APPROACH CONTROL OF A GAS EXCHANGE SOLENOID ACTUATOR FOR IC ENGINES

by

Chun Ming Tsai

B.A.Sc, Simon Fraser University, 2004

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF APPLIED SCIENCE
in the School
of
Engineering Science

SIMON FRASER UNIVERSITY
Fall 2007

APPROVAL

Name: Chun Ming Tsai

Degree: Master of Applied Science

Title of thesis: Approach Control of a Gas Exchange Solenoid Actuator for IC Engines

Examining Committee:

Dr. Parvaneh Saeedi
Assistant Professor, School of Engineering Science
Simon Fraser University
Chair

Dr. Mehrdad Saif
Professor, School of Engineering Science
Simon Fraser University
Senior Supervisor

Dr. Charles Robert Koch
Associate Professor,
Department of Mechanical Engineering
University of Alberta
Supervisor

Dr. Mehrdad Moallem
Associate Professor, School of Engineering Science
Simon Fraser University Surrey
Internal Examiner

Date Approved: December 11, 2007

Abstract

Application of solenoid valve actuators in internal combustion engines can facilitate operations such as variable valve timing for improved efficiency and emission. Unfortunately, smooth solenoid valve landing is hard to achieve due to limited control authority, limited bandwidth, and time varying disturbances. The resultant valve impact causes unacceptable noise and component wear on the engine. To solve this "soft seating" problem, the controller is further divided into approach and landing sub-controllers. The landing controller causes the valve to follow a smooth trajectory for a low-impact landing in the last portion of the valve flight. Before armature landing starts, the approach controller complements the landing control by setting a consistent initial condition for the landing trajectory. This thesis focuses on developing a cycle-adaptive approach controller that utilizes information from the repetitive operations of the engine valve. Additionally, a novel way of using induced voltages to identify disturbance pressure magnitudes is introduced.

keywords: cam-less engine, valve soft-seating control, nelder mead algorithm, iterative learning algorithm, electromagnetic devices, solenoids

subject terms: automatic control, control theory, valves, actuators

"People are like stained-glass windows.
They sparkle and shine when the sun is out,
but when the darkness sets in,
their true beauty is revealed only if there is a light from within.!"

— *Elisabeth Kubler-Ross*

Acknowledgments

I wish to thank my supervisors Dr. Koch and Dr. Saif for guiding me along on this project. Without your insight and guidance, I would not have been able to finish the project successfully. I also wish to thank people in my lab at SFU, Esmail, Qing, Weitien, Quongqing, Maryam, and Arina, for their encouragement. I also wish to thank everyone in Dr. Koch's Engine Control Lab in the Mechanical Engineering Department at the University of Alberta: Akmed, Ryan, David, Robert, and Adrian. Thank you for helping me conduct my experiments and helping me with adjusting to life in Edmonton. I have nothing but the fondest memories of Edmonton. Special thanks to Auto21 project for providing the funding for this research. Finally, I wish to thank my parents for all of their love and support.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Flexible control of engine valve operation is a key contributor to performance of a modern engine. The following three examples illustrates this point: 1. Replacing the camshaft, which fixes valve timing with respect to the crankshaft, allows engine operation to be optimized according to load and performance requirements [5]. 2. If valves can be selectively controlled, the unneeded cylinders can be selectively shutdown for fuel economy [6]. 3. One effective way to run an engine is in homogeneous charge compression ignition (**HCCI**) mode, which provides an efficient and cleaner combustion, requires full control of the engine valves to help maintain conditions suitable for auto-ignition [7].

To accomplish such flexible valve control (also known as Variable Valve Timing, **VVT**), many camshaft phase or lift altering mechanisms have been employed by various automotive companies. A brief search on variable valve timing results range from Alfa Romeo's "Twin Spark" [8] to Honda's "VTEC" [9] to Yamaha's "Variable Cam Timing" valve-train [10]. However, the most flexible valve-trains are the ones that eliminate the camshaft altogether. While there are a few actuators that qualify for camless engine operation, in this thesis our focus is on the electro-magnetic solenoid actuator, which is energy-efficient and powerful, but requires a sophisticated control strategy to avoid large valve landing impacts and to obtain "soft landing".

To solve the soft seating problem more effectively, the research undertaken in a Simon Fraser University - University of Alberta joint project is separated into an approach portion and a landing portion. The approach control takes place in the first part of the valve motion and attempts to deliver consistent final conditions which are the initial conditions for the second part of the control - the landing control. This thesis develops an approach control

scheme that uses the information from previous cycles to tune the feedforward control input such that the conditions at the beginning of the landing trajectory are kept constant.

In this opening chapter, the motivation behind development of the electromagnetic valve actuator is discussed. The first subsection talks about the benefits of variable valve actuator and homogeneous charge ignition and compression (HCCI). Then the structure and properties of electromagnetic valve (**EMV**) actuator itself are discussed along with the control problem associated with it. The chapter ends with a problem statement, and describes the thesis organization and my contributions in the thesis.

1.1 Benefit of Variable Valve Timing

By changing the timing of the engine intake and exhaust valves at different operating conditions, the engine can be optimized to achieve various objectives. Below are some, but not all the variable valve timing operations:

- Increase maximum power at high speed: If closing of the intake valve is delayed under high engine loads, the significant air-fuel momentum in the intake means that even after the piston starts moving up, fresh air continues entering the combustion chamber instead of being pushed out [11]. The result is better compression and more power.

- Improve engine efficiency at low speed: When the engine is under partial load, the intake valve would be closed early to reduce the suction force needed to bring the air in (pumping loss) [12]. In addition, because less air needs to be compressed, the engine generates less heat, and thus, the thermal efficiency is improved. These efficiency improvements translate into less fuel consumption at low engine loads.

- Reduce pollution using Exhaust Gas Recirculation (**EGR**): Traditionally, in order to reduce nitrogen-oxide (**NOx**) emissions generated from engines operating at high temperatures, EGR valves are placed in the engine to reduce the combustion temperature by recycling the exhaust back into the intake valve [13]. Variable valve actuation can remove the need for additional valves by regulating the overlaps between intake and exhaust valve openings or by closing the exhaust valve early [14].

- Deactivate unneeded cylinders: Switching a few cylinders off during low engine loads in a conventional camshaft driven engine requires additional machinery. If the valves can

be controlled individually, then cylinder deactivation would be as simple as sending different electrical signals. This results in a higher effective load on the remaining cylinders with the potential for increased engine efficiency [15].

For more strategies of VVT, see the review and analysis article [16]. Note the process of EGR can also be used to raise gas mixture temperature and cause auto ignition, which has many advantages over conventional spark or compression ignition methods when controlled properly.

1.1.1 Homogenous Charge Compression Ignition

Homogeneous charge compression ignition (**HCCI**) is a form of controlled auto ignition. In this mode of combustion, a homogenous fuel-air mixture enters the cylinder, and the combustion takes place through compression only; no spark or fuel injection is required. The mixture combusts more uniformly and at lower temperatures, which means less nitrogen oxide pollutant and no need for expensive exhaust treatment. Also, the mixture is very diluted and required fuel can be as much as 20 percent less compared to the baseline spark ignition engine [7].

Manipulating intake and exhaust valve timing not only raises the temperature of the gas but can also change the composition and motion of the gas mixture. For example, delaying the intake valve opening can increase the intake flow rate. These additional tuning capabilities are paramount for the HCCI operation to succeed [7].

1.2 The Electromagnetic Valve (EMV) Actuator

The requirements for the engine gas exchange valve for automotive application are stringent: short travel time (4ms), forces capable of overcoming engine blow-down pressure, energy efficiency, and heat resistance. The various actuators that are considered for camshaft replacement currently includes hydraulic [17], rotary motor [18], piezoelectric type actuators [19], and electromagnetic solenoid actuators [20].

Out of all these choices, the solenoid valve actuator utilized in this thesis stands out because it can exert a high non-contacting force over a short distance. The actuator is simple to construct and withstands the engine generated pressure and heat. The electro-mechanical actuator discussed here is manufactured by Telefunken Microelectronic GmbH

and it will be referred to as **TEMIC** in the rest of this thesis. The actuator is composed of two springs, two solenoids, and a shaft connecting a metal armature to a valve (see Figure 1.1). Both springs are always set in compression such that, in equilibrium, the armature sits in the middle. No hydraulic lash adjuster is present in the experimental test-bench. Two coils are used as electromagnets to attract the armature to one of the two ends across the total travel distance of 8mm.

To initiate the actuator for valve operation, only the closer coil is energized. As a result, the armature is attracted from its detent middle position to the top of the actuator (next to the closer coil). Since the armature is connected to the valve, this initialization closes the valve. To open the valve from this position, the current in the closer coil is drained to zero, causing the closer coil to stop holding the armature and allowing the armature to be pulled (by the springs) toward the opener magnet at the bottom. The opener magnet is then energized to attract and hold the armature to keep the valve open. During a valve closing event, the opposite occurs, and the armature travels from its initial position next to the opener coil to the closer coil at the top of the actuator.

1.2.1 Challenges

The challenge of the EMV controller design includes closed-loop bandwidth, non-linearity, robustness, and energy efficiency requirements. In order to meet the engine speed requirement of at least 4000 to 5000 RPM, springs are tensioned to 250N/mm so that the transition time is 4ms. For such short durations, bandwidth suffers from the significant delays needed to convert input voltage to magnetic force. Non-linearity exists because magnetic force is roughly inversely proportional to the square of the gap between the coil and armature. When the armature is far away, the pulling force is negligible; on the other hand, when the armature is close to the coil, the pulling can be undesirably large. Finally, to meet the robustness requirement, the valve controller must guard against variations of both engine combustion pressure and actuator parameters.

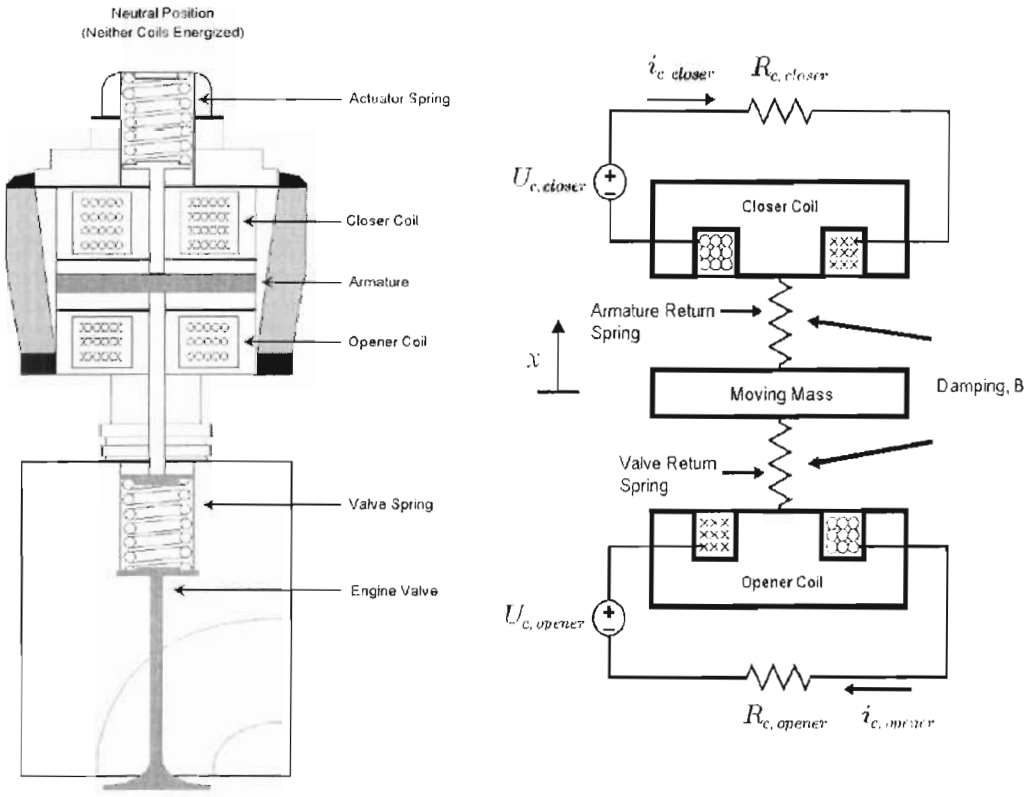

Figure 1.1: Cross sectional view of the electromagnetic valve actuator (left) and the corresponding schematic (right). Adopted from [1] with permission of the author.

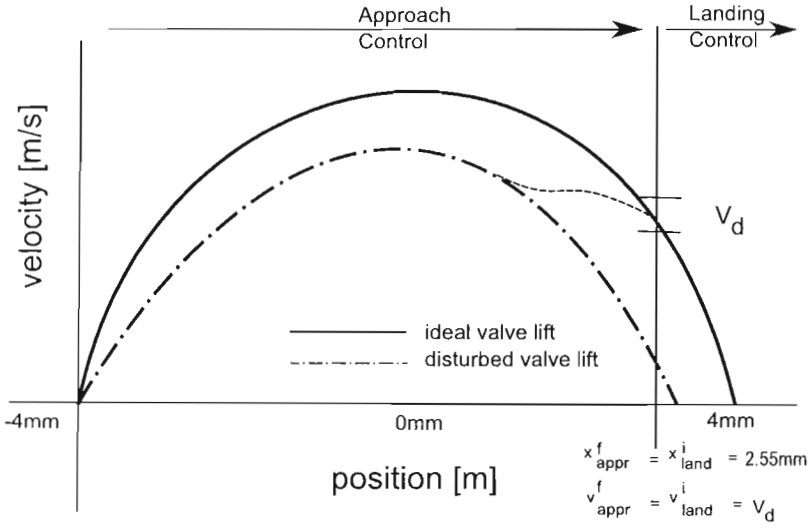

Figure 1.2: Illustration of control regions for the approach and landing control: setting a consistent initial conditions for the EMV landing control.

1.3 The Approach Controller for the EMV Actuator

The ultimate goal of the controller is to ensure the armature gently lands at one end of the actuator. An impact speed less the 0.1m/s is considered an effective guard against excessive mechanical wear and noise ([21]-[23]). To achieve this goal, our research group further divided the controller for TEMIC into two portions. The take-off/approach controller operates at the beginning and the middle of the armature flight to compensate for energy shortfalls due to disturbance and provides a good initial condition for the end controller. The end controller subsequently takes over for the rest of the valve travel and facilitates a soft landing. In Figure 1.2, the position 2.55mm is the transition point between the two controllers.

$$x^i_{land} = x^f_{appr} = 2.55mm \tag{1.1}$$

The superscript i and f stand for initial and final, while subscripts $appr$ and $land$ stand for approach and landing trajectory. In [24], a landing controller using flatness-based control assuming desired initial conditions v_d and i_d was successfully designed. The approach controller developed in this thesis aims complement [24] and compensate for the disturbances

taking place between -4mm to 2.55mm. The goal is to regulate the velocity and current at start of the landing control x^i_{land} are within a specific window.

$$v^f_{appr} = v^i_{land} \simeq v_d \quad \text{and} \quad i^f_{appr} = i^i_{land} \simeq i_d \tag{1.2}$$

The valve position velocity plot shown in Figure 1.2 illustrates this approach by showing the effects of approach control, symbolized by the dotted line connecting the disturbed trajectory back to the nominal trajectory.

Note that due to the symmetry of the solenoid actuator, the control for valve opening and closing are almost identical with the exception that exhaust valve opening encounters larger pressure disturbances from the engine. Consequently, most of the simulation and experiment in this thesis will be performed on valve closing without loss of generality.

1.3.1 Cycle Adaptive Control

Normal engine operation requires valves to open or close at between 10 to 100Hz; thus the electromagnetic actuator is a highly repetitive system. Therefore, the information from the previous valve events can be utilized as the basis for feedforward control in the current valve event. This thesis focuses on the cycle adaptive feedforward approach control of the EMV actuator. The use of feedforward control compensates for the small magnetic force at large airgap, while cyclical adaptation takes advantage of the repetitive nature of the engine valve operation. Note that the controller assumes the disturbance is much slower than the valve travel time (i.e., the residuals from previous iterations can be used for current iteration). In reality, this assumption is not always true, especially in the case of engine misfires. For the feed-forward controllers that accommodates drastic pressure variations between firings, see [25] on disturbance-estimator based EMV approach control.

1.4 Problem Statement

This thesis develops approach controllers which regulate the velocity and current at position x^f_{appr} (defined in Figure 1.2) to set values suitable to begin landing control. Under the assumption that the disturbance dynamic is much slower than the actuator spring dynamic, the approach controller must be robust enough to handle engine disturbances and the parameters variations in the actuator. Different control schemes will be investigated for comparison purpose. All controllers investigated will be validated in simulation; then depending on the simulation results, some will be ported to the TEMIC experimental actuator test bench at a supply voltage of 42 volts.

1.5 Thesis Contributions

The main contribution of the thesis comes from developing the cycle-adaptive controller for EMV actuator approach control, but some interesting disturbance estimation results are also presented briefly.

- A novel method to estimate in-cylinder pressure disturbances is proposed using the induced voltage from the release coil of the EMV actuator. The known induced voltage and the pressure value is used to develop an inverse model. Verifying the inverse model output using real induced voltage data shows a good match with the actual measured pressure value.

- A trajectory tracking iterative learning controller is implemented in simulation for approach control purpose. Unlike the existing EMV learning controller [23] and [2], the controller presented here requires no linearization. The simulation result shows convergence can be achieved if the supply voltage is sufficiently high.

- A terminal iterative learning controller, which requires no trajectory tracking, is developed and tested in simulation and experiment. A nullspace gradient projection method is shown to help steer the controller coefficient evolution.

- A direct search controller, which is based on solving a nonlinear programming problem using the Nelder Mead algorithm, is presented. The controller requires no trajectory tracking, allows greater freedom in defining the cost function, and can be constrained

against input and state saturation. The experimental result shows effective regulation of velocity and current against unknown actuator parameters variations, step disturbances, and ramp disturbances, at the end of the approach control trajectory.

1.6 Thesis Organization

Benefits of the VVT and HCCI operation on an engine, and how an EMV actuator is used to facilitate VVT, were discussed in this chapter. An explanation was given as to why the soft landing problem of the EMV must be resolved before it can be relied upon for VVT operation. Further, readers were introduced to the idea of dividing the control algorithm into approach and landing sub-controllers. Finally, the chapter presented the problem statement, described the thesis organization, and outlined the contribution of this thesis.

A literature review is provided in the second chapter. A brief discussion of the existing control schemes is presented: linear quadratic regulator, sliding mode, hybrid system, and H_∞ controllers. Emphasis is placed on feed-forward approach controllers such as iterative learning and extreme seeking algorithms. The third chapter presents a description of the EMV actuator model. In particular, emphasis is placed on how eddy currents and magnetic flux saturation is accommodated in the model. In addition, a pressure disturbance model along with an induced voltage pressure identification scheme is discussed.

In Chapter Four, three different solutions to the approach control problem are given: the nonlinear iterative learning controller, the terminal iterative learning controller, and the Nelder Mead direct-search controller. B-spline interpolation is also mentioned because it is applied in some of the controllers. Chapter Five describes the physical experimental setup, which includes the TEMIC actuator itself, the dSpace controller board, current protection circuit, the power supply and power electronics that run the actuator, and the sensors that provide position, voltage, and current feedback.

Experimental results are presented in Chapter Six. The data from the actual test bench are presented, which includes experimental results for the terminal iterative learning controller and the Nelder-Mead direct search controller. The results from the Nelder Mead algorithm can be further divided into three groups: setpoint changes, temperature-related disturbances, and simulated pressure disturbances. The last chapter is a summary of what was accomplished and the important insights gained during the research work. The thesis concludes with possible directions for future investigations.

Chapter 2

Literature Review

In this chapter, developments relevant to solving the EMV valve soft-seating problems are reviewed. The discussion starts with a review of the existing publications on the electromagnetic valve actuator model. The various existing control schemes are then presented. Because this thesis focuses on cycle adaptive control, more detailed explanation and observations are provided for cycle adaptive strategies such as iterative and extremum seeking control.

2.1 EMV Actuator Model Development

The EMV actuator was proposed as early as the 70s [26] and 80s [27]. As computing capabilities improved, model development followed suit. In 1990, [28] investigated a solenoid actuator with two springs . However, the actuator in [28] requires permanent magnets, which will demagnetize at high temperature [29], and thus is unsuitable for engine valve operation.

The most difficult aspects of EMV actuator modeling are the various nonlinear electromagnetic effects such as eddy currents, material saturation, and fringing; hence, much of the work focuses on this area. To account for these phenomena, the finite element method (**FEM**) is frequently applied [30, 31]. However, due to the computational complexity of the FEM, some publications attempt to model and design better electromagnetic actuators with methods such as electric-equivalent-network [32], reluctance network [33], and magnetic coupling network [34] .

For EMV actuators in camless engine applications, the modeling effort by [35], [20], and [36] are significant. The authors in [35] presented an analytic EMV model based on a

mass-spring oscillator system and a magnetic system that is divided into linear and saturated regions. Their subsequent work [37] points out the specific difficulties involved with the EMV armature soft landing: inherent instability, low control authority, and limited bandwidth. [37] also demonstrated that open-loop control is not feasible, and showcased a quick EMV armature-release scheme using voltage reverse polarity. Additionally, [36] contains a model of the EMV actuator with the switching power converter taking command from an automotive engine control unit (**ECU**). Also, [20] set out to determine the specifications required for a successful EMV deployment in the engine, which includes the requirement for measurement sensors as well as the actuator itself. [20] stated that an actuator must be built to withstand $-40°$C to 160°C with a travel time of $3 \sim 4$ ms and compatible with a 42 volt power supply. By analyzing the linearized plant at various operating points and finding the location of unstable poles, [20] concluded that a minimum sensor sampling rate of 7.5 khz and sensor accuracy of 10 μm are necessary.

More recently, [31] presented a lump parameter model that incorporates lookup tables with FEM results for improved model accuracy and computational efficiency. Also in [38], various measurement feedback schemes were investigated it was determined that the combination of position and flux feedback were most suitable. In [39], the authors used a hybrid system method to model the EMV actuator and convert the soft-seating problem into a valve profile regulation problem.

2.2 Existing Control and Observation Schemes

Since the millennium, at least two dozen reports related to the valve soft-seating controller have been published, covering various control strategies such as sensorless [22], linear quadratic regulator (**LQR**) [40], sliding mode [41], control Lyapunov function (**CLF**) [42], and flatness-based control [24]. In addition, cyclic adaptive controllers such as repetitive learning [43], iterative learning [2], and extremum seeking [3] have been employed. The following subsections provide brief explanations of these works.

2.2.1 Sensorless control

The term "sensorless control", was coined by the authors in [22] in 2000 to indicate an EMV controller that uses only current measurement instead of the expensive position measurement sensor. In [22], they proposed such a controller based on the following relationship between

the electrical and mechanical state of the actuator:

$$\frac{di}{dt} = \frac{i}{1 + k_0 d}\left(\frac{U_{coil} d}{\mu_0 N^2 A} - \frac{iv}{d}\right) \tag{2.1}$$

The terms U_{coil} , i, v, d refer to coil voltage, coil current, armature velocity, and armature airgap, while the rest of the terms are fixed coefficients. If the voltage is reduced to zero and the airgap is small, $1 \gg k_0 d$, the ratio of current and current derivative corresponds to the ratio between velocity and position:

$$\frac{di/dt}{i} = \frac{-v}{d} \tag{2.2}$$

If the control keeps the ratio $\frac{di}{dt}/i$ constant, then the ratio of velocity and airgap is the same as the airgap reduces to zero. Subsequently, the armature velocity is reduced during landing. Butzmann et al. reported an impressive result of average 0.1m/s impact velocity over 5000 iterations under laboratory conditions and an average of 0.2~0.3m/s in actual engine operations. In 2003, the robustness of the sensorless control scheme is improved by Gunselmann and Melbert with the addition of take-off and approach control schemes [44]. For take-off, they pointed out the induced voltage in the release coil can be used to detect disturbances. In addition, during the middle of the armature swing (also known as the approach stage), the current is allowed to increase freely until a specific current derivative is detected, and thus compensates for the disturbance.

Sensorless control strategies are cost efficient and computationally efficient. The main issue with the two sensorless control approaches is that they are heuristic rather than analytic. While feedback methods are identified, the details are lacking. For example, what is the current derivative value used to switch the approach control into 0v mode? What is the exact mechanism compensating for the take-off disturbance detected from induced voltage? Are there mathematical convergence analysis for landing velocity reduction in finite time?

2.2.2 Frequency-based linear control

Another way to facilitate soft-landing control is to linearize the EMV actuator dynamics and utilize the existing linear control strategies. The authors in [43] applied frequency domain system identification to find a voltage to position transfer function and then applied a proportional derivative (**PD**) controller for stabilization. Trajectory tracking is achieved through cyclic adaption of the feedforward input using repetitive learning control. They

reported satisfactory impact velocities but slow transition time of 10ms:

$$\text{PD controller} \qquad Gc(s) = \frac{kd \cdot s + kp}{s/\omega_4 + 1} \tag{2.3}$$

$$\text{repetitive controller} \quad u_{ff,j+1}(k) = Q(u_{ff,j}(k) + k_p \cdot Me_j(k)) \tag{2.4}$$

where $Gc(s)$ is the controller transfer function and $u_{ff,j+1}(k)$ is the feedforward input at sampling moment k of the iteration $j+1$. $e_j(k)$ is the tracking error and the rest of the coefficients are controller gains.

Subsequent papers by the same authors in 2002 and 2003 replaced PD control with linear quadratic regulator (**LQR**) control [40] and included the enhanced plant model to reflect saturation in short air gaps [23]. The LQR uses the linear model described as:

$$\dot{x}(t) = Ax(t) + Bu(t). \quad \text{and} \quad y(t) = Cx(t) \tag{2.5}$$

The cost function weights the energy expenditure and the need to reduce the states $x(t)$, armature position, and velocity, to zero:

$$J = \int_0^\infty \left(x^t(t)Qx(t) + Ru^2(t)\right) \cdot dt \tag{2.6}$$

The solution for minimized J can be found in the following (P matrix is the solution to the algebraic Riccati's equation[45]):

$$\text{LQR control input} \qquad u(t) = -R^{-1}B^T Px(t) \tag{2.7}$$

$$\text{algebraic Riccati's equation} \quad A^T P + PA - PBR^{-1}B^T P + Q = 0 \tag{2.8}$$

Their controller in [23] improved the transition time to 5ms, but the armature impacts are listed with an average of 0.278m/s, which is still above 0.1/ms.

2.2.3 Operating point based linear control

Another way to implement the linear controller is through equilibrium-point-based linearization. In 2002, [46] separated the system into two subsystems based on *far* and *close* equilibrium points. After linearizing the system at the equilibrium points, both linear models have the same form:

$$\frac{d}{dt}\begin{bmatrix} \triangle i \\ \triangle z \\ \triangle v \end{bmatrix} = \begin{pmatrix} a_{11} & 0 & a_{13} \\ 0 & 0 & a_{23} \\ a_{11} & a_{32} & a_{33} \end{pmatrix} \begin{pmatrix} \triangle i \\ \triangle z \\ \triangle v \end{pmatrix} + \begin{pmatrix} b_1 \\ 0 \\ 0 \end{pmatrix} \triangle V_c \tag{2.9}$$

The *far* model, which runs between 1mm to 8mm airgap, is used to construct a flux initialization controller, an LQR controller with feedback gains that penalize deviation from nominal catching current and overcome the low bandwidth problem. Another LQR controller uses the *near* model to bring the system to the equilibrium point which requires airgap and velocity to be zero. The results for the experiment in [46] is a mean value of 0.16m/s at an average transition time of 3.42ms using a 200 volt power supply.

2.2.4 Sliding mode sensorless control

The author in [47] and [41] presented a controller which uses sliding mode methodology to estimate the unmeasured states and to following a landing trajectory. He utilized a sliding mode observer that uses only current measurement to estimate both position and velocity:

$$\begin{aligned}
\dot{\hat{z}_1} &= \hat{z_2} - M_1\text{sign}(s_1) - M_2\text{sign}(s_2) \\
\dot{\hat{z}_2} &= \frac{1}{M_t}(-2K_s(\hat{z_1} - y) - Bz_2 - C\text{sign}(\hat{z_2}) - \hat{F}_h - \hat{F}_{md} + \hat{F}_{mu}) \\
&\quad -M_3\sin(s_1) - M_4\sin(s_2) \\
\dot{\hat{\lambda}}_u &= -f_{3u}(\hat{\lambda}_u, \hat{x}_u, V_{Lu})R + V_{su} - M_5\text{sign}(s_1) \\
\dot{\hat{\lambda}}_d &= -f_{3d}(\hat{\lambda}_d, \hat{x}_d, V_{Ld})R + V_{sd} - M_6\text{sign}(s_2)
\end{aligned} \tag{2.10}$$

The values $\hat{z_1}$, $\hat{z_2}$, $\hat{\lambda}$ are position, velocity, and flux estimates. s_1 and s_2 are errors between measurements and expected current values of the two coils. The sliding mode observer gains are the M_1 to M_6, while the rest of the symbols are related to the EMV system model. To show convergence, [41] defined a positive definite Lyapunov function:

$$V(\tilde{z_1}, \tilde{z_2}, \tilde{\lambda}_u, \tilde{\lambda}_d) = \frac{1}{2}\tilde{z}_1^T\tilde{z}_1 + \frac{1}{2}\tilde{z}_2^T z_2 + \frac{1}{2}\tilde{\lambda}_u^T\tilde{\lambda}_u\frac{1}{2}\tilde{\lambda}_d^T\tilde{\lambda}_d \tag{2.11}$$

The gains M_1 to M_6 can be chosen such that

$$\dot{V}(\tilde{z}_1, \tilde{z}_2, \tilde{\lambda}_u, \tilde{\lambda}_d) < 0 \tag{2.12}$$

A positive definite Lyapunov function with a negative definite derivative means the error will converge to zero in finite time; therefore the sensorless observer in [41] can determine the position and velocity variables with only current measurements.

To track a desired trajectory, the sliding mode controller reformulates the system dynamic equation so that the tracking errors (and their derivatives) are the state variables:

$$e_1 = r_u - \hat{z}_1, \dot{e}_1 = e_2, \dot{e}_3 = \ddot{e}_2 \tag{2.13}$$

A sliding mode equilibrium manifold, σ_1, for the upper landing trajectory is defined as

$$\sigma_1(e_1) = c_1 e_1 + c_2 e_2 + c_3 e_3 \tag{2.14}$$

where the variables c_1 to c_3 are weighting coefficients. To prove convergence of the tracking error, the Lyapunov function is again applied:

$$V_u(\sigma_1) = \frac{1}{2}\sigma_1^T \sigma_1 \tag{2.15}$$

A sliding mode voltage controller, V_{su}, is presented in the following form that enables a negative definite Lyapunov function time derivative.

$$\begin{aligned} V_{su} &= -U_u \text{sign}(\sigma_1) \\ U_u &\quad \text{is chosen such that} \quad \dot{V}_u(\sigma_1) < 0 \end{aligned} \tag{2.16}$$

The experimental result in [41] shows a mixed range of impact velocities from 0.35m/s to 0.05m/s at a slow valve opening/closing time of 20ms. The inconsistency seems to be due to a lack of proper approach control (only tuned open-loop was used) against disturbances. For example, [41] stated spring pre-compression changes after several high frequency cycles, which can contribute to the inconsistency of the valve performance.

2.2.5 Control Lyapunov function based control

[48] and [42] present a feedback controller for the EMV actuator utilizing the control Lyapunov function. In addition to soft-landing, the controller can perform variable valve lift (VVL), a technique beneficial to the HCCI operation [7]. The EMV model of [48] is formulated in the following manner:

$$\frac{dx}{dt} = f(x) + g(x)u$$

where x is the state vector (position, velocity, and flux), t is time, and u is voltage input. The control utilizes Sontag's feedback [49]:

$$u = -\frac{L_f V + \sqrt{L_f V^2 + ((L_g V)(L_g V)^T)^2}}{L_g V} \tag{2.17}$$

The variables $L_g V$ and $L_f V$ are the Lie derivative of the control Lyapunov function $V(x)$:

$$L_f V(x) = \frac{dV}{dx} f(x) \quad \text{and} \quad L_g V(x) = \frac{dV}{dx} g(x) \tag{2.18}$$

This defines the control Lyapunov function (**CLF**) candidate as,

$$V(x) = x^T P x \tag{2.19}$$

where the matrix P is the solution to algebraic Riccati equation,

$$PA + A^T P - Q - PBB^T P = 0 \tag{2.20}$$

with A and B as the linearized EMV system matrices with the operating point at the origin. At this point, [42] draws a parallel with LQR in which the matrix Q is defined as part of the cost function similar to Eq. 2.6. Compared to the LQR, [42] showed that CLF performs better during transient conditions. Overall, the CLF controller results in 0.1~0.2m/s landing impact and demonstrates armature levitation as close as 1mm away from the catching coils. However, the power supply used in the experiments is a 200v power supply instead of a 42 volt power supply.

2.2.6 Feedback linearization based control

[50] lists a proportional integral (**PI**) controller over an exact-feedback linearized EMV actuator model. The linearized system use magnetic force as input:

$$M\ddot{x} + B\dot{x} + Kx = f_m(x, I) \tag{2.21}$$

where M, B, K, I are mass, damping constant, spring constant, and coil current. The magnetic force f_m is determined by current and position x. The linearized input $f_m = f_c$ are chosen as

$$f_c = M\ddot{x}_d + Bx_d + K\dot{x}_d - (\bar{B} - B)\dot{e} - (\bar{K} - K)\int \dot{e} \tag{2.22}$$

where $e = x - x_d$ and $\bar{B}$, $\bar{K}$ are tuning variables. Because of the above control input, the close-loop tracking error dynamics is reduced to

$$\bar{M}\ddot{e} - \bar{B}\dot{e} + Ke = 0 \tag{2.23}$$

The rest is simply to choose $\bar{M}$ and $\bar{B}$ so that the differential equation above drives the error to zero. In [50], a disturbance estimator and compensator is also presented. However, no evidence of the estimator actually being implemented is shown, and no impact velocity statistics from the feedback-linearized PD controller are available.

2.2.7 Flatness based control

The work in [1] and [24] presentes a flatness-based landing controller for the EMV actuator. To utilize the flatness property of the EMV system, the authors first showed that all the states can be expressed as a function of the position measurement and its first and second order derivatives. The input, V, can thus be written as

$$V = \frac{f(y)}{i_c f'(y)} \left[m y^{(3)} - \dot{A}(y, \dot{y}) - 2\dot{y} F(y, i_c) \times \left[\frac{1}{c_2 - y} - \frac{f'(y)}{f(y)} \right] \right] + R_c i_c \tag{2.24}$$

where $A(y, \dot{y})$ and $F(y, i_c)$ represents mechanical and magnetic forces, i_c and y are the measured position and current, and R_c is the coil resistance. The triple derivative term $y^{(3)}$ in Eq. 2.24 can be replaced to achieve closed loop linear error dynamics

$$\tilde{y}^{(3)} = y_d^{(3)} - k_2 \ddot{\tilde{y}} - k_1 \dot{\tilde{y}} - k_0 \tilde{y} \quad \text{where} \quad \tilde{y} = y - y_d \tag{2.25}$$

Substituting Eq. 2.25 into Eq. 2.24, the close loop error dynamics is

$$\tilde{y}^{(3)} + k_2 \ddot{\tilde{y}} + k_1 \dot{\tilde{y}} + k_0 \tilde{y} = 0 \tag{2.26}$$

Selecting coefficients k_0 to k_2 to diminish the error over time achieves tracking and smooth landing. Soon's result shows an average of 0.1m/s impact velocity using a 42 volt power supply.

The authors in [25] employed a similar flatness-based end controller. To improve robustness, a disturbance estimator and a feed-forward controller is used to compensate for the energy lost before the landing trajectory starts. The disturbance pressure is estimated using a nonlinear observer and the work done by the disturbance over the distance, w_{gs}, is determined to be:

$$w_{gs}(L) = \gamma (c_1 L + c_2 \frac{L^2}{2} + c_3 \frac{L^3}{2}) \tag{2.27}$$

where γ is the estimated initial pressure. L is the armature position and c_1 to c_3 are curve fitting constants. To compensate for the work done by the disturbance, the catching coil delivers work, $W_d(L)$, in the following fashion:

$$W_d(L) = \gamma \int (c_1 + c_2(L - x_{stroke}) + c_3(L - x_{stroke})^2) dL \tag{2.28}$$

where $(L - x_{stroke})$ is the air gap to the catching coil. Note that this trajectory is chosen to allow the force to be compensated after the armature has swung pass the middle point. The magnetic co-energy equation is then applied to convert desired work into desired current to be used for feed-forward current control.

2.3 Existing Cycle Adaptive Control Schemes

This thesis focuses on a cycle adaptive strategy to implement the EMV feedforward controller (i.e., applying the error information from previous iterations to allow better decision making in the current iteration). Consequently, reviewing of controllers based on the same principle is of great importance. The author believes two publications are most significant: the iterative learning controller [2] and the extremum seeking controller [3] , both of which are published by the same research group in University of Michigan. In order to gain more insight about these cycle-adaptive controllers, the author of this thesis has simulated both systems using the available information in [2] and [3]. Using these simulation results, the effectiveness of the approaches are analyzed. Due to its similarity with iterative learning control and the lack of detailed information, the repetitive learning controller in [43] is not simulated.

2.3.1 Iterative learning control

In [2], an LQR controller based on a system model linearized around an equilibrium point is used to track a desired smooth landing trajectory for the last 0.25mm air gap. For the rest of the trajectory (i.e. between 8mm to 0.25mm airgap), a feedforward voltage u_l tuned by an iterative learning controller (**ILC**) is used. In addition to feedforward voltage, input to the LQR controller, y_r, is also tuned by ILC such that the trajectory tracking error is reduced as iteration increases.

In [2], the sampling index at 0.25 mm airgap is denoted as n_{fb}. The inputs computed from the ILC controller consist of u_l and y_r at different sample time:

$$\vec{u} = \begin{bmatrix} u_l[0] \\ \vdots \\ u_l[n_{fb}-1] \\ y_r[n_{fb}] \\ \vdots \\ y_r[n_{fb}+N-1] \end{bmatrix} = \begin{bmatrix} \vec{u_l} \\ \vec{y_r} \end{bmatrix} \in \Re^M \tag{2.29}$$

The ILC output is the measured trajectory after n_{fb}, for a total number of N samples.

$$\vec{y} = \begin{bmatrix} y[n_{fb}+1] & \dots & y[n_{fb}+N] \end{bmatrix}^T \tag{2.30}$$

As a result, the ILC linear model in [2] contains different numbers of inputs and outputs in contrast to the mainstream even-input/output ILC implementations surveyed by [51] and [52]. The matrix that relates input to output can be further divided into P^{φ}, which converts the feedforward voltage into the position at n_{fb}, and P^{y_r}, which is the convolution matrix of the LQR controller with respect to reference trajectory:

$$\vec{y} = \overrightarrow{y^{\varphi}} + \overrightarrow{y^{y_r}} = \underbrace{\begin{bmatrix} P^{\varphi} & P^{y_r} \end{bmatrix}}_{P} \begin{bmatrix} \vec{u_l} \\ \vec{y_r} \end{bmatrix} \tag{2.31}$$

The iterative learning controller is essentially an integrator. The weighted input and error from iteration k are added together to compute the input for iteration $k+1$:

$$u[k+1] = S(u[k]) + E(y_d - y[k]) \tag{2.32}$$

In case of [2], the weighting matrices are defined by $S \quad R\Lambda_s R^T$ and $E = \Lambda_E L^T$, where the orthonormal matrices, E and L, come from the singular value decomposition of the system matrix P in equation 2.31:

$$P = L \begin{bmatrix} \Sigma | & 0 \end{bmatrix} R \quad R \in \mathbf{R}^{M \times M} \quad \text{and} \quad L \in \mathbf{R}^{N \times N} \tag{2.33}$$

where Σ is an $N \times N$ diagonal matrix with diagonal entries, $\sigma_0 > \sigma_i > 0$. The work in [2] demonstrates that by choosing the ILC diagonal weighting matrices S and E, with diagonal entries, $s_i = 1, e_i = \sigma_0 \forall i \in [0, N-1]$ and $s_i = 0 \forall i \in [N, M-1]$, the computed input becomes the product of the pseudo-inverse matrix of P and the desired trajectory, y_d,

$$u^{\star} = R \begin{bmatrix} \Sigma^{-1} \\ 0 \end{bmatrix} L^T y_d = P^{\dagger} y_d \tag{2.34}$$

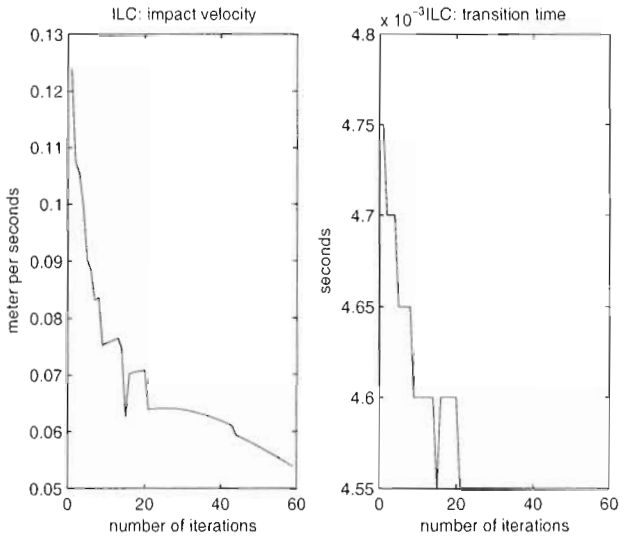

Figure 2.1: Armature impact velocity and iteration time versus iterations under ILC control: simulation using available data from [2].

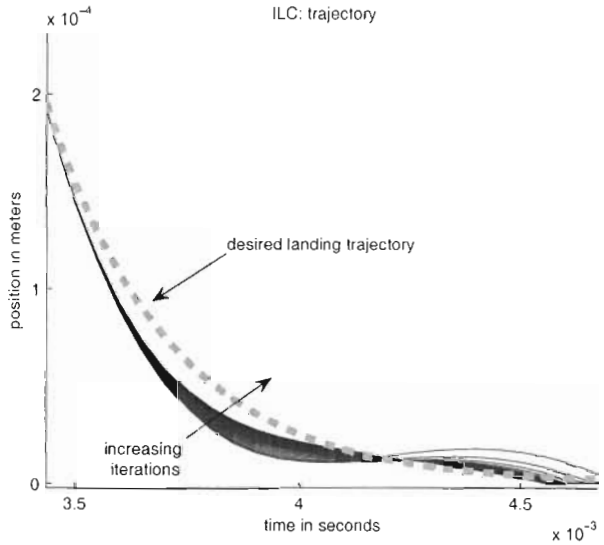

Figure 2.2: Evolution of armature landing trajectory under ILC control: tracking improves as iteration increases.

Simulation result

The work in [2] reported the ability to maintain a soft landing impact velocity of 0.04 m/s at an average transition time of 3.9ms. The author's simulation result shows impact velocity and time in Figure 2.1. Also, the evolution of the landing trajectory is available in Figure 2.2. One can see that the actual trajectory matches more closely with the desired trajectory as the number of iterations increase. The simulated result affirms the result in [2], but it also exposes the following issues:

- The duration of the valve travel changes, so the sample size of matrices change. The linear model becomes inadequate if the back pressure changes.
- Since armature bouncing is not modeled, the ILC controller will have problems distinguishing what can be learned and what must be avoided.
- Because the coil magnets in the EMV system can only pull the armature, the braking is only achieved through spring forces. The desired trajectory in the ILC may cause a problem if it requires armature braking beyond what the spring can allow. This problem is compounded by the integrative nature of the ILC.
- The trajectory tracking only takes place at the last 0.25mm in the overall 8mm of armature travel. Perhaps better modeling of the system (as opposed to Eq. 2.31) can allow tracking over a longer distance and can improve robustness.

2.3.2 Extremum seeking control

In [3], the extremum seeking strategy is used to tune one parameter of a nonlinear controller of the following form:

$$V_c = \begin{cases} \frac{K_1}{\gamma+z}v + \frac{K_2}{\beta+z} & \text{if } z \leq 2l - l^{-3} \\ 0 & \text{if } z \geq 2l - l^{-3} \end{cases} \tag{2.35}$$

where $2l$, v, and z are total armature travel, velocity, and airgap respectively, while the rest are controller parameters. Eq. 2.35 contains insight into the EMV system: the second term, $\frac{K_2}{\beta+z}$, ensures that large input are only applied at small airgap to ensure effective energy use and to combat the large inductance at the small airgap. The first term, $\frac{K_1}{\gamma+z}v$, is a nonlinear damping gain to shape the force usage. Most importantly, the parameter β shapes the

transition of the second term from small to large and is tuned by the extremum seeking cycle-adaptive controller.

To understand extremum seeking control, one must understand the theorem of averaging [53] first. If given a system of the form:

$$\frac{dx}{dt} = \epsilon f(t, x, \epsilon) \quad \epsilon > 0 \tag{2.36}$$

where $f(t, x, \epsilon)$ is smooth and periodic with a period T, then x is close to its average x_{av}

$$\begin{aligned} ||x(t,\epsilon) - x_{av}(t,\epsilon)|| &= O(\epsilon) \quad \text{where} \\ \frac{dx_{av}}{dt} &= \frac{\epsilon}{T}\int_0^T f(\tau, x, 0)d\tau \end{aligned} \tag{2.37}$$

The dynamics of the extremum seeking controller uses a sinusoidal input excitation to take advantage of the averaging theorem over a static nonlinearity, $Q[x + a\sin(\omega t)]$. The dynamic of the parameter, x, is designed to be

$$\frac{dx}{dt} = Q[x + a\sin(\omega t)]a\sin(\omega t) \tag{2.38}$$

To convert Eq. 2.38 into the form of Eq. 2.36, define $\tau = \omega t$ and $\tilde{x}(\tau) = x(\tau/\omega)$

$$\frac{d\tilde{x}}{d\tau} = \epsilon Q[\tilde{x}a\sin(\tau)]a\sin(\tau) \tag{2.39}$$

Applying the averaging theorem results in Eq. 2.37,

$$\frac{d\tilde{x}_{av}}{d\tau} = a^2\epsilon\left(\frac{1}{2\pi}\int_0^{2\pi} sin^2(\tau)d\tau\right)\frac{dQ}{d\tilde{x}}(\tilde{x}_{av}) \tag{2.40}$$

The above result is significant because it implies that the extremum of Q happens when the average of the variable x stops changing. Since the $\tilde{x}_{av}$ can only be stable if $\frac{d^2Q}{dx^2} < 0$, then the equilibrium $\tilde{x}_{av}$ is also a local maximum of Q. To seek a minimum instead of a maximum, replace the integrator sign with -1. In the case of the EMV actuator control in [3], the static nonlinearity, Q, is related to the valve impact noise, $S_{meas}[k]$, detected by microphone.

$$Q = (S_{des} - S_{meas}[k])^2 \tag{2.41}$$

The discretized extreme seeking excitation input is added to β in Eq. 2.35:

$$\begin{aligned} x[k+1] &= x[k] + \triangle T a\sin(\omega k\triangle T + \phi)(S_{des} - S_{meas}[k])^2 \\ \beta[k] &= x[k] + a\sin(\omega k\triangle T + \phi) \end{aligned} \tag{2.42}$$

where $\triangle T$ is the sampling period, $\omega = \pi/(\triangle T)$ and $\phi = \pi/2$

Simulation result

In Figure 2.3, the author's simulated extremum seeking controller produces a similar trend for impact velocity reduction as in [3]. Also, as described by [3], the method is simple to implement and the convergence speed is reasonable. However, one issue becomes obvious: tuning one variable is not enough to facilitate a landing speed of less than 0.1m/s. Eq. 2.35 has four variables and changing only one does not provide enough flexibility. Further investigations on extremum seeking control reveal that multiple parameter tuning is available only on a linear plant and at a reduced convergence rate, depending on the number of variables tuned [54].

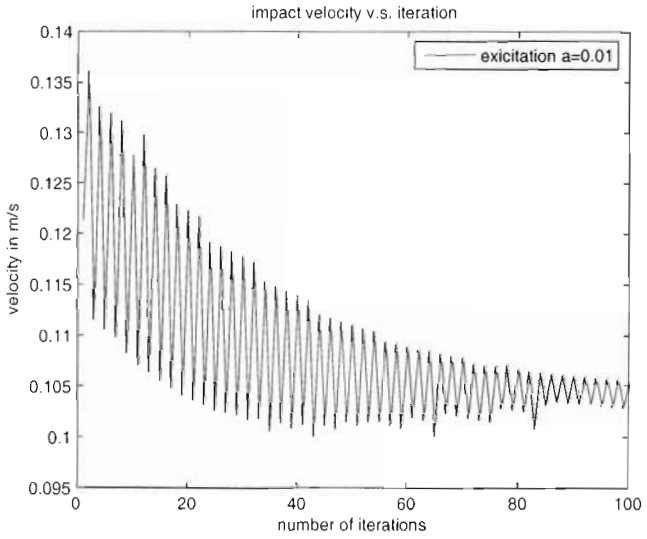

Figure 2.3: Armature landing velocity versus iterations under extremum seeking control: simulation using available data from [3].

2.4 Summary

In this chapter, the important publications on modeling and control of the electro-magnetic valve actuator were presented. The modeling section included modeling work for magnetic force generation and for applied valve control. The feedback control section included sensorless, various linearization-based, CLF based, sliding mode, and flatness-based controllers. Last, but not least, two cycle-adaptive controllers (iterative learning and extremum seeking) were introduced, simulated, and analyzed for their pros and cons. Table 2.1 lists the publications covered in the chapter.

Table 2.1: Various EMV control schemes

Contributor	Approach control /cycle-adaptation	Landing
Ch. Gunselmann and J. Melbert [44]	landing current level adaptation	quotient observation
C. Tai and T. Tsao [23]	repetitive learning	frequency identified LQR
W. Hoffmann and K. Peterson and A. G. Stefanopoulou [2]	iterative learning	equilibrium-linearized LQR
K. Peterson and A. G. Stefanopoulou [3]	extremum seeking	nonlinear feedback
P. Eyabi and G. Washington [41]	tuned open-loop	sliding mode tracking
K. S. Peterson and J. W. Grizzle and A. G. Stefanopoulou [42]	N.A. (extremum seeking is used in Peterson's thesis [48])	control Lyapunov function
S.K. Chung and C.R. Koch and A.F. Lynch [24]	tuned open-loop	flatness-based tracking
R. R. Chladny and C. R. Koch [25]	estimated energy-based feedforward compensation	flatness-based tracking

The approach controllers developed in this thesis complements the landing controllers in [24] or [41] and are based on cycle adaption similar to [2] and [3]. Since this thesis aims to improve upon the performance of the existing cycle-adaptive controllers for approach control, the issues identified (such as limitation on model, saturation, and parameter tuning) will be addressed in the following chapters based on the author's own controllers.

Chapter 3

Actuator Model and Disturbance Estimation

The model used for simulating the EMV actuator, developed by Soon Chung from the University of Alberta [1], is presented in this section. In addition to the actuator model, the disturbance model, which depends only on position and initial disturbance, is discussed. Lastly, a promising disturbance pressure estimation, based on induced voltage observations, is introduced and verified against various pressure data sets.

3.1 TEMIC Actuator Model

Ryan Chlandy and Soon Chung from the University of Alberta have developed the model for the TEMIC actuator. The work in Chlandy's master thesis [4] established the finite element model-based lookup table and verified it against experimental data from the MTS actuator loading machine. In Chung's master thesis [1], he further simplifies, Chladny's model into a lump parameter model, which is suitable for control system development purpose. The model discussed in this section is taken directly from Chung's thesis [1].

3.1.1 Linear mechanical model

The EMV actuator is basically a mass-spring-damper system with a non-linear magnetic applied force. Therefore, the mass-spring-damper portion is easily written as a second order

linear differential equation:

$$m\frac{d^2x}{dt^2} = F_{mag}(x, i_c) - K_s x - B\frac{dx}{dt} \tag{3.1}$$

where x is the armature position, F_{mag} is the magnetic force, and i_c is the coil current. The rest are coefficients defined as

$$\begin{aligned} \text{Spring constant} \quad K_s &= 250.98 N/mm \\ \text{Damping constant} \quad B &= 12.75 Ns/m \\ \text{Mass} \quad m &= 0.28 kg \end{aligned}$$

3.1.2 Current to magnetic force model

The model for magnetic force generation is much more complicated. Not only is the magnetic force nonlinear (roughly proportional to inverse distance squared), the effect of magnetic flux saturation and eddy current must be accounted for. The following function characterizes the current, i_c, to magnetic flux, $\lambda(x, i_c)$, relationship:

$$\lambda(x, i_c) = \lambda_s \left[1 - e^{-i_c f(x)}\right], \; i_c \geq 0 \tag{3.2}$$

where λ_s is the maximum saturated flux, and the function, $f(x)$, is fitted to describe saturation and parameterized as the following:

$$\begin{aligned} f(x) \qquad &= \frac{2C_1}{C_2 - x} + C_3 \\ \lambda_s = 0.076 Wb \quad &\text{and} \quad C_1 = 2.32e^{-2} mm/A \\ C_2 = 4.04 mm \quad &\text{and} \quad C_3 = 4.18e^{-2} A^{-1} \end{aligned} \tag{3.3}$$

To determine the force through the available flux, co-energy analysis is applied. First the flux is integrated with respect to the coil current, i_c, to find the co-energy, $W_c(x, i_c)$:

$$W_c(x, i_c) = \int_0^{i_c} \lambda(x, \xi) d\xi \tag{3.4}$$

The derivative of the co-energy with respect to distance is the magnetic force. The final expression of the magnetic force is derived by substituting in the flux equation (Eq. 3.2):

$$\begin{aligned} F_{mag}(x, i_c) &= \frac{dW_c(x, i_c)}{dx} \\ &= \frac{\lambda_s f'(x)}{f^2(x)} \left[1 - [1 + i_c f(x)] e^{-i_c f(x)}\right] \end{aligned} \tag{3.5}$$

where $f'(x)$ is derivative of $f(x)$ with respect to x.

3.1.3 Voltage-current Model

The basic form of the voltage-current model is an RL circuit. For a more accurate model of the valvetrain, the current, i, is separated into two parts: coil current and eddy current. The input, voltage contributes to the coil current i_c, which generates flux $\lambda(x, i_c)$, and eddy current, i_e, which converts energy into heat. On top of the coil resistance, R_c, the eddy current has its own dynamics which involves a second branch of resistance, $R_e(x, i_c)$, and inductance, $L_e(x, i_c)$. The expressions of the voltage-current model are given in the following:

$$U_c = R_c i + \frac{d}{dt}\lambda(x, i_c) \tag{3.6}$$

$$\frac{di_c}{dt} = \frac{1}{d\lambda(x, i_c)/di_c}\left(U_c - R_c i - \frac{d\lambda(x, i_c)}{dx}\frac{dx}{dt}\right) \tag{3.7}$$

$$= \frac{e^{i_c f(x)}}{\lambda_s f(x)}\left[U_c - R_c[i_c + i_e]\right] - \frac{f'(x)}{f(x)} i_c v \tag{3.8}$$

$$\frac{di_e}{dt} = \frac{1}{L_e(x, i_e)}\left[U_c - R_c(i_c + i_e) - R_e(x, i_c) i_e\right] \tag{3.9}$$

$$i = i_c + i_e \tag{3.10}$$

3.1.4 Complete EMV model

By combining all the models described in section 3.1.1 to 3.1.3, the complete model from the input voltage to the output position measurement can be assembled:

$$\frac{dx}{dt} = v \tag{3.11}$$

$$\frac{dv}{dt} = \frac{1}{m}\left[\frac{\lambda_s f'(x)}{f^2(x)}\left[1 - [1 + i_c f(x)]\, e^{-i_c f(x)}\right] - K_s x - Bv\right] \tag{3.12}$$

$$\frac{di_c}{dt} = \frac{e^{i_c f(x)}}{\lambda_s f(x)}\left[U_c - R_c[i_c + i_e]\right] - \frac{f'(x)}{f(x)} i_c v \tag{3.13}$$

$$\frac{di_e}{dt} = \frac{1}{L_e(x, i_e)}\left[U_c - R_c(i_c + i_e) - R_e(x, i_c) i_e\right] \tag{3.14}$$

$$\tag{3.15}$$

Unfortunately, $R_e(x, i_c)$ and $L_e(x, i_c)$ are highly complex due to their dependence on the position and current state. In the rest of the thesis, the eddy current is treated as a disturbance rather than as part of the system dynamic.

3.1.5 Challenges for control

The complete model captures the three major difficulties associated with the effective control of the EMV actuator.

1. The actuator can only act in one direction (i.e., it contains no braking capability). So once the velocity is too high, there is no way to slow down the actuator with the catching coil.

2. The magnitude of magnetic force is strongly dependent on position. It is roughly inversely proportional to the armature-valve gap distance squared. So if the armature is far away from the catching coil, the system suffers from low control authority. Figure 3.1 illustrates the force curve with respect to distance.

3. Before changing the magnitude of the pulling force in the actuator, the current in the coils must be changed. Unfortunately, the combination of stiff spring and system inductance in the EMV means that the current rise time is significant. To make matters worse, the system contains a phenomenon in which the closer the armature is to the catching coil, the harder it is to raise the current. Figure 3.2 illustrates this problem. It is a plot of the current derivative (Eq. 3.13) with respect to position while voltage and current are held constant. The multiple lines shows that the increasing armature velocity, results in a slower increase in the current near the end of the valve flight.

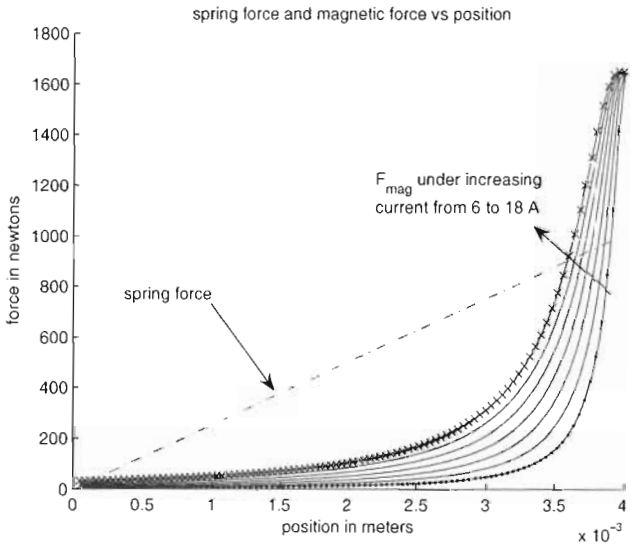

Figure 3.1: Force exerted from closer/upper electromagnet versus armature position. The force input drops off sharply with distance. (Simulated using Eq. 3.5)

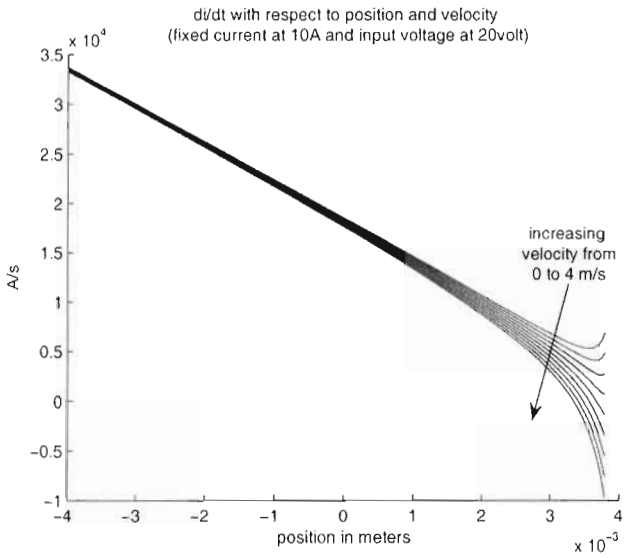

Figure 3.2: Time derivative of coil current versus armature position. The ability to raise current using the same voltage reduces as air gap distance decreases.

3.1.6 Pressure disturbance model

The major disturbance to the EMV actuator comes from the engine back pressure. During an exhaust valve opening, the actuator works against the pressure force inside the combustion chamber. According to the pressure experiment data gathered[1](see Figure 3.3), the disturbance pressure during the actuator's opening operation can be approximated reasonably by a model that depends only on position and initial pressure magnitude. The evidence of this is in Figure 3.4, in which all six normalized force-versus-position trajectories falls onto one single curve. From this curve, the disturbance model is determined:

$$F_{dist}(x) = A_{bar}\left(p_m(x+4e-3)^2 - q_m(x+4e-3) + r_m\right) \tag{3.16}$$
$$p_m = 22875/2 \; q_m = 987/8; \; r_m = -1353/2000$$

The parameter, A_{bar}, is the initial gauge back pressure value in atmosphere air pressure (bar), while p_m, q_m, and r_m are curve fitting coefficients. By knowing A_{bar}, one can reasonably predict the subsequent force by using the armature position. This approach simplifies the disturbance estimation process.

[1]This set of data is first referred to by Quong's co-op report [55]. However, the actual experimental work is done by Mercedes Benz in Germany before Quong's work.

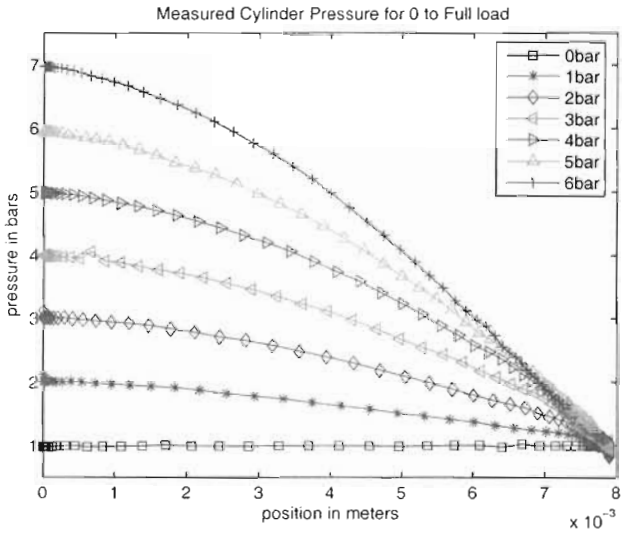

Figure 3.3: Engine cylinder back pressure versus armature position with back pressure varying from 0 to 6 bars.

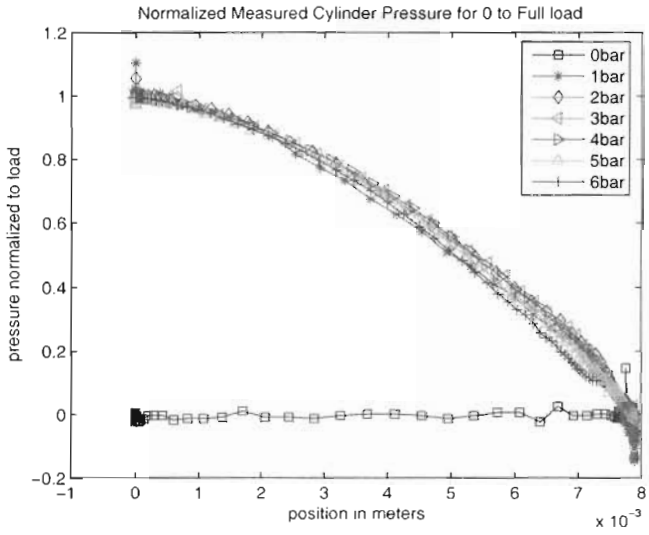

Figure 3.4: Normalized engine cylinder back pressure versus armature position with back pressure varying from 0 to 6 bars. All normalized pressure v.s. position trajectories fall onto the same curve with the exception of zero curve, which can't be normalized.

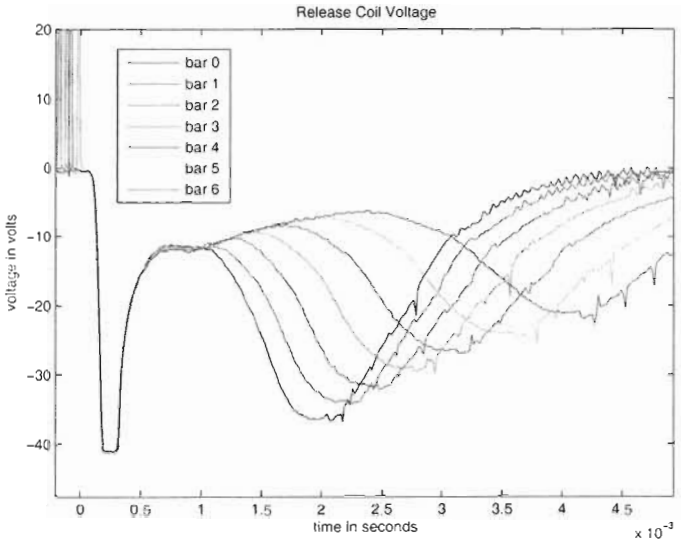

Figure 3.5: Release coil voltage v.s. time curves during the armature flight under 0 to 6 bars of engine back pressure.

3.2 Estimating Pressure Disturbance Via Release Coil Voltage Measurement

Determining the disturbance pressure can be achieved through processing the measured position (e.g., [55] and [56]). However, estimating pressure using position measurements can be difficult because position change is largely determined by spring force. An alternative would be to use release coil induced voltage as the input measurement. In 2003, Gunselmann and Melbert [44] observed that the release coil, when switched to open, will have an induced voltage at the beginning of the armature movement. They stated that the induced voltage is caused by the fast flux decrease in the release coil as the coil current is driven down to zero quickly by the power transistors. However, [44] does not provide the details on how pressure can be determined from induced voltage.

The experimental data (Figure 3.5) available to the author shows similarity with the induced voltage data from [44]. Therefore, an attempt is made to estimate pressure data using the induced release coil voltage with inverse model identification.

3.2.1 Pressure estimation though inverse model identification

In the EMV actuator, motion of the coil causes an induced voltage. Since the cylinder pressure changes the motion, a different induced voltage can be detected. Cylinder pressure is the cause of induced voltage; thus, the pressure is the input, and the voltage is the output. Using the release coil voltage measurement to determine the initial disturbance pressure requires the opposite. During the identification procedure, seven sets of pressure and induced voltage experimental data (shown in Figure 3.6) are used as inputs and outputs of the inverse model. The data sets represent zero to six times of the atmospheric pressure (0 to 6 bars). These seven data sets are separated into working and validation sets. The working set is used to identify the inverse model. Then the induced voltage data from the validation set is fed into the inverse model to test whether the output from the identified model matches with the actual values.

System identification using Matlab is applied to determine an inverse model that can take the coil voltage as the input and produce a pressure estimate as the output. The working data used in identification are taken from the experimental data with a 3 bar pressure input. In Figure 3.6, the data starts from 1ms to the time that the voltage reaches its most negative point. The reason for starting at 1ms is that all the coil voltages follows the same path (seen in Figure 3.5) before 1ms. The input/output data sets terminate not at a set time, but at a distance 0.3mm away from the release coil.

The experimental data set can be fitted to various models (frequency-based, state-space, polynomial, etc) in the Matlab model identification toolbox. The process model, *P2*, is chosen here because it provides best match between actual and model outputs. *P2* stands for a second order transfer function with no dead time. Matlab identifies the *P2* model from the working data as

$$G(s) = \frac{K}{(1+T_{p1}s)(1+T_{p2}s)}$$

$$K = 47987$$

$$T_{p1} = 3278.9 \quad \text{and} \quad T_{p2} = 0.004446$$

Testing the identified inverse model on the validation data produces excellent results. Figure 3.8 shows the model output for the verification data sets (the induced voltage and pressure pairs for bar 0, 1, 2, 4, 5, and 6). Note the jagged edges of the experimental data 3.8 are simply measurement noises. Also note that the pressure values listed are 1 bar above

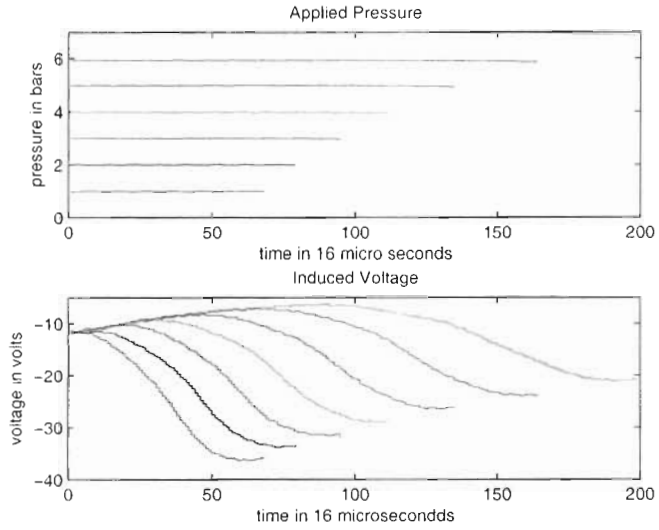

Figure 3.6: The input (release coil voltage) and output (known cylinder pressure) data sets for identifying and validating the inverse model used for pressure disturbance estimation.

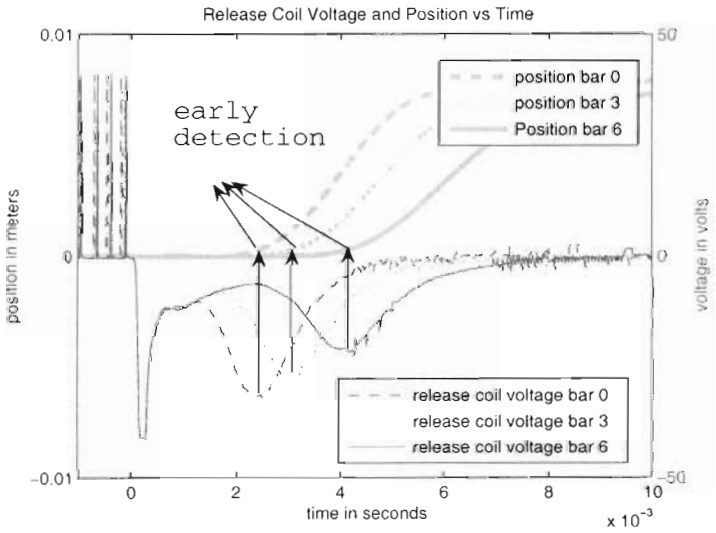

Figure 3.7: Armature position and induced voltage versus time. Inverse model input is available before the armature takes flight.

the disturbance because the atmospheric pressure. For example, at zero bar disturbance, the measurement reads 1 bar due to atmospheric pressure. The estimated pressure data from the inverse model are **within 0.025 bar of the actual pressure**.

While the process model produces the smallest prediction error, various other models with different orders have been tested. These resultant models can be examined through their impulse or frequency response, location of poles and zeros, correlation analysis of the residuals, and measured and modeled output comparison. In terms of the prediction errors, all the model errors are within 0.05 bar for all 6 validation data sets.

For example, auto-regressive model with exogenous input (ARX) model such as the following was tested:

$$A(q)y(t) = B(q)u(t) + C(q)e(t) \tag{3.17}$$

where $A(q)$, $B(q)$, and $C(q)$ are polynomial functions of the delay operator q^{-1}. The output, input, and error parameters are $y(t)$, $u(t)$, and $e(t)$ respectively. Closely related models to the ARX include *box-jenkins*, *output error*, and autoregressive moving average with exogenous input ($ARMAX$) models.

Finally, various orders of state space model were tested out as well:

$$x(t+1) = Ax(t) + Bu(t) + Ke(t) \tag{3.18}$$

$$y(t) = Cx(t) + Du(t) + e(t) \tag{3.19}$$

where A, B, C, D, K are fixed matrices to be determined by the identification procedures.

Once the disturbance pressure value at the beginning of the valve flight is available, it can be fed into the disturbance model in Eq. 3.16 as the initial disturbance value, A_{bar}. For the rest of the valve flight, the controller can utilize this disturbance pressure estimate to compensate for any energy shortfall.

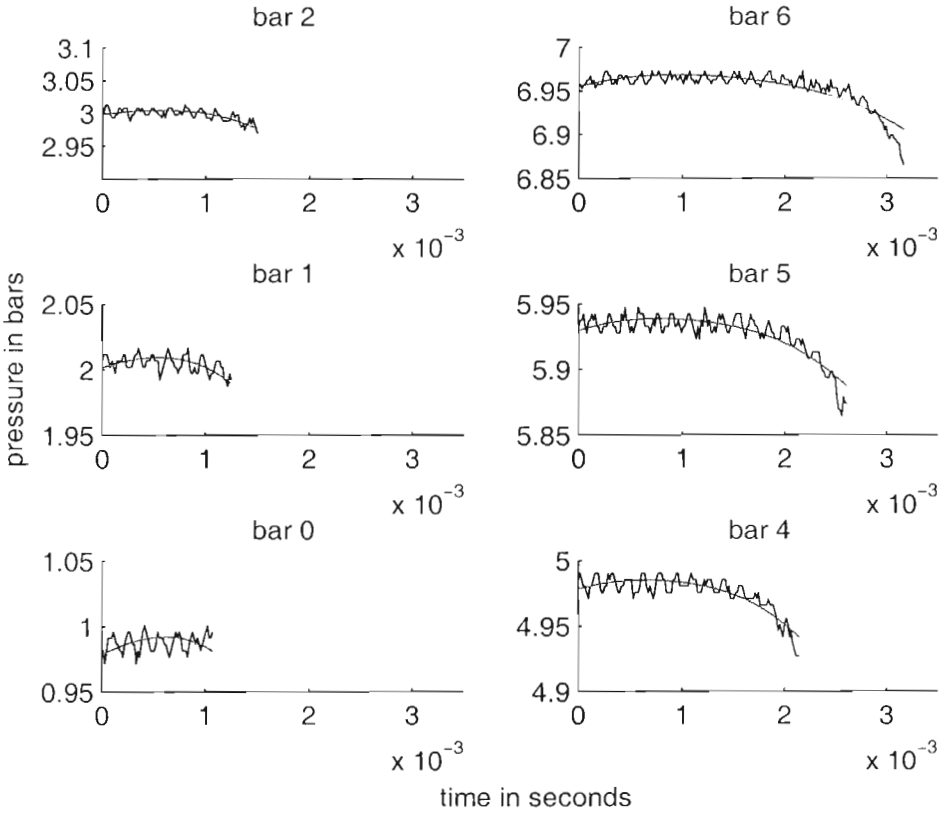

Figure 3.8: Actual and inverse model (Eq. 3.17) estimated engine back pressure from 0∼6 bar. The smooth lines are estimated, while the jagged lines are actual measured values. All estimation errors are less then 0.025 bars.

The following are the advantages of using an identified inverse model:

Early detection Using the inverse model, pressure data is available even as the armature is taking off. This is best illustrated in Figure 3.7, where the dip in the voltages are compared to position measurements. In contrast, the position-based method requires some position measurements to pass through in order to converge.

Robustness against measurement noise With input changing from -10 volt to -40 volts, the induced voltage's signal magnitude is much higher than any potential noise in voltage measurement. The same can not be said about the armature position measurement, where high signal to noise ratio comes at a steep cost.

Computational Simplicity Compared to filtering the position measurement for pressure estimation, the computation required for a model like Eq. 3.17 is small.

However, the practicality of this algorithm depends on how representative the validation results are. If the inverse model holds for most of the service life of the actuator in the engine environment, then there is no problem. However, if model identification must be performed often because the model coefficient changes quickly over time, then this method is not nearly as useful. Since no more experimental data is available at the time of this writing, the author is not uncertain how well inverse model identification will work in the actual valve operation. However, given all the advantages stated above, this method deserves further investigation.

3.3 Summary

In this chapter, the analytical EMV actuator model used for controller design and simulation in this thesis was introduced. The model includes equations for electrical, mechanical, and pressure disturbance. The electrical system model also includes nonlinear effects of eddy current and flux saturation. The pressure disturbance model uses the position dependence of the pressure forces to simplify the analytical equation. Lastly, in addition to the system model, a disturbance estimation method using inverse model identification was introduced and tested in this chapter with satisfactory results.

Chapter 4

Control Methodologies

To effectively tackle the electromechanical valve soft-seating problem, the controller is divided into two sections: the approach controller is responsible from armature takeoff at $-4mm$ to the controller transition point, x^f_{appr} at $2.55mm$. For the rest of trajectory, the landing controller is active (see Figure 1.2). The approach controller regulates the end conditions so that the landing controller can start landing trajectory tracking under consistent initial conditions. Specifically, the approach controller regulates the velocity, v^f_{appr}, and current, i^f_{appr}, to a window around the desired values at the end of the approach control.

In this thesis, the author attempts to facilitate effective approach control using cycle-adaptive feed-forward controllers that utilizes the information from previous valve events. In this chapter, three cycle-adaptive approach controllers are proposed, which includes two iterative learning controllers (Nonlinear ILC and Terminal ILC) and one Nelder Mead direct search controller. The simulated performance of the three controllers are discussed at the end of this chapter, and depending on the simulated performance, some of the controllers were tested on experimental test-bench.

4.1 Nonlinear Iterative Learning Control (Nonlinear ILC)

This section begins by adapting Peterson and Stefanopoulou's iterative learning control paper [2] from landing control to approach control. (A short overview of the paper [2] is available in the Chapter 2). To enable tracking over a longer distance, and thus improve upon the work in [2], a nonlinear iterative learning methodology is applied that requires no operating point-based linearization. Also, in the process changing from landing control to

approach control, the robustness of the iterative learning algorithm improves because it no longer suffers from the armature bouncing problem.

Iterative learning control is originally developed as a linear methodology. In [2], Peterson and Stefanopoulou linearized the system based on an operating point in order to adopt ILC to the high nonlinear EMV system. As a result of the operating point based linearization, controller tracking is only done in the last 0.3mm of the valve flight. The Nonlinear ILC method developed in this section attempts to alleviate this problem by using the convergence result derived by Sun and Wang [57].

The Nonlinear ILC developed in this section will track a trajectory between 1mm to x^f_{appr} at 2.55mm (recall that the armature travels between -4mm to 4mm). The reason for choosing this starting point at 1mm is that the control authority is very small before 1mm (see Figure 3.1). Ideally, tracking a longer trajectory (1.55mm as opposed to 0.3mm) would allow more time for control and thus a better result.

4.1.1 Convergence Analysis

The work in [57] outlines a convergence proof for iterative learning control of a continuous nonlinear system with discrete observation of the form:

$$\dot{x}(t) = F(x(t), u(t)) \tag{4.1}$$

$$y(jh) = G(x(jh)) \tag{4.2}$$

where h is the sampling period and j is the sample number. The controller is of the form:

$$u_{k+1}(jh) = u_k(jh) + \Phi_k(jh)\left[e_k(jh+h) - \sum_{i=0}^{\mu-1} \frac{h^i}{i!} e_k^{(i)}(jh)\right] \tag{4.3}$$

where u is the control input between 1mm to x^i_{land}, k is the iteration number, e is the tracking error, Φ is the learning gain, μ is the relative degree of the system, and d_k is the maximum bound for how input affects output derivative.

For the tracking error to converge, the learning gain ϕ must be selected such that

$$1 > \left|1 - \frac{h^\mu}{\mu!}\Phi_k(jh)d_k(jh)\right| \tag{4.4}$$

$$d_k(jh) = \sup\left[\frac{d}{du}L_f^{\mu-1}G(x_k(jh), u(k))\right] \tag{4.5}$$

The assumptions listed in [57] are:

1. The system is slowly varying.
2. The system is Lipschitz.
3. The trajectory is realizable.
4. The sampling period must be small with respect to the Lipschitz constants (see Eq. 29 of [57]).
5. The error to initial condition is bounded.

4.1.2 Computing desired trajectory to track

Due to the severe non-linearity of the system as explained in the previous chapter, the undisturbed trajectory can not be used as the desired trajectory; otherwise, the controller will immediately saturate the input voltage. A better idea is to interpolate the disturbed trajectory with the undisturbed trajectory in the position velocity plot to generate a new trajectory, $V_d(x)$.

$$V_d(x) = V_i + (V_u - V_i)\left(\frac{\tanh((x+\alpha)\beta) + 1}{2}\right) \tag{4.6}$$

where V_i is the disturbed trajectory and V_u is the undisturbed trajectory in the velocity-position domain. The function tanh goes between 1 and -1 and Eq. 4.6 interpolates $V_d(x)$ between V_u to V_i lines. Other interpolation methods are possible but Eq. 4.6 contains two adjustment parameters α and β that determine at which section and how quickly the lines V_u and V_i merge. Figure 4.1 illustrates such a trajectory. The interpolated line starts from the disturbed line and merges into the undisturbed line. Because the Nonlinear ILC requires both the desired velocity and position with respect to time, $x_d(t)$ is computed by integrating $V_d(x)$.

4.1.3 Simulation Results

To test the performance of the Nonlinear ILC controller, a pressure disturbance of 0.2 bar is applied to the simulated system. The goal of the controller is to ensure the end velocity, v^f_{appr}, stays around 2.9m/s by tracking the desired trajectory. The success in trajectory tracking can be seen from the error convergence plot in Figure 4.2 and trajectory evolution

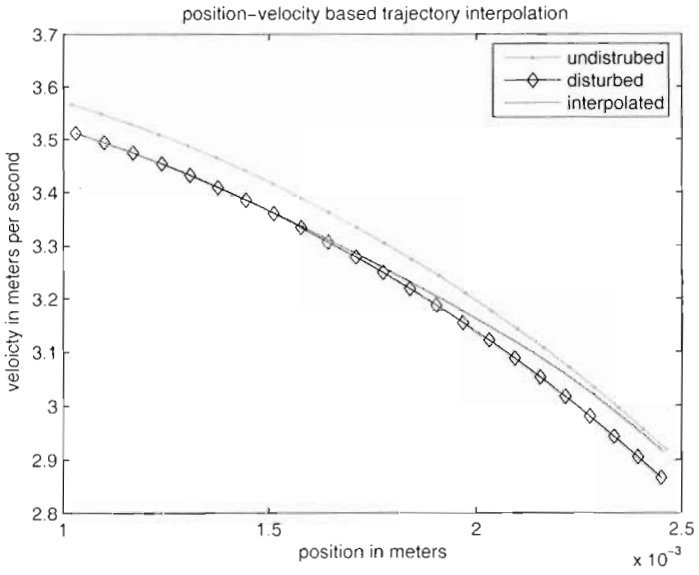

Figure 4.1: Interpolating between disturbed and undisturbed armature trajectory to provide a desired trajectory for the Nonlinear ILC controller.

plot 4.3. Unfortunately, successful trajectory tracking does not guarantee the end velocity condition. We contrast the terminal velocity error in Figure 4.4 with the tracking error norm in Figure 4.2. While the trajectory tracking error has converged, the terminal velocity error doesn't settle to zero. In addition, tracking is achieved with a high cost of input voltage (more than 100 V). The evolution of input voltage and current trajectory can be seen in Figure 4.5 and Figure 4.6. If the voltage input is limited to 42 V as in the real system, saturation will cause steady state errors in trajectory tracking and in terminal velocity (see Figure 4.4).

4.1.4 Discussion

The Nonlinear ILC as investigated in this section is not suitable for the approach control problem for the following reasons:

1. The further away from the catching coil the armature is, the harder it is to control its movement. The simulation shows that the distance of 1mm to 2.55mm is still too far away for tracking. It's nearly impossible to find a realizable trajectory. This negates the fact the controller needs no operating point based-linearization.

2. The goal of the approach controller is not about tracking a specific trajectory, but rather about maintaining the armature within a velocity window at the feedback activation point, x^f_{appr}. As long as the terminal velocity v^f_{appr} differs from the desired velocity v_d, regardless of how well the ILC tracks the desired trajectory, the performance is still unacceptable.

3. ILC can not control the current at the feedback activation point, x^f_{appr}, explicitly. If the current, i^f_{appr}, is too high or too low, control difficulty will arise for the landing controller because the control bandwidth problem under small airgap discussed in Chapter 3.

In conclusion, the nonlinear ILC approach controller is impractical for implementation in the actual system due to its tendency to saturate the input and its inability to directly control the end current and velocity.

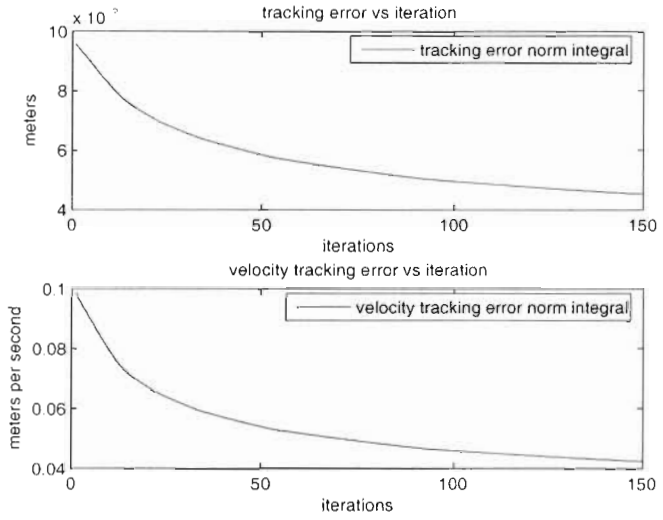

Figure 4.2: Armature position and velocity tracking error norm versus iterations under the Nonlinear ILC controller.

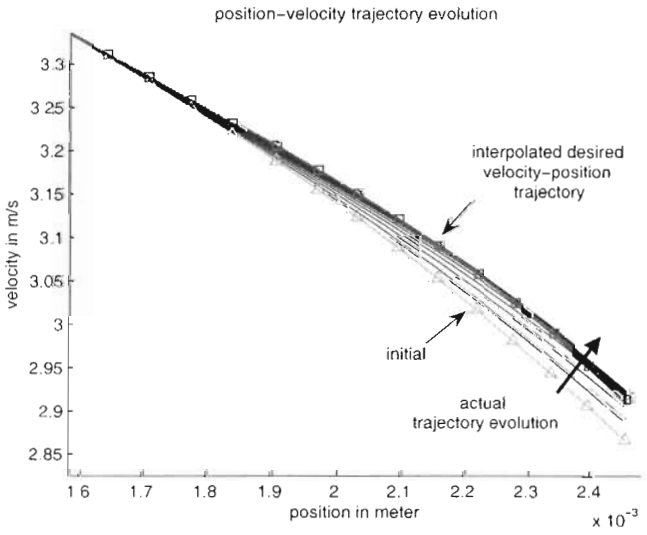

Figure 4.3: Armature position versus velocity trajectory evolution under the Nonlinear ILC controller. The actual armature trajectory approaches the desired as iteration increases.

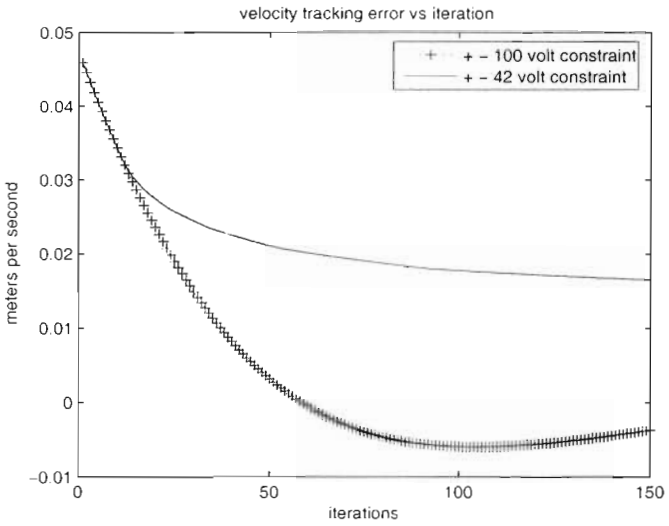

Figure 4.4: v^f_{appr} versus iterations under the Nonlinear ILC controller. The controller fails to eliminate error under $\pm$42V input constraint, but even with a $\pm$100 v input power supply, a small steady state error remains.

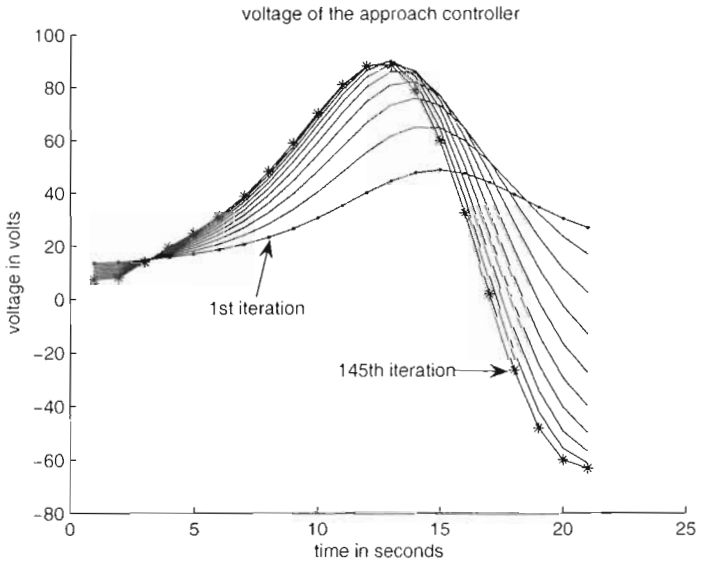

Figure 4.5: Input coil voltage evolution from the Nonlinear ILC controller. As the iteration increases, the voltage input calculated from the nonlinear ILC controller swings wildly and exceeds the ± 42V input constraint.

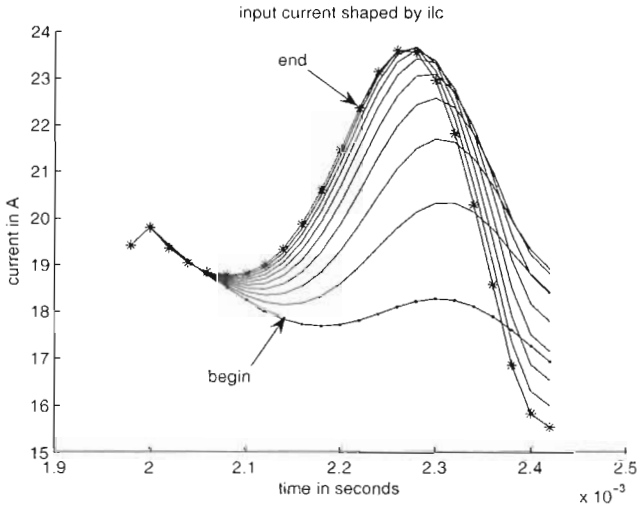

Figure 4.6: Input coil current evolution from the Nonlinear ILC controller. As the iteration increases, the current resulted from the nonlinear ILC controller swings wildly.

4.2 Terminal Iterative Learning Control (Terminal ILC)

Unlike the nonlinear iterative learning control, which strives to track a trajectory (i.e., position and velocity), the terminal iterative learning control focuses only on the two end states: current, i^f_{appr}, and velocity, v^f_{appr}, at the end of approach control. In addition, B-spline interpolation is employed in Terminal ILC to reduce number of parameters needed to define an input profile during a valve event. Lastly, feedback linearization is applied to simplify the prediction of the state and enable the monotonic convergence of the algorithm.

4.2.1 Existing literature on Terminal ILC

The Terminal ILC work done in this thesis follows the semi-conductor wafter processing work of Xu et al. [58], and the wheeled robot trajectory planning work of Oriolo et al. [59]. Xu's terminal ILC work is motivated by the problem of rapid terminal processing chemical vapor deposition (RTPCVD). It is a process where only the terminal output tracking error is available instead of the whole output trajectory. In his work, Terminal ILC learns from the terminal tracking error of the previous trials to adjust the current input interpolating coefficients. Oriolo's paper for wheeled robot trajectory planning uses the same idea for learning from terminal error. On top of that, the paper introduces the idea of "nullspace projection" that allows changing input parameters based on performance or the robustness-related cost function.

4.2.2 Feedback linearization

The convergence of Terminal ILC requires a linear plant that does not depend on the operating point. One way to accomplish this is to use feedback linearization to actively compensate for the system nonlinearity. For the sake of simplification, the eddy current model is not used in feedback linearization. The remaining model has the magnetic force as the only non-linear portion. To linearize the model in section 3.1 through feedback, the time derivative of the magnetic force is chosen to be the new linearized input.

$$U_{new} \equiv \frac{d}{dt}(F_{mag}(x, i_c)) \tag{4.7}$$

Voltage feedback linearization

The relationship between the magnetic force derivative and the input voltage can be derived from taking derivative of the magnetic force equation (Eq. 3.5):

$$\underbrace{\frac{d}{dt}(F_{mag}(x,i_c))}_{U_{new}} = \frac{dx}{dt}\cdot\frac{d}{dx}(F_{mag}(x,i_c)) + \frac{d}{di_c}(F_{mag}(x,i_c))\cdot\frac{di_c}{dt} \tag{4.8}$$

$$= \frac{v\lambda_s}{f(x)^2}\left[f''(x) - \frac{2f'(x)^2}{f(x)}\right](1-(1+i_cf(x))e^{-if(x)}) + \frac{if'(x)}{f(x)}(U_{des}-i_cR) \tag{4.9}$$

By doing some algebra on Eq. 4.9, the desired input voltage, U_{des}, can be computed given the linearized input and the three states (current, velocity, and position). The relationship between the voltage input and the new feedback-linearized input, dF_{mag}/dt, is stated by the next two equations:

$$U_{des} = \frac{f(x)}{f'(x)i}(U_{new} - G(x,i)v) + iR \tag{4.10}$$

$$G(x,i) = \frac{\lambda_s}{f(x)^2}\left(f''(x) - \frac{2f'(x)^2}{f(x)}\right)\left(1-(1+if(x))e^{-if(x)}\right) \tag{4.11}$$

Current feedback linearization

Instead of voltage control, one can also use current control to realize the magnetic force derivative required. To do so, the relationship between magnetic force and current is inverted. The magnetic force is computed based on current and position:

$$F(x,i_c) = \frac{\lambda_s f'(x)}{f^2(x)}\left[1-[1+i_cf(x)]e^{(-i_cf(x))}\right]$$

The above equation (same as Eq. 4.9) can be used to derive current analytically in terms of magnetic force derivative (the linearized input U_{new}) and measured position.

$$i_{des} = \frac{-1}{f(x)}\left[W_{-1}(g(x)/e)+1\right] \tag{4.12}$$

$$g(x) = \frac{U_{new}f^2(x)}{\lambda_s f'(x)} - 1 \tag{4.13}$$

where W_{-1} is the *Lambert W* function, and i_{des} is the current needed to realize the magnetic force U_{new}. The W_{-1} function is an inverse function of we^w. (i.e., Given an expression of we^w, W_{-1} function returns w).

The linearized EMV model after exact feedback linearization is provided below:

$$\underbrace{\begin{pmatrix} \dot{x} \\ \dot{v} \\ \dot{F_{mag}} \end{pmatrix}}_{\dot{X}_{lin}} = \underbrace{\begin{pmatrix} 0 & 1 & 0 \\ -k/m & -b/m & 1/m \\ 0 & 0 & 0 \end{pmatrix}}_{A} \underbrace{\begin{pmatrix} x \\ v \\ F_{mag} \end{pmatrix}}_{X_{lin}} + \underbrace{\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}}_{B} U_{new} \tag{4.14}$$

Limitations of feedback linearization

The feedback linearization equation for the EMV system (eq 4.10) will fail if the current is zero (current in the denominator). Therefore, feedback linearization can only be used when current is above a certain threshold. The bigger problem with the feedback linearization is the system constraints. The static linear inequality constraints (i.e., the ± 42 V input voltage constraint and 0 to 35 A current constraint), after feedback linearization, become dynamic nonlinear constraints. The upper and lower bounds of the linearized control are provided below:

$$-42 \leq U_{old} \leq 42 \Rightarrow -42 \leq \frac{f(x)}{f'(x)i}\left(U_{new} - G(x,i)v\right) + iR \leq 42 \tag{4.15}$$

Rearranging the inequality to obtain a relation on U_{new} provides the following dynamic constraint on U_{new}:

$$(-42 - iR)\frac{f'(x)i}{f(x)} + G(x,i)v \leq U_{new} \leq (42 - iR)\frac{f'(x)i}{f(x)} + G(x,i)v \tag{4.16}$$

4.2.3 Predicting the states at the end of approach control

Terminal ILC requires a predicted final state parameterized by the input coefficients [60]. Determining the final state involves incorporating the B-spline interpolation and integrating the linear state transition equation.

B-spline interpolation

Interpolating the inputs via basis function reduces the number of coefficients to calculate. B-spline basis function is selected because of its ability to represent complex curves with a low degree of polynomial functions while maintaining a smooth second order derivative.

In addition, B-spline's local property and its convex-hull bounding box property [61] are beneficial features for the controller parameterization.

To use B-spline, the entire input profile is divided into 8 uniform sections, and each section is represented by a second order polynomial. The selection of 8 sections comes from the trade-off between the degrees of freedom in specifying current profile and the real-time constraints in the implementation hardware. The interpolation function for each section is:

$$U^j_{new}(t^\star) = \underbrace{1/2 \begin{bmatrix} t^{\star 2} & t^\star & 1 \end{bmatrix} \begin{bmatrix} 1 & -2 & 1 \\ -2 & 2 & 0 \\ 1 & 1 & 0 \end{bmatrix}}_{\phi(t^\star)} \begin{bmatrix} c_j \\ c_{j+1} \\ c_{j+2} \end{bmatrix}^T \tag{4.17}$$

The star in the time variable $t^\star$ signifies the normalization for the time duration to 1. c_js are the control coefficients to be determined by the Terminal ILC algorithm. Because the coefficients, c_js, are shared between adjacent polynomials as required by the B-spline, only ten coefficients are needed to interpolate the entire input trajectory. Consequently, the vector of input coefficient is defined by:

$$C = \begin{bmatrix} c_1 & c_2 & \dots & c_{10} \end{bmatrix} \tag{4.18}$$

If the dF_{mag}/dt profile is not fixed to zero at the start, feedback linearization will demand a voltage that exceeds the 42 volt bound. To prevent this, the first two coefficients, c_1 and c_2 are fixed to zero to force the input profile to start from zero. Thus, the Terminal ILC needs to compute only eight out of ten coefficients.

Integrating for final value

After feedback linearization, the linear differential model and its solution is of the form:

$$\dot{X_{lin}} = AX_{lin} + BU_{new} \tag{4.19}$$

$$X_{lin}(T) = e^{AT}X_{lin}(0) + \int_0^T e^{A(T-\tau)}BU_{new}d\tau \tag{4.20}$$

The solution can be further simplified because U_{new} is parameterized on the B-spline basis. The predicted terminal value is a linear combination of initial conditions and input coefficients.

$$X_{lin}(T) = V_n(T)X_{lin}(0) + W_n(T)C \tag{4.21}$$

T is defined as the total time required for the armature to travel to the end of the approach control. Even though different control inputs will result in different Ts, the period T from the previous iteration is assumed to be roughly the same as the current one. The rationale is that the EMV system is dominated by the spring force and (to a lesser extent) pressure, and neither of those values are assumed to change much between consecutive iterations. The time at the end of the first-out-of-eight B-spline sections is defined to be $\triangle T = T/8$. To determine the matrices, $V_n(T)$ and $W_n(T)$, one can start by computing the states at time, $\triangle T$, and define two intermediate matrices, $V(\triangle T)$ and $W(\triangle T)$:

$$X_{lin}(\triangle T) = e^{A\triangle T}X_{lin}(0) + \int_0^{\triangle T} e^{A(\triangle T - \tau)}BU_{new}d\tau \tag{4.22}$$

$$= \underbrace{\left(e^{A\triangle T}\right)}_{\mathbf{V(\triangle T)}} X_{lin}(0) + \underbrace{\left(\int_0^{\triangle T} e^{A(\triangle T-\tau)}B\phi(\tau/(\triangle T))d\tau\right)}_{\mathbf{W(\triangle T)}} \begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix} \tag{4.23}$$

$$= V(\triangle T)X_{lin}(0) + [W(\triangle T)\ 0_{3\times 7}]\,C \tag{4.24}$$

The equation, $\phi(\tau/(\triangle T))$, is defined in Eq. 4.17. The predicted value at time, $t = 2\triangle T$, is almost the same:

$$\begin{aligned} X_{lin}(2\triangle T) &= V(\triangle T)X_{lin}(\triangle T) + [0_{3\times 1}\ W(\triangle T)\ 0_{3\times 6}]\,C \\ &= V^2(\triangle T)X_{lin}(0) \\ &\quad + ([V(\triangle T)W(\triangle T)\ 0_{3\times 7}] + [0_{3\times 1}\ W(\triangle T)\ 0_{3\times 6}])\,C \end{aligned} \tag{4.25}$$

One can see that a pattern is formed. By integrating and shifting the forced response matrix, $W(\triangle T)$, the terminal state at the end, $8\triangle T$, can be expressed as the following:

$$\begin{aligned} X_{lin}(T) = X_{lin}(8\triangle T) &= V^8(\triangle T)X_{lin}(0) \\ &\quad + \left(\sum_{i=0}^{7}\left[0_{3\times i}\ V^{7-i}(\triangle T)W(\triangle T)\ 0_{3\times(7-i)}\right]\right)C \end{aligned} \tag{4.26}$$

The terminal state prediction equation is thus

$$X_{lin}(T) = V_n(T)X_{lin}(0) + W_n(T)C \tag{4.27}$$

$$V_n(T) = V^8(\triangle T) \tag{4.28}$$

$$W_n(T) = \sum_{i=0}^{7}\left[0_{3\times i}\ V^{7-i}(\triangle T)W(\triangle T)\ 0_{3\times(7-i)}\right] \tag{4.29}$$

4.2.4 Convergence

Fundamentally, terminal ILC is an integrator for the past terminal errors. The algorithm updates the control coefficient through the following update law:

$$C_{k+1} = C_k + L * (X_d - X_k) \tag{4.30}$$

where subscript k is the iteration index, L is the learning gain, and $X_d - X_k$ denotes the terminal error from iteration k. The convergence of the algorithm is based on the evolution of the error. If from one iteration to the next iteration the error decreases, then eventually the terminal error reduces to zero. The following error dynamic equations show that under ideal conditions the error will reduce monotonically provided that the learning gain is selected such that the eigenvalue of matrix, $(I - W_n(T)L)$, is less than unity. In this thesis, the learning matrix, L, is computed as the pseudo-inverse of $W_n(T)$.

$$\begin{aligned}
\epsilon_{k+1} &= X_d - X_{k+1}(T) \\
&= X_d - V_n(T)X(0) - W_n(T)C_{k+1} \\
&= X_d - V_n(T)X(0) - W_n(T)(C_k + L\epsilon_k) \\
&= (X_d - V_n(t)X(0) - W_n(T)C_k) - W_n(T)L\epsilon_k \\
&= (X_d - X_k(T)) - W_n(T)L\epsilon_k \\
&= \epsilon_k - W_n(T)L\epsilon_k \\
\epsilon_{k+1} &= (I - W_n(T)L)\epsilon_k
\end{aligned} \tag{4.31}$$

Because terminal ILC is a linear methodology, its search direction does not exclude the area outside the dynamic constraints. So in manipulating U_{new} into achieving the terminal constraint, the EMV control coefficient often violates the voltage saturation (i.e., Eq. 4.16) bound or the current saturation bound. Next, an attempt is made to alleviate this problem by applying the nullspace gradient projection method developed by Oriolo [62].

Nullspace Gradient Projection

The main goal of nullspace gradient projection is to steer the coefficients away from the constraints while allowing ILC to continue reducing the terminal error. In this method, the nullspace matrix, $I - (W_n^{\dagger}W_n)$, is used for projection of the correction coefficient. The original terminal ILC controller (Eq. 4.30) is modified to:

$$C_{k+1} = C_k + L\epsilon_k + (-\alpha)(I - W_n^{\dagger}W_n)\nabla H_c \tag{4.32}$$

where $W_n^\dagger = W_n^T(W_n W_n^T)^{-1}$ and H_c is a cost function in which we penalize the control for reaching close to the boundary and α is a scalar step size in the gradient direction. For example, in order to limit the input voltage with ± 42 volts, we can select the cost function as

$$H_k = \max_{t\epsilon[0,T]} \left(\frac{|U_k(t)|}{35} \right)^{2p} + \epsilon_k^T Q \epsilon_k \tag{4.33}$$

where the matrix, Q, and the scalar variable, p, shifts the cost function weights to either the terminal errors or the constraints. The nullspace projection, in theory, does not alter the error convergence:

$$\begin{aligned} \epsilon_k &= X_d - V_n(T)X(0) - W_n(T)C_{k+1} \\ &= X_d - V_n(T)X(0) - W_n(T)\left(C_k + L\epsilon_k + (-\alpha)(I - W_n^\dagger W_n)\nabla H_c\right) \\ &= (I - LW_n(T))\epsilon_k + (-\alpha) * (W_n - W_n W_n^\dagger W_n)\nabla H_c \end{aligned} \tag{4.34}$$

Since we know that the matrix $W^\dagger$ is pseudo-inverse of the W_n matrix, the term $W_n - W_n W_n^\dagger W$ goes to zero:

$$\begin{aligned} (W_n - W_n W_n^\dagger W_n) &= W_n - \left(W_n W_n^T (W_n W_n^T)^{-1}\right) W_n \\ &= W_n - I * W_n = 0 \end{aligned} \tag{4.35}$$

Unfortunately, there are a few undesirable aspects to the nullspace gradient projection method. First of all, the derivative of the cost function, ∇H_c, is not known ahead of time. Computing numerical derivatives will significantly lengthen the convergence time. Secondly, the step size needed for descent must be found by trial and error. Oriolo recommended performing a line search via the Armijo's rule ([63]) but doing so also introduces long delays caused by the necessary functional evaluations. Most importantly, the projection of cost function gradient onto the null-space of input matrix W_n severely restricts the amount of adjustment possible to the control coefficient.

4.2.5 Simulation results

The simulation for terminal iterative learning control reveals that despite being prone to saturation, it is capable of reducing terminal error very quickly (usually under 50 iterations). When the initial condition is suitable, Terminal ILC can converge without saturation. In Figure 4.7, the pressure disturbance is set to be at 0.25 bar and the conditions at the end of approach control are initially not at the desired values. At the 50th iteration, the Terminal ILC controller is activated, and it takes less than 20 iterations for the Terminal ILC controller to regulate the terminal states back to the setpoint. Adding measurement noise into the system does not change the result (see Figure 4.8). Output from the controller, in this case, produces a reasonable voltage and current profile (see Figure 4.9). In addition, the voltage constraint can be visualized in the linearized input dF_{may}/dt domain. Figure 4.10 shows that the simple ± 42 input voltage constraint can no longer be presented as a straight line in the domain of magnetic force time derivative. The significance of Figure 4.10 is that feedback linearization complicates the handling of constraints.

In Figure 4.11 and 4.12, a different set of initial controller coefficient are used, and this time the controller must overcome a 0.25 bar pressure drop. While the convergence occurs as shown in Figure 4.11, the generated controller output results in input voltage saturation (42 V) and current saturation (35 A) in Figure 4.12. Logically, the pressure drop should cause the controller to lower the voltage. However, the simple feedback linearized integrator structure of the Terminal ILC can not incorporate any important insight.

Attempts to use nullspace gradient projection to complement the Terminal ILC causes changes of the controller output. Despite the significant difference in the controller output compared with the simple Terminal ILC output (Figure 4.13), voltage and current saturation still occurrs (Figure 4.14). Even though coefficient movement is reduced by the nullspace gradient projection shown in Figure 4.13, it still exceeds the system voltage and current constraints.

4.2.6 Discussion of Terminal ILC approach controller

Unfortunately, the attempt to avoid saturation by applying nullspace gradient projection did not yield any improvement for this system. While there is still no effective way to bound the input coefficients, terminal iterative learning control has demonstrated rapid setpoint regulation for many cases (with or without saturation). It is possible to try the simulation

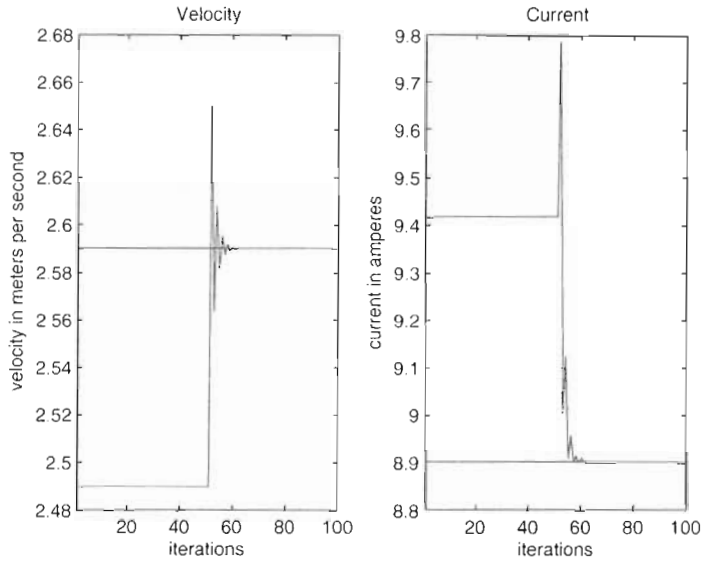

Figure 4.7: Simulated v^f_{appr} and i^f_{appr} versus iterations under Terminal ILC control. The controller is activated at 50th iteration and operates under no measurement noise.

cases that does not end in saturation in an actual experiment, especially for smaller error signals. Finally, the Terminal ILC algorithm is not computationally intensive (a simple lookup table can be prepared for computing the learning gain) and is thus suitable for realtime operations. Therefore, Terminal ILC was tested out in the test bench (without the nullspace correction, because the simulation result with nullspace correction is not favorable), and the results can be found in the experimental section of this thesis.

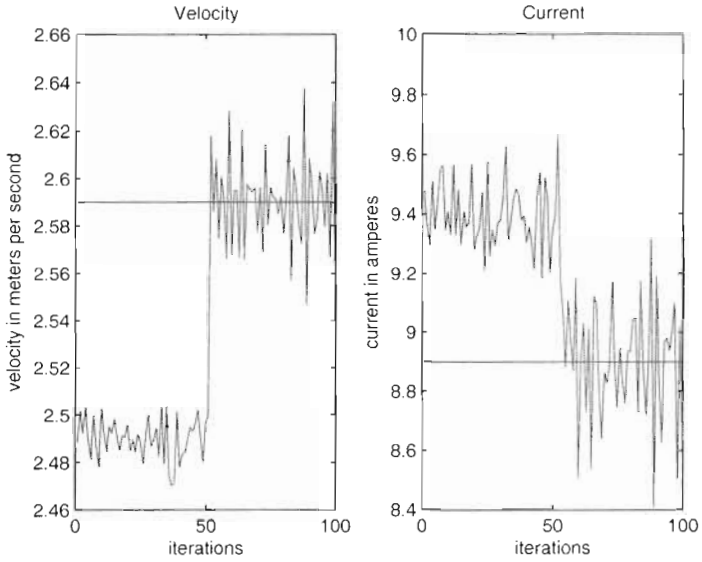

Figure 4.8: Simulated v^f_{appr} and i^f_{appr} versus iterations under Terminal ILC control. The controller is activated at 50th iteration and operates under measurement noise.

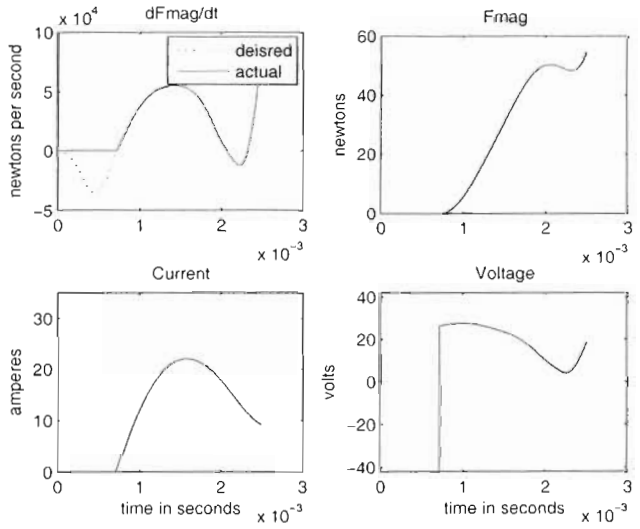

Figure 4.9: Coil current, input voltage, magnetic force from simulation for Figure 4.8.

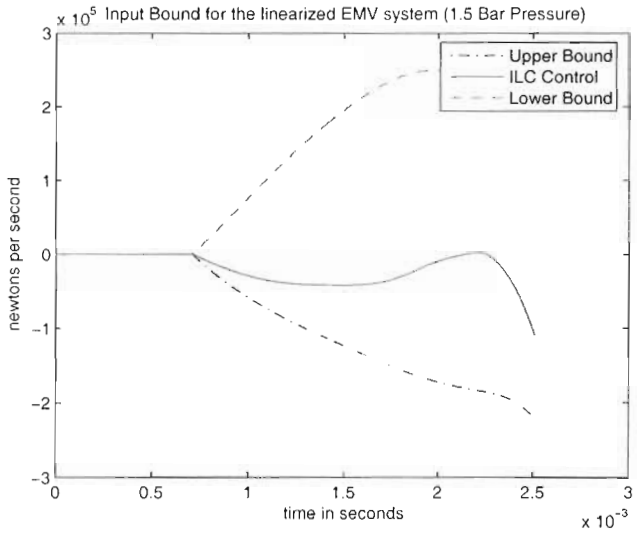

Figure 4.10: An illustration of feedback-linearization complicating the handling of constraints. The ± 42 V input voltage limitation in terms of the linearized input $\frac{dF}{dt}$.

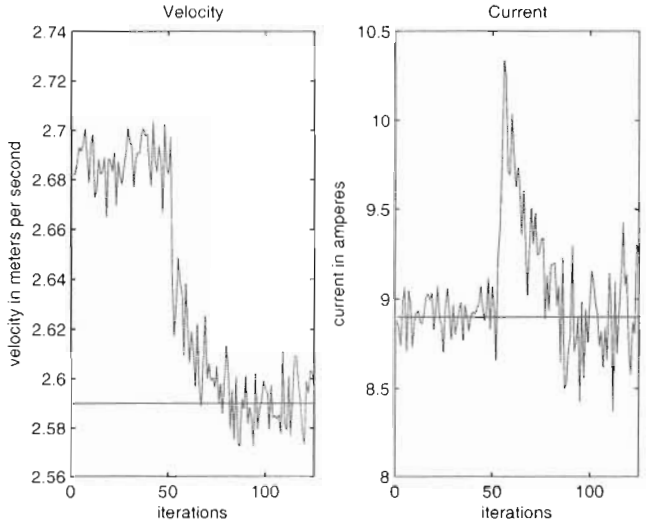

Figure 4.11: An illustration of Terminal ILC controller converging against input saturation and measurement noise. Simulated v^f_{appr} and i^f_{appr} versus iterations under Terminal ILC control.

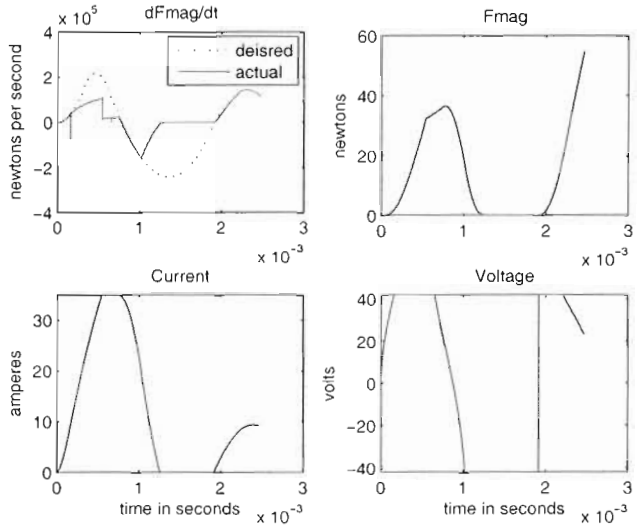

Figure 4.12: Current, voltage, magnetic force from Figure 4.11's simulation.

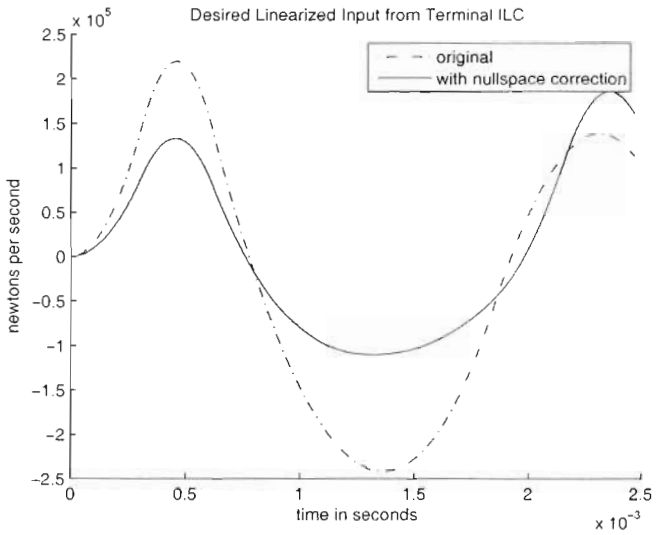

Figure 4.13: Terminal ILC controller output with and without nullspace gradient projection.

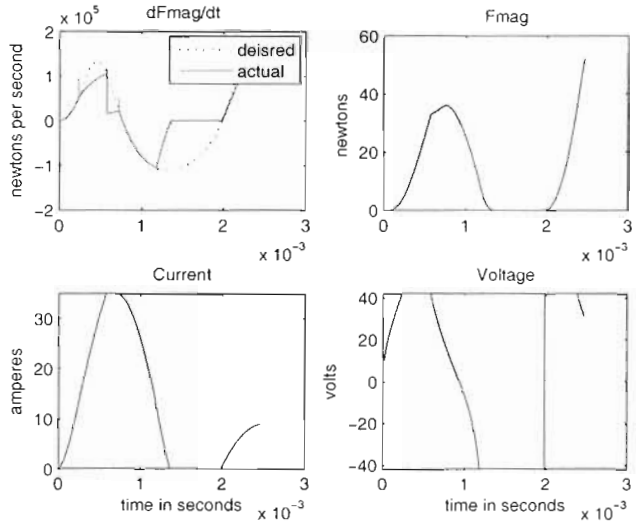

Figure 4.14: Current, voltage, magnetic force profile from a converged Terminal ILC controller with null-space gradient projection. The null-space correction fails to reduce input and state saturation in any meaningful amount.

4.3 Nelder Mead Simplex Algorithm

A different way to solve the EMV approach control problem is to apply the nonlinear programming methodology. A general nonlinear programming problem is of the form:

$$\begin{array}{lll} \text{minimize} & F(x) & \text{with respect to x} \\ \text{subject to} & g_i(x) = 0 & \text{for i=1,...}m_1\text{, } m_1 > 0 \\ \text{subject to} & h_j(x) >= 0 & \text{for j=}m_1\text{+1,...m, } m > m_1 \end{array}$$

In the case of EMV control, the cost function, $F(x)$, is a quadratic function of the terminal errors weighted by scalar factors, α and β:

$$F(c) = (\alpha(v^f_{appr} - v_d)^2 + \beta(i^f_{appr} - i_d)^2) \tag{4.36}$$

The voltage and current limitations contribute to the inequality constraints, $h_j(x)$. The upper bound coefficient, c_{ub}, interpolates the current profile produced under constant maximum voltage; c_{lb} interpolates the lowest current profile that still provides enough energy for landing:

$$h_1(c) = c_{ub} - c > 0 \quad \text{and} \quad h_2(c) = c - c_{lb} > 0 \tag{4.37}$$

The Nelder Mead Simplex Method (a.k.a downhill simplex) belongs to a branch of nonlinear programming algorithms called direct search, meaning that it does not need any derivative information such as gradient or hessian. By considering the rank of the objective values associated with a group of vertices, it determines the descent direction and step size. As it lowers the cost function, the entire group of vertices (simplex) moves and contracts, and eventually converges to a local minimum. To further understand the Nelder Mead algorithm, refer to [64].

Other derivative-free methods can be used for optimization (for example, pattern search, Rosenbrock's method, Powell's method, etc). For a detailed discussion, see the survey paper [65]) by Lewis, Torczon, and Trosset. Nelder Mead is selected here because it is easy to understand and computationally efficient to implement.

4.3.1 Position-based current input interpolation

Like the Terminal ILC, the Nelder Mead controller also uses B-spline interpolation to reduce the variables that need to be solved. Unlike the Terminal ILC's time-based interpolation,

however, the input interpolation of the Nelder Mead controller is position-based. Compared to the armature travel duration, which varies between iterations, the armature travel length is fixed to 8mm. Thus, a position based interpolation is a better solution. As stated in the section 4.2.3, Terminal ILC requires time-based interpolation because it needs to predict the terminal state. Note that before -3mm, the current profile is clamped to zero to limit the risk of current saturation damaging the experimental equipment. To be comparable to the experimental results, simulations follow this rule as well.

The other distinction is that the input coefficients in the Nelder Mead controller change the current profile without going through feedback linearization. The current tracking is done by using a simple on-off voltage controller. While foregoing feedback linearization eliminates the effect of modeling errors, tracking current this way is imprecise because of the ± 42 voltage constraint and system latency. Fortunately, the errors should be acceptable because Nelder Mead is a model free method. As long as a clear cause and effect relationship between input and output exists, the Nelder Mead controller can decrease the cost function to a local minimum.

4.3.2 Determining search direction and step size

The Nelder Mead algorithm involves five steps: order, reflect, expand, contract (inside/outside) and shrink. These five steps determine the search direction and the step size. In the next paragraph a brief description of the algorithm is provided. The detailed algorithm is presented in the following subsection.

The Nelder Mead algorithm uses *non-degenerate simplex*. The definition of simplex is a set of $n + 1$ points in n dimension. So if the input current profile is interpolated by n coefficients, then the simplex should have $n + 1$ different current profiles. A simplex is non-degenerate if the n vectors connecting any single vertex to the remaining vertices spans the entire space.

To find a search direction, the simplex is reordered to separate out the *worst* vertex. Then the *worst* vertex is moved in the general direction of the remaining vertices (represented by the average of the rest of the vertices). This newly moved point is called the reflection point. If the function evaluation at the reflection point is favorable, then we can move the point further in that direction for potential improvement (expansion). If the trial result is not favorable, then the next trial point can be moved closer to the worst point from average (*contract in*) or move toward the average from the reflection point (*contract out*).

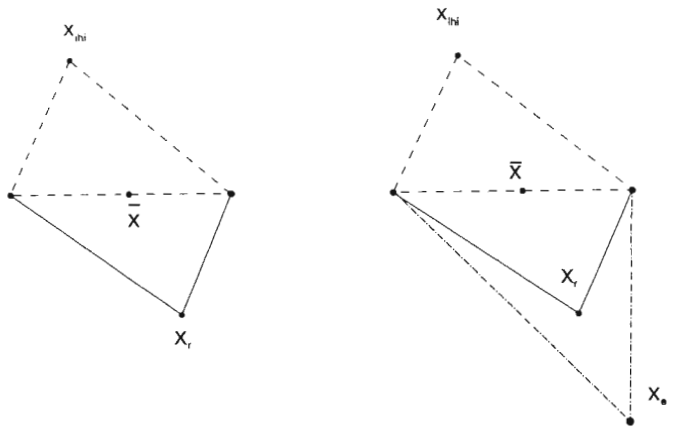

Figure 4.15: Nelder Mead algorithm steps: reflection (left) and expansion (right).

When all else fails, the vertex itself can be shrunk toward its *best* vertex. The goal is to eventually replace the *best* point in the simplex.

One iteration of the Nelder Mead algorithm

The detailed Nelder Mead algorithm is presented in this subsection. First the tuning parameters are introduced. ρ, χ, γ, and σ are reflection, expansion, contraction, and shrinkage coefficients. They determine the aggressiveness/conservativeness of the controller. They are constrained by the following rules:

$$\rho > 0,\ \chi > 1,\ \chi > \rho,\ 0 < \gamma < 1,\ \text{and}\ 0 < \sigma < 1$$

In this section's simulation work, these coefficients are defined as the following:

$$\rho = 1,\ \chi = 2,\ \gamma = 1/2,\ \text{and}\ \sigma = \frac{1}{2}$$

Next, the five steps are presented in detail. An iteration of the algorithm will start with *order* and *reflect*. Depending on the trial result of the reflection point, the algorithm may go to one of the expansion, contraction or shrinkage steps.

1. **Order**

   To start, $n + 1$ vertices are functionally evaluated to separate out the best, the worst,

and the second worst vertices.

$$f(x_1) \leq f(x_2) \leq f(x_3) \ldots \leq f(x_{n+1})$$

best vertex	$x_{ilo} = x_1$
second worst vertex	$x_{inhi} = x_n$
worst vertex	$x_{ihi} = x_{n+1}$
average vertex (exclude worst)	$\bar{x} = \sum_{i=1}^{n} \frac{x_i}{n}$

2. **Reflect**

   Then the worst point is reflected toward the average of the remaining vertices.

$$x_r = \bar{x} + \rho(\bar{x} - x_{ihi}) = (1 + \rho)\bar{x} - \rho x_{ihi} \tag{4.38}$$

   if $f(x_r) < f(x_{ilo})$ then go to expansion step
   else if $f(x_r) < f(x_{inhi})$ then go to replace x_{ihi} with x_r, terminate the iteration.
   else if $f(x_r) > f(x_{ihi})$ then go to the contraction step.
   An illustration of *reflection* is shown in the left plot of Figure 4.15.

3. **Expand**

   Then the expansion point is computed based on the coefficient , χ:

$$x_e = \bar{x} + \chi(x_r - \bar{x}) = \bar{x} + \rho\chi(\bar{x} - x_{ihi}) = (1 + \rho\chi)\bar{x} - \rho\chi x_{ihi} \tag{4.39}$$

   If $f(x_e) < f(x_r)$, then replace the x_{ihi} with x_e; otherwise, replace x_{ihi} with x_r and terminate the iteration. An illustration of *expansion* is shown at the left plot of Figure 4.15.

4. **Contract**

   If $f(x_{inhi}) \leq f(x_r) < f(x_{ihi})$, then go to contract outside step; otherwise, go to contract side step.

   (a) *contract inside*

$$x_{cc} = \bar{x} - \gamma(\bar{x} - x_{ihi}) = (1 - \gamma)\bar{x} + \gamma x_{ihi} \tag{4.40}$$

   If $f(x_{cc}) < f(x_{ihi})$, then replace x_{ihi} with x_{cc} in the simplex and terminate the current iteration, else go to the shrink step. An illustration of *contracting inside* is shown at the middle plot of Figure 4.16.

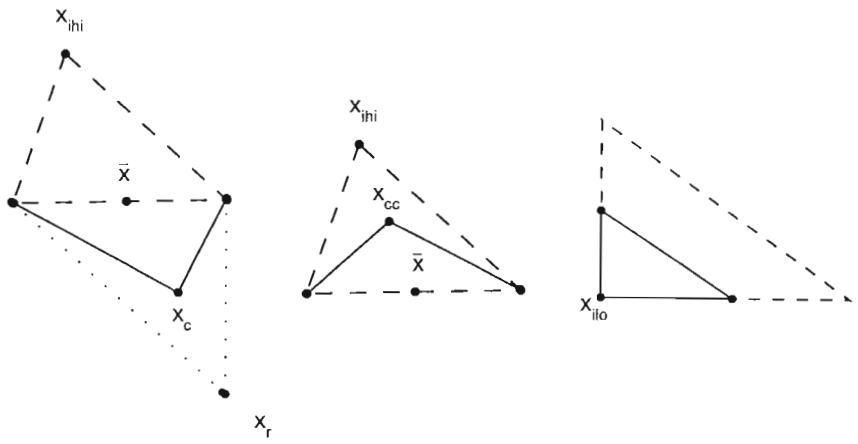

Figure 4.16: Nelder Mead algorithm steps: contraction outside (left), contraction inside (middle), and shrinkage (right).

(b) *contract outside*

$$x_c = \bar{x} + \gamma(x_r - \bar{x}) = \bar{x} + \gamma\rho(\bar{x} - x_{ihi}) = (1 + \rho\gamma)\bar{x} - \rho\gamma x_{ihi} \tag{4.41}$$

If $f(x_c) < f(x_{ihi})$, then replace x_{ihi} with x_c in the simplex and terminate the current iteration, else go to the shrink step. An illustration of *contracting outside* is shown at the left side plot of Figure 4.16.

5. **Shrink**

Evaluate f at the n points

$$v_i = x_{ilo} + \sigma(x_i - x_{ilo}) \tag{4.42}$$

Terminate the iteration. The unordered vertices of the simplex in the next iteration will consist of $x_{ilo}, v_2, v_3, \ldots, v_{n+1}$. An illustration of *shrinking* is shown in the right side plot of Figure 4.16.

Convergence occurs either when the cost function is reduced sufficiently or when the spacing between the simplex shrinks too much. If the cost reduction is unsatisfactory, even through the simplex has shrunk too much, then the algorithm is stuck at a local minimum. Under such a condition, re-establishing the simplex and resetting the algorithm may be beneficial.

4.3.3 Transforming input constraints

The original Nelder Mead algorithm is an unconstrained nonlinear programming algorithm. The typical solution for tuning constrained problems into non-constrained problems is to incorporate the constraints into the cost function (a.k.a barrier or penalty methods [63]). However, this approach is unnecessary because the EMV system only has simple input constraints. Instead of modifying the cost function, sine transformation can be applied to map the input coefficients into sine coordinates. The following explanation comes from John D'Errico's mfile [66] available on the Matlab®file exchange central website.

Suppose $x(i)$ is the original variable constrained by upper-bound, $UB(i)$, and lower-bound, $LB(i)$; one can solve optimization problem of $x(i)$ by solving the unconstrained optimization problem with $z(i)$, which is related with $x(i)$ through the following:

$$x(i) = LB(i) + (UB(i) - LB(i))\frac{(sin(z(i) + 1))}{2} \tag{4.43}$$

By inverting the above equation and checking the neccessary condition of arc-sine, we get an expression for $z(i)$

$$\begin{aligned} z_{temp}(i) &= \frac{2(x(i) - LB(i))}{UB(i) - LB(i)} - 1 \\ z(i) &= 2\pi + \arcsin\{\max(-1, \min(1, z_{temp}(i)))\} \end{aligned} \tag{4.44}$$

Since *sine* only varies between 0 and 1, modifying $z(i)$ will result in changing $x(i)$ within the bounds, $UB(i)$ and $LB(i)$. Note that in the case of the EMV control, those bounds are derived from operating/simulation experiences.

4.3.4 State machine implementation

The steps in section 4.3.2 belong to a typical optimization structure where an iteration of valve event is treated as a functional evaluation. In the realtime implementation on the dSpace card, the structure is different because the algorithm runs as an interrupt. At the end of the interrupt, the current profile for the next valve-swing must be computed. Table 4.3.4 illustrates how the state transition occurs based on the trial result of the previous iteration. Of equal importance, the table provides information on how the simplex vertex is updated and what the input for the next iteration is. With this table, one can easily code the algorithm in an interrupt setting.

Nelder Mead State Machine Transition Table				
current state	**if**	**replace simplex**	x_{trial}	**next state**[1]
reset	$i <= n+1$	$x_i = x_{trial}$, $i = i+1$	$x_{trial} = x_{init}, x_{trial}[i] = x_{init}[i] * \alpha$	reset
	else	$x_i = x_{trial}$,$i = 1$	$x_r = \bar{x} + \rho(\bar{x} - x_{ihi})$	reflect
reflect	$f(x_{trial}) < f(x_{ilo})$	-	$x_e = \bar{x} + \rho\chi(\bar{x} - x_{ihi})$	expand
	else if $f(x_{trial}) < f(x_{inhi})$	$x_{ihi} = x_{trial}$	$x_r = \bar{x} + \rho(\bar{x} - x_{ihi})$	reflect
	else if $f(x_{trial}) < f(x_{ihi})$	-	$x_c = \bar{x} + \gamma\rho(\bar{x} - x_{ihi})$	contract out
	else	-	$x_{cc} = \bar{x} - \gamma(\bar{x} - x_{ihi})$	contract in
expand	$f(x_{trial}) < f(x_r)$	$x_{ihi} = x_{trial}$	$x_r = \bar{x} + \rho(\bar{x} - x_{ihi})$	reflect
	else	$x_{ihi} = x_r$	$x_r = \bar{x} + \rho(\bar{x} - x_{ihi})$	reflect
contract out	$f(x_{trial}) = f(x_c) < f(x_r)$	$x_{ihi} = x_c$	$x_r = \bar{x} + \rho(\bar{x} - x_{ihi})$	reflect
	else	–	$x_s = x_{ilo} + \sigma(x_1 - x_{ilo})$	shrink
contract in	$f(x_{trial}) = f(x_{cc}) < f(x_{ihi})$	$x_{ihi} = x_{cc}$	$x_r = \bar{x} + \rho(\bar{x} - x_{ihi})$	reflect
	else	–	$x_s = x_{ilo} + \sigma(x_1 - x_{ilo})$	shrink
shrink	$i <= n+1$	$x_i = x_{trial}$, $i = i+1$	$x_s = x_{ilo} + \sigma(x_i - x_{ilo})$	shrink
	else	$x_{n+1} = x_{trial}$,$i = 1$	$x_r = \bar{x} + \rho(\bar{x} - x_{ihi})$	reflect

Table 4.1: Nelder Mead state machine

[1]If the next state is reflect, the vertices associated with minimum cost, x_{ilo}, maximum cost, x_{ihi}, and second maximum cost, x_{inhi}, must be identified. Subsequently, the reflection point, x_r, is calculated to be the next trial point, x_{trial}.

4.3.5 Convergence issue

The original paper [67] of the Nelder Mead algorithm is published in 1965. Since then, there has been thousands of references in scientific journals to it and various implementations of it in numerical packages. For example, both [68] and [66] have implementations of Nelder Mead. In fact, an entire book [69] in chemical engineering is devoted to its application. However, despite its conceptual simplicity and enduring popularity, there has been very little theoretical proof or analysis of it. The proof by Lagraias [64] is only R^1 while there exists a counter example by McKinnon [70] for C^2. In this thesis, the Nelder Mead algorithm is discussed in detail because it a useful solution to the EMV problem, both in simulation and experiment. Nevertheless, it important to keep in mind this theoretical deficiency.

4.3.6 Energy consumption

Being a cost function based algorithm, Nelder Mead can be fine tuned by changing the composition of its cost function. For example, if a current integral term is added to the cost function, then the algorithm will have to balance the cost for regulating the set-points, v^f_{appr} and i^f_{appr}, with the cost from energy expenditure. When properly weighted, the Nelder Mead controller can tune itself to search for minimum energy required to keep the controlled variables around the desired set-points.

$$\begin{aligned} \text{cost} = &(\frac{\text{velocity error}}{\text{velocity weight}})^2 + (\frac{\text{current error}}{\text{current weight}})^2 \\ &+\text{current integral weight} * \int i(t)dt \end{aligned} \tag{4.45}$$

4.3.7 Simulation result

In simulation, the Nelder Mead algorithm produces proper regulation of the terminal set-point under a reasonable time frame (100 steps of 40N of step disturbance). More importantly the resultant input coefficient stays within constraints. The first simulation test is to see what happens when the setpoint is changed. The result in Figure 4.17 shows that the setpoint is changed from [10A, 2.55m/s] to [12A, 2.61m/s] in 300 iterations. This result can be improved further speed-wise because the controller coefficients are set conservatively. The second and third examples are the step response against a 40N pressure increase and decrease (see Figure 4.18 and 4.19). It takes around 100 iterations for both velocity and current to converge back to the setpoint. The responses against a ramp disturbance of 40N

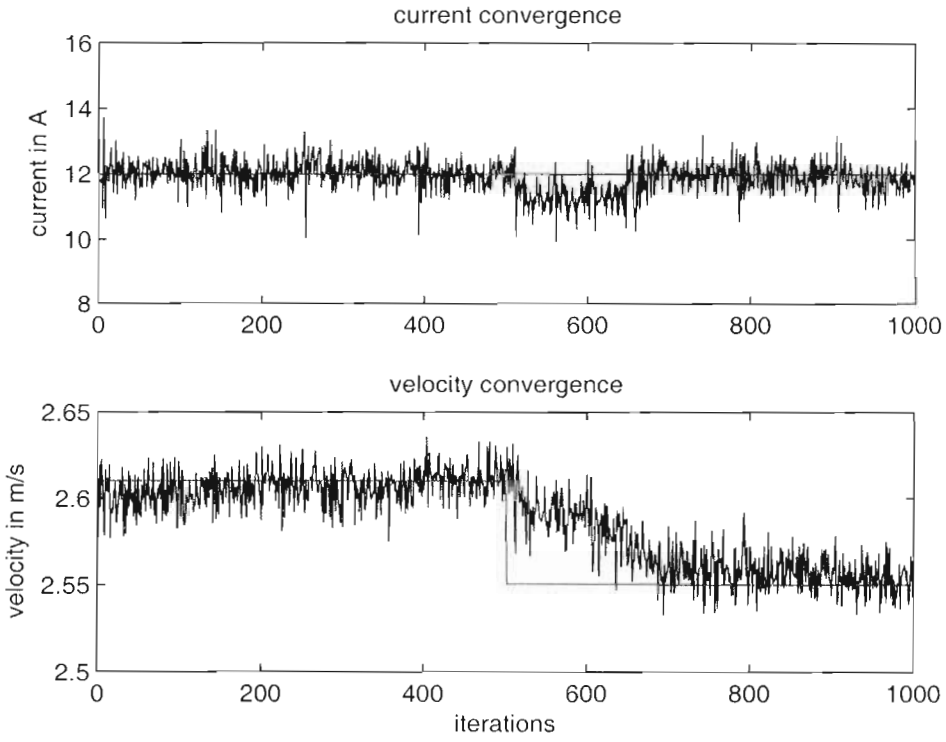

Figure 4.17: An illustration of the Nelder Mead controller's regulation against set-point change. Simulated v^f_{appr} and i^f_{appr} versus iterations. Set-point changes at the 500th iteration.

in 400 steps can be seen in Figure 4.20 and 4.21. The algorithm's performance against ramp disturbance is not as good because its simplex vertices quickly becomes obsolete with a ramp disturbance. Despite the larger transient (error of 0.05 m/s), the regulated result still returns to the setpoint. Inserting a current integral term into the cost function results in the current integral reduction from 16.5e-3 to 16.25e-3 ampere-second without changes in the terminal conditions (see Figure 4.22). Finally, Figure 4.23 illustrates the evolution of the input current in 600 iterations under the Nelder Mead controller. One can see that the command current is smooth even during the transient phase of the close-loop system. This result is much more practical and robust compared to the results from nonlinear and terminal iterative learning controllers.

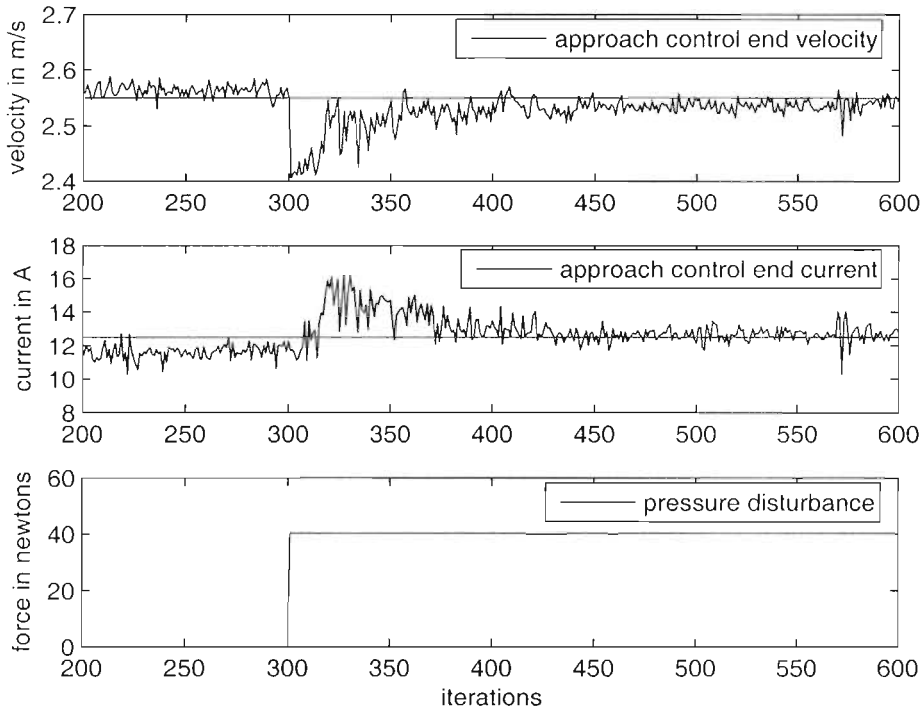

Figure 4.18: An illustration of the Nelder Mead controller's regulation against step pressure disturbance. The top and middle plot are simulated v_{appr}^{f} and i_{appr}^{f} versus iterations. The bottom plot shows a step disturbance pressure increase of 40N occuring at the 300th iteration.

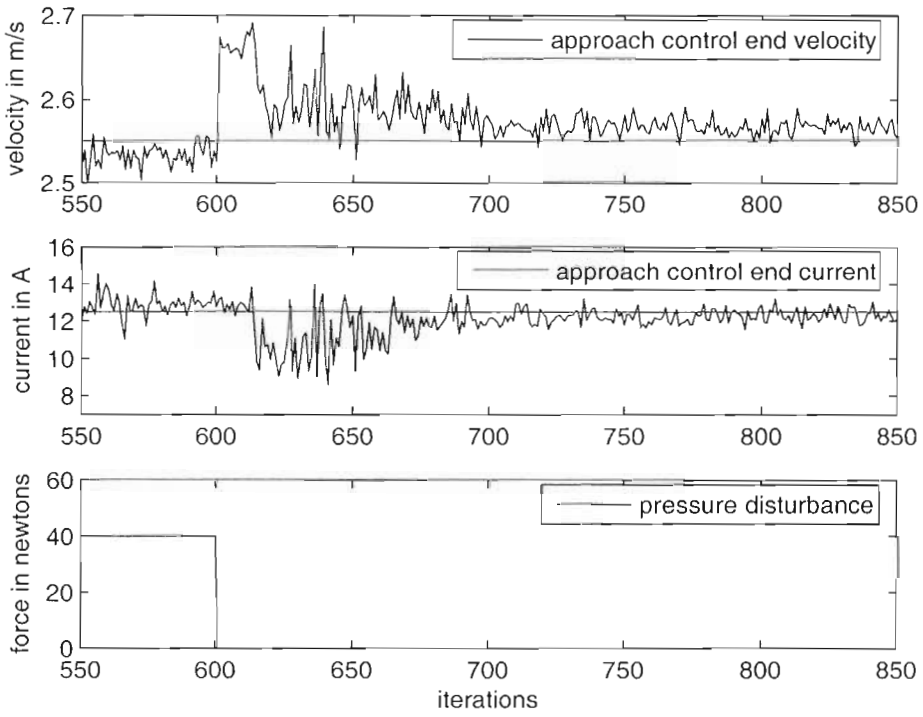

Figure 4.19: An illustration of the Nelder Mead controller's regulation against step pressure disturbance. The top and middle plot are simulated v^f_{appr} and i^f_{appr} versus iterations. The bottom plot shows a step disturbance pressure decrease of 40N occuring at the 600th iteration.

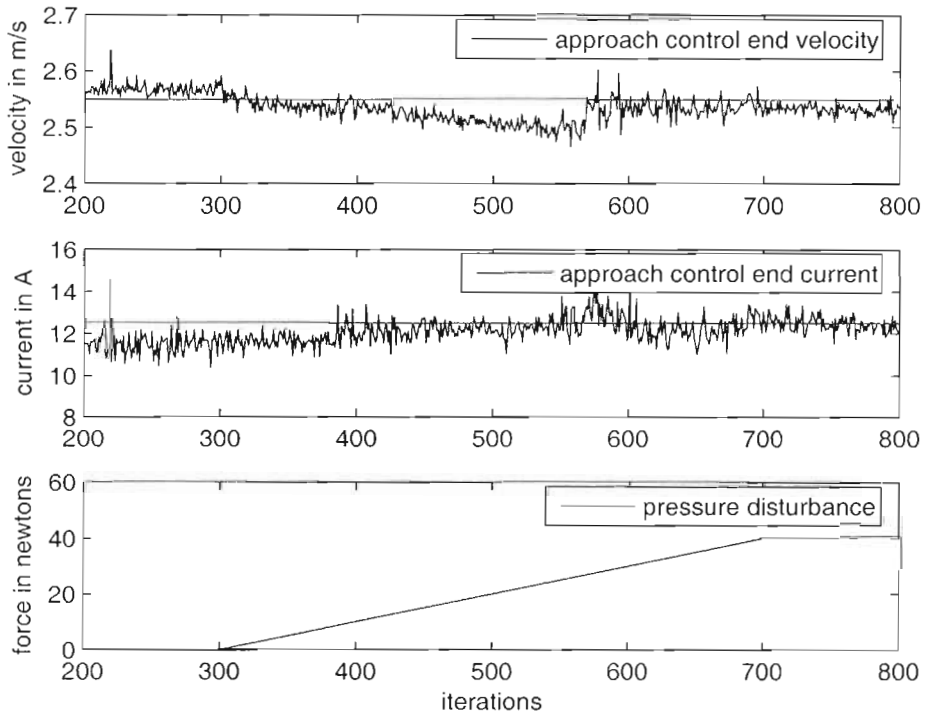

Figure 4.20: An illustration of the Nelder Mead controller's regulation against ramp pressure disturbance. The top and middle plot are simulated v^f_{apprr} and i^f_{appr} versus iterations. The bottom plot shows a ramp disturbance pressure at rate of 10N per 100 steps occuring at the 300th iteration.

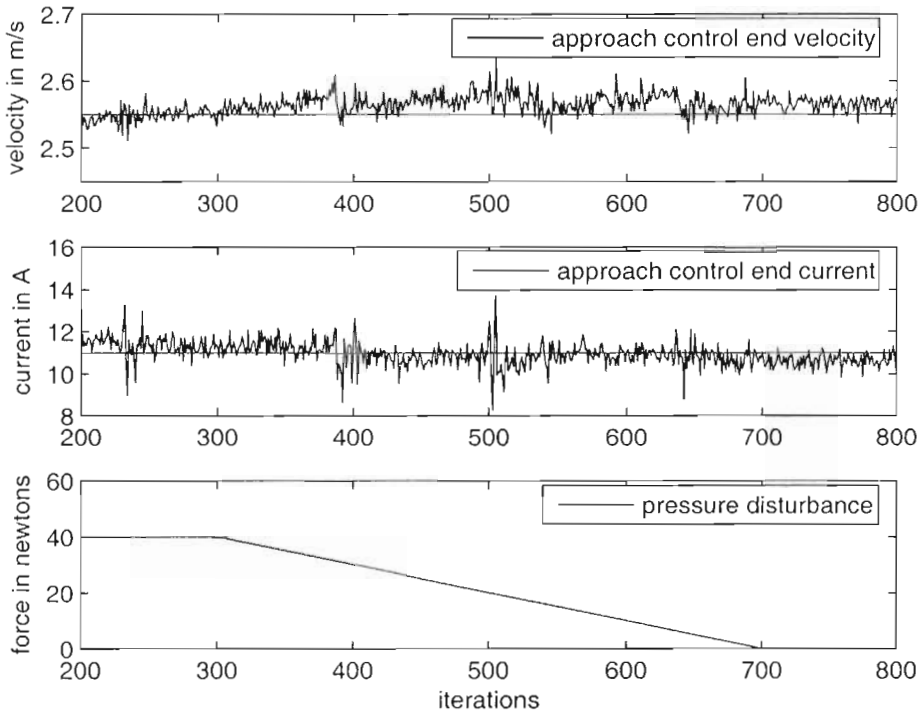

Figure 4.21: An illustration of the Nelder Mead controller's regulation against ramp pressure disturbance. The top and middle plot are simulated v^f_{appr} and i^f_{appr} versus iterations. The bottom plot shows a ramp disturbance pressure at rate of -10N per 100 steps occuring at the 300th iteration.

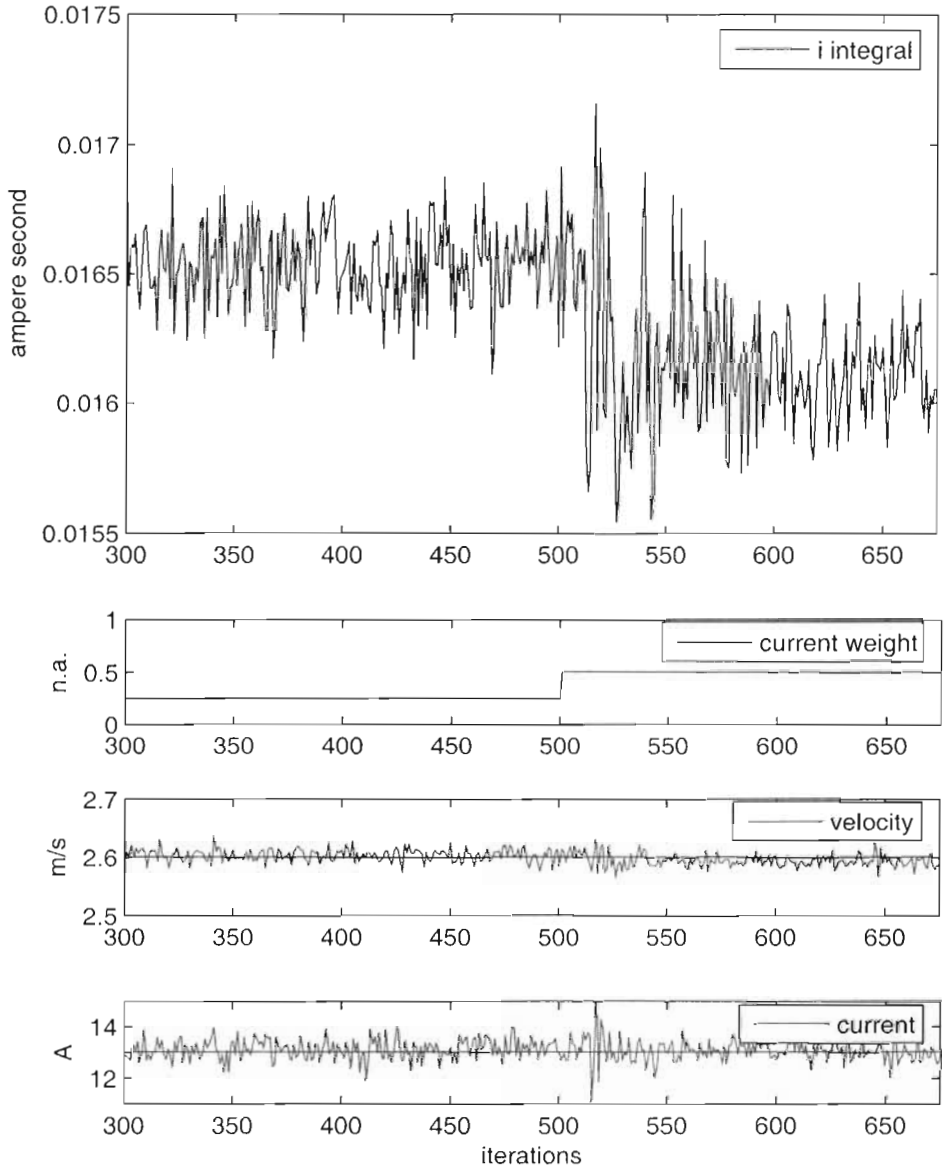

Figure 4.22: An illustration of energy reduction by adding a weighted current integral term to the cost function of the Nelder Mead controller. The top plot is current integral versus iterations. Second highest plot is the current integral weight in the controller's cost function versus iterations. The bottom two plots are simulated v^f_{appr} and i^f_{appr} versus iterations.

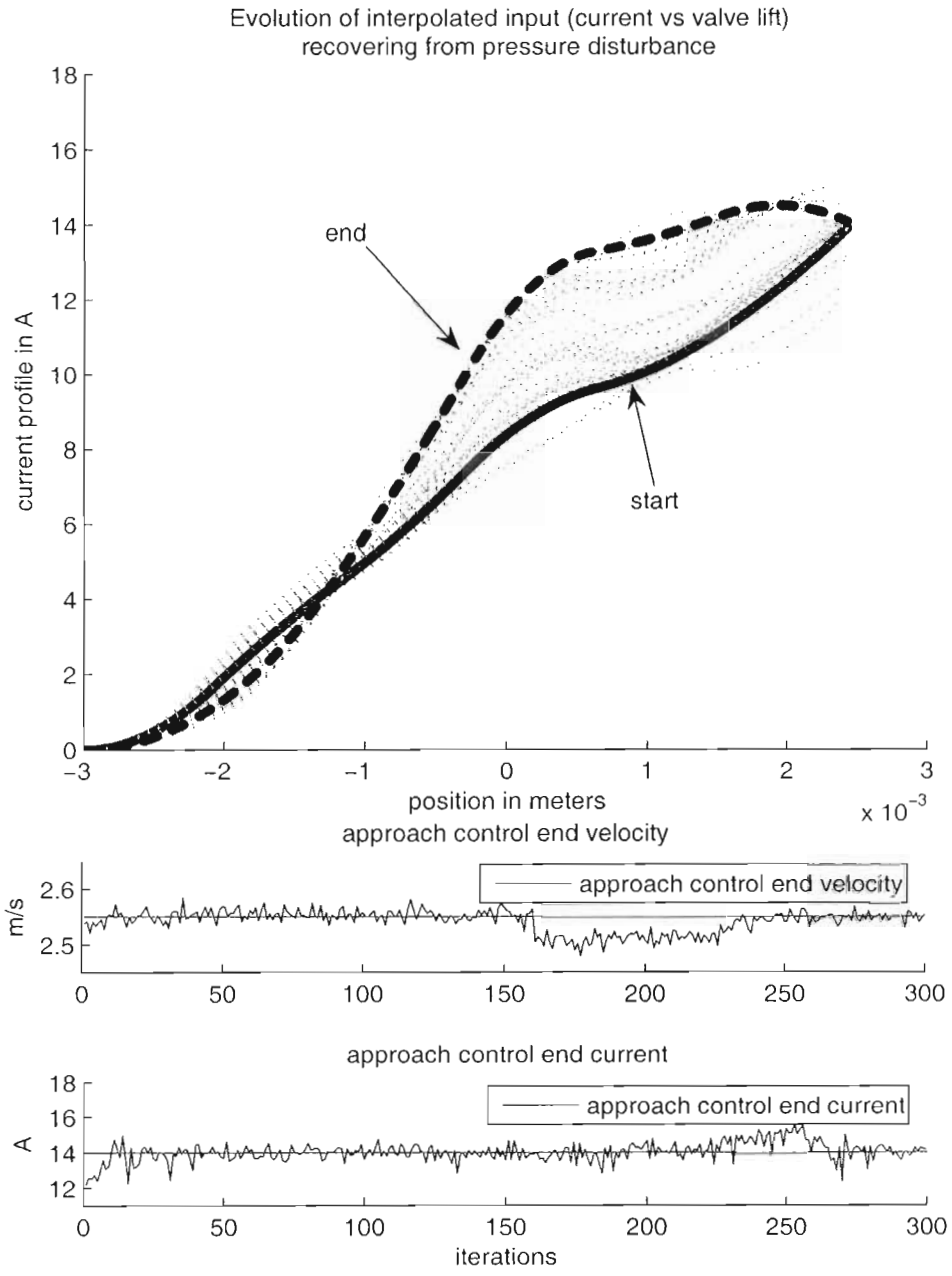

Figure 4.23: Input current profile evolution of the Nelder Mead controller regulating against step pressure increase. The top plot is the current profile versus position from iteration 1 to 300. The bottom two plots are simulated v^f_{appr} and i^f_{appr} versus iterations. Nelder-Mead controller acts between $x = -3mm$ to $x = 2.55mm$. The dip in velocity at 160th iteration is due to a step increase of disturbance force.

4.3.8 Discussion

Nelder Mead is a model-free and derivative-free method for unconstrained optimization. After transforming the input bounds, the Nelder Mead algorithm can be applied to optimize input coefficients for the minimum terminal error. Unlike Nonlinear ILC and Terminal ILC, the Nelder Mead controller is more robust because it is forced to stay within a safe bound. The simulation result shows that it can rectify a 40N step disturbance within 100 iterations and resist ramp disturbance at the rate of 10N per 100 iterations. It is remarkable that even though all its algorithmic steps are linear, this method can handle highly nonlinear systems such as the EMV actuator plant. Finally, the steps in the state machine require only linear combination, making Nelder Mead very suitable for real-time operation.

4.4 Summary

Initially, the investigation aimed to improve upon the ILC controller in [2] by applying nonlinear ILC convergence result from [57] for approach control. Unfortunately, the inability to find a realizable trajectory offsets the benefit of nonlinear ILC convergence. To avoid the need for tracking a trajectory and to ensure that terminal current is also controlled, the focus is shifted to the Terminal ILC strategy. While the Terminal ILC simulation provides speedy error convergence, it is also prone to input and state saturation. To avoid saturation, the Nelder Mead algorithm with constraint *sine* coordinate transformation is implemented.

The conclusion reached from the simulation result is that Nelder Mead is most suitable for further experimentation because its coefficient evolution is constricted, and there is much less chance of saturating the coil circuit and damaging the equipments. In addition, its ability to regulate setpoint against disturbance is adequate, and its state machine is also computationally simple enough for real time implementation. The Terminal ILC is the fastest converging algorithm, but it is also most prone to saturation. It makes sense to at least perform some experiments using the Terminal ILC controller for small disturbance cases where saturation is less likely. The nonlinear ILC, however, is not deemed to be suitable for experimentation due to the difficulty of finding a suitable trajectory for it to track. Table 4.4 summarizes the controller-specific properties determined in this chapter.

Table 4.2: Conclusion reached on the three approach control schemes

	Nonlinear ILC	Terminal ILC	Nelder Mead
Computational requirement	learning gain computation	learning gain computation required	state machine and some algebra
Terminal error	fairly slow	monotonic convergence	adequate convergence time
Trajectory tracking required	require realizable trajectory	terminal error driven	terminal error driven
Controller saturation	input usually saturate while tracking trajectory	saturation occurs very often	no saturation due to coefficient bounds
Steady state	steady state error occurs if input saturate	even if minor saturation occurs the terminal error can still be eliminated	error elimination can be achieved
Tested experimentally	No, due to lack of realizable trajectory	Yes, but only for small disturbance	Yes, acceptable for larger disturbances

Chapter 5

Experimental Setup

The control algorithm that showed promise using simulation are tested on a real actuator mounted on an experimental test-bed. In this chapter, the actuator test bench, measurement sensors, fail-safe devices, prototype board, and monitoring software are described in detail. The experiment setup described in this section is almost identical to [1]. The experiments of [1] and this work took place in the engine control laboratory in the University of Alberta's mechanical engineering building. The complete experimental setup can be seen in Figure 5.1.

5.1 The TEMIC Electromechanical Actuator

DaimlerChrysler AG donated a linear prototype solenoid actuator built by Telefunken Microelectronic GmbH to the UofA/SFU valve control research project. As shown in Figure 5.2 and 5.3, the actuator consists of two springs, two solenoid electromagnets, and a shaft connecting the steel armature in between the springs to the valve below. The characteristics of the actuators are:

- The springs are under the same compressive force so that the natural state of the armature (without any coil magnetic influence) is in the middle.
- The springs are tensioned so that the system's natural frequency is roughly 150Hz to handle maximum engine speeds of 5000 to 6000 rpm.
- The coil-wound electromagnets are used for releasing or landing.

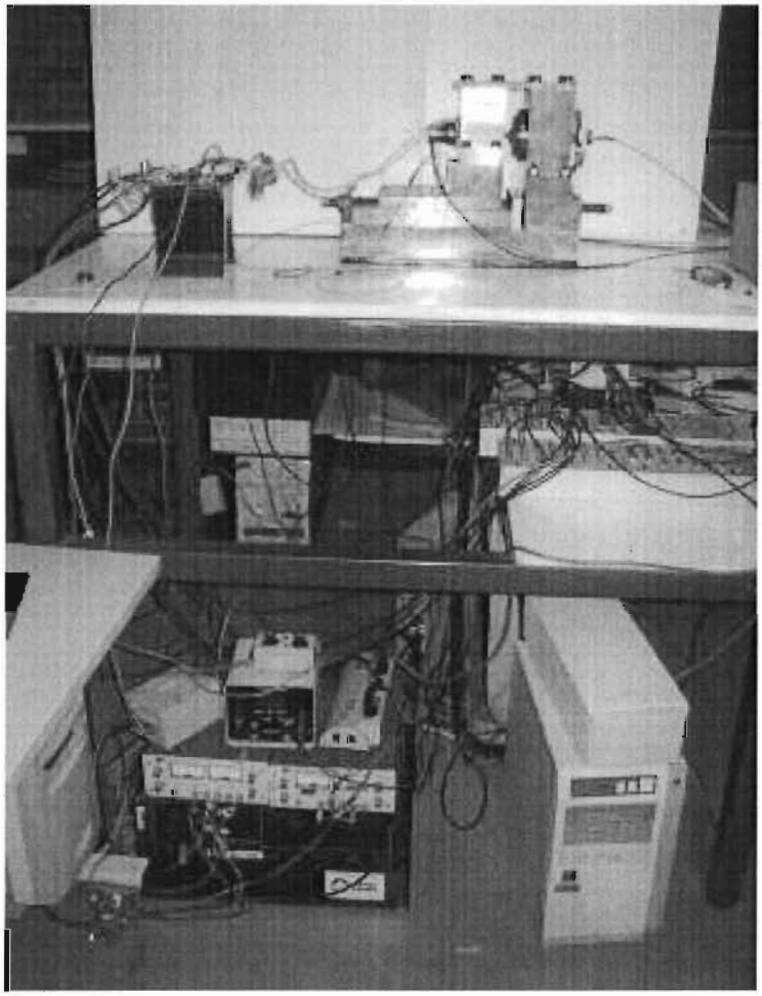

Figure 5.1: Complete actuator test-bench setup, including power supplies and power electronics.

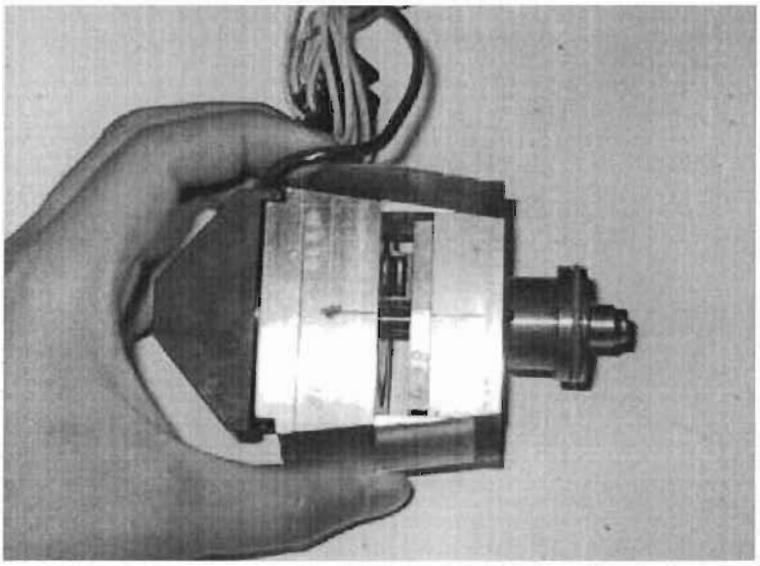

Figure 5.2: Closeup of the EMV actuator (without the springs).

- The closer coil consists of 79 turns and the opener coil consists of 72 turns. The flat pole and armature geometries are designed to provide large surface area and minimize the effect of fringing, at the cost of a nonlinear force vs distance relationship [71].
- Fine grain materials are used in the actuator construction to reduce eddy currents: sintered powder steel QStE500 is used in the housing of the actuator coils, while the coils themselves are made of Vacoflux silicon steel.
- Aluminium spacers are used between the two coils to ensure the flux flow passes through the steel armature only.

The test-bed is shown in Figure 5.4. The actuator and the valve are bolted down horizontally, and position sensors are placed at both ends of the assembly. The steel plate on top and the steel stage at the bottom clamps the valve actuator and valve assembly to restrict any movement other than in the horizontal direction. Underneath it all is a steel slab on top of neoprene rubber which is used to dampen vibration.

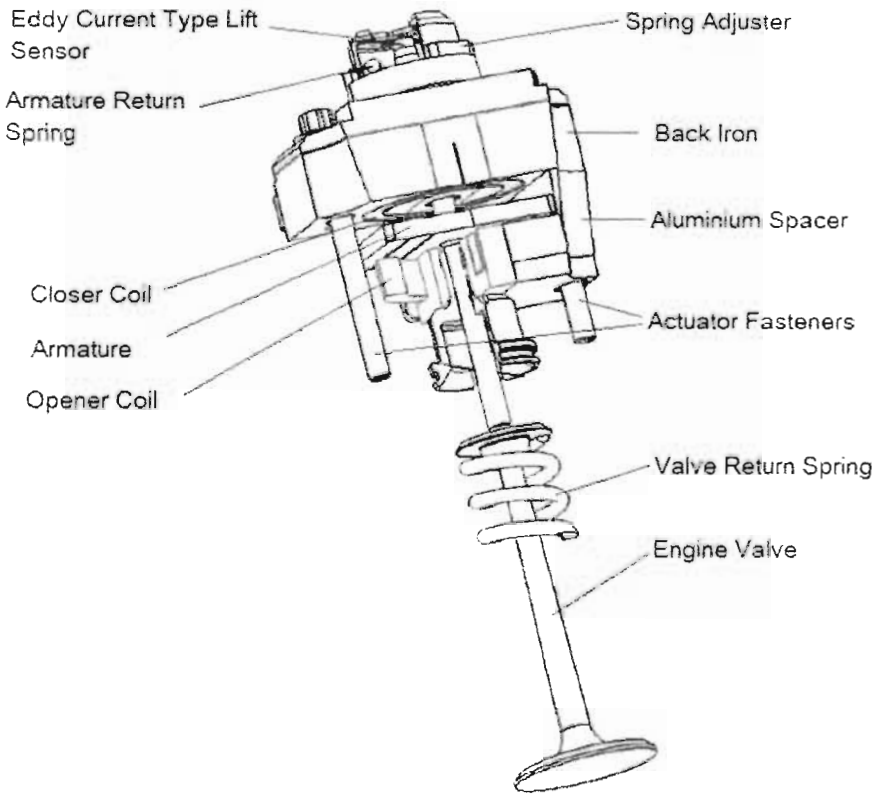

Figure 5.3: Cutaway of the EMV actuator, taken from [4] with permission of the author.

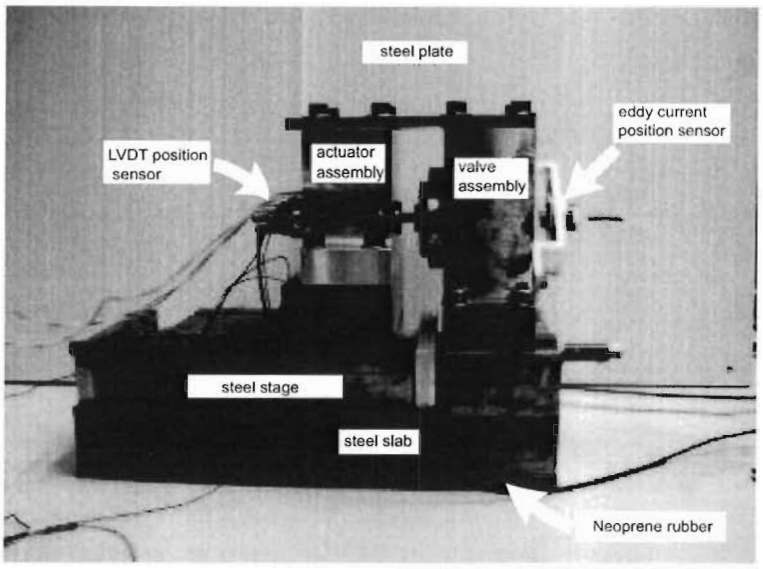

Figure 5.4: TEMIC actuator test-bed.

5.2 dSPACE DS1103 Board

The control algorithm is executed on the DS1103 control prototyping board from dSPACE corporation. The board contains two processors: an IBM Power PC 640e processor running at 400 MHz for the primary task and a Texas Instrument TMS320F240 DSP processor running at 20 MHz for handling I/O operations. Sixteen analog to digital convertors (**ADCs**) with either 12 or 16-bit channels and eight digital to analog converters (**DACs**) with 14-bit channels are available on the primary processor, while the I/O processor provides an additional sixteen 10-bit ADC input and four single-phase Pulse-Width Modulation (**PWM**) output channels. In addition, the board has its own expansion box (see Figure 5.5) that accommodates various I/O interfaces such as serial and parallel ports. Communication between the PC and the board passes through an industry standard architecture (ISA) card bus. The DS1103 board samples and executes commands at 50khz. In the case of the EMV approach control operation, the board reads the input voltage, current, and position measurements, computes the required command voltage, and then outputs the dedicated control signals to the respective power transistors located within the power electronics module.

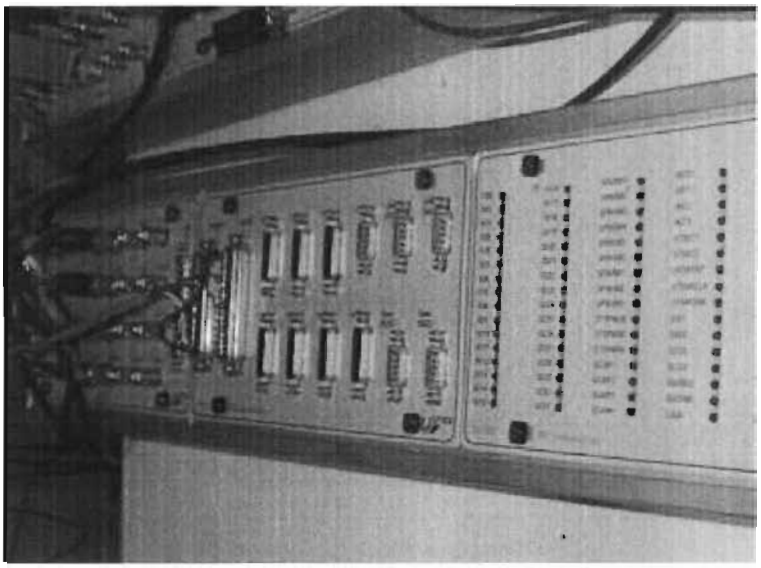

Figure 5.5: DS1103 board expansion box: i/o ports and LED panels.

5.2.1 ControlDesk Software

The communication between the DS1103 board and the host Windows PC relies on dSPACE proprietary software called ControlDesk. In addition to data monitoring and capturing, the user can compile controller code and tune controller parameters through ControlDesk. Figure 5.6 and 5.7 are the screen captures of the position and current plot utility, and the coefficient display/adjustment panel.

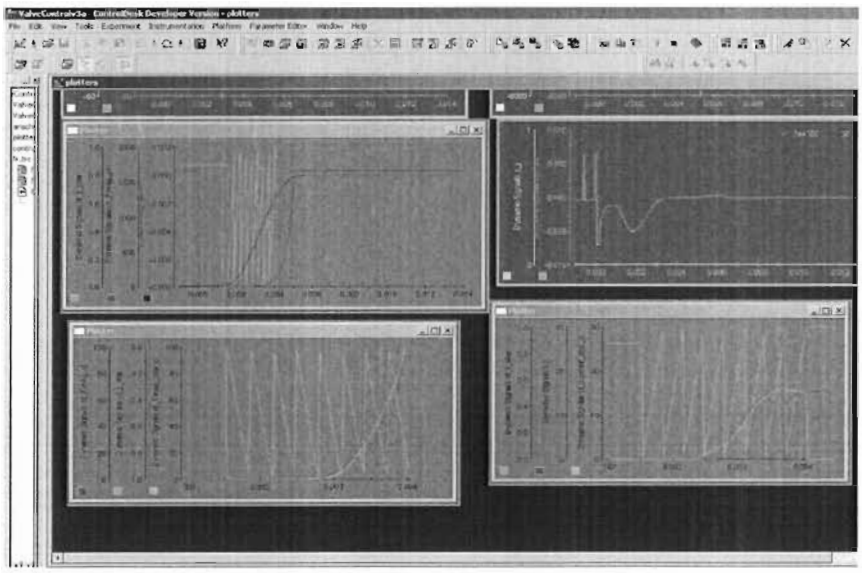

Figure 5.6: ControlDesk screen capture: plotting utilities.

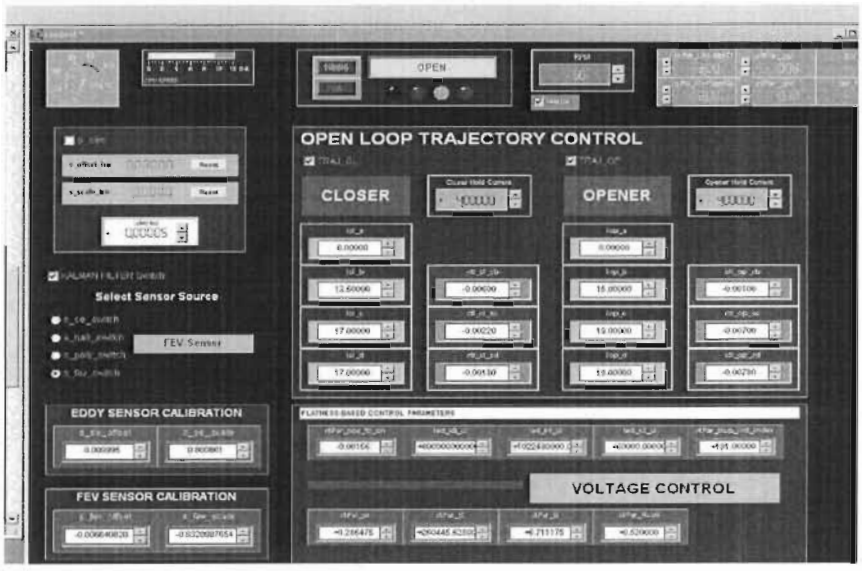

Figure 5.7: ControlDesk screen capture: control panels.

5.3 Power Management and Circuit Protection

5.3.1 Power electronics and power supply

The coil magnets on the EMV actuator are powered by two custom H-bridge circuit power electronics units developed by Bazooka Electronic Ltd. By switching two high speed insulated gate bipolar transistors (IGBT), three output power modes are available (0, 42, and (quasi)-42V) from the H-bridge circuit. The 42 volt limit is set in anticipation of an emerging voltage standard for future vehicles [72].

In Figure 5.8, the H-bridge controls the current flow by connecting two fly-back diodes and two IGBT transistors in the anti-parallel configuration. For +42 V mode, both transistor gate T1 and T2 are closed so that the potential across the actuator coil is 42 volts. For 0 V mode, transistor T2 is closed, but T1 is left open so the current circulates in a zero potential loop and slowly reduces to zero. The quasi -42 V mode requires both gate T1 and T2 to be open so that the potential difference across the actuator is -42 Volts, and the current is driven down to zero quickly. Once the current reaches zero, the reverse polarity can not be sustained and the voltage across the actuator is zero (thus, the name quasi -42 V mode). Note that transistor T1 can not be switched independently because it is powered by a subsidiary capacitor which only gets charged when T2 is closed. Consequently, to switch the power electronics between +42v to -42v requires going through 0 V mode; Figure 5.8 illustrates this process with transition arrows.

Shown in Figure 5.10, the two power supplies needed for the experiment are the 1 kilowatt Sorenson DCS60-18E, which drives the custom power electronics, and the Hewlett Packard 6236, which drives the power electronic circuitry and current measurement sensors. The Sorenson power supply is rated to provide supply voltages up to 60 volts direct current (**VDC**) at 18A, but according to [4], a short burst (less than 5ms) of 35A current is also possible. The Sorenson power supply is setup to supply 42 volts to power electronics which runs the EMV actuator. The Hewlett Packard 6236B triple output linear power supply is setup to provide ± 5 VDC for the over-current protection circuit and ± 15 VDC for the hall-effect sensors and opto-isolators.

Table 5.1: Switching transistors for power modes

voltage mode	transistor T1	transistor T2	effect on current
42	open	open	fast increase
0	closed	open	slow decrease
-42	closed	closed	fast decrease

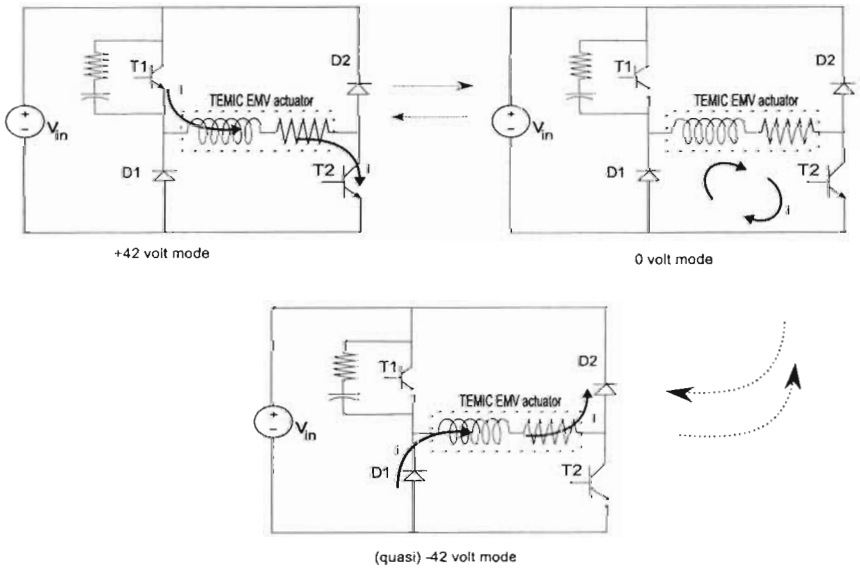

Figure 5.8: Power electronic output modes: 42,0, (quasi)-42.

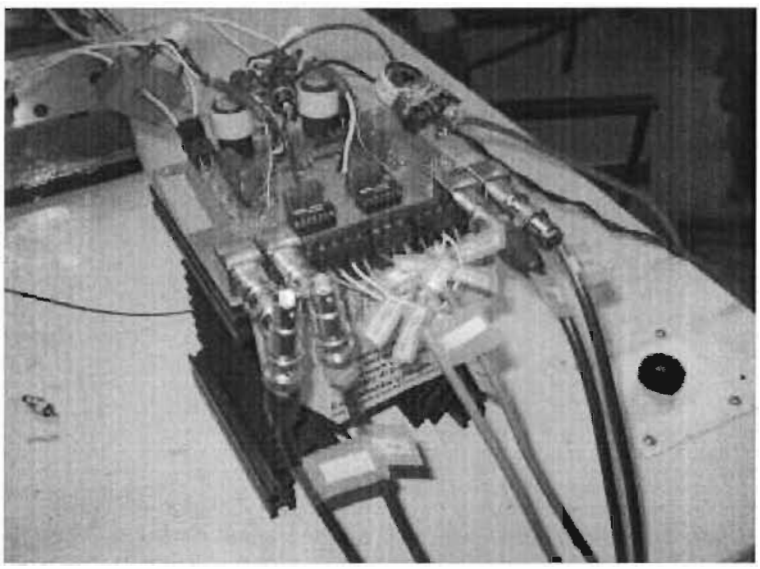

Figure 5.9: Power electronic circuit close up.

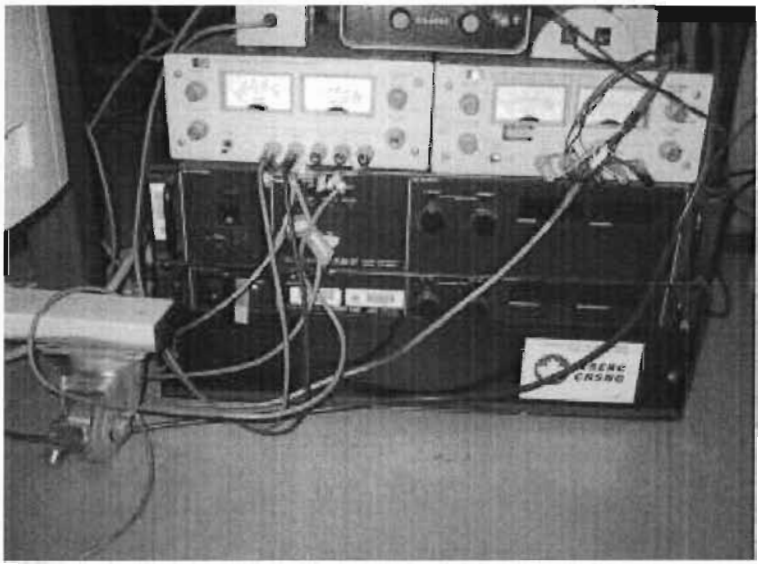

Figure 5.10: Sorenson and HP power supplies.

5.3.2 Over-current protection circuit and Opto-isolator

The most valuable equipment in the experimental setup are the EMV actuator and the dSPACE prototyping controller board. Sustained high current (above 35A at over 5ms) presents the most significant hazard because it can melt the coil magnet and ruin the EMV actuator. Therefore, an over-current protection circuit is needed. In addition, opto-isolators enable the DS1103 board to control power electronic circuitry without risking electrical spikes and surges.

The purpose of the over-current protection circuit is to shut down the Sorenson power supply when a sustained (>5ms) and unacceptably high current (>35A) level is detected in the event of software errors or power electronic failures. To perform this task, the over-current protection circuit (shown in Figure 5.11) low-pass filters the output of the hall-effect current sensor and compares the filtered result to a given threshold, to determine if the power supply should be shut down.

The inclusion of opto-isolators enables the DS1103 board and the power electronics to be connected without the risk of voltage spikes from the power electronics ruining the controller board. The 16 single channel opto-couplers (Figure 5.12) used in this experiment transmit data via light sensing at a speed of 10Mbit per second. For more information about the opto-isolator, see [73].

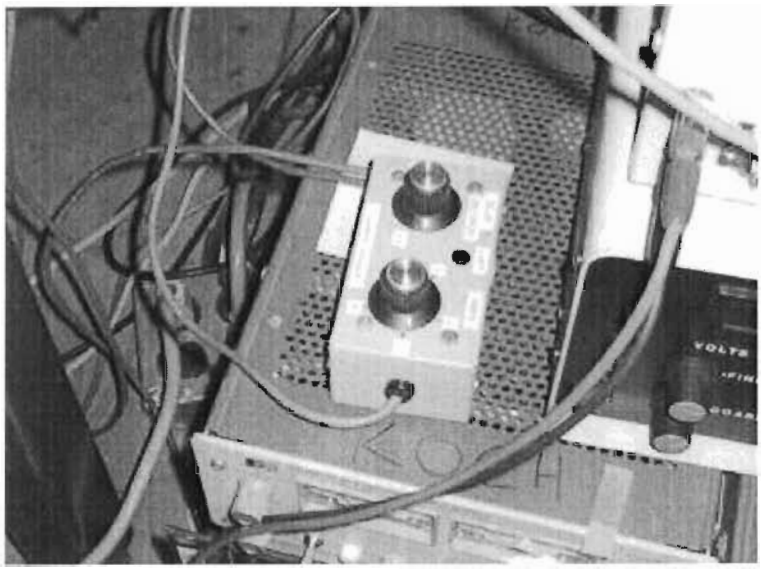

Figure 5.11: Over-current protection device (center).

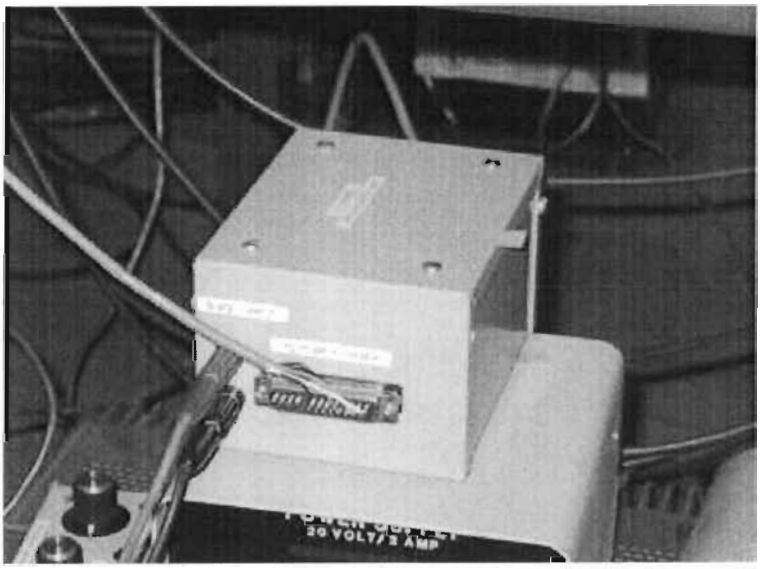

Figure 5.12: Opto-isolator.

5.4 Position, Voltage, and Current Sensor

Two position sensors are available in the experimental setup. The secondary sensor, not used for control and only for comparison with the primary sensor, is a custom built Eddy current type lift sensor [1](See Figure 5.13). The primary position sensor is a custom-made Linear variable differential transformer (**LVDT**) with a signal to noise ratio of 84dB and a bandwidth of greater than 20kHz. The documentation for this specific LVDT sensor is proprietary, the basic composition of the LVDT is shown in Figure 5.15: a powered primary coil (coil A) in the mid segment, two connected secondary coils (coil B) symmetrically wound in the two end segments, and a ferromagnetic core inserted coaxially without touching the coils. The two secondary coils are wound in series opposition to each other. If the core is placed in the middle position, the flux linkage from one secondary coil cancels the other; otherwise, depending on the axial displacement of the core, an imbalance of the flux linkage occurs and results in a voltage difference at the output of the secondary coils [74]. The actual LVDT sensor used in the experiment can be seen in Figure 5.14.

The current going into the EMV actuator is measured by the Hall-effect current sensor (LEM LA55-P) shown in Figure 5.16. In this sensor, current is detected by measuring the voltage of the semiconductor material induced by the magnetic field of the current-carrying wire. For voltage sensing, the potential across the actuator coils is measured by connecting a differential operational amplifier circuit with a configuration that gives a convenient fraction of the sensed voltage at output.

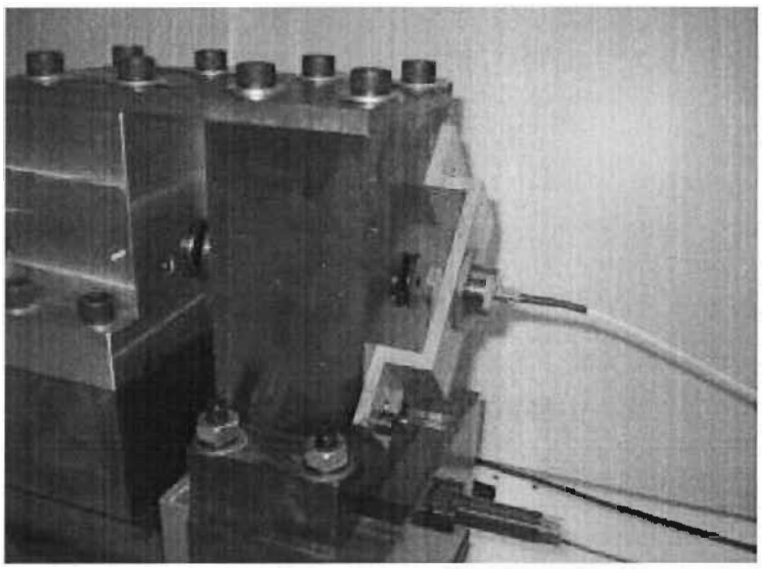

Figure 5.13: Eddy-current distancement sensor.

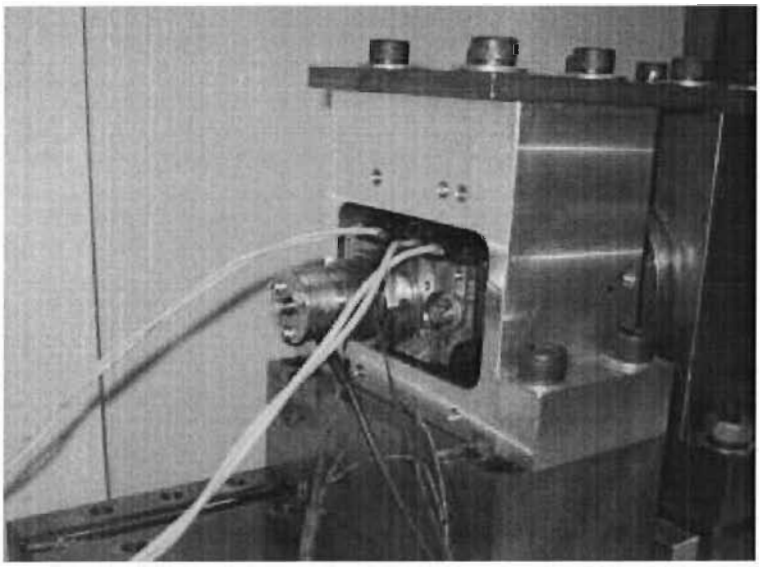

Figure 5.14: LVDT sensor placed in the front of the actuator housing.

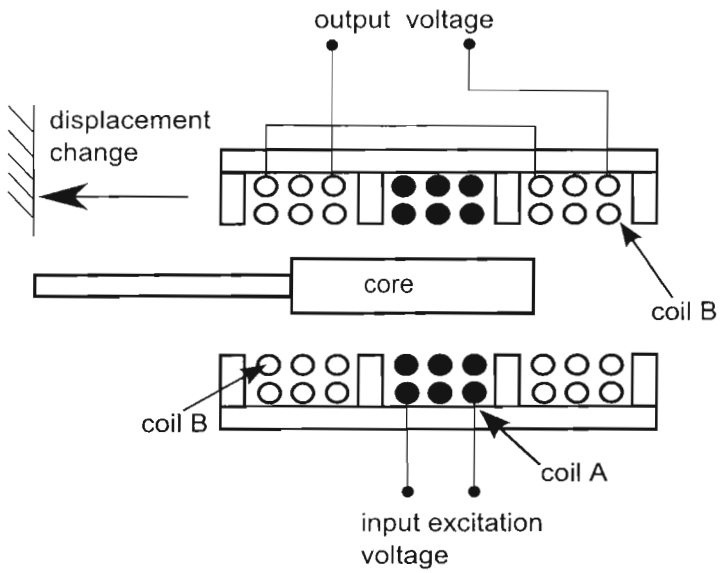

Figure 5.15: Illustration of an LVDT position sensor.

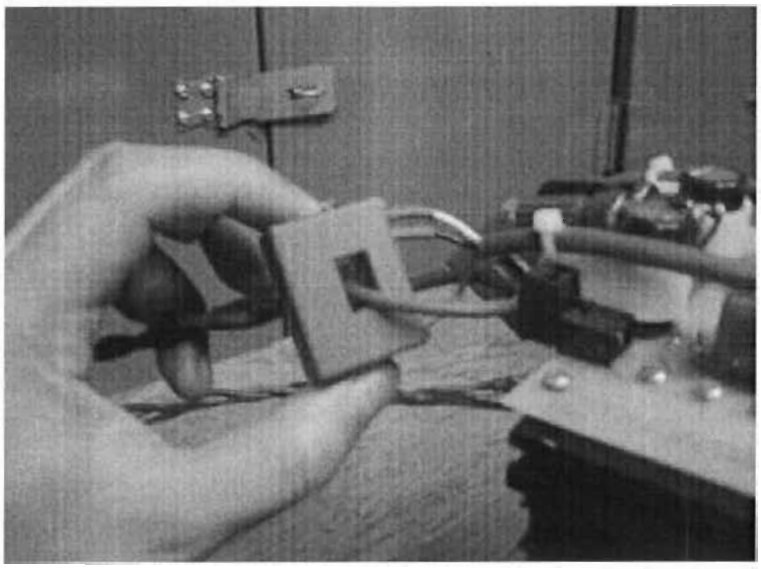

Figure 5.16: Hall-effect current sensor placed on top of the power electronics.

5.5 Summary

In this section, the hardware setup of the experiment discussed include: the EMV actuator, the position and current sensors, the power supplies, and the controller board. The realtime software programs on the prototype board and the monitoring-capturing program (ControlDesk) are also introduced. Last but not least, the functions of the over-current protection circuit and the opto-isolators are presented to illustrate the protection measures for the expensive EMV actuator and the dSPACE prototyping controller board. Table 5.2 gives a list of the devices used in the experiment.

Table 5.2: Devices used in the EMV approach control experiment

Device name	Description	Specifics
EMV actuator	electromagnetic actuator built by Telefunken Microelectronic GmBH, donated by DaimlerChrysler AG	8mm stroke and 250N/mm spring tension
dSPACE controller	DS1103 Controller board with PPC 604e processor and TMD320F240 slave processer	PWM and I/O output at 50kHz
Actuator power supply	Sorenson DCS60-18E programmable switching power supply	0-60VDC 0-18A Input 120VAC Max. output 1kW
Auxiliary power supply	HP 6236 triple output linear power supply	6VDC $\pm$20 VDC
Power Electronics	Custom H-bridge designed by Bazooka electronics	IGBT IRG4BC40W switching to 50khz with Cap at 200V @70A
Opto-Isolator	Optical linkage between DS1103 boar and power electronics	10Mbit/s $\pm$15 VDC input
LVDT displacement sensor	Custom linear variable differential transformer	Bandwidth $>$ 20kHz

Chapter 6

Experimental Results

This chapter starts with important issues encountered during experimentation, such as the system repeatability and the current controller latency. Then the experimental results of both the Terminal ILC and the Nelder Mead controller are presented. The data sets gathered for Nelder Mead controller are the focus of this section because the Terminal ILC controller failed to converge under experimental conditions.

The interpretation of the figures in this chapter requires an understanding of the approach control objective, which is to regulate terminal velocity v^f_{appr} and current i^f_{appr}. These two values are defined to be actuator states at position x^f_{appr}, which is the final position of the approach control. For this thesis, the position x^f_{appr} is defined to be at 2.55mm air gap away from the catching magnetic coil. For the division of approach and landing control, see the explanation in Section 1.3.

The results from the Nelder Mead controller are divided into two parts: the first part showcases terminal velocity v^f_{appr}, and current, i^f_{appr} against the unknown temperature/parameter related disturbance. The second part showcases its regulation against step and ramp disturbances from simulated cylinder back pressure.

6.1 Experimental Details

6.1.1 Consistency issue

The underlying assumption of the cycle-adaptive feed-forward algorithm is that useful control information for the current cycle can be learned from the previous iterations. If the

system parameters or disturbance forces change dramatically over consecutive iterations, then the tuning algorithm will fail. From the author's experience, the repeatability of the experimental setup between iterations can sometimes be inconsistent. Fortunately, so long as the actuator and the valve stay well lubricated, the output of the EMV test-bench usually changes slowly enough across cycles for the controller to be effective. The top and bottom plot in Figure 6.1 contrast these two conditions. Both graphs showcase the terminal velocity, v^f_{appr}. The top graph shows fast and drastic changes (as fast as 0.75 m/s within 15 samples) that will cause the cycle-adaptive algorithm be ineffective, while the bottom plot shows changes slow enough to be compensated for by a cycle-adaptive algorithm such as Nelder Mead.

6.1.2 Measurement interpolation

Because the cycle-adaptive algorithm in this chapter depends on the velocity, v^f_{appr}, and current, i^f_{appr}, at the end of the approach control (position x^f_{appr}), special attention is paid to the acquisition of these two estimates. Because no velocity sensor is used, armature velocity is estimated based on position measurement, using the velocity estimator in [1].

In addition to position measurement noise, the discrete nature of the digital system also introduces errors. Because the target, v^f_{appr} and i^f_{appr}, are fixed at a particular position instead of time, the velocity estimate for different time samples must be interpolated, using the velocity estimates before and after the desired position x^f_{appr}.

$$\begin{aligned} v^f_{appr} &= v_1 + \frac{(x^f_{appr} - x_1)}{x_2 - x_1}(v_1 - v_2) \\ i^f_{appr} &= i_1 + \frac{(x^f_{appr} - x_1)}{x_2 - x_1}(i_1 - i_2) \end{aligned} \tag{6.1}$$

The variables x, v, i, are the armature position, armature velocity, and catching coil current of the actuator. The subscript 1 and 2 stands for the measurement sample before and after the position x^f_{appr} passes.

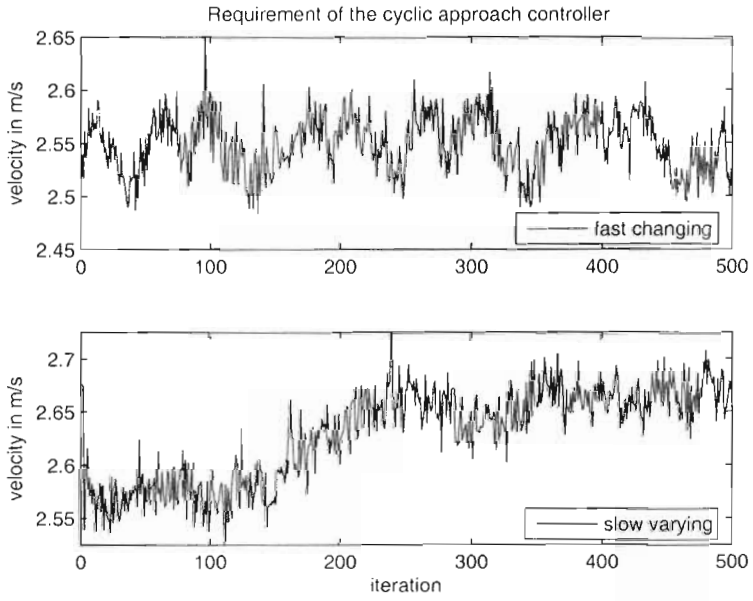

Figure 6.1: An illustration of cycle-adaptation requirement. Velocity at x^f_{appr} versus iterations under open-loop control. The cycle-adaptive controllers investigated in this thesis will not perform in the case of the top plot, in which the disturbance changes too quickly to allow learning. The bottom shows a system with disturbance that changes slowly enough to be compensated by the cycle-adaptive controllers.

6.1.3 On-off current controller

While Terminal ILC can use voltage control directly, voltage feedback linearization may have saturation and robustness issues. Therefore, the input magnetic force profile generated by the Terminal ILC controller is realized through current control similar to what is done in the Nelder Mead controller. The current control scheme used in the author's experiment utilizes a bang-bang controller. Specifically, the voltage is set to its maximum if the actual current is less than the desired current, while the voltage is zeroed quickly (by the pseudo -42 mode) when the desired current is less than the actual current. The actual C code implementation is the following:

```
dI = Isoll - Icl;
if ( dI > ctPar_zpr_hyst) {
    Sw_cl = C_ON_FAST;
    pwm_cl = 1.0;
} else if ( dI < -ctPar_zpr_hyst ) {
    Sw_cl = C_OFF_SLOW;
    pwm_cl = 0.0;
}
```

`dI` is the difference between the desired and actual current and `ctPar_zpr_hyst` is the hysteresis parameter. `Sw_cl` and `pwm_cl` are the power switching transistor's logic command and duty cycle explained in the experimental setup chapter. The discrete voltage states that `Sw_cl` can switch are defined as:

$$\text{Sw\_cl} = \begin{cases} 0 & \text{slow-on (42 volt)} \\ 1 & \text{fast-on (42 volt)} \\ 2 & \text{slow-off (0 volt)} \\ 3 & \text{fast-off (pseudo -42 volt)} \end{cases} \tag{6.2}$$

While both cycle-adaptive approach controllers use a B-spline current profile, different interpolation are used. The interpolation used in the Terminal ILC is time-based because the controller relies on time-based prediction of terminal states. In addition, the time duration needed for interpolation by the Terminal ILC controller is taken from the previous iteration. Assuming no drastic pressure variations between iterations, this simplification would be reasonable because the EMV system dynamic is dominated by spring forces. On the

other hand, the interpolation for the Nelder Mead direct search controller can be based on the position instead of time. Because the armature always travels between -4mm to 2.55mm during the approach control but its travel duration may vary cycle by cycle, position-based interpolation should be more robust than time-based interpolation. Figure 6.2 contains examples of the position-based interpolation (top) and time based-interpolation (bottom) for the Nelder Mead and Terminal ILC controllers respectively.

6.1.4 Coordinate change

Note that for the C code implementation, the coordinate system is shifted from -4mm to 4mm to -8mm to 0mm; this shift is reflected in the x-axis of the figures presented in this chapter. The point that approach control transition into landing control, x^f_{appr} or x^i_{land}, is now at position -1.45mm in the new coordinate.

Problems with the on-off current controller

The current profile approximation simplifies the approach control, but also introduces a latency issue. In Figure 6.2, one can see that the actual current roughly follows the desired current. The sawtooth-like waveform of the actual current is possibly due to the system inductance. Note the transistor logic states of `Sw_cl` for Figure 6.3 are defined by Eq. 6.2. If the effect of the current command is immediate, the actual current should change direction right after the logic signal changes. However, one can see in Figure 6.3, which is a zoomed-in presentation of Figure 6.2, it takes 2 to 4 samples after the voltage changes at position A, B, and C to reverse the direction of the current. This latency poses problems because it reduces current tracking accuracy. As a result, a small change of the desired current can cause the actual current to fluctuate much more. This problem can be seen when the approach control is implemented, especially for the terminal current i^f_{appr}.

Last, but not least, one must be careful to check if voltage saturation occurs. If the commanded current requires a steep increase, the desired and the actual current profile may not match because the 42 volt input voltage cannot raise current quickly enough. However, a slow current rise time is more of a system characteristic than a current control defect. This problem is alleviated by the coefficient upper bound constraint, c_{lb}, of the Nelder Mead controller. In contrast, the Terminal ILC controller tested in this chapter does not guard against saturation.

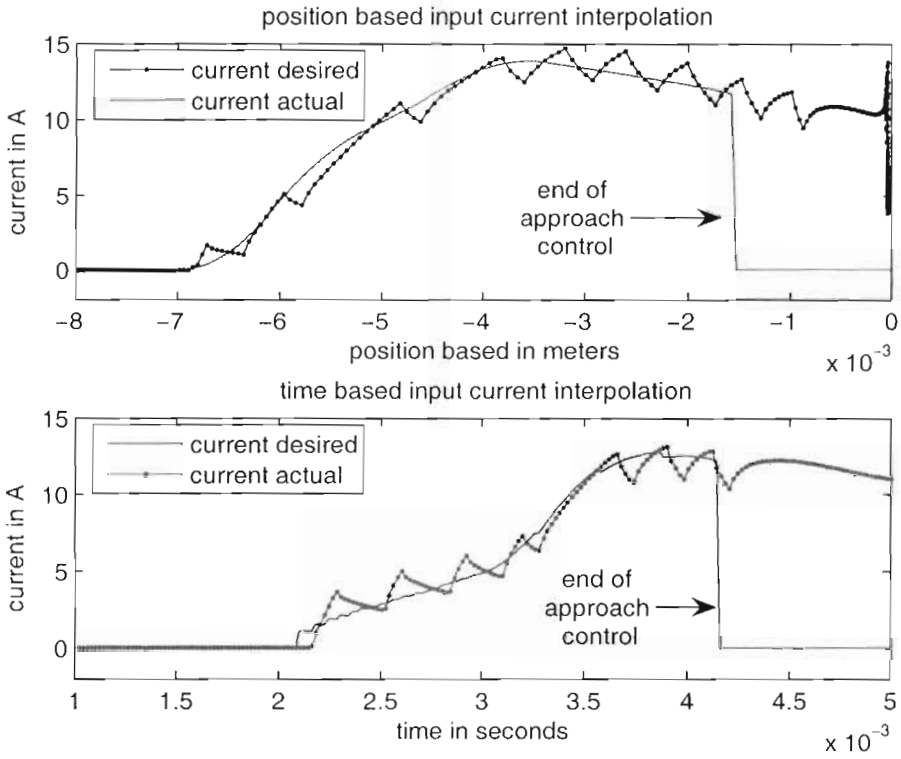

Figure 6.2: Position (top) and time-based (bottom) desired current profile interpolation. The actual current profiles are results of the on-off current controller tracking the desired current profiles.

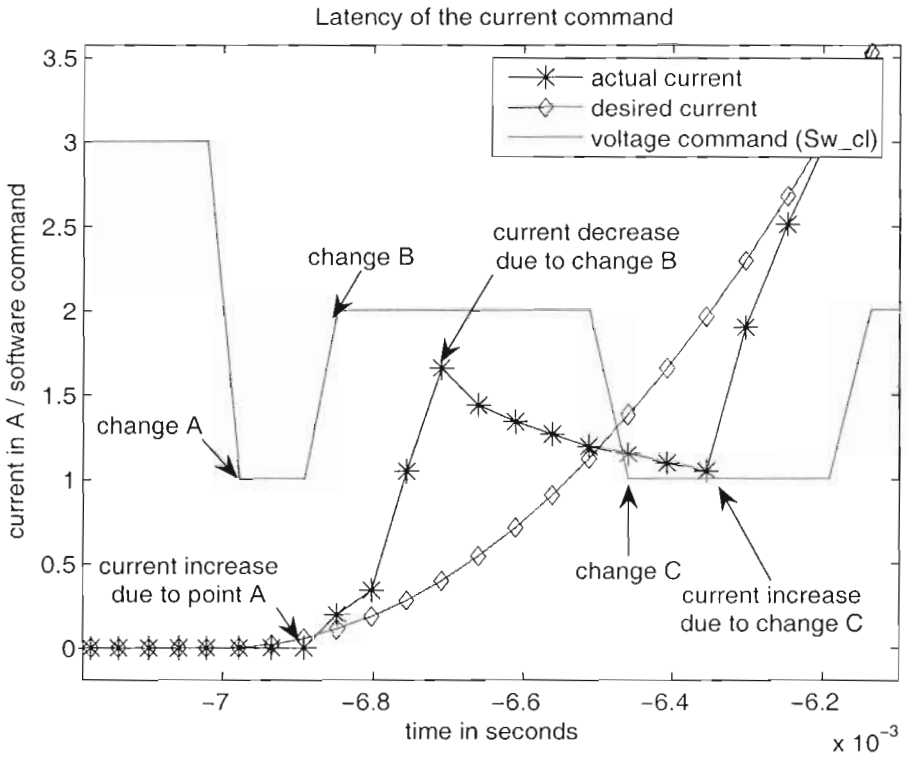

Figure 6.3: An illustration of the delayed response in current control. Voltage logic command and measured current versus time.

6.2 Experimental Results For Terminal Iterative Learning Controller

When applied to the actual plant, the Terminal ILC controller fails to regulate both velocity and current. In fact, the velocity at the end of the approach trajectory slows down significantly (see Figure 6.4). The evolution of the current input profile shows that the control coefficient requires a more drastic current amplitude change as time goes on. In Figure 6.5, which shows the input current profile at the 100th iteration, the actual current has problems following the desired current because the desired current is not physically realizable. Ultimately, the lack of error reduction causes the integrator in Terminal ILC to fail.

Despite being successful in simulation (see Figure 4.8 to 4.12), the Terminal ILC controller can not achieve convergence in the real world. The major problem are as follows:

1. The controller is too reliant on feedback linearization to keep the system linear.
2. The controller does not have enough control over the evolution of the control coefficients (i.e., there is no way to constrain the control coefficients)
3. The noisy velocity and current measurements also cause problems with the integrator of the learning controller.

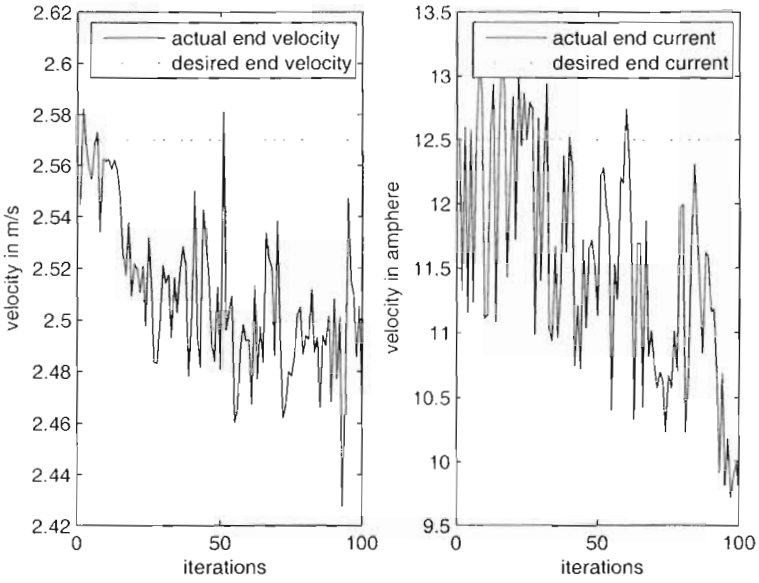

Figure 6.4: v^f_{appr} and i^f_{appr} versus iterations under Terminal ILC control. The Terminal ILC failed to eliminate errors on the test-bench.

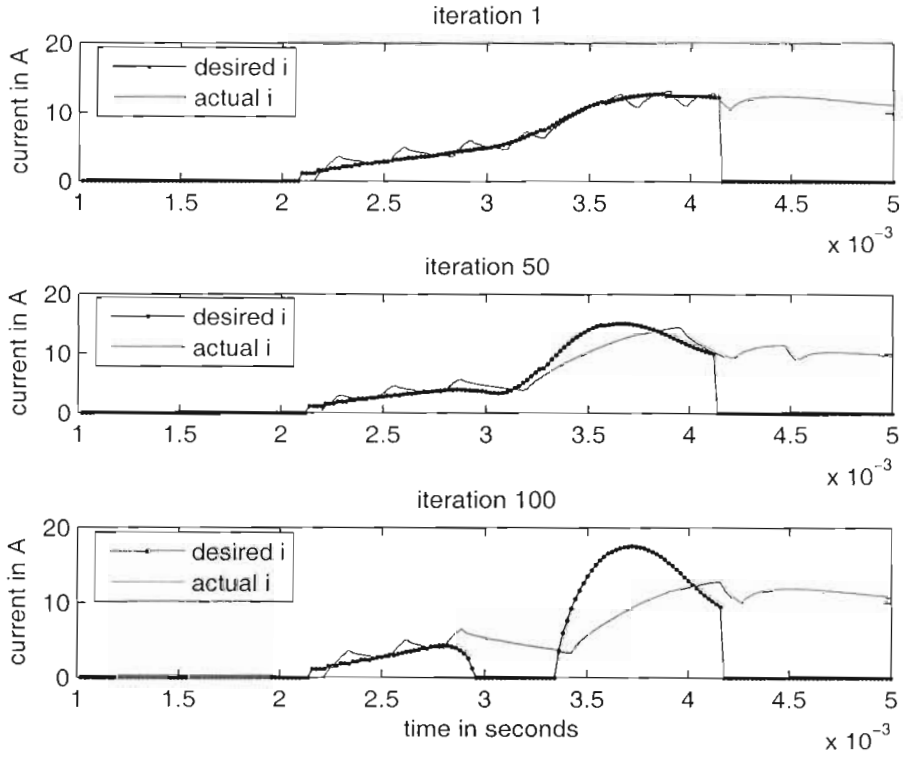

Figure 6.5: Desired and actual coil current versus time at iteration 1, 50, and 100 under Terminal ILC control. The current profile computed from Terminal ILC becomes un-realizable as iteration increases.

6.3 Experimental Results For the Nelder Mead Algorithm

Despite the problem with current control and noisy measurements described in section 6.1.2 and 6.1.4, the Nelder Mead simplex algorithm was able to converge after a few hundred iterations to the desired velocity and current (sometimes with some steady state errors in the current). The simplex algorithm has proven itself to be computationally efficient and robust for a real-time application such as the electromechanical valve.

Note that some over-sampling in the functional evaluations was done to prevent occasional noise spikes from affecting the ordering of the simplex vertices. The median of the samples is taken for the cost function computation because large noise spikes skew the average of the sample.

6.3.1 Convergence over set-points change

The experiments in Figure 6.6 and 6.7 are conducted by changing the velocity and current set-point values at the end of approach control. They show how the velocity and current converge to the desired state after a period in the transient state. The current went from 10 to 12 Amperes while the velocity increased from 2.55 to 2.65 m/s. In the case of Figure 6.6, both current and velocity are commanded to increase. In the case of Figure 6.7, the current, i^f_{appr}, is commanded to stay the same while the velocity is reduced from 2.62m/s to 2.56m/s. From experimental experiences, the author notes that the end velocity mostly converges well whereas the end current sometimes has steady state errors (less than 1A). Note that the over-sampling factor of 3 is determined to be adequate for noise/estiamtion error in the terminal velocity and current estimate.

The convergence to a different set-point is slow compared to the rest of the experiments (roughly 300 iterations to adapt to the new set-point). However, this experiment is done with a more conservative set of controller tuning parameters (ρ, χ, γ, and σ). The set-point-change experiment done in the simulation, using the same controller parameters, shows a similar convergence speed as illustrated in Figure 4.17. The matched results demonstrate that the simulation model developed in [1] is accurate enough so that the Nelder Mead controller can be designed in simulation and yield satisfactory result on the test-bench.

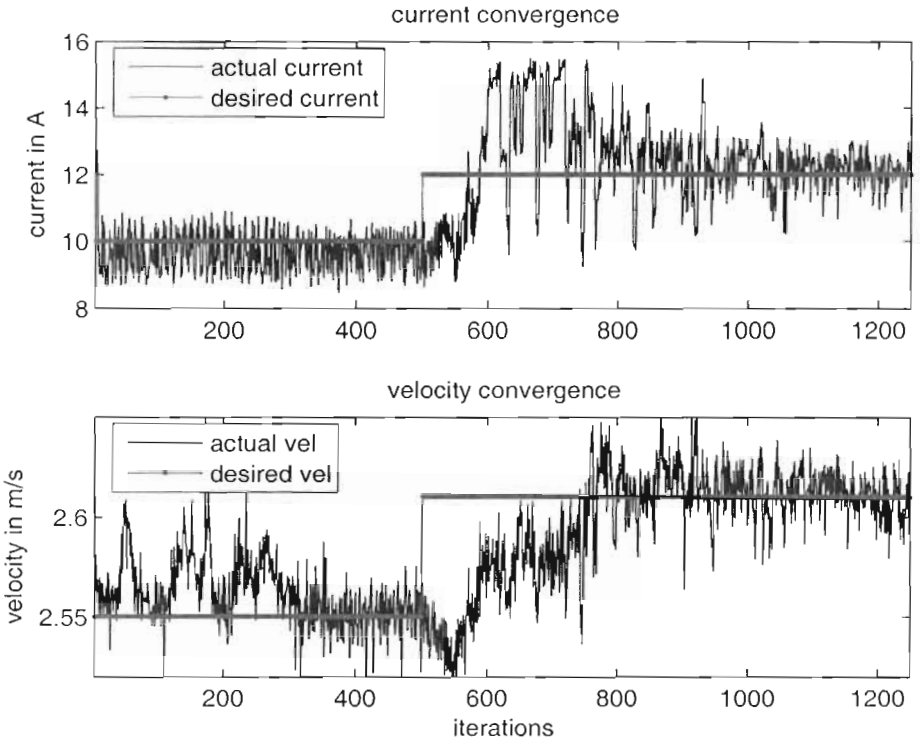

Figure 6.6: An illustration of the Nelder Mead controller's regulation against set-point change. v^f_{appr} and i^f_{appr} versus iterations. Both set-points increase at the 500th iteration.

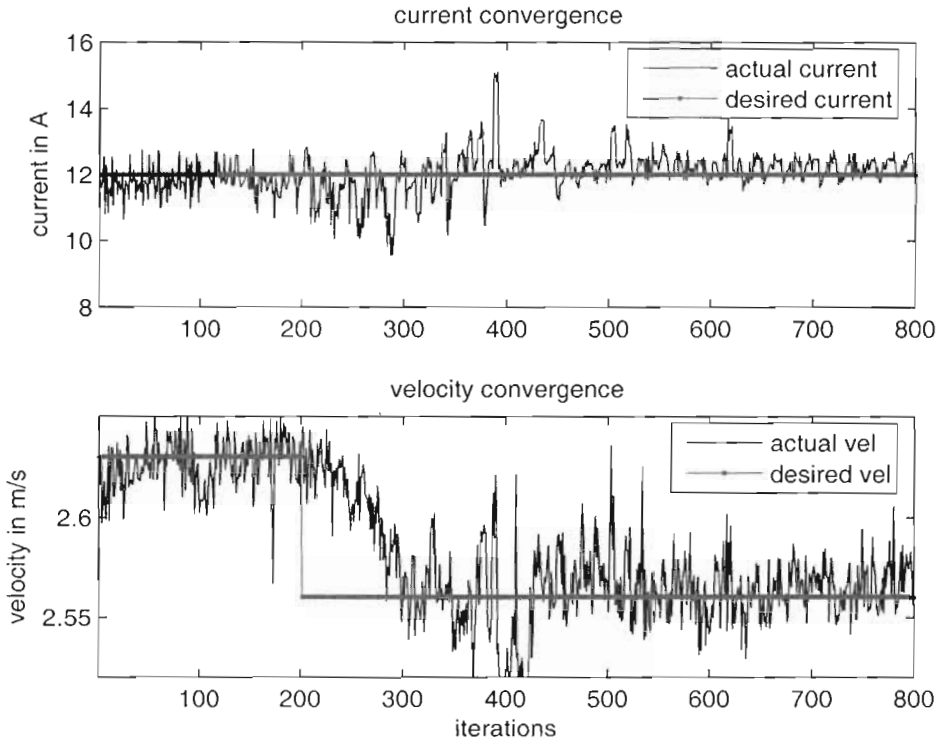

Figure 6.7: An illustration of the Nelder Mead controller's regulation against set-point change. v^f_{appr} and i^f_{appr} versus iterations Only the velocity set-point decreases at the 200th iteration.

6.3.2 Regulation against unknown disturbance

Even under laboratory condition, the valve actuator test bench setup is affected by an unknown time-varying disturbance. The author believes the cause of this disturbance is the same temperature-related variation in spring compression force reported in Eyabi's sensorless control paper [41].

To demonstrate that this problem can be alleviated, the Nelder Mead algorithm is run continuously for four thousand cycles. Following that, the controlled terminal conditions are compared to the open-loop results. One can see from Figure 6.8 that the Nelder Mead controller regulates the velocity to the desired point better than the open-loop controller. The histogram in Figure 6.9 show the distribution of the terminal velocity, v^f_{appr}, during the four thousand cycles. While both controllers keep the average velocity at the desired $2.6m/s$, the velocity variance resulted under Nelder Mead control is less then one third from the variance resulted under open-loop control. (0.2e-4m^2/s^2 compared to 0.69e-4m^2/s^2).

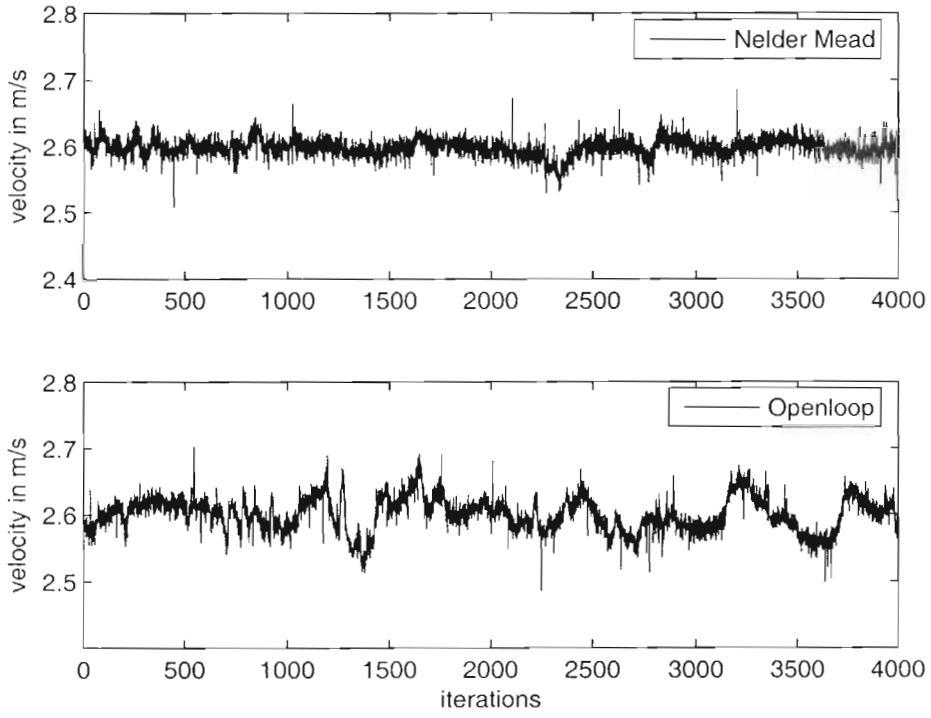

Figure 6.8: Regulated terminal velocity v_{appr}^{f} during 4000 valve events: Nelder Mead (top), open-loop (bottom).

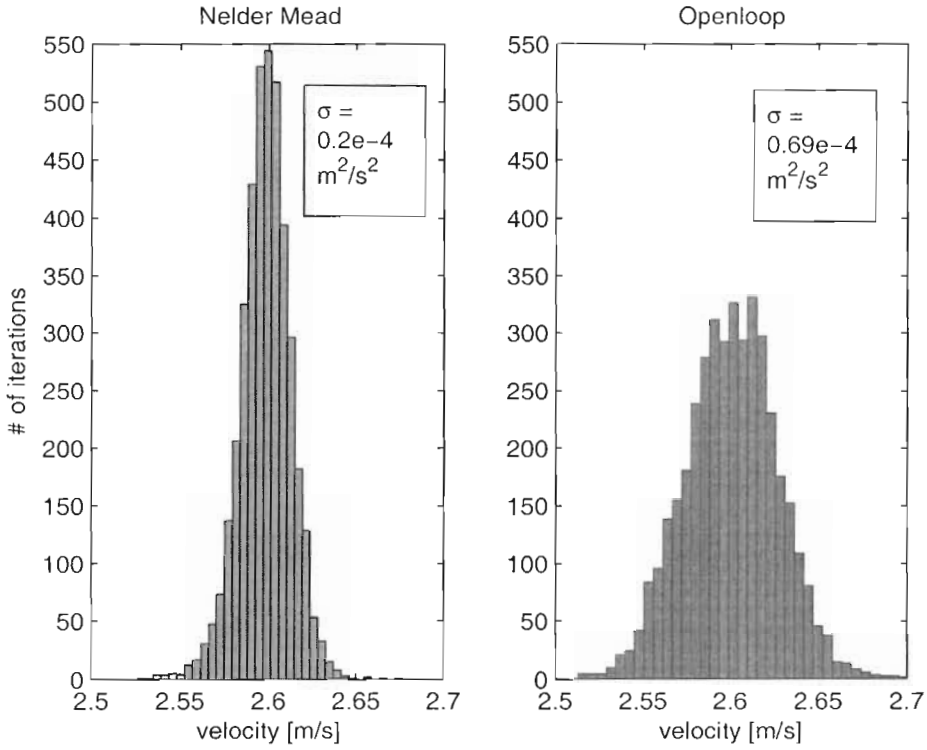

Figure 6.9: Histogram of v_{appr}^{f} after 4000 valve events: Nelder Mead (left) and open-loop (right).

6.3.3 Cold start regulation

The EMV actuator warms up from the ambient temperature due to ohmic losses in the coil. When the EMV actuator is first activated, the sudden increase in thermal energy affects the spring and the electromagnet. At room temperature, higher current is required to properly close or open the valve when the EMV test-bench initiates its operation. After several hundred motions, the initial coefficients for current control become excessive and cause the armature to land at higher impact speeds. A good approach controller should be able to regulate the speed of the armature around the desired point as quickly as possible during a cold start and than ease off on the control as the system heats up. Figure 6.10 shows that the Nelder Mead controller regulates the velocity to stay around 2.6 m/s in 150 iterations whereas the open-loop control requires 700 iterations to reach 2.6 m/s velocity.

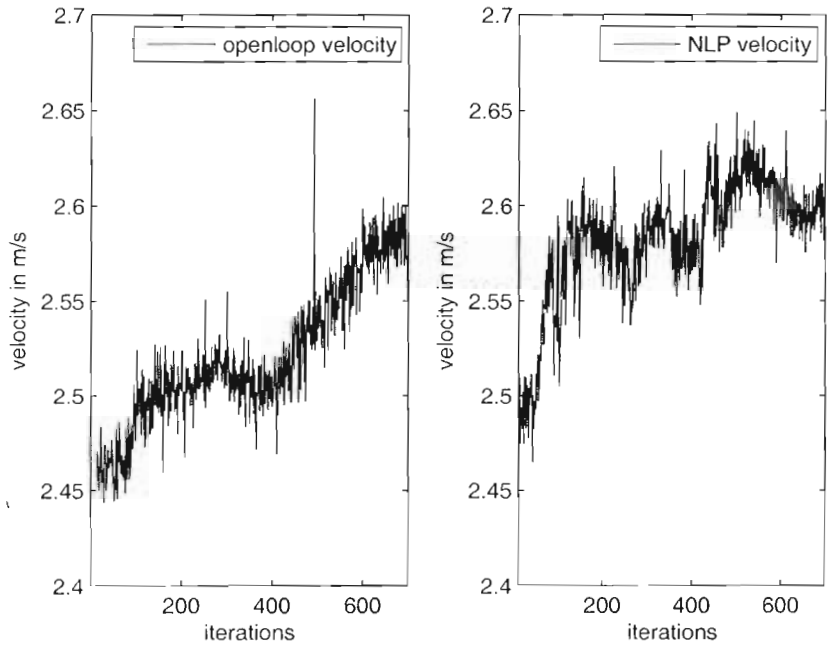

Figure 6.10: v^f_{appr} versus iterations under cold start condition: open-loop (left) and Nelder Mead (right). The Nelder Mead controller brings armature velocity to the desired much earlier then the open-loop control.

6.3.4 Improving energy efficiency

Placing a current integral during a valve motion into the controller's cost function allows energy consumption to be considered. The weight of the current integral must be much less than the weight on the velocity and position error to ensure that the set-points have higher priority. If the system is close enough to its required set-point, then the cost of the current integral will be more dominant, and the Nelder Mead algorithm will shape the current profile such that the system can be more energy efficient:

$$\text{cost} = (\frac{\text{velocity error}}{\text{velocity weight}})^2 + (\frac{\text{current error}}{\text{current weight}})^2 + \text{current integral weight} * \int i(t)dt \tag{6.3}$$

In Figure 6.11, one can see that the current integral (top graph) is lowered as soon as the weighting of the current integral is increased in the cost function. At the same time, the velocity and current is regulated around the set-point of 2.6 m/s and 12 A.

The result in Figure 6.11 can be compared to the simulation in Figure 4.22 of Section 4.3. The simulation reports current integral reduction of 16.5e-3 to 16.25e-3 ampere-seconds compared to the experimental result of 16.9e-3 to 16.7e-3 ampere-seconds. Aside from slightly higher current consumption perhaps due to unmodeled dynamics, the actual controller performed as expected in the simulation. The successful prediction demonstrates the utility of the simulation model for controller design.

Note, instead of a current integral, an energy expenditure term in Joules (integral of voltage times current) during a valve motion can also be used in the cost function. While this is not done in this thesis, the impact from such an additional cost function term is expected to be similar to that of a current integral term.

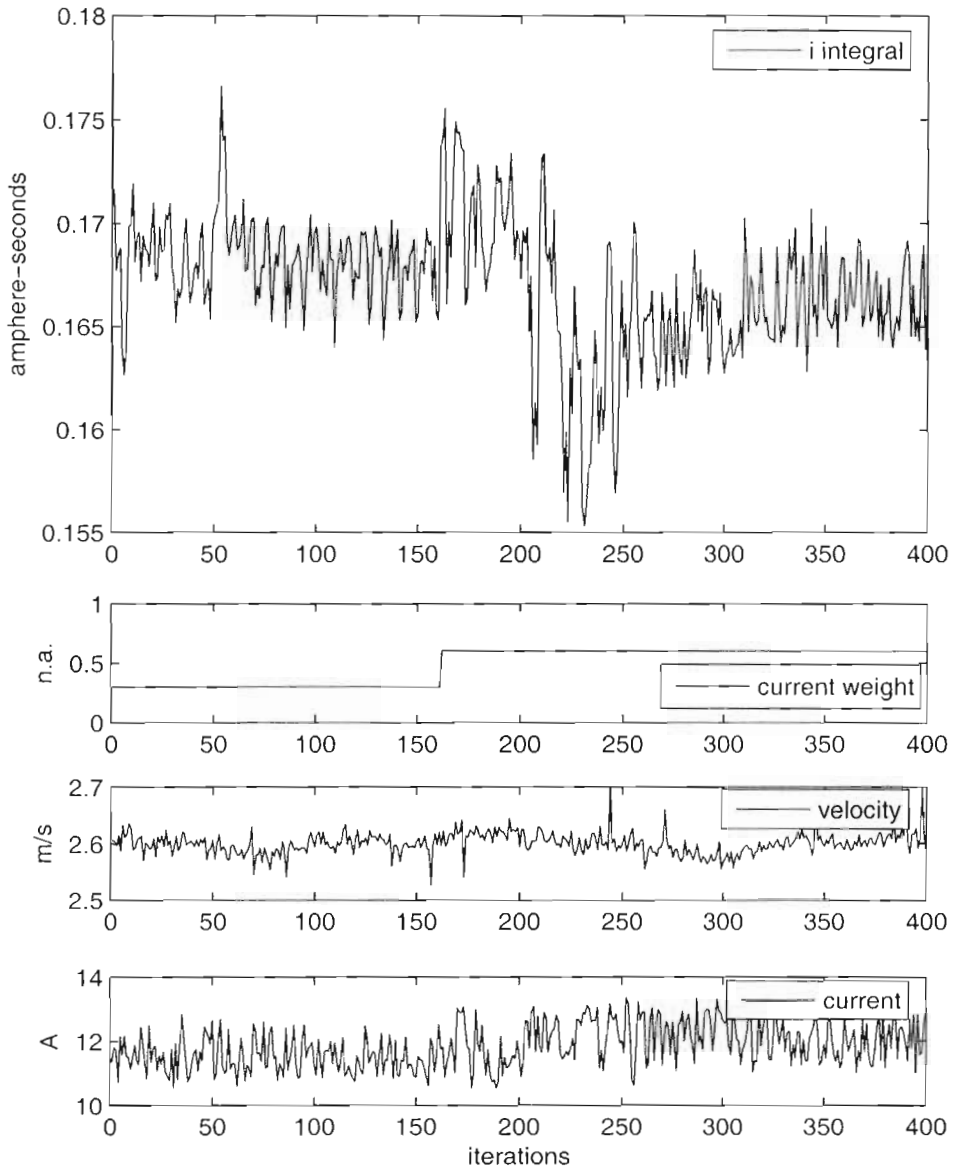

Figure 6.11: An illustration of energy reduction by adding a weighted current integral term to the cost function of the Nelder Mead controller. The top plot is coil current integral versus iterations. The second highest plot is the current integral weight in the controller's cost function versus iterations. The bottom two plots are v^f_{appr} and i^f_{appr} versus iterations. This result matches with the simulation in Figure 4.22

.

6.4 Simulating a Back Pressure via Release Coil

In section 6.3.2, the Nelder Mead Controller regulates the EMV coil current and armature velocity around the set-point under the time-varying unknown disturbances inherent to the system. Because the disturbance is unknown, it's impossible to quantify how well the the algorithm rejects disturbance. To truly test a controller's performance, a known disturbance source must be utilized.

This experimental testbench is not equipped with actual pressure forces on the valve. However, by using the information from the finite element model work in [31], a release coil current profile can be applied to hold back the armature during release as if a pressure valve is connected to the EMV test-bench. This substitution can be made because both forces from pressure disturbance and magnetic forces are position based (as explained in chapter 3). The current profile can be further simplified by position-based interpolation of current coefficients as shown in the C-code.

```
volatile double ct_i_pgeg_kl[] = {      /* current to simulate back
pressure*/
    0.325067,  5.203673,  7.714763,  9.692079, 11.420484,
   12.619973, 13.025574, 12.644924, 11.775298, 10.861866,
   10.278948, 10.116052, 10.050468,  9.388188,  7.354937,
    3.719086,  0.000000
};
```

The coefficients above are used to simulate 100N back pressure. Scaling of the force is done by multiplying by a factor. For example, Figure 6.12 shows the desired opener current for 90 N of back pressure. It also shows, in the dotted line, the actual current which tracks the desired through an on-off voltage controller. The top plot of 6.12 is position-based, and the bottom plot represents the same data set but is time-based.

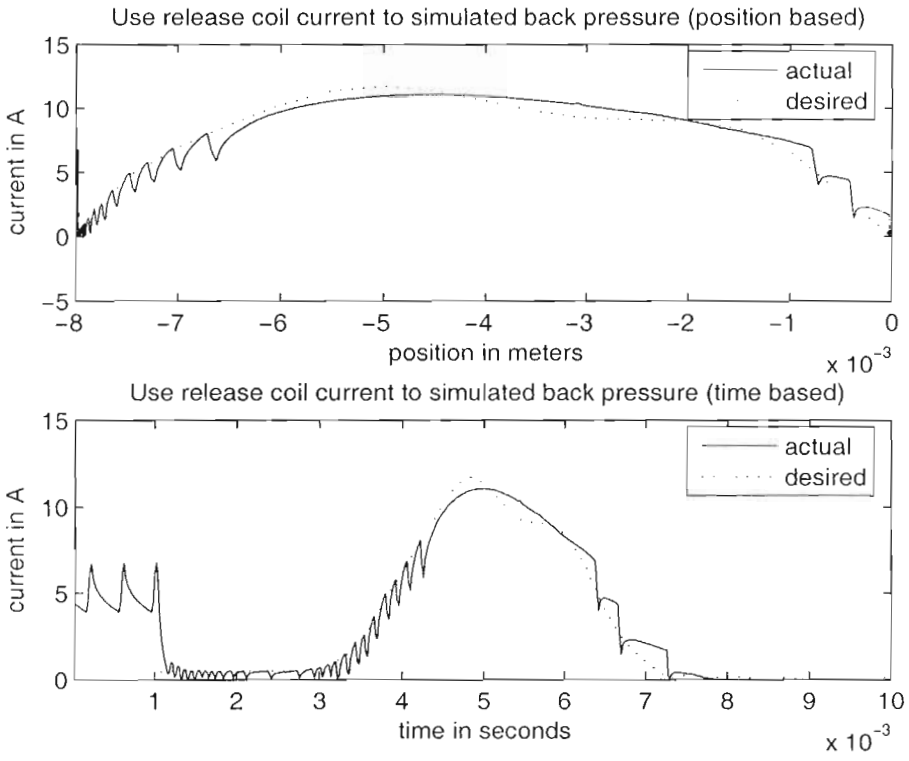

Figure 6.12: Release coil current profile needed to simulate 90N of cylinder back pressure. (top: current versus position, bottom: current versus time)

6.4.1 Maximum disturbance allowed and constraints determination

The simulated back pressures in this thesis are limited to less then 50N. To illustrate the reasoning behind this, the current needed for landing the valve against 90N of simulated back pressure is shown (the top plot of Figure 6.13). In this figure, most of the actual current is smooth due to saturation. At the end of the approach trajectory, the armature velocity is low (2.35m/s) and the current is very high (17.5A). In this case, saturation leaves no room for the controller to change the terminal velocity or current. At pressures above 100N, the armature fails to land due to lack of speed. Consequently, to allow the controller to regulate the terminal velocity and current within the nominal set-points, $2.45 \sim 2.6m/s$ at $10 \sim 14A$, the disturbance pressure is set to be no more than 50N.

The requirement corresponds roughly to 1 bar of cylinder back pressure. To see this, we can convert 1 bar into Pascal, and use the fact that the applied force is equal to pressure times the area of the valve:

$$\begin{aligned} & \text{force applied by 1 bar of pressure} \\ = \ & \underbrace{101325 * \text{pa/bar}}_{\text{pressure}} \times \underbrace{(\frac{25.4\text{mm} * 10^{-3}\text{m/mm}}{2})^2 \times \pi}_{\text{valve area}} \qquad (6.4) \\ = \ & 51.34 \text{ Newton} \qquad (6.5) \end{aligned}$$

The middle and bottom plot of the Figure 6.13 shows situations that are more suitable for feedforward tuning. The middle plot is used as the upper bound for the Nelder Mead controller coefficients, and the bottom plot is used as the lower bound. Because the Nelder Mead is not a model based algorithm, the selection of the two bounds does not have to be rigorous. For example, the coefficients in the middle and bottom plot of Figure 6.13 results in different terminal velocities (2.52m/s and 2.45m/s). However, as long as the results are within the desired velocity and current range ($2.45 \sim 2.6m/s$ at $10 \sim 14A$), these coefficients should be adequate.

Within these two bounds, the Nelder Mead algorithm tunes the current profile to regulate terminal velocity, v^f_{appr}, and current, i^f_{appr}, around the set-points, v_d and i_d.

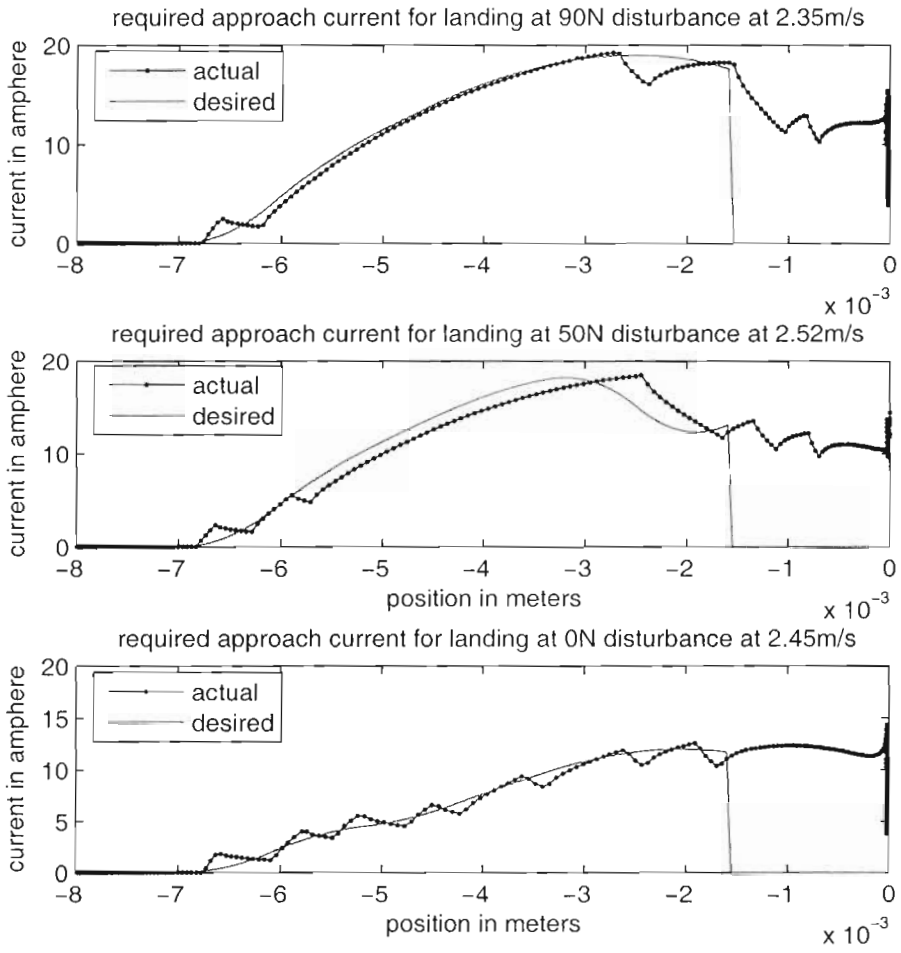

Figure 6.13: Current profiles required for valve landing at different pressure disturbance levels. (top: 90N, middle: 50N, and bottom: 0N)

6.5 Nelder Mead Algorithm Against Pressure Change

The experiment in this section is conducted by applying simulated step and ramp pressure disturbances, and recording the resultant velocity and current at the end of the approach control trajectory. The results show that the Nelder Mead controller is able to eliminate the effect of the disturbances on the terminal conditions, v^f_{appr} and i^f_{appr}, so that the landing trajectory starts at the specified initial conditions.

6.5.1 Step pressure change examples

To make an assessment of algorithm convergence speed, one can look at how many iterations are required for the Nelder Mead algorithm to guide the variables back to set-point under the disturbance. In example #1 (Figure 6.14), it takes around 100 steps for the velocity to return to the set-point under 40N of simulated pressure. In example #2 (Figure 6.15), it takes around 120 steps for the controller to return the velocity to the set-point under -40N of pressure disturbance. These results are similar to the step pressure change simulation shown in Figure 4.18 and 4.19 in the theory chapter. The experimental results are slightly slower (100 to 120 steps compared to slightly less than 100 steps in the simulation) because of the unmodeled noise and temperature related disturbances encountered in the experiments.

6.5.2 Cases for multiple step changes

Figure 6.17 shows the result from the Nelder Mead algorithm under a more realistic scenario by incorporating 4 step pressure changes in 600 cycles. As a contrast, the open-loop result under a similar disturbance pattern is shown in Figure 6.16. One can see that Nelder Mead algorithm regulated the terminal velocity v^f_{appr} back to the desired 2.55m/s around 100 steps after any step disturbance is applied. Some steady state errors did occur and shift the terminal current ,i^f_{appr}, from the desired 12.5m/s to 14m/s during the 50N pressure, but the additional current near the transition point, x^f_{appr}, is needed because of the saturation problem near the controller upper-bound (see Figure 6.13). The amount of disturbance that the controller can handle can be extended by either reducing the overall system inductance or by increasing the supply voltage.

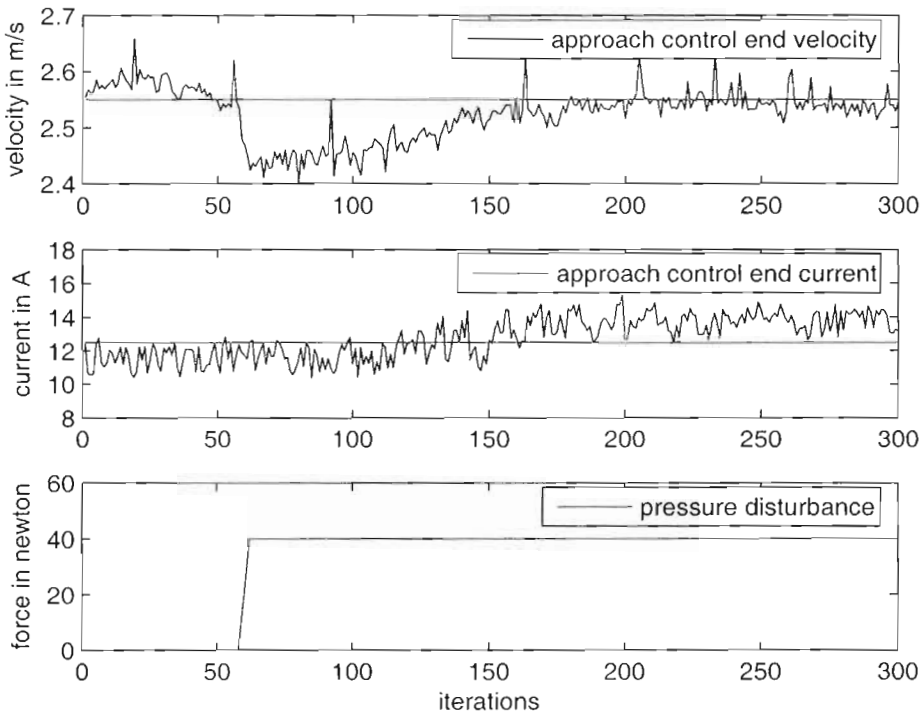

Figure 6.14: An illustration of the Nelder Mead controller's regulation against step pressure disturbance. The top two plots are v^f_{appr} and i^f_{appr} versus iterations under Nelder Mead control. The bottom plot shows a step disturbance pressure increase of 40N occuring at the 53th iteration.

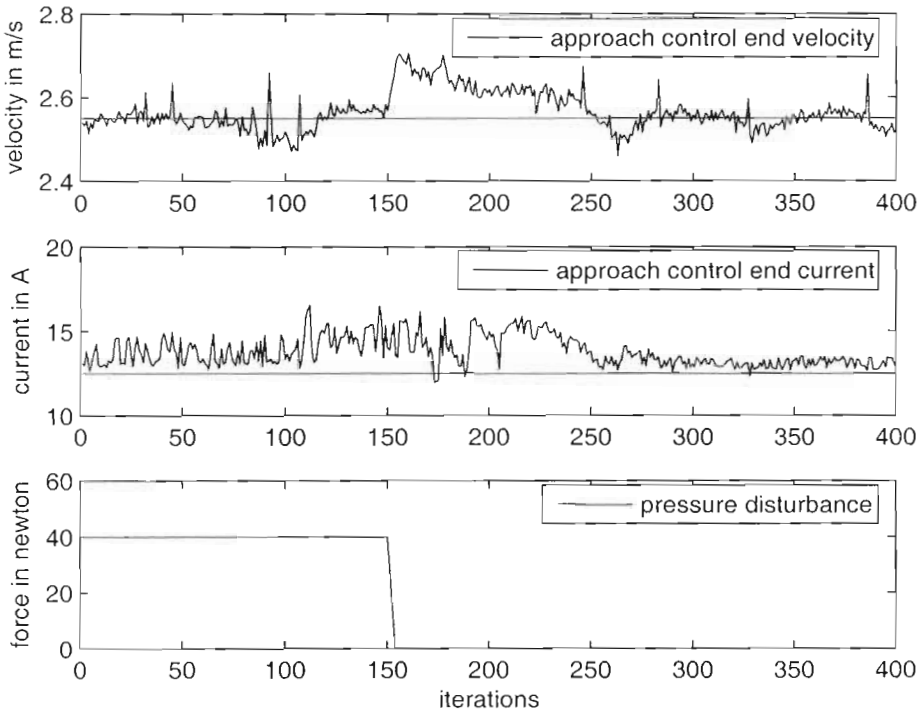

Figure 6.15: An illustration of the Nelder Mead controller's regulation against step pressure disturbance. The top two plots are v_{appr}^{f} and i_{appr}^{f} versus iterations under Nelder Mead control. The bottom plot shows a step disturbance pressure decrease of 40N occuring at the 151th iteration.

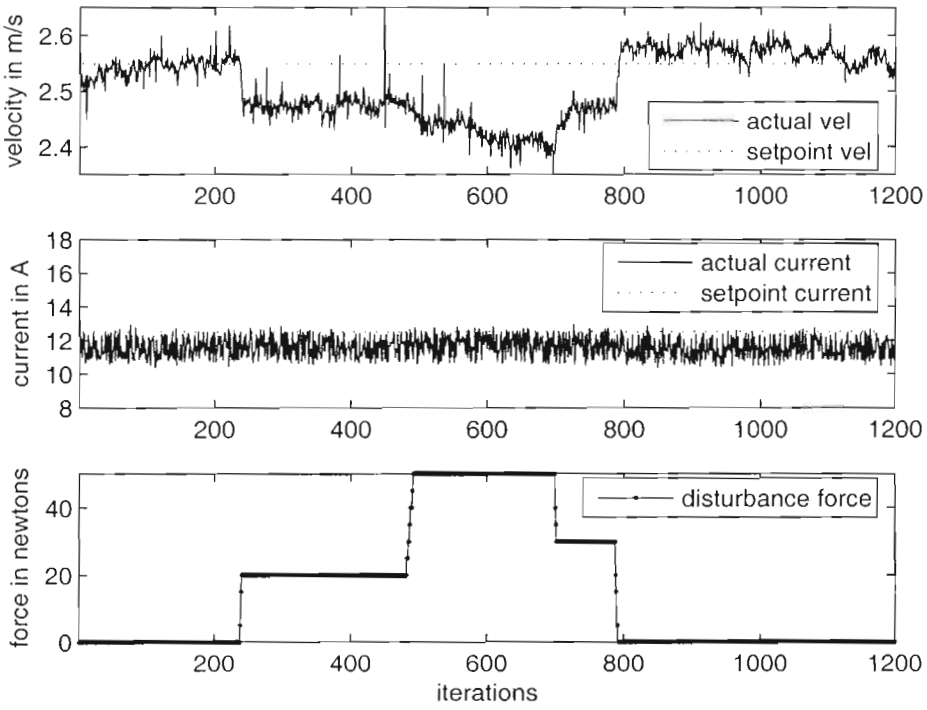

Figure 6.16: An illustration of the open-loop response under multiple step pressure changes. The top two plots are v^f_{appr} and i^f_{appr} versus iterations. The bottom plot records changes in the disturbance force.

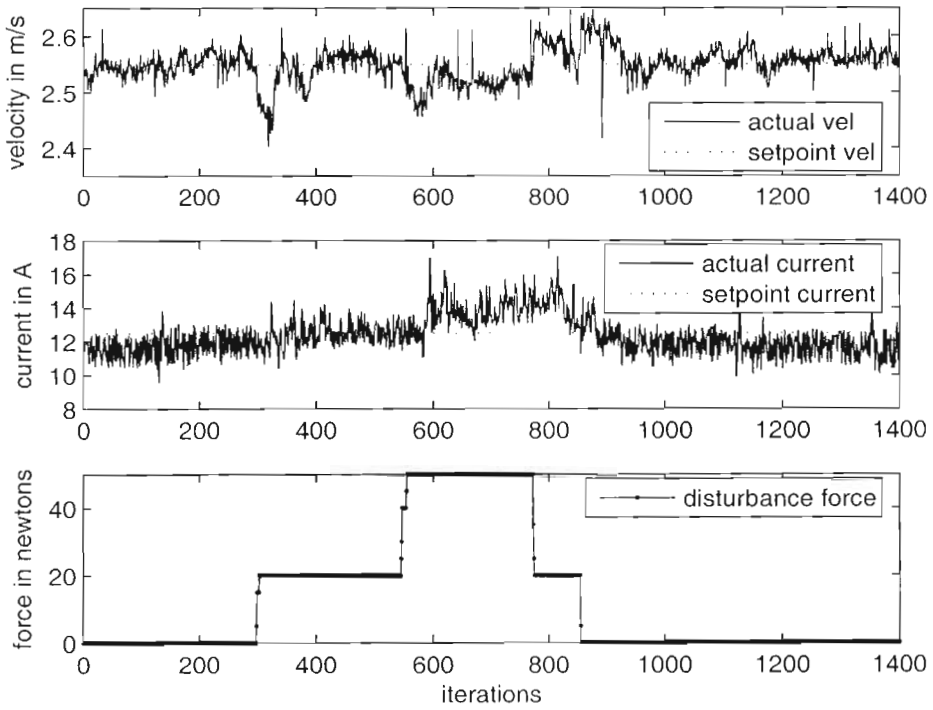

Figure 6.17: An illustration of the Nelder Mead controller regulating under multiple step pressure changes. The top two plots are v^f_{appr} and i^f_{appr} versus iterations. The bottom plot records changes in the disturbance force.

6.5.3 Ramp disturbance

Ramp disturbance presents a big challenge to the Nelder Mead controller because during a ramp, the vertices of the simplex are outdated very quickly. If any of the vertices are outdated, the resultant direction searched by the algorithm will not reduce the cost function. The effect is most pronounced during the *shrink simplex* operation. Usually the vertices of the simplex shrink toward the vertex associated with the lowest cost function, but because the evaluated cost function can be outdated, *shrinking* may not result is general reduction of the cost function. In fact, it is more likely that the average cost function will increase.

The experimental result shows that any ramp change faster than 10N per 100 iterations will cause problems for the Nelder mead controller. In this section, two examples are shown at this rate of change. Figure 6.18 shows a 30 Newton increase at 300 steps, while Figure 6.19 shows the same case but at decreasing pressures. As a comparison, an open-loop result under the same rate of ramp disturbance force is shown in Figure 6.20.

The experimental results can be compared to the ramp disturbance simulation in Figures 4.20 and 4.21 in the theory chapter. One can see that the results from the simulation and experimentation match, aside from the fact that the experimental plots contain the occasional noise spikes (possibility from realtime numerical errors) in addition to gaussian white noise.

6.5.4 Comparison with the results from the published literature

To put these results in context and compare them to the published literature is difficult because most publications do not focus solely on approach control, and measure their results in terms of terminal conditions, v^f_{appr} and i^f_{appr} at the transition point, x^f_{appr}. The closest match to our disturbance rejection result would be from the iterative learning control publication [2] by Hoffmann et al., whose simulation shows an iterative learning controller keeping the valve seating velocity below 0.1m/s against a pressure ramp increase of 20N at 100 iterations. Note that the controller in [2] uses a 200 volt power supply compared to our 42 volt power supply. While a higher voltage improves the control authority of the EMV system, the 42 volt supply is deemed to be the upcoming vehicle standard [72].

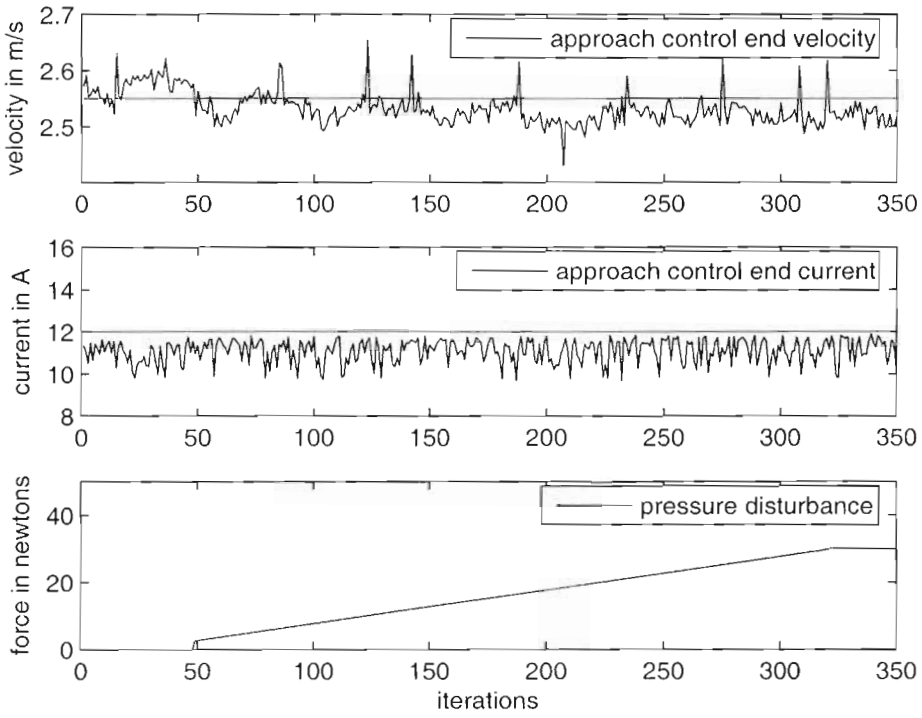

Figure 6.18: An illustration of the Nelder Mead controller's regulation against ramp pressure disturbance. The top two plots are v^f_{appr} and i^f_{appr} versus iterations. The bottom plot shows ramp disturbance pressure at rate of 10N per 100 steps occuring at the 49th iteration.

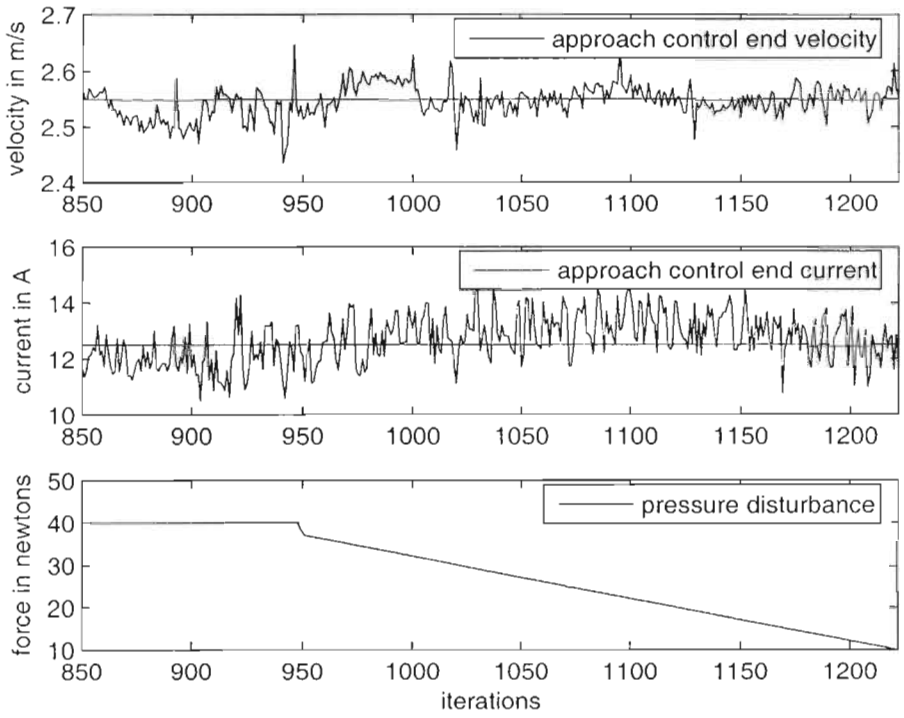

Figure 6.19: An illustration of the Nelder Mead controller's regulation against ramp pressure disturbance. The top two plots are v^f_{appr} and i^f_{appr} versus iterations. The bottom plot shows ramp disturbance pressure at rate of -10N per 100 steps occuring at the 950th iteration.

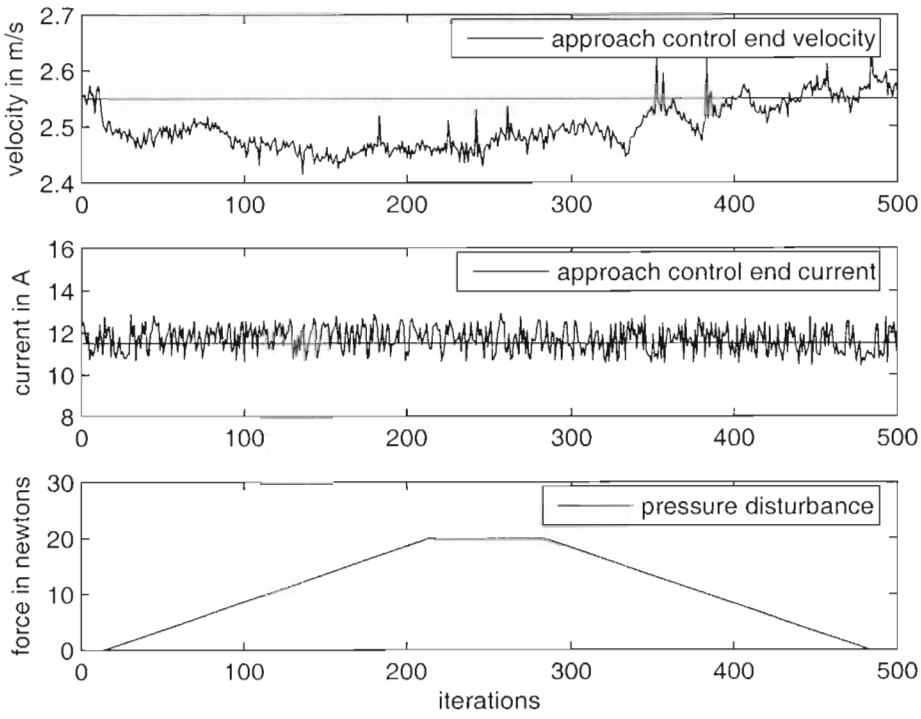

Figure 6.20: An illustration of the open-loop response under ramp pressure changes. Top and middle plots are v^f_{appr} and i^f_{appr} versus iterations. The bottom plot is the actual applied ramp disturbance.

6.5.5 Input coefficient evolution

In this section, the evolution of the current profile under the Nelder Mead algorithm is shown. Two examples are displayed: Figure 6.21 is the case where the actuator *heats up* and the system requires less energy to keep the same set-points at the end of the approach trajectory. Therefore, the Nelder Mead algorithm reduces the current profile during the approach control. The dotted lines are transients. Figure 6.22 shows the opposite situation where the controller raises the current profile to counter a step increase in back pressure at iteration 160 so that the velocity is kept the same.

The top graphs of Figure 6.21 and 6.22 are the evolution of current input with respect to valve lift tuned by the Nelder Mead algorithm; the middle and bottom plots validate that the controller keeps the end conditions at the desired set-point during these iterations. Again, the experimental results from this section are very similar to the simulation results presented in Figure 4.23, which shows the evolution of the input current profile under the Nelder Mead controller acting against a step pressure disturbance.

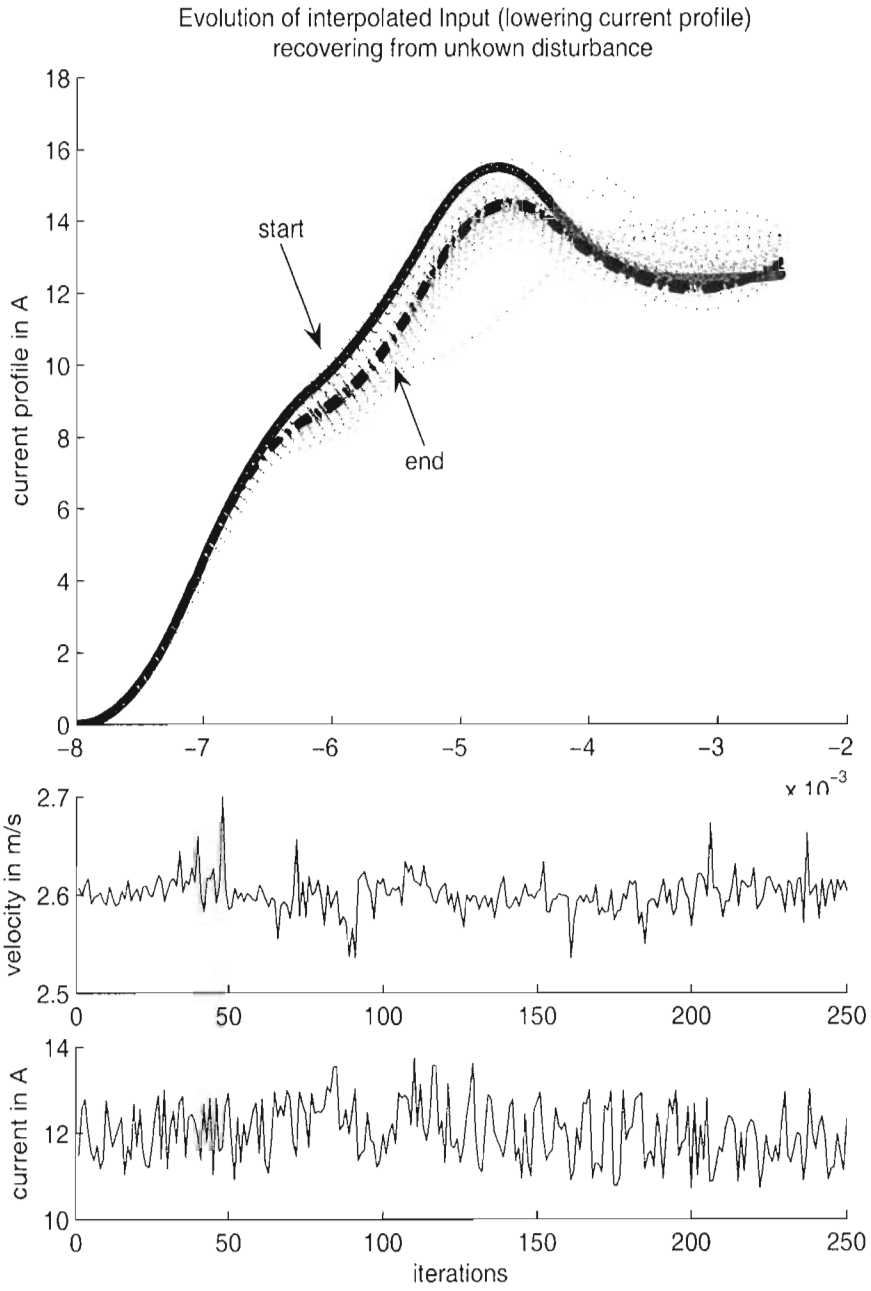

Figure 6.21: Evolution of the input current profile from the Nelder Mead controller regulating against unknown disturbance. The top plot is the current profile versus position from iteration 1 to 250. The bottom two plots are v^f_{appr} and i^f_{appr} versus iterations. The current profile is lowered to achieve the desired v^f_{appr}.

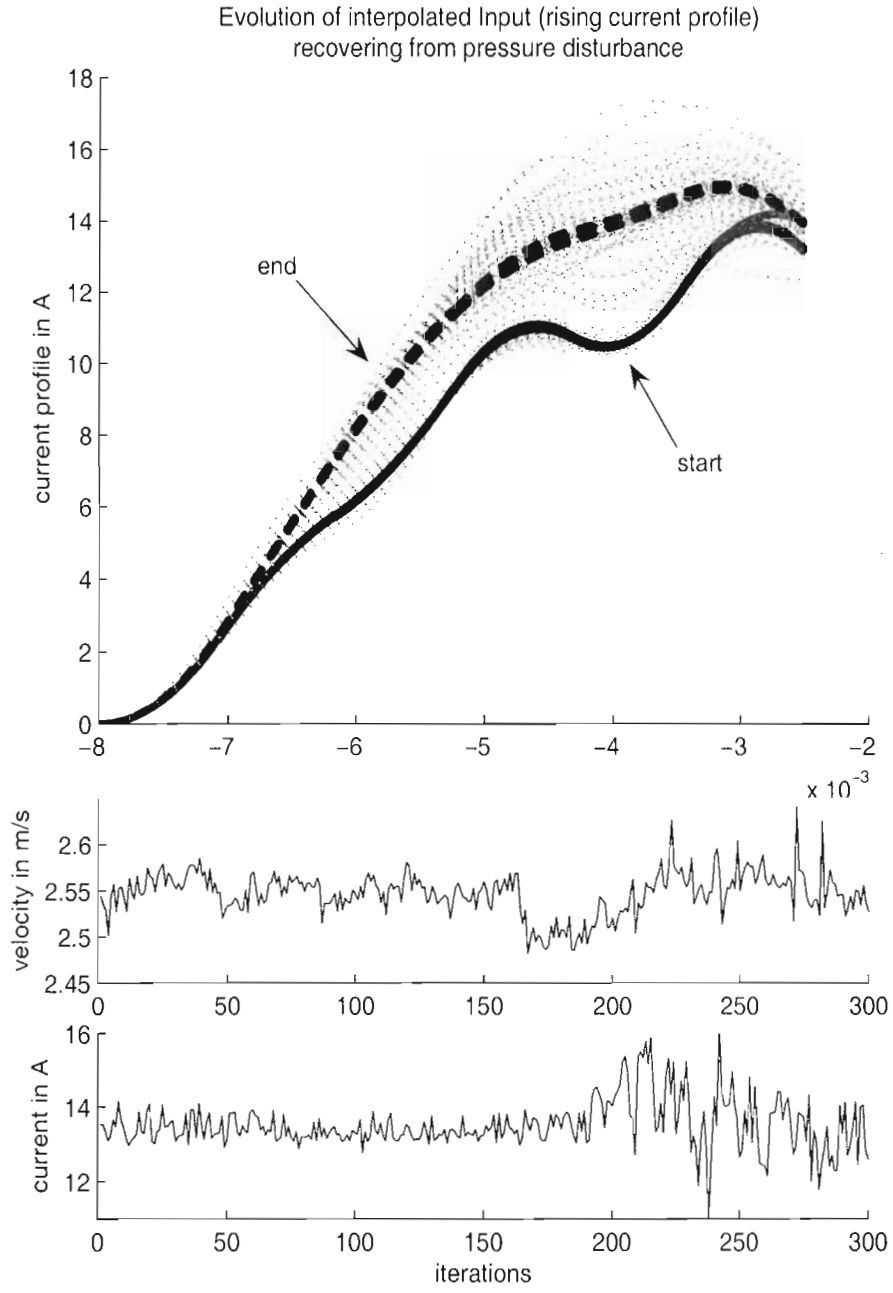

Figure 6.22: Evolution of the input current profile from the Nelder Mead controller regulating against a step pressure disturbance. The top plot is the current profile versus position from iteration 1 to 300. The bottom two plots are v^f_{appr} and i^f_{appr} versus iterations. The current profile is increased to compensate a step disturbance pressure force at 160th iteration.

6.6 Summary

The simulations performed in the controller algorithm section are tested experimentally in this section using the TEMIC electromagnetic actuator and dSpace 1103 board. Unfortunately, the Terminal ILC fails to converge because of the saturation issue and errors in feedback linearization. The results for the Nelder Mead algorithm fare much better and are similar to the simulated results in the theory chapter.

Overall, experimental tests for the Nelder Mead controller shows promising results. The temperature/parameter based disturbance was easily handled by the Nelder Mead controller. A properly weighted current integral added to the cost function results in a reduction in energy consumption without affecting the set-point regulation. The Nelder Mead controller demonstrated the ability reject disturbance pressure in step and ramp function. Last, but not least, the input current evolution is shown to be smooth and away from the saturation boundary. All the important tests and results in this chapter are summarized in table 6.1.

Table 6.1: Summary of experimental results

Experiments	**Nelder Mead controller performance**
Set-point change	Changes the set-point in 300 steps
Unknown disturbance	Reduces velocity variance to less than 1/3 compared to open-loop
Adding current integral into the cost function	Reduces current integral in an valve event from 16.9e-3 to 16.7e-3 ampere-second
Cold start performance	Reduces system warm up time from 700 iteration to 150 iterations
Step disturbance	Eliminates 40N step disturbance force in $100 \sim 150$ steps
Ramp disturbance	Rejects ramp disturbance force up to the rate of 10N per 100 steps

Chapter 7

Conclusion and Future Research

Camless engines equipped with electromagnetic solenoid valve actuators allows more flexibility in engine operation and can thus be optimized to achieve better torque characteristics, fuel economy, and emission control than conventional engines. However, the solenoid valve actuator suffers from hard landing impacts because of actuator nonlinearity, bandwidth limitations, and various environmental disturbances. Sophisticated controllers for the approach and landing stages of the valve flight are needed to avoid the unacceptable noise and component wear resulting from the large valve landing impact. This thesis investigated three cycle-adaptation based approach-controllers designed to provide consistent initial conditions for a subsequent solenoid valve landing controller.

7.1 Conclusion and Discussion

The following are the conclusions that can be drawn from the theoretical and experimental investigations of this thesis:

1. Pressure disturbance forces can be computed from the induced voltage in the release coil using an identified inverse model. The estimated pressure from validation data shows an average error less than 0.025 bar. This estimation method has the advantages of providing estimates early, having a high signal to noise ratio, and requiring little computational effort.

2. Because the EMV actuator has very limited control authority at larger airgap and is subjected to disturbances from engine back pressure, trajectory tracking for the

approach stage of the valve flight is virtually impossible. The inability to find a realizable trajectory to track leads to the poor simulation performance of the nonlinear iterative learning controller investigated in the theory chapter.

3. The terminal iterative learning controller is promising in terms of its convergence rate in simulations but more work needs to be done in feedback linearization and in constraining its coefficients. The application of the null-space gradient projection method does not provide enough control over the evolution of coefficients. The experimental results failed to converge because linearization failed, and the algorithm required current profile that is not physically realizable with a 42 volt power supply.

4. The Nelder Mead simplex algorithm, after sine coordinate transformation, can be applied to solve the approach control problem as a constrained nonlinear programming problem. Compared to the iterative learning and extremum seeking control, it has the advantages of not needing trajectory tracking and the ability to tune multiple parameters for greater optimization. In addition, it requires no actuator model and is efficient because it searches for the minimum without computing any derivative.

5. The experimental results of the Nelder Mead controller can be summarized by its performance against step, ramp, and temperature related disturbances. The controller eliminates 40N step disturbance in 100 to 150 steps, and is not effected by ramp disturbance below the rate of 10N per 100 steps. Compared to the openloop, it reduces the terminal velocity variance from 0.69 to 0.2 cm^2/s^2. In addition, adding a weighted current integral term into the cost function can reduce electrical energy consumption.

7.2 Future Research

The author proposes future research to be focused on the following areas:

1. A pressure chamber to supply disturbance pressure in the EMV test-bench setup is needed. It can be used to test both the proposed induced-voltage based pressure estimation and the Nelder Mead controller.

2. The estimates of terminal velocity and current at the end of the approach control

trajectory are rather noisy. This limits the convergence speed of cycle-adaptive controllers. Better interpolations and data smoothing schemes should be applied to improve these estimates.

3. The on-off current controller utilized in the experimental section is rather crude. If more work is done in the modeling, voltage-based feedback linearization may be applied to achieve much better current tracking and subsequently improve approach controller performance.

4. Terminal ILC would be much more effective if constraints and cost functions can be easily incorporated.

5. Nelder Mead is not the only direct search algorithm. Despite its advantages, its theoretical convergence conditions have not yet been found. Other direct search algorithms such as Rosenbrock and Pattern search should be investigated to see if they can perform better than the Nelder Mead algorithm on EMV actuator control.

6. Depending on how quickly disturbances change, the cost associated with a vertex can become outdated and should be replaced. One way to prevent this problem is to associate disturbance estimation information with the simplex vertices. If any earlier simplex vertex has disturbance estimate different from the rest, it should be replaced quickly. If this scheme is successful, the ramp disturbance rejection performance of the Nelder Mead controller should improve.

7. Approach controllers in this thesis are limited to using cycle-based compensation. However pressure disturbance can and does vary greatly between consecutive cycles (especially in the event of an engine misfire). More work should be done to combine the single -cycle-based approach controller with the cycle-adaptive approach controller. An ideal approach controller should combine the worst-case robustness of the single-cycle controller and the optimization capability of the cycle-adaption controller.

8. Most importantly, the EMV approach controller developed in this thesis should be combined with a landing controller such as the one developed by Chung [1] or Eyabi [41]. The landing impact reduction and disturbance rejection performance from the overall controller will be the ultimate measure of success for the work done in this thesis.

Appendix A

List of Program Files

A.1 Matlab Simulation Files

Table A.1: Nelder Mead controller simulation files

file name	file description
`RTNeaderMeader.m`	The Nelder Mead controller
`run_num_cost.m`	A function that calls Simulink model and computes cost
`emv_actuator.mdl`	Simulink actuator model based on [1], modified for Nelder Mead controller with additional blocks and scope to implement Terminal ILC control

Table A.2: Nonlinear ILC controller simulation files

file name	file description
`interpol_tanh.m`	Computes a desired trajectory for tracking
`nonlinear_ilc.m`	The Nonlinear ILC controller
`nilc.m`	A Matlab S-function that uses coefficient from ILC controller to generate current profile
`nilc_model.mdl`	Simulink actuator model based on [1], modified for nonlinear ILC with additional blocks and scope to implement Nonlinear ILC control

Table A.3: Terminal ILC controller simulation files

file name	file description
`Terminal_ILC.m`	Terminal ILC controller
`compute_voltage.m`	Performs voltage feedback linearization for the Terminal ILC controller
`bspline_itrpl.m`	Calculates interpolated current profile for the Terminal ILC controller
`tilc_emv_model.mdl`	Simulink actuator model based on [1], modified for Terminal ILC with additional blocks and scope to implement Terminal ILC control

Table A.4: Disturbance estimation files

file name	file description
`clbp0bar` ~ `clbp6bar`	Contains raw pressure data in Matlab ".mat" file format.
`DisturbanceID.sid`	The project file for Matlab's system identification toolbox for inverse model identification.
`mpr_cyl_plt.m`	Uses the seven pressure data sets `clbp0bar` ~ `clbp6bar` to showcase the effect of normalization on the data sets, and consequently the dependence of pressure forces on position.

A.2 Files for Experimentation

Table A.5: Terminal ILC controller C code

file name	file description
anschwing.c/h	Initiates the EMV control by swinging the armature to one of the coils from its middle resting position.
common.c/h	Defines all the important variables and structures shared by all files.
control.c/h	Interfaces with the tasks.c to allow different control methods to be called.
filter.c/h	Contains Kalman and moving average filter to provide velocity and acceleration information.
hw_inter.c/h	Initializes and communicates with the hardware to acquire measurements and execute voltage commands.
n_generator.c/h	Determines the frequency of the controller valve event.
ctrl_traj.c/h	Contains the actual approach controller using Terminal ILC algorithm and the landing controller using flatness-based control.
task_ctrl.c/h	Initiates the controller by registering the controller interrupts and the background tasks.
tasks.c	Defines the interrupt function that calls the approach and landing controllers.

Table A.6: Terminal ILC controller support lookup table

file name	file description
small_lw_map.txt	Lists desired current based on armature position and desired magnetic force (for position -0.37 to -0.28mm)
lw_map.txt	Lists desired current based on armature position and desired magnetic force (for position -0.28 to 4mm)
tilc_lg_table.txt	List Terminal ILC controller gain with armature swing duration as input.

Table A.7: Nelder Mead controller C code

file name	file description
anschwing.c/h	Initiates the emv control by swing the armature to the coil from its middle resting position.
common.c/h	Defines all the important variables and structures shared by all files.
control.c/h	Interfaces with the tasks.c to allow different control methods to be called.
filter.c/h	Contains Kalman and moving average filter to provide velocity and acceleration information.
hw_inter.c/h	Initializes and communicates with the hardware to acquire measurements and execute voltage commands.
n_generator.c/h	Determines the frequency of the controller valve event.
ctrl_traj.c/h	Contains the actual approach controller using Nelder Mead algorithm and the landing controller using flatness-based control (commented out)
task_ctrl.c/h	Initiates the controller by registering the controller interrupts and the background tasks.
tasks.c	Defines the interrupt function that calls the approach and landing controllers

Table A.8: ControlDesk Instrumentation files

file name	file description
control.lay	The panel that allows display and manual tuning of the controller parameters.
calibration.lay	The panel that calibrates sensors
anschwingen.lay	The panel that can be used to display and tune parameters for control initialization (swing the armature from middle to one of the coils)
itration_display	The panel that displays iteration-based terminal velocity and current.
plotters.lay	The panel that contains plots that display position and current plot in one valve motion.
fr.MK	The "Make" file for the entire controller file compilation
fr.trc	The trace file that keeps track of the parameters definition to be displayed and tuned on ControlDesk
ctrldesk_logging.py	The python logging script for storing values on the ControlDesk display panels
bk1_hand_flt7.par	The parameter file that stores values of parameters listed in fr.trc

Bibliography

[1] Soon K. Chung. Flatness-based end-control of a gas exchange solenoid actuator for internal combustion engines. Master's thesis, University of Alberta, 2005.

[2] Wolfgang Hoffmann, Katherine Peterson, and Anna G. Stefanopoulou. Iterative learning control for soft landing of electromechanical valve actuator in camless engines. *IEEE Trans on Control System Technology*, 11(2):174–184, 2003.

[3] K. Peterson and A. Stefanopoulou. Extremum seeking control for soft landing of an electromechanical valve actuator. *Automatica*, 2004.

[4] Ryan Chladny. Modeling and simulation of automotive gas exchange valve. Master's thesis, University of Alberta, 2003.

[5] M. Pischinger, W. Salber, F. van der Staay, H. Baumgarten, and H. Kemper. Benefits of the electromechanical valve train in vehicle operation. *SAE paper 2000-01-1223*, 2000.

[6] C. Sugimoto, H. Sakai, A. Umemot, Y. Shimizu, and H. Ozawa. Study on variable valve timing system using electromagnetic mechanism. *SAE paper 2004-01-1869*, 2004.

[7] P. Wolters, W. Salber, J. Geiger, and M. Duesmann. Controlled auto ignition combustion process with an electromechanical vale train. *SAE paper 2003-01-0032*, 2003.

[8] Alpha Romeo corp. 1.6 twin spark 16v, 2007. [Online; accessed 16-Nov-2007].

[9] Honda corp. The vtec breakthrough: solving century old dilemma, 2007. [Online; accessed 16-Nov-2007].

[10] Hot ribs online rigid hull inflatable boats magazine. Yamaha f350 outboard engine, 2007. [Online; accessed 16-Nov-2007].

[11] Asmus T. W. Valve events and engine operation. *SAE paper N0.800749*, 1982.

[12] Tuttle J. H. Controlling engine load byy means o flate intake-valve closing. *SAE paper N0.800794*, 1980.

[13] G.H. Abd-Alla. Using exhaust gas recirculation in internal combustion engines: a review. *Engergy Conversion and Management*, 43:1027–1042, 2002.

[14] A.G. Stefanopoulou and I. Kolmanovsky. Dynamic scheduling of internal exhaust gas recirculation systems. In *Proc. IMECE 1997, Sixth ASME Symposium on Advanced Automotive Technologies*, pages 671–678, 1997.

[15] M. Schechter and M. Levin. Camless engine. *SAE paper 960581*, 1996.

[16] H. Hong, G.B. Parvate-Patil, and B. Gordon. Review and analysis of variable valve timing strategies - eight ways to approach. *Proceedings of the Institution of Mechanical Engineers, Part D (Journal of Automobile Engineering)*, 218(D10):1179 – 200, October 2004.

[17] Z. Sun and D Cleary. Dynamics and control of an electro-hydraulic fully flexible valve actuation system. *Proceedings of the American Control Conference*, 4:3119–3124, 2003.

[18] Parlikar T., Chang W., Qiu Y., Seeman M., Perreault D., Kassakian J., and T. Keim. Design and experimental implementation of an electromagnetic engine valve drive. *IEEE/ASME Transactions on Mechatronics*, 10:482–494, 2005.

[19] C. Weddle and D. Leo. Embedded actuation system for camless engines. *Proceedings of the International Confrence on Adaptive Sturcture and Technologies*, 1998.

[20] Koch C. Lynch A. and Chladny R. Modeling and control of solenoid valves for internal combustion engines. *2nd IFAC Conference On Mechatronic Systems*, 41:317–322, 2002.

[21] Katherine S. Peterson and Anna G. Stefanopoulou. Rendering the electromechanical valve actuator globally asymptotically stable. *Proc. 2003 IEEE Conference on Decision and Control*, 2003.

[22] S. Butzmann, J. Melbert, and A. Koch. Sensorless control of electromagnetic actuators for variable valve train. *SAE paper 2000-01-1225*, 2000.

[23] C. Tai and T. Tsao. Control of an elctromechanical actuator for camless engines. *Proc. 2003 American Control Conference*, 2003.

[24] S.K. Chung, C.R. Koch, and A.F. Lynch. Flatness-based feedback control of an automotive solenoid valve. *IEEE Transactions on Control Systems Technology*, 15(2):394 – 401, March 2007.

[25] R. R. Chladny and C. R. Koch. Flatness-based tracking of an electromechanical vvt actuator with disturbance observer feed-forward compensation. IEEE Transactions on Control Systems Technology, accepted Aug 2007, 2007.

[26] K. Longstaff and S. Holmes. Internal combustion engines, 1975. U.S. Patent No. 3,882,833.

[27] F. Pischinger and P. Kreuter. Electromagnetically operating actuator, 1984. U.S. Patent No. 4,455,543.

[28] Bruno Lequesne. Fast-acting, long-stroke solenoids with two springs. *IEEE Transactions on Industry Applications*, 26(5), September/October 1990.

[29] David C. Jiles. *Introduction to magnetism and magnetic materials.* CRC Press, 2nd edition, 1998.

[30] R. E. Clark, G. W. Jewell, S. J. Forrest, and C. Maerky. Design features for enhancing the performance of electromagnetic valve actuation system. *IEEE Transactions on Magnetics*, 41(3), march 2005.

[31] R. Chladny, C. Koch, and A. Lynch. Modeling automotive gas-exchange solenoid valve actuator. *IEEE trans. on Magnetics*, 41(3):1155–1162, 2005.

[32] Piron M., P. Sangha, G. Reid, T.J.E. Miller, and D.M. Ionel. Rapid computer-aided design method for fast-acting solenoid actuators. *IEEE Transactions on Industry Applications*, 35(5):991–999, Sep/Oct 1999.

[33] C. Chillet and J.Y. Voyant. Design-oriented analytical study of a linear electromagnetic actuator by means of a reluctance network. *IEEE Transactions on Magnetics*, 37(4):3004–30111, July 2001.

[34] Klaus Gebauer Christoph Hartwig, Olaf Josef. Transients of electromagnetic valve train (emvt) actuator. *SAE paper 2004-01-1388*, 2004.

[35] Yan Wang, Anna Stefanopoulou, Mohammad Haghgooie, Ilya Kolmanovsky, and Mazen Hammoud. modeling of an electromechanical valve actuator for a camless engine. In *Proceedings 5th Int'l Symposium on Advanced Vehicle Control*, August 2000.

[36] J.Y. Xiang. Modeling and control of a linear electro-mechanical actuator (lema) for operating engine valves. In *Conference Record of the 2002 IEEE Industry Applications Conference. 37th IAS Annual Meeting (Cat. No.02CH37344)*, volume vol.3, pages 1943 – 9, Pittsburgh, PA, USA, 2002.

[37] Y. Wang, T. Megil, M. Haghgooie, K. Peterson, and A. Stefanopoulou. Modeling and control of electromechanical valve actuator. *SAE paper 2002-01-1106*, 2002.

[38] K. Peterson and A. Stefanopoulou. Current versus flux in the control of electromechanical valve actuators. In *Proceedings of 2005 American Control Conference*, Portland, Oregan, USA, June 2005.

[39] S. Di Gennaro, B. C. Toledo, and M. D. Di Benedetto. Nonlinear regulation of electromagnetic valves for camless engine. In *Proceedings of the 45th IEEE Conference on decision and control*, San Diego, CA, USA, 2006.

[40] C. Tai and T. Tsao. Control of an elctromechanical actuator for camless engines. *Proc. 2002 American Control Conference*, 2002.

[41] Peter Eyabi and Gregory Washington. Modeling and sensorless control of an electromagnetic valve actuator. *Journal of Mechatronics*, 16:159–175, 2006.

[42] K. S. Peterson, J. W. Grizzle, and A. G. Stefanopoulous. Nonlinear control for magnetic levitation of automotive engine valves. *IEEE Transactions on Control System Technology*, 14(2), March 2006.

[43] C. Tai, Andrew Stubbs, and T. Tsao. Control of an elctromechanical actuator for camless engines. *Proc. 2001 American Control Conference*, 2001.

[44] Ch. Gunselmann and J. Melbert. Improved robustness and energy consumption for sensorless electromagnetic valve train. *SAE paper 2003-01-0030*, 2003.

[45] F. L. Lewis and V. L. Syrmos. *Optimal control.* Wiley Interscience, 2nd edition, 1995.

[46] Katherine Peterson, Anna Stefanopoulou, Tom Megli, and Mohammad Haghgooie. Output observer based feedback for soft landing of electromechanical camless valvetrain actuator. In *Proceedings of 2002 American Control Conference*, Anchorage, AK, May 2002.

[47] Peter B. Eyabi. *Modeling and sensorless contrl of solenoidal actuators.* PhD thesis, Ohio State Universtiy, 2003.

[48] Katherine S. Peterson. *Control Methodologies for Fast and Low Impact Electromagnetic Actuators for Engine Valves.* PhD thesis, University of Michigan, Ann Arbor, 2005.

[49] E.D. Sontag. A universal construction of artstein's theorem on nonlinear stabilization. *Syst. Control Lett. (Netherlands)*, 13(2):117 – 23, August 1989.

[50] İ. Haskara, V. Kokotovic, and L. Mianzo. Control of an electro-mechanical valve actuator for a camless engine. *International Journal of Robust and Nonlinear Control*, 14:561–579, 2004.

[51] D.A. Bristow, M. Tharayil, and A.G. Alleyne. A survey of iterative learning control. *IEEE Control Systems Magazine*, 26(3):96 – 114, June 2006.

[52] Kevin L. Moore. Iterative learning control: An expository overview. *Applied and Computational Controls, Signal Processing, and Circuits*, 1(1), 1998.

[53] Hassan K. Khall. *Nonlinear Systems.* Prentice Hall, 3rd edition, 2002.

[54] Kartik B. Ariyur, Miroslav Krsti, and Cacute. *Real-Time Optimization by Extremum-Seeking Control.* Wiley, 1st edition, September 2003.

[55] Michael Quong. System identification and end control of an electromagnetic valve actuator. work term report, University of Alberta, 2004.

[56] Ryan R. Chladny. *Modeling and Control of Automotive Gas Exchange Valve Solenoid Actuator*. PhD thesis, University of Alberta, 2007.

[57] Mingxuan Sun, Danwei Wang, and Youyi Wang. Sampled-data iterative learning control with well-defined relative degree. *Int. J. Robust Nonlinear Control (UK)*, 14(8):719 – 39, May 2004.

[58] Jian-Xin Xu, Yangquan Chen, Tong Heng Lee, and S. Yamamoto. Terminal iterative learning control with an application to rtpcvd thickness control. *Automatica*, 35(9):1535 – 42, Sept. 1999.

[59] G. Oriolo, S. Panzieri, and G. Ulivi. Learning optimal trajectories for nonholonomic systems. *International Journal of Control*, 73(10):980 – 91, July 2000.

[60] G. Oriolo, S. Panzieri, and G. Ulivi. An iterative learning controller for nonholonomic mobile robots. *International Journal of Robotics Research*, 17(9):954 – 70, Sept. 1998.

[61] David F. Rogers. *An Introduction to NURBS With Historical Perspective*. Academic Press, 2001.

[62] Giuseppe Oriolo, Stefano Panzier, and Giovanni Ulivi. Learning optimal trajectories for non-holonomic systems. *Int. J. Control*, 73(10):980–991, 2000.

[63] Mokhtar S. Bazaraa, Hanif D. Sherali, and C. M. Shetty. *Nonlinear Programming: Theory and Algorithms*. Wiley, 2006.

[64] J.C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright. Convergence properties of the nelder-mead simplex method in low dimensions. *SIAM Journal of Optimization*, 9(1):112–147, 1998.

[65] R.M. Lewis, V. Torczon, and M.W. Trosset. Direct search methods: then and now. *J. Comput. Appl. Math. (Netherlands)*, 124(1-2):191 - 207, December 2000.

[66] John D'Errico. Understanding fminsearchbnd. made availabe on MATLAB Central File Exchange, 2005.

[67] J.A. Nelder and R. Mead. A simplex method for function minimization. *The Computer Journal*, 7:308 313, 1965.

[68] William Press, Saul Teukolsky, William Vetterling, and Brian Flannery. *Numerical Recipes in C*. Cambridge University Press, Cambridge, UK, 2nd edition, 1992.

[69] E. Polak. *Computational methods in optimization: A unified approach*. Academic Press, 1971.

[70] K. I. M. McKinnon. Convergence of the nelder-mead simplex method to a nonstationary point. *SIAM Journal on Optimization*, 7:148–158, 1998.

[71] H. Roters. *Electromagnetic Devices.* John Wiley and Sons, 1941.

[72] J. G. Kassakian, H.C. Wolf, J. M. Miller, and C. J. Hurton. Automotive electrical systems circa 2005. *IEEE Spectrum*, pages 22–27, 1996.

[73] W. J. Mooney. *Optoelectronic Devices and Principles.* Prentice-Hall, New Jersey, 1981.

[74] Clarence W. de Silva. *Sensor and actuators.* CRC Press, Boca Raton, FL, 2007.