2018

A Transformative Process Control Solution

Simone Castanho Nobrega de Almedia Soares
Cleveland State University

A TRANSFORMATIVE PROCESS CONTROL SOLUTION

SIMONE CASTANHO NÓBREGA DE ALMEIDA SOARES

Bachelor of Mechatronic Engineering

University of São Paulo

January 2008

submitted in partial fulfillment of requirements for the degree

MASTER OF SCIENCE IN ENGINEERING

at the

CLEVELAND STATE UNIVERSITY

MAY 2018

We hereby approve this thesis

for

SIMONE CASTANHO NÓBREGA DE ALMEIDA SOARES

Candidate for the Master of Science in Electrical Engineering degree

for the Department of

Electrical Engineering and Computer Science

and

CLEVELAND STATE UNIVERSITY´S

College of Graduate Studies by

____

Dissertation Committee Chairperson, Dr. Zhiqiang Gao

____

Department & Date

____

Committee Member’s, Dr. Lili Dong

____

Department & Date

____

Committee Member’s, Dr. Hanz Richter

____

Department & Date

Student’s Date of Defense: May 3, 2018

ACKNOWLEDGEMENTS

I would like to specially thank my advisor Dr. Zhiqiang Gao for everything he has taught me along the last two years; during this period I have unmeasurably learned from him in several different aspects; first with his inspiring classes, where I have learned how to do research, read/enjoy books, technical papers and how to value the principles from pioneers. In a second stage, the research; he has given me a lot of guidance and I can surely say that I was very lucky to hear his talks about ADRC philosophy, which was a kind of fuel to engage me in it. And at a third stage he has shown me in a couple different projects how academy and industry can work together, which ended up being a huge help for me to find a job. I'm also truly grateful to Dr. Lili Dong and Dr. Hanz Richter for serving on my committee.

I would like to also express a deep thanks to Mr. Gregg Calhoun, who has shared with me lots of his practical experiences. I would like to also thank my lab mates Yu Hu, Dr. Ruiqing Zhang, Dr. Lei Wang, Dr. Han Zhang, Dr. Mario Ramirez-Neria and Dr. Rafal Madonski.

My parents are for sure special part of it, since they were the ones that first gave me the opportunity to study and taught me the importance of it. Without this solid foundation I wouldn't be able to do that. Another important reference for me is my grandfather Tista, who was the first engineer in the family and has always inspired me.

Finally, a big thanks to my husband and daughter who have been my fulltime partners for everything, who have also shared with me these amazing and challenging experience of living abroad. My husband Diego is my source of energy, he is always supporting and encouraging me. My adorable daughter Isadora has helped me with all her lovely smiles, funny faces and kisses.

A TRANSFORMATIVE PROCESS CONTROL SOLUTION

SIMONE CASTANHO NÓBREGA DE ALMEIDA SOARES

ABSTRACT

Knowing that a technology invented almost hundred years ago (PID controller) is still dominating industrial process control, a historical review was done to understand how the control field evolved. Model dependency and high level of mathematics appear as the main reasons that prevent other technologies from penetrating the engineering practice. A relatively novel methodology introduced by J. Han in 1998 called Active Disturbance Rejection Control (ADRC) came with characteristics that matches process control needs and restrictions on model dependency. This study will present a ***transformative*** solution for process control based on that. The control algorithm is designed and discretized for digital implementation in PLC or DSC. The tuning process is explained in a logical and intuitive way based on time and frequency domain characteristics. The idea was to use the language familiar to industry practitioners. To show its applicability, a case study was done for server's temperature control; and the results show energy savings of 30% when compared to PID controllers. This solution is not yet optimal, since it is generally applicable for a wide range of processes, but it aims to be a step further in process control.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

This chapter brings a historical perspective of industrial process control, starting with the invention of steam engine until what is nowadays used in the industries. Then the emerging solutions are analyzed and, based on industry needs, the research problem is formulated on how to apply a novel methodology and create a transformative solution for industrial process control.

1.1. A Historical Perspective of Process Control

The first Industrial Revolution, which took place between 18^{th} and 19^{th} centuries, brought the transition from predominantly agrarian and rural societies to industrial and urban ones. That was guided by the changes in production methods, which until that time were mainly manual using just simple tools, but after the Revolution the development of machine tools and the concept of factories started to rise.

The development of steam engine is one of the most import elements of this revolution. In 1712 Thomas Newcomen developed the first practical steam engine, and in 1784 James Watt improved his work and made steam engine able to power machinery, locomotives and ships [1]. Watt's invention, the ***steam engine governor***, was the most significant ***control*** development till that time; it was the first time a feedback system was applied for process control. The shaft of the steam engine is connected to a flyball mechanism that is itself connected to the throttle of the steam engine. The system is designed in a way that when the speed of the engine increases, the flyball spread apart and a linkage causes the throttle on the steam engine to be more closed; thus the speed of the engine is reduced, which causes the flyball to come back together. When properly designed, the flyball governor maintains a constant speed of the engine, roughly independent of the loading conditions.

In 1919 Trinks authored the book "Governors and the governing of prime movers" [2] it was one of the first times that the principles of ***"governing"*** were studied, but its concept was not made very clear at that time. Maybe because that was not a wide enough field to warrant a separate course in engineering school. Trinks' study of governing comprised of two distinct parts, one being the treatment of the governor as a mechanism, and the other one being the interaction between the governor and the prime mover. He also mentions that in the evolution of the art of governing, many principles have been used, but one by one they were discarded, and at that time only one principle was left: "A force is produced by the quantity to be measured; it is balanced by an external known force such as is derived from springs or weights". The principle in question has been pronounced defectively and faulty, because to cause governor to act, it is necessary a change occurs in the quantity to be kept constant. Even though the concept was not too clear, the Trinks's book was really useful for the foundation theory of automatic control, as mentioned by Eckman [3] in his 1945 book "Principles of industrial process control".

Bennett in his paper about the history of automatic control [4] mentions that in the beginning of 20th century the sales of instruments grew rapidly, which made the feedback controllers widely spread in industries, not only for boiler control of steam generation but also for stabilization of temperature, pressure and flow control in process industry. Most of them were designed without clear understanding of the dynamics of the system to be controlled or the measuring and acting devices used for control. As applications multiplied the engineers got confused, because some controllers that worked well in one application were unsatisfactory for another application, sometimes leading to instability. In 1922, Nicholas Minorsky presented a clear analysis and formulated a three-term control law called ***PID***, the controller is mostly designed empirically, and it does not require a mathematical model of the physical process.

Many books about classic control theory started to be published after the second world war (1945 and later) with some clearer concepts [3, 5, 6, 7, 8, 9, 10]. The impetus of military demands in the Second World War permitted a concentration of a large amount of efforts in the field and boosted the study of automatic control. Examples of subjects studied are: speed governing, temperature control, automatic airplane piloting, automatic machine operation, artillery fire control. In 1945 Eckman [3] came with very illuminated principles of this new field. He stated: "***Automatic control*** can be defined as the maintenance of a balance state in a process by measuring one of the conditions representing the balance and providing an automatic counteraction to any change in the condition. The balance in the process may be a balance of any form of energy, very often heat or pressure". Another concept mentioned by him and other authors is the importance of measuring means, and its ***lags***. Eckman mentions *"Measurement of the variable is the basis for the control action* *since the response of the controller depends upon the detection of changes in the controlled variable"* and *"A change in the measured variable is not instantly detected by the measuring means of any controller. That is, all controllers indicate what the controlled variable was, not what it is. Thus, we say that the controller has a measuring lag"*. That principle tells us that no control problem should be instrumented without due consideration of the lag factors. In 1967 Shinskey came with another interesting book [8], he was a systems engineer at Foxboro Company, and he already understood that to be effective the control system had to be designed to fit the particular needs of the specific process, and the more intelligence the designer could put together the greater are the chances of success, and that intelligence is not necessarily coming from graduate level mathematics, but from a deep understanding of the problem.

The purpose of automatic control mentioned in those books, still very true: ***efficiency and economy***. It eliminates the element of human error and provides a continuous steady response in counteracting changes in the balance of the process. Advances in measuring accuracy and power with high speed, made man unable to compete with these features and it makes automatic control so attractive. Not only that it is possible to reduce the amount of manual labor required but it may also be possible to achieve a higher degree of performance that wouldn't otherwise be possible. Automatic control pays for itself in saving of fuel, processing materials, labor, and in the increased value of the product because of greater output or increased quality.

Nowadays cost reductions and quality improvements are major concerns in process industry, mainly due to the global competition. Safety and stringent environmental regulations are also relevant. Automatic control has the role to take care of that and improve process performance. Many different studies show PID (proportional-integral-derivative) controllers are the heart of process control engineering for the last eight decades, where more than ***95%*** of control loops relies on that [11]. Looks like industry is stuck in a theory that has almost hundred years now. Worse still is that [12] reported that 80% of PID controllers are badly tuned; 30% operates in manual; and 25% use default factory settings, implying that they have not been tuned at all. Historically the main concern of process control engineers was determining the controller parameters to meet the specifications, rather than design of the controller itself. This practice is also known as ***tuning*** and it was first done in 1934 to implement a PD controller for a process modeled by an integrator plus delay model. Subsequently, tuning rules were defined for PI and PID controllers assuming the process was exactly modelled by a first order plus time delay system or pure delay model. Since then more than a thousand different rules were already published [11]. Those are strong evidences that process control needs something new.

1.2. Emerging Solutions

Much theoretical work has been done in the control field since the creation of PID, but is still not commonly applied in industry yet. A recent survey [13] by industry committee of IFAC asked to rank the impact of different advanced control techniques and the result shows their admission that the “crown jewel” of modern control theory didn’t make much dent and ranked at the bottom of the list. On the other hand, PID which was in the list just for calibration purpose, is at the top of the list, 22% ahead of the second place Model Predictive Control. Modern control theory relies on the premise that the dynamics of the physical process to be controlled can be modeled mathematically. Having the model, the next step would be to describe the design objective in another mathematical model or as a cost function to be minimized. Theories from Kalman filters to H_∞ represent huge progress made in the last 50 years. Some techniques came to overcome the model dependence issue, like Robust Control and Adaptive Control, which are more tolerant with the unknowns in the systems. However, the level of mathematics required to understand and apply is beyond the ability of an average engineer; the dependency on mathematical model also limits the appeal of the advanced control techniques to industry practitioners [14].

TABLE I: IFAC survey on industrial impact

TABLE 1 A list of the survey results in order of industry impact as perceived by the committee members.

Rank and Technology	High-Impact Ratings	Low- or No-Impact Ratings
PID control	100%	0%
Model predictive control	78%	9%
System identification	61%	9%
Process data analytics	61%	17%
Soft sensing	52%	22%
Fault detection and identification	50%	18%
Decentralized and/or coordinated control	48%	30%
Intelligent control	35%	30%
Discrete-event systems	23%	32%
Nonlinear control	22%	35%
Adaptive control	17%	43%
Robust control	13%	43%
Hybrid dynamical systems	13%	43%

Model the process dynamic behavior is done based on the laws of chemistry and physics. Some processes can be as simple as first order systems (heat exchange, stirred-tank blending, single tank level control, etc), or second order systems (level control of two tanks in series, etc). However, they can also have complicated higher order dynamics and delays (generally associated with the transportation of the material or energy in the process or caused by processing time or by the accumulation of time-lags in a number of simple dynamic system connected in series). Many studies utilize the approach of approximating higher-order transfer functions models with lower order models, mainly first or second order systems with time delay, that have similar dynamic and steady state conditions [15]. Even similar processes may have different characteristics because of the variations in sensors and actuators, positioning, pipes and tanks sizes, and processors etc. Those are all factors that influences the system modeling. In 1989, Gunter Stein, who was the chief scientist at Honeywell and an adjunct professor of MIT, gave a very famous Bode Lecture, where he describes how the available bandwidth in the closed-loop system can be obtained from analyzing all system's components [16].

Breaking the model dependency, Han came up with a novel control methodology called Active Disturbance Rejection Control (ADRC), it was first introduced in English literature in 2001 [17]. As observer-based technique that practically requires no model, it links the powerful tools of modern control with the simplicity and generality of conventional PID controller. The basic concept of ADRC is based on the assumption, that all external and internal disturbances, including (even strongly nonlinear) dynamics, can be lumped as a total disturbance and effectively estimated and consequently rejected by the application of Extended State Observer (ESO). The practical linear simplification of that nonlinear approach was proposed by Gao [14] and it made ADRC more acceptable for control engineers without sufficient background in mathematics. This linear ADRC still requires adjusting of some tuning parameters.

ADRC provided improvement in control performance in comparison to conventional PID controller. It is disturbance rejection based, it does not rely on model information and it requires relatively simple mathematics background. Those are all characteristics that the industry was looking for. So, why isn't ADRC widely spread in industry yet? Some of the major difficulties are:

- Relatively novel technology;
- No English books have been written on ADRC yet that engineers can easily understand, only academic papers;
- Choice of the appropriate ADRC order: theoretically this is chosen based on the relative degree of the controlled process, however determining that can be difficult in some processes;
- Lack of relatively easy-to-use and reliable (robust) ADRC tuning rules.

Some studies already proposed the rules of thumb for ADRC tuning [18, 19, 20], but the industry needs something more. The objective of this study is to understand how ADRC can be designed or tuned for daily control problems in industry and create a new and better general-purpose controller. PID is still dominating today because of it is simple and user friendly. ADRC was already proved to be both powerful and simple, but it needs to adapt to control engineers and practitioner's language, so that practitioners can start replacing PID controllers with ADRC and then understand the benefits it can bring. Results presented in the Chapter 4 shows that it can bring up to 30% savings. And it is truly a ***transformative*** process control solution.

1.3. Outline

The rest of the thesis is organized as follows. Chapter II brings a quick background on PID and ADRC technologies and explains the design of ADRC. Chapter III presents the logical process of tuning the controller parameters of the proposed general-purpose controller. Servers temperature control case study is presented in Chapter IV, followed by concluding remarks and future work in Chapter V.

CHAPTER II

A GENERAL-PURPOSE CONTROLLER FOR PROCESS CONTROL

This chapter introduces some background on PID and ADRC technologies as they are applied to process control and make visible a parallel between them to show the benefits of the latter. Then, the design of this new general-purpose controller is explained, together with its discrete implementation.

2.1. Background

Most process control books have a chapter on PID controller design and tuning. Marlin [21] mentions that PID has been successful in process control industry since 1940s and remains the most often used algorithm today. The same algorithm is found in petroleum processing, steam generation, polymer processing and many more. This success is a result of many beneficial features of the algorithm that made possible a creation of a general-purpose controller, with generally acceptable performance. However, it is already known that in nearly no case it is an "optimal" controller.

A PID controller continuously calculates an error value $e(t)$ as the difference between a desired setpoint and a measured process variable and applies a correction based on proportional, integral and derivative terms (with K_p, Ki and K_d gains, respectively). The overall control function can be expressed mathematically as:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)d\tau + K_d \frac{de(t)}{dt} \quad (1)$$

In the standard form of the equation, K_i and K_d are respectively replaced by K_p/T_i and K_pT_d; the advantage of this being that T_i and T_d have more understandable physical meaning, as they represent the integration and derivative times.

What made this algorithm so popular is that it is a ***model free*** algorithm. That is, it relies only on the response of the measured process variable, not on the knowledge or a model of the process to be controlled. As mentioned before it can be used in many different process, this ***flexibility*** is achieved through several adjustable parameters, which can be selected to shape the behavior of the closed-loop systems. The process of choosing this value is also called ***tuning***. Being such a ***simple algorithm*** is also an attractive feature for both analog and digital implementations. It is however not as important today, due to the availability of inexpensive digital controller, but this feature was crucial to its initial use. Nowadays, most commercial digital control systems have a pre-configured PID controller block available.

The applicability of PID is so widely known that is possible to find it in the application guidelines of all common control loops. For example, Seborg [15] provides some guidance for different process control:

- Flow and pressure control loops, that are widely spread in process industries, are characterized by fast responses (order of seconds) with essentially no time delay. Disturbances tend to be frequent but small, and most are high frequency noise due to upstream turbulence, valve changes and pump vibration. For flow control, PI controllers are normally used.

- Level control is normally done with P or PI controllers, integral action is often required to remove offsets in the liquid level.
- Temperature loops are difficult to state because of the wide variety of process, and their different time scales. The presence of delays and/or multiple capacitances will usually place a stability limit to the controller gain. PID controllers are commonly employed to provide more rapid responses.

Even with the advantages and features mentioned above, surveys still report that over 30% of industrial control loops still operate in manual mode [12]. The problem is that finding controllers parameters that will work properly at all process operating points is seldom possible. Changes in the process characteristics can happen with time due to a variety of reasons, including changes in the equipment and instrumentation, different operating conditions, different products running through the process, environment disturbances, etc.

2.2. ADRC Background

Differently from the classic control school that believed that the problem of control is a problem of stability and optimality of feedback systems using advanced mathematical tools, ADRC came from a school that believes the reason for the use of automatic control is the presence of disturbance. What is more, the mathematical model of the physical process is not to be trusted, since unexpected disturbances can make the model no longer accurate, thus not reliable for controller synthesis.

The ADRC was originally proposed by Professor Jinqing Han and introduced to English literature in 2001 [17]. It used nonlinear equations, but a more practical way of implementing and tuning with linear equations was created by Gao in 2003 [18] . The central idea is to overcome uncertainties coming from internal dynamics and external disturbances, as they can be estimated and compensated for in real time. This estimation is calculated with plant input-output data in real time, not from a pre-defined mathematical model.

The input-output data brings to the controller information, and that establishes a bridge between empirical error-based controllers (PID) and full model-driven (Modern Control techniques) controllers, getting from each what they have best, improving performance and robustness. As stated by pioneers, the problem of control is a ***problem of lag*** and is possible to say that ADRC is very powerful because the information from input signal brings to the controller ***lead*** compared to simple error driven controllers, without the necessity of complete mathematical model of the controlled system.

This new design is applicable to general n^{th} order, nonlinear, time-varying, single-input and single-output (SISO) or multi-input and multi-output systems (MIMO).

The real word process control problems usually do not have detailed information, like perfect mathematical model, and work in a world where uncertainties definitely exist. In order to overcome all these uncertainties, many different studies have been already applied, utilizing the ADRC in process control applications. For example,in 2007 Zhou [22] applied it to tension and velocity regulations found in web process lines, in 2009 Zheng developed a strategy to overcome decoupling problems in an extrusion temperature control and could reach 30% energy savings [23], in 2014 Madonski applied ADRC structure to a constant water pressure management system, and obtained improved results compared to industrial off-the-shelf PID [25], Li proposed a disturbance rejection based controller for high precision temperature control of a semi-batch emulsion polymerization reactor [26]. In 2016 Sun [27] applied ADRC to a regenerative heater in a 1000MW power plant in China. In 2017 in Latin America Garrán [28] applied ADRC for coupled tanks level control.

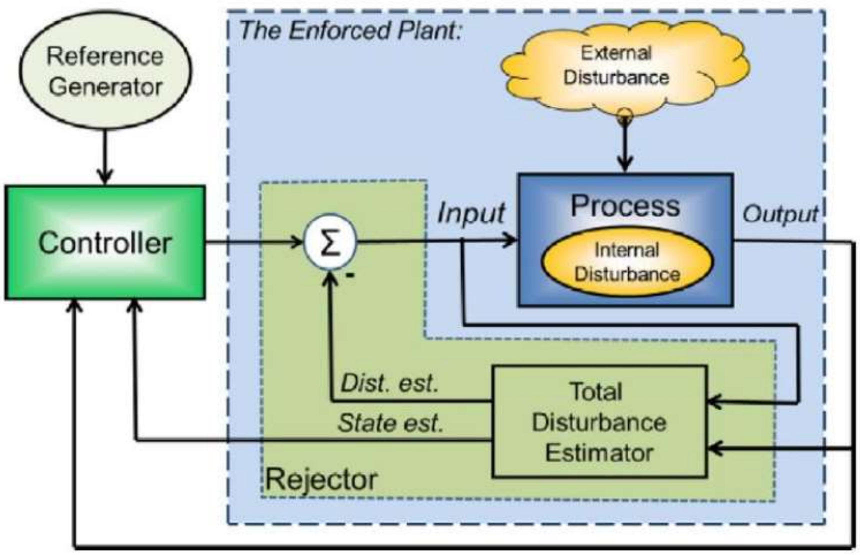

Figure 1: Disturbance rejection principle [24]

2.3. Controller Formulation

The ADRC came with a simple structure, just like PID, but with more information added, which can definitely bring advantages to process industry that today is dominated by the PID controllers. To explain ADRC as a general solution, first order systems were chosen in this work, since as already mentioned before, many processes have first order characteristics or can be approximated with first order plus time-delay system models.

First order systems can have their input-output dynamics represented with an exemplary first order differential equation:

$$\dot{y} = -ay + bu \tag{2}$$

u and y are respectively the system input and output. For example, in a room with temperature controlled with a steam heater, the opening of the steam control valve is the input to the system and the measured temperature is the system output. The transfer function of the considered first order system (2) can be written as:

$$\frac{Y(s)}{U(s)} = \frac{b}{s+a} = \frac{K}{\tau s + 1} \tag{3}$$

where K and τ are system gain and time constants, respectively, and are related to the system response. For example, if a unitary step is given as an input, the response will look like below, where gain is of value the output will reach at steady state, and time constant is the time the process response is still 63.2% complete, since the first order process does not respond instantaneously to a sudden change in the input, at 4 τ the response is closer to complete, 98.2%. In the room with controlled temperature, a step response means changing the steam valve position, for example from 10% to 20% open, and then waiting for the temperature change.

First order systems plus time delay (FOPTD) have similar response, the only difference is that the response is shifted (time delay period). Figure 3 shows an example how this type of models can approximate higher order systems.

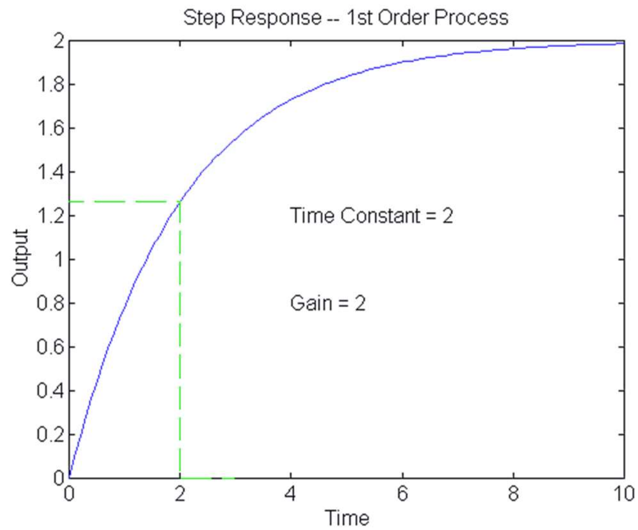

Figure 2: First order processes response to a unitary step.

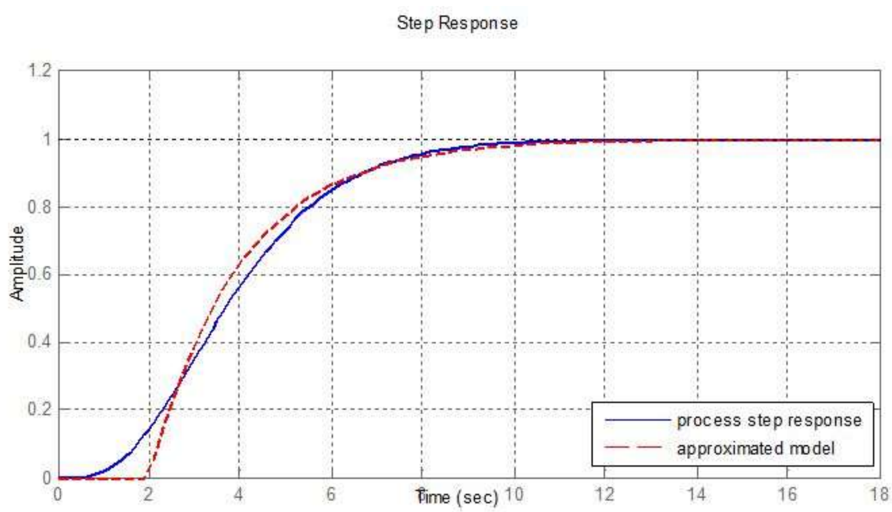

Figure 3: Higher order system and approximated FOPTD system.

In the disturbance rejection-based paradigm, a first order system can be seen as an integral plant perturbed by *f(y,w,t)*, that contains disturbances caused by plants dynamics and external disturbances.

$$\dot{y} = f(y, w, t) + bu \tag{4}$$

The goal of ADRC is to estimate *f*, and cancel it out, reducing the plant to a pure integral canonical form. But why? Reducing the plant to a pure integration creates a simple and straight relation between the control signal and the first derivative (rate of change) of the variable we want to control.

$$\dot{y} \approx u_0 \tag{5}$$

To make that possible, *u* has its control law defined as below, where b_0 is the approximated value of *b* and $\hat{f}$ the estimation of *f*.

$$u = \frac{-\hat{f} + u_0}{b_0} \tag{6}$$

The approximated value for b_0 can be obtained from a simple open loop step test, its response that should have a curve similar to the one seen in Figure 2. From that curve, approximated values of gain and time constants can be obtained and using (3) we have that $b_0=K/\tau$.

To estimate *f,* Han proposed the use of an Extended State Observer (ESO), here is where the modern control observer theory comes in to help. Observers are a mathematical tool that is presented in the state space form that can estimate unavailable state variables. If we consider *y* a state variable and *f* as an extended state, (4) can be rewritten in state space form:

$$\begin{gathered}
\begin{bmatrix} \dot{y} \\ \dot{f} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} y \\ f \end{bmatrix} + \begin{bmatrix} b \\ 0 \end{bmatrix} u + \begin{bmatrix} 0 \\ 1 \end{bmatrix} h \\
y = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} y \\ f \end{bmatrix}
\end{gathered} \tag{7}$$

From that, an observer is designed to calculate in real time the estimated state variables like bellow:

$$\begin{bmatrix} \dot{\hat{y}} \\ \dot{\hat{f}} \end{bmatrix}, = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \hat{y} \\ \hat{f} \end{bmatrix} + \begin{bmatrix} b_0 \\ 0 \end{bmatrix} u + L(y - \hat{y})$$

$$\hat{y} = [1 \quad 0]\hat{x} \tag{8}$$

where $\begin{bmatrix} \hat{y} \\ \hat{f} \end{bmatrix}$ is the vector with estimated variables and $L = \begin{bmatrix} l_1 \\ l_2 \end{bmatrix}$. Vector L identifies the location of the observer eigenvalues and Gao parameterized the observer gain vector to be a function of only one tuning parameter, known as the observer bandwidth, ω_o. The observer gains are determined as:

$$|I\lambda - A + LC| = (s + \omega_o)^2 \tag{9}$$

where $A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$, $C = [1 \quad 0]$ and results in $l_1 = 2\omega_o, l_2 = \omega_o^2$.

For those who are maybe not very comfortable with state space notation, the observer can easily be rewritten in the time-differential equation form:

$$\begin{cases} \dot{\hat{y}} = \hat{f} + b_0 u + 2\omega_o(y - \hat{y}) \\ \dot{\hat{f}} = \omega_o^2(y - \hat{y}) \end{cases} \tag{10}$$

The observer equations give us the real time estimation of f that we need for disturbance rejection, but also give us online estimation of y. This estimation will be used for controller feedback, instead of real measurement feedback, later sections will explain it in more details, but this act like a low-pass filter of the measurement signal, helping the controller against noisy measurements.

Now that the estimation of disturbances is done, it is possible to go back to the controller itself. To control an integral plant, a simple proportional controller is enough for feedback error converge to zero as time goes to infinity.

$$u_0 = k_p(r - \hat{y}) \tag{11}$$

Substituting that in (6), and then in (4) results in:

$$\dot{y} = k_p(r - \hat{y}) \tag{12}$$

and the controller transfer function can be constructed as:

$$\frac{Y}{R} = \frac{k_p}{s+k_p} \tag{13}$$

the pole location can give us the controller bandwidth ω_c:

$$k_p = \omega_c \tag{14}$$

A schematic can help understand how the equations above are interconnected. The green part is a simulation of the first order process that is being controlled, the blue part is the disturbance rejection part, the rectangular block is the ESO itself (eq. (10)) and the other blocks do the rejection part (eq. (6)), the pink part is the proportional controller (eq. (11)) and the white box is the reference signal.

The constants ω_c, ω_o and b_0 are considered the tuning parameters of ADRC. Tuning is not the exact word since they are not just empirical values and have a logical explanation that will be detailed in later sections.

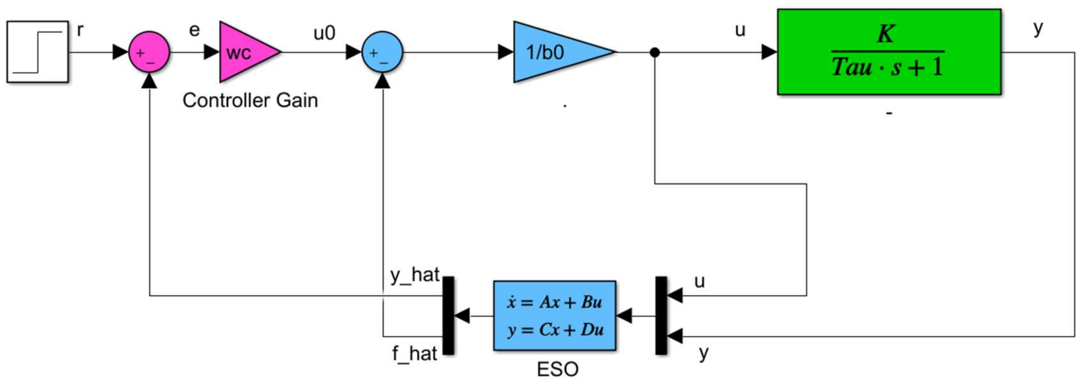

Figure 4: First order ADRC.

Many processes, like temperature loops, contain time-delays which is problematic for the controller design. To solve the problem of time-delay, Zhao [29] proposed a modified ADRC solution. A time-delay block is added to delay the control signal before it goes into the ESO. This way both ESO inputs are synchronized, since the system output that is naturally delayed because of its dynamics.

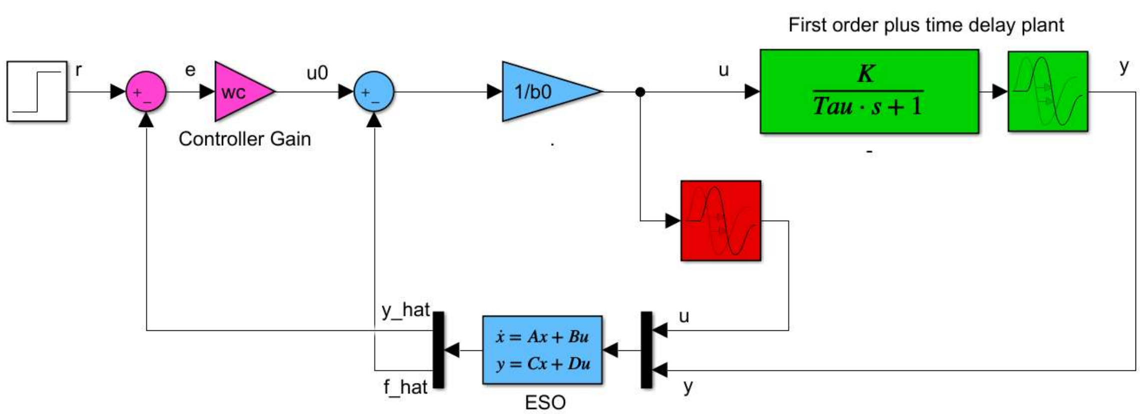

Figure 5: Modified ADRC for first order systems with time-delay.

To do this modification, it is needed to know, or at least have a rough idea, of how much delay the controlled system has. That can also be estimated based on a simple open loop step test.

2.3.1 ADRC Transfer Function: With controller transfer function and the system transfer function, it is possible to derive the loop gain transfer function and then do the frequency analysis of the closed loop system. Frequency analysis is an important tool of classical control theory for controller design that helps engineers that are used to think in terms of controller' bandwidth.

Tian in [30] derived ADRC transfer function in the form of a two-degree-of-freedom closed loop system.

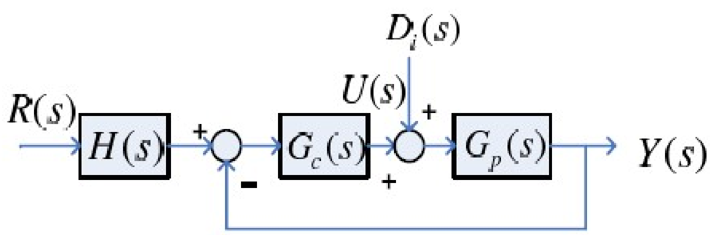

Figure 6: Two degree of freedom ADRC [30]

For a first order ADRC without delay, the transfer function is:

$$G_C = \frac{(2\omega_c\omega_o + \omega_o^2)s + \omega_c\omega_o^2}{b_0 s(s + \omega_c + 2\omega_o)} \tag{15}$$

In Tian's work he showed with loop gain frequency response a remarkable level of consistency in bandwidth and stability margins against significant internal parameter variations in the plant (*a* parameter from eq. (2)). One of the main contributions of this study uses the loop gain frequency response to show how the estimation of b_0 can also help in controller tuning and stability (Section 0).

2.4. Implementation in DCS

Industrial controllers are normally configured inside PLC (Programmable Logic Controller) or DCS (Distributed Control Systems). As mentioned before more than 95% of them are configured with PID logic. To have ADRC in the industrial language the first step would be to do a discrete implementation of it in one of these digital controllers. Discrete or digital means that the algorithm runs in three steps: i) inputs are read, ii) controller logic run and calculate the outputs, and iii) the calculated outputs are sent to the field, these three steps together run in a predetermined, cyclically repeated period of time that is called scan cycle or scan time.

Discrete implementation of ESO was already demonstrated by Miklosovic in 2006 [31]. He compared various discrete implementations and concluded that current discrete formulation is superior to predictive one, reducing the delay associated with sampling times, and also demonstrated that ZOH (zero-order-holder) improves estimation accuracy and stability compared to Euler-based methods, without additional complexity.

Based on that, a discrete formulation of a first order ADRC was done. Equations (6) and (10) in discrete time domain can be written like:

$$u[k] = \frac{-\hat{f}[k] + \omega_c(r[k] - \hat{y}[k])}{b_0} \tag{16}$$

$$\begin{cases} \hat{y}[k] = (1 - l_1)\left(\hat{y}[k-1] + T_s\hat{f}[k-1] + T_s b_0 u[k]\right) \\ \hat{f}[k] = \hat{f}[k-1] + l_2\left(y[k] - \hat{y}[k-1] - T_s\hat{f}[k-1] - T_s b_0 u[k]\right) \end{cases} \tag{17}$$

where the observer gains are defined as $l_1 = 1 - \beta^2$, $l_2 = (1-\beta)^2 \frac{1}{T_s}$ for $\beta = e^{-\omega_o T_s}$

Based on the above theory, ADRC was implemented first in Simulink software and later in *DeltaV* (v. 10.3) a well-known DCS in chemical and petrochemical plants.

The colors of the blocks in the following figures are related with continuous representation of ADRC (Figure 4). As already mentioned before, the ADRC implementation does not require advanced mathematics, which can be notice below.

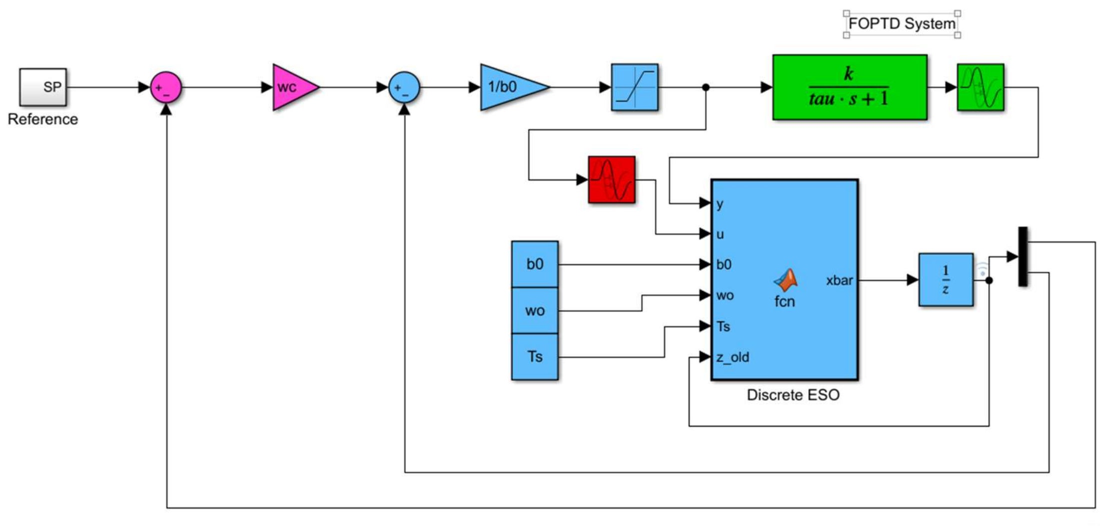

Figure 7: Discrete ADRC in Simulink for first order systems with or without delay

The most complicated part is the ESO logic, but the next figure shows the code implemented inside that block is relatively simple and short (only 11 lines), where the most difficult equation is the exponential, while term all the rest are just simple additions and multiplications.

```matlab
function xbar = fcn(y,u,b0,wo,Ts,z_old)

xhat = zeros(2,1);
xbar = zeros(2,1);

beta=exp(-wo*Ts);

lc1=1-beta*beta;
lc2=(1-beta)*(1-beta)/Ts;

xhat(1)=z_old(1)+Ts*(z_old(2)+b0*u);
xhat(2)=z_old(2);

e = y-xhat(1);
xbar(1)=xhat(1)+lc1*e;
xbar(2)=xhat(2)+lc2*e;
```

Figure 8: Discrete ESO function block

The *DeltaV* implementation is very similar to Simulink implementation, since both use function blocks. The difference is that *DeltaV* requires two different implementations for systems with and without delay. What happens is that the delay block cannot have its delay time set to zero, like was done in Simulink for cases without time delay. Again, the same colors were used to make readers understanding easier. The green box shows process simulation, unfortunately the license available in the lab was just a *DeltaV* Simulation license and a real process cannot be connected to the software. The orange box was added to include disturbance simulation. The pink box shows the proportional controller which only a subtraction and a multiplication blocks were needed. The blue box contains the ESO algorithm, a subtraction, a multiplication, a division and limit function block were used, and the most complicated part is an expression implemented inside a calculation block, that has the same function of the expression done in Simulink.

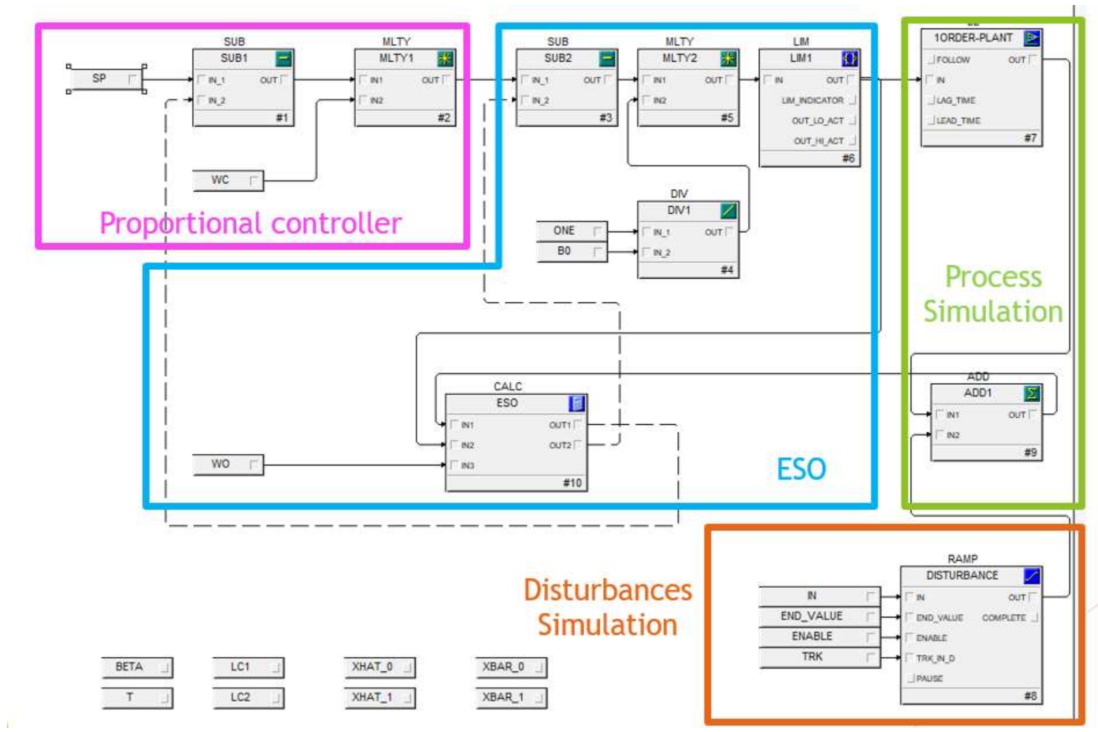

Figure 9: Discrete implementation of ADRC in DeltaV for systems without delay

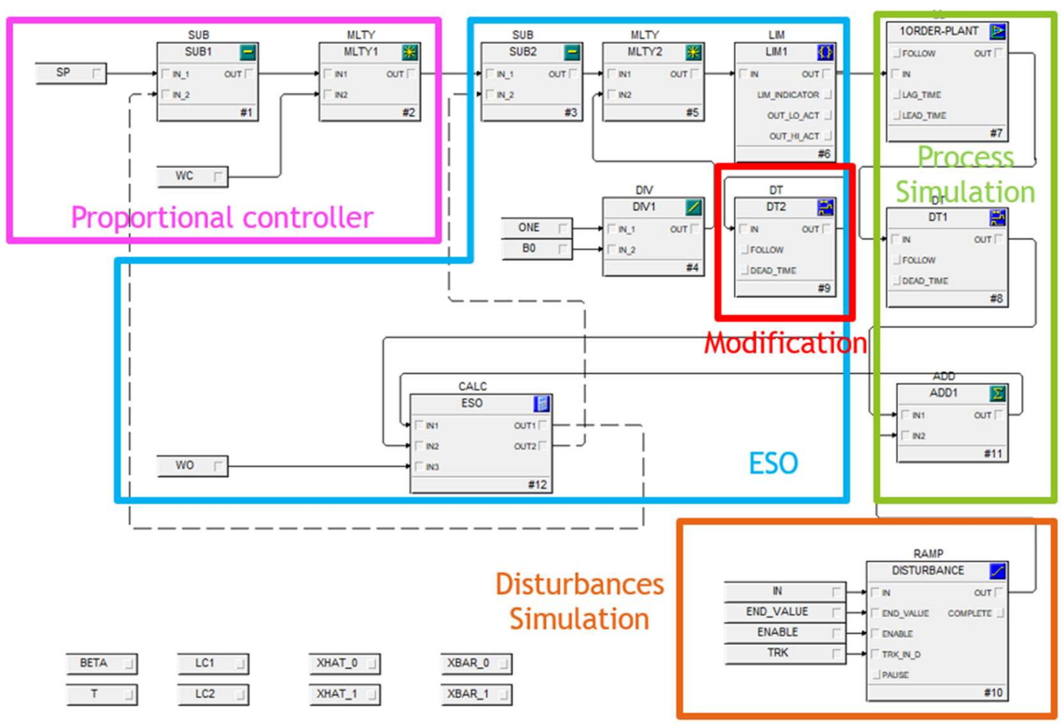

Figure 10: Discrete implementation of ADRC in DeltaV for systems with delay

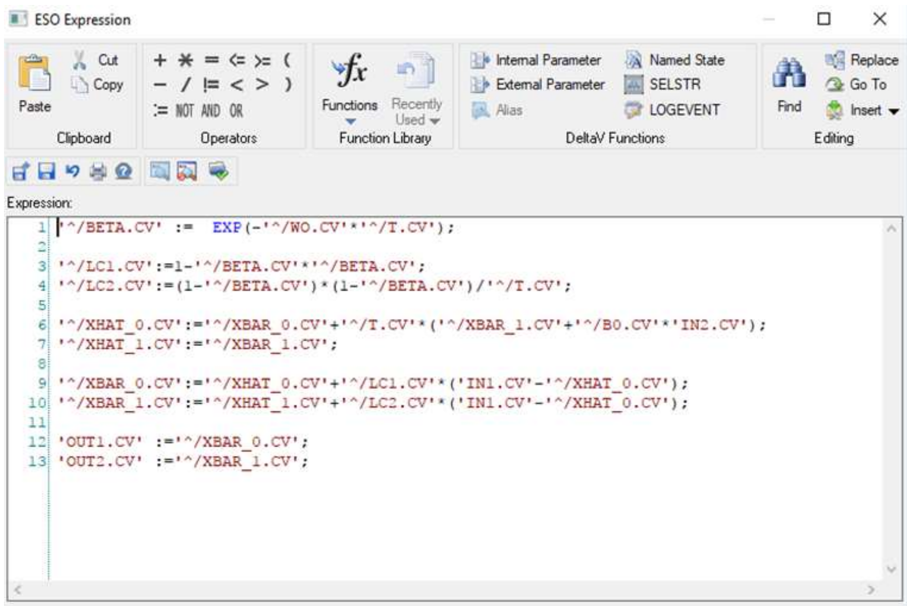

Figure 11: ESO function block in *DeltaV*

The simulation of this new general-purpose controller ADRC based algorithm was done for both cases, without and with time delay and the responses were compared to the response of built-in *DeltaV* PID block. Figures below show the algorithms responses (green for ADRC, red for PID), setpoint (blue) and control signals (pink for ADRC and yellow for PID). In both cases the lead advantage of ADRC can be seen in the conducted setpoint tracking and disturbance rejection tests.

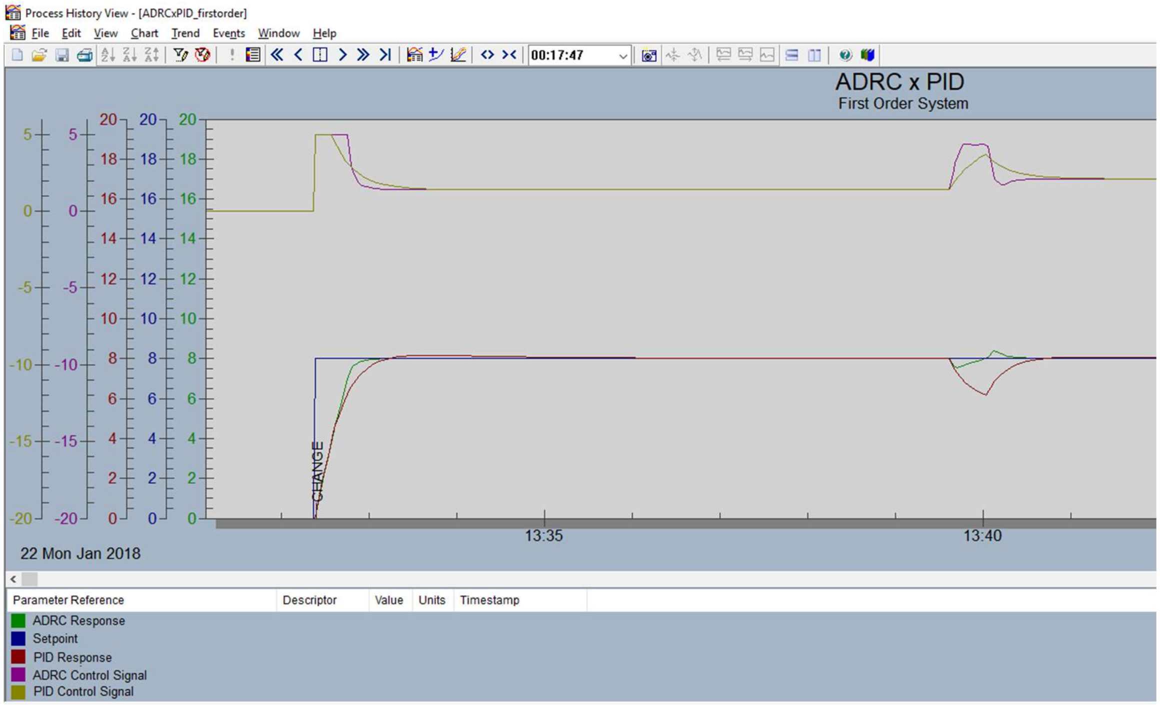

Figure 12: System without delay simulation (comparison to PID controller)

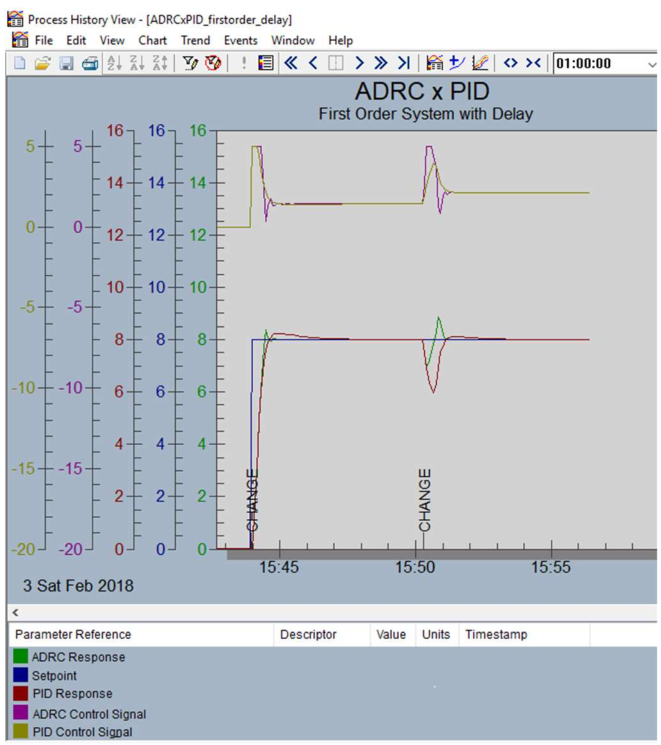

Figure 13: System with delay simulation (comparison to PID controller)

CHAPTER III

CONTROLLER CHARACTERISTICS AND TUNING

In contrary to the popular PID-like solutions in industry, which are mostly empirically tuned, this chapter presents the systematic method of tuning the controller parameters of the proposed ADRC solution. This tuning process can be applied to any system with approximated response of first order plant with or without delay. In our lab at CSU, there is simple level control experiment that can be used as an example. Since this chapter addresses the issues of tuning for practitioners, some ***tips*** are provided to help them applying ADRC to different processes.

3.1. Testbed

The testbed used for the experiment is from controls laboratory of CSU where a simple level control can be simulated. The figure bellow shows a scheme of the testbed. The focus of this study are first order systems with or without delay, and this process can simulate first order process. The plant consists of one column, one variable speed pump to control the inlet flow and one valve on the bottom of the tank, which is constantly open. A second pump was used as a disturbance to the system, removing water at a certain period.

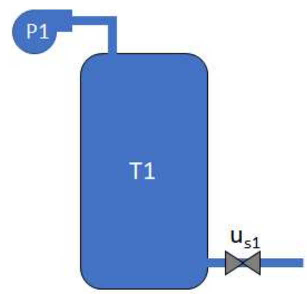

Figure 14: Level control testbed scheme

Based on the scheme above it is possible to determine a mathematical model for simulation. The following flow equation can be written:

$$A\frac{dh_1}{dt} = Q_1 - Q_{out} \tag{18}$$

where A is the section area of the tanks and h_1 is the height of water in the tank. The flows can be rewritten with Bernoulli's equation as bellow:

$$Q_{out} = u_s\sqrt{2gh_1} \tag{19}$$

$$Q_1 = c_1u_1 \tag{20}$$

where $u_s = S_n\alpha$, S_n is the section area of the valve, and α is a multiplier of proportionality, with $\alpha \in [0,1]$, where 0 means completely shut off and 1 means completely open. Term u_1 is the command given to the pump.

Substituting that in (18):

$$\dot{h_1} = \frac{c_1u_1}{A} - \frac{\sqrt{2g}}{A}u_s\sqrt{h_1} \tag{21}$$

It was assumed that the valve in the bottom is continuously open and no leaking occurs in the tank, and no restrictions are placed on the valve; that all means $u_s = S_n$.

Equation (21) is nonlinear, a linearization is done bellow:

$$\Delta\dot{h}_1 = \frac{c_1\Delta u_1}{A} - \frac{\sqrt{2g}}{2A} S_n \frac{1}{\sqrt{|h_{10}|}} \Delta h_1 \tag{22}$$

where h_{10} is the linearization point.

TABLE II: Constant values defined based on laboratory testbed

g	386.088 in/s^2	***c$_1$***	0.5673 V in^3/s
A	7.94 in^2	***S$_n$***	0.025 in^2
h$_{10}$	12 in	-	-

Substituting the constant values in (22), the following transfer function will be utilized for simulation of the first order level control system:

$$\begin{aligned} &\frac{\Delta h_1}{\Delta u_1} = \frac{0.07144}{(s+1.2629x10^{-2})} = \frac{b}{(s+a)} \quad \text{or} \\ &\frac{\Delta h_1}{\Delta u_1} = \frac{5.658}{79.19s+1} = \frac{K}{\tau s+1} \end{aligned} \tag{23}$$

The controller goal is to fill the tank 8 inches height and keep that for all the simulation time. A disturbance is added to the system at 150s when a second pump start removing water from the tank at constant flow (0.16 in/s) for 25 seconds.

The discrete ADRC was tested with simulated process in Simulink and a real experiment connected also to Simulink via a real time data acquisition board. To start time domain analysis, the sampling time of discrete controller was set to 0.2 seconds, the smallest sampling time available in *DeltaV* software. The smaller the sampling time is, the closer the system is to continuous time and less stability problems, but that also means more controller load. In the frequency analysis section, the sampling time will be changed and analyzed.

3.2. Time Domain Characteristics

To start the controller tuning, the time domain characteristics were analyzed. In this section, setpoint profile and controller bandwidth selection are going to be discussed.

3.2.1 How setpoint profile can impact the system response: Going back to Han's study [17], he proposed a new controller that would track the reference input in shortest time possible without overshoot, by including a profile generator. Putting that in simple words, it means that the reference should have a profile that the physical system is able to respond. Process control industry is normally only worried about how to tune PID controller gains, nothing else, and setpoints are almost always changed in steps. But it is important to be careful with what is asked to the actuator. Instant changes like steps means infinite control effort needed, and no actuator is able to do that in real systems.

The graphic bellow shows the system response for a step change setpoint and for a smooth profile setpoint. The smooth profile was defined in a way that the level should reach its setpoint in less than 120 seconds.

This is a first order example, so no overshoot is observed even with step setpoint, but it is possible to see the control signal saturated, for the step setpoint case. Saturation means no control, so it is desirable that the control signal never saturates, that is also important because "not saturated" means the controller still has some room for possible disturbances. Defining a smoot profile can also mean energy savings, and better usage of your equipment.

It was mentioned before that the setpoint does not normally change in process control, but there are exceptions. For those cases where setpoints do frequently change, what is presented here should be taken into account. This is the **first tip** for practitioners: think about what you are asking your controller to do: regulation or setpoint tracking.

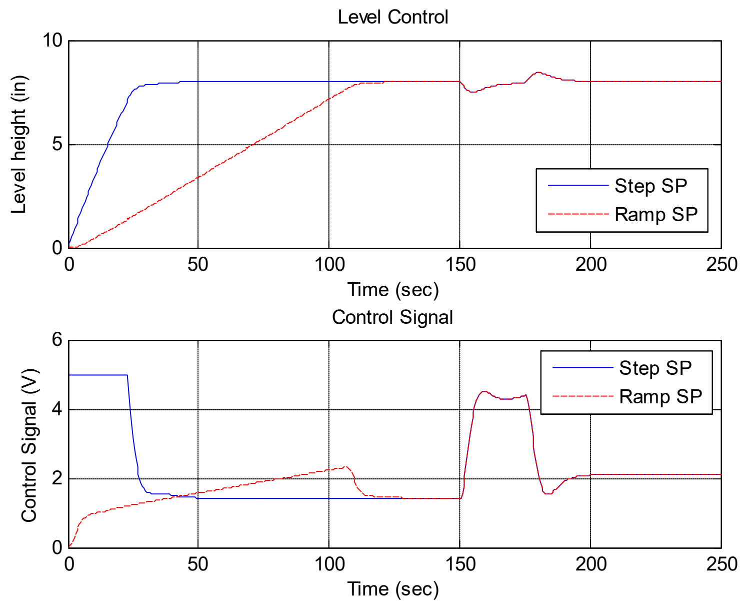

Figure 15: System response and control signal for step and ramp setpoints

3.2.2. Controller bandwidth selection: After the reference profile is defined, it is now time to tune controller gains: b_0, ω_c and ω_o. The ADRC requires minimum information about the system to be controlled: system order and b_0. In this study, first order systems are used for the formulation of this new general-purpose controller, so system order is already defined. And from mathematical model of the physical process b_0 was calculated and equals 0.07. An approximated value for b_0 can also be found without system model from open loop step tests as mentioned in section 2.3. This is the **second tip** for practitioners.

Based on Gao's study [18], tuning parameters should start with controller gain (ω_c) and observer gain (ω_o) defined with equal values. Controller gain (ω_c) is also called controller bandwidth, that is related to the speed of the system response. Since it was defined that the level should raise to 8 inches in less than 2 min, we can use it to find the required closed loop bandwidth ($\overline{\omega}_c$) using the conventional root locus method. The settling time of 120 seconds corresponds to 4 τ and $\overline{\omega}_c = {}^{1}/_{\tau}$, and, as communicated by Gao, a good starting point would be $\omega_c = \omega_o = 5{\sim}10\overline{\omega}_c$. This is considered the **tip number 3** for practitioners, below is shown practical results of it.

Taking in to consideration the level control case, it would mean a starting point for controller and observer gain of $0.16{\sim}0.3$ rad/s. The graphic bellow shows the system response in simulation for different values of ω_c and ω_o, starting from 0.1 rad/s, a little bellow the above mentioned required bandwidth, until 0.7 rad/s.

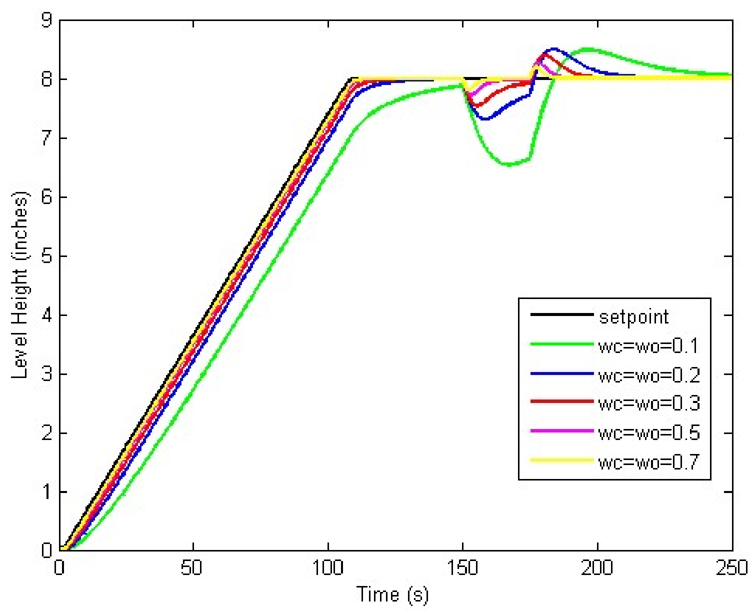

Figure 16: Simulation system time response for different ω_c and ω_o

It is easy to observe that the response is much better and acceptable for gains above 0.2 rad/s. The table below prove with integral of absolute error (IAE) calculations, that the method utilized is a good way to find the initial value of controller bandwidth, since bellow that value the error increases more rapidly and after that point the error decreases more slowly.

TABLE III: Time response IAE calculation

$\omega_c = \omega_o$ (rad/s)	$e = r - y$ IAE	$e = y - \hat{y}$ IAE	$e = f - \hat{f}$ IAE
0.1	782.4	141.5	33.0
0.2	332.4	41.9	19.4
0.3	194.0	21.7	14.6
0.5	97.7	12.9	9.1
0.7	62.3	10.4	6.6

From figure 16 it is possible to observe that as we increase the gain the better the response gets, so what would be the upper limit? The answer was already given by Gao in [18], the noise would be the limiting factor. Since we have the testbed, instead of including white noise in simulation it was used the real experiment. Figures 17 and 18 shows the graphics for time response and control signal in experiment. It is possible to observe that with the increase of gains the control signal is getting more and more noisy. When ω_c= ω_o=0.5, the noise presented in the control signal was not acceptable, and the time response improvement was small compared to the loss obtained in the control signal. More details about noise will be discussed in the frequency analysis section.

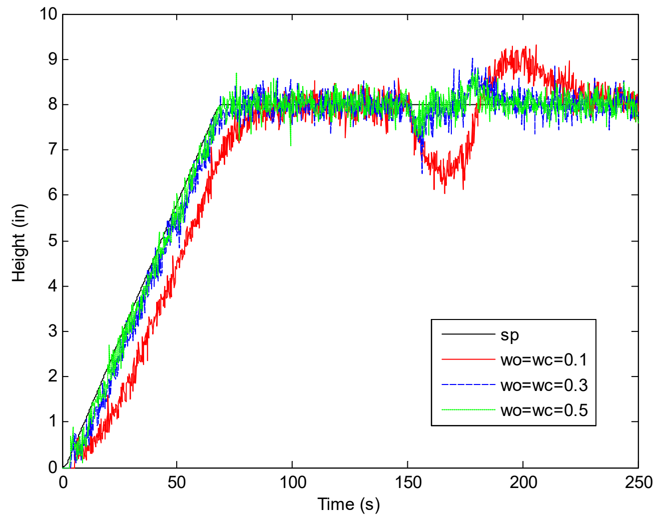

Figure 17: Experiment time response for different ω_c and ω_o

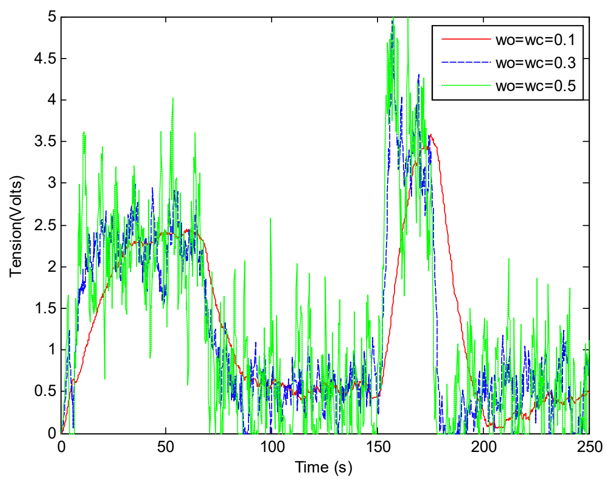

Figure 18: Experiment control signal for different ω_c and ω_o

3.3. Frequency Domain Characteristics

To start a little deeper analysis, the frequency domain is studied. This section brings the main contribution of this study. With physical process model (section 3.1), ADRC transfer function (section 0) and sampling transfer function (T_s is the defined sampling time) it is possible to obtain the loop gain transfer function by multiplying all of them. The next figure shows the Bode plot of loop gain transfer function for different controller and observer bandwidth.

$$Loop\ gain\ tf = \frac{b}{(s+a)} \frac{(2\omega_c\omega_o + \omega_o^2)s + \omega_c\omega_o^2}{b_0 s(s+\omega_c+2\omega_o)} e^{-\frac{T_s}{2}s} \tag{24}$$

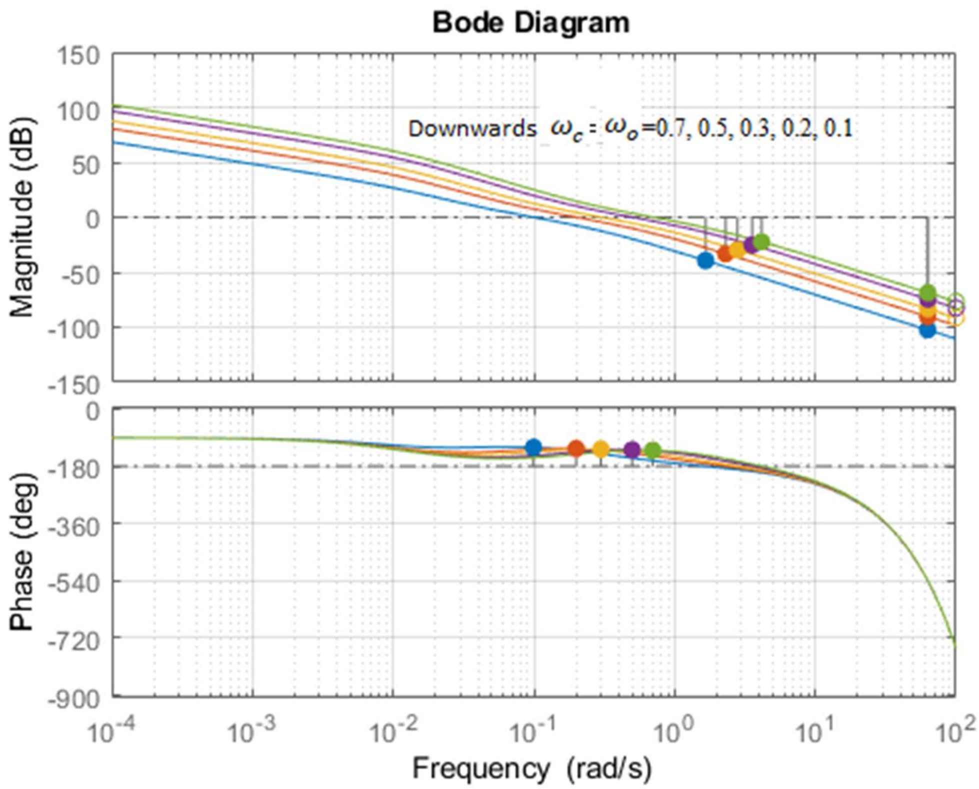

Figure 19: Loop gain bode plot for different ω_c and ω_o

TABLE IV: Phase margin under different ω_c, ω_o

$\omega_c = \omega_o$ (rad/s)	Phase Margin (Degrees)
0.1	59.8
0.2	55.6
0.3	53.8
0.5	51.7
0.7	50.2

The study made in time domain showed us the increase in controller and observer gains would give us better responses, but noise was a limitation factor. The frequency domain analysis shows us another limitation for gains' increase, which is the system stability. From the graph above it is possible to check that for bigger gains lower phase margin is obtained. Phase margin is the distance in degrees your system is from the instability (-180°), so the bigger the phase margin is the more stable your system is going to be.

3.3.1. Sampling time effect: Frequency domain can also show us the impact of sampling time increasing. For fixed controller gains the loop gain bode plot is done for different sampling time.

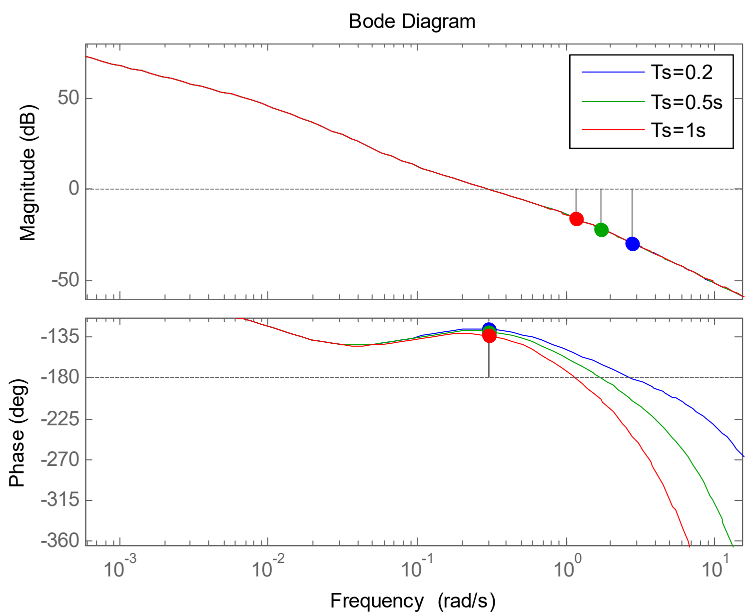

Figure 20: Loop gain bode plot under different sampling time

Sampling time only affects the phase graph and the bigger it is the faster system cross the unstable limit (-180 degrees). The table V shows phase margin for different controller gains and sampling times.

There is always a tradeoff that practitioners have to consider, reducing sampling time can give more stability to the closed loop system but it also increases the controller load. The minimum sampling time needed is related to the speed of your system, if it runs at lower frequencies a bigger sampling time can be chosen, but if runs at high frequencies you would need small sampling time, for example flow loops as already mentioned before are fast loops (in the order of few seconds) so a smaller sampling time is needed, 1 second or less, but temperature loops are slow (in the order of minutes or hours) so bigger sampling time can be tolerated, like few seconds. This is **tip number 4,** choose sampling time according to your system needs.

TABLE V: Phase margin under different ω_c, ω_o and T_s

$\omega_c = \omega_o$ (rad/s)	T_s=0.2s	T_s=0.5s	T_s=1s
0.1	59.8	59	57.5
0.2	55.6	53.9	51.0
0.3	53.8	51.2	47.0
0.5	51.7	47.4	40.3
0.7	50.2	44.1	34.1

3.3.2. Term b_0 as a tuning parameter: Until now the value of b_0 was kept constant with the value calculated from the physical system model, but it is known that it is not easy to have the system modeled for every case and defining b_0 sometimes can be tough.

In (24), we have b and b_0, where the first one is the real value from your system and the second one is the estimated value chosen for controller implementation. This way the ratio b/b_0 can tell relation between both. The closer this ratio is to 1, the better the estimation is to a real value. The bode plot for different relations of b/b_0 were done and an interesting point was observed for the phase margin. The table below shows the results obtained:

TABLE VI: Phase margin under different ω_c, ω_o and b/b_0

$\frac{b}{b_0}$ / $\omega_c = \omega_o$	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**1**	**1.1**	**1.3**	**1.5**	**1.8**	**2**
0.1	61.0	61.2	61.2	60.9	60.4	59.8	58.4	57.6	56.0	53.6	52.1
0.2	54.5	55.5	55.9	56.1	55.9	55.6	55.15	54.0	52.6	50.4	49.0
0.3	52.1	53.3	53.9	54.1	54.1	53.8	53.4	52.3	50.9	48.7	47.3
0.5	49.8	51.1	51.8	52.0	52.0	51.7	51.3	50.1	48.6	46.3	44.7
0.7	48.4	49.7	50.3	50.6	50.5	50.2	49.7	48.3	46.7	44.1	42.5

Evaluating the phase margin table above, made us think that for fixed controller gains the phase margin looks like to have a maximum value at some b/b_0 ratio. Then it was tried to find the mathematical equation that would generate the graph phase margin *vs* b/b_0. But the phase itself is not dependent of b/b_0 ratio, as can be seen in the plot below. The phase margin graph is exactly the same for different b/b_0 ratios. What changes is the crossover frequency, as we increase the ratio (in other words, decrease b0) the crossover frequency is bigger.

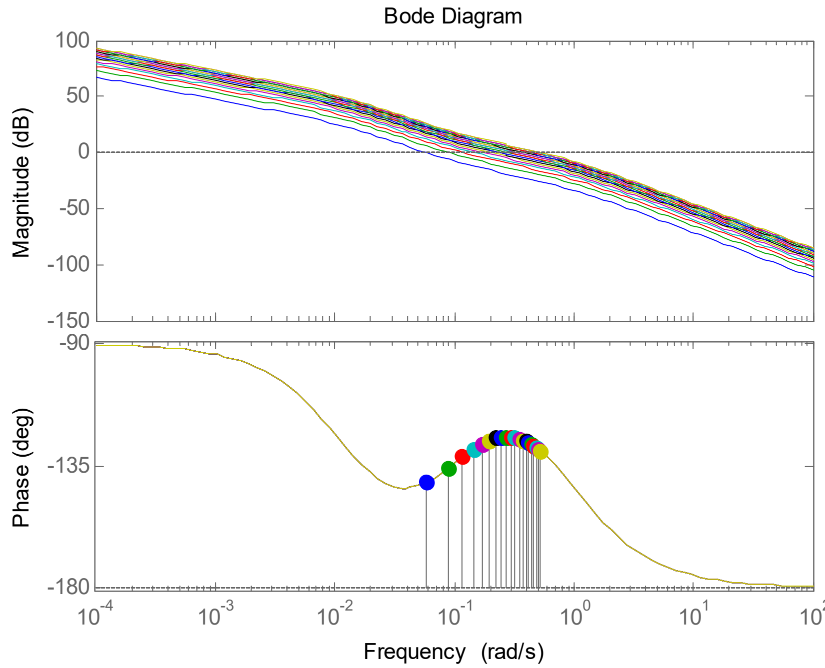

Figure 21: Loop gain bode plot for different *b/b₀*

Finding an analytical equation for that was tough, so it was generated a program in Matlab to calculate phase margin for many different values for b/b_0 ratio and plot that. To start with the simpler case, it was considered the continuous case, no sampling included, and the figure 22 was obtained.

It is possible to observe that as controller and observer gains (ω_c and ω_o) are increased the maximum point approximates to the ratio =1. It is what we would normally think that for a best controller design the more accurate estimation you have the better. But the opposite was observed for discrete cases, when the sampling transfer function (T_s=0.2 seconds) was included, figure 23. An even bigger difference can be seen when the sampling time was increased to 1 second, figure 24.

Since most of the industry controllers are digital nowadays, it is possible to say that the first approach (Figure 22), where continuous the case was considered, is true in theory but is not applicable in real world. The conclusion of this is that with b/b_0 ratio less than 1 we could obtain better stability results for the controller. And it can be used as a tuning parameter when stability is a problem.

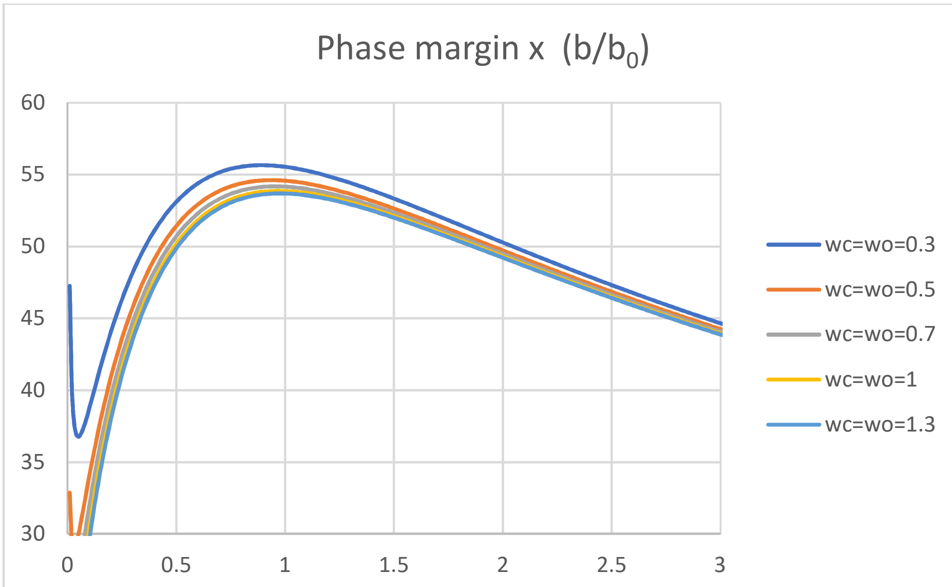

Figure 22: Phase margin for different b/b_0 – continuous time

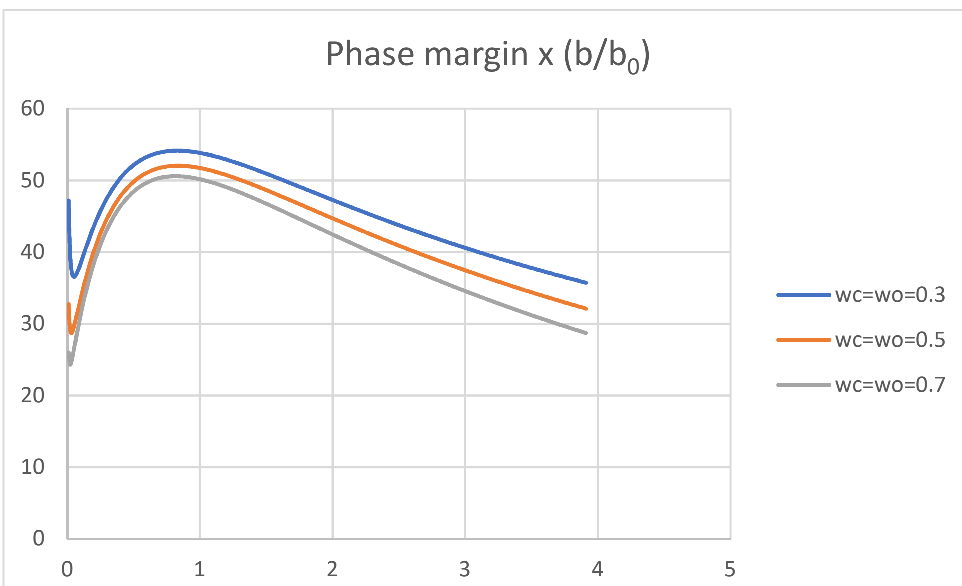

Figure 23: Phase margin for different b/b_0 – discrete time (T_s=0.2s)

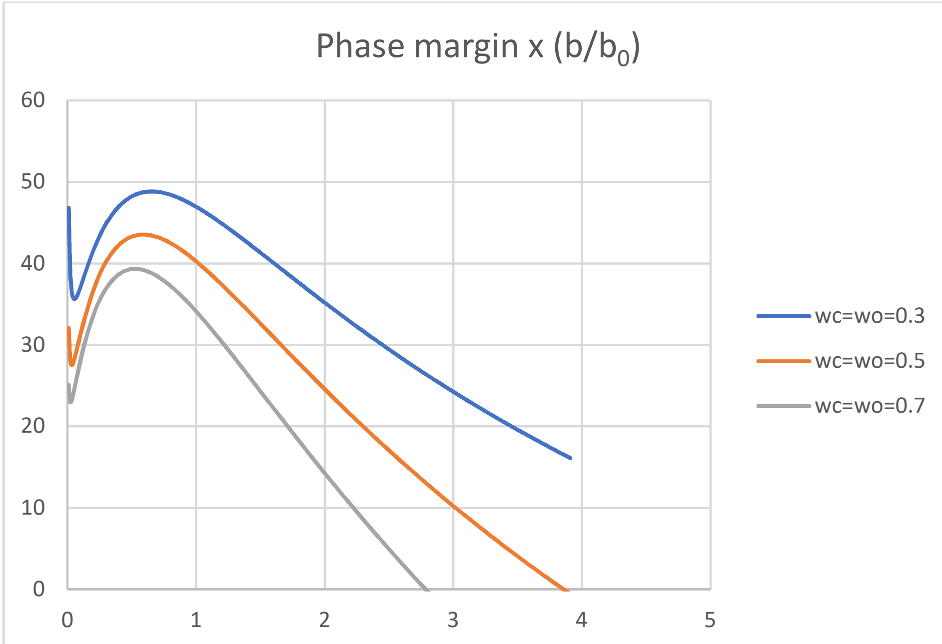

Figure 24: Phase margin for different b/b_0 – discrete time ($T_s=1s$)

That brought the curiosity of what would happen if an even bigger sampling or delays where included to the system. Considering the delay is measured in percentage of time constant of the plant, the graph bellow shows the results for delays =1%, 2.5%, 3.75% and 5% of τ.

From it is possible to observe that the curve for time delay equals to 5% of τ, the maximum point does not exist anymore, and the phase margin is reduced drastically. That shows b_0 is also an important tuning parameter, it can help stability in systems with delay. The table VII shows some important values from graphs.

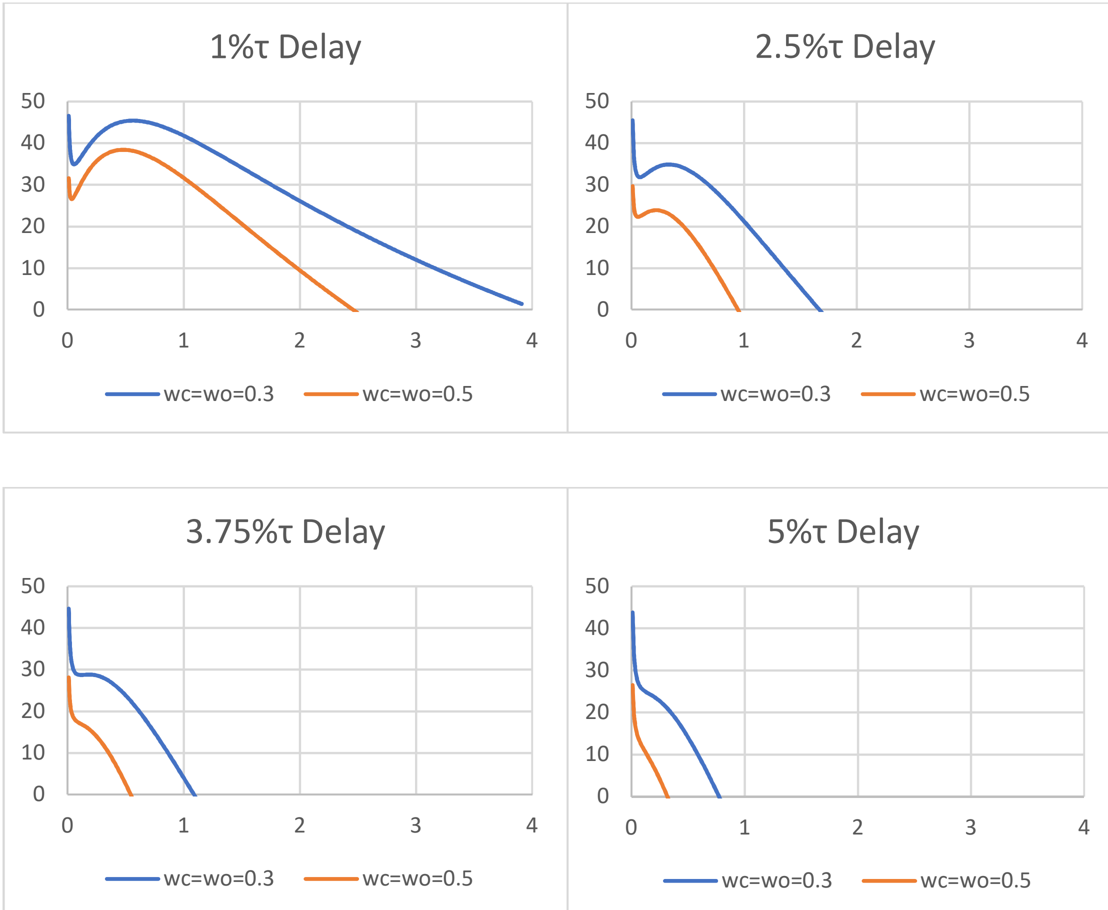

Figure 25: Phase margin for different b/b_0 and different time delays

TABLE VII: Max. phase margin under different time delays ($\omega_c=\omega_o=0.3\ rad/s$)

	Curve presented maximum point (Yes/No)	Phase margin at Maximum point	b/b_0 at Maximum point
Continuous time	Y	55.66	0.89
T_s**=0.1s**	Y	54.14	0.83
T_s**=1s**	Y	48.83	0.65
T_d**=1% τ**	Y	45.42	0.56
T_d**=2.5% τ**	Y	32.61	0.33
T_d**=3.75% τ**	Y	28.80	0.19
T_d**=5% τ**	N	N/A	N/A

The above table shows that when the delay is increased the maximum phase margin happens with bigger value of b_0. Increase the b_0 also has its drawbacks. If we look to at the schematic in Figure 4, the control signal will be divided by this amount before goes to the system and having bigger value that will impact in a smaller control effort, that will slow down the system response. This is **tip number 5,** increase b_0 can improve stability of your controller, but slower response will be seen.

3.3.3. Spectrum analysis-based observer bandwidth selection: A spectrum analysis was another tool used to study the controller behavior, more specifically the ESO behavior. Before doing the spectrum analysis of ESO, the spectrum of system disturbances was plotted. The schematic structure of the simulation is depicted in figure 26. For a first order plant, white noise was injected as *u,* and colored noise was injected as external disturbance.

The figure 27 shows the spectrum plotted in a semi log scale to have a clear view at lower frequencies, where the disturbances are normally concentrated. In this example we can see that disturbances have high power at frequencies lower than 0.1 rad/s.

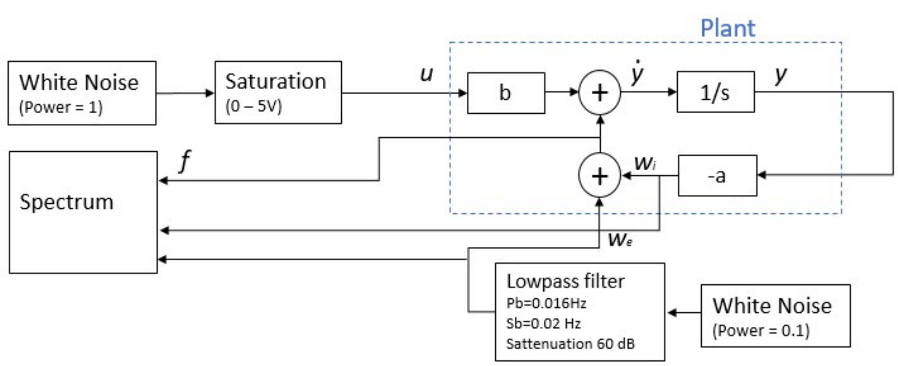

Figure 26: Logic used to plot spectrum of disturbances

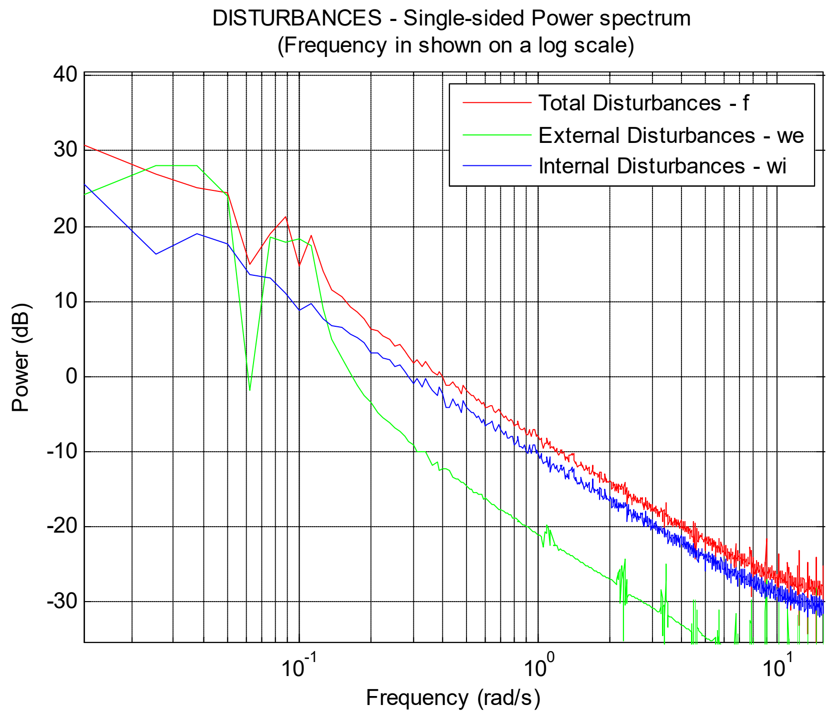

Figure 27: Spectrum of disturbances

The spectrum analysis of ESO was done with the spectrum of the remaining disturbance signal, that means $f - \hat{f}$, that can show how efficient the estimation was done. Figure 28 shows the structure that was simulated in Simulink. This simulation was repeated for different observer bandwidths (ω_o) and plotted together in Figure 29.

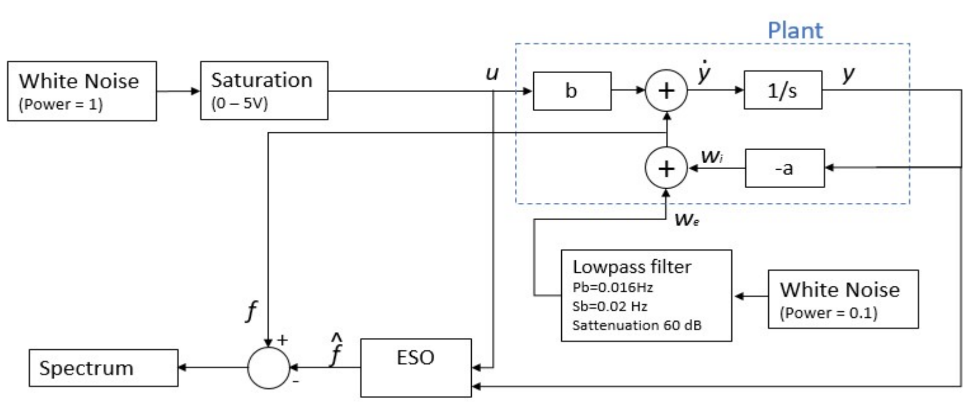

Figure 28: Logic used to plot spectrum of ESO

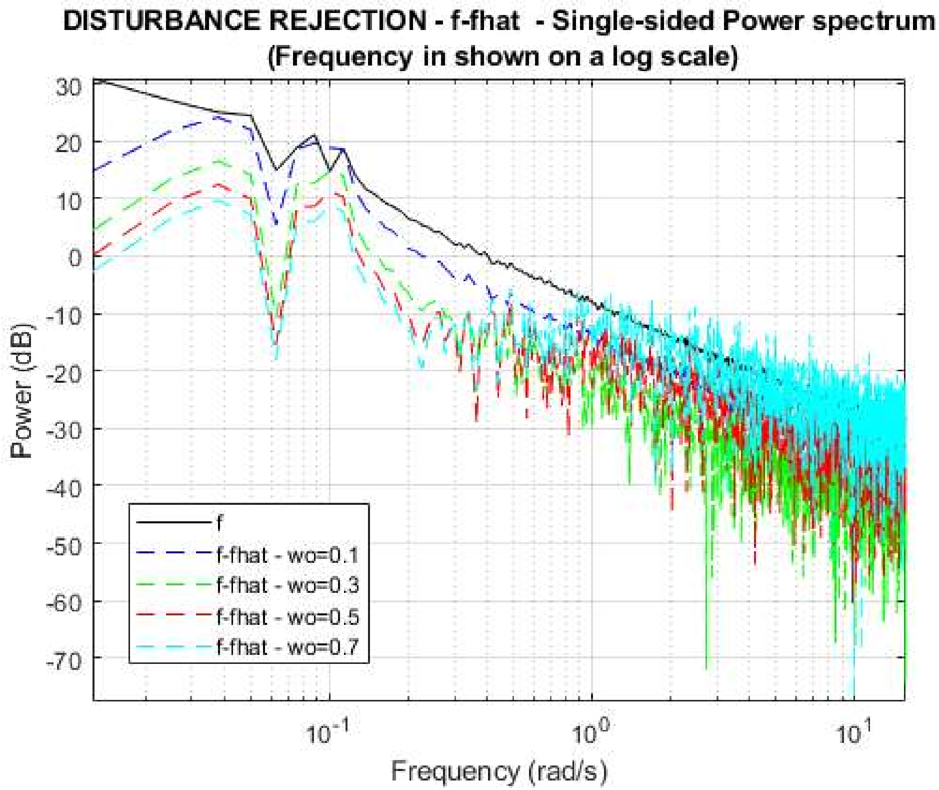

Figure 29: Spectrum of disturbances

It is possible to observe at lower frequencies that; the more we increase the observer bandwidth the lower is the level of system uncertainty $(f - \hat{f})$. And similar to what happened when the controller bandwidth selection was studied for values under the observed maximum disturbance frequency, the estimation is not satisfactory and for values above that point the estimation is better. From this graph is also possible to observe at higher frequencies that as we increase the observer bandwidth, we also increase the noise in our system (higher frequencies). If your measurement is not very noisy that can be done also for real systems, from (4) we can estimate real f as the first derivative of the measured signal minus control signal times b_0 and $\hat{f}$ is the ESO output. Based on that the ***last tip*** for practitioners would be to try to identify for the known disturbances what is its maximum frequency, and that can help in observer bandwidth (ω_o) tuning.

3.4. Guidelines for Practitioners

The proposed general solution controller needs to have your parameters tuned according to the process it is going to be applied, that is going to be done by the practitioners. As we could see from sections above this process is very intuitive and logical. Based on their knowledge of the process to be controlled they can use this chapter as a guide, bellow a brief summary of the tips are given:

1. Setpoint profile: if the process to be controlled requires good abilities of setpoint tracking, a smooth profile for reference signal could be defined instead of just step setpoints because that can help in increasing energy savings as well as decreasing overshooting and equipment wear and tear;

2. Estimation of b_0: an approximated value for b_0 can found from open loop step tests as mentioned in section 2.3.

3. Controller bandwidth (ω_c) selection: ω_c is related to the speed of system response, and a good starting point would be to define the desired settling time ($T_s = 4\tau$) for the system and based on that find the initial value for controller bandwidth $\omega_c = 5 \sim 10 * 1/\tau$.

4. Sampling time selection: should be selected according to your system needs, a balance between system speed and controller load must be kept in mind;

5. b_0 tuning: increasing b_0 can improve stability of your controller when needed, but slower response will be seen;

6. Observer bandwidth (ω_o) selection: if the frequency of disturbances are known that can be used as a starting point for the observer bandwidth (ω_o) selection.

CHAPTER IV

A CASE STUDY: SERVER TEMPERATURE REGULATION

After the detailed analysis is done, testing it on other processes is the next step of this study. For this part, a real temperature control problem is selected. Temperature loops are good examples of first order systems with time-delay. The problem of temperature regulation inside computers servers was picked, which especially for the SSD drives (that nowadays have higher density and temperature) is a challenging issue.

The number of data centers is rapidly growing throughout the world, fueled by the increasing demand of remote storage and cloud computing services. Computational density has also been increasing over the years. With those aspects comes the problem of high temperature inside the servers. ASHRAE [32] has published a trend of heat load increasing in the last years in IT equipment.

Combining the information from Figure 30 with an increasing number of datacenters, it will consequently increase the power consumption of datacenters all around the world. Figure 31 shows an estimated yearly energy consumption of data centers for the next few years [33].

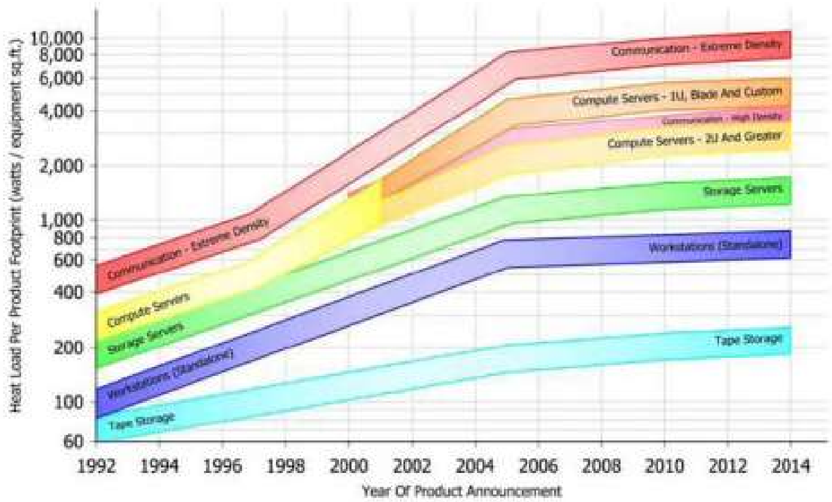

Figure 30: Heat load trend for IT equipment

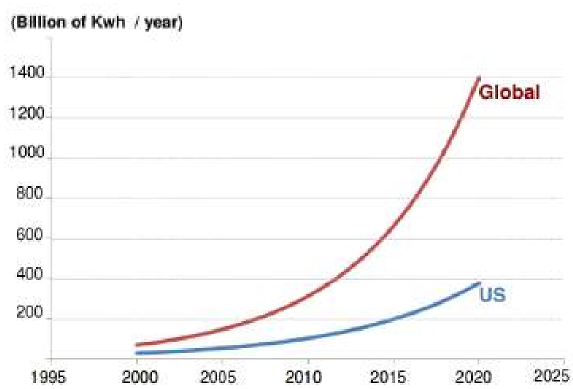

Figure 31: Estimated yearly energy consumption of data centers

This is one of the main reasons why power consumption is turning out to be a critical issue in the design and operation of servers and data centers nowadays. For the year of 2006 the Environmental Protection Agency (EPA) reported that 60 billion kWh, or 1.5% of the total U.S.A. electricity consumption, was used to power data centers [34]. And it is expected to increase considerably in the upcoming years. Several studies have shown that for every 1W of power used to operate a server, an additional 0.5-1W of power is required to cool it in data centers.

Thermal management of datacenters is now such a hot topic in research. Some studies cover entire data center temperature management. For example, Chen [35], based on sensor data, predicts the server temperatures in real time and optimize the temperatures setpoints and cold air supply rates of cooling systems, as well as the speeds of servers' internal fans, to minimize their overall energy consumption. Huang [36] shows that there is a tradeoff between the power of HVAC that is used to cool server inlet air flow in a data center cooling zone and the power from cooling fans inside individual servers. Pradelle [37] proposes an optimal fan setting, which simultaneously minimizes the power leakage and fan power consumption.

Other studies concentrate on the temperature management of server itself (mainly fan control optimization, the same direction this study is taking). Work [38] presents a model-based approach to manage fan power and provide optimal cooling and energy efficiency and [39] presents a PID neural network with fan-power based optimization.

4.1. Experimental Testbed

In order to simulate the real problem of temperature control inside servers, a testbed was constructed. A real 1U server chassis with PWM fans (Delta FFB0412SHN) are going to be used in the experiment. There are four fans that can be controlled one by one. To mimic the SSD drives, copper blocks (100x70x3mm) were used and heated with a foil heater, to emulate the SSD workload. The higher the workload is, the higher the SSD temperature gets. The temperature of each block is measured with a type K thermocouple attached to the block. To control the heaters, fans, as well as to read the temperature and create the control logic, a Simulink Real Time Explorer was used. In the experiment, two copper blocks were used, each block represents one SSD, each block was positioned in front of two fans. Each set of 1 block and 2 fans are called "Zone".

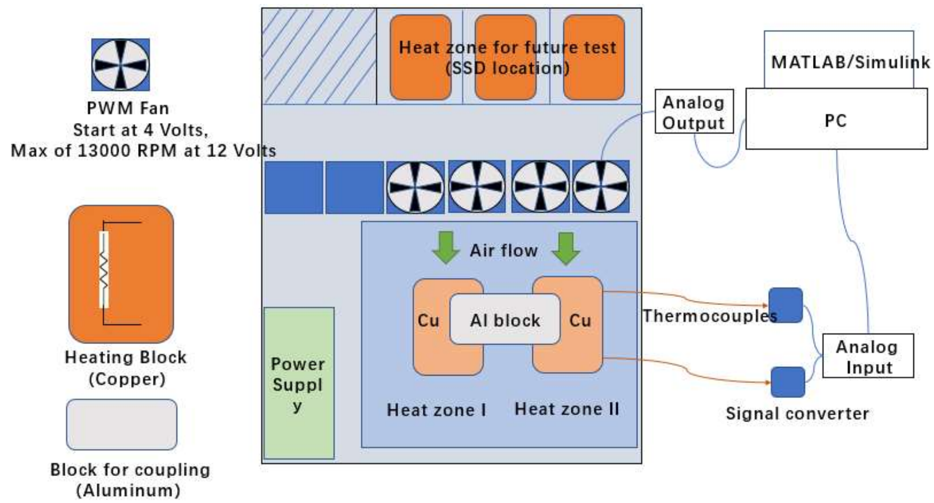

Figure 32: Testbed scheme

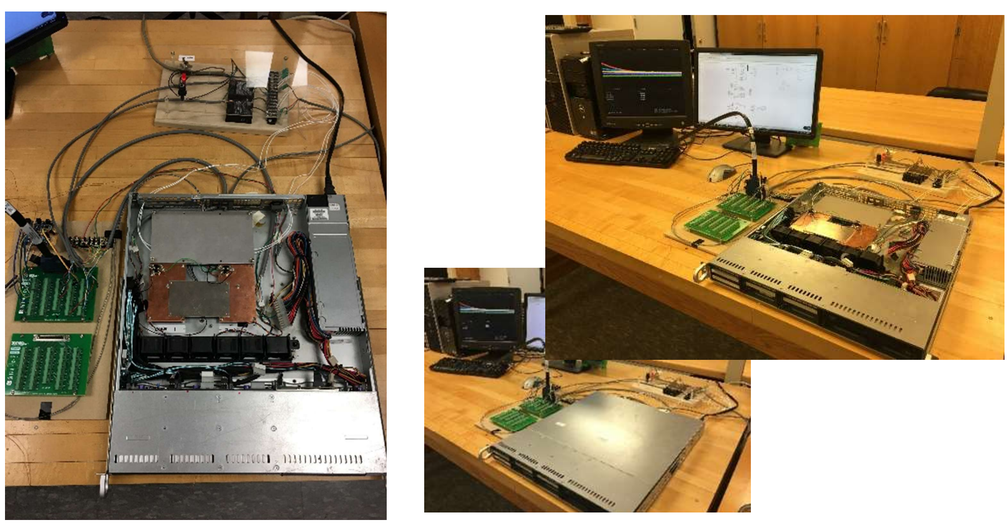

Figure 33: Testbed setup photos

After the hardware was set up, the next step was to derive a mathematical model to capture the behavior of the real system. To do that, many different open loop tests were conducted. Five different levels (0, 20, 50, 70 and 100%) of cooling and heating for each zone were selected. Figure 34 shows some of the open loop tests done with only one zone heated all the time (3000s) with heater at some level and cooled after 2000 seconds at some level of fan speed. The second zone temperature variation is just due to the coupling effect - heater is off all the time and fans at minimum speed.

Figure 35 shows some tests done with two zones heated all the time (3000s) with heaters at the same level and after 2000 seconds cooled at the same level of fan speed.

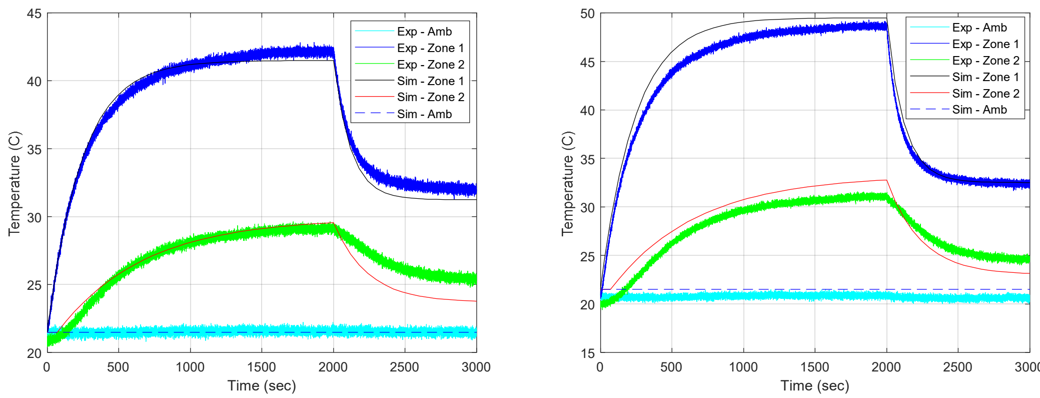

Figure 34: Open loop for one zone (a)heat/cool level = 50% (b) heat/cool level = 70%

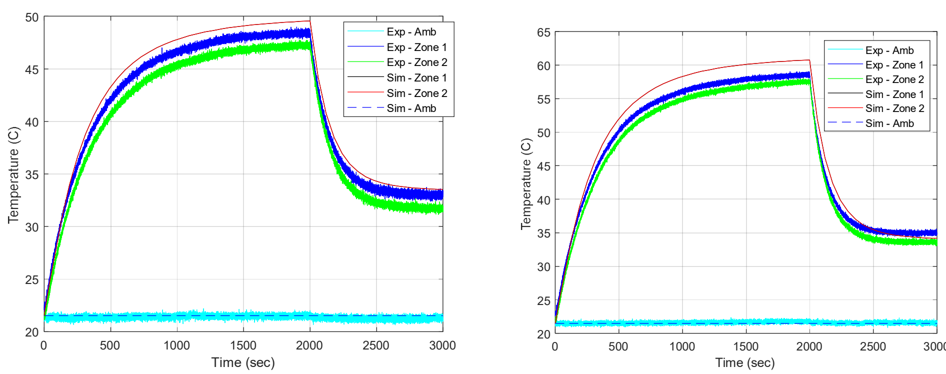

Figure 35: Open loop for two zones (a)heat/cool level = 50% (b) heat/cool level = 70%

The above tests verified that the considered temperature control system can be represented (with satisfactory accuracy) with a first order model. Based on the study and open loop tests, the below equation represents the model of the considered physical system (Wang [38]):

$$\begin{bmatrix}\Delta T_1\\ \Delta T_2\end{bmatrix} = \begin{bmatrix} \dfrac{k_{h1}}{\tau_{h1}s+1} & \dfrac{(1-h_c)k_{h2}}{(1+2.6h_c)\tau_{h2}s+1}e^{-td_{hc}s} \\ \dfrac{(1-h_c)k_{h1}}{(1+2.6h_c)\tau_{h1}s+1}e^{-td_{hc}s} & \dfrac{k_{h2}}{\tau_{h2}s+1} \end{bmatrix}\begin{bmatrix}Q_1\\ Q_2\end{bmatrix} - \begin{bmatrix} \dfrac{k_{c1}}{\tau_{c1}s+1} & \dfrac{(1-c_c)k_{c2}}{(1+2.6c_c)\tau_{c2}s+1} \\ \dfrac{(1-c_c)k_{c1}}{(1+2.6c_c)\tau_{c1}s+1} & \dfrac{k_{c2}}{\tau_{c2}s+1} \end{bmatrix}\begin{bmatrix}FS_1^{NR}\\ FS_2^{NR}\end{bmatrix} \quad (25)$$

where ΔT_n, Q_n, FS_n , h_c and c_c are, respectively, the temperature variation on zone n, heating generated on zone n, fan speed of zone n, heating coupling factor, and cooling coupling factor. As the two zones were designed in the same way, the gains and time constants were considered same for both. From open loop tests, the range of variation of the above gains, time constants, and factors has been defined. From this range, one set of values was picked that was judged to cover a wide change in the scenarios to start the implementation of the closed loop controllers, in Figures 34 and 35 the red line is the simulation result using this set of values.

TABLE VIII: Modeling parameters

Heating			Cooling		
	Range	**Selected value**		**Range**	**Selected value**
$k_{h1} = k_{h2}$	[37,43]	33	$k_{c1} = k_{c2}$	[23,38]	38
$\tau_{h1} = \tau_{h2}$	[200,240]	240	$\tau_{c1} = \tau_{c2}$	[80,120]	120
h_c	[0.55,0.65]	0.65	c_c	[0.4,0.5]	0.5
td_{hc}	[50,70]	50	td_c		5

4.2. Controller Design

The temperature regulation problem was reformulated in terms of disturbance rejection and the general purpose first order ADRC controller was implemented. From the obtained model, we can state that our temperature variation is a function of heating (workload) and fan speed, from its own zone or from the coupling zone. Heating is not a controllable variable, only its minimum and maximum values are known, but the uncertainty related to when and how much it varies, will be considered as a disturbance in the control system. The fan speed is controllable, but the coupling factor makes the control problem much trickier, as the fans of each zone affects both zones temperature. The coupling problem can be also treated as a disturbance rejection problem. Putting that into disturbance rejection framework, heating and cooling coupling will be parts of the total disturbance (*d*). That been said, each zone can be treated as a simple first order system, and the rate of change in temperature can be written as:

$$\Delta\dot{T}_n = d_n + b_{0n}FS_n \qquad (26)$$

The proposed general purpose ADRC based controller was implemented in Simulink, one for each zone separated, its structure is seen in Figure 36.

The SSD drives have a built-in algorithm to control its temperature by reducing its performance after reaching certain predefined temperature limits. The control objective is to not let the temperature reach that limit of performance degradation. This way the controller setpoint should be set some degrees below that level. But how many degrees? The answer to that question is: the controller setpoint should be set in a way that the maximum overshoot does not exceed that point of performance degradation. The higher the setpoint, the lower fan speed requirements will be and consequently less power consumption and energy savings.

The real SSD drives start throttling at 60°C, and shutdown at 70°C. In the experimental open loop tests, it is possible to observe that the maximum temperature they could reach is around 70°C. If the setpoint for the controller is set close to 60°C, the fans probably won't need to work hard, this way it was decided to lower this setpoint for the existing experiment to 40°C. And it was assumed that the throttling point would start at 45°C, and shutdown at 50°C.

Settling time is not a requirement from the point of view of reaching the throttling limit but one interesting performance indicator would be the overshoot rate (sum of overshoot time simulation run time). This indicator will be shown for two temperature limits: over setpoint and over 45°C (throttling limit).

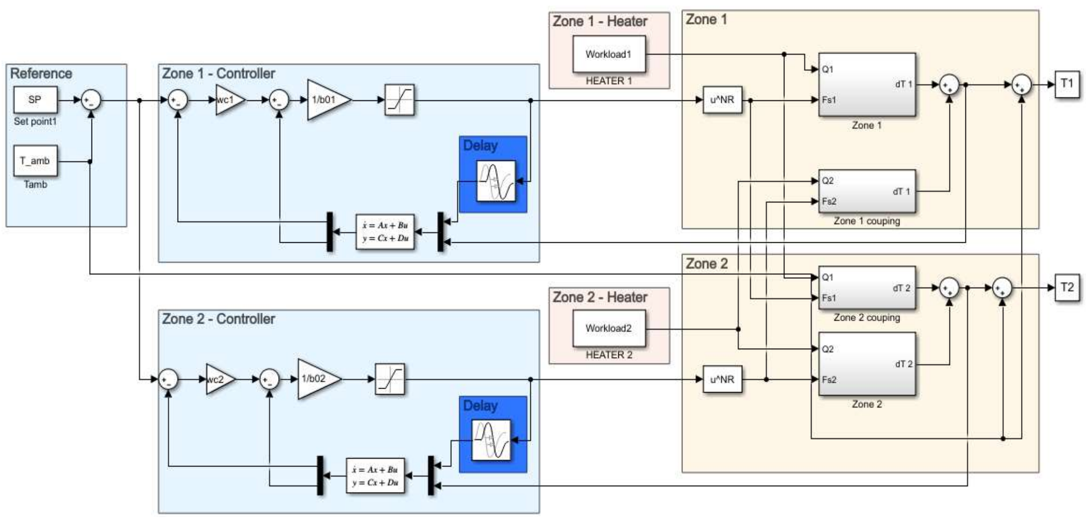

Figure 36: Two zones controller simulation

The main task for this controller is disturbance rejection, not setpoint tracking, since the workload changes without previous notice. To simulate a real operation of the server, the workload of zone one will change every 800s (~13min) and zone two every 1600s (~27min). It was considered that server minimum workload is 20%, so the workload is never lower than that. In the figure bellow, an user-defined profile is presented that emulates the changeable system workload

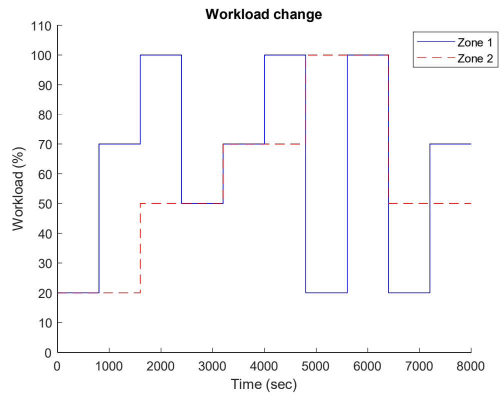

Figure 37: User-defined workload profile (disturbance)

4.3. Simulation Results

The controller was tuned using the same technique presented in chapter 3. This section presents the results for simulation and experiment with the temperature setpoint equal to 40 °C. The graph below shows the temperature against time, as well as the fan speed against time. The results obtained with a PID controller are also shown in order to quantitatively compare the results.

From figures 38 and 39, it is easily seen that the ADRC has a much better response than PID. Some performance indicators are used to compare the response of ADRC and PID controllers:

- peak temperature - maximum overshoot
  (that indicate if the controller reached the throttling or shutdown limit);

- overshoot percentage rate over setpoint
  (sum of time with temperature over setpoint/ simulation run time);

- overshoot percentage rate over throttling limit (45°C)
  (sum of time with temperature over 45°C / simulation run time);

- energy consumption.

It is known that fans power consumption is related to the third power of fans speed. This way the integration of the third power of fans speeds was used to compare the total energy consumed by the fans in each case.

$$Power \propto \int_0^{T_{run}} FS_i^3 dt \tag{27}$$

That integration won't provide exactly power consumption, so the PID integration was chosen to normalize all the values. This way it is possible to have a good-enough approximation of how much efficiency or deficiency the new controller can bring.

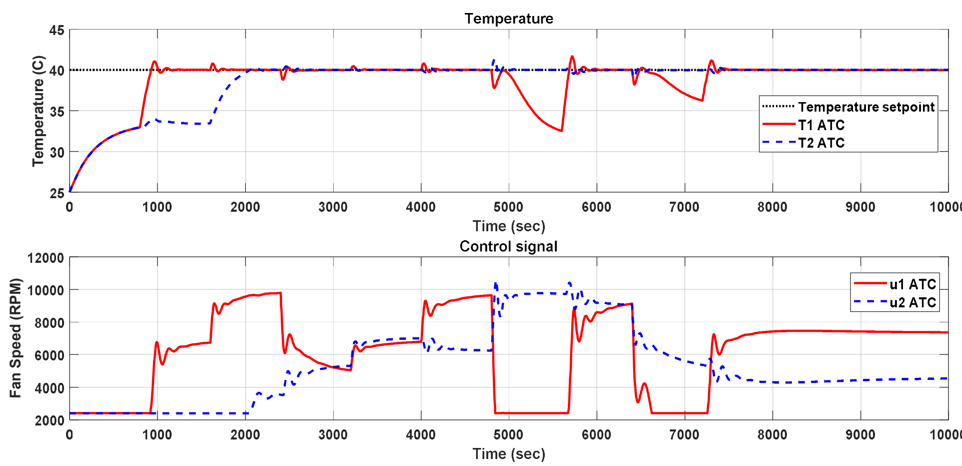

Figure 38: ADRC performance

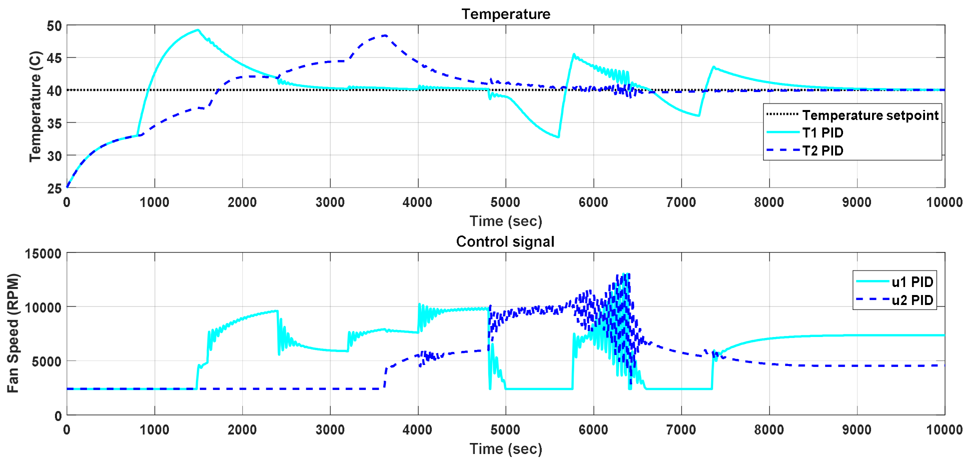

Figure 39: PID performance

Tables IX and X show the performance indicators for Zones 1 and 2. With the setpoint set at 40°C, it is possible to check that, in terms of peak temperature and overshoot rate, the new controller is much better than PID, the maximum overshoot is less than 5%, while PID is over 20%. Also, the overshoot rate for temperature above the setpoint for PID is 8 times bigger than ADRC, and ADRC never reached the throttling limit, while PID exceeded that almost 10% of time.

In terms of overall energy consumption, if both methods were considered with same setpoint, the ADRC is worse than PID. However, that is not a fair comparison, since ADRC kept the temperature most part of the time in a lower level, what means that more energy was required. In order to have a more fair comparison, and show the benefits of ADRC over PID, the ADRC controller setpoint was increased to 43°C. Even with that increase in setpoint, ADRC continues to have better results in terms of overshoot and never reached the throttling limit, leading it to an energy consumption lower than PID (around 13% less).

If the setpoint is pushed even higher, to have overshoot indicators closer to PID method, the savings in energy consumption can reach almost 30%. For this particular scenario and used criteria, it shows its unique strength in actively reducing the overshoot, overshoot rate, and consequently letting the setpoint be elevated and in that case reaching the main target of power consumption savings.

TABLE IX: Zone 1 performance indicators

	PID	ADRC		
Setpoint (°C)	**40**	**40**	**43**	**45**
Peak Temperature (°C)	49.2	41.7	44.5	46.4
Overshoot rate (%) > Setpoint	88.0	10.1	10.2	11.2
Overshoot rate (%) > 45°C	9.7	0	0	11.2
Energy consumption (normalized with PID result)	1	1.09	0.86	0.72

TABLE X: Zone 2 performance indicators

	PID	ADRC		
Setpoint (°C)	**40**	**40**	**43**	**45**
Peak Temperature (°C)	48.4	41.3	44.3	46.4
Overshoot rate (%) > Setpoint	58.5	7.6	7.2	6.1
Overshoot rate (%) > 45°C	8.3	0	0	6.1
Energy consumption (normalized with PID result)	1	1.12	0.87	0.71

4.4. Experimental Verification

Based on the simulation results from the above section, PID and ADRC methods are selected to be tested on our experimental testbed. The test is performed under the same user-defined varying workload, as shown in the simulation part. The ambient temperature at the inlet of the server chassis is represented here as *Ambient inlet*.

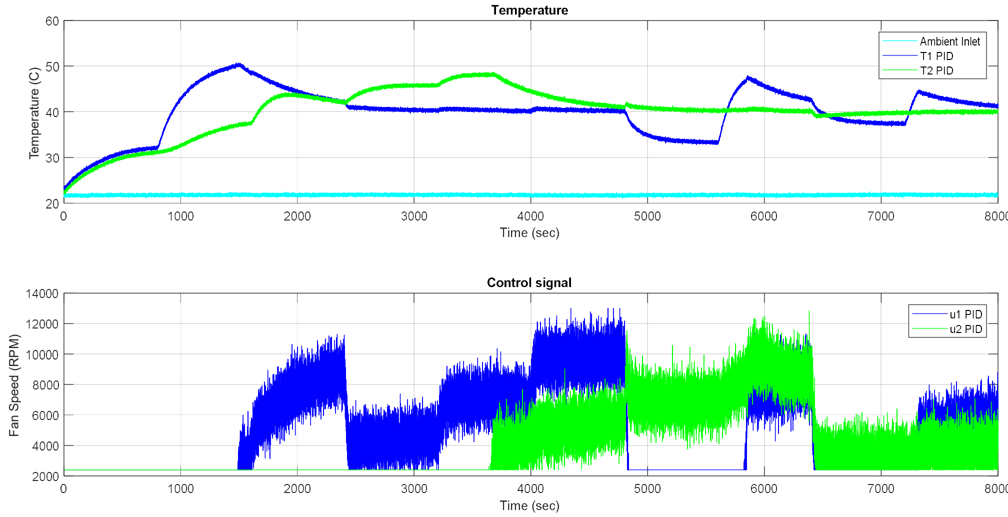

Figure 40: PID experiment results

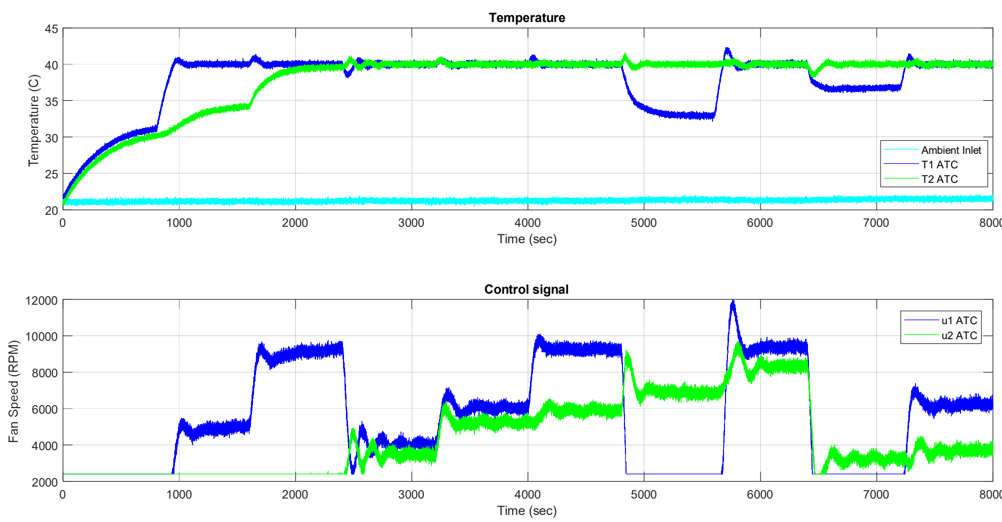

Figure 41: ADRC experiment results

Figures 40 and 41 show the results of PID and ADRC. They are similar to the ones obtained in the simulation. One extra point, seen in the conducted experiment, is that the PID control signal contains more noise than the ADRC control signal, which can be considered as one more advantage of ADRC.

The general purpose ADRC controller is proved to be an effective control method. By comparing the results with PID method, ADRC shows unique advantages in dealing with system overshoot and overshoot percentage rate. The ADRC made the setpoint limit elevation possible and consequently reaching energy savings up to 30%.

CHAPTER V

CONCLUSION AND FUTURE WORK

5.1. Conclusion

In this thesis it was investigated why industrial process control still relies on an almost one hundred years old methodology called PID and why new technologies are still not used. This popular algorithm has many advantages such as simple mathematics, model free design, flexible to many different applications, and with generally good performance. But to have good performance its parameters must be carefully tuned, however, and it is mostly done empirically. Today there are more than 1000 different ways of tuning PID controllers. Many surveys show that in most cases PID controllers are badly tuned or run in manual or works with factory default settings. It's an industry wide problem!

Modern control theories are mainly supported by modeling the physical process, but that can be very challenging for process industry. Overcoming this model dependency, Han in 1998 introduced a novel methodology called Active Disturbance Rejection Control (ADRC), where the central idea is to estimate and mitigate uncertainties coming from disturbances and plant dynamics in real time and this estimation is calculated based on plant input-output signals. Based on ADRC, a general-purpose controller is presented in this study, which has the potential to ***transform*** the process control industry and bring it to another era. It retains the simplicity of PID, but with much better performance and no model dependency. This solution is not intended to be optimal for a particular process, since it is general and applicable for a wide range of processes, but it takes a step further in the right direction is establishing a new platform for process control.

In direct contrast to PID, the tuning process of this new solution is very intuitive and logical. Based on the knowledge of the process and what the control system requirements, the initial tuning parameters setting can be easily found. The main contribution of the work in this thesis comes from this tuning approach, where it was demonstrated that the parameter b_0 can also be used as a tuning parameter.

Practical results from a case study about temperature control on servers show the improvements this controller can bring for a very challenging problem that contains delays, coupling effects and lots of disturbances. A much better response was obtained in terms of overshoot what made possible to increase the setpoint of temperature and to obtain 30% of energy saving.

5.2. Future Research

Though a lot has been accomplished in this thesis, it is only a beginning and there is still much to be worked on. One problem yet to be solved is to implement the same type of solution for higher order systems for problems where the second order solution presented here is not sufficient.

Furthermore, in some cases the model information exists and it should be utilized, not wasted. Because, as mentioned before, the more information the controller has the better its response would be. It is therefore a logical next step to add model information to the general solution in order to generate a particular solution for various applications, with improved performance, robustness, and stability margins.

REFERENCES

[1] L. T. C. Rolt and J. S. Allen, The Steam Engine of Thomas Newcomen, New York : Neale Watson Academic Publications, Inc., 1977.

[2] W. Trinks, Governors and Governing of Prime Movers, New York: D. Van Nostrand Company, 1919.

[3] D. P. Eckman, Principles of Industrial Process Control, New York: John Wiley & Sons, Inc., 1945.

[4] S. Bennett, "A brief history of automatic control," *IEEE Control Systems Magazine,* vol. 16, no. 3, pp. 17-25, June 1996.

[5] H. Chesnut and R. W. Mayer, Servomechanisms and Regulating System Design, New York: Wiley, 1951.

[6] A. C. Hall, "Application of Circuit Theory to the Desing of Servomechanisms," *Journal of the Franklin Institute,* vol. 242, no. 4, pp. 279-307, 1946.

[7] G. C. Newton, L. A. Gould and J. F. Kaiser, Analytical Design of Linear Feedback Controls, New York: John Wiley and Sons, Inc, 1957.

[8] F. G. Shinskey, Process Control Systems, New York: McGraw-Hill, Inc, 1967.

[9] E. S. Smith, Automatic Control Engineering, New York: McGraw-Hill Book Company, inc, 1944.

[10] H. S. Tsien, Engineering Cybernetics, New York: McGraw-Hill, 1954.

[11] A. O'Dwyer, "PI and PID controller tuning rules: an overview and personal perspective," in *IET Irish Signals and Systems Conference*, Dublin, 2006.

[12] P. V. Overschee and B. De Moor, "RaPID: The End of Heuristic PID tuning," *IFAC Proceedings Volumes,* vol. 33, no. 4, pp. 595-600, 2000.

[13] T. Samad, "A survey on industrial impact and challenges thereof," *IEEE Control Systems Magazine,* pp. 17-18, February 2017.

[14] Z. Gao, "Active disturbance rejection control: a paradigm shift in feedback control system design," *Proceedings of American Control Conference,* pp. 2399-2405, 2006.

[15] D. E. Seborg, T. F. Edgar and D. A. Mellichamp, Process Dynamics and Control Second Edition, John Wiley & Sons, Inc., 2004.

[16] G. Stein, "Respect the Unstable," *IEEE Control Systems Magazine,* pp. 12-25, August 2003.

[17] Z. Gao, Y. Huang and J. Han, "An alternative paradigm for control system design," *Proceeding of IEEE Conference on Decision and Control,* vol. 5, pp. 4578-4585, 2001.

[18] Z. Gao, "Scaling and parametrization based controller tuning," *Proceedings of the 2003 American Control Conference,* vol. 6, pp. 4989-4996, 2003.

[19] L. Wang, Q. Li, C. Tong, Y. Yin, Z. Gao, Q. Zheng and W. Zhang, "On control design and tuning for first order plus time delay plants with significant uncertainties," in *American Control Conference (ACC)*, Chicago, 2015.

[20] P. Nowak, J. Czeczot and T. Klopot, "Robust tuning for first-order reduced active disturbance rejection controller," in *21st International Conference on Methods and Models in Automation and Robotics (MMAR)*, Miedzyzdroje, Poland, 2016.

[21] T. Marlin, Process Control, Designing Processes and Control Systems for Dynamic Performance, McGraw-Hill College, 2000.

[22] W. Zhou and Z. Gao, "An active disturbance rejection approach to tension and velocity regulations in web processing lines," in *16th IEEE International Conference on Control Applications*, Singapure, 2007.

[23] Q. Zheng and Z. Gao, "An energy saving, factory-validated disturbance decoupling control design for extrusion processes," in *10th World Congress on Intelligent Control and Automation*, Beijing, 2012.

[24] Z. Gao, "On the Centrality of Disturbance Rejection in Automatic Control," *ISA Transactions,* vol. 53, no. 4, pp. 850-857, 2014.

[25] R. Madonski, M. Nowicki and P. Herman , "Application of active disturbance rejection controller to water supply system," in *33rd Chinese Control Conference*, Nanjing, 2014.

[26] D. Li, Z. Li, Z. Gao and Q. Jin, "Active disturbance rejection-based high-precision temperature control of a semibatch emulsion polymerization reactor," *Industrial & Engineering Chemistry Research,* pp. 3210-3221, February 2014.

[27] L. Sun, D. Li, K. Hu, K. Y. Lee and F. Pan, "On tuning and practical implementation of active disturbance rejection controller: a case study from a regenerative heater in

a 1000MW power plant," *Industrial & Engineering Chemistry Research,* pp. 6686-6695, May 2016.

[28] P. T. Garrán and G. Garcia, "Desing of an optimal PID controller for a coupled tanks system employing ADRC," *IEEE Latin America Transactions,* vol. 15, no. 2, pp. 189-196, 2017.

[29] S. Zhao and Z. Gao, "Modified active disturbance rejection control for time-delay systems," *ISA Transactions,* vol. 53, no. 4, pp. 882-888, 2013.

[30] G. Tian and Z. Gao, "Frequency response analisys of active disturbance rejection based on control system," in *16th IEEE International Conference of Control Applications*, Singapure, 2007.

[31] R. Miklosovic, A. Radke and Z. Gao, "Discrete implementation and generalization of the extended state observer," in *American Control Conference*, Minneapolis, Minnesota, USA, 2006.

[32] ASHRAE Technical Committee (TC) 9.9 Mission Critical Facilities, Technology Spaces, and Electronic Equip, "IT Equipment Thermal Management and Controls," American Society of Heating, Refrigerating and Air Conditioning Engineers, Inc., 2012.

[33] C. L. Belady, "Projecting Annual New Datacenter Construction Market Size, Global Foundation Services," Global Foundadtion Services, March 2011.

[34] U.S. Environmental Protection Agency (EPA), "Report to Congrees on Server and Data Center Energy Efficiency," 2007.

[35] J. Chen, R. Tan, G. Xing and X. Wang, "PTEC: A system for predictive thermal and energy control in data centers," in *IEEE Real-Time Systems Symposium*, Rome, Italy, 2014.

[36] W. Huang, M. Allen-Ware, J. B. Carter, E. Elnozahy, H. Hamann, T. Keller, C. Lefurgy, J. Li, K. Rajamani and J. Rubio , "TAPO: thermal-aware power optimization techniques for servers and data centers," in *International Green Computing Conference and Workshops*, Orlando, FL, 2011.

[37] B. Pradelle, N. Triquenaux, J. C. Beyler and W. Jalby, "Energy-centric Dynamic Fan Control," *Computer Science Research and Development,* vol. 29, no. 3-4, pp. 177-185, 2013.

[38] Z. Wang, C. Bash, N. Tolia, M. Marwah, X. Zhut and P. Ranganathan, "Optimal fan speed control for thermal management of servers," in *ASME Interpack Conference*, San Francisco, 2009.

[39] C. Lee and R. Chen, "Optimal self tuning PID controller based on low power consuption for a server fan cooling system," *Sensors,* vol. 15, no. 5, pp. 11685-11700, 2015.

[40] J. G. Ziegler and N. B. Nichols, "Optimum settings for automatic controllers," in *trans. ASME*, 1942.

[41] J. G. Ziegler and N. B. Nichols, "Process lags in automatic control circuits," in *Trans. ASME*, 1943.

[42] C. Zhao and D. Li, "Control design for the SISO system with the unknown order and the unknown relative degree," *ISA Transactions,* vol. 53, no. 4, pp. 858-872, 2014.

[43] P. Albertos, P. Garcia, Z. Gao and T. Liu, "Disturbance rejection in process control," in *11th World Congress on Intelligent Control and Automation*, Shenyang, 2014.

[44] Q. Zheng and Z. Gao, "Predictive active disturbance rejection control for processes with time delay," *ISA Transactions,* vol. 53, no. 4, pp. 873-881, 2014.

[45] D. Li and Z. Gao, "Tuning method for second-order active disturbance rejection control," in *30th Chinese Control Conference*, Yantai, 2011.

[46] K. Ogata, Modern Control Engineering, Pearson Education, Inc., 2010.

[47] M. C. Popescu, G. Manoela and A. D. L. P. Popescu, "Implementation of new solution software for three tank system control," 2018.

[48] Y. Hou, Z. Gao, F. Jiang and B. Boulter, "Active disturbance rejection control for web tension regulation," in *40th IEEE Conference on Decision and Control*, Orlando, 2001.