IMPLEMENTATION OF FOURIER REPETITIVE CONTROL IN LABVIEW

A Thesis
Submitted to the Graduate Faculty
of the
North Dakota State University
of Agriculture and Applied Science

By

Fan He

In Partial Fulfillment of the Requirements
for the Degree of
MASTER OF SCIENCE

Major Department:
Electrical and Computer Engineering

April 2014

Fargo, North Dakota

North Dakota State University
Graduate School

Title

Implementation of Fourier Repetitive Control in LabVIEW

By

Fan He

The Supervisory Committee certifies that this ***disquisition*** complies with North Dakota State University's regulations and meets the accepted standards for the degree of

MASTER OF SCIENCE

SUPERVISORY COMMITTEE:

Dr. Jacob Glower
Chair

Dr. Mark Schroeder
Co-Chair

Dr. Dan Ewert

Dr. Lawrence Reynolds

Approved:

04/01/2014
Date

Dr. Rajesh Kavasseri
Department Chair

ABSTRACT

This thesis presents position control as well as current (torque) control of a DC servo motor with a periodic input. Position control is achieved by using Fourier repetitive control scheme. Simulation on position control is carried out and results discussed.

Experiment is executed based on a testing platform that is built using two, mechanically linked DC servo motors and LabVIEW to incorporate the Fourier repetitive control algorithms. The experiment results validate the feasibility of controlling motor positions via Fourier repetitive control scheme. Suggestions on improving the implementation of Fourier repetitive control theory are also made. Experiment results on current (torque) control are presented. The controlling of both position and current of the motors at the same time is also demonstrated.

ACKNOWLEDGEMENTS

Foremost, I would like to express my sincere gratitude to my advisor Dr. Glower, for his dedication and continuous help in my research and thesis writing. I highly appreciate the encouragement from him that inspires me achieving more. I would also like to thank my co-advisor Dr. Schroeder in helping me in every aspect throughout my graduate study.

I am thankful for Jeff Erickson's dedication in helping arranging instruments and building the test platform.

I highly appreciate my supervisory committee members Dr. Ewert and Dr. Reynolds for their time and effort in guiding my work.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDIX FIGURES

CHAPTER 1. INTRODUCTION

1.1. Introduction

In this thesis, the problem of implementing a control scheme which controls both the position as well as the current (torque) of a DC servo motor with a periodic input is investigated. There are several reasons why this problem is important. First, Left Ventricular Assist Devices (LVADs) have been proposed as a means of helping failing hearts recover from injury. These devices accept the flow of blood from the heart and, in turn, pump the blood to the rest of the body, as presented in Figure 1. In this way, the heart continues to work, avoiding heart atrophy while the majority of the work is off-loaded to the AVD. Moreover, the work done by the heart can be controlled and modified, allowing for the heart muscle to be exercised and strengthened – with the hope of eventual healing of an injured heart muscle [Dipla 1998] [Dandel 2005] [Girdharan 2012] [Girdharan 2004] [Goldenstein 1998] [Hetzer 2001] [Levin 1995] [Rose 2001] [Schmid 1999] [Zafeiridis 1998].

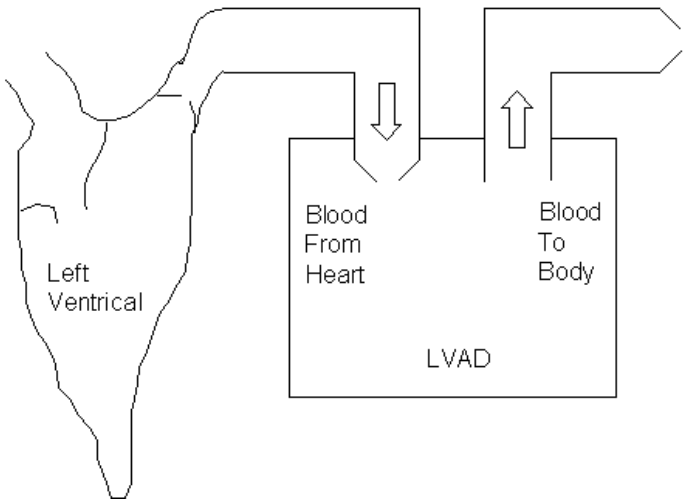

Figure 1. Left Ventricular Assist Device (LVAD) which receives blood from the heart and pumps blood to the body.

Since the heart beat is essentially a periodic signal, the LVAD needs to provide a periodic pressure while accepting a periodic flow from the heart.

A second application is in the design of artificial stents: a tube to replace a section of artery. To test an artificial stent, one needs to test under conditions it will see in practice: i.e. a periodic pressure and flow resulting from the heart beat.

There are numerous methods that have been proposed to control both pressure and flow – these are summarized in Chapter 2. One which shows significant promise in simulation is to use two motors: one of which controls the flow, the other controls the pressure at a specified location as presented in Figure 2 [Girdharan 2012].

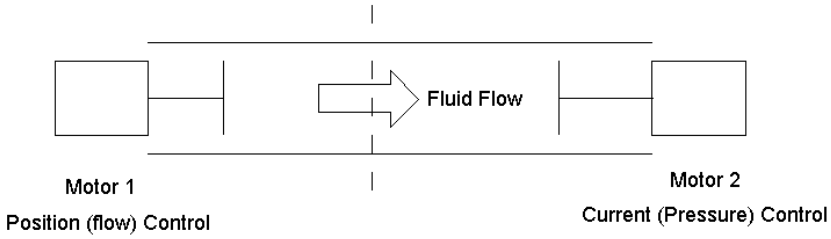

Figure 2. Use of two motors to control both flow and pressure.

An equivalent control objective is to control motor angle and current. Angle is the integral of speed (flow). By controlling angle, you can also control speed (and hence flow when connected to a pump.) Current is related to torque through the motor's torque constant. Likewise, controlling current is analogous to controlling torque.

Since both pressure and flow are periodic for steady-state flow from the heart, a controller which can both

- Track periodic set points, and
- Reject periodic disturbances with the same period

is required. One such controller which shows promise is Fourier Repetitive Control [Glower 1995]. Repetitive Control is a type of model-reference adaptive control where the set point is periodic with a known period [Hara 1988]. Using this knowledge, a periodic input can be determined by estimating the weightings for a set of periodic basis functions. [Glower 1995]. Fourier Repetitive Control is simply a Repetitive Control scheme which uses sine and cosine functions for this basis function as presented in Figure 3.

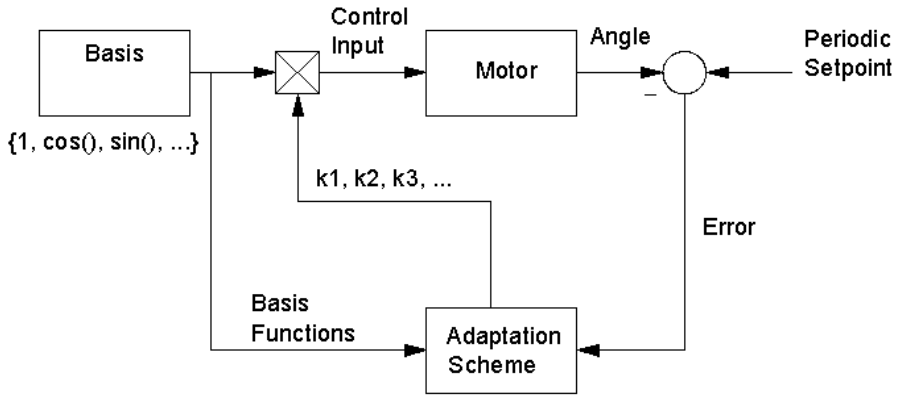

Figure 3. Fourier repetitive controller configuration.

The advantage of such a scheme is that

- Periodic set points can be tracked with no error so long as the input required for tracking (termed the "perfect model following conditions" [Landau 1979]) are satisfied.
- Periodic disturbances can be rejected as well, again so long as the perfect model following conditions are satisfied.

Fourier Repetitive Control has been demonstrated in simulation studies [Glower 1995b] [Tsao 2000]. It has not, however, been demonstrated in hardware while are also tracking a desired force (current) profile. This thesis intends to remedy this.

1.2. Thesis Statement

In this thesis, the feasibility of controlling both the angle and current to a pair of coupled DC servo motors is to be demonstrated. Specifically, the position of the motor will be through a hardware implementation of a Fourier Repetitive Control scheme through LabVIEW. The tracking accuracy and modifications required for implementing these schemes in hardware will also be noted.

1.3. Thesis Outline

This thesis is organized as follows. In Chapter 1, the motivation for controlling both position and current is presented along with the objective for this thesis.

In Chapter 2, previous work for controlling position and force is presented.

In Chapter 3, a Fourier Repetitive Control scheme is implemented on a DC servo motor and in simulation. The ability of this scheme to track periodic inputs and reject periodic disturbances is presented.

In Chapter 4, a current controller for the second DC servo motor is presented. The ability of the controller to track periodic set points and reject periodic disturbances is also demonstrated in hardware.

In Chapter 5, the position and current controllers are used at the same time, demonstrating that both can be controlled in hardware.

Chapter 6 summarizes this work and conjectures on where this work will lead in the future.

This is followed by the references used in this work.

Appendix A describes a Left Ventricular Assist Device in detail.

Appendix B presents modeling of a DC Servo Motor.

CHAPTER 2. PREVIOUS WORK

2.1. Introduction

In this thesis, the problem of controlling in hardware both the angle of a DC servo motor as well as the current (torque) produced by that motor is investigated. This is similar to the problem of controlling pressure and flow from an LVAD or through an artificial stent. Over the past few decades, several approaches have been proposed to do this.

One approach is termed Impedance Control. With this approach, rather than prescribing the absolute position or torque from a motor, the differential equation relating the two is to be controlled. These techniques are presented in Section 2.2.

A second approach is termed Hybrid Control. Here, the controller switches between position and torque control. This technique is presented in Section 2.3.

A third approach uses two separate motors, coupled together. One of the motors is used to control the position while the other controls the torque. This is the technique used in this thesis and is presented in Chapters 3-5.

2.2. Impedance Control

Impedance control is a relatively new area of controls systems dating back to 1985 [Hogan 1985]. Several approaches have been taken to date. These include attempts to modify a position controller so that it regulates impedance [Hogan 1985], attempts to modify a force controller so that it regulates impedance [Volpe 1993], attempts to switch between position and force controllers (termed hybrid control) [Anderson 1988] [Lee 1991], and model following controls. In this chapter, a brief description of these designs along with their shortcomings will be presented.

2.2.1. Introduction

Several types of feedback control are commonly used. In position control, a plant with a strictly proper transfer function is to be controlled. Examples include controlling the angle or speed of a motor, the temperature in a room, etc. In force control, a plant with an improper transfer function is to be controlled. The most common example is the force or torque applied by a motor where the torque is constant times the applied current. A third type of control is impedance control.

With impedance control, the differential equation relating the input and output is to be regulated. Examples for when such a controller is desirable would be with impedance matching for a large flexible structure or providing a specified load for another motor.

2.2.2. Model following control

An inherent property of feedback control is the dynamics of the closed-loop system are different from the open-loop system. This property is the basis of the root-locus design technique. Similarly, any feedback control law changes the dynamics of the plant - allowing one to think of such controllers as a type of impedance control. This was noted by Whitney [Whitney 1985] and [Spong 1996]. Similarly, they observed that impedance control was identical to feedback control with a specific feedback controller that resulted in the desired closed-loop dynamics.

A more concrete method to obtain the desired impedance is to use a reference model to define the desired behavior of the system as presented in Figure 4.

When the feedback gains, K, are fixed, a Model Reference Control approach is obtained. When the gains are adjustable based upon the tracking error, a Model Reference Adaptive Control approach is obtained [Landau 1979].

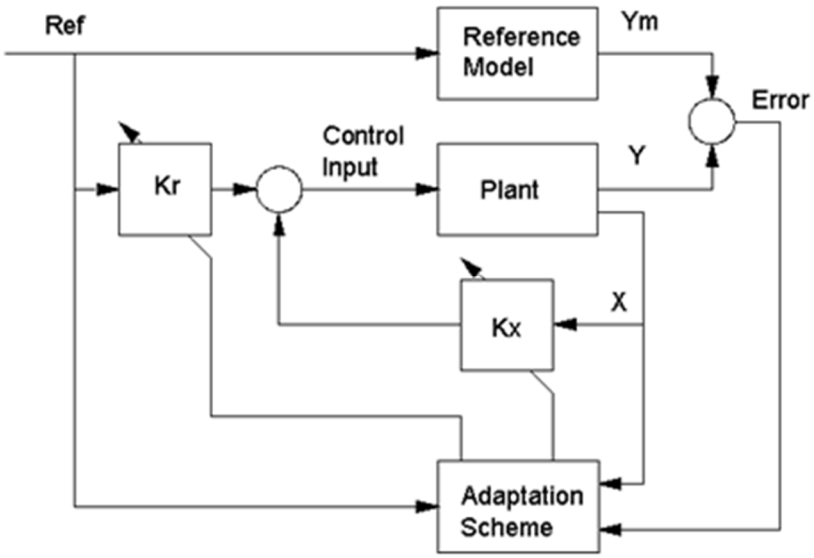

Figure 4. Model reference adaptive control.

While there is nothing inherently wrong with these methods, they tend to work best when all the states of the plant can be measured or estimated.

If the plant reacts to the input fast enough relative to the reference model, a simpler system can be obtained. If the reference model defines the desired output for a given impedance (i.e. transfer function), the feedback controller tracks this set point in a Model Following Control approach as presented in Figure 5.

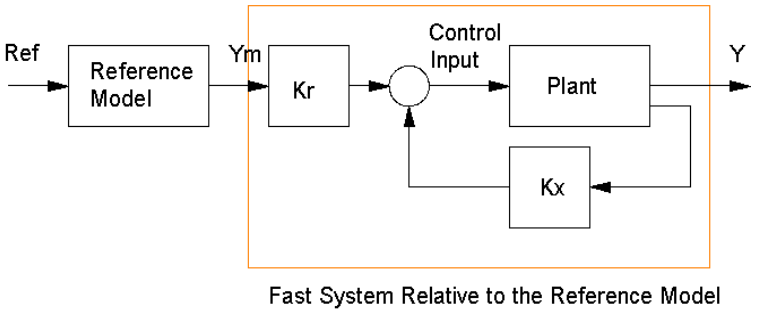

Figure 5. If the plant can be made fast relative to the reference model, a model following controller can be used.

2.2.3. Impedance control based upon position control

One problem with impedance control is that when the plant such as a robot comes in contact with the environment (thus creating a force and position at the output - both being necessary for an impedance), a second feedback loop is created which involves the impedance of the environment.

Hogan [Hogan 1985] investigated the effect of this second feedback look when applied to robot manipulators in contact with the environment. Specifically, Hogan considered the problem of trying to make the tip of a robot behave like an arbitrary spring or dash-pot when the operator pushed on it. In essence, he considered the problem of controlling the impedance (or admittance) of a robotic manipulator. Since position control is well understood, Hogan started with a position controller. Since the desired position is a function of force from equation (1), an outer feedback loop was added. With this loop, the tip force was measured, which then altered the position set point as

$$X_{ref} = Y_{ref} F_{ext} \tag{1}$$

where *Yref* is the desired admittance, *Xref* is the desired motion, and *Fext* is the external force applied by the environment. When in contact with a compliant environment, the tip position also affects the tip force depending upon the characteristics of the environment. This creates a second feedback loop presented in Figure 6.

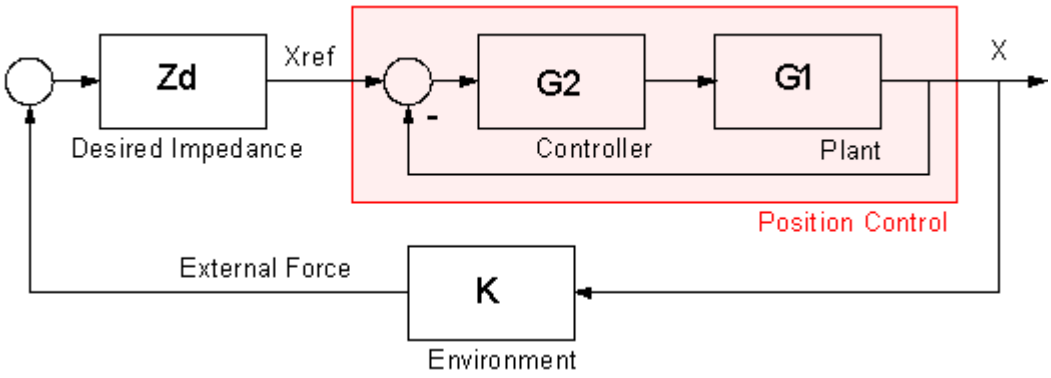

Figure 6. Impedance control based upon position control.

Note from Figure 6 the following. In free space, the environment has a gain of zero: as the robot moves, the tip force remains zero. This eliminates the outer feedback loop. As a result, stability is assured by stabilizing the position control loop.

When in contact with a surface, however, the environment provides feedback between tip position and tip force. For stiff surfaces, this gain can be extremely large, which, being a part of a feedback loop, can cause instabilities [DeSchutter 1997] [Dutta 2002].

This problem with instabilities when in contact with the environment was noted by Eppinger [Eppinger 1986]. He also noted that these instabilities are aggravated by backlash, higher-order dynamics, and sampling effects.

To analyze the effect of the environment on the net system's stability, Hogan [Hogan 1987] defined the desired impedance of a robotic manipulator to be

$$F_{external} = Ms^2X + BsX + K(x - x_0) \tag{2}$$

where M, B, and K are the desired impedance parameters. Coupling the dynamics of the environment with the dynamics for a robot result in

$$\Phi\Theta'' + C\Theta' = J^T(F_{actuator} - F_{external}) \tag{3}$$

The torque on the actuator required to obtain the desired impedance (equation 1) becomes:

$$\tau_a = J^T W^{-1} M^{-1}(K(X_0 - X) - BX') + J^T W^{-1}(JI^{-1}C\Theta' - J'\Theta') + J^T(1 - W^{-1}M)F_{external} \tag{4}$$

Given the measurements of the robot tip, its velocity, and external forces, the torque applied by the actuator to achieve the desired impedance can be calculated. This can also be framed in terms of joint motion and velocities by substituting in the robot kinematics equations.

Several problems exist with this solution. First, the torque is a function of the external force, which is a direct function of the torque. This creates an algebraic loop which adversely affects the stability of the system.

Second, the stability of the overall system is related to the dynamics of the environment. With this method, you need to know the environment's dynamics before you can determine the stability of the overall system – a constraint which is counterintuitive.

Third, the desired impedance

$$X = F_{ext} * \frac{1}{Z_d} \tag{5}$$

is not met. The transfer function from the external force to the tip position in Figure 6 is

$$X = \left(\frac{G_1 G_2}{1 + G_1 G_2} \right) Z_d F_{external} \tag{6}$$

The first term on the right-hand side of (6) defines how fast the plant can respond to a change in its set point. If this term is one, the desired impedance is achieved. This requires the position control loop be made arbitrarily fast. The speed of most physical systems is severely limited, however. This results in the desired impedance being limited to the bandwidth of the position controller.

Mills [Mills 1996] expanded upon Hogan's design by replacing the feedback term with a feedforward term. This resulted in a more robust system by breaking the feedback loop through the environment. Without feedback, however, the precision of the force and position control is reduced. For some applications where the precise force is not important, this tradeoff may be acceptable.

By designing an impedance controller around a position controller, several problems result. First, the actual impedance is often not the desired impedance. Secondly, the actual impedance depends on the impedance of the environment, which is undesirable. Third, stability of the overall system may depend upon the impedance of the environment.

2.2.4. Impedance control based upon force control

A second method proposed to build a system which regulates impedance is built around a force controller as presented in Figure 7. First, a quickly responding force controller is designed. Since the desired force is related to the position of the robot by the desired impedance, this force set point is developed by measuring the actual tip position and computing the desired force.

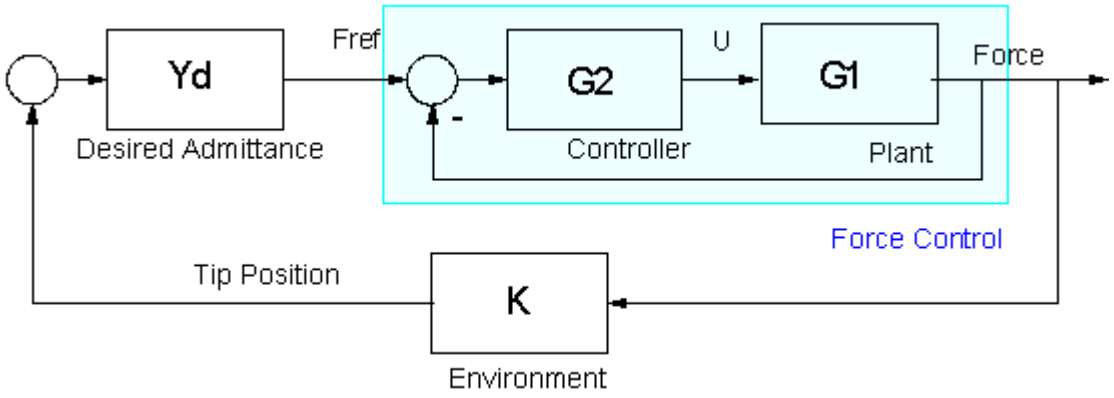

Figure 7. Configuration of an impedance controller based upon a force controller.

The heart of such a scheme is a force controller. Force control methods are summarized by Whitney [Whitney 1985] where the instabilities resulting from contact with the environment are noted. These methods are summarized by Kazerooni [Kazerooni 1986a] [Kazerooni 1986b], DeSchutter [DeSchutter 1997] who noted that the impedance of the environment needs to be known to a high degree of precision with this method.

Volpe [Volpe 1993] argues that impedance control is actually a subclass of force control where proportional gain feedback is used. He also noted that force control requires the inverse-dynamics of the system be computed - which can be a complex problem.

This approach was used by Lee [Lee 1991]. Lee noted that under the dynamic interaction between the manipulator and the environment it is difficult to achieve force as well as position control. In such situations, Lee proposed to compromise the accuracy between force and position control in a way to achieve the desired behavior of the manipulator which is defined by the differential equation relating the error in interaction force and the error in position. This led to the development of the concept of generalized impedance control, which is defined by:

$$M_d\left(\frac{d^2x}{dt^2}-\frac{d^2x_d}{dt^2}\right)+B_d\left(\frac{dx}{dt}-\frac{dx_d}{dt}\right)+K_d(x-x_d)=B_f\left(\frac{dF_e}{dt}-\frac{dF_e}{dt}\right)+K_f(F_e-F_c) \tag{7}$$

where B_f and K_f are the impedance parameters, M_d, B_d, and K_d are the desired impedance, and F_e and F_c are the actual and desired contact forces. By solving for F_c, the desired contact force can be determined. A force controller can then be driven to this set point.

The problem with Lee's approach is that the relation between the force error and position error to be:

$$\frac{F_e(s)-F_c(s)}{X(s)-X_d(s)}=(M_d s^2+B_d s+K_d)*\frac{1}{B_f s+K_f} \tag{8}$$

Hence we can see that the actual impedance is off the desired impedance. Secondly, this control law requires the derivative of the measured force, which is a noisy signal. This noise is amplified by requiring its derivative in (8).

Goldenberg [Goldenberg 1988] proposed a slightly different arrangement, as presented in Figure 8. With this method a force controller is designed using the inverse dynamics of the environment (1/Ze below). A second feedback loop is then added which regulates the tip position of the robot so that it behaves as a desired impedance (Zd).

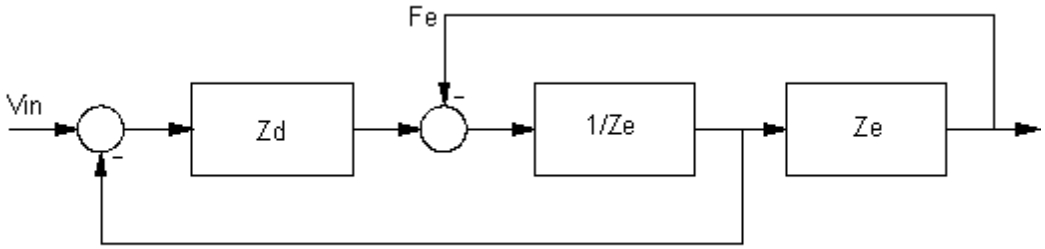

Figure 8. Force and impedance control.

One problem with this method is the feedback loop on the right is unstable with a loop gain of -1. Ignoring this detail, with this controller, the transfer function becomes:

$$\frac{Fe}{Vin - Ve} = \frac{1}{2} * Z_d, \tag{9}$$

$$\Rightarrow \frac{Fe}{Vin - \frac{Fe}{Ze}} = \frac{1}{2} * Z_d \tag{10}$$

After mathematical manipulation we have:

$$\frac{Fe}{Vin} = \frac{Z_d * Z_e}{2Z_e + Z_d} \tag{11}$$

where Z is the actual impedance.

As we can see from (11), the actual impedance differs from the desired impedance and it is dependent on the impedance of the environment. As mentioned before, it is also unstable with an algebraic loop with a gain of one.

The problems with controlling both position and force were also addressed by Vukobratovic [Vukobratovic 1996a] [Vukobratovic 1996b] [Vukobratovic 1998] where on-line learning and adaptive schemes were proposed to learn and compensate for uncertainties in the environment.

2.3. Hybrid Control

A third proposed method to develop an impedance controller uses both a position controller as well as a force controller as presented by Raibert [Raibert 1981] and presented in Figure 9. One of the earliest proposals for hybrid control dealt with controlling a robot as it goes from free space to the surface of an object.

When the robot is in free space, the force on the tip is zero. In this case, you need to control the position of the robot. When the robot is in contact with a rigid surface, however, you lose your ability to control position. Instead, you need to switch over to a force control scheme to prevent damaging the robot or the surface. In essence, a form of switching control is mandated.

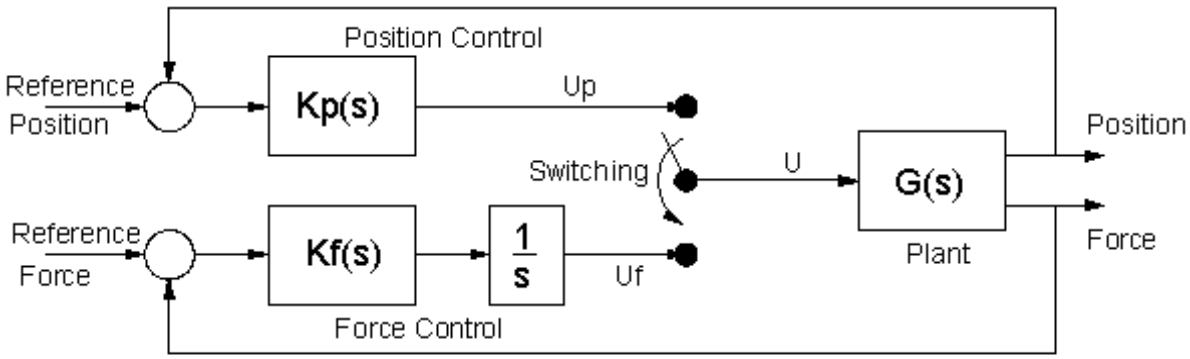

Figure 9. Typical configuration of a hybrid (switching) controller.

When a robot comes in contact with the environment, the robot needs to switch from position to force control. This switching can cause instabilities, as noted by Eppinger [Eppinger 1986], Hyde [Hyde 1993] with the problem of controlling that transition addressed by Hyde [Hyde 1994] and Tarn [Tarn 1996].

This type of switching control law was also proposed as a type of impedance control by Anderson [Anderson 1988] and Liu [Liu 1991]. The problem with this approach is that with impedance control, you are not controlling force or position. Instead, you are trying to regulate the differential equation which relates the two. To approximate an impedance controller, Anderson [Anderson 1988] proposed including a selection matrix, S. This matrix selects whether you are controlling position (S=1) or force (S=0) through the control law

$$U = S \cdot U_{position} + (I - S) \cdot U_{force} \tag{12}$$

Further, if you wish to control impedance, S can be selected so that you are controlling a weighted average of position and force.

There are several problems with this method as well. First, stability is an issue. In Figure 9, K_p and K_f are selected to be stabilizing controllers. By setting S=0 or S=1, a stabilizing controller is selected. By setting 0<S<1, however, you are averaging stabilizing controllers. There is to the author's knowledge no proof that the average of stabilizing controllers is also stabilizing.

Second, by setting 0<S<1, the control law is trying to regulate a weighted average of position and force. As a result, neither position nor force is controlled to the set point. Furthermore, the differential equation relating the two is not set, meaning that the controller does force the impedance to some desired differential equation.

2.4. Two Motor Solution

Another approach proposed by Girdharan [Girdharan 2012] is to use two motors: one to control the flow (a position controller) and the second to control the pressure (force control). The advantage of this approach is that each motor only focuses on one task. Control techniques for position and force control can also be used for each motor. Since the two motors are driving the same system, however, each will look like a disturbance on the other.

One control scheme which shows promise for this solution is a Fourier Repetitive Controller [Glower 1995]. Since the motors will receive blood flow from the left ventricle, all signals are essentially periodic. Repetitive controllers are a type of controller designed specifically for such systems.

2.4.1. What is repetitive control

Repetitive control is a type of model-reference adaptive control (MRAC) [Hara 1988]. With a hyperstability-based MRAC, such as proposed by Landau [Landau 1979], the control input is defined as the product of a set of known quantities (typically the states of the plant) and unknown constants:

$$U = \sum k_i \cdot f(ref, x, \frac{dx}{dt}, ...) \tag{13}$$

If a set of constants, k_i, exist which force the plant to follow the reference model (termed the perfect model following conditions), these constants can be estimated using various adaptive control schemes [Landau 1979, Popov 1963].

Note that $f()$ is simply a basis function: a pre-defined set of functions which spans the input (U) required to force the plant to follow the reference model. If the set point is periodic, then the input, U, must also be periodic.

Repetitive control is a subset of MRAC where the basis function chosen is a periodic function which spans the (unknown) input required for perfect model following. The variations of repetitive control result from how this basis function is chosen.

2.4.2. Time-slot repetitive control

Time-Slot Repetitive Control was the first form proposed [Hara 1988]. Since the ideal input is periodic in time T, it can be approximated with n constants defined over n time slots

$$u(t) = \sum k_i \delta(t - iT/n) \tag{14}$$

where each constant is estimated using adaptive control schemes such as

$$\nu = e + \frac{de}{dt} \tag{15}$$

$$k_i = \nu\delta(t - iT/n) + \int_0^t \nu\delta(t - iT/n)d\tau \tag{16}$$

where $\delta(t)$ is the discrete-time delta function:

$$\delta(t) = \begin{cases} 1 & 0 < t < T/n \\ 0 & otherwise \end{cases} \tag{17}$$

This is presented in Figure 10.

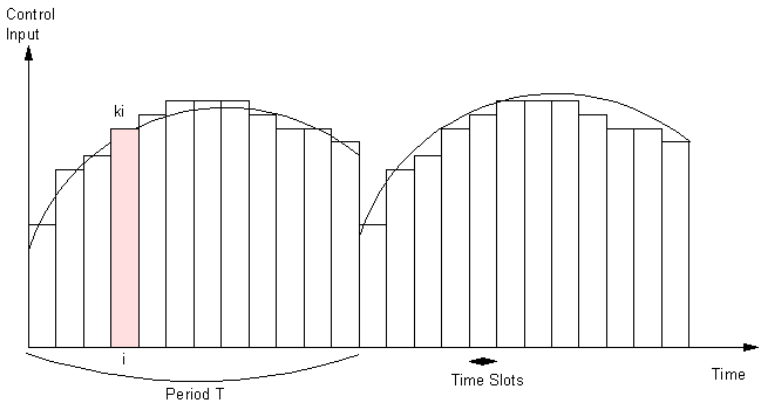

Figure 10. Time-slot repetitive control defines N-time slots each period. The amplitude of the control input at each time slot is estimated using MRAC schemes.

The advantage of this approach is any periodic input can be approximated with this basis. It also assumes nothing about the ideal input: jumps and discontinuities have no effect on the approximation. The disadvantage is that a large number of time slots are required for an accurate approximation. This results in a large number of parameters being estimated, which slows down the adaptation process [Glower 1988].

Slight variations of this scheme use trapezoids or other polynomials to approximate the ideal input in each time slot. These reduce the number of time slots required at the cost of more parameters needing to be estimated in each time slot. A summary of these schemes is presented in [Glower 1995b].

2.4.3. Fourier repetitive control

Since the ideal input is periodic, it can also be represented by its Fourier series.

$$u(t) = b_0 + \sum a_i \sin(n\omega t) + b_i \cos(n\omega t) \tag{18}$$

Selecting *n* terms from the Fourier series as the basis function results in a Fourier Repetitive Control [Glower 1995].

$$\nu = e + \frac{de}{dt} \tag{19}$$

$$b_0 = \nu + \int_0^t \nu dt \tag{20}$$

$$a_i = \nu \sin(i\omega t) + \int_0^t \nu \sin(i\omega t) d\tau \tag{21}$$

$$b_i = \nu \cos(i\omega t) + \int_0^t \nu \cos(i\omega t) d\tau \tag{22}$$

Depending upon the waveform being followed, a Fourier Repetitive Controller can result in better tracking using fewer terms [Glower 1995b]. Estimating fewer terms results in a faster converging adaptive control scheme [Glower 1988].

2.5. Chapter Summary

In the previously described methods for controlling the impedance of a system, several problems were observed. In simulation, these problems can be overcome by using two separate motors: one to control position and the other to control force. The following chapters will investigate the implementation of such a scheme on two physical motors.

CHAPTER 3. POSITION CONTROL DESIGN

3.1. Introduction

In this chapter, hardware and LabVIEW implementation of a position controller will be presented. It can be seen from the analysis in Chapter 2 that the system is composed of two distinct controls, namely, position (or velocity) control and current control. The position control is achieved with a servo motor (including a DC motor, a motor driver and an encoder), a PC running the Fourier repetitive control algorithm implemented in LabVIEW, and a DAQ card for signal IO. The current control is achieved using a servo motor and an arbitrary waveform generator (Agilent, Santa Clara, CA) and will be discussed in greater detail in Chapter 4.

The servo motor's angular velocity is mathematically equivalent to blood flow into and out of an AVD. Integrating the desired angular velocity results in the angle, or position, of the motor. Similarly, in this thesis, a motor angular position controller is designed, simulated and implemented.

3.2. System Structure

The components of the position controller includes a PC with LabVIEW 2009 installed and the Fourier repetitive control algorithm implemented, a DC motor that simulates the motor inside an AVD, an analog servo drive that supplies adequate current to drive the DC motor, the output voltage of which is proportionally controlled by its input voltage from the PC, an encoder that converts the position information to a digital code, an I/O board which serves as D/A and A/D converters and interfaces the PC with both the analog servo drive and the encoder. The list of components of the position controller is indicated in Table 1.

Table 1. Details of the main components of position control

Component	Description
DC motor	WGYMN-4200, Servo Systems Co., Montville, NJ.
Analog servo drive	30A8ACT, ADVANCED Motion Controls, Camarillo, CA. Key Specs: Peak current: 30A, Continuous current 15A, Power supply: AC.
PC	DakTech Computers, Fargo, ND. Key Specs: Intel ® Core ™ 2 Quad CPU Q8400 2.66 GHz; Windows XP Professional SP 3.
I/O board	NI PCI-MIO-16E-4 (NI 6040 E), National Instruments, Austin, TX. Key Specs: Analog Input Resolution: 12 bits, Maximum Sampling Rate: 250kS/s (Multiple-channel scanning); Analog Output Resolution: 12 bits, Max update rate: 400kS/s (Non-FIFO Mode).
Encoder	ARS25-FF501024, SICK Inc., Minneapolis, MN. Key Specs: Measurement range: 1 revolution, Encode: Gray, Resolution: 1024.
DC power supply	RSR HY3002-3, RSR Electronics, Inc. Output voltage: two 0-30V; Output current: two 0-2A; Fixed output, 5V, 3A. Serial or parallel operation.
15MHz Function/Arbitrary Waveform Generator	Agilent 33120A. Key Specs: Sine: 100μHz-15MHz, 50mVpp-10Vpp; Harmonic Distortion on sine wave for DC-20kHz: -70dBc, total harmonic distortion: <0.04%.

A visual representation of the system is as shown in Figure 11.

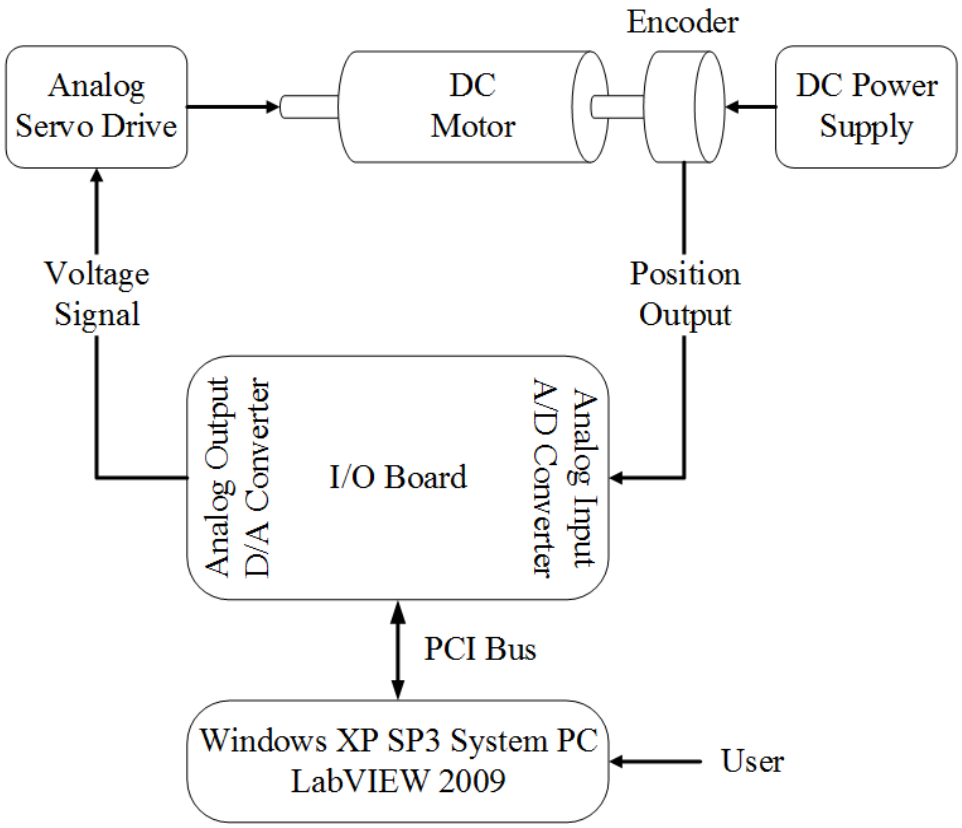

Figure 11. Functional block diagram of position controller.

3.3. Algorithm: Fourier Repetitive Control

Fourier repetitive control is adopted to control the position of the motor. The mathematical model, simulation and results follow.

3.3.1. Mathematical model – MRAC model

Model Reference Adaptive Control (MRAC) is a control scheme that contains a reference model that defines how the plant is to behave, a controller, and an adjustment mechanism to tune

parameters of the controller to force the plant to behave like the reference model as presented in Figure 12.

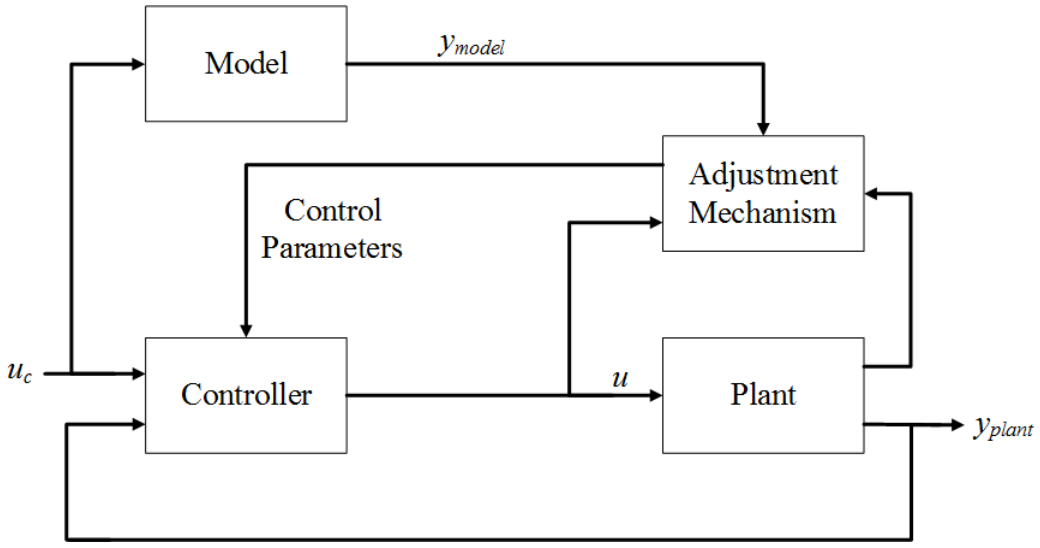

Figure 12. Model reference adaptive control (MRAC) block diagram.

Since all that matters in the algorithm is the output of the reference model, its output can be replaced with the position the motor is to follow. The purpose of the controller is to estimate the input required to force the plant to follow the model. As long as the input required for perfect tracking is spanned by the basis function chosen, the perfect model following conditions are satisfied and the plant will track the model (eventually) [Glower 1997].

Since the output set point is periodic at 1Hz, the input required to force perfect model following will also be periodic at 1Hz. Hence, one choice for the basis function is the Fourier series expansion of $u(t)$

$$u(t) = b_0 + \sum a_i \sin(n\omega t) + b_i \cos(n\omega t) \tag{23}$$

where ω is the fundamental frequency or first harmonic of the repetitive rate.

The tracking error e is defined as the difference between the plant output and the reference model output:

$$e = \theta - \theta_m \tag{24}$$

where θ is the angle of the actual motor while θ_m is the angle of the reference model.

a_i and b_i can be estimated using the following algorithm based upon hyperstability techniques [Glower 1995] with guaranteed stability as shown by Landau [Landau 1979] and Popov [Popov 1963].

$$v = e + \frac{de}{dt} \tag{25}$$

$$b_0 = v + \int_0^t v d\tau \tag{26}$$

$$a_i = v \sin(i\omega t) + \int_0^t v \sin(i\omega t) d\tau \tag{27}$$

$$b_i = v \cos(i\omega t) + \int_0^t v \cos(i\omega t) d\tau \tag{28}$$

Note that it is important that the plant is a stable system (i.e. not a type-1 system) [Glower 1988] [Glower 1989]. Likewise, this algorithm assumes that the plant includes a stabilizing controller, such as unity feedback.

3.3.2. Simulation

To test the above adaptive control scheme, this algorithm is simulated using National Instruments LabVIEW 2013 with Control Design and Simulation add-on Module installed on a PC (Windows 8.1 pro). The simulation begins with defining a plant (the transfer function of a motor) using zero-pole-gain:

$$\frac{\theta(s)}{V(s)} = \frac{36}{s(s+6)} \tag{29}$$

In LabVIEW, this system is input as presented in Figure 13.

Figure 13. Plant model setup.

In LabVIEW, a plant model can be defined by filling out a Plant Model cluster with Model name, Sampling Time, Gain and arrays of Zeros and Poles. For this simulation, the plant model has a gain of 36, two poles that are 0 and -6 and no zeros. LabVIEW automatically displays the Plant Model Equation based on Plant Model using "CD Draw Zero-Pole-Gain Equation.vi" provided by the Control Design and Simulation Module as shown in Figure 13. The back panel is presented in Figure 14.

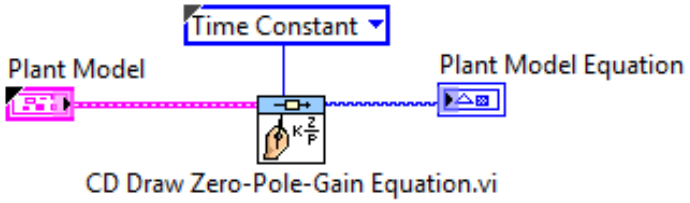

Figure 14. A "CD Draw Zero-Pole-Gain Equation.vi" for displaying equations based on plant model cluster.

The simulation is executed in a "Control & Simulation Loop" as shown in Figure 15. A total of 100 data points are collected with a sampling rate of 10 Samples / second. The reference signal θ_m is defined as a 1 Hz (or an angular speed of 2π rad/s) sine wave with an offset of 0 and an amplitude of 1; Error e is then calculated by taking the difference of the process output θ and model output θ_m. Seven bases functions all with a magnitude of 1 are used including a DC term, sine and cosine at the fundamental frequency (1 Hz in this case), sine and cosine at the second harmonic (2 Hz), and sine and cosine at the third harmonic (3 Hz). The control signal is a summation of 8 components that are b_0, $a_1 \sin(2\pi t)$, $b_1 \cos(2\pi t)$, $a_2 \sin(4\pi t)$, $b_2 \cos(4\pi t)$, $a_3 \sin(6\pi t)$, $b_3 \cos(6\pi t)$ and $-\theta$, which serves as negative feedback that stabilizes the controller. The implementation of this repetitive controller in LabVIEW is presented in Figure 15. Simulation studies to verify this controller does track periodic set points are presented in the following sections.

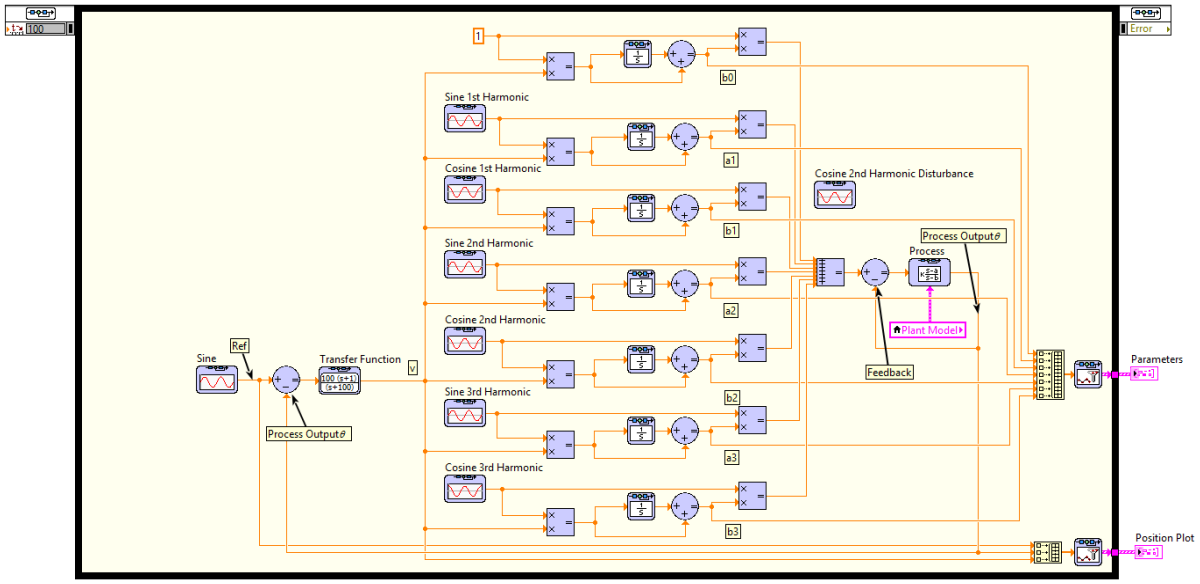

Figure 15. LabVIEW control & simulation loop back panel for tracking half-rectified sine wave.

3.3.3. Simulation results

First, consider the case where the set point (reference model output) is a pure 1Hz sine wave. Since this is a linear system, the input, $u(t)$, should also be a 1Hz sine wave – but at an unknown amplitude and phase shift. The results of this algorithm when tracking a 1Hz sine wave are presented in Figure 16. Note that the error in Figure 17 and Figure 17 is driven to zero as expected. Also note that the coefficients for the 1Hz terms in Figure 18 and Figure 19 are converge to a constant while the remaining coefficients remain at zero as expected.

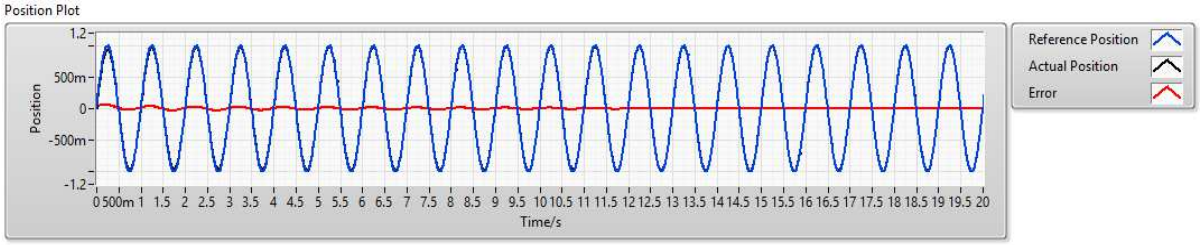

Figure 16. Position plot from 0 to 20 seconds.

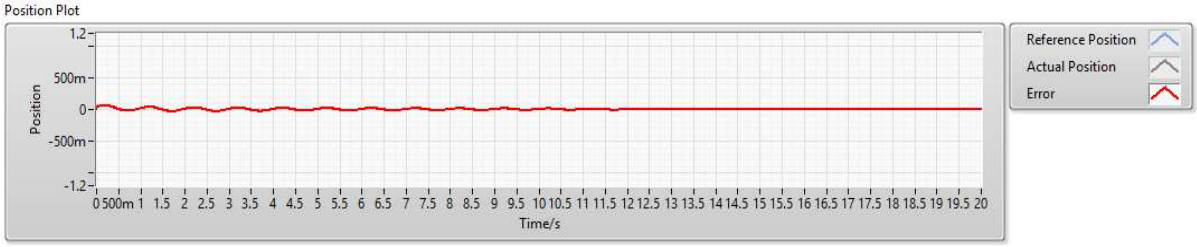

Figure 17. Error e from 0 to 20 seconds.

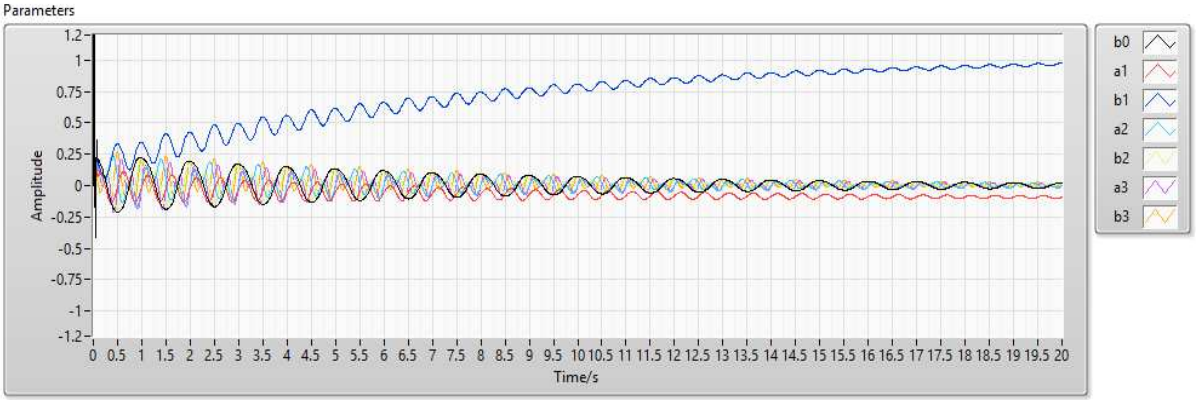

Figure 18. The coefficients including b_0, a_1, b_1, a_2, b_2, a_3 and b_3 from 0 to 20 seconds.

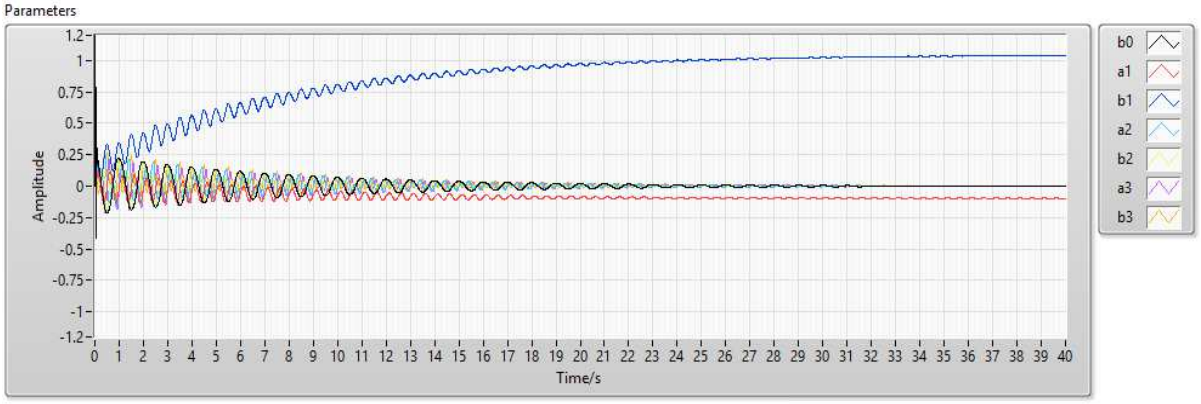

Figure 19. The coefficients including b_0, a_1, b_1, a_2, b_2, a_3 and b_3 from 0 to 40 seconds.

Next, consider the case where the set point is changed to a 2Hz sine wave as presented in Figure 20. Again, this should result in the input, $u(t)$, required for perfect model following to be a 2Hz sine wave as well – but at an unknown amplitude and phase shift.

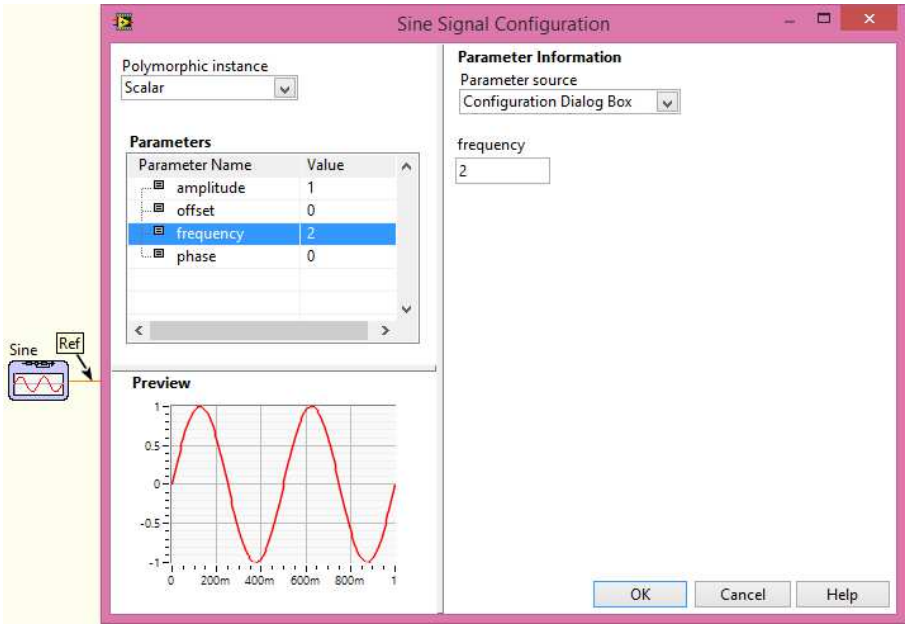

Figure 20. Tracking signal configuration.

Simulation results are presented in Figures 21-23. As expected, the output converges to the 2Hz set point (Figure 21) and the error is driven to zero (Figure 22). Moreover, the coefficients of the 2Hz components of the input (a_2 and b_2) converge to constants while the other terms converge to zero as expected (Figure 23).

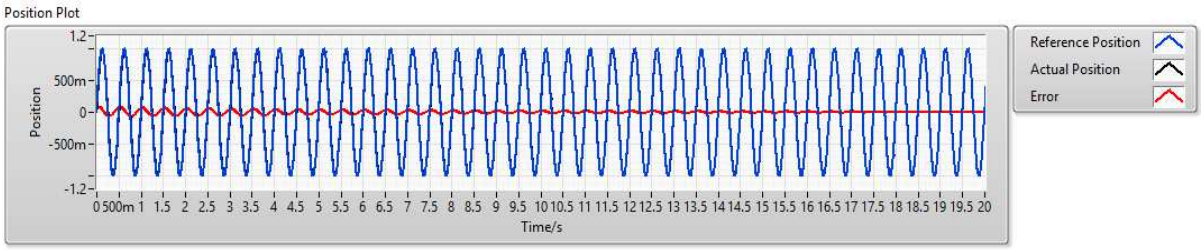

Figure 21. Position plot from 0 to 20 seconds.

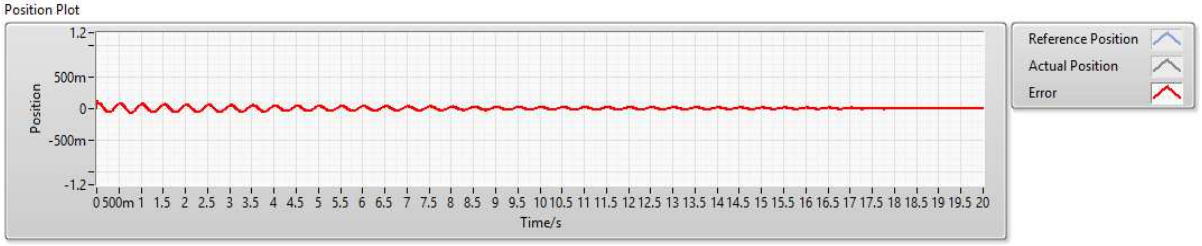

Figure 22. Error e from 0 to 20 seconds.

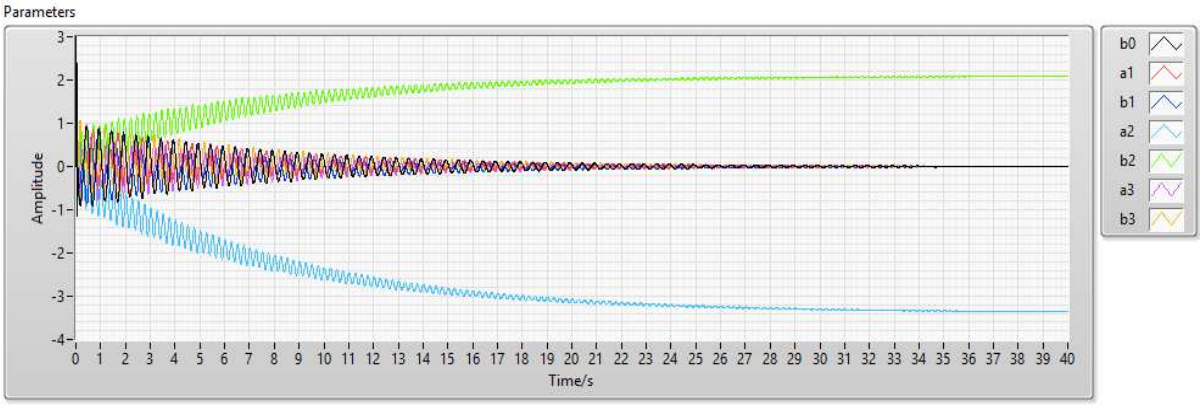

Figure 23. The coefficients including b_0, a_1, b_1, a_2, b_2, a_3 and b_3 from 0 to 40 seconds.

Next, consider the case where a disturbance with a period of 1 second affects the plant. Again, in this case, this should result in the input required for perfect model following to be

periodic in 1Hz as well. To illustrate this, a 1Hz set point is used along with a 2Hz disturbance. This should result in perfect model following (since the ideal input required for perfect model following is spanned by the basis function chosen). The input, $u(t)$, in this case should contain both a 1Hz term (due to the 1Hz set point) and a 2Hz term (to reject the disturbance).

The implementation of this test is presented in Figure 24.

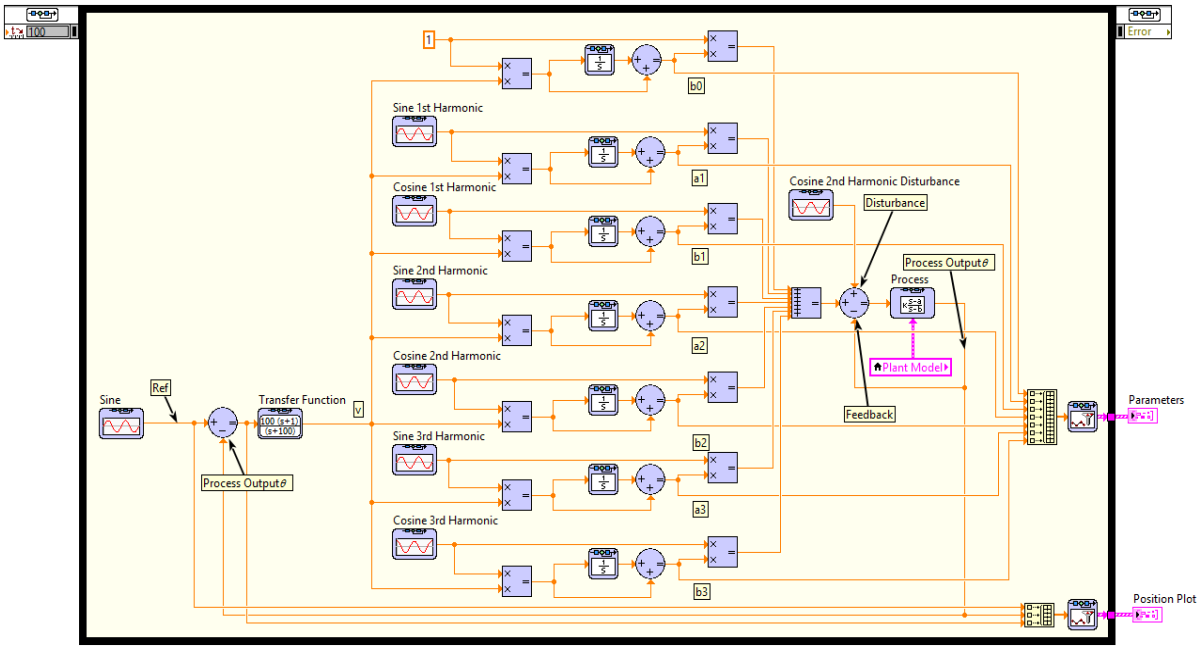

Figure 24. LabVIEW control & simulation loop back panel for disturbance rejection.

The output of the plant is presented in the following figures. Note that the output follows the 1Hz set point in spite of the 2Hz disturbance as expected (Figure 25) and the error is driven to zero (Figure 26). Moreover, the input, $u(t)$ has both a strong 1Hz component (b_1) and 2Hz component (b_2) as expected (Figure 27 and Figure 28).

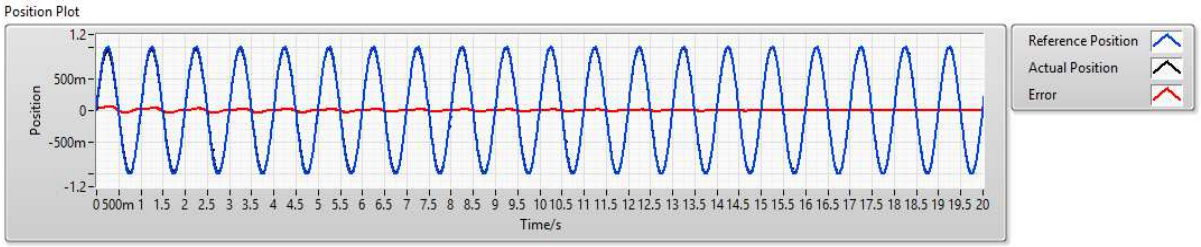

Figure 25. Position plot from data point 0 to 20 seconds when rejection disturbance of second harmonic sine wave.

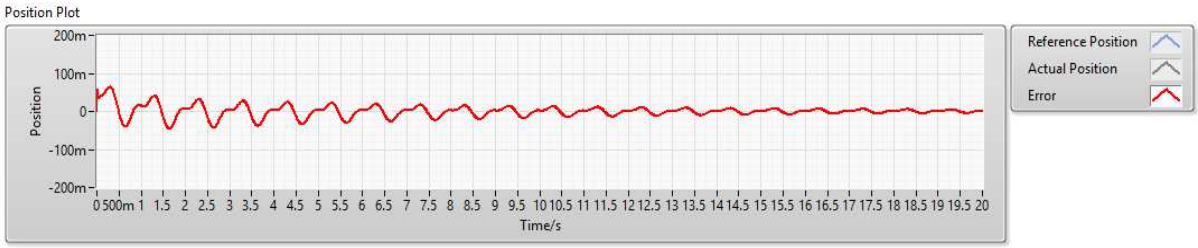

Figure 26. Error e from data point 0 to 20 seconds when rejecting disturbance of second harmonic sine wave.

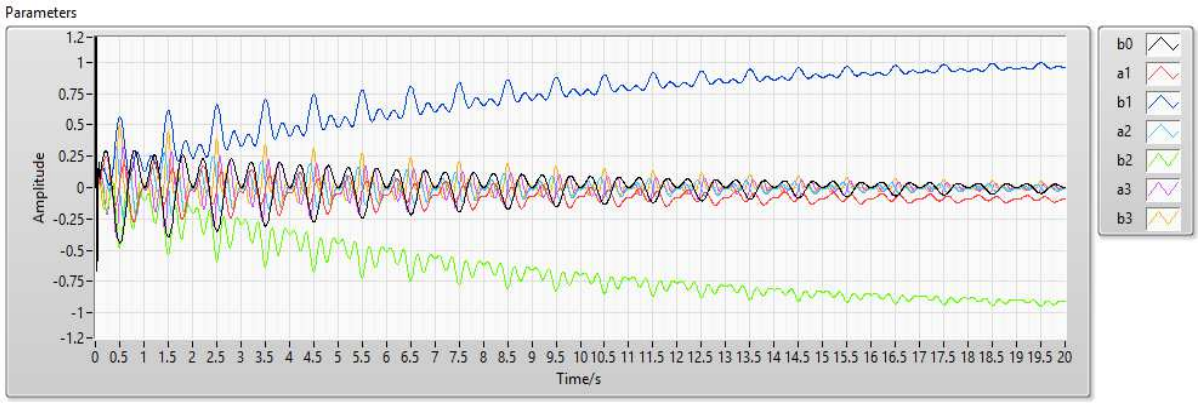

Figure 27. The coefficients including b_0, a_1, b_1, a_2, b_2, a_3 and b_3 from 0 to 20 seconds when rejecting disturbance of second harmonic sine wave.

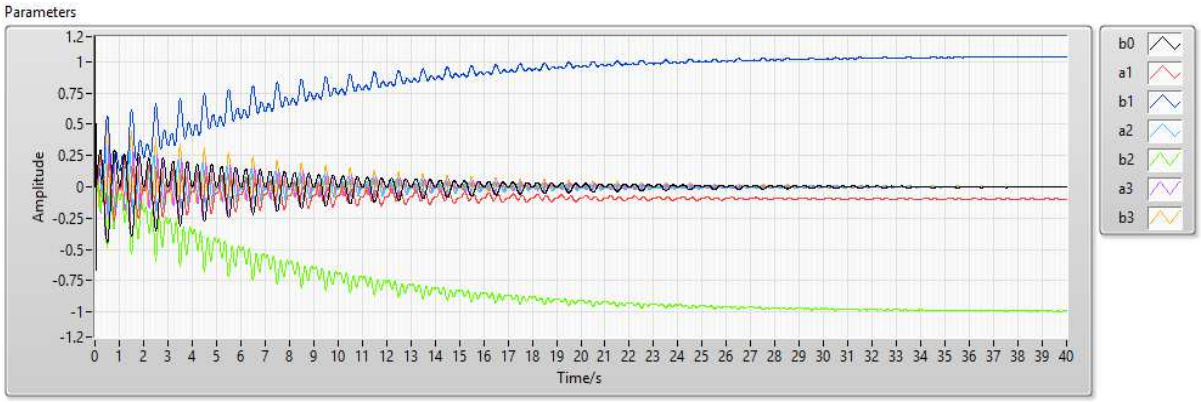

Figure 28. The coefficients including b_0, a_1, b_1, a_2, b_2, a_3 and b_3 from 0 to 40 seconds when rejecting disturbance of second harmonic sine wave.

From the previous examples, it is evident that the repetitive Fourier control algorithm works when the set point is such that the basis function spans the input required for perfect model following. This brings up the question of what happens if the basis for $u(t)$ does not span the necessary space? To illustrate what happens then, consider the case where the set point is a half-rectified sine-wave. In this case, the Fourier series expansion of the set point is:

$$f(t) = \frac{A}{\pi} + \frac{A}{2}\sin(\omega_0 t) - \frac{2A}{\pi}\sum_{k=2,4,6\ldots}^{\infty}\frac{1}{k^2-1}\cos(k\omega_0 t) \tag{30}$$

when the half-rectified sine wave has an amplitude A, angular frequency ω_0 and $f(0) = 0$.

Note that the set point is an infinite series with energy beyond the 3rd harmonic as presented in Figure 29. Likewise, the input required for perfect model following is also an infinite series – i.e. it is not spanned by the basis function chosen for $u(t)$. Hence, prefect model following cannot be achieved.

Instead, the adaptive control algorithm should do the best it can, given the basic function chosen. Since the error is never driven to zero, the algorithm should converge eventually, but a

small ripple in the error and parameter estimates is expected as it continues to search for the 'best' coefficients for the basis function.

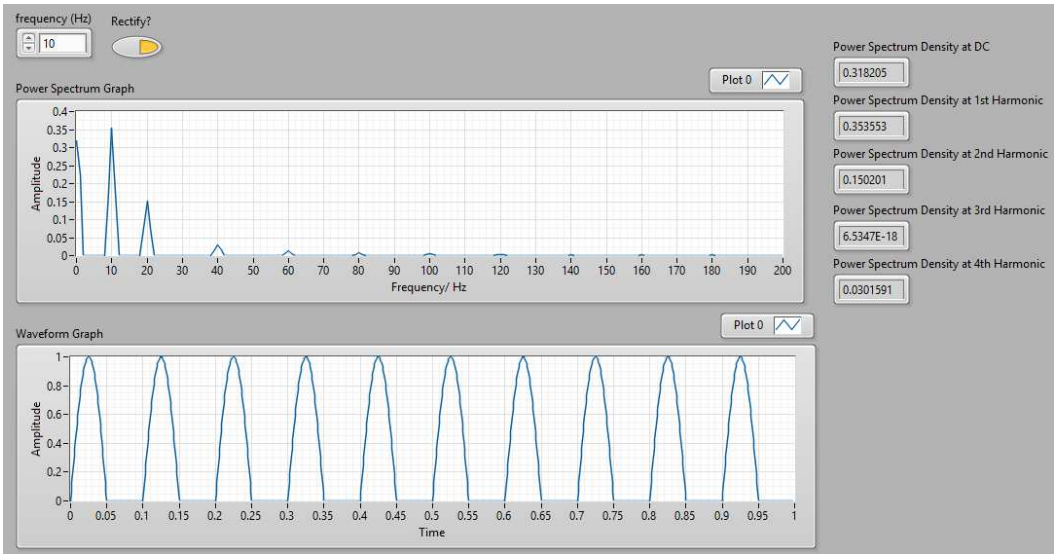

Figure 29. When half rectified sine wave is 10 Hz, the power spectrum shows 31.82% energy is distributed at DC, 35.36% of energy is contributed by 1st harmonic (10 Hz), 15.02% of energy is contributed by the 2nd harmonic (20 Hz), 0% of energy is contributed by the 3rd harmonic and 3% of energy is contributed by the 4th harmonic (40 Hz).

In LabVIEW, half-wave rectification can be achieved by adding an expression node to the sine wave output (Figure 30). The expression node compares the sine output value against 0 and outputs the greater value.

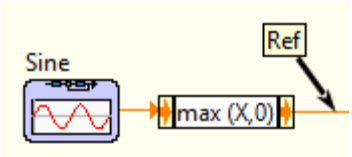

Figure 30. Half-Rectified sine wave generation in LabVIEW.

The simulation results are shown in Figure 31 through Figure 33. The output of the plant closely follows the set point (Figure 31) with the error being driven close to zero, but never

converges to zero (Figure 32 and Figure 33). This is expected since the perfect model following conditions are not satisfied. The estimates of the Fourier coefficients of the ideal input (Figure 34 and Figure 35) converge to constants with 1Hz periodic disturbances. This too is expected since the error is never driven to zero and is periodic in 1Hz. Likewise, the estimates of the Fourier coefficients continue their search.

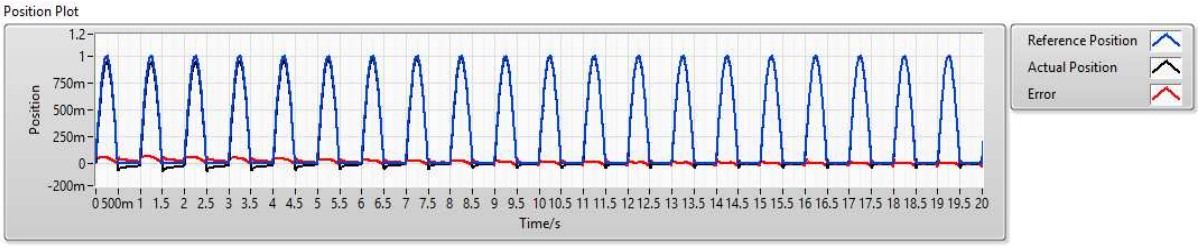

Figure 31. Position plot from 0 to 20 seconds.

Figure 32. Error *e* from 0 to 20 seconds.

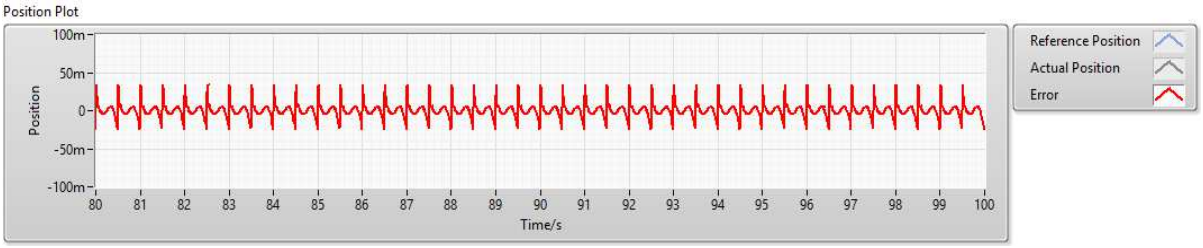

Figure 33. Error *e* from 80 to 100 seconds.

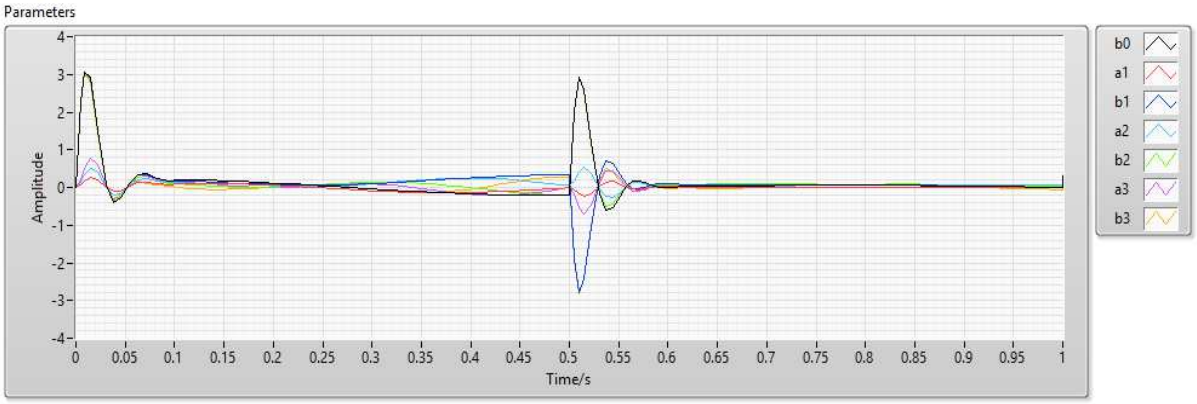

Figure 34. The coefficients including b_0, a_1, b_1, a_2, b_2, a_3 and b_3 from 0 to 1 second.

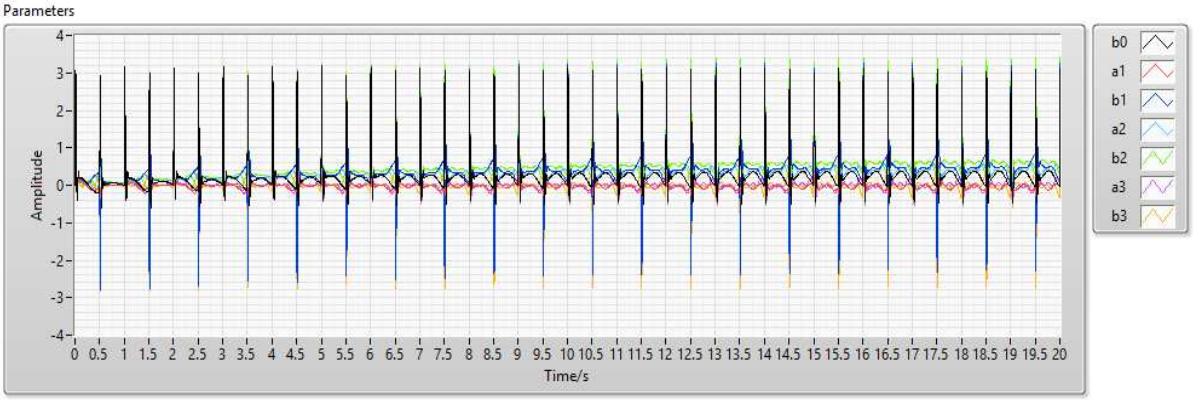

Figure 35. The coefficients including b_0, a_1, b_1, a_2, b_2, a_3 and b_3 from 0 to 20 seconds.

From the above simulation results, it can be concluded that Fourier repetitive control scheme effectively tracks first and second harmonic and half-rectified sine waveforms and it also rejects disturbances of first and second harmonic sine waves.

3.4. Implementation

3.4.1. Hardware description

The DC power supply supplies power to the encoder. The encoder (ARS25-FF501024, SICK Inc., Minneapolis, MN) outputs position information in Gray code. The mechanical interface is a 3/8'' shaft and the electrical connection type is axial connector MS 19 pin (Figure 37. Encoder MS 19 pin pinout.). The encoder is mechanically coupled to the shaft of the DC motor with a mechanical coupler (Figure 36), and the DC power supplies 14 Volts to the encoder via a National Instruments CB-68LPR 68 pin I/O connector block. The encoder pinout is indicated in Table 2.

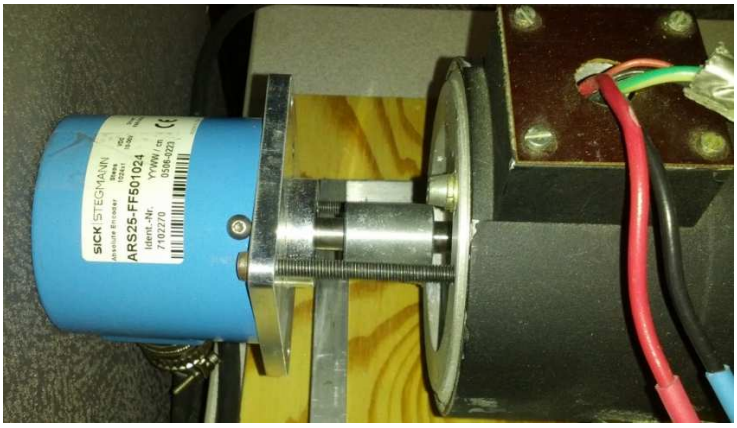

Figure 36. The mechanical connection between the encoder and DC motor.

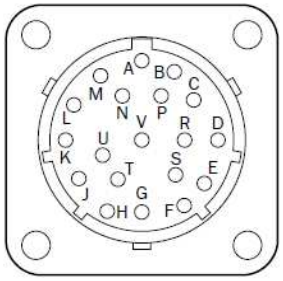

Figure 37. Encoder MS 19 pin pinout.

Table 2. Encoder MS 19 pin pinout

Function	Cable Color	19 Pin Out	Connect to NI PXI-6040E Pin	Connector Block Pin
Bit 0	Lilac	A	AI-0	68
Bit 1	white/brown	B	AI-1	33
Bit 2	white/green	C	AI-2	65
Bit 3	white/yellow	D	AI-3	30
Bit 4	white/grey	E	AI-4	28
Bit 5	white/pink	F	AI-5	60
Bit 6	white/blue	G	AI-6	25
Bit 7	white/red	H	AI-7	57
Bit 8	white/black	J	AI-8	34
Bit 9	brown/green	K	AI-9	66
Bit 10	brown/yellow	L	Insulated	
Bit 11	brown/grey	M	Insulated	
Bit 12	brown/pink	N	Insulated	
Bit 13	brown/blue	P	Insulated	
cw/ccw	brown	S	Insulated	
GND	blue	T	AI Sense	62
Store	pink	R	Insulated	
+Vs	red	V		To 14V DC
Set Zero	grey	U	Insulated	

The output of the encoder is encoded in 14bit (Bit 0 – Bit 13) Gray code. Due to the limited number of digital input ports on the DAQ card (NI PXI-6040E, 8 digital I/O), analog input ports (AI0 – AI9) were used instead to receive the position output. According to the specification of the encoder, a Logic High = 0.7×Vs and a Logic Low = 0 – 0.3×Vs, where Vs is the operating voltage supplied to the encoder. When the operating voltage is fixed at 14V, a Logic High

equals 9.8V or larger while a Logic Low equals 0 – 4.2V based on the specification. However, measurements show a Logic High is typically around 9.9V and a Logic Low is typically around 0V. A 5V threshold is selected to separate Logic High from Logic Low. Any voltage input that is under 5V is regarded as a Logic Low while any voltage above 5V is considered to be a Logic High.

The LabVIEW program receives the output from the encoder and compares the received data against the 5V threshold. A subvi (graytobinary.vi) was developed to convert the Gray code to binary code (Figure 38) by executing the following code:

```
for (converted = 0; gray; gray >>= 1) {
     converted ^= gray;
   }
   return converted;
```

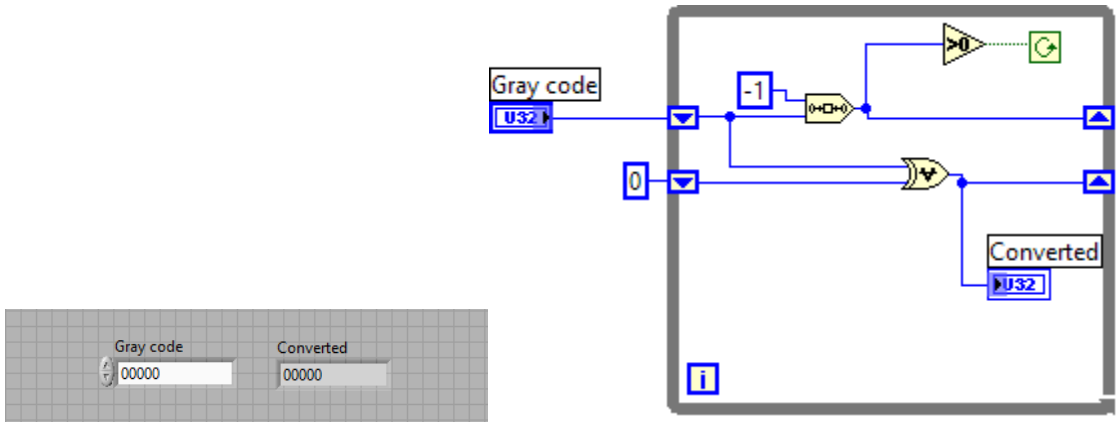

Figure 38. Graytobinary.vi a) front panel and b) block diagram.

The resolution of the encoder is $2^{10} = 1024$ and only 10 output ports are used. In LabVIEW, only channel AI0 – AI9 are used. The setup of the physical channel is shown in Figure 39.

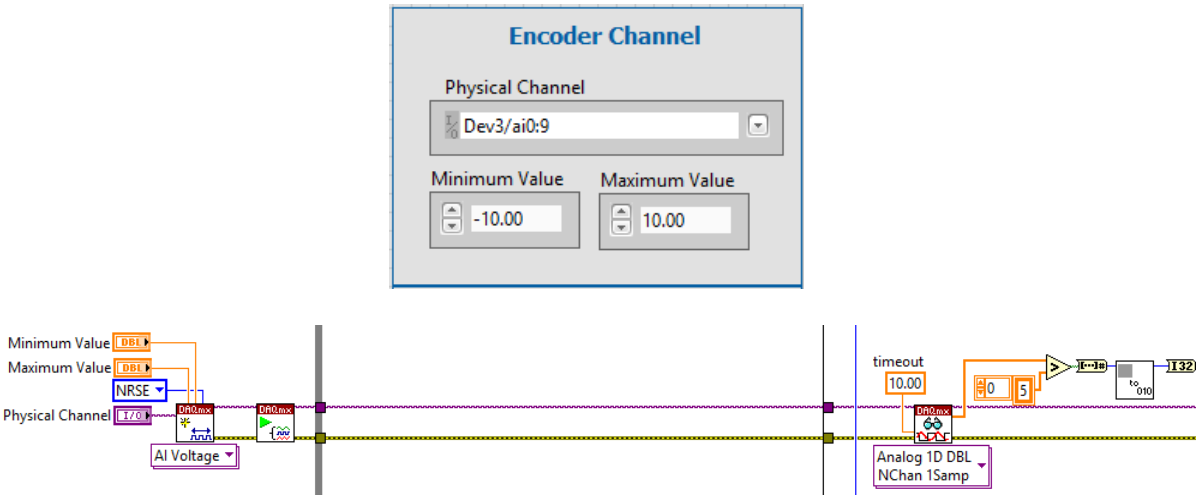

Figure 39. Physical channel setup for analog inputs.

Because the output of an absolute encoder is only position-relevant, when the position increases from a number lower than 1024 to 1024, instead of displaying a number larger than 1024, the output of the next position is 0. This causes a "jump" from 1024 to 0 and vice versa and is harmful in providing feedback to the controller. In order to address this problem, the author devised an algorithm that determines the rotational direction and calculates the true position. The basic idea is to first find if the current input is 600 points larger than the previous input, if is, then that implies a "jump" from 0 to 1024 just occurred and thus the real position is equal to the previous position + (current input – previous input) – 1024; if not, then the program compares the input difference against -600, if true, then that means a "jump" from 1024 to 0 just happened and the real position = previous position + (current input – previous input) + 1024. If both comparisons are false, then no "jumps" has occurred and the real position = previous position + (current input – previous input).

An illustration of the modular 1024 correction is presented in Figure 40. It can be seen from Figure 41 that when the motor is in uniform linear rotation, the position output of the encoder "jumps" to 0 when it reaches a certain value (1024 in this case). After the aforementioned code is added, the program remembers the previous position and calculates the rotation direction so no error occurs. These "jumps" are removed with the proposed algorithm as presented in Figure 42.

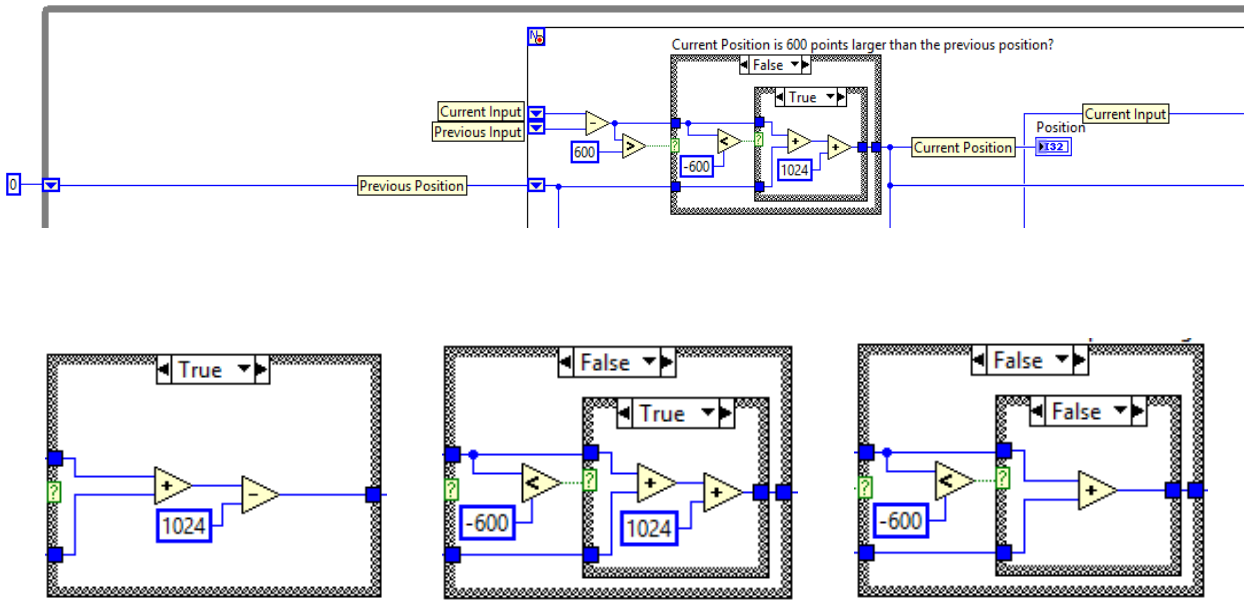

Figure 40. Block diagram of the position integrator.

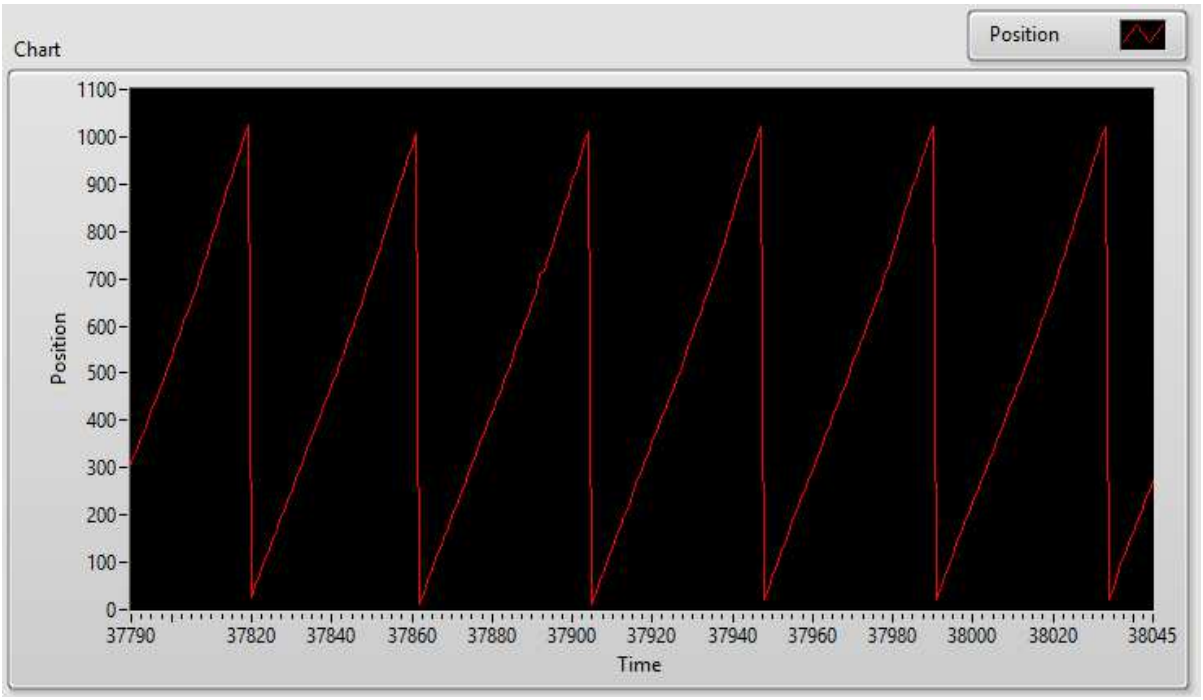

Figure 41. Position output when motor is in uniform linear rotation (before position correction).

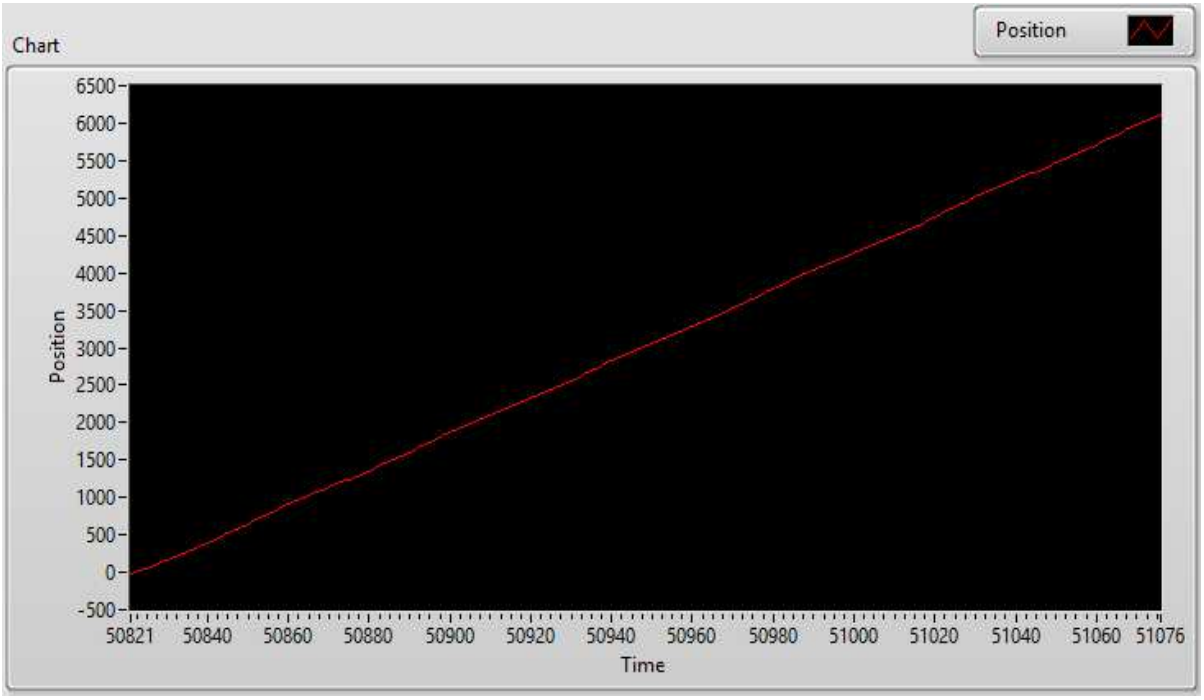

Figure 42. Position output when motor is in uniform linear rotation (after position correction).

The program has two loops – a for loop (outer loop) and a while loop (inner loop). The outer loop is where waveforms are generated. The tracking waveform and internal model waveforms are generated and data points are calculated and are output as an array of data. The inner loop runs at a constant time interval of 50 ms. The array of data from the outer loop is indexed and processed one by one. The motor angular position is calculated and control voltage updated based on calculated error e at a set time interval of 50 ms.

At the center of the tracking waveform generator is a Basic Function Generator (Figure 43). It creates a waveform based on the inputs including signal type, frequency, amplitude, Phase (deg) and Sampling Info. It is worth noting that the Sampling Info dictates sampling frequency and number of samples each run (cycle). The sampling frequency is set to be 20 Hz and number of samples is calculated based on tracking frequency:

$$\#s = \frac{20}{f} \tag{31}$$

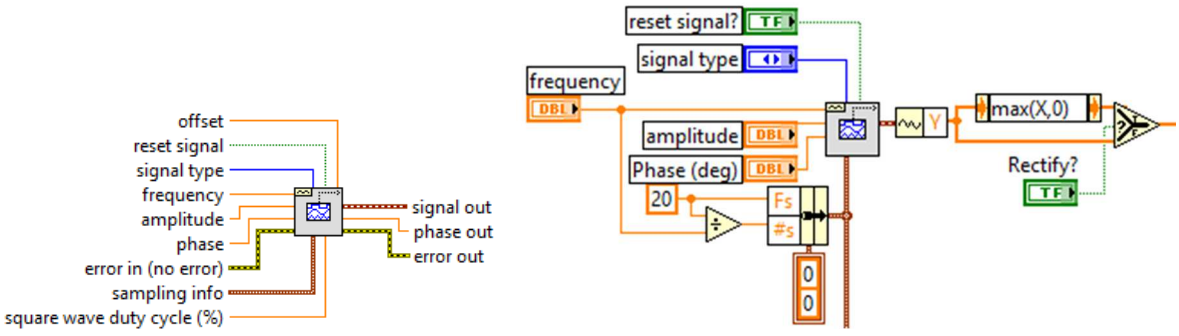

Figure 43. Block diagrams of the basic function generator vi and of the tracking waveform generator.

The internal model generator is composed of a 0.2Hz sine and cosine wave (1st harmonic), a 0.4Hz sine and cosine wave (2nd harmonic), and a 0.6Hz sine and cosine wave (3rd harmonic) as presented in Figure 44. Note that the frequencies relative to the previous simulation is time scaled by 1/5th. This scaling was necessary due to the size of the motors used as well as current limitations of the power supply. By slowing down the simulation by 5x, the acceleration (and hence current) is reduced by 25x. In this way, the effectiveness of the Fourier Repetitive Control algorithm can be tested in spite of having a power supply which is limited in its output. Other than slowing down the basis function 5x as well, the adaptive control algorithm should not change in any way.

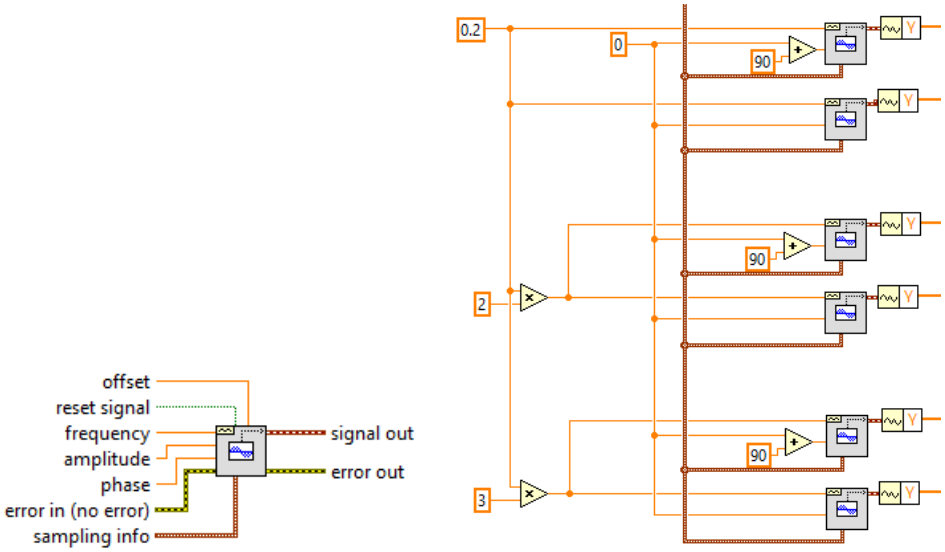

Figure 44. Block diagrams of the "Sine Waveform Generation" (left) and "Internal Model Implementation" (right) in LabVIEW.

The sine waveform generator stops after each call. To ensure the internal model and tracking signal outputs are periodic, both the internal model and tracking signal generator are put inside a for loop that continuously run until a Stop button on the front panel is pressed.

The control voltage is composed of several components:

- A 2nd harmonic disturbance with controllable magnitude,
- A feedback that stabilizes the system,
- A constant term,
- A first, second, and third harmonic sine and cosine, and
- A selectable constant term or stiction compensation, which will be discussed in section 3.5.

All of the components are summed and written to an analog output channel (AI1 in this study) that controls a motor drive (Figure 45).

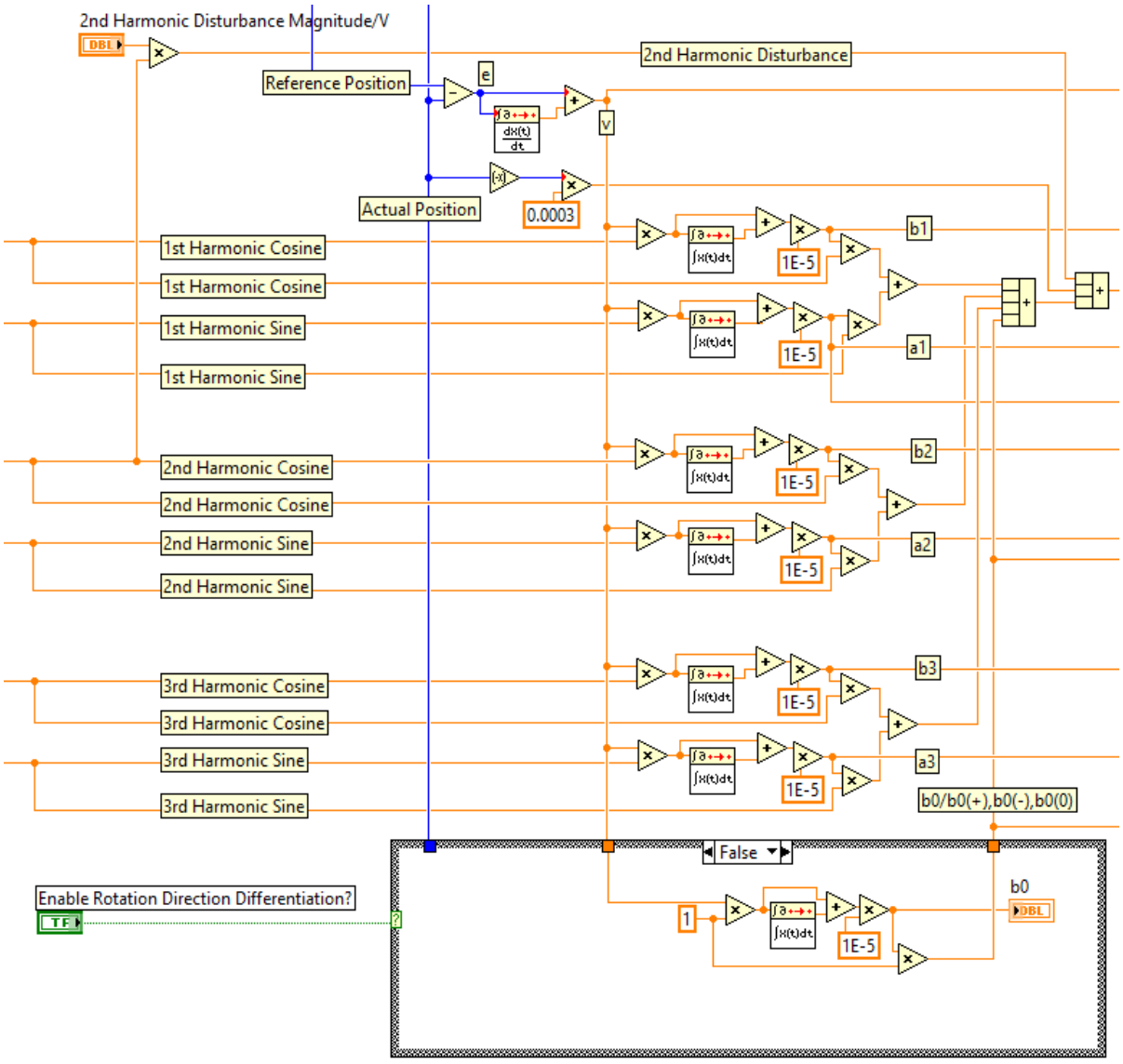

Figure 45. Block diagram for voltage output calculation (summation).

The control signal is supplied by an Analog Output port from the DAQ card (NI PXI-6040E, 2 analog output ports including AO 0 and AO 1). AO 1 is used to supply the control signal from the algorithm to a DC servo motor controller. It is obvious that the DAQ has only limited power output (5mA at 10V) and does not supply enough current to drive a DC motor. An analog servo

drive (30A20ACT, Advanced Motor Controls) interfaces the control signal from the DAQ analog output port to the DC motor.

3.4.2. Hardware setup

The analog servo drive receives a ±10V analog command and outputs a maximum continuous current of 15 Amps that drives the DC motor. The control signal is connected to Pin 4 and 5, which are +Ref In and –Ref In, respectively. The differential reference input operates within a ±10V range.

According to the specification of the analog servo drive, a total of 4 modes can be selected by flipping 10 dip switches. For the purpose of supplying a voltage controlled voltage, the analog servo drive is set in VOLTAGE mode by setting the dip switch as shown in Table 3:

Table 3. DIP switch status for voltage controlled voltage output

Switch Number	Status	Description
SW1	ON	Voltage feedback.
SW2	OFF	IR compensation.
SW3	ON	Current loop proportional gain adjustment
SW4	OFF	Inner (current) loop integral gain adjustment.
SW5	ON	Current scaling.
SW6	OFF	Current limit ratio.
SW7	OFF	Current loop integral gain.
SW8	OFF	Outer loop integration.
SW9	OFF	Outer loop integral gain adjustment.
SW10	OFF	Test/Offset.

The front panel of the code is shown in Figure 46. It is divided into 2 parts – left and right.

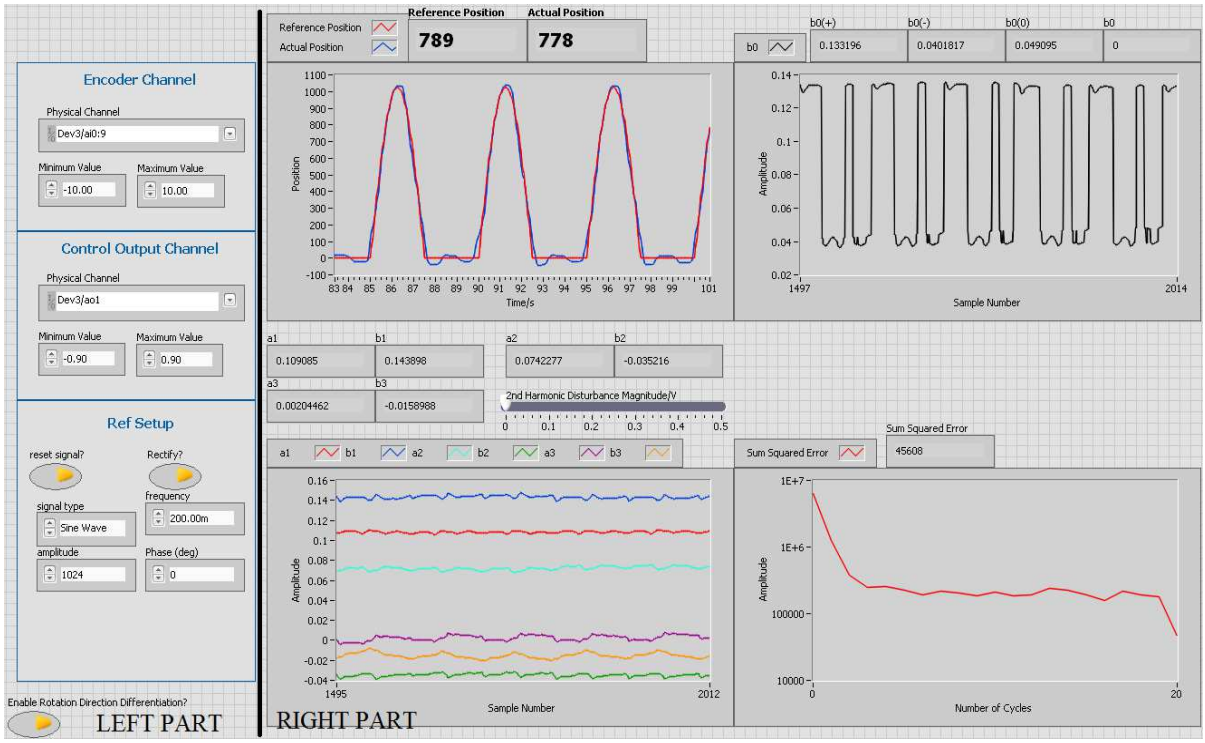

Figure 46. Front panel of the position control user interface.

The left part of the front panel are settings of the encoder input channel, control output channel and reference setup. It is has three sections, namely, Encoder Channel, Control Output Channel and Ref Setup. The button "Enable Rotation Direction Differentiation?" controls the enable status of stiction compensation as discussed in section 3.5

The DAQ being used for this research shows up on NI Measurement and Automation Explorer as "Dev3". Ten analog input channels (ai0 through ai9) are used for encoder input and hence the physical channel for the encoder is set up as "Dev3/ai0:9. The control output directly feeds into the differential reference input ports of the analog servo drive. Analog output channel 1 is used in this case and the physical channel is set up as "Dev3/ao1". Encoder Channel and Control Output Channel settings are not changed throughout the research.

The Ref Setup section specifies the desired position profile. The reset signal? boolean is default to true, that ensures the position profile is periodic. The rectify? boolean controls whether the waveform will be rectified. Signal type enum including Sine Wave, Triangle Wave, Square Wave and Sawtooth Wave controls the shape of the waveform. Sine Wave is the default option. Frequency, amplitude and Phase (deg) specify the frequency, amplitude and phase (in degree) of the waveform, respectively. The first harmonic or fundamental frequency for this research is 0.2Hz, or 200mHz. The default value of amplitude is 1024, or one full revolution.

The right part of the front panel contains 4 charts, numerical displays and a numerical control slider (2nd Harmonic Disturbance Magnitude/V). The top left chart shows reference position (blue line) and actual (measured) position (red line) with time in seconds as its *x* axis. The two numerical displays present the set position and actual (measured) position in real time. The top right chart displays the constant term b_0 or the other three terms $b_0(+)$, $b_0(-)$ and $b_0(0)$ combined by toggle between "Enable Rotation Direction Differentiation?" Boolean. More detailed discussion will be carried out in section 3.5. The bottom left chart shows the real-time values of a_1 (red), b_1(blue), a_2 (cyan), b_2 (green), a_3 (magenta) and b_3 (orange) with sample number (the number of each calculation, which happens every 50 ms) as its *x* axis. A "2nd Harmonic Disturbance Magnitude/V" numerical controller is for controlling the second harmonic disturbance supplied to the output. The bottom right chart shows sum squared error of each cycle.

3.4.3. Position control results

To test the hardware implementation of this algorithm, the same set points and disturbances which were used in simulation are applied to the actual motors. In order to quantify the "quality" of the tracking, the sum-squared error for the last cycle after 100 seconds (two cycles) is used.

$$SSE = \sum_i e_i^2 \tag{32}$$

3.4.3.1. Tracking a pure sine wave

First, consider the case of tracking a first-harmonic sine wave with no disturbance. This is done by sliding the "2nd Harmonic Disturbance Magnitude/V" to the leftmost position and set the parameters in the Ref Setup section as in Table 4 before starting the program:

Table 4. Front panel setup for tracking first harmonic sine wave

Front Panel Control Name	Value
Reset signal?	True
Rectify?	False
Signal type	Sine Wave
Frequency	200.00m
Amplitude	1024
Phase (deg)	0
Enable Rotation Direction Differentiation?	True

The experiment results are shown in Figure 47. Note that the output is a 1Hz sine wave as desired (top left figure). b0 (top right) is fluctuating within a small range (-0.013 to -0.003). The Fourier coefficients (lower left) contain 1st harmonic terms (a1 and b1) while the other

coefficients are nearly zero, again as expected. Finally, the sum-squared error (lower right) does not go to zero – but converges to approximately 160,000.

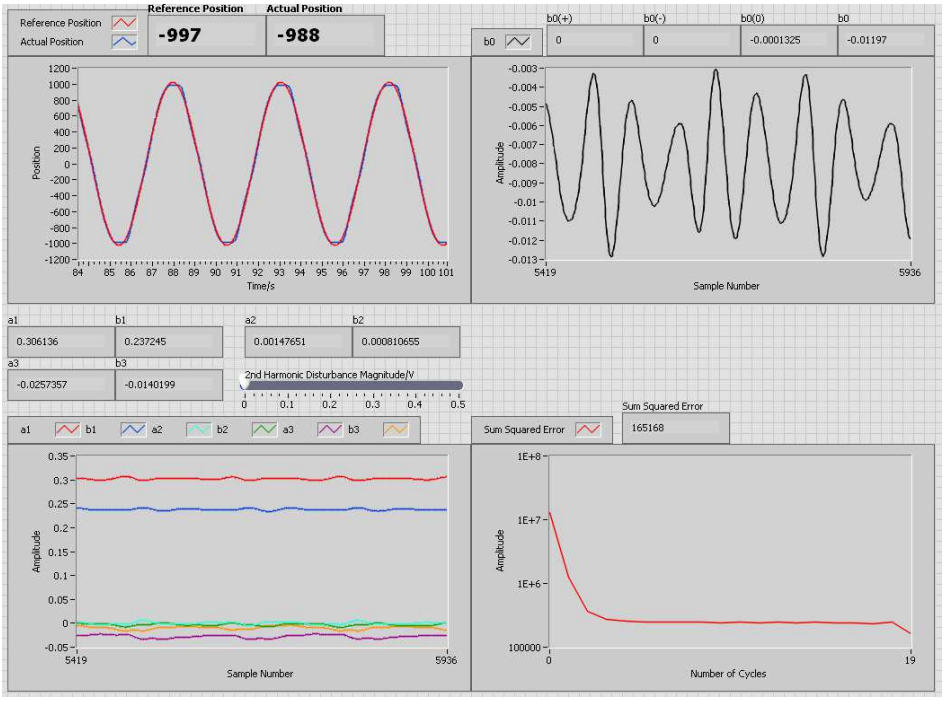

Figure 47. Experiment results on tracking first harmonic sine wave with no disturbance.

Next, consider the case where the set point is a 2Hz sine wave. The results from repeating this experiment at 100 seconds is presented in Figure 48. As predicted by the simulation, the motor tracks the set point (top left) and the Fourier coefficients for the 2nd harmonic (a_2 and b_2) converge to a constant while the other parameters become nearly zero. b_0 (top right) is also close to zero as predicted. The final SSE is around 210,000 (bottom right).

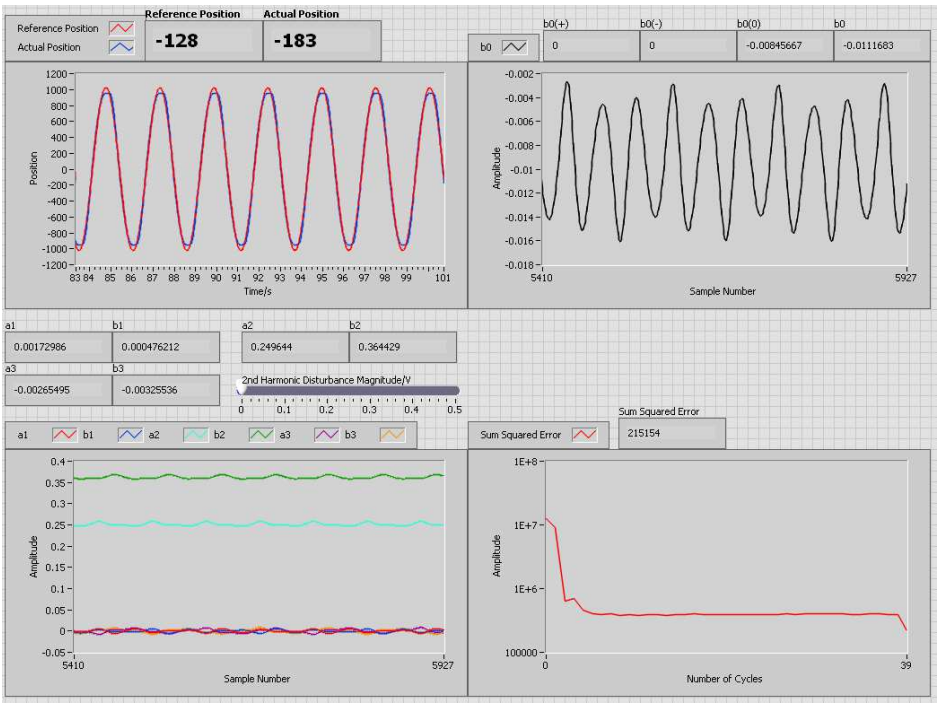

Figure 48. 2nd harmonic (400 mHz) reference profile as input at 100 seconds. a_1, b_1, a_3 and b_3 are close to zero while a_2 is between 0.1 and 0.15 and b_2 is between -0.3 and -0.35.

3.4.3.2. Tracking a half-wave rectified sine wave

Next, consider the case where the motor is to track a half-wave rectified sine wave. From previous simulation experiments, it is expected that both first harmonic (a_1 and b_1) as well as second-harmonic (a_2 and b_2) terms should be present in steady-state. This is exactly what happens with actual hardware as presented in Figure 49 and 50. In Figure 49, the tracking for the first 10 seconds is presented. Note that the plant does not track the set point well over this time span. In Figure 50, the tracking at 100 seconds is presented. Note that the tracking is fairly good (top left) and the Fourier coefficients contain both 1st and 2nd harmonics as the simulation results predicted (bottom left). b_0 on the top right displays a periodic fluctuation that ranges from 0.084 to 0.1.

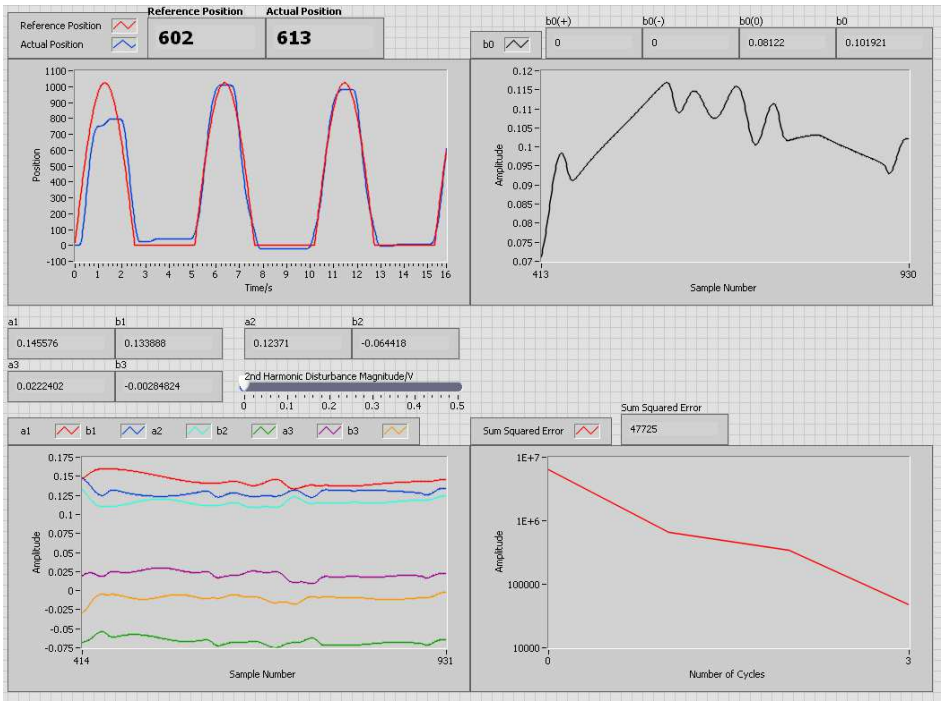

Figure 49. The waveform of the first 15 seconds when the reference profile is a half-rectified sine wave.

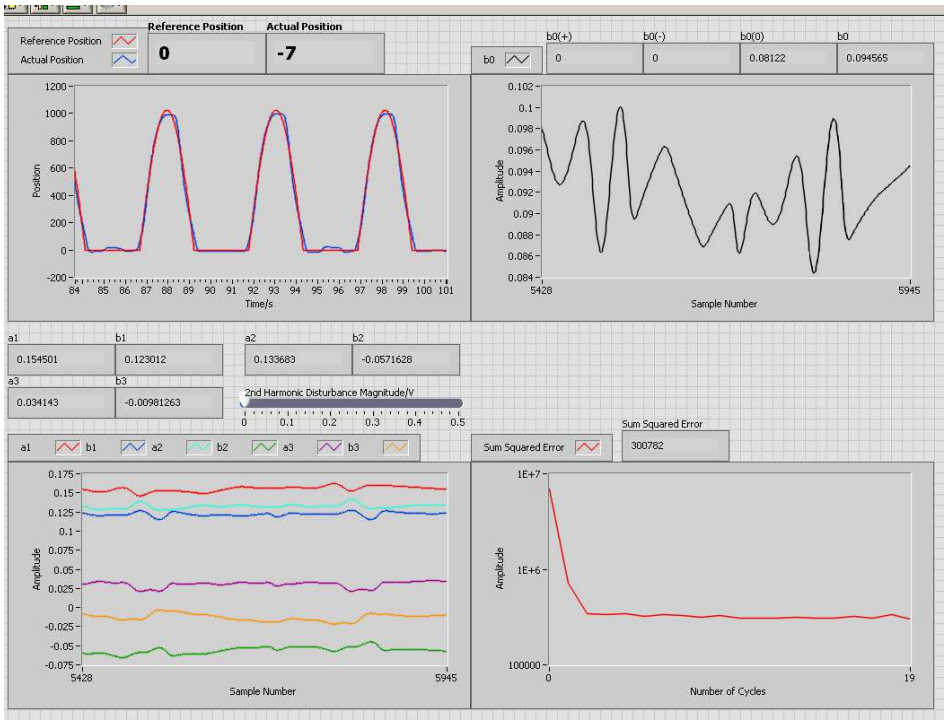

Figure 50. The waveform of 85 to 100 seconds when the reference profile is a half-rectified sine wave.

The Fourier coefficients at 100 seconds are summarized in Table 5.

Table 5. Tracking results for different tracking waveforms

Tracking Waveform	Final SSE	Final RMS Error (degrees)
$1024\sin(2\pi \times 0.2t)$	244880	12.30
$1024\sin(2\pi \times 0.4t)$	387794	15.48
$\max\{1024\sin(2\pi \times 0.2t), 0\}$	336204	14.41

3.4.3.3. Second harmonic disturbance

To illustrate the system behavior when a second harmonic (0.4 Hz) disturbance is added to the control signal output, a "2nd Harmonic Disturbance Magnitude/V" numeric control was added to the program, Figure 51.

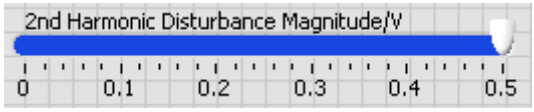

Figure 51. Numeric control for controlling the 2nd harmonic disturbance.

Because a second harmonic disturbance is added to the program, when tracking a certain profile, the coefficients for compensating second harmonic disturbance a_2 and b_2 should be non-zero values, in addition to the other coefficients when tracking the corresponding profile. The SSE should still be a small value when stabilize since the second harmonic bases should offset the disturbance.

The results of tracking the 1st harmonic sine wave profile when the 2nd harmonic disturbance presents is in Figure 52. The actual positions match the profile position well (top left) and in addition to the non-zero a1 and b1, which are expected when tracking 1st harmonic sine wave, there is a pronounced second harmonic component b2, with a magnitude of -0.5. The constant term b0 fluctuates in a small range (-0.02 to -0.006) around zero (top right). The final SSE is 320911, corresponding to an RMS error of 14.08 degrees.

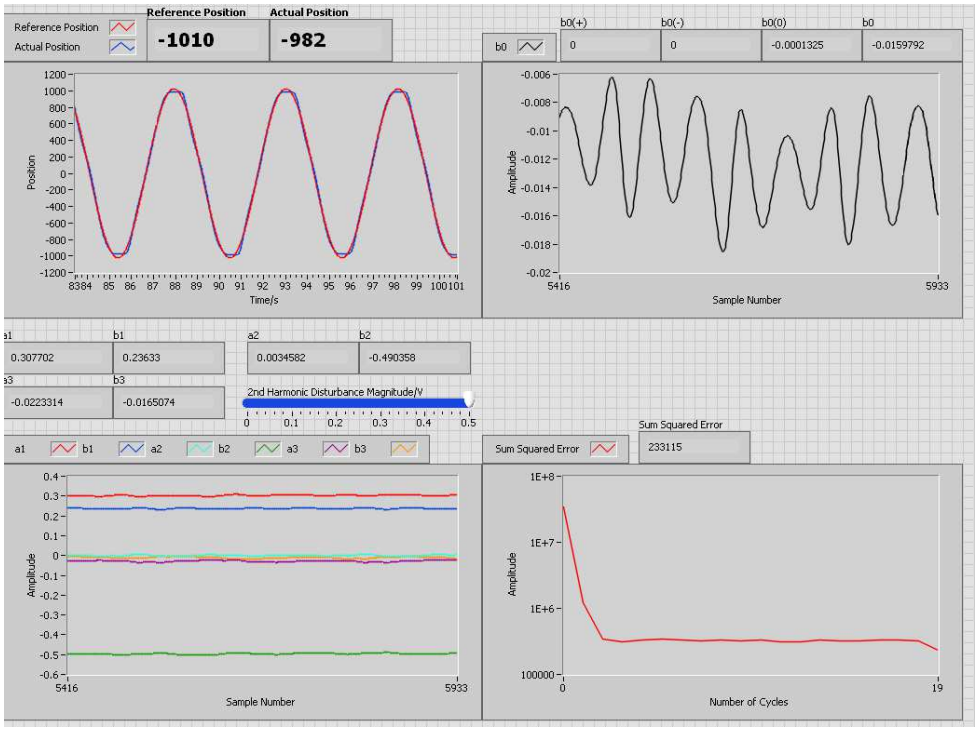

Figure 52. Tracking 1st harmonic sine wave when the 2nd harmonic disturbance presents.

When tracking a second harmonic sine wave with added 2nd harmonic disturbance, only the coefficients for the second harmonic bases (a_2 and b_2) should be non-zero values while the other coefficients remain zero. The results in Figure 53 display anticipated results – good tracking (upper left), near zero values of coefficients except for a_2 and b_2 (lower left and upper right),

decreased SSE over time and final SSE is 418926 (corresponding to an RMS error of 16.09 degrees).

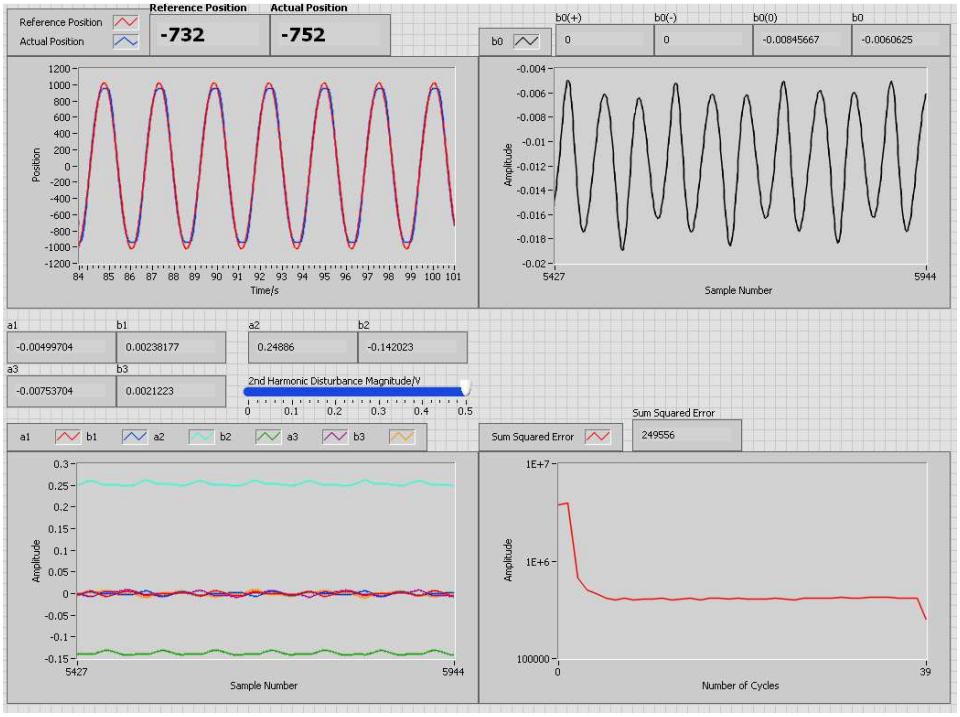

Figure 53. Tracking 2nd harmonic sine wave when the 2nd harmonic disturbance presents.

The tracking results when tracking a half rectified sine wave with the 2nd harmonic disturbance should be the same as tracking half rectified sine wave without disturbance, only a_2 and/or b_2 are different because of the disturbance. The results shown in Figure 54 validate the prediction.

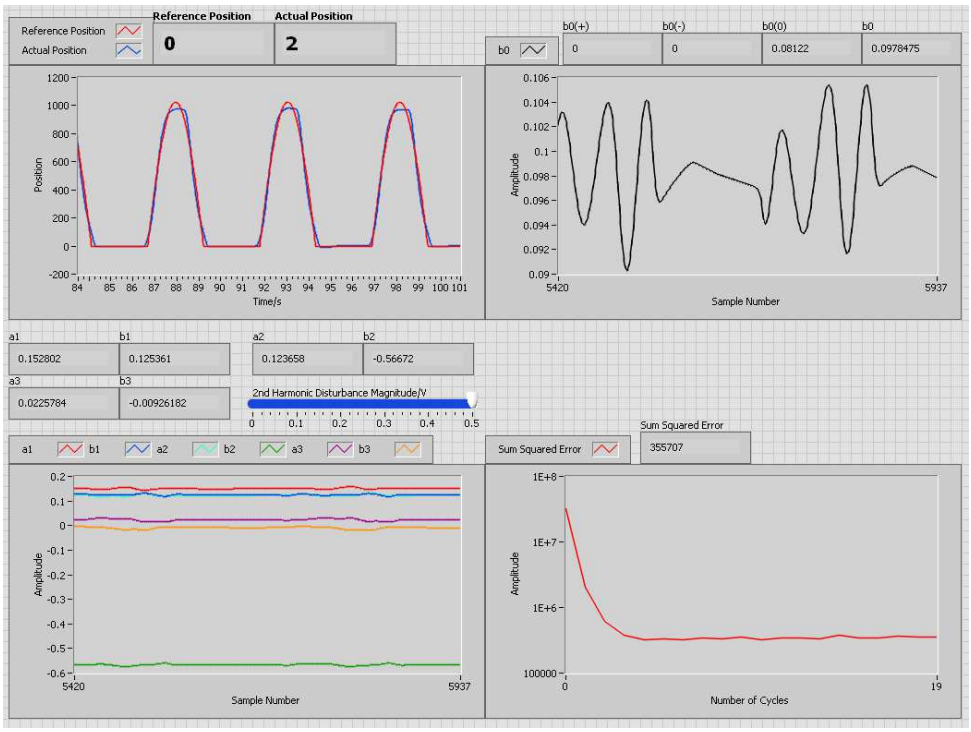

Figure 54. Tracking half rectified sine wave when the 2nd harmonic disturbance presents.

The tracking results at 100 seconds are summarized in Table 6.

Table 6. Tracking results for different tracking waveforms

Tracking Waveform	Final SSE	Final RMS Error (degrees)
$1024\sin(2\pi \times 0.2t)$	320911	14.08
$1024\sin(2\pi \times 0.4t)$	418926	16.09
$\max\{1024\sin(2\pi \times 0.2t), 0\}$	350802	14.72

It can be seen from the final error that the algorithm effectively rejects the second harmonic disturbance.

3.5. Algorithm Improvement: Static Friction Compensation

With the adaptive control scheme used, perfect tracking depends upon the basis chosen spanning the input required for perfect model following (termed the perfect model following conditions [Landau 1979], [Popov 1963]). In simulation studies, this was achieved and the error was driven to approximately zero in all cases save for the half-wave rectified sine wave set-point. When applied to an actual motor, however, perfect tracking was never obtained. The error in the tracking tended to be a square wave related to the direction of motion. This can be explained by the presence of static friction in the motors.

In order to cancel the effect of static friction, an eighth term is added to the basis function of (3.1) equal to the sign of the direction of motion:

$$u(t) = b_0 + c_0 sign\left(\frac{d\theta}{dt}\right) + \sum_{i=1}^{3} a_i \sin(n\omega t) + b_i \cos(n\omega t) \tag{33}$$

The implementation of stiction compensation implementation is shown in Figure 55.

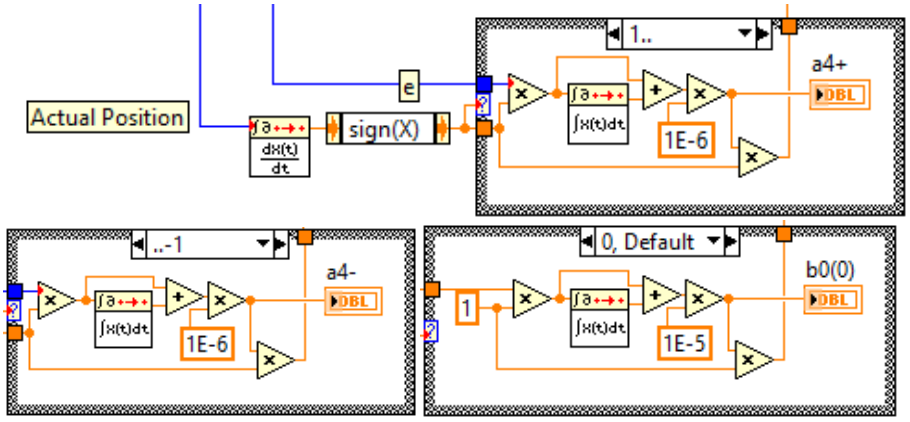

Figure 55. Block diagram of stiction compensation implementation in LabVIEW.

When tracking 1st harmonic sine without disturbance, it can be seen that there are mismatches at crest and trough of the tracking waveform (Figure 56). When the stiction compensation is enabled, it can be seen from Figure 57 that the mismatches at crest trough of the waveform are reduced:

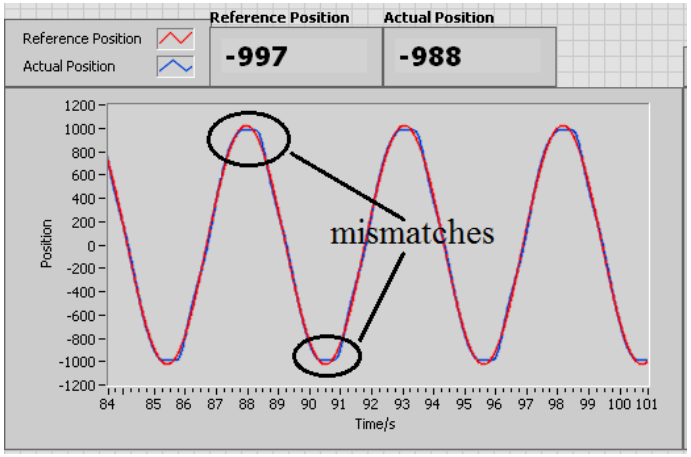

Figure 56. Tracking 1st harmonic sine without disturbance, showing mismatches without stiction compensation.

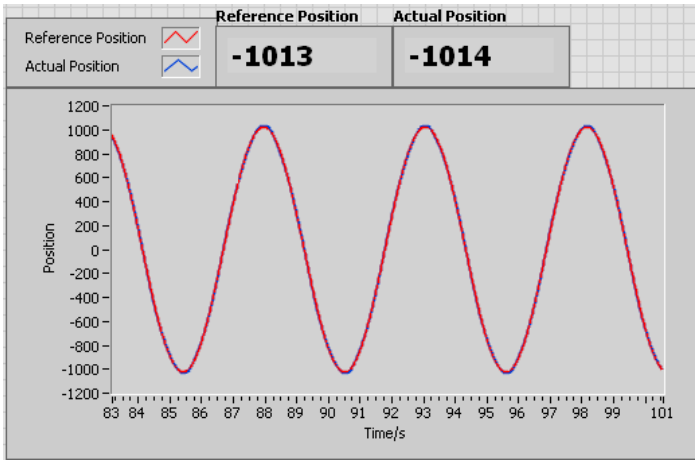

Figure 57. Tracking 1st harmonic sine without disturbance, showing improvements in mismatches at crest and trough with stiction compensation.

With stiction compensation, the final Sum Squared Error also reduced to 26291 (RMS error of 4.03 degrees) as presented in Figure 58, compared to SSE = 244887 (RMS error of 12.30 degrees) without stiction compensation.

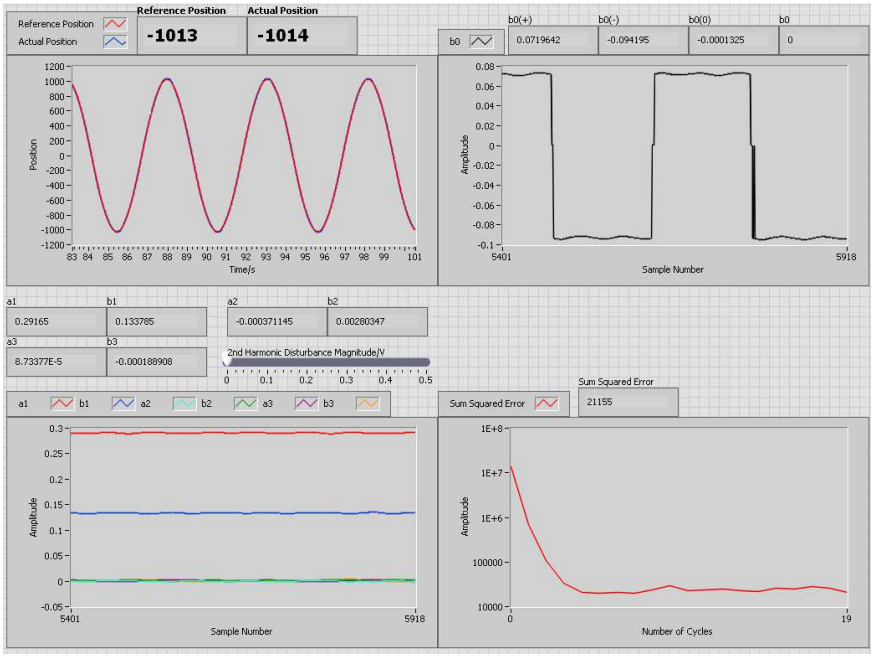

Figure 58. Tracking 1st harmonic sine wave without disturbance, when stiction compensation is enabled.

3.6. Conclusion

This chapter presented a position controller including both hardware and software composition, simulation and experiment results. It has been found that the control algorithm is able to track a sinusoidal waveform profile when that frequency is included in the bases. It also effectively rejects disturbances with known frequency. The experiment differs from simulation in that actual motors have stiction that cannot be compensated for by only using only a constant term and harmonics. A stiction compensation technique is proposed by the author to address the problem that shows improvements.

CHAPTER 4. CURRENT CONTROL

4.1. Chapter Introduction

In this chapter, the hardware for controlling the current to a DC servo motor is presented. Current is controlled since it is equal to the torque produced by the motor through the torque constant. The current is also used as a torque disturbance for the position controller presented in the following chapter.

4.2. Methodology

A DC servo motor controller (Advanced Motion Controls, 30A8, 20-80V DC input, 15A continuous, 30A peak output) configured in CURRENT mode is used to supply current to a DC motor that is identical to the DC motor used in position control development. The supply voltage for the motor controller is 20 – 80 VDC. The DC power supply (RSR HY3002-3) has maximum voltage output of 30V DC and is used to supply power to the servo motor controller.

The control signal is provided by the Agilent arbitrary waveform generator. It has the capacity of supplying a sinusoidal waveform of 200 mHz, 400 mHz and 600 mHz frequency.

It is expected that when set to CURRENT mode, the DC servo motor controller acts as a voltage controlled current source. The current output should be proportional to control voltage input.

4.3. Design and Develop

According to the specification, the DC servo motor controller can be set to CURRENT mode by setting the DIP switches on the controller as illustrated in Table 7.

Table 7. DIP switch for setting 30A8 motor servo controller in CURRENT mode

Switch Number	Status	Description
SW1	OFF	Voltage feedback.
SW2	OFF	IR compensation.
SW3	ON	Current loop proportional gain adjustment
SW4	OFF	Inner (current) loop integral gain adjustment.
SW5	ON	Current scaling.
SW6	OFF	Current limit ratio.
SW7	OFF	Current loop integral gain.
SW8	ON	Outer loop integration.
SW9	OFF	Outer loop integral gain adjustment.
SW10	OFF	Test/Offset.

The DC servo motor controller has two connectors – the Power Connector and the Signal Connector. Power is supplied by the DC power supply. Connect the DC power supply GND to pin 4 (PWR GND) on the Power Connector and DC power supply 30V output to pin 3 (HIGH VOLT) on the power connector. Pin 1 and pin 2 of the Power Connector are negative motor output and positive motor output, respectively.

The signal output from the waveform generator is connected to Pin 4 and 5 of Signal Connector, which are +Ref In and –Ref In, respectively. In this research, the differential reference input is ±0.5 V. The wiring schematic is shown in Figure 59.

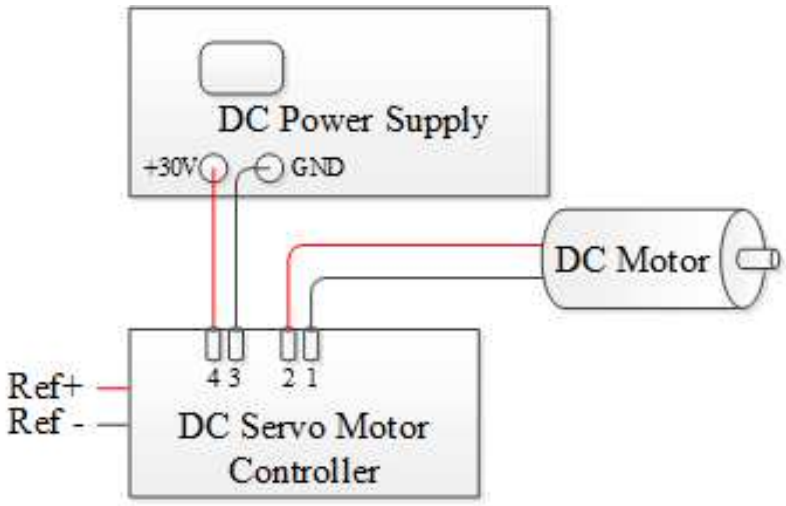

Figure 59. Current control wiring schematic.

4.4. Testing/Verification

The motor controller drives the motor using Pulse Width Modulated (PWM) signal. The pulse frequency tends to be in the order of tens of mega Hz. Hence, the current cannot be directly measured by measuring voltage across a resistor that is in series with the motor controller output. A Current Monitor Out port (pin 8) on the Signal Connector is used to measure current output. According to the specification, the analog voltage output of this channel is proportional to the current output. The scaling is 3.8 A/V.

Due to the PWM control scheme used by the motor controller, the environment is contaminated with electromagnetic noise. To cope with the noisy environment, a 220 μF capacitor is used to filter out spikes. That results in a phase lead of $\pi/2$ which is compensated for later in LabVIEW.

Three tests were run in order to test the dynamic response of the DC servo drive. First, a 0.2Hz, 500mV peak sine wave is applied to the current amplifier while the position control

motor is turned off. In Figure 60, the command signal and the actual current is plotted. Note that the current follows the set point as desired.

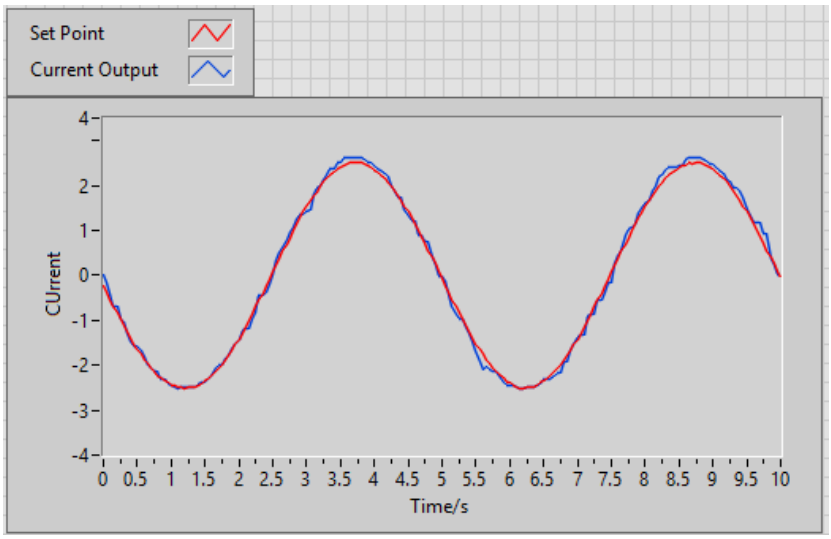

Figure 60. Current setpoint (red) and actual current output (blue) for a 0.2 Hz sinusoidal input.

Next, a 0.4Hz sinusoidal setpoint is applied to the current amplifier while the position controller is turned off. The results are presented in Figure 61. Note that the current again tracks the desired set point.

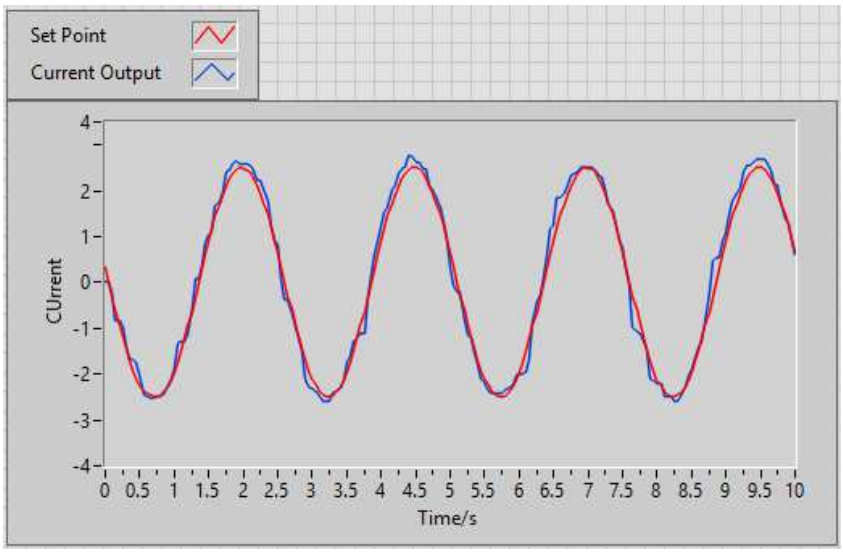

Figure 61. Current setpoint (red) and actual current (blue) for a 0.4 Hz sinusoidal input.

Finally, a 0.2Hz ½ rotation sinusoidal setpoint is applied to the position control while a 0.4Hz sinusoidal setpoint is applied to the current amplifier with the results presented in Figure 62. Note again that the current controller tracks the set point nearly perfectly.

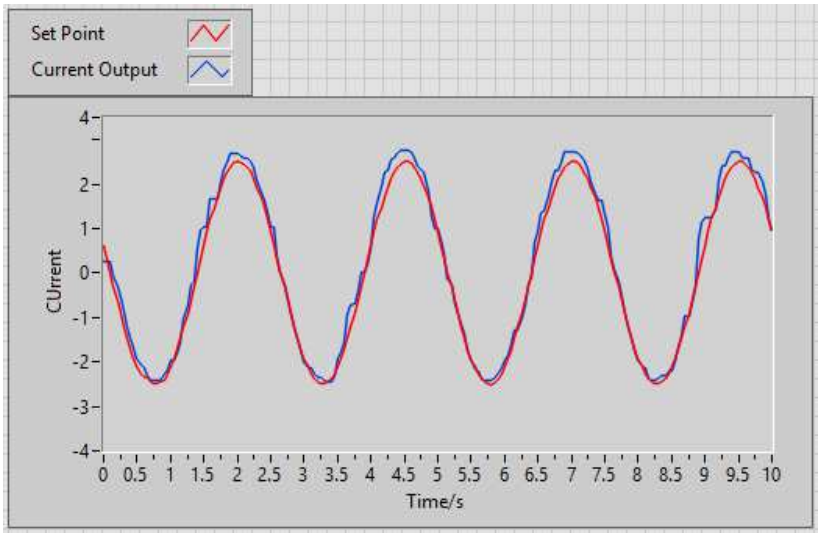

Figure 62. Current setpoint (red) and actual current (blue) for a 0.4 Hz sinusoidal input for current control and a 0.2 Hz sinusoidal input for position control.

4.5. Results

From the above analysis it can be concluded that the 30A8 DC servo controller can be used to supply current linear to control voltage to the DC motor used in this research. The DC power supply provides adequate power to the controller and the motor. The linearity performances in both DC and dynamic performance when tracking 200 mHz and 400 mHz AC are also proven to be adequate for this research.

CHAPTER 5. COMBINED POSITION AND CURRENT CONTROL

5.1. Chapter Introduction

In this chapter, the position control (detailed in Chapter 3) and current control (detailed in Chapter 4) are combined together. The two theoretically identical DC motors are mechanically coupled together via a coupler as shown in Figure 63. The idea is that while one motor controls the position of the pair, the other motor controls the torque at the junction of the motors by controlling the current to the secondary motor (which is related to torque by the motor's torque constant.) Likewise, with this combination, both position and torque can be controlled at the junction of the two motors.

This chapter presents the test results for simultaneous control of position and torque.

5.2. Test Setup

Because these two motors are mechanically coupled as presented in Figure 63, they share the same position at any given moment. The left motor is driven by the DC servo motor controller in CURRENT mode, which in turn is controlled by a waveform generator. Therefore, the current drives the motor can be controlled and modulated via selecting the desired waveform on the waveform generator. This current is proportional to the torque produced by the motor through its torque constant.

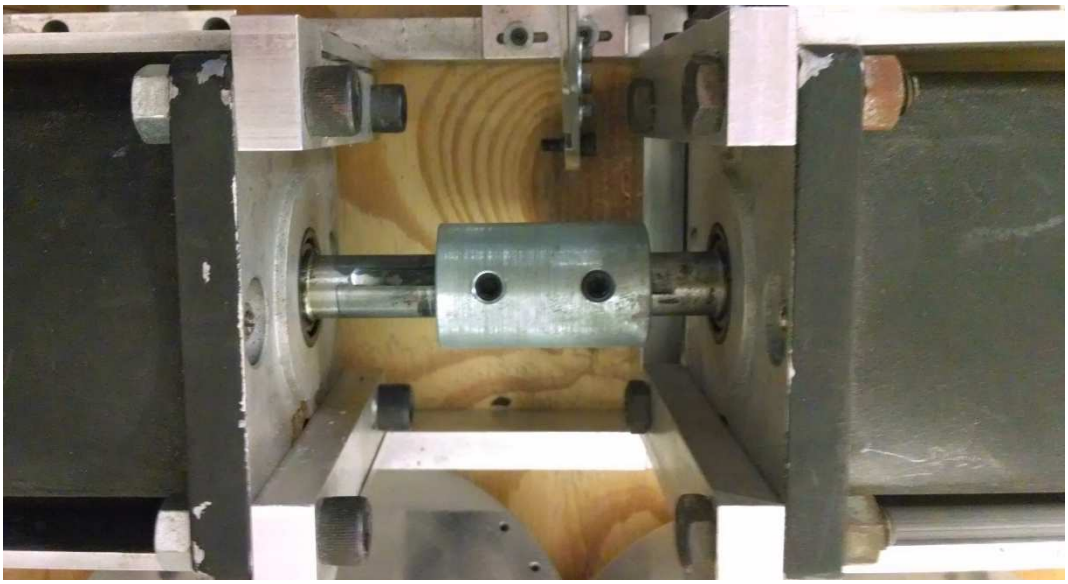

Figure 63. The coupler that mechanically couples two motors.

The motor on the right side is driven by a standalone DC servo motor controller whose power source is from the wall plug. It is set in VOLTAGE mode, which means its voltage output is proportional to the control voltage. The Fourier repetitive control algorithm from Chapter 3 controls the position of this motor with a sampling rate of 50ms. Experiments are done on various tracking profiles and disturbance profiles.

The same LabVIEW program used in position control (Chapter 3) is used in this chapter. The "2nd Harmonic Disturbance Magnitude/V" slider is set to 0 so there is no disturbance within the Fourier repetitive control algorithm itself. The "Enable Rotation Direction Differentiation?" Boolean is set to True to compensate for the stiction.

5.3. Results

First, consider the case where the position controller tracks a 200mHz sine wave (1st harmonic) while the current controller tracks a 400mHz sine wave (2nd harmonic), the results of

which are presented in Figure 63 and Figure 64. Initially, the tracking is poor (top left plot in Figure 63) – but as the adaptation algorithm determines the required weightings on the Fourier coefficients for $u(t)$, the error is quickly reduced. After 100 seconds (Figure 64), the position controller is tracking the set point very well (top left plot of Figure 64). Furthermore, the Fourier coefficients (lower left plot of Figure 64) include both the first harmonic (a_1 and b_1) to track the set point as well as second harmonic terms (a_2 and b_2) to reject the 2nd harmonic disturbance from the current control motor.

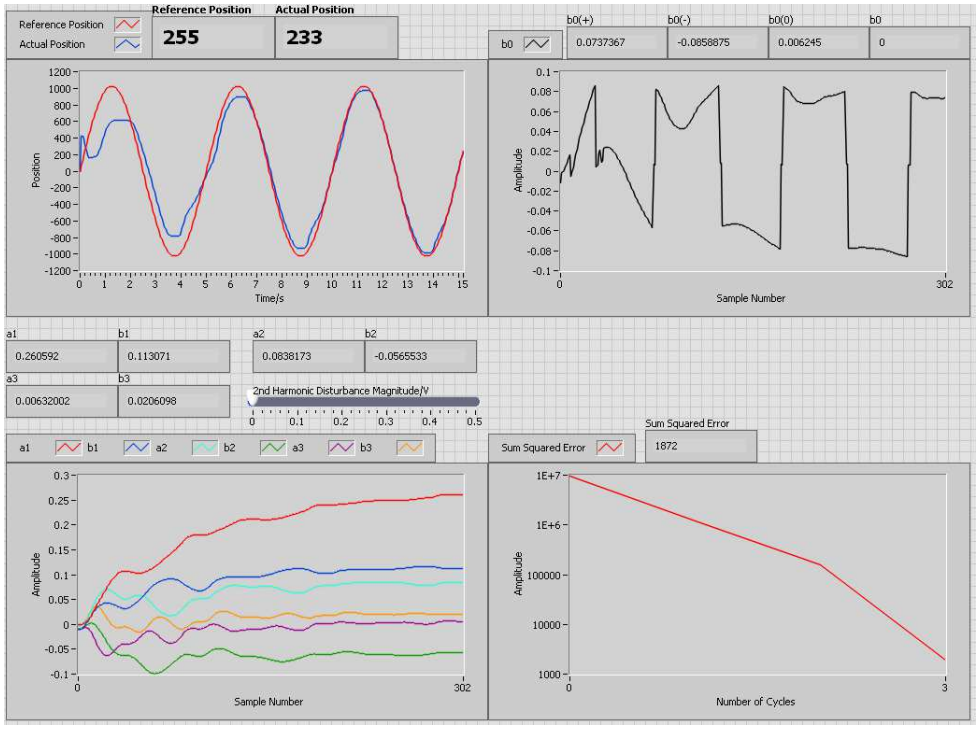

Figure 64. The first 15 seconds of tracking 200 mHz sine wave with 400 mHz external disturbance on the position control motor.

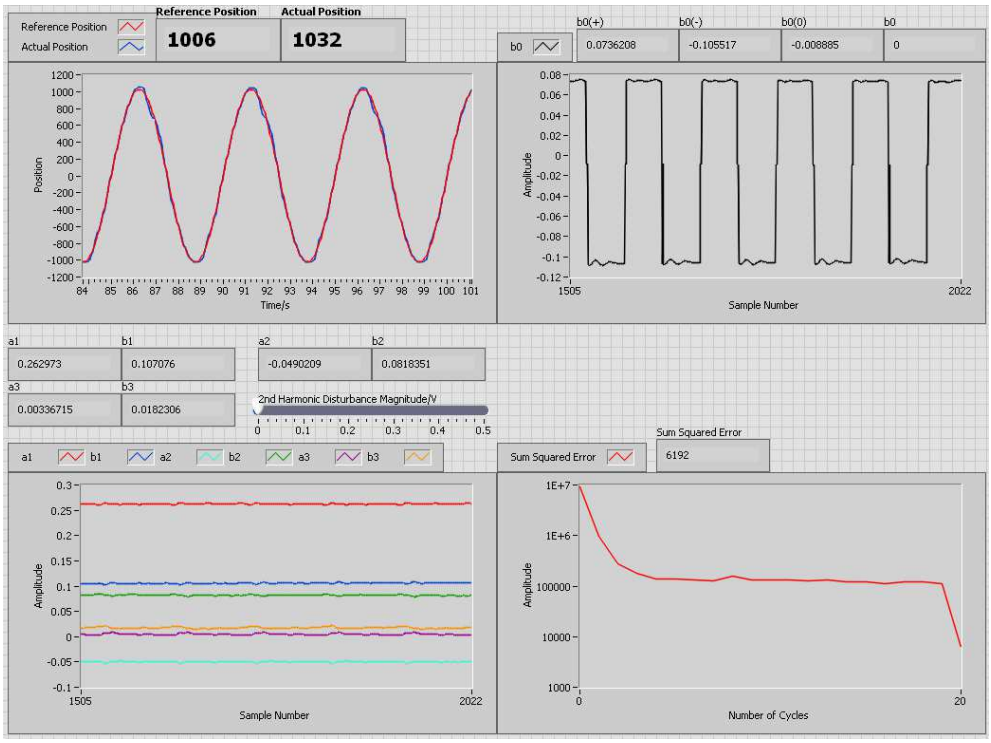

Figure 65. The last 15 seconds of tracking 200 mHz sine wave with 400 mHz external disturbance on the position control motor.

The current in the second motor is presented in Figure 65. Note that the current is tracking the 400 mHz sine wave (2nd harmonic) in spite of the 20 mHz disturbance provided by the position of the first motor.

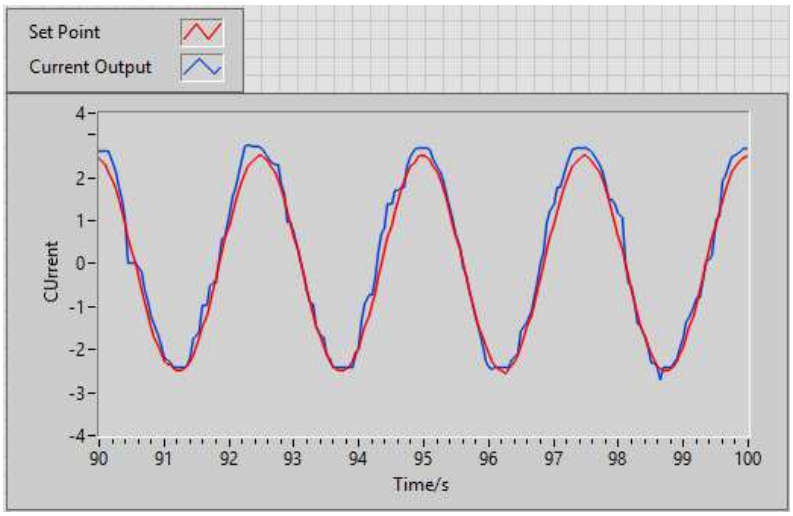

Figure 66. The last 10 seconds of tracking a 400 mHz sine wave with a 200 mHz external disturbance on the current-control motor.

Next, consider the case where the disturbance is at the same frequency as the set point. In this experiment, the position controller is to track a 0.2Hz sine wave while the current controller also tracks a 0.2Hz sine wave. The result of this experiment is presented in Figure 66- Figure 68. In Figure 66, the tracking error for the first 15 seconds is presented. Note that initially the tracking is again poor due to the initial values of the Fourier coefficients for $u(t)$ being off. After a short time, however, the tracking error is reduced. After 100 seconds (Figure 67), the position controller tracks the set point in spite of the disturbance on torque provided by the second motor (top left plot of Figure 67). Moreover, the Fourier coefficients are primarily the first harmonic (a_1 and b_2) as expected (lower left plot of Figure 67).

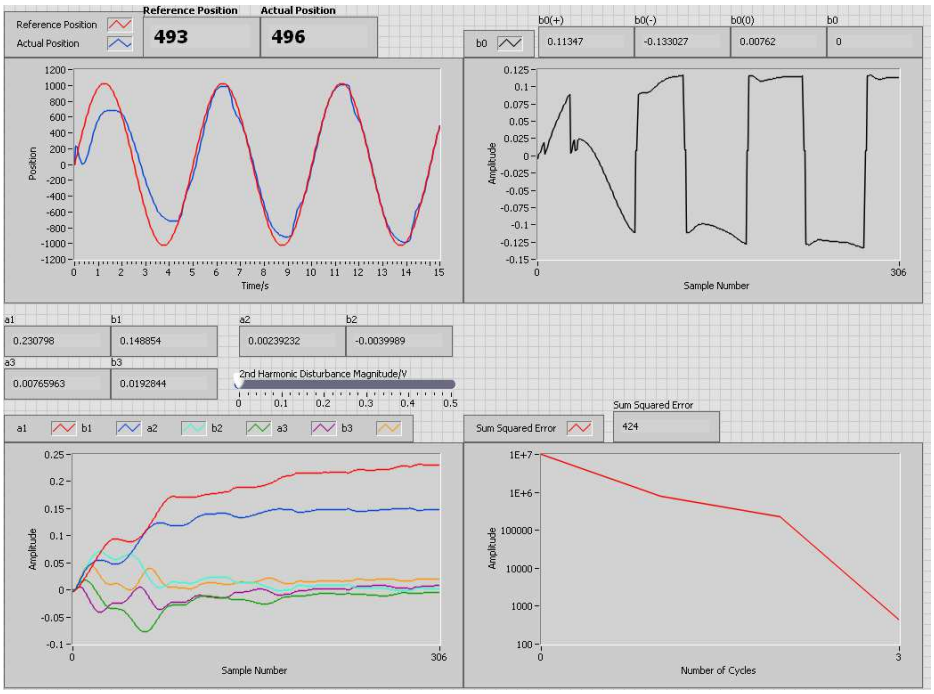

Figure 67. The waveforms of 0 to 10 seconds when the disturbance is 0.2 Hz sine wave.

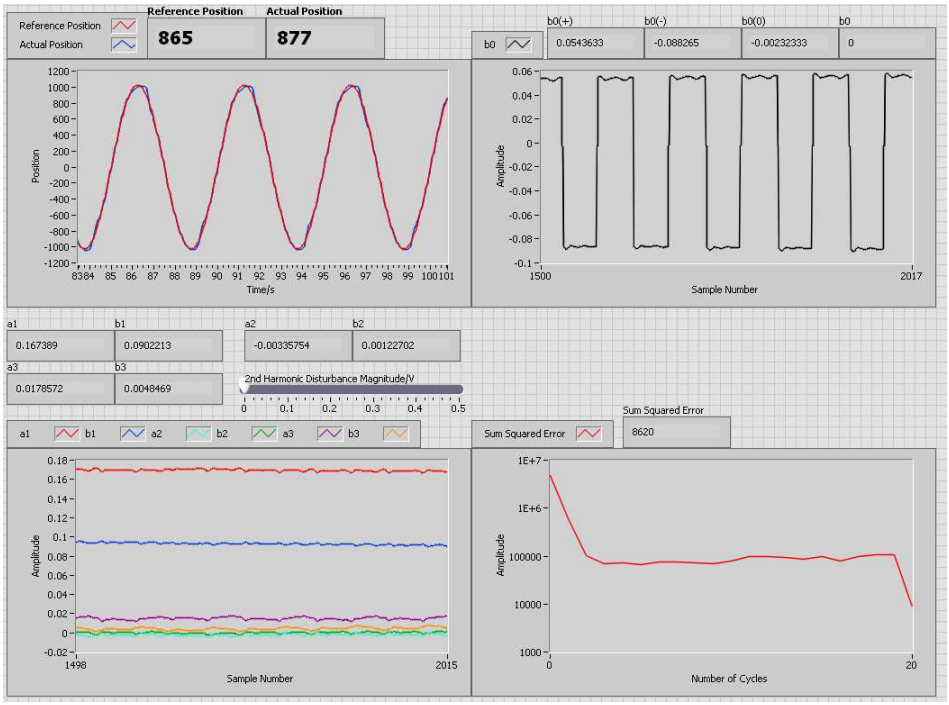

Figure 68. The waveforms of 85 to 100 seconds when the disturbance is 0.2 Hz sine wave.

The resulting current provided to the second motor is presented in Figure 68. Note that the motor tracks the set point in spite of the position disturbance from the 2nd motor.

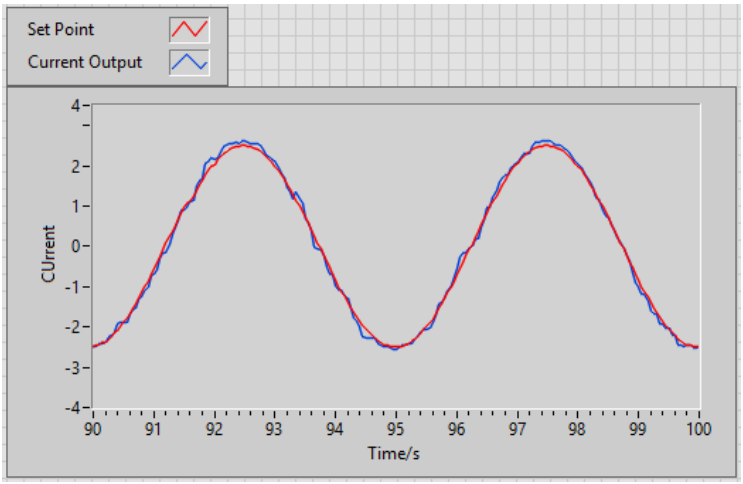

Figure 69. Current to the second motor from 90 to 100 seconds when the disturbance is a 0.2 Hz sine wave.

Finally consider the case where the set point is not spanned by the basis function chosen. In this case, it is expected that the adaptive control algorithm will to the best it can given the basis function provided – but the tracking error will be non-zero for position. Current should track regardless.

In Figure 69 – Figure 71, the tracking of a 1Hz triangle wave (which has harmonics out to infinity) is presented. As before, the initial tracking error is significant (top left corner of Figure 70). After 100 seconds, the error is reduced but is non-zero (top left corner of Figure 70). The Fourier coefficients converge to constants (lower left corner of Figure 70) – which are the best constants the adaptive control scheme can find for minimizing the tracking error.

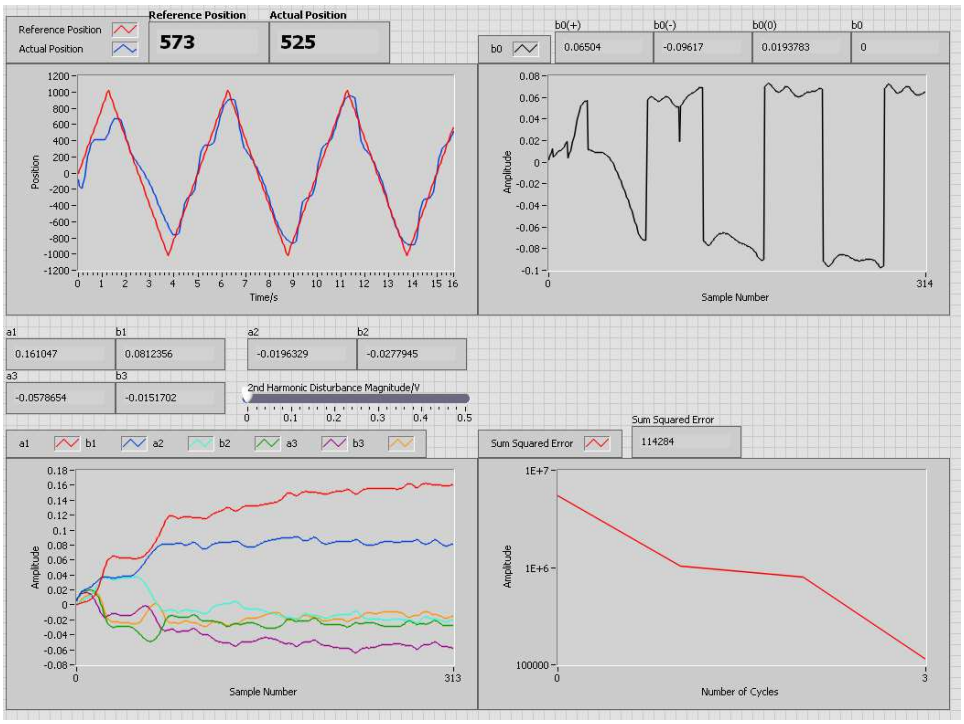

Figure 70. The waveforms of 0 to 15 second when the disturbance is 0.2 Hz sawtooth wave.

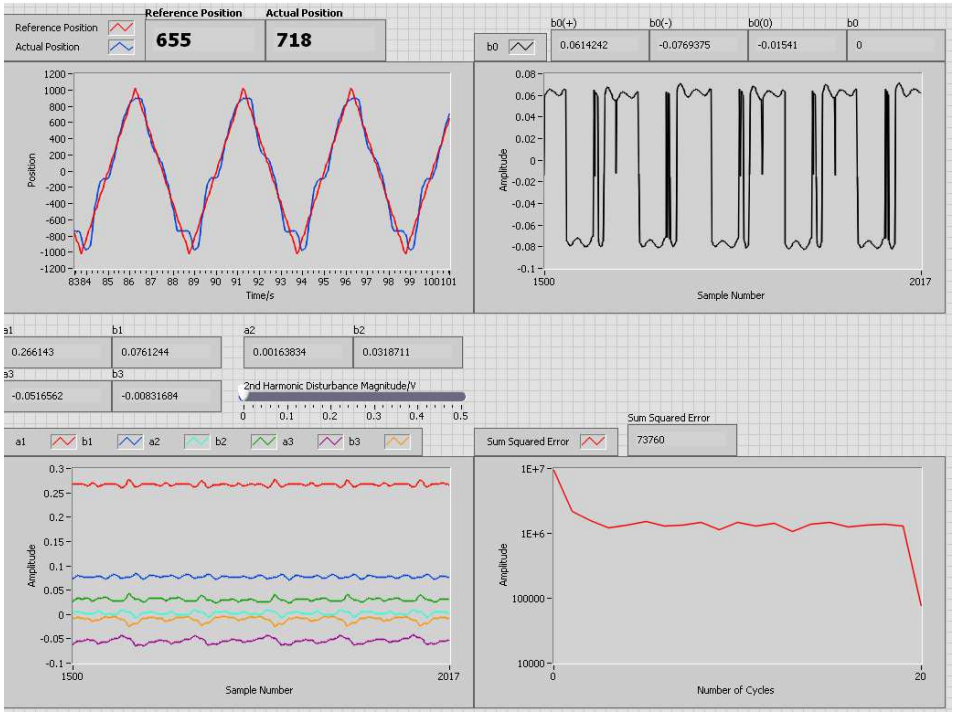

Figure 71. The waveforms of 85 to 100 second when the disturbance is 0.2 Hz sawtooth wave.

In spite of the position tracking a triangle wave, the current controller continues to follow its set point as presented in Figure 72 as expected.

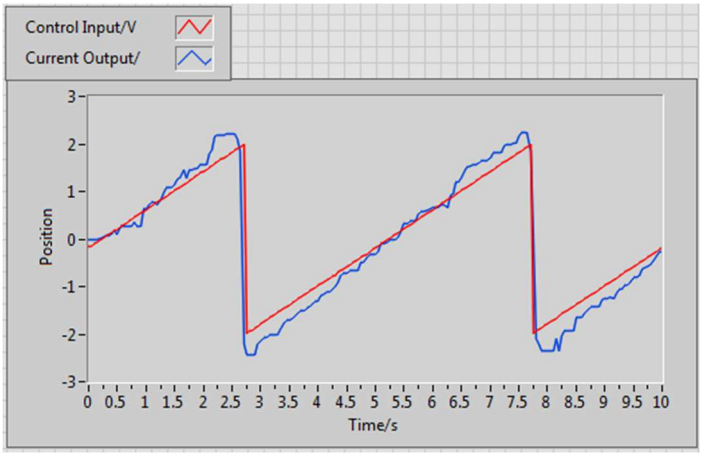

Figure 72. Current to the second motor from 0 to 10 seconds when the position controller tracked a 1 Hz triangle wave.

5.4. Chapter Summary

In this chapter, the position and current controller were put together. It was shown that the position controller was able to reject disturbances provided by the current controller:

- At the same frequency as the position controller,
- At a different frequency of the position controller, and
- When the set point for the position controller contained more terms than the first three harmonics of the Fourier series.

In the last case, tracking errors resulted as expected, but the algorithm did as well as it could with the basis function given. Furthermore, the current controller was able to track its set point in spite of the disturbances provided to it by the position controller.

In short the objective of this thesis has been met: both position and current were controlled using LabVIEW.

CHAPTER 6. CONCLUSION AND FUTURE WORK

In this chapter, the conclusion from this research is drawn and future work is discussed.

6.1. Conclusion

In this thesis, the feasibility of controlling position (or velocity) and torque (or current) at the same time using Fourier repetitive control algorithm was presented. The results show ideal performance in both tracking a waveform and rejecting disturbances that exist in the base frequencies. This property enabled the possible applications in the development of Left Ventricular Assist Device (LVAD) control algorithms and testing of artificial stents.

The need for designing a controller that better controls periodic signals for robots undergoing repetitive tasks arose in the 1980s. Since the idea of repetitive control was independently developed by Arimoto *et al.*, Casalino and Craig in 1984, extensive research was done in this field including the development of several variations of the repetitive control scheme. This research is a continuation of this trend. Fourier repetitive control, as one of the variations, was proven to have better tracking using fewer terms [Glower 1995b]. This research focused on the implementation of the Fourier repetitive control scheme on DC motors and demonstrated the performance of the Fourier repetitive control is as expected.

6.2. Future Work

There are several possibilities that the algorithm can be further improved. Adopting a real-time operation system for the control algorithm implementation, for instance – will improve the tracking result. The author noticed that the SSE increases when other tasks were performed while the algorithm is running. This is likely due to the fact that the Windows XP Sever Pack 3 is

not a real time operating system, and executing other tasks delays the control signal output and causes the SSE to rise. A real-time operating system will eliminate glitches like this.

Furthermore, for the purpose of applying such control to an LVAD in the future, mock simulation *in vitro* needs to be carried out. The tools for the Fourier repetitive control implementation are DC motors. The motor position (velocity) and motor torque (current) are the parameters controlled. However, in order to apply such a control algorithm onto the LVADs, it has to involve pumps and fluid. As is presented in Appendix A and B, the blood flow is analogous to angular velocity of the motor while the blood pressure is analogous to the current of the motor – they are mathematically identical. It is promising that the same control algorithm should work on a fluid dynamic system and eventually applied to LVADs.

REFERENCES

1. [Anderson 1988] Anderson, R.J. Spong, M.W., 1988, "Hybrid impedance control of robotic manipulators," IEEE Journal of Robotics and Automation, Oct 1988 Volume: 4, Issue: 5 pp: 549-556.
2. [Dandel 2005] Dandel M, Weng Y, Siniawski H, Potapov E, Lehmkuhl HB, Hetzer R. Long-term results in patients with idiopathic dilated cardiomyopathy after weaning from left ventricular assist devices. Circulation 2005;112:Suppl:I-37
3. [DeSchutter 1997] Joris DeSchutter, Kerman Bruyninckx, Wen Zhu,, Mark Spong, "Force Control: a bird''s eye view," IEEE CSS/RAS International Workshop on Control Problems in Robotics and Automation: Future Directions, San Diego, CA December 1997, available from www.sciencedirect.com
4. [Dipla 1998] Dipla K, Mattiello JA, Jeevanandam V, Houser SR, Margulies KB. Myocyte recovery after mechanical circulatory support in humans with end-stage heart failure. Circulation. 1998; 97:2316–2322.
5. [Dutta 2002] Ashish Duttaa, Goro Obinata, "Impedance control of a robotic gripper for cooperation with humans," Control Engineering Practice 10 (2002) 379–389
6. [Eppinger 1986] Eppinger, S. and Seering, W., 1986, "On Dynamic Models of Robot Force Control," in: Proceedings of the IEEE International Conference on Robotics and Automation, pp. 29–34.
7. [Girdharan 2004] Giridharan GA, Ewert DL, Pantalos GM, et al: Left ventricular and myocardial perfusion responses to volume unloading and after-load reduction in a computer simulation. ASAIO J 50: 512–518, 2004.

8. [Girdharan 2012] Giridharan GA, Cheng RC, Glower JS, Ewert DL, Sobieski MA, Slaughter MS, Koenig SC. Control strategies for afterload reduction with an artificial vasculature device. ASAIO J. 58(4):353-62, 2012 Jul-Aug.
9. [Glower 1988] J.S. Glower, On the Specification of Adaptation Rates in MRAC Systems, Ph.D. dissertation, The Ohio State University, 1988.
10. [Glower 1989] J.S. Glower, "Adaptation rates for MRAC systems," The International Journal of Control, vol 50, no. 5, pp. 1645-1666, November 1989.
11. [Glower 1997] Glower, Jacob S. "Adaptation rates for repetitive control schemes." International Journal of Adaptive Control and Signal Processing 11.6 (1997): 533-547.
12. [Glower 1995] Glower, J. S., “Repetitive Fourier Control of Robots,” Proceedings of the American Control Conference (1995).
13. [Glower 1995b] Glower, Jacob S. "Comparison of Repetitive Control Schemes." (1995).
14. [Goldenberg 1988] Goldenberg, A.A, 1988, "Implementation of force and impedance control in robot manipulators," IEEE Journal of Robotics and Automation, Apr 1988 pp 1626-1632 vol.3
15. [Goldenstein 1998] Goldstein DJ, Oz MC, Rose EA. Implantable left ventricular assist devices. N Engl J Med 1998; 339:1522-33.
16. [Hara 1988] S. Hara, Y. Yammato, T. Omata, and M. Nakano, "Repetitive Control System: A new type servo system for periodic exogenous signals," IEEE Transactions on Automatic Control, vol. 33, no. 7, pp. 659-667, 1988.
17. [Hetzer 2001] Hetzer R, Müller J, Weng Y, Meyer R, Dandel M. Bridging-to-recovery. Ann Thorac Surg. 2001; 71:S109 –S113.

18. [Hogan 1985] N. Hogan, 1985, “Impedance Control: An Approach to Manipulation (3 parts)”, ASME Journal of Dynamic Systems, Measurement, and Control, 107:1-24.

19. [Hogan 1987] N. Hogan, 1987 (?), "Stable execution of contact tasks using impedance control, "Proceedings. 1987 IEEE International Conference on Robotics and Automation, Mar 1987, Volume: 4, pp 1047- 1054

20. [Hyde 1994] J.M. Hyde and M.R. Cutkosky, 1994, "Controlling Contact Transition", IEEE Control Systems Magazine, 14(1): 25-30.

21. [Hyde 1993] James M. Hyde and Mark R. Cutkosky, "Contact Transition Control: An Experimental Study," 1993 IEEE International Conference on Robotics and Automation, 2-6 May 1993, page(s): 363-368 vol.1, Meeting Date: 05/02/1993 - 05/06/1993, Atlanta, GA, USA

22. [Kazerooni 1986a] Kazerooni, H., Houpt, P. K., and Sheridan, T. B., 1986a, “Robust Compliant Motion for Manipulators, Part 2: Design Method,” IEEE Journal of Robotics and Automation, Vol. RA-2, No. 2, pp. 93–105.

23. [Kazerooni 1986b] Kazerooni, H., Sheridan, T. B., and Houpt, P. K., 1986b, “Robust Compliant Motion for Manipulators, Part 1: The Fundamental Concepts of Compliant Motion,” IEEE Journal of Robotics and Automation, Vol. RA-2, No. 2, pp. 83–92.

24. [Landau 1979] 25. Landau, I. D., Adaptive Control: The Model Reference Approach. New York: Marcel Dekker, 1979.

25. [Lee 1991] S Lee, H S Lee, 1991, "Intelligent Control of Manipulators Interacting with an Uncertain Environment Based on Generalized Impedance," Proc. IEEE Int. Symp. on Intell. Control Date: 13-15 Aug 1991, pp. 61-66.

26. [Levin 1995] Levin HR, Oz MC, Chen JM, Packer M, Rose EA, Burkhoff D. Reversal of chronic ventricular dilation in patients with end-stage cardiomyopathy by prolonged mechanical unloading. Circulation 1995; 91:2717- 20.

27. [Liu 1991] Liu, G.J.; Goldenberg, A.A., 1991, "Robust hybrid impedance control of robot manipulators," IEEE Journal of Robotics and Automation, Apr 1991 Page(s):287 - 292 vol.1

28. [Mills 1996] James K. Mills, 1996, "Simultanous Control of Robot Manipulator Impedance and Generalized Force and Position," Mechanism and Machine Theory, Vol. 31, No. 8, pp. t069-1080,

29. [Popov 1963] 26. Popov, V. M. "Solution of a New Stability Problem for Controlled Systems," Automation andRemote Control, 24, 1-23 (1963)

30. [Raibert 1981] M. Raibert and J.J. Craig, 1981, “Hybrid Position/Force Control of Manipulators”, ASME Journal of Dynamic Systems, Measurement, and Control, 103(2): 126-133.

31. [Rose 2001] Rose EA, Gelijns AC, Moskowitz AJ, et al. Long-term use of a left ventricular assist device for end-stage heart failure. N Engl J Med 2001; 345:1435-43.

32. [Schmid 1999] C. Schmid, D. Hammel, M. Deng, M. Weyand, H. Baba, T.D.T. Tjan, G. Drees, N. Roeder, C. Schmidt, and H.H. Scheld. Ambulatory care of patients with left ventricular assist devices. American Heart Association Circulation, (100):224–228, 1999.

33. [Spong 1996] Spong, M. W., De Schutter, J., Bruyninckx, H., & Wen, J. (1996). Control of Robots and Manipulators. The control handbook, 1339-1351.

34. [Tarn 1996] T.J. Tarn, Y. Wu, N. Xi, and A. Isidori, 1996, "Force Regulation and Contact Transition Control", IEEE Control Systems Magazine, 16(1): 32-40.

35. [Tsao 2000] Tsao, Tsu-Cliin, Yao-Xin Qian, and Mahadevamurty Nemani. "Repetitive control for asymptotic tracking of periodic signals with an unknown period." Urbana 51 (2000): 61801.

36. [Volpe 1993] Richard Volpe and Pradeep Khosla, "A Theoretical and Experimental Investigation of Explicit Force Control Strategies for Manipulators," IEEE Transactions on Automatic Control 38[11] November 1993, pp. 32-33.

37. [Vukobratovic 1996a] M. Vukobratovic and R. Stojic, 1996, "On Position/Force Control of Robot Interacting with Dynamic Environment in Cartesian Space", ASME Journal of Dynamic Systems, Measurement and Control, 118(1): 187-92.

38. [Vukobratovic 1996b] Miomir Vukobratovic and Dusko Kati, "“Stabilizing Position/Force Control of Robots Interacting with Environment by Learning Connectionist Structures,"” Automatica, Vol. 32, No. 12, pp. 173.’’;--1739. 1996

39. [Vukobratovic 1998] Miomir Vukobratovic, Radoslav Stojic, Yuri Ekalo, "Contributions to the Position/Force Control of Manipulation Robots Interacting with Dynamic Environment – A Generalization," Automatica, Vol 34, No 10, pp 1219 - 1226, 1998.

40. [Whitney 1985] Whitney, D. E., 1985, “Historical Perspective and State of the Art in Robot Force Control,” in: Proceedings of 1985 International Conference on Robotics and Automation.

41. [Zafeiridis 1998] Zafeiridis A, Jeevanandam V, Houser SR, et al. Regression of cellular hypertrophy after left ventricular assist device support. Circulation. 1998; 98:656–662.

APPENDIX A. LEFT VENTRICULAR ASSIST DEVICES (LVAD)

The mechanical model of the AVD is shown in Figure A1. A rotational motor is used to control an arm with a length r. The arm drives a piston to move up and down, therefore changes the volume of a blood chamber that is anastomosed to the aortic arch. Blood flow directions are controlled by valves. From the analysis of the mechanical structure of the AVD above, the relationship between rotational angle θ (small amount) and vertical displacement of the piston x is as follows:

$$x = r\theta \tag{34}$$

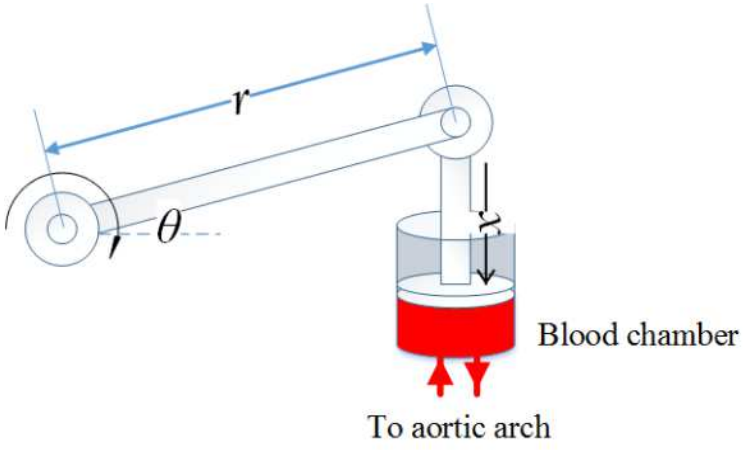

Figure A1. The mechanical model of an AVD.

Assume the shape of the blood chamber is a cylinder and the cross section area of the cylinder is A, volume V of blood inside the blood chamber is

$$V = Ax = Ar\theta \tag{35}$$

Flow Q is defined as the changing rate of volume:

$$Q = \frac{dV}{dt} = Ar\frac{d\theta}{dt} \tag{36}$$

The magnitude of angular velocity is defined as the rate of change of angular displacement:

$$\omega = \frac{d\theta}{dt} \tag{37}$$

Combine (33) and (34) yields

$$Q = Ar\,\omega \tag{38}$$

Since A and r are constants, flow Q is proportional to the magnitude of angular velocity ω.

$$Q \propto \omega \tag{39}$$

APPENDIX B. ARMATURE-CONTROLLED DC MOTOR

An armature-controlled DC motor has a fixed magnet field, either established by a field coil and current or a permanent magnet as presented in Figure B1.

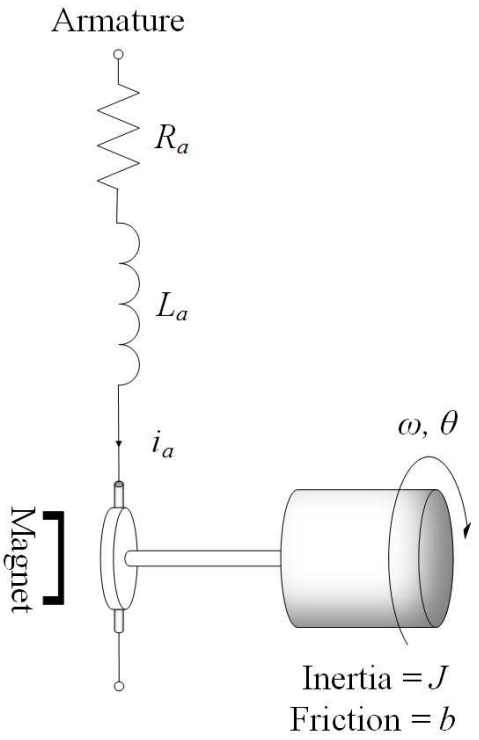

Figure B1. Wiring diagram of an armature-controlled DC motor.

The motor torque $\tau_m(t)$ of an armature-controlled DC motor is illustrated below:

$$\tau_m(t) = K_m i_a(t) \tag{40}$$

where K_m is a function of the permeability of the magnetic material, in this case, K_m is a constant value. Thus, the motor torque $\tau_m(t)$ is proportional to the armature current $i_a(t)$.

$$\tau_m(t) \propto i_a(t) \tag{41}$$

By definition, torque is the cross product of displacement vector and force vector, or the magnitude of torque can be represented as:

$$\tau_m(t) = r\left\|\mathrm{F}(t)\right\| \tag{42}$$

When in this case, the arm vector and the force vector are perpendicular to each other.

Assume the blood inside the blood chamber is incompressible, the blood pressure inside the blood chamber can then be calculated by using the definition of pressure:

$$p(t) = \frac{\left\|\mathrm{F}(t)\right\|}{A} \tag{43}$$

From the equations (38), (40), and (41), one may find that blood pressure $p(t)$ is proportional to the armature current $i_a(t)$:

$$p(t) \propto i_a(t) \tag{44}$$

The following analysis is in s domain for simplicity.

For motor 1, the controller supplies an armature current $I_a(s)$ that is proportional to the controller voltage. The controller voltage follows sinusoidal waveform, thus providing a disturbance $T_d(s)$ as follows:

$$T_d(s) \propto \frac{s}{s^2 + \omega^2} \tag{45}$$

For motor 2, the armature voltage instead of the current is directly controlled by the voltage supplied to the motor controller 2.

$$V_a(s) = (R_a + L_a s) I_a(s) + V_b(s) \tag{46}$$

where R_a is the resistance and the L_a is the inductance of the armature. $V_b(s)$ is the induced electromotive-force voltage that counteract the voltage supplied to the armature (Lenz's law). Its value is proportional to the motor speed. Therefore,

$$V_b(s) = K_b \omega(s) \tag{47}$$