2019

The Incorruptible Integrator: A Streamlined Approach to IMC-PID Controller Tuning

Sam Wisotzki
University of Vermont

The Incorruptible Integrator: A Streamlined Approach to IMC-PID Controller Tuning

A Thesis Presented

by

Sam Wisotzki

to

The Faculty of the Graduate College

of

The University of Vermont

In Partial Fullfillment of the Requirements
for the Degree of Master of Science
Specializing in Electrical Engineering

August, 2019

Defense Date: April 25, 2019
Thesis Examination Committee:

Hamid R. Ossareh, Ph.D., Advisor
Alan Ling, Ph.D., Chairperson
Luis A. Duffaut Espinosa, Ph.D.
Cynthia J. Forehand, Ph.D., Dean of the Graduate College

Abstract

In automakers' never-ending quest to reduce emissions and improve performance, the turbocharger represents a major step in advancing these goals. By repurposing waste exhaust and compressing the air intake, they are able to increase overall power. One critical control loop in the turbocharger is control of boost pressure via the wastegate. This is a highly nonlinear process and experimental data has shown that a gain-scheduled PID (proportional integral derivative) controller developed with IMC (internal model control) tuning methodology is an effective means to control boost pressure. Motivated by this successful implementation of IMC-PID tuning in the automotive world, this work hopes to extend and analyze that framework.

Traditionally, the success of an IMC controller depends on the accuracy of the plant model. This research challenges this view and investigates using IMC with a gain-integrator-delay (GID) model identified at a critical frequency, regardless of the actual plant. The GID model is useful because of its simplicity to characterize and its ability to be translated to the ubiquitous PID controller easily. Three design techniques are developed: (1) design for post-hoc tuning, (2) design for closed loop bandwidth, and (3) design for phase margin. In addition, these techniques are investigated via a Monte Carlo simulation to determine efficacy for when there exists plant/model mismatch. Finally, the three techniques are applied to control the speed of an inertia disk on the Quanser QUBE$^{\text{TM}}$ Servo 2 device.

Acknowledgements

Most importantly, I'd like to thank all the members of my examination committee, Dr. Alan Ling, Dr. Luis A. Duffaut Espinosa, and especially my advisor, Dr. Hamid Ossareh. Your patience and understanding was greatly appreciated throughout my three years as your student. Thank you all for the opportunity to study at the University of Vermont, it truly was a pleasure.

I'd also like to thank my wonderful girlfriend, Taline, and her partner in crime, Ellie, for their love, support and countless fetch breaks throughout this entire process.

Table of Contents

List of Figures

1 Introduction

1.1 Motivation

Today's automakers are following a trend of downsizing their engines to create more and more fuel efficient cars. Increased fuel efficiency comes at the expense of a reduction of engine output power. One of the key components to mitigate these losses is the turbocharger. These devices are used to recapture and re-purpose the engine's exhaust thereby increasing overall fuel efficiency (and reducing CO_2 emissions) without giving up the significant engine output power associated with a more fuel-efficient car. These devices utilize exhaust gasses to spin a turbine which powers a compressor which then compresses the air intake, and this forces more air (and fuel) into the engine, creating more power.

The two major actuators of this system are the throttle, which controls air intake to the engine, and the wastegate, which controls the boost pressure (or the pressure at the throttle inlet). To output a desired power, the air intake to the engine (throttle actuation) must be delicately balanced with the wastegate actuation. The original genesis of the problem researched in this thesis is the control of boost pressure via the wastegate. The industry standard control topology is a PID controller with scheduled gains [1].

Wastegate control is difficult due to many nonlinearities present in the engine [2, 3], as well as the fact that optimal gain scheduling is a labor-intensive process for the calibrator, as modifying the three PID tuning parameters to their desired values (at multiple operating regions) is nontrivial.

With that in mind, this research presents a novel design process to make it much easier to obtain these PID gains. The standard IMC (internal model control) framework naturally

allows for single-parameter tuning, however, instead of a high-fidelity plant model, a much simpler gain-integrator-delay (GID) model is used in this work. The novelty lies not in the use of this particular model, but in how its parameters are characterized. Was it possible the previous validated success of this design process was a fluke, or can this technique perform well in general, for plants not similar to integrators? In addition, is it possible to design for classical control system metrics, like bandwidth or phase margin? This research hopes to investigate and analyze these questions.

1.2 Prior Work: IMC-PID tuning

PID control has its origins in the 1930s [4], and the number of papers about PID control is very quickly approaching the number of stars in the known universe. These ubiquitous controllers are extremely effective for many applications. A summary of major findings can be found in [5–7].

Once the control engineer decides to use a PID-based topology, the next biggest step is in deciding how to tune the operating parameters, or gains, of the controller. The most famous of these tuning methods is the Ziegler-Nichols method, which results in aggressive gain and overshoot but great disturbance rejection [8]. Other popular methods are the Cohen-Coon method [9], the Tyreus-Luyben method [10], and IMC-based tuning [11]. These methods are compared against each other in [12].

This work utilizes the IMC-based method of obtaining PID gains, introduced in [11]. IMC is particularly useful for gain-scheduling due to its tuning parameter(s) trading off speed with robustness. A background of IMC theory can be found in [13–16] and has been widely used in practice, most notably in the process industry [17–24].

1.3 Prior Work: Boost Pressure Control

Boost pressure control via the wastegate actuator is a much-studied topic in the literature [25–32]. IMC to get PID gains is a popular solution path for this problem, and most of the work done in this space is with respect to applying clever new tools to create better IMC models. For example, in [29], parameter estimation is used in conjunction with the certainty equivalence principle to simultaneously identify the plant and inverse plant dynamics (a more accurate means of identifying the IMC framework). The LTI IMC tools are expanded to nonlinear systems in [31] through developing a novel quasi-LPV (linear parameter varying) model. Further, a complex mean value model for the wastegate is developed in [32] using three submodels. In general, these approaches show the use of high-fidelity models produces predictably good results. This thesis is concerned with the use of a notably non high-fidelity model, the GID.

With that in mind, much IMC-based work has indeed been done with respect to model reduction [33–36], the use of lower-order models [37–39], and most relevantly to this work, the control of integrating processes [40–42].

The research proposed in this work is most similar to the work in [43], but this research's approach does not require knowledge of the process dynamics (but does require the physical process), is not based on an approximation (and is exact around a designed frequency), and can be quickly implemented in a calibration setting. In addition, the novelty here is not the use of a GID model (or in the literature sometimes referred to a IPD - integrating plus dead time), rather, it is how the GID's parameters are characterized (the interim step) and how the eventual PID controller is tuned (the final step). For a full description of the

process, see Section 3.

1.4 Summary of Contributions and Outline of Thesis

This thesis will do (3) major things. After providing a background of PID and IMC methods in Section 2, the novel approach to designing the GID model's parameters will be developed, in Section 3. In particular, the GID model is matched to the plant around a frequency, ω_0, with the option to design for: post-hoc tuning, closed-loop bandwidth, or phase margin. The first design approach was developed in order to give the system calibrator flexibility in shaping the response of the system. The latter two approaches were developed to guarantee useful metrics for the system. The innovation of this research lies here: the unique design of the IMC filter to enact these design approaches in conjunction with system characterization around an ω_0 is what is novel.

Next, these three design approaches are investigated via Monte Carlo analysis when there exists plant/model mismatch, to see if this general technique can be applied to plants not similar to the wastegate, in Section 4. Third, the three approaches are applied to a physical system, the QUBE$^{\text{TM}}$ servo-2, in Section 5. The goal here is to experimentally validate the Monte Carlo analysis, confirming that the design techniques can be applied in general.

Finally, future work and a summary are presented in Section 6.

2 Review of IMC-PID methods

2.1 Proportional-Integrative-Derivative (PID) Control

Consider the following standard unity-feedback system:

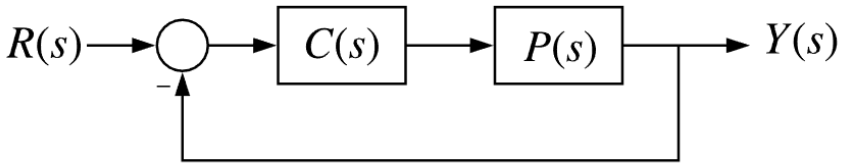

Figure 1: LTI system with feedback, where $C(s)$ is the designed controller, $P(s)$ is the plant or process, $R(s)$ is the reference signal, and $Y(s)$ is the system output. This will be referred to as the *classical* feedback structure.

For this system, $R(s)$ is considered the reference or input, and $Y(s)$ is the output. $P(s)$ is known as the plant or process; the thing whose behavior will be controlled by $C(s)$, the controller.

The typical goal of a control system is to design a controller in order to manage or modify the behavior of the plant so that the reference signal is tracked. This work will utilize the PID framework, shown in Figure 2.

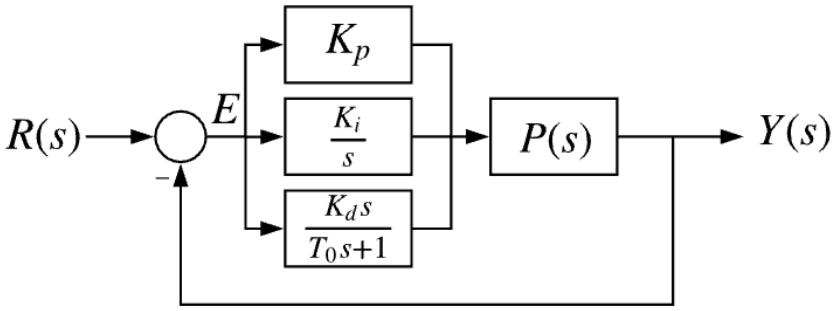

Figure 2: PID controlled system. Note the derivative term has an augmented filter, $\frac{1}{T_0 s+1}$. This is to combat the issues associated with sensor noise.

This figure introduces a new signal, $E = R - Y$. This is known as the error signal, and each of these gains can be thought about as applying a different multiplying factor of the error to the plant. The proportional gain applies a linear scaling factor to the error signal. The integral term, $\frac{K_i}{s}$ increases not only with the error signal, but also as a function of the time for which the error has persisted. If the goal is to track a step reference, then at least one integrator ($\frac{K_i}{s}$) is required in the open loop transfer function, $P(s)C(s)$.

Finally, there's the derivative action, $K_d s$, which is oftentimes accompanied by the first-order filter, $\frac{1}{T_0 s+1}$. This is because the D controller takes into consideration the rate of change of the error, and is a limiting factor on overshoot. Without this filter (and ample sensor noise, i.e. the feedback signal is not accurate or smooth), the resulting control action will be noisy and undesirable. For a more thorough background on general PID theory, see Ch. 4.3 of [44].

Note that there exist countless tweaks and alterations to this classical PID structure, such as adding a pre-filter to modify the system's bandwidth [45], adding feedforward action to the full structure [46], or adding logic to prevent integrator wind-up caused by actuator saturation [47]. In order to allow for the most general use of the techniques proposed in this work, I will assume the classical structure as described in Figures 1, 2.

2.2 Internal Model Control (IMC)

IMC theory presents a different approach to controller design - instead of designing a controller to a plant; a plant model is directly incorporated into the controller and feedback loop. This general structure is visualized in Figure 3. Within this structure, two new blocks are introduced: $\hat{P}(s)$, the plant model, and $Q(s)$, the IMC design parameter.

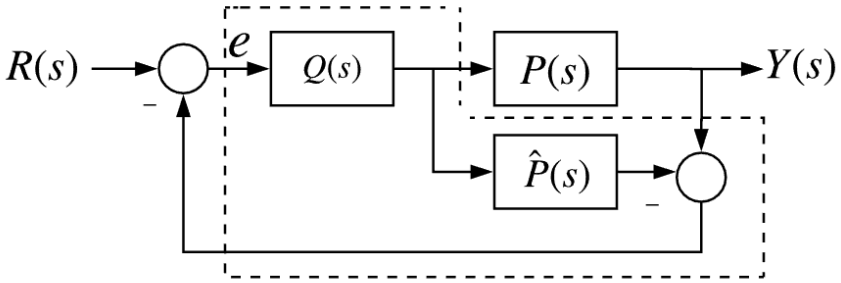

Figure 3: IMC system for plant $P(s)$ and plant model $\hat{P}(s)$. Term $Q(s)$ contains the model inversion term and a design filter term.

Notice how the error signal is now a function of the difference between the plant and the plant model; this is why model fidelity is so important; the conventional (and indeed literature-proven) wisdom is that the better the model, the better the controller with IMC. This will not be the case for the GID-based design methods in this work, as we are assuming a non-accurate model from the get-go. Moreover, part of the contribution of this work is to show that a good model of the process is not needed to achieve good performance.

Also note that the standard IMC structure can be used as a feedback control system in of itself, but for this work (and many others), IMC is a means to an end: the eventual goal is a PID controller, and through rearranging of blocks, the classical control structure can be obtained, in Figure 1, where:

$$C(s) = \frac{Q(s)}{1 - Q(s)\hat{P}(s)}. \tag{1}$$

The goal of IMC is to select $Q(s)$ to minimize the H_2 norm of the tracking error, $\|e\|_2$ [13]. To accomplish this, decompose $\hat{P}(s)$ into:

$$\hat{P}(s) = \hat{P}_A(s)\hat{P}_M(s),$$

where $\hat{P}_A(s)$ is an all-pass transfer function that contains all delays and non-minimum phase zeros of $\hat{P}(s)$, and $\hat{P}_M$ contains the rest of the (minimum-phase) dynamics. It can be shown that [13] the following is the optimal choice of $Q(s)$ that minimizes $\|e\|_2$ (for a unit step reference):

$$Q(s) = \hat{P}_M^{-1}(s).$$

In practice, the above $Q(s)$ cannot be implemented because it is typically improper, and it does not have proper high frequency roll-off properties required for robustness. For these reasons, $Q(s)$ is typically augmented with a low-pass filter, $F(s)$:

$$Q(s) = \hat{P}_M^{-1}(s)F(s). \tag{2}$$

The filter $F(s)$ is typically chosen to be strictly proper with a unity dc-gain. Note that with this filter, the IMC controller is not H_2-optimal. However, if F has sufficiently large bandwidth, then the filter's effects are negligible and the controller will be *close* to optimal.

With $Q(s)$ in (2) and $C(s)$ in (1), the closed-loop transfer function from R to Y is given by:

$$T_{cl}(s) := \frac{P(s)C(s)}{1 + P(s)C(s)} = \frac{P\hat{P}_M^{-1}F}{1 + F(P\hat{P}_M^{-1} - \hat{P}_A)}. \tag{3}$$

Assuming the plant model is accurate (i.e., $P(s) = \hat{P}(s)$), the closed-loop transfer function

becomes:

$$T_{cl}(s) = \hat{P}_A(s)F(s), \tag{4}$$

which is directly proportional to $F(s)$. Thus, the parameter(s) of $F(s)$ are typically tuned to trade-off performance (i.e. bandwidth) and robustness (i.e. margins).

2.3 System Characterization

Before introducing the control scheme, it is important to describe the major assumption required for its use. First and foremost, because the technique is extracting information from the real-life plant, access to the plant is required. Next a system characterization (sometimes known as system *identification*) technique needs to be performed, of which there are numerous time-domain and frequency-domain methods [5, 48, 49]. More modern techniques use bleeding-edge technology like neural networks [50], and genetic algorithms [51].

One such frequency-domain based, system-calibrator-friendly method of system characterization (which has been utilized by the previous experimental implementation of these design methods) is the relay-feedback experiment. Here, a bounded square-wave is introduced in place of the controller and pertinent information can be extracted from the plant output (another bounded waveform). The output waveform's amplitude and frequency are determined by the system dynamics [52]. For the analysis sections of this thesis, the data from system characterization is presumed to be known, and in the experimental section, a simpler approach of inputing a sinusoid and measuring the output sinusoid's gain and phase is used.

3 GID-based IMC-PID design

In short, the user has (3) general design options, (1) design for post-hoc tuning, (2) design for closed-loop bandwidth, or (3) design for phase margin. Depending on this choice, the two design parameters, ω_0, the frequency at which to match the GID model to the actual plant, and γ, the IMC tuning parameter contained in $Q(s)$ (to be introduced) are either chosen directly or indirectly for the user.

3.1 General Methods

To begin, the gain-integrator-delay (GID) structure is assumed for the plant model:

$$\hat{P}(s) = \frac{K_{GID}}{s} e^{-T_d s}, \tag{5}$$

and model parameters K_{GID} and T_d are chosen so that $P(j\omega_0) = \hat{P}(j\omega_0)$, where ω_0 is the frequency at which the plant is matched to the plant model. In other words,

$$K_{GID} = \omega_0 |P(j\omega_0)|, \tag{6}$$

$$T_d = \frac{1}{\omega_0} \left(-\angle P(j\omega_0) - \frac{\pi}{2} \right). \tag{7}$$

This specific plant model is useful for two major reasons: first, in conjunction with the filter, $F(s)$, this model results in a 2nd order controller, $C(s)$ (in the sense of the classical feedback structure), which allows it to be easily translated to existing PID architecture. Secondly, its two degrees of freedom are easy to determine in the sense of system characterization.

Next, the following structure for $Q(s)$ is developed:

$$Q(s) = \frac{s}{K_{GID}} F(s) \tag{8}$$

Note the omission of the delay in $\hat{P}(s)$, as delays are not invertible for causal systems analysis. Using this filter, the final IMC controller can be derived:

$$C(s) = \frac{Q(s)}{1 - Q(s)\hat{P}(s)} = \frac{1}{K_{GID}} \left(\frac{sF(s)}{1 - F(s)e^{-T_d s}} \right). \tag{9}$$

Note the delay in $C(s)$ here comes from the direct use of $\hat{P}(s)$.

The filter, $F(s)$ now needs to be designed to convert this controller to a PID. This provides two major design constraints: (1) a pole must be introduced at $s = 0$ while cancelling the existing zero at $s = 0$, and (2) the infinite-dimension exponential needs to be eliminated.

To ensure that the controller indeed has a pole at $s = 0$, we must select $F(s)$ such that the denominator $\left(1 - F(s)e^{-T_d s}\right)$ has two roots at $s = 0$, one to cancel the zero and the other to introduce an integrator. To ensure a repeated root at $s = 0$, the denominator must satisfy $\frac{d}{ds}(1 - F(s)e^{-T_d s})|_{s=0} = 0$, which can be simplified to $F'(0) - T_d F(0) = 0$, where $'$ denotes the derivative with respect to s. F must satisfy $F(0) = 1$ or unity dc gain (from Section 2.2), so the above condition becomes $F'(0) = T_d$. A first order, strictly proper filter cannot satisfy these conditions. Thus, the following novel second-order filter is selected:

$$F(s) = \frac{\beta s + 1}{(\alpha s + 1)^2}, \tag{10}$$

where α is a free parameter, and β must be selected such that $\beta = 2\alpha + T_d$ to create 2 poles at $s = 0$.

To eliminate the exponential, the first-order Padé approximant is used:

$$e^{-T_d s} \approx \frac{\frac{-T_d}{2}s + 1}{\frac{T_d}{2}s + 1}. \tag{11}$$

Combining these two design constraints yields the following controller (in terms of the classical structure):

$$C(s) = \frac{1}{K_{GID}} \left(\frac{(\beta s + 1)(\frac{T_d}{2}s + 1)}{s\left(\frac{\alpha^2 T_d}{2}s + \alpha^2 + 2\alpha T_d + \frac{T_d^2}{2}\right)} \right). \tag{12}$$

Finally, to obtain the PID controller gains, simply match coefficients to the standard PID transfer function form:

$$\begin{aligned} C_{PID}(s) &= K_p + \frac{K_i}{s} + \frac{K_d s}{T_0 s + 1} \\ &= \frac{s^2(K_p T_0 + K_d) + s(K_p + K_i T_0) + K_i}{s(T_0 s + 1)}. \end{aligned} \tag{13}$$

This mapping yields the following formulas for the PID gains:

$$\begin{aligned} K_i &= \frac{1}{K\eta}, \\ K_p &= \frac{1}{K\eta}(\beta + \frac{T_d}{2} - T_0), \\ K_d &= \frac{1}{K\eta}(T_0^2 - T_0(\beta + \frac{T_d}{2}) + \frac{\beta T_d}{2}), \\ T_0 &= \alpha^2 \frac{T_d}{2\eta}, \end{aligned} \tag{14}$$

where $\eta = \alpha^2 + 2\alpha T_d + \frac{T_d^2}{2}$.

The only remaining issue is the selection of the free parameter α. In order to greatly simplify the analysis, let $\alpha = \gamma T_d$, where γ is a constant. To show why this implementation is useful, substitute $\alpha = \gamma T_d$ in the expression for the filter $F(s)$ (10) to obtain the frequency response:

$$F(j\omega) = \left.\frac{(2\gamma+1)T_d s + 1}{(\gamma T_d s + 1)^2}\right|_{s=j\omega} = \frac{(2\gamma+1)(\omega T_d)j + 1}{(\gamma(\omega T_d)j + 1)^2}.$$

Note that in this expression, ωT_d appear together. The same observation can be made for the Padé approximation (11). This allows for introducing the change of variable $\omega' = \omega T_d$, which removes T_d from the closed-loop transfer function and allows for analysis purely as a function of γ, which allows for *single parameter* tuning, and will become more useful in the next section. With this substitution, the PID gains become:

$$\begin{aligned} K_p &= \frac{1}{K_{GID}T_d}\left(\frac{8\gamma^3 + 20\gamma^2 + 16\gamma + 3}{(2\gamma^2 + 4\gamma + 1)^2}\right), \\ K_i &= \frac{1}{K_{GID}T_d^2}\left(\frac{2}{2\gamma^2 + 4\gamma + 1}\right), \\ K_d &= \frac{1}{K_{GID}}\left(\frac{(4\gamma^2 + 5\gamma + 1)^2}{(2\gamma^2 + 4\gamma + 1)^3}\right), \\ T_0 &= T_d\left(\frac{\gamma^2}{2\gamma^2 + 4\gamma + 1}\right). \end{aligned} \tag{15}$$

3.2 Theoretical Case and Metrics for Analysis

The goal of this section is to introduce what I call the *theoretical* or *perfect* IMC case of a GID plant with a GID plant model, i.e. $P(s) = \hat{P}(s)$. This will serve as the standard of comparison for the plant/model mismatch analysis, as the best-case use of IMC is when the plant and model are identical. Utilizing the theoretical IMC topology, certain metrics

can be established in order to quantify the success of the design technique.

The three major metrics looked at in this research are phase margin (PM), gain margin (GM), and closed-loop bandwidth (BW). Margins help determine the overall system's stability and likelihood of going unstable if gain or phase (i.e. delay) is injected into the system (higher is better, and negative margins indicate instability).

Phase margin is measured at the gain crossover frequency, ω_{gc}, or when the open-loop Bode plot has a gain of 1 (0dB), with the following value:

$$PM = 180^\circ + \angle P(j\omega_{gc}),$$

where $\angle P(j\omega_{gc})$ is negative for practical systems.

Gain margin is measured at the phase crossover frequency, ω_{cp}, or when the open-loop Bode plot has a phase of -180°, with the following value:

$$GM = \frac{1}{|P(j\omega_{cp})|}.$$

Bandwidth corresponds to the speed of a system, typically higher is better. This is measured as the first crossing of -3dB in the closed-loop Bode plot. For more information/derivations regarding margins and bandwidth, see [44].

Below are figures of the previously mentioned metrics vs. the tuning parameter, γ, for the theoretical case. These were generated by setting $P(s) = \hat{P}(s) = \frac{K_{GID}}{s} e^{-sT_d}$, utilizing the change of variables $\omega' = \omega T_d$, and using substitution $\alpha = \gamma T_d$. This makes the open-loop transfer function independent of T_d and K_{GID} (recall the use of a model inversion

term in the open-loop transfer function eliminating the K_{GID}).

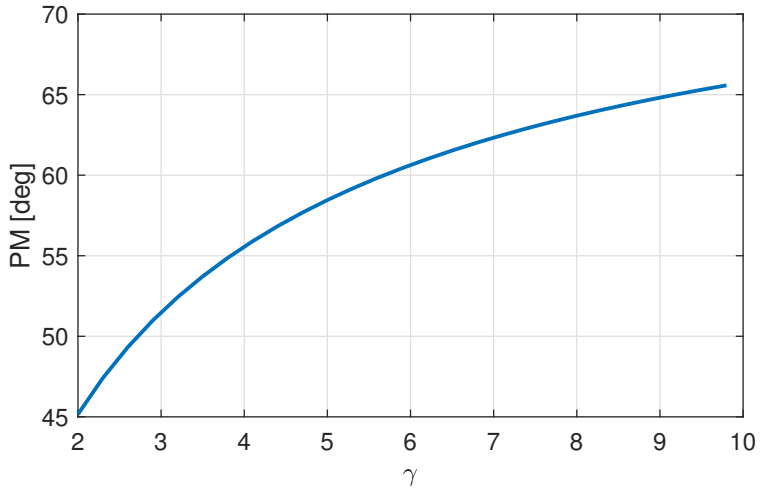

Figure 4: Theoretical phase margins for varying γ. A close fit is given by $PM \approx 67.42 - 39.53e^{-0.298\gamma}$, with R^2 value of 0.999.

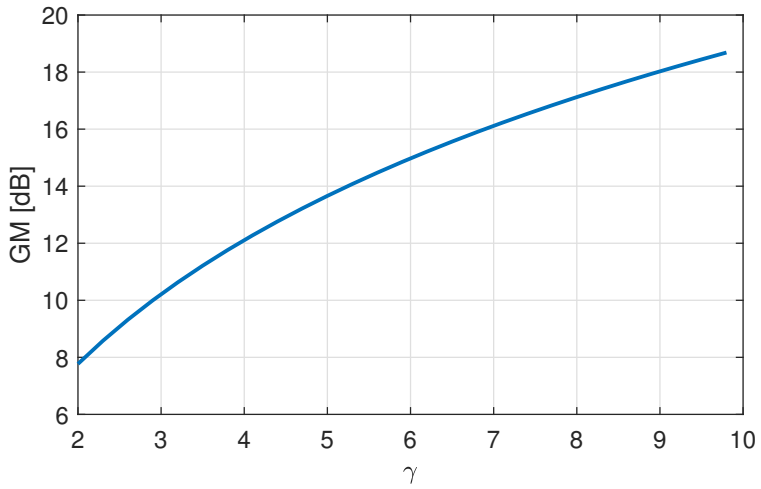

Figure 5: Theoretical gain margins for varying γ. A close fit is given by $GM \approx 22.75 - 20.53e^{-0.163\gamma}$, with R^2 value of 0.999.

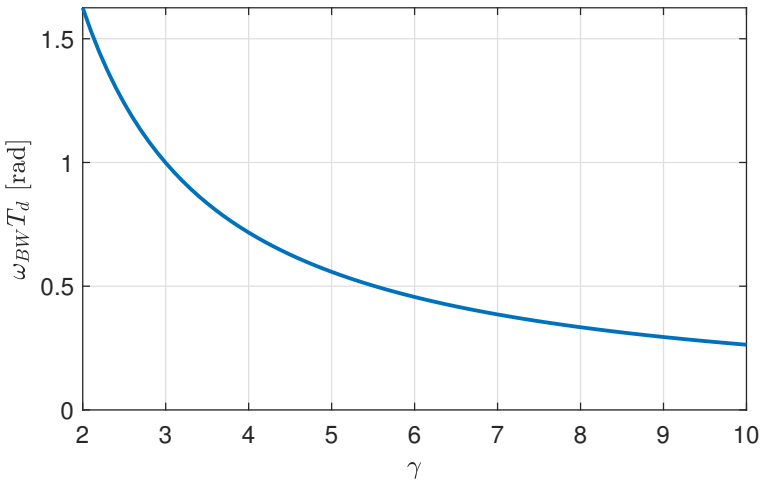

Figure 6: Theoretical product of closed-loop bandwidth and T_d. This was done to generalize the results regardless of the system. A close fit is given by $\omega_{BW}T_d \approx -0.086 + 3.29/\gamma$, with R^2 value of 0.998.

There are a couple key takeaways from these plots. First, margins increase and bandwidth decreases monotonically as γ increases, allowing for single parameter tuning. Also, a fundamental limitation of control is revealed: the system cannot be made too fast if the system must at the same time be robust (which agrees with standard control theory).

3.3 Analysis of the Padé Approximation

Recall from Section 2.2 that an ideal IMC controller minimizes the H_2 norm of the tracking error $\|e\|_2$. In reality, $Q(s)$ must be augmented with the low-pass filter $F(s)$, which no longer is H_2-optimal. However, if the bandwidth of $F(s)$ is large, its effect will be negligible. Using similar reasoning, the GID-based IMC controller developed in Section 3.1 is not optimal because of the filter $F(s)$ and, additionally, because of the Padé approximation. This section will study the impact of γ on the optimality of the IMC controller, in terms of this approximation.

To begin, note that the Padé approximation, (11), and the exponential, $e^{-j\omega T_d}$, have the same gain at each ω, but different phases. Thus, let the phase error, θ, be defined by the difference in their phases:

$$\theta = -2\text{atan}\left(\frac{\omega T_d}{2}\right) + \omega T_d.$$

Note that, similar to Section 3.2, ωT_d appears in the above expression as a product. Hence, θ is a function of ωT_d. Note also that the y-axis of Figure 6 is $\omega_{BW} T_d$. Hence, at $\omega = \omega_{BW}$ (the measured 3dB bandwidth), θ can be expressed explicitly as a function of γ, by inverting the function in Figure 6 and using the above expression for θ. The result is illustrated in Figure 7.

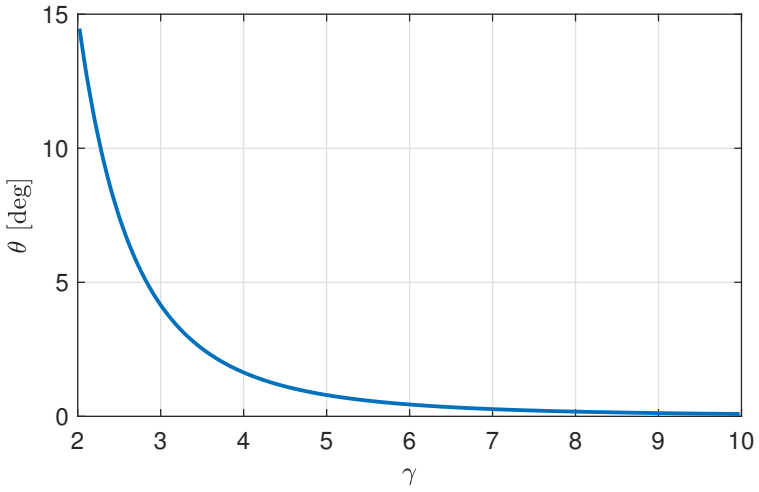

Figure 7: Error of the Padé approximantion's phase as a function of γ.

From this, it is evident that the larger the γ, the smaller the error in the Padé approximation at the closed-loop bandwidth (and, consequently, at all lower frequencies).

We can now combine the above ideas: the larger the value of γ, the better the accuracy

of the Padé approximation and the closer the resulting controller is to optimal.

3.4 Design Approach (DA) 1: Design for Post-hoc Tuning

The idea with this approach is to give the most flexibility to the end user, and this approach is the only method that allows for modification of the controller post-implementation.

First, the plant's phase crossover frequency, ω_{cp}, is determined. This frequency occurs when the phase of the plant is $-180°$, and is associated with the point where any more phase (lag, or delay) will produce instability. This can be done through the Ziegler-Nichols tuning method [8], wherein, a proportional controller is used in tandem with the system and the proportional gain is increased until sustained oscillations occur at the output, indicating the cusp of instability. The period of these oscillations is measured and the inverse value, frequency, is the ω_{cp}. This ω_{cp} becomes the ω_0 and the plant model parameters K_{GID}, T_d are identified through system characterization at ω_0 using equations (6) and (7).

Finally, the user has free choice over γ. The figures in Section 3.2 show what varying γ does to the margins (i.e. robustness) and the bandwidth (i.e. speed of system). If the system's response is not desirable, all one does is change the choice of γ (thereby changing the PID controller gains as described in (15); no other plant identification step at other ω_0 values is necessary, i.e. the values of K_{GID}, T_d are fixed after system characterization.

The downside here is that since γ isn't chosen to guarantee any design metric, the actual PM, GM, and BW will not match the theoretical values in the previous figures.

3.5 Design Approach (DA) 2: Design for Closed-loop Bandwidth (BW)

This design approach serves to refine Design Approach 1 by assigning ω_0 to be the desired closed-loop bandwidth. As it turns out, not all closed-loop bandwidths are assignable. Therefore, an acceptable range for ω_0 will be developed.

A phase margin minimum of 45° is a typical industry goal, and with that in mind, a lower bound of the acceptable ω_0 values will be established. Looking at Figs. 6 and 4, a PM of 45° corresponds to $\gamma = 2$, which also corresponds to an $\omega_{BW}T_d = \omega_0 T_d$ value of approximately $\pi/2$. Looking at the phase of the GID plant (the theoretical case):

$$\angle P(j\omega_0) = \angle \left(\frac{K_{GID}}{j\omega_0} e^{-j\omega_0 T_d} \right) = -\pi/2 - \omega_0 T_d,$$

together with $\omega_0 T_d < \pi/2$ from above, leads to the following bound on the plant phase:

$$\angle P(j\omega_0) > -\pi.$$

In addition, a GID model has phase strictly less than $-\pi/2$ due to the integrator. Thus, to ensure that K_{GID} and T_d can be chosen to match the GID model to the plant at ω_0, the plant phase must satisfy $\angle P(j\omega_0) < -\pi/2$. These requirements bound the acceptable range of plant phases at ω_0 (thereby bounding ω_0) as:

$$-\pi < \angle P(j\omega_0) < -\pi/2. \tag{16}$$

The ω_0 associated with the value of $\angle P(j\omega_0) = -\pi$ is the plant's phase crossover frequency, the same as determined from the Ziegler-Nichols analysis. The ω_0 associated

with the value of $\angle P(j\omega_0) = -\pi/2$ can be found using a modified relay feedback experiment, with the controller appended with an integrator [53].

With this bound on ω_0 defined, now the design process can be described. First, a desired closed-loop bandwidth, ω_0, is chosen within the range detailed in (16). The frequency response of the plant is then identified at this frequency via system characterization - the parameters of the GID model are selected using (6) and (7). Finally, the value of γ is selected to assign ω_0 as the closed-loop bandwidth. This is achieved using the theoretical plot of $\omega_{BW}T_d$ vs. γ, Figure 6. Since the plant is matched to the GID model at this frequency, the bandwidth is guaranteed to be assigned exactly.

If the system's response needs to be modified post-hoc, a new ω_0 must be chosen, which then indirectly chooses a new γ. This adds additional overhead, requiring a new system characterization step for every new ω_0. Despite this, the user has direct control over the bandwidth, or speed, of the response.

3.6 Design Approach (DA) 3: Design for Phase Margin (PM)

Sometimes closed-loop bandwidth is not the best design parameter. With that in mind, it is possible to modify design approach 2 to design for a another common systems metric: phase margin. Unfortunately, it is not possible to design for GM as the ω_0 associated with the phase crossover frequency (where the GM is measured) will violate the bounds described in (16).

The parameter γ is easy to ascertain - simply read off the plot of PM vs γ, Figure 4. Similar to the previous section, an acceptable range of PM values to assign will also be developed. As a lower bound, the typical industry goal of 45° will be used. To find an

upper bound, I will utilize the trendline on the plot of PM vs γ for large γ:

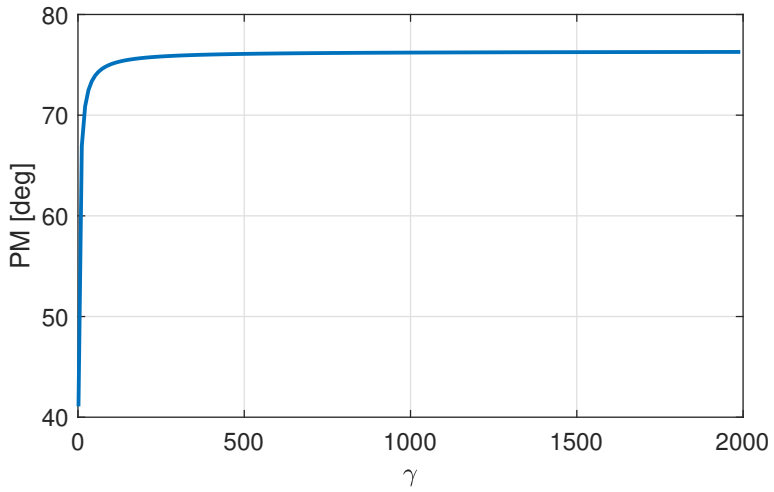

Figure 8: Plot of PM vs γ for large γ. This determines the upper bound on the range of acceptable values for which to assign PM.

These two ideas bound the acceptable range of PM values to:

$$PM_{des} \in [45°, 75°]. \tag{17}$$

Now is the process to pick the value of ω_0. To assign the correct phase margin, the idea is to match the frequency response of the GID model to the actual plant at the gain cross-over frequency of $P(s)C(c)$ (i.e., the frequency at which the phase margin is calculated). With this in mind, in order to greatly simplify the process, a plot of PM vs $\omega_{gc}T_d$ is developed, created in a similar manner to Figure 4 for the theoretical case. This plot can be seen in Figure 9.

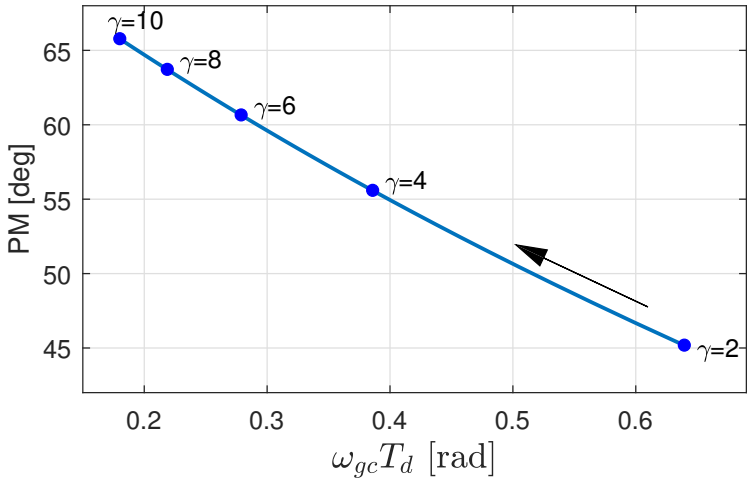

Figure 9: Plot of the quantity PM vs $\omega_{gc}T_d$, used in assigning the gain crossover frequency to assign a desired PM. Plot generated for the theoretical case of GID plant, GID plant model.

Now, the design process is as follows. First, utilizing Figure 9, the value of $\omega_{gc}T_d$ is determined from an initial desired PM. Unfortunately, this does not give the gain crossover frequency directly because the value of T_d is still unknown. To address this issue, the following workaround is developed. Using (7), the phase of the GID model at the crossover frequency, ω_{gc} (which is to be determined), is $-\omega_{gc}T_d - \pi/2$. Therefore, utilizing the $\omega_{gc}T_d$ value from Fig. 9, calculate the plant phase, $-\omega_{gc}T_d - \pi/2$. Using this value, identify the frequency (x-value on the Bode plot) at which the plant phase is equal to this value (y-value on the bode plot), which can be accomplished experimentally with the two-channel relay autotuner described in [54]. This frequency will become $\omega_0 = \omega_{gc}$. The GID parameters are then selected as usual using (6) and (7).

3.7 Illustrative Example

For the following examples, consider the plant:

$$P(s) = \frac{43e^{-0.05s}}{(0.2s+1)^2}, \tag{18}$$

which represents an experimentally validated and linearized model of a turbocharged gasoline engine from wastegate to boost pressure.

3.7.1 Design Approach 1

First, the value of ω_{cp} is found to be 13.8 rad/s. This becomes ω_0. Next, say we want a relatively fast system and don't care about the margins as much, so select $\gamma = 2$. Since we have chosen the parameters ω_0, γ directly, the GID model parameters can be found from Equations (6), (7), and the PID gains can be found from (15).

This results in a system with PM = 41.3°, GM = 9.18 dB, and BW = 14.4 rad/s. Say we wanted higher margins instead - with this design approach we have the opportunity to easily change γ for a better response. For example, a $\gamma = 4$ results in: 67.5° PM, 12.9 dB GM, and 9.60 rad/s BW.

To visualize the above, below is the open-loop Bode plot, illustrating the matching of the GID-based $\hat{P}C$ to the actual plant's PC:

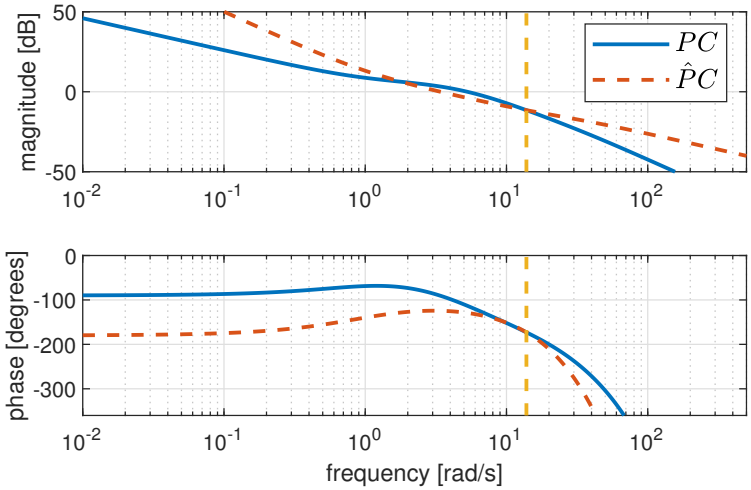

Figure 10: Open loop Bode plots of both the 2nd order system with GID designed controller (PC), and the GID system with GID designed controller ($\hat{P}C$) identified at $\omega_0 = \omega_{cp} \approx 13.8$ rad/s, post-hoc tuning case.

Finally, to get a better understanding what the changing of γ does to the closed-loop system, below are step responses of the two closed-loop systems, one for $\gamma = 2$, the other for $\gamma = 4$:

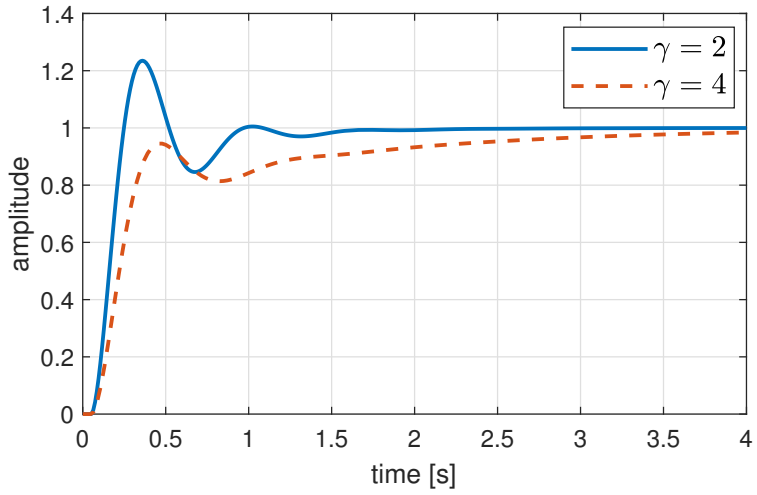

Figure 11: Closed-loop unit step responses of the 2nd order system designed with the GID plant model, post-hoc tuning case, $\gamma = 2, 4$.

3.7.2 Design Approach 2

Knowing that the maximum assignable $\omega_0 = \omega_{cp}$ value is 13.8 rad/s from the previous section, select the slightly smaller $\omega_0 = 10$. Now perform system characterization to obtain frequency response data, for the calculation of K_{GID}, T_d using (6), (7).

Next, we calculate $\omega_0 T_d = 1.14$, and using Figure 6, $\gamma = 2.68$. Finally, the PID gains can be calculated from (15). These system parameters result in a system with PM = 61.5°, GM = 12.6 dB, assigned BW of 10 rad/s.

Below is the *closed loop* Bode plot for both systems, validating that the GID model is matched to the plant at $\omega_0 = 10$, and the 3dB (closed-loop) bandwidth has been assigned to ω_0.

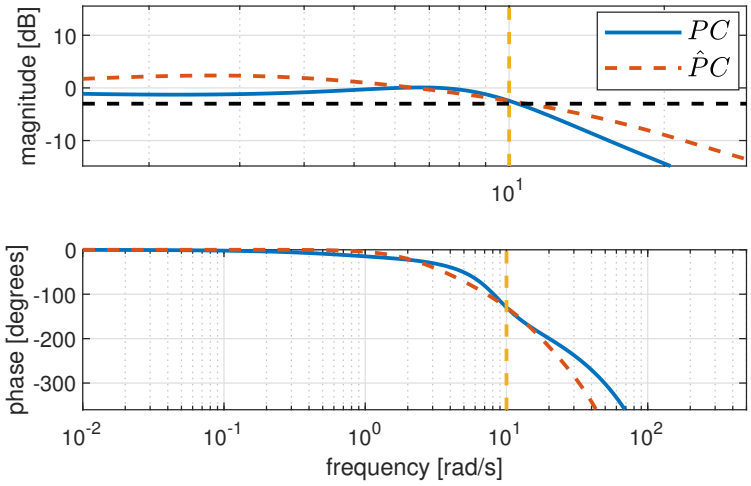

Figure 12: Closed loop Bode plots of both the 2nd order system with GID designed controller (PC), and the GID system with GID designed controller ($\hat{P}C$) identified at $\omega_0 = 10$, design for BW case. The horizontal dotted line is at -3dB, demonstrating the BW was assigned to $\omega_0 = 10$ rad/s.

From this plot, it is evident the bandwidth has correctly been assigned to $\omega_0 = 10$. The phase margin of the system is 63.5°, greater than the theoretical 47° from Figure 4, and

the gain margin is 13 dB, greater than the theoretical 9dB. These values are not close due to the GID model not being matched to the actual plant at the crossover frequencies, i.e. from where the margins are measured.

3.7.3 Design Approach 3

First, select a suitable desired phase margin, $PM_{des} = 55°$. This is within the range specified by (17). Now γ can be read off Figure 4, with a value of 3.83. Now comes the tricky part of selecting ω_0. First, read the value of $\omega_{gc}T_d$ associated with $PM_{des} = 55$ off of Figure 9, with a value of 0.39 rad. Now using (7), the phase of the GID model is $\frac{-\pi}{2} - 0.39 = -1.97$ rad. Finally, find the frequency (x-value on the plant's Bode plot) associated with this phase, this value becomes $\omega_0 = 5.61$ rad/s (and is the vertical dotted line on Figure 13).

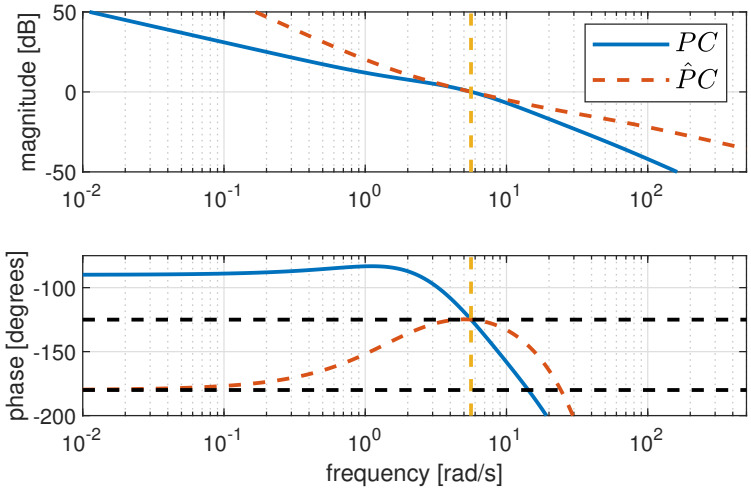

Figure 13: Open loop Bode plots of both the 2nd order system with GID designed controller (PC), and the GID system with GID designed controller ($\hat{P}C$) identified at $\omega_0 = 5.61$, the gain-crossover frequency associated with a PM of 55°. The horizontal dotted lines at -180° and -125° show that a PM of 55° has correctly been assigned.

This plot shows that the plant has been set equal to GID model at the gain crossover

frequency, with a $PM_{des} = 55^{\circ}$ being assigned. This design results in a GM of 11.89dB, very close to the theoretical 11.82dB. This is because not only are the phase crossover frequencies so close together, but at this frequency, the magnitude plots are extremely close. As Section 4 will show, this is most likely due to the nature of the plant: this plant has no complex poles or zeros, which produces a very well-behaved (and close-to-GID) Bode plot.

4 Monte Carlo Analysis for the Case of Plant-Model Mismatch

4.1 Executive Summary

As the literature shows, IMC has been proven to work great if the plant dynamics are close to the plant model's dynamics. This section hopes to figure out how far from GID dynamics one can go (within reason) before margins degrade.

With that in mind, a Monte Carlo experiment was devised to observe each design approach's success (as compared to the theoretical case) over a wide range of plant parameters, chosen to be approximately a factor of 10 greater/less than the validated wastegate transfer function, (18).

Four general plant styles were chosen, a first order, 2nd order with real poles, 2nd order with complex poles, and a 2nd order plant with real poles and a zero. For a description of how the specific plant parameters chosen, see the upcoming subsections. Each design approach was applied to these plants; approach 1 utilized $\gamma \in [2, 10]$, approach 2 utilized 20 ω_0 values associated with the range (16), and approach 3 utilized 10 PM_{des} values within the range detailed in (17). These quantities were chosen to generate sufficient data.

This section will create a vast number of plots, all of which can be found in Appendix B, and only the most interesting plots and data trends will be highlighted in this section. To provide a sense of all the overall trends in a single plot, see the plot below of mean GM for every case:

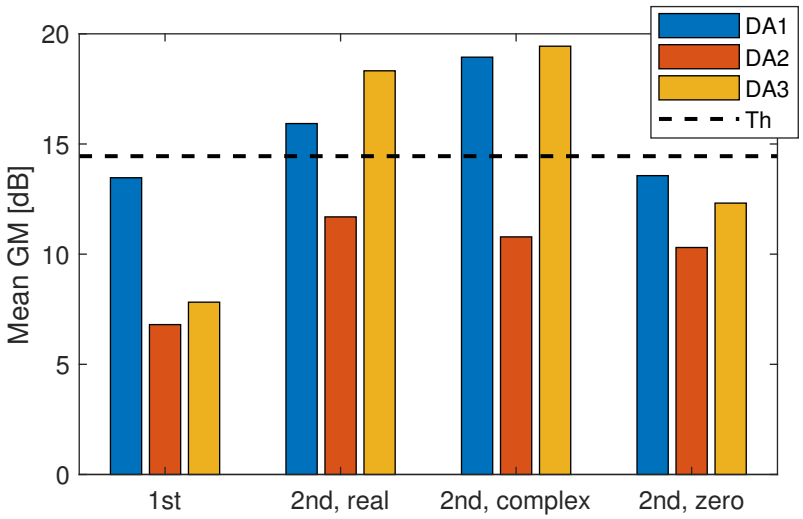

Figure 14: Mean GM for all cases. DA = Design approach, Th = theoretical case, i.e. from Figure 5. These values were arrived at by averaging all the GM values obtained from the MC analysis.

This plot shows that on the average, the four plant types looked at will perform comparably to one another, and deviate not too far from the theoretical best-case scenario. This data here does not tell the full story, however. Much of the larger-than-theoretical values are associated with many lower-than-theoretical values. On the whole though, this data indicates this technique as fairly effective for each of the plants tested. Negative margins were only experienced in the Design Approach 3 cases, and will be explained in Section 4.3. This means that (nearly) all cases tested are stable, with fairly high mean GM.

All *metric vs* γ plots can be found in Appendix B. The general synopsis is that the experimental data follow the general trends of higher γ, higher margins and lower bandwidth. In addition, the more complicated the plant (more poles/zeros, especially complex ones), the larger the spread of these values, with some (a minority but still not insignificant) going unstable, as stated previously.

4.1.1 Monte Carlo End Condition

With any Monte-Carlo analysis (and especially with this one), an important question to ask is 'when will I have simulated enough data/systems'? There exist numerous techniques to determine this in the literature, many of which are summarized in [55].

Here, I looked at the phase margin variation, and approximated the standard error of its mean (SEM) via bootstrapping [56]. I stopped sampling random systems when the SEM was less than 1% of the mean. In practical terms, at the tail end of the sampling, the mean PM for the 1st order case varied by less than 1°:

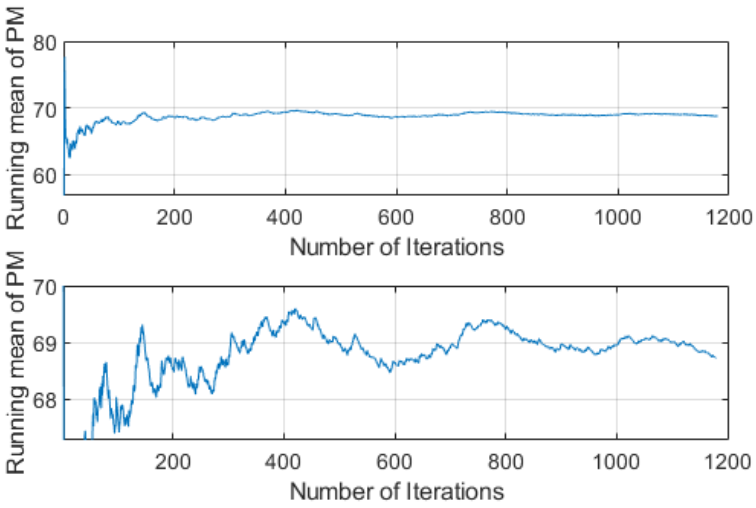

Figure 15: Running mean of the PM, for the 1st order case, $\gamma = 3$. Lower plot is zoomed, note the convergence and minimal amount of variance.

I made sure to apply this process for every Monte Carlo experiment. However, for the case where PM is designed for/is assigned - the algorithm had to be modified slightly - GM was used as the ending parameter and the end condition was increased to 5% due to numerical timeouts. For a more complete description of this process, see Appendix A.

4.1.2 Discussion of the *real* margins/bandwidth

An indirect result of this analysis has been an in-depth discussion about how we define margins and the 3dB bandwidth - at the first ω_{cp}/3dB crossing or the last? This issue manifests itself as bimodal data - some (measured) data points that are closely grouped to the theoretical curves, and some that are much lower (or even higher).

This also becomes an issue because it begs the question - how do I verify that the 3dB bandwidth has been assigned - if design approach 2 picks a bandwidth off of the theoretical curve and the measured value doesn't match does that mean the process has not been accomplished correctly? To make a long story short, due to the multiple crossings, despite the bandwidth being assigned correctly, the measured 3dB bandwidth (the first crossing) does not match the theoretical value. This is evident with the following plot of $\omega_{BW}T_d$ vs γ for one of the first order case in Section 4.2:

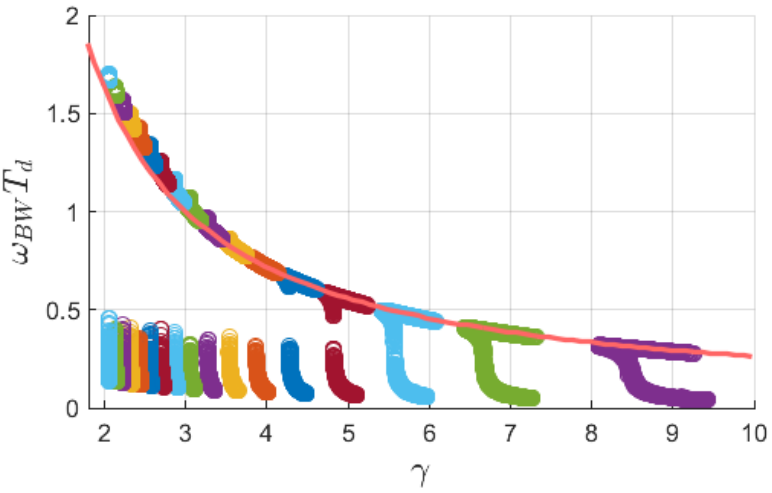

Figure 16: The measured bandwidth (circles) should always fall on the theoretical curve, as the γ value is picked directly off the curve. However, if you use the strict definition of first 3dB crossing for bandwidth, you get this bimodal behavior in the data. That is why for all plots of this nature, I operate under the assumption of picking the 3dB crossing closest to the theoretical curve.

The multiple crossings can be seen within the Bode plot of an example system falling in the lower section:

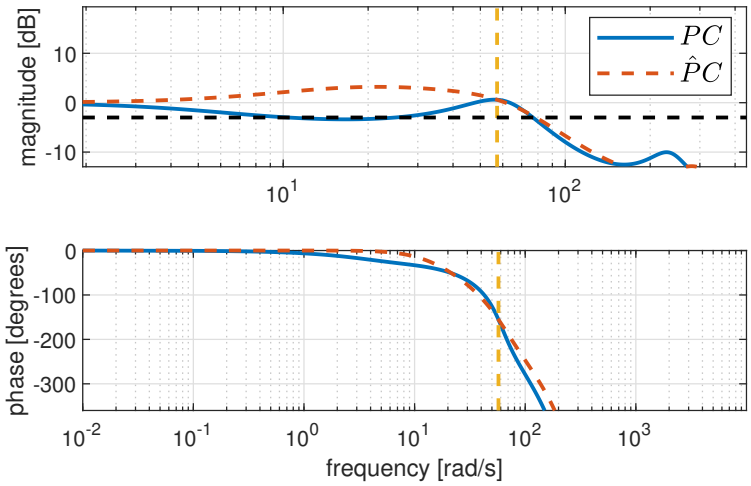

Figure 17: This CL bode plot demonstrates the discrepancy observed in the previous figure. The horizontal dotted line is at -3dB.

With this in mind, the following tweak is implemented - for measuring 3dB bandwidth, pick the 3dB crossing closest to the theoretical one. For measuring margins, since the first crossing is most important for these metrics (where is the *first* point of instability?), I will stick with using the accepted definition of first crossover frequency to measure margins.

4.2 First Order Case

Consider a plant of the form:

$$\frac{k}{\tau s + 1} e^{-sT}. \tag{19}$$

For this case (and all other cases containing gain k, time constant τ, and delay T), the parameter k is uniformly distributed over the region $[4, 400]$, and the parameters τ and T are logarithmically distributed over the region $[0.03, 3]$ and $[0.01, 1]$, respectively. This is to

have uniform pole distribution across the interval. This is accomplished by first generating $x \in [\log(0.03), \log(3)]$ and $y \in [\log(0.01), \log(1)]$ uniformly and then letting $\tau = 10^x$ and $T = 10^y$. In addition, systems whose delay was greater than their time constant ($T > \tau$) were rejected due to this behavior not being common in physical systems.

4.2.1 Design Approach 1

Figure 18 shows the box plots of $\omega_{BW}T_d$, PM and GM of the systems, respectively, as a function of the tuning parameter γ for Design Approach 1. The theoretical curves are also superimposed.

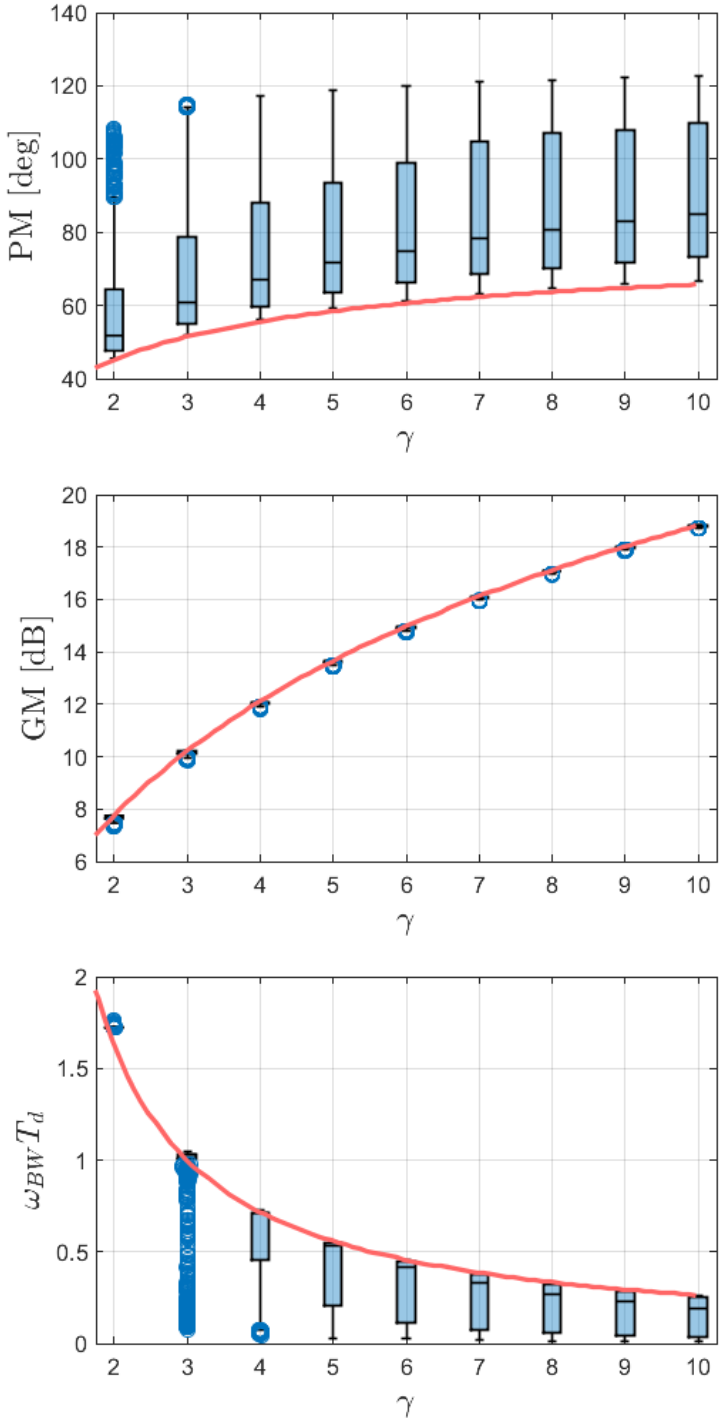

Figure 18: 1st order case, DA1.

It can be seen that the gain and phase margins are positive, with mean very close to,

if not better than, the theoretical case.

In fact, for the case of GM, the random systems are closely grouped around the theoretical curve. This is because for Design Approach 1, the GID model is set equal to the randomly generated plant at the plant's phase crossover frequency, which can be shown to be close to the phase crossover frequency of PC. Because of this, and the fact that 1st order systems do not have the more complicated dynamics (indicating one phase crossover frequency instead of multiple), the GM values are closely grouped around the theoretical curve.

4.2.2 Design Approach 2

Figure 19 shows the plots of the stability margins for Design Approach 2. Again, the phase margins exceed the theoretical values and gain margins do not dip below the typical benchmark of 6 dB. Note that a plot of BW is not provided as bandwidth is assigned exactly - for this plot, see Appendix B.

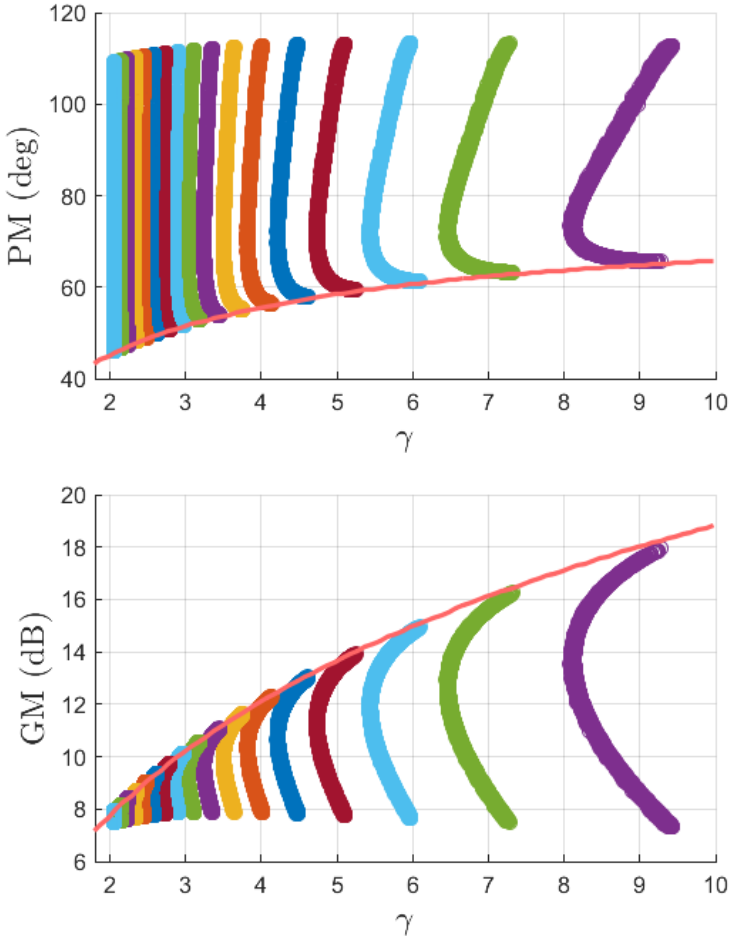

Figure 19: 1st order case, DA2.

As an aside, coming to grips with a choice of ω_0 can feel very arbitrary. At best, one has a choice over ω_0 *relative* to the ω_{cp}. But how does one know if one bandwidth is *better* than another? How do you compare the choice of bandwidth?

That is why I'd like to introduce one more metric for analysis for design approach 2. Call this the *error* of the choice of ω_0. The idea here is that even though there is an inherent relationship between bandwidth and speed of response, the choice of ω_0 is still fairly arbitrary. With that in mind, define *error* in terms of the closed loop transfer

functions:

$$\left\| \frac{PC}{1+PC} - \frac{\hat{P}C}{1+\hat{P}C} \right\|_2, \tag{20}$$

where $||.||_2$ is the $\mathcal{H}_2$ norm. In order to normalize the data, the resulting difference was divided by the $\mathcal{H}_2$-norm of the second term in the difference, creating a value relative to the theoretical, perfect-GID-model case. This analysis can only be applied to Design Approach 2, as no other design technique allows free selection of ω_0. The idea here is to provide a quantitative value on what could be considered the *best* choice of ω_0, in terms of closeness to the theoretical case.

With that in mind, for all the randomly generated systems in this Monte Carlo section, the following curve can be generated:

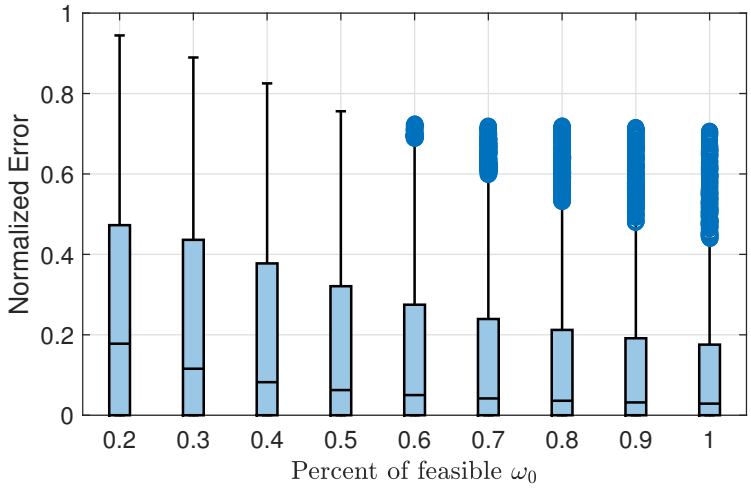

Figure 20: Value of the normalized error as a function of choice of ω_0, 1st order case for Design Approach 2.

Each tick on the x-axis represents that percentage of the feasible range of ω_0, as described in (16). These data indicate that the closer the assigned bandwidth, ω_0, is to ω_{cp}, the closer the closed-loop dynamics will perform to the theoretical case.

4.2.3 Design Approach 3

Finally, Figure 21 displays the margins for design approach 3. The plot of PM can be found in Appendix B.

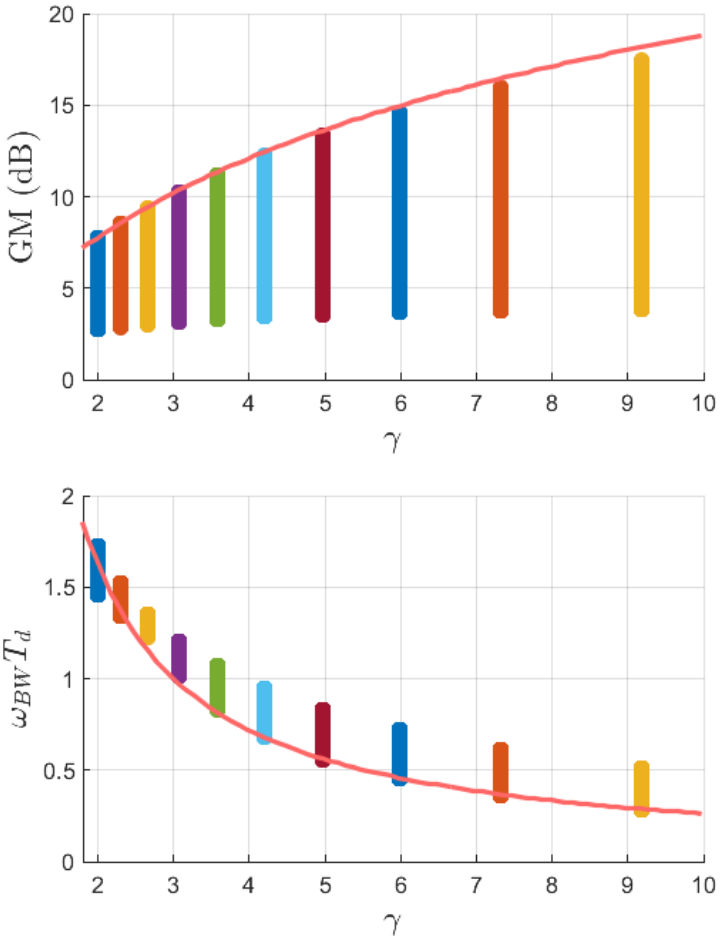

Figure 21: 1st order case, DA3.

Gain margins are still high, but are lower than Design Approach 2. The $\omega_{BW}T_d$ values are closely grouped around the theoretical curve, but are not exact due to the lack of a guarantee on bandwidth when phase margins are assigned.

Overall, for all 1st order systems considered, each design approach performs well, with

no unstable systems and acceptable margins (that will on occasion dip below the industry standards but are acceptable on the average). These data would indicate that any design approach can be applied to a reasonable 1st-order plant, in terms of closeness to (18).

4.3 Second Order cases

Below are the transfer functions used for this section of the analysis:

Second Order Case with Real Poles For this case, plants are of the form:

$$\frac{k}{(\tau_1 s + 1)(\tau_2 s + 1)} e^{-sT}. \tag{21}$$

Second Order Case with Complex Poles For this case, plants are of the form:

$$\frac{k\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} e^{-sT}. \tag{22}$$

And ω_n was treated like τ in the previous cases, varying logarithmically over the interval $[0.03, 3]$. The parameter ζ was varied uniformly over $[0, 1]$. For this case, systems were rejected if $\omega_n t > 1$.

Second Order Case with a Zero Finally the effects of a zero were analyzed. Plants are of the form:

$$\frac{k(\tau_3 s + 1)}{(\tau_1 s + 1)(\tau_2 s + 1)} e^{-sT}. \tag{23}$$

The zeros looked at were all in the left-half of the complex plane, nonminimum-phase dynamics were ignored.

The general story here is that the added complexity of poles/zeros will shift the Bode plot's crossover values further away from the theoretical ones. Sometimes, this works to our advantage - sometimes the margins vastly exceed their theoretical counterparts. However, despite this, there is much more spread in the values than the 1st order case.

Most notably, some of the systems actually go unstable. See the complex poles, DA3 case GM plot below:

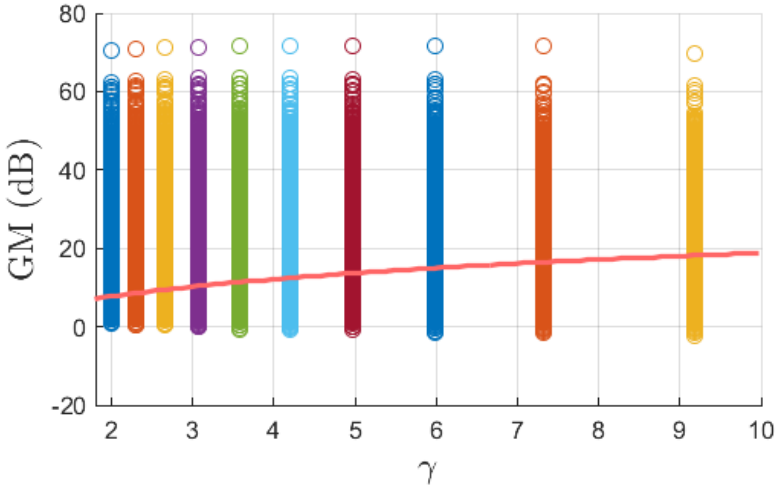

Figure 22: Plot of GM vs γ for the 2nd order case with complex poles, design approach 3.

Looking at two systems' step responses; one in the $\gamma \approx 6$ column with slightly negative margins (right), and the same system but with a smaller γ value ($\gamma \approx 5$ column), but with slightly positive margins (left):

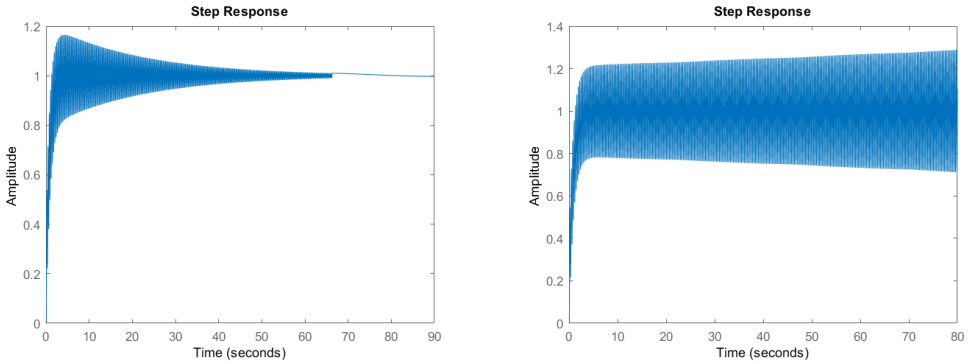

Figure 23: Step responses of a system with slightly positive GM (left), and slightly negative GM (right).

Figure 23 indeed confirms that some of the systems have gone unstable. To explain this, take a look at this system's Bode plot and Root locus:

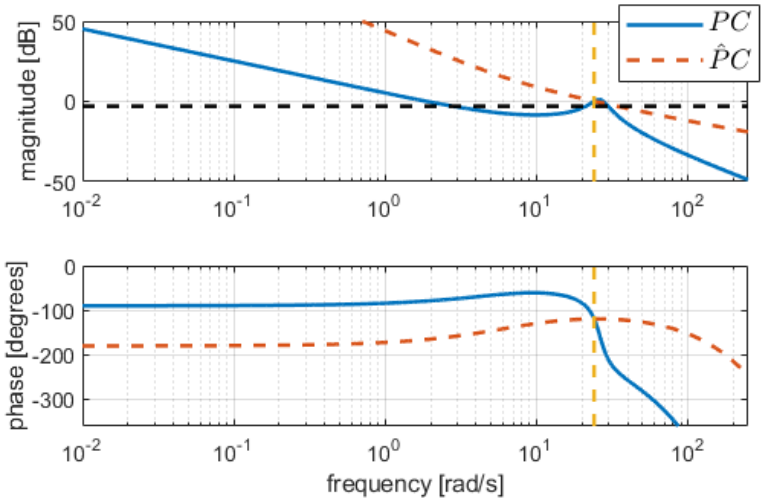

Figure 24: The system in question's open-loop Bode plot.

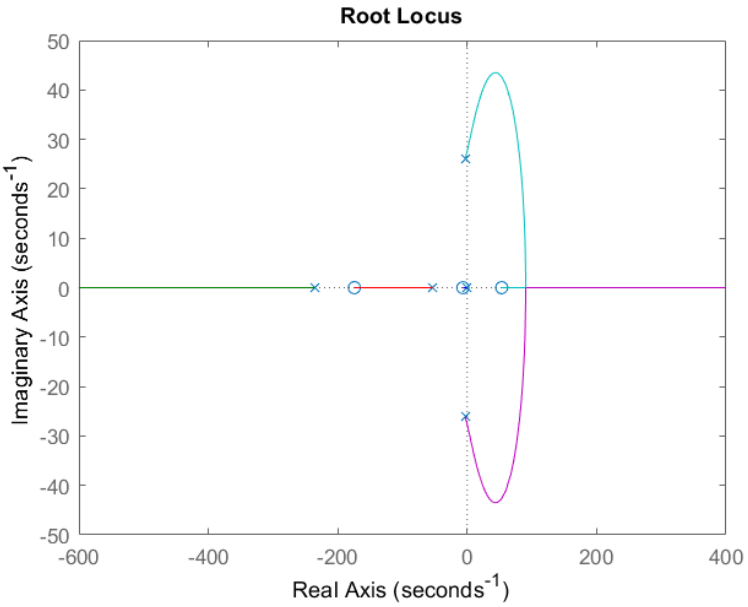

Figure 25: The system in question's Root Locus plot.

Figure 24 shows a kink in the phase plot of PC around the -180° crossing, where the GM is measured - due to PC's phase crossover being right where the magnitude plot goes positive, you get a negative GM (instability).

Figure 25 confirms my suspicions - there exists a very small window of gains that make the system stable. This is why, for the same system, $\gamma \approx 5$ (lower gain) was stable, but $\gamma \approx 6$ (higher gain) was unstable. Indeed, going back to Figure 22, there are no systems that have negative GM for the smaller γ values (i.e. small PM assignment) - the gain associated with these values keeps the closed-loop system stable.

In summary, for first order plants, any DA can be applied. However, with more complicated plants, it becomes harder to predict the behavior of the system, and with DA3, it is possible for the system to go unstable. As a final note, the previous implementation of these design techniques on the second-order turbocharger system performed well.

5 Physical Implementation

The three proposed design techniques will now applied to a physical process, the Quanser QUBE^{TM} servo-2 device [57]. A picture of this device can be seen in Figure 26.

Figure 26: Quanser QUBE^{TM} servo-2 device.

The system is a brushless DC motor controlled by an input voltage. The direct output is a position, as measured by an optical encoder. In my analysis, I add a derivative filter to this output in order to control speed, instead of position. This was done to simplify the analysis. For a full description of the Simulink models used, see Appendix C.

A simple plant identification experiment has revealed the device's dominant dynamics are first order, with the following transfer function:

$$P(s) \approx \frac{1379}{0.18s + 1}$$

This is used for didactic purposes only; never in the analysis is this $P(s)$ used, instead the physical device itself is used to gather data.

As demonstrated in Section 3.1, all three design approaches require the selection of two

parameters: ω_0, the frequency at which to match the GID plant to the physical system, and γ, the tuning parameter.

Once an ω_0 is chosen, the plant needs to be identified, i.e. $|P(j\omega_0)|$ and $\angle P(j\omega_0)$ need to be found. For the plant/model mismatch analysis (Section 4), this is trivial because the plant transfer function is known. However, when implementing on a physical device, this data needs to be ascertained experimentally.

The simplest approach is to exploit LTI systems analysis through 'picking' a value off of the Bode plot, in other words, input a sinusoid directly to the plant and measuring the resulting sinusoid's gain and phase. This was the approach taken in my work. Other methods do exist for system characterization, one of the most popular being relay feedback [52].

5.1 Design for Post-hoc Tuning

In this approach, the design frequency, ω_0, is chosen to be the plant's phase cross-over frequency, ω_{cp} (recall that phase cross-over frequency is the frequency at which the phase of the plant is -180 degrees). Thus, we first identify ω_{cp}. This can be done easily through Ziegler-Nichols analysis [8], which sets the K_i and K_d gains to zero while gradually increasing the K_p gain until sustained oscillations are reached at the output. This process is illustrated in Figure 27 below.

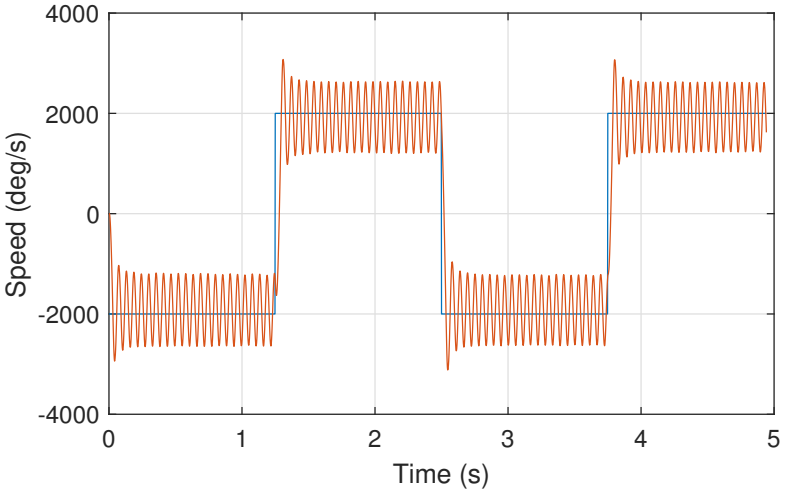

Figure 27: A square wave (continuous step input) was input to the closed-loop system, with just a proportional controller. This gain was increased until sustained oscillations were present at the output, as evidenced by the plot. To get ω_{cp}, the sinusoid's period was measured. The resulting gain of the sinusoid not used in my work, but it is used to compute the standard Ziegler-Nichols controller [8].

The frequency of these oscillations is $\omega_{cp} = \omega_0 \approx 114$ rad/s. Next the model parameters at this ω_0 are found by inputting a sinusoid to the plant and measuring the resultant sinusoid's gain and phase values. This is done below in Figure 28. Now the GID model parameters K_{GID} and T_d are selected using (6) and (7).

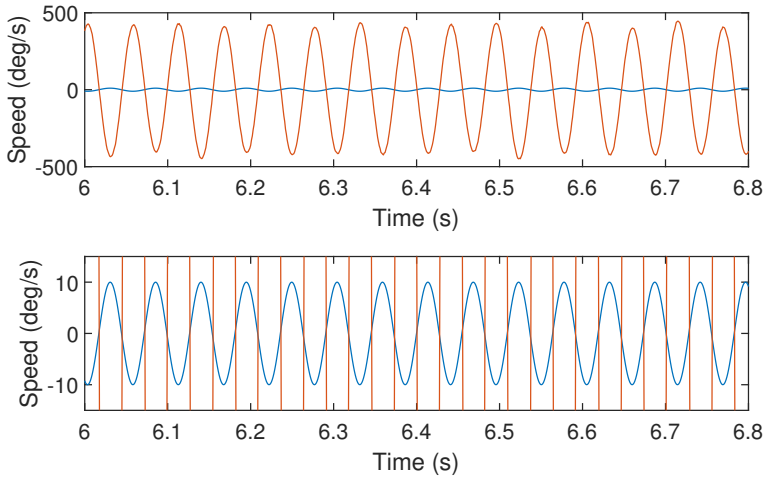

Figure 28: System characterization at the phase crossover frequency, ω_{cp}. Lower plot is zoomed. The nearly 180 degree phase shift confirms that ω_{cp} was found correctly (within measurement error).

Finally, the γ value is chosen to the designer's liking, and the PID gains are computed from (15). To get a sense of which γ to select, as dictated from Figures 4, 5, and 6, higher γ's generally correspond to lower bandwidth (i.e., slower response) but better robustness (i.e., higher gain and phase margins).

Below is a plot of a few step responses for PID controllers designed in this way. In addition, the standard Ziegler-Nichols tuned PID controller is added for a state of comparison, using the P-controller gain, K_u, associated with sustained oscillations, and the resulting period of the oscillations, P_u, the PID gains can be calculated [58]:

$$\begin{aligned} K_p &= 0.6K_u, \\ K_i &= 1.2\frac{K_u}{P_u}, \\ K_d &= 0.075K_uP_u. \end{aligned} \tag{24}$$

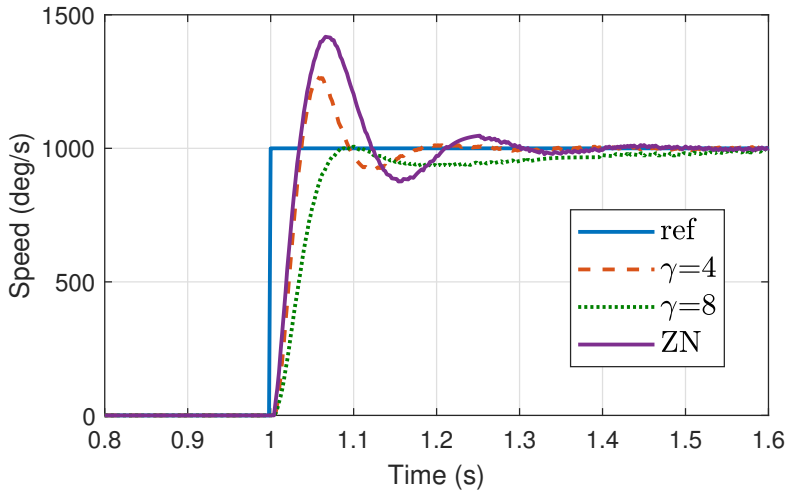

Figure 29: Sample step responses for design approach 1: design for post-hoc tuning. Smaller γ indicates faster response but lower margins, as echoed by the theoretical plots.

This figure is nicely tied to Figure 6 in showing speed of the response (BW) as a function of γ. With a lower choice of γ, it is possible to closely replicate the Ziegler-Nichols PID controller, and the beauty of this design approach is that it offers the flexibility to change this response if desired.

5.2 Design for Bandwidth

Next, the process of designing for a desired closed-loop bandwidth was implemented on the device. Knowing that ω_{cp} is approximately 114 rad/s from the previous section, I choose two ω_0 values slightly smaller than this, and to ensure they are feasible, I verify $\angle P(j\omega_0)$ is within the bounds described in (16).

For my first choice, I used $\omega_0 = 100$ rad/s in order to get a fast response that gave me enough buffer so that I wasn't teetering on the boundary condition of 114 rad/s. For the second value, I wanted to choose a slower response to resemble the choice of $\gamma = 8$

in the previous section, with $\omega_0 = 40$ rad/s. These choices of ω_0 are associated with a phase values, $\angle P(j\omega)$, of 143° and 122° respectively. Again, to ascertain phase of the plant experimentally, the process as described in the introduction of this section was used, input a sinusoid and measure the resulting sinusoid's gain and phase as compared to the input.

To determine whether these PID gains have correctly assigned a closed-loop bandwidth, a small-amplitude sinusoid of frequency ω_0 rad/s was input to the closed-loop system, and then the resulting output sinusoid's gain was confirmed to be $\frac{1}{\sqrt{2}}$ of the input sinusoid (i.e., -3dB). This process is illustrated below in Figure 30

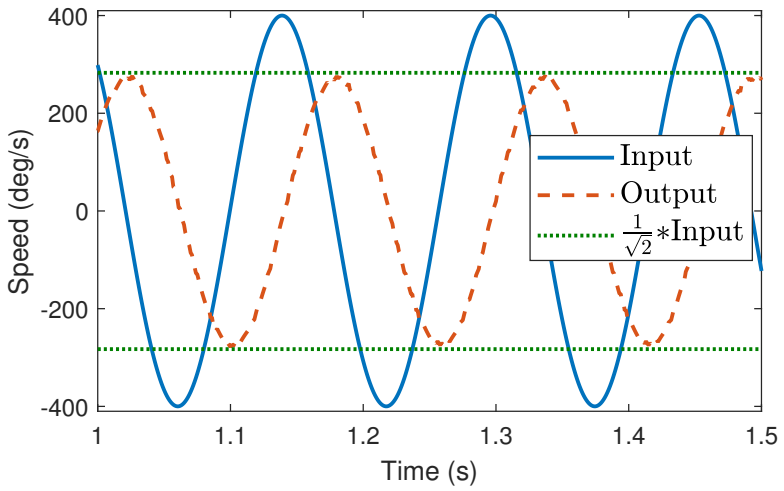

Figure 30: This plot serves to confirm that the closed-loop bandwidth was assigned. A small-amplitude (400 deg/s) sinusoid of frequency $\omega_0 = 40$ rad/s was input to the closed-loop system and the output sinusoid's gain is confirmed to be $\frac{1}{\sqrt{2}} * 400 \approx 282$ deg/s. This process is the equivalent of finding the -3dB crossing on the system's Bode plot, but experimentally.

Finally, below are step responses for the chosen $\omega_0 = 100, 40$ values:

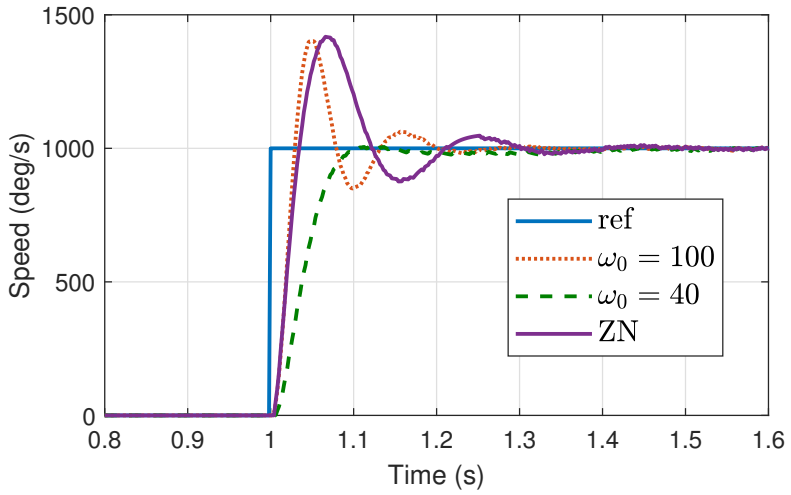

Figure 31: Sample step responses for design approach 2: design for closed-loop BW.

The larger choice of ω_0 is fairly close to the Ziegler-Nichols controller, as ZN is based around ω_{cp}, and the lower choice of ω_0 demonstrates the amount of depth the designer has for shaping the system's step response.

5.3 Design for Phase Margin

I start with two choices for a desired phase margin, $PM_{des} = 50°, 60°$. The following data will be for $PM_{des} = 50°$. The value of γ can be read off Figure 4 directly, $\gamma = 2.7$. Now, the ω_0 needs to be selected. As mentioned previously, this will be the frequency associated with the phase of $-\pi/2 - \omega_{gc}T_d$, or $-119.5°$. To find this ω_0, the two-channel relay autotuner in [54] could be used; however, I opted to use sinusoidal inputs at the plant of increasing frequency until the desired output phase was reached, with $\omega_0 \approx 38$ rad/s. The gain and phase of the plant were then measured at this frequency (system identification). These values were $\angle P(j\omega_0) = -119.4°$, and $|P(j\omega_0)| = 227.5$. Finally, the GID model was identified from these values and the PID gains were determined from (15).

To measure the PM experimentally, a sinusoid was input to the open-loop system (controller and plant) at the input frequency of ω_0. Because ω_0 was chosen to be the gain-crossover frequency, this ω_0 should be where the gain of the system is equal to 1 and the phase is used to measure the PM. The resulting output sinusoid had a gain of 1.07 and a phase of $-134.5°$, indicating a phase margin of $-(-180 + 134.5) = 45.5°$, which is within operator/measurement error of the desired 50°. A plot of the resulting sinusoid can be seen below in Figure 32. For more details and the Simulink model used to measure PM experimentally, see Appendix C.

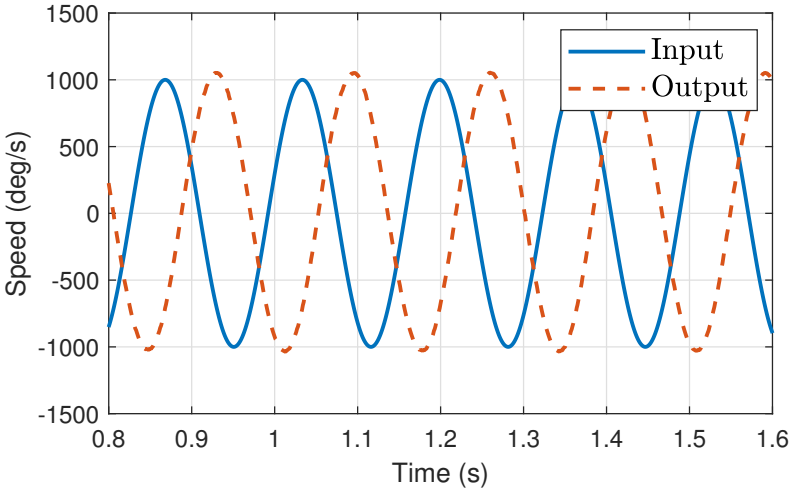

Figure 32: This plot confirms that the PM was correctly assigned to 50°, within operator error. The input sinusoid has a frequency, $\omega_0 = \omega_{gc} \approx 38$ rad/s. Since ω_{gc} is where the PM is measured, the output sinusoid should have a gain of 1 (which it does), and the phase margin is the difference between 180 and the phase of the output waveform.

Finally, below are a couple step responses for this system, for $PM_{des} = 50°, 60°$. These values were chosen to be within the range of acceptable PM values, as described by (17).

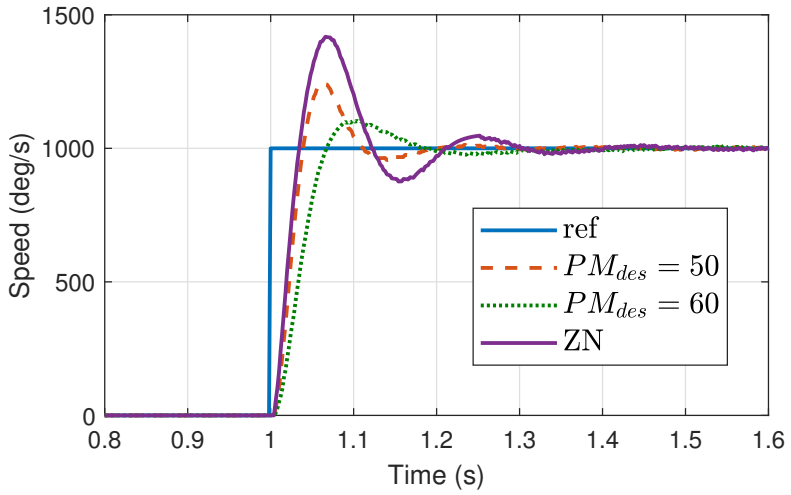

Figure 33: Sample step responses for design approach 3: design for PM.

The steps responses shown here are very similar to the previous sections' steps, with the added benefit of the operator having assigned the useful metric of phase margin throughout the process. This process should be implemented only if post-hoc modification of the step is not desired, as this process is the most labor-intensive.

5.4 Summary

These experiments have confirmed and validated the three design paradigms as effective means for calculating PID gains. Designing for post-hoc tuning is the easiest to accomplish and is the most user-friendly, in offering the option to tune the response with a single parameter. The latter two methods, while requiring more steps, are able to ensure desirable metrics on the closed-loop system.

6 Conclusions and Future Work

The control of boost pressure within the turbocharger is a difficult problem, and engineers much smarter than me have laid the foundation of using the gain-scheduled PID controller topology to tackle it. Morari, Ziegler/Nichols, and others made picking these PID gains much easier. Using the IMC tuning fundamentals, this thesis made it even easier for the control engineer to pick these PID gains, with (3) new approaches to the tuning methodology.

These methods were investigated via Monte Carlo analysis for use with plants not similar to the original use-case of boost pressure. In general, the technique works best for first order systems, and the addition of complex poles or a zero will cause variation and degradation in the margins. Finally, teehe Monte Carlo work was extended by implementing the designs on the Quanser QUBE$^{\text{TM}}$ Servo-2 device, confirming that the scope of the design is not limited to the automotive realm.

Topics of future research include discrete-time analysis of the design process, further analysis on the effect of plant/model mismatch, and potentially the analogous extension of this technique for use with higher-fidelity models.

References

[1] P. Moulin, J. Chauvin, and B. Youssef, "Modelling and control of the air system of a turbocharged gasoline engine," *IFAC Proceedings Volumes*, vol. 41, no. 2, pp. 8487 – 8494, 2008. 17th IFAC World Congress.

[2] J. E. Hadef, S. Olaru, P. Rodriguez-Ayerbe, G. Colin, Y. Chamaillard, and V. Talon, "Explicit nonlinear model predictive control of the air path of a turbocharged spark-ignited engine," in *2013 IEEE International Conference on Control Applications (CCA)*, pp. 71–77, Aug 2013.

[3] Y. Li, X. Zhou, Y. Hu, and H. Chen, "Air path system control of turbocharged gasoline engine based on fuzzy pid," in *The 27th Chinese Control and Decision Conference (2015 CCDC)*, pp. 941–946, May 2015.

[4] S. Bennett, "Development of the pid controller," *IEEE Control Systems Magazine*, vol. 13, pp. 58–62, Dec 1993.

[5] K. J. Åström and T. Hägglund, *PID controllers: theory, design, and tuning*, vol. 2. Instrument society of America Research Triangle Park, NC, 1995.

[6] K. H. Ang, G. Chong, and Y. Li, "Pid control system analysis, design, and technology," *IEEE Transactions on Control Systems Technology*, vol. 13, pp. 559–576, July 2005.

[7] K. J. Åström and T. Hägglund, "The future of pid control," *Control engineering practice*, vol. 9, no. 11, pp. 1163–1175, 2001.

[8] J. G. Ziegler and N. B. Nichols, "Optimum settings for automatic controllers," *Transactions of ASME*, vol. 64, pp. 759 – 768, 1942.

[9] G. H. Cohen and A. G. Coon, "Theoretical consideration of retarded control," *Transactions of ASME*, vol. 75, pp. 827 – 834, 1953.

[10] P. Cominos and N. Munro, "Pid controllers: recent tuning methods and design to specification," *IEE Proceedings-Control Theory and Applications*, vol. 149, no. 1, pp. 46–53, 2002.

[11] D. E. Rivera, M. Morari, and S. Skogestad, "Internal model control: Pid controller design," *Industrial and engineering chemistry process design and development*, vol. 25, no. 1, pp. 252–265, 1986.

[12] W. Tan, J. Liu, T. Chen, and H. J. Marquez, "Comparison of some well-known pid tuning formulas," *Computers & Chemical Engineering*, vol. 30, no. 9, pp. 1416 – 1423, 2006.

[13] M. Morari and E. Zafiriou, "Robust process control," *Chemical Engineering Research and Design*, vol. 65, pp. 462–79, 1987.

[14] I. G. Horn, J. R. Arulandu, C. J. Gombas, J. G. VanAntwerp, and R. D. Braatz, "Improved filter design in internal model control," *Industrial & engineering chemistry research*, vol. 35, no. 10, pp. 3437–3441, 1996.

[15] C. E. Garcia and M. Morari, "Internal model control. a unifying review and some new results," *Industrial & Engineering Chemistry Process Design and Development*, vol. 21, no. 2, pp. 308–323, 1982.

[16] B. Bequette, *Process control: modeling, design, and simulation*. Upper Saddle River, NJ, USA: Prentice Hall Press, first ed., 2002.

[17] O. Galan, J. A. Romagnoli, and A. Palazoglu, "Real-time implementation of multi-linear model-based control strategiesan application to a bench-scale ph neutralization reactor," *Journal of Process Control*, vol. 14, no. 5, pp. 571 – 579, 2004.

[18] X. Litrico, "Robust imc flow control of simo dam-river open-channel systems," *IEEE Transactions on Control Systems Technology*, vol. 10, pp. 432–437, May 2002.

[19] D. Rupp and L. Guzzella, "Adaptive internal model control with application to fueling control," *Control Engineering Practice*, vol. 18, no. 8, pp. 873 – 881, 2010.

[20] M. Awais, "Application of internal model control methods to industrial combustion," *Applied Soft Computing*, vol. 5, no. 2, pp. 223 – 233, 2005.

[21] S. Qi, H. Wang, and Y. Zhou, "Application of internal model control in the production of pvc," in *2012 IEEE International Conference on Automation and Logistics*, pp. 122–126, Aug 2012.

[22] W. Liang and J. Wang, "Application of internal model control in main steam temperature system," in *Proceedings of the 33rd Chinese Control Conference*, pp. 3511–3514, July 2014.

[23] Z. Xinping, L. Quanshan, W. Huan, W. Wenxin, J. Qibing, and P. Lideng, "The application of model pid or imc-pid advanced process control to refinery and petrochemical plants," in *2007 Chinese Control Conference*, pp. 699–703, July 2006.

[24] C. Cheng and M.-S. Chiu, "Adaptive imc controller design for nonlinear process control," *Chemical Engineering Research and Design*, vol. 85, no. 2, pp. 234 – 244, 2007.

[25] A. Thomasson and L. Eriksson, "Model-based throttle control using static compensators and imc based pid-design," in *IFAC workshop on engine and powertrain control, simulation and modeling*, 2009.

[26] D. Schwarzmann, R. Nitsche, J. Lunze, and A. Schanz, "Pressure control of a two-stage turbocharged diesel engine using a novel nonlinear imc approach," in *IEEE international conference on control applications*, pp. 2399–2404, IEEE, 2006.

[27] S. Hong, I. Park, and M. Sunwoo, "Model-based gain scheduling strategy for an internal model control-based boost pressure controller in variable geometric turbocharger system of diesel engines," *Journal of Dynamic Systems, Measurement, and Control*, vol. 138, no. 3, p. 031010, 2016.

[28] A. Y. Karnik and M. Jankovic, "Imc based wastegate control using a first order model for turbocharged gasoline engine," in *American Control Conference (ACC), 2012*, pp. 2872–2877, IEEE, 2012.

[29] Z. Qiu, M. Santillo, M. Jankovic, and J. Sun, "Composite adaptive internal model control and its application to boost pressure control of a turbocharged gasoline engine," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 6, pp. 2306–2315, 2015.

[30] S. Rohde and M. Philipp, "Combined boost pressure and knock control system for si engines including 3-d maps for control parameters," tech. rep., SAE Technical Paper, 1989.

[31] Z. Qiu, J. Sun, M. Jankovic, and M. Santillo, "Nonlinear internal model controller design for wastegate control of a turbocharged gasoline engine," *Control Engineering Practice*, vol. 46, pp. 105 – 114, 2016.

[32] A. Thomasson, L. Eriksson, O. Leufvn, and P. Andersson, "Wastegate actuator modeling and model-based boost pressure control," *IFAC Proceedings Volumes*, vol. 42, no. 26, pp. 87 – 94, 2009. 2nd IFAC Workshop on Engine and Powertrain Control, Simulation and Modeling.

[33] A. Lekka, M. C. Turner, and P. P. Menon, "Full and reduced order imc anti-windup compensators for a class of nonlinear systems with application to wave energy converter control," in *2013 American Control Conference*, pp. 4861–4866, June 2013.

[34] Y. Arkun and F. Kayihan, "A novel approach to full cd profile control of sheet-forming processes using adaptive pca and reduced-order imc design," *Computers & Chemical Engineering*, vol. 22, no. 7, pp. 945 – 962, 1998.

[35] S. Saxena and Y. V. Hote, "Load frequency control in power systems via internal model control scheme and model-order reduction," *IEEE Transactions on Power Systems*, vol. 28, pp. 2749–2757, Aug 2013.

[36] M. Tavakoli-Kakhki and M. Haeri, "Fractional order model reduction approach based on retention of the dominant dynamics: Application in imc based tuning of fopi and fopid controllers," *ISA transactions*, vol. 50, no. 3, pp. 432–442, 2011.

[37] S. Majhi and D. P. Atherton, "Obtaining controller parameters for a new smith predictor using autotuning," *Automatica*, vol. 36, no. 11, pp. 1651 – 1658, 2000.

[38] E. Jacob and M. Chidambaram, "Design of controllers for unstable first-order plus time delay systems," *Computers & Chemical Engineering*, vol. 20, no. 5, pp. 579 – 584, 1996.

[39] W. S. Lee and J. Shi, "Modified imc-pid controllers and generalised pid controllers for first-order plus dead-time processes," in *7th International Conference on Control, Automation, Robotics and Vision, 2002. ICARCV 2002.*, vol. 2, pp. 898–903 vol.2, Dec 2002.

[40] D. S. Kumar and R. P. Sree, "Tuning of imc based pid controllers for integrating systems with time delay," *ISA Transactions*, vol. 63, pp. 242 – 255, 2016.

[41] P. K. Paul, C. Dey, and R. K. Mudi, "Imc-pid controller for pure integrating process with large dead time," in *Proceedings of The 2014 International Conference on Control, Instrumentation, Energy and Communication (CIEC)*, pp. 76–80, IEEE, 2014.

[42] L. Wang and W. Cluett, "Tuning pid controllers for integrating processes," *IEE Proceedings-Control Theory and Applications*, vol. 144, no. 5, pp. 385–392, 1997.

[43] S. Skogestad, "Simple analytic rules for model reduction and pid controller tuning," *Journal of Process Control*, vol. 13, no. 4, pp. 291 – 309, 2003.

[44] G. F. Franklin, J. D. Powell, A. Emami-Naeini, and J. D. Powell, *Feedback control of dynamic systems*, vol. 3. Addison-Wesley Reading, MA, 1994.

[45] S. Harris and D. Mellichamp, "Controller tuning using optimization to meet multiple closed-loop criteria," *AIChE Journal*, vol. 31, no. 3, pp. 484–487, 1985.

[46] M. Araki and H. Taguchi, "Two-degree-of-freedom pid controllers," *International Journal of Control, Automation, and Systems*, vol. 1, no. 4, pp. 401–411, 2003.

[47] M. V. Kothare, P. J. Campo, M. Morari, and C. N. Nett, "A unified framework for the study of anti-windup designs," *Automatica*, vol. 30, no. 12, pp. 1869–1883, 1994.

[48] P. Danckwerts, "Continuous flow systems. distribution of residence times: P. v. danckwerts, chem. engng sci.2: 1-13, 1953," *Chemical Engineering Science*, vol. 50, no. 24, p. 3855, 1995. Frontiers of Chemical Engineering Science.

[49] A. J. Isaksson and S. F. Graebe, "Brief analytical pid parameter expressions for higher order systems," *Automatica (Journal of IFAC)*, vol. 35, no. 6, pp. 1121–1130, 1999.

[50] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Transactions on Neural Networks*, vol. 1, pp. 4–27, March 1990.

[51] K. Kristinsson and G. A. Dumont, "System identification and control using genetic algorithms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 22, pp. 1033–1046, Sept 1992.

[52] Y. Chen, C. Hu, and K. L. Moore, "Relay feedback tuning of robust pid controllers with iso-damping property," in *Decision and Control, 2003. Proceedings. 42nd IEEE Conference on*, vol. 3, pp. 2180–2185, IEEE, 2003.

[53] T. Liu, Q.-G. Wang, and H.-P. Huang, "A tutorial review on process identification from step or relay feedback test," *Journal of Process Control*, vol. 23, no. 10, pp. 1597 – 1623, 2013.

[54] M. Friman and K. V. Waller, "A two-channel relay for autotuning," *Industrial & Engineering Chemistry Research*, vol. 36, no. 7, pp. 2662–2671, 1997.

[55] M. J. Gilman, "A brief survey of stopping rules in monte carlo simulations," in *Proceedings of the Second Conference on Applications of Simulations*, pp. 16–20, Winter Simulation Conference, 1968.

[56] B. Efron, "Nonparametric estimates of standard error: The jackknife, the bootstrap and other methods," *Biometrika*, vol. 68, no. 3, pp. 589–599, 1981.

[57] "Qube servo-2 device description." `https://www.quanser.com/products/qube-servo-2/`. Accessed: 2019-03-27.

[58] K. J. Åström, T. Hägglund, C. C. Hang, and W. K. Ho, "Automatic tuning and adaptation for pid controllers-a survey," *Control Engineering Practice*, vol. 1, no. 4, pp. 699–714, 1993.

Appendices

A Monte Carlo End Condition

No matter what smart method one chooses to describe an end condition for a Monte Carlo simulation, there will have to be one or multiple assumptions somewhere. Despite using the fancy technique of approximating the standard error of the mean via bootstrapping, I still had to decide to say 1% error is acceptable. Moreover, I had to define the nature of my bootstrapping technique: how many samples over which to take the mean is sufficient? Entire papers can be written on this subject.

Nonetheless, below is the main concept as implemented:

1. Collect the randomly distributed data (for this work, this was PM)
2. Run the following *bootstrapping* routine every so often (past a certain threshold, the number of times I ran this was 5% of that threshold number of samples):
   (a) randomly look at a sample size of values without replacement multiple times, and calculate the mean
   (b) Repeat this many times
   (c) Compute the standard deviation of the data collected in (b), thus approximating the standard error of the mean (SEM).
3. Stop when the SEM $< 1\%$ of the mean of the data in bullet (a).

Below is the code utilized (again, most of the values are arbitrary and come from

repeatedly testing the process until the curve in Figure 15 was obtained):

```
% Requires: (1) data for testing of end condition (PM,GM,BW),
%               assumes no NaN entries
% Modifies: nothing
% Effects: Randomly looks at a sample size of (1), calculates that mean.
%           This is repeated many times, and the std dev of those means is
%           calculated. When this std dev is less than 1% of total mean of
%           (1), return 1, else return 0. Logic: these calcs approximate the
%           standard error of the mean, when SEM <1% mean, you have enough
%           data.

function out=endMonteCarlo(testArray)
    sampleSize=700;
    maxRepeats=2000;
    if numel(testArray)<sampleSize
        error('testArray smaller than sample size');
    end
    meanFull=mean(testArray);
    meanArray=nan(maxRepeats,1);
    for k = 1:maxRepeats
        samples=datasample(testArray,sampleSize,'Replace',false);
        meanArray(k)=mean(samples);
    end
    sem=std(meanArray);

    if sem ≤ 0.01*meanFull
        out = true;
    else
        out=false;
    end
end
```

Not shown here is how this subroutine is called. Essentially, the Monte Carlo simulation runs for 2000 systems straightaway, then every 100 systems after that, this routine is called, until the standard error of the mean passes the threshold. And due to a numerical tolerance imposed when calculating the phase crossover frequency, some systems were unable to be used - meaning I could get 100% of systems calculated but it would take hours, not minutes to run.

B Supplemental Monte Carlo Plots

This section displays every *metric vs* γ plot created from the Monte Carlo analysis. The goal here is to supplement what was discussed in Section 4. Let DA = design approach.

The bandwidth plots for DA2 (and the PM plots for DA3) should be a direct match of the theoretical plots, as these values are calculated by reading directly off the curves (and I use the convention of measuring the 3dB bandwidth closest to the theoretical curve). Any variation for these plots is due to numerical tolerances within Matlab.

Design approach 1 utilizes box-and-whisker plots where the shaded-in region represents the interquartile range, and any explicit data points outside the upper and lower quartiles (whiskers) are larger than 1.5x the interquartile range. The remaining plots are scatter plots, because with DA2/DA3, γ no longer becomes the independent variable.

B.1 First Order Case, DA1

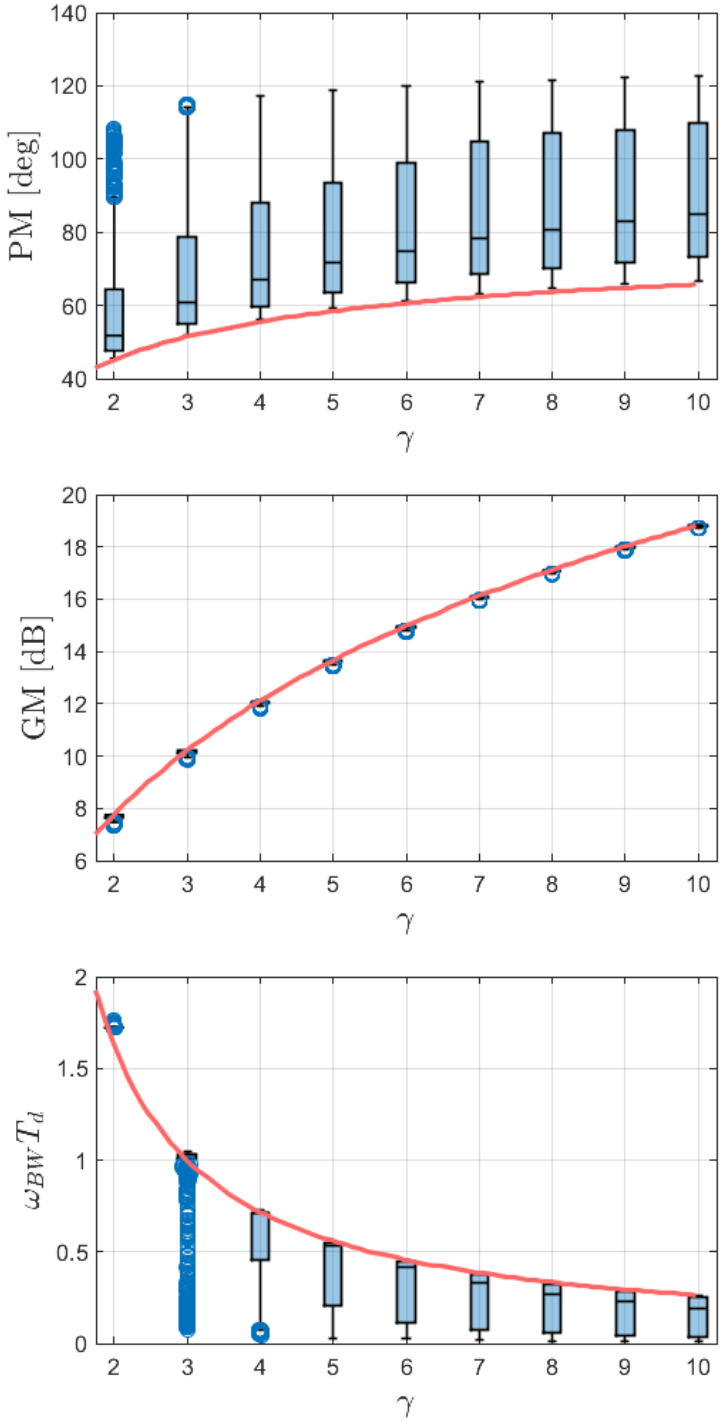

Figure 34: 1st order case, DA1.

B.2 First Order Case, DA2

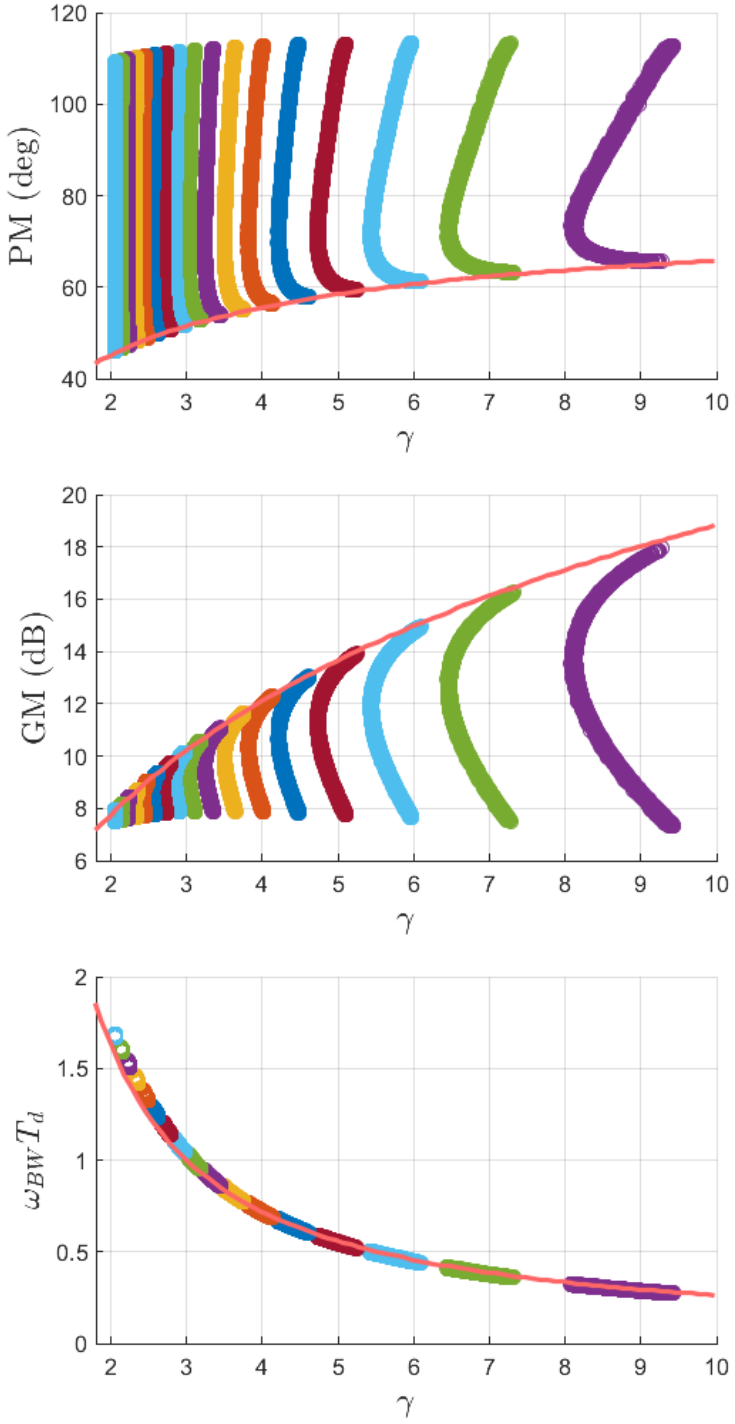

Figure 35: 1st order case, DA2.

B.3 First Order Case, DA3

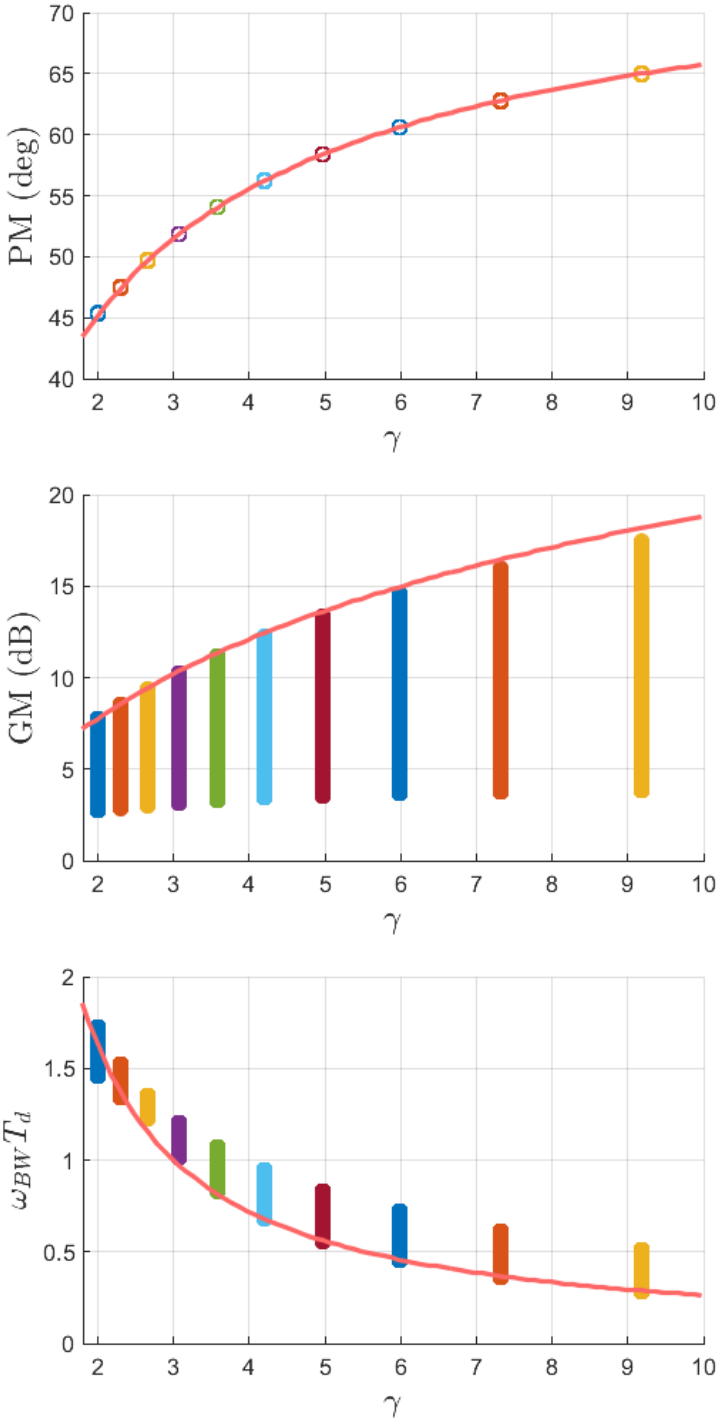

Figure 36: 1st order case, DA3.

B.4 Second Order Case With Real Poles, DA1

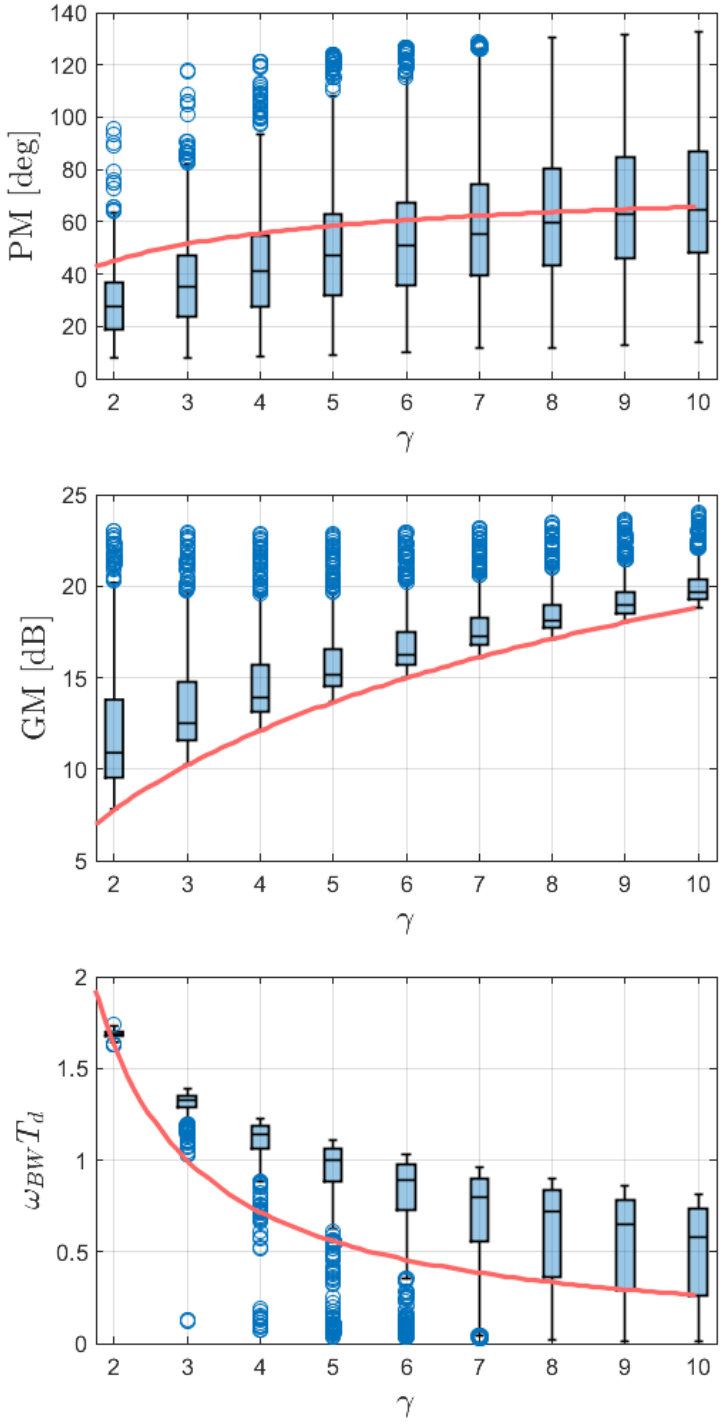

Figure 37: Second Order Case With Real Poles, DA1.

B.5 Second Order Case With Real Poles, DA2

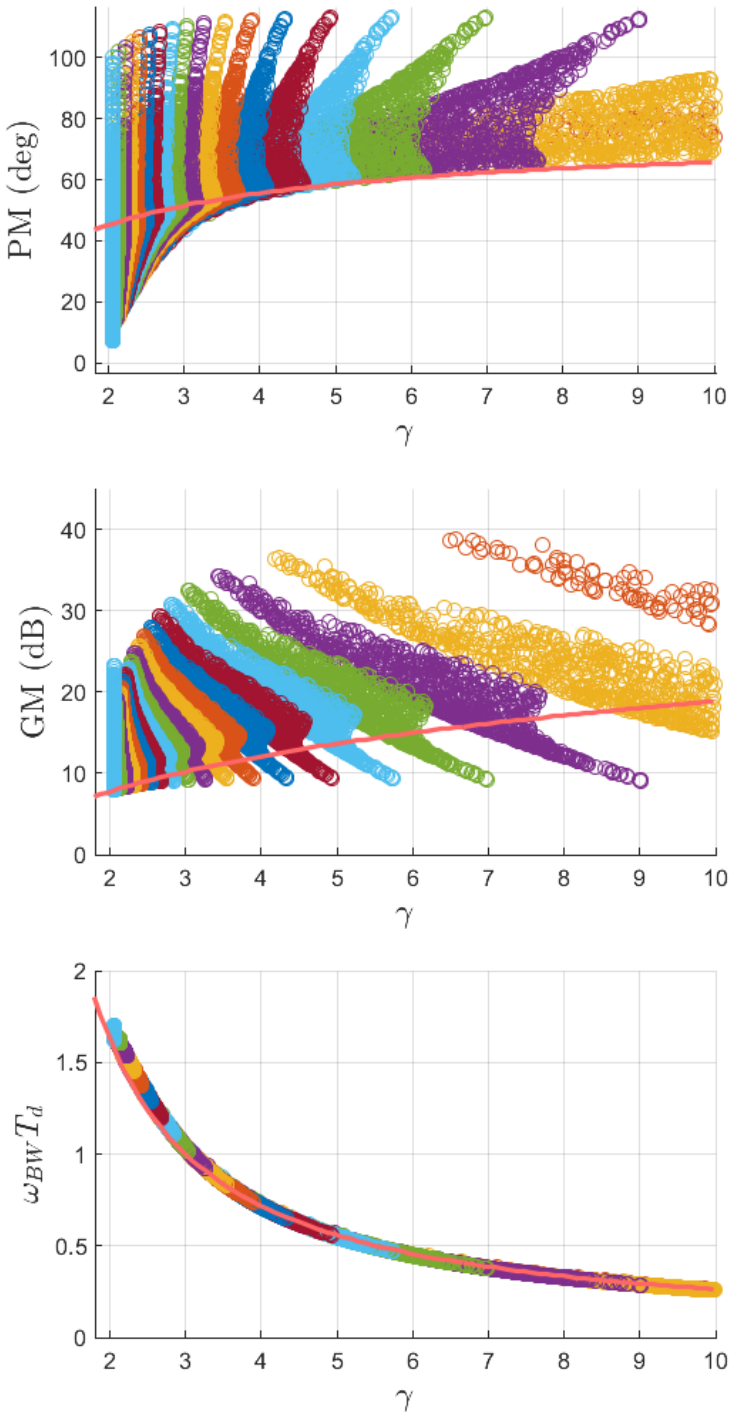

Figure 38: Second Order Case With Real Poles, DA2.

B.6 Second Order Case With Real Poles, DA3

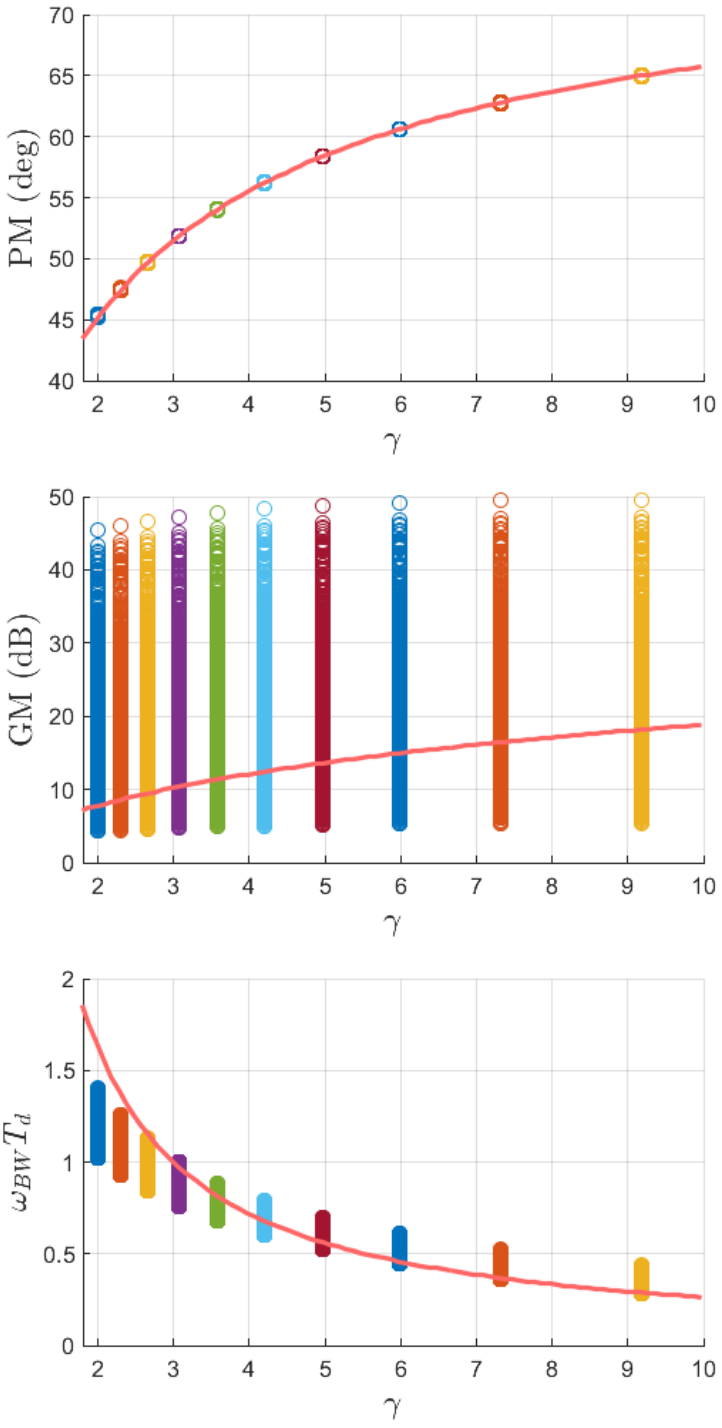

Figure 39: Second Order Case With Real Poles, DA3.

B.7 Second Order Case With Complex Poles, DA1

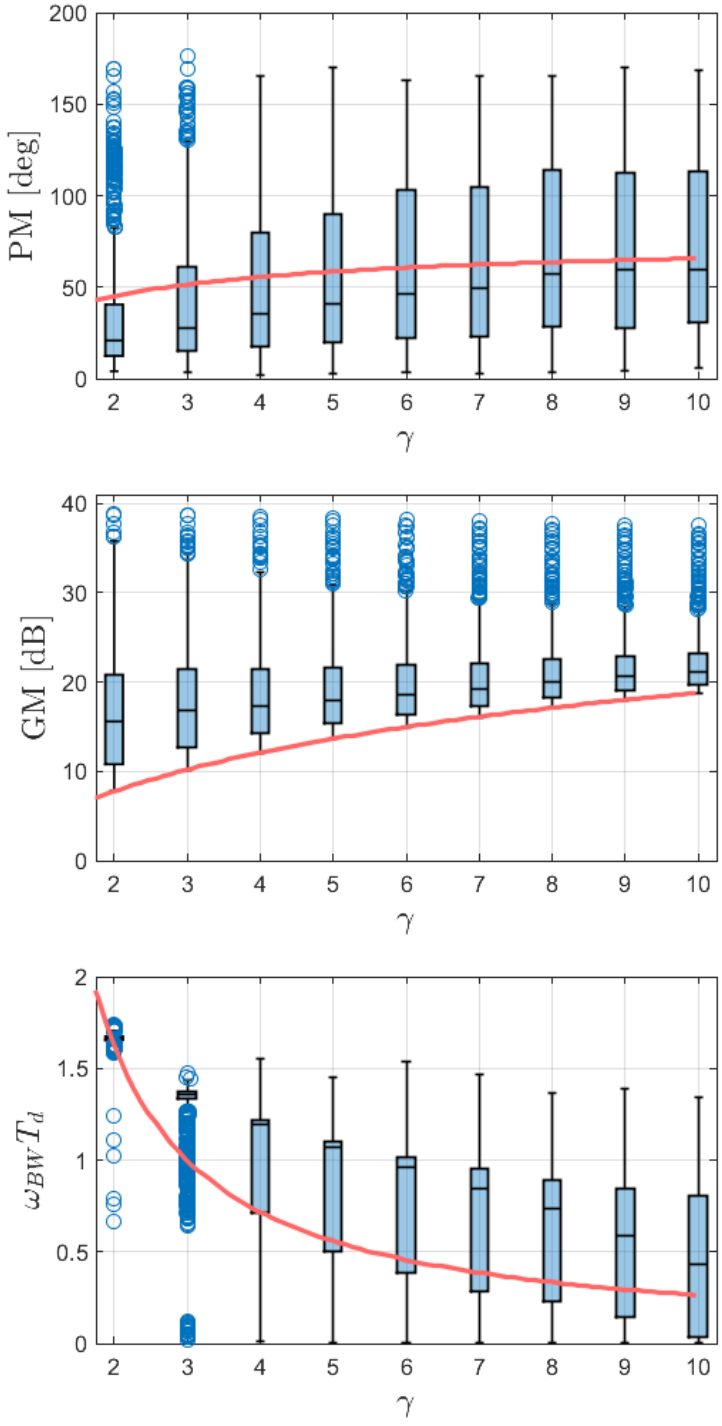

Figure 40: Second Order Case With Complex Poles, DA1.

B.8 Second Order Case With Complex Poles, DA2

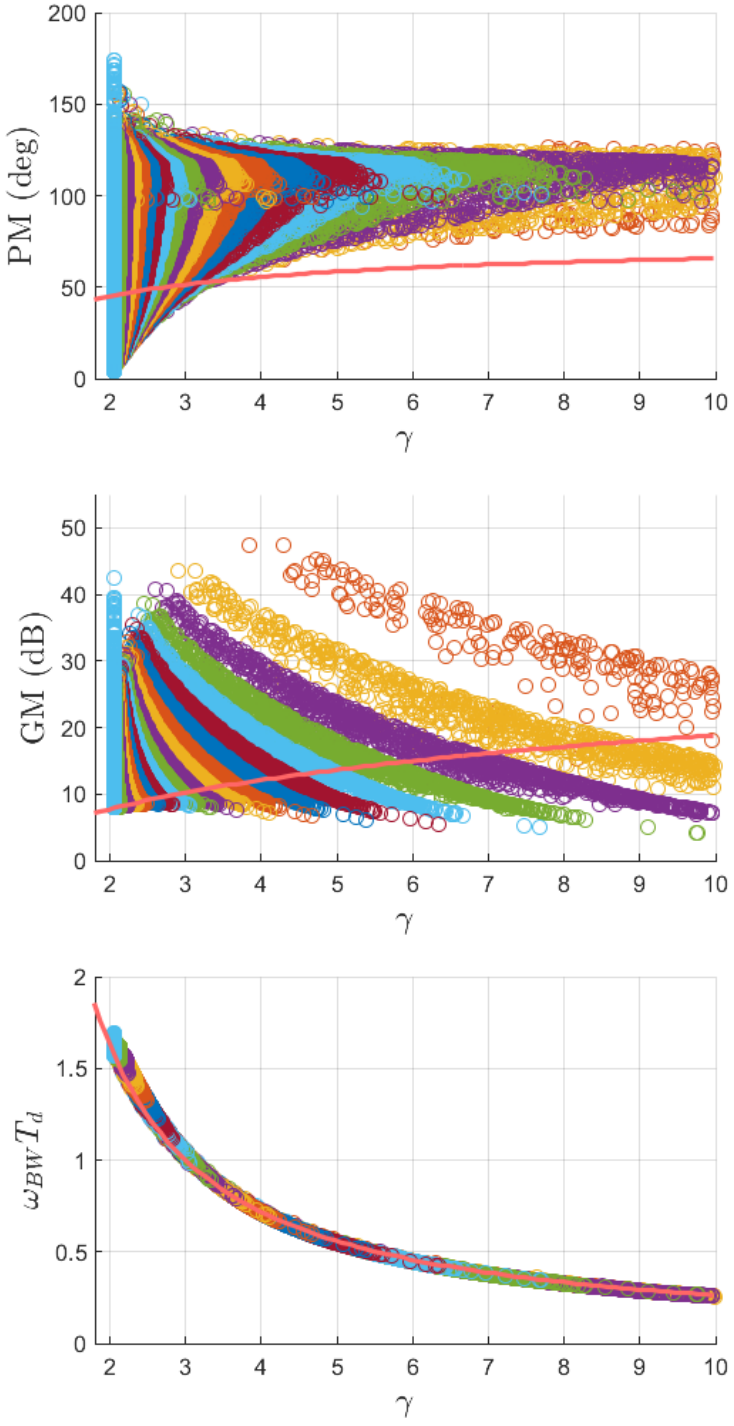

Figure 41: Second Order Case With Complex Poles, DA2.

B.9 Second Order Case With Complex Poles, DA3

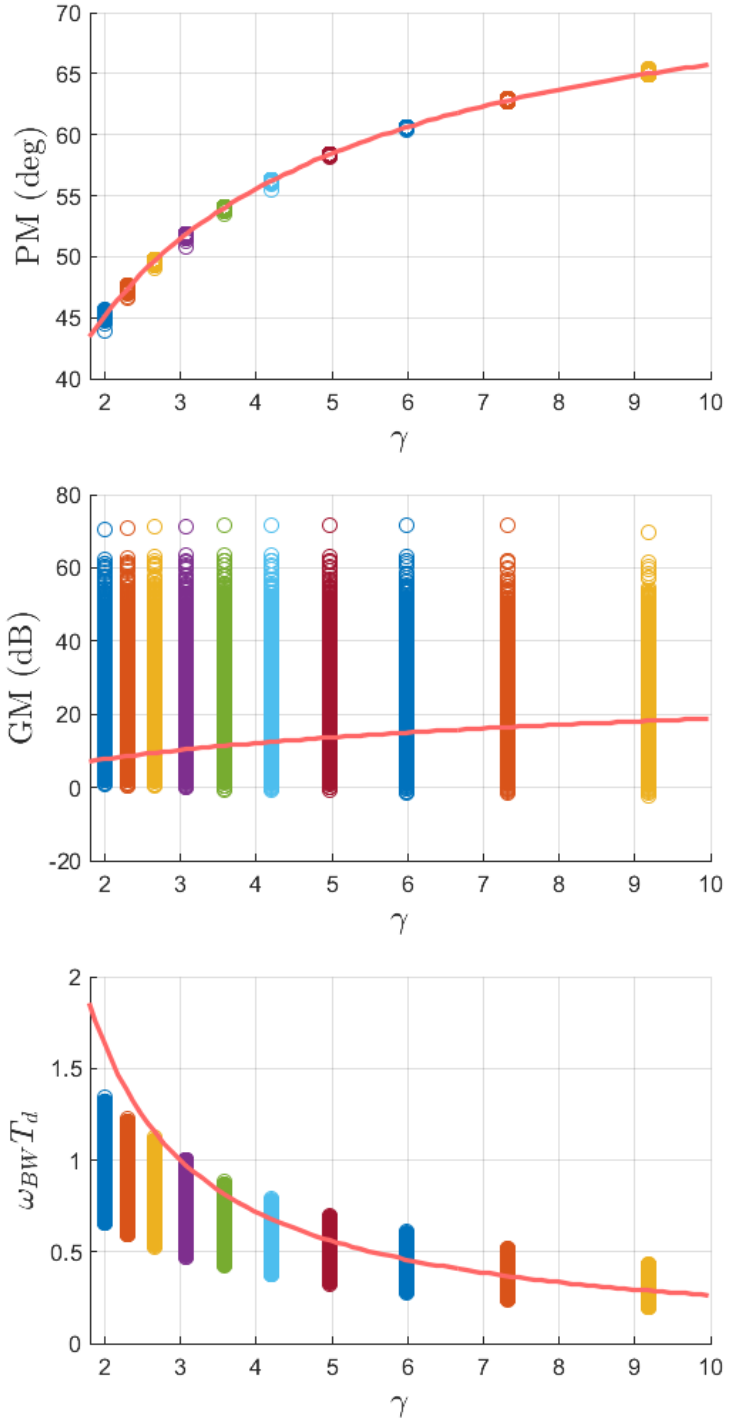

Figure 42: Second Order Case With Complex Poles, DA3.

B.10 Second Order Case With a Zero, DA1

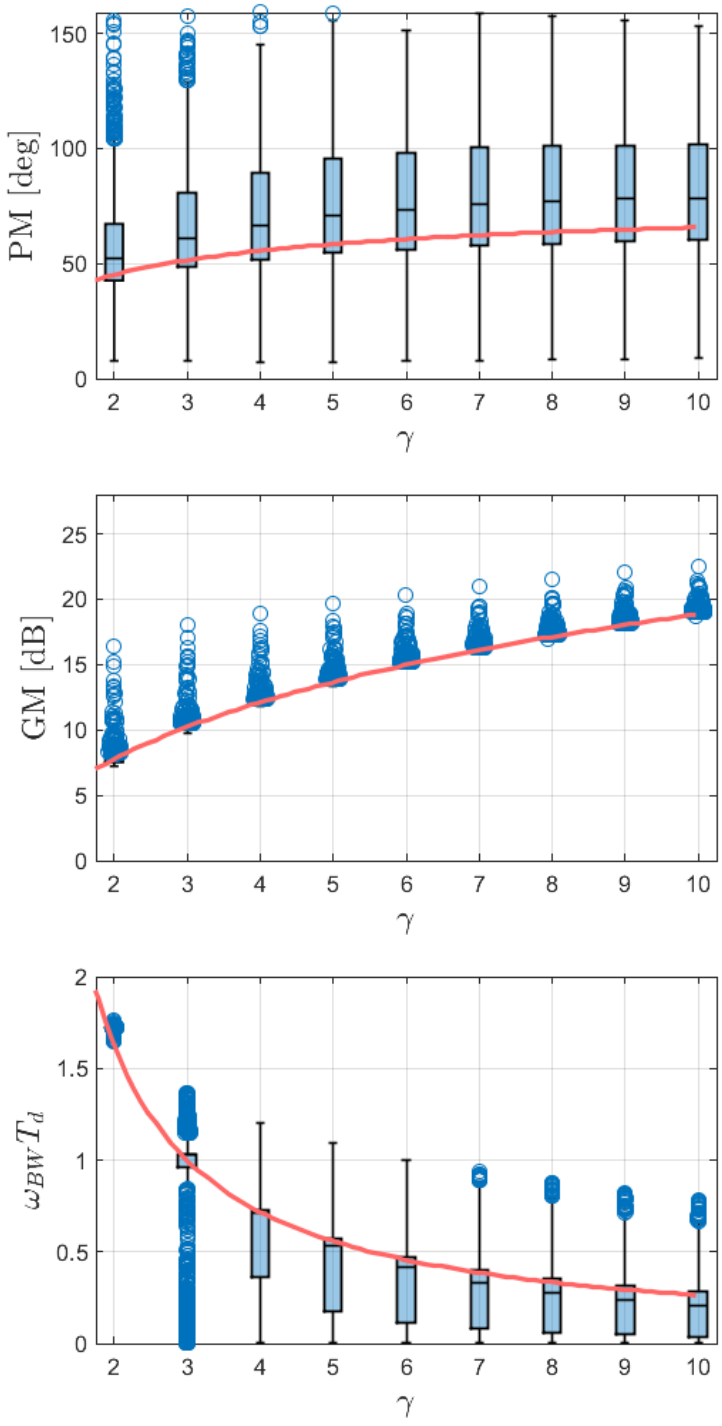

Figure 43: Second Order Case With a Zero, DA1.

B.11 Second Order Case With a Zero, DA2

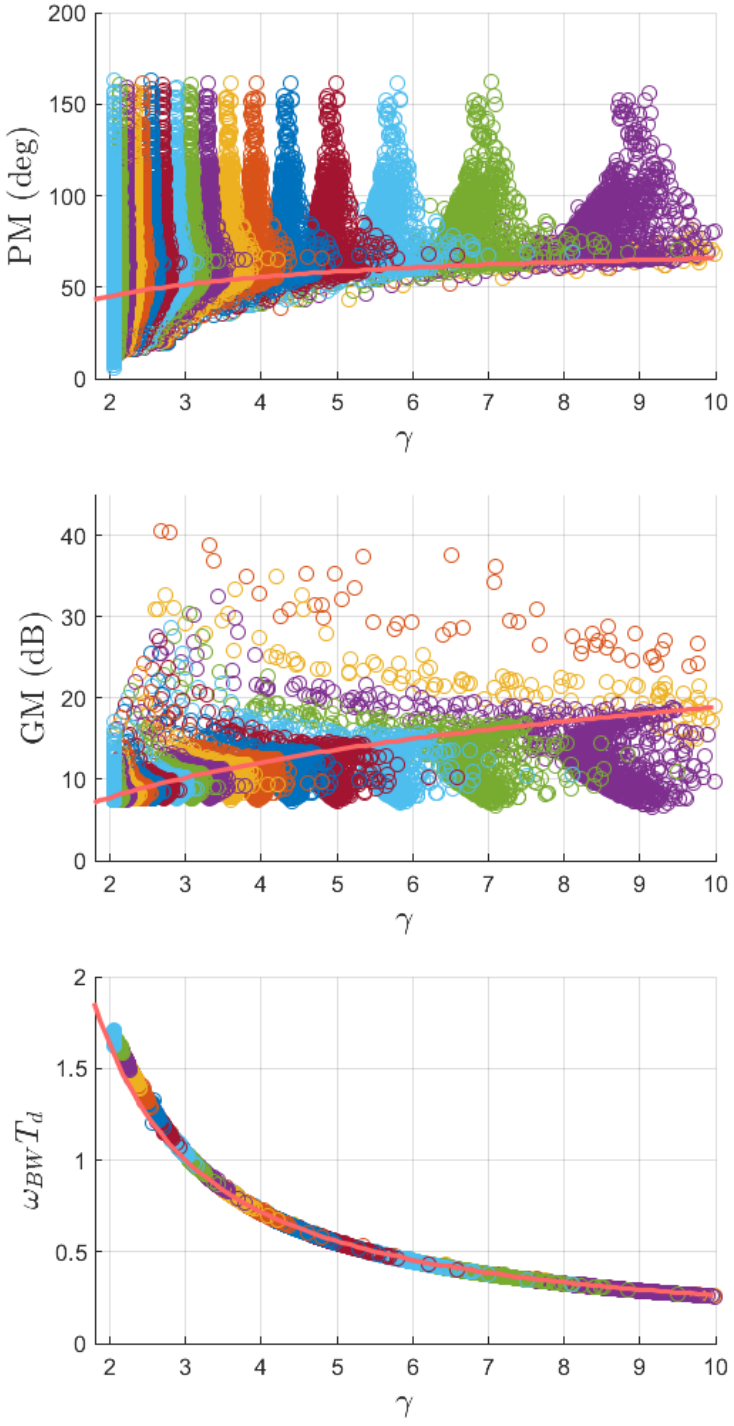

Figure 44: Second Order Case With a Zero, DA2.

B.12 Second Order Case With a Zero, DA3

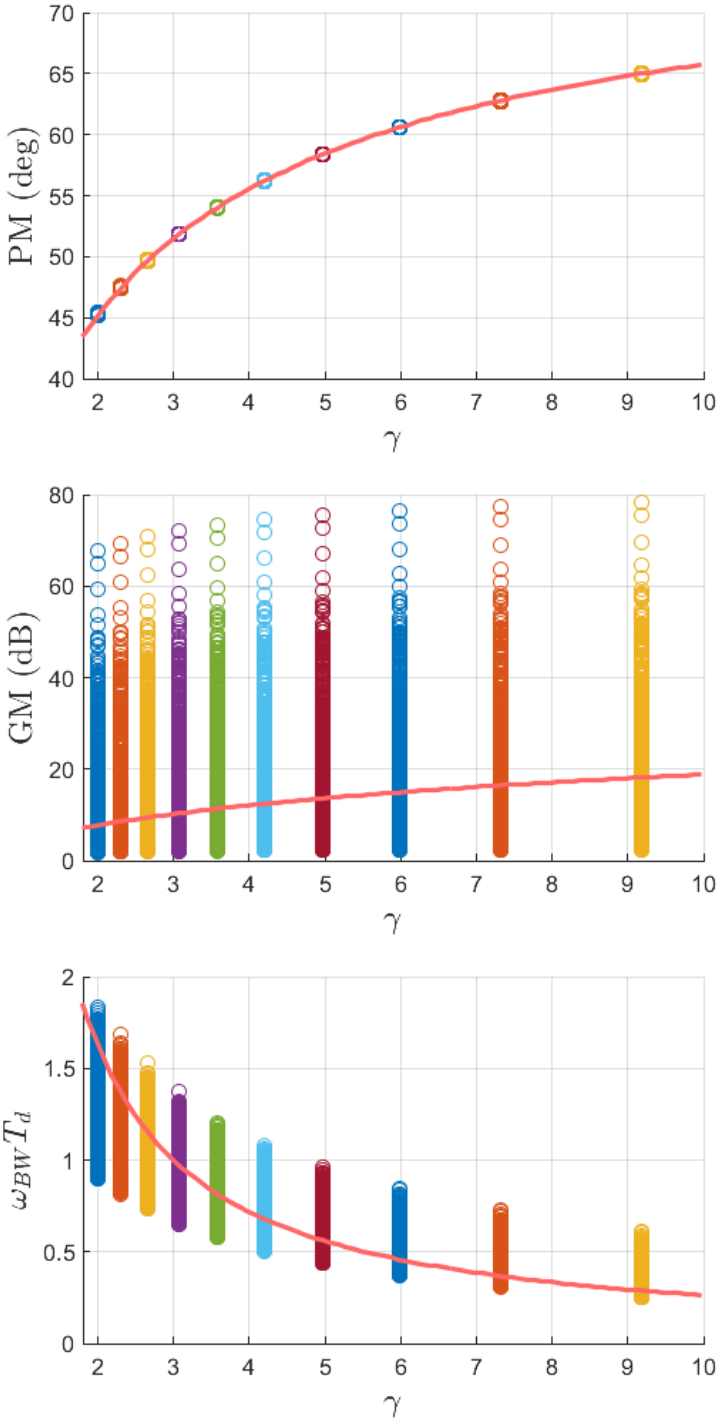

Figure 45: Second Order Case With a Zero, DA3.

C Simulink Implementation of Designs

This section serves to summarize how I gathered and validated the data on the Quanser device. The process is fairly straightforward, and Simulink is extremely user-friendly. The key to this is to remember the three important Quanser blocks: HIL initialize, HIL read, and HIL write. The combination of the read/write blocks can be viewed as the plant, $P(s)$.

In addition, since the device outputs a position by default (in the form of discrete counts around the axis of rotation), this value had to be converted to degrees with the conversion factor, $\frac{360}{2048}$ deg/counts. Finally, since we wish to control speed, this value needs to be modified with a derivative filter. The time constant for this filter was determined through trial-and-error and does a good job of suppressing the noise. The larger the time constant, the more noise suppression, but the slower the response.

C.1 Step Response

Below is the general method of measuring the step response of the closed-loop system. A step size of 1000 deg/s was used in this work.

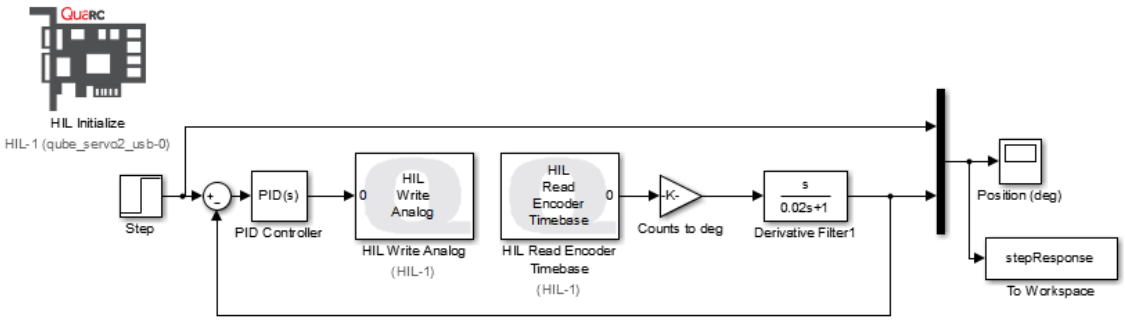

Figure 46: Topology for gathering step response data.

C.2 Measuing Bandwidth Experimentally

Next Figure 47 shows the general idea behind how to measure bandwidth experimentally. Note the only difference is the use of a signal generator instead of the step function at the input. This technique can be seen as picking a value off of the Bode plot, or frequency response.

A sinusoid is input to the closed-loop system at a frequency of the expected 3dB bandwidth, which for design approach 2, will be ω_0. If the output sinusoid has a gain of -3dB, or is $\frac{1}{\sqrt{2}}$ of the input, then the input sinusoid's ω_0 is indeed the bandwidth. A background of the origin and development of 3dB bandwidth can be found in [44].

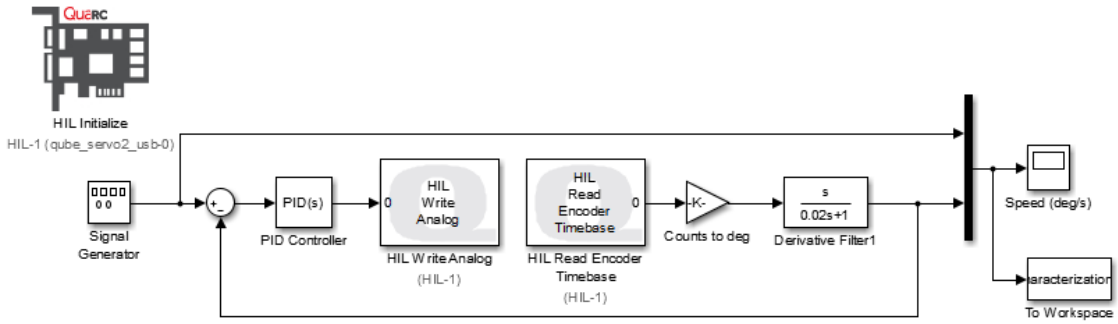

Figure 47: Topology for measuring closed-loop bandwidth experimentally.

C.3 Measuring Phase Margin Experimentally

Finally, Figure 48 demonstrates how phase margin can be measured experimentally. Note how the system is now open-loop, as this is how PM is defined.

Phase margin is defined as:

$$-\left(-180^\circ - T(j\omega_{gc})\right),$$

where ω_{gc} is the gain crossover frequency, or the frequency where the gain is 1 (0dB), and $T(j\omega)$ is the open-loop transfer function [44]. For my analysis, this will be $P(j\omega)C(j\omega)$, where P is the Quanser device and C is the PID controller.

Since design approach 3 effectively assigns the gain crossover frequency, all I have to do is measure the system's phase at this frequency, ω_0, and add 180 degrees. Again, to do this, a sinusoid was used at the input and the phase was measured by the output sinusoid's phase.

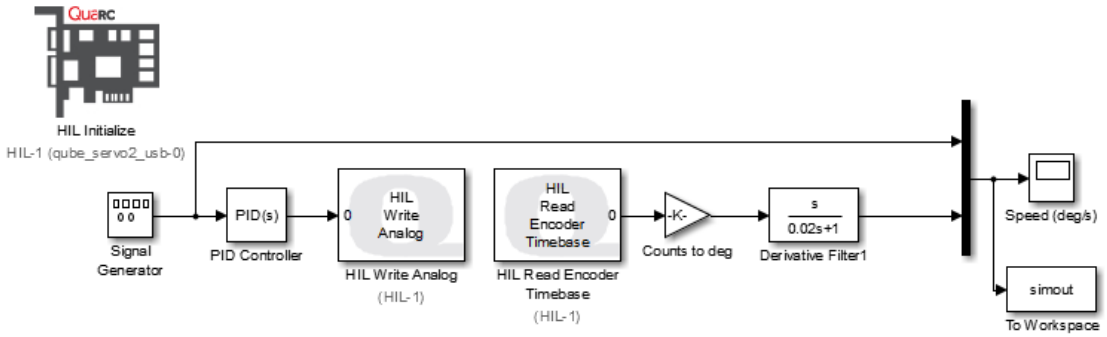

Figure 48: Topology for measuring phase margin experimentally.