General Predictive Control: An Investigation of Perceived Limitations

Thesis presented for the degree of Master of Science

in the Department of Electrical Engineering

University of Cape Town

South Africa

by

Christopher Domenic Cecchini

February, 2012

I know the meaning of plagiarism and declare that all the work in this document, save for that which is properly acknowledged, is my own.

Christopher Domenic Cecchini

2012

Table of Contents

List of Figures

Nomenclature

Interpretation of Mathematic Operators

$\rightarrow$	'tends to'
$\Longrightarrow$	'implies'
$(.)_{i,j}$	Denotes the coefficient of an arbitrary matrix located at row i and column j
$(.)_i$	Denotes the i^{th} row of an arbitrary matrix or vector
z	The forward shift operator such that $f(t) * z = f(t + T_s)$
z^{-1}	The backward shift operator such that $f(t) * z^{-1} = f(t - T_s)$
$\mathbf{Z}$	Z-transform operator
$\mathbf{Z}_m$	Modified z-transform operator

Acronyms/Definitions

GPC	General Predictive Control
MPC	Model Predictive Control
PID	Proportion, Integral, Differential
PZC	Pole/Zero Cancellation
RK4	Runga-Kutta 4th Order
SPGPC	Smith Predictor Generalized Predictive Control
ZOH	Zero Order Hold function

$(t + k)$	This denotes a time in the past or future that is an integer (k^{th}) multiple of the discrete sampling time offset to the present time, t
na	The degree of the pole polynomial in the CARIMA model
nb	The degree of the zero polynomial in the CARIMA model
dB	Decibel - the magnitude of the gain with regard to a Bode plot

DC	Designating a signal or property taken at a frequency of zero
Plant	An open-loop system with no control scheme implemented also known as a physical plant

Polynomials/Transfer Functions

$D(z^{-1})$	The disturbance rejection model utilised in Sinusoidal Setpoint Tracking
$g(s)$	The transfer function representing the physical plant
$g_h(z)$	The transfer function representing the digital model of the plant with the step invariant transform applied
$R(z^{-1})$	Polynomial used to calculate closed loop pole positions for GPC
$S(z^{-1})$	Polynomial used to calculate closed loop pole positions for GPC
$T(z^{-1})$	A prefilter design polynomial in the CARIMA model

Signals

$u(t)$	The input at time t
$w(t)$	The setpoint at time t
$y(t)$	The output at time t
$\mathbf{u}$	A column vector containing optimal incremental input values
$\mathbf{w}$	A column vector containing setpoint values
$\mathbf{y}$	A column vector containing predicted output values

Symbols

α	The slope of the ramp setpoint in relation to sampling time
β	Open loop system gain

δ The error squared weighting variable

Δ The difference operator, $1 - z^{-1}$

ϵ A small number just greater than zero

λ The input squared weighting variable

ϕ_c Designation for a closed loop characteristic equation

τ The total system dead time

θ The non-integral (fractional) portion of the total system dead time, τ, considering integral dead time to be an integral multiple of the sampling time, T_s

Variables

a The inverse of the plant settling time

d The number of integral dead time samples present in a digital system

j The number of steps into the future the prediction model calculates

N_1 The start of the Prediction Horizon

N_2 The end of the Prediction Horizon

N_u The control horizon

t The variable designating continous time

T_s Digital sampling interval

s Laplace variable

Acknowledgments

Firstly to Prof Martin Braae for his guidance and assistance, without which this dissertation would not have been possible. To my co-supervisor, Sammy Tsoeu, for his invaluable insight into the inner workings of MPC. To my office mates and friends, particularly Dave Moore, Thabo Koetje and Tony Ho, for your guys help and support. A special mention must be made to Douwe Egberts for their unbelievably tasty coffee, keeping me up and going during the long nights.

Finally, to my wife, Lee-Anne, for her relentless determination and (often) less-than-gentle encouragement. You are my pillar.

Abstract of the Thesis

General Predictive Control: An Investigation of Perceived Limitations

by

Christopher Domenic Cecchini

Master of Science in Electrical Engineering

University of Cape Town, South Africa, 2012

General predictive control is a well-known subset of the Model-based Predictive Contollers and has been applied to many industrial and commercial applications. The concepts of step, ramp and sinusoidal setpoint tracking, pole zero cancellation, non-minimum phase systems and dead time are some of the difficulties presented to the modern control algorithm. This dissertation investigates how GPC handles these difficulties and what modifications to the basic algorithm have been applied in the pursuit of stable, fast and efficient control.

CHAPTER 1

Introduction

Automatic control has been conceptualized for over 2 000 years with one of the earliest documented inventions being a water clock created by Ctesibius of Alexandria [9]. Since then, new control techniques have been coming to the fore, one of the earlier notable techniques being classic PID control arising circa 1890 in governor designs [5]. Many others have followed including linear quadratic, sliding-mode and state-based control. Model Predictive Control (MPC), the class of controllers utilizing future predictions to calculate the optimal current control action, is one of the more recent algorithms [10]. General predictive control (GPC) is a subset of MPC that inherits the MPC architecture.

Control theory aims to realize autonomous control of a system, that is the output of the system tracks a desired setpoint without the need for human intervention. Some systems are more difficult to control than others. Examples of unwanted system properties include inherently unstable systems, non-minimum phase systems including those with integral and fractional dead time, ringing, signal constraints and pole-zero cancellation. Depending on the control architecture applied to the problem, some of these problems are known to be easily solved. For example, the Smith predictor is used to compensate for dead time [25] however PID control is known to have difficulties counteracting this phenomenon.

A comprehensive survey of available literature revealed that GPC has its own potential strengths and weaknesses. For example, GPC can integrate constrained signals and integral dead time into the control algorithm with ease [10, 25]. Some of these strengths

are well known and stem from the fundamental properties of the algorithm but there are other areas that have been studied to a lesser degree.

The literature survey further revealed a list of potential problems for GPC. These include; asymptotic tracking of ramp and sinusoidal setpoints, open-loop unstable and non-minimum phase systems, pole/zero cancellation, non-integral, variable and unknown dead time, model mismatch, non-linear systems and multivariable systems. After an in-depth study of the information gathered during the initial survey, the following potential problems were chosen for analysis in this dissertation:

1. Asymptotic Tracking of Step, Ramp and Sinusoidal Setpoints
2. Pole/Zero Cancellation
3. Integral and Non-Integral Dead time
4. Unstable Open-Loop and Non-Minimum Phase Systems

Problem 1 was chosen as it was not a well documented problem with regard to ramp and sinusoidal setpoints. Problem 3 and 4 were chosen due to the relatively high frequency that these types of systems are encountered in industry. Problem 2 was chosen as it is regarded to be an issue for GPC and forms a basis for the analysis of Problem 4.

A comprehensive review of these topics is presented in Chapter 2. Solutions have been proposed in the literature for all problems under question. These solutions are critiqued and analyzed.

Asymptotic tracking of setpoints is considered first. There is a substantial amount of information available relating GPC and tracking of constant setpoints (or step functions) [14, 33, 34, 36, 20]. However there is very little readily-available information on asymptotic tracking of ramp and sinusoidal setpoints. A number of researchers have considered these setpoints and there have been recent advances [10, 35] in those areas although the

practical applications of these algorithms are still limited. In Chapter 3, the standard GPC algorithm is adapted for tracking ramp setpoints asymptotically, presented and verified through a simulated example. This algorithm is compared mathematically to a proposed solution for setpoint tracking in [10].

Pole/Zero cancellation (PZC) is the second classic control problem considered although it is typically dependent on the relationship between the controller and the plant. It is well known that a problem with PZC is rooted in input disturbance rejection and to a lesser degree robustness to model mismatch [24]. An analysis of the GPC algorithm is conducted in Chapter 4 to verify if PZC occurs in GPC and if so, under what conditions and what solutions are available to allow optimal control of the plant. Additionally, the chapter will serve as a technical introduction to Chapter 5.

Chapter 5 investigates the effects of using the GPC algorithm for systems with integral and non-integral dead time. An extension of this problem is variable or unknown dead time. Variable and unknown dead time has been shown to be handled effectively in [36] and will not be covered in this dissertation. Integral dead time is handled effectively in GPC [13, 10, 25]. The case of non-integral dead time has solutions proposed and verified through example in [10]. The implications of non-integral dead time are investigated regarding the zero created during the generation of the digital model of the plant.

Non-minimum phase systems (NMP), which include the class of unstable open-loop systems, are a classic control problem. They are one of the first problems considered when analyzing a control algorithm as they are associated with the plant and not with a secondary system, such as the controller, or a relationship between the two. The seminal works regarding GPC purported to be able to control this class of systems but the stability and transient dynamics are dependent on the choice of prediction and control horizons [13, 22]. It is shown that by choosing the prediction and control horizons carefully, GPC can be made to resemble the well-known dead-beat controller [14] and that concept is

extended into a pole-placement algorithm [12]. This enables practical control of such systems using the GPC algorithm as a basis.

Conclusions have been drawn and are presented in Chapter 6. Based on these conclusions, Chapter 6 also discusses potential future work.

1.1 Simulation Notes and Methods

This dissertation makes use of MatLab as a simulation tool. All simulations were programmed independently of a preconfigured toolbox. Unless otherwise stated, the following conditions apply to simulations:

- Systems are simulated in discrete time and the output calculated using the RK4 algorithm to simulate continuous time.
- All plants simulated are controllable and stabilizable.

This dissertation makes reference to time-dependent signals containing offset integer increments e.g. $f(t+k)$ for some integer k. Unless otherwise stated, this is to be taken as a time offset of kT_s as this is the mathematically correct representation.

The algorithms used to perform research and provide simulated data for the dissertation are the original work of the author. The following is a list of algorithms developed:

- Standard GPC
- Standard GPC for Systems with Integral/Non-Integral Dead Time
- Standard GPC for Ramp Setpoints
- Standard GPC for Sinusoidal Setpoints
- Smith Predictor GPC for Step/Ramp Setpoints

- Augmented GPC with Appropriate $D(z^{-1})$
- Calculating the Closed Loop Pole Positions for Varying λ

CHAPTER 2

Literature Review

Information found in a review of available literature in the fields of setpoint tracking, pole zero cancellation, dead time and generalized predictive control is presented below. This information aims to provide the reader with a foundation for the investigations presented in proceeding chapters.

2.1 Asymptotic Tracking of Step, Ramp and Sinusoidal Setpoints

The standard GPC algorithm has the ability to include any future knowledge of the setpoint trajectory into the controller to yield offset free tracking. While this works well for step setpoints and ramp setpoints to a degree, problems arise with sinusoidal setpoints due to inaccuracies inherent in the optimal predictor [30].

Step responses are the most common type of setpoints encountered in industry. Examples include controlling of temperature in a reactor, fluid level in a distillation column, voltage and frequency regulation in uninterruptible power supplies, hovering altitude of a helicopter, etc. GPC has been applied to a variety of different systems including oscillators, motors, computer networks, chemical reactors and the administration of anaethesia during surgery [14, 33, 34, 36, 20, 23]. In all of these systems, a step setpoint has been applied and successful control has been established.

Ramp setpoints are far less common. However, ramp setpoint tracking is becoming

utilized more frequently as new technologies become available [4]. Certain systems require a smooth transition between setpoint levels that is more readily viewed as a ramp. Fast changes in temperature, for example, can cause stresses to form in a reactor or pipe which would reduce the equipment's functional lifetime so a slower temperature change is required. A ramp setpoint-tracking algorithm, and its application to a simulated system, is presented in [10]. It utilizes the GPC algorithm's ability to inject setpoint data into the optimal control increment calculation to compensate for a ramp setpoint.

Sinusoidal tracking is typically found in power electronics [28] such as voltage tracking in buck/boost converters but also in aerospace applications [19]. It has largely been ignored in the predictive control community until recently [35] due to lack of practical applications driving the research. In [30], a potential problem is indicated regarding GPC and sinusoidal setpoint tracking: the controller contains prediction mismatch. It is shown in [35] that, as per the Internal Control Model Principle [15], to track a sinusoidal signal asymptotically, the generator of that signal must be included in the control structure. Utilizing that knowledge, the algorithm proposed in [35] was shown to track sinusoids asymptotically.

2.2 Pole Zero Cancellation

Pole/zero cancellation is a phenomenon by which the poles or zeros of a system are canceled numerically by the zeros or poles, respectively, of a serially connected system. In classical control this is typically the zeros of a controller canceling the poles of the plant to introduce the more favourable dynamics of the controller poles.

There are many instances when PZC has been used as an effective tool in controlling a system. One of the main applications is in adaptive controllers [11, 1, 32]. However these adaptive controllers have problems with input disturbance rejection because the output response, with regard to an unmeasurable input disturbance, follows the dynamics of

the open-loop plant poles. One method of reducing the effect of unmeasurable input disturbances is to alter the PZC to use pole shifting [32].

Pole shifting reduces the system's robustness to model mismatch [18]. The problem of input disturbance rejection is not limited to adaptive controllers and has been reported in prefixed controllers [18].

2.3 Dead Time

Digital control is a necessity for modern control techniques. Analogue components are impractical, if not impossible, to use in constructing a GPC controller, for example.

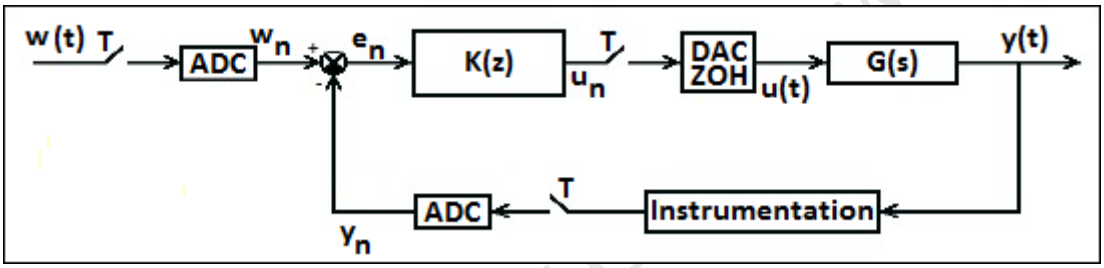

Figure 2.1: The Digital Control Scheme

Fig. 2.1 shows how the digital controller replaced the analogue controller with the addition of samplers, ADCs and DACs [8]. Controlling a real plant with a digital controller, however, required new modeling techniques to convert the real (s-plane) plant model into a digital (z-plane) model [8]. These modeling techniques included the z-transform and the modified z-transform to convert a model in s to a model in z, the former applying to systems with no or integral dead time and the latter being applied to systems with non-integral (fractional) dead time. A table giving some of the common z-transforms and modified z-transforms can be found in most digital control textbooks.

GPC has been applied to systems with integral, non-integral, variable and unknown dead time [37, 10, 13, 14, 12]. All sources have reported that the standard algorithm,

with some minor changes, can control systems with all types of dead time. However, no mention has been made of the conditions under which a system with non-integral dead time can become unexpectedly unstable depending on the chosen sampling time. This topic will be further covered in Chapter 5.

2.4 Non-minimum Phase Systems

Continuous-time, non-minimum phase systems are systems that contain one or more poles or zeros with positive, non-zero real parts. In this dissertation, the term 'non-minimum phase system' defines a plant with one or more zeros that have positive, non-zero real parts. A plant with poles that contain positive, non-zero real parts will be referred to as 'open-loop unstable'.

In terms of a digital system, a 'non-minimum phase system' is that system with one or more zeros that have a magnitude of greater than unity i.e. a zero that lies outside the unit circle in the z-plane. Similarly an 'open-loop unstable' system is that system with one or more poles that have a magnitude of greater than unity i.e. a pole that lies outside the unit circle in the z-plane.

GPC was purported to be able to control these classes of systems in [13]. One of the main problems with controlling non-minimum phase and unstable open-loop systems in the past was the lack of computational power for solving the GPC algorithm with large prediction horizons [22].

An algorithm is proposed in [22] and reinforced in [29, 10] that provides a way to guarantee stability for the class of systems under discussion in addition to others. This algorithm, termed Stable Generalized Predictive Control, makes use of an internal control loop to stabilize the system before constructing the GPC control algorithm around it.

The difficulties with controlling NMP systems have been circumvented and a solution

was found to be in place in the form of an extension to the GPC algorithm. This adaption makes use of dead beat and pole-placement techniques to force the closed-loop poles to predetermined positions [12, 14].

2.5 General Predictive Control

The most defining characteristic about MPC and consequently GPC is its implementation of Receding Horizon Theory. An example of this theory is of a person driving a car. The driver is able to see a certain distance ahead, noting bends or obstacles, and uses this information to make decisions on when to turn the steering wheel and by what degree. The field of vision of the driver moves as the car drives forward. In the same way, GPC predicts, at each time step, the output of a system for a number of time steps ahead to calculate a series of optimal control actions but only applies the first control action.

The computational burden of implementing a GPC controller was relatively high at the time the algorithm was being proposed and investigated. This hindered the ability to use large prediction horizons as the matrix inversion required exponentially increasing computations to solve. This was only a temporary setback as contemporary computers are able to perform many more computations per second than previously. This, with the use of precalculation, has allowed GPC to be used in many areas previously not available [6, 21].

The GPC algorithm has been successfully implemented in both the CARIMA (Controlled Auto-Regressive Integrated Moving-Average)- and State Space models [13, 27] however the CARIMA-based model is used in this dissertation. It takes the form:

$$A(z^{-1})y(t) = B(z^{-1})z^{-d}u(t-1) + \frac{T(z^{-1})}{\Delta}e(t) \tag{2.1}$$

where $\Delta = (1 - z^{-1})$ and the polynomials are defined by:

- $A(z^{-1}) = 1 + a_1 z^{-1} + ... + a_{na} z^{-na}$
- $B(z^{-1}) = b_0 + b_1 z^{-1} + ... + b_{nb} z^{-nb}$
- $T(z^{-1}) = 1 + t_1 z^{-1} + ... + t_{nt} z^{-nt}$

The time-varying signals $u(t)$, $y(t)$ and $e(t)$ are the plant input, output and a zero-mean white noise respectively. The backward shift operator, z^{-1}, is defined such that $z^{-1} f(t) = f(t-1)$. The poles of the plant are defined by the roots of $A(z^{-1})$ and the zeros by the roots of $B(z^{-1})$. The plant dead time is defined by z^{-d}. The polynomial, $T(z^{-1})$, can be defined, depending on the disturbance type being modeled [13, 14, 12, 10], as unitary in the case of zero mean white noise, a disturbance polynomial, often termed 'coloured noise' in literature, indicating a moving average error or a design polynomial used in conjunction with the 'coloured noise' polynomial. This dissertation will make use of the last definition; $T(z^{-1})$ is a design polynomial used to filter signals as shown below.

A full review of the GPC prediction model, control law and closed-loop relationship can be found in the seminal works, [13, 14, 12] as well as most recent MPC/GPC textbooks [10]. This dissertation will provide a summary of these topics.

2.5.1 The Prediction Model

To be able to utilize a receding horizon, a method of prediction must be used that is based on current and past values of the relevant signals. This method is found in recursion of a form of the Diophantine equation. There are other methods available to achieve the prediction model [2, 10] however this dissertation will not cover them.

The Diophantine equation, for a j-step ahead prediction, and the consequent generation of the prediction equation is given in [13, 10] but will be shown here for completeness:

$$T(z^{-1}) = E_j(z^{-1})\Delta A(z^{-1}) + z^{-j}F_j(z^{-1}) \tag{2.2}$$

where E_j and F_j are uniquely defined with degrees $j-1$ and na respectively and $j \in \mathbb{Z}$, $j > 0$. In Appendix A, it is shown that there are unique solutions for E_j and F_j for a given $A(z^{-1})$ and j. With this result in mind, the prediction model can be generated.

Consider (2.1) for a time in the future, $t+j$.

$$A(z^{-1})y(t+j) = B(z^{-1})u(t+j-d-1) + \frac{T(z^{-1})}{\Delta}e(t+j) \tag{2.3}$$

Multiplying both sides of the equation by $E_j(z^{-1})\Delta$ yields:

$$E_j(z^{-1})\Delta A(z^{-1})y(t+j) = E_j(z^{-1})B(z^{-1})\Delta u(t+j-d-1) + E_j(z^{-1})T(z^{-1})e(t+j) \tag{2.4}$$

To note here, as the degree of $E(z^{-1})$ is $j-1$, all the disturbance/noise terms in the model are in the future. That being said, the optimal future prediction assumes that all expected future disturbance/noise terms are zero. Therefore the optimal j-step ahead prediction can be represented as:

$$E_j(z^{-1})\Delta A(z^{-1})y(t+j) = E_j(z^{-1})B(z^{-1})\Delta u(t+j-d-1)$$

Substituting $E_j(z^{-1})\Delta A(z^{-1}) = T(z^{-1}) - z^{-j}F_j(z^{-1})$ from (2.2) yields:

$$
\begin{aligned}
(T(z^{-1}) - z^{-j}F_j(z^{-1}))y(t+j) &= E_j(z^{-1})B(z^{-1})\Delta u(t+j-1) \\
T(z^{-1})y(t+j) &= z^{-j}F_j(z^{-1})y(t+j) + E_j(z^{-1})B(z^{-1})\Delta u(t+j-1) \\
T(z^{-1})y(t+j) &= F_j(z^{-1})y(t) + E_j(z^{-1})B(z^{-1})\Delta u(t+j-1)
\end{aligned}
$$

dividing through by $T(z^{-1})$, the equation simplifies to:

$$
y(t+j) = F_j(z^{-1})y_f(t) + E_j(z^{-1})B(z^{-1})\Delta u_f(t+j-d-1) \tag{2.5}
$$

with $y(t)$ and $u(t)$ represented as filtered versions of themselves such that $y_f(t) = \dfrac{y(t)}{T(z^{-1})}$ and $u_f(t) = \dfrac{u(t)}{T(z^{-1})}$.

As the degree of $E_j(z^{-1})$ is $j-1$, each iteration of the prediction equation will contain, at least, components of $\Delta u(t)$ based at a current or future time but may also contain terms dependent on past values of $\Delta u(t)$. To obtain the prediction equation explicitly in terms of past and future signals, consider the substitution $G_j(z^{-1}) = E_j(z^{-1})B(z^{-1})$ with:

$$
\mathbf{G_j}(z^{-1}) = \begin{bmatrix} G_1(z^{-1}) \\ G_2(z^{-1}) \\ \vdots \\ G_j(z^{-1}) \end{bmatrix} = \begin{bmatrix} g_0 & g_1 & \dots & 0 \\ g_0 & g_1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ g_0 & g_1 & \dots & g_{(j+nb)} \end{bmatrix}
$$

The prediction equation can now be rewritten in a matrix/vector form:

$$
\mathbf{y} = \mathbf{Gu} + \mathbf{G'}(z^{-1})\Delta u(t-1) + \mathbf{F}(z^{-1})y(t) \tag{2.6}
$$

where the matrices and vectors, $\mathbf{G}$, $\mathbf{G}'$ and $\mathbf{F}(z^{-1})$ are defined as:

$$\mathbf{G} = \begin{bmatrix} g_0 & 0 & \dots & 0 \\ g_1 & g_0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ g_{j-1} & g_{j-2} & \dots & g_0 \end{bmatrix}$$

$$\mathbf{G}' = \begin{bmatrix} (G_1(z^{-1}) - g_0)z \\ (G_2(z^{-1}) - g_0 - g_1 z^{-1})z^2 \\ \vdots \\ (G_j(z^{-1}) - g_0 - g_1 z^{-1} - ... - g_{j-1} z^{-(j-1)})z^j \end{bmatrix}$$

$$\mathbf{F}(z^{-1}) = \begin{bmatrix} F_1(z^{-1}) \\ F_2(z^{-1}) \\ \vdots \\ F_j(z^{-1}) \end{bmatrix}$$

and the output and incremental input vectors, $\mathbf{y}$ and $\mathbf{u}$ are defined as:

$$\mathbf{y} = \begin{bmatrix} y(t+N_1) \\ y(t+N_1+1) \\ \vdots \\ y(t+N_2) \end{bmatrix}$$

$$\mathbf{u} = \begin{bmatrix} \Delta u(t-d) \\ \Delta u(t-d+1) \\ \vdots \\ \Delta u(t+N_u-d-1) \end{bmatrix}$$

The prediction model now contains two distinct portions: one that is dependent on future values of u and one that is dependent on past and current values of u and y. These parts are termed the forced and free responses respectively. With this information in mind, the prediction equation is rewritten in matrix format:

$$\mathbf{y} = \mathbf{G}\mathbf{u} + \mathbf{f} \tag{2.7}$$

where $\mathbf{G}\mathbf{u}$ is the forced response and $\mathbf{f}$ is the free response.

An important consideration when formulating the prediction is deciding what values to assign to the prediction and control horizons. There are 3 different horizon-related variables to consider: the start of the prediction horizon (N_1), the end of the prediction horizon (N_2) and the control horizon (N_u).

For the purposes of this dissertation, unless otherwise noted, it is assumed that $N_2 = N_1 + N_u - 1$ and $N_1 = d + 1$ where d is the number of integral dead time steps of the plant. That is, the control horizon is the same length of the prediction horizon although the start and end points of the horizons are not necessarily the same.

The literature has provided guidelines as to what the default choices for the horizons should be, depending on the plant to be controlled. The general consensus is that the start of the prediction horizon should be taken to be at $d + 1$ to avoid rank deficiency in the $\mathbf{G}$ matrix, the end of the prediction should encompass the rise time of the plant but at a minimum, for non-minimum phase systems, be greater than the degree of $B(z^{-1})$ to include all non-minimum phase behaviour. A short control horizon saves on computational cost but, as this is not a concern, it will be set to the length of the prediction horizon for this dissertation.

Until recently, the choice of prediction horizon was limited by computing power. The concept of 'bigger is better' was applied as it was assumed that more knowledge of the response of the system yielded better control or at least allowed for the possibility for better control [13, 31].

Recent literature has highlighted that a long prediction horizon may not provide optimal control [30, 29], especially when applying the GPC algorithm to non-constant setpoints. More specifically, the choice of control horizon should be short enough to not incorporate rapidly changing setpoint dynamics so as to force the controller to act more quickly to future setpoint changes. For control of unstable open loop plants, the choice of prediction horizon is important. A long prediction horizon will lead to a $\mathbf{G}$ matrix containing small numbers in the top rows and large numbers in the bottom rows, potentially leading to rank deficiency.

2.5.2 The Control Law

GPC obtains the optimal control increment, defined by $\Delta u(t)$, by minimizing a cost function over a specified prediction horizon. The cost function is of quadratic form, minimizing both the error squared and the input squared values. Weighting variables in the form of $\delta(j)$ and $\lambda(j)$ are included as added degrees of freedom. It is worth noting that other cost functions may be utilized and that is the prerogative of the control engineer. This dissertation is investigating the GPC algorithm which makes use of the above-mentioned cost function. The cost function is:

$$J(N_1, N_2, N_u, \delta(j), \lambda(j)) = \sum_{j=N_1}^{N_2} \delta(j)[y(t+j) - w(t+j)]^2 + \sum_{j=1}^{N_u} \lambda(j)[\Delta u(t+j-1)]^2 \quad (2.8)$$

The error weighting function is $\delta(j)$ and the control action weighting function is $\lambda(j)$. They are typically defined such that $\lambda(j) = \lambda \geq 0$ and $\delta(j) = \delta \geq 0$ although δ is typically taken to be 1. A small or zero value of λ will make the controller fast and aggressive but more sensitive to varying λ or model mismatch while a large value will increase robustness but make the controller sluggish as the available input is limited [16]. The setpoint is defined by $w(t)$.

Rewriting (2.8) into matrix form and minimizing with respect to $\mathbf{u}$ yields the prediction of the optimal control increment sequence. This is a long if simple mathematical derivation and, while not included in this dissertation, is available in seminal works and contemporary GPC textbooks. The optimal control sequence is defined by:

$$\mathbf{u} = (\mathbf{G}^T\mathbf{G} + \lambda\mathbf{I})^{-1}\mathbf{G}^T(\mathbf{w} - \mathbf{f}) \tag{2.9}$$

However, since only the first control increment is applied, (2.9) is rewritten as:

$$\Delta u(t) = \mathbf{K}(\mathbf{w} - \mathbf{f}) \tag{2.10}$$

where $\mathbf{K}$ is the first row of the $(\mathbf{G}^T\mathbf{G} + \lambda\mathbf{I})^{-1}\mathbf{G}^T$ matrix.

2.5.3 Closed Loop Relationship

The closed loop relationship for unconstrained GPC is calculated using the following well-known control structure for linear systems originally proposed in [3] and seen in [10]:

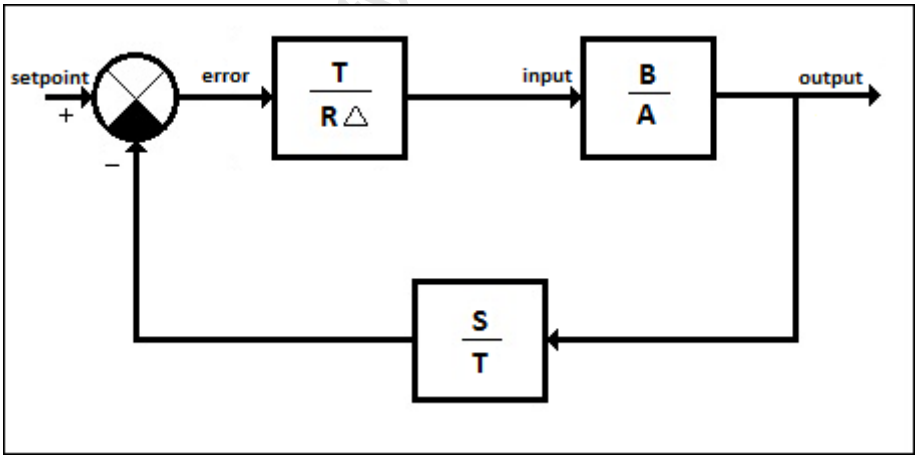

Figure 2.2: The Closed Loop Relationship

The control law is:

$$R(z^{-1})\Delta u(t) = T(z^{-1})w(t) - S(z^{-1})y(t) \tag{2.11}$$

where $T(z^{-1})$ is a prefilter as defined previously. Combining (2.11) and (2.1) and applying recursion of the Diophantine equation yields [13, 10]:

$$\left[T(z^{-1}) + z^{-1}\sum_{i=N_1}^{N_2} k_i G_i^{'}\right]\Delta u(t) = T(z^{-1})\sum_{i=N_1}^{N_2} k_i w(t) - \sum_{i=N_1}^{N_2} k_i F_i y(t)$$

where k_i are the elements of the $\mathbf{K}$ vector, I_i and F_i are elements of the free response.

$R(z^{-1})$ and $S(z^{-1})$ are therefore defined as:

$$R(z^{-1}) = \frac{T(z^{-1}) + z^{-1}\sum_{i=N_1}^{N_2} k_i G_i^{'}}{\sum_{i=N_1}^{N_2} k_i} \tag{2.12}$$

$$S(z^{-1}) = \frac{\sum_{i=N_1}^{N_2} k_i F_i}{\sum_{i=N_1}^{N_2} k_i} \tag{2.13}$$

The output is expressed as a function of setpoint and disturbance by substituting (2.11) into (2.1):

$$\begin{aligned} y(t) = & \frac{B(z^{-1})T(z^{-1})z^{-1}}{R(z^{-1})A(z^{-1})\Delta + B(z^{-1})S(z^{-1})z^{-1}}w(t) \\ & + \frac{T(z^{-1})R(z^{-1})}{R(z^{-1})A(z^{-1})\Delta + B(z^{-1})S(z^{-1})z^{-1}}e(t) \end{aligned} \tag{2.14}$$

and therefore the characteristic equation is given by:

$$\phi_c = R(z^{-1})A(z^{-1})\Delta + B(z^{-1})S(z^{-1})z^{-1} \tag{2.15}$$

CHAPTER 3

Setpoint Tracking

Proposed Problem: The literature review of Chapter 2 indicated that, while many applications of GPC were applied to constant step inputs, there was very little to no available literature regarding asymptotic tracking of ramp and sinusoid setpoints. The basic GPC algorithm, due to its Type 1 controller nature, will not track ramp or sinusoid setpoints unless it is altered.

Objective: The standard GPC algorithm will be extended to include considerations for asymptotic tracking of ramp setpoints. The results for this implementation will be compared to a method proposed in the literature that uses a Smith Predictor to generate the predicted outputs of the open loop model.

Similarly to ramp setpoint considerations, the standard algorithm will be adapted to track sinusoidal setpoints asymptotically. It is expected that the adapted algorithm will fail in this regard due to the inability of the standard GPC algorithm to fulfill the conditions of the internal model control principle for a sinusoidal setpoint. The results of this adaptation will be compared to a more complex algorithm proposed in literature that makes use of an alternative disturbance rejection polynomial to include a sinusoidal generator as a system in the control structure, fulfilling the requirements of the internal model control principle.

3.1 Ramp Setpoints

An algorithm for tracking ramp setpoints effectively and without steady-state error is proposed in [10]. This method injects dynamics from the setpoint into the minimization of the cost function. A different form of the method proposed in [10] will be shown here and compared to the proposed method.

3.1.1 Proposed Solution

The GPC algorithm has the ability to include the setpoint structure in the minimization as per (2.9). If the setpoint structure is known, this can be included in the minimization, however typically the setpoint structure is not known, for example in a cascade control loop, and as such approximations of the assumed setpoint need to be made. Consider a ramp setpoint of unknown slope, $\frac{\alpha}{T_s}$, starting at time t. The signal is defined by its generating function for k, $(\forall k \in \mathbb{Z}, k \geq 0)$ and for an arbitrary initial value $w(t) = w_i$:

$$w(t+k) = w_i + \alpha k \tag{3.1}$$

At time t, if the control system detects that the setpoint rate of change is non-zero by a simple difference equation $(w(t) - w(t - T_s))$, an α value is generated using that difference equation such that:

$$\alpha = w(t) - w(t-1) \tag{3.2}$$

With this approximation in mind, the vector of setpoint values used in the minimization of the cost function can be viewed as:

$$\mathbf{w} = w(t) \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} + \begin{bmatrix} 0 \\ \alpha \\ \vdots \\ (N_u - 1)\alpha \end{bmatrix} \tag{3.3}$$

Including this setpoint vector in the minimization of the cost function is a simple task and an example will be used to illustrate this. A plant was chosen to be not 'well-behaved' with oscillatory, under damped poles, a non-minimum phase zero and a period of oscillation of 15 seconds, denoted by:

$$g(s) = \frac{-0.104s + 0.193}{s^2 + 0.105s + 0.215}$$

The system has a 98% settling time of approximately 80 seconds and DC gain of 0.9. With the settling time and period of oscillation in mind, the sampling time (T_s) was chosen to be 1 second. The prediction horizon (N_2) was chosen to be 20, equal to a quarter of the plant settling time, and, after some trial and error, a λ value of 10 was chosen to sufficiently damp the closed loop response. Lower values of λ did provide stable control however the response was very oscillatory. The ramp variable, α, was calculated for each sampling instance based on the current and previous setpoint values.

The graphs in Fig. 3.1 show the system response to a varied setpoint structure. Ramp compensation was introduced midway through the simulation and as can be seen, the steady state error was reduced from approximately -0.22 units to effectively zero (taking numerical inaccuracy and rounding errors into account).

A negative unit step change is performed by the setpoint at time $t = 341$ seconds. This step caused a higher amplitude output and input response than was expected for the same magnitude step change seen earlier in the simulation. This is due to the α value at that point being taken as the magnitude of the step change which yielded a large corrective

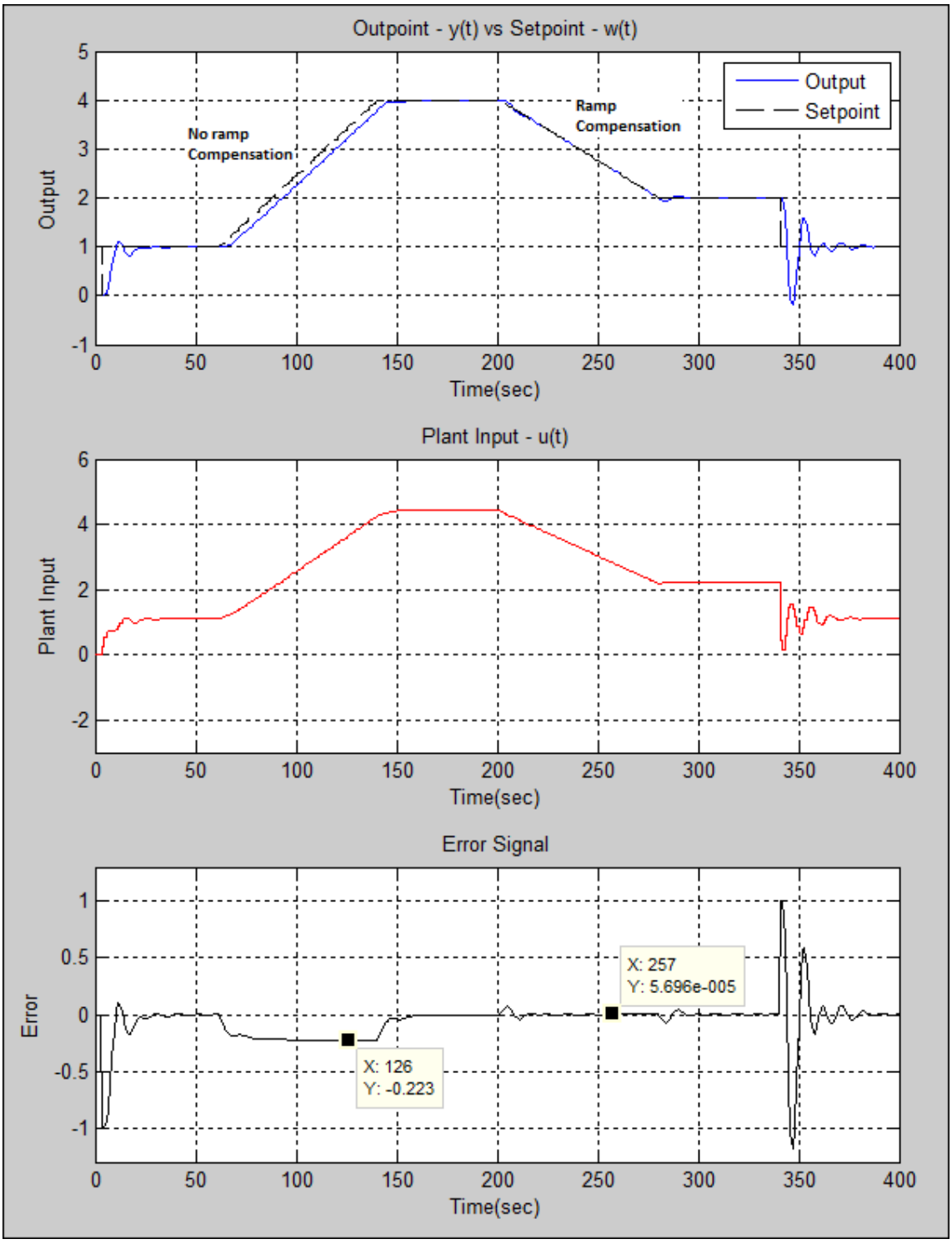

Figure 3.1: Ramp Tracking Output Response

output from the controller. To circumvent this problem, a check was programmed into the simulation to determine if the setpoint had shifted from a constant value to a ramp. This solution is deemed 'α-Correction' in this dissertation.

The α-Correction check determined if, at time t, the rate of change of the setpoint was equivalent to the calculated rate of change for the previous sample instance. That is; the α value at time t was equal to the α value at time $t - T_s$. Equivalence indicated that the setpoint was a ramp of slope α and a difference in values indicated the setpoint had performed a step change and that the augmented setpoint vector would not be included in the optimal control increment calculation. This implementation degraded the performance of the ramp compensation marginally but reduced the amplitude response of the output and plant input signals significantly. These results can be seen in Fig. 3.2, specifically at time $t = 341$ seconds.

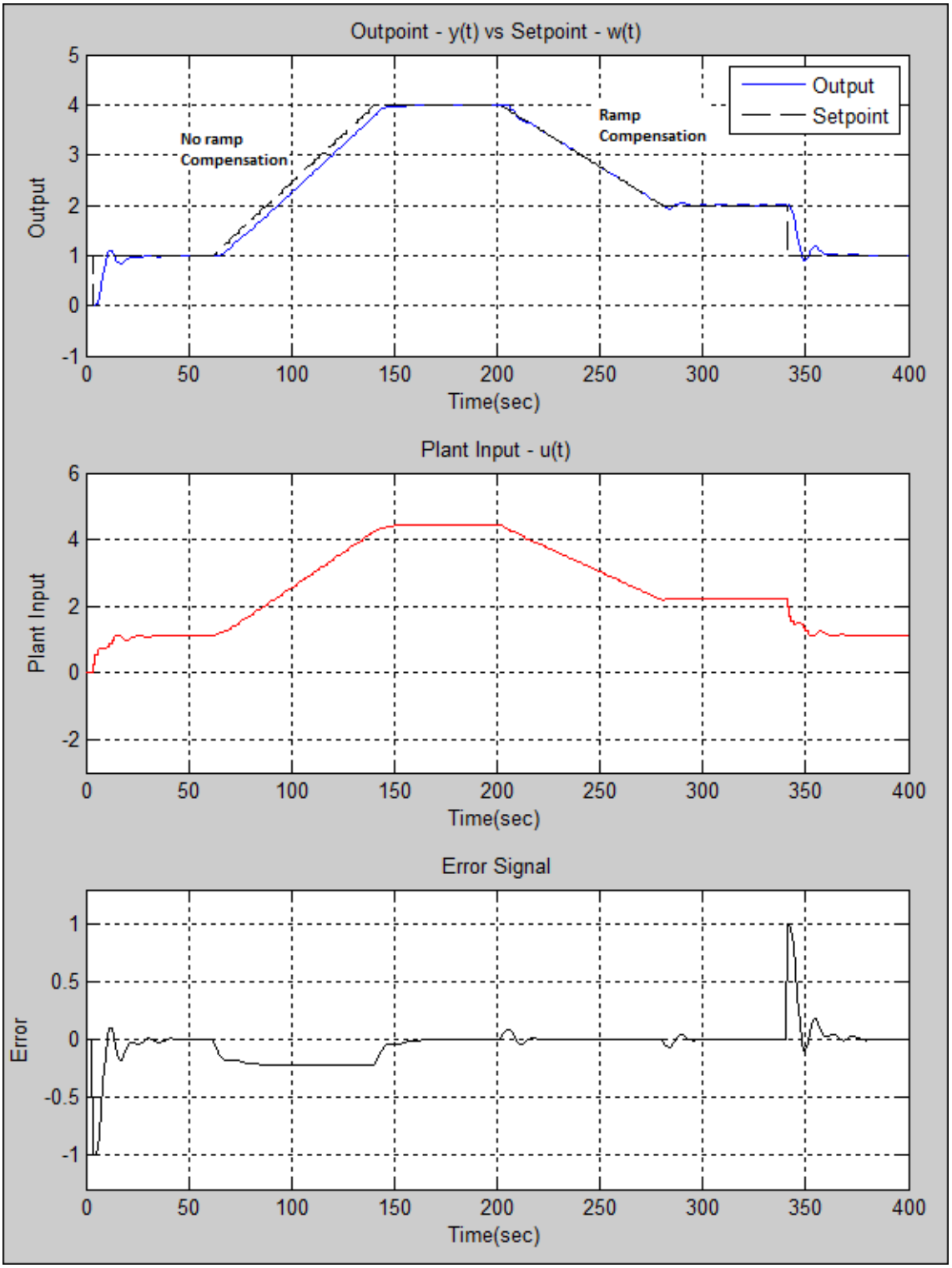

Figure 3.2: Ramp Tracking Output Response - α-Correction Implemented

3.1.2 Ramp Tracking Using The Smith Predictor

The method proposed in [10] is a deviation from the above-mentioned method in that it utilizes a Smith predictor to compute the prediction equation and consequently the optimal control increment. This section aims to compare results of the two methods.

As before, the Smith Predictor General Predictive Control (SPGPC) method utilizes the standard GPC cost function, (2.8). The Smith predictor calculates a j-step ahead prediction by utilizing the CARIMA equation, (2.1), to calculate $y(t+j)$ using the previous recursion, $y(t+j-1)$. The first recursion is shown below. For simplicity consider the CARIMA model defined by a system with dead time, d, $T(z^{-1}) = 1$, $N_1 = d+1$, $N_2 = d+N$, $N_u = N$, $\delta(j) = 1$ and $\lambda(j) = \lambda$. As stated previously, for future predictions $e(t)$ is assumed to be zero.

$$A(z^{-1})y(t) = z^{-d}B(z^{-1})u(t-1) + \frac{e(t)}{\Delta}$$

$$A(z^{-1})\Delta y(t) = B(z^{-1})\Delta u(t-d-1) + e(t)$$

$$y(t) = (\hat{A}(z^{-1}) - A(z^{-1}))y(t-1) + B(z^{-1})\Delta u(t-d-1) + e(t) \quad (3.4)$$

$$y(t+1) = (\hat{A}(z^{-1}) - A(z^{-1}))y(t) + B(z^{-1})\Delta u(t-d) \quad (3.5)$$

where $A(z^{-1}) = 1 + z^{-1}\hat{A}(z^{-1})$. By substituting (3.4) into (3.5), $y(t+1)$ can be expressed in terms of current and past values of $y(t)$, $\Delta u(t+d-1)$ and the current control increment, $\Delta u(t)$. Performing this recursively for j steps, $y(t+j)$ can be expressed as:

$$
\begin{aligned}
y(t+j) = {} & (1-a_1)y(t+j-1) + (a_1-a_2)y(t+j-2) + ... \\
& + (a_{na-1}-a_{na})y(t+j-na-2) + (a_{na})y(t+j-na-1) \\
& + b_0\Delta u(t+d+j-1) + ... + b_{nb}\Delta u(t+d+j-1-nb)
\end{aligned} \tag{3.6}
$$

This prediction equation can be summarized in a vector/matrix format as follows:

$$
\mathbf{y} = \mathbf{Lu} + \mathbf{Hu_b} + \mathbf{Sy_b} \tag{3.7}
$$

where $\mathbf{y}$ and $\mathbf{u}$ are defined as previously. Matrices $\mathbf{L}$, $\mathbf{H}$ and $\mathbf{S}$, are of dimension (N x N), (N x nb) and (N x $na+1$) respectively. Vectors, $\mathbf{u_b}$ and $\mathbf{y_b}$, are the past nb and $na+1$ values of the input increments and output respectively.

Substituting (3.7) into the cost function and minimizing with respect to $\mathbf{u}$ yields the control law:

$$
\begin{aligned}
(\mathbf{L}^T\mathbf{L} + \lambda\mathbf{I})\mathbf{u} &= -\mathbf{L}^T\mathbf{Sy_b} - \mathbf{L}^T\mathbf{Hu_b} + \mathbf{L}^T\mathbf{w} \\
\mathbf{Mu} &= \mathbf{P}_0\mathbf{y_b} + \mathbf{P}_1\mathbf{u_b} + \mathbf{Rw} \\
\Delta u(t) &= \mathbf{qP}_0\mathbf{y_b} + \mathbf{qP}_1\mathbf{u_b} + \mathbf{qRw}
\end{aligned} \tag{3.8}
$$

where $\mathbf{q}$ is the first row of $\mathbf{M}^{-1}$. The control law can be viewed as a series of coefficients acting on the input, output and setpoint signals:

$$\Delta u(t) = [l_{y1}\ l_{y2}\ ...\ l_{y(na+1)}] \begin{bmatrix} y(t+d) \\ y(t+(d-1) \\ \vdots \\ y(t+(d-na-1)) \end{bmatrix}$$

$$+ [l_{u1}\ l_{u2}\ ...\ l_{u(nb)}] \begin{bmatrix} \Delta u(t+(d-1)) \\ \Delta u(t+(d-2)) \\ \vdots \\ \Delta u(t+(d-nb)) \end{bmatrix} \tag{3.9}$$

$$+ l_{w1} w(t)$$

$$\Delta u(t) = \mathbf{l_y y_b} + \mathbf{l_u u_b} + l_w w(t) \tag{3.10}$$

where $w(t) = w(t+1) = w(t+N)$. The coefficients are calculated as follows:

$$\mathbf{qP}_0 = [l_{y1}\ l_{y2}\ ...\ l_{y(na+1)}]$$

$$\mathbf{qP}_1 = [l_{u1}\ l_{u2}\ ...\ l_{u(nb)}]$$

$$l_{w1} = \sum_{i=1}^{N} q_i \sum_{j=1}^{N} r_{i,j}$$

and $r_{i,j}$ are the coefficients of the $\mathbf{R}$ matrix. This definition holds true only for constant setpoints. However, the setpoint under consideration is defined by $w(t) = w(t-1) + \alpha$. Consider the reference term in the control law:

$$\mathbf{qRw} = \mathbf{hw} = \sum_{i=1}^{N} h_i \; w(t+i) = \sum_{i=1}^{N} h_i \; [w(t) + i \; \alpha] = \sum_{i=1}^{N} h_i \; w(t) + \alpha \sum_{i=1}^{N} h_i \; i \qquad (3.11)$$

This can be expressed as:

$$\mathbf{hw} = l_{w1} \; w(t) + l_{w2} \; \alpha \qquad (3.12)$$

and the optimal control increment is defined by:

$$\Delta u(t) = \mathbf{l_y y_b} + \mathbf{l_u u_b} + l_{w1} w(t) + l_{w2} \; \alpha \qquad (3.13)$$

The literature provides a way to generate coefficients for a first order system. To be able to test the algorithm on a second order system, the prediction equation, (3.6), needed to be expanded. Recursion of (3.6) for a general form of a second order plant yielded the following equations used to generate the coefficients, $\mathbf{l_y}$, $\mathbf{l_u}$ and $\mathbf{l_w}$, for a system defined by:

$$(1 + a_1 z^{-1} + a_2 z^{-1}) y(t) = (b_0 + b_1 z^{-1}) \Delta u(t-1) + \frac{e(t)}{\Delta}$$

The first rows L, H and S are defined by:

$$S_1 = [(1 - a_1) \;\; (a_1 - a_2) \;\; (a_2)]$$

$$H_1 = b_1$$

$$L_{1,1} = b_0$$

$$L_{2,1} = A_1 b_0 + b_1$$

and $\mathbf{L}$ is lower triangular such that:

$$\mathbf{L} = \begin{bmatrix} l_1 & 0 & \dots & 0 \\ l_2 & l_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ l_N & l_{N-1} & \dots & l_1 \end{bmatrix}$$

and the subsequent coefficients are defined, for $j \in \mathbb{Z}; 2 \leq j \leq N_2$, by:

$$S_{j,1} = S_{1,1} S_{j-1,1} + S_{j-1,2}$$

$$S_{j,2} = S_{1,2} S_{j-1,1} + S_{j-1,3}$$

$$S_{j,3} = S_{1,3} S_{j-1,1}$$

$$H_j = H_1 S_{j-1,1}$$

$$L_{j,1} = b_0 S_{j-1,1} + b_1 S_{j-2,1}$$

This algorithm was applied to the example from Section 3.1.1. It was expected that this algorithm will track the ramp setpoints asymptotically as there has been inclusion of a ramp generating function in the control structure. The original control parameters were not sufficient to achieve stable control. The large prediction horizon generated large controller coefficients that the control weighting function was not able to compensate for. A value of $\lambda = 100$ was found to stabilize the system however the output was very oscillatory and a large offset to the ramp setpoint was observed as shown in Fig. 3.3.

Reducing the prediction horizon to 5 and varying λ yielded various responses from the system. A value of $\lambda = 2.5$ was found to provide near-zero error to the ramp setpoints.

The SPGPC algorithm did not yield results in line with claims made in literature though it is noted that the literature only provided numerical examples for a first order plant. Asymptotic tracking of ramp setpoints was achieved only for a single value of λ indicating the the controller lacked the necessary generating function in the control structure to track a ramp asymptotically. This is akin to the finite error seen in systems lacking the requisite Type 1 characteristics to track a ramp setpoint asymptotically [7].

Fig. 3.5 shows the relationship between increasing λ and the sum of the error squared generated during tracking of a ramp (specifically, for a time between 122 and 280 seconds). The nature of the graph alludes to a link between λ and the perceived gain of the system though this topic will be discussed further in Chapters 4 and 5.

The implementation of the method is initially time consuming, especially for higher order systems, however the overall number of online computations is reduced as only a simple vector multiplication is required to compute the control law. For large prediction horizons, the control law coefficients become large compounding the effects of numerical inaccuracies, rounding errors and/or prediction mismatch. This could potentially cause unstable behaviour, requiring large λ values to compensate that may lead to stable but sluggish controller behaviour.

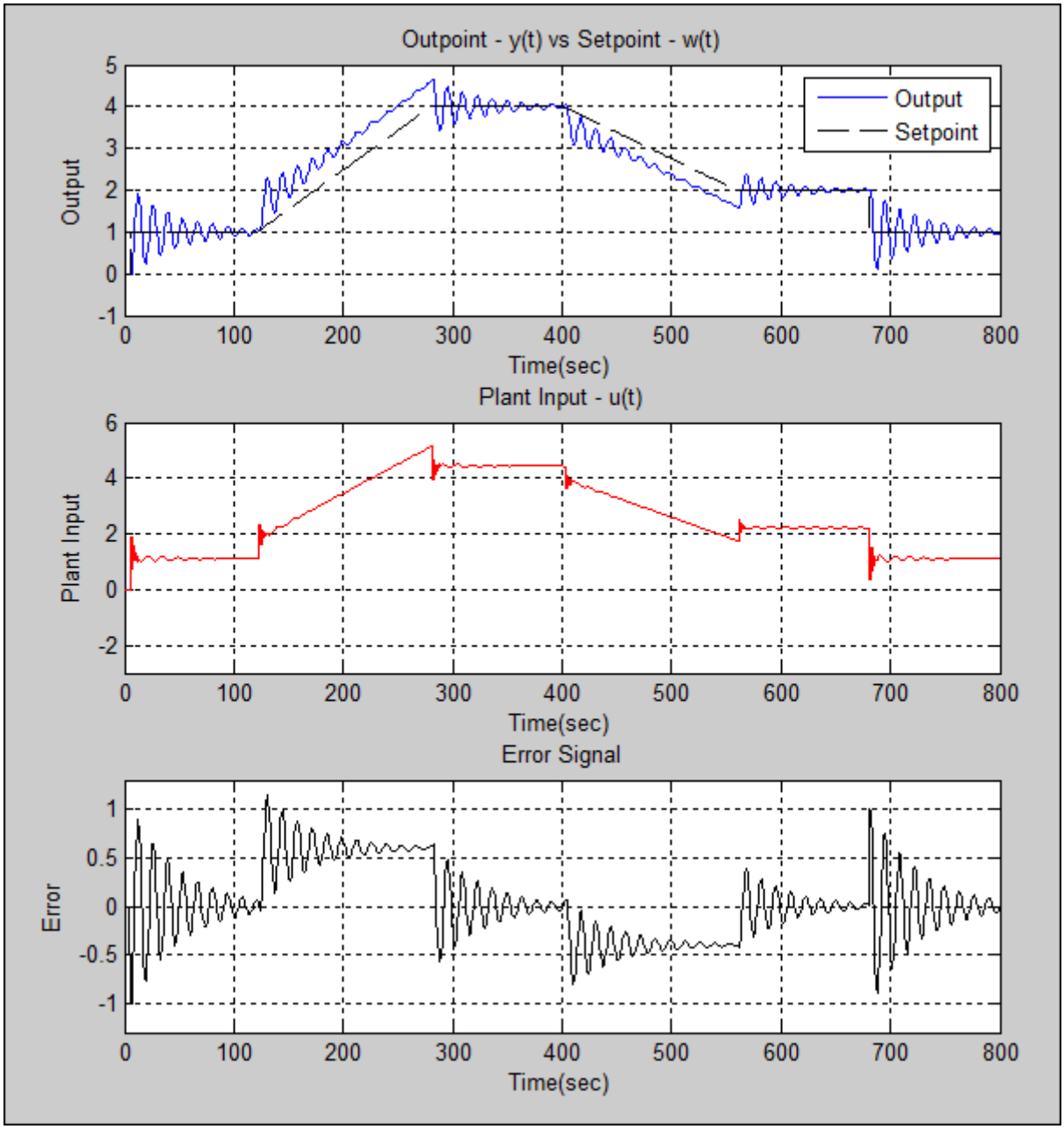

Figure 3.3: Ramp Tracking Output Response - $\lambda = 100$

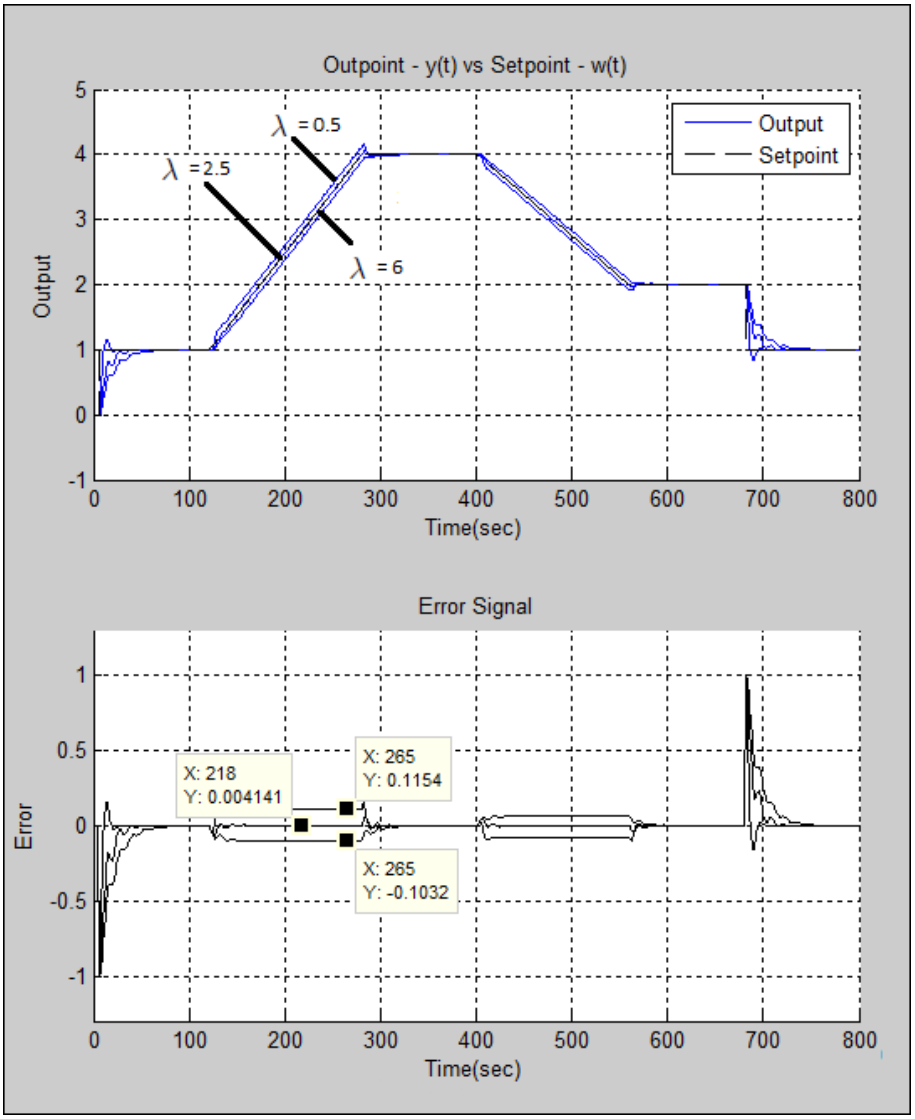

Figure 3.4: Ramp Tracking Output Response - Varying λ

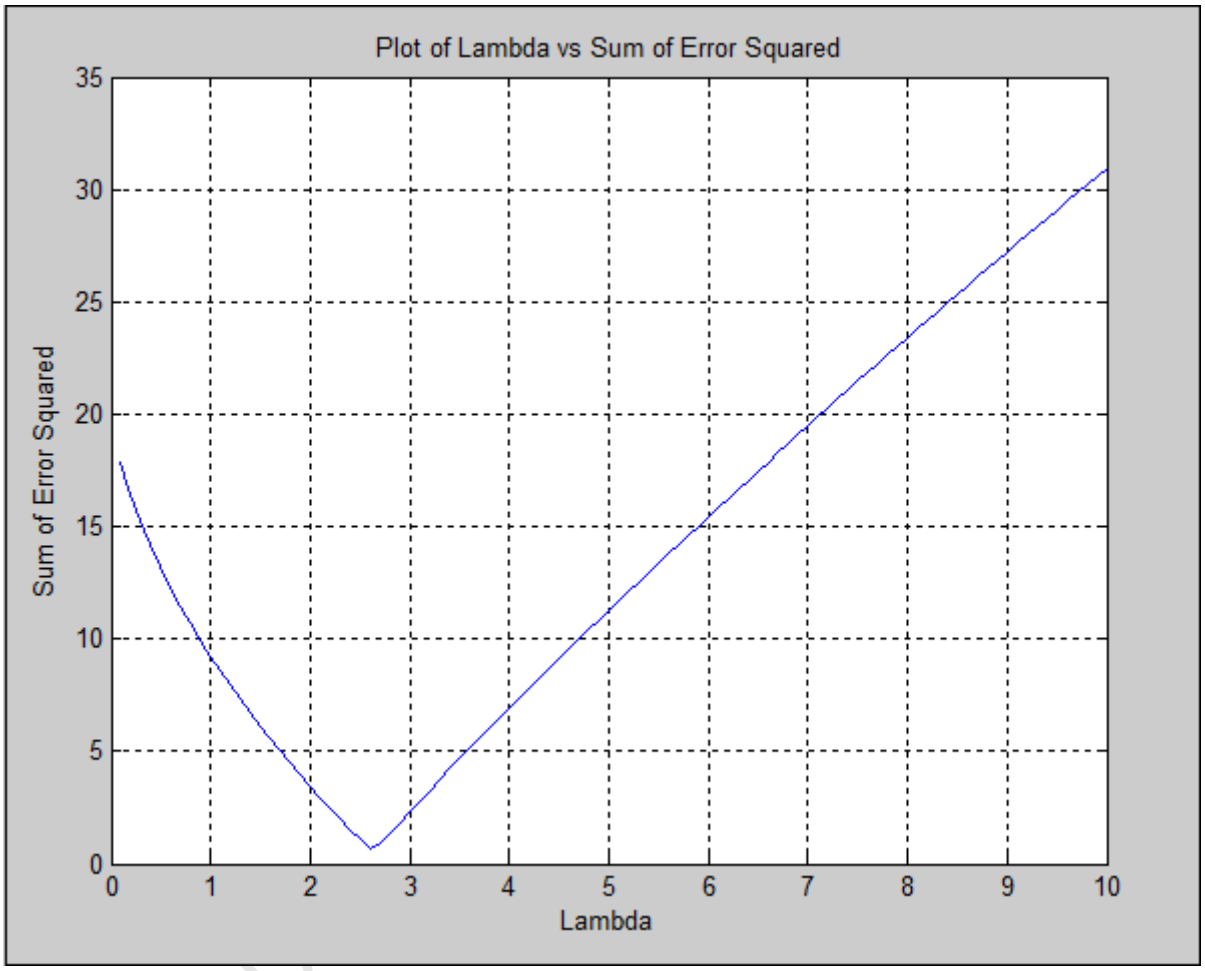

Figure 3.5: λ vs Sum of Error Squared

3.2 Sinusoidal Setpoints

An article, [35], has provided the basis of an algorithm that allows for error free sinusoidal setpoint tracking. This algorithm will be discussed briefly and compared to the standard GPC algorithm utilizing setpoint data injection into the optimal control sequence calculation.

3.2.1 Proposed Solution

To illustrate some design considerations when dealing with open loop unstable systems as well as show effective setpoint tracking of a sinusoid, an unstable non-minimum phase plant was chosen:

$$g(s) = \frac{-s+3}{120s^2+7s-1} \tag{3.14}$$

This plant contains a stable component with a time constant of 8, therefore a 1 second sampling time was chosen. Care was taken when considering the prediction and control horizons. Literature in [29] has made reference to design considerations with regard to open loop unstable systems. As the predictor makes use of the open loop model, the prediction and control horizons should not be too short or too long to avoid prediction errors. Initially, N_u and N_1 were set to 1, a prediction horizon was chosen as $N_2 = 25$ and λ set to 5.

The simulation was coded to use the standard GPC algorithm for the first 120 seconds, and to incorporate setpoint data into the control calculation for the second 120 seconds. The frequency of the setpoint was made available to the controller during the second half of the simulation to attempt to track the setpoint asymptotically. From this information, the sinusoidal setpoint trajectory was calculated at each sampling instance (t_k) using

the previous and current setpoint value, $w(t_k)$ and $w(t_k - 1)$ and incorporated into the calculation of the optimal control increment.

Results from the above example are shown in Fig. 3.6. As was expected, due to the lack of future information of the setpoint, the algorithm tracked the sinusoidal input with finite error. Incorporating setpoint data reduced the error significantly as is seen from time $120 \leq t \leq 240$. However, asymptotic sinusoidal tracking was not achieved.

The Bode plot, shown in Fig. 3.7, indicated that the open loop system behaviour was in line with a low pass filter with a 3dB attenuation at a frequency of approximately 0.06 $rad.s^{-1}$, compared to the setpoint frequency of 0.1 $rad.s^{-1}$. The simulation was repeated with an easier setpoint signal of frequency 0.035 $rad.s^{-1}$ in an attempt to achieve asymptotic tracking.

The results from altering the setpoint frequency displayed a marked decrease in error as seen in Fig. 3.8 in line with what was expected. However asymptotic tracking was not achieved. Despite previous assumptions, the requirements of the Internal Control Model Principle for tracking a sinusoid have not been fulfilled. A generating function of the sinusoidal input has not been included in the control strucutre. The next section attempts to address this failing.

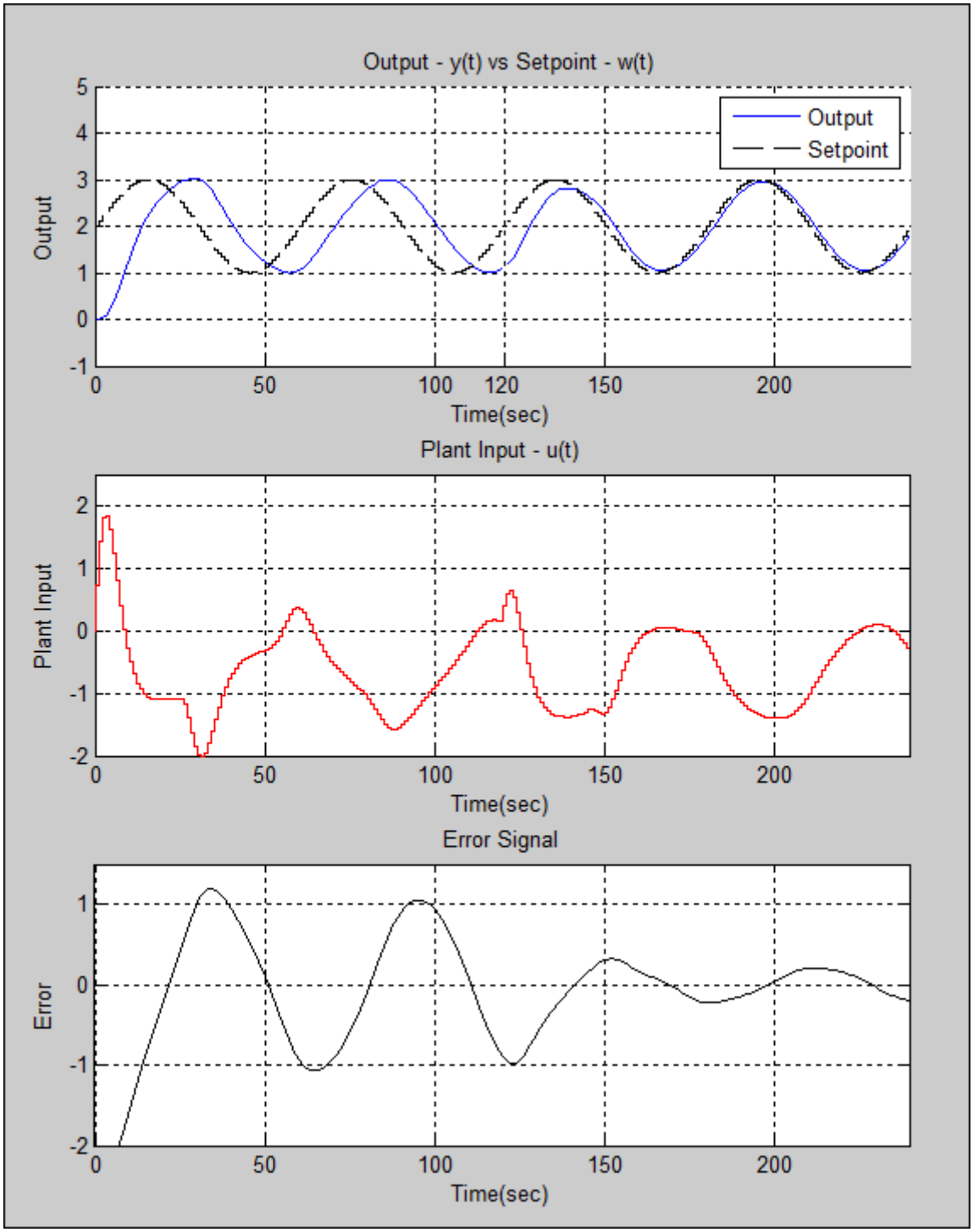

Figure 3.6: Sinusoidal Tracking Output Response - Proposed Method

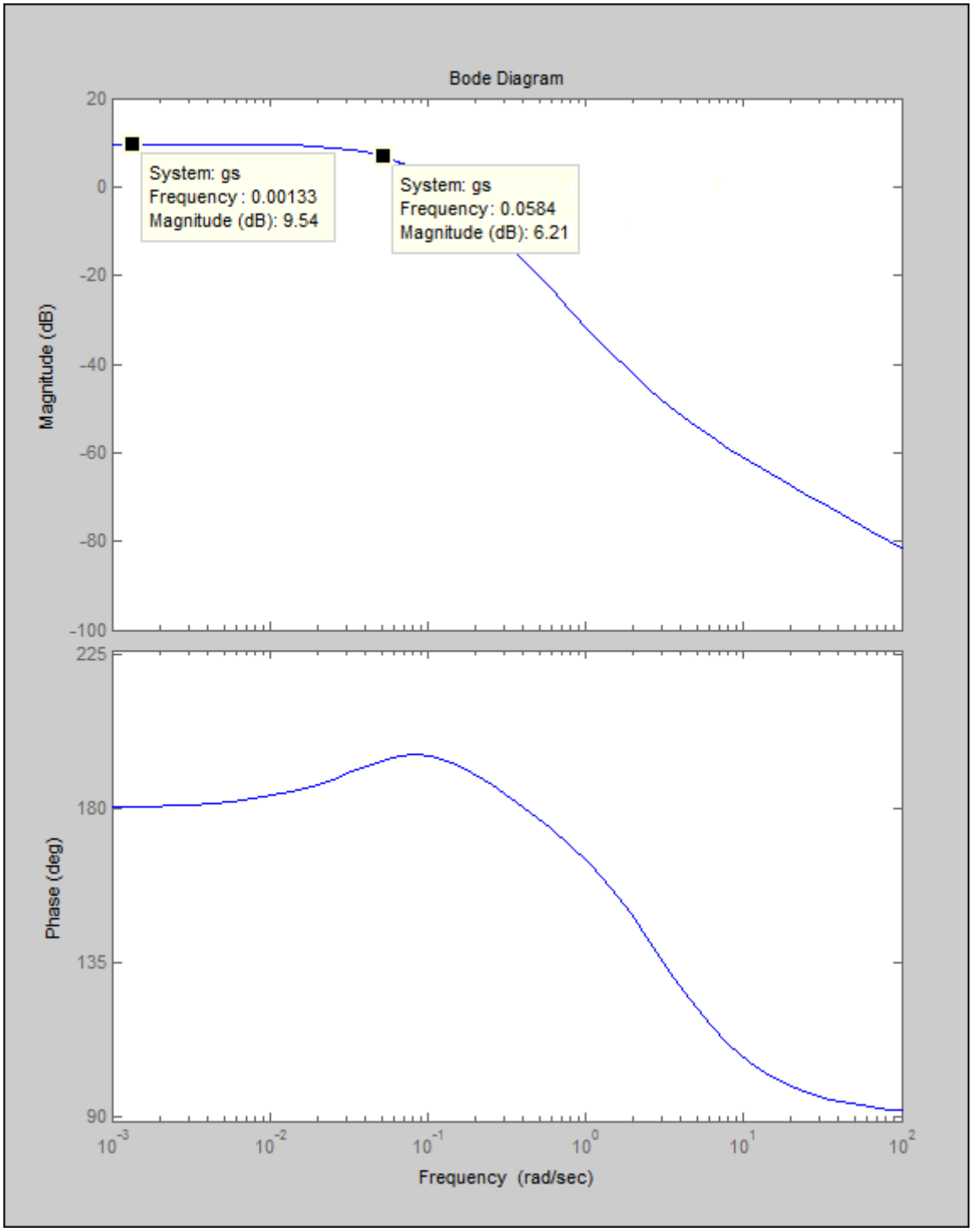

Figure 3.7: Bode Plot for Open Loop System

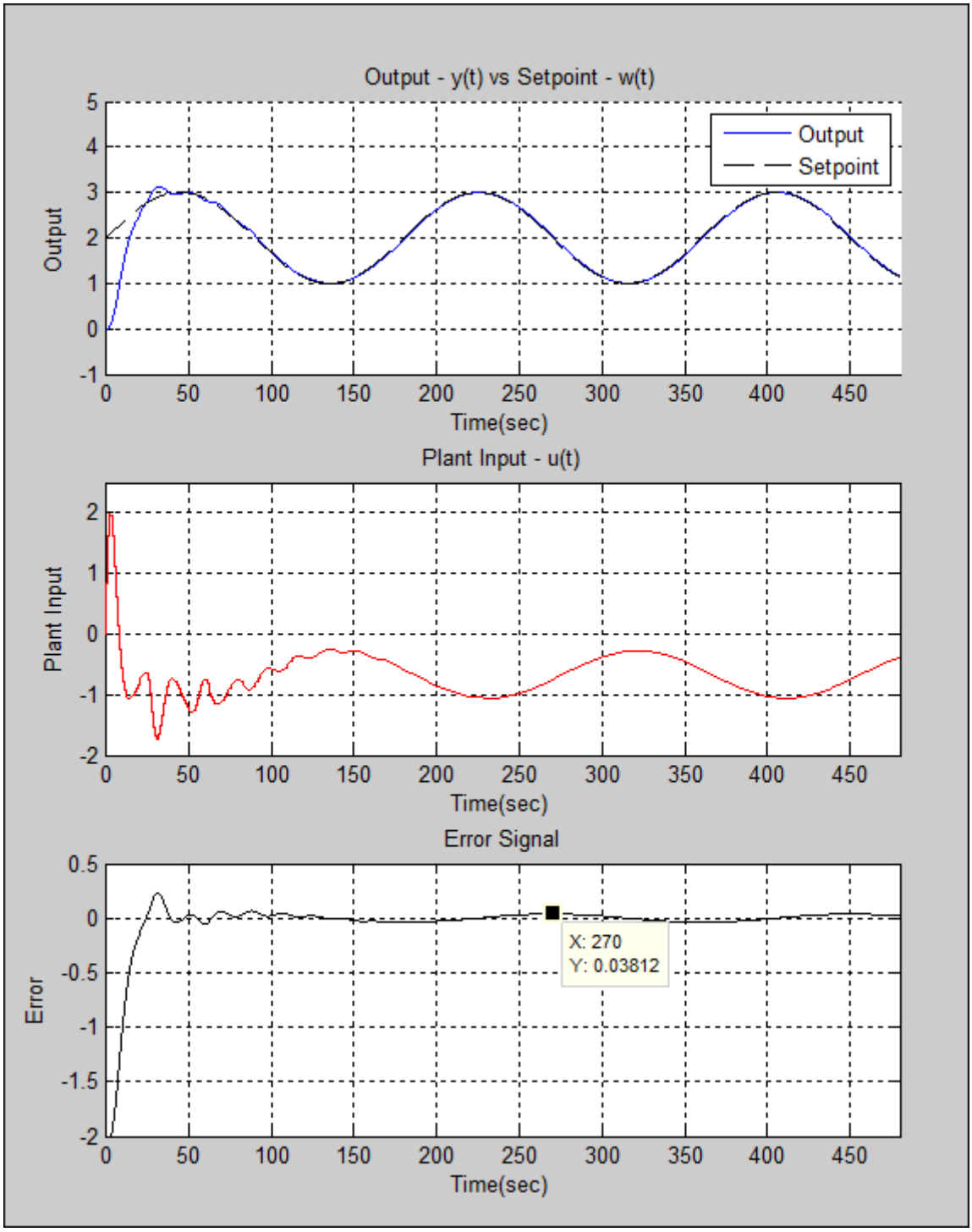

Figure 3.8: Output for Corrected Setpoint Frequency

3.2.2 Tracking Sinusoids Using an Appropriate Disturbance Rejection Model

A generalization of the CARIMA model can be viewed as:

$$A(z^{-1})y(t) = B(z^{-1})z^{-d}u(t) + \frac{T(z^{-1})}{D(z^{-1})}e(t) \tag{3.15}$$

where $D(z^{-1})$ is referred to as the disturbance rejection polynomial. Disturbance rejection is not the topic under consideration but this polynomial can be utilized for another more useful task applicable to sinusoidal setpoint tracking. Typically, $D(z^{-1}) = \Delta = 1 - z^{-1}$ as most GPC applications use constant setpoints. The reason for this is that the disturbance rejection polynomial contains the requisite dynamics for asymptotic tracking of a constant setpoint. This will be shown below.

Consider the traditional control structure proposed by [3]:

$$R(z^{-1})D(z^{-1})u(t) = T(z^{-1})w(t) - S(z^{-1})y(t) \tag{3.16}$$

The Internal Model Control Principle requires a sinusoidal generator of the same frequency as the setpoint to be included in the control structure to be able to realize asymptotic tracking of a sinusoidal setpoint. As per the control strucutre above, it can be seen that selecting the correct $D(z^{-1})$ allows the control engineer to incorporate a sinusoidal generating function into the control structure. For a sinusoidal setpoint of known frequency, ω, an appropriate disturbance rejection polynomial is [29]:

$$D(z^{-1}) = 1 - 2z^{-1}cos(\omega) + z^{-2} \tag{3.17}$$

The example, performed in Section 3.2.1, was repeated using the same parameters but

using a different $D(z^{-1})$ to attempt asymptotic tracking.

Fig. 3.9 shows unstable behaviour. Consider that for a long prediction and control horizon, the algorithm attempts to minimize the errors and control signal for the entire horizon based on the open loop characteristics. Due to the unstable nature of the open loop system, the utilization of the future knowledge can lead to generation of large control increments that, while optimal for that horizon, lead to overall unstable behaviour.

To attempt to realize stable control, the controller parameters were adjusted, after some experimentation, to $\lambda = 1$ and $N_2 = 10$. Fig. 3.10 shows a stable, almost asymptotic response. There is a very small, but evident error present in the output. The simulation was repeated for $1 \leq \lambda \leq 10$ and the errors were documented in Fig. 3.11. The error was not linear in relation to λ. Claims of asymptotic tracking hold true for the literature, but the simulation data shows small finite error, the root cause of which can be attributed to numerical inaccuracies in the simulation process.

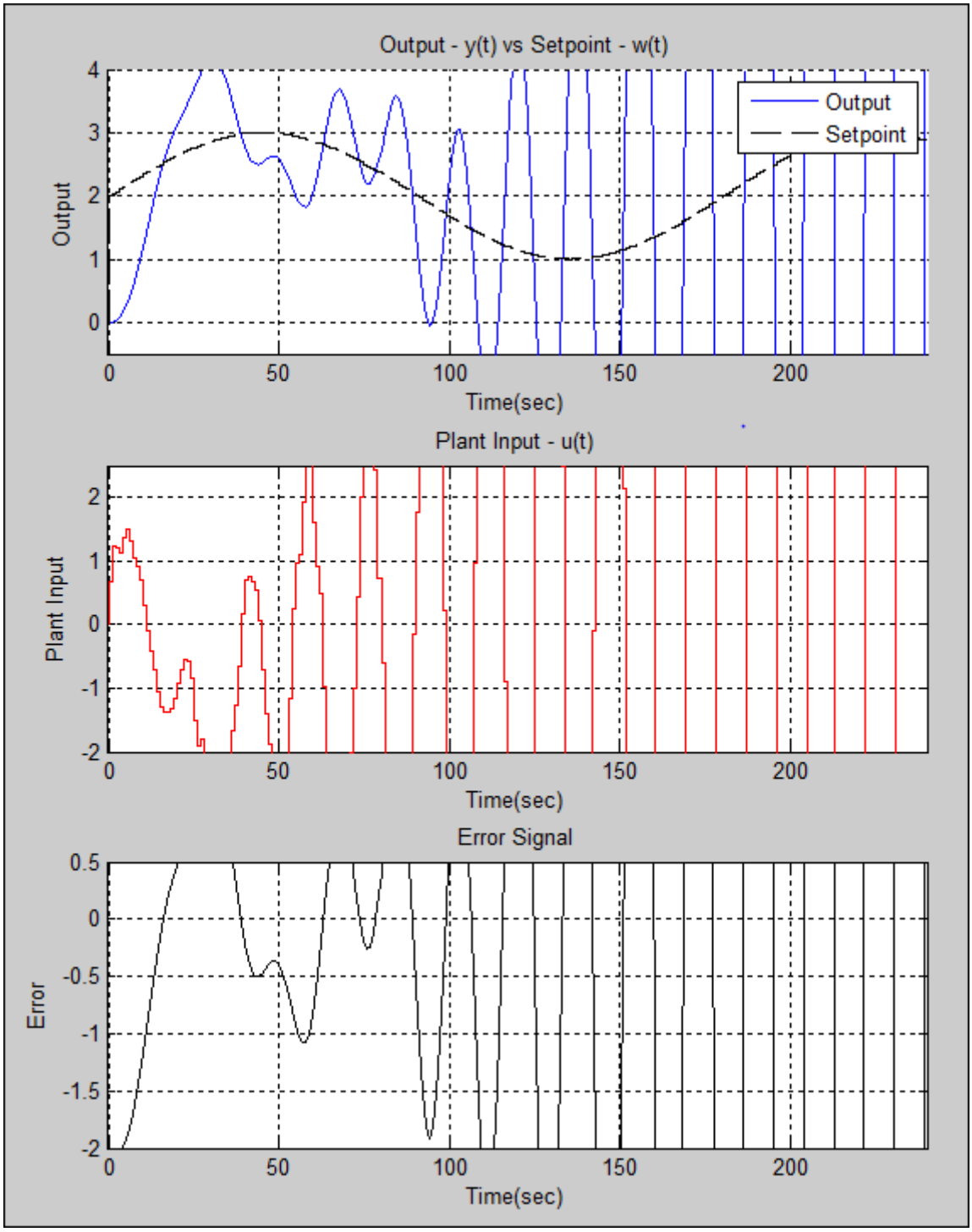

Figure 3.9: Sinusoidal Tracking Output Response - Unstable

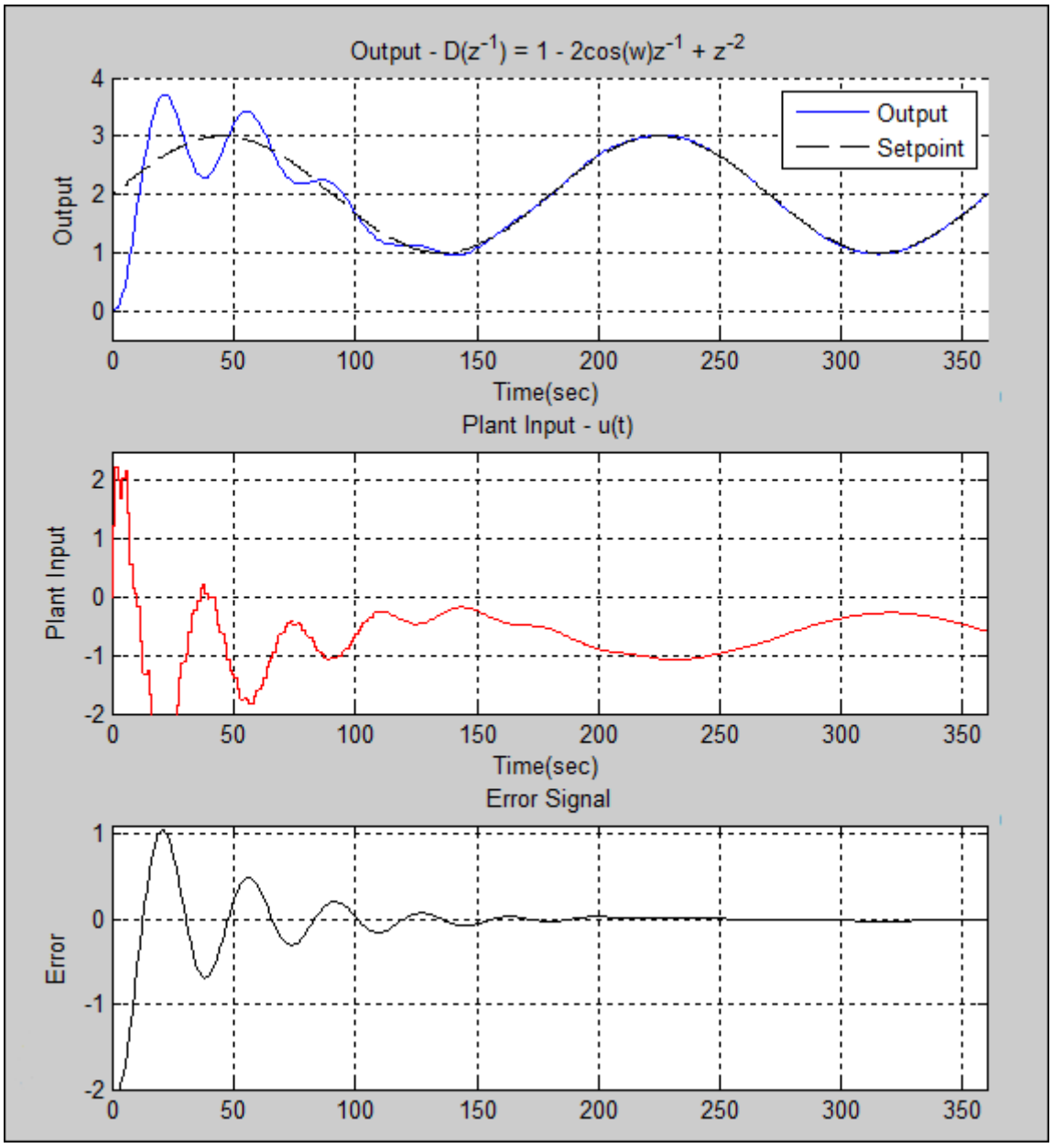

Figure 3.10: Sinusoidal Tracking Output Response - Stable

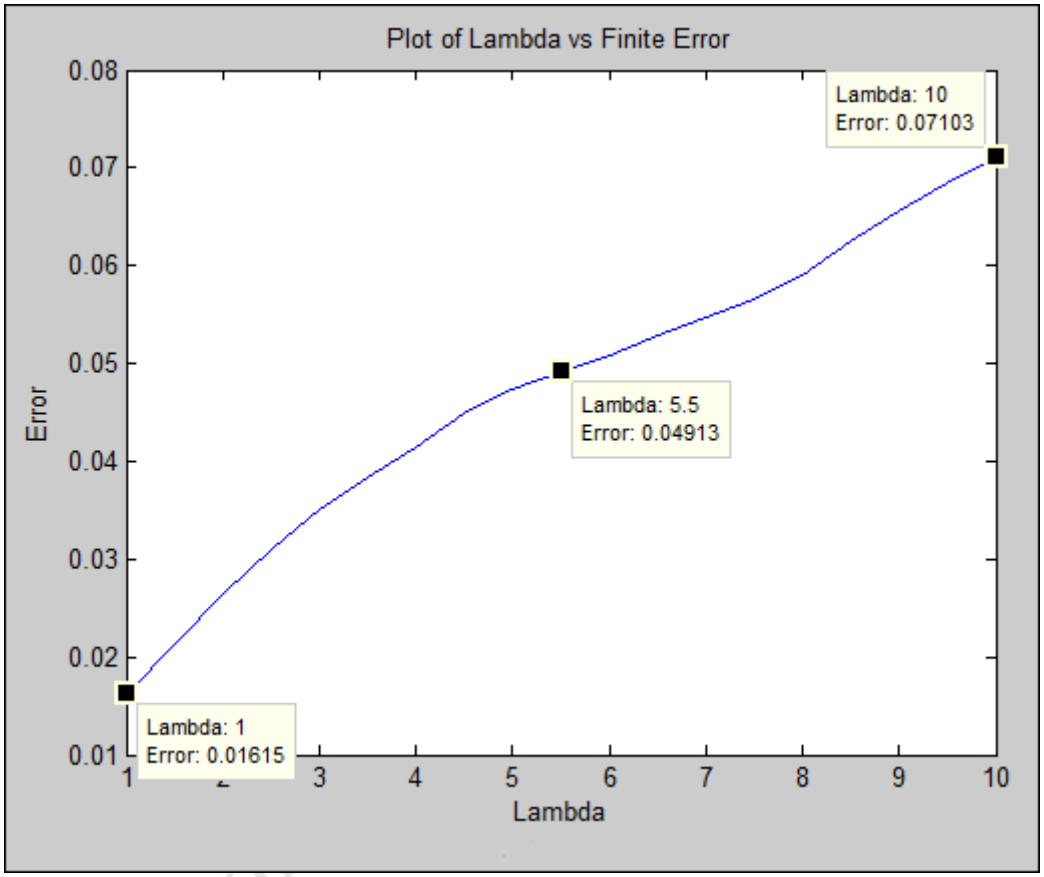

Figure 3.11: λ vs Finite Error for Sinusoid Tracking

CHAPTER 4

Pole/Zero Cancellation

Proposed Problem: The review of the literature presented in Chapter 2 states that Pole/Zero Cancellation (PZC) does occur under certain conditions for the GPC algorithm. It is shown that a non-zero value of λ prevents PZC however there has been no definitive statement on how to determine what value of λ will provide the desired closed loop pole positions.

Objective: This chapters aims to verify that PZC occurs in a GPC controller under certain conditions. It is specifically relevant to systems with non-integral dead time. As such, this investigation will be used as a basis for further work presented in Chapter 5.

4.1 The GPC Algorithm

The closed loop characteristic equation, (2.15), for a GPC-controlled system indicates that elements of the open loop zeros are present in the closed loop pole polynomial. From (2.15), if the term containing the open loop pole polynomial, $A(z^{-1})$, is made to be zero, the closed loop poles will be positioned on the open loop zeros. That is, if the control weighting parameter, λ, is set to 0, minimization of the cost function, J, results in a cancellation law that attempts to remove the plant dynamics by using an inverse plant model in the controller [13].

An analysis of the minimization of the cost function confirms this. Consider $\mathbf{G}$, from (2.7), that contains the step response data of the open loop system and, as such, can be loosely viewed as containing the open loop poles and zeros. Minimization of the cost function with $\lambda = 0$ yields:

$$\mathbf{u} = (\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T(\mathbf{w} - \mathbf{f})$$

$$\mathbf{u} = \mathbf{K}(\mathbf{w} - \mathbf{f})$$

The equations for the GPC closed loop relationship shown in Chapter 2 are repeated here for convenience:

$$\phi_c = R(z^{-1})A(z^{-1})\Delta + B(z^{-1})S(z^{-1})z^{-1}$$

$$R(z^{-1}) = \frac{T(z^{-1}) + z^{-1}\sum_{i=N_1}^{N_2} k_i G_i'}{\sum_{i=N_1}^{N_2} k_i}$$

$$S(z^{-1}) = \frac{\sum_{i=N_1}^{N_2} k_i F_i}{\sum_{i=N_1}^{N_2} k_i}$$

As per the equations shown in Chapter 2, for $\lambda = 0$:

$$\mathbf{K} = [\frac{1}{b_0} \quad 0 \quad ... \quad 0]$$

$$G_1' = [b_1 + b_2 z^{-1} + ... + b_{nb} z^{-(nb-1)}]$$

which yields

$$R(z^{-1}) = \frac{1 + z^{-1} k_1 G_1'}{k_1} = b_0 + b_1 z^{-1} + ... + b_{nb} z^{nb} = B(z^{-1})$$

This results in the closed loop characteristic equation reducing to:

$$\phi_c = B(z^{-1})[A(z^{-1})\Delta + S(z^{-1})z^{-1}] \tag{4.1}$$

which clearly shows that the zeros of the open loop system are present in the closed loop characteristic equation for $\lambda = 0$.

A simple example confirms this theory:

$$(1 - 0.8z^{-1})y(t) = (1 + 1.5z^{-1})u(t-1) + \frac{e(t)}{\Delta} \tag{4.2}$$

with $\lambda = 0$, $N_1 = 1, N_2 = N_u = 10$ and $T_s = 0.05$.

The root locus of (4.2) is shown in Fig. 4.1. The open loop poles are shown as blue crosses, the open loop zeros as blue circles and the closed loop poles as red squares. The closed loop poles are sitting on the open loop zero and at the origin as expected. Zeros at the origin represent zeros at $-\infty$ in the s-plane and have no effect due to their non-dominance.

The position of the closed loop pole was calculated as per (2.15) and shown on Fig. 4.1 to be at $z = -1.5$. As was expected, this closed loop pole is at the same position as the open loop zero and as such, the response of the controlled system was unstable.

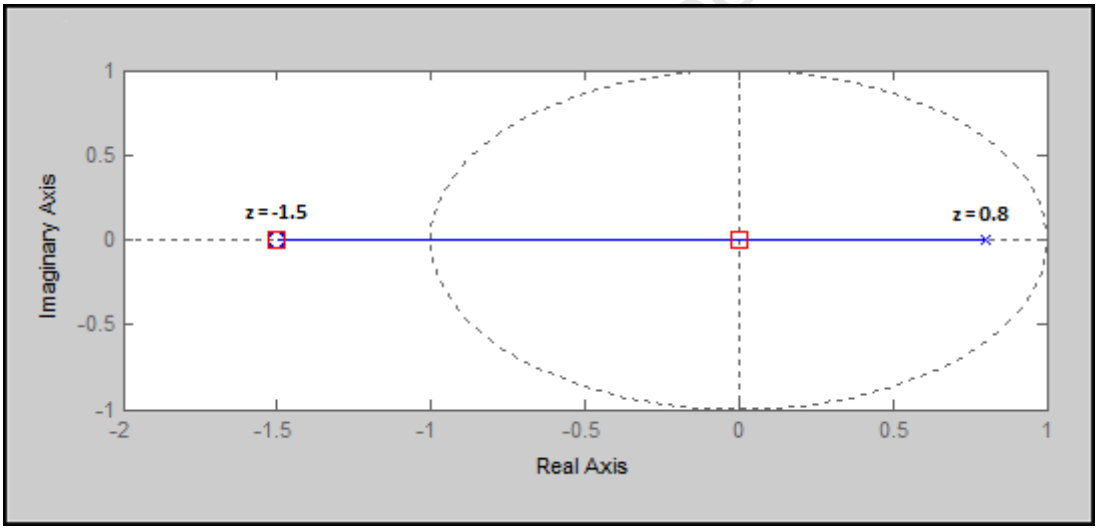

Figure 4.1: Root Locus for $g(z)$ with $\lambda = 0$

The effect of increasing λ from zero shifts the closed loop poles towards the open loop pole positions [13]. An analysis of the closed loop characteristic functions confirms this.

From (2.9) and (2.10):

$$\lim_{\lambda \to \infty} \Delta u(t) \quad \Longrightarrow \quad K = [0,\ 0,\ ...,\ 0] \tag{4.3}$$

Therefore from the equation for $R(z^{-1})$:

$$R(z^{-1}) = \frac{T(z^{-1}) + z^{-1} \sum_{i=N_1}^{N_2} k_i I_i}{\sum_{i=N_1}^{N_2} k_i} \tag{4.4}$$

the coefficients of $R(z^{-1})$ will tend to infinity for all values of z^{-1} resulting in the 'size' of $R(z^{-1})$ tending to infinity.

The closed loop pole positions can be found by setting the characteristic function equal to zero:

$$A(z^{-1})\Delta + \frac{z^{-1} S(z^{-1}) B(z^{-1})}{R(z^{-1})} = 0 \tag{4.5}$$

and, as $R(z^{-1})$ tends to ∞, the second term tends to 0 and the closed loop poles are equivalent the open loop poles and an integrator pole.

Re-examining the previous example for $\lambda = 10$ yields the following pole zero diagram and step response graphs:

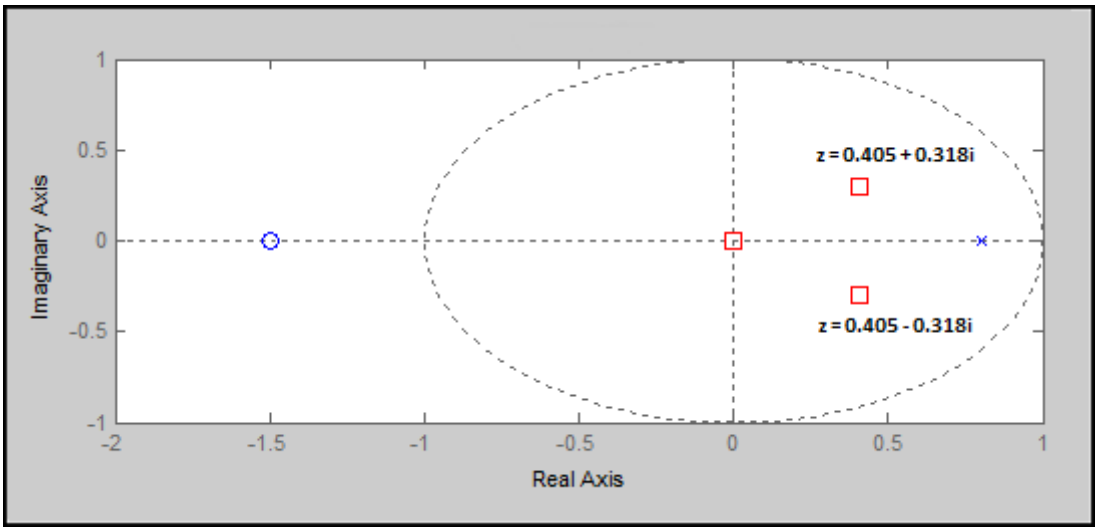

Figure 4.2: Pole and Zero Diagram for $\lambda = 10$

As can be seen from Fig. 4.3, the closed loop poles situated in the unit circle have yielded a stable response. It is not evident that the closed loop pole position with regard to varying λ are not visible on a standard root locus. Varying λ moves the closed loop poles from open loop zero positions to open loop pole positions, the opposite effect to increasing the system gain. As such, varying $\frac{1}{\lambda}$ can be thought of having a similar effect to varying the system gain in regard to the position of the closed loop poles. This will be investigated in the next chapter.

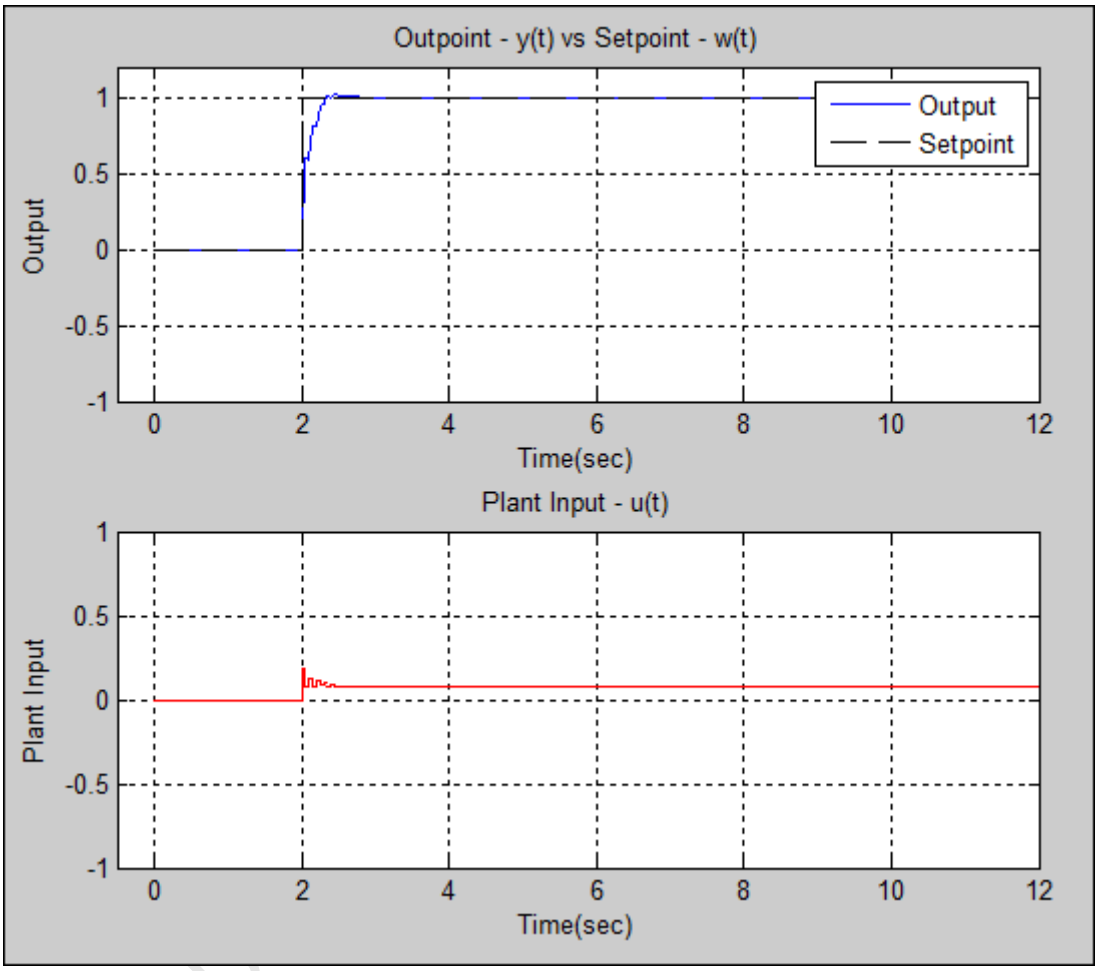

Figure 4.3: Step Response of the Closed Loop System

CHAPTER 5

Dead Time

Proposed Problem: The literature reviewed in Chapter 2 indicated that integral dead time is not a concern for the GPC control structure. Researchers have extended the standard algorithm to handle variable and unknown dead time with success. A subtle and potentially problematic issue arises with an effect dead time has on digital modeling, specifically for the non-integral dead time class. The effect in question is the zero created during the digital modeling of systems with dead time and whether it is inside or outside the unit circle. These potential non-minimum phase systems, for conditions found in Chapter 4, present unstable behavior.

Objective: The implementation of the GPC control structure in the case of integral and non-integral dead time cases will be investigated. The effect of λ, the fractional dead time, θ, and the system gain on the closed loop poles of a controlled system will be analyzed.

5.1 Integral Dead Time Case

The literature reviewed [13, 10] details some design considerations concerning the dead time that come into effect when choosing the initial prediction horizon, N_1. It should be taken that $N_1 \geq d + 1$ where d is the number of integral dead time instances in the digital model. Choosing N_1 to be smaller than the dead time will yield rows of zeros in the $\mathbf{G}$ matrix from (2.9). This, however, is a problem. Having d initial rows of zeros in the $\mathbf{G}^T\mathbf{G}$ matrix will create n zeros at the start of the $\mathbf{K}$ vector from (2.10) which do not contribute to the calculation of the optimal control increment.

In addition, for the case of a null control weighting variable ($\lambda = 0$), having rows of zeros in the $\mathbf{G}$ matrix will cause the matrix to become singular and therefore not invertible. As the control law relies on inverting $(\mathbf{G}^T\mathbf{G} + \lambda\mathbf{I})$, this can cause the controller to attempt to produce infinitely large control signals. This problem is solved by choosing the control weighting variable to be non-zero, positive such that the inversion is dependent on $det(\mathbf{G}^T\mathbf{G} + \lambda\mathbf{I})$ which is non-singular for positive λ.

Consider a first order system with model:

$$(1 - 0.946z^{-1})y(t) = 0.1351z^{-10}u(t-1) + \frac{e(t)}{\Delta} \tag{5.1}$$

where the dead time is 0.5 seconds, $T_s = 0.05$ seconds and the rise time of the plant is 3.6 seconds. The chosen design variable were $N_1 = 1,\ N_u = 1,\ N_2 = 3$ and $\lambda = 0$. This yielded an unstable output as was expected due to the rows of zeros present in the $\mathbf{G}$ matrix. Increasing the initial control horizon to $d+1$ rectified the problem of $(\mathbf{G}^T\mathbf{G} + \lambda\mathbf{I})$ being singular.

As seen in Fig. 5.1, the response of the system was similar to that of the well-known deadbeat controller [13], in that all the closed loop poles were placed at the origin. This

was confirmed by analysis of the characteristic equation (2.15).

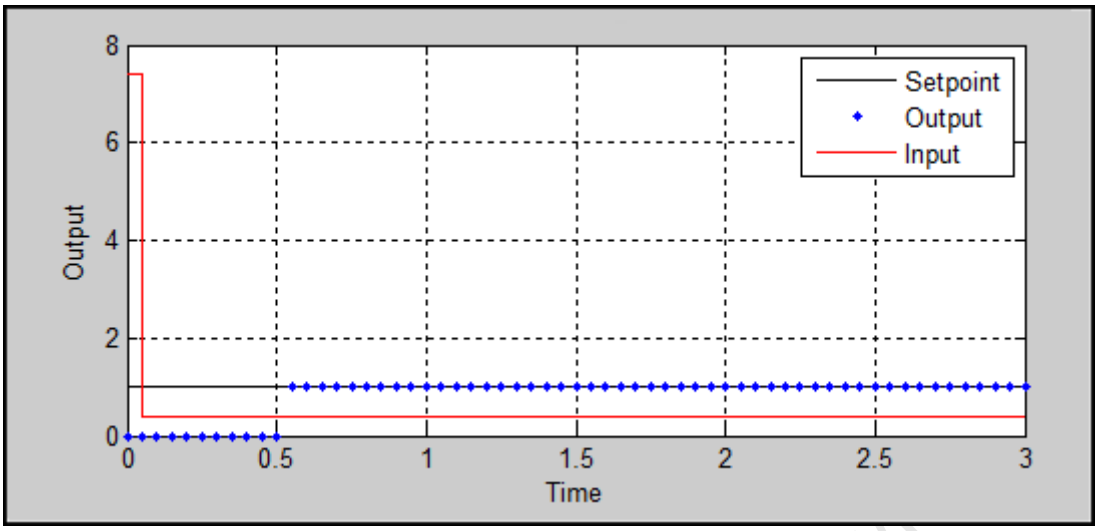

Figure 5.1: Dead Beat Response

5.2 Non-Integral Dead Time

The modified z-transform differs from the standard transform in that it is applied specifically to systems with non-integral dead-time, that is dead-time that is not an integer multiple of the digital sampling time, T_s. For this class of systems, the modified z-transform creates a zero in the digital model which is an approximation of the behavior of the fractional dead-time [26].

5.2.1 The Modified z-Transform

A short example will be used to illustrate the relationship between the fractional dead-time and the zero that is created by the modified z-transform. Consider the class of first order with fractional dead-time systems of the form:

$$g(s) = \beta \frac{1}{\frac{s}{a} + 1} e^{-\tau s} \tag{5.2}$$

where β is the system gain, a is the inverse of the plant time constant and τ is the total dead-time. The dead-time, τ, is a combination of integral (dT_s) and non-integral (θ) dead-time and is described by $\tau = dT_s + \theta$ such that: $d \geq 0$, $d \in \mathbb{Z}$, $0 \leq \theta < T_s$. A step invariant transform (ZOH) will be applied to ensure the correct form of the digital model is generated. The digital model takes the form:

$$g_h(z) = \frac{(z-1)}{z}\mathbf{Z}\left[\frac{g(s)}{s}\right] \tag{5.3}$$

$$g_h(z) = \frac{(z-1)}{z}\beta\mathbf{Z}\left[\frac{a}{s(s+a)}e^{-(dT_s s)}e^{-(\theta s)}\right] \tag{5.4}$$

where $\mathbf{Z}$ is the z-transform operator. However, as the dead time of this system is non-integral, the modified z-transform must be used. This yields:

$$g_h(z) = \beta\frac{(z-1)}{z}z^{-n}\mathbf{Z}_m\left[\frac{a}{s(s+a)}e^{-\theta s}\right] \tag{5.5}$$

Evaluating (5.5) using the standard transform tables yields:

$$g_h(z) = \beta\frac{(z-1)}{z}\frac{z(1-e^{-amT_s}) + (e^{-amT_s} - e^{-aT_s})}{(z-1)(z-e^{-aT_s})}z^{-d} \tag{5.6}$$

A new variable, m, is defined by $m = 1 - \dfrac{\theta}{T_s}$, $1 > m \geq 0$. Therefore, the position of the zero created by approximating the behavior of the fractional dead-time is:

$$z = \frac{e^{-aT_s} - e^{-amT_s}}{1 - e^{-amT_s}} \tag{5.7}$$

A graph relating m and the zero positions is shown in Fig. 5.2 for arbitrary values of $a = 1$ and $T_s = 1$.

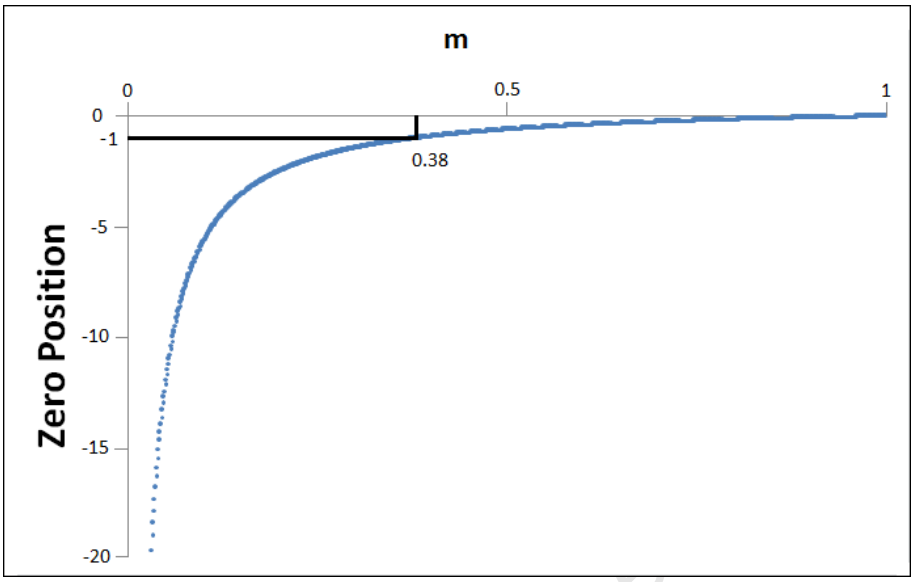

Figure 5.2: A Graph of m vs the Position of the Created Zero

Of note is that Fig. 5.2 shows only one sampling time interval. As τ is increased and moves from one integral dead time segment to the next, the graph repeats and as such is cyclical in nature. For these arbitrary values of a and T_s, a value of $0.38 \leq m \leq 1$ translates to a zero located in the unit circle. Varying the value of a and T_s result in a different value for m for the $z = -1$ point however the shape of the graph remains the same.

The concepts and results presented for systems with integral dead time are consistent for systems with non-integral dead time except for one distinct difference: modeling a continuous system with non-integral dead time by use of the modified z-transform creates a zero that is included in the zero polynomial of the digital model.

Chapter 4 relates how pole/zero cancellation can occur with GPC. These concepts will be used to perform an analysis of how the input control weighting, λ, the non-integral

system dead time, θ, and the system gain, β, affect the closed loop pole positions.

5.2.2 The Effect of λ on the Closed Loop Pole Positions

The control weighting factor was shown to affect the position of the closed loop poles in Chapter 4. For a value of $\lambda = 0$, pole/zero cancellation occurred and the closed loop poles were positioned on top of the open loop zeros. As λ was increased to ∞, the closed loop poles tended towards the open loop pole positions along loci similar to that of a standard root locus but in reverse. The following example shows the effect of varying λ.

Consider the first order system with fractional dead time:

$$g(s) = \frac{1}{10s+1}e^{-8.6s} \tag{5.8}$$

which has a digital model of the form:

$$g(z) = \frac{0.03921z + 0.05595}{z - 0.9048}z^{-9} \tag{5.9}$$

when $T_s = 1$. The GPC control algorithm was applied with $\lambda = 0$ and $N_2 = 3$. The position of the created zero was calculated to be $z = -1.427$.

The closed loop pole positions, calculated from (2.15), were plotted on an extended root locus plot for varying $\lambda = [0\ 3^{-10}\ 1^{-9}\ 3^{-9}\ ...\ 1^{6}]$.

The closed loop system contains 10 poles and 10 zeros. Fig. 5.3 shows the path that two of the closed loop poles follow moving from the open loop zero positions to the open loop pole positions. The remaining closed loop poles are placed at the origin and are canceled by the zeros of the controller.

Markers A and B correspond to two of the closed loop poles when $\lambda = 0$. A value of

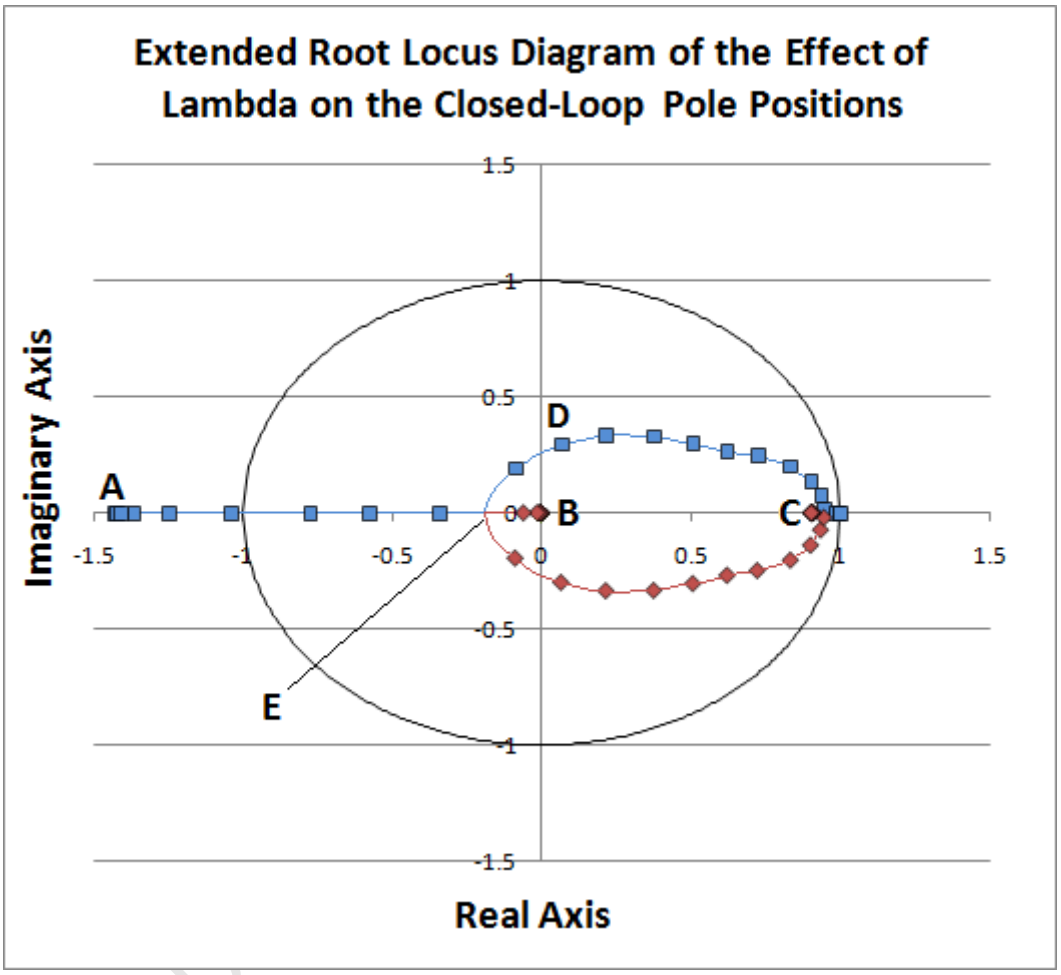

Figure 5.3: Extended Root Locus Relating λ to the Closed Loop Pole Positions

$\lambda \approx 1.2 * 10^{-7}$ corresponds to the point where the closed loop pole initially at A moves inside the unit circle and becomes stable but ringing.

As λ is increased, the closed loop poles move along the real axis and then breaks away at point E ($z \approx -0.1677;\ \lambda \approx 0.0002$) before returning to the real axis at the break in point near C and finally moving to the open loop pole positions at marker C. Marker D shows the closed loop pole positions corresponding to $\lambda = 0.001$. Marker C corresponds to $z = 0.9048$ and $z = 1$ where $\lambda \Longrightarrow \infty$. This indicates that the closed loop pole positions are very sensitive (at least initially) to changes in λ.

Thus far all research has indicated that increasing λ will draw all the closed loop poles of a system into the unit circle. A system with fractional dead time was theorized to determine if it was possible to encounter a controlled structure such that at least one closed loop pole would remain outside the unit circle for any value of λ. This system contained one negative zero outside the unit circle and two poles inside the unit circle. Digital modeling of that system yielded a model with an additional negative zero:

$$(1 - 1.35z^{-1} + 0.405z^{-2})y(t) = (1 + 3.8z^{-1} + 3.6z^{-2})u(t-1) + \frac{\Delta}{e(t)} \tag{5.10}$$

Applying the GPC control structure yielded a controlled system with five sets of poles and zeros. Varying λ from 0 to 10^{10} for that system generated closed loop pole loci, three of which are shown in Fig. 5.4. The remaining two loci are not shown as the pole/zero set remains at the origin for all values of λ used.

The closed loop pole loci begin at the location of the open loop zeros, in this case at $z = 0, 0, 0, -2, -1.5$ and, as λ is increased, move to the open loop pole positions (the open loop was augmented with an integrator pole). An interesting observation is that the locus does an about-turn at $z = -1.4$ and $z = -0.4$ before continuing into the unit circle and terminating at the open loop pole positions. Despite the initial assumption that varying

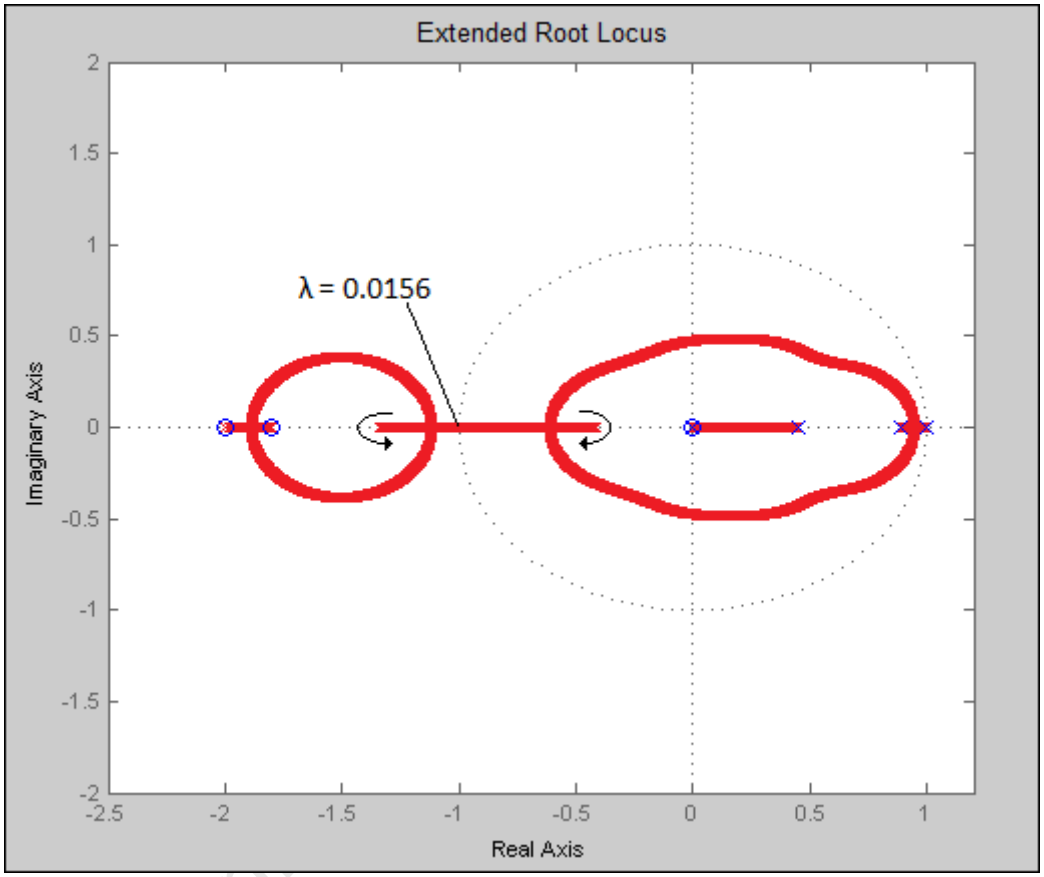

Figure 5.4: Extended Root Locus of λ vs Closed Loop Pole Positions

$\frac{1}{\lambda}$ had the same, or at least similar, effect as varying the system gain, this extended root locus shows that this is not the case. It is also noted that this system requires a value of $\lambda > 0.0156$ to pull the closed loop poles inside the unit circle.

5.2.3 The Effect of θ on the Closed Loop Pole Positions

Altering the fractional dead time changes the position of the created zero as shown in Fig. 5.2. This alters shape of the extended root locus including the break away point denoted by E in Fig. 5.3. The approximate break away points were calculated for an m value ranging from 0.1 to 0.9. In addition the minimum λ value that places the closed loop poles inside the unit circle was found and is noted in table 5.1.

m	**Position of Created Zero**	**Break Away Point**	λ
0.9	z = -0.106	z = -0.047	0
0.8	z = -0.238	z = -0.093	0
0.7	z = -0.408	z = -0.132	0
0.6	z = -0.634	z = -0.161	0
0.5	z = -0.951	z = -0.174	0
0.4	z = -1.427	z = -0.168	1.2×10^{-7}
0.3	z = -2.212	z = -0.143	1.7241×10^{-10}
0.2	z = -3.806	z = -0.120	1.8864×10^{-14}
0.1	z = -8.564	z = -0.052	ϵ

Table 5.1: Table of Break Away Points and Stabilizing λ for Variable m

For the last λ result, the numerical accuracy of the simulation program was insufficient to find a suitable value for λ and as such a value of ϵ can be assumed safely to place the closed loop poles in the unit circle.

The results shown in Tab. 5.1 indicate that the break away point is influenced the

most when the created zero is at $z = -1$. This indicates that the zero is most problematic near $z = -1$ and becomes less so as it moves along the real axis to $-\infty$.

The decreasing values of λ indicate that the rate at which the closed loop poles move along the extended root locus is faster the farther away the created zero is.

An implication is that, for a system with zero control weighting and a marginally stable, ringing output; a decrease of the fractional dead time may move the position of the created zero outside the unit circle resulting in an unstable output. Extending this idea, for a system with small λ, decreasing the fractional dead time would have the same effect.

5.2.4 The Effects of System Gain on the Closed Loop Pole Positions

Another variable that has an effect on the closed loop pole positions is the gain of the open loop system, more specifically when the gain is not modeled correctly. This topic has been addressed in literature [17] but will be discussed further here.

Consider the closed loop characteristic equation:

$$\phi_c = RA\Delta + \beta BSz^{-1} \tag{5.11}$$

where the gain of the open loop system is given by β and the sum of the coefficients of B divided by the sum of the coefficients of A equals one. In Chapter 4, it was shown that varying $1/\lambda$ has similar effects to varying the system gain with regard to the closed loop pole positions.

An analysis of the first order system (5.8) was performed, varying the system gain as shown in Fig. 5.5. θ was set to 0.5 and λ varied, such that $\lambda_1 > \lambda_2 > \lambda_3$, to produce the curves shown which are parameterized by the system gain. The open loop zeros are

situated at $z = -0.951$ and $z = 0$.

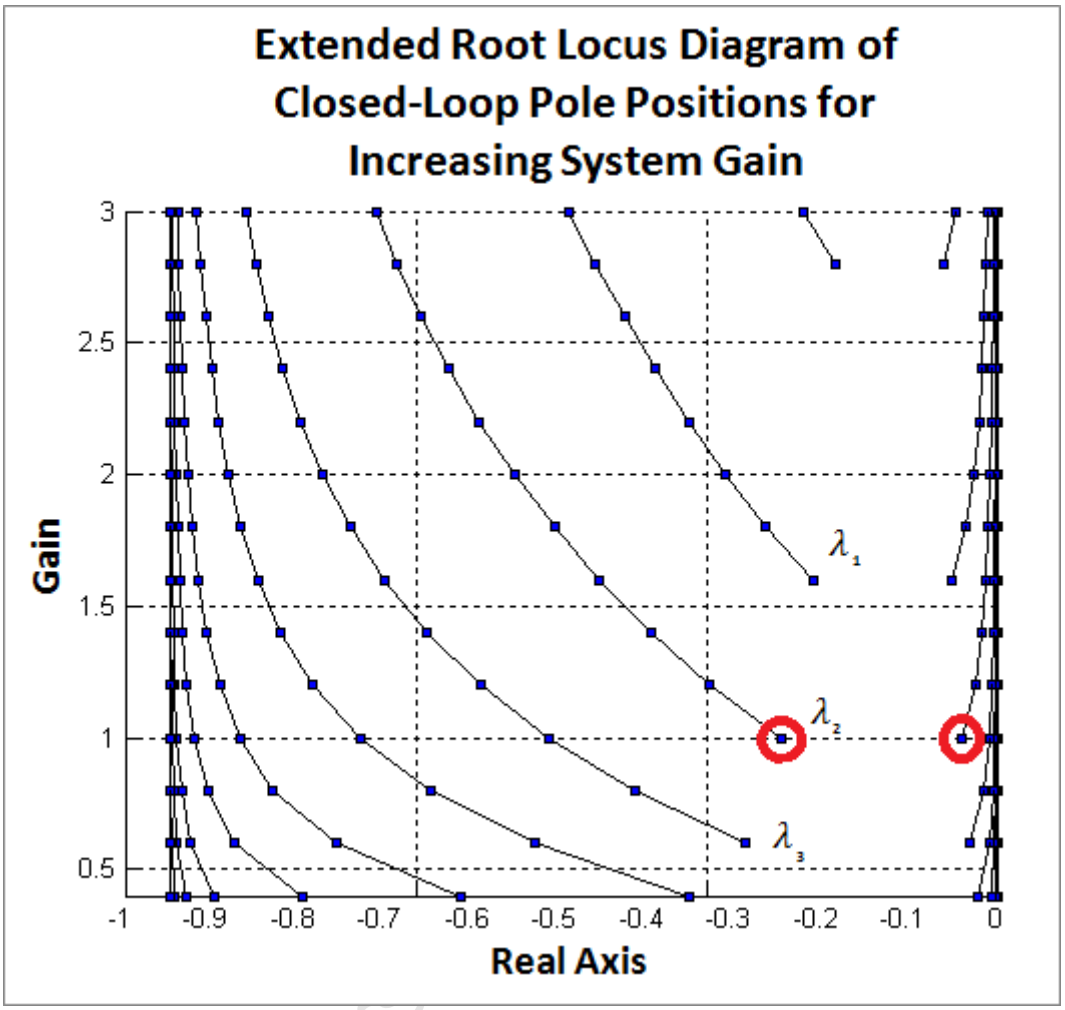

Figure 5.5: Extended Root Locus of System Gain vs. Closed Loop Pole Positions

This extended root locus plot contains the poles that are on the real axis before they meet and their complex parts become non-zero.

Consider the circled poles. For a gain of one and a particular value of λ, these closed loop poles sit at $z \approx -0.25$ and $z \approx -0.03$. For that value of λ_1, increasing the system gain to 3 pulls the closed loop poles towards the open loop zeros.

For marginally stable systems with non-minimum phase zeros, underestimating the

gain will cause the closed loop poles to be drawn back to the open loop zeros pulling them outside the unit circle, rendering the output unstable.

CHAPTER 6

Conclusions

6.1 General Conclusions

The literature survey concluded that non-minimum phase systems were easily solvable using a variety of methods, specifically pole placement and stable generalized predictive control. An analysis confirmed this and the topic was not investigated further.

Integral dead time proved to be one of the strengths of GPC, stemming from the predictive nature of the control algorithm. The problems associated with non-integral dead time were shown to be the same as that of the non-minimum phase systems in that pole/zero cancellation can occur. As such, this was easily solvable but still relevant to the dissertation.

GPC has been suc

Pole/zero cancellation was shown to exist for a zero value of λ. It was found that the extended root locus plot, comparing varying λ to the closed loop pole positions, acted similarly to varying the inverse of the gain for a standard root locus. However, further analysis showed that this was not the same effect merely a similarity for well behaved systems.

An analysis of the effects of the GPC tuning parameter, λ, the fractional portion of the system dead time, θ, and the open loop system gain, β, on the closed loop pole positions was performed. The results revealed that small initial changes in λ cause large movements in the closed loop pole positions along their extended root loci, allowing small values of

λ to prevent pole/zero cancellation, for example.

Underestimating the open loop system gain caused the closed loop poles to be drawn towards the open loop zero positions, potentially causing a system to become unstable. Decreasing θ such that the created zero moved towards negative infinity was shown to cause instability if the created zero was near $z = -1$ and the system marginally stable.

6.2 Future Work

- Chapter 3 yielded results in line with requirements of the Internal Model Control Principle. A disturbance rejection polynomial of the correct form is required for setpoint tracking of non-step functions. A property of the disturbance rejection polynomial is that it can be used to reject input disturbances and a comparison between designing the controller to reject disturbances and implementing an appropriate disturbance rejection polynomial would be performed. This work could be further extended to case studies for practical applications as this area is currently lacking in the literature.

- Chapters 4 and 5 relate how the closed-loop pole positions were very sensitive to small initial increases in λ. Further investigation could yield a quantifiable relationship between these two variables, allowing a more accurate choice of the value of λ, dependent on the control requirements. This could be further extended to finding quantifiable relationships between the closed-loop pole positions and the system gain, β, and fractional dead time, θ.

Bibliography

[1] S. Abe, M. Ogawa, M. Shoyama, T. Zaitsu, S. Obata, and T. Ninomiya. Dynamic characteristics of digitally controlled conveter with pole-zero-cancellation technique. In *TENCON 2010 - 2010 IEEE Region 10 Conference*, 2010.

[2] P. Albertos and R. Ortego. On generalized predictive control: Two alternative formulations. *Automatica*, 25:753–755, 1989.

[3] K. Astrom and B. Wittenmark. *Computer-Controlled Systems Theory and Design, Third Edition*. Prentice Hall, 1997.

[4] J. Becedas, I. Payo, V. Feliu, and H. Sira-Ramirez. Generalized proportional integral control for a robot with flexible finger gripper. In *Proceedings of the 17th World Congress The International Federation of Automatic Control*, 2008.

[5] S. Bennet. *A history of control engineering*. Peter Peregrinus Ltd, 1993.

[6] C. Bordons and E.F. Camacho. A generalized predictive controller for a wide class of industrial processes. *Control Systems Technology, IEEE Transactions on*, 6:372–387, 1998.

[7] M. Braae. *Control Engineering - 1*. UCT Press (Pty) Ltd, 1996.

[8] M. Braae. *Control Engineering - 2*. UCT Press (Pty) Ltd, 1996.

[9] Encyclopaedia Britannica. Ctesibius of alexandria. `http://www.britannica.com/EBchecked/topic/145475/Ctesibius-Of-Alexandria`, January 2011.

[10] E.F. Camacho and C. Bordons. *Model Predictive Control*. Advanced Textbooks in Control and Signal Processing. Springer, 2003.

[11] B. Choi and C. Choi. An effective pole-zero cancellation in feedforward controllers for nonminimum phase systems. In *American Control Conference, 1997. Proceedings of the 1997*, volume 6, pages 3907 –3908 vol.6, June 1997.

[12] D. Clarke and C. Mohtadi. Properties of generalized predictive control. *Autom*, 25:859–875, 1989.

[13] D.W. Clarke, C. Mohtadi, and P.S. Tuffs. Generalised predictive control - part i. the basic algorithm. *Automatica*, 23(2):137–148, 1987.

[14] D.W. Clarke, C. Mohtadi, and P.S. Tuffs. Generalised predictive control - part ii. extensions and interpretations. *Automatica*, 23(2):149–160, 1987.

[15] B. Francis and W. Wonham. The internal model principle of control theory. *Automatica*, 12:457–465, 1976.

[16] J.L. Garriga and M. Soroush. Model predictive control tuning methods: A review. *Industrial & Engineering Chemistry Research*, 49(8):3505–3515, 2010.

[17] H.W. Gomma. Stability analysis for generalized predictive control (gpc) with different classes of uncertain systems. In *Computer Aided Control System Design, 2006 IEEE International Conference on Control Applications, 2006 IEEE International Symposium on Intelligent Control, 2006 IEEE*, pages 984 –989, oct. 2006.

[18] S.F. Graebe and R.H. Middleton. Stable open loop poles: to cancel or not to cancel? In *Decision and Control, 1995., Proceedings of the 34th IEEE Conference on*, volume 1, pages 311 –316 vol.1, December 1995.

[19] R.A. Hess and Y.C. Jung. An application of generalized predictive control to rotorcraft terrain-following flight. *Systems, Man and Cybernetics, IEEE Transactions on*, 19(5):955 –962, sep/oct 1989.

[20] E. Katende and A. Jutan. Nonlinear predictive control of complex processes. *Industrial & Engineering Chemistry Research*, 35(10):3539–3546, 1996.

[21] R. Kennel, A. Linder, and M. Linke. Generalized predictive control (gpc)-ready for use in drive applications? In *Power Electronics Specialists Conference, 2001. PESC. 2001 IEEE 32nd Annual*, volume 4, pages 1839 –1844 vol. 4, 2001.

[22] B. Kouvaritakis, J.A. Rossiter, and A.O.T. Chang. Stable generalised predictive control: an algorithm with guaranteed stability. *Control Theory and Applications, IEE Proceedings D*, 139(4):349 – 362, July 1992.

[23] M. Mahfouf, A. Asbury, and D. Linkens. Unconstrained and constrained generalised predictive control of depth of anaesthesia during surgery. *Control Engineering Practice*, 11:1501–1515, 2003.

[24] R.H. Middleton and S.F. Graebe. Slow stable open-loop poles: to cancel or not to cancel. *Automatica*, 35(5):877 – 886, 1999.

[25] J.E. Normey-Rico and E.F. Camacho. *Control of Dead-time Processes*. Advanced Textbooks in Control and Signal Processing. Springer, 2007.

[26] K. Ogata. *Discrete-Time Control Systems*. Prentice Hall, 1994.

[27] A.W. Ordys and D.W. Clarke. A state-space description for gpc controllers. *International Journal of Systems Science*, 24(9):1727–1744, 1993.

[28] L.F.A. Pereira, G. Bonan, G. Thomaz, and D.F. Coutinho. Robust psc control design for sinusoidal tracking in uninterruptible power supplies. In *Power Electronics Conference, 2009. COBEP '09. Brazilian*, pages 389 –396, 272009-oct.1 2009.

[29] J. Rossiter. *Model-Based Preditive Control - A Practical Approach*. CRC Press, 2004.

[30] J. Rossiter and G. Valencia-Palomo. Feed forward design in mpc. In *Proceedings of the European Control Conference*, 2009.

[31] P. Scokaert. Infinite horizon generalized predictive control. *International Journal of Control*, 66:1:161–175, 1997.

[32] J. Sikaundi and M. Braae. Preventing pole-zero cancellation for improved input disturbance rejection in iterative feedback tuning systems. In *Novel Algorithms and Techniques In Telecommunications, Automation and Industrial Electronics*. Springer Netherlands, 2008.

[33] P. Thirusakthimurugan and P. Dananjayan. A new generalized predictive controller for the speed control of pmbldc motor. *Third International Conference on Information and Automation for Sustainability*, 1:7–12, 2007.

[34] T.T.C Tsang and D.W. Clarke. Generalised predictive control with input constraints. *Control Theory and Applications, IEE Proceedings D*, 135:451–160, 1988.

[35] L. Wang and J.A. Rossiter. Disturbance rejection and set-point tracking of sinusoidal signals using generalized predictive control. In *47th IEEE Conference on Decision and Control*, 2008.

[36] T. Wang, L. Zhou, P. Han, and Q. Zhang. Complete compensation for time delay in networked control system based on gpc and bp neural network. *2007 International Conference on Machine Learning and Cybernetics*, 2:637, 2007.

[37] V. Wertz, R. Gorez, and K. Zhu. A new generalized predictive controller application to the control of processes with uncertain dead-time. In *26th Conference on Decision and Control*, 1987.

APPENDIX A

Recursion of the Diophantine Equation

This appendix serves to prove that the Diophantine equation used in Chapter 2 can be recursively calculated to provide unique solutions that will be used to generate a prediciton equation.

Consider, for a given $A(z^{-1}) = 1 + a_1 z^{-1} + ... + a_{na} z^{-na}$ where na is the degree of $A(z^{-1})$ and $\tilde{A}(z^{-1}) = A(z^{-1})(1 - z^{-1})$, the following Diophantine equation can be solved for any arbitrary j ($j \geq 0, j \in \mathbb{Z}$):

$$1 = E_j(z^{-1})\tilde{A}(z^{-1}) + z^{-j}F_j(z^{-1}) \tag{A.1}$$

with E_j and F_j uniquely defined with degrees $j-1$ and $na-1$ respectively such that:

$$E_j(z^{-1}) = e_{j,0} + e_{j,1}z^{-1} + ... + e_{j,j-1}z^{-(j-1)}$$

$$F_j(z^{-1}) = f_{j,0} + f_{j,1}z^{-1} + ... + f_{j,na}z^{-na}$$

Consider for j and $j+1$ the Diophantine equations are:

$$1 = E_j(z^{-1})\tilde{A}(z^{-1}) + z^{-j}F_j(z^{-1}) \tag{A.2}$$

$$1 = E_{j+1}(z^{-1})\tilde{A}(z^{-1}) + z^{-(j+1)}F_{j+1}(z^{-1}) \tag{A.3}$$

Subtracting (A.2) from (A.3) yields the following:

$$0 = \tilde{A}(z^{-1})(E_{j+1}(z^{-1}) - E_j(z^{-1})) + z^{-j}(z^{-1}F_{j+1}(z^{-1}) - F_j(z^{-1})) \tag{A.4}$$

$$0 = \tilde{A}(z^{-1})(\tilde{E}(z^{-1}) + e_{j+1,j}z^{-j}) + z^{-j}(z^{-1}F_{j+1}(z^{-1}) - F_j(z^{-1}))$$

and, as the degree of $E_j(z^{-1})$ is $j-1$ and the degree of $E_{j+1}(z^{-1})$ is j:

$$E_{j+1}(z^{-1}) - E_j(z^{-1}) = \tilde{E}(z^{-1}) + e_{j+1,j}z^{-j} = e_{j+1,j}z^{-j}$$

Therefore:

$$0 = \tilde{A}(z^{-1})e_{j+1,j}z^{-j} + z^{-j}(z^{-1}F_{j+1}(z^{-1}) - F_j(z^{-1}))$$

$$0 = \tilde{A}(z^{-1})e_{j+1,j} + z^{-1}F_{j+1}(z^{-1}) - F_j(z^{-1})$$

and as the leading coefficient of $\tilde{A}(z^{-1})$ is one:

$$e_{j+1,j} = f_{j,0}$$

$$f_{j+1,i} = f_{j,i+1} - f_{j,0}\tilde{a}_{i+1} \qquad i = [0, 1, ..., na-1]$$

Extending (A.1) to include the $T(z^{-1})$ as shown in (2.2) does not alter the generation of $E_j(z^{-1})$ or $F_j(z^{-1})$ as $T(z^{-1})$ acts on $y(t)$ and $u(t)$. Therefore the predicted output, $y(t+j)$, will include filtered signals, $y^f(t)$ and $u^f(t)$ as in (2.5).