THESIS

CONTROL SYSTEM DESIGN FOR PLASMA POWER GENERATOR

Submitted by

Aishwarya Sankaran

Department of Electrical and Computer Engineering

In partial fulfillment of the requirements

For the Degree of Master of Science

Colorado State University

Fort Collins, Colorado

Summer 2022

Master's Committee:

Advisor: Peter M Young

Edwin Chong
Charles Anderson

ABSTRACT

CONTROL SYSTEM DESIGN FOR PLASMA POWER GENERATOR

The purpose of this research is to develop advanced control strategies for precise control over power delivery to nonlinear plasma loads at high frequency. A high-fidelity MATLAB/Simulink simulation model was provided by Advanced Energy Industries, Inc (AE) and the data from this model was considered as the actual model under consideration. The research work requires computing a mathematical model of the plasma power generator system, analyzing and synthesizing robust controllers for individual operating points, and then developing a control system that covers the entire the grid of operating points. The modeling process involves developing computationally simple near-linear models representing relevant frequencies and operating points for the system consisting of nonlinear plasma load, RF Power Amplifier, and a Match Network. To characterize the (steady-state) mapping from power setpoint to delivered power the steady-state gains of the system are taken under consideration. Linear and nonlinear system identification procedures are used to adequately capture both the nonlinear steady-state gains and the linear dynamic model response. These near-linear or linear models with uncertainty description to characterize the robustness requirements are utilized in the second stage to develop a grid of robust controller designed at linear operating points. The controller from μ-synthesis design process optimizes robust performance for allowable perturbations as large as possible. It does all this while guaranteeing closed-loop stability for all allowable perturbations. The final stage of the research focuses on developing Linear Parameter Varying (LPV) controllers with non-linear offset. This single controller covers the entire operating range, including the case that the desired signals to track may vary over wide regions

of the operating envelope. LPV controllers allows actual power to track the changing setpoint in a smooth manner over the entire operating range.

ACKNOWLEDGEMENTS

This thesis has become a reality only with the support and unwavering encouragement from many individuals. I would like to extend my sincere thanks to all of them.

First and foremost, I would like to convey my special gratitude to my Advisor Dr Peter M. Young for his guidance in this research and imparting his knowledge through amazing lectures in *Control Theory, Robust Control Systems, and Nonlinear Control Systems.* His humbleness and expertise in the subject motivated me in pursuing academic and career goals in the field of Control Systems.

I am indebted to Advanced Energy Industries and Dr Chad S. Samuels for identifying the potential in me and providing me with a great opportunity to work on their project. Dr Chad's prompt response and timely feedback was key for the successful completion of this endeavor.

I would like to convey my special thanks to my committee members, Dr Edwin Chong, and Dr Charles Anderson for their valuable suggestions that helped me deepen my knowledge of the subject.

Finally, I would like to thank my parents, relatives and friends for their constant love and support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

AE Advanced Energy Industries
LPV Linear Parameter Varying
MIMO Multi-Input Multi-Output
NN Neural Networks
LTI Linear Time Invariant
PI Proportional-Integral
CSU Colorado State University
RF Radio Frequency
MLP Multi-Level Pulsing
M&C Measurement and Control
MPSoC Multi-Processor System on Chip
CW Continuous wave
DLP Digital Light Processing
HDL Hardware Description Language
FPGA Field Programmable Gate Array
FIL FPGA-in-the-loop
HIL Hardware-in-the-loop
SIL Software-in-the-loop
Vphase Phase Voltage
Pdel Power Delivered
G(z) Linear Dynamic Model
Δ Uncertainty
FFT Fast Fourier Transform
GUI Graphical User Interface
DC Direct Current
PID Proportional-Integral-Derivative
SoC System on Chip
VHDL Very High-Speed Integrated Circuit (VHSIC) Hardware Description Language

CHAPTER 1

INTRODUCTION

This research project is in collaboration with Advanced Energy Industries based in Fort Collins and it involves developing advanced control strategies for radio frequency power supply. AE is a leading supplier of RF plasma generators which delivers power to nonlinear plasma loads at high frequency.

Plasma processing systems are commonly utilized to change the surface qualities of materials in a range of sectors. For example, many processes in the production of modern integrated circuits use plasmas for etching sub-micrometer features or depositing atomically thin layers of materials.

A plasma processing system typically consists of a processing chamber and a power supply system for creating and maintaining plasma inside the chamber. The plasma is an electrical load with a characteristic impedance that the power generator drives. The impedance of a processing plasma, on the other hand, is not always constant and might change based on process circumstances and other variables. Variations in plasma impedance may have an unfavorable effect on the generator's power delivery, which is normally optimized solely for a specific load impedance. Due to changes in the physical properties of the plasma at different power levels, these variances may cause unwanted drifts or perturbations in process variables such as etch or deposition rates. As a result, plasma processing systems frequently include impedance matching and control mechanisms or circuitry that react to variations in plasma impedance and maintain desirable levels of plasma power supply.

In order for the generators to distribute power efficiently, these power supplies often use a network to match the plasma load impedance. To cope with changes in impedance and at least keep the average impedance matched, the generator is modulated at a higher rate than the network used to match the impedance. The plasma load in the plasma chamber can vary depending on the gas combination, gas pressure, and changes in the plasma device. The plasma generator has quick dynamics, enabling for rapid and high-level power setpoint changes. It's worth noting that the load impedance matched at the generator is a function of the load's power. The output power, on the other hand, is a function of the matched impedance. The system's unique feedback loop makes control design an intriguing and demanding task. We will be looking into three major areas for the development of control systems as described in the sections below.

1.1 SYSTEM LEVEL MODELLING

Beyond the conventional goal of generating a relatively realistic model of the physical system under examination, the modeling efforts discussed here have certain unique goals. The models provided here are primarily intended to aid in the design, analysis, and optimization of control systems. As a result, we must possess the following characteristics:

• System level models, which include all physical components as well as any current control scheme features that may be kept in the design.

• Capture both linear and nonlinear effects adequately.

• Relatively high overall accuracy.

• Structures that are simple, such as nonlinear-static and linear-dynamic.

• Low order, for example, is computationally simple.

• Cover the whole operating envelope, which may need the use of many interconnected models for various operating points.

• Define the nominal performance as well as the uncertainty ranges.

The fundamental purpose of system level modeling is to develop a process for efficiently obtaining models, as mentioned above. In later stages of the project, these models will be used to design controller

1.2 CONTROLLER DESIGN

To establish the robustness requirements, the robust controller design approaches developed here rely on a (gain-scheduled) set of linear models at a grid of operating points, with uncertainty descriptions. The goal becomes that of finding a controller **K**(z) minimizing the peak value of μ across frequency for a single setpoint. In order to apply the μ synthesis design technique to our problem, we first need to decide on the design system interconnection. This contains the physical model of our system, but it includes much more than that. In addition, the Design Interconnection will specify:

1. What uncertainties are present in the system?

2. Which disturbance signals act on the system?

3. Which signals do we wish to penalize in the optimal control problem?

4. What are the control signals, and measurements, available to us?

1.3 ROBUST CONTROLLER DESIGN

Having developed a grid of robust controllers across the operating envelope utilizing μ synthesis, with the controller operating points spaced apart by about 500W, we also need the controller to handle arbitrary waveforms transitioning across multiple operating points. Hence, we develop Linear Parameter Varying (LPV) controllers to handle the entire operating range. To achieve this, we implement an interpolation algorithm such that at each sample time we measure power and compute its nearest two neighbors in terms of controller operating points.

1.4 THESIS STRUCTURE

The structure followed in this thesis is as follows:

- Chapter 1: Introduction to the research topic and brief description of the research
- Chapter 2: Literature review
- Chapter 3: Introduction to the problem statement through AEs current system overview.
- Chapter 4: Discussion of the theoretical approaches to system modeling, controller design around a setpoint, and LPV control system design
- Chapter 5: Discussions include the process of applying the theoretical methods in the AE system and the results
- Chapter 6: Conclusion and future work

CHAPTER 2

LITERATURE REVIEW

A technique and apparatus for altering interactions between a nonlinear load and an electrical generator is presented in an AE patent published [1] in 2009. The engine of the generator receives a control input as the main control signal. The main control signal governs at least one of the electrical generator's output power, output current, and output voltage given to the nonlinear load, the engine being a power amplifier or a converter. Sensors are used to measure the impedance of the nonlinear load. Finally, a compensation signal is fed to the generator that renders the transfer function of the output power of the generator with respect to the main control signal substantially insensitive to variations in the impedance of the nonlinear load.

A second AE patent published in 2017 [2] entails a method for adjusting the source impedance of a generator as a control mechanism that responds to changes in the plasma impedance and maintain desired levels of power delivery to the plasma. As an example, a method includes generating a first signal and applying it to the first input of a combiner. A second signal is generated and applied to the second input of the combiner. These two signals are then combined to produce power that can be delivered to the plasma load. A controller variable impedance is provided to the isolation port of the combiner. To modify the source impedance of the power supply system the controllable variable impedance can be adjusted.

The textbook chapter by Skogestad [3] discusses how to represent uncertainty and analyze its effects on structured singular value μ. To analyze the robust stability of our uncertain system we made use of the $M\Delta$-structure, where M represents the transfer function of the new feedback system generated by the uncertainity. The D-K iteration method combining H_∞-synthesis and μ-

analysis synthesizes an optimal μ-controller by minimizing the peak value over frequency of the upper bounds for a scaled problem.

CHAPTER 3

AE CURRENT SYSTEM OVERVIEW

The overall scheme portrayed in figure 3.1 below depicts the control architecture currently in use by AE.

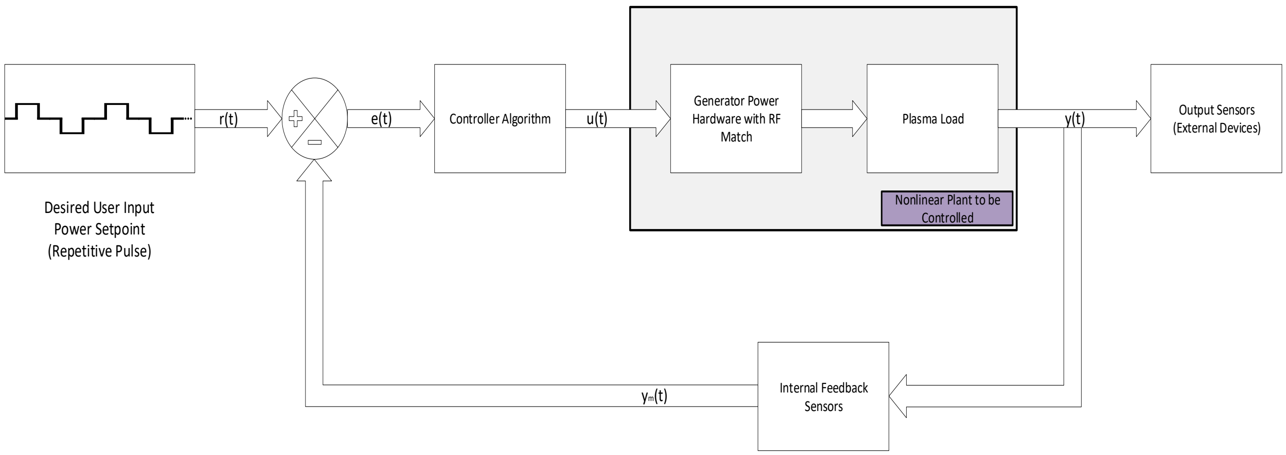

Figure 3.1: Advanced Energy Current Control Architecture

This appears to be a standard feedback control strategy. The feedback control algorithm employs PI control, or simply integral control, in conjunction with an integrator reset mechanism that remembers and resets the integrator for each system state/power level encountered. Industry demands have shifted to multi-level pulsing, which has made the integrator's reset scheme substantially more difficult. As a result, AE is looking into improved control systems that can meet the more demanding performance standards, especially without the use of any kind of integrator resetting mechanism.

However, while industry rivalry has previously looked into the use of feedforward and adaptability for similar systems, with some success, robustness and performance requirements are always increasing.

The control system design will be tackled chapters 4 and 5 of the thesis, so we'll save a comprehensive discussion of these components, as well as the research issues/approaches that go with them, until then. For the time being, we are focusing on establishing system models that will help the subsequent design work in section 4.2, which will be detailed in the sections that follow in this thesis report. Before we discuss system modeling approaches, let us mention and describe the components in the Digital Twin model of the plasma power system provided by AE.

3.1 DIGITAL TWIN

The Digital Twin essentially involves simulating the real-world AE generator systems, matches, and interactions with plasma chambers in the context of power delivery to a high degree of accuracy.

Main Functional Blocks The model consists of four major blocks representing:

1. The RF Generator
2. The Match Hardware
3. The Plasma Reactor
4. The Sensing Hardware

Figure 3.2: Digital twin main functional blocks

The RF Generator block represents the hardware and functionality of the generator. Since the RF Generator block is the only block of concern for this research, we will briefly describe the major constituents under this system.

1. The user Multi-Level Pulsing (MLP) setpoint input and creation.

2. The measurement and control (M&C) Multi-Processor System on Chip (MPSoC).

3. The RF Power Amplifier.

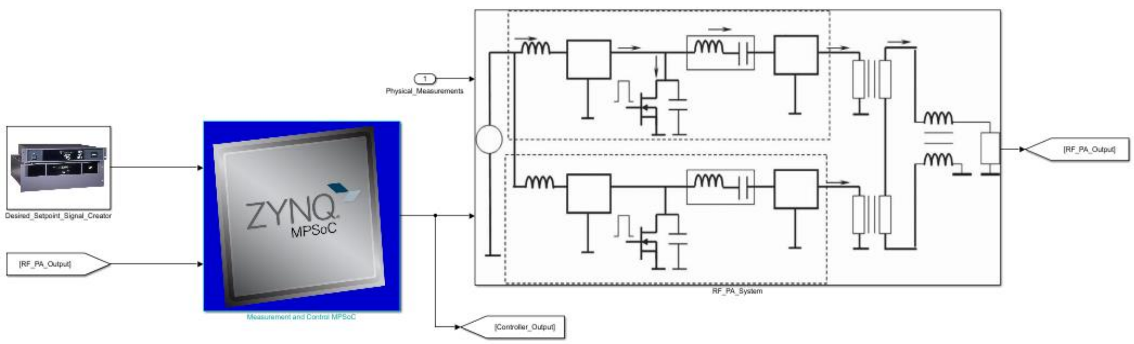

Figure 3.3: The RF Generator Hardware Components and Functionality

The user MLP setpoint input and creation This block ensures the desired CW/DLP the user desires is transformed into a meaningful signal that is sent to the onboard M&C MPSoC. This block uses solely Simulink components that can be automatically converted into embedded C and/or HDL for Rapid Prototyping.

The M&C MPSoC This block embodies the centralized control algorithms and measurements processing. The processed measurements are passed through the measurement hardware, the output of which is passed to the controller which outputs the electric control signals which are then fed to the whole voltage, power delivery, match, and frequency control interconnected hardware system. In Fig. 3.4 the M&C MPSoC consists of two hardware and functional elements: 1. The measurement unit. 2. The controller system. In fact, the RF Measurement unit as well as the controller system have been designed solely with blocks that can automatically be converted into embedded C/HDL code that can go directly onto a generator. Furthermore, the accuracy by which the modeling of these hardware elements and their functionality have been modeled can allow the Digital Twin end-user to run HIL simulations through MATLAB/Simulink's FIL/HIL/SIL connection capabilities.

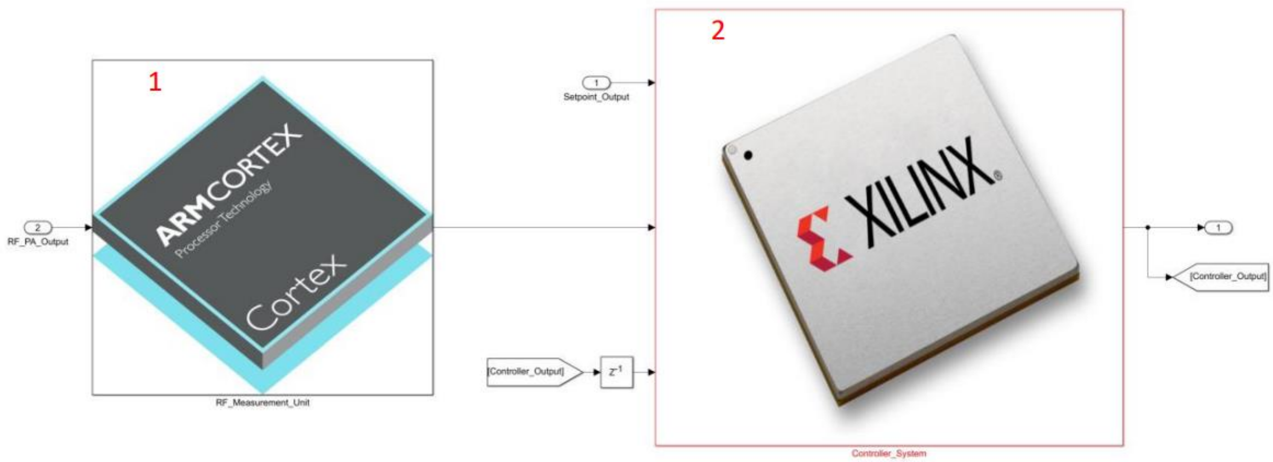

Figure 3.4: The M&C MPSoC's Hardware Components and Functionality

RF Power Amplifier The RF Power Amplifier unit consists of the following three functional hardware blocks: 1. The Sensor Systems. 2. The Digital Down Conversion Systems. 3. The Calibration Systems. The RF PA Outputs are selected and passed to the sensor systems. The sensors then convert these quantities into their digital (discretized and quantized) equivalents which are then passed to the calibration systems that determine based on the generator/match/reactor system's characteristic impedance the actual values of the forward power, reflected power, and load power. These values are then passed to output interface which collects them all together and outputs them from the RF Measurement unit as a single bus signal.

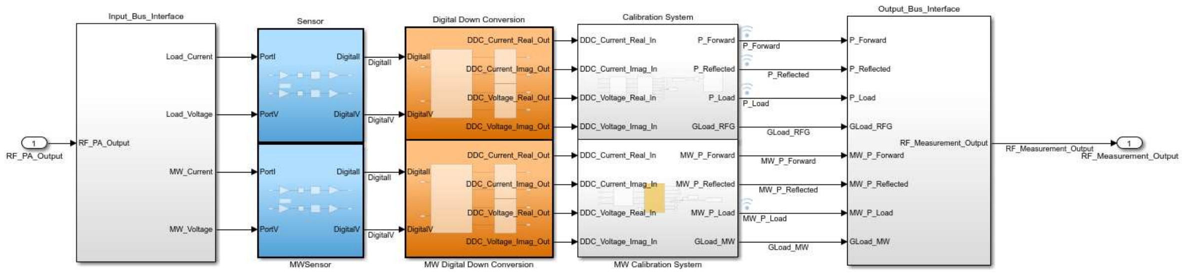

Figure 3.5: The RF Measurement Hardware Components and Functionality

CHAPTER 4

THEORETICAL APPROACH

The plasma power generator system can be represented by mathematical equations which are useful in the analysis and design of control systems. Let us look into the theoretical approaches into system modeling in the subsections that follow.

4.1 SYSTEM LEVEL MODELING

Beyond the conventional goal of generating a relatively realistic model of the physical system under examination, the modeling efforts discussed here have certain unique goals. The models provided here are primarily intended to aid in the design, analysis, and optimization of control systems. As a result, we must possess the following characteristics:

- System level models, which include all physical components as well as any current control scheme features that may be kept in the design.
- Obtain both linear and non - linear effects adequately.
- Relatively high overall accuracy.
- Structures that are simple, such as nonlinear-static and linear-dynamic.
- Low order, for example, is computationally simple.
- Cover the whole operating envelope, which may need the use of many interconnected models for various operating points.
- Define the nominal performance as well as the uncertainty ranges.

4.1.1 MODEL DEVELOPMENT APPROACH

In order to deliver the attributes discussed in the preceding subsection we propose a general model development structure of the form illustrated in figure 4.1. For the generator functioning, the major scheduling parameters are Impedance and Frequency. The values are entered into a grid of available Nonlinear Static Models in the Nonlinear Steady State Model Lookup. For specified values of Impedance and Frequency, this model component is essentially a nonlinear gain (no dynamics) that captures the fluctuation of the overall system gain with power level.

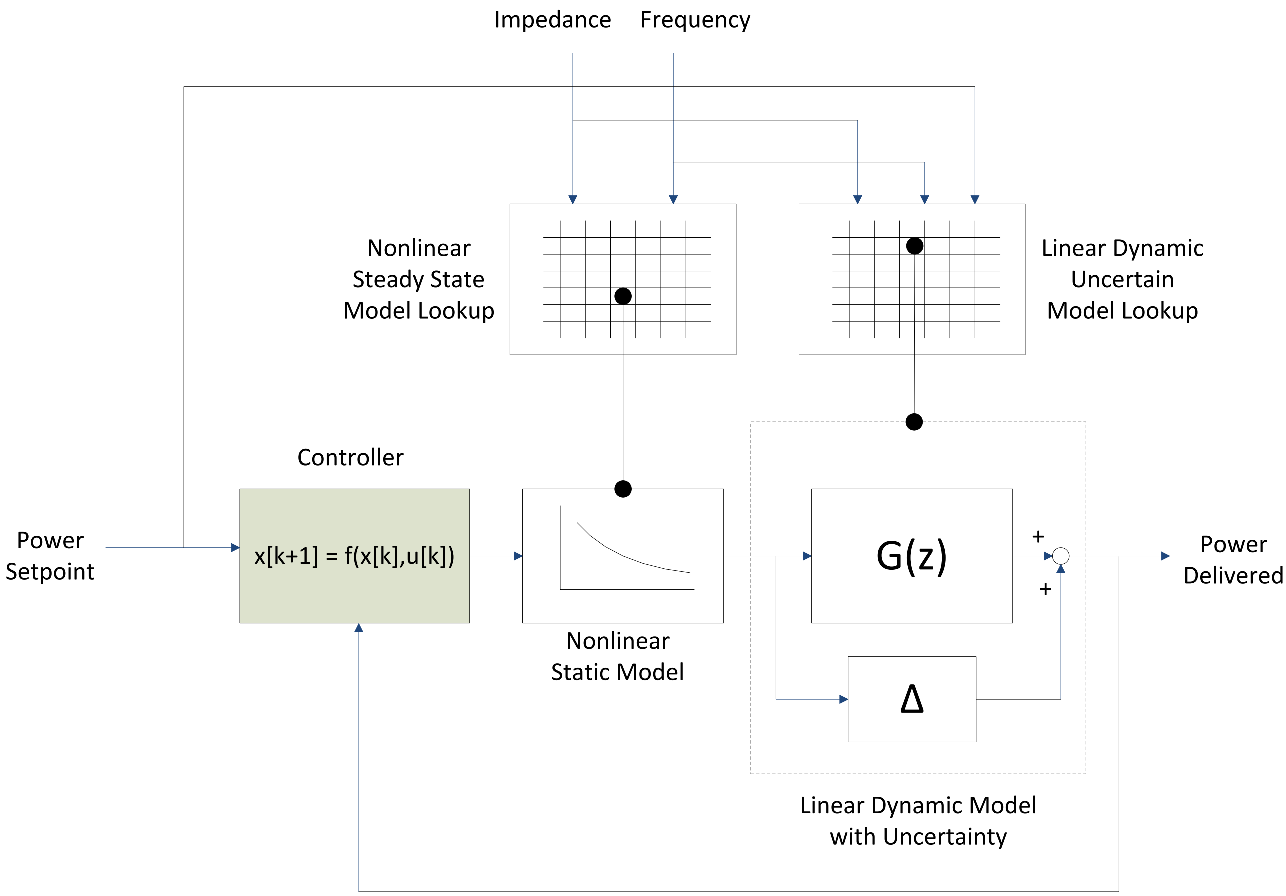

Figure 4.1: Model Development Architecture

The Linear Dynamic Uncertain Model Lookup block receives the fundamental scheduling variables, Impedance and Frequency, as well as the precise Power Setpoint. An array of Linear Dynamic Model with Uncertainty elements is contained in this block. The Linear Dynamic Uncertain Model Lookup picks a specific Linear Dynamic Model with Uncertainty, which becomes the system model's next component. It depicts the system's linear dynamics at this exact operational point (Impedance, Frequency, and the Power Setpoint). The model is often a discrete-time LTI system (Pulse Transfer Function – G(z), or State Space representation) that represents the tiny signal dynamics in the vicinity of the operational point.

4.1.2 UNCERTAINTY CHARACTERIZATION

Uncertainty characterization is also included in linear dynamics. This captures the dynamics' unpredictability, and it will be a key component of the robust controller design processes, allowing for a-priori consideration of specific stability and performance guarantees. A Controller could be part of the overall system, or it could be part of the existing control strategy that we choose to integrate. The final result of the whole endeavor is a model from *Power Setpoint* to *Power Delivered* for the overall system. The model is specifically designed for controller design, with an emphasis on optimization, robustness, and adaptation/learning.

To completely characterize the model, we must examine the variability of the data in relation to our system model, which is accomplished by repeatedly applying the same inputs and describing the output variability as shown in figure 4.2. In this section, we will describe the process for uncertainty characterization, which can be readily applied to an actual physical setup with access to experimental data. This amounts to placing bounds on the size of the uncertainty Δ.

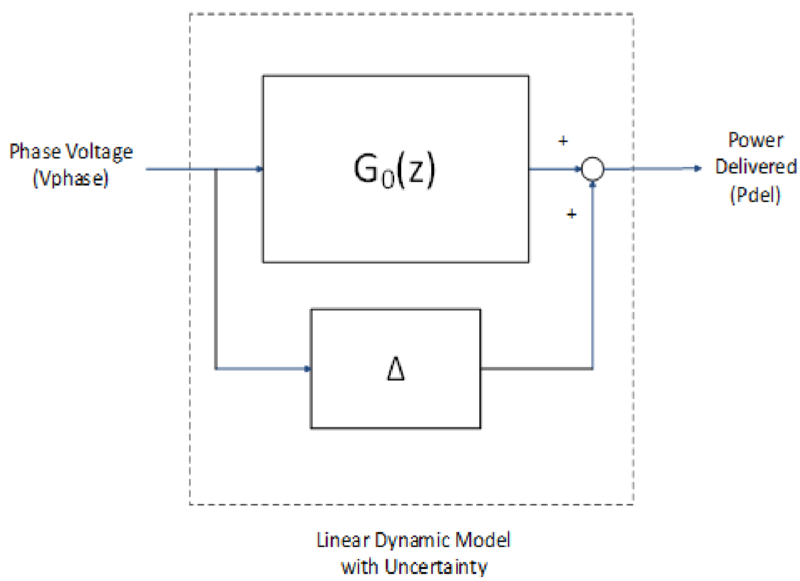

Figure 4.2: Linear-plus Uncertainty Model for Robust Controller Design

The first step is to create the nominal model using the process that will be described in sections 5.1.1 and 5.1.2. Then apply the test signal Vphase to each operational point and collect the time series data Vphase[k] and Pdel[k]. Apply a Fast Fourier Transform (FFT) to this time series data to get Vphase($e^{j\Omega}$) and Pdel($e^{j\Omega}$), which are frequency domain representations. A variety of tools, such as the Matlab fft command, can be used to accomplish this.

This procedure is performed several times, resulting in several copies of Gdata($e^{j\Omega}$). To characterize the uncertainty size in the frequency domain, we now use the nominal model transfer function by choosing W($e^{j\Omega}$) as the smallest limit satisfying:

$$| \, tf\,1(e^{j\Omega}) - G_{data}(e^{j\Omega}) \, | \leq | \, W(e^{j\Omega}) \, |$$

The value of $|W(e^{j\Omega})|$ now characterizes the uncertainty size, since we guarantee that for all possible uncertainties $\Delta(z)$ we have:

$$|\Delta(e^{j\Omega})| \leq | \, W(e^{j\Omega}) \, |$$

As a result, our uncertainty is constrained by $|W(e^{j\Omega})|$. All that is left to do is fit a stable, minimum-phase transfer function W(z) to the frequency response magnitude data $|W(e^{j\Omega})|$ for (robust) controller design (ignoring the phase response). We built tools (coded as the Matlab fitsysmag command) based on the Matlab Robust Control Toolbox commands genphase and fitsys to accomplish this. The resulting stable, minimum-phase weight W(z) is suited for robust controller design approaches.

4.2 CONTROLLER DESIGN

In this section we discuss the controller design approaches, with a primary focus on robust control. To specify the robustness requirements, the robust controller design approaches developed here rely on a (gain-scheduled) set of linear models at a grid of operational points, with uncertainty descriptions from the system modeling efforts.

4.2.1 ROBUST CONTROL FOUNDATION

For any matrix (or vector), M, we denote the largest singular value as $\bar{\sigma}(M)$. Given a transfer matrix P(z) we denote its H-infinity norm by:

$$\|\boldsymbol{P}\|_\infty \doteq \sup_{\Omega\in[0\pi]} \bar{\sigma}(\boldsymbol{P}(e^{j\Omega}))$$

The above statement holds for the general MIMO case, but note that in the SISO case this definition reduces to:

$$\|\boldsymbol{P}\|_\infty \doteq \sup_{\Omega\in[0\ \pi]} \left|\boldsymbol{P}(e^{j\Omega})\right|$$

which is simply the peak value on the Bode (magnitude) plot.

The structured singular value, μ, is a (constant) matrix function. It depends upon the underlying block structure of the uncertainties, which is defined as follows. The block structure K is a set of positive integers:

$$K = (k_1, \dots, k_m)$$

which specifies the dimensions of the perturbation blocks.

This determines the set of allowable (block-diagonal) constant matrix perturbations, namely:

$$X_K = \{\Delta = \text{block diag}(\Delta_1^C, \dots, \Delta_m^C) \colon \Delta_i^C \in C^{k_i \times k_i}\}$$

Now the structured singular value, $\mu_K(M)$, of a matrix M, with respect to a block structure *K,* is defined as:

$$\mu_K(M) = \left(\min_{\Delta\varepsilon X_K}\{\bar{\sigma}(\Delta) : \det(I - \Delta M) = 0\}\right)^{-1}$$

This function is being introduced since it is at the heart of robust control analysis and design. We'll explore later how to calculate/optimize the structured singular value over frequency to solve general robust stability and performance difficulties. To move forward, we'll go over some key tools from optimal and robust controller design theory.

4.2.2 CONTROLLER SYNTHESIS TOOLS

Consider the problem of generic synthesis depicted in Figure 4.3. P(z) is the nominal transfer matrix (assumed to be specified a-priori), and K(z) is the controller in this illustration (to be designed). A (block-diagonal) structured uncertainty perturbs the closed loop system. Exogenous disturbances, error signals, measurements, and control signals are represented by the signals w,z,y,u in this diagram. We will soon propose a precise definition for structured uncertainty, which describes unknown or unmodeled dynamics.

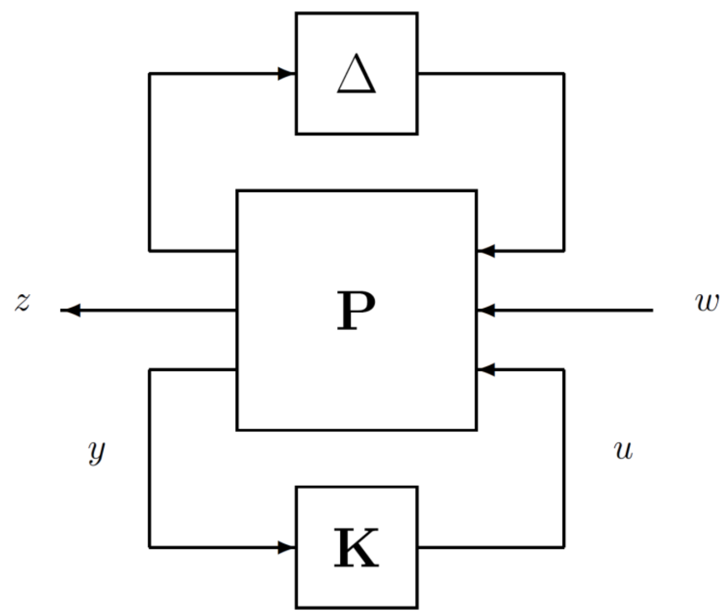

Figure 4.3: Design Interconnection for Controller Synthesis

We will explore the general design challenge of selecting K(z) in order to achieve robust performance, that is, selecting K(z) such that the perturbed closed loop system is stable and the worst-case gain from exogenous disturbances (w) to error signals (z) is modest, for all permitted. Note that this is what is meant by "robust performance," since it means that despite perturbations, the system will remain stable and massive disturbances will only create modest error signals. Of course, this ambiguous assertion will be rendered mathematically rigorous and also computable, for both analysis and controller design.

A set of structured dynamic perturbations are defined as:

$$M(X_K) \doteq \{\boldsymbol{\Delta}\ stable{:}\, \boldsymbol{\Delta}(e^{j\Omega}) \in X_K \text{ for all } \Omega \in [0 \quad \pi]\}$$

In this perspective, the frequency response of any unmodeled dynamics is seen as the constant matrix uncertainty stated earlier (specifically X_K). It's also worth noting that the earlier-mentioned block diagonal structure (X_K) enables us to solve situations with numerous sources of uncertainty.

The transfer matrix from w to z is denoted by $\mathbf{T_{zw}}$ (z). Our performance aim is to keep the gain of this transfer matrix low (because it is the transfer matrix from disturbance to error), which we will measure as its H_∞ norm. As a result, our steadfast performance target might be described as follows: *Choose **K**(z) so that the perturbed closed loop system in figure 3.4 is stable, and* $\|T_{zw}\|_\infty \leq 1$, *for all* $\Delta \varepsilon M(X_K)$ *with* $\| \Delta \|_\infty < 1$.

Note that the above result is applicable to MIMO systems and the result is normalized, with performance/uncertainty bounds set to unity. In reality, however, any desired level of robustness and performance may be specified because all signals/uncertainties can be weighted and the weights are simply absorbed into P(z). The selection of an appropriate P(z) and weights is, in fact, at the heart of the design process.

Since it can be proven that robust performance in figure 4.3 is exactly similar to robust stability in figure 4.4, this problem can be transformed into a μ synthesis problem. We've introduced an extra "performance" uncertainty Δ_p here, which closes the loop from w to z in figure 4.3 and results in this equivalency.

The nominal closed loop transfer matrix derived from P(z) and K(z) is defined as M(P,K)(z) (see figure 5.2). The following well-known robust performance theorem is derived from these definitions. Assume that the nominal system M(P,K)(z) is stable (by selecting K(z) appropriately). The perturbed closed loop system in figure 4.3 is then stable, and this holds true for $\|T_{zw}\|_\infty \leq 1$ *for all* $\| \Delta \|_\infty < 1$ iff.

$$\sup_{\Omega \in [0 \pi]} \mu_{\hat{R}}(\boldsymbol{M}(\boldsymbol{P},\boldsymbol{K})(e^{j\Omega})) \leq 1$$

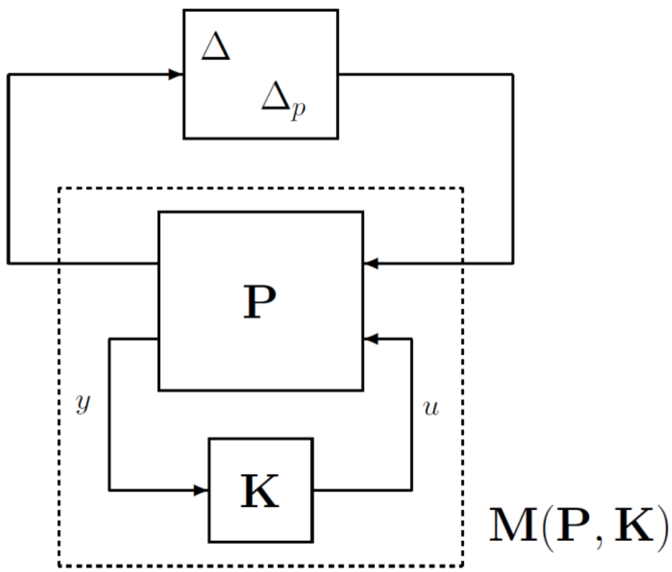

Figure 4.4: Equivalent Robust Stabilization Problem

Robust control theory is based on this result. Simply take the nominal closed-loop system M(z), run $M(e^{j\Omega})$ as a frequency sweep, and compute at each frequency μ(M). The resulting cross-frequency peak value of precisely solves the robust stability and performance difficulties highlighted in figure 4.3. We can also reorganize any linear fractional interconnection of systems and (many) uncertainties into the conventional robustness analysis form of figure 4.3, as previously indicated.

4.2.3 μ-SYNTHESIS FOR CONTROLLER DESIGN

Given the preceding result, our goal shifts to the generic synthesis problem of finding a controller K(z) that minimizes the peak value of μ across frequency:

$$\inf_{\boldsymbol{K} stabilizing} \sup_{\Omega \in [0 \pi]} \mu_{\widehat{K}}(\boldsymbol{M}(\boldsymbol{P}, \boldsymbol{K})(e^{j\Omega}))$$

This controller improves robust performance (i.e., it minimizes the gain from w to z in figure 4.3 for permitted uncertainties as large as possible) (robustness). All of this is accomplished while ensuring closed-loop stability for all permissible perturbations $\boldsymbol{\Delta}$.

The design approach for μ-synthesis incorporates the following stages:

1. Begin by collecting information and creating a model for the nominal physical system.
2. Select a suitable interconnection structure for modeling the (perturbed) system, taking into account the uncertainty structure that resilience is sought. This includes disturbances as well as signals that should be penalized or minimized.
3. Determine the appropriate weights to reflect the required performance standards as well as any information about the uncertainties that is available (uncertainty characterization).
4. In Matlab, implement the μ-synthesis design.

Despite the fact that these methods are based on computer-aided optimization, the process is far from automated, and engineering judgment is still necessary.

4.2.4 ROBUSTNESS ANALYSIS TOOLS

Since we are concerned with analysis results in this subsection we assume both **P**(z) and **K**(z) (and hence **M((P,K)**(z)) have already been chosen by the design engineer. Furthermore, the design engineer has chosen a stabilizing controller **K**(z), i.e., the nominal closed-loop system **M**(z) is stable. Consider **M**(z):

$$M(z) = \begin{pmatrix} M_{11}(z) & M_{12}(z) \\ M_{21}(z) & M_{22}(z) \end{pmatrix}$$

Then we have the following robustness analysis results:

Robust Stability: *The perturbed closed loop system in figure 4.3 is stable for all* $\| \Delta \|_\infty < 1$ *iff:*

$$\sup_{\Omega \in [0 \pi]} \mu_K(\boldsymbol{M}_{11}(\boldsymbol{P}, \boldsymbol{K})(e^{j\Omega})) \leq 1$$

Robust Performance: *The perturbed closed loop system in figure 4.3 is stable, and* $\|T_{zw}\|_\infty \leq 1$*, for all* $\| \Delta \|_\infty < 1$ *iff:*

$$\sup_{\Omega \in [0 \pi]} \mu_{\hat{R}}(\boldsymbol{M}(\boldsymbol{P}, \boldsymbol{K})(e^{j\Omega})) \leq 1$$

4.3 LINEAR PARAMETER VARYING (LPV) CONTROL

By this stage let us assume we would have built a grid of reliable controllers that span the operating envelope, with the controller operating points spaced roughly 500W apart. The overall control scheme covers the complete operating envelope (in this case, 0 - 4,500W), while each individual controller only runs in a limited range (say, up to 500W from its designed operating point). This is enough to keep any specified setpoint values in place. However, we must modify this strategy if we want to handle arbitrary waveforms that may transition between many operational locations. The development of a single Linear Parameter Varying (LPV) controller that can manage the whole operating range is critical. LPV controllers are reliant on a set of factors. Although this dependency is nonlinear, the controllers themselves are linear.

4.3.1 ROBUST LPV CONTROLLER

For the robust controller consider the Controller–K-Linear block in Figure 5.40 (discussed later in section 5.3). The robust control algorithm's basis is the Discrete-Time State-Space block, which can be formally represented as:

$$\begin{aligned} x[k+1] &= Ax[k] + Bu[k] \\ y[k] &= Cx[k] + Du[k] \end{aligned}$$

Note that this is Linear Time-Invariant (LTI) since the A, B, C, D matrices are constant. Now consider allowing these matrices to vary as:

$$\begin{aligned} x[k+1] &= A(\rho)x[k] + B(\rho)u[k] \\ y[k] &= C(\rho)x[k] + D(\rho)u[k] \end{aligned}$$

where ρ is a parameter vector. This represents an LTI system for frozen values of ρ; however, when is permitted to vary, this explains an LPV system. It's also worth noting that each of the matrices; $A(\rho)$, $B(\rho)$, $C(\rho)$, and $D(\rho)$ could have a nonlinear reliance on this collection of values.

CHAPTER 5

APPLICATION TO AE SYSTEM AND RESULTS

In this section we will look into application of the theoretical approaches mentioned in the previous chapter onto the plasma power generator digital twin model provided by AE, and subsequently carry out system modeling, control design and robust LPV control design.

5.1 MODELS

The process and results of carrying out system modeling and identification is detailed in the subsections that follow.

5.1.1 MODELING PROCESS

Static nonlinear models are based on the steady-state gains of the system under consideration. As a result, the constant output levels for given constant inputs can be used to characterize them. This is usually accomplished through the system's step reaction.

Consider a 1,000W Power Setpoint as an example of one operating point. In figure 5.1, the results of this AE simulation model are shown.

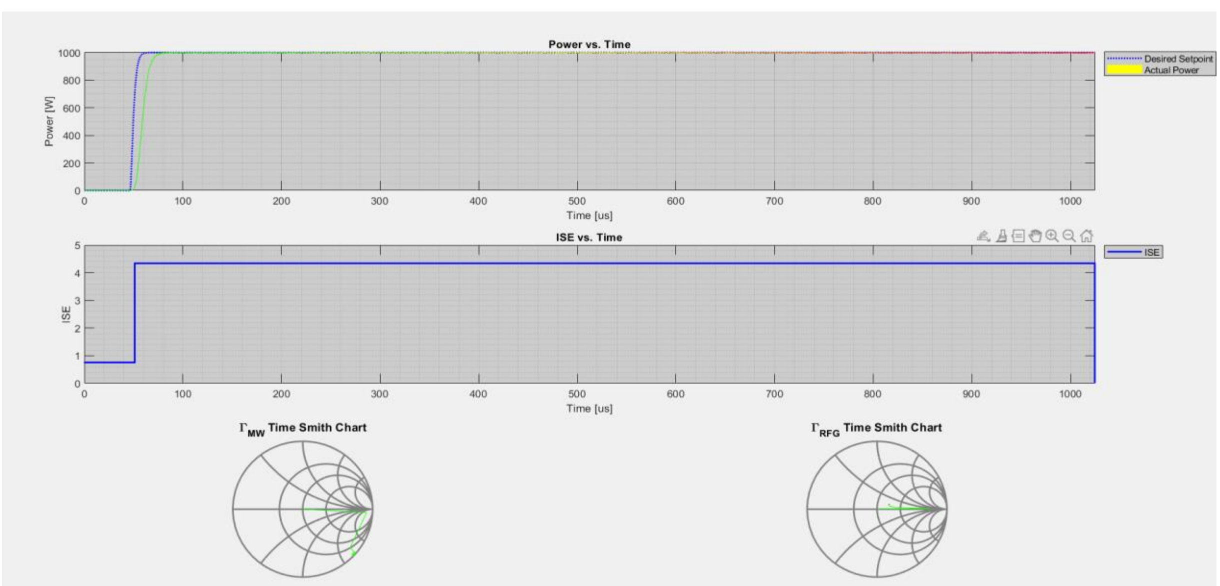

Figure 5.1: AE Simulation Model System Step Response for 1000 W

We wish to characterize the mapping from Phase Voltage (Vphase) to the Power Delivered (Pdel) by extracting the steady-state values for these signals. These steady state values are named Vphasess and Pdelss for steady-state Vphase and steady-state Pdel respectively. Small data windows can be averaged if necessary to reduce the impact of noise and other factors.

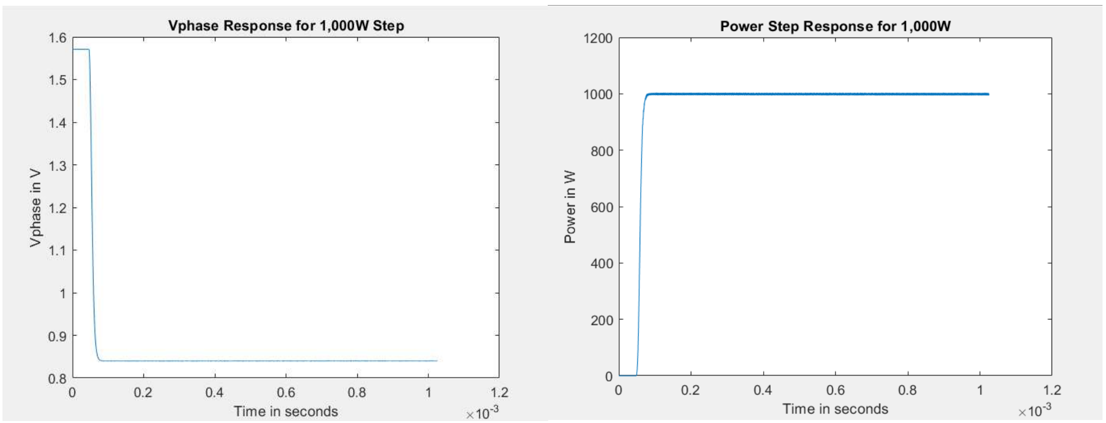

Figure 5.2: Vphase and Pdel for a Power Setpoint Step of 1000 W

Carrying out the steady state analysis for the signals in figure 4.3 results in:

Vphasess = 0.84V, Pdelss = 998W

This corresponds to our first data point in the nonlinear static mapping, namely:

Vphasess = 0.84V $\rightarrow$ Pdelss = 998W

We can now repeat this operation across the whole operating envelope from 500 - 4,500W.

Carrying out this process results in the data shown in table 5.1.

Table 5.1: Steady State Values for Vphasess and Pdelss across Operating Envelope

setpoint	vphasess	pdelss
500	1.045	502
1000	0.839	998
1500	0.692	1499
2000	0.576	2003
2500	0.480	2497
3000	0.396	3003
3500	0.320	3506
4000	0.253	4012

Table 5.1 now provides us with the whole static nonlinear model (for Impedance = 50 and Frequency = 13.4MHz) for the entire working envelope. In practice, this might now be used as a lookup table with the required interpolation. The fitting of a static nonlinear curve to this data is likewise straightforward, as shown in figure 5.3.

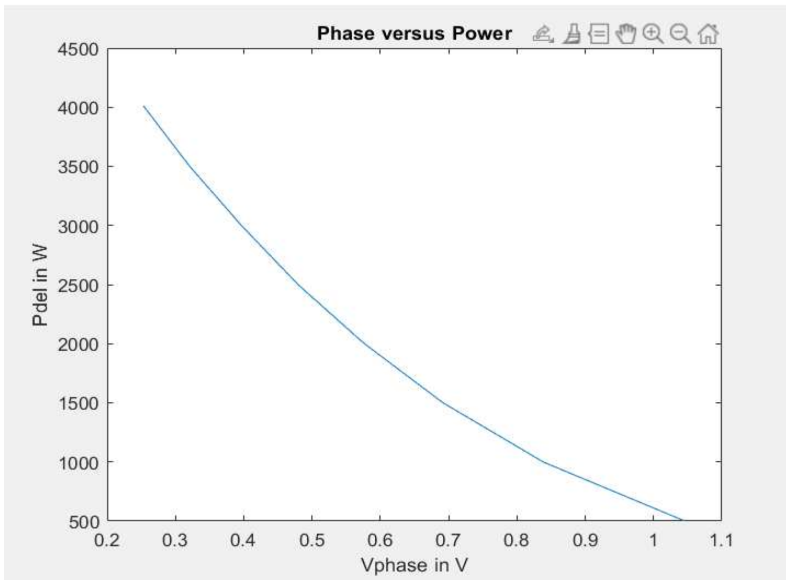

Figure 5.3: Nonlinear Static Gain from Vphase to Pdel

This static increase is evidently nonlinear. The nonlinearity looks to be minimal, and the function is close to being affine. This may not always be the case, but in any event, we may characterize any system of interest using the process presented here.

5.1.2 LINEAR DYNAMIC MODELS

To simulate the dynamic response, we require an input signal that is centered around the target operating point of 1,000W, but includes tiny signal changes away from it. For example, a signal made consisting of tiny increments superimposed on a steady-state 1,000W signal can be used. There are various options for such a signal, however we utilized a two-sided 25W square wave superposed on a steady 1,000W signal for demonstration purposes. Figure 5.4 depicts the simulation findings.

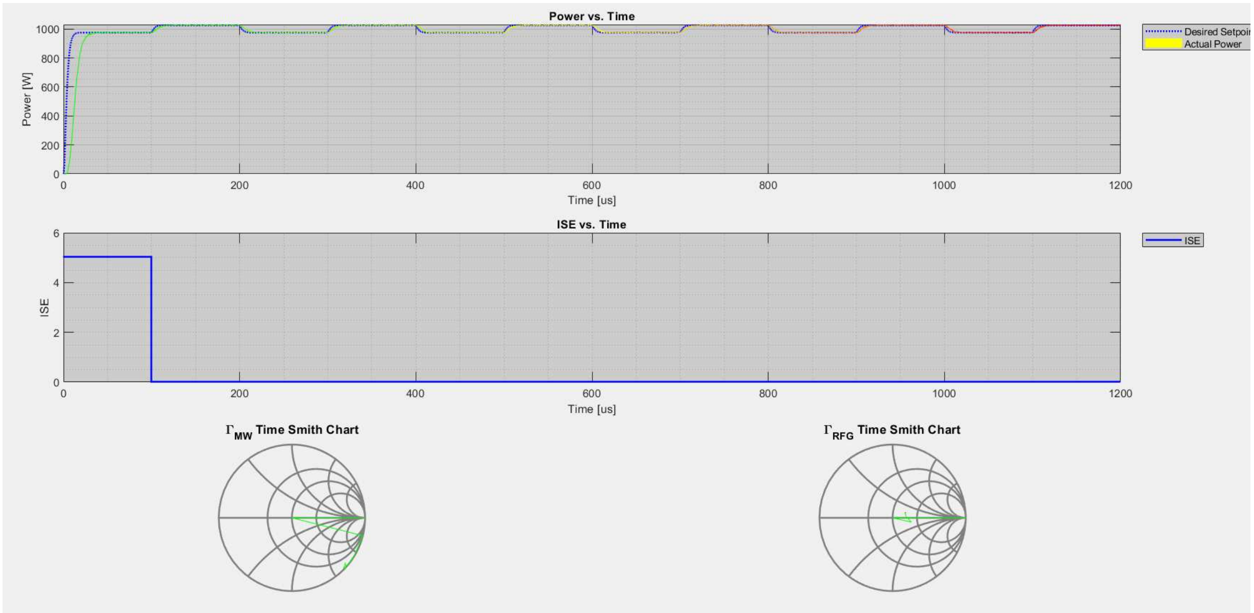

Figure 5.4: AE Simulation Model Dynamic Response around 1000 W

Extracting the Vphase and Pdel signals from the AE simulation data yields the signals in figure 5.5.

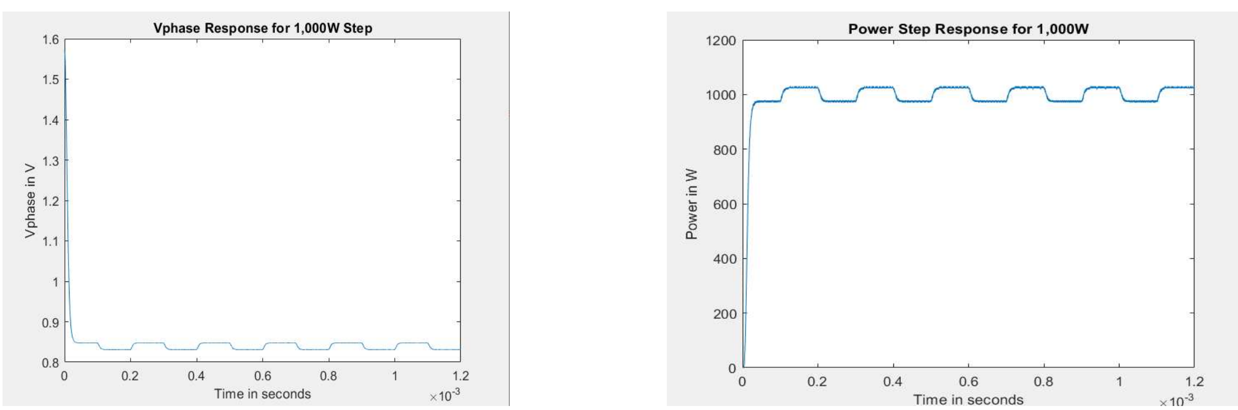

Figure 5.5: Vphase and Pdel Signals for Dynamic Signal around 1000 W

We must first preprocess these signals to allow for a better match before we can proceed with model fitting. The first step is to eliminate the (nonlinear) steady-state offsets, which were already taken into account in our model. Table 5.1 provides an easy way to obtain the values (in this case for an operating point of 1,000W, which is highlighted). This will result in signals that are (in both cases) centered around zero, which is ideal for a linear mapping. Figure 5.6 depicts the linearized signals.

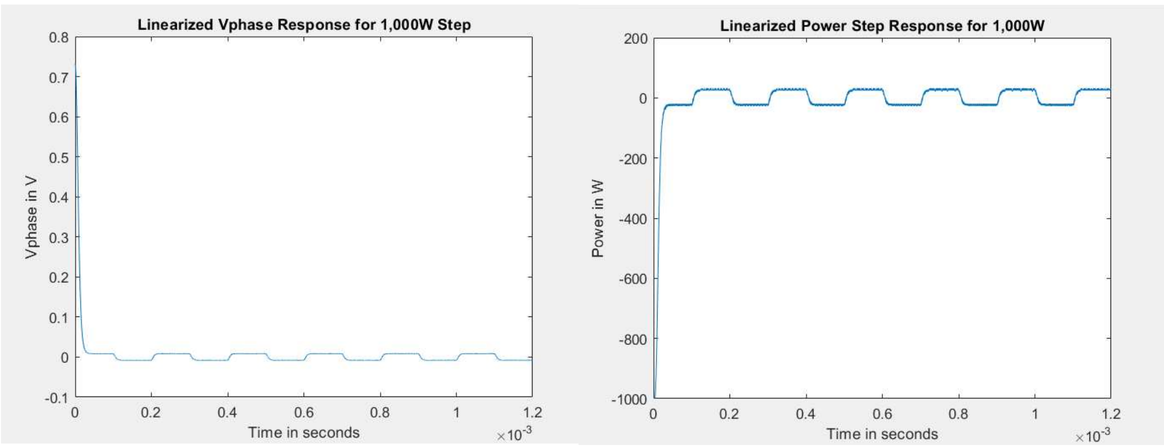

Figure 5.6: Vphase and Pdel Linearized Prior to Fitting

Our original operating point is now mapped to the origin for both signals (prior to fitting a linear model). Since the linear model is only intended to capture dynamics in the neighborhood of the operating point, we mask the first part of the signal (in this case the first 0.15 milliseconds) to remove this transient as shown in figure 5.7.

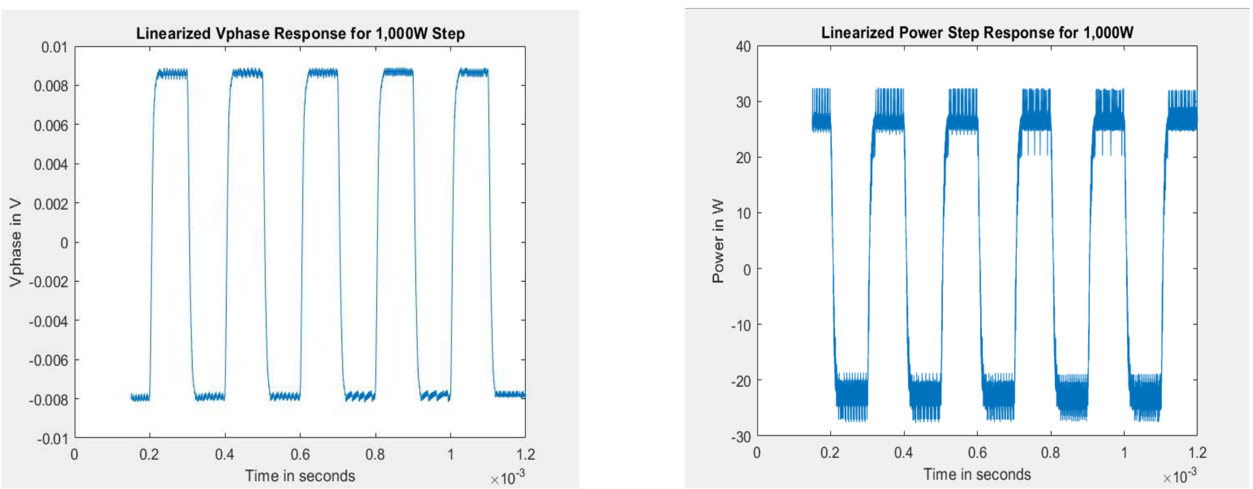

Figure 5.7: Linearized Vphase and Pdel - PreProcessed Prior to Linear Model Fit

The signals are now ready to be fitted into a linear dynamic model. To fit a Discrete-Time Transfer Function model operating at a sample rate of Ts = 2.5x10-7 seconds, we use some of the Matlab System Identification tools (from the AE model) [6]. The GUI interface was used, as it provides a variety of appropriate techniques and alternatives. Figure 5.8 illustrates a sample screenshot.

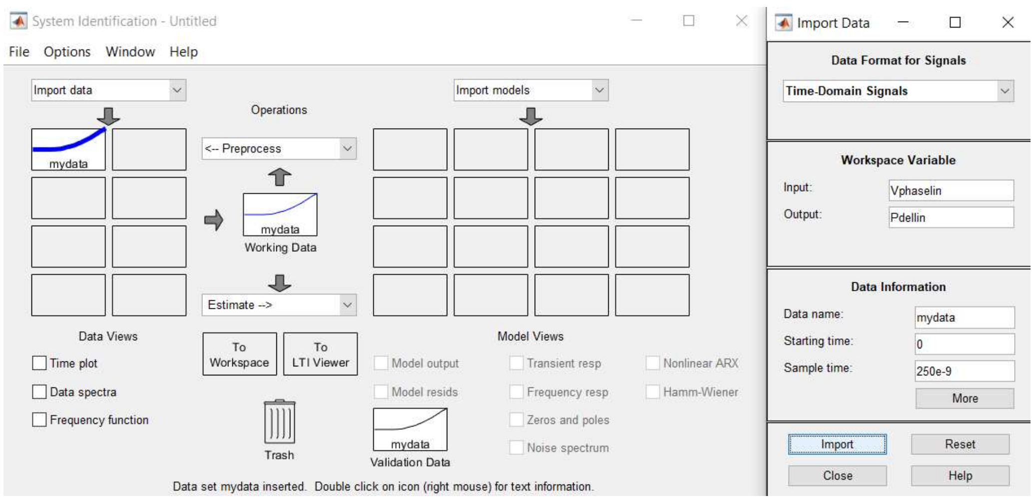

Figure 5.8: Matlab System Identification GUI Screenshot

Figure 5.9 shows the input-output (u1-y1) signals that correspond to the preprocessed Vphase and Pdel data from figure 5.6.

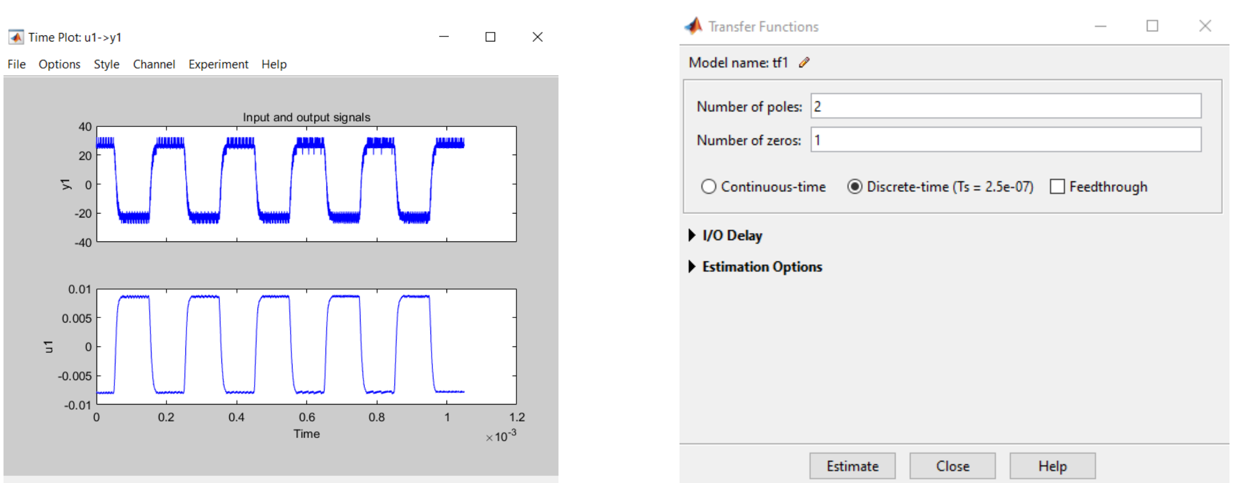

Figure 5.9: System ID Matlab Tool for Linear System Fit

Selecting a 2nd order linear model fit results in the fitted response shown in figure 5.10, which corresponds to the following transfer function:

-2060 z^-1

----

1 + 0.1592 z^-1 - 0.4763 z^-2

Name: tf1

Sample time: 2.5e-07 seconds

Discrete-time identified transfer function.

Parameterization:

Number of poles: 2   Number of zeros: 1

Number of free coefficients: 3

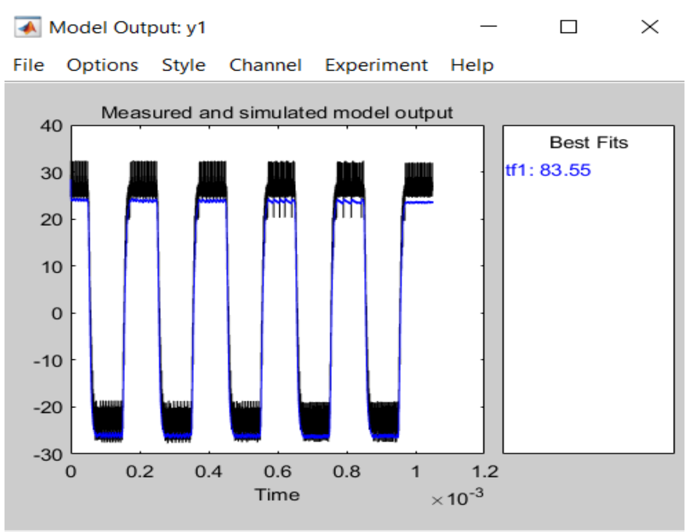

Figure 5.10: Linear Dynamic 2nd Order Fit (Blue) to Simulation Data (Black) for Pdel

Despite the low order transfer function, the fit error is observed to be fairly minimum (model data in blue, and original simulation data in black). Indeed, we expect this fit to be accurate enough for its intended purpose, however, proof of that fact will have to wait till later.

We now have a linear model fit that seems to work well. Of course, the linearized signals from figure 5.7 were used in this fit. It's important to remember that the actual signals (while still blanking the first 0.15 milliseconds to avoid the initial transient) look like figure 5.11.

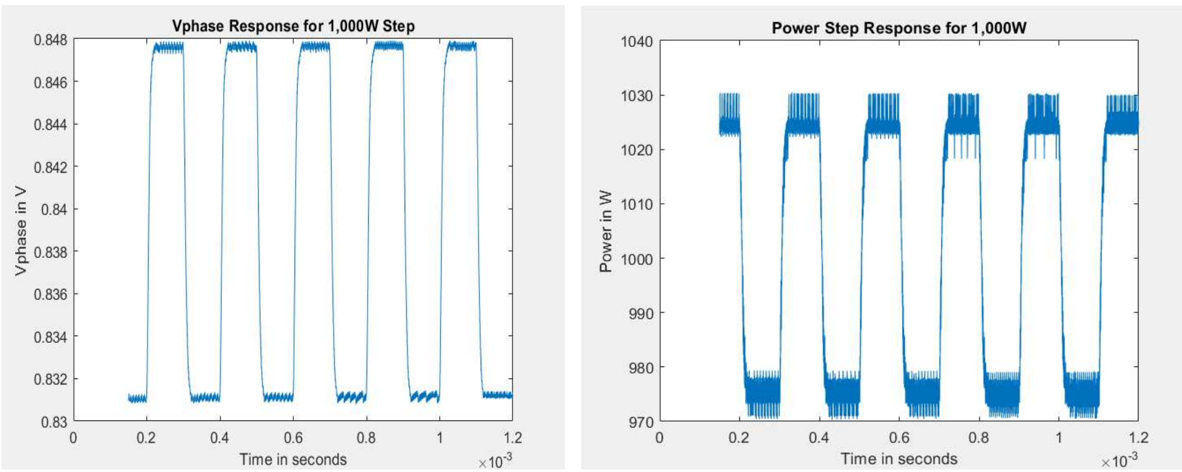

Figure 5.11: Vphase and Pdel Dynamic+Steady State Response

This is the actual response of the AE system. We must include both the nonlinear steady-state gains and the linear dynamic model response to reconstruct this response. This is easily accomplished by creating a simple simulation model like the one shown in figure 5.12.

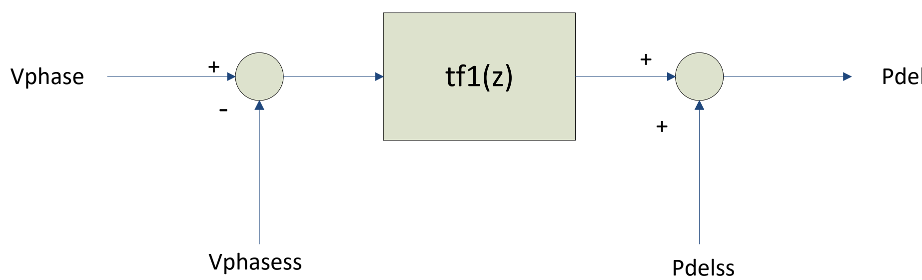

Figure 5.12: Simple Identified Model for AE *Digital Twin* Model

This model incorporates both linear and nonlinear effects. Figure 5.13 shows a comparison of the output of the simple model in figure 5.12 with the output of the high-fidelity AE Digital Twin model (see figure 3.2) full size on the left and zoomed in (on a typical part) on the right.

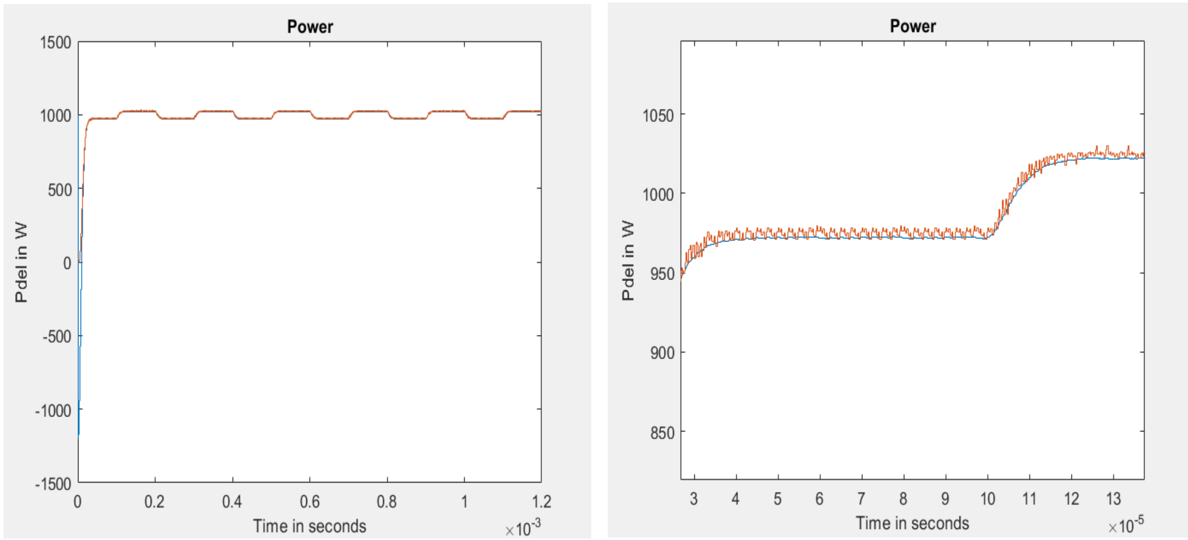

Figure 5.13: *Digital Twin* Simulation (Red) Compared to Simple Model (Blue)

The simple model adequately captures both steady-state and transient/dynamic effects (note simple model data is in blue, and Digital Twin simulation data is in red). This approach is only useful for minor alterations in the selected operating point (in this case 1,000W). The fit is very good in the vicinity of the operating point at 1,000W, as can be observed. A large initial transient in the simple model (Blue) is due to the fact that we are starting from 0W, which is a long way from our operating point of 1,000W, and so our simple model is not very accurate in this region. This will not be a problem for controller design because controllers will be built similarly around operating points, with a scheduling method that combines them into a single nonlinear control approach for the whole operating envelope.

Table 5.1 depicted the static nonlinear model data across this operating envelope. Tables 5.1and 5.2, when combined, provide a complete set of simple models, such as those shown in figure 5.12, for the whole operating envelope.

Table 5.2: Linear Transfer Function Models Across Operating Envelope

Setpoint	Transfer Function (2 poles, 1 zero)	fit
500	$\dfrac{-5036\, z^{-1}}{1 + 0.6558\, z^{-1} + 0.807\, z^{-2}}$	77.83
1000	$\dfrac{-2060\, z^{-1}}{1 + 0.1592\, z^{-1} - 0.4763\, z^{-2}}$	83.55
1500	$\dfrac{-9076\, z^{-1}}{1 + 0.6762\, z^{-1} + 0.625\, z^{-2}}$	88.5
2000	$\dfrac{-1.105e04\, z^{-1}}{1 + 0.6803\, z^{-1} + 0.5291\, z^{-2}}$	89.92
2500	$\dfrac{-2915\, z^{-1}}{1 + 0.08739\, z^{-1} - 0.5743\, z^{-2}}$	86.68
3000	$\dfrac{-3157\, z^{-1}}{1 + 0.147\, z^{-1} - 0.6306\, z^{-2}}$	90.31
3500	$\dfrac{-1.48e04\, z^{-1}}{1 + 0.6819\, z^{-1} + 0.3666\, z^{-2}}$	91.82
4000	$\dfrac{-3581\, z^{-1}}{1 + 0.1499\, z^{-1} - 0.6696\, z^{-2}}$	89.32

5.2 CONTROLLER DESIGN

In this section we discuss the controller design approaches with a primary focus on robust control. As previously stated, we will treat the Digital Twin (Matlab/Simulink model) from Figure 3.2 as if it were the true physical system under discussion.

5.2.1 DESIGN INTERCONNECTION P(Z) FOR AE SYSTEMS

In addition to containing the physical model ($\mathbf{G_0}(z)$), the design interconnection $\mathbf{P}(z)$ will specify the following:

- Uncertainties are present in the system (Δ)

- Disturbance signals act on the system (w)
- Signals to be penalized in the optimal control problem (z)
- Control signals (u), and measurements (y), available to us

This sets the basic framework for the design tradeoffs. After several different iterations, we finally settled on a design interconnection of the form shown in figure 5.14. First note that if we make the obvious definitions:

$$w = \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} \qquad z = \begin{pmatrix} z_1 \\ z_2 \end{pmatrix}$$

then figure 5.14 fits exactly into the structure of figure 4.3, and so this defines a design interconnection **P**(z). Indeed, if we omit the dashed boxes for Δ and **K**(z) in figure 5.14, then what is left is exactly **P**(z).

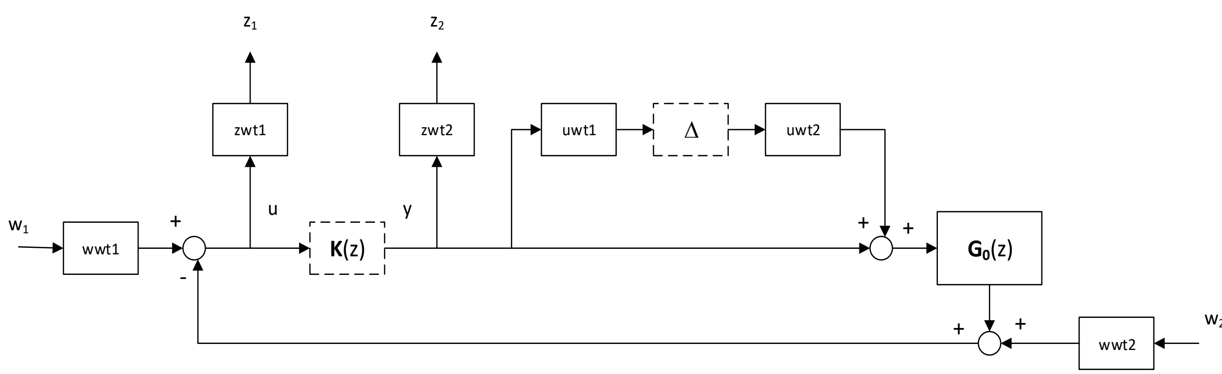

Figure 5.14: General Template for Design Interconnection **P**(z)

The measurements (y) and control signals (u) in this case are Pdel and Vphase respectively (i.e., Delivered Power and Phase Voltage). We choose to employ (weighted) multiplicative unstructured uncertainty at the plant's input because we have insufficient precise knowledge about the uncertainty sources. This is based on our design experience with various systems, where this uncertainty description has shown to be quite useful.

Consider first the disturbance signals (w). The reference input is represented by the signal w1. This is the signal to be monitored, and it is part of our primary performance goal, but the controller treats it as a disturbance that is attempting to increase the tracking error in this formulation. At first glance, this may appear to contradict our common sense, although it is mathematically comparable in terms of optimization. The w2 signal denotes measurement noise. Although it functions at the same time as w1, it is more convenient to have it as a separate signal for weight selection and to prevent some technical concerns that can lead to a singular control problem.

Now consider the penalty signals (z). This is the set of signals that the controller optimization will attempt to reduce. The principal penalty signal z1 is the tracking error for the delivered power. The major purpose of the control strategy is to reduce this error. However, if we simply penalized this signal, we might end up with a singular control problem or one that relies on excessive control activity (which in practice would saturate Vphase). As a result, we've included a z2 penalty on control authority. This eliminates the problems caused by the aforementioned issues.

5.2.2 PERFORMANCE AND UNCERTAINTY WEIGHT SELECTION

The selection of weighting functions is crucial to both robust controller design procedures like μ -synthesis and optimum controller synthesis methodologies like H_∞-optimal control. These weights define the tradeoffs between performance and robustness goals, as well as the optimization criteria.

We specify the continuous-time form of the weights in what follows for convenience of presentation and understanding (and this is how they will be specified in the design interconnection – see figure 5.15 later). The design process, on the other hand, automatically converts them to discrete-time using something similar to a bilinear transformation:

$$W(z) = W(s)|_{s=\frac{2}{T_s}\left(\frac{z-1}{z+1}\right)}$$

where for the AE designs presented here the sample period is T_s = 250ns (2.5×10^{-7} seconds).

Uncertainty Weights: For illustrative purposes here these may be simply chosen as:

$$uwt1 = uwt2 = \sqrt{0.07}$$

representing about 7% multiplicative (relative) uncertainty.

Disturbance Inputs: The major disturbance input is w1, which is the reference input (desired power level) to be tracked, with a weight of wwt1. This weight could be chosen to reflect the signal's spectrum. Because we can't expect to follow extremely high frequency (or very fast) signals, it's usually a low-pass filter, and it's also usually rigorously correct (consider the limiting values of the above argument as frequency tends to infinity). The true reference input (for intended power level) in our situation is normally a series of step functions, but the AE system filters this using a critically-damped second-order system to provide a more sensible signal to track. In light of this, the following is a good alternative for wwt1:

$$wwt1 = \frac{K{\omega_0}^2}{s^2 + 2\xi\omega_0 s + {\omega_0}^2}$$

where reasonable parameter choices are $\zeta = 1$ (critical damping), ω_0 = 2,000 rad/s, and K = 10^{-3}

The disturbance input w2 is used to account for measurement noise, and it typically goes all the way up to high frequencies. As a result, biproper (i.e., not strictly proper) or constant functions are commonly used. A constant function (representing a flat noise spectrum, or in other words white noise) is a viable choice in this case, such as:

$$wwt2 = 10^{-5}$$

Since the DC gain of *wwt1* is given as K = 10^{-3}, this is a noise level about 1% of that value.

Penalty Signals: The tracking error (z1) is the principal penalty signal (for desired power level). This weight's spectrum is often set to match the bandwidth of the desired tracking performance (and once again, for reasons discussed earlier, it is usually strictly proper). A first order transfer function of the form:

$$zwt1 = \frac{\widehat{K}}{s + p}$$

where reasonable parameter values are $\widehat{K} = 10^5$ and p = 10^3

To avoid the usage of excessive control authority for Vphase, it is critical to add a penalty on z_2 signal (which in practice will just saturate). Because substantial high-frequency changes in control authority are particularly undesirable, the weight zwt2 is usually a high-pass filter or a constant. In any event, due to the limiting situation (as frequency goes to infinity) of the foregoing argument, it is generally biproper (not exactly proper). A decent choice for the AE system was discovered to be:

$$zwt2 = 5$$

Figure 5.15 depicts the final connectivity, which was developed in Simulink (it implements the P(z) illustrated schematically in figure 5.14).

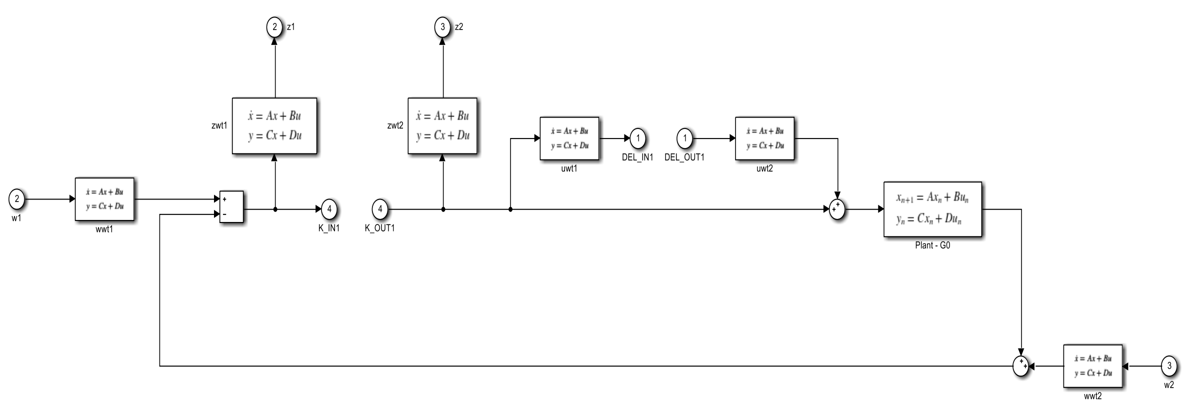

Figure 5.15: Simulink Design Interconnection **P**(z)

5.2.3 SIMULATION MODELS AND IMPLEMENTATION ON AE DIGITAL TWIN

The AE *Digital Twin* Matlab/Simulink model is shown in figure 3.2. It gives a complete closed-loop system model, including the AE control scheme. Figure 5.5 is the AE setpoint and Vphase controller which forms the basis of the AE control scheme.

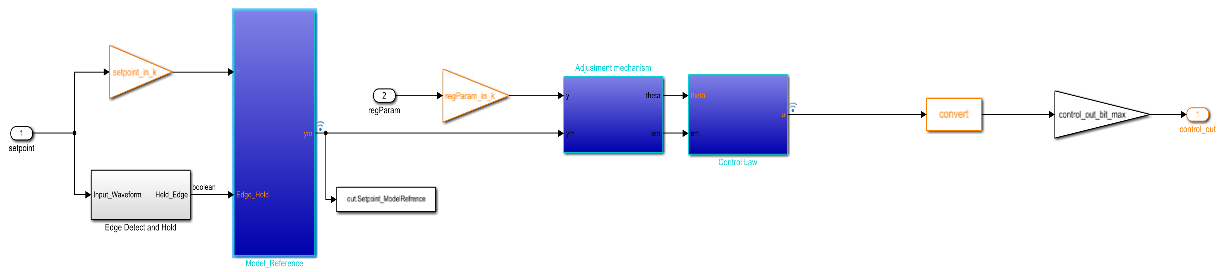

Figure 5.16: Full AE Control Scheme

The initial portion of the control system (Model Reference) generates the filtered setpoint (using a second-order filter). The central portion of the controller from figure 5.16, which is shown in figure 5.17, we see that it consists of two main components. The first component (Adjustment mechanism) is responsible for adaption (based on L1 adaptive control theory). PID control is implemented in the second component (Control Law).

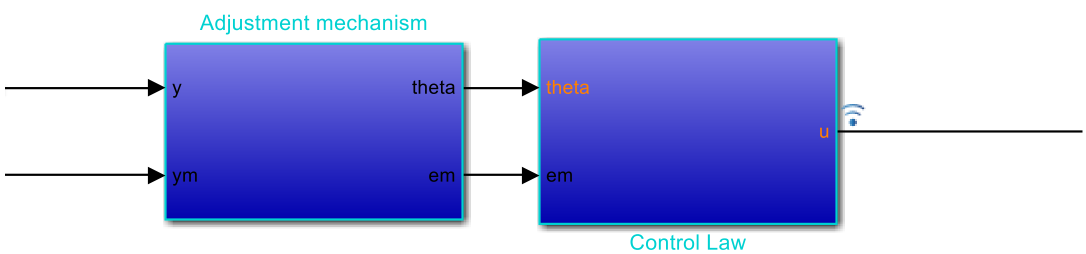

Figure 5.17: AE Vphase Controller

To test our controller designs on the AE *Digital Twin* we simply swap out the controller in figure 5.17 with our new controller. The overall control scheme is illustrated in figure 5.18, and the (zoomed-in) Vphase controller is shown in figure 5.19.

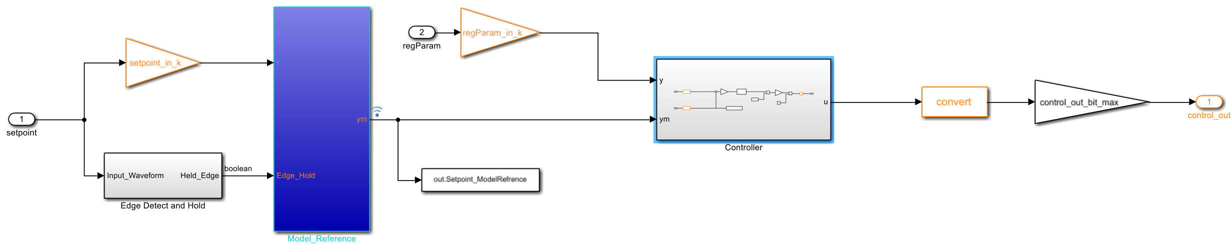

Figure 5.18: New Full Control Scheme

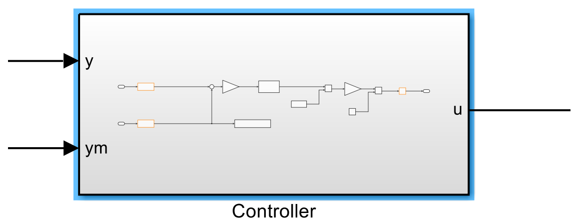

Figure 5.19: New Vphase Controller

The robust controller is implemented in the block *Controller-K-Linear*, as a State Space representation. Nonlinear offsets are implemented as can be seen in figure 5.20, which also implements several other variable changes and formats, so that it interacts correctly with the rest of the *Digital Twin* nonlinear simulation.

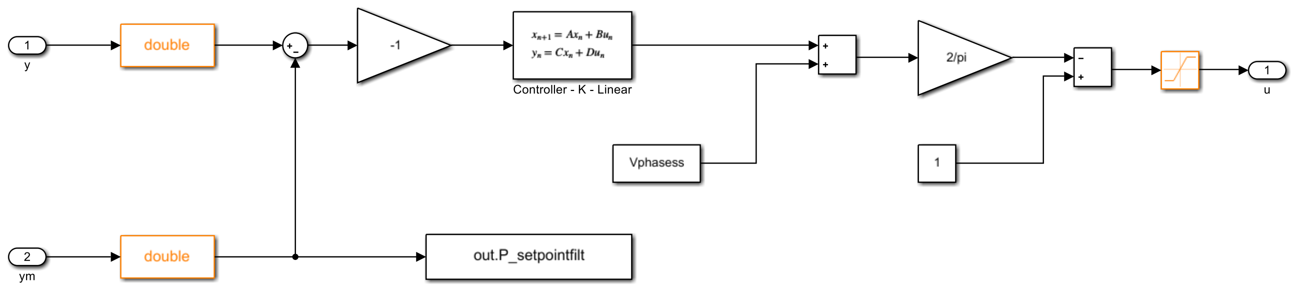

Figure 5.20: Robust Controller *Digital Twin* Implementation

Our controller designs now have full nonlinear simulation capability thanks to the Digital Twin implementation. However, we'd like to examine what the simple linear models (used for design) predict for performance for comparison purposes. As a result, we created a simple linear simulation model in Simulink, which is depicted in figure 5.21.

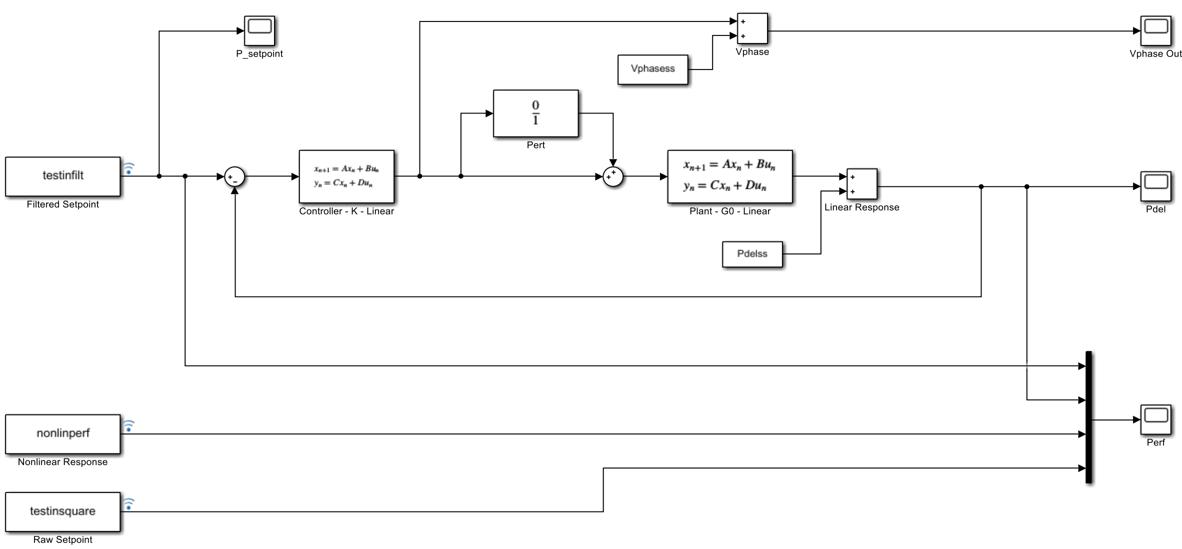

Figure 5.21: Linear Simulation Model

The linear simulation model in figure 5.21 uses the same (State Space) linear controller (Controller-K-Linear) as the nonlinear simulation model in figure 5.20. As illustrated in figure 5.12, the plant model has been replaced with the simple discrete-time transfer function model G0(z) with appropriate offsets. The design templates and simulation models mentioned in this paragraph are now being utilized to create and test robust controllers for the AE system, as detailed in the section below.

5.2.4 BASELINE AE SYSTEM

To establish a baseline, we examine the performance of the current AE control system. We perform the Digital Twin simulation with a test signal fluctuating around a setpoint of 1,000W. Figure 5.22 depicts the outcome.

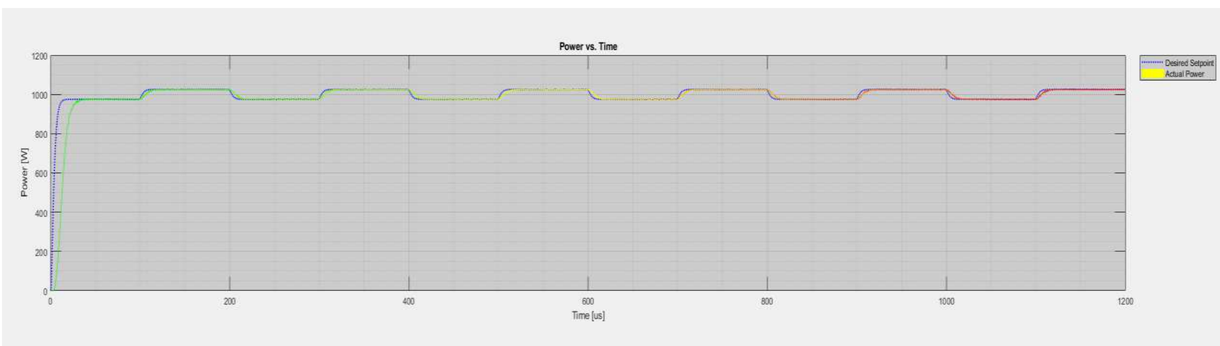

Figure 5.22: Baseline Performance of Existing AE Control System

This control system works effectively, and in figure 5.23, we can observe the transient reaction of one of the stages in greater detail (Desired Power Level is shown in blue and Delivered Power is in yellow).

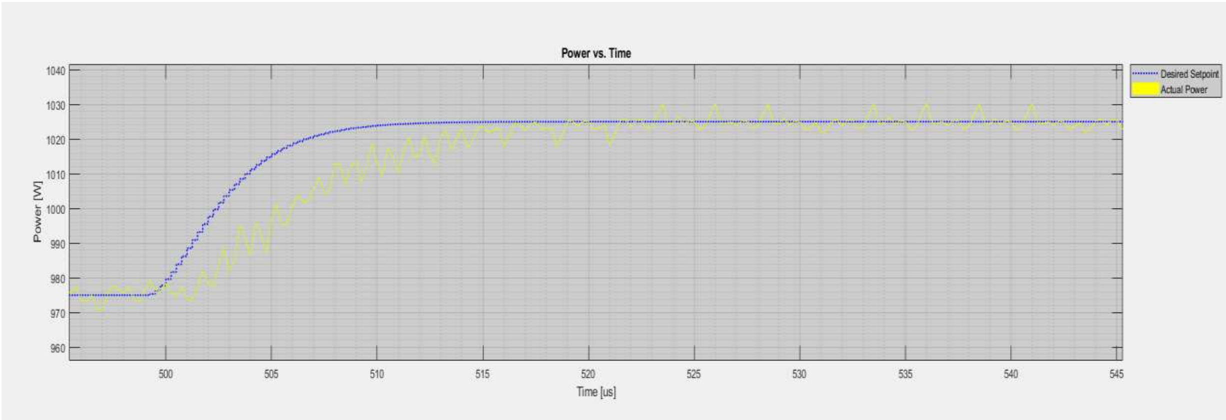

Figure 5.23: Baseline Transient Performance of Existing AE Control System

The reaction is smooth and stable, with no overshoot or oscillation, as can be seen (though there is some ripple due to noise etc.). It quickly settles to a steady state tracking error of zero. Figure 5.24 (overall response) and figure 5.25 (response) show the results of using the baseline AE control system on our simple linear model for comparison (zoomed in on one of the steps). We simplified the AE control model for the linear model to a linear PID controller without adaptation (to make the entire system linear), whereas the Digital Twin simulation employs the full AE control scheme with PID plus adaptation.

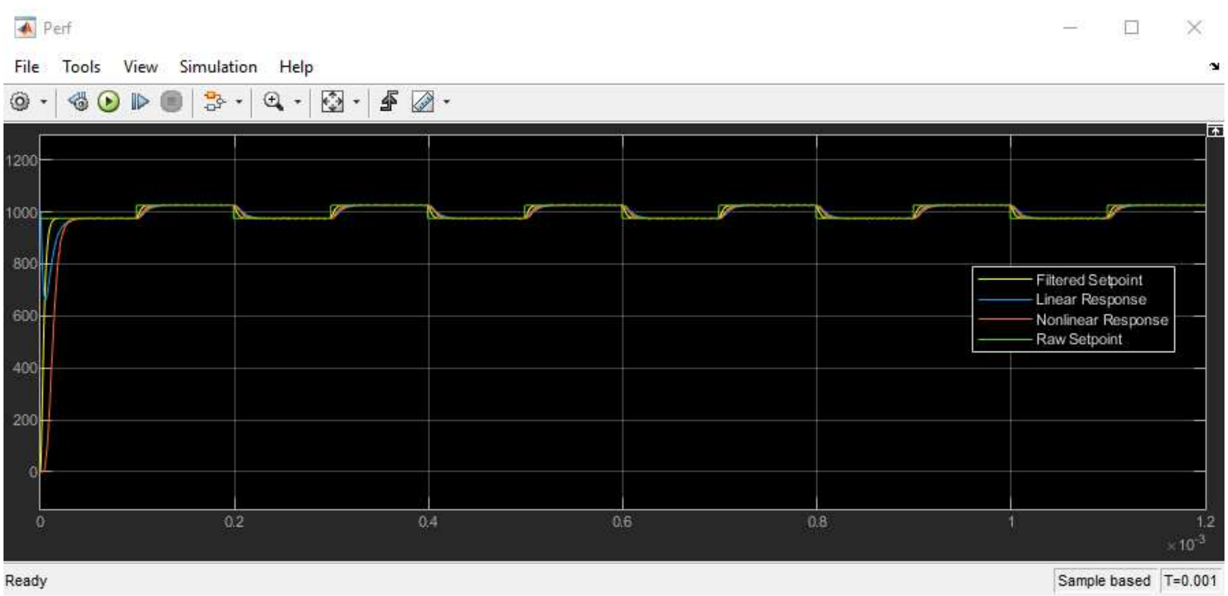

Figure 5.24: Baseline AE Control System on Linear Model

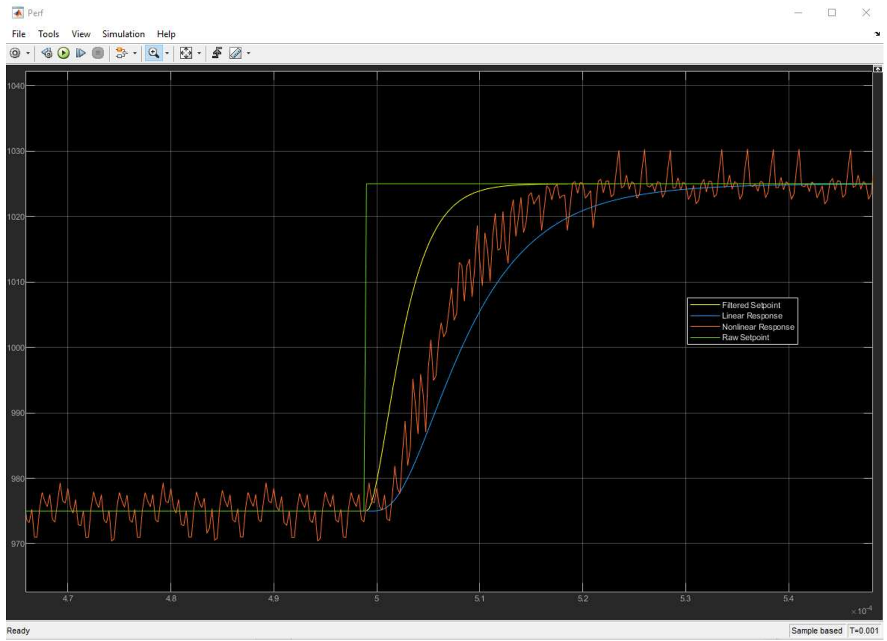

Figure 5.25: Baseline Transient Performance on Linear Model

Note that these linear model plots show 4 traces:

- Raw Setpoint (Green) - the input square wave for the desired power level
- Filtered Setpoint (Yellow) - the above setpoint after the second order filter
- Nonlinear Response (Red) - delivered power from nonlinear simulation
- Linear Response (Blue) - delivered power from linear simulation

For comparison, the first three traces (Raw Setpoint, Filtered Setpoint, and Nonlinear Response) were simply recorded from the Digital Twin nonlinear simulation. The linear model simulation is only responsible for the final trace (Linear Response). Since the information from the Digital Twin nonlinear simulation is already stored in this plot, we will only show these linear model simulation graphs from now on. The linear model provides a pretty realistic approximation of the nonlinear response, as seen in figure 5.25. It's also smooth and stable, with no overshoot or oscillation, and it settles to zero steady-state tracking error in a similar amount of time. However, it excludes higher-order effects (such as ripple due to noise).

5.2.5 μ-SYNTHESIS DESIGN AT A SETPOINT

Using the tools discussed previously, we constructed a -synthesis controller for a setpoint of 1,000W using the linear model (G0(z)). Figures 5.26 (total response) and 5.27 (individual reaction) demonstrate the results (zoomed in on one of the steps).

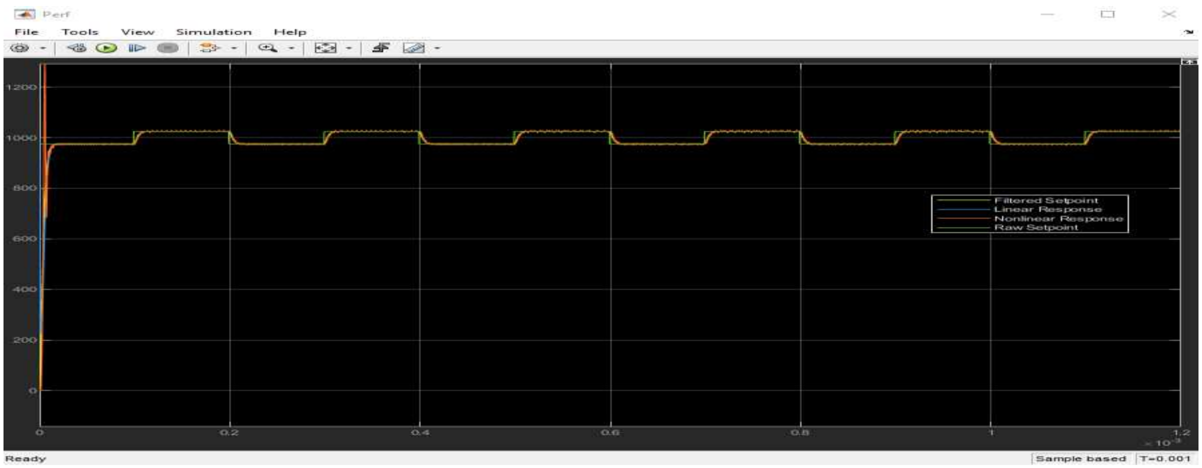

Figure 5.26: Response of μ-Synthesis Controller Designed for 1,000W Setpoint

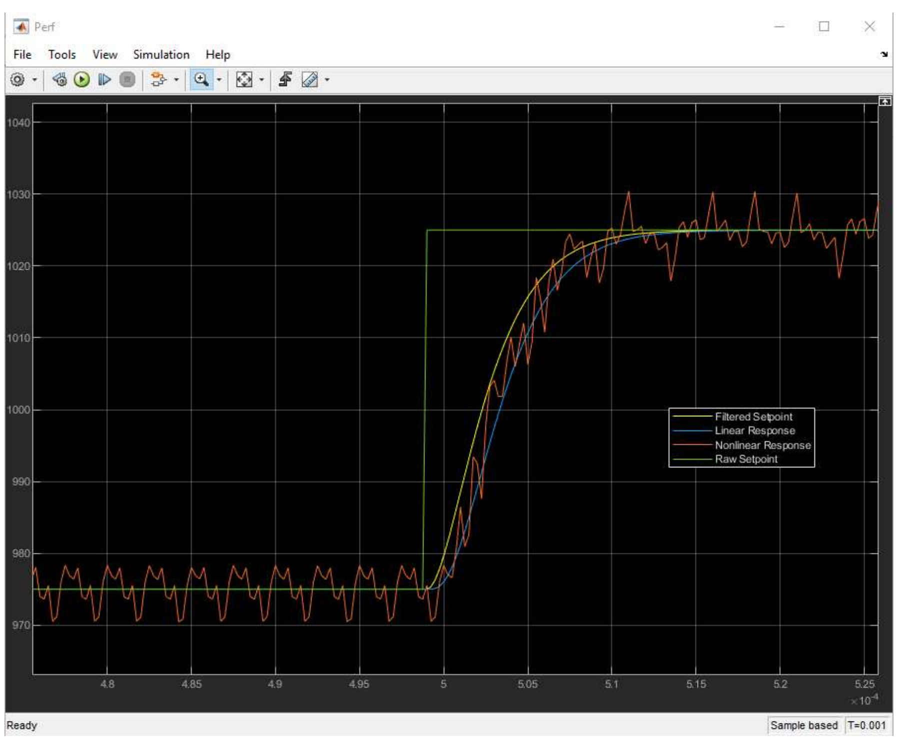

Figure 5.27: Response of μ-Synthesis Controller Designed for 1,000W Setpoint (zoomed in)

In comparison to the present AE control technique in this operating zone, the reaction is clearly improved (compare figures 5.25 and 5.27). It keeps the characteristics of a smooth and stable response, with no overshoot or oscillation, and it settles to zero steady state tracking error. However, the response time has improved dramatically. Indeed, whether comparing the nonlinear (red) or linear (blue) responses to the filtered setpoint (yellow), it is clear that the responses almost completely track it, with very little lag. The linear model is an excellent approximation of the nonlinear response, and both models demonstrate nearly perfect tracking of the filtered setpoint. As a result of our simulation results, the μ-synthesis controller (without adaptation) is capable of delivering great outcomes.

5.2.6 ROBUSTNESS ANALYSIS

In order to carry out robust performance analysis for the μ-synthesis controller we need a μ plot across frequency is shown in figure 5.28.

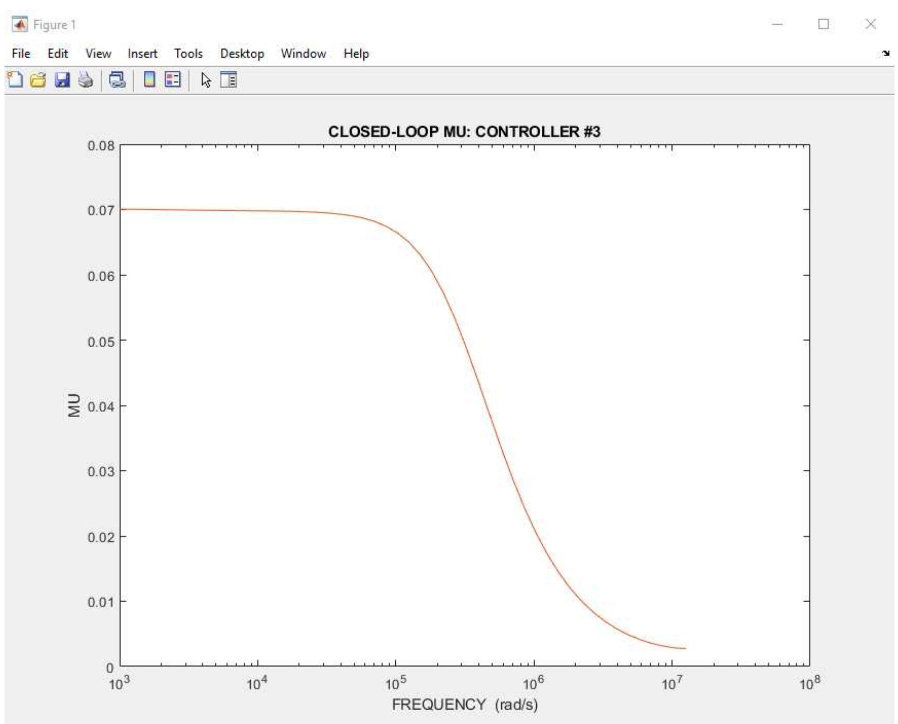

Figure 5.28: Robust Performance μ-Analysis Plot

We note that the peak μ value across frequency is given as:

$$\sup_{\Omega \in [0\pi]} \mu_{\tilde{R}}(\boldsymbol{M}(\boldsymbol{P},\boldsymbol{K})(e^{j\Omega})) = 0.07$$

As a result, stable performance at the specified levels is attained (peak value below unity). This means that even if worst-case perturbations at the stated levels occur, the system is guaranteed to stay stable and deliver the specified performance. This is when the power of robustness results comes into play. They provide stability and performance guarantees in the worst-case scenario.

5.2.7 μ-SYNTHESIS DESIGN ACROSS THE OPERATING ENVIRONMENT

We can simply repeat this method for a grid of setpoints over the operating environment now that we've defined our design process for a setpoint. Figures 5.29, 5.30, and 5.31 demonstrate designs with operating points of 1,000, 1,500, and 2,000 watts, respectively. Figure 5.32 shows a zoomed-in plot for the 2,000W level.

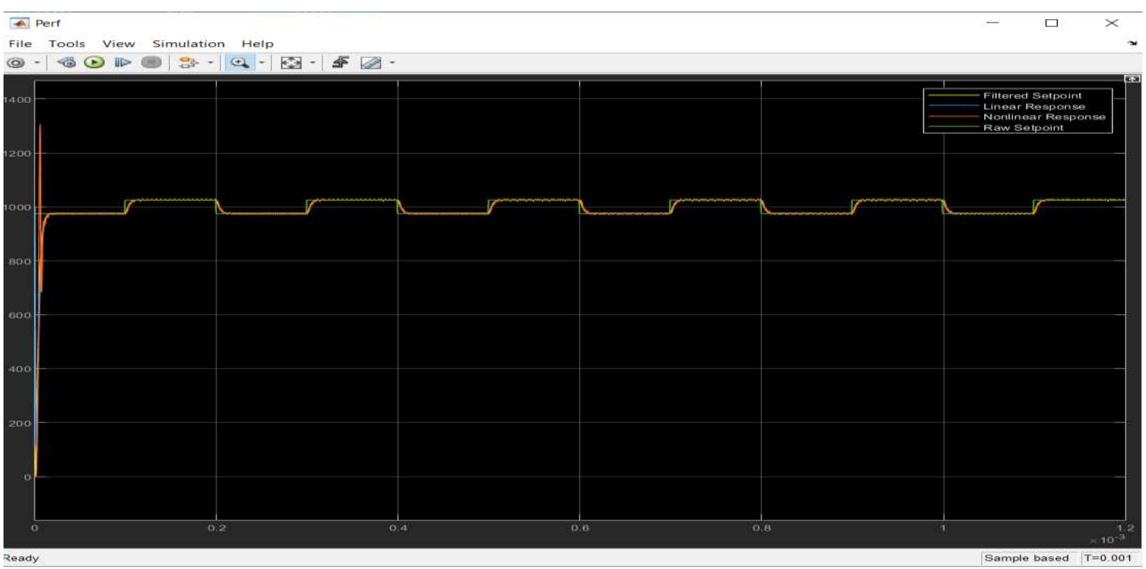

Figure 5.29: Performance of μ-Synthesis Controller Designed for 1,000W

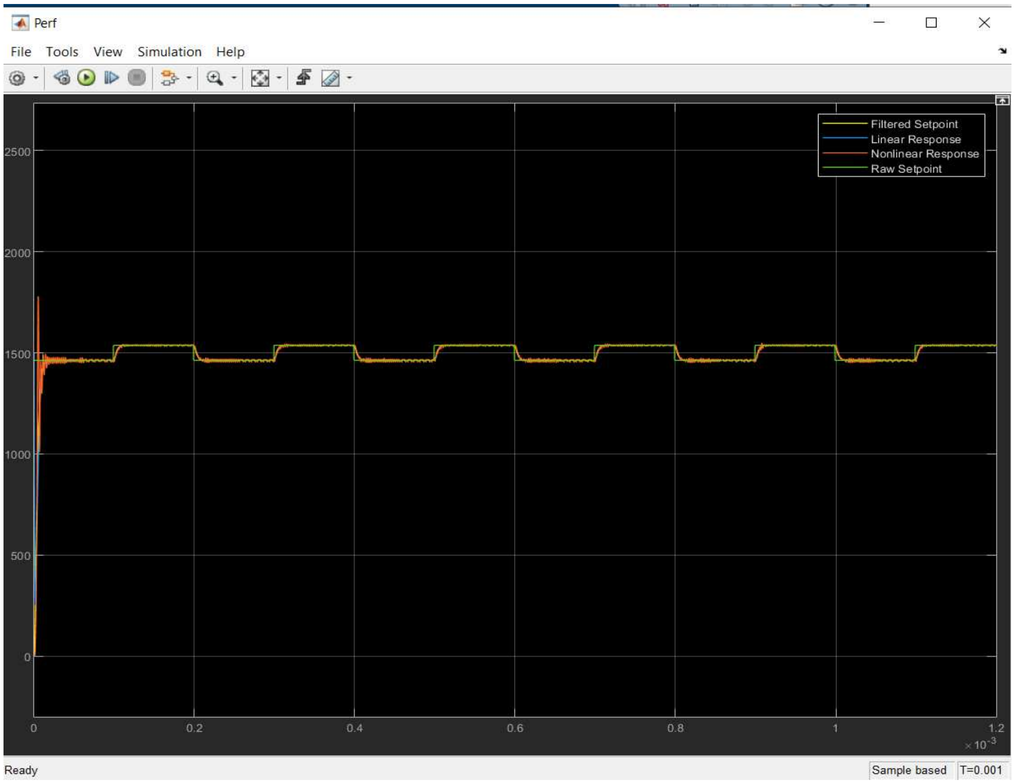

Figure 5.30: Performance of μ-Synthesis Controller Designed for 1,500W

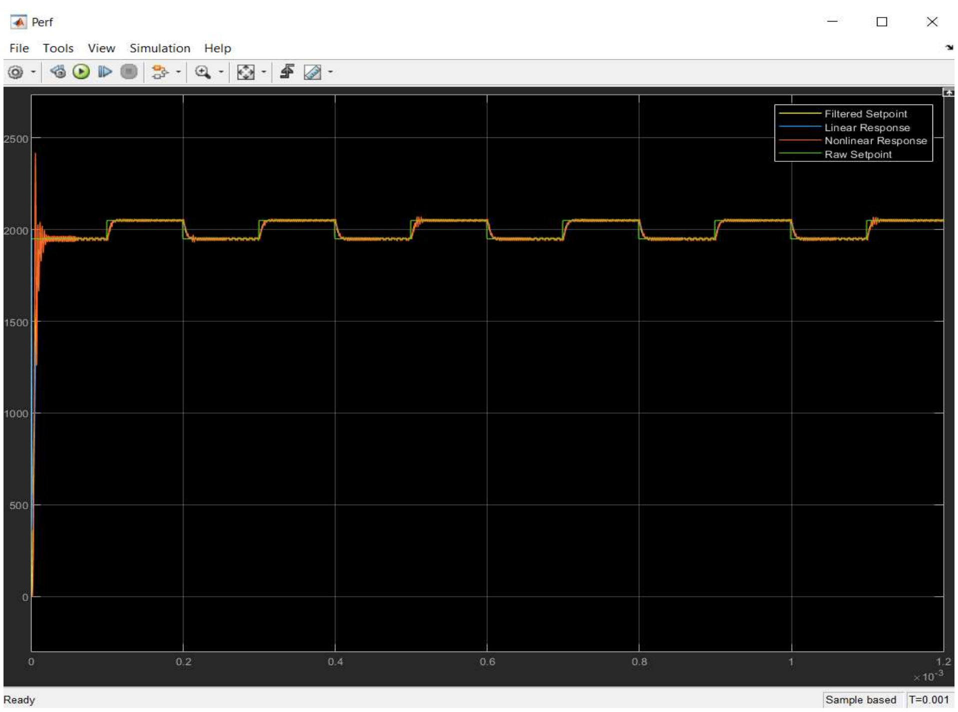

Figure 5.31: Performance of μ-Synthesis Controller Designed for 2,000W

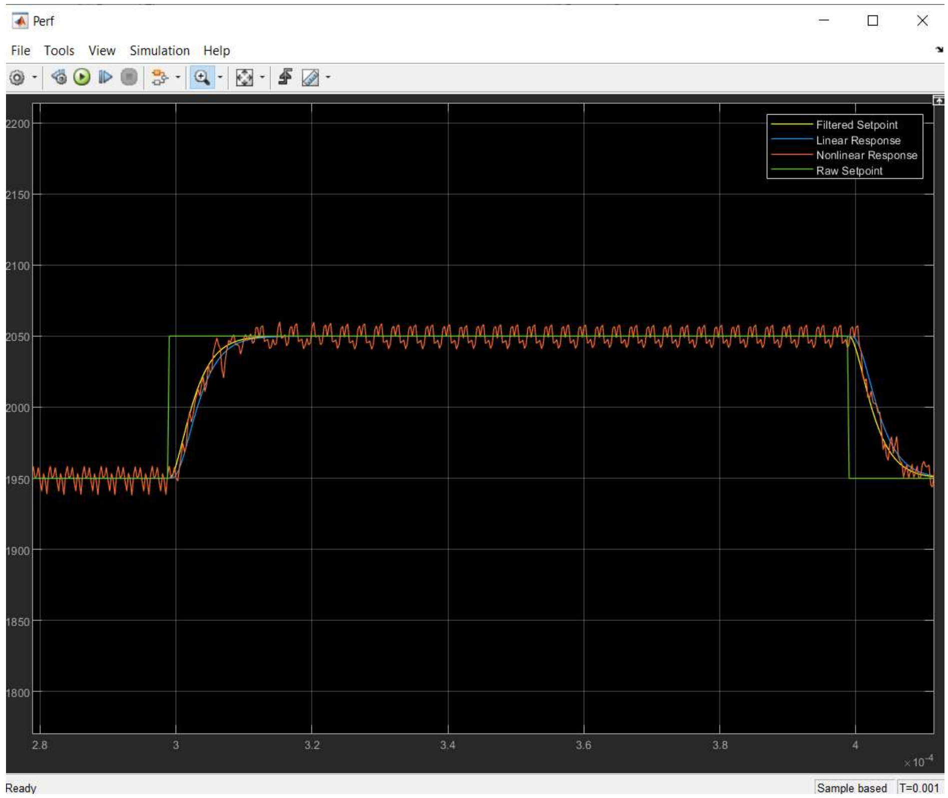

Figure 5.32: Performance of μ-Synthesis Controller Designed for 2,000W (zoomed in)

5.2.8 ROBUST CONTROLLER OPERATING RANGE

In this stage, we analyze how the controllers designed up to this point operate at setpoints other than the one for which they were designed. Our first investigations revealed that even minor deviations from the operating point resulted in poor performance and even instability. We discovered the problem after some inquiry, and the solution required us to review the system identification process. Although the linear models (G0(z)) in table 5.2 (section 5.1.2) were correct for modeling, they were not suitable for controller design. This is because no effort was taken to ensure that the models varied smoothly between operating points, which could have an adverse effect on the controller design. There was also some overfitting, which can lead to superfluous data. To solve this, we re-fit the data from table 5.2 using models that are limited to the following:

$$\frac{-C}{z-a}$$

where C>0 and a>0 (to avoid pathological sampling issues in the controller). This process resulted in the models shown in table 5.3.

Table 5.3: Refit for Linear Transfer Function Models across Operating Envelope

Setpoint (in W)	$\mathbf{G_0}(z)$
500	$\frac{-1310}{z - 0.41}$
1000	$\frac{-1405}{z - 0.53}$
1500	$\frac{-1500}{z - 0.65}$
2000	$\frac{-1500}{z - 0.65}$
2500	$\frac{-1575}{z - 0.62}$
3000	$\frac{-1690}{z - 0.65}$
3500	$\frac{-1780}{z - 0.65}$
4000	$\frac{-1860}{z - 0.65}$

The generated models (for G0(z)) were then utilized to re-design the -synthesis controllers throughout the operating envelope. The resulting performance at each setpoint was comparable to that shown in figures 5.29–5.32. The performance far from the design operating point, on the other hand, was much enhanced, as illustrated in figures 5.33–5.37. All of these data use the same controller, which was intended for a 1,500W operating point but was tested over a wide range of working points.

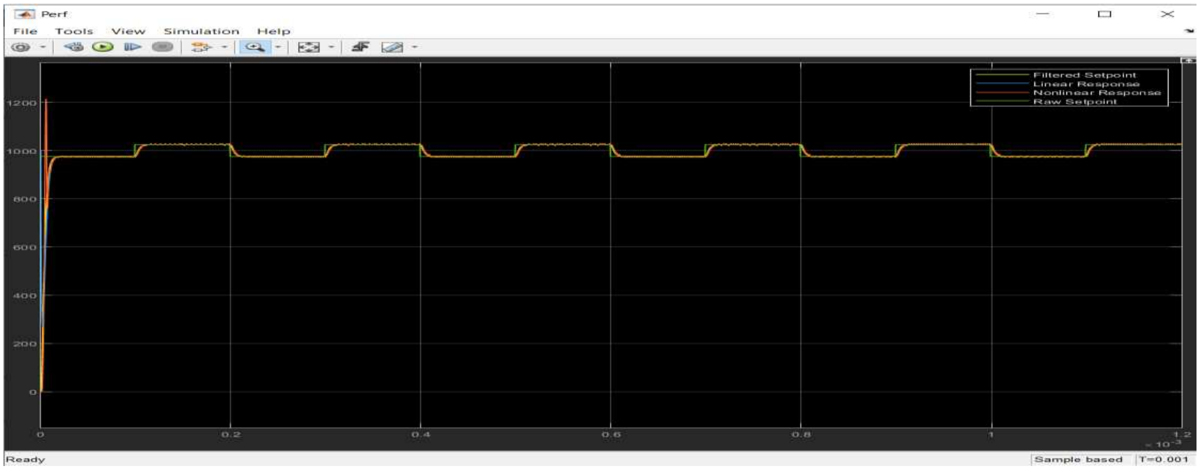

Figure 5.33: Robust Controller Designed at 1,500W and Tested at 1,000W

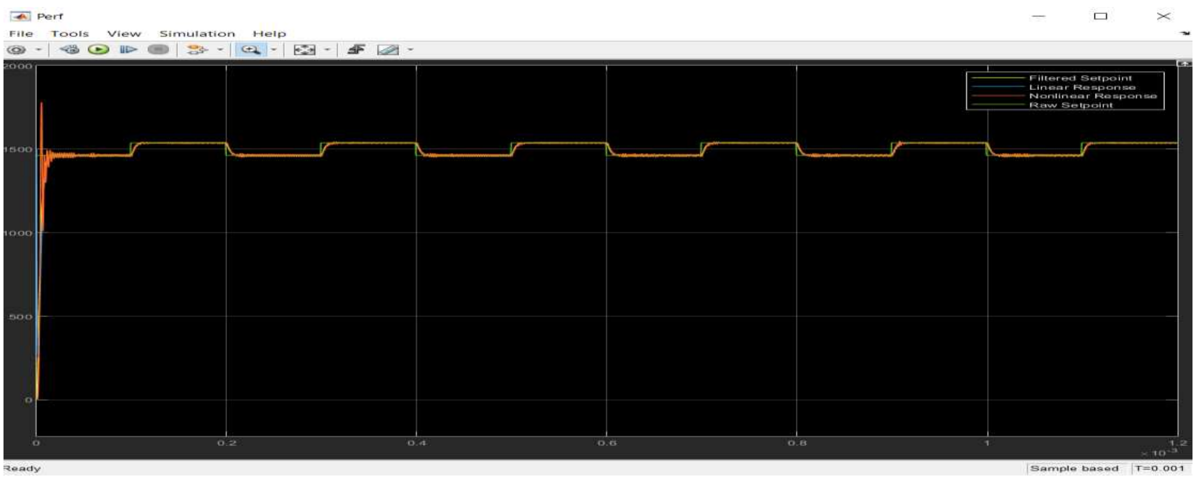

Figure 5.34: Robust Controller Designed at 1,500W and Tested at 1,500W

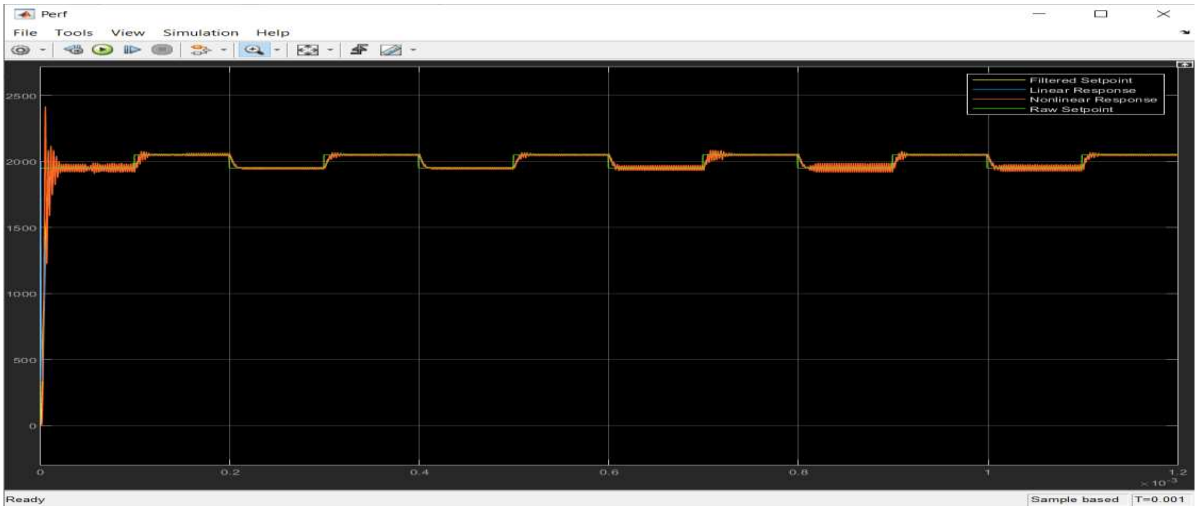

Figure 5.35: Robust Controller Designed at 1,500W and Tested at 2,000W

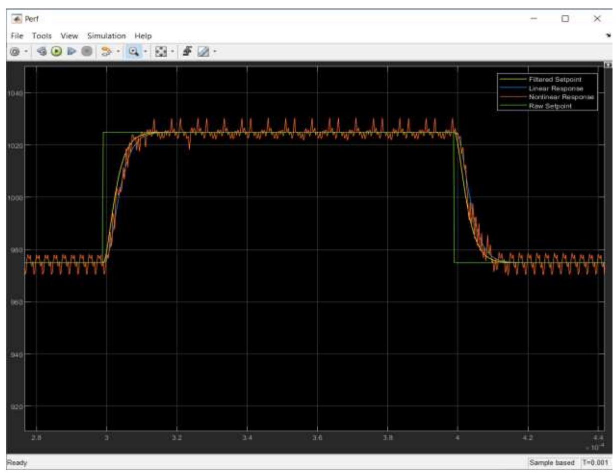

Figure 5.36: Robust Controller Designed at 1,500W and Tested at 1,000W (zoomed in)

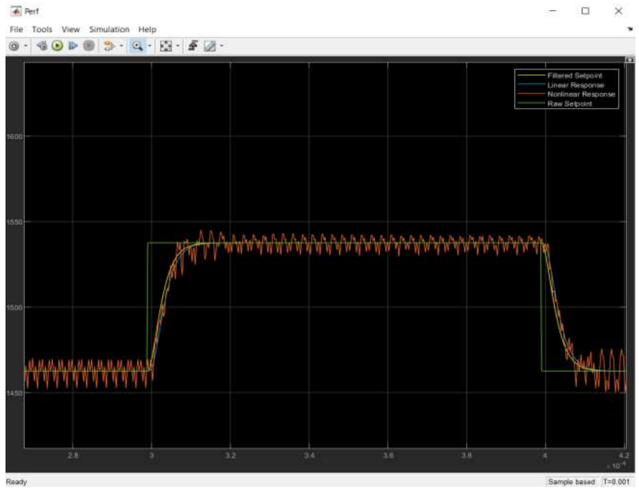

Figure 5.37: Robust Controller Designed at 1,500W and Tested at 1,500W (zoomed in)

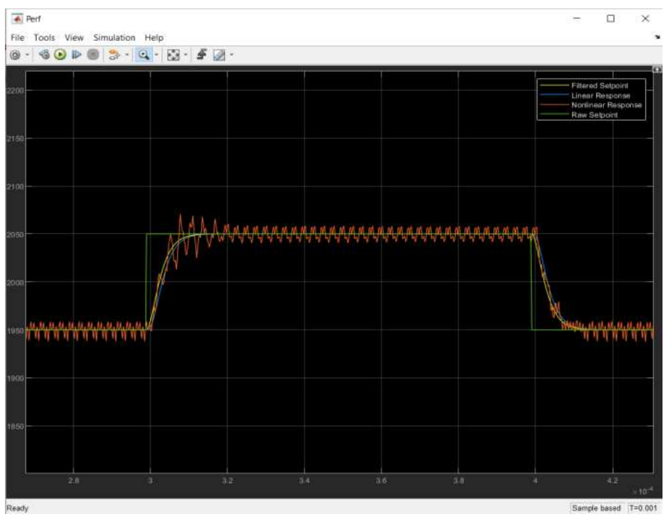

Figure 5.38: Robust Controller Designed at 1,500W and Tested at 2,000W (zoomed in)

The sturdy controller is shown to be capable of delivering excellent performance across a wide working range. It's worth noting that, while performance is excellent at operating locations other than the design point, it degrades significantly as you move away from it. Figures 5.37 and 5.38, for example, show the identical 1,500W controller being tested at 1,500W and 2,000W, respectively. It obviously outperforms the competition in terms of design. Compare figures 5.32

and 5.38 to further explain this point. Both of these images depict the performance at 2,000W. Figure 5.32, on the other hand, employs a 2,000W controller, and figure 5.38 uses a 1,500W controller. Moving away from the design point, slight decline in performance is noted.

These are great findings, and our testing has proven that the robust control design technique provided here can deliver high levels of robust performance throughout a wide working range, with a grid of such controllers covering the entire operating range. However, one should not expect that a single powerful controller will cover the entire needed working range. This is implausible for many systems, and it turns out to be the case for the AE system under consideration. When we operate a single controller outside of its design parameters, performance and even stability can suffer. Figure 5.39 shows an attempt to use the 1,500W controller design for a 2500 W setpoint. The response is visibly unstable and therefore, a grid of robust controllers, rather than a single one, is required to cover the specified operating range.

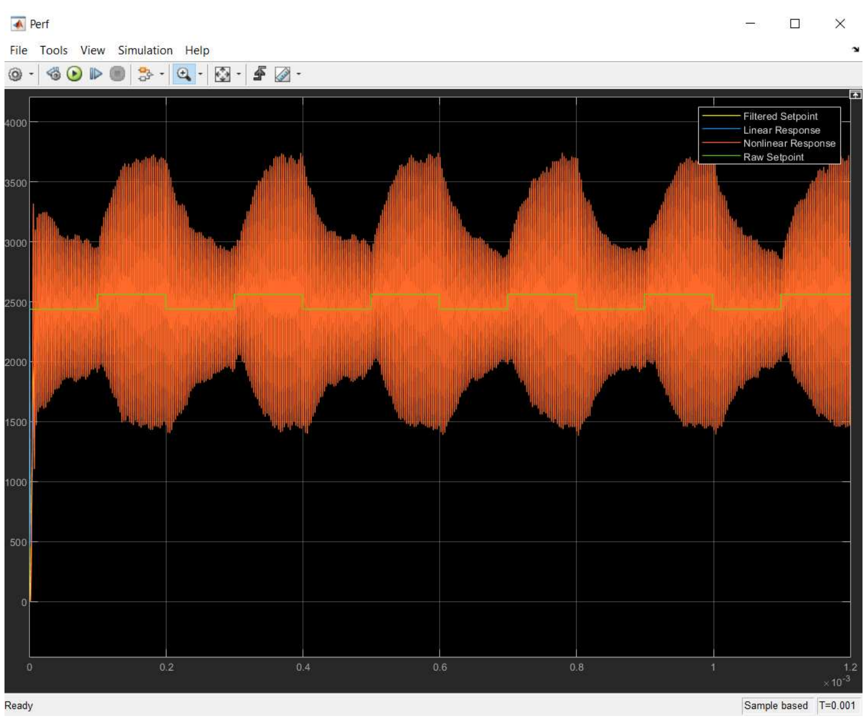

Figure 5.39: Unstable Response for Controller Designed at 1,500W and Tested at 2,500W

5.3 ROBUST LPV CONTROLLER

It's important to note that in order to link a grid of controllers, each of which is designed to work over a specific range, the handoff between controllers must be flawless (sometimes referred to as bumpless transfer). The following is how we propose to accomplish this. We use the measured power and compute its nearest two neighbors in terms of controller operating points at each sample time, k. We choose the A,B,C,D matrices for the two controllers in question and use an interpolation algorithm to construct the A,B,C,D matrices for our controller at that sample time, k. Every sample undergoes the same procedure. When we use the measured power as our parameter, it's clear that we're dealing with an LPV system.

The LPV controller implemented in Matlab/Simulink is shown below in figure 5.40. In addition to the usual error input, e (which is fed into the controller), it also receives the measured power, y, which constitutes the varying parameter for the LPV controller (i.e., determines the appropriate A, B, C, D matrices).

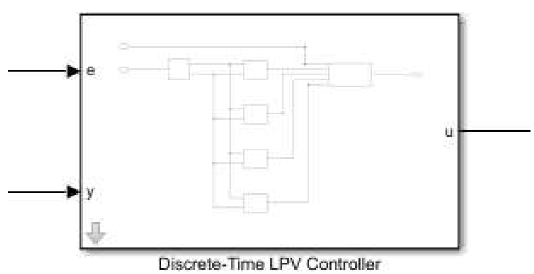

Figure 5.40: LPV Controller Implemented in Matlab/Simulink

Figure 5.41 shows the implementation structure inside the Discrete-Time LPV Controller block. The measured power, y, is used to first compute the grid interpolation point (k,f), and then to calculate the resulting A, B, C, D matrices for the LPV controller. Finally, as a Discrete-Time State-Space system, the LPV controller employs these A, B, C, D matrices to process the error signal, e, and generate the corresponding output.

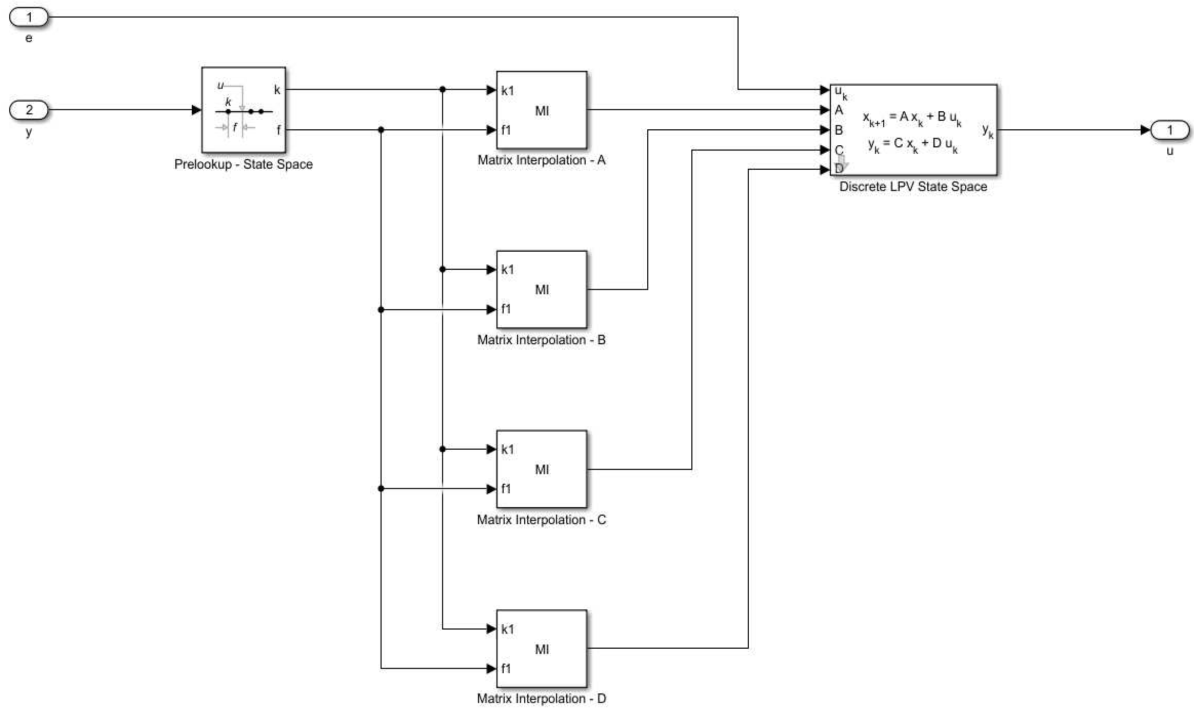

Figure 5.41: LPV Controller Implementation Structure

The overall LPV control system architecture implemented on the AE *Digital Twin* is shown below in figure 5.42. For comparison, recall the single robust controller implementation in figure 5.20, and it can be observed that the structure is largely similar, but with some enhancements, which we describe below.

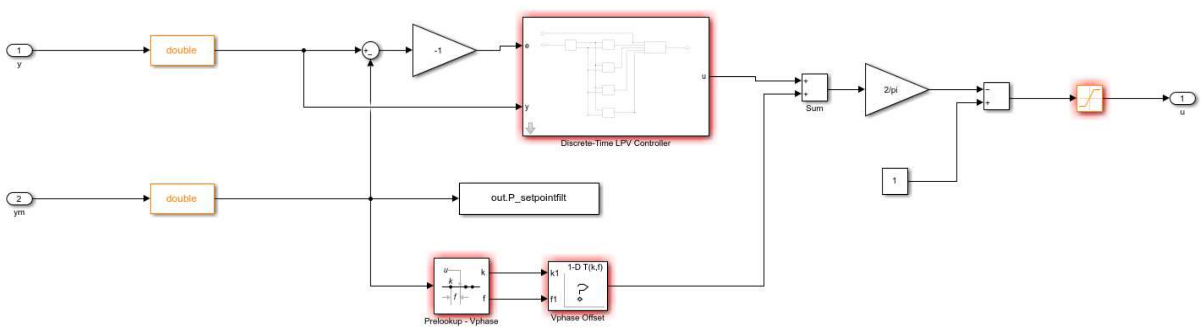

Figure 5.42: LPV Controller Implementation on AE System

The main change of course is the use of the *Discrete-Time LPV Controller* block described above (which replaces the *Controller-K-Linear* Discrete-Time State-Space block from earlier). As described earlier it receives both the error signal, e, and the measured power, y, which are utilized as described above. The next major change is the *Vphase Bias* block, which is shown below in figure 5.43.

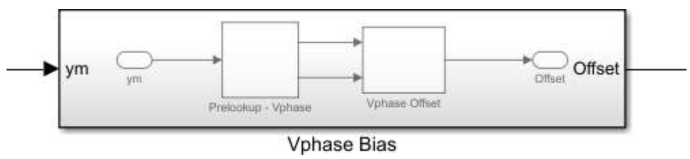

Figure 5.43: Vphase Bias Nonlinear Offset

This block computes the steady-state nonlinear offset needed for Vphase (instead of the simple constant offset shown in figure 5.20). It employs a similar interpolation strategy (grid point calculation followed by interpolation). However, because this is a static nonlinear function with no dynamics, it is not as complex as the previous LPV controller. It should be noted that the grid interpolation point in this case is not based on measured power, but rather on the power setpoint, ym. This essentially implies that Vphase Bias block is a feedforward term for the overall controller. It ensures that the transient response is excellent while responding rapidly to the setpoint changes.

5.4 RESULTS AND DISCUSSION

The AE Digital Twin Simulink model was used to test the design approaches described up to this point. We used test signals that covered the entire operating range of 0 – 4,500W. Figures 5.44 and 5.45 show typical simulation run examples. This controller design approach clearly produces solutions that run across the entire operating envelope, with actual power tracking the desired setpoint/trajectory at all times. It should be noted that the previous work described in the μ synthesis controllers delivered solutions at a grid of operating points but was incapable of moving between operating points or moving far from the current operating point.

Figure 5.44: Simulation run for LPV controller on AE *Digital Twin* with Multi-Level Input

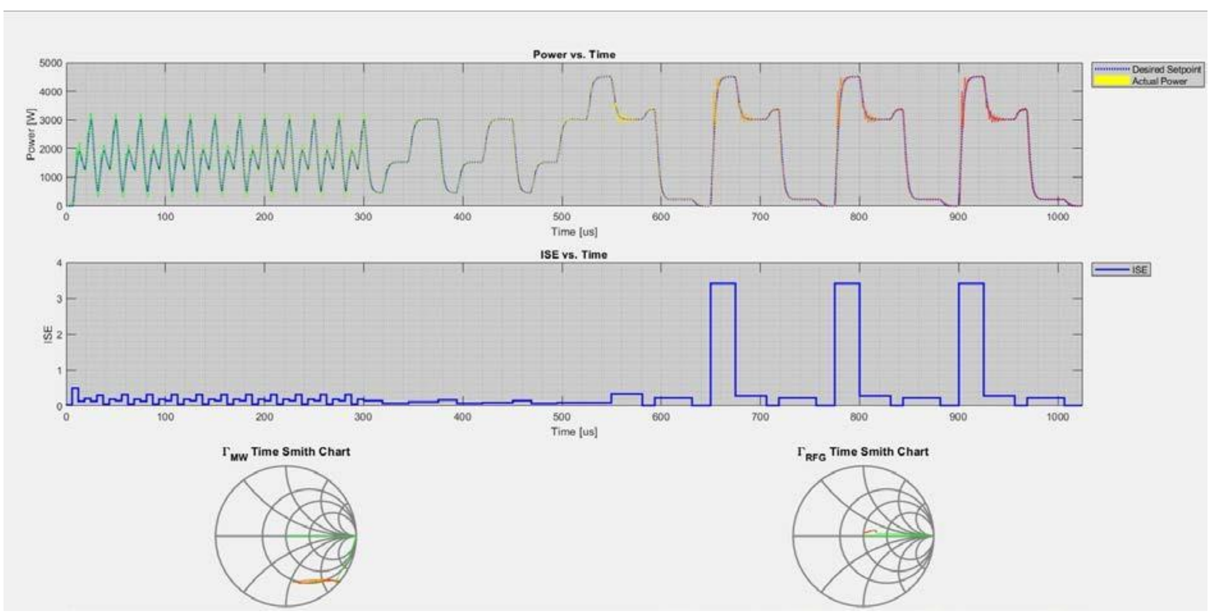

Figure 5.45: Simulation run for LPV controller on AE *Digital Twin* with Trajectory Following

The existing AE control scheme includes a second order filter on the reference input. It serves several functions, so we retain it in the control system here. Therefore, the desired-setpoint signal is not a pure square wave, but rather a filtered one.

The desired-setpoint signal in figure 5.44 clearly shows the switching between setpoints/levels. It is also evident that the control scheme handles level switching without issue. Over the entire operating range, the actual power tracks the changing setpoint accurately, validating the LPV controller design process.

In figure 5.45 the first part of the desired-setpoint signal varies in a continuous fashion, rather than just switching between setpoints. The actual power level follows this trajectory flawlessly, demonstrating that it can do more than just switch between and hold setpoints.

CHAPTER 6

CONCLUSION AND FUTURE WORK

This research commenced by successfully carrying out a literature survey and review of the current state of art of the plasma load dynamics and the RF plasma generator electronics in analytical and numerical terms. From the existing Simulink AE digital-twin model and data, a series of system identification procedures were carried out at each of the operating points across an operational range from 500 – 4500 W. A nonlinear static model fitting is achieved by interpolation of the data points. From dynamic response simulations, a linear dynamic model fitting is performed using Matlab System Identification tools. The actual simple-model of the system includes both the nonlinear steady-state gains and the linear dynamics, and it adequately captures both these effects. Therefore, it forms a nominal model for our plant. For uncertainty characterization, the variation in the outputs is characterized by applying the same inputs and is bounded. This forms the basis for robust controller design approaches. A design interconnection for the robust control analysis is developed based in the uncertainties, disturbance signals, signals to be penalized, and control signals present in the system. To ensure that the optimization problem does not treat all penalty signals equally, robust control designs such as μ-synthesis and H_∞-control choose weights that set the tradeoffs between performance and robustness. μ-synthesis design across the operating envelope provided a grid of robust controllers to cover the desired operating range. Applying it to the Digital Twin system provided by AE depicted a significant improvement over the existing control scheme. To be able to handle arbitrary waveforms as part of multi-level setpoint power inputs, a single Linear Parameter Varying (LPV) controller is developed for the entire operating range.

A bumpless transfer method is necessary for a seamless handoff between these controllers and is implemented in Matlab/Simulink as an LPV controller block. This control scheme performed well over the operating range on the AE Digital Twin with multi-level input as well as trajectory following. We created a process for designing robust LPV controllers that makes use of advanced tools from robust control theory to improve the robust performance of the resulting closed-loop systems. Based on the structured singular value, this robust performance comes with strict stability and performance guarantees. These tools enable the creation of a single robust LPV controller that covers the desired operating range.

For future work, it is also desirable to implement some form of adaptation. This could be to compensate for unknown a-priori plant dynamics or variations over time. A variety of adaptation schemes involves the use of L1 adaptive control. These schemes can be used to improve the capabilities of existing control approaches in a variety of ways, one of which is to use a multiplicative architecture. We can also implement the advanced control systems in System on Chip (SoC) devices by coding the control systems in VHDL and C++.

BIBLIOGRAPHY

[1] Gideon van Zyle, Method and apparatus for modifying interactions between an electrical generator and a nonlinear load, August 4 2009, US Patent 7,570,028 B2

[2] Gideon van Zyle, System and method for control of high efficiency generator source impedance, August 22 2017, US Patent 9,741,544 B2

[3] Young, P., Sankaran, A. (2021). *Advanced Energy's New Distributed Control System Architecture – The Core - Part I: Single Device Control System* Stage 1 Report

[4] Young, P., Sankaran, A. (2021). *Advanced Energy's New Distributed Control System Architecture –The Core - Part I: Single Device Control System* Stage 2 Report

[5] Young, P., Sankaran, A. (2021). *Advanced Energy's New Distributed Control System Architecture – The Core - Part I: Single Device Control System* Stage 3 Report

[6] Sigurd Skogestad and Ian Postlethwaite, *Multivariable Feedback Control Analysis and Design,* John Wiley and Sons, Second Edition, pp. 293-351, 2001

[7] MathWorks, *Simulink User Guide,* r2020b ed., 2020.