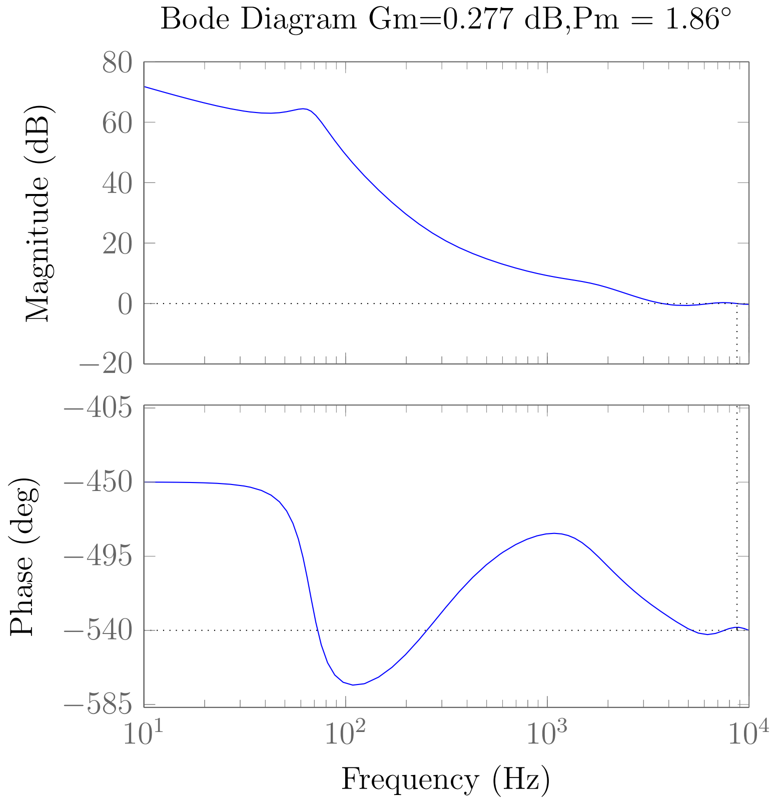

Figure 3.18: Gain and Phase margins of two pole controller

Since the maximum of the S_y function occurs at a high frequency it is possible to add real poles at the frequency where the maximum value is located. Adding two real poles at z=0.28 and 0.3 reduces the value of the maximum of S_y down to 22.4dB and shifts the maximum to a lower frequency. Landau in [1] recommends to choose real pole locations below 0.5. This ensures that they have little affect on the system dynamics since they are much faster poles than the dominant ones.

After the two auxiliary real poles have been added, the maximum of S_y has shifted close to the attenuation frequency meaning that a notch filter can be used to reduce the value further. Two notch filters: one at 2800Hz and one at 1703Hz were added to the system to increase the modulus margin further. These filters can be added to the closed loop system by using Equation 2.22 and adding the poles of the filters to the desired closed loop polynomial P. Using the Diophantine equation the polynomials R and S can be calculated to give the desired closed loop polynomial P. The resulting system has a maximum of S_y of 4.1dB, a gain margin of 9.44dB and a phase margin of 40.6°. Figures 3.19 and 3.20 show the final sensitivity function and open loop bode plot.

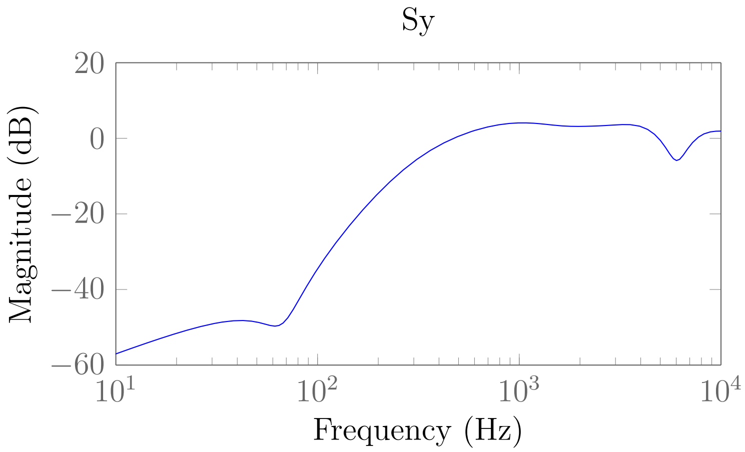

Figure 3.19: Output Sensitivity function of final RST controller

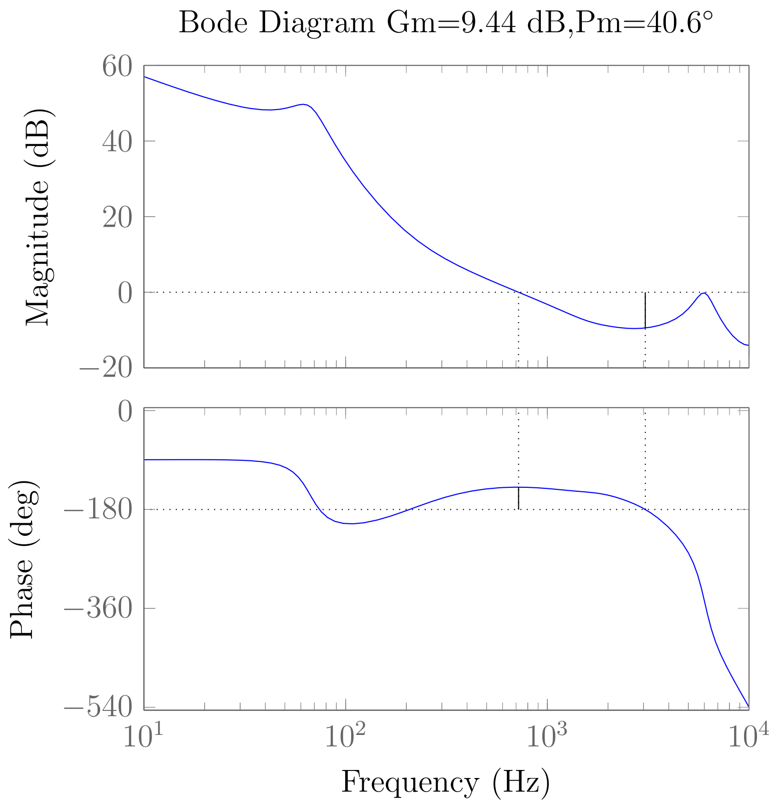

Figure 3.20: Gain and Phase margins of final RST controller

The final closed loop pole locations can be seen in Figure 3.21. The closed loop frequency spectrum of the system is mostly determined by the dominant pole locations chosen at the beginning. The extra poles are only used to improve the stability of the system and have little effect on the frequency response of the closed loop system. There are an extra two poles at the origin since the characteristic polynomial must have 10 poles for the Diophantine equation to be solved for this system [1].

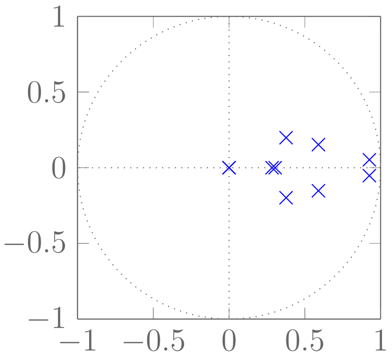

Figure 3.21: Closed loop poles of RST controller

A second RST controller was implemented which had 200 Hz dominant poles instead of 300 Hz as in the previous system, but used a tracking polynomial for T as shown in Section 2.3 Equation 2.24. For this controller the 300 Hz dominant poles from the first controller had to be reduced to 200 Hz for the controller to be stable in the real system with the addition of the T polynomial. Otherwise, the controller was designed in the same manner as the 300 Hz dominant pole one.

RST Controller Simulations

The RST controller was simulated using MATLAB Simulink with the same test signals as the ILC. Three sine waves of 1°amplitude were used as a reference: one at 5 Hz, 10 Hz, and 20 Hz. The performance of the RST controller was then compared to the performance of the basic PID controller. The performance of the RST was much better than the PID controller and was comparable to the Iterative Learning Controller.

The addition of the T polynomial did not improve the results when compared to the 300 Hz RST controller. This is most likely because the dominant poles in the controller with the T polynomial had to be reduced to 200 Hz to keep the experimental system stable.

Frequency	**PID RMS Error** μrad	**RST RMS Error** μrad	**RST T poly RMS Error** μrad
5 Hz	614.72	23.35	38.65
10 Hz	775.91	47.86	76.06
20 Hz	807.08	90.69	142.47

Table 3.3: RMS tracking error comparison between PID and RST controller

Frequency	**PID Peak-to-Peak Error** μrad	**RST Peak-to-Peak Error** μrad	**RST T poly Peak-to-Peak Error** μrad
5 Hz	1813.5	63.41	109.37
10 Hz	2271.7	124.79	215.30
20 Hz	2304.8	233.51	403.30

Table 3.4: Peak-to-Peak tracking error comparison between PID and RST controller

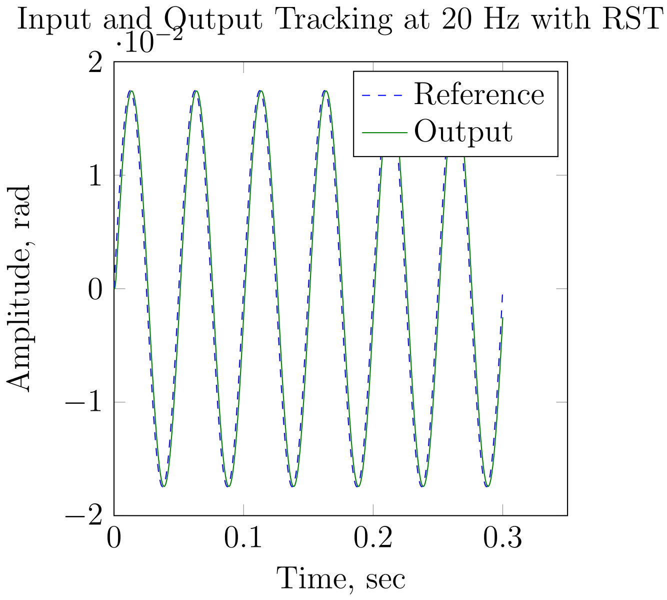

Figure 3.22: Input and Output of RST simulation

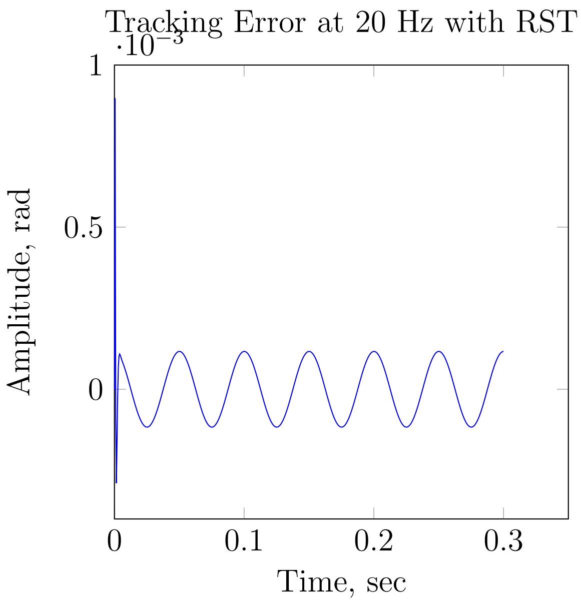

Figure 3.23: Tracking Error of RST Controller at 20Hz

3.2.3 Adaptive Feed Forward Cancellation

Design of AFC controllers is much simpler than the previous ILC and RST controllers since there are only two parameters to adjust: the resonant frequency ω_n and the phase advance ϕ_n. The choice of ω_n corresponds to the desired frequency of the disturbance that must be rejected. This is normally done by examining the FFT of the error signal and selecting the largest peaks to be attenuated. Typically there will be one large error at an initial frequency and then the next multiples of that frequency will be a slightly smaller value. The phase advance parameter can be set to the phase of the closed loop controller (without the AFC filters) at $\omega = \omega_n$ using Equation 2.31.

If the frequencies of the disturbance are known ahead of time then all the

AFC filters can be designed at once. Sometimes the addition of one AFC filter may change the major frequencies in the error signal. In this case it is useful to add one AFC filter at a time while checking the FFT of the error after each pass and deciding which frequency the next resonant filter should be placed at. Before any AFC filters can be designed, the closed loop bode plot when only a PID controller is used must be computed. Using the closed loop bode plot, the required phase advances of each AFC resonator can be selected.

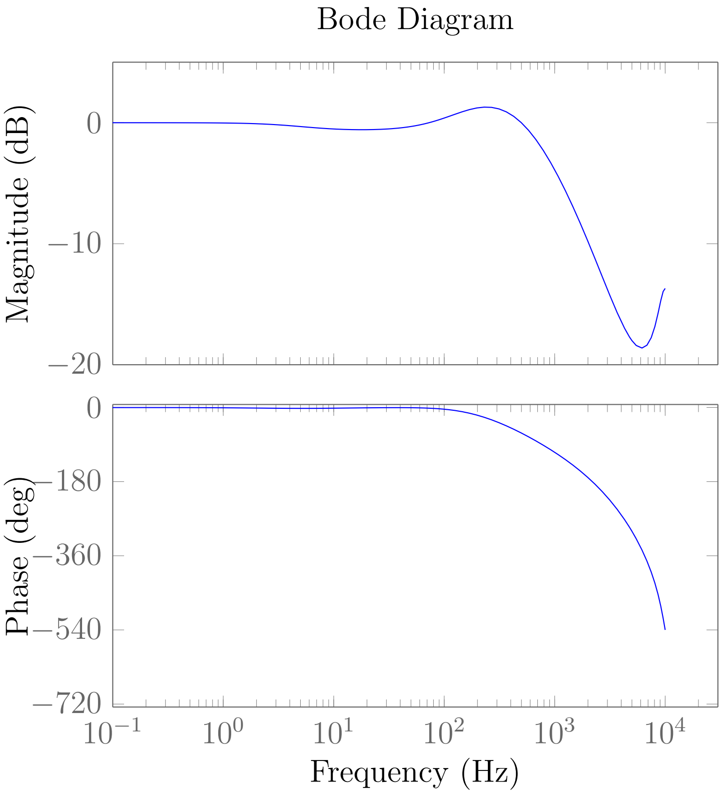

Figure 3.24: Closed Loop Bode plot of plant and PID

AFC Simulation

The desired signal to track is a 20 Hz 1°amplitude sine wave. Using Simulink the tracking error frequency spectrum using just a PID controller was simulated. As was expected the largest error is at 20 Hz. Figure 3.25 shows the open loop system with only one AFC filter enabled at 20 Hz. The phase advance parameter at this frequency was chosen as 0. This filter was designed by converting the continuous filter in Equation 2.30 to its discrete form using MATLAB.

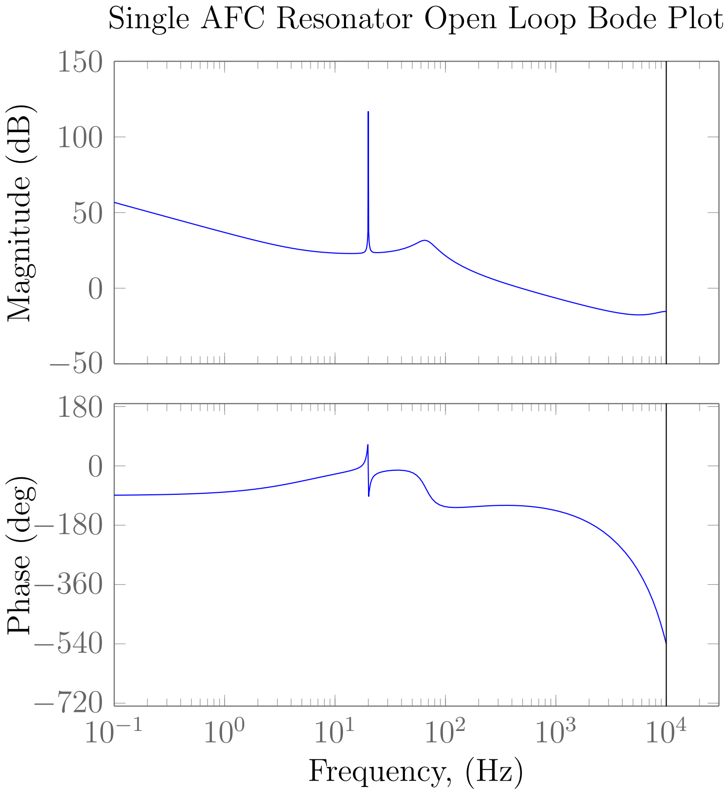

Figure 3.25: Open Loop Bode plot of one AFC resonator and PID

Using one AFC filter was enough to completely remove the tracking error at

20 Hz. It took around 1.5 seconds for the error to be reduced to a steady state RMS value of 1.5e-10 rad. The AFC controller also had comparable performance for the 5 Hz and 10 Hz sine waves.

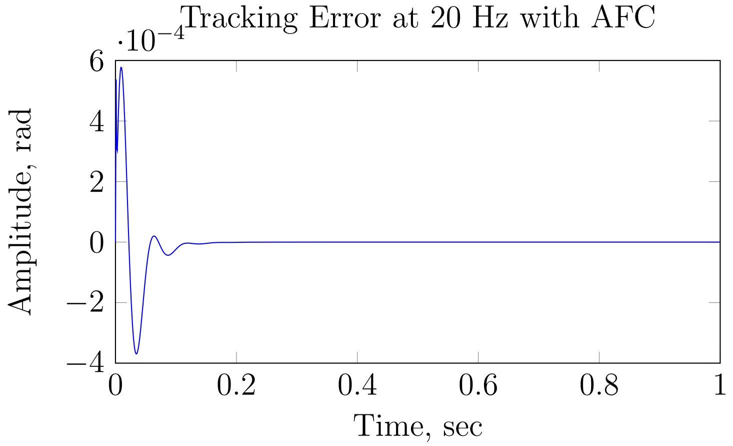

Figure 3.26: Error signal when tracking a 20 Hz reference

The experimental results are not expected to perform as well as the simulations because there are usually many more harmonics that must be removed in the real system.

3.3 Conclusions

The AFC controller performed the best of all the other systems since a sinusoidal reference was used for tracking. Because it was a simulation with no added noise or disturbances, the AFC was able reduce the tracking error to essentially zero. The RST controller was also able to get the second best results since there were no uncertainties in the mathematical model affecting the results in the simulation. Since the PID controller was tuned on the real system then implemented on the simulation it performed the worst of all the controllers. The ILC performed better than the PID but had an RMS error which was double of the RST controller.

Chapter 4

Experimental Results and Comparisons

The following chapter will outline the experimental results obtained with the different controllers and compare them to their simulations. The experimental setup as well as some implementation details will be outlined. Finally the different controllers will be compared with each other and a combination of the ILC and AFC controller will be tested as well.

4.1 Experimental Set-up

A Polaris Controller and Amplifier combination was used to control the two axis laser scanning system. The laser scanning system consists of two Canon Galvanometers mounted on an aluminum block. Figure 4.1 shows this experimental set-up. The Canon galvanometers have a range of ± 0.35 rad for a total range of 0.7 rad. The actuators have a built-in optical encoder with a resolution of 0.77 μrad.

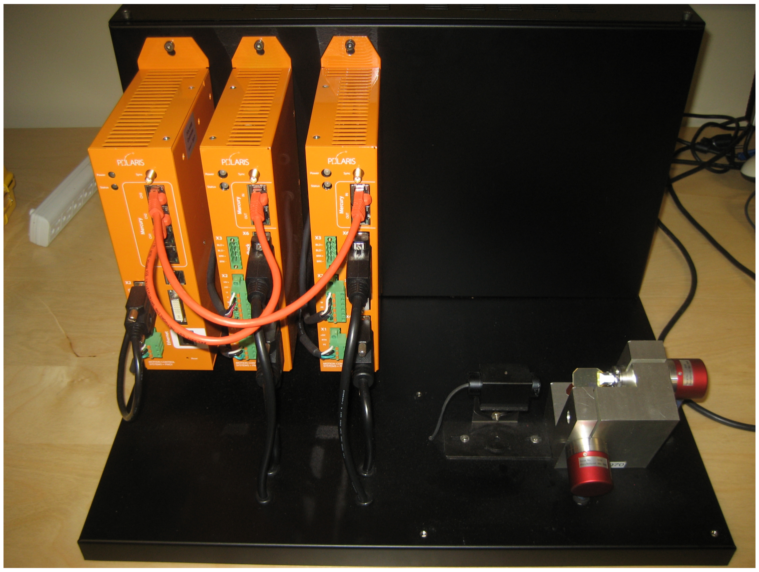

Figure 4.1: Controller, amplifiers and galvanometer set-up

The Polaris Controller is responsible for calculating trajectories and sending position, velocity and acceleration set points to the Polaris PWM amplifier. The amplifier contains an FPGA which has a custom made SPU (Servo Processing Unit) which is capable of double floating point addition, subtraction, multiplication as well as other common CPU operations like branching and comparing. The FPGA also supports up to eight 15th order real time digital IIR filter blocks which can be programmed with any coefficients. All controllers are implemented using a custom assembly language on the SPU. Simple controllers such as the PID are implemented using a difference equation in assembly code and the more complicated ones such as the AFC, ILC, and RST controllers are implemented using the dedicated hardware filters. The filters can be easily designed using MATLAB or Octave and copied onto the SPU. Since the SPU supports double floating point precision, filter coefficient quantization is not an issue.

The following experimental results are based on using only one of the galvanometers.

4.2 Baseline PID Controller Experimental Results

The first controller tested was the standard PID controller. This controller was designed using Table 2.1 from Chapter 2 and was hand tuned by adjusting the PID gains in real time until the desired performances were achieved. The goal with the PID tuning was to achieve the fastest possible response with little overshoot and quick settling time. These general requirements would result in a fast controller which allowed good tracking results. The proportional, integral and derivative gains were set as high as possible while keeping the system stable.

One of the draw backs of the PID implemented in the hardware controller was that the derivative term was not low pass filtered. A high derivative gain was used to keep the system from oscillating but the high gain resulted in a fairly noisy drive signal. The derivative gain was kept high since the tracking was still better with it than without it.

The same tests performed in the simulations were repeated on the experimental system. Three 1° sine waves were used as reference signals: a 5 Hz, 10 Hz and 20 Hz frequency wave was used. The following table shows the experimental results. The peak to peak value was calculated by subtracting the absolute minimum value from the absolute maximum value; therefore it corresponds to the worst error in the data.

Frequency	PID Sim RMS Error μrad	PID Sim Peak-to-Peak Error μrad	PID RMS Error μrad	PID Peak-to-Peak Error μrad
5 Hz	614.72	1813.5	464.25	2086.7
10 Hz	775.91	2271.7	576.13	2021.8
20 Hz	807.08	2304.8	798.80	2182.7

Table 4.1: Tracking error of PID controller

The experimental values are considerably better than the simulated results with a percentage improvement of 24%, 25% and 1% for the 5 Hz, 10 Hz and 20 Hz signals respectively. The differences are most likely due to plant modelling errors in the simulation. The PID tuning in the simulation was found by tuning the real system and then applying that tuning to the simulation. Since both the simulation and real system used the same PID gains, it shows that there are some unmodelled dynamics in the simulation even though the model had a 98% fit. Figure 4.2 shows the experimental tracking error when using a PID controller.

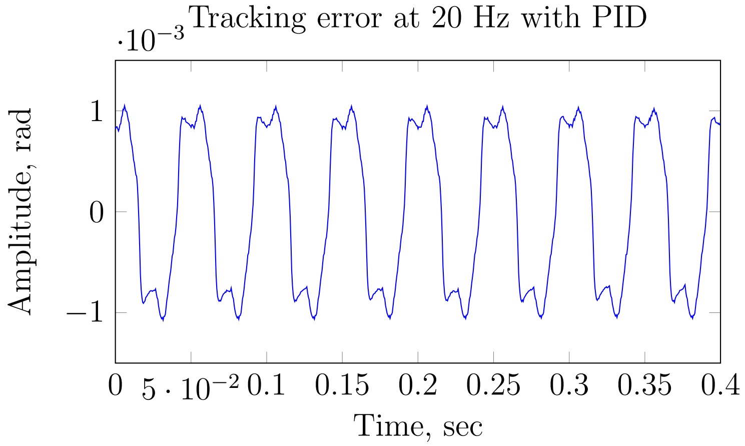

Figure 4.2: Error signal when tracking a 20 Hz reference with PID

The error signal does not appear to be a clean sinusoidal wave like the simulated error. This is most likely due to the noise added by the derivative term and the nonlinearities of the galvanometer.

4.3 ILC Experimental Results

The Iterative Learning Controller designed and simulated in Chapter 3 was implemented on the real galvanometer system. The ILC performed considerably better than the experimental PID but not as good as the ILC simulations. Table 4.2 below shows the experimental results.

Frequency	**ILC Sim RMS Error** μrad	**ILC Sim Peak-to-Peak Error** μrad	**ILC RMS Error** μrad	**ILC Peak-to-Peak Error** μrad
5 Hz	43.05	119.52	139.58	1254.2
10 Hz	105.85	299.26	175.39	1522.1
20 Hz	215.07	609.95	367.87	2099.1

Table 4.2: Tracking error of ILC controller

The experimental results of the ILC controller were 224%, 66% and 71% worse than the simulated values. This is most likely due to the increased noise in the real system that did not get completely removed as well as errors in the learning filter L. The Q filter bandwidth also has a major affect on the error since the system cannot learn past the cut off frequency of 300 Hz. A Q filter with a 400 Hz cut off was tested in simulations and showed to have considerably better results but it was not stable when applied to the real system.

Looking at the error plots of the simulated and experimental tests it can be seen that there are many more harmonics in the experimental system. The simulation had a much cleaner error signal which was a pure 20 Hz oscillation at steady state. Further differences between the two experiments may be due to the fact that the inverse is approximated using a plant model which could have errors, this would compound any errors when designing the learning filter. Since the ILC controller should be robust to modelling uncertainties it still considerably improved the performance of the galvanometer system when comparing it to using only a PID controller.

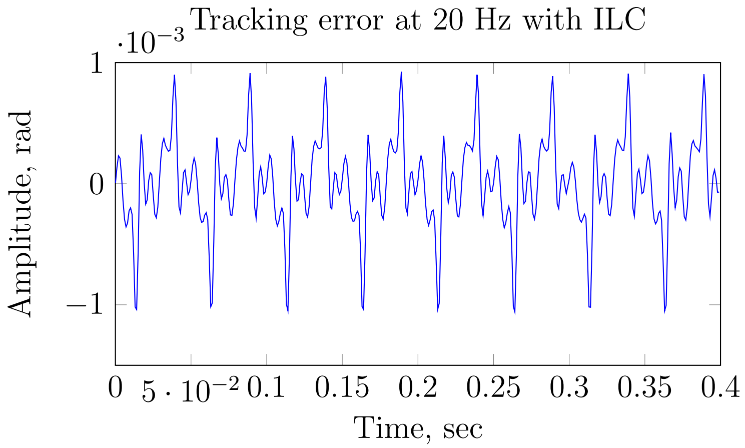

Figure 4.3: Error signal when tracking a 20 Hz reference with ILC

4.4 RST Controller Experimental Results

The next controllers implemented were the two RST controllers simulated in Chapter 3. The first one had dominant poles at 300 Hz with T set as a gain like in Equation 2.23 and the second controller had dominant poles at 200 Hz and T was set as a polynomial as shown in Equation 2.24.

The previous tests in Sections 4.2 and 4.3 have shown that there are some modelling uncertainties which have affected the performance of the previous controllers. Of all the controllers used in this work, the RST is the most dependent on an accurate model. Table 4.3 shows the results when using the first RST controller with 300 Hz dominant poles. Because of the model uncertainties it performed the worst out of all controllers tested.

Frequency	**RST Sim RMS Error** μrad	**RST Sim Peak-to-Peak Error** μrad	**RST RMS Error** μrad	**RST Peak-to-Peak Error** μrad
5 Hz	23.35	63.41	2399.4	7138.2
10 Hz	47.86	124.79	4792.2	13772.0
20 Hz	90.69	233.51	9536.6	27078.6

Table 4.3: Tracking error of RST controller

Since the RST controller is a pole placement controller, the location of those poles depend heavily on dynamics of the real system. It appears that the resulting closed loop polynomial is not close to the simulated system.

Looking at Figure 4.4 it can be seen that there is a significant amount of phase delay in the signal which is partly why there is such a large tracking error. Delaying the reference to counteract the delay is possible but the galvanometer is normally used in coordinated motion with several other axes. To correctly coordinate the delay necessary would be very difficult. The RST controller can be modified to include two undamped complex zeros at the resonant frequency as part of the fixed part H_S (eqn. 2.19) to act the same way as the AFC controller, but when this was added the resulting RST controller was unstable.

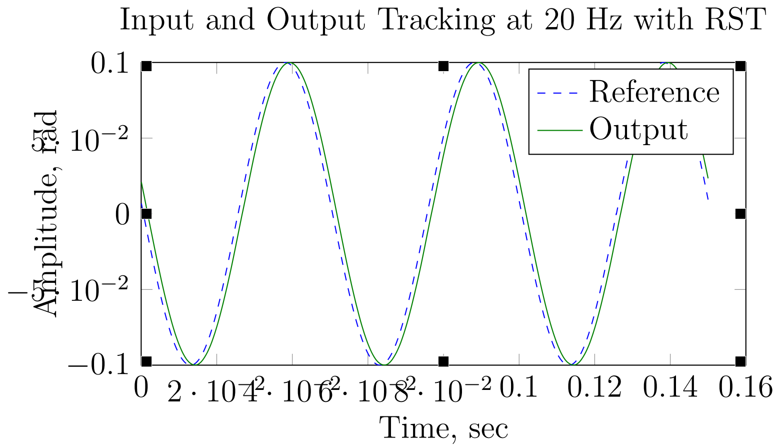

Figure 4.4: Input and output signal at 20 Hz with RST

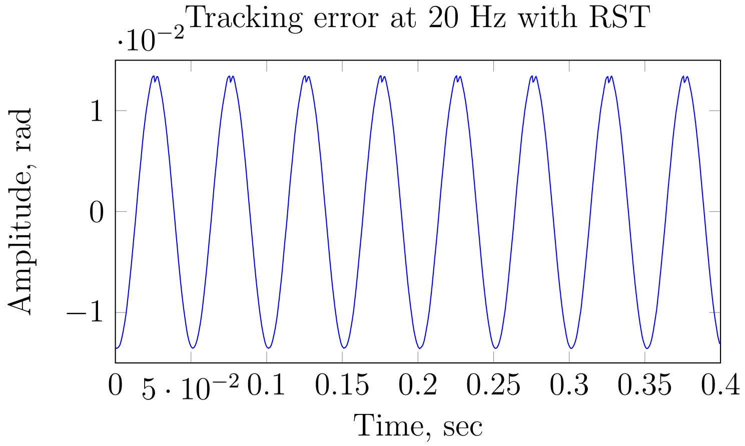

Figure 4.5: Error signal when tracking a 20 Hz reference with RST

The second RST controller with the T polynomial performed significantly better than the 300 Hz RST controller since it removed the majority of the phase delay as can be seen in Figure 4.6. It also had a smaller RMS error than the baseline PID controller, but the higher peak-to-peak error because of the overshoot at the top of each sine wave.

Frequency	RST Sim RMS Error μrad	RST Sim Peak-to-Peak Error μrad	RST RMS Error μrad	RST Peak-to-Peak Error μrad
5 Hz	38.65	109.37	179.13	2205.6
10 Hz	76.06	215.30	374.63	3128.0
20 Hz	142.47	403.30	571.05	5931.9

Table 4.4: Tracking error of RST with T polynomial controller

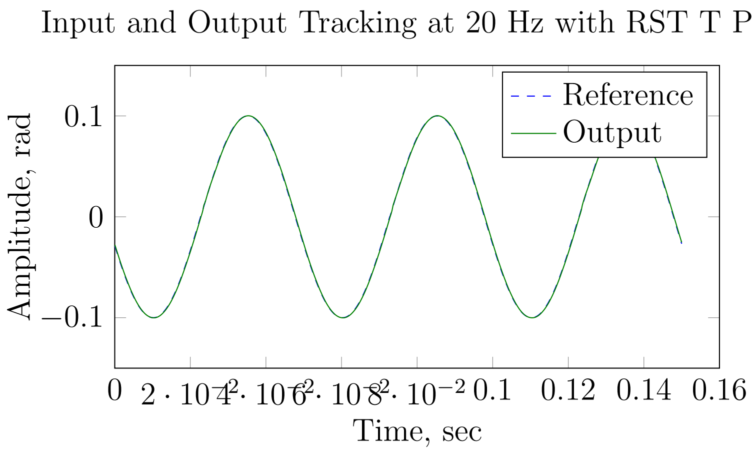

Figure 4.6: Input and output signal at 20 Hz with RST T Poly

The error plot of the RST controller with T polynomial is shown below. The galvanometer position crosses the reference signal and overshoots a the peak of the sine wave which causes the sudden drop in the tracking error.

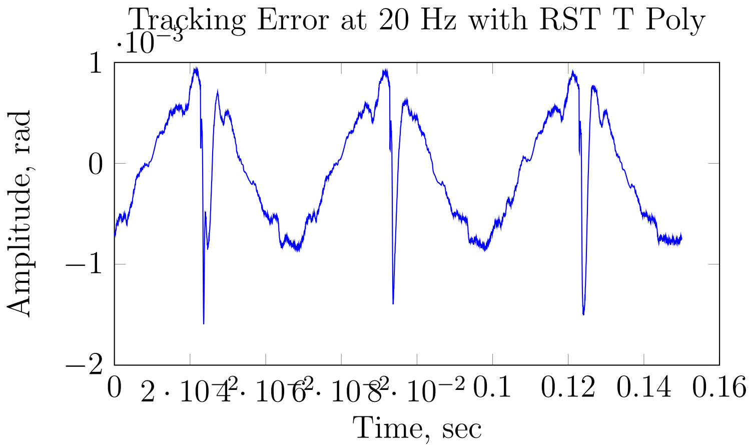

Figure 4.7: Error signal when tracking a 20 Hz reference with RST T Poly

4.5 AFC Controller Experimental Results

The AFC controller used for the experiments was a little different than the one used for the simulations in Chapter 3. In the simulations only one AFC filter was necessary since there was no noise or harmonics in the system. The experimental AFC used four AFC filters each with a gain of 300 and a phase advance of 0. The phase of the closed loop system was small at the frequencies of interest therefore ϕ_n could be set to zero using Equation 2.31. The centre frequency was chosen by sending the reference signal without any AFC filters present and observing the FFT plot of the error. The frequency of the largest peak was chosen as the frequency of the first AFC filter. The frequencies of the other filters were also chosen in the same way; the reference was sent and the FFT of the error was measured after each AFC filter was added.

The large gain was chosen so that each AFC filter worked in a broader frequency band as explained in Section 2.3. Each input signal had a different set of AFC filters. The final frequencies for the AFC filters were set as 5 Hz, 15 Hz, 20 Hz and 25 Hz for the 5 Hz input signal; 10 Hz, 20 Hz, 30 Hz and 50 Hz for the 10 Hz signal; 20 Hz, 60 Hz, 100 Hz and 140 Hz for the 20 Hz input signal. In all cases the filter frequencies chosen were multiples of the input signal.

The following table compares the experimental and simulated values of the AFC controllers. For the simulation only one AFC filter was necessary since there was no noise or other harmonics in the error, but all four were used in the experimental tests.

Frequency	**AFC Sim RMS Error** μrad	**AFC Sim Peak-to-Peak Error** μrad	**AFC RMS Error** μrad	**AFC Peak-to-Peak Error** μrad
5 Hz	2.057e-11	1.638e-10	42.58	494.81
10 Hz	9.402e-12	2.935e-11	56.51	535.77
20 Hz	5.059e-12	1.504e-11	46.66	409.74

Table 4.5: Tracking error of AFC controller

Looking at Figure 4.8 there still seems to be large peaks at around 20 Hz intervals which were attenuated but not completely removed.

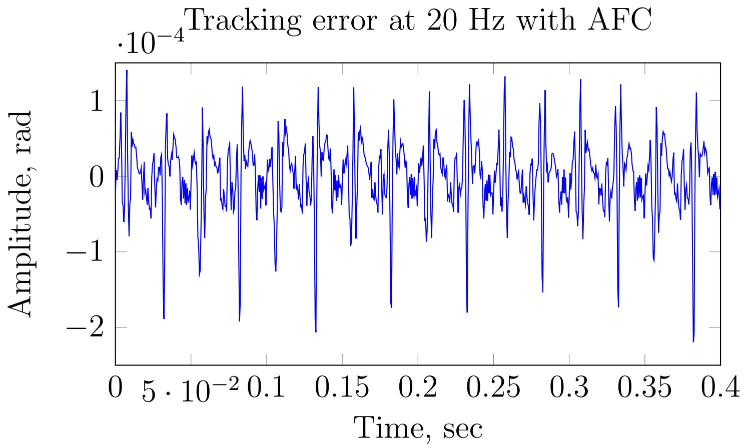

Figure 4.8: Error signal when tracking a 20 Hz reference with AFC

The experimental AFC filter worked very well but was not nearly as accurate as the simulation. This is due to the increased noise and harmonics in the real system that were not present in the simulations. Theoretically one could add more AFC filters to the real system until all noise and harmonics were removed but this would not practically be possible because of the space limitations for hardware filters in the FPGA.

4.6 AFC and ILC Combined Experimental Results

After performing the ILC experiment there still was a very periodic error signal present. This suggests combining the ILC controller with some AFC filters to remove the resulting periodic error. Only three AFC filters with much smaller gains than the pure AFC controller were used. The addition of a fourth AFC filter made the system unstable; therefore it was not used. The results using this combination are shown in Table 4.6 below.

Frequency	**ILC+AFC RMS Error** μrad	**ILC+AFC Peak-to-Peak Error** μrad	**ILC RMS Error** μrad	**ILC Peak-to-Peak Error** μrad
5 Hz	115.89	1258.5	139.58	1254.2
10 Hz	160.22	1209.2	175.39	1522.1
20 Hz	186.22	1173.4	367.87	2099.1

Table 4.6: Tracking error of ILC+AFC controller

As expected, the addition of the AFC filters improved the tracking response of the purely ILC controller. For the 5 Hz and 10 Hz signals there was not a very large improvement but there was an improvement of 49% for the 20 Hz signal.

The large 20 Hz peaks present in Figure 4.3 of the ILC controller were reduced by 54% with the addition of the AFC filter, which coincides with the reduction of the RMS and Peak-to-Peak errors.

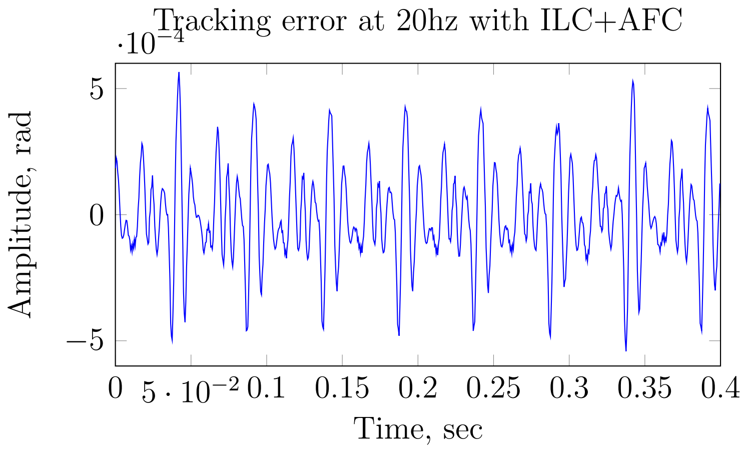

Figure 4.9: Error signal when tracking a 20 Hz reference with ILC+AFC

The results of the combined ILC and AFC controller were considerably better than using only an ILC controller, but it did not improve upon the performance of only using the AFC filters by themselves.

4.7 Comparisons and Conclusions

With the exception of the PID, all of the experimental performances were worse than their simulated performances. By looking at Table 4.7 it can be seen that the controllers which were heavily dependent on a model of the galvanometer performed worse than controllers which could be tuned online.

In the simulated system the AFC performed the best which was also the case for the experimental controller.

Frequency	**PID RMS Error** μrad	**AFC RMS Error** μrad	**ILC RMS Error** μrad	**ILC+AFC RMS Error** μrad	**RST T Poly RMS Error** μrad
5 Hz	464.25	42.58	139.58	115.89	179.13
10 Hz	576.13	56.51	175.39	160.22	374.63
20 Hz	798.80	46.66	367.87	186.22	571.05

Table 4.7: Tracking error comparison of the different controllers

Although there were modelling errors the ILC was still able to perform better than the PID and RST controllers. This shows the power of learning controllers which can adapt to meet the requirements of the real system. The addition of the AFC filters further improved the performance of the ILC controller. The ILC controller seemed to make the error signal contain more harmonics when compared to the PID only controller which made it harder for the AFC filters to completely remove the rest of the error. The AFC controller worked the best since the user was able to systematically remove the maximum errors in the system; but there was still noise and harmonics in the error which could not be completely removed since there were only four AFC filters available.

Chapter 5

Conclusion and Future Work

In this thesis several controllers were simulated and then implemented onto an FPGA processor. Their performances were compared with each other and with their expected simulated results. All controllers except the PID performed better in the simulation due to system noise and modelling uncertainties that were not represented in the model.

Almost all controllers were able to improve on the basic PID and the controllers which were not fully dependent on an accurate galvanometer model performed the best. The AFC controller had the best results since it was possible to systematically remove disturbances in the system, essentially tuning it to the requirements of the trajectory. The ILC was the second best since it was a learning controller and was not completely reliant on the system model. The ILC controller was able to improve on the error with each iteration. The RST controller had one of the best simulated performances but the worst experimental performance. Since the RST controller is a pole placement controller, the accuracy of the model is very important for good results.

5.1 Future Work

There are some possibilities for future work to implement these algorithms on a two axis galvanometer system and test the final combined path. The next step would be to use all the algorithms on a four axis CNC laser machine, then examine the overall performance of all the four axes working in unison.

Although the galvanometer model had a very good fit to the experimental data it would be interesting to know if using a different modelling technique could improve the performance of the RST controller. Or if reducing the noise in the system further by using a band limited differentiator could improve the results of the other controllers.

Bibliography

[1] I.D. Landau and G. Zito. *Digital Control Systems: Design, Identification and Implementation.* Communications and Control Engineering Series. Springer-Verlag London, Ltd., 2006.

[2] Jing Feng and Jiang Zhi. Digital implementation of a galvanometric optical scanner based on dsp and fpga. In *Mechatronic Science, Electric Engineering and Computer (MEC), 2011 International Conference on*, pages 1899–1902, Aug 2011.

[3] Hyo-Sung Ahn, Yang-Quan Chen, and K.L. Moore. Iterative learning control: Brief survey and categorization. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 37(6):1099–1121, Nov 2007.

[4] Yang-Quan Chen, K.L. Moore, Jie Yu, and Tao Zhang. Iterative learning control and repetitive control in hard disk drive industry - a tutorial. In *Decision and Control, 2006 45th IEEE Conference on*, pages 2338–2351, Dec 2006.

[5] Suguru Arimoto, Sadao Kawamura, and Fumio Miyazaki. Bettering operation of robots by learning. *Journal of Robotic Systems*, 1(2):123–140, 1984.

[6] M. Phan and R. LONGMAN. A mathematical theory of learning control for linear discrete multivariable systems. In *AIAA/AAS Astrodynamics Conference, Minneapolis, MN*, pages 740–746, 1988.

[7] I. Rotariu, B.G. Dijkstra, and M. Steinbuch. Comparison of standard and lifted ilc on a motion system. In IFAC, editor, *3rd IFAC Symposium on Mechatronic Systems*, pages 211–216, 2004.

[8] B.G. Dijkstra and O.H. Bosgra. Extrapolation of optimal lifted system ilc solution, with application to a waferstage. In *American Control Conference, 2002. Proceedings of the 2002*, volume 4, pages 2595 – 2600 vol.4, 2002.

[9] Kevin L. Moore. *Iterative Learning Control for Deterministic Systems.* Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1993.

[10] M. H. A. Verwoerd. *Iterative learning control: a critical review.* PhD thesis, University of Twente, The Netherlands, January 2005.

[11] M. K. Maaziz, P. Boucher, and D. Dumur. A new rst cascaded predictive control scheme for induction machines. In *Control Applications, 1999. Proceedings of the 1999 IEEE International Conference on*, volume 2, pages 927–932 vol. 2, 1999.

[12] M. Khanchoul and M. Hilairet. Design and comparison of different rst controllers for pmsm control. In *IECON 2011 - 37th Annual Conference on IEEE Industrial Electronics Society*, pages 1795–1800, 2011.

[13] M.B. Camara, H. Gualous, F. Gustin, A. Berthon, and B. Dakyo. Dc/dc converter design for supercapacitor and battery power management in hybrid vehicle applications #x2014;polynomial control strategy. *Industrial Electronics, IEEE Transactions on*, 57(2):587–597, 2010.

[14] I.-D. Landau, J. Langer, D. Rey, and J. Barnier. Robust control of a 360 deg; flexible arm using the combined pole placement/sensitivity function

shaping method. *Control Systems Technology, IEEE Transactions on*, 4(4):369–383, 1996.

[15] B.A. Francis and W.M. Wonham. The internal model principle of control theory. *Automatica*, 12(5):457 – 465, 1976.

[16] M. Bodson, Alexei Sacks, and P. Khosla. Harmonic generation in adaptive feedforward cancellation schemes. *Automatic Control, IEEE Transactions on*, 39(9):1939–1944, Sep 1994.

[17] Marten F. Byl, Stephen J. Ludwick, and David L. Trumper. A loop shaping perspective for tuning controllers with adaptive feedforward cancellation. *Precision Engineering*, 29(1):27–40, Jan 2005.

[18] Xiaoqin Zhou and Longfei Hu. An improved adaptive feedforward cancellation for tool trajectory tracking in diamond turning of freeform optics. In *Mechanic Automation and Control Engineering (MACE), 2010 International Conference on*, pages 3590–3593, June 2010.

[19] W. Messner and M. Bodson. Design of adaptive feedforward controllers using internal model equivalence. In *American Control Conference, 1994*, volume 2, pages 1619–1623 vol.2, June 1994.

[20] Xiaodong Lu. *Electromagnetically-Driven Ultra-Fast Tool Servos for Diamond Turning*. PhD thesis, Massachusetts Institute of Technology, 2005.

[21] Kiam Heong Ang, G. Chong, and Yun Li. Pid control system analysis, design, and technology. *Control Systems Technology, IEEE Transactions on*, 13(4):559–576, July 2005.

[22] Lennart Ljung. *System Identification (2Nd Ed.): Theory for the User*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1999.

[23] Ioan Dor Landau and Alireza Karimi. Recursive algorithms for identification in closed loop: A unified approach and evaluation. *Automatica*, 33(8):1499 – 1523, 1997.

[24] Pierre J. Brosens. Scanning speed and accuracy of moving magnet optical scanners. *Optical Engineering*, 34(1):200–207, 1995.

[25] Theodore Mirtchev, Robert Weeks, and Sergey Minko. Optimizing the feedback control of galvo scanners for laser manufacturing systems. *Proc. SPIE*, 7750:77500T–77500T–7, 2010.