

SHAPE MEMORY ALLOY TORQUE TUBE DESIGN OPTIMIZATION FOR
AIRCRAFT FLAP ACTUATION AND CONTROL

A Thesis

by

JOHN LUKE ROHMER

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Chair of Committee,	James G. Boyd
Co-Chair of Committee,	Dimitris C. Lagoudas
Committee Members,	Darren Hartl
	Ibrahim Karaman
Head of Department,	Rodney Bowersox

December 2016

Major Subject: Aerospace Engineering

Copyright 2016 John Luke Rohmer

ABSTRACT

Shape memory alloys comprise a unique class of material that is able to undergo a thermally driven, solid-solid phase change. This transformation is characterized macroscopically by the generation of large inelastic strains which may be recovered while supporting significant load. This process can be harnessed to do useful work as an actuator, and indeed, shape memory alloys possess one of the greatest actuation work densities of all active materials. It is because of this that researchers and engineers are interested in using these alloys to create powerful, lightweight actuators for several aerospace applications. In current aircraft designs, hydraulic systems represent a large proportion of the total aircraft mass. However, shape memory alloy torque tubes may provide a lightweight alternative. This thesis documents research done to study and optimize the structural design and PID controller parameters of an inductively heated shape memory torque tube providing feedback control of the aircraft control surfaces. The system electro-thermomechanical response under variable loading is modeled and implemented in Python. The Design of Experiments methodology is utilized to identify important design parameters. Finally, the structural and control design space is explored using particle swarm optimization to achieve an optimum PID controller response. Experiments are used to calibrate the SMA constitutive model and to validate the time-domain control response simulation. It was found that this method is a viable solution for designing SMA torque tubes for use as aircraft control surface actuators.

“Show up. Work hard. Be kind. Take the high road.”

–George Meyer

ACKNOWLEDGEMENTS

I wish to thank those many people and organizations who made this work possible. First and foremost, my advisors, Dr. Boyd and Dr. Lagoudas, and my committee members, Dr. Hartl and Dr. Karaman, whose guidance, encouragement and occasional splash of cold water have proved time and time again to be invaluable to my research and professional development. Jim Mabe and Dr. F. Tad Calkins of The Boeing Company, for their support and loan of test samples and equipment, as well as extensive technical input on this project. Ceylan Hayrettin, for his assistance in setting up the experiments, and finally, to all those who have contributed to this work and influenced my time at Texas A&M. This includes Parikshith Kumar who first trained me to be an experimentalist, Rob Wheeler, Chris Bertagne, Dr. Majid Tabesh, Dr. Theocharis Baxevanis, Mahdi Mohajeri, Chris Calhoun, Hande Ozcan, Nathan Key, Edwin Peraza, Behrouz Haghgouyan, Sameer Jape, Dr. Parikshith Kumar, Ken Cundiff, Keegan Colbert, Rodney Inmon, Dr. Amine Benzerga, Dr. Mohammad Naraghi, Dr. John Valasek, Alex Solomou, Bonnie Reid and Ashley Brown.

Finally, thanks to my parents, Dwayne and Cheryl Rohmer, my sister Christina Rohmer and my grandmother Agnes Rohmer, for their ongoing love and support.

TABLE OF CONTENTS

	Page
ABSTRACT	ii
ACKNOWLEDGEMENTS	iv
TABLE OF CONTENTS	v
LIST OF FIGURES	vii
LIST OF TABLES	x
1. INTRODUCTION	1
1.1 Aerodynamic Loading	3
1.2 Methods of Heating/Cooling of Shape Memory Alloys	6
1.2.1 Methods of Heating	6
1.2.2 Induction Heating	7
1.2.3 Methods of Cooling	8
1.3 Proportional-Integral-Derivative Control Method	9
1.4 Design Optimization	10
1.5 Outline of Thesis	12
2. GOVERNING EQUATIONS	14
2.1 Constitutive Modeling of Shape Memory Alloys	14
2.1.1 Reduced Constitutive Model for 1-D Torsion	19
2.1.2 Minor Hysteresis Loop Modification of the SMA Constitutive Model	20
2.2 Series Equivalent Circuit Modeling of Induction Heating	21
2.2.1 Skin Depth	23
2.2.2 Power Applied to the Torque Tube	24
2.2.3 Electrical Impedance	25
2.2.4 Electrical Efficiency and Power Factor	27
2.3 System of Differential Equations in Time	28
2.3.1 Mechanical Equilibrium	28
2.3.2 First Law of Thermodynamics	30
2.4 Proportional-Integral-Derivative (PID) Control Law	31
3. COMPUTATIONAL MODELING AND OPTIMIZATION	35

3.1	Design of Experiments	36
3.2	Particle Swarm Optimization of Design and Control Response	36
3.2.1	Optimization Problem Description	42
4.	EXPERIMENTAL DESIGN	43
4.1	Electromagnetic Characterization: Impedance Measurement	43
4.2	Complete System Testing with PID Control and Realistic Loading	44
5.	RESULTS	48
5.1	Experimental Calibration of Computational Model	48
5.1.1	Thermal Calibration	48
5.1.2	SMA Calibration	49
5.2	Calibrated Computational Results	52
5.3	Computational and Experimental Results for Determination of Electrical Properties	52
5.4	Computational and Experimental Results for PID Control of an SMA Torque Tube	54
5.5	Controller Optimization Problem	59
5.6	Complete Optimization Problem	63
5.7	Optimization Discussion	68
6.	SUMMARY	69
APPENDIX A. PARTICLE SWARM OPTIMIZER WITH SMA CONSTITUTIVE MODEL IN PYTHON		79

LIST OF FIGURES

FIGURE	Page
1.1 Shape Memory Alloy (SMA) phase diagram.	2
1.2 Two way shape memory effect	3
1.3 3/8" Nitinol torque tubes used for control experiments.	4
1.4 Schematic description of the SMA torque tube experiment.	4
1.5 Illustration of a plain flap configuration.	5
1.6 Theoretical hinge moment coefficients of a plain trailing-edge flap [3].	6
1.7 Illustration of global and local optima.	11
2.1 Effect of partial loop modification on the behavior of a single isobaric cycle.	21
2.2 Effect of partial loop modification on the behavior of multiple isobaric cycles.	22
2.3 Homogeneous conducting annular space with sinusoidal current. The amplitude of the current density vector at an instant of time versus the radial position, r , is indicated in blue.	24
2.4 Conceptual drawing of the inductively heated torque tube as modeled with linear loading on the free end.	26
2.5 Equivalent electrical circuit used in the development of the induction heating model.	27
2.6 Schematic drawing of thermomechanical loading path on SMA phase diagram.	30
2.7 Mapping of the controller output to the heating and cooling process inputs.	32
2.8 Pulse Width Modulation (PWM) carrier wave with control function. .	33

2.9	Comparison of the rotation and convection coefficients between the continuous control methodology (Left) and the PWM methodology (Right) in the time domain.	34
3.1	Single loop Python script algorithm.	37
3.2	Python script algorithm with PID control.	38
3.3	Example SMA response of complete cycle implementation.	39
3.4	Example command function (green) and simulated response (blue).	39
3.5	Initial DoE results from JMP statistical analysis software.	40
4.1	Nitinol torque tube with induction coil.	45
4.2	LabVIEW control panel.	46
4.3	Controlled torque tube experimental setup.	47
5.1	Heating and cooling curves for calibration of the forced convective cooling coefficient.	49
5.2	Heating and cooling curves for calibration of the natural convective cooling coefficient.	50
5.3	Temperature rotation response of the SMA torque tube at various load levels.	50
5.4	Measurement of SMA torque tube actuation modulus. Courtesy of The Boeing Company.	51
5.5	Temperature-rotation from computational model with 2.5 N-m applied moment.	52
5.6	Temperature-rotation partial cycle from computational model with 2.5 N-m applied moment without the minor loop modification.	53
5.7	Temperature-rotation partial cycle from computational model with 2.5 N-m applied moment and with the minor loop modification.	53
5.8	Simulated rotation for sine wave input.	55
5.9	Simulated PID controller output for a sine wave input.	55
5.10	Control and measured rotations for a sine wave input.	56

5.11	LabVIEW PID controller output for a sine wave input.	56
5.12	Simulated rotation for square wave input.	57
5.13	Simulated PID controller output for a square wave input.	57
5.14	Control and measured rotations for a square wave input.	58
5.15	LabVIEW PID controller output for a square wave input.	58
5.16	Control optimization convergence and simulated rotation.	60
5.17	Progression of the value of local best objective function for each particle during controller optimization.	61
5.18	Global best solution controlled thermomechanical response following controller optimization.	62
5.19	Convergence of structural and electrical properties.	64
5.20	Convergence of control parameters and pulse width modulation carrier frequency.	65
5.21	Progression of the value of the local best objective function for each particle during complete system optimization.	66
5.22	Global best solution controlled thermomechanical response following complete system optimization.	67

LIST OF TABLES

TABLE	Page
1.1 Potential heating and cooling methods for shape memory alloys actuator components.	7
3.1 Design points used in DoE study.	36
5.1 Dimensions and published properties of SMA torque tube.	48
5.2 Calibrated properties of Nitinol torque tube.	51
5.3 Measured torque tube electrical properties.	54
5.4 Experimental and simulated control parameters.	54
5.5 Controller particle swarm optimization initial seed ranges and lower bounds.	59
5.6 Controller particle swarm optimization best solution.	60
5.7 Complete particle swarm optimization initial seed ranges and lower bounds.	63
5.8 Complete particle swarm optimization best solution.	64

1. INTRODUCTION

Developments in materials and structures technology have been responsible for many performance improvements in aerospace systems during recent decades. Furthermore, these improvements have come while meeting the challenges and ever more stringent requirements of reliability, performance and cost effectiveness [47]. In recent years, the aerospace industry has found a renewed interest in returning to structural techniques originally pioneered by the Wright Brothers, namely morphing aerostructures [67]. Morphing aerostructures, also known as smart, active or reconfigurable structures, can be used to dynamically manipulate the aircraft flight characteristics to optimize performance for changing operating conditions [15, 16, 67, 71]. Numerous implementations of these smart structures have been proposed for both aircraft and spacecraft including morphing thermal radiators, flap actuators, noise reduction devices, solar sail deployment, orbital release mechanisms, re-configurable rotor blades, and deployable rotor blade devices [10, 15, 16, 25, 33, 39, 43, 47]. While many more applications for smart materials exist, each of these listed have in common that they employ shape memory alloy (SMA) components.

SMA's are a unique class of structural material which are capable of recovering apparently permanent deformations of up to 10% through a solid-solid, diffusionless phase change which enables their use in adaptive structures, motors and actuators [13, 55]. This property is manifested in two different thermomechanical processes, pseudoelasticity and the Shape Memory Effect (SME). These are initiated when the stress-temperature state of the system crosses the transformation boundaries shown in Figure 1.1. The SME is of interest in this work which, as shown in Figure 1.2, is characterized by the generation of inelastic strain under load at tem-

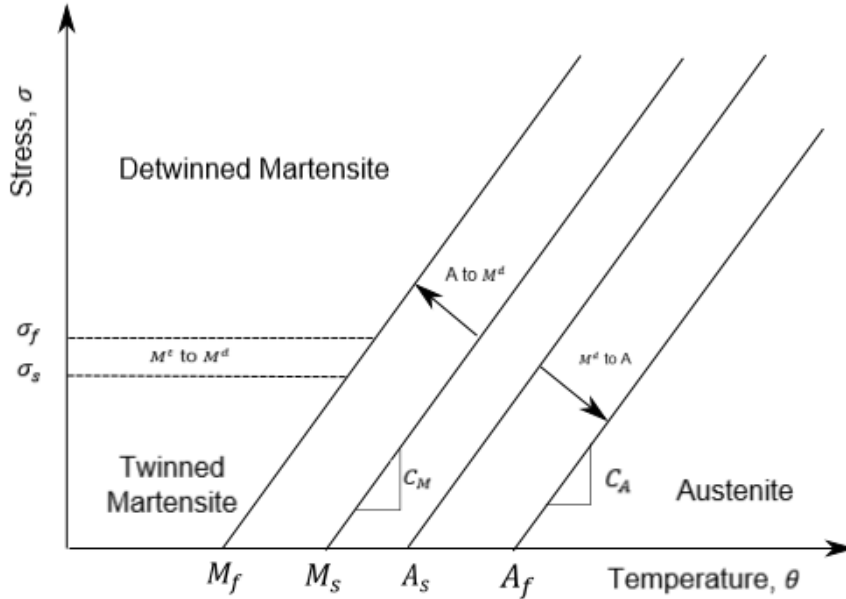


Figure 1.1: Shape Memory Alloy (SMA) phase diagram.

peratures below the reverse transformation threshold. This transformation strain is then recovered through a thermally induced transformation of the material from martensite into austenite [39, 57].

Several of the SMA applications mentioned above are implementations of two-way, SMA torque tubes. This type of actuator is of particular interest for applications such as rotor blade/wing twisting and flap/aileron deployment where size and weight are an issue [25, 33, 43]. A common feature of these applications is the presence of spring or aerodynamic loads which must be included in any simulation predicting the SMA response in these applications. This aspect is notably absent in the existing body of literature for this problem [22].

The Boeing Company has identified several issues presently impeding the aerospace implementation of SMA technologies. These include, among other things, the im-

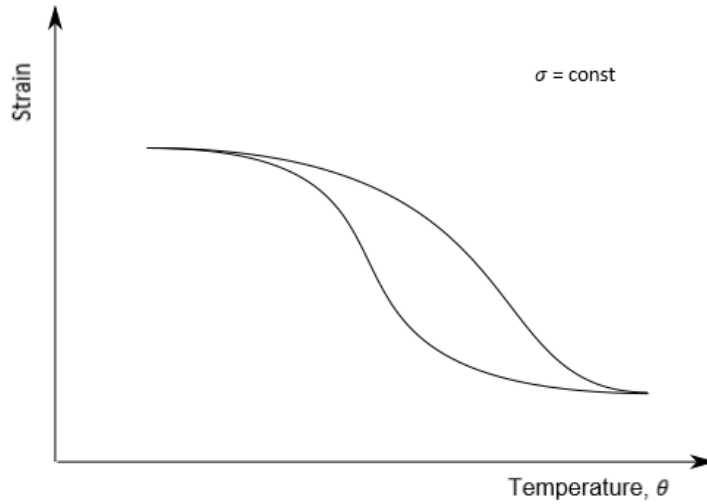


Figure 1.2: Two way shape memory effect

plementation of SMA actuators into otherwise passive structures (i.e. composites), development of improved computational tools for conceptual development and design optimization [15, 16]. It is the goal of this thesis to help address these concerns for SMA torque tube actuators such as those shown in Figure 1.3.

The SMA torque is assumed, for this work, as being thin walled, fixed on one end, having a uniform temperature, θ , end rotation, ϕ and applied moment, T , as shown in Figure 1.4.

1.1 Aerodynamic Loading

The salient application in this work is the actuation of aircraft control surfaces. As the SMA material behavior is stress dependent, the hinge moment must be characterized. Classical aerodynamics theory provides a model for the hinge loading on plain flaps such as that shown in Figure 1.5 under the assumption that the airflow does not separate from the wing surface [3].



Figure 1.3: 3/8" Nitinol torque tubes used for control experiments.

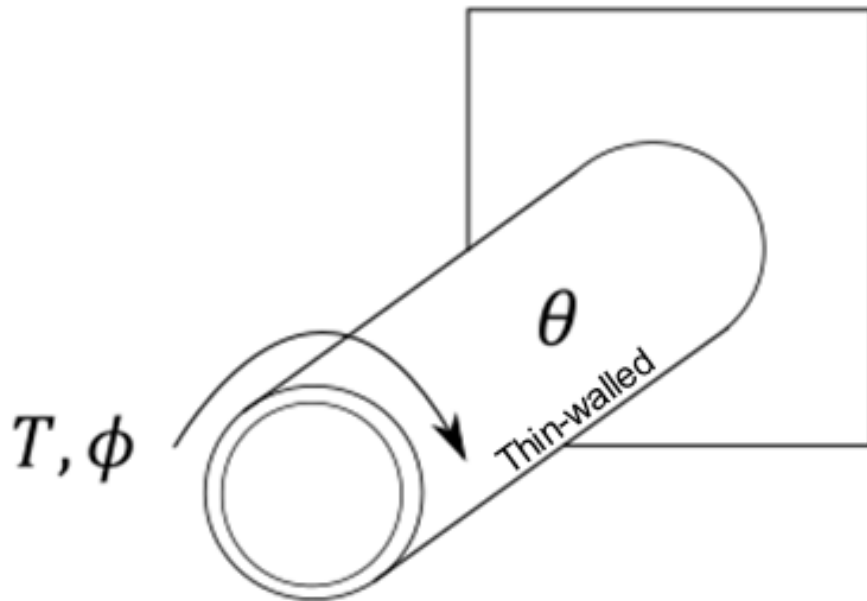


Figure 1.4: Schematic description of the SMA torque tube experiment.



Figure 1.5: Illustration of a plain flap configuration.

The section aerodynamic hinge moment may be written as

$$T_{aero,sec} = \frac{1}{2}\rho_{\infty}V_{\infty}^2c_f^2c_h \quad (1.1)$$

where

$$c_h = c_l \left(\frac{dc_h}{dc_l} \right) + \delta \left(\frac{dc_h}{d\delta} \right), \quad (1.2)$$

V_{∞} and ρ_{∞} are the speed and density of the oncoming airflow, c_f is the chord length of the flap, δ is the flap angle relative to the airflow and $\frac{dc_h}{d\delta}$ is the hinge moment coefficient as defined in Figure 1.6. Additionally, c_l is the section lift coefficient of the wing and $\frac{dc_h}{dc_l}$ is the rate of change of hinge moment with coefficient of lift [3].

Given the full width of the aileron and neglecting the lift portion of the hinge moment coefficient, the moment applied to the aileron hinge, comprised of the SMA torque tube, T_{aero} , may then be approximated as

$$T_{aero} = \frac{dc_h}{d\delta} \frac{1}{2}\rho_{\infty}V_{\infty}^2S_f c_f \delta \quad (1.3)$$

where S_f is the area of the flap.

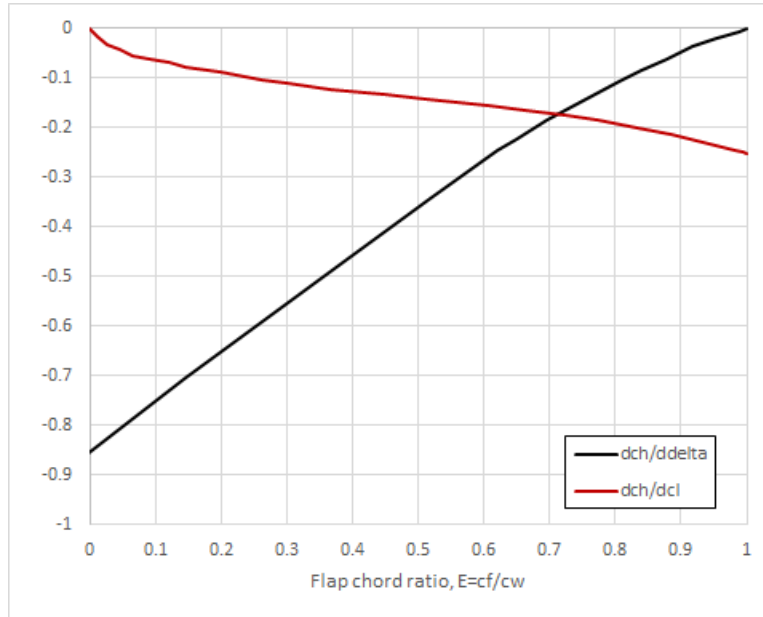


Figure 1.6: Theoretical hinge moment coefficients of a plain trailing-edge flap [3].

1.2 Methods of Heating/Cooling of Shape Memory Alloys

The maximum frequency of cyclic actuation of Nitinol is limited by the fact that shape memory behavior is driven by a controlled change in temperature and Nitinol has a low thermal conduction coefficient [47, 51]. Because of this, methods of quickly introducing and rejecting thermal energy must be considered. Methods of heating and cooling Nitinol actuators have been thoroughly explored in the literature and are summarized in Table 1.1.

1.2.1 Methods of Heating

Of the methods of heating listed in Table 1.1, methods which generate body heat, such as electromagnetic induction heating or direct application of current, are now seen as an enabling technology for the utilization of SMA torque tube actuators in certain applications as compared to methods where heat is applied via conduction

Table 1.1: Potential heating and cooling methods for shape memory alloys actuator components.

Heating	Cooling
Direct resistive[14, 30, 48, 55, 68]	Free convection (air)[55, 70]
Capacitance-assisted resistive[51, 55]	Liquid immersion[21, 55]
Conductive[8, 26, 55]	Forced air/liquid convection[55, 70]
Convective heating [21, 55]	Peltier effect[55, 62]
Radiative (including laser)[31, 44, 55]	Heat sinking[8, 55]
Induction heating[55, 71]	<i>Cool Chips</i> technology[55, 59]
Chemical fuel[50, 64]	

at the surface [56]. The direct application of electric current for heat generation and induction heating each have benefits and drawbacks. Direct resistance heating is a more efficient method of heating, however induction heating is somewhat more flexible in its application in terms of geometry of the object being heated and maximum power and has been selected for this study [42].

1.2.2 Induction Heating

In induction heating, an alternating current is applied to a coil wrapped around the body being heated, generating an alternating electromagnetic field. This field generates eddy currents within the enclosed body [19]. These currents create a Joule heating effect within the actuator itself. The resulting body heat is capable of increasing the Nitinol temperature rapidly and evenly if the induction heater is properly configured. Two electrical models for studying induction heating which have been proposed are the series equivalent circuit (SEC) model, such as that proposed by [6] which is based on a series electric circuit equivalency to the magnetic circuit [28] and the transformer equivalent circuit (TEC). Additional methods using finite element analysis (FEA) also exist. Experimental comparison of these methods shows that the FEA method provides the most accurate results of these choices. However, the

closed form analytical expressions of the SEC and TEC solutions allow them to be used, if not as definitive performance computations, as easily programmed and applied design guides [65]. The SEC model is selected for this work and is described in detail in the next section.

1.2.3 Methods of Cooling

Forced convective cooling is selected to be the primary mode of heat transfer driving forward transformation of the torque tube. It is defined by

$$\dot{Q}_{out} = -hA(\theta - \theta_{\infty}) \quad (1.4)$$

where A is the area of the inside surface of the torque tube [7].

All heat transfer is assumed to be through forced convection to a fluid flowing through the inside of the tube. Therefore the effective convection coefficient, h , is a function of the velocity of the fluid, the overall geometry of the system and the condition of the surrounding environment.

An important consideration for forced convective cooling of the type described is the selection of the fluid to be used and its velocity. It is known in literature that the convection coefficient depends on whether the flow can be classified as laminar or turbulent.

This classification is determined by the Reynolds number, Re .

$$Re = \frac{DV\rho}{\mu} = \frac{DV}{\nu} \quad (1.5)$$

In this context, D is the inner diameter of the torque tube, V is the average velocity of the fluid, ρ is the density of the fluid, μ is the dynamic viscosity of the fluid and $\nu = \frac{\mu}{\rho}$ is the kinematic viscosity [17, 27, 61]. In commercial tubing, pipe flow is

typically laminar where $Re \leq 2300$ and turbulent for $Re \geq 4000$. In the interim, the flow is in transition and cannot be determined [17, 7, 61].

In this work, the effective convection coefficient will be characterized experimentally using the First Law of Thermodynamics given by

$$hA(\theta - \theta_\infty) = -mc \frac{d\theta}{dt} \quad (1.6)$$

under the assumptions of a constant material volume and mass, that the only heat being transferred to or from the material is due to convective cooling and that the torque tube has a homogeneous temperature distribution. This differential equation has the solution

$$\theta(t) = \theta_\infty + e^{-\frac{hA}{cm}t} (\theta_0 - \theta_\infty). \quad (1.7)$$

By curve fitting this solution to the experimentally measured cooling process with and without active cooling, the natural and forced convection coefficients may be determined.

1.3 Proportional-Integral-Derivative Control Method

Proportional-Integral-Derivative (PID) control is a well known process control method with widespread implementation due to its robust performance across many applications and simple formulation [23, 34]. This function is defined by proportional, integral and derivative terms [34] which allow the controller to respond to various aspects of the system performance [34].

There exist two common formulations of the PID control function. The first is defined in the time domain by

$$\aleph(t) = k_P \epsilon|_t + k_I \int_{t_0}^t \epsilon dt + k_D \frac{d\epsilon}{dt}|_t \quad (1.8)$$

where ε is defined as the difference between the simulated rotation angle and the targeted angle at time, t . The PID control law is defined as a three-term function of ε . The second form is defined in the parallel time-domain by

$$\aleph(t) = k_P \left(\varepsilon|_t + \frac{1}{T_I} \int_{t_0}^t \varepsilon dt + T_D \frac{d\varepsilon}{dt} \Big|_t \right), \quad (1.9)$$

which is an easily derived form of the first formulation.

The parallel form is commonly used in industry and indeed is utilized by the LabVIEW controller implemented in the later described experiments. This formulation uses an integral time constant, $T_I = \frac{k_P}{k_I}$ and a derivative time constant $T_D = \frac{k_D}{k_P}$ [32, 34].

PID control contrasts with the nonlinear "Bang-Bang" control method in which, for single mode control, a high state and a low state are selected between based on the satisfaction of a control criterion [9, 58].

$$u(t) = \begin{cases} u_{max} & \text{if } f(t) > f_{crit} \\ u_{min} & \text{if } f(t) < f_{crit} \end{cases} \quad (1.10)$$

1.4 Design Optimization

Optimization has been defined as "(1) a systematic change, modification, adaptation of a process that aims to (2) achieve a pre-specified purpose" [46]. There exist several important aspects in any optimization process. These include the design parameters which may be varied by the designer, the objective function which is used to evaluate or "score" the particular configuration of design parameters, and the design constraints which limit the space of possible parameter combinations to those which will result in a physically permissible, acceptable response [53, 46, 60]. Each of these

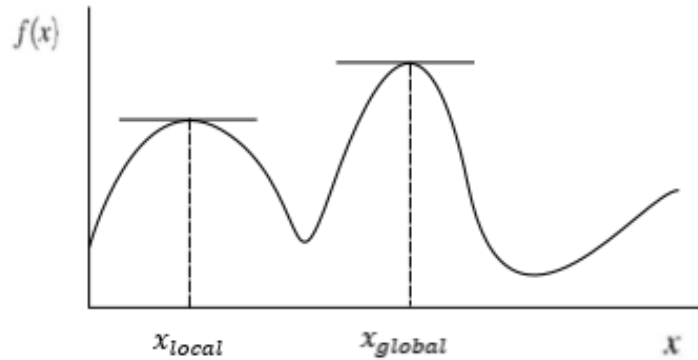


Figure 1.7: Illustration of global and local optima.

components of the optimization must be considered and defined. An example of a constraint is the maximum stress which must not be exceeded at the design loading.

Optimization algorithms may be divided into three broad categories, meta-heuristic, gradient based and direct-search techniques [72]. The direct-search techniques such as that proposed by [29] may be used for problems with a discontinuous or non-differentiable solution space but are typically relegated to niche applications [2, 38]. The remaining choice is between gradient based and meta-heuristic algorithms.

A common feature of optimization problems is the presence of local as well as global optima [52]. While gradient methods often have the tendency to become trapped in local optima, such as that shown as x_{local} in Figure 1.7, heuristic methods often contain mechanisms by which the optimizer may escape from local optima in search of better solutions [35, 73].

The Particle Swarm method of Optimization (PSO) is a heuristic technique originally proposed by Kennedy and Eberhart in 1995 as a method for solving problems in both engineering and the behavioral sciences [36]. It is simple to implement and exhibits robust behavior. Because of its simplicity, heuristic nature and history of

use in electromagnetic applications, this optimization method was chosen over other options such as the genetic algorithms [54].

One method of accelerating and improving the design optimization process is the use of the Design of Experiments (DoE) procedure. DoE process is a statistical method by which important parameters in a design process may be identified and unimportant parameters eliminated from focus. It also identifies interaction effects which may impact the design process [5, 18, 69]. In the most general sense, NIST defines design of experiments as “a systematic, rigorous approach to engineering problem-solving that applies principles and techniques at the data collection stage so as to ensure the generation of valid, defensible, and supportable engineering conclusions. In addition, all of this is carried out under the constraint of a minimal expenditure of engineering runs, time, and money” [1]. DoE processes differ from the traditional scientific method by varying multiple variables simultaneously in order to accelerate the process and gain information on the interaction of parameters [45]. In the present implementation, the DoE utilizes a series of test cases distributed over the design space of all variables to identify, using statistical analysis, those variables which are most influential in the value of the objective function.

Using the information provided by the DoE, an efficient optimization routine, considering only those variables which are most important, may be implemented.

1.5 Outline of Thesis

In order to use SMA torque tubes in mission critical applications, the shape memory behavior, the controlled thermomechanical and electromagnetic responses as well as any specific design considerations must all be well understood. The goal of this research is to advance this understanding through theoretical development and experimentally validated, computational analysis. The remaining sections will

address these objectives.

Section 2 will develop the equations and concepts which govern the behavior of SMA torque tubes and reduce them to usable formulations. This includes the basic principles of continuum mechanics (kinematics, conservation laws and constitutive equations), modeling of the induction heating, thermo-electric response and an introduction to PID control.

Section 3 details the numerical implementation of the governing equations in Python, the design optimization algorithm and analysis of the results of the design of experiments process.

Section 4 describes the experiments utilized to validate the governing equations and computational implementation.

Section 5 provides a summary of the results. The computational and experimental findings are compared and discussed.

Section 6 summarizes and concludes this work with suggestions for future efforts.

2. GOVERNING EQUATIONS

As this project studies the behavior and design methodology of SMA torque tube actuators in the context of an operational system, a number of theoretical concepts have been incorporated. These include detailed thermomechanical modeling of shape memory alloys, electromagnetic induction heating, thermodynamics, thin airfoil theory, mechanics of materials and basic control theory.

2.1 Constitutive Modeling of Shape Memory Alloys

Shape memory alloy transformation and transformation-induced shape change are governed by the laws of thermodynamics and analyzed using various micromechanics based and phenomenological models. A recently developed phenomenological model [40] which derives from [13] was selected.

This model is built upon the Gibbs free energy, defined as $G = G(\sigma, \theta, \xi^t)$, which accounts for the external state variables, σ and θ and the internal state variables, $\xi^t = (\varepsilon^t, \xi, g^t)$. The Gibbs free energy is decomposed into austenite, martensite and “mixing” components.

$$G(\sigma, \theta, \varepsilon^t, \xi, g^t) = (1 - \xi)G^A(\sigma, \theta) + \xi G^M(\sigma, \theta) + G^{mix}(\sigma, \varepsilon^t, g^t) \quad (2.1)$$

The Gibbs free energy, stated in Equation 2.1, may be written as:

$$G^\psi(\sigma, \theta) = -\frac{1}{2\rho_m}\sigma : S^\psi\sigma - \frac{1}{\rho_m}\sigma : \alpha(\theta - \theta_0) + c^\psi \left[(\theta - \theta_0) - \theta \ln \frac{\theta}{\theta_0} \right] - s_0^\psi\theta + u_0^\psi \quad (2.2)$$

where $\psi = A, M$ and the Gibbs free energy of mixing, G^{mix} , may be stated as

$$G^{mix}(\sigma, \varepsilon^t, g^t) = -\frac{1}{\rho_m} \sigma : \varepsilon^t + \frac{1}{\rho_m} g^t. \quad (2.3)$$

Evolution of the transformation strain is governed according to the following equations under the assumption that all inelastic strain is due to change in the martensite volume fraction. Reorientation of martensite is absent from this model and its inclusion is unnecessary for the present application.

The transformation strain tensor is defined such that Λ_{fwd}^t is applied for forward transformation and Λ_{rev}^t is utilized for reverse transformation.

$$\dot{\varepsilon}^t = \Lambda^t \dot{\xi}, \quad \Lambda^t = \begin{cases} \Lambda_{fwd}^t = \frac{3}{2} H^{cur} \frac{\sigma'}{\bar{\sigma}}, & \dot{\xi} > 0 \\ \Lambda_{rev}^t = \frac{\varepsilon^{t-r}}{\xi^r}, & \dot{\xi} < 0 \end{cases} \quad (2.4)$$

ξ^r and ε^{t-r} are the martensite volume fraction and transformation strain tensor at transformation reversal. The effective Mises stress is

$$\bar{\sigma} = \sqrt{\frac{3}{2} \sigma' : \sigma'} \quad (2.5)$$

where σ' is the deviatoric stress tensor. The maximum transformation strain, H_{cur} , which saturates at H_{max} under high loading, is given by Equation 2.6.

$$H^{cur}(\bar{\sigma}) = \begin{cases} H_{min}, & \bar{\sigma} \leq \bar{\sigma}_{crit} \\ H_{min} + (H_{sat} - H_{min}) (1 - e^{-kt(\bar{\sigma} - \bar{\sigma}_{crit})}), & \bar{\sigma} \geq \bar{\sigma}_{crit} \end{cases} \quad (2.6)$$

The time rate of change of hardening energy \dot{g}^t is related to $\dot{\xi}$ by:

$$\dot{g}^t = f^t \dot{\xi}, \quad f^t = \begin{cases} f_{fwd}^t = \frac{1}{2}a_1 (1 + \xi^{n_1} - (1 - \xi)^{n_2}) + a_3, & \dot{\xi} > 0 \\ f_{rev}^t = \frac{1}{2}a_1 (1 + \xi^{n_3} - (1 - \xi)^{n_4}) + a_3, & \dot{\xi} < 0 \end{cases} \quad (2.7)$$

where f^t is an empirically calibrated, power law hardening function [40].

Utilizing the Coleman-Noll procedure and the Second Law of Thermodynamics, the generalized thermodynamic force may be computed. The Second Law of Thermodynamics which governs the generation of entropy may be stated by the Clausius-Duhem inequality

$$\frac{D}{Dt} \left(\int_{\Omega} \rho_m s dV \right) + \int_{\partial\Omega} \frac{\mathbf{q}}{\theta} \cdot \mathbf{n} dS - \int_{\Omega} \frac{\rho_m r}{\theta} dV \geq 0 \quad (2.8)$$

or

$$\rho_m \dot{s} + \frac{1}{\theta} \operatorname{div}(\mathbf{q}) - \frac{1}{\theta^2} \mathbf{q} \cdot \nabla \theta - \frac{\rho_m r}{\theta} \geq 0 \quad (2.9)$$

where s is the specific entropy per unit mass. From experimental experience, it is known that heat flows spontaneously only from hot to cold, therefore Equation 2.9 may be restated as the Clausius-Planck Inequality [39].

$$\rho_m \dot{s} + \frac{1}{\theta} \operatorname{div}(\mathbf{q}) - \frac{\rho_m r}{\theta} \geq 0 \quad (2.10)$$

The total thermodynamic force conjugate to ξ , Π , may then be stated and incorporated into the Second Law of Thermodynamics,

$$\begin{aligned} (\sigma : \Lambda^t + \frac{1}{2} \sigma : \tilde{S} \sigma + \sigma : \tilde{\alpha} (\theta - \theta_0) - \rho_m \tilde{c} \left[(\theta - \theta_0) - \theta \ln \left(\frac{\theta}{\theta_0} \right) \right] \\ + \rho_m \tilde{s}_0 \theta - \rho_m \tilde{u}_0 - f^t) \dot{\xi} = \Pi \dot{\xi} \geq 0 \end{aligned} \quad (2.11)$$

where (\sim) denotes the difference in a given value between that of the pure martensite and pure austenite states. It must then be recognized from Equation 2.11 that Π must be positive when $\dot{\xi}$ is positive and negative when $\dot{\xi}$ is negative.

From the Second Law of Thermodynamics, as stated in Equation 2.11, the critical thermodynamic driving force, Y^t , may be derived for forward and reverse transformation. The transformation surfaces may then be defined as Φ^t .

$$\Phi_{fwd}^t = \Pi_{fwd}^t - Y_{fwd}^T \quad (2.12)$$

$$\Phi_{rev}^t = -\Pi_{rev}^t - Y_{rev}^T \quad (2.13)$$

The threshold values vary according to the following equation where D captures its stress dependency.

$$Y_{fwd}^t(\sigma) = Y_0^t + D\sigma : \Lambda_{fwd}^t, \quad Y_{rev}^t(\sigma) = Y_0^t + D\sigma : \Lambda_{rev}^t \quad (2.14)$$

There are a number of parameters in this model which must be calibrated. This is performed using the following four conditions plus an additional thermodynamic requirement.

1. Beginning of forward transformation, $\Phi_{fwd}^t(\sigma = 0, \theta = M_s, \xi = 0) = 0$
2. Ending of forward transformation, $\Phi_{fwd}^t(\sigma = 0, \theta = M_f, \xi = 1) = 0$
3. Beginning of reverse transformation, $\Phi_{rev}^t(\sigma = 0, \theta = A_s, \xi = 1) = 0$
4. Ending of reverse transformation, $\Phi_{rev}^t(\sigma = 0, \theta = A_f, \xi = 0) = 0$
5. Continuity of Gibbs Free Energy $\int_0^1 f_{fwd}^t d\xi + \int_1^0 f_{rev}^t d\xi = 0$

These conditions give rise to the following parameters.

$$a_1 = \rho_m \tilde{s}_0 (M_f - M_s) \quad (2.15)$$

$$a_2 = \rho_m \tilde{s}_0 (A_s - A_f) \quad (2.16)$$

$$a_3 = -\frac{a_1}{4} \left(1 + \frac{1}{n_1 + 1} - \frac{1}{n_2 + 1} \right) + \frac{a_2}{4} \left(1 + \frac{1}{n_3 + 1} - \frac{1}{n_4 + 1} \right) \quad (2.17)$$

$$\rho_m \tilde{u}_0 = \frac{\rho_m \tilde{s}_0}{2} (M_s + A_f) \quad (2.18)$$

$$Y_0^t = \frac{\rho_m \tilde{s}_0}{2} (M_s - A_f) - a_3 \quad (2.19)$$

Calibration of \tilde{s}_0 and D requires computation of the slope of the transformation surface at a reference stress, σ^* . By the Kuhn-Tucker condition which states that $\Phi^t \dot{\xi} = 0$, it is recognized that, in 1-D:

$$d\Phi^t = \partial_\sigma \Phi^t d\sigma + \partial_\theta \Phi^t d\theta + \partial_\xi \Phi^t d\xi = 0 \quad (2.20)$$

Given a series of known calibration conditions where $\xi = 0, 1$, and the additional assumption in 1-D that $\Lambda_{fwd}^t = \Lambda_{rev}^t$, the remaining parameters may be computed.

$$C_M = \frac{-\rho_m \tilde{s}_0}{(1 - D) (\Lambda^t + \sigma \partial_\sigma \Lambda^t) + \sigma \left(\frac{1}{E^M} - \frac{1}{E^A} \right)} \Bigg|_{\sigma=\sigma^*} \quad (2.21)$$

$$C_M = \frac{-\rho_m \tilde{s}_0}{(1 + D) (\Lambda^t + \sigma \partial_\sigma \Lambda^t) + \sigma \left(\frac{1}{E^M} - \frac{1}{E^A} \right)} \Bigg|_{\sigma=\sigma^*} \quad (2.22)$$

C_M and C_A may be determined experimentally and the above equations may be

rearranged to finally obtain \tilde{s}_0 and D .

$$\rho_m \tilde{s}_0 = \frac{-2(C^M C^A) \left[H^{cur}(\sigma) + \sigma \partial_\sigma H^{cur}(\sigma) + \sigma \left(\frac{1}{E^M} - \frac{1}{E^A} \right) \right]}{C^M + C^A} \Bigg|_{\sigma=\sigma^*} \quad (2.23)$$

$$D = \frac{(C^M - C^A) \left[H^{cur}(\sigma) + \sigma \partial_\sigma H^{cur}(\sigma) + \sigma \left(\frac{1}{E^M} - \frac{1}{E^A} \right) \right]}{(C^M + C^A) [H^{cur}(\sigma) + \sigma \partial_\sigma H^{cur}(\sigma)]} \Bigg|_{\sigma=\sigma^*} \quad (2.24)$$

The rate of change of the martensite volume fraction, $\dot{\xi}$ is found by rearranging and substituting terms into the derivative of Π in time from Equation 2.11.

$$\dot{\xi} = \left(-\frac{\partial \Pi}{\partial \xi} \right)^{-1} \left(\frac{\partial \Pi}{\partial \sigma} \dot{\sigma} + \frac{\partial \Pi}{\partial \theta} \dot{\theta} \right) \quad (2.25)$$

$$\dot{\xi} = \left(-\frac{\partial \Pi}{\partial \xi} \right)^{-1} \left((\Lambda^t + \tilde{S}\sigma) \dot{\sigma} + \rho_m \tilde{s}_0 \dot{\theta} \right) \quad (2.26)$$

2.1.1 Reduced Constitutive Model for 1-D Torsion

Because the shape memory behavior occurs in torsion only, the constitutive model simplifies considerably. The cases in Equation 2.4 are equal for forward and reverse transformation.

$$\dot{\varepsilon}^t = \Lambda^t \dot{\xi}, \quad \Lambda^t = \begin{cases} \Lambda_{fwd}^t = \frac{3}{2} H^{cur} \frac{\sigma'}{\bar{\sigma}}, & \dot{\xi} > 0 \\ \Lambda_{rev}^t = \frac{\varepsilon^t - r}{\xi^r}, & \dot{\xi} < 0 \end{cases} = \frac{3}{2} H^{cur} \frac{\sigma'}{\bar{\sigma}} \quad (2.27)$$

Furthermore, since $\sigma_{ij} = \tau, (i, j) \in [(1, 2), (2, 1)]$,

$$\bar{\sigma} = \sqrt{\frac{3}{2} \sigma'} : \sigma' = \sqrt{\frac{3}{2} 2\tau^2} = \sqrt{3} |\tau| \quad (2.28)$$

$$\Lambda^t = \frac{\sqrt{3}}{2} H^{cur} \text{sgn}(\tau) \quad (2.29)$$

Recognizing that $\tilde{\alpha} = 0$ in torsion and making the additional assumption that \tilde{c} is negligible, Equation 2.11 may be rewritten.

$$\left(2\tau\Lambda^t + 4(1 + \nu)\tau^2 \left(\frac{1}{E_M^{el}} - \frac{1}{E_A^{el}}\right) + \rho_m\tilde{s}_0\theta - \rho_m\tilde{u}_0 - f^t\right) \dot{\xi} = \Pi\dot{\xi} \geq 0 \quad (2.30)$$

Finally for the 1-D case in torsion, the rate of change of the martensite volume fraction given by Equation 2.26 reduces as shown.

$$\dot{\xi} = \left(-\frac{\partial\Pi}{\partial\xi}\right)^{-1} \left(\left(\frac{\sqrt{3}}{2}H_{cur}\text{sign}(\tau) + \left(\frac{1}{G_M^{el}} - \frac{1}{G_A^{el}}\right)\tau \right) \dot{\tau} + \rho_m\tilde{s}_0\dot{\theta} \right) \quad (2.31)$$

2.1.2 Minor Hysteresis Loop Modification of the SMA Constitutive Model

The SMA constitutive model presented above is part of a larger category of thermomechanical models which, while they are well suited to modeling complex multi-axial loading paths, are known to perform poorly when modeling the minor loop hysteresis response. Minor loops are characterized by having a cyclic temperature range not completely spanning between M_f and A_f [12, 20, 41].

A proposed modification which has been shown to improve the minor loop fidelity of the model may be accomplished by altering the forward and reverse hardening functions, $f_{fwd}^t(\xi)$ and $f_{rev}^t(\xi)$ so that $f_{fwd}^t(\hat{\xi}_{fwd}(\xi))$ and $f_{rev}^t(\hat{\xi}_{rev}(\xi))$ where

$$\hat{\xi}_{fwd}(\xi) = \frac{1}{1 - \xi^f}\xi - \frac{\xi^f}{1 - \xi^f}, \quad 0 \leq \xi^f < 1, \quad (2.32)$$

$$\hat{\xi}_{rev}(\xi) = \frac{1}{\xi^r}\xi, \quad 0 < \xi^r \leq 1 \quad (2.33)$$

and ξ^f is the martensite volume fraction at the end of reverse transformation and ξ^r is the martensite volume fraction at the end of forward transformation [12].

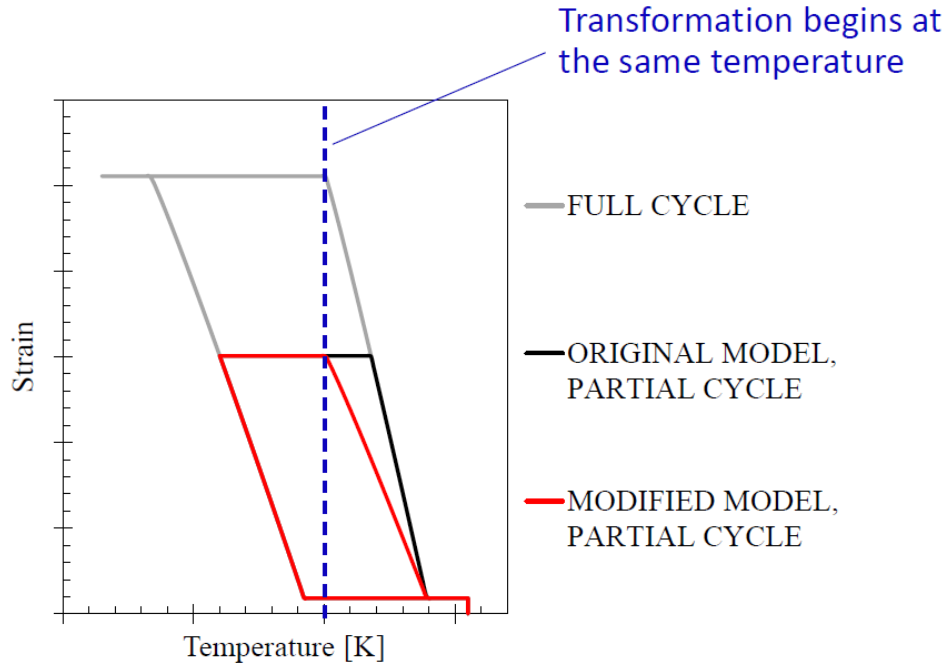


Figure 2.1: Effect of partial loop modification on the behavior of a single isobaric cycle.

The impact of this modification is that forward and reverse transformation always initiates at the martensite and austenite start transformation temperatures for a given stress level regardless of the current martensite volume fraction. This effect is shown for a single cycle in Figure 2.1 and for multiple cycles in Figure 2.2.

2.2 Series Equivalent Circuit Modeling of Induction Heating

Electromagnetic induction heating is seen as an enabling technology for SMA torque tube actuators [56, 55, 71]. In order to effectively use and design systems with this technology, several aspects of the electrical system must be characterized. These include the power applied to the torque tube, the electrical impedance across the induction coil, the skin depth, the electrical efficiency and the power factor.

The series equivalent circuit (SEC) model provides a rough estimate of these

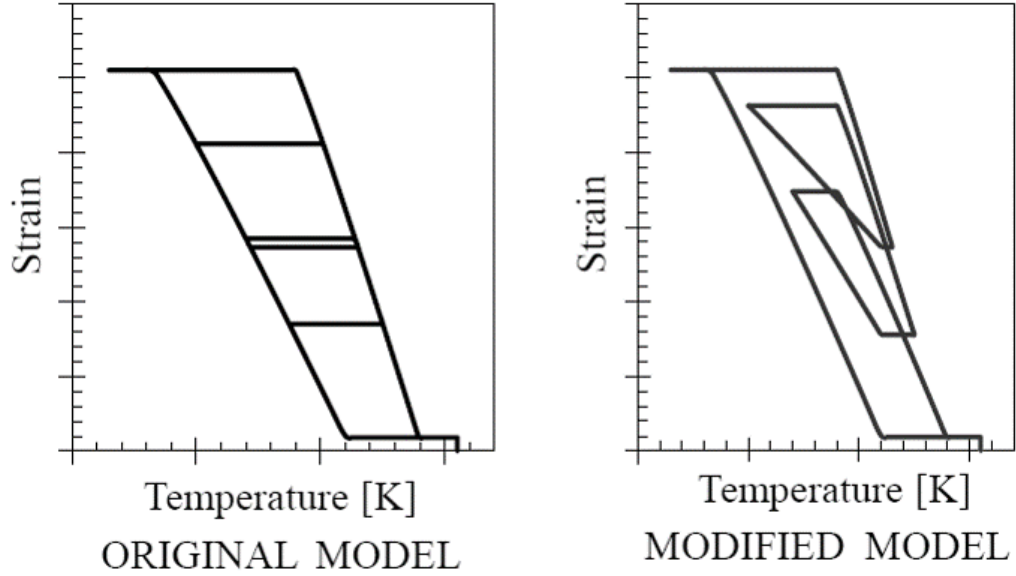


Figure 2.2: Effect of partial loop modification on the behavior of multiple isobaric cycles.

quantities by analyzing the path and distribution of the magnetic flux passing through the work piece (Nitinol torque tube) and through the surrounding induction coil and open space and converting this magnetic flux distribution into an equivalent electrical circuit [6, 19, 65]. This model is limited to cases where the work piece is completely surrounded by the coil and the length of the workpiece is equal to or greater than the diameter of the coil. Additionally, it is assumed that the workpiece and coil geometry are uniform along their length [6].

The RMS magnetic field intensity is defined at the outer surface by

$$H = \frac{N_c I_{coil}}{l} \quad (2.34)$$

for a given number of induction coil turns, N_c , RMS current in the induction coil, I_{coil} and length l .

Important shape parameters in this model are γ , p and q which are functions of the skin depth, δ_E [19].

$$\gamma_E = \frac{R_o t}{\delta_E^2} \quad (2.35)$$

$$p = \frac{\gamma_E}{1 + \gamma_E^2} \quad (2.36)$$

$$q = \frac{1}{1 + \gamma_E^2} \quad (2.37)$$

Finally an empirical correction factor is included in this model. For single layer coils with copper windings thicker than the skin depth of the copper coils, δ_c , a good value typically used is $k_r = 1.15$ [6, 19].

2.2.1 Skin Depth

Although total power applied to the torque tube increases with the frequency of the applied current, that power tends to become more concentrated at the surface nearest the coil of the material being heated, as shown in Figure 2.3. A metric for the depth of penetration, the skin depth,

$$\delta_E = \frac{1}{k} = \sqrt{\frac{2\rho}{\mu\omega}} \quad (2.38)$$

can be derived from Maxwell's Equations given an electrical resistivity, ρ , the magnetic permeability, $\mu = \mu_r \mu_0$, $\mu_0 = 4\pi \times 10^{-7} \text{ N/A}^2$ and the angular frequency of the electrical current in the coil, ω [49, 6, 11, 19, 42, 55, 66].

In order to heat the material evenly and ensure that transformation occurs as uniformly as possible across the radius of the material, the frequency must be selected appropriately for the geometry and material of the structure being heated.

An additional effect of skin depth is on the electrical resistance of a current conducting body. As frequency is increased and the current concentrates on the surface

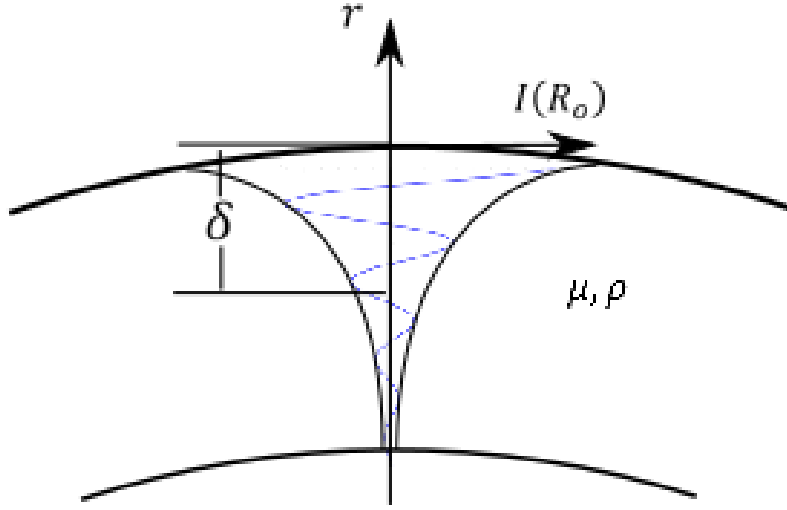


Figure 2.3: Homogeneous conducting annular space with sinusoidal current. The amplitude of the current density vector at an instant of time versus the radial position, r , is indicated in blue.

of the conductor, the effective conducting cross-sectional area of the conductor decreases. Therefore, the electrical resistance of the material increases according to the angular frequency of the applied signal.[4, 6]

2.2.2 Power Applied to the Torque Tube

The power applied to the SMA torque tube is a function of the geometric and material properties of the SMA as well as the design of the coil providing the power to the SMA.

According to the SEC model given by [6, 19], power may be expressed as

$$P = \mu_0 \pi f 2H^2 l A_w p \quad (2.39)$$

where $A_w = \pi R_o^2$.

2.2.3 Electrical Impedance

The most efficient energy transfer from the AC power supply to the torque tube actuator, shown in Figure 2.4 with an induction coil and assumed mechanical load, is obtained when the impedance of the power supply is matched with the electrical impedance across the induction coil [19].

The electrical impedance is defined by a real component and an imaginary reactance. The relative magnitude of the real and imaginary portions is responsible for the phase difference between the voltage and current signals while the magnitude of the impedance determines the ratios between the amplitudes of the current and electric potential. These quantities are defined below [6, 19, 65]. This system is primarily inductive, therefore the capacitive portion of the reactance can be neglected.

$$\mathbf{E} = \mathbf{Z}\mathbf{I} \quad (2.40)$$

$$\mathbf{Z} = R + Xj \quad (2.41)$$

$$X = \omega L - \frac{1}{\omega C} \approx \omega L \quad (2.42)$$

The impedance may also be expressed in polar form.

$$\mathbf{Z} = \|\mathbf{Z}\|(\cos\varphi + \sin\varphi j) = \|\mathbf{Z}\|\angle\varphi \quad (2.43)$$

$$\|\mathbf{Z}\| = \sqrt{R^2 + X^2} \quad (2.44)$$

$$\varphi = \tan^{-1}\frac{X}{R} \quad (2.45)$$

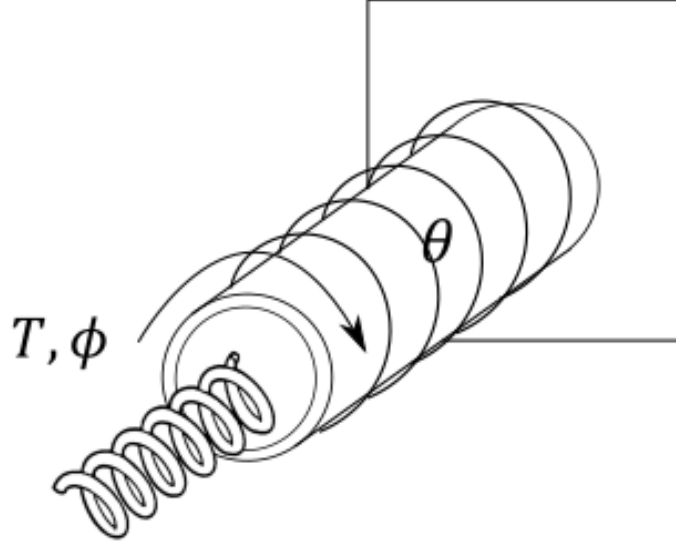


Figure 2.4: Conceptual drawing of the inductively heated torque tube as modeled with linear loading on the free end.

A method of approximating the electrical impedance exists based on a hypothetical relationship between the magnetic circuit and the equivalent electrical circuit [6, 19]. This method provides an accurate prediction of the impedance which may be used for design purposes.

In this model, the electrical impedance is defined as

$$\mathbf{Z} = (R_w + R_c) + j(X_g + X_w + X_c) \quad (2.46)$$

where R_w , R_c , X_g , X_w and X_c are defined as

$$R_w = K\mu_r p A_w \quad (2.47)$$

$$R_c = Kk_r \pi r_c \delta_c \quad (2.48)$$

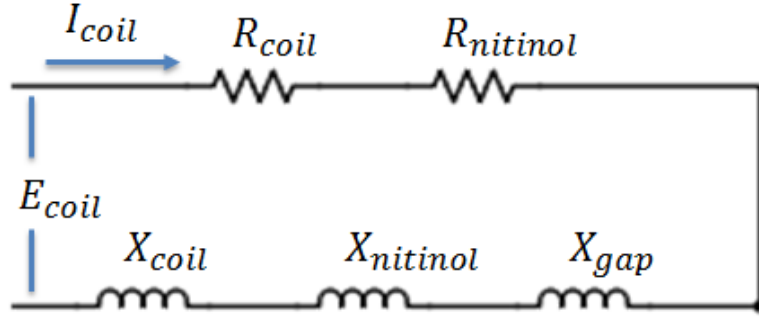


Figure 2.5: Equivalent electrical circuit used in the development of the induction heating model.

$$X_g = K A_g \quad (2.49)$$

$$X_w = K \mu_r q A_w \quad (2.50)$$

$$X_c = K k_r \pi r_c \delta_c \quad (2.51)$$

$$K = 2\pi f \mu_0 \left(\frac{N_c^2}{l} \right) \quad (2.52)$$

and shown schematically in Figure 2.5.

2.2.4 Electrical Efficiency and Power Factor

From the model discussed above, the efficiency of electricity utilization and transfer is determined by two related parameters, the Coil Efficiency, η and the Coil Power Factor, $\cos\phi$. These are defined by:

$$\eta = \frac{R_W}{R_C + R_W} \quad (2.53)$$

$$\cos\phi = \frac{R_W + R_C}{\|\mathbf{Z}\|} \quad (2.54)$$

The electrical efficiency gives the portion of total power which heats the torque tube. The power factor is defined as the ratio of the real power and the apparent power transferred to the induction coil. A low power factor indicates a large amount of reactive power in the circuit which is unavailable to heat the SMA.

In order to maximize the total electrical efficiency of the system, each of these parameters should be maximized and both the real and imaginary components of the power supply impedance should be matched to the impedance of the "loaded" induction coil.

2.3 System of Differential Equations in Time

The thermomechanical behavior of the torque tube system is governed by two equations. These are mechanical equilibrium in rotation and the First Law of Thermodynamics.

2.3.1 Mechanical Equilibrium

Mechanical equilibrium is defined as follows under a quasi-static assumption. Derivation of this equation begins with the traditional solution for twist of a tube under torsion with the addition of the inelastic transformation strain, γ^{tr} , where $\gamma = \gamma^{el} + \gamma^{tr}$, and proceeds as follows.

$$\tau = \frac{TR}{J} = G(\gamma - \gamma^{tr}) \quad (2.55)$$

$$\phi = \gamma \frac{l}{R} = T \frac{l}{GJ} + \gamma^{tr} \frac{l}{R} \quad (2.56)$$

$$\frac{TR}{J} = G(\gamma - \gamma^{tr}) = G \left(\frac{\phi R}{l} - \gamma^{tr} \right) \quad (2.57)$$

The total moment is a sum of constant moment and aerodynamic loading. For the purposes of this work, it is convenient to model the aerodynamic loading as a linear spring in torsion. Equation 1.3 becomes Equation 2.58 with the given equivalent spring constant, k_{sp} , carried forward in the system modeling and experiments. δ is the amount of rotation in radians relative to an unloaded reference angle.

$$T_{aero} = k_{sp}\delta \quad (2.58)$$

$$T = k_{sp}(\phi - \phi_0) + T_0 \quad (2.59)$$

$$k_{sp}(\phi - \phi_0) \frac{R}{J} + T_0 \frac{R}{J} = G \left(\frac{\phi R}{l} - \gamma^{tr} \right) \quad (2.60)$$

$$\left(k_{sp} \frac{R}{JG} - \frac{R}{l} \right) \phi - k_{sp} \frac{R}{JG} \phi_0 + T_0 \frac{R}{GJ} = -\gamma^{tr} = -2\varepsilon^{tr} = -2 \int_t \Lambda^{tr} \dot{\xi} dt \quad (2.61)$$

This development results and can be rearranged into Equation 2.62.

$$\begin{aligned} \phi &= \frac{1}{k_{sp} \frac{R}{GJ} - \frac{R}{l}} \left[k_{sp} \frac{R}{GJ} \phi_0 - T_0 \frac{R}{GJ} - 2 \int_t \Lambda^{tr} \dot{\xi} dt \right] \\ &= \frac{1}{k_{sp} \frac{R}{GJ} - \frac{R}{l}} \left[k_{sp} \frac{R}{GJ} \phi_0 - T_0 \frac{R}{GJ} - \gamma^{tr} \right] \end{aligned} \quad (2.62)$$

It is important to recognize that G varies with ξ along with ρ_e resulting in full coupling between the electromagnetic and thermomechanical responses.

$$G^{el}(\xi) = \left(G_A^{el-1} + \xi \left(G_M^{el-1} - G_A^{el-1} \right) \right)^{-1} \quad (2.63)$$

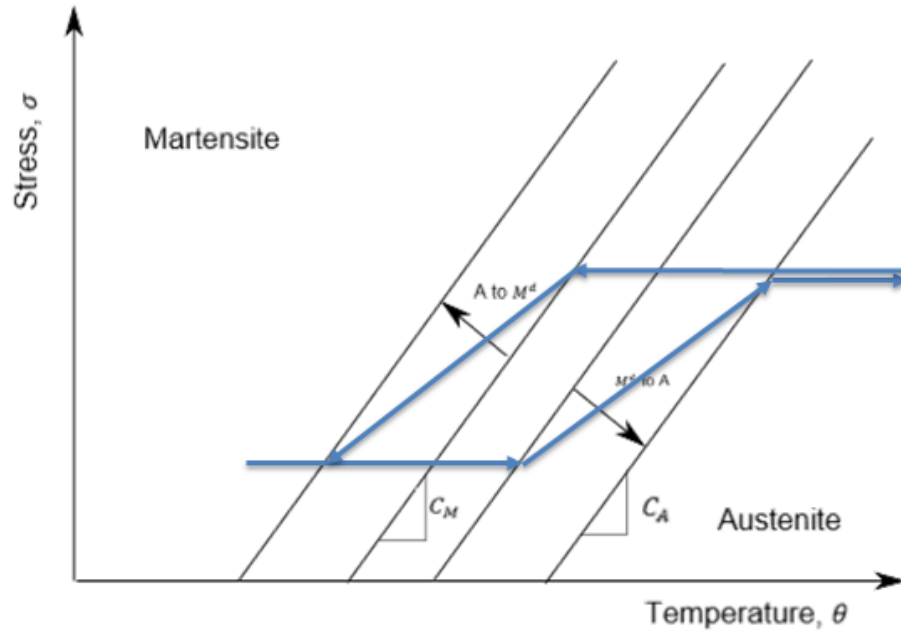


Figure 2.6: Schematic drawing of thermomechanical loading path on SMA phase diagram.

Because of the nature of the variable loading in the present problem, the SMA response can be shown schematically for complete transformation cycles on the SMA phase diagram as shown in Figure 2.6.

2.3.2 First Law of Thermodynamics

With the latent heat of transformation of the SMA, the First Law of Thermodynamics may be stated as

$$V \rho_m c \dot{\theta} = \dot{Q}_{in} - \dot{Q}_{out} + \dot{Q}_{LH} \quad (2.64)$$

the components of which are defined as

$$\dot{Q}_{in} = P \quad (2.65)$$

$$\dot{Q}_{out} = hA(\theta - \theta_\infty) \quad (2.66)$$

$$\dot{Q}_{LH} = \begin{cases} V(-Y + \rho_m \tilde{s}_0 \theta) \left(\frac{\rho_m \tilde{s}_0}{\partial_\xi f_{rev}^t} \right) & \text{if } \dot{\xi} > 0 \\ -V(Y + \rho_m \tilde{s}_0 \theta) \left(\frac{\rho_m \tilde{s}_0}{\partial_\xi f_{fwd}^t} \right) & \text{if } \dot{\xi} < 0 \end{cases}. \quad (2.67)$$

Finally, these can be combined as

$$\dot{\theta} = \begin{cases} \frac{1}{V\rho_m c} [P - hA(\theta - \theta_\infty)] & \text{if } \dot{\xi} = 0 \\ \frac{1}{V\rho_m c} \left[P - hA(\theta - \theta_\infty) + V(-Y + \rho_m \tilde{s}_0 \theta) \left(\frac{\rho_m \tilde{s}_0}{\partial_\xi f_{rev}^t} \right) \right] & \text{if } \dot{\xi} > 0 \\ \frac{1}{V\rho_m c} \left[P - hA(\theta - \theta_\infty) - V(Y + \rho_m \tilde{s}_0 \theta) \left(\frac{\rho_m \tilde{s}_0}{\partial_\xi f_{fwd}^t} \right) \right] & \text{if } \dot{\xi} < 0 \end{cases}. \quad (2.68)$$

2.4 Proportional-Integral-Derivative (PID) Control Law

For the initial modeling, the PID control parameter, \aleph , is then mapped to a coil current and convection coefficient according to the following law. This is shown graphically in Figure 2.7.

- $-1 \leq \aleph \leq 0$: $I = 0$, $h = h_{MIN} - \aleph(h_{MAX} - h_{MIN})$
- $0 \leq \aleph \leq 1$: $I = \aleph I_{MAX}$, $h = h_{MIN}$
- $\aleph < -1$: $I = 0$, $h = h_{MAX}$
- $\aleph > 1$: $I = I_{MAX}$, $h = h_{MIN}$

Ensuring acceptable performance of this controller requires proper tuning of the parameters k_P, k_I, k_D . Unfortunately, when the system being controlled is highly

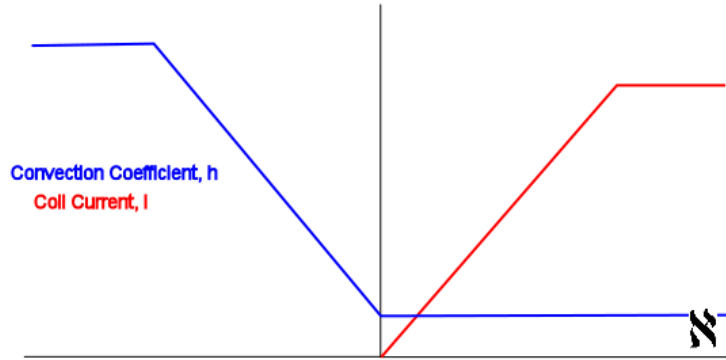


Figure 2.7: Mapping of the controller output to the heating and cooling process inputs.

nonlinear, these parameters can prove extremely difficult to select using traditional techniques [23]. For the present problem, the controller is faced with nonlinear power input as a function of current as well as a significant hysteresis in the SMA response. Because of this, these parameters will be optimized alongside the geometric parameters as detailed in the following section. This technique has been demonstrated in the literature [53].

Due to experimental limitations, pulse width modulation (PWM) was selected to provide controlled cooling. Under this system, the convection coefficient is alternated between two, experimentally determined, high and low values corresponding to active and natural convective cooling.

Under this control methodology, the heating mode remains identical to the previous scheme, as does cooling when heating is active and when the control law saturates. Controller saturation and duty cycle mapping occurs according to the following control laws:

- $-1 \leq \aleph \leq 0$: $I = 0$
- $0 \leq \aleph \leq 1$: $I = \aleph I_{MAX}$, $h = h_{MIN}$

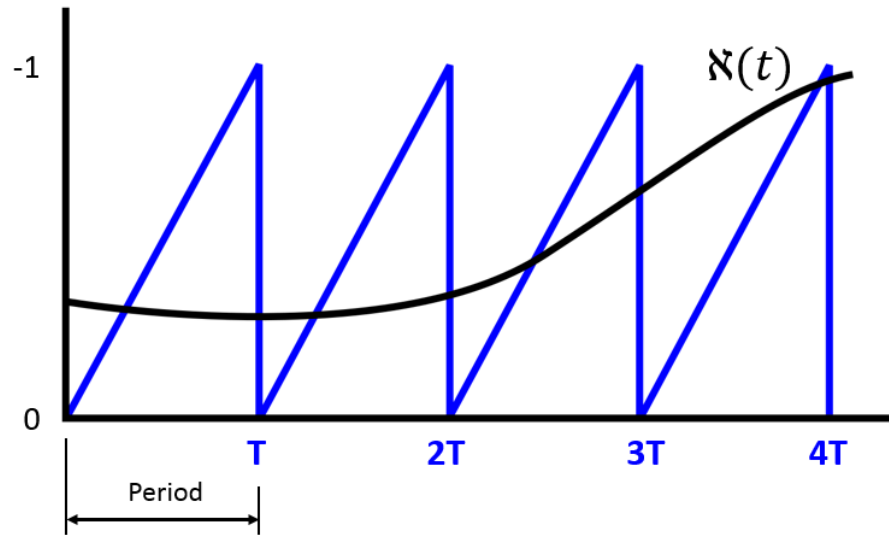


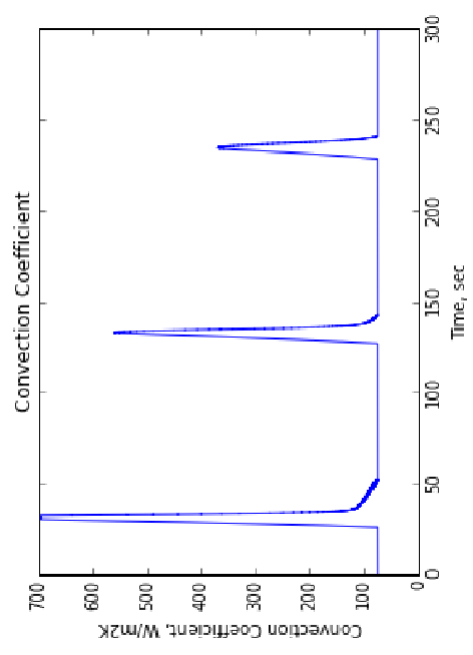
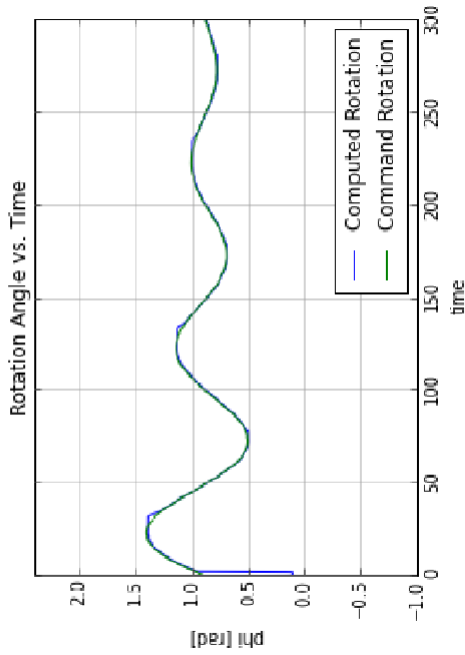
Figure 2.8: Pulse Width Modulation (PWM) carrier wave with control function.

- $\aleph < -1$: $I = 0$, $h = h_{MAX}$
- $\aleph > 1$: $I = I_{MAX}$, $h = h_{MIN}$

Modulation of the cooling by PWM, when $-1 \leq \aleph \leq 0$, is accomplished according to the following procedure. A sawtooth carrier wave of period, T , oscillating between the values of 0 and -1 is computed throughout the simulation [63].

Whenever the amplitude of the control function, \aleph , exceeds that of the carrier wave, shown in Figure 2.8, active cooling is set to the "on" state. When it is less than the carrier function, it is in the "off" state. Early modeling efforts described later have demonstrated that this methodology produces similar control response to the "continuous" control mode, even at low carrier frequencies, as shown in Figure 2.9.

Continuous Cooling Control



PWM Cooling Control

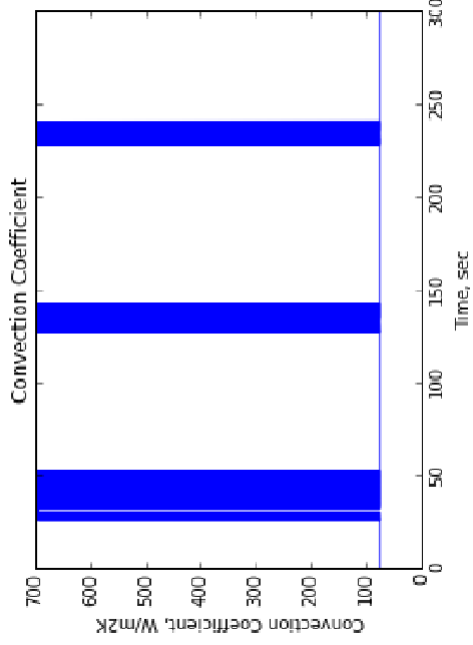
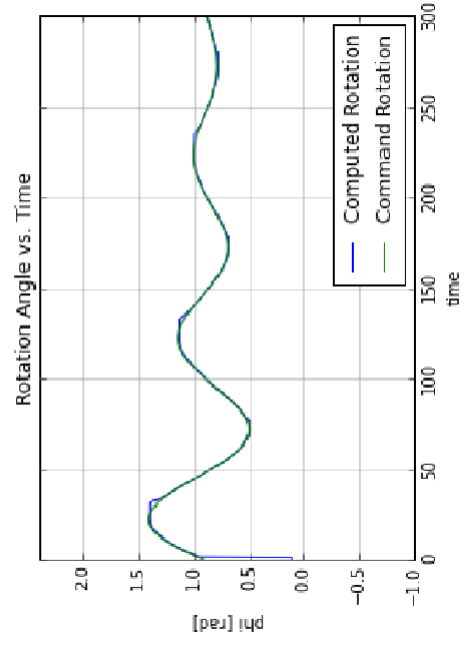


Figure 2.9: Comparison of the rotation and convection coefficients between the continuous control methodology (Left) and the PWM methodology (Right) in the time domain.

3. COMPUTATIONAL MODELING AND OPTIMIZATION

The governing equations of this system have been implemented into a custom Python script using a time stepping algorithm based on the Euler method which simulates the response of the SMA torque tube under combined static/aerodynamic torsional loading.

The simplest Euler method uses a first-order Taylor Series approximation to compute sequential values of a differential equation starting from a known initial condition. For example, the differential equation given by Equation 3.1 may be approximated by Equation 3.2 [24].

$$\frac{dy}{dx} = f(x, y), \quad y(x_0) = y_0 \quad (3.1)$$

$$y(x) = y(x_0) + y'(x_0)(x - x_0) + O(h^2) \quad (3.2)$$

The coupled differential equations 2.68 and 2.62 are implemented in a similar manner.

This algorithm has been implemented in Python with several options and variations which allow the study of various aspects of the SMA response. The primary code actuates the SMA, starting in martensite, to complete rotation and back again. This allows for direct comparison to SMA material characterization experiments and to verify the accuracy of the SMA parameters and constitutive model. The algorithms for the complete cycle and controller variant programs are given in Figures 3.1 and 3.2. An example output is shown in Figure 3.3. This code also contains an optional arbitrary rotation command capability in which a predefined function (i.e. a sine wave) is generated to command rotation through a PID controller and the thermomechanical response is simulated. An example of the controller variant

Table 3.1: Design points used in DoE study.

R_i [m]	t [m]	l [m]	Frequency [Hz]	Wire Diameter [m]
0.001	0.0015	0.2	50000	.00367
0.003	0.0020	0.3	150000	.00291

program rotation output is given in Figure 3.4. Two variants of this code permit the Design of Experiments process to be executed to understand the impact of various design and control parameters and to implement a Particle Swarm Optimization Algorithm routine.

3.1 Design of Experiments

For the initial DoE, the effects of the inside radius, thickness and length of the torque tube as well as the frequency of the coil electric current and the diameter of the wire making up the tightly packed induction coil are studied. The impact of each of these parameters on the time required for a single cycle, the maximum angle of rotation, the power factor and the electrical efficiency are evaluated. The DoE was run with and without the minor loop modification to the constitutive model.

As is shown in Figure 3.5, all of the tested variables except for the torque tube length have an effect on the time required for a single cycle of operation. This result was expected and verified analytically using the First Law of Thermodynamics.

3.2 Particle Swarm Optimization of Design and Control Response

The particle swarm optimization algorithm operates by (1) generating a well distributed series of independent agents or "particles" within the bounded hyperspace of all design variables. In addition to a position, each particle is also assigned an initial velocity in each of the design dimensions. (2) The objective function for each agent is then evaluated, and the location of each is recorded as an individual best

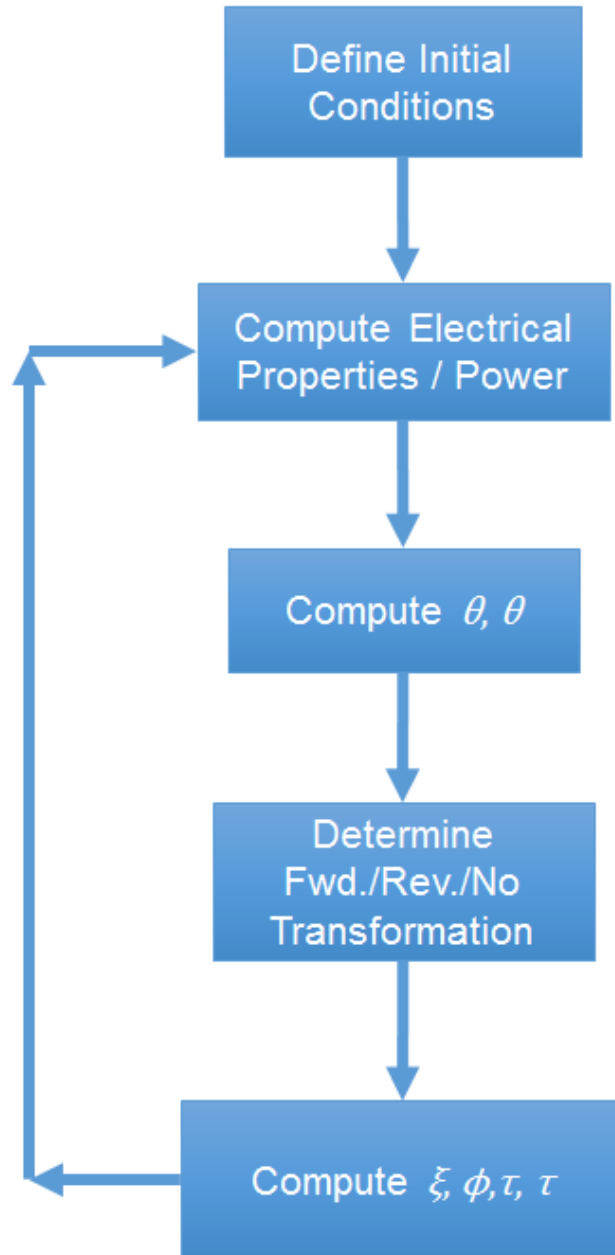


Figure 3.1: Single loop Python script algorithm.

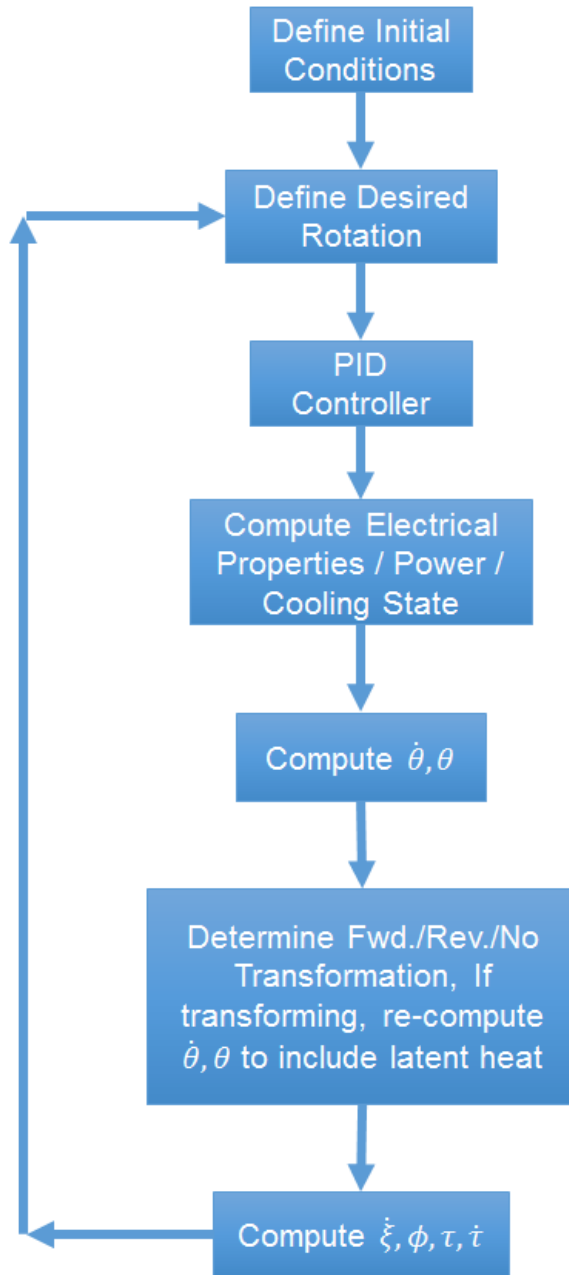


Figure 3.2: Python script algorithm with PID control.

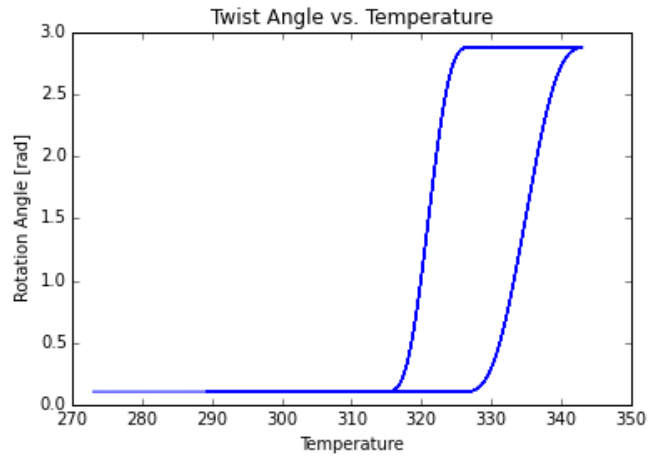


Figure 3.3: Example SMA response of complete cycle implementation.

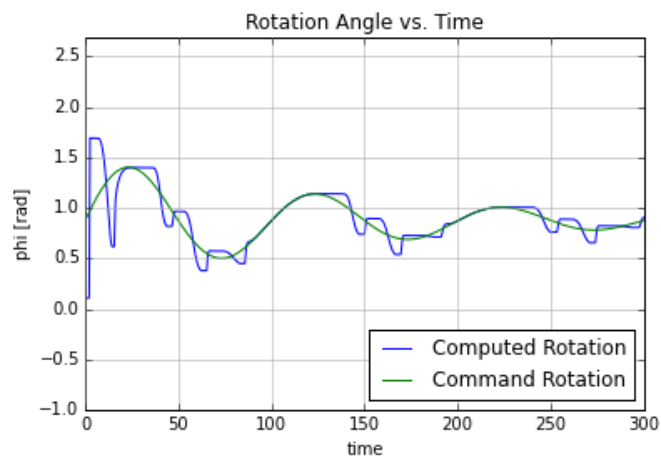


Figure 3.4: Example command function (green) and simulated response (blue).

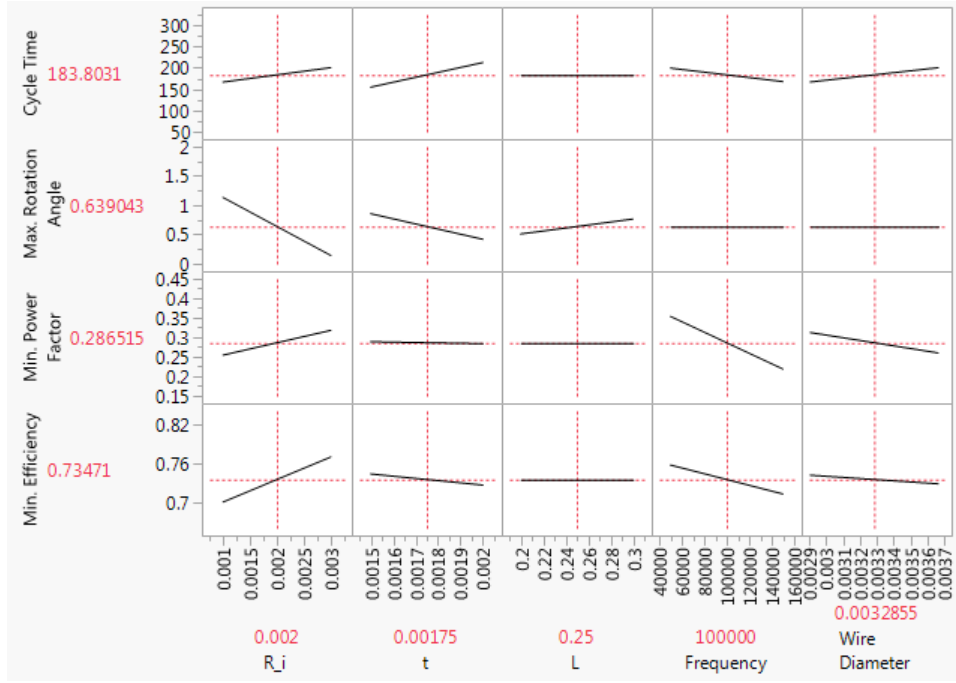


Figure 3.5: Initial DoE results from JMP statistical analysis software.

and the location of the overall best solution is recorded as a global best. (3) Once the objective function is computed for each agent, the velocity, v_n^i , and position, r_n^i , for each is updated according to Equations 3.3 and 3.4. The process then returns to Step 2 and repeats until convergence is achieved or a specified number of cycles are completed [36, 46, 54, 37].

$$v_n^i(t + \Delta t) = wv_n^i(t) + c_1\chi_1 [r_n^{i,L} - r_n^i(t)] \Delta t + c_2\chi_2 [r_n^{i,g} - r_n^i(t)] \Delta t \quad (3.3)$$

$$r_n^i(t + \Delta t) = r_n^i(t) + \Delta tv_n^i(t) \quad (3.4)$$

The components of Equation 3.3 are selected to incorporate various physical and/or social concepts into the optimization model. An "inertial" factor, w , is first incorporated to include present behavior into the decision making process for the

next increment. The magnitude of w affects the stability of direction of the particle convergence. A value too small results in unstable behavior when faced with a rapidly changing known global optimum location. This is especially troublesome early on in the optimization process. Conversely, a magnitude too large of w results in a particle unresponsive to both its own and the social knowledge of the landscape from previous increments. Next a variable c_1 is defined as a "cognitive" factor which weights the acceleration of the particle towards that particle's individual best solution. Finally, a third, "social" factor, weighted by c_2 , is included to direct the acceleration of the particle towards the best known global solution. Both c_1 and c_2 are pre-multiplied by χ_1 and χ_2 which are randomly generated numbers where $\chi_n \in [0, 1]$. Good values for w , c_1 and c_2 have been found and published in literature as 1, 2 and 2 respectively. This combination of parameters results in a particle which tends to overshoot its best known solution about half of the time. It has also been suggested in literature that stability of the optimization process may be regulated through the use of a "speed limit" [37, 54]. Optimization runs with and without this limit and with different tuning parameters will be executed for comparison of results and speed of convergence.

Another aspect of this optimization method given considerable attention is the behavior of the particles when they encounter a boundary. Variations explored in literature include an "absorbing" boundary where the velocity of the offending particle is brought instantly to zero in the relevant spatial dimension. Another variation "reflects" the particle back off of the boundary with equal and opposite velocity. The final type permits the particle to violate the boundary, but no score is computed for the particle while outside of the designated space [46, 54]. In this implementation, the particle is stopped at the wall and the objective function is computed. However, the computed acceleration and velocity of the particle are permitted to progress

naturally as though no boundary were present.

3.2.1 Optimization Problem Description

For this work, two related optimization problems are solved. 1) The PID controller driving the torque tube to be tested experimentally. 2) The geometry and controller properties of torque tube with identical properties will be optimized simultaneously. Both optimizations will use the RMS error in the command versus simulated rotation as the objective function to be minimized. The RMS Error is defined as

$$\epsilon_{RMS} = \sqrt{\frac{\sum_t \epsilon^2}{\frac{\Delta t}{t}}} = \sqrt{\frac{\sum_t (\phi - \phi_c)^2}{\frac{\Delta t}{t}}} \quad (3.5)$$

For the controller optimization, the terms k_P , k_I and k_D of the PID controller along with the carrier frequency for the pulse width modulated cooling will be varied. For the complete optimization, the inside radius, R_i , the thickness, t , the wire diameter and the frequency of the induction current, f will also be varied.

Constraints on the maximum shear stress, the minimum power factor and the minimum electrical efficiency will also be implemented along with bounds on the parameters corresponding to physical limitations (i.e. inner radius must be positive).

4. EXPERIMENTAL DESIGN

Two experiments were conducted in order to validate the theory and computational methods used. The first is designed to evaluate the impedance across the loaded induction coil. The second will measure the thermomechanical and controller response.

4.1 Electromagnetic Characterization: Impedance Measurement

As shown in Equation 2.44, impedance is defined by two parts, a real-valued resistance and an imaginary valued reactance, the sum of which may be represented in polar coordinates. The impedance of an electrical component may be measured by several means. The most common method used by electricians is the use of an LCR meter. However inexpensive versions of these devices are typically limited to AC frequencies of 50-60 Hz. As induction heating uses frequencies on the order of 10-100 kHz, another method must be devised.

The phase angle of the value of the impedance is equal to the phase difference in the phasors between the sinusoidal electrical potential across the component and the associated alternating current. Furthermore, the amplitude of the impedance is equal to the ratio of the amplitudes between the potential and the current signals. This suggests that the impedance may be calculated by taking measurements of the current and voltage amplitudes and their relative phase. The voltage signal is easily measured in both DC and AC circuits. Additionally, for DC, the current may be obtained by measuring the potential drop across a shunt resistor of known value wired in series with the component of interest. This is not possible, however, in high-frequency AC circuits due to the effect of the skin depth on the resistance of the shunt resistor. This means that a dedicated current sensor such as a current clamp or hall

effect sensor must be employed. Additionally, the current probe utilized must be designed for operation in the both the frequency and current domain of the problem. A Tektronix TCP 300 Series current probe was used for this experiment. In order to get the relative phase of the two signals, the differential voltage probe and the current probe each output analog signals to an oscilloscope on opposing channels. This setup is shown in Figure 4.1. The results of this experiment were compared to the values predicted by Equations 2.37 through 2.46 as given in Table 5.3.

4.2 Complete System Testing with PID Control and Realistic Loading

In addition to the electrical properties, the thermomechanical SMA model and computed control behavior were validated against experiments. A torque tube provided by The Boeing Company was subjected to a series of constant moments. The angle of rotation of the tube was controlled using a bi-modal PID controller implemented in LabVIEW and recorded by sensors. Heating was effected through an Am-brell Induction heater and controlled cooling through pulse width modulated bursts of shop compressed air through the inside of the torque tube. Variables measured included the temperature, and rotation angle of the torque tube.

The effective convection cooling coefficient, with and without pressurized air passing through the torque tube, h , was be determined prior to the test. Once these and the SMA parameters required to calibrate the model were determined, a series of input control functions was generated and converted by a PID controller, implemented in LabVIEW, into command signals transmitted by a NI USB-6211 data acquisition card. These command signals modulate the current output of the induction heater and the duty cycle of the solenoid valve controlling the shop compressed air used for cooling. moment was applied by suspending a dead load from a wheel fixed to the free end of the torque tube. Rotation of the torque tube was measured using

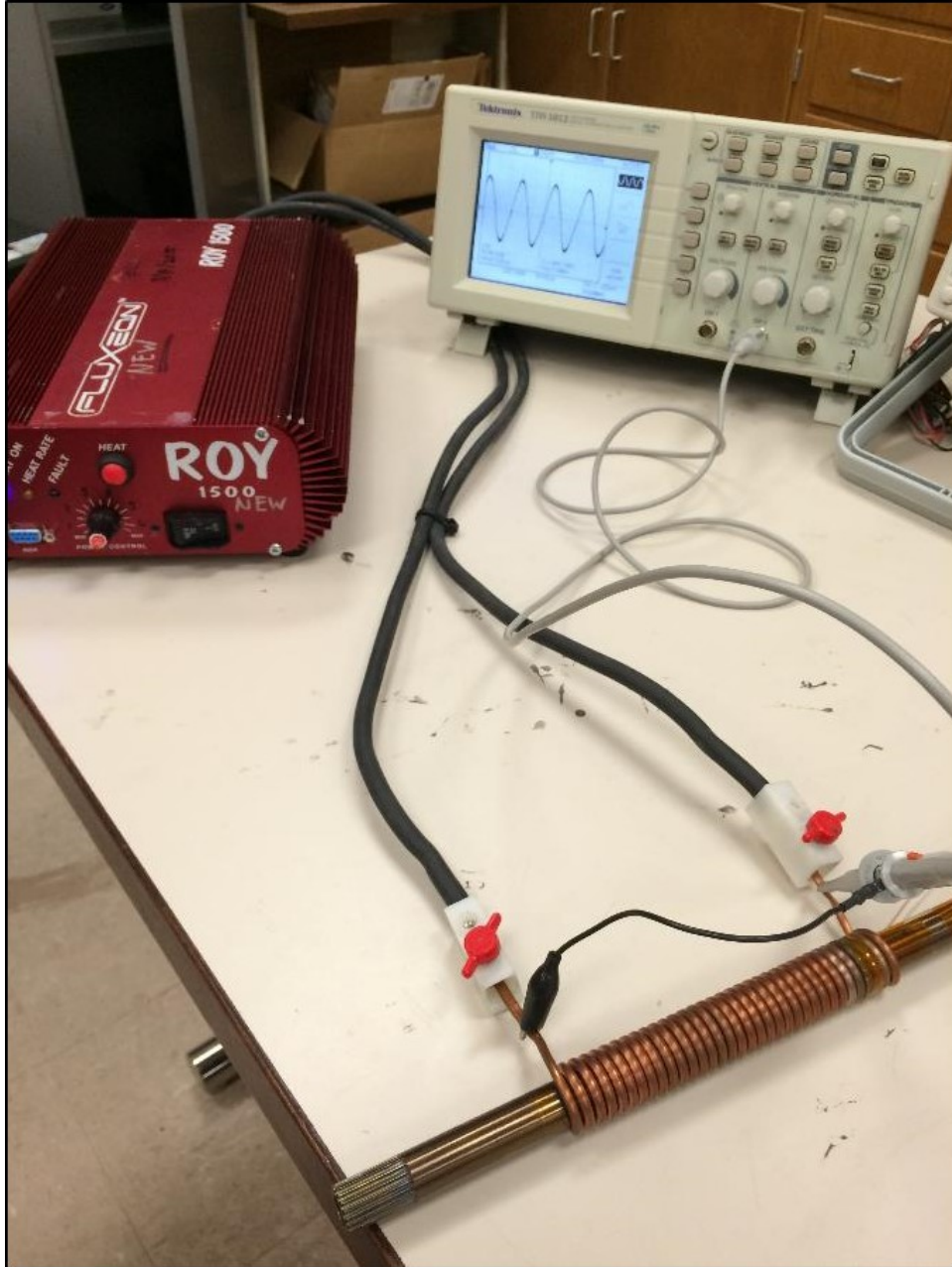


Figure 4.1: Nitinol torque tube with induction coil.

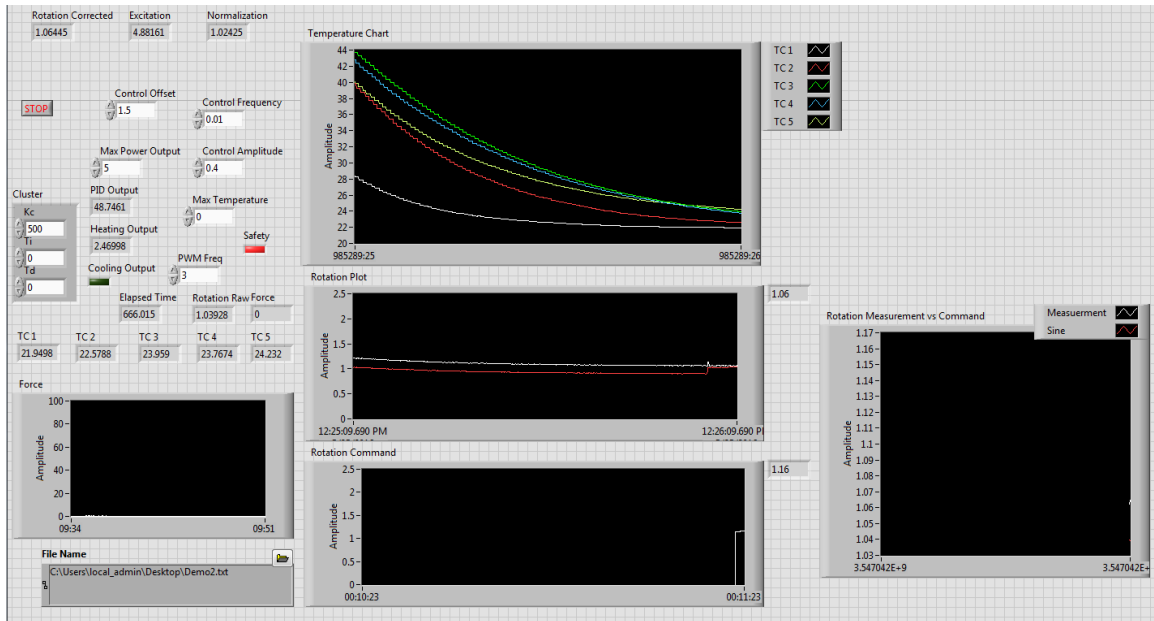


Figure 4.2: LabVIEW control panel.

a potentiometer and temperature by a series of T-type thermocouples. Rotation and force were logged using the NI DAQ card mentioned above and temperatures was recorded using a Measurements Computing USB-TC thermocouple reader. The water-cooled induction coil is comprised of 1/4 inch copper tubing. The LabVIEW interface is shown in Figure 4.2 and the hardware is shown in Figure 4.3.

Using this experimental design, the thermomechanical and control responses of the real-world system were directly compared to the time-domain output of the computational model.

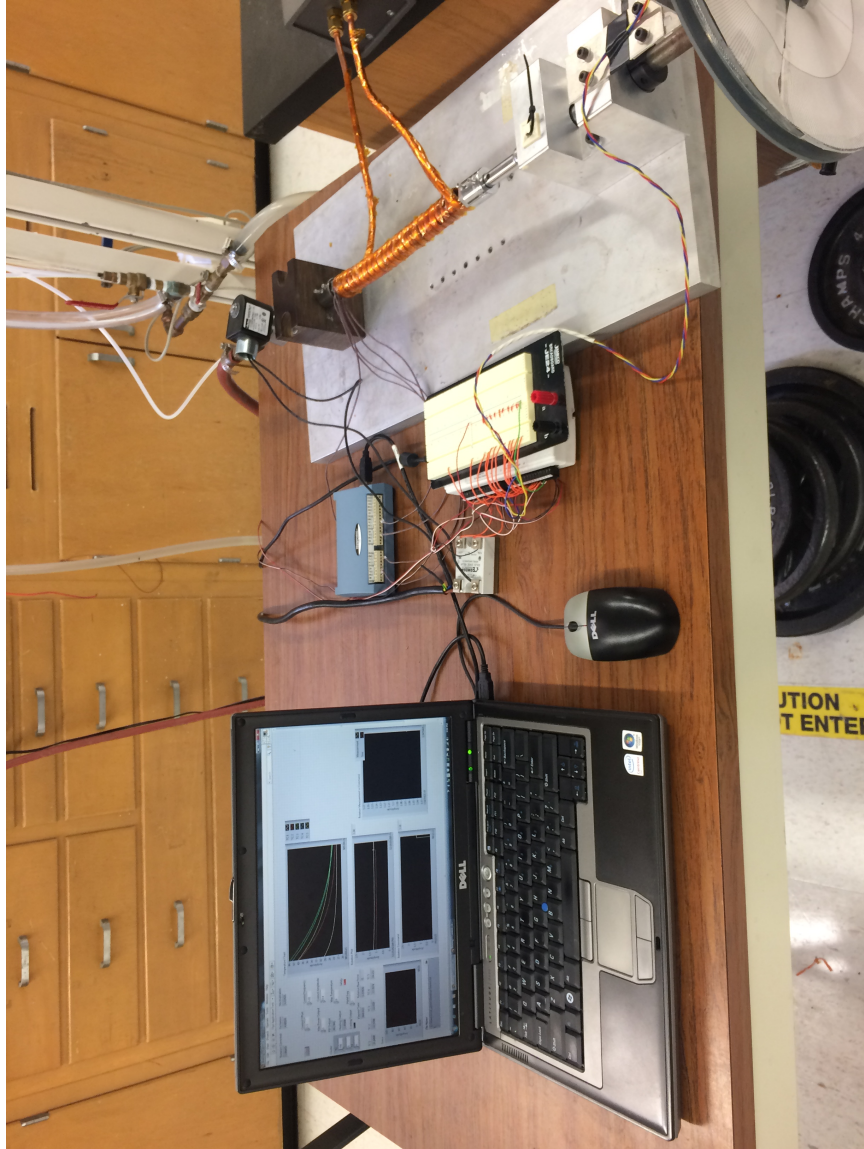


Figure 4.3: Controlled torque tube experimental setup.

5. RESULTS

5.1 Experimental Calibration of Computational Model

Calibration of the computational model is comprised of two parts, calibration of the forced and natural cooling coefficients, and calibration of the SMA model. This was accomplished using the experimental setup described in the previous section. The torque tube shown in Figure 1.3 was used for this experiment. Specifications for the tube are given in Table 5.1

5.1.1 Thermal Calibration

In order to calibrate the forced and natural convection coefficients, the SMA torque tube was heated past the observed austenite start temperature, and permitted to cool. This process was repeated three times for forced convection where the solenoid valve regulating the compressed shop air was set to the open position, and again three times where the solenoid valve remained closed. The results of these cycles are shown in Figures 5.1 and 5.2.

The solution to differential equation 1.6, Equation 1.7, was fit to the cooling portion of these curves.

Table 5.1: Dimensions and published properties of SMA torque tube.

Inside Radius, R_i	2.81 mm
Outside Radius, R_o	4.76 mm
Length, L	203 mm
Mass density, ρ_m	$6450 \frac{kg}{m^3}$
Specific Heat, c	$850 \frac{J}{KgK}$
Poisson Ratio, ν	.33[55]
Electrical Resistivity (A/M), ρ_M	$82/76 \mu\Omega cm$ [55]
Relative Permeability, μ_r	1.002[55]

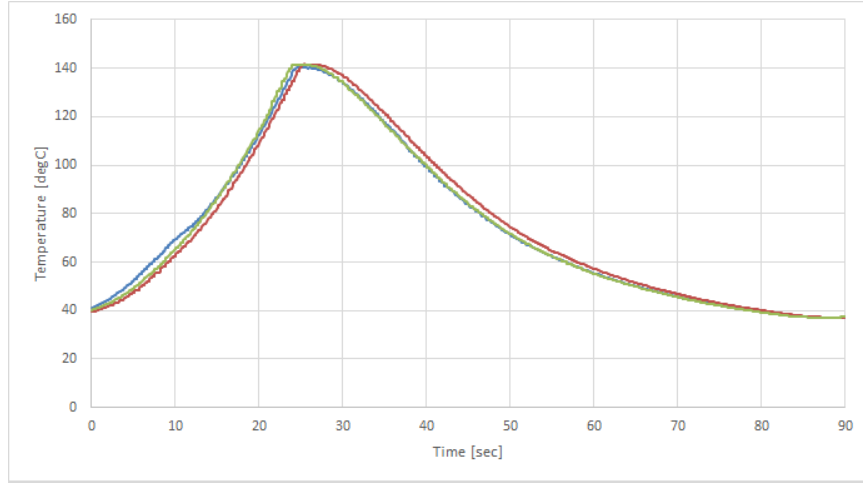


Figure 5.1: Heating and cooling curves for calibration of the forced convective cooling coefficient.

It was found that the exponent of Equation 1.7, $\frac{hA}{c_h h}$, had a value of .0499 for forced convection and a value of .00549 for natural convection.

Using the assumption that all heat loss occurred across the inner surface of the torque tube, the forced convection cooling coefficient, h_{forced} had a value of $695 \frac{W}{M^2K}$ and the natural convection cooling coefficient, $h_{natural}$ had a value of $76.5 \frac{W}{M^2K}$.

5.1.2 SMA Calibration

The elastic and transformation properties and parameters are determined by thermally cycling the SMA at varying load levels and measuring the thermomechanical response. From the data in Figures 5.3 and 5.4 the elastic moduli, transformation temperatures, Clausius-Clapeyron slopes and transformation strains may be determined. The internal state variables of the SMA model are computed as a function of these values.

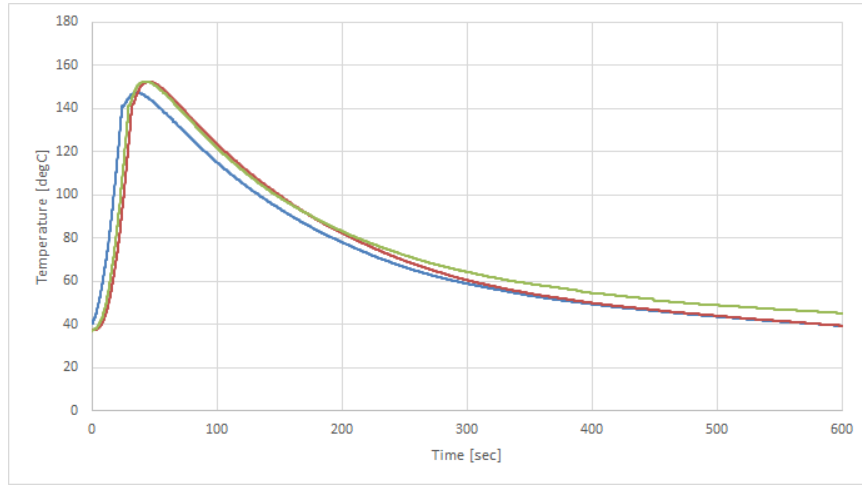


Figure 5.2: Heating and cooling curves for calibration of the natural convective cooling coefficient.

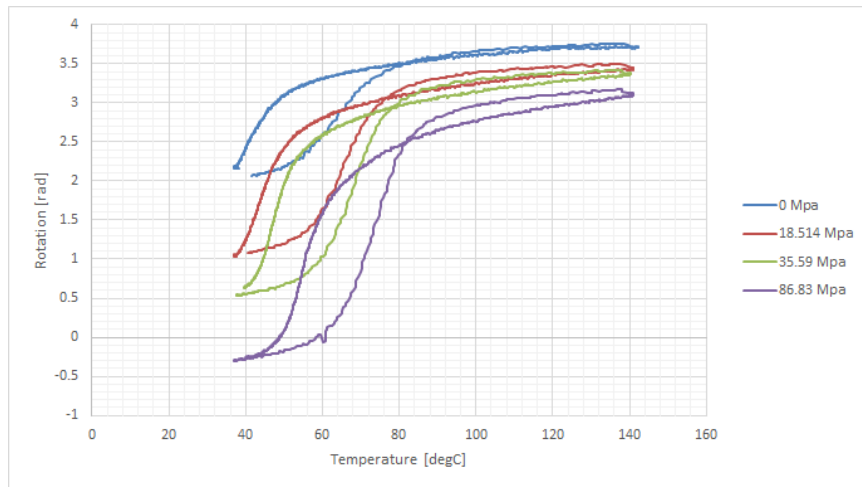


Figure 5.3: Temperature rotation response of the SMA torque tube at various load levels.

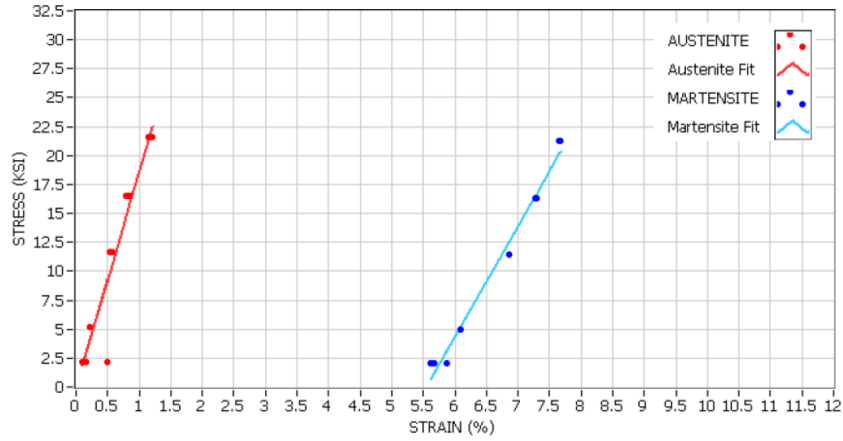


Figure 5.4: Measurement of SMA torque tube actuation modulus. Courtesy of The Boeing Company.

Table 5.2: Calibrated properties of Nitinol torque tube.

A_s	326.1 K
A_f	344.4 K
M_s	321.6 K
M_f	309.4 K
C_M	6.29 MPa/degC
C_A	7.61 MPa/degC
H_{max}	.038
H_{min}	.02
k_t	.0357 MPa^{-1}
σ^{crit}	3.89 MPa

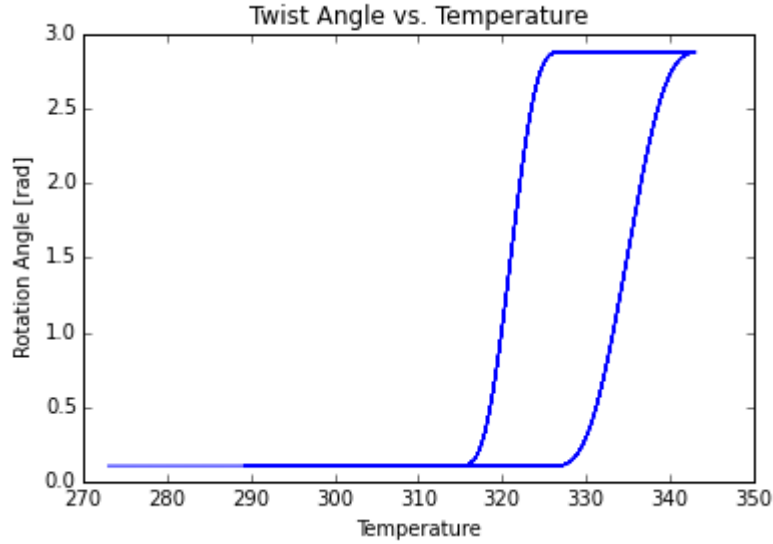


Figure 5.5: Temperature-rotation from computational model with 2.5 N-m applied moment.

5.2 Calibrated Computational Results

Following the characterization of this material, complete and partial cycles of the tested were simulated using the material properties in Table 5.2. A complete cycle is shown in Figure 5.5 and a series of minor transformation loops corresponding to the response in Figure 3.4 without and with the minor loop modification are shown in Figures 5.6 and 5.7. The complete cycle appears to match the calibration data fairly well.

5.3 Computational and Experimental Results for Determination of Electrical Properties

The impedance of an unrelated torque tube supplied by The Boeing Company was characterized using the procedure described in the previous section and compared to that computed by the SEC induction heating model. The measured and computed

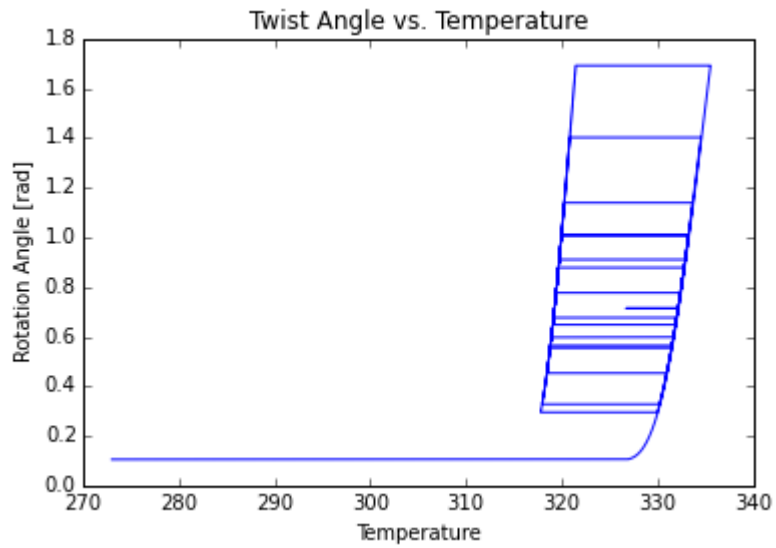


Figure 5.6: Temperature-rotation partial cycle from computational model with 2.5 N-m applied moment without the minor loop modification.

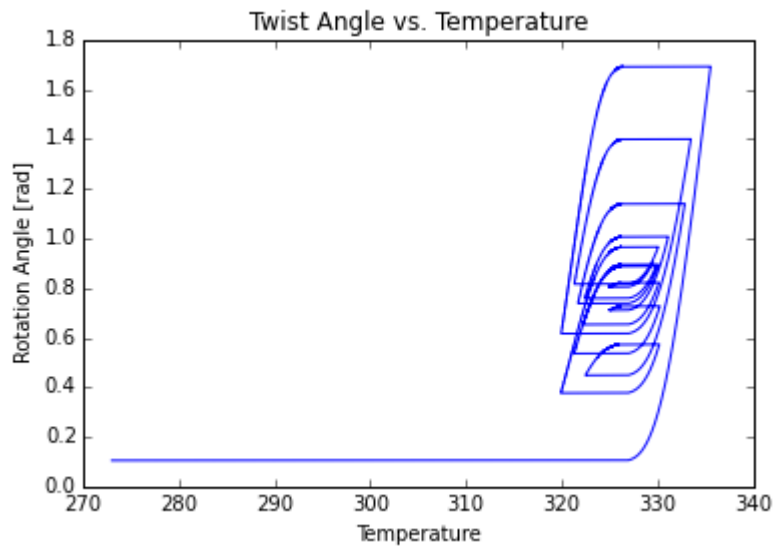


Figure 5.7: Temperature-rotation partial cycle from computational model with 2.5 N-m applied moment and with the minor loop modification.

Table 5.3: Measured torque tube electrical properties.

Sample	Impedance
Measurement 1	457m Ω / \angle 66.16 $^\circ$
Measurement 2	472m Ω / \angle 71.45 $^\circ$
Average Measured	464m Ω / \angle 68.85 $^\circ$
Computed	505m Ω / \angle 70.9 $^\circ$

Table 5.4: Experimental and simulated control parameters.

Parameter	Value
k_P	20
k_I	0
k_D	0
PWM Carrier Frequency [Hz]	3

results are given in Table 5.3.

5.4 Computational and Experimental Results for PID Control of an SMA Torque Tube

The un-optimized control parameters given in Table 5.4 were selected and implemented in both the experiment via LabVIEW and the simulation in Python for comparison in commanding both sine and square wave functions.

The computational rotation and control results for a sine wave input are given in Figures 5.8 and 5.9 and for square wave input in Figures 5.12 and 5.13 while the experimental results are given in Figures 5.10, 5.11, 5.14 and 5.15.

Two interesting features in the comparison between the simulated and experimental responses are the lack of oscillatory control response in the simulation which is present in the experimental result and a somewhat faster transformation on cooling in the square wave response. It has been suggested that these discrepancies are due to the thin wall formulation of the thermomechanical problem, however a

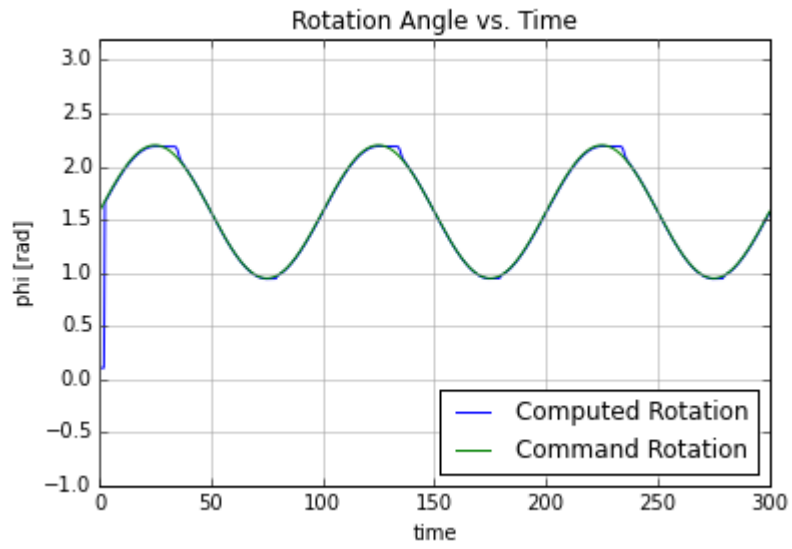


Figure 5.8: Simulated rotation for sine wave input.

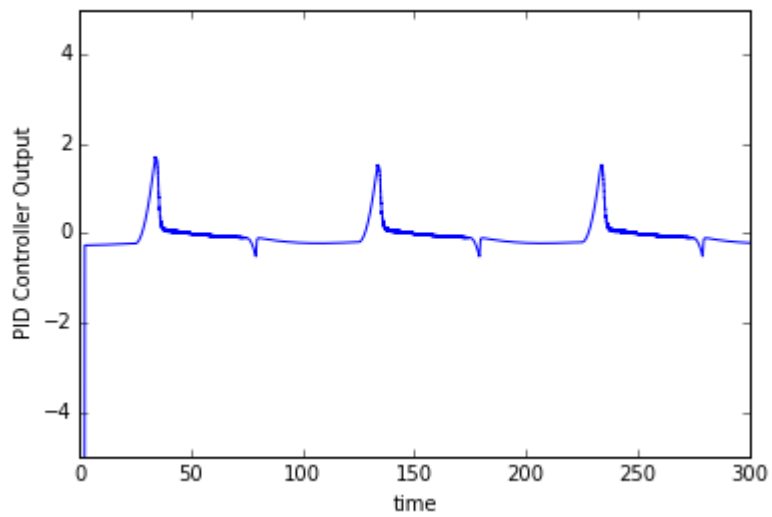


Figure 5.9: Simulated PID controller output for a sine wave input.

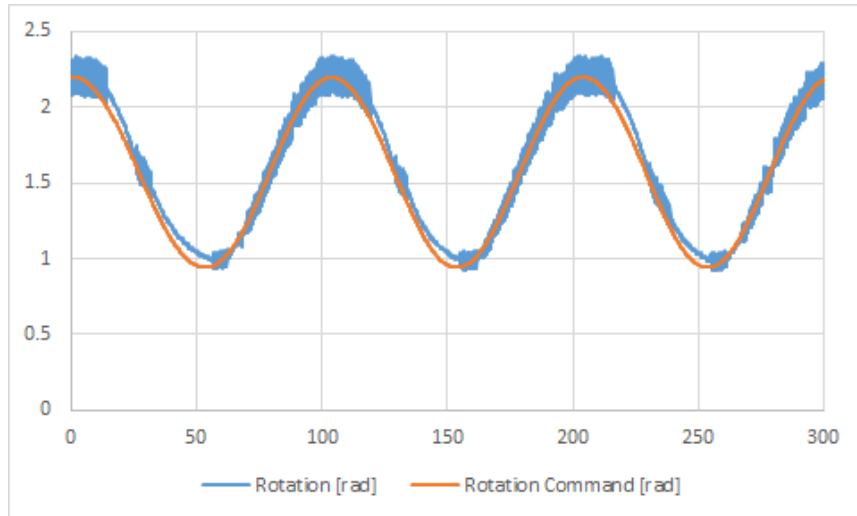


Figure 5.10: Control and measured rotations for a sine wave input.

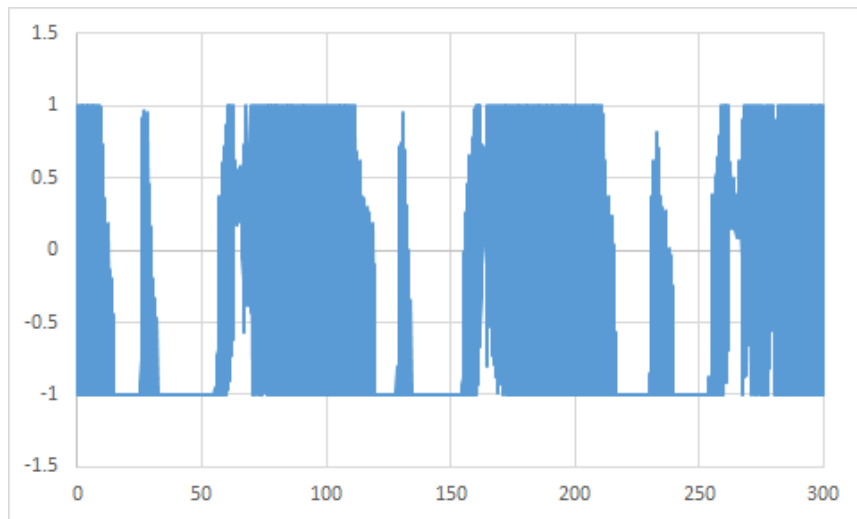


Figure 5.11: LabVIEW PID controller output for a sine wave input.

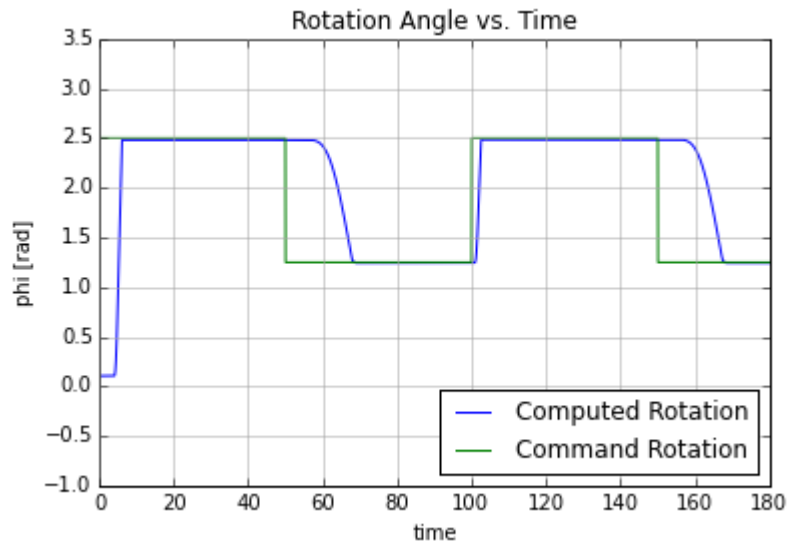


Figure 5.12: Simulated rotation for square wave input.

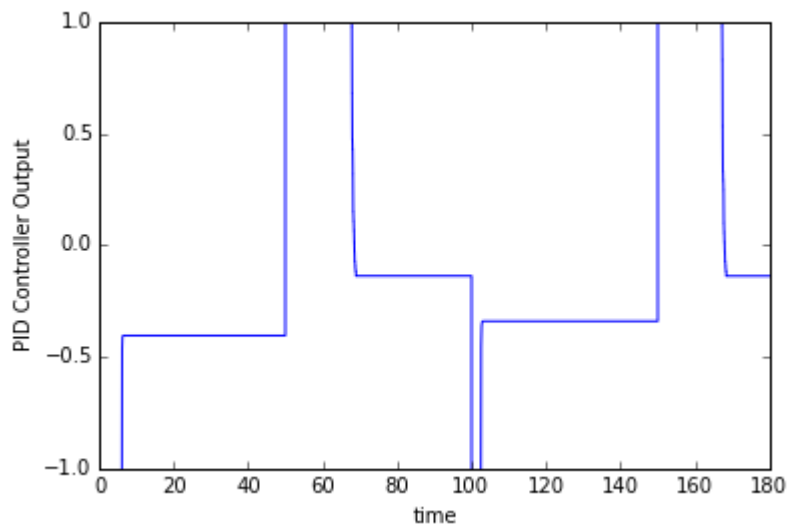


Figure 5.13: Simulated PID controller output for a square wave input.

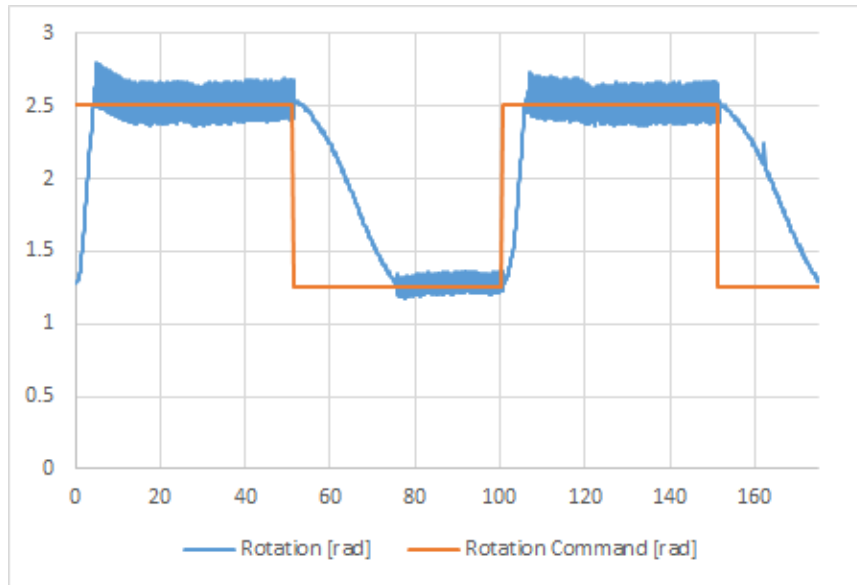


Figure 5.14: Control and measured rotations for a square wave input.

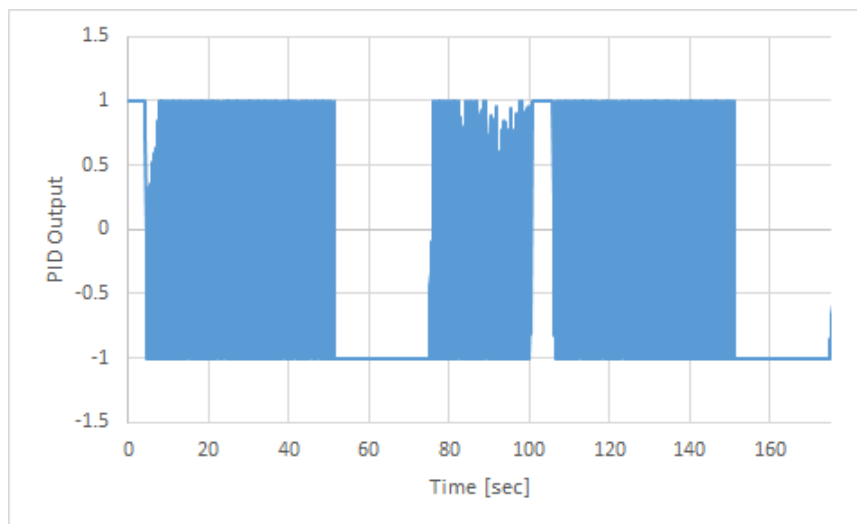


Figure 5.15: LabVIEW PID controller output for a square wave input.

Table 5.5: Controller particle swarm optimization initial seed ranges and lower bounds.

Parameter	Initial Range	Lower Bound
k_P	[0, 10]	0
k_I	[0, 5]	0
k_D	[0, 5]	0
PWM Carrier Frequency [Hz]	[0.1, 3]	0.001

thick-walled solution is beyond the scope of this project and is suggested for future work.

The minor loop modification to the constitutive model is employed for the remainder of this work.

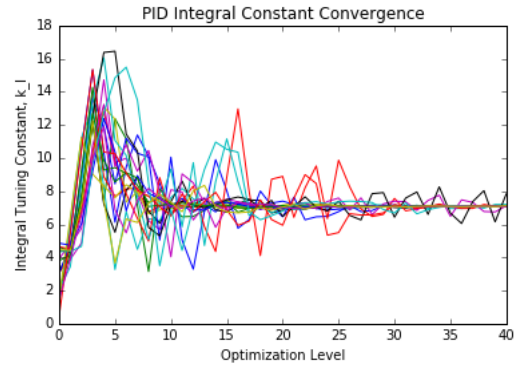
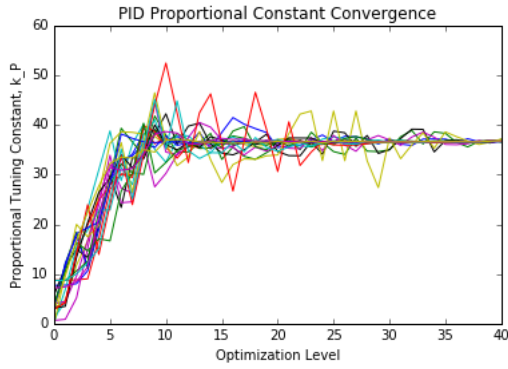
5.5 Controller Optimization Problem

Restricting the optimizer to the geometry and properties of the torque tube given by Table 5.1, frequency of the electrical signal to 200 kHz, and the number of coil turns to 24, optimal control parameters were determined using particle swarm optimization as follows. A 7 N-m/rad aerodynamic loading plus an additional 2.54 N-m constant load to an SMA torque tube with the properties listed in Table 5.2. The convection coefficients used were $695 \frac{W}{m^2K}$ for forced convection and $76 \frac{W}{m^2K}$ for natural convection. The maximum coil current was 125A.

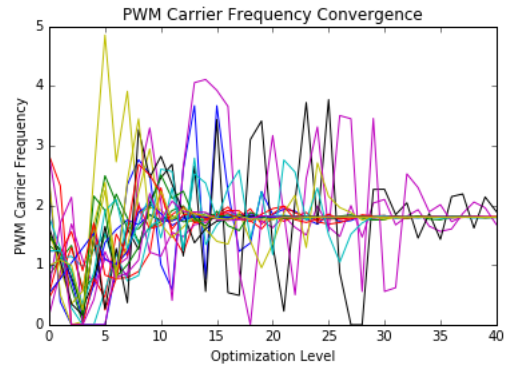
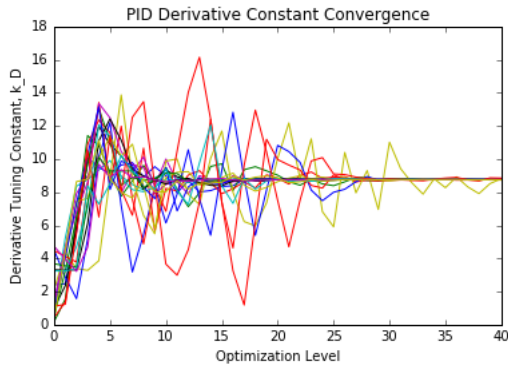
The starting values and lower variable bounds for the optimization were as shown in Table 5.5. The control parameters of the global best solution are given in Table 5.6. Additionally, the convergence of the controller parameters is shown in Figure 5.16.

Table 5.6: Controller particle swarm optimization best solution.

Parameter	Best Solution
k_P	36.623
k_I	7.0841
k_D	8.7595
PWM Carrier Frequency [Hz]	1.8027
RMS Rotation Error	1.9585E-10



(a) PSO convergence for proportional PID tuning constant. (b) PSO convergence for integral PID tuning constant.



(c) PSO convergence for derivative PID tuning constant. (d) PSO convergence for pulse width modulation carrier frequency.

Figure 5.16: Control optimization convergence and simulated rotation.

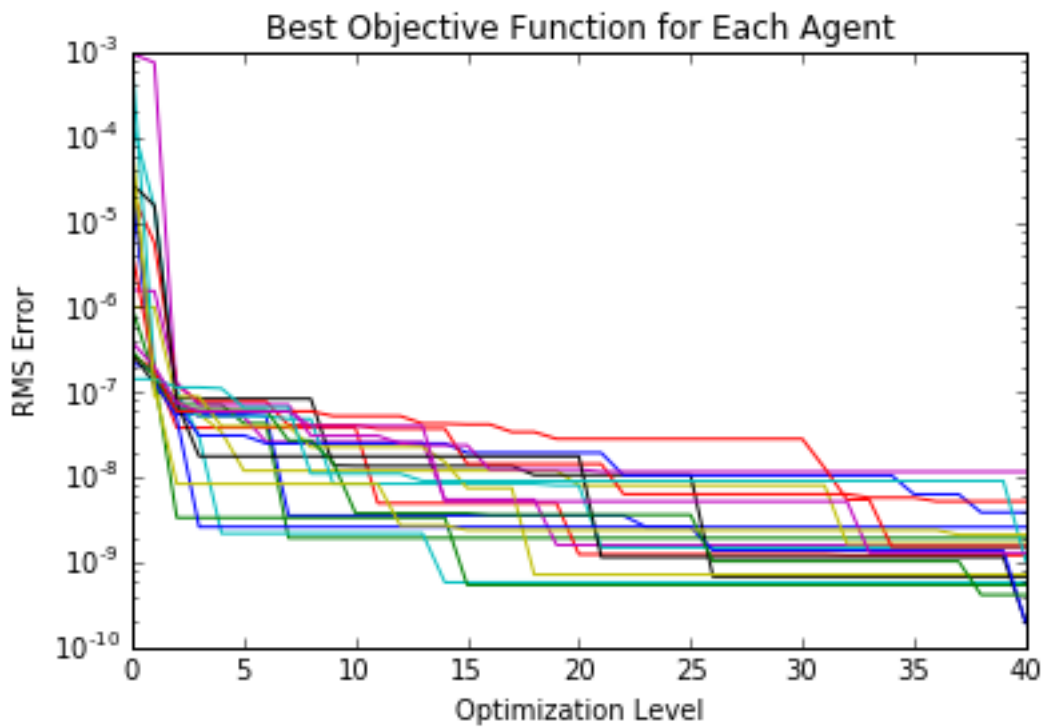
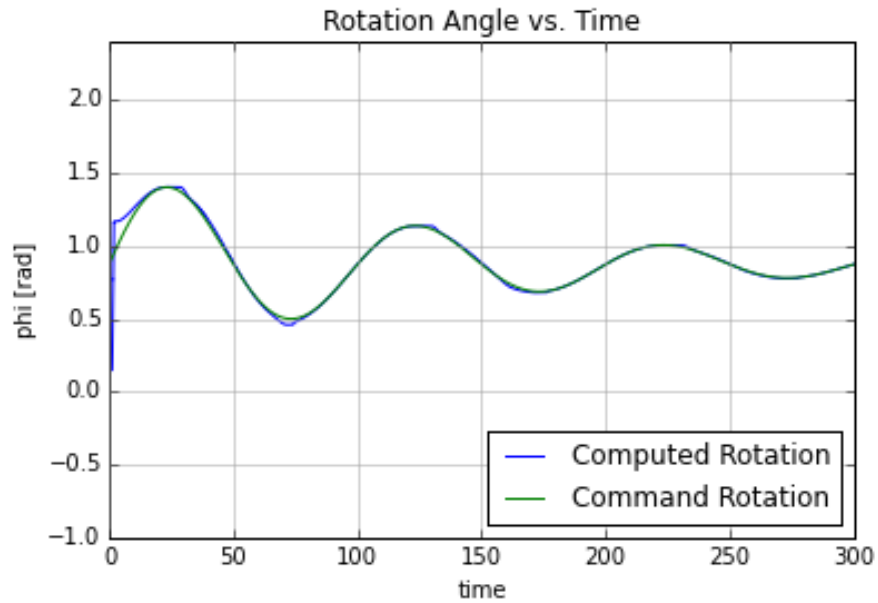
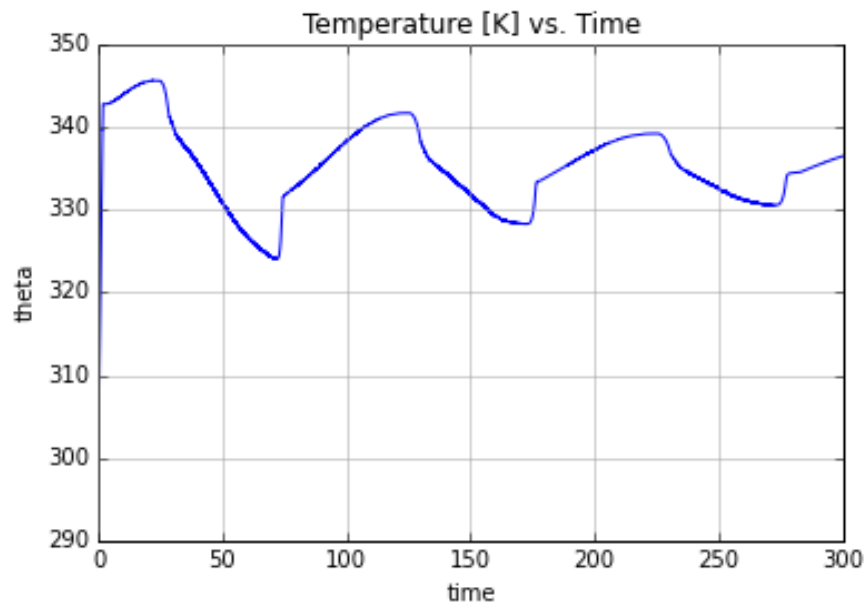


Figure 5.17: Progression of the value of local best objective function for each particle during controller optimization.



(a) Optimized controller rotation response for set torque tube and coil geometry and induction heater frequency.



(b) Optimized controller temperature response for set torque tube and coil geometry and induction heater frequency.

Figure 5.18: Global best solution controlled thermomechanical response following controller optimization.

Table 5.7: Complete particle swarm optimization initial seed ranges and lower bounds.

Parameter	Initial Range	Lower Bound
R_i , [m]	[0.0015, 0.01]	0.0015
t , [m]	[0.0001, 0.01]	0.0001
<i>WireRadius</i> , [m]	[0.00075, 0.002]	0.00075
Electrical Frequency, [kHz]	[10, 300]	0.005
k_P	[0, 10]	0
k_I	[0, 5]	0
k_D	[0, 5]	0
PWM Carrier Frequency [Hz]	[0.1, 3]	0.001

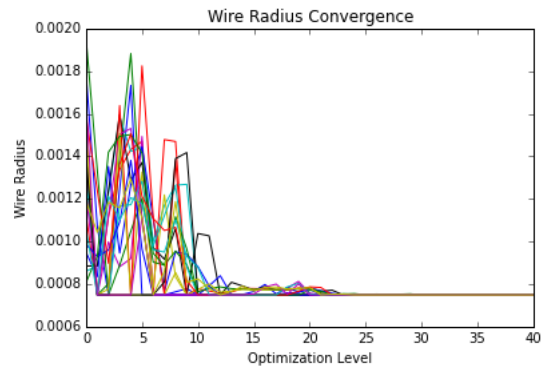
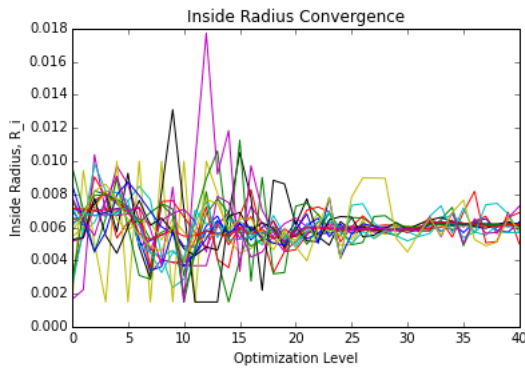
5.6 Complete Optimization Problem

Using this same optimization scheme, the entire torque tube design was optimized, including the geometry and control parameters. Again, a 7 N-m/rad aerodynamic loading plus an additional 2.54 N-m constant load, and the SMA properties listed in Table 5.2 were used. The convection coefficients used were $695 \frac{W}{m^2K}$ for forced convection and $76 \frac{W}{m^2K}$ for natural convection. The maximum coil current was 125A.

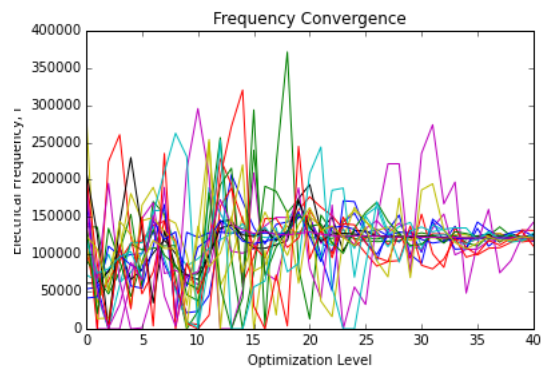
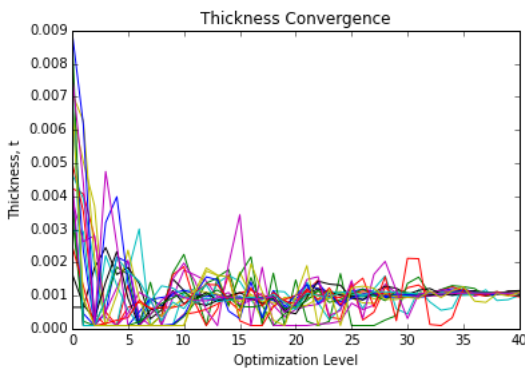
The starting values and lower variable bounds for the optimization were as shown in Table 5.7. Additionally, valid solutions were restricted to a maximum shear stress of 250 MPa, a minimum power factor of .25 and a minimum electrical efficiency of .25. It should be noted that the power factor and electrical efficiency change throughout the simulation as the electrical resistivity is a function of the martensite volume fractions. The dimensions and control parameters of the global best solution are given in Table 5.8. Additionally, the convergence of the torque tube and wire dimensions is shown in Figure 5.19 and that of the controller parameters is shown in Figure 5.20.

Table 5.8: Complete particle swarm optimization best solution.

Parameter	Best Solution
R_i , [m]	.00615
t , [m]	.00108
$WireRadius$, [m]	0.00075
Electrical Frequency, [kHz]	122130
k_P	11.745
k_I	12.962
k_D	0
PWM Carrier Frequency [Hz]	4.2715
RMS Rotation Error	1.8363E-11
Minimum Power Factor	0.6224
Minimum Electrical Efficiency	0.8586

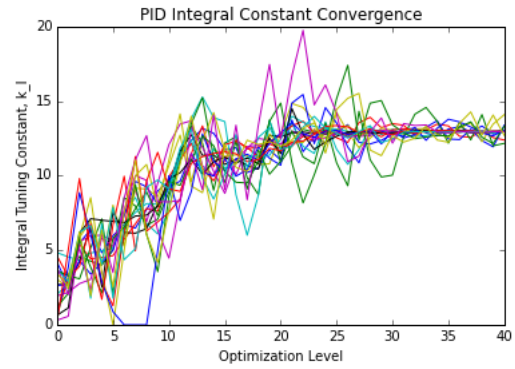
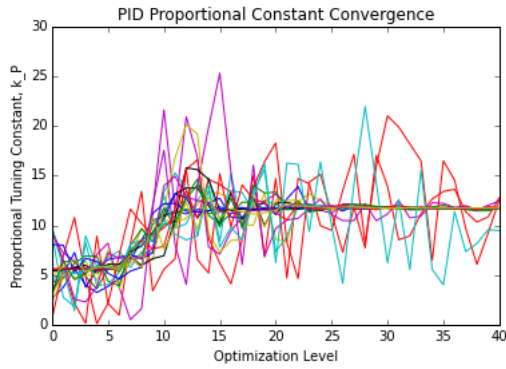


(a) PSO convergence for inside radius of torque tube. (b) PSO convergence induction coil wire radius.

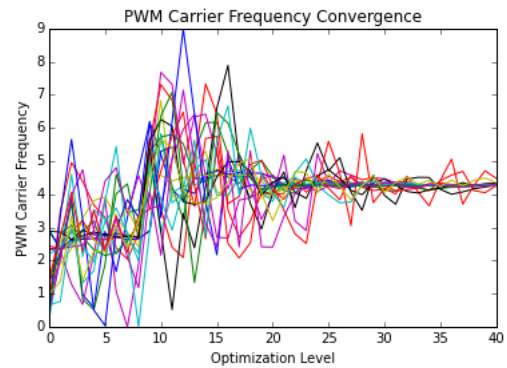
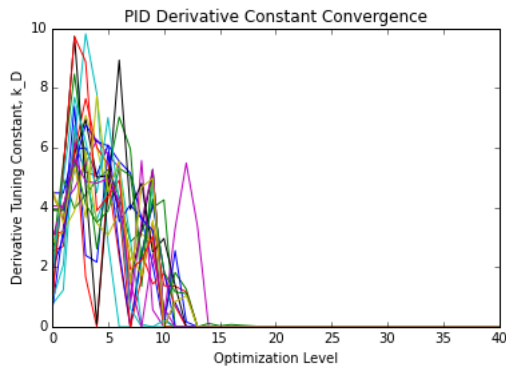


(c) PSO convergence for torque tube thickness. (d) PSO convergence for frequency of current passing through induction coil.

Figure 5.19: Convergence of structural and electrical properties.



(a) PSO convergence for proportional PID tuning constant. (b) PSO convergence for integral PID tuning constant.



(c) PSO convergence for derivative PID tuning constant. (d) PSO convergence for pulse width modulation carrier frequency.

Figure 5.20: Convergence of control parameters and pulse width modulation carrier frequency.

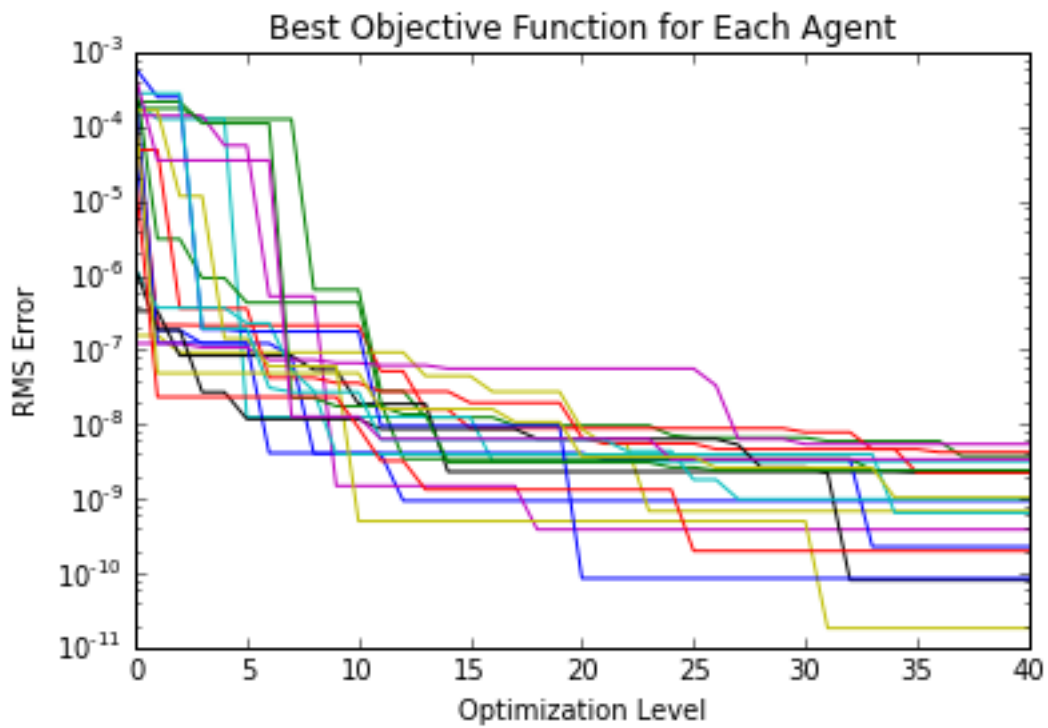
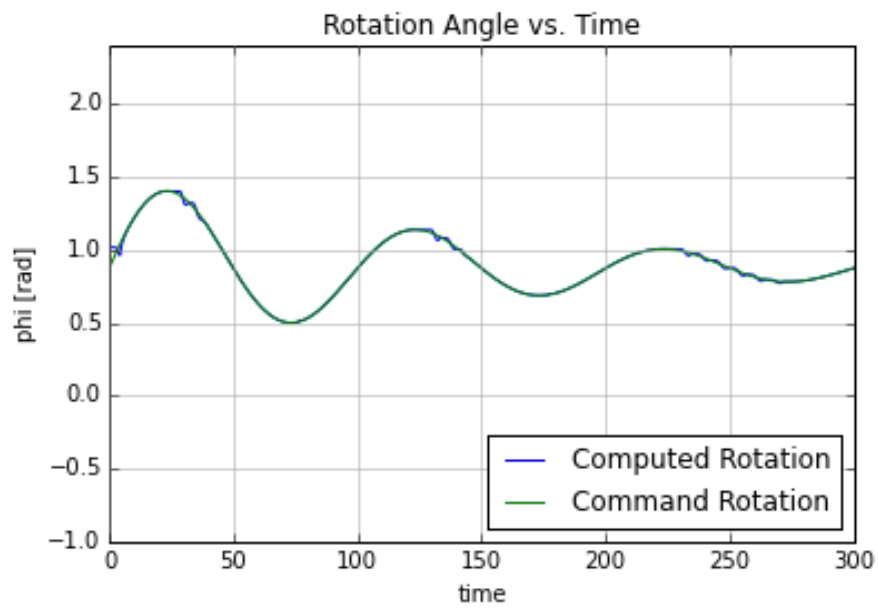
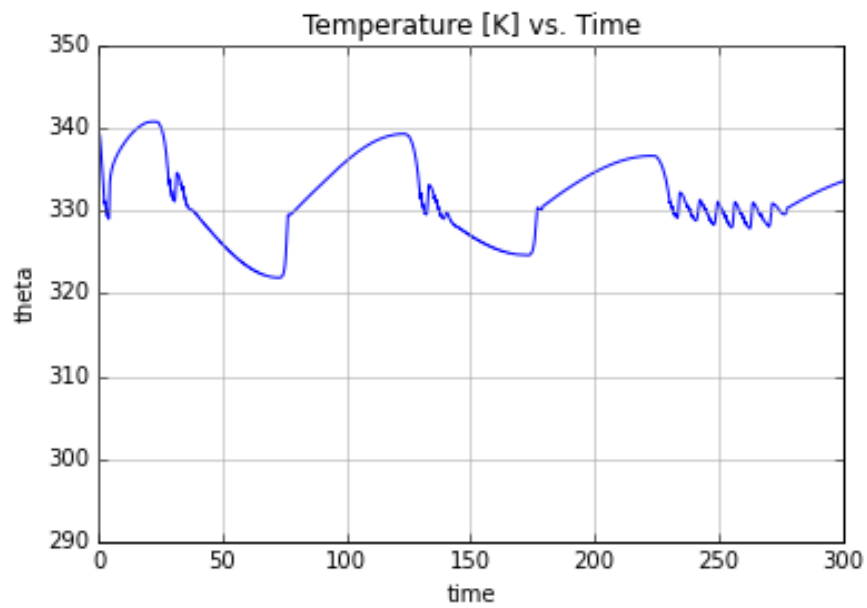


Figure 5.21: Progression of the value of the local best objective function for each particle during complete system optimization.



(a) Simulated rotation response of optimized torque tube and controller with solution data.



(b) Simulated temperature response of optimized torque tube and controller with solution data.

Figure 5.22: Global best solution controlled thermomechanical response following complete system optimization.

5.7 Optimization Discussion

These two optimizations have resulted in torque tube systems of very different geometries and controller parameters. By permitting the geometry to "float" along with the controller tuning constants in pursuit of an optimum control response, the RMS error in the rotation is reduced by an additional order of magnitude. Additionally, the complete system optimization naturally drove to a geometric design with electrical efficiency and power factor well above the imposed design limits.

6. SUMMARY

Shape memory alloy actuators hold great promise for aerospace applications due to their high actuation work density and the potential for simultaneous utility as structural components. One implementation which has been proposed is SMA torque tubes used in place of cables and/or hydraulics as aircraft flap actuators. Previous attempts at implementation and computational studies have shown that the performance of these actuators may be greatly enhanced through the use of induction heating and that for this application, variable loading should be included in any computational model used to simulate the shape memory response. Both of these technical challenges have been addressed and studied in this work.

As part of this thesis, the thermomechanical, electromagnetic and SMA constitutive responses have been modeled in a coupled numerical analysis implemented in the Python programming language. The SMA model was calibrated experimentally and the design of experiments process was utilized to identify important design parameters. The computational modeling was validated against experimental results and limitations of this implementation have been identified. The computational model was finally utilized for design and controller optimization. Throughout this work, special emphasis was placed on the utilization of induction heating and on feedback control.

Bibliography

- [1] NIST/SEMATECH e-Handbook of Statistical Methods, <http://itl.nist.gov/div898/handbook/>, June 2013.
- [2] MathWorks: What is Direct Search?, <http://www.mathworks.com/help/gads/what-is-direct-search.html>, 2016.
- [3] I. H. A. Abbott and A. E. von Doenhoff. *Theory of Wing Sections; Including a Summary of Airfoil Data*. Dover Publications, 1949.
- [4] H. A. Abdelbagi. Skin and Proximity Effects in Two Parallel Plates. Master's thesis, Wright State University, 2007.
- [5] M. J. Anderson and P. J. Whitcomb. *DOE Simplified : Practical Tools for Effective Experimentation*. Boca Raton, FL : CRC Press, [2007], 2015.
- [6] R. M. Baker. Design and Calculation of Induction-Heating Coils. *Trans. AIEE, Part II: Applicat. Ind. Transactions of the American Institute of Electrical Engineers, Part II: Applications and Industry*, 76(1):3140, 1957.
- [7] T. Baumeister and L. S. Marks. *Standard Handbook for Mechanical Engineers*. McGraw-Hill, 1967.
- [8] B. C. Bayer, S. Sanjabi, C. Baetz, C. T. Wirth, S. Esconjauregui, R. S. Weatherup, Z. H. Barber, S. Hofmann, and J. Robertson. Carbon Nanotube Forest Growth on NiTi Shape Memory Alloy Thin Films for Thermal Actuation. *Thin Solid Films*, 519(18):6126 – 6129, 2011.
- [9] L. Bayn, J. M. Grau, M. M. Ruiz, and P. M. Surez. An Algorithm for Bang-Bang Control of Fixed-Head Hydroplants. *International Journal of Computer Mathematics*, 88(9):19491959, 2011.

- [10] C. L. Bertagne, D. J. Hartl, and T. J. Cognata. Analysis of Highly Coupled Thermal-Structural Responses in Morphing Radiative Bodies. *23rd AIAA/AHS Adaptive Structures Conference*, Feb 2015.
- [11] P. Blockley. Skin Effect, http://peterblockley.tripod.com/documents/skin_effect.pdf, November 2003.
- [12] Z. Bo and D. C. Lagoudas. Thermomechanical Modeling of Polycrystalline SMAs Under Cyclic Loading, Part IV: Modeling of Minor Hysteresis Loops. *International Journal of Engineering Science*, 37:1205 – 1249, 1999.
- [13] J. G. Boyd and D. C. Lagoudas. A Thermodynamical Constitutive Model for Shape Memory Materials. Part I. The Monolithic Shape Memory Alloy. *International Journal of Plasticity*, 12(6):805842, 1996.
- [14] L. C. Brinson, A. Bekker, and S. Hwang. Deformation of Shape Memory Alloys Due to Thermo-Induced Transformation. *Journal of Intelligent Material Systems and Structures*, 7(1):97–107, 1996.
- [15] F. T. Calkins and J. H. Mabe. Shape Memory Alloy Based Morphing Aerostructures. *Journal of Mechanical Design*, 132(11):111012, 2010.
- [16] F. T. Calkins, J. H. Mabe, and R. T. Ruggeri. Overview of Boeing’s Shape Memory Alloy Based Morphing Aerostructures. In *ASME 2008 Conference on Smart Materials, Adaptive Structures and Intelligent Systems*, pages 885–895. American Society of Mechanical Engineers, 2008.
- [17] Y. A. Çengel and J. M. Cimbala. *Fluid Mechanics: Fundamentals and Applications*. McGraw-Hill series in mechanical engineering. McGraw-Hill Higher Education, 2006.

- [18] V. Czitrom. One-Factor-at-a-Time versus Designed Experiments. *The American Statistician*, (2):126, 1999.
- [19] J. Davies and P. Simpson. *Induction Heating Handbook*. McGraw-Hill, 1979.
- [20] S. M. Dutta and F. H. Ghorbel. Differential Hysteresis Modeling of a Shape Memory Alloy Wire Actuator. *IEEE/ASME Transactions on Mechatronics*, 10(2):189–197, 2005.
- [21] J. D. Ertel and S. A. Mascaro. Dynamic Thermomechanical Modeling of a Wet Shape Memory Alloy Actuator. *Journal of Dynamic Systems Measurement and Control-Transactions of the ASME*, 132(5), 2010.
- [22] D. H. Friedman, S. Bieniawski, and D. J. Hartl. Simulation and Control Design for Shape Memory Alloy Torque Tubes. *ASME 2011 Conference on Smart Materials, Adaptive Structures and Intelligent Systems, Volume 2*, 2011.
- [23] Z.-L. Gaing. A Particle Swarm Optimization Approach for Optimum Design of PID Controller in AVR System. *IEEE Transactions on Energy Conversion IEEE Trans. On Energy Conversion*, 19(2):384391, 2004.
- [24] C. F. Gerald and P. O. Wheatley. *Applied Numerical Analysis. 7th ed.* Boston : Pearson/Addison-Wesley, 2004.
- [25] D. J. Hartl and D. C. Lagoudas. Aerospace applications of shape memory alloys. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 221(4):535552, Jan 2007.
- [26] D. J. Hartl, D. C. Lagoudas, F. T. Calkins, and J. H. Mabe. Use of a Ni60Ti Shape Memory Alloy for Active Jet Engine Chevron Application: I. Thermo-mechanical Characterization. *Smart Materials and Structures*, 19(1), 2010.

- [27] P. G. Hill and C. R. Peterson. *Mechanics and Thermodynamics of Propulsion*. Addison-Wesley Publishing Company, 2nd edition, 1992.
- [28] C. A. Holt. *Introduction to Electromagnetic Fields and Waves*. Wiley, 1963.
- [29] R. Hooke and T. A. Jeeves. "Direct Search" Solution of Numerical and Statistical Problems. *J. ACM*, 8(2):212–229, April 1961.
- [30] S. Huang, M. Leary, T. Ataalla, K. Probst, and A. Subic. Optimisation of Ni-Ti Shape Memory Alloy Response Time by Transient Heat Transfer Analysis. *Materials and Design*, 35:655–663, 2012.
- [31] P. V. Hull, S. L. Canfield, and C. Carrington. A Radiant Energy-Powered Shape Memory Alloy Actuator. *Mechatronics*, 14:757 – 775, 2004.
- [32] National Instruments. *LabVIEW PID and Fuzzy Logic Toolkit User Manual*. Austin, Texas, June 2009.
- [33] A. P. Jardine, J. D. Bartley-Cho, and J. S. Flanagan. Improved Design and Performance of the SMA Torque Tube for the DARPA Smart Wing Program. *Smart Structures and Materials 1999: Industrial and Commercial Applications of Smart Structures Technologies*, Sep 1999.
- [34] M. A. Johnson, M. H. Moradi, and J. Crowe. *PID Control: New Identification and Design Methods*. Springer, 2005.
- [35] A. Kangas, J. Kangas, and M. Kurttila. *Decision Support for Forest Management*. Springer, 2008.
- [36] J. Kennedy and R. Eberhart. Particle Swarm Optimization. In *Proceedings of IEEE International Conference on Neural Networks, 1995.*, volume 4, pages 1942–1948 vol.4, Nov 1995.

- [37] J. F. Kennedy, R. C. Eberhart, and Y. Shi. *Swarm Intelligence*. Morgan Kaufmann Publishers, 2001.
- [38] T. G. Kolda, R. M. Lewis, and V. Torczon. Optimization by Direct Search: New Perspectives on Some Classical and Modern Methods. *SIAM Rev. SIAM Review*, 45(3):385482, 2003.
- [39] D. C. Lagoudas. *Shape Memory Alloys: Modeling and Engineering Applications*. Springer, 2008.
- [40] D. C. Lagoudas, D. Hartl, Y. Chemisky, L. Machado, and P. Popov. Constitutive Model for the Numerical Analysis of Phase Transformation in Polycrystalline Shape Memory Alloys. *International Journal of Plasticity*, 32-33:155183, 2012.
- [41] D. C. Lagoudas, J. J. Mayes, and M. M. Khan. Simplified Shape Memory Alloy (SMA) Material Model for Vibration Isolation. In *Proceedings of SPIE - The International Society for Optical Engineering*, volume 4326, pages 452–461, Aerospace Engineering Department, Texas A&M University, 2001.
- [42] S. Lupi, M. Forzan, and A. Aliferov. *Induction and Direct Resistance Heating: Theory and Numerical Modeling*. Springer International Publishing, 2014.
- [43] J. H. Mabe, F. T. Calkins, G. S. Bushnell, and S. R. Bieniawski. Aircraft Systems with Shape Memory Alloy (SMA) Actuators, and Associated Methods, February 1 2011. US Patent 7,878,459.
- [44] S. Maeda, K. Abe, K. Yamamoto, and O. Tohyama. Active Endoscope with SMA (Shape Memory Alloy) Coil Springs. In *Annual International Workshop on Micro Electro Memchanical Systems*, Micro electro mechanical systems: an investigation of micro structures, sensors, actuators, machines and systems, pages 290 – 295, 1996.

- [45] E. Martz. Getting Started with Factorial Design of Experiments (DOE) <http://blog.minitab.com/blog/understanding-statistics/getting-started-with-factorial-design-of-experiments-doe>, March 2013.
- [46] S. M. Mikki and A. A. Kishk. Particle Swarm Optimization: A Physics-Based Approach. *Synthesis Lectures on Computational Electromagnetics*, 3(1):1103, 2008.
- [47] A. K. Noor, S. L. Venneri, D. B. Paul, and M. A. Hopkins. Structures Technology for Future Aerospace Systems. *Computers & Structures*, 74(5):507–519, 2000.
- [48] A. Pathak, D. Brei, and J. Luntz. Experimental Characterization of the Convective Heat Transfer from Shape Memory Alloy (SMA) Wire to Various Ambient Environments. In *Student's papers*, page 12 pp, Ascona, Switzerland, October 6-9, 2008 2008. Pathak,A.; Mechanical Engineering, University of Michigan, 2250 GG Brown, Ann Arbor, MI 48109-2126, USA, apathak@umich.edu.
- [49] Z. B. Popovic and B. D. Popovic. *Introductory Electromagnetics: Practice, Problems and Labs*. Prentice Hall, 2000.
- [50] P. Pranda, R. A. Gagliano, and T. M. K. Thampan. Heat Transfer in SMR Tubes, September 10 2013. US Patent 8,529,849.
- [51] J. Qiu, J. Tani, D. Osanai, Y. Urushiyama, and D. Lewinnek. High-Speed Response of SMA Actuators. *International Journal of Applied Electromagnetics & Mechanics*, 12(1/2):87, 2000.
- [52] C. R. Reeves, editor. *Modern Heuristic Techniques for Combinatorial Problems*. John Wiley & Sons, Inc., New York, NY, USA, 1993.

- [53] Gilberto Reynoso-Meza, Xavier Blasco, Javier Sanchis, and Miguel Martinez. Controller Tuning Using Evolutionary Multi-Objective Optimisation: Current Trends and Applications. *Control Engineering Practice*, 28:5873, 2014.
- [54] J. Robinson and Y. Rahmat-Samii. Particle Swarm Optimization in Electromagnetics. *IEEE Trans. Antennas Propagat. IEEE Transactions on Antennas and Propagation*, 52(2):397407, 2004.
- [55] R. N. Saunders. Modeling the Effects of Induction Heating on Arbitrary Shape Memory Alloy Components. Master's thesis, Texas A&M University, College Station, TX, 2015.
- [56] R. N. Saunders, J. G. Boyd, D. J. Hartl, J. K. Brown, F. T. Calkins, and D. C. Lagoudas. A Validated Model for Induction Heating of Shape Memory Alloy Actuators. *Smart Materials and Structures*, 25(4):045022, 2016.
- [57] S. Seelecke and I. Muller. Shape Memory Alloy Actuators in Smart Structures: Modeling and Simulation. *Applied Mechanics Reviews Appl. Mech. Rev.*, 57(1):23, 2004.
- [58] J. J. E. Slotine and W. Li. *Applied Nonlinear Control*. Prentice Hall, 1991.
- [59] M. Sreekumar, M. Singaperumal, T. Nagarajan, M. Zoppi, and R. Molfino. Recent Advances in Nonlinear Control Technologies for Shape Memory Alloy Actuators. *Journal of Zhejiang University-SCIENCE A*, 8(5):818–829, 2007.
- [60] W. Stadler. *Multicriteria Optimization in Engineering and in the Sciences*. Plenum Press, 1988.
- [61] R. S. Subramanian. Heat Transfer To or From a Fluid Flowing Through a Tube. <http://web2.clarkson.edu/projects/subramanian/ch302/notes/conv1.pdf>. *Clarkson University*, 2007.

- [62] B. Sun, J. Zhang, X. Wu, and W. Liao. Single-Joint Driving System of Bionic Finger Based on Shape Memory Alloy. In *Micro-NanoMechatronics and Human Science (MHS), 2012 International Symposium on*, pages 225–228, Nov 2012.
- [63] J. Sun. *Dynamics and Control of Switched Electronic Systems: Advanced Perspectives for Modeling, Simulation and Control of Power Converters*, chapter Pulse-Width Modulation, pages 25–61. Springer London, London, 2012.
- [64] Y. Tadesse, A. Villanueva, S. Priya, C. Haines, D. Novitski, and R. Baughman. Hydrogen-Fuel-Powered Bell Segments of Biomimetic Jellyfish. *Smart Materials and Structures*, 21(4), 2012.
- [65] L. Takau and P. Bodger. Low Frequency Modelling of Induction Heaters Using Series Equivalent Circuit, Transformer Equivalent Circuit and Finite Element Analysis. *2013 Australasian Universities Power Engineering Conference (AUPEC)*, 2013.
- [66] B. Thidé. *Electromagnetic field theory*. Upsilon Books Uppsala, Sweden, 2004.
- [67] C. Thill, J. Etches, I. Bond, K. Potter, and P. Weaver. Morphing Skins. *The Aeronautical Journal*, 112(1129):117–139, 2008.
- [68] T. L. Turner, C. L. Lach, and R. J. Cano. Fabrication and Characterization of SMA Hybrid Composites. In *Proceedings of SPIE - The International Society for Optical Engineering*, volume 4333, pages 343–354, NASA Langley Research Center, Structural Acoustics Branch, Mail Stop 463, 2001.
- [69] M. Uy and J. K. Telford. Optimization by Design of Experiment Techniques. *2009 IEEE Aerospace Conference*, 2009.
- [70] R. T. Watkins, J. A. Shaw, and D. S. Grummon. Shape Memory Thermal Lag and Superelastic Rate Sensitivity of SMA Cellular Structures. In *Proceed-*

ings of SPIE - The International Society for Optical Engineering, volume 8689, University of Michigan, 2013.

- [71] J. Webster. Shape Memory Material Actuation, January 5 2006. US Patent App. 11/168,545.
- [72] O. Yeniay. Comparative Study of Algorithms for Response Surface Optimization. *Mathematical & Computational Applications*, 19(1):93 – 104, 2014.
- [73] X. Zhang, H. Liu, X. Wang, L. Dong, Q. Wu, and R. Mohan. Speed and Convergence Properties of Gradient Algorithms for Optimization of IMRT. *Medical Physics*, 31(5), 2004.

APPENDIX A

PARTICLE SWARM OPTIMIZER WITH SMA CONSTITUTIVE MODEL IN PYTHON

```
1 from __future__ import division
2 from numpy import *
3 from math import *
4 from sys import *
5 import matplotlib.pyplot as plt
6 pi = 3.14159265359
7 mu_0 = 4*pi*10**-7 # Permeability of Free Space, H/m
8
9 # This program has been prepared to simulate the time-
   domain, control response of two-way shape memory
   alloy torque tubes.
10
11 # The following features have been included in this
   simulation:
12 #   - Built-in PID control based on rotation
13 #   - Torsional spring loading meant to approximate
   aerodynamic loading
14 #   - Series equivalent circuit model for induction
   heating
15 #       - Electrical efficiency
16 #       - Power factor
17 #       - Electrical impedance
18 #       - Power into SMA torque tube
19 #   Full electro-thermomechanical coupling of material
   properties
20
21 # The constitutive model authored by Dimitris Lagoudas,
   Darren Hartl, Yves Chemisky, Luciano Machado, Peter
   Popov and published in International Journal of
   Plasticity 32-33 (2012) 155-183 was used in the
   creation of this code.
22
23 # John L. Rohmer
24 # Graduate Student
25 # Dept. of Aerospace Engineering
```

```

26 # Texas A&M University
27 # Cell: 940-704-5898
28 # john.rohmer@tamu.edu
29
30
31 #Format Output Files
32 fout = open('PSOData.txt', 'w')
33 fout.write('Run, R_i, t, Wire_Diameter, Frequency, kP,
            kI, kD, PWM_Frequency, RMS_Error, Max_Rotation_Angle
            , Min_Power_Factor, Min_Efficiency\n')
34 fout.close()
35 fout = open('PSOBest.txt', 'w')
36 fout.write('Level, R_i, t, Wire_Diameter, Frequency, kP
            , kI, kD, PWM_Frequency, RMS_Error, Max_Rotation
            Angle, Min_Power_Factor, Min_Efficiency\n')
37 fout.close()
38 runcount = 0
39
40 # Set Command Mode
41     # CM = 1: Complete cycles
42     # CM = 2: Arbitrary rotation control
43 CM = 2
44
45 # Set command rotation function
46 def command(t, runtime):
47     # CM == 1: Complete Cycles
48     if CM == 1:
49         UPC = 10
50         LPC = -10
51         if t == 0: return UPC, runtime
52     else:
53         if phic_vec[-1] == UPC:
54             if xi[-1] <= .001: return LPC, runtime
55             else: return UPC, runtime
56         if phic_vec[-1] == LPC:
57             if xi[-1] >= .999:
58                 if runtime == 0: runtime = t
59                 return UPC, runtime
60             else: return LPC, runtime
61     # CM == 2: Arbitraty control mode
62     if CM == 2:
63         return (2.25+1)/2-.75 + (2.25-1)/2*sin

```

```

        (.01*2*3.14159*t)*2**(-.01*t), runtime
64
65 # Use Minor Loop Modification
66 # 0: no, 1: yes
67 MLM = 1
68
69 #Set cooling mode
70 mode = 2
71     #Mode 1: Continuous control
72     #Mode 2: Pulse width modulation
73     #If mode == 2 set PWM carrier frequency,
        carrierfreq
74
75 #####
76 ### Optimization Settings ###
77 #####
78
79 Npar = 8           # Number of parameters being optimized
80 Nagents = 20 #Number of particles traversing the design
        hyperspace
81 Nlevels = 40 #Number of iterations
82
83 w = 1 # Interial Factor
84 c1 = 2 # Cognitive Factor
85 c2 = 2 # Social Factor
86
87
88 #####
89 ### Constraints ###
90 #####
91
92 maximumstress = 250000000
93 minimumpowerfactor = .25
94 minimumelectricalefficiency = .25
95
96 #Variable Sequence
97     #0. R_i
98     #1. t
99     #2. wire radius
100     #3. frequency
101     #4. k_p
102     #5. k_i

```

```

103     #6. k_d
104     #7. PWM Frequency
105 #Initial Variable Ranges
106
107 seedrange = [ [.0015, .01],
108               [.0001, .01],
109               [.00075, .002],
110               [10000, 300000],
111               [0, 10],
112               [0, 5],
113               [0, 5],
114               [.1, 3] ]
115
116 #Parameter Bounds
117 LBoundR_i = .0015 #0. R_i
118 LBoundt   = .0001 #1. t
119 LBoundWire = .00075 #2. wire radius
120 LBoundf   = 5 #3. frequency
121 LBoundk_P = 0 #4. k_p
122 LBoundk_I = 0 #5. k_i
123 LBoundk_D = 0 #6. k_d
124 LBoundPWM = .001 #7. PWM Frequency
125
126 #####
127 ### Optimization Initial Conditions ###
128 #####
129 #Generate Data Matricies Full of Zeros
130 position = [[0 for agent in range(Nagents)] for level
131             in range(Nlevels+1)] for parameters in range(Npar)]
132 velocity = [[0 for agent in range(Nagents)] for level
133             in range(Nlevels+1)] for parameters in range(Npar)]
134 objective = [[0 for agent in range(Nagents)] for level
135             in range(Nlevels+1)]
136
137 maxtable = [[0 for agent in range(Nagents)] for
138             level in range(Nlevels+1)]
139 minetatable = [[0 for agent in range(Nagents)] for
140               level in range(Nlevels+1)]
141 minPowerFactortable = [[0 for agent in range(Nagents)]
142                        for level in range(Nlevels+1)]
143
144 localbestobjective = [[0 for agent in range(Nagents)]
145                      for level in range(Nlevels+1)]

```

```

138 globalbestobjective = [[0 for a in range(2)] for level
    in range(Nlevels+1)]
139
140 localbestposition = [[0 for parameters in range(Npar)]
    for agent in range(Nagents)]
141 globalbestposition = [0 for parameters in range(Npar)]
142
143 #Set Initial Particle Positions, initial velocities set
    to zero
144 parcounter = 0
145 while parcounter < Npar:
146     agentcounter=0
147     while agentcounter < Nagents:
148         position [parcounter][0][agentcounter] = random.
            uniform(seedrange [parcounter][0], seedrange [
            parcounter][1])
149         agentcounter += 1
150     parcounter += 1
151
152 #####
153 ### Begin Optimization Loop ########
154 #####
155 level = 0
156 levelcounter = 0
157 while levelcounter <= Nlevels:
158     agent = 0
159     while agent < Nagents:
160         runcount += 1
161         R_i = position [0][level][agent]
162         R_o = position [0][level][agent]+position [1][
            level][agent]
163         thickness = position [1][level][agent]
164         wirediameter = 2 * position [2][level][agent]
165         f = position [3][level][agent]
166         kp = position [4][level][agent]
167         ki = position [5][level][agent]
168         kd = position [6][level][agent]
169         carrierfreq = position [7][level][agent]
170
171         #####
172         # Input Parameters #
173         #####

```

```

174
175     # Program Parameters
176     time = 300
177     delta_t = .001
178     theta_max = 400
179
180     # Environment and Operational Parameters
181     I_coil_max = 125
182     h_max = h_active = 695
183     h_min = h_natural = 76
184     theta_0 = theta_inf = 293
185
186     #Aerodynamic Loading Parameters
187     dchdphi = -.65 #[ rad^-1 ]
188     rho_infty = 1.225 #[ kg/m^3 ]
189     V_infty = 25 #[ m/s ]
190     c_e = .15 #[ m ]
191     S_e = .1875 #[ m^2 ]
192     k_spring = -dchdphi*.5*rho_infty*V_infty**2*S_e
193     *c_e #[ Nm/rad ]
194     phi0 = 0 #Reference angle
195     #Applied constant moment, N-m
196     M = 2.54
197     #Geometry
198     L = .203
199     R_coil = R_o + wirediameter / 2
200     R_0 = (R_o+R_i)/2
201     J_P = pi/32*((R_o*2)**4-(R_i*2)**4)
202     A_innersurf = R_i*2*pi*L
203     V = pi*(R_o**2-R_i**2)*L
204     #Coil Electrical Properties
205     rho_cu = 1.68e-8
206     mu_r_cu = .9999
207     N = L/wirediameter #Number of induction coil
208     turns
209     k_r = 1.15 #Coil geometry correction factor
210     #Nitinol Electrical Properties
211     rho_e_M = 82e-8
212     rho_e_A = 76e-8
213     mu_r = 1.02
214     #####
215     #Nitinol Thermomechanical Properties

```

```

214 #####
215 rho_m = 6450
216 c_h = 850
217 nu=.33
218 #Martensite Properties
219 G_M = 6.9e9
220 E_M = G_M*2*(1+nu)
221 c_M = 6.29e6 # Slope of martensite Clausius-
           Clapeyron curve at reference stress, Pa/K
222 #Austenite Properties
223 G_A = 12e9
224 E_A = G_A*2*(1+nu)
225 c_A = 7.61e6 # Slope of austenite Clausius-
           Clapeyron curve at reference stress, Pa/K
226 Ms = 48.6+273 # K
227 Mf = 36.4+273 # K
228 As = 53.1+273 # K
229 Af = 71.4+273 # K
230 H_max = .0769/2
231 H_min = .04/2
232 sigma_crit = 3.89e6 # Critical Mises Stress, Pa
233 kt = .0357 # MPa^-1
234 sigma_star = 200e6 #Reference Stress, Pa
235 # n1-n4: Hardening coefficients
236 n1 = .3
237 n2 = .3
238 n3 = .3
239 n4 = .3
240 #Computed model parameters
241 if sigma_star < sigma_crit:
242     H_curSigstar = H_min
243     dH_curSigstar = 0
244 else:
245     H_curSigstar = (H_min + (H_max-H_min)
           *(1-e**(-kt*((sigma_star-sigma_crit)
           /1000000))))
246     dH_curSigstar = (H_max-H_min)*kt
           /1000000*e**(-kt*((sigma_star-
           sigma_crit)/1000000))
247 ds = (1/rho_m) * (1/(c_M+c_A)) * -2 * (c_M*c_A)
           * (H_curSigstar + sigma_star*dH_curSigstar+
           sigma_star*(1/E_M-1/E_A))

```

```

248     du = ds/2*(Ms+Af)
249     a1 = rho_m*ds*(Mf-Ms)
250     a2 = rho_m*ds*(As-Af)
251     a3 = -1*a1/4 * ( 1 + 1/(n1+1) - 1/(n2+1) ) + a2
           /4 * ( 1+1/(n3+1) - 1/(n4+1) )
252     D = (c_M-c_A) * (H_curSigstar + sigma_star*
           dH_curSigstar+sigma_star*(1/E_M-1/E_A)) / ( (
           c_M+c_A)*(H_curSigstar+sigma_star*
           dH_curSigstar) )
253     Y0 = rho_m*ds/2 * (Ms-Af) - a3 + D*sigma_star*
           H_curSigstar
254     Pivec = [0]
255     # Initialize variables
256     t_elapsed = [0]
257     xi = [.999]
258     theta = [theta_0]
259     phi = [M*L/(J_P*G_M)]
260     tau = [M*R_o/(J_P*G_M)]
261     Zvec = [0]
262     etavec = [1]
263     PowerFactorvec = [1]
264     Powervec = [0]
265     stepvec = [0]
266     epsilon_tr = [0]
267     Lambda_vec = [0]
268     I_coil_vec = [0]
269     h_vec = [0]
270     errorvec = [0]
271     phic_vec = [0]
272     controller_vec = [0]
273     xi_f = .001
274     xi_r = .999
275     xi_f_vec = [xi_f]
276     xi_r_vec = [xi_r]
277     RMSerrorsum = 0
278     throwback = 0
279     PTerm = 0
280     ITerm = 0
281     DTerm = 0
282     controller = 0
283     taut = tau[-1]
284     phit = phi[-1]

```



```

285     thetat = theta[-1]
286     xit = xi[-1]
287     fdrv = 'r'
288     step = 1
289     phic = 0
290     controltimer = 0
291     carrierwave = 0
292     runtime = 0
293     #Begin incrementing the simulation
294     while t_elapsed[-1] < time-delta_t/2:
295         go = 1
296         #Compute current target
           rotation
297         phic, runtime = command(t_elapsed[-1],
           runtime)
298         phic_vec.append(phic)
299         #Compute PID controller output
300         error = phi[-1]-phic
301         PTerm = error
302         ITerm = ITerm + error*delta_t
303         DTerm = (error-errorvec[-1])/delta_t
304         controller = kp*PTerm + ki*ITerm + kd*
           DTerm
305         controller_vec.append(controller)
306         errorvec.append(error)
307
308         #Definition of cooling mode
309         if mode ==1: #Continuous Cooling
           Control, Continuous Current Control
310             if controller >= 0:
311                 I_coil = 0
312                 h = h_natural + (
           h_active-h_natural)*
           controller
313             if controller >= 1:
314                 h = h_active
315             if controller < 0:
316                 h = h_natural
317                 I_coil = abs(controller
           )*I_coil_max
318             if I_coil >= I_coil_max
           : I_coil = I_coil_max

```

```

319             if theta[-1] >=
                 theta_max: I_coil = 0
320 if mode == 2: #PWM Cooling Control,
                 Continuous Current Control
321             if controller >= 0:
322                 I_coil = 0
323                 if controller >=
                     carrierwave: h=h_max
324                 else: h=h_min
325             if controller < 0:
326                 h = h_natural
327                 I_coil = abs(controller
                     )*I_coil_max
328                 if I_coil >= I_coil_max
                     : I_coil = I_coil_max
329                 if theta[-1] >=
                     theta_max: I_coil = 0
330                 carrierwave += carrierfreq*
                     delta_t
331                 if carrierwave >= 1+delta_t*
                     carrierfreq/2:
332                     carrierwave = 0
333                 I_coil_vec.append(I_coil)
334                 h_vec.append(h)
335
336                 # Recompute shear modulus as a function
                     of martensite volume fraction
337                 G = ( 1/G_A + xit * ( 1 / G_M - 1 / G_A
                     ) )*(-1)
338                 # Recompute the electrical resistivity
                     of Nitinol
339                 rho_e = rho_e_A + xit * (rho_e_M-
                     rho_e_A)
340                 # Recompute Nitinol and copper skin
                     depth
341                 delta = sqrt(2*rho_e/(mu_r*mu_0*2*pi*f)
                     )
342                 delta_cu = sqrt(2*rho_cu/(mu_r_cu*mu_0
                     *2*pi*f))
343                 # Compute power and electrical shape
                     parameters from induction heater for
                     step

```

```

344 gamma = 2*R_o*(R_o-R_i)/(2*delta**2)
345 p = gamma/(1+(gamma)**2)
346 q = 1/(1+(gamma)**2)
347 EPower = 2*pi*f * I_coil**2 * mu_0*mu_r
          * N**2 * (pi*(R_o**2-R_i**2)/L)*p
348
349 # Determine sign(dtheta/dt) => Fwd/rev
          transformation
350 dthetalin=(1/(rho_m*c_h*(pi*(R_o**2-R_i
          **2)*L))) * (EPower + h*pi*2*R_i*L*(
          theta_inf-theta[-1]))
351 if dthetalin >= 0: fdrv = 'r'
352 else: fdrv = 'f'
353          #Compute Maximum Transformation
          Strain, H_cur
354 if sqrt(3)*taut < sigma_crit: H_cur =
          H_min
355 else: H_cur = (H_min + (H_max-H_min)
          *(1-e**(-kt*sqrt(3)*(taut/1000000))))
356          #Compute (Reduced)
          Transformation Tensor, 1-D
          Torsion
357 Lambda = (sqrt(3)/2.) * H_cur
358 Lambda_vec.append(Lambda)
359          #Compute Hardening Function
360          # MLM == 0: Minor loop
          modification inactive
361          # MLM == 1: Minor loop
          modification active
362 if MLM == 0:
363     if fdrv == 'f':
364         ft = .5*a1*(1+xit**n1
          -(1-xit)**(n2))+a3
365         dft = .5 * a1 * ( n1*
          xit**(n1-1) + n2*(1-
          xit)**(n2-1) )
366     if fdrv == 'r':
367         ft = .5*a2*(1+xit**n3
          -(1-xit)**(n4))-a3
368         dft = .5 * a2 * ( n3*
          xit**(n3-1) + n4*(1-
          xit)**(n4-1) )

```

```

369         if MLM == 1:
370             if fdrv == 'f':
371                 xi_hat = 1/(1-xi_f
                    -.0004)*xit-(xi_f
                    -.0004)/(1-xi_f
                    -.0004)
372                 ft = .5*a1*(1+xi_hat**
                    n1-(1-xi_hat)**(n2))+
                    a3
373                 dft = 1/(1-xi_f-.0004)
                    * .5 * a1 * ( n1*
                    xi_hat**(n1-1) + n2
                    *(1-xi_hat)**(n2-1) )
374             if fdrv == 'r':
375                 xi_hat = 1/(xi_r+.001)*
                    xit
376                 ft = .5*a2*(1+xi_hat**
                    n3-(1-xi_hat)**(n4))-
                    a3
377                 dft = 1/xi_r * .5 * a2
                    * ( n3*xi_hat**(n3-1)
                    + n4*(1-xi_hat)**(n4
                    -1) )
378             #Compute thermodynamic driving force ,
                Pi
379             Pi = 2*taut*Lambda + 4*(1+nu)*taut
                **2*(1/EM-1/EA)+rho_m*ds*thetat-
                rho_m*du-ft
380             #Determine if transformation is
                occurring for forward (f) /
                reverse (r) transformation
381             Y = Y0+D*sqrt(3)*taut*Lambda
382             if fdrv == 'r':
383                 step = 1
384                 if Pi <= -Y:
385                     if xi[-1] > 0.001 and
                        xi[-1] <= 0.999: step
                            = 2
386                     else: step = 3
387             if throwback == 1: step = 1
388             if throwback == 2: throwback =
                0

```

```

389     if fdrv == 'f':
390         #if step == 4 or 5 or 6:
391         step = 4
392         if Pi >= Y:
393             step = 3
394             if xi[-1] < 0.999 and
                 xi[-1] >= 0.001: step
                 = 5
395             else: step = 6
396         if throwback == 2: step = 4
397         if throwback == 1: throwback =
                 0
398     stepvec.append(step)
399     #Reverse Transformation
                 Direction
400     if fdrv == "r":
401         #Before Transformation
402         if step == 1 and go == 1:
403             dtheta=(1/(rho_m*c_h*(
                 pi*(R_o**2-R_i**2)*L)
                 ))*(EPower + h*pi*2*
                 R_i*L*(theta_inf-
                 theta[-1]))
404             thetat = theta[-1]+
                 dtheta*delta_t
405             phit = 1/(k_spring*R_0
                 /(G*J_P)-R_0/L)*(
                 k_spring*R_0/(G*J_P)*
                 phi0 - 2*epsilon_tr
                 [-1] - M*R_0/(G*J_P))
406             xit = xi[-1]
407             epsilon_tr_t =
                 epsilon_tr[-1]
408             go = 0
409             if throwback == 0:
410                 if Pi<=Y:
411                     step =
                         2
412             else:
413                 if thetat >=
                     theta_res:
414                     step =

```

```

415                                     2
                                     throwback
                                     = 0
416     #During Transformation
417     if step == 2 and go==1:
418         Pi = -Y
419         dtheta = (1/(rho_m*c_h
                    *(pi*(R_o**2-R_i**2)*
                    L)) * (EPower
420                        + h*pi*2*R_i*L
                        *(theta[-1]-
                        theta_inf)
421                        - V*(Y+rho_m*ds
                        *theta[-1]) *
                        rho_m*ds * (
                        dft)**-1 ) )
422         thetat = theta[-1] +
                    dtheta*delta_t
423         theta_dot = (thetat-
                    theta[-1]) / delta_t
424         tau_dot = (tau[-1]-tau
                    [-2]) / delta_t
425         dxidt = ( (dft)**-1 * (
                    rho_m*ds * theta_dot
                    + (2*(1-D)*Lambda +
                    4*tau[-1]*(1/GM-1/
                    GA)) * tau_dot ) )
426         go = 0
427         if dxidt < 0:
428             epsilon_tr_t =
                    epsilon_tr
                    [-1] + Lambda
                    *abs(dxidt)*
                    delta_t
429             xit = xi[-1] +
                    delta_t *
                    dxidt
430             phit = 1/(
                    k_spring*R_0
                    /(G*J_P)-R_0/
                    L)*(k_spring*
                    R_0/(G*J_P)*

```

```

                                phi0-2*
                                epsilon_tr_t-
                                M*R_0/(G*J_P)
                                )
431 Moment=k_spring
                                *(phit-phi0)
                                + M
432 taut = Moment*
                                R_0/J_P
433 Lambda_vec.
                                append(Lambda
                                )
434 if xit <=
                                0.001:
435     step = 3
436     xit = 0.001
437     xi_f = xit
438 #After Transformation
439 if step == 3 and go == 1:
440     dtheta=(1/(rho_m*c_h*(
                                pi*(R_o**2-R_i**2)*L)
                                ))*(EPower + h*pi*2*
                                R_i*L*(theta_inf-
                                theta[-1]))
441 thetat = theta[-1]+
                                dtheta*delta_t
442 phit = phi[-1]
443 xit = xi[-1]
444 epsilon_tr_t =
                                epsilon_tr[-1]
445 Lambda_vec.append(0)
446 go = 0
447 #Forward Transformation
                                Direction
448 if fdrv == "f":
449     #Before Transformation
450     if step == 4 and go == 1:
451         dtheta=(1/(rho_m*c_h*(
                                pi*(R_o**2-R_i**2)*L)
                                ))*(EPower
452         + h*pi*2*R_i*L
                                *(theta_inf-

```

```

453         theta[-1]))
thetat = theta[-1]+
         dtheta*delta_t
454 phit = phi[-1]
455 xit = xi[-1]
456 epsilon_tr_t =
         epsilon_tr[-1]
457 go = 0
458 if throwback == 0:
459     if Pi>=Y:
460         step =
            5
461 #During Transformation
462 if step == 5 and xi[-1] < .999
    and go == 1:
463     Phit = Y
464     dtheta = (1/(rho_m*c_h
        *(pi*(R_o**2-R_i**2)*
        L))) * (EPower + h*pi
        *2*R_i*L*(theta[-1]-
        theta_inf)
465         + V*(-Y+rho_m*
            ds*theta[-1])
            * rho_m*ds *
            (dft**(-1) )
466     thetat = theta[-1] +
        dtheta*delta_t
467     theta_dot = -1*(thetat-
        theta[-1]) / delta_t
468     tau_dot = (tau[-1]-tau
        [-2]) / delta_t
469     dxidt = ( (dft)**(-1) * (
        rho_m*ds * theta_dot
        +
470         (2*(1-D)*Lambda
            + 4*tau
            [-1]*(1/GM
            -1/GA)) *
            tau_dot ) )
471 if dxidt > 0:
472     epsilon_tr_t =
        epsilon_tr

```



```

[-1] - Lambda
*abs(dxidt)*
delta_t
473 xit = xi[-1] +
      delta_t *
      dxidt
474 phit = 1/(
      k_spring*R_0
      /(G*J_P)-R_0/
      L)*(k_spring*
      R_0/(G*J_P)*
      phi0-2*
      epsilon_tr_t -
      M*R_0/(G*J_P)
      )
475 Moment=k_spring
      *(phit-phi0)
      + M
476 taut = Moment*R_0/J_P
477 if (taut-tau[-1])/(
      thetat-theta[-1]) >
      c_M:
478     step = 4
479     theta_res = (
      taut-tau[-1])
      *(1/c_M)+
      theta[-1]
      throwback = 2
480 if xit <= 0:
481     xit = 0.001
482 if xit >=.999:
483     xit = .999
484     step = 6
485 xi_r = xit
486 go=0
487 #After Transformation
488 if step == 6 and go == 1:
489     dtheta=(1/(rho_m*c_h*(
490         pi*(R_o**2-R_i**2)*L)
         ))*(EPower + h*pi*2*
         R_i*L*(theta_inf-
         theta[-1]))

```

```

491         thetat = theta[-1]+
              dtheta*delta_t
492         phit = phi[-1]
493         xit = xi[-1]
494         epsilon_tr_t =
              epsilon_tr[-1]
495         Lambda_vec.append(0)
496         go=0
497
498         # Compute other electrical properties
499         K = 2*pi*f*mu_0*(N**2/L)
500         R_w = K*mu_r*p*pi*R_o**2
501         R_c = K*(k_r*pi*R_coil*delta_cu)
502         X_g = K*pi*(R_coil**2-R_o**2)
503         X_w = K*mu_r*(q)*pi*R_o**2
504         X_c =K*k_r*pi*R_coil*delta_cu
505         Z = (R_w+R_c)+(X_g+X_w+X_c)*1j
506         eta = R_w/(R_c+R_w)
507         PowerFactor = (R_w+R_c)/abs(Z)
508         Powervec.append(EPower)
509         xi.append(xit)
510         theta.append(thetat)
511         phi.append(phit)
512         tau.append(taut)
513         t_elapsed.append(t_elapsed[-1]+delta_t)
514         Zvec.append(Z)
515         etavec.append(eta)
516         PowerFactorvec.append(PowerFactor)
517         Pivec.append(Pi)
518         epsilon_tr.append(epsilon_tr_t)
519         xi_f_vec.append(xi_f)
520         xi_r_vec.append(xi_r)
521         #RMS Error
522         RMSerrorsum += (phic_vec[-1]-phi[-1])**2
523         RMSerror = sqrt(RMSerrorsum/(t_elapsed[-1]/
              delta_t))
524         print '_'
525         print 'RMS_Error:_', RMSerror
526         print 'Maximum_Shear_Stress:_', max(tau)
527         print 'Minimum_Electrical_Efficiency:', min(
              etavec)
528         print 'Minimum_Power_Factor:', min(

```

```

    PowerFactorvec)
529     print 'Average_Impedance_Magnitude:_', mean(
        absolute(Zvec))
530     if runtime == 0: runtime = t_elapsed[-1]
531     #Convert radians to degrees
532     phil = []
533     for i in phi:
534         phil.append(i*360/(2*pi))
535         #Reorient results to match experimental
            axis
536     phiflip = []
537     phicflip = []
538     for rot in phi: phiflip.append(-1*rot)
539     for rotc in phic_vec: phicflip.append(-1*rotc)
540
541     maxtau = max( max(tau),abs(min(tau)) )
542     mineta = min(etavec)
543     minPowerFactor = min(PowerFactorvec)
544
545     #####
546     # Plot Results #
547     #####
548     actline , = plt.plot(t_elapsed , phi)
549     cmdline , = plt.plot(t_elapsed , phic_vec)
550     plt.legend([actline ,cmdline],[ 'Computed_
        Rotation' , 'Command_Rotation' ], loc=4)
551     plt.title('Rotation_Angle_vs._Time')
552     plt.xlabel('time')
553     plt.ylabel('phi_[rad]')
554     plt.grid(b=True, which='both', color='0.65',
        linestyle='-')
555     plt.xlim([0, time])
556     plt.ylim([ min([min(phi),min(phic_vec)])-1 ,
        max([max(phi),max(phic_vec)])+1 ])
557     plt.savefig('time-phi.png')
558     plt.show()
559
560     plt.plot(t_elapsed , theta)
561     plt.title('Temperature_[K]_vs._Time')
562     plt.xlabel('time')
563     plt.ylabel('theta')
564     plt.grid(b=True, which='both', color='0.65',

```

```

        linestyle='-')
565 plt.savefig('time-theta.png')
566 plt.show()
567
568 plt.plot(theta, phi)
569 plt.title('Twist_Angle_vs._Temperature')
570 plt.xlabel('Temperature')
571 plt.ylabel('Rotation_Angle_[rad]')
572 plt.savefig('temp-angle.png')
573 plt.show()
574
575 plt.plot(t_elapsed, Pivec)
576 plt.xlabel('Time_Elapsed')
577 plt.ylabel('Pi')
578 plt.savefig('Pivec.png')
579 plt.show()
580 ##### Print Data to Output File
581
582 print 'Writing_to_File'
583 print 'Run:', runcount
584 print 'Level:', level
585 print 'Agent:', agent
586 print kp, ki, kd
587 fout = open('PSOData.txt', 'a')
588 fout.write(str(runcount) + ', ' + str(R_i) + ',
        _' + str(thickness) + str(wirediameter) + ', _'
        + str(f) + ', _' + str(kp) + ', _' + str(ki)
        + ', _' + str(kd) + ', _' + str(carrierfreq) +
        ', _' + str(RMSError) + ', _' + str(abs(min(phi
        ))) + ', _' + str(min(PowerFactorvec)) + ', _'
        + str(min(etavec)) + '\n')
589 fout.close()
590
591 objective[level][agent] = RMSError
592 maxtautable[level][agent] = maxtau
593 minetatable[level][agent] = mineta
594 minPowerFactortable[level][agent] =
        minPowerFactor
595
596 if maxtau <= maximumstress and mineta >=
        minimelectrical efficiency and
        minPowerFactor >= minimumpowerfactor:

```

```

597         constraintsmet = True
598     else:
599         constraintsmet = False
600
601     if level == 0:
602         localbestobjective[level][agent] = RMSerror
603         for counter in range(0,Npar):
604             localbestposition[agent][counter] =
                position[counter][level][agent]
605     else:
606         if RMSerror < localbestobjective[level-1][
            agent] and constraintsmet == True:
607             localbestobjective[level][agent] =
                RMSerror
608             for counter in range(0,Npar):
609                 localbestposition[agent][counter] =
                    position[counter][level][agent]
610         else:
611             localbestobjective[level][agent] =
                localbestobjective[level-1][agent]
612             for counter in range(0,Npar):
613                 localbestposition[agent][counter] =
                    position[counter][level-1][agent]
                ]
614
615     if agent == 0:
616         levelglobalbest = [agent,RMSerror]
617     else:
618         if RMSerror < levelglobalbest[1] and
            constraintsmet == True:
619             levelglobalbest = [agent,RMSerror]
620
621     agent += 1
622
623     #Update Global Best Position
624     if level == 0:
625         globalbestobjective[level] = levelglobalbest
626         for counter in range(0,Npar):
627             globalbestposition[counter] = position[
                counter][level][levelglobalbest[0]]
628     else:
629         if levelglobalbest[1] < globalbestobjective[

```

```

        level - 1][1]:
630         globalbestobjective[level] =
            levelglobalbest
631         for counter in range(0,Npar):
632             globalbestposition[counter] = position[
                counter][level][levelglobalbest[0]]
633     else:
634         globalbestobjective[level] =
            globalbestobjective[level-1]
635     #Print Global Best Solution and Data to File
636     fout = open('PSOBest.txt','a')
637     fout.write( str(level) + ',_' + str(
        globalbestposition) + ',_' + str(
        globalbestobjective[level]) + '\n')
638     fout.close()
639
640     level += 1
641     levelcounter += 1
642     if level in range(0, Nlevels+1):
643         #Update agent velocities and positions
644         agent = 0
645         while agent < Nagents:
646             parameter = 0
647             while parameter < Npar:
648                 velocity[parameter][level][agent] = (
649                     velocity[parameter][level-1][agent
650                     ]*w
651                     - random.random()*c1*(position[
        parameter][level-1][agent] -
        localbestposition[agent][
        parameter])
652                     - random.random()*c2*(position[
        parameter][level-1][agent] -
        globalbestposition[parameter])
653                     )
654                 #Velocity Limiter
655                 if abs(velocity[parameter][level][agent
        ]) > (seedrange[parameter][1] -
        seedrange[parameter][0]):
656                     velocity[parameter][level][agent] =
                        sign(velocity[parameter][level][
        agent])*(seedrange[parameter][1] -

```

```

        seedrange [parameter] [0])
656 #Set new position
657 position [parameter] [level] [agent] =
        position [parameter] [level - 1] [agent] +
        velocity [parameter] [level] [agent]
658
659 if parameter == 0:
660     if position [parameter] [level] [agent]
661         < LBoundR_i:
662         position [parameter] [level] [
663             agent] = LBoundR_i
664         #velocity [parameter] [level] [
665             agent] = 0
666 if parameter == 1:
667     if position [parameter] [level] [agent]
668         < LBoundt:
669         position [parameter] [level] [
670             agent] = LBoundt
671         #velocity [parameter] [level] [
672             agent] = 0
673 if parameter == 2:
674     if position [parameter] [level] [agent]
675         < LBoundWire:
676         position [parameter] [level] [
677             agent] = LBoundWire
678         #velocity [parameter] [level] [
679             agent] = 0
680 if parameter == 3:
681     if position [parameter] [level] [agent]
682         < LBoundf:
683         position [parameter] [level] [
684             agent] = LBoundf
685         #velocity [parameter] [level] [
686             agent] = 0
687 if parameter == 4:
688     if position [parameter] [level] [agent]
689         < LBoundk_P:
690         position [parameter] [level] [
691             agent] = LBoundk_P
692         #velocity [parameter] [level] [
693             agent] = 0
694 if parameter == 5:

```

```

680         if position [parameter] [level] [agent
681             ] < LBoundk_I:
682             position [parameter] [level] [
683                 agent] = LBoundk_I
684             #velocity [parameter] [level] [
685                 agent] = 0
686         if parameter == 6:
687         if position [parameter] [level] [agent
688             ] < LBoundk_D:
689             position [parameter] [level] [
690                 agent] = LBoundk_D
691             #velocity [parameter] [level] [
692                 agent] = 0
693         if parameter == 7:
694         if position [parameter] [level] [agent
695             ] < LBoundPWM:
696             position [parameter] [level] [
697                 agent] = LBoundPWM
698             #velocity [parameter] [level] [
699                 agent] = 0
700         parameter += 1
701         agent += 1
702
703 #####
704 #### Optimization Plotting Segment ####
705 #####
706 plt.plot(position [0])      #0. R_i
707 plt.title('Inside_Radius_Convergence')
708 plt.xlabel('Optimization_Level')
709 plt.ylabel('Inside_Radius,_R_i')
710 plt.savefig('Convergence-R_i.png')
711 plt.show()
712
713 plt.plot(position [1])      #1. t
714 plt.title('Thickness_Convergence')
715 plt.xlabel('Optimization_Level')
716 plt.ylabel('Thickness,_t')
717 plt.savefig('Convergnce-t.png')
718 plt.show()
719
720 plt.plot(position [2])      #2. wire radius

```



```

713 plt.title('Wire_Radius_Convergence')
714 plt.xlabel('Optimization_Level')
715 plt.ylabel('Wire_Radius')
716 plt.savefig('Convergnce-R_wire.png')
717 plt.show()
718
719 plt.plot(position[3])    #3. frequency
720 plt.title('Frequency_Convergence')
721 plt.xlabel('Optimization_Level')
722 plt.ylabel('Electrical_Frequency , f')
723 plt.savefig('Convergnce-f.png')
724 plt.show()
725
726 plt.plot(position[4])    #4. k_p
727 plt.title('PID_Proportional_Constant_Convergence')
728 plt.xlabel('Optimization_Level')
729 plt.ylabel('Proportional_Tuning_Constant , k_P')
730 plt.savefig('Convergnce-k_P.png')
731 plt.show()
732
733 plt.plot(position[5])    #5. k_i
734 plt.title('PID_Integral_Constant_Convergence')
735 plt.xlabel('Optimization_Level')
736 plt.ylabel('Integral_Tuning_Constant , k_I')
737 plt.savefig('Convergnce-k_I.png')
738 plt.show()
739
740 plt.plot(position[6])    #6. k_d
741 plt.title('PID_Derivative_Constant_Convergence')
742 plt.xlabel('Optimization_Level')
743 plt.ylabel('Derivative_Tuning_Constant , k_D')
744 plt.savefig('Convergnce-k_D.png')
745 plt.show()
746
747 plt.plot(position[7])    #7. PWM Frequency
748 plt.title('PWM_Carrier_Frequency_Convergence')
749 plt.xlabel('Optimization_Level')
750 plt.ylabel('PWM_Carrier_Frequency')
751 plt.savefig('Convergnce-PWMfreq.png')
752 plt.show()
753
754 plt.semilogy(localbestobjective)    #Overall

```

Convergence Behavior

```
755 plt.title('Best_Objective_Function_for_Each_Agent')
756 plt.xlabel('Optimization_Level')
757 plt.ylabel('Objective_Function')
758 plt.savefig('LocalBestObjective.png')
759 plt.show()
```