University of Alberta

Robustness versus Performance Tradeoffs in PID Tuning

by

Mohammad Sadegh Amiri

A thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Master of Science

in

Chemical Engineering

Department of Chemical and Materials Engineering

Fall 2009
Edmonton, Alberta

Examining Committee

Sirish L. Shah, Chemical and Materials Engineering

Charles Robert Koch, Mechanical Engineering

Stevan Dubljevic, Chemical and Materials Engineering

Abstract

Proportional, integral and derivative (PID) controller tuning guidelines in process industry have been in place for over six decades. Nevertheless despite their long design history PID tuning has remained an 'art' and no single comprehensive solution yet exists. In this study various considerations, with new and different perspectives, have been taken into account in PID tuning design. This study explores the issue of PID tuning from a practical point of view with particular focus on robust design in the presence of typical problems in process industry: process changes, valve stiction effects and unmeasured disturbances.

The IMC tuning rule is recommended for setpoint tracking, while in the case of regulation, a newly proposed tuning rule, based on a combination of IMC and Ziegler-Nichols method, is demonstrated to give satisfactory results. The results were evaluated by simulation and were also validated on a computer-interfaced pilot scale continuous stirred tank heater (CSTH) process.

Acknowledgements

All the praise is due to the Lord who created the glory of statics and dynamics. I need to confess that I am still wondering how this thesis is reaching to an ending point with around 6 months overshoot as different hardships, ups and downs came to the place especially for finding the general direction of this research effort. I would love to greatly thank God for enabling me to finish as without his especial will it would not pass the initial stages of completion.

I would like to thank my parents Mohammad and Fatima for their great support and encouragement in my life. The role they had during this period was essential and indispensable to this thesis.

I am very grateful to my supervisor Dr. Sirish L. Shah for his great supervision during my study at the University of Alberta. His rich insights and experience both in academia and mentorship were big help for me. The patience he showed in this period was remarkable. He taught me how to be a self-starter and find my own way at my work independently. His patience and eager for learning inspired me a lot during my MSc studies.

I would like to acknowledge the contribution of Dr. Dave Shook who humbly set up the OPC software which enabled me to read and write the data in a real time stream from the CSTH process using MATLAB. Furthermore, I learned good lessons from him during our joint work on a boiler control problem. That case study was quite interesting as it involved with inverse response in its behavior.

Many thanks goes to Dr. Harigopal Raghavan the former research manager of our group (now with Albian-Sands, Shell) who suggested a starting point for my work and helped me through regular meetings we had and valuable comments he made.

I would be glad to thank Dr. Mehrdad Sahebsara and Dr. Iman Izadi members of products and alarm management groups in Matrikon Company. Mehrdad helped

me on basic concepts of control in the initial stages of the research and gave the fundamental idea for the ZNIMC tuning rule proposed in this thesis. On the other hand, I had productive discussion meetings on both academic and non-academic topics with Iman who currently holds the research manager position in our group.

I would also like to thank Dr. Rohit Patwardhan and Dr. Chris McNabb, solutions and products managers of Matrikon Company. The discussions I had with Dr. Shah and them gave me valuable comments to develop the TUNIX Simulation Package which contributed greatly to this thesis and to the industrial presentations I made during my stay in Edmonton.

I should acknowledge Dr. M. A. A. Shoukat Choudhury, assistant professor of Bangladesh University of Engineering and Technology for providing the dynamic model for the valve stiction phenomenon.

I want to name Dr. Babu Joseph, the professor and chair of the department of chemical and biomedical engineering at University of South Florida for giving feedback and proofreading of part of this work.

I would like to thank Dr. Ramon Vilanova from Universitat Autonoma de Barcelona in Spain who gave the computer codes on his proposed robust tuning rule for tracking. The reproduction of his results and the discussions we had was quite instructive to me in my academic life.

Thanks a lot to my groupmates in the CPC group for the good group meetings we had at the university of Alberta.

I would like to thank the staff of Chemical and Materials Engineering Department at UoA. That includes Dr. R. Hayes the graduate advisor, Lily Laser and Marion Pitchard for their help in university related purposes.

Finally I want to especially thank my dear brother Mohammad Saeed who mentored me step by step and sometimes week by week for accomplishing such a big project. I owe him a lot for his brilliant ideas he taught me.

Contents

List of Tables

List of Figures

0.1 Nomenclature

a : oscillation amplitude
C : controller
d : relay amplitude
D : disturbance
DM : delay margin
e : error
E : expectation
GM : gain margin
I : identity matrix
IAE : integral of absolute error
IE : integral of error
ISE : integral of squared error
$ISTE$: integral of squared time and error
$ITAE$: integral of time and error
J : J factor
k_i : integral action
K_c : gain (controller)
K_p : gain (process)
K_u : ultimate gain
l : upper bound for uncertainty
L : loop
M_S : maximum sensitivity
M_T : maximum complementary sensitivity
MV : manipulating variable
N : noise
OP : controller output
P : process
P_n : nominal process
PM : phase margin
r : IMC filter order
S : sensitivity
S_D : disturbance sensitivity
S_U : control action sensitivity
T : complementary sensitivity
T_M : desired time constant (Vilanova)
T_s : sampling time
T_v : total variation
u : controller output
V : valve
Y : output
Y_{SP} : setpoint
z : robustness factor

0.2 List of Symbols

$||H||_2$: Euclidean norm

$||H||_\infty$: infinity norm

α : dimensionless factor$(\frac{\theta}{\tau})$

δ : impulse function

Δ_a : additive uncertainty

Δ_m : multiplicative uncertainty

ΔP : modelling error

ϵ : IMC filter time constant

ω : frequency

ω_{gc} : gain crossover frequency

ω_{pc} : phase crossover frequency

ω_{sc} : sensitivity crossover frequency

P_u : ultimate period

τ : time constant

τ_c : desired closed loop time constant

τ_d : derivative time

τ_f : derivative filter time constant

τ_i : integral time

θ : deadtime

θ_0 : original deadtime

Chapter 1

A Literature Review on PID Controller Tuning + Thesis Objectives

1.1 Preface

PID controllers have been used in industry for more than seven decades [31]. From the onset of their introduction, the problem of effective tuning of the controllers has been an active area of research and discussion. One of the first pioneers of PID controller tuning were Ziegler and Nichols who proposed a very simple and handy method for regulatory control in 1942 [47]. Since then, a large number of methods have been suggested to address this critical issue based on different design objectives. To this date, the number of proposed tuning rules for PID controllers has surpassed one thousand and still is increasing [31]. It has been reported that 293 tuning rules out of 408 separately explored tuning rules were proposed after 1992 [31]. This chapter covers some of important tuning rules found in the literature.

A survey in [19] revealed that of the PID controllers loops evaluated at a particular time in industry, around 30% were operating in the manual mode and furthermore 65% of them were producing less variance in the manual than in the feedback mode. These statistics clearly show poor tuning of PID controllers and that open loop control in some instances delivered better results than closed loop control (see 2.1.1).

The tuning rules covered here are mainly concerned with Single Input Single Output (SISO) systems. Other structures such as cascade and multiple inputs / multiple outputs (MIMO) are not covered in this briefing.

The handbook of PI and PID controller tuning techniques by O'Dwyer [30] has gathered a comprehensive collection of tuning rules and is certainly useful for looking up a specific tuning technique.

Some of the methods for PID controller tuning both in frequency and time domains are presented in this chapter; however the content is not exhaustive due to the high volume of research papers in this area. It may not be an exaggeration to say that the most difficult review in control theory belongs to PID controllers due to the large volume of literature in this field.

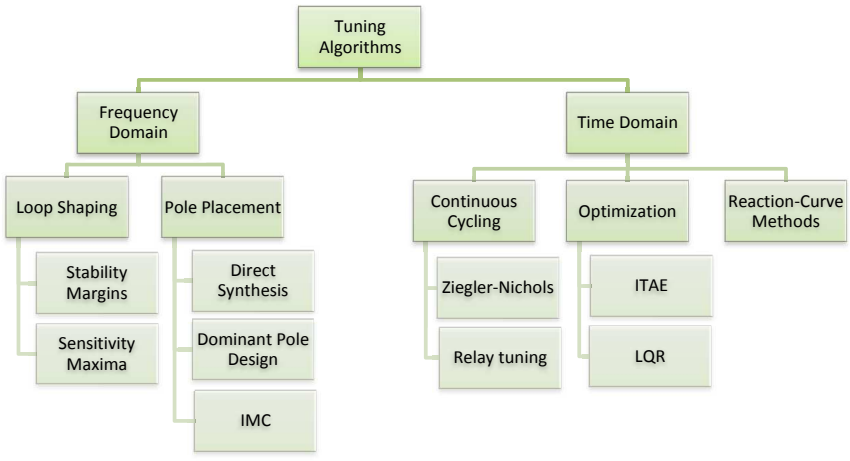
A couple of good review papers are used as the main framework of this chapter and the reader can refer to these for a more detailed analysis. This author would like to boldly acknowledge the great work of Unar et al. (1996) [41], van der Zalm et al. (2004) [42], and O'Dwyer (2006) [31].

In the literature for PID controller tuning, the author sees a need for classification of existing methods to present the tuning rules in a systematic manner.

1.2 Classical Classification of PID Controller Tuning Algorithms

PID controller tuning methods can be divided into time domain and frequency domain methods. For each class, different types of algorithms have been proposed. Time domain methods can be further subdivided to continuous cycling methods, optimization methods, and reaction-curve methods. For the frequency domain methods we have pole placement methods and loop shaping methods. Figure 1.1 shows a tree diagram of these structures.

Figure 1.1: Organization of various PID controller tuning methods

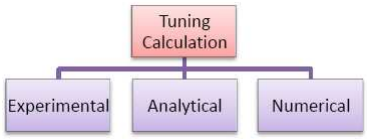


The classification is based on the approach that each tuning rule uses to calculate controller parameters. Nevertheless, overlapping between tuning categories does

occur and the mentioned classes are not mutually exclusive. For example, the Ziegler-Nichols technique belongs to both continuous cycling and reaction-curve method categories. The user can decide to choose any one based on the availability of data/model and ease of the use. In addition to the algorithms in the above table, there are also other nonconventional methods not reviewed here.

Rather than the inherent algorithm used in each tuning rule, it is also possible to classify the tuning rules based on the computational method. Figure 1.2 depicts such an organization.

Figure 1.2: Classification of PID controller tuning rules based on the computational approach



Some old and new tuning techniques in each of the classes mentioned are described in the following sections.

1.3 Time Domain Methods

1.3.1 Continuous Cycling Techniques

Continuous cycling technique consist of tuning rules which allow the process to reach to the verge of instability and by observing the output response it is possible to calculate tuning parameters.

The main advantage of this class is that they are model free and easy to perform. However, a number of disadvantages of such methods are listed below [31]:

- Instability needs to be observed by increasing the proportional gain
- Unique performance could not be achieved due to the empirical nature
- The trial and error nature of the method
- The need to upset the process variable during the test
- Confounding of limit cycle with instability bounds
- Doubt in safety and practicality of the algorithm

Modifications to the main theory have been proposed to use decay ratio of 0.25 or a phase lag of 135° instead of instability to mitigate some of the above concerns [30].

Ziegler-Nichols Class of tuning rules

This class of tuning rules (which includes the Ziegler-Nichols rule and its modifications) tries to find the controller parameters by measuring the ultimate gain and period of a system. The empirical procedure would be carried out as follows: First, the controller is set to proportional only mode. By gradually increasing the controller gain, one would finally encounter continuous cycling. By noting the period of oscillations in the critically stable mode (P_u) and the controller gain which is referred to as ultimate gain (K_u), it is feasible to calculate the PID controller parameters.

The Ziegler-Nichols tuning rule [47] was proposed in 1942 based on a number of trial and error experiments conducted to produce 1/4 decay ratio for regulatory control. That decay ratio corresponds to a damping factor of 0.21 [42]. The disadvantage of the Ziegler-Nichols tuning rule is that it is aggressive (but obviously that was not the drawback at that time) and it also suffers from poor robustness [1].

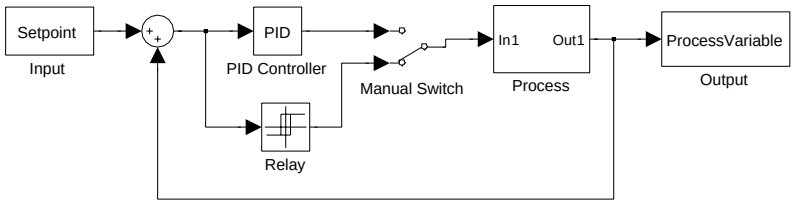
Hang et al. [21] proposed a refined Ziegler-Nichols rule in 1991. They incorporated a setpoint weighting factor to proportional action and claimed lower overshoots and better setpoint tracking for their tuning rule. It is possible to show that such a parameter adds a feedforward mechanism in addition to feedback.

Relay-Based Tuning

Sometimes instead of manually increasing the proportional gain, a relay generator is used to produce the continuous cycling situation. This method was proposed by Astrom and Hagglund in 1994 and was claimed as one of the most important method commercially used [24]. In this technique, the ultimate period is read in the same way as before while for ultimate gain the equality shown in Eqn. 1.1 can be used:

$$K_u = \frac{4d}{\pi a} \tag{1.1}$$

Figure 1.3: Relay-based tuning schematic



In the above equation, d and a represent relay and output oscillations amplitudes respectively. Using a relay has two advantages over the previous method [42]. The

first benefit of using a relay is that the process would not be driven to instability and furthermore other points on the Nyquist curve could be identified by adding hysteresis and integrators to the plain relay. A book on relay tuning was written by Yu in 2006 which can be served as a reference for more details [45].

1.3.2 Optimization Techniques

The idea for this class of tuning rule is to minimize a performance function of error and time with determined orders. In this way, parameters which give the optimal solution can be calculated. One general format of the optimization function is given in Equation 2.24. These tuning rules are normally optimized to give good performance for setpoint tracking or regulation in a system. The optimization methods are powerful [5], they can solve whatever criteria are specified by the designer. However, their drawback is that there can be several local minima for a system which could 'mask' the optimal solution. Computational intensity is another concern for this class of methods.

1.3.3 Integral of Error and Time Methods

These methods are formulated by optimizing (minimizing) a function of error and possibly time with corresponding powers (weights). (See section 2.6.1 for examples of these functions). When the speed becomes more important in the design, the cost function would be penalized by higher orders of time. The choice of those orders is hence a design factor based on problem conditions.

Lopez and coworkers [25] proposed a tuning rule for this class. However, among the set of tuning rules in this category, ITAE is claimed (1990) to yield better performance [32]. Nishikawa's tuning rule [29] also received attention in the literature as a powerful method based on an exponential time weighted ISE.

1.3.4 Linear Quadratic Regulator Methods

There are many tuning rules in the literature based on LQR techniques. The methods proposed by Athans [6], Williamson [44], Parker [34], Calovic and Cuk [12] are a few of these.

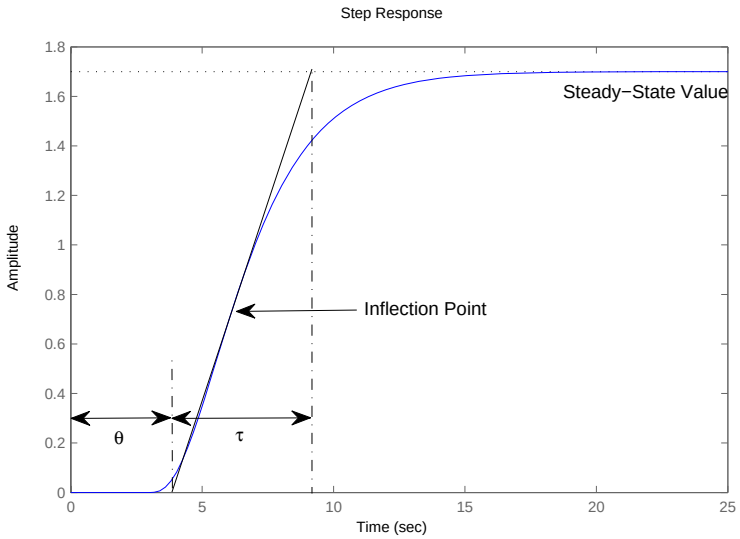
1.3.5 Reaction-Curve Techniques

This class of tuning rules, sometimes called *step response methods*, represents the simplest form of calculating tuning parameters. The process is excited by a simple

step test in open loop condition and some characteristic properties are then measured [31]. Figure 1.4 depicts the details of such a procedure.

These methods are simple to carry out and were mostly proposed in the beginning period of the tuning era. However, they do not always provide accurate models due to disturbances in the process. A large step test might be required to achieve the desired Signal to Noise Ratio (SNR) [36].

Figure 1.4: Reaction curve of a system



An example of this category can be found in [47]. The Cohen and Coon tuning rule [18] was proposed in 1953 for rejecting process disturbances and was designed again based on the quarter amplitude ratio criterion and calculates the controller parameters given an open loop step response. The tuning rule has the disadvantage of poor (oscillatory) performance in some conditions.

In 2001, Mann [26] introduced the normalized deadtime factor and categorized the FOPTD models into three regions based on that factor. It is possible to obtain the tuning parameters based on desired overshoot for each category. The drawback of this technique is that the tuning equations are rather complex.

1.4 Frequency Domain Methods

1.4.1 Pole Placement Techniques

These types of tuning rules rely on designing (placing) the closed loop poles of a process. There are different approaches for doing that; some place only the closed

loop poles, others may not only place the poles but also shape the zeros as well. In contrast, by using conventional PID controller, it may not be possible to place all of the poles, so only the dominant pole is placed for this scenario.

The design notion in this class is based on re-assigning the system's poles with faster modes. The drawback, however, is that some modes may become uncontrollable due to pole/zero cancellation and the performance degrades if they become excited.

Direct Synthesis

Direct synthesis allows an engineer to design the analytical solution for a closed loop dynamic of a given process. Equivalently, all of the zeros and poles of a system would be placed by using such techniques. It is straightforward to see that for first order plus time delay (FOPTD) and second order plus time delay SOPTD models, PI and PID controllers can help achieve the desired performance. The number of poles that can be placed is equal to the number of controller parameters. Therefore these techniques can be used for process models with the maximum order of 2 if a PID controller is selected [42].

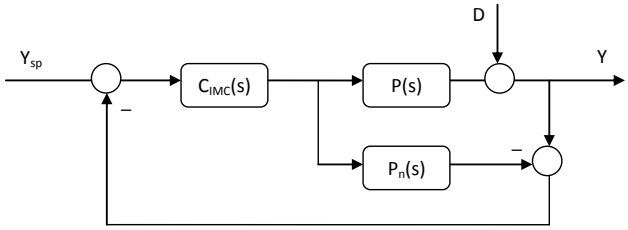
Dominant Pole Design

Contrary to the direct synthesis method, it is not always possible to assign all of the poles and zeros of a system using a PID controller, this case mostly happens in more complicated processes with higher orders. For such cases, the design would mainly focus on placing the slowest or dominant pole of a system.

Internal Model Control

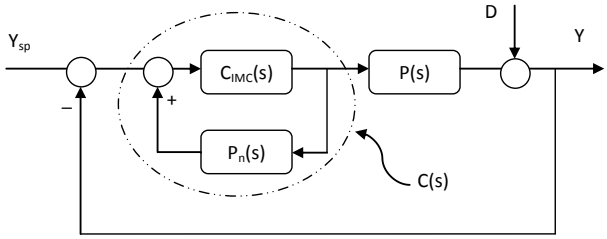
IMC design technique is one of the popular methods used in control theory. It has some attractive features which distinguish it from others as a powerful and robust design tool. Accordingly special emphasis is given to this method in this chapter and in the whole thesis. Consider the following IMC schematic:

Figure 1.5: Original IMC structure



In this block diagram, $C_{IMC}(s)$ represents the IMC controller while $P(s)$ and $P_n(s)$ represent the actual and nominal models of the process, respectively. Interestingly, the above diagram can be rearranged and redrawn as a conventional feedback loop as shown in Figure 1.6.

Figure 1.6: Re-arranged IMC structure



In this figure, the IMC controller and the conventional controller one can be converted to each other by the following equations:

$$C(s) = \frac{C_{IMC}(s)}{1 - C_{IMC}P_n(s)} \tag{1.2}$$

$$C_{IMC}(s) = \frac{C(s)}{P_n(s)[1 + C(s)]} \tag{1.3}$$

One of the excellent features of IMC formulation is that it can take into account the uncertainties inside the process model and hence it is worthy of attention in robustness analysis.

Using the above configuration, the dynamics from the setpoint variable (SP) and disturbance variable (DV) to the process variable (PV) can be explained via:

$$Y(s) = \frac{C_{IMC}(s)P(s)}{1 + C_{IMC}(s)\Delta P(s)}Y_{sp}(s) + \frac{1 - C_{IMC}(s)P_n(s)}{1 + C_{IMC}(s)\Delta P(s)}D(s) \tag{1.4}$$

Let $\Delta P(s)$ denote an additive model plant mismatch (i.e. $P(s) - P_n(s)$). From the setpoint tracking criterion ($\frac{Y(s)}{Y_{sp}(s)} = 1$) and the disturbance rejection criterion ($\frac{Y(s)}{D(s)} = 0$), we can deduce that the following relation should be hold:

$C_{IMC}(s) = P_n(s)^{-1}$ The IMC controller acts as exact inverse of the nominal process (1.5)

The procedure to design an IMC controller is explained in the next section.

IMC Design Procedure [10]

Consider a process with a transfer function of

$$P_n(s) = \frac{N(s)}{D(s)} e^{-\theta s} \tag{1.6}$$

in which $N(s)$ and $D(s)$ represent the numerator and denominator polynomials of the process. Assuming that the process is stable and there are no RHP zeros, the IMC controller has the following structure:

$$C_{IMC}(s) = \frac{D(s)}{N(s)(\epsilon s + 1)^r} \tag{1.7}$$

Here r denotes the relative order of $\frac{N(s)}{D(s)}$ [1] and ϵ is the so called IMC filter time constant which plays an important role in the robust design of a system. To avoid excessive amplification of noise, a lower limit on ϵ needs to be imposed based on [10]:

$$\epsilon \geq \left\{ \lim_{s \to \infty} \left(\frac{D(s)N(0)}{20 s^r N(s) D(0)} \right) \right\}^{1/r} \tag{1.8}$$

As mentioned earlier, ϵ remains a user-defined parameter based on design requirements (for example to accommodate model uncertainty). For a special case of a FOPTD process, the IMC controller has the form $\frac{\tau s+1}{K_p(\epsilon s+1)}$.

The design method for models with RHP zero and also inherently unstable processes is a little bit different and can be found in the literature ([28] and [10]). One issue involving IMC technique is that it does not always result in a PID controller structure. This necessitates some approximations to derive the tuning rules.

The IMC concept was originally introduced in 1980 by Morari and his colleagues work [27]. Other researchers in this field are Rivera et al. [35], Chien et al. [14] and Chia et al. [13]. IMC methodology is model based and is claimed to result in better responses compared to other methods [35]. In [35], the following advantages are mentioned for IMC:

- Direct incorporation of uncertainty in IMC design
- The user defined desired closed loop time constant can manipulate the trade-off between control performance and robustness against model plant mismatch

A list of IMC-based controller settings was developed by Chien and Fruehhauf in 1990 [15]. IMC has proved to be a popular and robust tuning rule in both theory and

[1] The relative order of the polynomial fraction of (A(s))/(B(s)) is defined as the order of B minus the order of A

practice. One of the advantages of IMC is the introduction of the desired closed loop time constant which can be used by operators to manipulate the degree of robustness. The IMC tuning rule is suitable for setpoint tracking; however, it may perform poorly against disturbances.

Vilanova (2008) [43] proposed a robust IMC based ISA tuning rule for setpoint tracking. Two user defined parameters are incorporated, T_M (desired closed loop time constant) to manipulate the speed of response and the z factor for direct tuning of robustness. The main advantage of the tuning rule is an increased degree of freedom for the design, but at the same time that can also be assumed as its disadvantage. There are no clear guidelines for specifying these parameters which may be sometimes confusing. The tuning rule is discussed more thoroughly in chapter 3.

Shamsuzzoha and Lee (2007) [37], reported that IMC demonstrates sluggish disturbance rejection, especially when the deadtime to time constant ratio is small. To alleviate this problem they proposed an IMC-PID tuning method for improved disturbance rejection. It was claimed that the proposed rule provides better response for lag-dominant processes where the controllers are tuned to have the same degree of robustness based on maximum sensitivity.

An extension of IMC design for digital PID controllers has been presented by Zhu and Saucier [46].

1.4.2 Loop Shaping Techniques

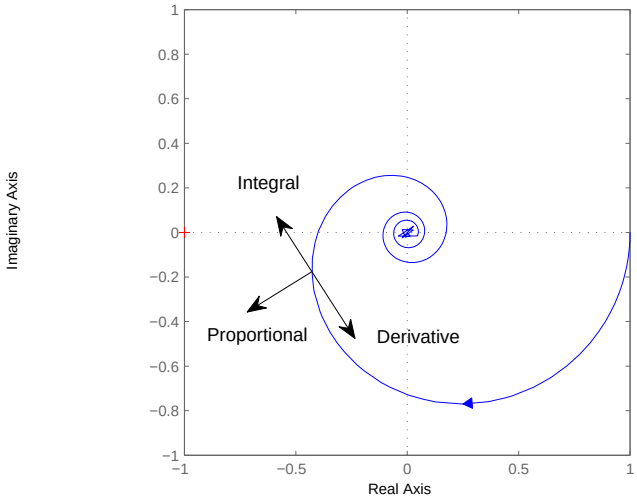
These techniques try to shape the Nyquist curve of a given system. Different frequency domain indices can be used such as gain and phase margins and sensitivity maxima to obtain the desired frequency behavior. Those indices normally impose circular constraints on the Nyquist plain and the curve should be designed to fall outside those regions. Figure 2.7 shows an example of a loop shaping design. It is normal to solve an optimization problem simultaneously with the loop shaping.

The effects of controller parameters on a Nyquist plot are graphically shown in Figure 1.7.

Astrom and Haglund developed their (A-H) tuning rule in 1984 [3]. The methodology is based on the effect of each controller parameter in shaping the Nyquist curve as described in [5]. The advantage of this method is that the controller can be designed based on a desired gain and phase margin and does not need a process model, however, it has been reported that it will not perform well in cases of high deadtime [41].

Astrom and Hagglund [4] also suggested the A(strom)MIGO tuning rule for in-

Figure 1.7: The Nyquist curve shaped by PID controller parameters



creased robustness. The tuning rule is a modification of the MIGO tuning rule which alleviates the problem of the derivative cliff [5]. The original term stands for M-constrained Integral Gain Optimization and therefore tries to maximize the integral gain which is an important factor in regulatory control. The tuning is calculated subject to the constraint of 1.4 for maximum sensitivity.

Ho et al. [22] also proposed a tuning rule based on gain and phase margins for PI and PID controllers. Due to the nonlinear relation of stability margins in respect to controller parameters, some approximations are made in the tuning rule to get an analytical solution. The authors suggested a PI controller for FOPTD and a PID controller for SOPTD models. It has been reported that the performance of this tuning rule is comparable to IMC [41].

A frequency domain method proposed by Barnes et al. [7] makes use of the least squares and tries to fit the system to the desired open loop plot.

1.5 Thesis Objectives

There are a large number of PID controller tuning rules in textbooks and generally in the literature for the control design. However for an engineer it is difficult to decide upon which to employ. The reason is simple, there is less attention to develop a guideline and manual for making the decision than the proposition of the tuning rules themselves.

The focus of this MSc thesis is on analytical tuning rules of PID controller for FOPTD processes and the theme is to evaluate the robustness of such techniques. A quite recent robust technique (ISA-based IMC by Vilanova [43]) was picked as the initial starting point for this arena. During the reproduction of the results in the Vilanova's paper, the idea of using a negative derivative action was suggested by the tuning rule for increased robustness. A considerable amount of time was spent on that idea to show the usefulness of application of such a structure. The results are presented in the next chapter 3.

The next effort was to evaluate and compare some of the tuning rules in the literature. It is difficult to tell which tuning rule is the best even when one considers the same design objectives in section 4.2. The tuning rules are still evolving and on the other hand research papers only provide comparisons with a handful of well known techniques. A lack of standard benchmarks and a comprehensive evaluation on proposed tuning rules are among the shortcomings in this context [31]. The other fact on the ground which is often ignored is the presence of process abnormalities. Those may include valve stiction, Model Plant Mismatch and noisy measurements. The describing equations for some of these phenomena (especially valve stiction) are quite complex and an analytical solution may not be feasible to be obtained. To perform such a global evaluation, the TUNIX Simulation Package was developed under the MATLAB environment to carry out and organize computer simulations. The lack of such a standard and convenient tool is highly felt in this regard. For more info refer to Appendix D.

This research effort deals with PID controller tuning in a different and unique way. First, the tuning rules are categorized based on servo and regulatory requirements. Then, for each category an evaluation and comparison is conducted to analyze the robustness behavior. A new tuning rule is also proposed for regulatory control to fill the gap which was seen by the industry practitioners. The results in the simulation environment are furthermore validated on a real pilot process. The content was organized to be practical and results seemed to be of interest for some of the engineers working in industry. Thus this thesis contributes comprehensiveness and practicality

to the field of PID controller tuning.

1.6 Organization of the Thesis

In chapter 1 we discussed about various PID controller tuning techniques in both frequency and time domains. In chapter 2, some relevant basic theory for this thesis is covered. In chapter 3 one of the robust techniques proposed by Vilanova is described. Some well-known techniques in a real simulation environment are evaluated and compared in the presence of prevalent process abnormalities in chapter 4. General tuning guidelines for PID controller are also suggested in this chapter. In chapter 5 some of the simulation results are validated on a real continuous stirred tank heater (CSTH) case study. Conclusion are drawn and future directions for this work are suggested in chapter 6.

Organization of the appendices is as follows: Appendix (A) consists of the paper titled "Guidelines for Robust PID Controller Tuning of FOPTD Processes" which has been submitted to the CSChE 2009 conference in Montreal. Appendix (B) covers the details of identification of the CSTH process using first principle models and experimental approaches; Appendix(C) contains an introductory talk about MATLAB's OPC toolbox which is useful for connecting the MATLAB environment to read/write real process variable values. The "TUNIX Simulation Package" an interactive and simple GUI designed by the author is introduced in Appendix (D) to automate and facilitate the study throughout this research work.

Chapter 2

Introduction to Theory

2.1 The Control Problem

The basic concept of control is to maintain output (sometimes called the process variable) as close as possible to the setpoint or the target. To illustrate this, consider the following conventional feedback loop:

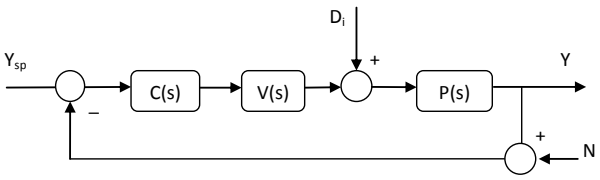


Figure 2.1: Schematic of a feedback single input - single output (SISO) loop

For a linear system, which is the case in this study, the dynamics could be shown as transfer functions using the Laplace transform and the Laplace variable s. The symbols in the figure correspond to:

Y_{sp} : Process setpoint	Y : Process variable or Output	
P : Process model	C : Controller model	V : Valve model
D_i : Input (Load) disturbance	N : Measurement noise	

The dynamics from different inputs to the process output are given by the following equations:

$$G_{Y_{sp}\to Y} = \frac{Y(s)}{Y_{sp}(s)} = \frac{C(s)V(s)P(s)}{1 + C(s)V(s)P(s)} \tag{2.1a}$$

$$G_{D_i\to Y} = \frac{Y(s)}{D_i(s)} = \frac{P(s)}{1 + C(s)V(s)P(s)} \tag{2.1b}$$

$$G_{N\to Y} = \frac{Y(s)}{N(s)} = \frac{-C(s)V(s)P(s)}{1 + C(s)V(s)P(s)} \tag{2.1c}$$

As seen here, the stability of all transfer functions is determined by a single characteristic polynomial. However, the numerator terms are different and should be dealt appropriately according to control objectives.

2.1.1 The Control Dilemma

Consider Figure 2.2 where the process output and control action for a first order plus time delay (FOPTD) process are depicted. Two cases were investigated: one where no feedback is present and the other where a PID controller takes care of the process control.

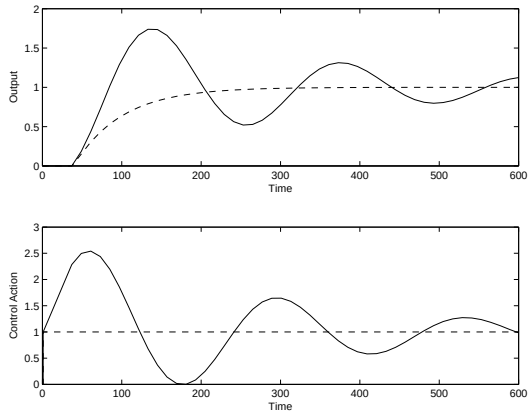


Figure 2.2: Manual control gives better dynamics!

Interestingly, the oscillatory response belongs to the feedback system and the smooth output is obtained using no controller. Hence, using no controller is better than using one. This indeed happens when the controller is mistuned with respect to the process model. The phenomenon highlights the need for reliable and robust tuning method.

2.2 Choice of Process and Controller

The main focus of this thesis is on First Order Plus Time Delay (FOPTD) systems. FOPTD models are quite well used for approximating process behaviors and are most

suitable for overdamped or nonoscillatory dynamics. The input-output behavior can be represented by:

$$P(s) = \frac{K_p}{\tau s + 1} e^{-\theta s} \tag{2.2}$$

The process gain, time constant, and dead time are denoted by K_p, τ, and θ, respectively, in the above equation. FOPTD models are also sometimes known as three parameter models.

The PID controller dynamics in the time domain is shown by the following differential equation:

$$u(t) = K_c \cdot (e(t) + \frac{1}{\tau_i} \int_0^\infty e(t)dt + \tau_d \frac{de(t)}{dt}) \tag{2.3}$$

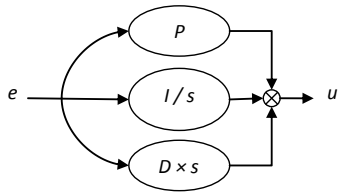


Figure 2.3: Various actions of an ideal PID controller, $P = K_c$, $I = K_c/\tau_i$, and $D = K_c\tau_d$

Kc is the controller gain, τ_i is the reset or integral time, and τ_D is rate or derivative time. The three different actions are illustrated in Figure 2.3.

The PID controller structure in this study is chosen as a noninteractive industrial ISA controller [43] with a derivative filter having a transfer function of:

$$C(s) = K_c \cdot (1 + \frac{1}{\tau_i \cdot s} + \frac{\tau_d \cdot s}{\tau_f \cdot s + 1}) \tag{2.4}$$

Each PID controller parameter has a philosophy behind its usage; while the controller gain takes care of the speed of response, integral time eliminates the offset and furthermore keeps a past memory of the system's behavior. The derivative part is used to add an anticipatory feature for future dynamics. The secret behind PID controller success is its simplicity to be easily understood by operators and its ability to do the control task for most of control loops using only the three tuning parameters. A simplistic design rule of thumb suggests settings such as $\tau_i/\tau_d = 4$ [1] and $\tau_f/\tau_d = 0.1$ [23].

The high frequency gain for above controller is computed as:

[1] The same source [23] claims a value close to 2.5 as an optimal solution for most stable plants

$$\lim_{\omega \to \infty} |G_c(j\omega)| = K_c \cdot (1 + \frac{\tau_d}{\tau_f}) \tag{2.5}$$

To avoid excessive amplification of noise inside the feedback loop Brosilow and Joseph [10] suggest a controller design that ensures $|\frac{G_c(j\infty)}{G_c(0)}| < 20$.

The expanded form of the controller equation is sometimes useful in PID controller design:

$$C(s) = \frac{a_2 \cdot s^2 + a_1 \cdot s + a_0}{b_2 \cdot s^2 + s} \tag{2.6}$$

The two forms are equivalent and can be converted easily using the following equations:

$$\begin{aligned} K_c &= a_1 - b_2 a_0 & \tau_i &= \frac{a_1}{a_0} - b_2 \\ \tau_d &= \frac{a_2}{a_1 - b_2 a_0} - b_2 & \tau_f &= b_2 \end{aligned} \tag{2.7}$$

2.2.1 Time Delay Approximation

One of the difficult factors in process control is the time delay due to the transportation or analysis lag. This is an irrational term from a mathematical point of view. However, it can be approximated in terms of rational transfer functions using Taylor and Pade expansions. First and second order Taylor series expansions for time delay can be written as:

$$e^{-\theta s} \approx 1 - \theta s \approx \frac{1}{1 + \theta s} \qquad \text{Fist Order Taylor} \tag{2.8a}$$

$$e^{-\theta s} \approx \frac{1}{1 + \theta s + \theta^2 s^2} \qquad \text{Second Order Taylor} \tag{2.8b}$$

First and second order Pade expansions for time delay can also be written as:

$$e^{-\theta s} = \frac{e^{-\frac{\theta}{2}s}}{e^{+\frac{\theta}{2}s}} \approx \frac{1 - \frac{\theta}{2}s}{1 + \frac{\theta}{2}s} \qquad \text{First Order Pade} \tag{2.9a}$$

$$e^{-\theta s} \approx \frac{1 - \frac{\theta}{2}s + \frac{\theta^2}{12}s}{1 + \frac{\theta}{2}s + \frac{\theta^2}{12}s} \qquad \text{Second Order Pade} \tag{2.9b}$$

Pade approximations are twice as accurate as Taylor approximations, meaning a 2^{nd} order Pade would do the job of a 4^{th} order Taylor approximation.

Figure 2.4 compares the trends of those approximations against a real function. The 2nd order Pade approximation seems to give enough accuracy up to critical frequency, however it has a complicated form which makes its use impractical in some techniques.

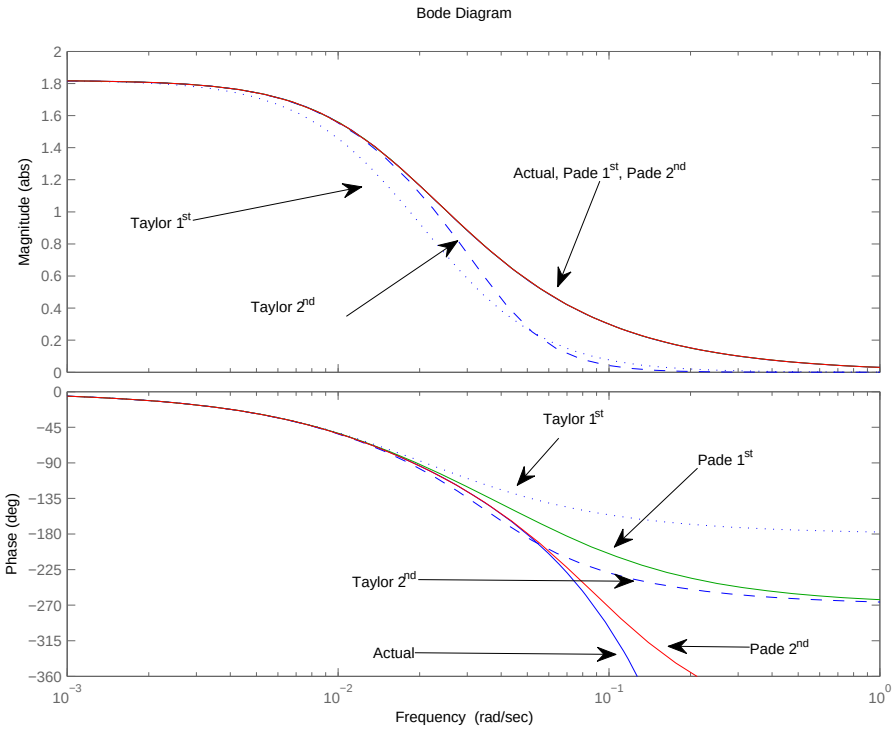


Figure 2.4: Accuracy of deadtime approximations against a real trend for the transfer function: $G(s) = \frac{1.82e^{-38s}}{60s+1}$

2.3 Model Order Reduction

Many chemical processes behave in a complex way. That is they behave nonlinearly or have a high order linear model. Nevertheless, it is feasible to approximate them with a low order model which captures most of their dominant dynamics. In most of the studies done on these processes, three major classes of models are used: Integrating, FOPTD, and SOPTD (second order plus time delay). The remaining problem is how to approximate our process effectively. Based on the situation we propose two solutions:

1. If identification data is available

   In this case we first try to identify a model with low order, if the correlation tests fail, we might consider two options. One is to let the data fit our desired model even if it violates the confidence intervals. The second wiser option in the mind of the author is to isolate the slower dynamics (which are of interest in chemical engineering) and feed them back to the identification process. For this purpose a low-pass filter could be used.

2. If only a high order LTI transfer function is available
   For this case, we follow the half rule mentioned by Skogestad. This method does not address nonlinear and integrating processes and another technique should be used instead.

2.3.1 Skogestad's Half Rule [38]

This technique depends on defining an effective time delay for our model. As mentioned, the time delay could be approximated by a first order Taylor expansion (see Equation (2.8a)). That indeed means that the lags and leads inside a transfer function are convertible to an equivalent amount of delay and vice versa.

In this method, the effective delay will consist of the true time delay, process leads, half of the dominant time constant, and is inclusive of all other minor ones. The exact algorithm is as follows;

Let us assume that the process is represented by:

$$G(s) = K_p e^{-\theta_0 s} \frac{\prod_{j=1}^{m} (\beta_j s + 1)}{\prod_{i=1}^{n} (\alpha_i s + 1)} \tag{2.10}$$

where α_i are process time constants in a descending order dominated by α_1. It is desirable to approximate the above model by an SOPTD system with a transfer function of the type:

$$G(s) = \frac{K_p e^{-\theta s}}{(\tau_1 s + 1)(\tau_2 s + 1)} \tag{2.11}$$

Clearly, a FOPTD model could be obtained by setting $\tau_2 = 0$

- Calculating the time constants
  for a FOPTD system we consider:

$$\tau_1 = \alpha_1 + \frac{\alpha_2}{2} \qquad \textit{and} \qquad \tau_2 = 0 \tag{2.12a}$$

$$\theta = \theta_0 + \frac{\alpha_2}{2} + \sum_{i=3}^{n} \alpha_i \tag{2.12b}$$

  For an SOPTD we could modify the equations to:

$$\tau_1 = \alpha_1 \qquad \textit{and} \qquad \tau_2 = \alpha_2 + \frac{\alpha_3}{2} \tag{2.13a}$$

$$\theta = \theta_0 + \frac{\alpha_3}{2} + \sum_{i=4}^{n} \alpha_i \tag{2.13b}$$

- Adjustment for negative zeros:
  The associated time constants are added to the delay calculated in the previous section:
  $$\theta' = \theta + |\beta| = \theta - \beta \tag{2.14}$$

- Adjustment for small positive zeros:
  The associated time constants are simply subtracted from the delay calculated in the previous section
  $$\theta' = \theta - \beta \tag{2.15}$$

- Adjustment for large positive zeros (if $\beta > \theta/2$):
  In this case, the zero should not be subtracted from the modified delay. Instead it could be canceled by a larger time constant:
  $$\frac{\beta_j s + 1}{\alpha_i s + 1} = \frac{1}{(\alpha_i - \beta_j)s + 1} \tag{2.16}$$

It is interesting that this technique is derived intuitively. Half of the second dominant pole is equally divided between the effective time constant and the delay. In this way a faster response could be expected for tuning the controller. It was claimed that this technique gives good results [38].

2.4 Process Abnormalities

The dynamics of PID controllers designed using some of conventional methods show excellent and satisfactory trends in simulation. Nevertheless, the same structure in a real process may not function accordingly. This deviation from ideal conditions is due to abnormalities and nonlinearities present in the process which are not considered in the design process. They may include valve stiction and saturation, model uncertainty, and measurement noise. Modeling valve stiction is not a trivial job; however a brief explanation is borrowed from [17] to complement the analysis. The model designed based on this definition is used for process simulation. The interested reader can refer to the book by Choudhury et al.(2008) [16] for a more comprehensive discussion.

2.4.1 Valve Stiction

One of well-known abnormalities prevailing in the process industry is valve stiction. The term "stiction" was originally created by combining the two key terms of "static

friction". Based on a study by Bialkowski [8] 30% of all oscillatory responses in an average plant are due to valve stiction, thus the impact of this phenomenon needs to be addressed.

The first step in understanding this nonlinear phenomenon is to devise a model which can be used for simulation. Modeling requires an in depth knowledge about the real mechanism of stiction, such as mass of moving parts, spring constant and other forces inside a conventional valve. Some approaches to this can be found in the literature and it is initially desirable to define some of the related terminologies quoted from Choudhury, Thornhill and Shah [17]. These are the standards used by the American National Standard Institute (ANSI) to describe the valve stiction.

- ***Backlash***: "In process instrumentation, it is a relative movement between interacting mechanical parts resulting from looseness when the motion is reversed."

- ***Hysteresis***: "Hysteresis is that property of the element evidenced by the dependence of the value of the output, for a given excursion of the input, upon the history of prior excursions and the direction of the current traverse"

- ***Deadband***: "In process instrumentation, it is the range through which an input signal may be varied, upon reversal of direction, without initiating an observable change in output signal."

- ***Deadzone***: "It is a predetermined range of input through which the output remains unchanged, irrespective of the direction of change of the input signal."

The effects of these parameters can be seen in Figure 2.5(a). The stiction model proposed by [17] is used in this thesis. Figure 2.5(b) explains the mechanism and stages of a full cycle valve stiction.

Basically, stiction causes the manipulating variable (MV) to behave differently from the the controller output (OP) (see the block diagram in Figure 2.1). Stiction only happens when the valve comes to a stop which is the case when the control action changes directions.

Figure 2.5(b) consists of four consecutive phases: deadband, stickband, slip-jump and a moving phase. To describe a full cycle, suppose that the valve is stuck and at rest at point (A). As the controller output increases, the valve does not move until the controller output overcomes the deadband (B) in addition to stickband (C). When the controller output reaches this point the valve undergoes a little slip-jump until point (D) and it continues through the moving phase to point (E). The valve could

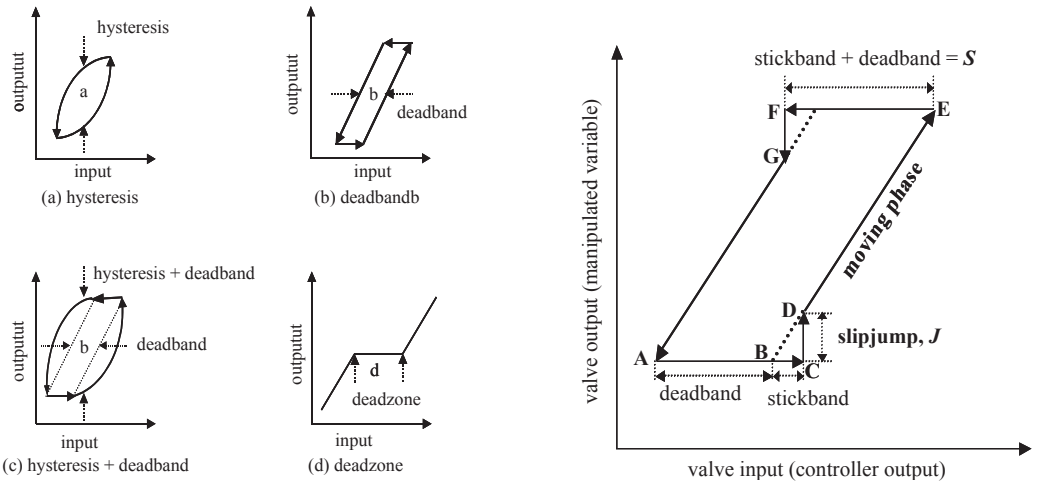


Figure 2.5: (a) Dynamics of hysteresis, deadband, and deadzone (adopted from ANSI/ISA-S51.1-1979) and (b) The relationship between OP and MV in the presence of valve stiction, Courtesy of Choudhury et al. [17]

also stick during this phase due to slow movement. The second half of stiction period is generally the exact reverse of the first part.

The nature of a stiction problem is based on the original cause. Some stiction scenarios are depicted in Figure 2.6. Each row represents a possible case starting from no stiction. The left column shows the variations of MV and OP with time while the right column shows the trends of these parameters against each other.

2.4.2 Model Uncertainty

Model uncertainty or Model Plant Mismatch (MPM) is a prevalent condition in the process industry. It refers to the case when the identified model does not match the real process. The uncertainty could happen in two forms: first when there exists a nonstructural mismatch, that is, the model structure is correct but the parameters are miscalculated and second when model structures such as model order do not match each other (i.e., structural mismatch). Two important variables can be used to quantify the mismatch, multiplicative and additive model errors. We denote the uncertainty by $\Delta(s)$ and the relations between the real process and model are then:

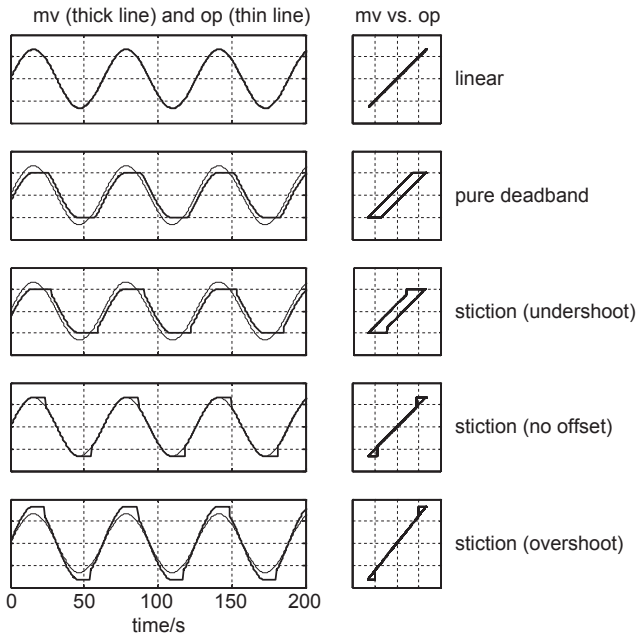


Figure 2.6: Various stiction scenarios, Courtesy of Choudhury et al. [17]

$$P(s) = P_n(s) + \Delta_a(s) \qquad \textit{additive} \tag{2.17a}$$
$$P(s) = 1 + \Delta_m(s)P_n(s) \qquad \textit{multiplicative} \tag{2.17b}$$

P(s) and $P_n(s)$ represent the real and nominal models. Therefore

$$\Delta_a(s) = P(s) - P_n(s) \qquad \textit{additive} \tag{2.18a}$$
$$\Delta_m(s) = \frac{P(s) - P_n(s)}{P_n(s)} \qquad \textit{multiplicative} \tag{2.18b}$$

It is usually hard to obtain an exact mathematical equation describing the mismatch; despite that, it is normal to define an upper bound or confidence interval for the modeling error which can easily be used for derivations.

$$l(\omega) \geq \max_{\omega}(|\Delta(j\omega)|) \tag{2.19}$$

2.5 Nominal Sensitivity Functions

Sensitivity functions are important variables in linear time invariant (LTI) systems. They are closely related to system's dynamic Equations (2.1a), (2.1b), and (2.1c)

which describe the feedback behavior. By definition [20]:

$$T(s) = \frac{C(s)P(s)}{1 + C(s)P(s)} = \frac{Y(s)}{Y_{sp}(s)} = G_{Y_{sp} \to Y} \tag{2.20a}$$

$$S(s) = \frac{1}{1 + C(s)P(s)} = \frac{Y(s)}{N(s)} = G_{D_o \to Y} \tag{2.20b}$$

$$S_D(s) = \frac{P(s)}{1 + C(s)P(s)} = \frac{Y(s)}{D_i(s)} = G_{D_i \to Y} \tag{2.20c}$$

$$S_U(s) = \frac{C(s)}{1 + C(s)P(s)} = \frac{Y(s)}{U(s)} = G_{U \to Y} \tag{2.20d}$$

Given these equations, sensitivity functions are called:

$$T(s) : \text{Nominal complementary sensitivity}$$

$$S(s) : \text{Nominal sensitivity}$$

$$S_D(s) : \text{Nominal input-disturbance sensitivity}$$

$$S_U(s) : \text{Nominal control sensitivity}$$

The nominal sensitivity function is actually the transfer function from the disturbance to the process variable and the complementary sensitivity shows the dynamics from the setpoint to the process variable. The philosophy behind these names is explained in the next section.

Sensitivity functions are important factors in representing the trade-off between various control objectives and impose important constraints on the system's robustness. Sensitivity functions in Equations (2.20) exhibit some interesting properties. While the sensitivity function(S) is a measure of how the system rejects disturbances, it is also a measure of the degree of robustness as it will be discussed in a later section. Note that the sum of the sensitivity and complementary sensitivity functions is always equal to unity:

$$S(s) + T(s) = 1 \tag{2.21}$$

2.5.1 Sensitivity and Model Uncertainty

Consider the closed-loop transfer function from a setpoint to the process variable, denoted by T(s)

$$T(s) = G_{Y_{sp} \to Y} = \frac{P(s)C(s)}{1 + P(s)C(s)} \tag{2.22}$$

The effect of P(s) on T(s) can be investigated by taking the derivative of the above equation:

$$\frac{\partial T(s)}{\partial P(s)} = \frac{C(s)[1+P(s)C(s)] - C(s)^2 P(s)}{[1+P(s)C(s)]^2} = \frac{C(s)}{[1+P(s)C(s)]^2}$$
$$\frac{\partial T(s)}{T(s)} = \frac{1}{1+P(s)C(s)} \frac{\partial P(s)}{P(s)} = S(s) \frac{\partial P(s)}{P(s)} \tag{2.23}$$

We see the sensitivity function here plays an important role in determining the robustness of the system to changes in process dynamics. Based on Equation (2.23), it is desirable to keep the sensitivity function as low as possible; however it is impossible to fully achieve this over the whole frequency region and hence a compromise has to be sought (see Bode's Integral Theorem 2.7).

For an open-loop system it is straightforward to show that S(s) is equal to unity. Using feedback will normally decrease the sensitivity of the system against process changes ($\frac{1}{1+P(s)C(s)} < 1$). This is actually one of main reasons to use feedback for control [5].

2.5.2 Sensitivity and Disturbance Rejection

Sensitivity is also closely related to the way the process rejects a disturbance. Equation 2.20b shows that for frequencies where $|S(j\omega)| < 1$ the system attenuates the disturbance and for frequencies where $|S(j\omega)| > 1$ the disturbance is amplified inside the loop.

2.5.3 Graphical Representation

Sensitivity functions have graphical representations on the Nyquist plot. Considering Figure 2.7, the magnitude of the sensitivity function is the inverse of length of the vector from the critical point [-1 0] to the loop curve. The unit sensitivity circle divides the curve into the two sections; disturbances with frequencies inside the circle are amplified while those in the outer section become attenuated. The sensitivity cross-over frequency (ω_{sc}) marks the threshold.

The maximum sensitivity (M_S) is also represented as the inverse of the distance from the nearest point to the critical point and exhibits the maximum amplification of feedback. A design constraint normally shows itself as a circle centered at the critical point; the loop curve should be designed in such a way to fall outside that area. For M_T, this constraint is again in the shape of a circle with the center at $[-{M_T}^2/({M_T}^2-1), 0]$ and radius of $M_T/({M_T}^2-1)$.

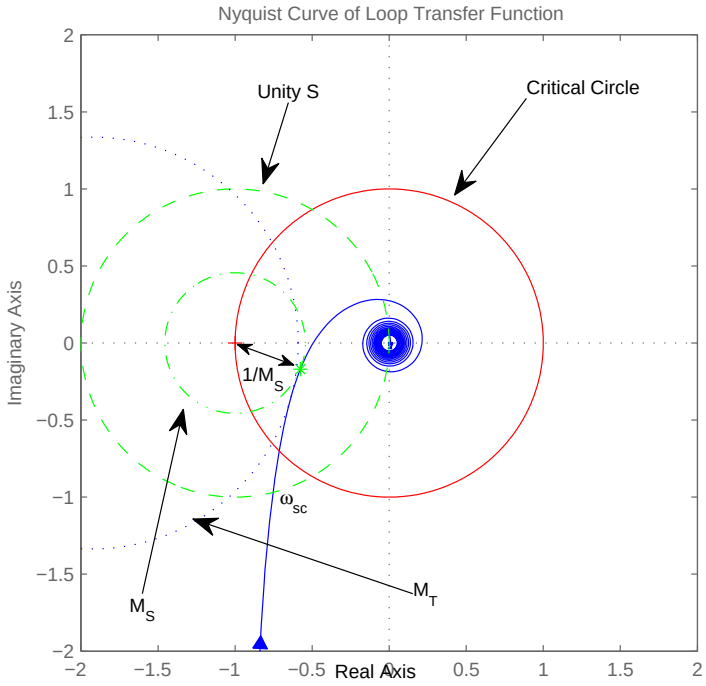


Figure 2.7: Graphical representation of sensitivity function and peripherals

2.6 Performance versus Robustness

The variety of proposed tuning rules in the literature makes the selection of the appropriate rule difficult. In order to evaluate and ensure the competitiveness of a tuning rule, it is essential to define some indices for a fair comparison. This section will go through some of those for both performance and robustness of feedback control.

2.6.1 Performance Metrics

Performance indices are divided into two subcategories: output performance measures how the system behaves in terms of process variable with respect to the setpoint, and controller performance measures the variations of the actuator. Both frequency and time domain measures are discussed.

Output Performance

a) *Time Domain*: The integrations of process variables over time is usually in the scope of interest. Different versions are used based on specific emphasis. The general formulation can be represented by:

$$ITAE(p,q) = \int_0^\infty t^q |e(t)|^p dt \tag{2.24}$$

Some of the famous ones have the structure of: IAE (p=1, q=0), ISE (p=2,q=0), and ISTE(p=1,q=1).

b) *Frequency Domain*: J_{SP} and J_D are normally used to measure the performance for setpoint-tracking and disturbance rejection as follows:

$$J_{SP} = ||\frac{1}{s}S(s)||_\infty = \max_\omega |\frac{S(j\omega)}{\omega}| \tag{2.25a}$$

$$J_D = ||\frac{1}{s}S_D(s)||_\infty = \max_\omega |\frac{S_D(j\omega)}{\omega}| \tag{2.25b}$$

Controller Performance

a) *Time Domain*: Total variation of control effort can be used as denoted by T_v:

$$T_v = \int_0^\infty |u(t)|dt \tag{2.26}$$

c) *Frequency Domain*: J_U was proposed as an index of control performance:

$$J_U = ||S_U(s)||_\infty = \max_\omega |C(j\omega)S(j\omega)| \tag{2.27}$$

2.6.2 Robustness Metrics

Robustness metrics are defined only for the frequency domain and measure the ability of a system to perform in nonideal conditions such as in the presence of model uncertainty. In this section, sensitivity peaks and stability margins are covered.

Sensitivity Peaks

The maximum of sensitivity function and its complementary function can be calculated via:

$$M_S = \max_\omega |S(j\omega)| = 1/\min_\omega |1 + P(j\omega)C(j\omega)| \tag{2.28a}$$

$$M_T = \max_\omega |T(j\omega)| = \max_\omega |\frac{P(j\omega)C(j\omega)}{1 + P(j\omega)C(j\omega)}| \tag{2.28b}$$

M_S actually shows the maximum amplification of disturbance by the system. On the other hand, M_T represents the maximum gain of the feedback loop. Figure 2.8 depicts M_S and M_T for a conventional system.

Stability (Gain, Phase and Delay) Margins

Two important quantities in linear control theory are gain and phase margins. They form two frequency domain metrics for the degree of stability and robustness of a system.

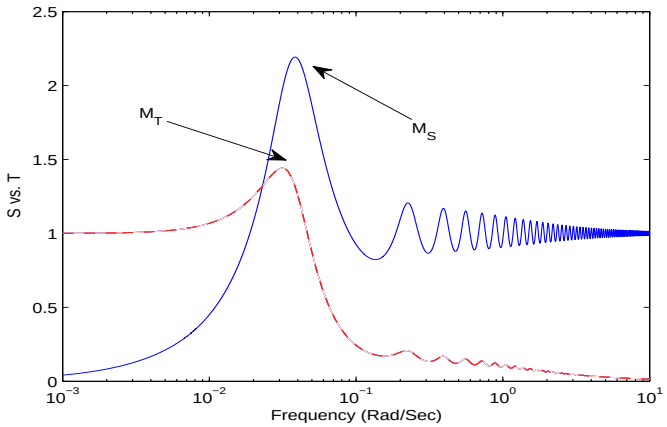


Figure 2.8: Sensitivity and its complementary functions, peaks and trends [$K_p = 1.8, \tau = 60, \theta = 38, K_c = 0.8, \tau_i = 60, \tau_D = 5, \tau_f = 1$]

The Gain margin (GM) is defined as the inverse of modulus of frequency response at phase crossover frequency (ω_p). The Phase margin (PM) is defined as the difference of frequency response angle at gain crossover frequency (ω_g) and the critical phase of $-\pi$. In mathematical notation:

$$GM = \frac{1}{|G(j\omega_p)|}^2 \tag{2.29a}$$

$$PM = 180^\circ + \varphi\{G(j\omega_g)\} \tag{2.29b}$$

The stability theorem states that a system is critically stable if and only if $GM \leq 1$.

Table 2.1 outlines nominal values for phase and gain margins in addition to sensitivity maxima. The data can be used as rules of thumb for controller design.

Table 2.1: Nominal values for GM, PM, M_S, and M_T

Nominal Values	Gain Margin	Phase Margin	Maximum S	Maximum T
Minimum	2	45	1.2	1
Maximum	5	60	2	1.5

Figure 2.9 shows the concept of phase and gain margins on a Nyquist plot.

One other related property is the delay margin which is defined as the maximum amount of extra delay that can be tolerated by a system without jeopardizing its stability. This parameter has a close relation to phase margin i.e.,

[2]Gain margin is sometimes defined in decibel units; in that case: $GM(dB) = -20log(|G(j\omega_p)|)$

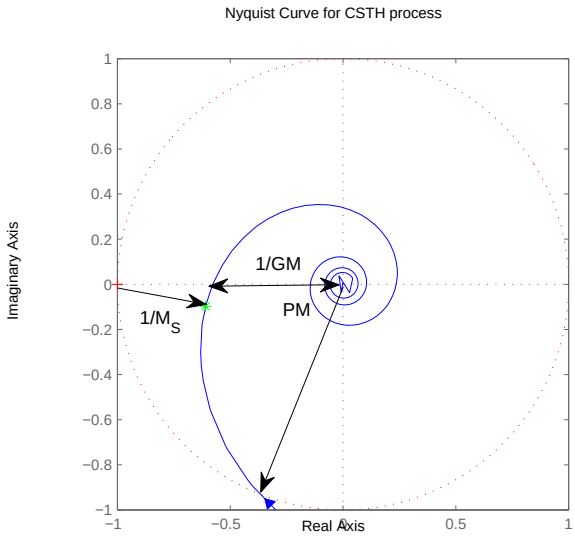


Figure 2.9: Nyquist representation of phase and gain margins

$$DM = \frac{PM}{\omega_g} \times \frac{\pi}{180} \tag{2.30}$$

Delay margin gives a more insightful physical sense of the system's stability than the phase margin.

2.6.3 Bounds of Sensitivity on Phase and Gain Margins

Interestingly, gain and phase margins are related to the maximum sensitivity. Indeed, sensitivity maxima define a lower limit on the amount of stability margins. That is:

$$GM \geq \frac{M_S}{M_S - 1} \qquad and \qquad GM \geq 1 + \frac{1}{M_T} \tag{2.31}$$

$$PM \geq 2 \cdot arcsin(\frac{1}{2M_S}) \qquad and \qquad PM \geq 2 \cdot arcsin(\frac{1}{2M_T}) \tag{2.32}$$

2.7 Bode's Integral Theorem

As noted before, it is favorable to keep sensitivity function as low as possible. However, there exists a mathematical relation on the value of sensitivity for delayed systems governed by:

$$\int_0^\infty ln(|S(j\omega)|)d\omega = 0 \tag{2.33}$$

Intuitively, the meaning of this equation is that the sensitivity cannot be kept low over the entire frequency region. If in some region its value is low; then it may have to be unintentionally high in another region. This phenomenon is sometimes known as the waterbed effect [36], that is when you push push down a particular point, another region pops out.

2.8 Robustness Stability Theorem

Assume that $G_0(s)$ is stable and $l(\omega)$ is governed by Equation (2.19). Figure 2.10 shows the basic idea behind the theorem. To ensure the stability of a system in the presence of uncertainty it is sufficient to hold the following relation:

$$|1 + C(s)P(s)| < |C(s)P(s)\Delta P(s)| \tag{2.34}$$

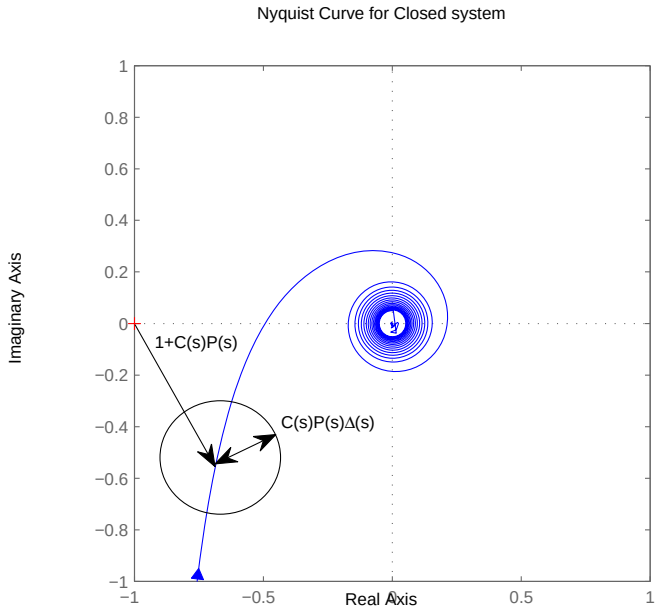


Figure 2.10: Robustness stability theorem

By substituting an upper bound for model uncertainty from Equation 2.19 and finally rearranging the variables we can define:

The robustness stability theorem states that the closed loop dynamics of a system is robustly stable if and only if:

$$[T_0(j\omega)] < \frac{1}{l(\omega)} \qquad \text{for all } \omega \tag{2.35}$$

Note that this theorem expresses a sufficient condition for stability and hence the designs based on the above equation are conservative.

Chapter 3

Review of Robust ISA Tuning Method

The very first formative idea of this research effort was to find a robust PID controller tuning which can fairly overcome the uncertainties present inside the process. Definitions and formulations for types of MPM are given in 2.4.2. The focus is on FOPTD systems controlled by an ISA-standard-based noninteractive PID controller.

In 2007, Vilanova proposed a robust PID controller tuning technique [43] which was set as a benchmark for further investigations in this thesis. To become familiar with this tuning rule, a short description of the method follows.

3.1 Summary of an IMC Based Robust ISA PID Scheme Proposed by Vilanova [43]

The proposed tuning rule is derived using frequency domain techniques based on IMC methodology. The process is assumed to be FOPTD and the PID controller has an ISA based structure.

The time delay is approximated by a first order Taylor series expansion. A model plant mismatch (MPM) margin is then introduced by defining a multiplicative modeling error (Equation 2.18b) and using a weight function which encompasses the acceptable family of plants around a nominal process. By using an analogy between IMC formulation and conventional representation, the necessary condition to stabilize a system is obtained. This H_∞ [1] inequality relation is furthermore transformed to

[1] Two continuous norms are mostly used in the literature: Euclidean and infinity norms By definition the Euclidean norm of a system is the root mean square of its impulse response:

$$||G||_2 = \sqrt{\frac{1}{2\pi}\int_{\infty}^{\infty} trace\{G(j\omega)^T G(j\omega)\}d\omega} \tag{3.1}$$

a min-max optimization problem and by using lemma 3.1 in the original paper, the problem is solved to get a set of robust tuning rules.

Note that two user-selected parameters: z (robustness factor) and τ_M (closed-loop time constants) are introduced into the derivation to allow the trade-off between robustness and performance. z and τ_M parameters should be chosen in such a way to satisfy H_∞ constraint.

In the next section of paper a simulation example is presented to show how the method is applied. By assuming a fixed value for τ_M the original author tries to calculate the optimal value for z by using the H_∞ inequality which is shown graphically in Figure 2 of Vilanova's paper. Tracking responses for the nominal and the family of real plants are also depicted to prove the suitability of the proposed method.

Next, an automatic tuning rule is obtained from the general rule mentioned before. By assuming critical values for phase margin and maximum sensitivity ($PM > 45°$ and $M_S < 2$), performance and robustness parameters are furthermore eliminated to lead to automatic tuning equations.

The proposed tuning rule is then evaluated by computer simulation. Three different process models have been assumed to analyze the rule for different working conditions: lag dominant, delay dominant and balanced lag and delay. To test the competitiveness of proposed tuning rule, two other tuning rules: SIMC (Skogestad et al.) [38] and AMIGO (Astrom et al.) [5] have been chosen for comparison.

Simulation results are presented for setpoint tracking and regulation along with corresponding control actions. Sensitivity (S) and complementary sensitivity (T) functions are also illustrated in a separate graph. Other numerical criterions are also computed for the sake of completeness. For robustness, maximum sensitivity (M_S) and maximum complementary sensitivity (M_T) are used. The performance is divided into two subcategories: output performance in which the integrated absolute error (IAE) (Equation 2.24) is used and input performance is measured using the total variation (T_v) (Equation 2.26) of control action. The latter is sampled at the rate of

---

This norm gives out the steady state covariance of output to a unit white noise input:

$$||G||_2 = \lim_{t\to\infty} E\{y(t)^T y(t)\} \tag{3.2}$$

$$E\{w(t)w(t)^T\} = \delta(t-\tau)I \tag{3.3}$$

By contrast, the infinity norm represents the peak gain of a system's frequency response and for SISO systems it is defined as:

$$||G||_\infty = \sup_\omega |G(j\omega)| \tag{3.4}$$

The Euclidean norms become infinity when the system is unstable or has a non-zero high frequency gain. However this is the case for infinity norm when a system has a pole on the imaginary axis. [11]

one tenth of the process time constant.

The results were organized in a table to find the best rule for each criterion. In conclusion, it was claimed that the Vilanova technique behaved better than others in tracking control.

The article was an interesting starting point for further research and therefore the results (graphs and tables) were initially reproduced to validate the claims.

During simulations it was confirmed that the Vilanova technique is indeed a robust PID tuning rule. However, there exists a trade-off between robustness and performance and this makes the tuning rule somehow sluggish under ideal conditions compared to other techniques. The guidelines in "Control Objectives" (see section 4.2) can be helpful for making the final decision.

One other interesting result was found during the reproduction of materials. It was suggested that a negative derivative action favors robustness under some special circumstances. That was more prominently seen in delay dominant regions.

Some authors suggested that the abnormal behavior is an artifact of the tuning rule or may be due to the tight constraint on the optimization problem initially posed to derive the rule; that is the constraint on the H_∞ problem is so strict that it leads to a nonrealistic derivative solution. However the τ_d and N parameters used in deriving the tuning rule have the same signs, making the derivative filter stable for that specific example. Nevertheless more study was deemed necessary to clarify the claims.

To further investigate this phenomenon, the relationships between controller parameters and robustness factor (z) at a fixed deadtime are calculated and plotted in Figure 3.1.

As described before, one of the advantages of the Vilanova tuning rule is that the user has two degrees of freedom in the design phase: the desired closed loop time constant which is adjusted by τ_c and the z factor which manipulates desirable robustness characterization.

The Vilanova technique recommends using less derivative action in favour of more robustness (3rd graph in Figure 3.1). Essentially, beyond $z = 2$ using negative derivative time is claimed to be helpful. The most important and difficult factor in controlling of an FOPTD process is deadtime and this means the amount of necessary robustness should be determined based on the magnitude of the delay term. Figure 3.2 gives the minimum robustness factor for various time delays as recommended by the Vilanova tuning rule.

Based on this plot and previous discussions we conclude that the Vilanova technique suggests a negative derivative action for deadtime to time constant ratios beyond 0.8-0.9.

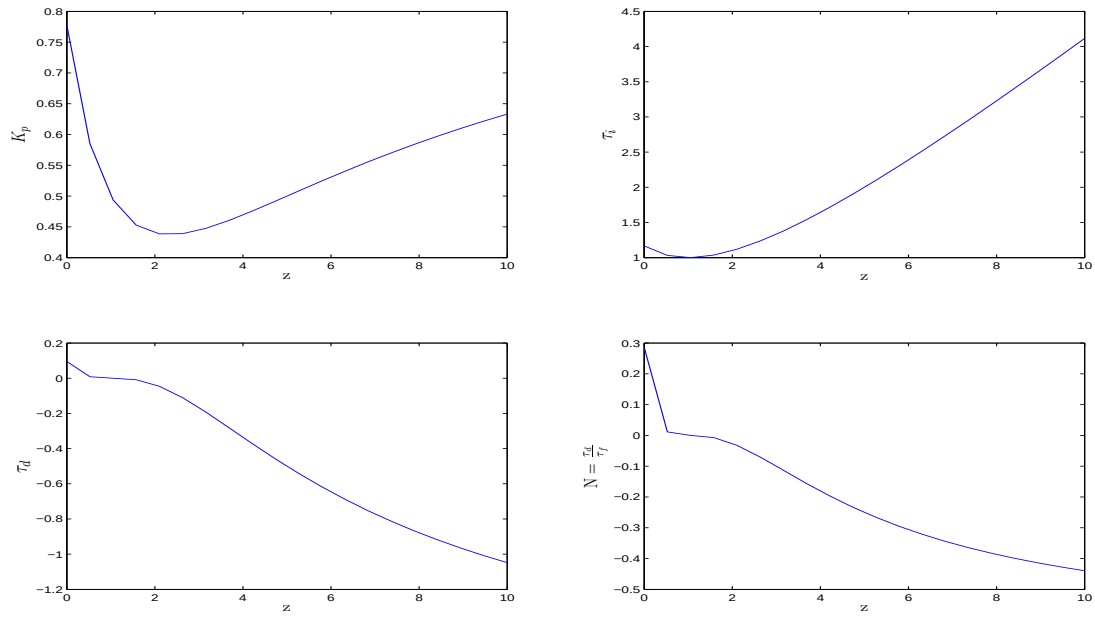


Figure 3.1: PID controller parameters as a function of robustness factor (z)

3.2 A PID Controller with Negative τ_d

The previous study and discussions triggered a new idea in the research of controller tuning. Could a negative derivative action be helpful in feedback systems at all? As far as we know negative derivative time is impossible to use in the current industrial PID controllers. What happens if we use such a structure in the control design and implementation?

To address this fundamental problem, the feasibility of the idea from stability point of view was investigated. The problem was divided into two parts: first to see whether closed loop poles are stable and second to ensure that the controller behaves in a minimum phase fashion when negative derivative action is used.

1. Closed loop poles

   Although the initial derivation by Vilanova was obtained based on a stability assumption for FOPTD systems, we prove that the poles are stable using an arbitrary PID controller that satisfies certain conditions.

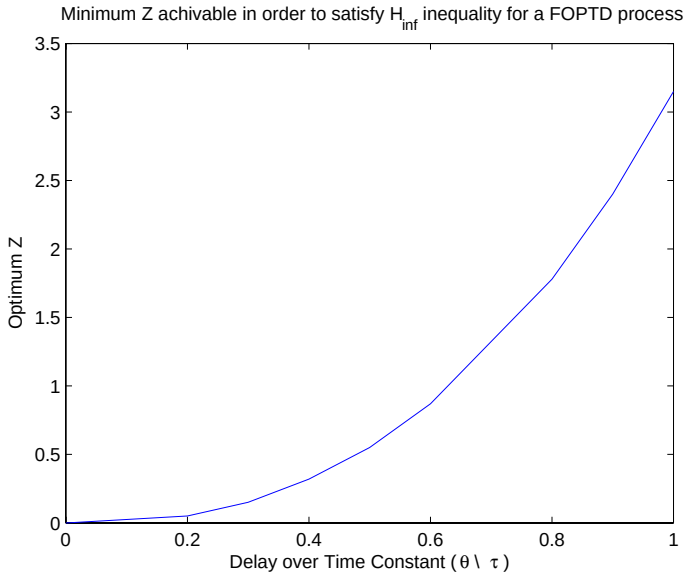


Figure 3.2: Optimal z recommended by the Vilanova tuning rule

3.2.1 Stability Proof of a Closed Loop System Controlled by a PID Compensator with Negative Derivative Action

Process and Controller Definition

$$P(s) = \frac{K_p}{\tau s + 1} \tag{3.5}$$

$$C(s) = Kc(1 + \frac{1}{\tau_i s} + \tau_d s) \tag{3.6}$$

The characteristic equation can be calculated as:

$$\Delta(s) = 1 + L(s) = 1 + K_p K_c \frac{\tau_d \tau_i s^2 + \tau_i s + 1}{\tau_i s + 1}$$

$$\text{Assuming } k = K_p K_c$$

$$\Delta(s) = 0 \rightarrow \tau_i(k\tau_d + \tau)s^2 + \tau_i(k+1)s + k = 0 \tag{3.7}$$

The necessary and sufficient condition for a second order polynomial to have stable roots (poles) is that all of its coefficients should be positive. Therefore the following condition should hold to ensure stability:

$$\tau_d > -\frac{\tau}{k} \tag{3.8}$$

Chemical processes normally have large time constants; therefore the derivative action can be significantly negative without jeopardizing the stability.

CSTH Process Example (no deadtime)

$$P(s) = \frac{1.82}{60s + 1} \tag{3.9}$$

PID controller parameters are set as: $K_c = 0.8$, $\tau_i = 79$, $\tau_d = -10$, and $\tau_f = 1$.

It is possible to prove stability using Nyquist and Bode plots:

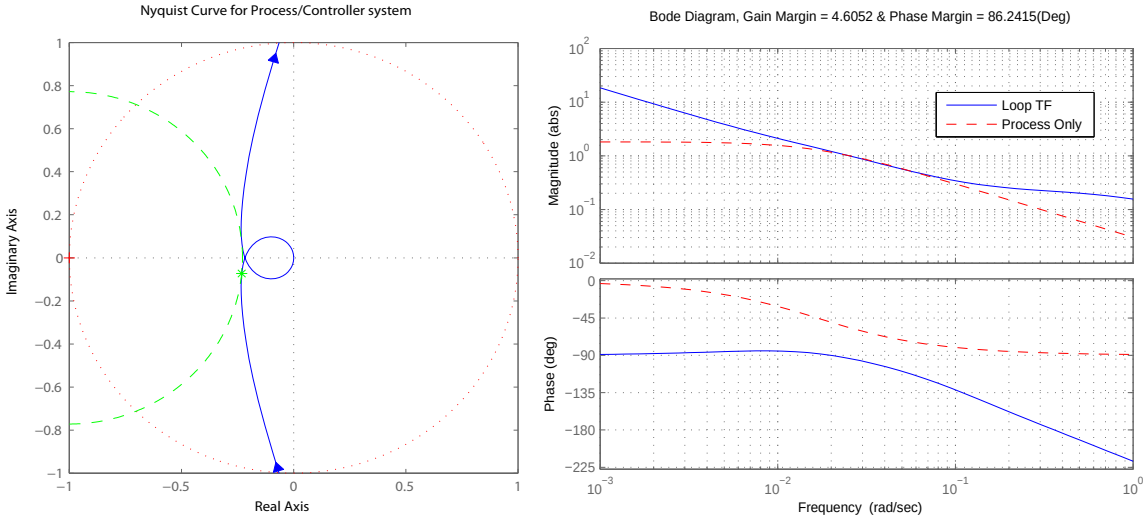


Figure 3.3: Nyquist and Bode plots of the case study

It is obvious that the system is stable and even the gain and phase margins are higher than expected. For this example the specific derivative action could be increased even to -40 without fear of instability.

CSTH Process Example (with deadtime)

The stability of the CSTH process was investigated using root locus diagrams for variable gain, time constant, and deadtime. The model is assumed to be the same as in the Equation 3.9 but with a deadtime of 38 seconds. A third order Pade approximation was used to calculate the closed loop poles of the system. The beauty of colors was utilized in these figures with hotter colors corresponding to the higher values of the parameter of interest. The closed loop poles of the nominal process are marked with pink circles.

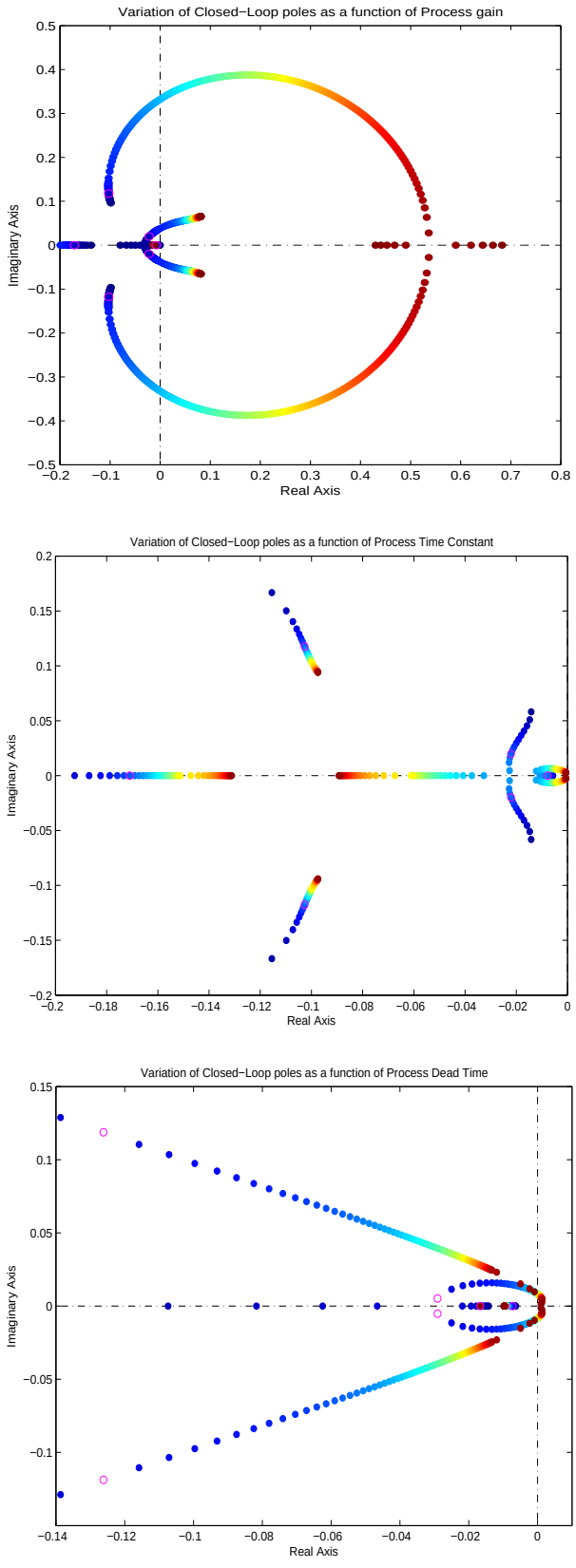


Figure 3.4: Root locus plots for CSTH system as a function of gain, time constant, and deadtime [$K_c = 0.32$, $\tau_i =$, $\tau_d = -5.75$, and $\tau_f = 10$]

The analysis on the root locus plots showed that a factor of 4.0 for the gain and 5.2 for the deadtime is needed to drive the system to the verge of instability. However; in the case of the time constant the system proved to be stable for the whole domain of the examination.

2. Closed loop zeros

   In this section we first prove that the closed loop poles calculated by the Vilanova tuning rule are stable, then we investigate the general case.

3.2.2 Minimum Phase Proof of a System Tuned by the Vilanova (ISA-based) Rules

The following expression shows an expanded form of the controller transfer function:

$$C(s) = \frac{\tau_i(\tau_f + \tau_d)s^2 + (\tau_i + \tau_f)s + 1}{(\tau_i s)(\tau_f s + 1)} \tag{3.10}$$

For ISA based tuning:

$$\tau_f = \frac{\tau_d}{N} \Rightarrow \tau_f + \tau_d = \frac{\tau_d}{N}(N + 1) \tag{3.11}$$

Based on the tuning rule:

$$\tau_f = \frac{\tau_d}{N} = T_M \frac{\rho + z}{\rho + T_M} \tag{3.12}$$

$$N + 1 = \frac{T}{T_i}\frac{\rho}{L}\frac{(\rho + T_M)}{(\rho + z)} \tag{3.13}$$

The latter two variables are always positive, hence the postulate is proven.

3.2.3 Proof for a General First Order System

Assuming no zero/pole cancellations occur using an arbitrary PID controller structure, the open and closed loop zeros coincide with each other. Therefore by looking at PID controller transfer function, zeros turn out to be roots of the following equation:

$$f(s) = s^2\tau_i(\tau_f + \tau_d) + s(\tau_f + \tau_i) + 1 \tag{3.14}$$

To confirm the stability, the following condition should be satisfied:

$$\tau_d > -\tau_f \Rightarrow |\tau_d| < |\tau_f| \tag{3.15}$$

It seems that the derivative action should be less than its filter which is apparently unusual. At the end of the chapter we will discuss this issue to draw a final conclusion.

3.3 Justification

Two case studies are presented in this section to rationalize the use of negative derivative action. They are only simple justifications and should not be considered proofs.

- Delayed response example

  Consider the step response of the underdamped second order system shown in Figure 3.5. It has been assumed that this response belongs to a feedback delayed process controlled by a PID controller. The full line represents the real process variations (PV) while the dashed line shows the dynamics that the controller is actually seeing. This situation may arise due to the transportation delay (L) present inside the process.

  Suppose the process goes through point (A) in Figure 3.5. Circle (A) represents the point which the controller is observing. Note that the actual PV is decreasing whereas the apparent PV (point A) is still growing. The same misleading situation happens at point (B) where the PVs have reverse trends.

  It is well known that the derivative action adds an anticipatory capability to the feedback control. Nevertheless, in this particular circumstance the derivative action is unexpectedly doing the wrong duty due to the inconsistent dynamic trend. Application of a negative derivative action in such cases which might be surprising in the first place, could prove to be advantageous for delayed processes.

- Inverted pendulum case study

  The second example refers to the familiar inverted pendulum control case study in Figure 3.6. The problem is defined to move a cart with an inverted pendulum

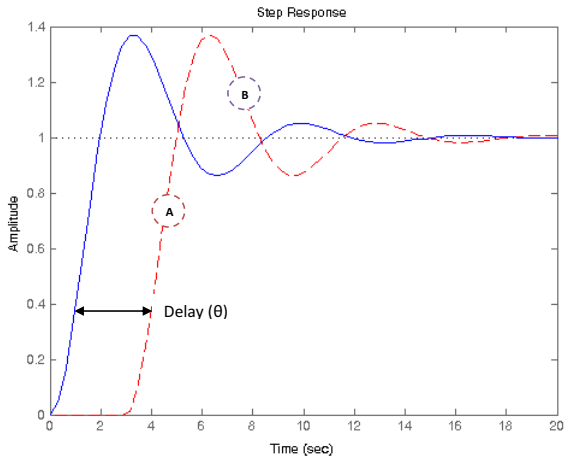


Figure 3.5: Two viewpoints of an underdamped system response

mounted on top from the initial position (x=0) to the final position (x=L) without unbalancing the pendulum. Two approaches can be considered:

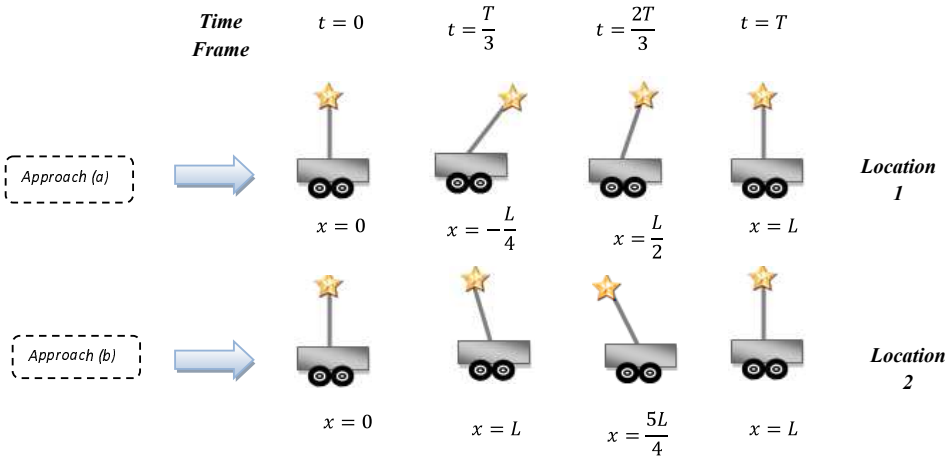


Figure 3.6: Inverted pendulum case study

The more straight-forward way of doing the job, approach (b), is to move the cart slightly after the destination point and then returning it to the final position. This so called overshoot reaction is done to preserve the balance of the pendulum. The second tricky way, (approach (a)), is to initially drive the cart against the destination direction and then move it to the final position. In this case an undershoot response is imposed by the controller (driver). The latter approach reinforces the notion of negative derivative action, as it employs an

initial response to counteract the future events.

We have shown the feasibility of the use of negative derivative action; however, whether it is useful or not is still an open question. Why bother if no extra advantage is gained? A look in the literature shows that the issue has been discussed earlier especially with respect to IMC derivations. During IMC controller design, it is probable to end up with a non-PID controller structure, and in special cases we may even face a negative derivative format. Following are two recommendations from other researchers to alleviate the problem:

1. Substitute the derivative part with a first order filter [10].
2. Set $\tau_d = 0$ if you have encountered negative derivative action [40].

The above mentioned solutions seem to be shortcuts to simply ignore the issue. However, we are interested to have an analysis and study possible advantages and drawbacks of using negative derivative action which is discussed in the next section.

3.4 Possible Advantages of Using Negative Derivative Action

3.4.1 Making closed loop dynamics closer to what was originally designed

Application of negative derivative time will enables us to have a more accurate controller structure and it also expands the available selection set for such a controller.

As noted before, IMC formulation does not always end up in a realizable or sensible solution (for instance, the outcome might be a controller with unstable poles). Using a direct structure eliminates the need for a nonexact filter or complete elimination of the derivative part.

3.4.2 Making the Process More Robust

In section 3.1, it was pointed out that the Vilanova tuning rule (which claims to be a robust method) calculates a negative τ_d for highly delayed systems. To investigate the possible advantage of this, we tried to derive the mathematical equations of the gain/phase margins and sensitivity functions. However, due to the high degree of complexity, it was not feasible to solve the equations and obtain an analytical expression. Instead, a particular process model was analyzed for further investigations.

Consider a process with the transfer function $P(s) = \frac{e^{-0.4s}}{0.2s+1}$

The controller parameters of this system are calculated using the Vilanova tuning rule. By keeping all of the controller parameters constant and incrementally decreasing the derivative time from a positive value to a negative one, the reaction of the system is observed on the Nyquist plot. The results are plotted in Figure 3.7:

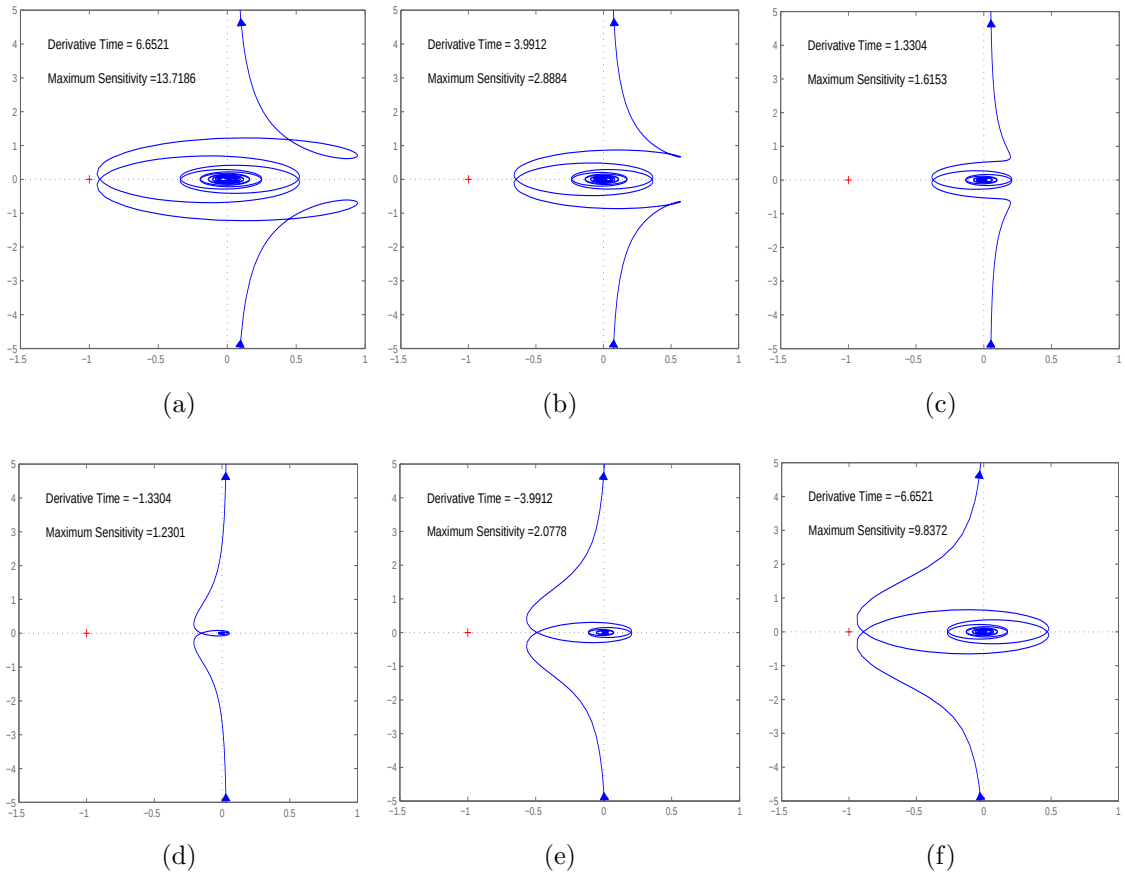


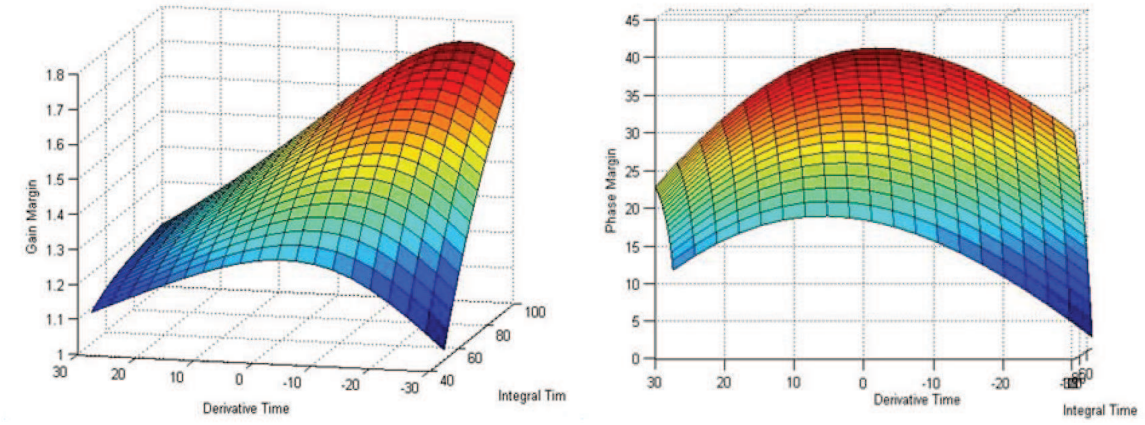
Figure 3.7: Behavior of the Nyquist curve as a function of derivative time (τ_d decreases from a positive value to a negative value, from the top left plot to the bottom right plot) - $z = 3.18$, $T_M = 1$, $K_c = 0.34$, $\tau_i = 0.68$, $\tau_f = 2.08$.

A more interesting and general 3D graph for the gain and phase margins is also presented in Figure 3.8 to help in understanding the concept:

3.4.3 Reduction in High Frequency Measurement Noise

Equation 2.5 calculates the high frequency gain of an ISA-based PID controller:

$$|C(\infty)| = K_c \cdot (1 + \frac{\tau_d}{\tau_f})$$



(a) (b)

Figure 3.8: Phase and gain margins as a function of PID controller parameters

It is clear from the above equation that a negative derivative time decreases the amount of high frequency gain. This means that the controller could actually decrease the amount of sensor noise circulating inside the feedback loop.

3.5 Disadvantages of Using Negative Derivative Action

A disadvantage of using negative derivative action is the lack of support by major PID controller venders. This could be just because they do not use ISA-based PID format (which is assumed as a default here) in their DCS. Another drawback of this scheme is the little benefit gained in some circumstances, especially when the ratio of deadtime to time delay is not high. We also observed that in order to make the obtained yield significant, the ratio of $\frac{\tau_d}{\tau_f}$ should be increased which causes the system to enter the nonminimum phase region. The control designers normally avoid non-minimum phase situations as they are notorious of being troublesome. These considerations show that analysis is required before implementing negative derivative action in each specific case. We do not recommend using negative derivative action due to these facts.

3.6 Summary

In this chapter, the tuning rule proposed by Vilanova was described and it was shown how a negative derivative could possibly improve the system. The proofs for sta-

bility of poles and zeros were also presented. The advantages and disadvantages of using negative derivative action were discussed and finally the procedure is not recommended. However, negative derivative action would be a good topic for future research in the context of nonminimum phase systems.

Chapter 4

Robust PID Controller Tuning of FOPTD Processes

4.1 Abstract

The purpose of this chapter is to recommend an optimal (or at least near optimal) PID tuning rule for a first order plus time delay (FOPTD) process. At this time, no single comprehensive solution has yet been found for this problem. That means no generalized rule would give you the desired response under different circumstances. However, it is possible to design an optimal tuning rule for a single process if the model is well known. To cope with this difficulty we propose a compromise solution. We divide the process models into different regions and compare some of the well-known techniques on each of them using computer aided simulation. In this way a guideline for each domain can be devised. Different performance indices are used to obtain a fair quantitative comparison and qualitative graphs are provided to supplement the analysis.

4.2 Control Objectives

Before starting an analysis, the objectives of a system controller should be reviewed. Any PID controller design should clarify its position with respect to these objectives:

- Tracking of operator's setpoints
- Rejection of process disturbances
- Robustness to model uncertainties
- Attenuation of sensor noise

- Stability and safety considerations

In this chapter the first three items on the above list are considered for the design and in case of robustness, only robustness against unstructural model uncertainty is discussed. However, in the attached paper in Appendix A, problems of valve stiction and sensor noise are included in the design as well.

4.3 Choice of Desired Closed Loop Time Constant

In some more recent tuning rules (such as the IMC formulation), a provision is set for the operator to specify the closed loop time constant. The choice of this user-defined quantity has a critical role in the final dynamic response.

As a general guideline, the closed loop time constant should be less than its open-loop counterpart to ensure the minimum speed of response($\frac{\tau_c}{\tau} < 1$). There are a variety of suggestions in the literature to make the decision easier. Several examples follow [36]:

$$\tau_c < \theta < \tau \qquad \text{(Chien and Fruehauf, 1990)} \qquad (4.1)$$

$$\tau_c = \theta \qquad \text{(Skogestad, 2003)} \qquad (4.2)$$

$$\frac{\tau_c}{\theta} < 0.8 \text{ and } \tau_c > 0.1\tau \qquad \text{(Rivera et al., 1986)} \qquad (4.3)$$

It has also been reported in [38] that for a robust performance $\tau_c \geq \theta$ should be maintained. Changing τ_c would compromise between two competing classes of parameters [38]:

1. Fast speed of tracking response and disturbance rejection corresponding to closed loop settling time

2. Stability margins and robustness of system in addition to the required control effort

Keep in mind that increasing τ_c gives you less speed but more stability. In this sense, previous guidelines could be fine tuned according to the application of interest. Here the operator's experience comes into play for a better process performance.

The second rule of thumb is used throughout this thesis wherever no other guideline is specified.

4.4 Defining FOPTD Process Domains

FOPTD models, also known as three parameter models, are commonly used for simplified analysis in process control. Process gain, time constant, and deadtime are the varying parameters in these models. According to the literature of controller design [5] process gain has the simplest effect on the calculation of controller parameters in the tuning. By deriving the closed loop transfer function of a system it is clear that the product of $Kp \times K_c$ should remain constant and therefore the controller gain is a function of inverse K_p $[K_c = f(1/K_p)]$. Intuitively we can omit this parameter from our analysis with no harm to the generality of the problem.

Both θ and τ have the dimension of time (sec). It has been observed that the ratio of these parameters has an important effect on the final response. We define alpha as the ratio of:

$$\alpha = \frac{\theta}{\tau} \tag{4.4}$$

Here we start the discussion based on the different values which alpha can take. In our view, the process model could fall in one of five categories:

$$\text{Process is } \begin{cases} \text{Pseudo-Pure Lag} & if \quad \alpha < 0.2 \\ \text{Lag Dominant} & if \quad 0.2 < \alpha < 0.5 \\ \text{Balanced Lag and Delay} & if \quad 0.5 < \alpha < 1 \\ \text{Delay Dominant} & if \quad 1 < \alpha < 3 \\ \text{Pseudo-Pure Delay} & if \quad \alpha > 3 \end{cases} \tag{4.5}$$

The boundaries set for this classifications are not sharp and may need slight adjustments depending on the specific situation.

Figure 4.1 shows the domains graphically:

4.5 Process Simulation

4.5.1 Defining the Test Batch

In order to implement a fair test, we have defined a standard benchmark to use in the different modeling regions mentioned before:

$$G(s) = \frac{e^{-\alpha}s}{s+1} \tag{4.6}$$

Several different values of α have been selected to perform the tests and and evaluation is exercised in each modeling region. The data are presented in Table 5.1.

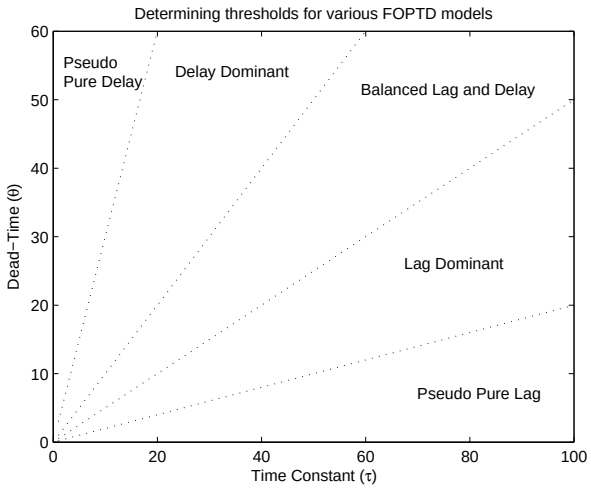


Figure 4.1: Various FOPTD modelling zones

Table 4.1: Chosen process models for each FOPTD region

	P-Pure Lag	Lag Dom.	Balanced Lag&Delay	Delay Dom.	P-Pure Delay
α	0.1	0.4	0.7	1.5	3

To make validation easier, we matched the system for each scenario to fit our experimental model.

4.6 Methodology

Various analyses have shown [1] that to achieve good performance and robustness characteristics for a feedback system, it is essential to differentiate between regulatory and tracking mechanisms. That means you need to design two sets of tuning parameters for each fixed system. One solution to this challenge is to use a controller with two degrees of freedom which is not in the scope of this study.

Accordingly, our approach here is to present two optimal techniques for both step-tracking and disturbance rejection. The overall theme of controller design is based on a robust technique which exhibits satisfactory control performance. An important point is that our effort attempts to provide a simple and handy method useful to the process industry. That means the method will not be sophisticated but provide acceptable results for process practitioners. Trial and error is used to fine tune suitable controller parameters and evaluation is carried out by both qualitative and quantitative techniques. The main emphasis is given to the IMC technique due to its superior robustness characteristics.

4.6.1 Setpoint Tracking

Based on the results in [28], the IMC technique shows good performance and robustness for step-tracking and it is quite well tested in industry. The main challenge for this method is how an operator should determine the desired closed loop time constant. During this analysis we test different choices for τ_c and for each FOPTD modeling region the best value is proposed by trial and error. Here the design is conducted so that the response would give a 15% overshoot in the case of 10% MPM in the worst case direction. We found this to be a suitable approach to obtain a moderately robust tuning rule.

4.6.2 Disturbance Rejection

Industry professionals complain that the IMC tuning technique does not perform well for rejecting process disturbances. Our tests confirmed the same observation. To present a good tuning rule we searched for other conventional techniques and found that the Ziegler-Nichols tuning rule is able to reject disturbances effectively. However, the Ziegler-Nichols rule exhibits poor robustness. This led us to compare the parameters calculated by Ziegler-Nichols and IMC tuning rules and finally resulted in our proposed method for disturbance rejection. For the proposed tuning rule, τ_c has been fine tuned to obtain little or no undershoot for the second response peak, again in the presence of 10% MPM. However, that condition is not exhaustive and therefore the final guidelines found at the end of this chapter for tracking are used as well for regulation. The procedure has been carried out adaptively.

4.7 Cross Comparison of IMC and Ziegler-Nichols Tuning Techniques

In this section the differences between Ziegler-Nichols and IMC tuning rules are compared. We first consider the FOPTD process in Equation 2.2. The PID controller parameters are calculated for different ascending ratios of time delay to time constant (α) from 0.1 to 3. The desired closed-loop time constant is set to be equal or higher than the time delay for IMC. The calculated ratios for controller gain, integral time, and derivative time are plotted in Figures 4.2, 4.3, and 4.4 respectively.

- Controller Gain

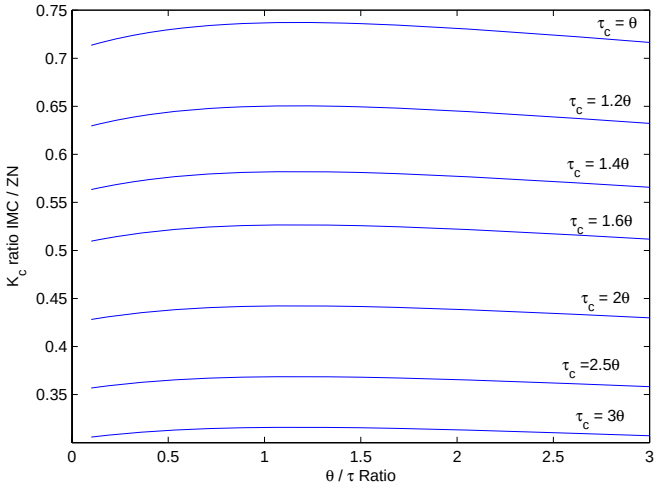


Figure 4.2: Controller gain ratios calculated by (IMC/ZN) as a function of τ_c

- Integral Time

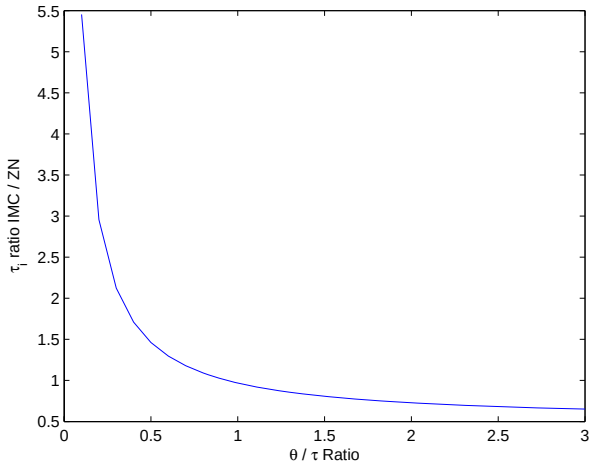


Figure 4.3: Integral time ratios calculated by (IMC/ZN)

- Derivative Time

Figure 4.2 shows that Ziegler Nichols is a more aggressive tuning method than IMC rule. This could explain the low robustness of this technique and the oscillatory responses observed in practice. The integral graph shows a great discrepancy between IMC and Ziegler-Nichols. The calculated τ_i for ZN has a much lower value in pseudo pure lag and lag dominant regions. Interestingly, simulation tests showed that the

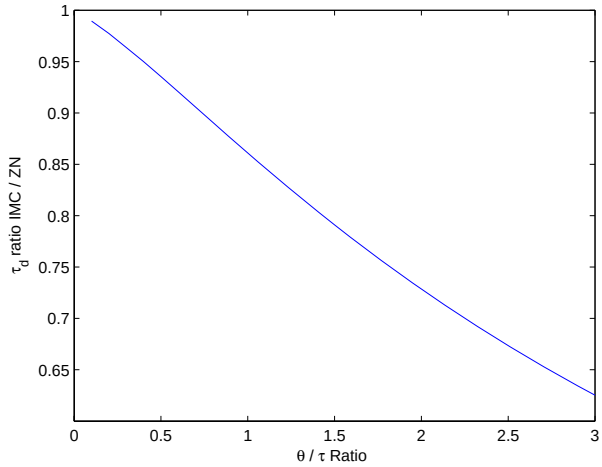


Figure 4.4: Derivative time ratios calculated by (IMC/ZN)

IMC controller does not perform well in those areas. Figure 4.5 showcases an example to clarify this idea:

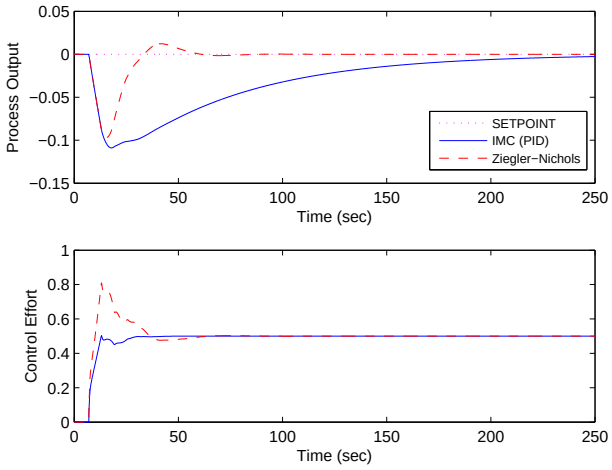


Figure 4.5: Dynamic responses of IMC and ZN in the lag dominant zone($\alpha = 0.1$)

To find the root cause, we look at the integral of error for a system excited by a unit disturbance [23]:

$$\begin{aligned} \text{Integral of Error (IE)} &= \int_0^\infty e(t)dt = -\int_0^\infty y(t)dt \\ &= -\lim_{s\to 0}(s \cdot \frac{1}{s}Y(s)) = \frac{-1}{k_i} \end{aligned} \tag{4.7}$$

The Integral of error is inversely proportional to the integral action. That means lower integral time would improve the rejection of entering disturbances (assuming nonoscillatory dynamics). From the graph of integral time ratio (Figure 4.3) we now understand the reason behind the poor behavior of IMC technique. Lower than sufficient integral time has caused a degradation of the closed loop dynamics. Surprisingly, if you look at the relation of the integral time for IMC rule in case of FOPTD systems:

$$\tau_I = \tau + \frac{\theta}{2} \tag{4.8}$$

It is clear that you always get a low integral action for lag dominant processes and a change is needed to alleviate the problem.

4.8 The Proposed Tuning Technique for Regulatory Control (ZNIMC Tuning Rule)

In the previous section we saw the defects of IMC and Ziegler Nichols techniques and investigated the reasons behind their misbehaviors. In our design methodology we would like to have a tuning rule that performs well and has good robustness at the same time. In other words, a tuning rule that performs as well as the Ziegler-Nichols rule and is as robust as the IMC technique. As discussed before that aim is impossible to achieve so an optimal intermediate solution is sought in this analysis.

Our proposed solution keeps the core IMC structure to ensure the desired robustness and substitutes the integral time formulation with the Ziegler-Nichols tuning rule. In this way a compromise solution is obtained. The obtained PID controller tuning technique is hereafter called "ZNIMC" tuning rule. Equation (4.9) shows the ZNIMC rule mathematically. The ZNIMC tuning technique can also be viewed as a detuned version of the Ziegler-Nichols to increase the robustness.

$$K_c K_p = \frac{\tau + \theta/2}{\tau_c + \theta/2} \qquad \tau_i = \frac{P_u}{2} \qquad \tau_d = \frac{\theta\tau}{2\tau + \theta} \tag{4.9}$$

where P_u is the ultimate period in the critically stable condition.

An important advantages of the ZNIMC rule is that it has the conventional user defined desired closed loop time constant in the same way as IMC does. This makes it greatly superior to the Ziegler Nichols tuning rule. The proposed tuning algorithm was presented to industrial practitioners as a simple solution for lag-dominant processes [2]. In the next section we evaluate the ZNIMC tuning rule and compare it with other conventional rules; and in the same way as setpoint tracking we propose a guideline for robust PID tuning design.

4.9 Results

1. Pseudo-Pure Lag Case ($\alpha = 0.1$)

$$P(s) = \frac{1.82 \cdot e^{-6s}}{60 \cdot s + 1}$$

- Setpoint Tracking ($\tau_c = 7$)

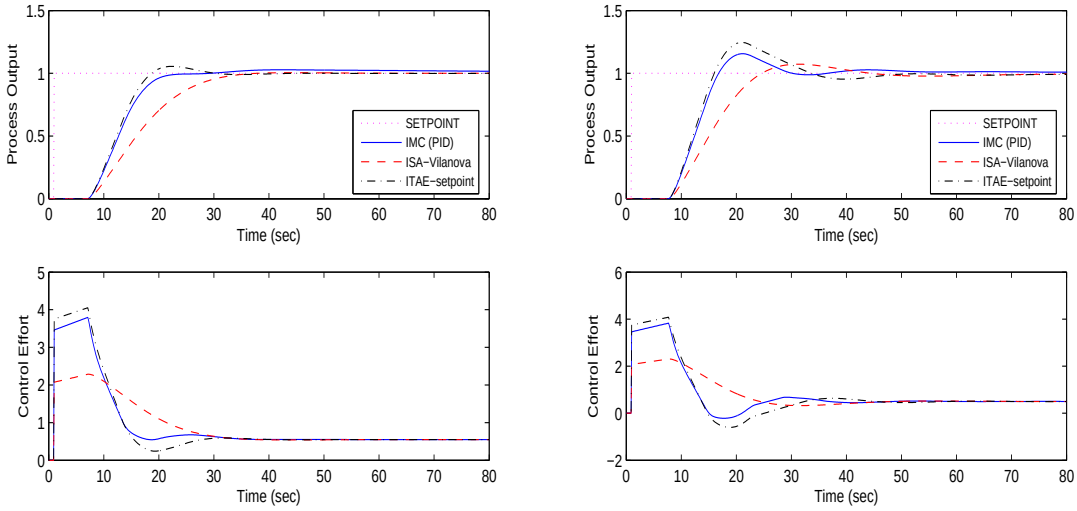


Figure 4.6: Responses for $\alpha = 0.1$, Nominal (left) and with 10% MPM (right)

Tuning	IMC	Vilanova	ITAE
K_c	3.46	2.07	3.75
τ_i	63	60.19	76.79
ISE	11.08	13.43	10.82
$ITSE$	64.86	99.75	61.26
T_v	57.7	54.48	57.05
$Max(y)$	1.03	1.01	1.06
GM	2.42	4.12	2.5
PM	71.13	68.79	66.6
DM_n	1.95	3.17	1.69
M_S	1.72	1.41	1.71
M_T	1	1	1
J_{SP}	10	15.94	11.24
J_D	18.2	29.01	20.46
J_U	40.36	2.97	42.93

Tuning	IMC	Vilanova	ITAE
τ_d	2.86	0.16	2.18
τ_f	0.29	10.36	0.22
ISE	11.17	12.93	11.27
$ITSE$	65.99	90.03	70.56
T_v	50.1	47.33	49.99
$Max(y)$	1.16	1.07	1.25
GM	1.89	3.09	1.91
PM	61.97	62.77	55.7
DM_n	1.35	3.09	1.91
M_S	2.15	1.6	2.17
M_T	1	1	1
J_{SP}	9.09	14.49	10.22
J_D	18.2	29.01	20.46
J_U	41.65	3.37	43.8

- Disturbance Rejection ($\tau_c = 7.2$)

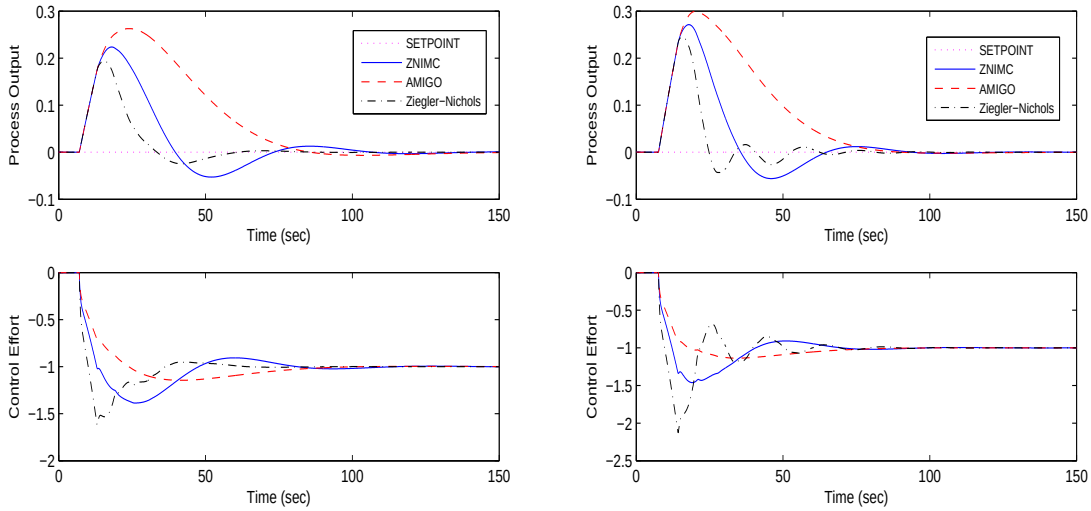


Figure 4.7: Responses for $\alpha = 0.1$, Nominal (left) and with 10% MPM (right)

Tuning	**ZNIMC**	**AMIGO**	**ZN**
K_c	3.39	2.58	5.39
τ_i	11.55	25.2	11.55
ISE	0.77	1.94	0.35
$ITSE$	17.26	57.72	5.9
T_v	147.1	143.7	147.8
$Min(y)$	-0.05	-0.01	-0.02
$Max(e)$	0.22	0.26	0.19
GM	2.63	3.28	1.65
PM	35.8	59.6	36.22
DM_n	0.93	2.17	0.63
M_S	1.68	1.45	2.58
M_T	1.75	1.18	1.67
J_{SP}	14.7	13.51	9.91
J_D	4.66	9.76	2.49
J_U	39.5	29.35	69.45

Tuning	**ZNIMC**	**AMIGO**	**ZN**
τ_d	2.86	2.91	2.89
τ_f	0.29	0.29	0.29
ISE	0.89	2.05	0.47
$ITSE$	18.31	56	7.89
T_v	147.3	144.1	147.9
$Min(y)$	-0.06	0	-0.04
$Max(e)$	0.27	0.30	0.24
GM	2.06	2.57	1.29
PM	32.7	57.08	26.13
DM_n	0.72	1.74	0.36
M_S	2.00	1.65	4.58
M_T	1.84	1.17	3.62
J_{SP}	13.7	11.6	14.63
J_D	4.53	9.76	2.37
J_U	40.66	29.85	76.9

2. Lag Dominant Case ($\alpha = 0.4$)

$$P(s) = \frac{1.82 \cdot e^{-24s}}{60 \cdot s + 1}$$

- Setpoint Tracking ($\tau_c = 27$)

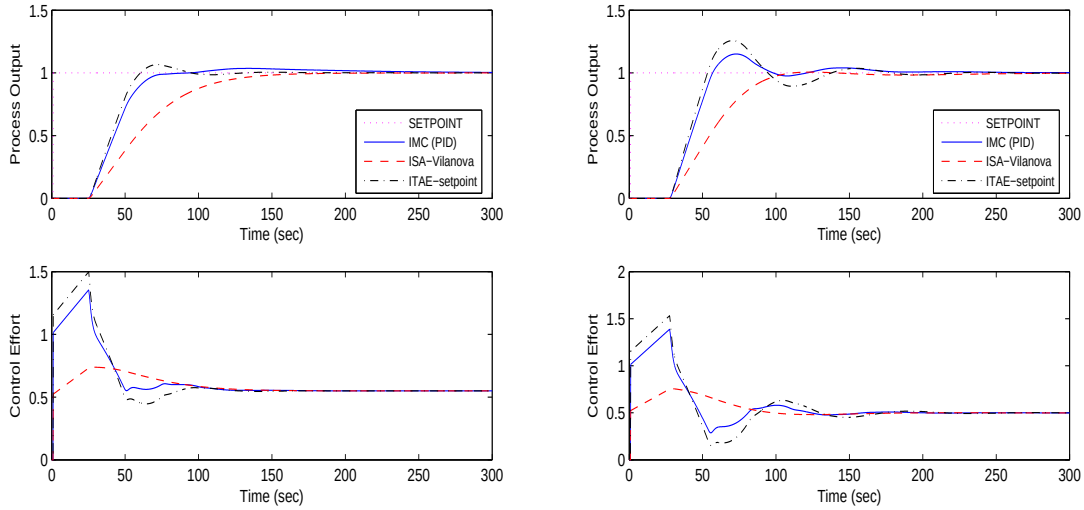


Figure 4.8: Responses for $\alpha = 0.4$, Nominal (left) and with 10% MPM (right)

Tuning	IMC	Vilanova	ITAE
K_c	1.01	0.52	1.16
τ_i	72	60.75	81.37
ISE	37.17	48.39	35.78
$ITSE$	734.9	1335.22	668.85
T_v	161.56	147.95	161.73
$Max(y)$	1.04	1	1.07
GM	2.37	4.12	2.25
PM	70.79	68.79	66.57
DM_n	1.9	3.17	1.54
M_S	1.75	1.41	1.84
M_T	1	1	1
J_{SP}	39	63.76	38.7
J_D	70.98	116.05	70.43
J_U	11.86	0.79	13.34

Tuning	IMC	Vilanova	ITAE
τ_d	10	0.23	7.89
τ_f	1	41.42	0.79
ISE	37.89	46.36	37.98
$ITSE$	761.26	1170.81	806.68
T_v	146.82	135.63	146.96
$Max(y)$	1.15	1.01	1.26
GM	1.87	3.15	1.74
PM	63.88	65.76	55.87
DM_n	1.39	2.61	1.04
M_S	2.18	1.57	2.42
M_T	1.21	1	1.49
J_{SP}	35.45	57.97	35.18
J_D	70.98	116.05	70.43
J_U	12.26	0.88	13.7

- Disturbance Rejection ($\tau_c = 28.8$)

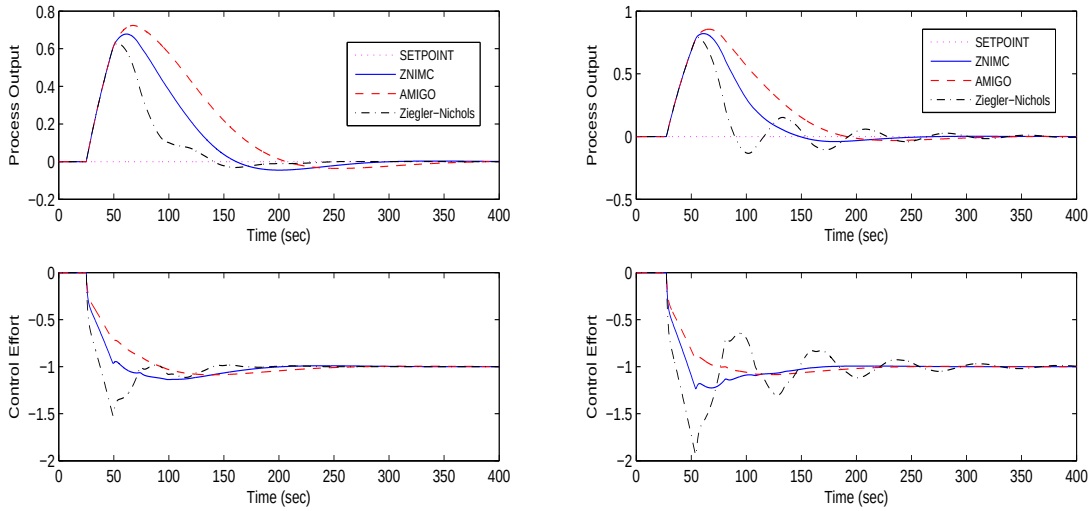


Figure 4.9: Responses for $\alpha = 0.4$, Nominal (left) and with 10% MPM (right)

Tuning	**ZNIMC**	**AMIGO**	**ZN**
K_c	0.97	0.73	1.51
τ_i	42.11	46.08	42.11
ISE	22.92	35.37	12.5
$ITSE$	1639	2947	733
T_v	375	364	384
$Min(y)$	-0.05	-0.04	-0.03
$Max(e)$	0.68	0.72	0.63
GM	2.56	3.28	1.6
PM	52.3	60.1	45.94
DM_n	1.31	1.98	0.76
M_S	1.66	1.45	2.68
M_T	1.18	1.07	1.7
J_{SP}	39.29	45.86	29.88
J_D	44.42	63.5	28
J_U	11.27	8.3	19.64

Tuning	**ZNIMC**	**AMIGO**	**ZN**
τ_d	10	10.71	10.53
τ_f	1	1.07	1.05
ISE	27	40	17.7
$ITSE$	1810	3107	1109
T_v	377	367	385
$Min(y)$	-0.04	-0.03	-0.13
$Max(e)$	0.82	0.86	0.79
GM	2.01	2.6	1.27
PM	47.7	57.65	32.87
DM_n	1.02	1.65	0.43
M_S	2.03	1.64	4.83
M_T	1.24	1.09	3.85
J_{SP}	36.5	41.29	57.03
J_D	43.9	63.3	27.85
J_U	11.6	8.45	21.91

3. Balanced Lag and Delay Case ($\alpha = 0.7$)

$$P(s) = \frac{1.82 \cdot e^{-42s}}{60 \cdot s + 1}$$

- Setpoint Tracking ($\tau_c = 48$)

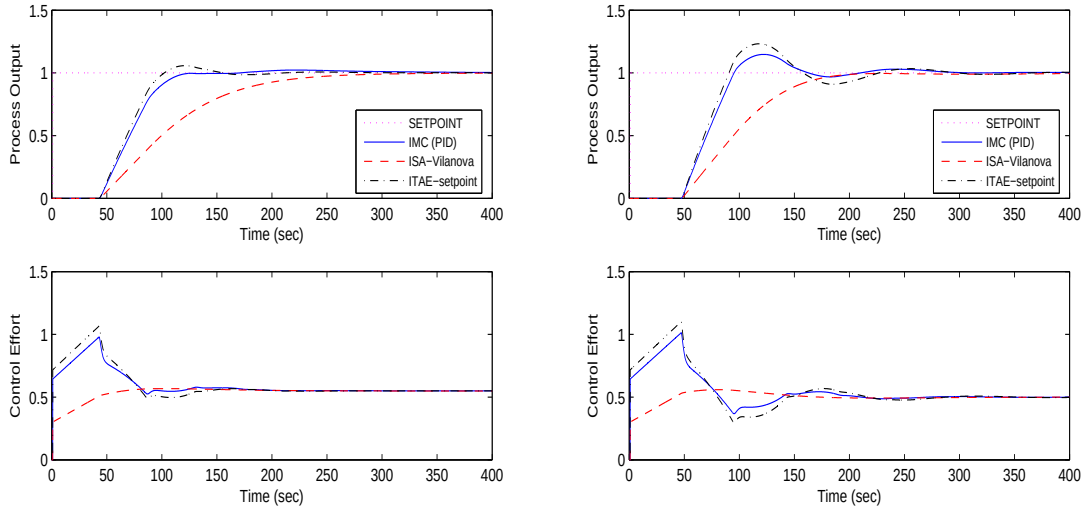


Figure 4.10: Responses for $\alpha = 0.7$, Nominal (left) and with 10% MPM (right)

Tuning	IMC	Vilanova	ITAE
K_c	0.65	0.3	0.72
τ_i	81	61.32	86.52
ISE	62.02	83.21	60.19
$ITSE$	2020.45	3971.66	1888.61
T_v	182.4	159.09	183.95
$Max(y)$	1.02	1	1.06
GM	2.4	4.12	2.26
PM	71.14	68.79	68.91
DM_n	1.93	3.17	1.67
M_S	1.73	1.41	1.83
M_T	1	1	1
J_{SP}	69	111.59	66.21
J_D	125.58	203.08	120.5
J_U	7.53	0.55	8.33

Tuning	IMC	Vilanova	ITAE
τ_d	15.56	-0.27	13.27
τ_f	1.56	72.49	1.33
ISE	63.67	80.07	63.6
$ITSE$	2136.71	3510.22	2216.27
T_v	168.01	148.88	169.12
$Max(y)$	1.15	1	1.23
GM	1.91	3.22	1.78
PM	65.99	66.14	61.12
DM_n	1.51	2.7	1.22
M_S	2.12	1.55	2.34
M_T	1.14	1	1.38
J_{SP}	62.73	101.44	60.19
J_D	125.58	203.08	120.5
J_U	7.78	0.56	8.57

- Disturbance Rejection ($\tau_c = 50.4$)

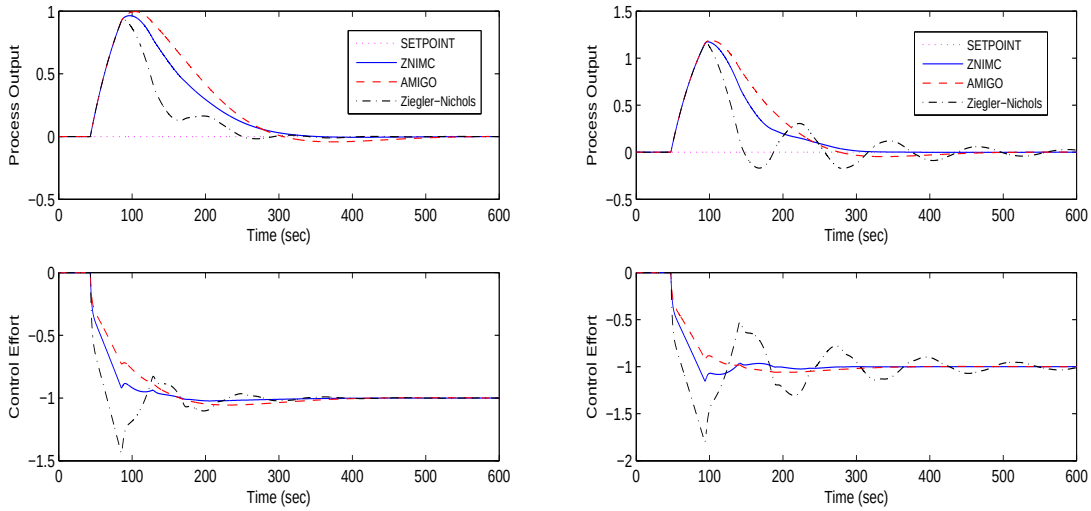


Figure 4.11: Responses for $\alpha = 0.7$, Nominal (left) and with 10% MPM (right)

Tuning	**ZNIMC**	**AMIGO**	**ZN**
K_c	0.62	0.46	0.96
τ_i	68.71	56.7	68.71
ISE	72.2	92.03	43.02
$ITSE$	8386	11509	4198
T_v	538	532	560
$Min(y)$	-0.01	-0.04	-0.02
$Max(e)$	0.96	1.00	0.93
GM	2.51	3.29	1.56
PM	64.7	62.29	54.93
DM_n	1.71	1.98	0.91
M_S	1.67	1.44	2.79
M_T	1.00	1.02	1.8
J_{SP}	62.12	74.86	50.9
J_D	110.2	122.55	71.38
J_U	7.25	5.29	12.7

Tuning	**ZNIMC**	**AMIGO**	**ZN**
τ_d	15.56	17.36	17.18
τ_f	1.56	1.74	1.72
ISE	85.43	107.39	62.25
$ITSE$	9432	12793	6821
T_v	544	538	564
$Min(y)$	0.0	-0.05	-0.17
$Max(e)$	1.17	1.19	1.16
GM	2.00	2.64	1.25
PM	59.9	59.41	40.67
DM_n	1.36	1.68	0.53
M_S	2.03	1.62	4.99
M_T	1.06	1.04	4
J_{SP}	56.06	68.93	97.07
J_D	110.2	122.63	71.38
J_U	7.47	7	14.37

4. Delay Dominant Case ($\alpha = 1.5$)

$$P(s) = \frac{1.82 \cdot e^{-90s}}{60 \cdot s + 1}$$

- Setpoint Tracking ($\tau_c = 100$)

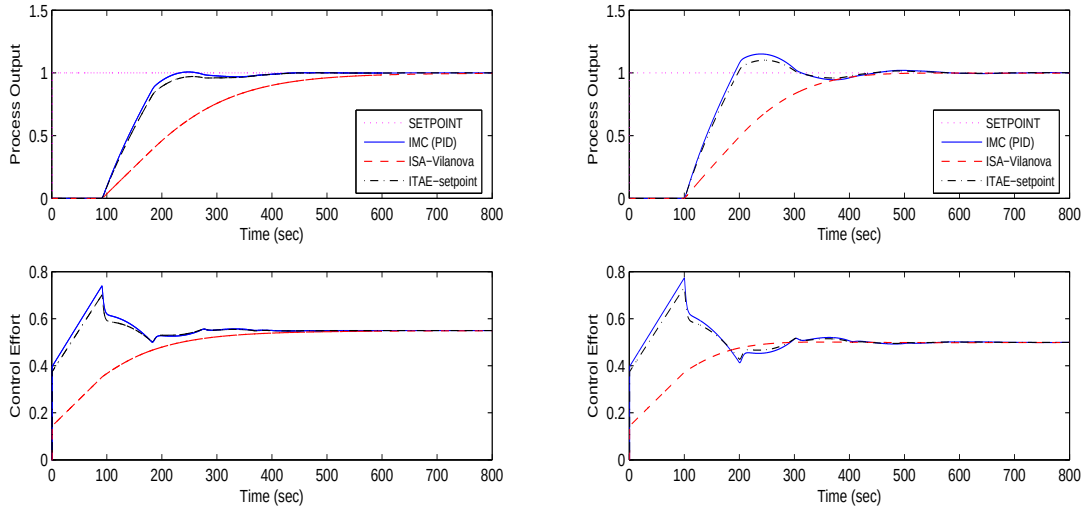


Figure 4.12: Responses for $\alpha = 1.5$, Nominal (left) and with 10% MPM (right)

Tuning	IMC	Vilanova	ITAE
K_c	0.4	0.14	0.38
τ_i	105	62.82	104.12
ISE	123.91	175.8	126.26
$ITSE$	7995.26	17829.88	8362.44
T_v	331.89	280.61	327.88
$Max(y)$	1.01	1	1
GM	2.39	4.12	2.51
PM	70.73	68.79	71.87
DM_n	1.89	3.17	2.04
M_S	1.73	1.41	1.67
M_T	1	1	1
J_{SP}	144.99	239.1	152.29
J_D	263.88	435.15	277.16
J_U	4.66	0.55	4.39

Tuning	IMC	Vilanova	ITAE
τ_d	25.71	-4.28	26.93
τ_f	2.57	155.34	2.69
ISE	128.98	170.71	129.73
$ITSE$	8754.73	16085.96	8754.34
T_v	309.72	267.31	306.45
$Max(y)$	1.15	1	1.1
GM	1.96	3.35	2.06
PM	65.95	65.55	67.64
DM_n	1.55	2.73	1.7
M_S	2.06	1.52	1.95
M_T	1.08	1	1
J_{SP}	131.81	217.37	138.45
J_D	263.89	435.17	277.16
J_U	4.82	0.5	4.54

- Disturbance Rejection ($\tau_c = 108$)

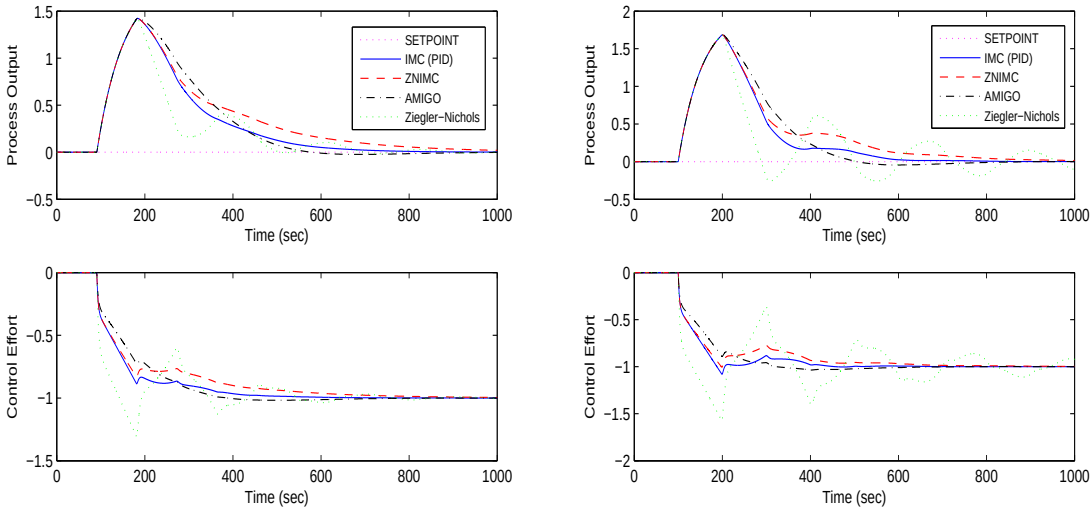


Figure 4.13: Responses for $\alpha = 1.5$, Nominal (left) and with 10% MPM (right)

Tuning	**IMC**	**AMIGO**	**ZN**
K_c	0.38	0.27	0.58
τ_i	105	78.75	130.02
ISE	245.3	283.7	181
$ITSE$	52266	62931	36185
T_v	846	842	876
$Min(y)$	0	-0.02	-0.02
$Max(e)$	1.42	1.43	1.42
GM	2.52	3.38	1.49
PM	71.9	66.26	73.72
DM_n	2.04	2.1	1.43
M_S	1.67	1.42	3.04
M_T	1	1	2.04
J_{SP}	153	157.5	123.03
J_D	278.4	286.64	223.92
J_U	4.38	3.15	8.23

Tuning	**IMC**	**AMIGO**	**ZN**
τ_d	25.71	31.03	32.51
τ_f	2.57	3.1	3.25
ISE	299.1	341.9	264.8
$ITSE$	62382	74454	61931
T_v	860	856	887
$Min(y)$	0.00	-0.04	-0.26
$Max(e)$	1.69	1.69	1.68
GM	2.07	2.79	1.24
PM	67.6	62.49	63.42
DM_n	1.69	1.78	0.98
M_S	1.95	1.57	5.16
M_T	1.00	1	4.15
J_{SP}	139.1	144.66	192.91
J_D	278.4	286.65	223.92
J_U	4.52	3.22	10.01

5. Pseudo-Pure Delay Case ($\alpha = 3$)

$$P(s) = \frac{1.82 \cdot e^{-180s}}{60 \cdot s + 1}$$

- Setpoint Tracking ($\tau_c = 200$)

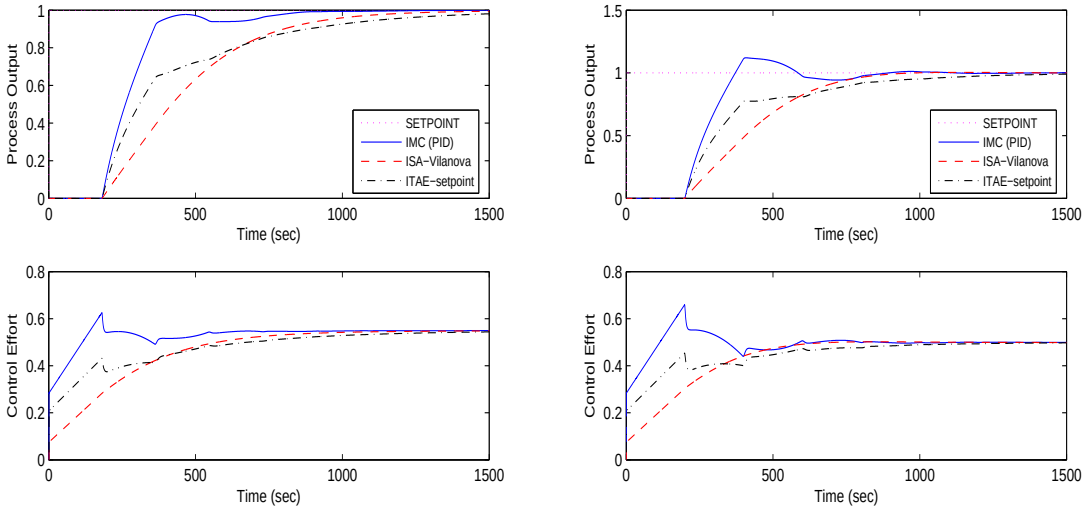


Figure 4.14: Responses for $\alpha = 3$, Nominal (left) and with 10% MPM (right)

Tuning	IMC	Vilanova	ITAE
K_c	0.28	0.08	0.21
τ_i	150	65.64	168.3
ISE	243.11	355.84	293.61
$ITSE$	30757.75	72498.55	52406.72
T_v	796.45	693.36	713.77
$Max(y)$	1.01	1	0.98
GM	2.44	4.12	2.98
PM	70.7	68.79	80.53
DM_n	1.89	3.17	3.4
M_S	1.71	1.41	1.51
M_T	1	1	1
J_{SP}	289.94	478.08	443.42
J_D	527.68	870.08	807
J_U	3.33	0.55	2.45

Tuning	IMC	Vilanova	ITAE
τ_d	36	-21.54	51.28
τ_f	3.6	310.68	5.13
ISE	254.52	349.01	287.34
$ITSE$	33984.08	67085.07	46581.78
T_v	743.21	657.98	673.98
$Max(y)$	1.15	1.01	0.99
GM	2.07	3.43	2.47
PM	65.37	64.97	77.73
DM_n	1.56	2.72	2.96
M_S	1.95	1.51	1.68
M_T	1	1	1
J_{SP}	263.6	434.69	403.18
J_D	527.73	870.24	807.16
J_U	3.44	0.5	2.55

- Disturbance Rejection ($\tau_c = 216$)

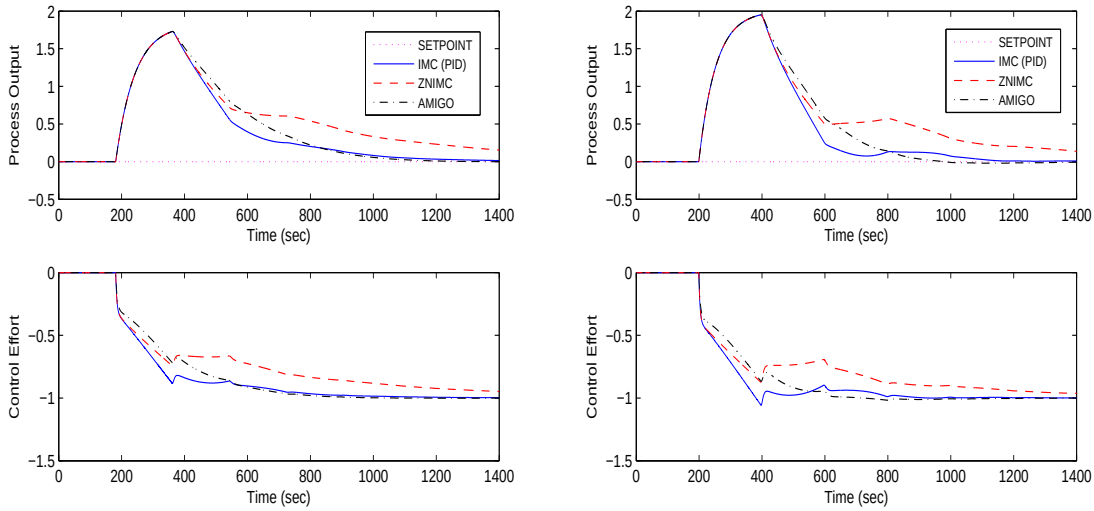


Figure 4.15: Responses for $\alpha = 3$, Nominal (left) and with 10% MPM (right)

Tuning	**IMC**	**ZNIMC**	**AMIGO**
K_c	0.27	0.27	0.19
τ_i	150	230.31	116.13
ISE	603.9	746.9	686
$ITSE$	224112	335425	267580
T_v	1094	955	1067
$Min(y)$	0	0	0
$Max(e)$	1.73	1.73	1.73
GM	2.57	2.58	3.51
PM	71.9	87.4	69
DM_n	2.04	3.68	2.23
M_S	1.65	1.64	1.4
M_T	1	1	1
J_{SP}	305.9	469.3	331.2
J_D	556.8	854	603.78
J_U	3.13	3.13	2.22

Tuning	**IMC**	**ZNIMC**	**AMIGO**
τ_d	36	36	47.37
τ_f	3.6	3.6	4.74
ISE	749.6	881.4	834.6
$ITSE$	280338	384082	325996
T_v	1121	989	1097
$Min(y)$	0	0	-0.02
$Max(e)$	1.95	1.95	1.95
GM	2.18	2.21	2.91
PM	66.9	85	64.58
DM_n	1.39	2.67	1.89
M_S	1.86	1.83	1.52
M_T	1	1	1
J_{SP}	278.1	426.7	301.62
J_D	556.8	854.2	603.84
J_U	3.23	3.23	2.28

4.10 Discussion

The results from the simulation study exhibited promising tuning algorithms. For each case of servo and regulatory control, three different tuning rules were picked. As the response plots hinted, all of them were worthy of consideration and it is not a good idea to completely rule out any of them. The selection of the best is well related to the testing situation and the considerations for a specific application.

In each category, three tuning rules are intentionally chosen in this way: one with very good performance, one with very good robustness and a compromise tuning solution. Table 4.2 ranks these tuning rules as described.

Table 4.2: Ranking of tuning rules based on performance and robustness

$\uparrow$ *Performance*	**Setpoint Tracking**	**Disturbance Rejection**	$\downarrow$ *Robustness*
	ITAE	Ziegler-Nichols	
	IMC	ZNIMC	
	Auto-Vilanova	AMIGO	

For each category the best performing tuning rule gives the best results in ideal conditions, but as the uncertainty inside the system worsens, more robust tuning rules become superior in performance. These results validate the trade-off between performance and robustness for a given process. As it is almost impossible to have an exact model of the system; use of ITAE and Ziegler-Nichols techniques are not recommended in case of non-ideal conditions.

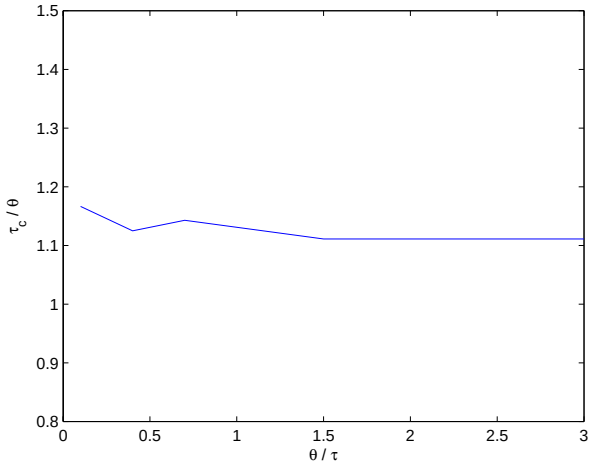


Figure 4.16: Recommended choice of desired closed-loop time constant for setpoint tracking

The results proved that the proposed ZNIMC tuning rule is a good technique in

simulation and results in a compromised solution. It has minor to moderate robustness characteristics and hence is recommended to industrial practitioners for implementation based on these findings.

For the desired closed loop time constant, Figure 4.16 provides a guideline to choose τ_c for setpoint tracking and we have used a similar guideline in case of disturbance rejection.

We conclude that a value around 1.2 times the deadtime is appropriate for the desired closed loop time constant. That would correspond to a 15% overshoot in the presence of 10% MPM.

To set standard values for the PID controller design, performance and robustness indices for the best controller in each modelling region are presented in Tables 4.3 and 4.4. For the case of setpoint tracking the results are consistent and can be used as excellent rules of thumb for the controller design. For the discussed degree of robustness we recommend a gain margin of 2.4, a phase margin of 70°, a maximum sensitivity of 1.7, a maximum complementary sensitivity of 1, and a normalized delay margin of 1.9. Interestingly, the suggested value for M_S is in complete accordance with the design formulation used in [23]. The very low value of variance between data points and wide scope of α make the recommendations highly reliable.

For the case of regulatory control, from the data in Table 4.4 it is not easy to draw meaningful patterns. The reason is that all of the indices are derived using the loop transfer function ($L(s) = P(s) \cdot C(s)$) which is good for describing the servo behavior (based on Equation 2.1a). However in the case of regulatory control the dynamics are not solely determined by the loop transfer function, they are also a function of the process model itself (see Equation 2.1b). That means the indices cannot be defined based only on L(s), they should be defined for each specific modeling region. Therefore, no general index can be defined for the whole design domain. In view of these facts, loop shaping methods will not give optimal results and should not be used for PID regulatory control designs in the framework described above.

The normalized delay margin represented itself as a better criterion than the phase margin. Quantitative data for regulatory control in the lag dominant region show that a difference of less than 15° in the phase margin could result in a normalized delay margin ratio of more than 2.5 between Ziegler-Nichols and AMIGO tuning rules. In other words, the discrepancy can be seen more clearly using the normalized delay margin. This is especially useful in cases of nonstructural robustness analyses.

The values for J-factors were not constant for either servo or regulatory control. Figure 4.17 shows the trends of these variables as a function of α. J-factors are not much suitable to be used as controller design indices due to their variability.

Table 4.3: Calculated robustness and performance indices for the best case scenario (setpoint tracking)

α	GM	PM	M_S	M_T	DM_n	J_{SP}	J_D	J_U
0.1	2.42	71.1	1.72	1	1.95	10	18	40.4
0.4	2.37	70.8	1.75	1	1.9	39	71	11.9
0.7	2.4	71.1	1.73	1	1.93	69	126	7.5
1.5	2.39	70.7	1.73	1	1.89	145	264	4.7
3	2.44	70.7	1.71	1	1.89	290	528	3.3

Table 4.4: Calculated robustness and performance indices for the best case scenario (disturbance rejection)

α	GM	PM	M_S	M_T	DM_n	J_{SP}	J_D	J_U
0.1	2.63	35.8	1.68	1.75	0.93	14.7	4.66	39.5
0.4	2.56	52.3	1.66	1.18	1.31	39.3	44.4	11.3
0.7	2.51	64.7	1.67	1	1.71	62.1	110	7.25
1.5	2.52	71.9	1.67	1	2.04	153	278	4.38
3	2.57	71.9	1.65	1	2.04	306	557	3.13

Simulations proved that the ZNIMC tuning rule performs well in lag dominant and balanced lag and delay modeling zones. However, as alpha approaches unity the ZNIMC tuning rule converges to the conventional IMC rule and beyond that threshold the conventional IMC rule gives dominant results.

"Robustness" is a general term and it has to be defined before the design as one of the control objectives. In this thesis, I investigated situations both with mild degree of robustness and with higher demands of robustness. The controller design in Chapter 4 was carried out based on 10% unstructural MPM in model parameters. The outcome was seen to be robust enough for the identification errors, however, it may not be reasonable when process shifts in the operating points occur. For higher demands of robustness, the results in Appendix A seem to be more relevant where 30% of MPM is assumed for the controller design.

PID controllers are not recommended for delay dominant processes, higher order systems and inherently oscillatory systems [5]. The simulation results presented in the Appendix A showed that the application of derivative action should be limited in noise-rich environments; nonetheless, use of a suitable derivative filter can be relieving. The author also discourages the use of derivative action in the presence of valve stiction.

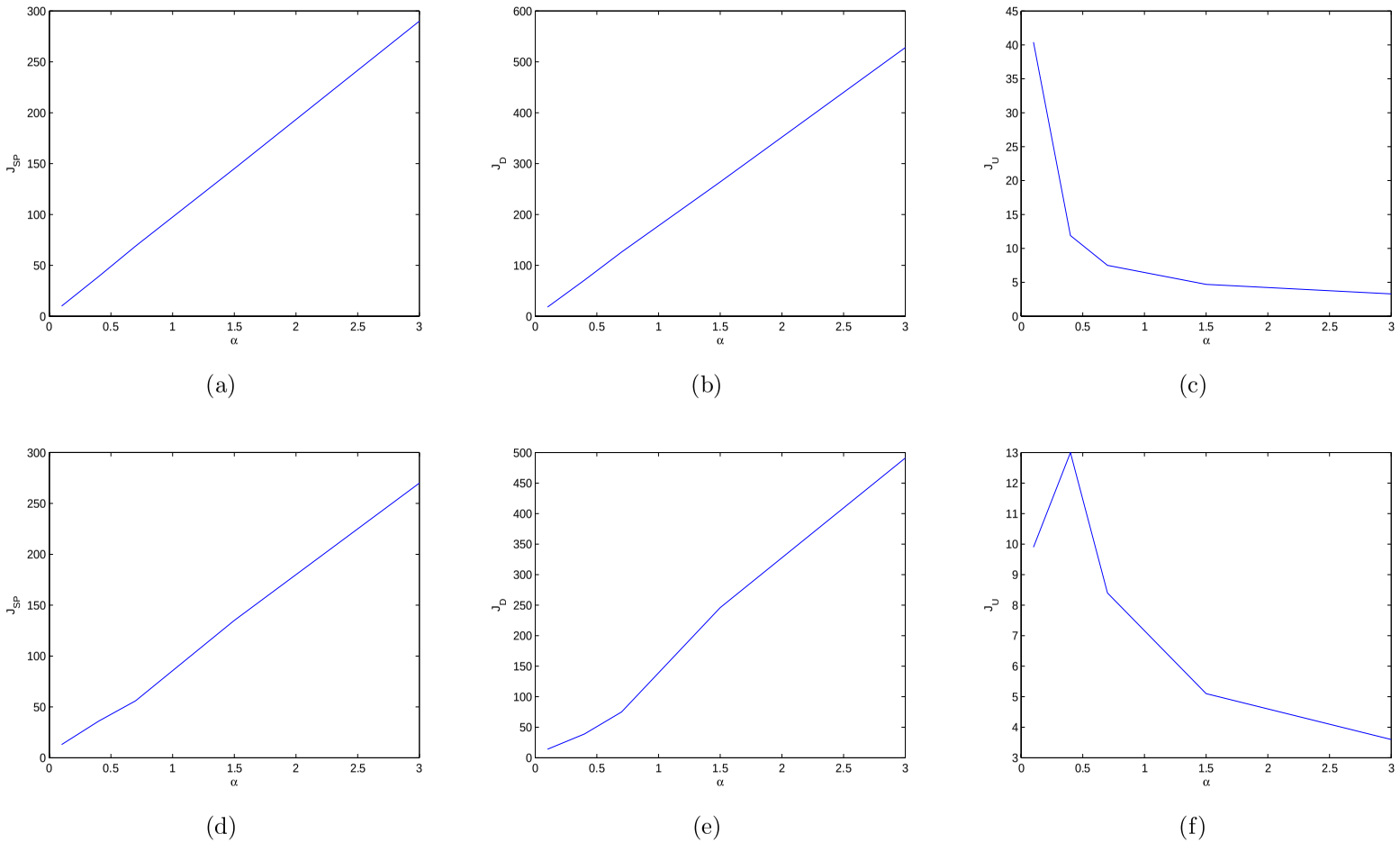


Figure 4.17: J-factors for setpoint tracking [top] and disturbance rejection [bottom], (best case scenario)

Chapter 5

Experimental Design for Robust PID Controller Tuning of Pilot Scale Continuous Stirred Tank Heater

So far we have seen the performance of tuning rules in a simulation environment. However, it has been observed that simulation results may not mimic reality. To confirm and substantiate the findings in the previous chapter, we implemented the techniques on a computer interfaced pilot process located in the process control laboratory in the Department of Chemical and Materials Engineering at the University of Alberta.

The process chosen was the continuous stirred tank heater (CSTH). The objective was to maintain the outlet water temperature at a desired setpoint by regulating the amount of steam flowing through the immersed coils. The process is inherently second order, nevertheless a FOPTD model was obtained for the sake of tuning. In this way, some degree of unstructural MPM was incorporated into the ensuing analysis.

For more comprehensive information about the process and the procedures for identification of the model used here refer to Appendix B.

5.1 Process Model and Choices of PID Controller

The process model from steam flow (kg/hr) to tank output temperature ($^{\circ}C$) was identified as a FOPTD model with a steady-state gain of 1.82, a time constant of 60 seconds, and a deadtime of 38 seconds (See Appendix B.3)

$$P(s) = \frac{1.82}{60s+1}e^{-38s}$$

The process is interfaced to the deltaV (Emerson) distributed control system (DCS). The controller gain for this specific type of interface is defined in dimensionless (normalized) units. To convert the calculated controller parameters to suit the system, the following equality was used:

$$K_c(dimensionless) = K_c(conventional) \times \frac{\text{Range of process varaible}(PV)}{\text{Range of controller action}(OP)} \quad (5.1)$$

For this specific control loop we have:

$$K_c(dimensionless) = K_c(conventional) \times \frac{(100-0)^{\circ}C \ Water \ Temp.}{(40-0)\frac{Kg}{hr} Steam} = 2.5K_c \quad (5.2)$$

The calculated parameters for the PID controller obtained using various tuning rules are presented in Table 5.1:

Table 5.1: Calculated controller parameters

	Name of tuning	K_c	K_c(DCS)	τ_i	τ_d
Servo	ITAE-track (1997) [39]	0.78	1.95	85.3	12.1
	IMC ($\tau_c = 1.2\theta$) (1990) [15]	0.67	1.68	79.0	14.4
	Vilanova (2008) [43]	0.33	0.83	61.2	0
Regulation	Ziegler-Nichols (1942) [47]	1.04	2.60	63.1	15.8
	ZNIMC (2009)	0.67	1.68	63.1	14.4
	AMIGO (2004) [4]	0.5	1.25	54.6	16
	IMC-dist (2007) [37]	0.57	1.43	67.1	8.5

The nominal operating conditions of the CSTH process are given in Table 5.2 (borrowed from Table B.1)

Table 5.2: CSTH process deterministics

Process Variable	**Value**
Water level inside the tank	20cm
Cold water flow	4.8 kg/min
Nominal steam input	10 Kg/hr
Temperature measuring sensor	thermocouple #2
Manual valve position	50%
Ambient temperature	$21^{\circ}C$

5.2 Experimental Procedure

The main objective of this exercise is to evaluate the tuning rules and the robustness of several different PID controllers. Three different scenarios with no, low and significant model plant mismatches were investigated.

In the first scenario, the process was assumed to be operating around the nominal point and exactly matched the identified model, that is, with no MPM consideration.

In the next scenario, the nominal level of water was dropped by 25% to test the suitability of tuning methods in that condition. Decreasing the water level decreased the time constant of the process and also increased the gain and deadtime, though not a significant change of deadtime. (see Appendix B.5.5).

For the last scenario, the measuring sensor was switched to a more downstream sensor (from #2 to #3). This was done using the instrument selector provided by DeltaV DCS. This caused the amount of deadtime to increase by a factor of about 40% (see Table B.3) which is considered a large uncertainty.

In this way, the robustness of the tuning techniques was analyzed by experimentally inducing mismatches to the CSTH process.

To test the tracking performance of the tuning rules, the setpoint was changed from 30 to $35°C$. After the response settled, the setpoint was returned to its original value to start the next test.

For regulatory control, a disturbance was introduced into the process at $30°C$ by opening the hot water valve and maintaining a steady 0.5 kg/min flowrate. The level however was controlled by a level controller and after some time it again reached its nominal (20cm) value by an automatic decrease of cold water flow. This change nevertheless had an interacting effect on the temperature loop used for this study.

The same tests were conducted for the MPM cases, this time by adding 25% MPM in the level (L=15cm) and switching the thermocouple from location #2 to #3.

The attending tuning rules for comparison were chosen to be the same as those in Chapter 4. That is, ITAE, IMC, and Vilanova for setpoint tracking and Ziegler-Nichols, ZNIMC, and AMIGO for regulatory control. We also added a new IMC based tuning rule proposed by Shamsuzzoha and Lee (2007) [37] for regulatory control to extend the scope of comparison.

5.3 Results

5.3.1 Scenario(a): Nominal process

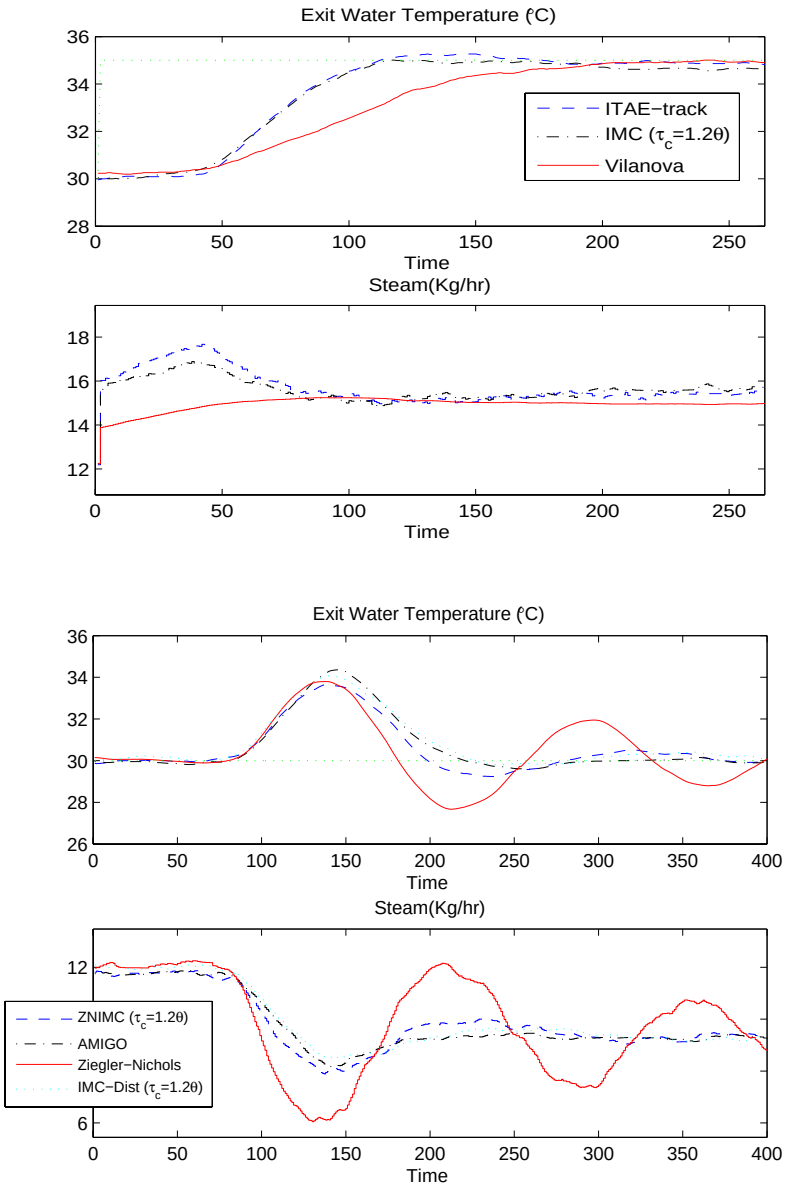


Figure 5.1: Setpoint tracking (top) and regulatory (bottom) behaviour at nominal conditions

5.3.2 Scenario(b): Water level dropped by 25% (L=15cm)

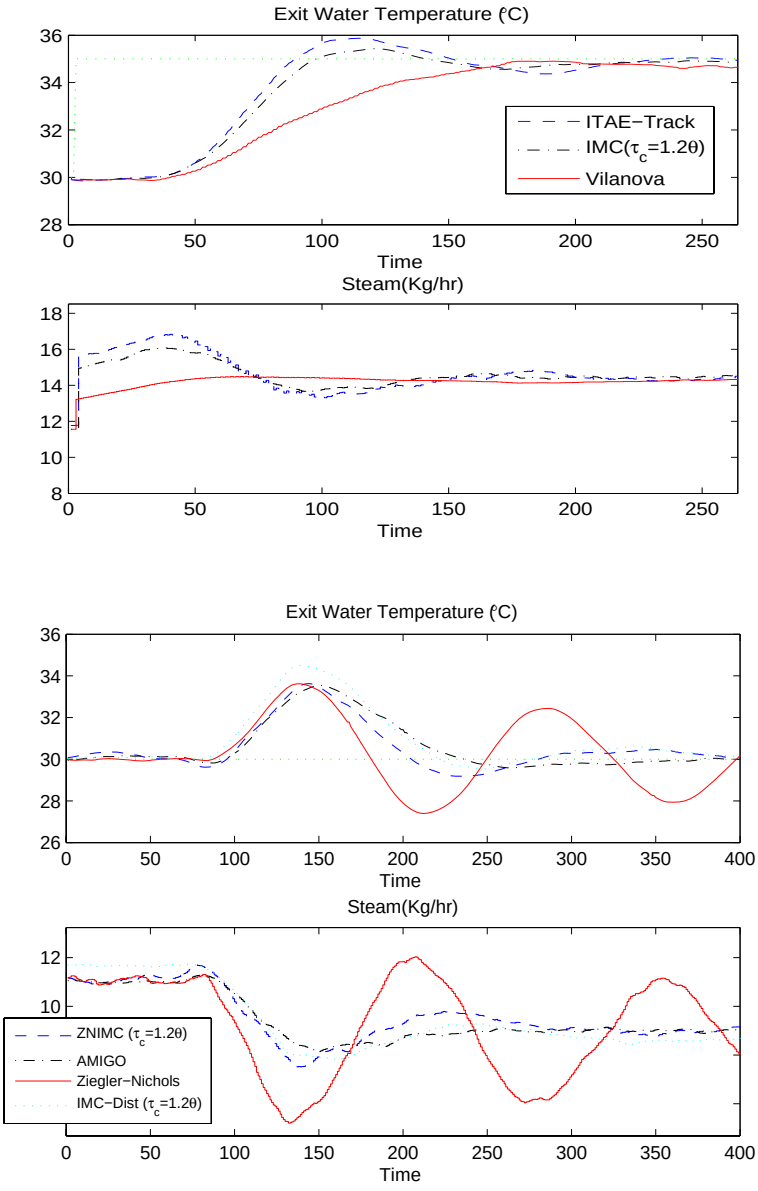


Figure 5.2: Setpoint tracking (top) and regulatory (bottom) behaviour in the presence of 25% MPM (Level=15cm)

5.3.3 Scenario(c): Thermocouple switched from location #2→#3

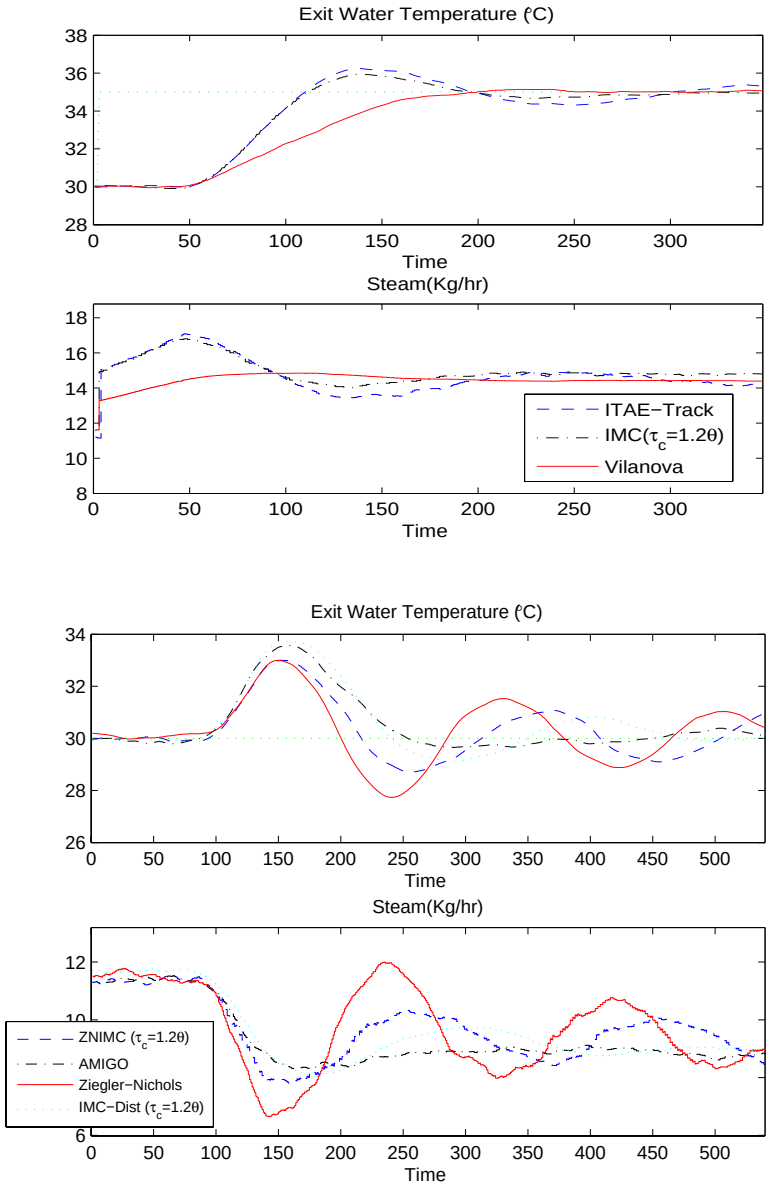


Figure 5.3: Setpoint tracking (top) and regulatory (bottom) behaviour when deadtime is increased

5.3.4 Time domain numerical data

In order to compare tuning rules in a quantitative way, it is possible to measure the performance of the response variable and the control effort by using the time domain indices mentioned in section 2.6.1. Here we used ISE and ITSE for the output and T_v

to measure the control activity or effort. Tables 5.3, 5.4, and 5.5 present the results obtained from the experimental data.

Table 5.3: Table of comparison between ISE, ITAE and T_v indices: Scenario(a)

Tuning method	ISE	ITSE	T_v	
ITAE-Track	1460	**46638**	124	Servo
IMC PID ($\tau_c = 1.2\theta$)	**1417**	46987	98	
Vilanova	1817	83002	**56**	
Ziegler-Nichols	1048	193500	594	Regulation
ZNIMC ($\tau_c = 1.2\theta$)	**705**	**103157**	250	
AMIGO	911	132614	245	
IMC-Dist ($\tau_c = 1.2\theta$)	909	132382	**244**	

Table 5.4: Table of comparison between ISE, ITAE and T_v indices: Scenario(b)

Tuning method	ISE	ITSE	T_v	
ITAE-Track	**1474**	**48951**	177	Servo
IMC PID ($\tau_c = 1.2\theta$)	1504	49790	117	
Vilanova	1915	83116	**52**	
Ziegler-Nichols	1186	262651	689	Regulation
ZNIMC ($\tau_c = 1.2\theta$)	**610**	**93253**	241	
AMIGO	675	104759	**231**	
IMC-Dist ($\tau_c = 1.2\theta$)	1108	166121	263	

Table 5.5: Table of comparison between ISE, ITAE and T_v indices: Scenario(c)

Tuning method	ISE	ITSE	T_v	
ITAE-Track	**1860**	84954	230	Servo
IMC PID ($\tau_c = 1.2\theta$)	1820	**74338**	226	
Vilanova	2129	103494	**80**	
Ziegler-Nichols	844	196780	511	Regulation
ZNIMC ($\tau_c = 1.2\theta$)	674	141018	674	
AMIGO	815	**134968**	**130**	
IMC-Dist ($\tau_c = 1.2\theta$)	1013	179281	200	

The integration time was approximately set to from 0 to 280 seconds in scenarios (a)-(b) and from 0 to 360 seconds in scenario (c) for tracking and from 30 to 430 seconds in scenarios (a)-(b) and from 90 to 540 seconds in scenario (c) for regulation measured from the beginning of each test. The integration was carried out numerically using the trapezoidal technique.

The numerical data for the best performing tuning are highlighted in each section. For tracking, the best result in the nominal case and also in the reduced water level

condition was obtained using ITAE with a small margin compared to IMC. In the case of increased deadtime, ITAE was better in ISE but IMC performed better using the ITAE measure considering the time penalty. In all cases the Vilanova tuning rule resulted in the least control action effort as measured by T_v.

For regulation the best results were obtained by the ZNIMC tuning rule in scenarios(a) and (b); however, in scenario(c) AMIGO performed better compared to ZNIMC. On the basis of the required control effort, AMIGO was generally superior, nevertheless, the IMC-Dist tuning rule was slightly better for the nominal process.

5.4 Discussion

The performance and robustness of selected PID controller tuning methods were evaluated on a real process. The validation results for the real process did not fully match the simulation outcome. Especially for regulatory control, simulation responses did not exhibit undershoot as seen in the real case. The real process gain seemed to be more than that of the model and the results were sometimes oscillatory specially for Ziegler-Nichols tuning rule. The possible causes could be error in the identification process, nonlinearity of the process and even interactions between temperature and level loops.

Another possibility for the observed differences is a variation in disturbance models between the simulation and the experiment. In simulation, a step load disturbance was applied, but in the experiment the disturbance transfer function was not P(s), i.e. $D(s) \neq P(s)$. One of the issues in evaluating tuning rules for regulation is that analysis in scientific reports is mostly carried out based on load disturbance assumptions while something different could occur beyond the simulation. Unfortunately load disturbance simulation is the only available regulatory test tool which tuning rules could be evaluated upon.

Regardless of the discrepancy, the IMC tuning rule proved to be good for tracking; however, the response of the ITAE tuning technique was quite similar. ISA-based Vilanova tuning method was overall very robust but otherwise very sluggish. We recommend using it where significant robustness is needed or when minimal control action is desired and/or when facing a saturation problem.

The proposed tuning rule (ZNIMC) did well for regulation in the real process both in nominal case and in the presence of small MPM with reduced water level. It did not perform well in scenario (c) using thermocouple #3. The reason could be the high degree of uncertainty in that case, around 40% in the estimated deadtime. The amount of assumed mismatch for deadtime was considered to be 10% in the design

stage (see Chapter 4). Therefore more robustness is needed in similar cases. Therefore the desired closed loop time constant should be fine tuned to suit the application requirements. Identification methods normally calculate the amount of uncertainty associated with each parameter and this could be exploited to tune PID controllers. The experimental results also showed that Ziegler-Nichols is too aggressive and hence its use is discouraged in this context.

Use of the ISTE index is preferred over IATE because in long runs it eliminates the effect of small deviations from the setpoint observed under steady-state condition of the experimental results.

Chapter 6

Conclusions and Future Directions

This research effort considered the tuning of PID controllers from a practical viewpoint and did not deal deeply with mathematical derivations common in process control theory. The author feels that the mathematical tools needed to solve this problem are far beyond what is required; the reason why this seemingly simple problem still exists is likely a result of insufficient understanding of the problem itself. Lack of a comprehensive approach to the tuning problem over time is also another factor.

A motivation for this study was to find a simple tuning rule which could be used for regulatory control as industry practitioners need practical guidelines that are easy to implement for tuning PID controllers in process plants. Another motivation for this study was the comments and feedback from industrial practitioners who were interested in exploring alternative tuning rules for disturbance rejection in their tuning software TaiJi-PID.

The theme of the thesis was on controller robustness against performance degradation in cases of model plant mismatch and in situations where process abnormalities are encountered. The study conducted in chapter 4 was mainly concerned with MPM; however, the results presented in the attached paper in Appendix A considers both conditions.

As mentioned in section 4.2, for each control loop there are several control objectives to accomplish. Stability is an important consideration for processes with aggressive operating conditions (i.e., temperature and pressure); nevertheless, for most chemical processes that is not an issue (assuming they are open loop stable). Other than that, a controller faces tracking and regulation objectives. A study by Morari [28] showed that it is not possible to have optimal solutions for both situations simultaneously. Our simulations confirmed this idea. A promising solution for that problem is to use controllers with two degrees of freedom where the main controller is

designed for regulation and a prefilter is used to obtain the desired tracking dynamics.

The choice of regulation or tracking also depends on the specific application. For example, in cascade configurations the inner loop should be tuned based on tracking as it receives the commands(setpoints) from the master loop. Another good example is in MPC applications. Due to frequent changing of setpoints by the MPC server, it is suggested that the client loops be tuned for tracking.

Another point to re-emphasize is the trade off between robustness and performance which is an inevitable constraint due to the non-optimal structure of the PID controllers. In the beginning era of control theory the focus was on maximizing the loop performance, however, in the late 80s robustness grew in importance albeit at the cost of lower performance. In Chapter 4 some tuning rules were ranked according to the robustness and performance. The results again confirmed the trade off between those variables. The best tuning rule in the context of this thesis was the one that gave a compromised behavior. The IMC tuning rule [15] was recommended to be used for setpoint tracking of PID controllers. The proposed tuning rule for regulation (ZNIMC) also exhibited a compromise behavior and gave a reliable tuning technique when the desired closed loop time constant was selected appropriately.

The tuning rule proposed by Vilanova also demonstrated a high degree of robustness for setpoint tracking; however, owing to higher robustness, its response is somewhat sluggish. The author recommends using this tuning rule when higher robustness and lower controller action (both in magnitude and variance) are demanded.

Based on the results and discussions I had during research work, it seems that the best results for PID controller tuning use optimization techniques. The main issue here is the right problem formulation (i.e., selection of the cost function and appropriate weighting factors). A drawback is the local minima trap where the optimal solution is not reached. The author recommends using optimization methods in commercial software packages. On the other hand, for analytical tuning rules, the best result seems to be obtained using pole-placement techniques, especially IMC formulation which deals with that problem systematically. These are all good areas for future studies.

Simulations supported the fact that the bottleneck for the maximum controller performance seems to be highly related to the amount of deadtime in FOPTD processes. Accordingly, using the Smith predictor method for delay dominant modeling regions could be helpful.

The selection of performance and robustness indices in a design approach should be based on problem requirements. I found that the normalized delay margin was a better physical interpreter of the system than phase margin and hence using it for

design is recommended. A nominal value of 1.9 can be estimated for this parameter. The downside of using a normalized delay margin is that it cannot be easily converted to the phase margin and therefore its use is not straightforward in the frequency domain analysis.

In this study, conventional tuning rules were tested and evaluated against each other using the TUNIX Simulation Package. Some did not perform well especially when the nominal model and the process did not match, and some exhibited similar trends to each other with little gain. However, based on our comparison all of the tuning rules discussed in this thesis seemed to be worthy of consideration for future works.

SISO (single input single output) systems were the major focus of this analysis. However MIMO (multiple inputs multiple outputs) cases are of course a good area for future researchers (how could the results be modified for those systems?). There exist a considerable multivariable literature as well especially for TITO (two input two output) systems which the interested reader could refer to. Such problems involve decoupling issues and the construction of decentralized matrices and thus requires additional mathematical tools and knowledge of the linear algebra.

Tuning rules were tested on FOPTD models which are used in industry for modeling self regulating processes. Future analyses could consider integrating and second order processes. The author evaluated continuous time domain systems while industrial controllers are nowadays digital; therefore an extension of this study to sampled data systems would be beneficial to the field of controller tuning.

PID controllers do not provide the most optimal solutions to control problems. However, it seems that most of the potential of PID controllers can now be utilized by means of available tuning rules especially in the case of setpoint tracking. Recently a new LQR based algorithm was suggested by Pannocchia et al. [33] to replace the conventional PID controller. It has incorporated two tuning parameters: Regulator and Estimator, and by using these, it claims higher performance and better robustness than current PID controllers. However, it seems that even if it succeeds as an alternative to PID controller, it will take a long time for it to be adopted.

Bibliography

[1] M. S. Amiri. *A tutorial on various PID controller tuning techniques*. Industrial Presentation, Matrikon, Edmonton, AB, Canada, March 2009.

[2] M. S. Amiri and M. Sahebsara. *Regulatory control using a proposed PID controller tuning technique*. Industrial Presentation, Matrikon, Edmonton, AB, Canada, April 2009.

[3] K. J. Astrom and T. Hagglund. Automatic tuning of simple regulators with specification on phase and amplitude margins. *Automatica,*, 20:645–651, 1984.

[4] K. J. Astrom and T. Hagglund. Revisiting the Ziegler-Nichols step response method for PID control. *Journal of Process Control*, 14(6):635 – 650, 2004.

[5] K. J. Astrom and T. Hagglund. *Advanced PID Control*. ISA - The Instrumentation, Systems, and Automation Society, Research Triangle Park, NC 27709, 2005.

[6] M. Athans. On the design of PID controllers using optimal linear quadratic theory. *Automatica*, 7:643–647, 1971.

[7] T. J. D. Barnes, L. Wang, and W. R. Cluett. A frequency domain design method for PID controllers. *Proceedings of the 1993 American Control conference*, 1:890–894, 1993.

[8] W. Bialkowski. Dreams vs. reality: A view from both sides of the gap. *Control systems*, pages 283–294, 1992.

[9] J. Bolton. *Copper Piping and Pin Hole Leaks*. http://www.gomestic.com/Do-It-Yourself/Copper-Piping-and-Pin-Hole-Leaks.88385, March 2008.

[10] C. Brosilow and B. Joseph. *Techniques of model-based control*. Prentice Hall, 2002.

[11] N. A. Bruinsma and M. Steinbuch. A fast algorithm to compute the H_∞ norm of a transfer function matrix. *Systems & Control Letters*, 14:287–293, 1990.

[12] M. S. Calvoic and N. M. Cuk. Proportional-integral-derivative realisation of optimal linear quadratic regulators. *Proceedings of IEE*, 121(11):1441–1443, 1974.

[13] T. L. Chia and I. Lefkowitz. Model based technique helps tune PID loops control. page 36, 1991.

[14] I. Chien and P. S. Fruehauf. Consider IMC tuning to improve controller performance. *Chemical Engineering Progress*, 33, 1990.

[15] I. L. Chien and P. S. Fruehauf. Consider IMC tuning to improve controller performance. *Chem. Eng. Progress*, 86 (10):33, 1942.

[16] M. A. A. Shoukat Choudhury, S. L. Shah, and N. F. Thornhill. *Diagnosis of Process Nonlinearities and Valve Stiction: Data Driven Approaches*. Springer, 2008.

[17] M. A. A. Shoukat Choudhury, N. F. Thornhill, and S. L. Shah. Modelling valve stiction. *Control Engineering Practice*, 13(5):641 – 658, 2005.

[18] G. H. Cohen and G. A. Coon. Theoretical considerations of retarded control. *Trans. ASME*, 75:827, 1953.

[19] D. B. Ender. Process control performance: not as good as you think. *Control Eng.*, pages 180–190, September 1993.

[20] G. Goodwin, S. Graebe, and M. Salgado. *Control System Design*. Prentice Hall, Imperial College Press, 2001.

[21] C. C. Hang, K. J. Astrom, and W. K. Ho. Refinement of the ziegler-nichols tuning formula. *IEE Proceedings*, 138(2):111–118, 1991.

[22] W. K. Ho, C. C. Hang, and L. S. Cao. Tuning of PID controllers based on gain and phase margin specification. *Automatica*, 31(3):497–502, 1995.

[23] B. Kristiansson. *PID Controllers Design and Evaluation*. PhD thesis, Chalmers University of Technology, Goteborg, Sweden.

[24] M. Lelic. *PID Controllers in Nineties (Presentation)*. Corning Incorporated, Science and Technology Division, Corning, NY, 12/7 1999.

[25] A. M. Lopez, P. W. Murril, and S. L. Smith. Controller tuning relationships based on integral performance criteria. *Instrum. Tech.*, 14(11):57, 1967.

[26] G. K. I. Mann. Time domain based design and analysis of new PID tuning rules. *In IEE Proceedings, Control Theory and Applications*, 148(3):251–261, 2001.

[27] M. Morari, S. Skogestad, and D. E. Rivera. Implications of internal model control for PID controllers, proceedings of the american control conference. page 661, San Diego, CA, 1984.

[28] M. Morari and E. Zafirou. *Robust Process Control*. Prentice Hall, Englewood Cliffs, NJ:, 1989.

[29] Y. Nishikawa, N. Sannomia, T. Ohta, and H. Tanaka. A method for auto tuning of PID control parameters. *Automatica,*, 20(3):321–332, 1984.

[30] A. O'DWYER. *Handbook of PI and PID Controller Tuning rules*. Imperial College Press., second edition, 2006.

[31] Aidan O'Dwyer. PI and PID controller tuning rules: an overview and personal perspective. *ISSC 2006, Dublin Institute of Technology*, pages 161–166, 28-30 June 2006.

[32] K. Ogata. *Modern Control Engineering*. Prentice Hall, second edition, 1990.

[33] G. Pannocchia, N. Laachi, and J. Rawlings. A candidate to replace PID control: SISO-constrained LQ control. *AIChE*, 51(4):1178–1189, April 2005.

[34] K. T. Parker. Design of proportional-integral-derivative controller by the use of optimal linear regulator theory. *Proceedings of IEE*, 119(7):911–914, 1972.

[35] D. E. Rivera, M. Morari, and S. Skogestad. Internal model control for PID controller design. *Ind. Eng. Chem. Proc. Des. & Dev.*, 25:252–265, 1986.

[36] D. Seborg, T. Edgar, and D. Mellichamp. *Process Dynamics and Control*. Prentice Hall, second edition, 2004.

[37] M. Shamsuzzoha and M. Lee. IMC-PID controller design for improved disturbance rejection of time-delayed processes. *Ind. Eng. Chem. Res.*, 46:2077–2091, 2007.

[38] S. Skogestad. Probably the best simple PID tuning rule in the world. page 276h, Reno (NV), USA, November 2001. Annual AIChE Meeting.

[39] C. A. Smith and A. B. Corripio. *Principles and Practice of Automatic Control*. John Wiley, New York, second edition, 1997.

[40] Su Whan Sung, Tai yong Lee, and Sunwon Park. Optimal PID controller tuning method for single-input/single-output processes. *AIChE Journal*, 48(6):1358–1361, 2002.

[41] M. A. Unar, D. J. Maurray-Smith, and S. F. A. Shah. Design and tuning of fixed structured PID controllers a survey. Technical report, Glasgow University, 1996.

[42] G. M. van der Zalm. Tuning of PID-type controllers: Literature overview. *DCT-report*, 2004.

[43] R. Vilanova. IMC based robust PID design: Tuning guidelines and automatic tuning. *Journal of Process Control*, 18(1):61 – 70, 2008.

[44] D. Williamson and J. B. Moor. Three term controller parameter selection using sub optimal regulator theory. *IEEE transactions on Automatic Control*, AC-16:82–83, 1971.

[45] C. C. Yu. *Autotuning of PID Controllers: Relay Feedback Approach (Advances in Industrial Control)*. Springer, second edition, 2006.

[46] J. M. Zhu and M. F. Saucier. An IMC based extended PID controller for sampled data systems. *Proceedings of the 1992 American control conference,*, 1:601–606, 22-24 June 1992.

[47] J. G. Ziegler and N. B. Nichols. Optimum settings for automatic controllers. *Trans. ASME*, 64:759, 1942.

Appendix A: GUIDELINES ON ROBUST PID CONTROLLER TUNING FOR FOPTD PROCESSES

Mohammad Sadegh Amiri and Sirish L. Shah*
Accepted by the 8th World Congress of Chemical Engineering, August 2009, Montreal, PQ

September 16, 2009

Abstract

Proportional-Integral-Derivative (PID) controller tuning guidelines in the process industry have been in place for more than six decades. Nevertheless despite their long design history, PID controller tuning has remained an 'art' and no single comprehensive solution yet exists. Various considerations have been taken into account in PID tuning, but in this study tuning is carried out from a different perspective. The conventional methods seldom take real process abnormalities such as model plant mismatch and/or valve stiction into account. In this paper we attempt to view the issue of PID tuning from this perspective and look for a robust design against real problems in the process industry. Various well known PID techniques are evaluated and compared on a benchmark First Order Plus Time Delay (FOPTD) processes. Extensive simulation results are presented and afterwards an acceptable tuning is picked and recommended. At the end some concluding remarks and remedies are suggested.

Key-words: ISA PID controller - FOPTD model - robust tuning technique - process abnormality: valve stiction, model plant mismatch - disturbance rejection

*Department of Chemical and Materials Engineering, University of Alberta, CANADA Email: amiri1@ualberta.ca, sirish.shah@ualberta.ca

1 Introduction

Proportional-Integral-Derivative (PID) controllers are one of the most essential parts of process control. Thanks to their simplicity and efficiency, they have found wide usage and acceptance in every chemical plant. It is estimated that around thousands of control loops are commissioned in an ordinary refinery and a majority of them are PIDs. Based on an industrial study [11] only about 5-10% of control loops are non-PIDs. O'DWYER [11] has suggested that the PID controller is not recommended for delay dominant processes.

This wide range of application is a significant motivation to introduce reliable tuning methods. There are many tuning techniques available in the literature; however, none of them offer an ideal comprehensive solution to PID performance.

There are some key issues in the tuning of the controller for optimal process performance, an area that needs attention. Considering the facts that process models are rarely precise and accurate plus the existence of process disturbances requires that these issues need to be taken into account when PID controllers are tuned.

The purpose of this work is to evaluate and compare some well known tuning rules and to recommend guidelines for robust tuning methods from a practical point of view. In this paper tuning is carried out to ensure robustness in the presence of abnormalities such as model-plant-mismatch and valve stiction mentioned in the very last paragraph.

1.1 Control Objectives

Consider the block diagram of a feedback system:

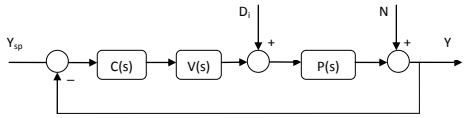


Figure 1: Schematic of a control feedback system

In general there are five main control objectives in the design and configuration of a control loop

- Tracking of operator's setpoints
- Regulation in the presence of process disturbances
- Robustness to model uncertainties
- Attenuation of sensor noise
- Stability and safety considerations

Please note that these objectives are not necessarily mutually inclusive but rather they are contradictory as suggested by Bode's integral theorem. In summary there is always a trade-off between robustness and performance. The existence of such a trade-off between control objectives makes it necessary to prioritize the design for each application. This essentially confirms the idea that there is no single panacea for the tuning of an optimal controller.

In this paper we are more concerned about robust design of system against process uncertainties and its trade-off against overall performance. The first criterion however may unwillingly make the closed-loop system sluggish at the expense of poor performance; nevertheless a treatment to enhance performance is suggested in a later section.

1.2 Choice of process and controller

In this study we have focused on First-Order-Plus-Time-Delay (FOPTD) systems with following dynamics

$$G_p(s) = \frac{K_p}{\tau s + 1} \cdot e^{-\theta s} \tag{1}$$

FOPTD processes are a common type of systems given that the dynamics of most overdamped processes can be sufficiently well characterized by such models. They are also known as three parameter models; process gain, time constant and deadtime being the varying parameters.

For the purpose of evaluation we use the following standard ISA form of the PID controller as widely used in literature with a filter applied to the derivative part.

$$G_c = K_c(1 + \frac{1}{\tau_I \cdot s} + \frac{\tau_D \cdot s}{\tau_f \cdot s + 1}) \tag{2}$$

Based on the recommendation of Brosilow [4] to avoid noise amplification inside feedback loop the ratio of $\frac{\tau_f}{\tau_D}$ is selected to be 0.1. This indeed ensures the system to have $|\frac{G(j\infty)}{G(0)}| < 20$.

2 PROBLEM ANALYSIS

By deriving the closed loop transfer function of the system it is clear that the product of $K_p \times K_c$ should remain constant and therefore the controller gain is a function of inverse K_p. $[K_c = f(1/K_p)]$.

Both τ and θ have the same dimension of time. It is observed that the ratio of the two variables has an important effect on the final response. Towards this end, we define the following dimensionless parameter:

$$\alpha = \frac{\tau}{\theta} \tag{3}$$

Here we start the discussion based on different values which alpha could take. In our view, the FOPTD process model could take anyone of the three categories:

Process type belongs to :

$$\begin{cases} \text{Lag Dominant} & if \quad \alpha < 0.5 \\ \text{Balanced Lag \& Delay} & if \quad 0.5 < \alpha < 1 \\ \text{Delay Dominant} & if \quad \alpha > 1 \end{cases} \tag{4}$$

Please note that the boundaries set for the classification are not sharp and they may need some

slight adjustments. The FOPTD modeling regions are shown graphically in Figure 2.

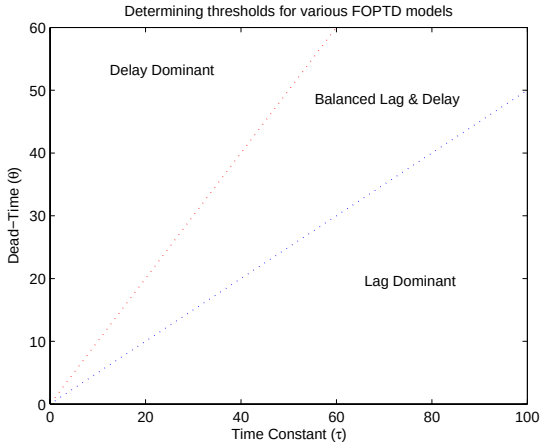


Figure 2: FOPTD modeling zones

3 PROCESS SIMULATION

To compare and evaluate the quality of tuning rules, we have attempted to simulate the performance of controllers designed for each modeling zone. An extensive set of simulation runs have made, however due to the lack of space we will only present the result for just one model per each zone. Conventional step-tracking and disturbance rejection tests are used for comparison.

3.1 Defining the Test Batch

In order to implement a fair test, a standard benchmark was defined for use in different modeling regions:

$$G_p(s) = \frac{e^{-\alpha s}}{s+1} \qquad \alpha \in \{0.2, 0.7, 1.3\} \qquad (5)$$

The following robust tuning methods (such as IMC [5], AMIGO [1], and Vilanova [14]) have been picked. As mentioned before the emphasis was given on performance in the presence of process abnormalities.

3.2 Process abnormalities

To test tuning methods, we introduced three different abnormalities prevalent in process industry: Model Plant Mismatch (MPM), valve stiction and sensor noise.

The problem definition in this paper is as follows: in case of uncertain systems 30% MPM was imposed. That means that every FOPTD parameter was computed based on a 30% error in the gain, time constant and delay in the worst case direction. It is equivalent to higher than normal estimated value of process gain and deadtime and a reduced estimate of time constant.

The valve stiction is introduced with the dead-band including stick-band of 2 and a slip jump of magnitude 1 (See [6]). The valve stiction block is adopted from the study by Choudhury et al. [7]. Finally measurement noise with a variance of $1e^{(-4)}$and zero random seed was applied directly to the process.

The step-tracking and disturbance rejection tests are presented in the same plot. The magnitudes of steps are set equal to 1 for setpoint tracking and a step size of -0.5 in the disturbance variable for regulatory test case. The setpoint is entered at the start of the simulation and the disturbance enters the process once the setpoint level is reached. The results are shown in Figures 3, 4 and 5.

Numerical robustness data corresponding to a single change of each FOPTD parameter in the closed loop condition is provided in Table 1. The models used here are the same as before. For each parameter $\{K_p, \tau, \theta\}$ a 30% positive and negative change is applied to validate the worst case as well as favourable MPM situations. The nominal closed-loop sensitivity plots are also depicted for further reference.

4 RESULTS

The setpoint tracking and disturbance rejection responses in the presence of the process abnormalities were drawn in Figures 3, 4, and 5. Additionally, the sensitivity and complementary sensitivity plots were presented for each modeling region.

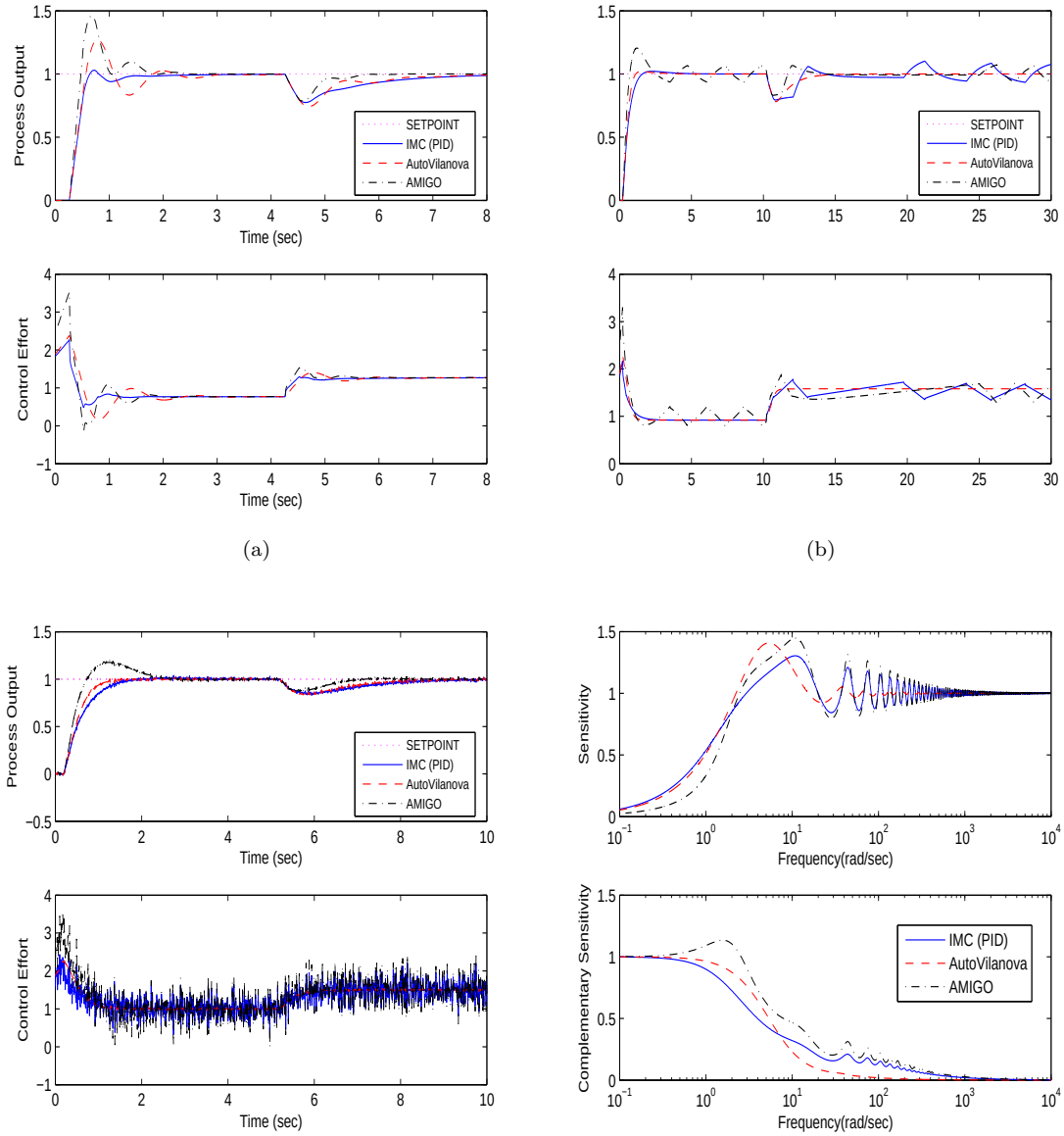


Figure 3: Responses for $\alpha = 0.2$, in presence of (a): 30% mismatch, (b): valve stiction, (c): sensor noise - (d): Sensitivity and complementary sensitivity behaviors obtained by various tuning rules

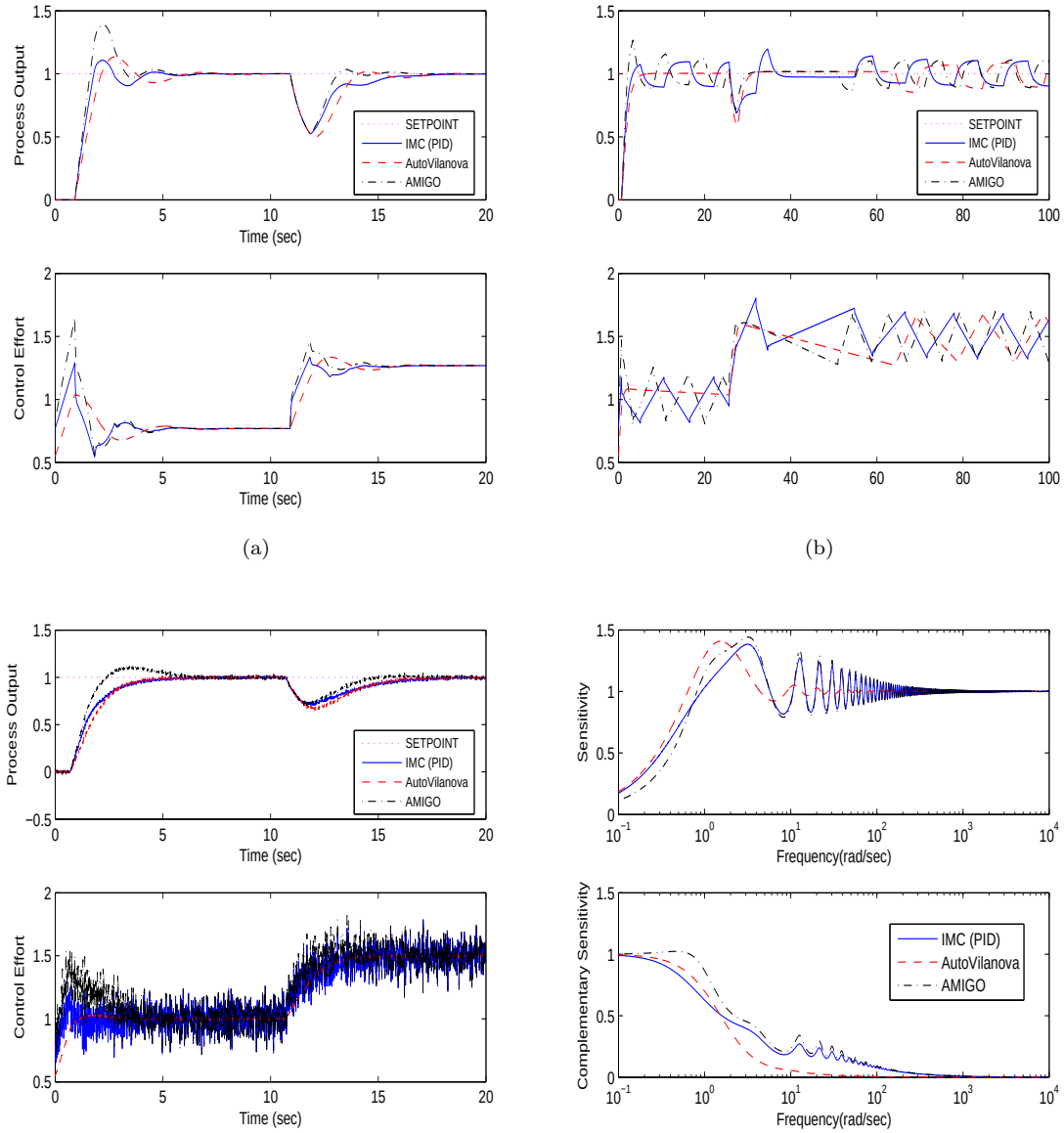


Figure 4: Responses for $\alpha = 0.7$, in presence of (a): 30% mismatch, (b): valve stiction, (c): sensor noise- (d): Sensitivity and complementary sensitivity behaviors obtained by various tuning rules

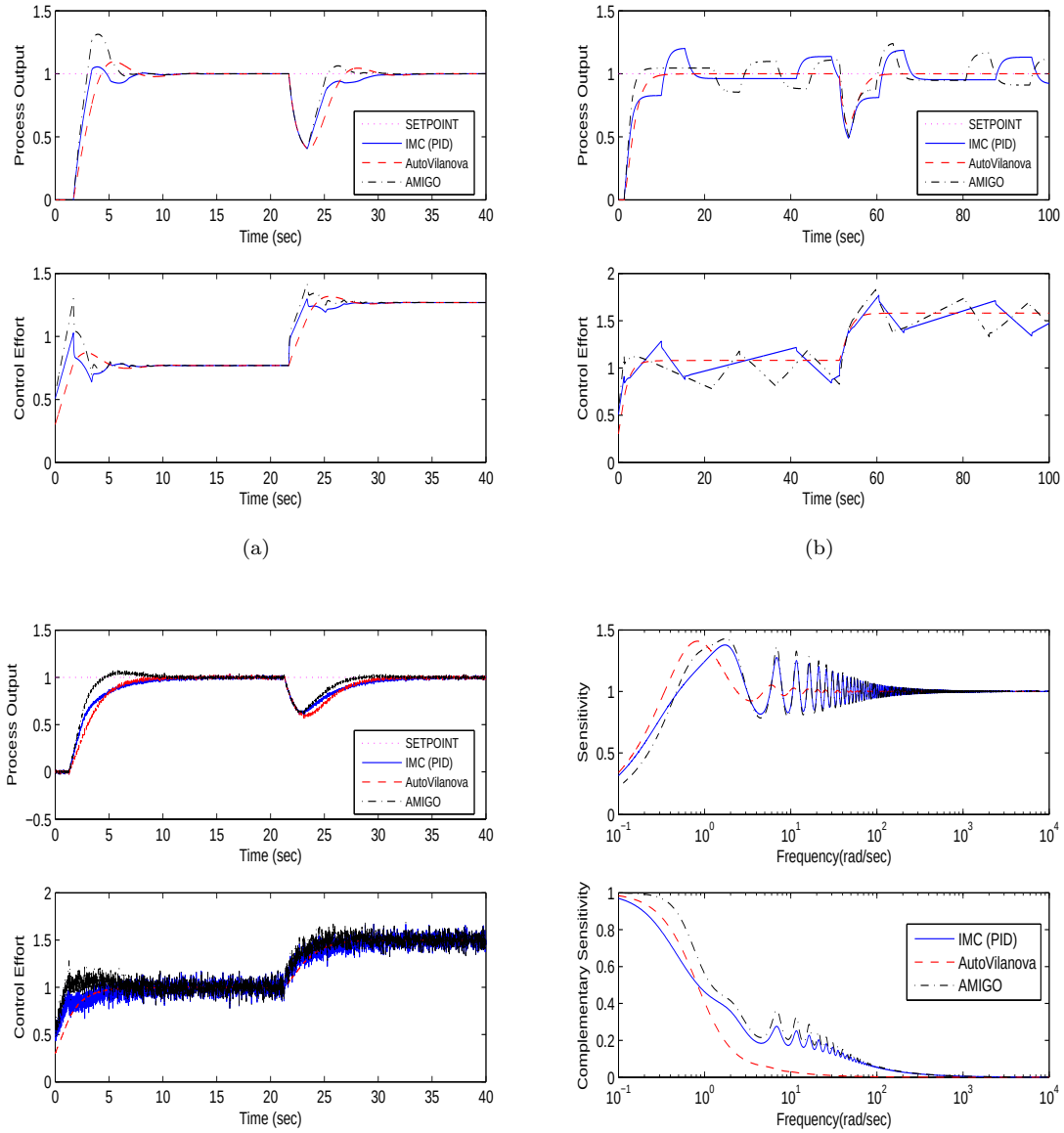


Figure 5: Responses for $\alpha = 1.3$, in presence of (a): 30% mismatch, (b): valve stiction, (c): sensor noise- (d): Sensitivity and complementary sensitivity behaviors obtained by various tuning rules

4.1 Numerical Results

The robustness indices corresponding to each modelling zone were tabulated in the Tables 2, 3, and 4 to give an indication of the robustness.

Table 1: Recommended controller design values for robustness margins

Recommended	GM	PM	DM	MS	MT
Low	2	45	NA	1.2	1
High	5	60	NA	2	1.5

Table 2: Variations of robustness measures by changing FOPTD parameters

Lag-dominant Case						
IMC		GM	PM	DM	MS	MT
+30%	K_p	3.36	77	0.6	1.43	1
-30%		6.23	83	1.24	1.2	1
+30%	τ	5.64	72	0.94	1.22	1
-30%		3.08	82	0.66	1.49	1
+30%	θ	3.78	74	0.77	1.38	1
-30%		4.93	86	0.89	1.26	1
Vilanova						
+30%	K_p	3.17	62	0.44	1.58	1
-30%		5.89	75	0.99	1.27	1
+30%	τ	5.24	66	0.75	1.32	1.02
-30%		3	69	0.48	1.6	1
+30%	θ	3.18	62	0.57	1.58	1
-30%		5.88	75	0.69	1.27	1
AMIGO						
+30%	K_p	2.52	58	0.32	1.67	1.15
-30%		4.69	60	0.56	1.28	1.11
+30%	τ	4.24	52	0.46	1.31	1.22
-30%		2.32	64	0.34	1.77	1.05
+30%	θ	2.85	51	0.35	1.58	1.22
-30%		3.65	68	0.47	1.38	1.08

Table 3: Variations of robustness measures by changing FOPTD parameters

Balanced Lag & Delay Case						
IMC		GM	PM	DM	MS	MT
+30%	K_p	2.81	74	1.68	1.56	1
-30%		5.23	82	3.54	1.24	1
+30%	τ	4.62	72	2.33	1.29	1
-30%		2.7	86	2.35	1.59	1
+30%	θ	3.18	71	2.13	1.48	1
-30%		4.11	85	2.55	1.32	1
Vilanova						
+30%	K_p	3.17	62	1.54	1.58	1
-30%		5.89	75	3.48	1.27	1
+30%	τ	4.86	63	2.2	1.39	1
-30%		3.37	75	2.27	1.48	1
+30%	θ	3.18	62	2.01	1.58	1
-30%		5.88	75	2.43	1.27	1
AMIGO						
+30%	K_p	2.53	57	1.03	1.66	1.07
-30%		4.7	67	2.07	1.27	1
+30%	τ	4.17	56	1.38	1.34	1.09
-30%		2.42	69	1.4	1.71	1
+30%	θ	2.88	52	1.18	1.6	1.13
-30%		3.57	71	1.6	1.39	1

Abbreviations used in the tables stand for [GM: Gain Margin][PM: Phase Margin][DM: Delay Margin][MS: Maximum Sensitivity][MT: Maximum Complementary Sensitivity]

5 CONCLUSION

The results given in the last section reveal that the task of control becomes harder as alpha becomes larger. In general it makes sense to have a detuned controller in this case.

Some tuning rules behaved better in some modeling zones: for example the AMIGO [1] tuning rule had good responses in delay dominant region. The tuning rules proposed by Vilanova [14] also provided an overall robust tuning solution.

The extensive simulations, not included here because of lack of space, showed that other well known tuning techniques such as Tyreus-Luyben [9], Cohen-Coon [8], ITAE [13] and Ziegler-Nichols [15] no longer satisfy uprising control demands. They sometimes give oscillatory dynamics or end up being unstable especially in case of severe abnormalities.

From the comparison of tuning techniques it has been deduced that the IMC formulation gives the best solution in terms of performance and robustness. In most cases the correct choice of the desired closed loop time constant (τ_c) is of great importance. A rule of thumb to select the desired closed loop time constant is given in Table 2. In simulations, IMC [5] PI controller exhibited itself as a more robust controller compared with PID version considering the same τ_c.

The presence of measurement noise proved to be challenging especially in dealing with the required control effort. The tuning rules with large derivative time constants caused a greater variance in valve stem position. We recommend using a PI controller in noise rich environments. Although all of the techniques have derivative filters mounted inside the controller, using stronger derivative filter could also help.

In summary, in abnormal conditions (i.e. presence of valve stiction, model plant mismatch and sensor noise), we recommend using IMC tuning with the derivative action turned off (PI version) for servo-control and AMIGO tuning technique for regulatory-control. Please note that in the design of tuning constants there is always a major side effect due to trade-off between robustness and performance. To avoid excessive sluggishness of response we suggest using a two-degree of freedom controller where possible.

References

[1] K.J. Astrom and T. Hagglund. Revisiting the Ziegler-Nichols step response method for PID control. *Journal of Process Control*, 14(6):635 – 650, 2004.

[2] K.J. Astrom and T. Hagglund. *Advanced PID Control*. ISA - The Instrumentation, Systems, and Automation Society, Research Triangle Park, NC 27709, 2005.

[3] W. Bialkowski. Dreams vs. reality: A view from both sides of the gap. *Control systems*, pages 283–294, 1992.

[4] C. Brosilow and B. Joseph. *Techniques of model-based control*. Prentice Hall, 2002.

[5] I.L. Chien and P.S. Fruehauf. Consider IMC tuning to improve controller performance. *Chem. Eng. Progress*, 86 (10):33, 1942.

[6] M. A. A. Shoukat Choudhury, S. L. Shah, and N. F. Thornhill. *Diagnosis of Process Nonlinearities and Valve Stiction: Data Driven Approaches*. Springer, 2008.

[7] M. A. A. Shoukat Choudhury, N. F. Thornhill, and S. L. Shah. Modelling valve stiction. *Control Engineering Practice*, 13(5):641 – 658, 2005.

[8] G.H. Cohen and G.A. Coon. Theoretical considerations of retarded control. *Trans. ASME*, 75:827, 1953.

[9] W.L. Luyben and M.L. Luyben. *Essentials of Process Control*. McGraw-Hill, New York, 1997.

[10] M. Morari and E. Zafirou. *Robust Process Control*. Prentice Hall, Englewood Cliffs, NJ:, 1989.

[11] A. O'DWYER. *Handbook of PI and PID Controller Tuning rules*. Imperial College Press., second edition, 2006.

[12] D. Seborg, T. Edgar, and D. Mellichamp. *Process Dynamics and Control*. Prentice Hall, second edition, 2004.

[13] C.A. Smith and A.B. Corripio. *Principles and Practice of Automatic Control*. John Wiley, New York, second edition, 1997.

[14] R. Vilanova. IMC based robust PID design: Tuning guidelines and automatic tuning. *Journal of Process Control*, 18(1):61 – 70, 2008.

[15] J.G. Ziegler and N.B. Nichols. Optimum settings for automatic controllers. *Trans. ASME*, 64:759, 1942.

Table 4: Variations of robustness measures by changing FOPTD parameters

Delay Dominant Case						
IMC [5]		GM	PM	DM	MS	MT
+30%	K_p	2.84	74	3.13	1.55	1
-30%		5.28	82	6.57	1.24	1
+30%	τ	4.47	74	4.23	1.3	1
-30%		2.91	83	4.5	1.52	1
+30%	θ	3.21	71	3.95	1.48	1
-30%		4.13	85	4.73	1.32	1
Vilanova [14]						
+30%	K_p	3.17	62	2.87	1.58	1
-30%		5.89	75	6.46	1.27	1
+30%	τ	4.46	65	3.99	1.42	1
-30%		3.75	73	4.29	1.41	1
+30%	θ	3.18	62	3.73	1.58	1
-30%		5.88	75	4.51	1.27	1
AMIGO [1]						
+30%	K_p	2.58	60	1.92	1.64	1.03
-30%		4.79	72	4.14	1.27	1
+30%	τ	4.08	61	2.6	1.38	1.02
-30%		2.62	71	2.81	1.62	1
+30%	θ	2.9	56	2.3	1.62	1.07
-30%		3.55	75	3.08	1.39	1

Table 5: Rule of thumb guidelines on determining τ_c (IMC formulation)

FOPTD Modeling Zone	Closed-loop time constant
Lag Dominant	$\tau_c = max\{\frac{\tau}{2}, 2\theta\}$
Balanced lag and delay	$\tau_c = \gamma\theta$
Delay dominant	Performance based $\rightarrow \tau_c = \theta$ Maximum Robustness $\rightarrow \tau_c = \gamma\theta$
Hint: Use $\gamma = 2$ for PID and $\gamma = 1.5$ for PI controller structures	

Appendix B

Continuous Stirred Tank Heater (CSTH) with Time Delay Process

B.1 Process Description

The CSTH process (P-4) consists of a transparent glass tank which has two input streams; one regulates the cold water inlet and the other regulates the inlet for hot water. The contents of the tank are heated by a steam coil mounted inside the tank. Two inlet stream baffles cause the inlet water to enter the tank without causing the level to fluctuate much and also ensure good mixing. In this way, the water temperature inside the tank can be assumed to be constant at steady-state conditions.

Figure B.1: CSTH process

B.1.1 Process Control

Two control loops act on this process; the first controls the fluid level inside the tank, while the other takes care of the output water temperature. Both of these goals are accomplished using conventional PID controllers. An MPC controller is also installed on this machine to test advanced control strategies in the higher level class.

In our test, we were concerned only about the temperature loop. This is traditionally known as a single input single output (SISO) system. Furthermore, we used

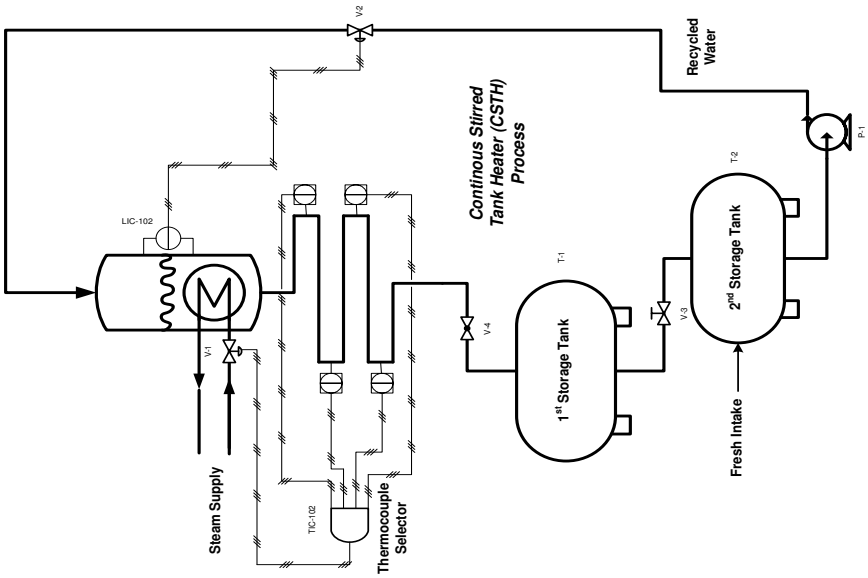


Figure B.2: Schematic diagram of CSTH process

only the cold water stream and the hot water valve was kept shut throughout the experiment. It is desirable to have a model from the steam flow setpoint to the exit water temperature of the tank.

In order to change the delay of the process, the output line was initially designed as long section of a transport process and four thermocouples are placed at different downstream locations. Therefore various transportation delays could be entertained by selecting the appropriate sensor location.

This structure would be useful to investigate the robustness of tuning rules against a process model (i.e., time delay uncertainty). Inside the process, there exists a thermocouple selector which could be used to switch the active readings of the system to any of the thermocouples. It is even possible to use the median or average of all sensors.

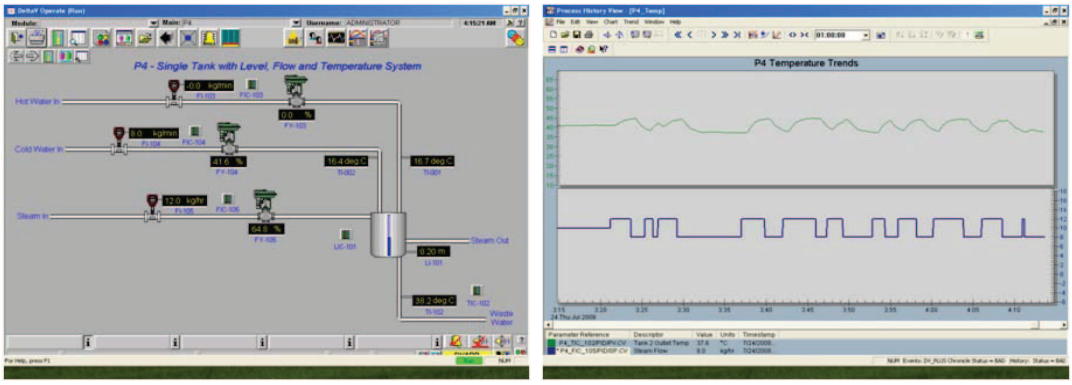
Please notice that manipulating the inlet cold water flow rate can change the transportation delay. Decreased flow rate corresponds to increased deadtime of the system and vice versa.

This pilot scale has been interfaced to a DeltaV DCS from Emerson Process Management. This allowed realtime interaction between the process and the operator. Almost all important process quantities are superimposed on the process flow diagram.

PID controllers in the process can be set into three different working schemes: the "manual" setting is used by the operator to have the process in an open loop condition, the "auto" setting is the closed loop instruction format to control the process around the operator's specified setpoint, and in the "cascade" when the slave loop is closed but the setpoint is determined by the master control loop.

B.2 Process Variables

Real-life processes rarely behave linearly. For our analysis a linear model of the process was required. To obtain a linear system, we linearized the non-linear system around



(a) (b)

Figure B.3: Left panel: DeltaV human machine interface (HMI) for the CSTH process, right panel: Process History View during identification

Table B.1: CSTH process deterministics

Process Variable	**Value**
Water level inside the tank	20cm
Cold water flow	4.8 Kg/min
Nominal steam input	10 Kg/hr
Temperature measuring sensor	Thermocouple #2
Manual valve position	50%
Ambient Temperature	$21°C$

a chosen operating point. In this case, we assumed fixed values for the variables other than the manipulating and controlled variables and we only excited the MV around a narrow neighborhood of its steady-state value.

The level inside the tank was set to be 20cm. The manual output valve was positioned at 50%. The steam setpoint was changed around the nominal value of $10(\pm 2)$ kg/hr. The thermocouple #2 was used for sensing the exit measurements. Finally the room temperature was about $21.2°C$ during the test. Table 5.2 summarizes this data.

B.2.1 Process Disturbances

There are a variety of disturbances: room temperature plus cold water and steam flow rates fluctuations are some examples. A more important disturbance happens due to the interaction of control loops with each other, common in the MIMO systems. Even though our analysis was based on a single variable (constant level), changing the temperature could cause a disturbance on water level because of the change in liquid density. We can see that even in this simple case the decoupling principle does not hold true 100%.

B.3 Process Identification

To obtain a model for this process, system identification techniques will be used throughout this section. The algorithms and peripheral considerations for process identification are beyond the scope of this thesis. However, we attempt to present the essence of a hands-on process identification with an eye for some practical aspects.

B.3.1 Abstract

The CSTH process in this special case is a SISO system. The manipulating variable (MV) is selected to be the steam flow setpoint which can affect our controlled variable (CV), the output water temperature.

B.3.2 Preliminary Step Tests

In order to find the initial operating points and to determine the approximate time constant, the process is excited with some basic step tests.

B.3.3 Input Signal Design

There are a variety of input structures for system identification; white noise is the best identifying signal; but the most impractical due to the large variance in control action. Other options are RBS (random binary signal), PRBS (pseudo random binary signal) and sine waves. RBS is the most commonly used excitation signal in industrial identification and it is used here.

There are two important defining characteristics for every RBS. One is the two bands of signal between which the process is going to oscillate. Here 8-12 kg/hr are assumed for the steam flow setpoint and is chosen symmetrically around a nominal value of 10 kg/hr. The other important quantity is the excitation frequency band; the lower band is chosen as zero whereas the upper is chosen using the relation $\omega_{max} = \frac{k \cdot T_s}{\pi \cdot \tau}$. For this identification k is selected as 2, the time constant is obtained from section B.3.2.

For choice of the sampling time, there are a variety of criteria; here we used one tenth (of the dominant time constant) rule of thumb, leading to 5 sec as T_s. With the sample time and levels decided, the following command can be used to generate our desired identification signal in the MATLAB:

$$u = idinput(N, type, band, levels) \tag{B.1}$$

N is the number of generated signal data points which corresponds to an identification time span. In our case, about three hours is deemed sufficient to fully excite the system.

B.3.4 Application of the Identification Signal

In applying the RBS signal to the system, we needed to have the MATLAB connected to the process. OPC standards can be used to establish a connection between the process server and the host machine where MATLAB is running on it.

The OPC toolbox was used in the MATLAB to allow direct communication with the process. Consequently, data could be read and written from/to the real-time working process. We then wrote a code in which given an input matrix, the process is excited at each sampling time using a timer and the output signal is measured and

stored in the workspace at the end of the run. For more information on the OPC toolbox the reader is referred to Appendix C.

B.3.5 Analysis of Output Data

The first preliminary step to obtain an output signal was to find whether there existed any bad data during the process of identification. One method to find inconsistent data is outlier detection, that is, the points outside the confidence limit around the arithmetic mean. A one step ahead (horizon) prediction model of the system was estimated and then subtracted from the original data, then the outlier test was applied and inappropriate data is pinpointed. The confidence limit was set as five times of standard deviation. The result is depicted in the following graph.

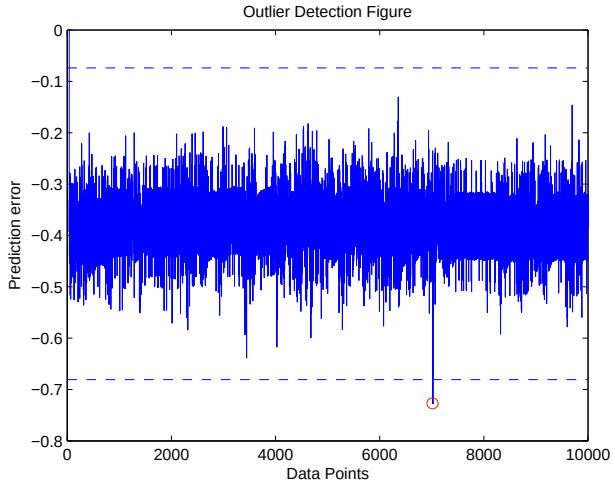


Figure B.4: Outlier detection on CSTH data

As observed, there is an outlier at the sampling time around 7000 sec; the average of neighborhood data points was used for correction. A more perfect treatment would perform two replacements, first with average data and then by estimating a model for the second substitution. In this way the effective trend of the data could be translated more strongly into the original data.

Input and outputs of the CSTH process are shown Figure B.5. The upper graph depicts CV variations while MV is plotted in the bottom graph.

A fair amount of sensor and process noise can be seen in the measurements. This is a realistic fact about industrial processes and is considered an important issue to be considered in the task of control.

The impulse response of data was calculated using the "cra" function in MATLAB. It showed the correlation between different data points at different time delays. Process data is supposed to be uncorrelated at least before the process starts to show reaction. Using this fact, the amount of time delay can be estimated. A time delay of 35-36 seconds is visually estimated while by using "delayest" function a delay equal to 38 seconds was obtained (compare this with Figure B.6).

In the next stage, an identification model was obtained using the ARX model which is one of the simplest formats used for dynamic systems. An equation like $A(q^{-}1) * y(t) = B(q^{-}1) * u(t) + e(t)$ was used to fit the data. In this equation, A and B are the the ARX model parameters which should be determined and e(t) represents the white noise signal.

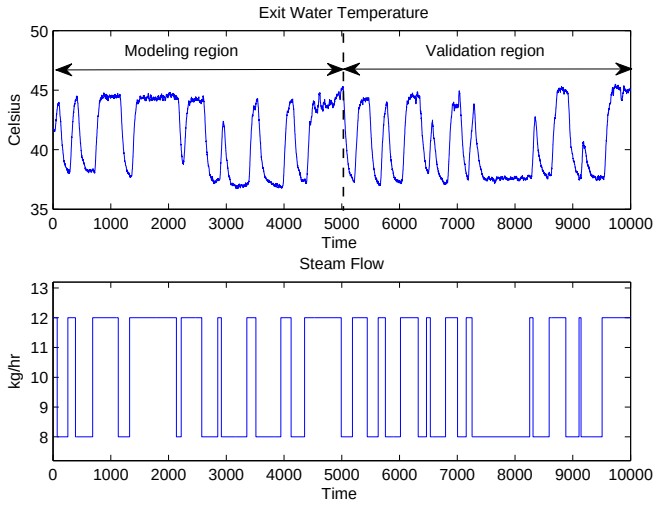


Figure B.5: PV and MV trends in CSTH identification

To check for the 'goodness-of-fit' of the model, two residual tests are normally conducted. The auto-correlation test on the residual error validates the quality of the model whereas the cross-correlation test between the input and the residuals gives an indication of the dynamics that may have been missed.

After the convergence of identification parameters is ensured, it is desirable to test the model against a real process. To do this, the identification data was divided into two parts, one part, called the training data, is used to obtain the model while the other part, the validation data, is used to check the suitability of model in real world conditions. This is also called as an infinite (horizon) step prediction test. Figure B.8 shows that our model successfully passed the test.

To evaluate the quality of the calculated model, step and impulse responses of the model were plotted. The responses represent the degree of stability of the identified model.

Other traditional tools for analyzing data and obtaining a model are frequency based methods used in signal processing. The Periodogram and power spectrum of input and output data as well as the empirical estimate of transfer function from MV to CV were used for that purpose.

Finally, the frequency response of the obtained model was plotted against the periodogram and power spectrum responses in one single graph. This gives an indication of the type of dynamics captured by the simplified model.

In Figure B.11 we see that the phase response of our model behaved well at low frequencies which is more important in the context of process control (slow dynamics). However, the amplitude response showed a little gain mismatch between our model and the real process. This might be explained as being due to process nonlinearities inherent inside the process. One way to modify the model to match the reality is to multiply model gain with an appropriate correcting factor.

B.4 Process Model

The process model was identified twice. In the first RBS excitation a second order model was obtained as:

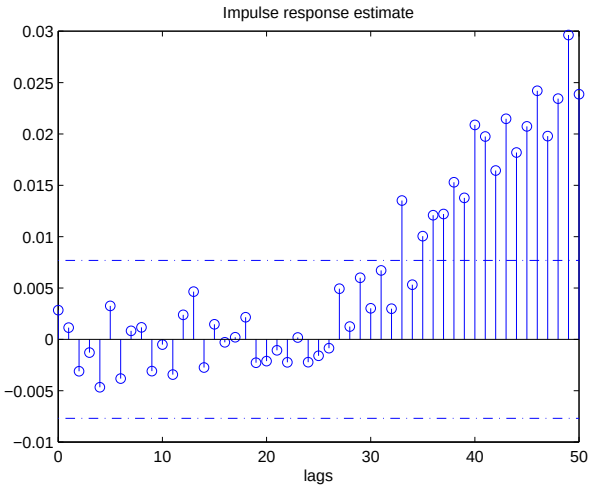


Figure B.6: Estimation of CSTH deadtime

$$P(s) = 1.77\frac{(1.47s+1)e^{-35s}}{632s^2+56s+1} \tag{B.2}$$

This model has a natural time constant of approximately 25 seconds and a damping ratio of 1.11. Therefore the process could be well modeled using a first order model. In the next step a lower order model was identified by means of a different RBS signal with reduced bandwidth in order to neglect higher dynamics (refer to section 2.3 for information on model order reduction). That is:

$$P(s) = \frac{1.82 \cdot e^{-38s}}{60s+1} \tag{B.3}$$

This is well known as a first order plus time delay (FOPTD) model, which is suitable to predict the behavior of nonoscillatory processes. Relative deadtime is ($\alpha = L/T = 0.63$) and is therefore considered to be moderately difficult to control.

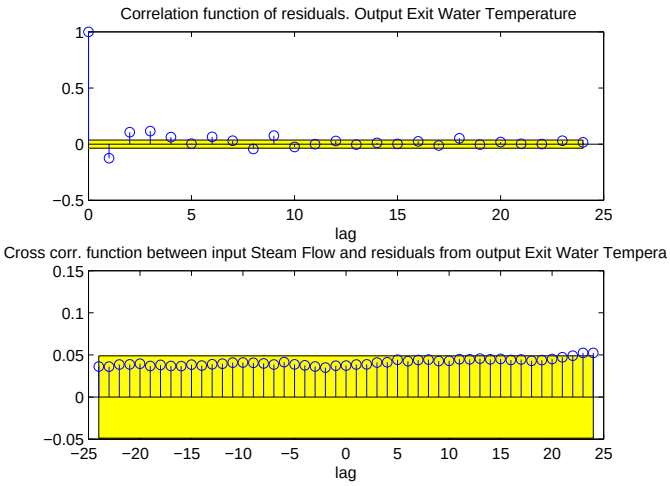


Figure B.7: Auto-correlation and cross-correlation tests

B.5 Mathematical Derivation of First Principle Models for the CSTH Process

B.5.1 Overview

In this section, we use material and energy balances throughout the derivation of level and temperature models. This can be briefly represented by:

$$\textit{Input - Output } \{\textit{+ Generation / - Consumption}\} = \textit{Accumulation} \tag{B.4}$$

In this pilot scale process, there are three manipulating variables (MVs): cold water, hot water, and steam flow. These variables affect the water level and the output temperature of water, the two controlled variables (CVs). The model for the whole system is derived; however in real experiments, cold water and steam are used as active variables and hot water flow can be considered as a source of disturbance to the process. A schematic of the process is shown in Figure B.12:

B.5.2 Assumptions Used to Derive First Principle Models of CSTH Process

- Process is assumed to be linear and therefore the superposition principle holds
- Output flow is linearly dependent (with constant R) on the water level
- The process tank is considered as a lumped heat-transfer object (the tank content is stirred by means of mounted baffles)
- Dynamics of valves and pipes have been neglected

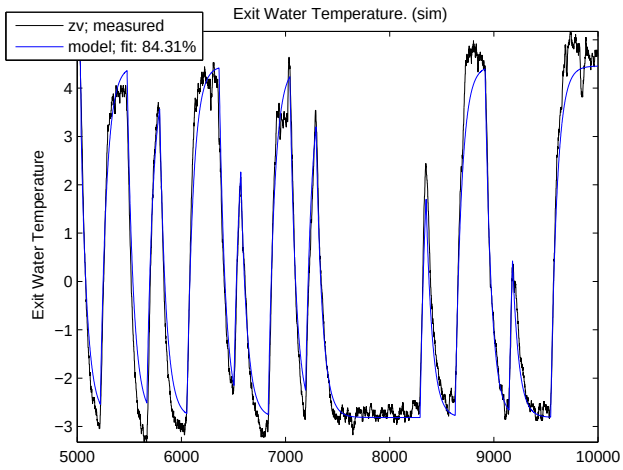
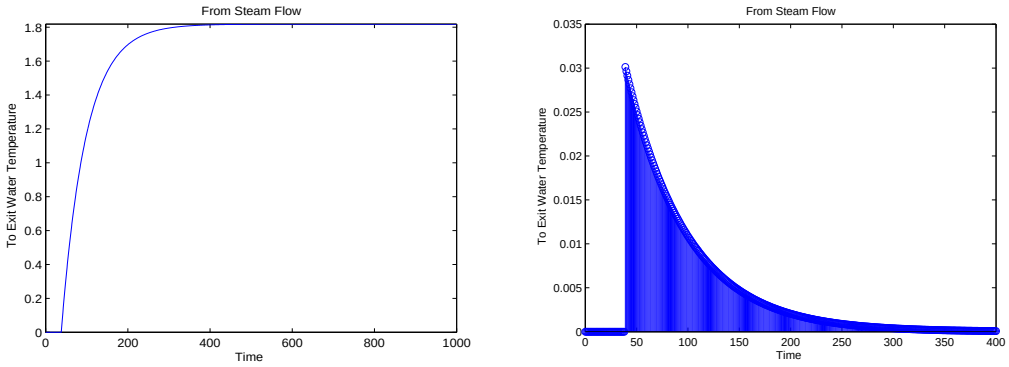


Figure B.8: Validation of the process model against real data



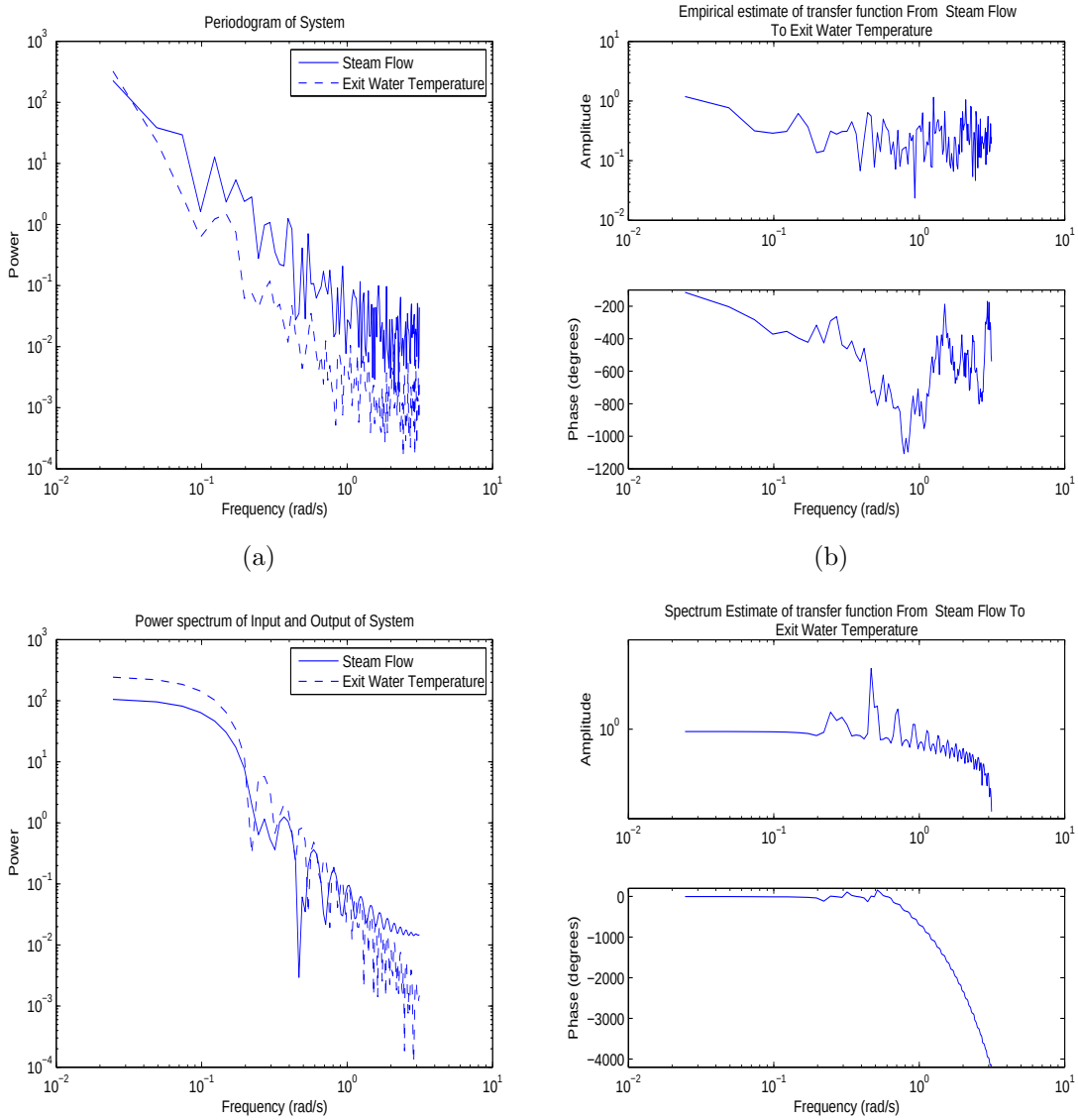
(a) (b)

Figure B.9: Step and impulse responses of identified model

B.5.3 Nomenclature

u : Water velocity
l : Characteristic length
ν : Kinematic viscosity
ρ : Water density
A : Tank cross section area
l_e : Effective length
$\dot{m}$: Mass flowrate
θ : Residence time
T_∞ : Ambient temperature
h_{fs} : Latent heat of condensation
h : Convection heat transfer coefficient
p : Pitch of coil
Re : Reynolds number (pipe)
d_i : Inner piping diameter
μ : Static viscosity
L : Water level
V : Volume
f : Volumetric water flowrate
R : Valve constant
T : Temperature
C_p : Heat capacity
q : Heat flowrate
U : Overall heat transfer coefficient
r : Radius of coil

(B.5)



(a) (b)

(c) (d)

Figure B.10: Periodogram and Empirical estimate of data

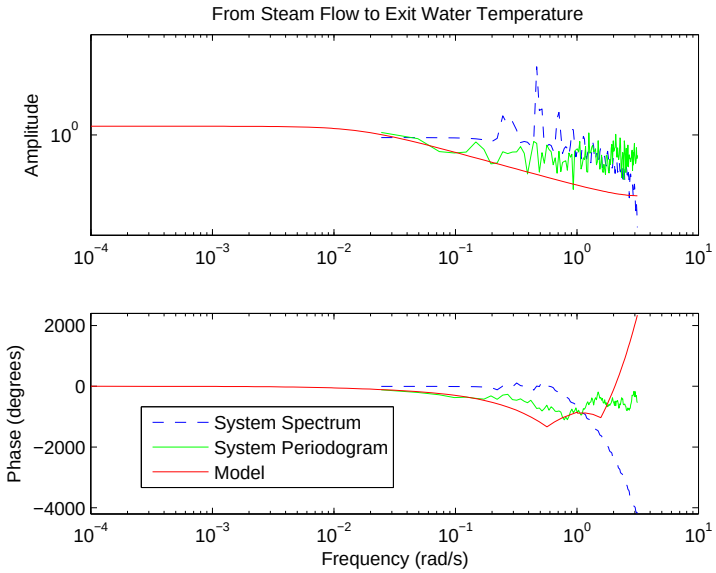


Figure B.11: Comparison of identified model, periodogram and empirical estimate

B.5.4 Theoretical Approximation of Time Delay for CSTH Process

The time delay in CSTH process system was calculated theoretically using fluid mechanic principles and real measurement information (e.g. a tape measure was used).

Physical length measurements

The measurements of process and calculated parameters are summarized in Table [2]. The length of piping was measured from the bottom of the tank to the corresponding temperature sensor.

Table B.2: Physical data for CSTH process

Transparent tank diameter	16 cm
Tank cross section area	$201cm^2$
Pipe OD	2.9 cm
Outside perimeter	9cm
Nominal pipe size	3/4”
Pipe ID	2.74cm
Active fluid area	$5.9cm^2$
Piping length until	
Thermo#1	15cm
Thermo#2	500cm
Thermo#3	705cm
Thermo#4	1155cm

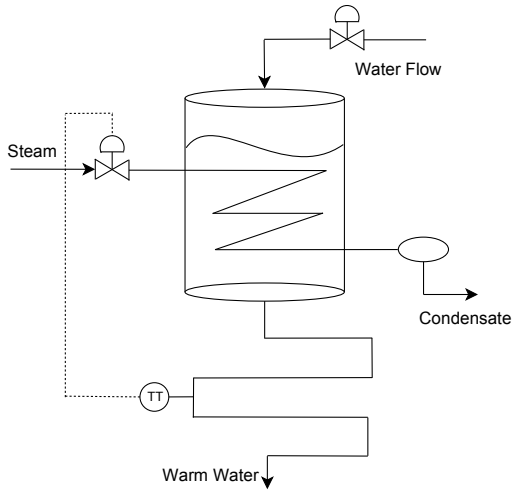


Figure B.12: Schematic of flow streams to / out of CSTH

To calculate the active surface area, wall thickness was taken from the table in [9] for nominal 3/4" copper piping and schedule of M.

Theory

The most famous dimensionless parameter in fluid mechanics is the Reynolds number which is used to determine flow regimes of moving fluids (here water) in carrying lines. By definition:

$$Re_o = \frac{u \cdot l}{\nu} = \frac{\rho \cdot u \cdot d_i}{\mu} \tag{B.6}$$

where u is the fluid local or average velocity, l is the characteristic length (equal to pipe diameter d). ν and ρ are kinematic and static viscosities of water, respectively.

A Reynolds number greater than the critical value (2300 for pipes) means that the flow regime is turbulent and a plug velocity distribution is observed inside the piping.

Consider a pipeline in which a fluid is flowing. The relation between velocity and volumetric flowrate at any single moment can be expressed by:

$$f = \frac{dV}{dt} = \vec{u} \cdot A \tag{B.7}$$

We can extend the formulation for a period of time by defining an average linear velocity over a corresponding time interval and find the desired deadtime:

$$\bar{f} = \frac{\Delta V}{\Delta t} = \bar{u} \cdot A \Rightarrow \Delta t = \frac{\Delta V}{\bar{f}} = \frac{A \cdot l_e}{\bar{f}} = \text{Nominal time delay} \tag{B.8}$$

Results

By plugging numbers into the equation we obtained:

u inside the tank = 0.4 cm/sec (B.9)

u inside the piping = 23.3 cm/sec

Calculated value for Reynolds number = 4870

$\Rightarrow$ flow regime is turbulent

The time constant inside the tank was 50.25 seconds; this is also known as the residence time of a vessel and is used as an important factor in process design. Table B.3 summarizes the lag and delays for this process.

Table B.3: Lag and Delays of CSTH

Residence time of tank	50.25 sec
Delay to Thermo#1	1 sec
Delay to Thermo#2	37.1 sec
Delay to Thermo#3	52.5 sec
Delay to Thermo#4	86.1 sec

During our analysis, thermocouple #2 was used as our base temperature reading. Using the above calculated values we can estimate the time delay until 2nd thermocouple to be about 37.1 seconds. (compare this to 38 seconds obtained from process identification). This confirmed our approximation of time delay and validated the model (less than 1 second error).

Discussion

Several possible reasons for the discrepancy between the theoretical and experimental values of the time delay are listed below:

- The reaction time of the temperature sensor and other carrying instruments is not taken into account in the theoretical approach.
- The inside diameter of the piping may not be calculated accurately.
- There could be a calibration error in the cold water flow indicator
- There is no definite way to calculate the time delay inside the tank because the steam coil is immersed in the water

B.5.5 Level Loop Model

The dynamics between input cold/hot water flow rates and the water level can be described by the following differential equation:

$$f_{in} - f_{out} = \frac{dV(t)}{dt} = A\frac{dL(t)}{dt} \qquad MassBalance \qquad \text{(B.10)}$$
$$f_{out}(t) = \frac{L(t)}{R} \Rightarrow f_{in}(t) - \frac{L(t)}{R} = A\frac{dL(t)}{dt}$$

Input and output volumetric flows, water level and, valve constant are denoted in these equations as f_{in}, f_{out}, L and, R, respectively.

We know from fluid mechanics that the downstream flow is a function of the square root of the level; nevertheless, a linear approximation is used here to obtain an LTI model. And finally we can present the first order model in the Laplace domain:

$$\text{Laplace Transform} \Rightarrow \left\langle \frac{L(s)}{f_{in}(s)} = \frac{R}{\tau s + 1} \right\rangle \tag{B.11}$$
$$K_p = R \text{ and } \tau = A \cdot R$$

The residence time of the process is defined as the amount of time in which the throughput is being treated under process conditions and it can be calculated as:

$$\theta = \frac{V}{f_{out}} = \frac{A \cdot L}{L/R} = A \cdot R = \tau \tag{B.12}$$

The residence time inside the tank becomes equal to the model time constant. This shows that some control data (e.g., from process identification) can be considered as valuable information for process designers or operators.

B.5.6 Heat Flow Loop Model

Steam as MV

In the same fashion, the model from steam flowrate to output water temperature can be analytically written as:

$$\dot{m}_s h_{f,s} + \rho_w f_w C_{p,w}(T_{in} - T_{out}) = \rho_w V C_{p,w} \frac{dT_{out}}{dt} \tag{B.13}$$

By rewriting the equation we get:

$$\dot{m}_s(t) \frac{h_{f,s}}{\rho_w C_{p,w}} + f_w(T_{in} - T_{out}) = V \frac{dT_{out}(t)}{dt} \tag{B.14}$$

Applying the Laplace operator yields the following first order LTI model:

$$\left\langle \frac{T(s)}{\dot{m}_s} = \frac{K_p}{\tau s + 1} \right\rangle \tag{B.15}$$
$$K_p = \frac{h_{f,s}}{\rho_w C_{p,w} f_w} \text{ and } \tau = \frac{V}{f_w} = \theta$$

Hot or cold water flow as MV

$$\rho_w f_w C_{p,w}(T_{in} - T_{out}) = \rho_w V C_{p,w} \frac{dT_{out}}{dt} \tag{B.16}$$
$$f_w(T_{in} - T_{out}) = V \frac{dT_{out}}{dt}$$

Clearly, this differential equation is nonlinear and needs to be linearized around an arbitrary working point. A first order Taylor's series expansion for a multivariable function has been used.

$$f(x,y) = f(x_0,y_0) + \frac{\partial f}{\partial x}|_{x_0,y_0}(x-x_0) + \frac{\partial f}{\partial y}|_{x_0,y_0}(y-y_0) + O(h^2) \tag{B.17}$$

$$f_w T_{in} - f_{w0} T_{out} - f_w T_{out0} = V\frac{dT_{out}}{dt} \tag{B.18}$$

finally

$$\left\langle \frac{T_{out}(s)}{f_w(s)} = \frac{K_p}{\tau s + 1} \right\rangle \tag{B.19}$$
$$K_p = \frac{T_{in} - T_{out0}}{f_{w0}} \text{ and } \tau = \frac{V}{f_{w0}} = \theta$$

It can be deduced from Equation B.20 that the tank temperature will behave as an integrating process if no initial flow is present. In other words the temperature is highly regulated by initial storage of CSTH enthalpy.

Hot or cold water temperature as MV

$$f_w(T_{in} - T_{out}) = V\frac{dT_{out}}{dt} \tag{B.20}$$

$$\left\langle \frac{T_{out}(s)}{T_{in}(s)} = \frac{K_p}{\tau s + 1} \right\rangle \tag{B.21}$$
$$K_p = 1 \text{ and } \tau = \frac{V}{f_w} = \theta$$

Multivariable formulation

The total effect of the variables for a linear system can be shown as a sum of the effects created by each one independently. That is:

$$\begin{bmatrix} T \\ L \end{bmatrix} = \left(\begin{bmatrix} G_{11} & G_{12} & G_{13} \\ G_{21} & G_{22} & G_{23} \end{bmatrix} \right) \times \begin{bmatrix} \dot{m}_s \\ f_w \\ T_w \end{bmatrix} \tag{B.22}$$

G sub-matrices are defined as follows:

$$G_{ij} = \frac{K_{p,ij}}{\tau \cdot s + 1} \text{ and } \tau = \theta = \frac{V}{f_w} = \frac{4.02L}{4.8L/min} = 50.25sec \tag{B.23}$$

$$K_{p,11} = \frac{h_{f,s}}{\rho_w C_{p,w} f_w} = \frac{2260KJ/Kg}{1Kg/L \times 4.18KJ/Kg°C \times 4.8L/min \times 60min/hr}$$
$$= 1.88 \frac{°C}{Kg/hr(steam)}$$

$$K_{p,12} = \frac{(T_{in} - T_{out0})}{f_{w0}} = \frac{(21-30)}{4.8L/min} = -1.8 \frac{°C}{L/min(water)}$$

$$K_{p,22} = R = \frac{20cm}{4.8L/min} = 4.17 \frac{cm}{L/min(water)}$$

$$K_{p,13} = 1 \ , \ K_{p,21} = 0, K_{p,23} = 0$$

B.6 Characterization of steam flow valve

The steam flow controller [FIC-105] regulates the amount of inlet steam by manipulating the corresponding air2open valve. By carrying out an open loop step test, the dynamics from the steam setpoint to the actual valve position could be identified as a FOPTD model with gain of 0.5, time constant of 0.8 and a deadtime of 2 seconds.

Flow loops are normally controlled by a PI controller without derivative action to avoid the effect of high frequency noise. By using the IMC tuning rule, the controller parameters could be set at 0.2(normalized) for K_c and 1 sec for τ_i.

Different values of steam flow could be read as a function of valve position. In that way, the characteristic curve of valve was obtained as shown in Figure B.13. The behavior is similar to a linear valve with a deadband.

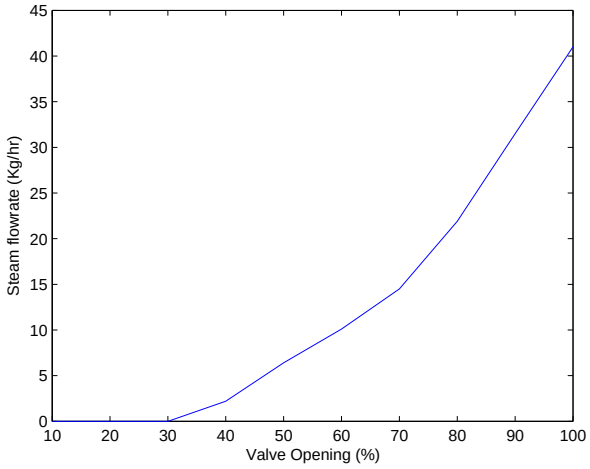


Figure B.13: Characteristic curve: steam valve

B.7 Estimation of Overall Heat-Transfer Coefficient Using Experimental Data

Newton's law describes the rate of convection heat transfer as:

$$q = hA(T - T_\infty) \tag{B.24}$$

The transfer area in this case is the area of coils immersed in water. We first calculate the helix surface area to use in the equations.

B.7.1 Approximation of helix surface area

Consider Figure B.14 [1]. To calculate the surface area, we assume only one complete turn of the helix. It is possible to approximate the length of one turn as the diagonal of a squared triangle which has following dimensions

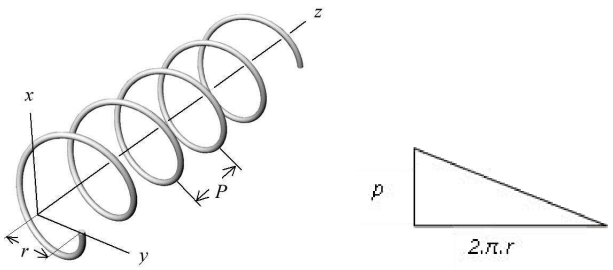


Figure B.14: A helix curve (left) and approximated expansion of one full round(right)

Here p denotes the pitch which is the vertical distance between two rounds and r denotes the radius of the helix. Therefore the length of one turn of the pitch is almost equal to: (assuming p=2cm and helix diameter=10cm)

$$l = \sqrt{p^2 + 4(\pi r)^2} = \sqrt{2^2 + 4(5\pi)^2} = 31.48cm$$

To find the total surface area, one can simply calculate the surface for one turn and multiply it by the number of rounds. Taking n equal to 5.5 complete rounds and the piping diameter as 1.5cm we can calculate the surface area:

$$A = n\pi dl = 5.5 \times \pi \times 1.531.48 = 815.9cm^2$$

Returning to our problem, we know the rate of heat-transfer is equal to the amount of heat dissipated through the steam coils. i.e.,

$$\begin{aligned} q = UA(T - T_\infty) &= \dot{m}_s \times h_{f,s} \\ \Rightarrow U &= \frac{\dot{m}_s \times h_{f,s}}{A(T - T_\infty)} \end{aligned} \tag{B.25}$$

Inserting the numerical data we obtain:

[1] courtesy of Advanced Tubular Technologies, Inc. (http://advancedtubular.com)

$$U = \frac{10 KgSteam/hr \times 2260 \times 10^3 J/KgSteam}{3600 sec/hr \times 815.9 \times 10^{-4} \times (100 - 30)} = 1099 W/m^{2\circ} C \tag{B.26}$$

By referring to the table in [http://www.engineeringtoolbox.com/overall-heat-transfer-coefficients-d_284.html] we find that the overall heat transfer coefficient between steam and water when the transferring media is mild-steel has an approximate value of 1050 W/m2 C. Indeed, the figure provides a good rule of thumb value for that kind of condensation (less than 5% error).

B.8 Conclusions

- All CSTH process dynamics can be represented by first order plus time delay (FOPTD) models.
- Interestingly, time constants for different models were the same and equal to the residence time of the tank. This implies that level and temperature loops undergo the same lag in their responses. However, due to higher deadtime for the temperature loop (especially using downstream thermocouples), it is rationally a harder control goal.
- The transfer functions and their corresponding gains have similar dimensions. Dimensional analysis can be used to guess the final parameters and validate obtained equations.
- Although it seems that steam flow does not affect the water level, but in reality some changes are observed during experiments due to changes in water density. This indicates control loops are interacting with each other; nevertheless, they can be modeled fairly independently (i.e., decoupling holds).

Appendix C

OPC Toolbox in MATLAB

In order to establish a connection to a DCS using a remote computer, it is required to know the language which a specific DCS operating system uses. However, it is easier to use a standard interface called OPC developed for this purpose. In this appendix we discuss details of the OPC toolbox and some of the basic commands associated with it and show examples of its application on the CSTH process.

C.1 What is OPC?

OPC (Object Linking and Embedding or OLE for Process Control) provides open connectivity and standardization to the field of automation and process control. The roles it plays is similar to that of drivers in Microsoft Windows. The problem of interfacing between client and server machines and applications were resolved to a great extent by using this technology.

As per the standards of the OPC foundation: "OPC is a series of standards specifications. The first standard (originally called simply the OPC Specification and now called the Data Access Specification) resulted from the collaboration of a number of leading worldwide automation suppliers working in cooperation with Microsoft. Originally based on Microsoft's OLE COM (component object model) and DCOM (distributed component object model) technologies, the specification defined a standard set of objects, interfaces and methods for use in process control and manufacturing automation applications to facilitate interoperability. The COM/DCOM technologies provided the framework for software products to be developed."

C.2 OPC in MATLAB

MATLAB has dedicated a separate toolbox to OPC technology. The collection of functions and related blocks in Simulink provide the user enough capability to acquire and store the information on the OPC server in a real time manner. This is made feasible by using OPC Data Access Standards. The structure of OPC toolbox is defined as a hierarchy and the concepts of object oriented programming have been implemented.

At one side of the interface we have the server which is represented by a server-ID. The server consists of various items organized by some branches based on their category. Each item could represent a physical field measurement (such as temperature), or a property of a device on the server (such as setpoint of a PID controller) or even a property of the supervisory control and data acquisition (SCADA).

On the other side of interface we have the client workstation which is represented by a client data access point. Each item on the server has a unique identifier on the client machine which is called an item tag. You can query some of the properties associated with server items such as value, quality and the time stamp from MATLAB.

C.2.1 Basic Commands for Connecting and Communicating to the OPC Server

This section demonstrates a simplified way of connecting to an OPC server. First of all you must define an data access point to the server. For this you need to know the name and the address of the host machine.

```
da = opcda('localhost', 'server.name');
```

You may obtain the server.name by invoking the following command:

```
hostInfo = opcserverinfo('localhost')
```

By default the created data access point (da) is offline. Use connect(da); to activate the communication. To retrieve the server item values, a data group should be defined first and then each item can be placed inside the group by calling the corresponding item tag.

```
grp1 = addgrp(da);
itm1 = additm(grp1, 'item.ID.1');
```

If you do not know the exact tag name, you may browse the tree-like name space on the server and find it by following the branches to the desired item leaf.

```
ns = getnamespace(da);
```

To obtain the value of a server item the "read" function can be used. In normal conditions, this function uses OPC cache for the requested information which takes less time but may not be fully up to date. To get instantaneous item value add the 'device' tag.

```
val1 = read(itm1, 'device');
```

The reverse operation is made possible by:

```
write(itm1, value);
```

After you are done with the operation, the cleaning up functions can be used to clear the memory footprint.

```
disconnect(da);
delete(da);
clear da grp1 itm1;
```

In the next sections we cover two example codes of using OPC for CSTH process.

C.3 OPC in Use Example1 . CSTH Start-Up

```
.........................CSTH_Initialization.m .........................

da=opcda('localhost','OPC.DeltaV.1'); connect(da);
da %Shows the properties of the data access point
CSTH=addgroup(da);
steam_setpoint=additem(CSTH,'P4_FIC_105/PID/SP.CV');
level_setpoint=additem(CSTH,'P4_LIC_101/PID/SP.CV');
exitwater_temp=additem(CSTH,'P4_TIC_102/PID/PV.CV');
%Tank-Level Initialization
%We would like to maintain the level inside the tank to the
%nominal value of 20 cm.  We use the controller to take care of
%it (closed-loop).
write(level_setpoint,0.2); pause(60);
write(steam_setpoint,10); pause(600);
.....................................................................
```

C.4 OPC in Use Example2 . CSTH Identification

```
.........................CSTH_OPC_Callback.m ..........................

global i n; % i :  current number of timer execution
ExitWater_T(i)=read(exitwater_temp,'device');
write(steam_setpoint,IDSignal(i)); clc;
disp('************************************************************');
disp(['Process-Identification is in progress :
',num2str(round(i/n*100)),' % Completed']);
disp(['MV is Steam Flow = ',num2str(IDSignal(i)),' Kg/hr']);
disp(['CV is Exit Tank Water Temperature =
',num2str(ExitWater_T(i).Value),' C']);
disp('************************************************************');
i=i+1;
.............................CSTH_Timer.m ............................

global i n;
n=length(IDSignal); %Number of timer execution = length of input
identification signal matrix to CSTH
i=1; % i is current number of timer execution
sampling_time=1; %second
t=timer('TimerFcn','CSTH_OPC_Callback','Period',sampling_time,...
'ExecutionMode','FixedRate','TasksToExecute',n,...
'BusyMode','error','ErrorFcn','CSTH_ID_errorlog.m');
%TimerFunction represents the code which needs to be excuted in
%every timer call.  Period is the time between two executive
%timer callbacks.  TasksToExecute is the number of timer
%callbacks.  ExecutionMode tunes the way that time is taken care
%between two callbacks
start(t);
.....................................................................
```

The preceding codes demonstrate a procedure to identify the temperature loop (MV:steam) of the CSTH process while the level remains unchanged. A timer was used to read and write the data at each identification sampling time.

C.5 References

OPC foundation official website @ www.opcfoundation.org
OPC User's Guide, Mathworks, www.mathworks.com

Appendix D

TUNIX Simulation Package

The TUNIX simulation package [1] is a simple tuning toolbox developed under the MATLAB environment in 2008/09. This graphical user interface (GUI) was designed to facilitate the analysis of the different tuning rules for linear time invariant (LTI) systems. TUNIX is a collection of various tools in the control system toolbox which are organized in a systematic manner to conveniently carry out simulations in an interactive fashion. The toolbox was developed during my masters research work at the University of Alberta. In that time, the need for such a standard and user friendly tool to design control systems based on PID controllers was greatly felt by the author. Figure D.1 shows the main window of the program.

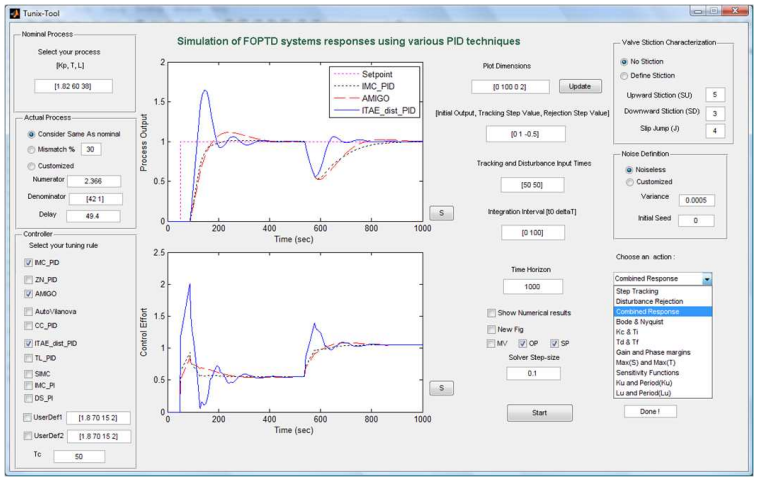


Figure D.1: TUNIX Main Frame

The toolbox can be used to easily compare the performance of new and existing tuning rules with each other. Both qualitative and quantitative tools are included to make the decision for picking a tuning rule easier.
Some of the feature of this toolbox are listed below:

- Support for conventional models

[1] To obtain a copy of this software, you may contact the author by sending an email to amiri1@ualberta.ca

It is possible to define a process in TUNIX using integrating plus time delay (IPTD), first order plus time delay (FOPTD), and second order plus time delay (SOPTD) models. Process gain K_p, time constant τ, deadtime θ, and the damping ratio ξ may need to be specified for each of these models. The dynamics of most of the processes can be well captured using above mentioned models. For more related information on model order reduction refer to 2.3.

- Taking into account nominal / real process models

  While the PID controllers in a system is designed based on the nominal model (normally obtained from the process identification), the real process may not behave in the same way. The phenomenon which is referred as model plant mismatch (MPM) in the literature, can be useful in the robustness analysis of a system.

  The available options are no MPM, percentage of non-structural MPM (i.e., each parameter is biased to the worst case direction, positive shift for process gain, deadtime and negative shift for time constant and damping ratio), and a customized LTI model for the process in case of the structural mismatch.

- Tuning library of well known techniques

  Some of the familiar tuning rules are collected in the software which can be selected in the controller window. It is feasible to input user-defined PID controller name and parameters. You can even import a tuning rule from a m-code to the software. For some of tuning rules, a provision is set to specify the desired closed loop time constant τ_c which is supported by TUNIX. The derivative filter can also be defined using the derivative filter ratio $\frac{\tau_f}{\tau_d}$.

- Valve stiction and saturation considerations

  Valve stiction is a situation in which the manipulating variable (MV) is not changing in the same way as the controller output (OP). In other words, the actuator is not able to follow the controller commands. This prevalent condition is due to the sticky valves in the processes and claimed to be the cause for more than 30% of the oscillations in industrial control loops [8].

  The model which is used in this software is adopted by courtesy of Choudhury et al. [17]. There are three parameters which should be defined to use the model: upward stiction (SU), downward stiction (SD) and slip jump (J). For a homogenous stiction set $SD = SU$. A rule of thumb by Choudhury suggests setting $SU = SD < 1$ for small stiction, $1 < SU = SD < 5$ for moderate, and $SU = SD > 5$ for severe stiction scenarios. For all of the previous rules set $J = SU = SD$. A more comprehensive discussion on the valve stiction topic can be found in 2.4.1.

  The valve saturation can also be modeled using TUNIX. The controller output higher and lower limits need to be specified. Currently integral anti-windup techniques are not supported by the software; however that is a possibility for the future versions.

- Inclusion of measurement noise

  Measurement noise is a prevalent condition in the control loops. It is possible to add a Gaussian distributed random noise in TUNIX by defining the variance and the initial seed of the signal. To ensure the repeatability of the signal use same seed values (i.e., to preserve the same signal for the different simulations in order for a fair comparison).

D.1 How can I do simulations using TUNIX Simulation Package?

The procedure is simple and straightforward. First select your model by pressing the process model button icon, then select the set of the participating tuning rules in the simulation in the controller section. You can add valve stiction, saturation and sensor noise if applicable in the corresponding sections. Select your time or frequency horizon and select an appropriate action from the variety of tools provided by the software and finally start the simulation.

D.2 Analyzing Tools

You can easily test setpoint tracking and regulatory control using TUNIX. It was also designed to have both of the tests (combined) in a single simulation run. Additionally, there are a couple of interpretive tools included in the software. Bode and Nyquist plots can be useful in the evaluation of the controllers and the degree of their stability. You can also plot the sensitivity and complementary sensitivity functions. Process variable (PV) versus controller output variable (OP) plot can be a good tool in investigating the valve stiction.
In addition to the graphical tools, there are a variety of indices for performance and robustness which can be used as quantitative tools for comparing different tuning techniques. Gain margin, phase margin, sensitivity maxima, ITAE, total variation of the controller are some examples. To enable this feature, select options and tick "Show Performance/Robustness indices".

D.3 Real Time Simulations

It is sometimes desirable to analyze the real time effect of changing an specific parameter on the controller design. For example, it would be pleasant to observe the effects of controller parameters on the Nyquist diagram. To accommodate this need, TUNIX was designed in a way to carry-out interactive simulations. You may change the controller parameters or the desired closed loop time constant incrementally by using the designed sliders and watch the resulting effects instantaneously. This can be a nice capability especially for the final fine tuning of the controller and also may be used as an instructional tool in the academia to show the students the effects of the control parameters in a sensible manner.

D.4 Simulation Options and Other Capabilities

You can specify other properties for the simulations. For instance, the time you want the signals to enter the simulation environment or the limits of the axes of the plots. The integration interval is also another user-defined property. To extend the capabilities of the software it is also possible to export the figures in eps and png formats which can be easily used in Latex and MS Office.