PRESSURE CONTROL METHODS FOR A PRESSURIZED COOLING SYSTEM WITH REPETITIVE CYCLES OF TRANSIENT DYNAMIC BEHAVIOR

by

Michael Stephen Smith

A dissertation submitted to the faculty of
The University of North Carolina at Charlotte
in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in
Electrical Engineering

Charlotte

2015

Approved by:

____
Dr. Sukumar Kamalasadan

____
Dr. Yogendra Kakad

____
Dr. Abasifreke Ebong

____
Dr. K. Scott Smith

ABSTRACT

MICHAEL STEPHEN SMITH. Pressure control methods for a pressurized cooling system with repetitive cycles of transient dynamic behavior. (Under the direction of DR. SUKUMAR KAMALASADAN)

Pressurizer (PZR) behavior in typical pressurized water reactor (PWR) nuclear power plants is well understood. Pressure control is relatively straightforward for a typical PWR, since the larger fluid system normally operates in a very steady state condition. However, pressure control is more complicated for a pressurizer in a pulsed cooling system, such as the cooling system for a tokamak fusion reactor. During normal operation the tokamak's plasma is pulsed, instead of staying at a constant value, which results in temperature swings between the plasma pulses. This design characteristic means that (a) coolant temperatures fluctuate over a larger range during normal operation than typical PWRs experience and (b) fluid volume also varies as the coolant temperature changes, since fluid (water) density is a function of temperature.

Pressurizer pressure control is typically accomplished with an on/off and proportional control strategy in PWRs. However, this approach alone may not meet the desired control performance for a pulsed cooling system. Therefore, a dynamic model based control design approach is proposed that permits modification online as the process dynamics change by uniquely combining a hybrid control technique with a method to improve system knowledge. This research includes contributions to PZR control modeling, dynamic simulation inputs, adaptive-optimal and adaptive-dynamic control, and system knowledge. Simulations support that this approach enables greater control of the process during transients than is achievable with a conventional control approach.

ACKNOWLEDGMENTS

I want to acknowledge and thank Dr. Sukumar Kamalasadan for his assistance, guidance, and understanding throughout the course of this project. His proficiency and knowledge was immensely helpful. I also want to thank Dr. Walter Van Hove from Oak Ridge National Laboratory (ORNL) for his contribution to this project. Finally, I would like to acknowledge and thank several other people, including members of my advisory committee, for their assistance throughout the course of this project. These individuals are: Dr. Yogendra Kakad, Dr. Abasifreke Ebong, and Dr. K. Scott Smith. Most importantly, I would like to thank my family for encouraging me during my pursuit of this goal.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

This chapter introduces the basic concepts for nuclear fission and fusion reactors and provides an overview of pressurized cooling systems for these types of reactors, as a basis for understanding the focus of this research effort. A survey is presented of the recent advances in pressure control for a pressurized cooling system. Additionally, and most importantly, this chapter introduces the challenges associated with pressure control for a pressurizer in a pulsed system that created the need to address this research topic.

Pressurizer (PZR) behavior in typical pressurized water reactor (PWR) nuclear power plants is well understood. Pressure and level control are relatively straightforward for a typical PWR, since the larger fluid system normally operates in a very steady state condition. However, pressure control is more complicated for a pressurizer in a pressurized cooling system with repetitive cycles of transient dynamic behavior, such as a tokamak fusion reactor cooling system. During normal operation the tokamak's plasma is pulsed, instead of remaining at a constant value, which results in temperature swings between the plasma pulses. This design characteristic means that (a) coolant temperatures fluctuate over a larger range during normal operation than typical PWRs experience and (b) fluid volume also varies as the coolant temperature changes, since fluid (water) density is a function of temperature.

In PWRs, pressurizer control is typically accomplished with an on/off and proportional control strategy. However, for a tokamak cooling system, this approach alone may not meet the desired control performance, considering the varying process conditions described (i.e., a pulsed cooling system). Therefore, advanced control strategies (e.g., adaptive control, neural networks, and model predictive control) may have applicability to this type of problem.

In this research a dynamic model based control design approach is proposed that permits modification online as the process dynamics change. A new control method (i.e., called LiMeRICK) is presented that uniquely combines a hybrid control technique (i.e., called LiMe) with a technique to improve system knowledge (i.e., called RICK). The RICK method (used to improve the pressurizer system knowledge) is based on Recursive Least Squares Identification (RLS ID), a Kalman filter, and a model corrector. The LiMe technique (used to improve pressurizer pressure control) is a hybrid control architecture that combines a Linear Quadratic Regulator (LQR) and Minimum Variance (MV) controller. Combining these two new methods (i.e., LiMe and RICK) enables enhanced control during transient conditions. Contributions of this research are described in Section 1.7. Simulations are presented (e.g., see CHAPTER 6) that demonstrate greater process control during transients is achieved with the new control method (e.g., LiMeRICK) than is achievable with a conventional (e.g., on/off and proportional) control approach.

1.1 Pressurized Water Reactor (PWR) – Nuclear Fission Overview

The basic process for all Rankin cycle power plants (e.g., fossil fuel, nuclear fission and nuclear fusion) is very similar [17]. Water is heated to produce steam; the steam is then used to turn a turbine/generator to produce electricity [33]. The main difference is the heat source produced (i.e., nuclear fusion, nuclear fission, fossil fuels, etc.).

Nuclear fission reactors operate on the principal of splitting the nucleus of heavy atoms, such as uranium, into nuclei of lighter atoms. During the fission process energy is released in the form of thermal energy (i.e., heat). This thermal energy can be captured and transformed to produce electricity, which is a readily transmitted form of energy that can be used to power many common devices and components.

Fission occurs when an incident neutron collides with the nucleus of a heavy target atom, such that the neutron enters the nucleus. This results in excitement of the nucleus to a higher energy level. Fission occurs once the excitation energy exceeds the critical energy level required for binding a nucleus. Consequently, the atom's nucleus splits (fissions) into (two) smaller fragments (i.e., different atoms), because of the high energy level, as shown in EQUATION 1.

In addition to the fission fragments, neutrons are released (from the fission reaction) that are not constrained to a nucleus. Much of the energy is released in the form of fission fragment kinetic energy and radiation. In a light water reactor, the stray neutrons (fast neutrons) from a fission reaction are slowed down with the help of a moderator (water) to become thermal neutrons. Thermal neutrons have very little kinetic energy and are essentially in thermal equilibrium with the surrounding material. This is desired, since a thermal neutron is much more easily captured by another nucleus than a fast neutron, which tends to bounce off a nucleus rather than be absorbed. Different fission fragments are possible depending on the type of atom being split. Shown below is a typical fission reaction [35]:

$${}_{0}^{1}n + {}_{92}^{235}U \rightarrow \left({}_{92}^{236}U\right)^{*} \rightarrow {}_{55}^{140}Cs + {}_{37}^{93}Rb + 3\left({}_{0}^{1}n\right)$$

EQUATION 1

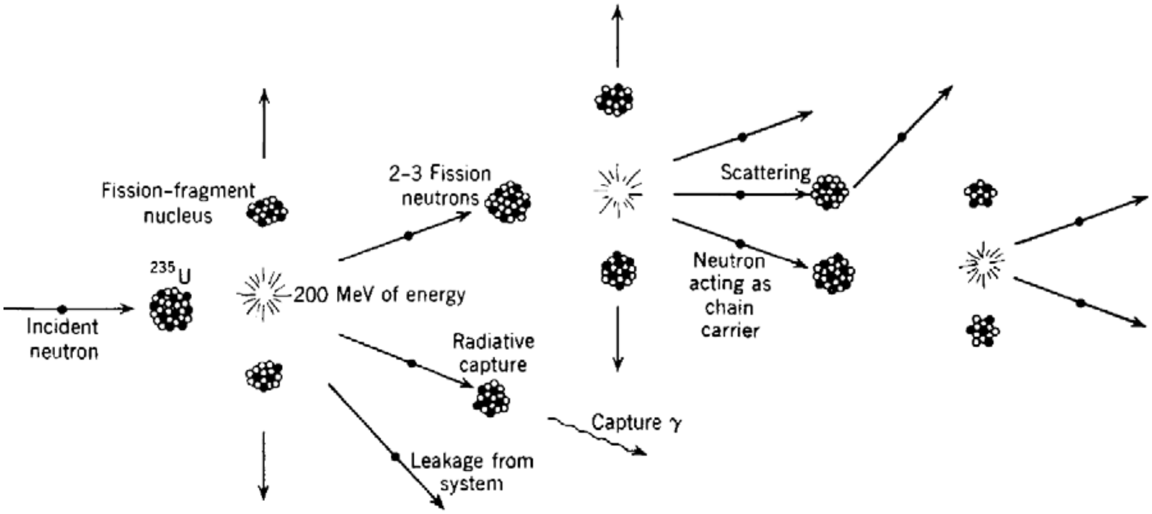


FIGURE 1: Nuclear fission chain reaction overview [34]

The two most common types of Nuclear Power Plant (NPP) fission reactors are Pressurized Water Reactors (PWRs) and Boiling Water Reactors (BWRs), with PWRs being more prevalent and the focus of this chapter, as it introduces the concept of a pressurized cooling system. FIGURE 2 provides a high level functional schematic of a PWR (adapted from [33]). A PWR consists of two separately enclosed loops: a primary cooling loop and a secondary steam loop. The secondary loop consists of two phases. The primary loop includes a reactor, primary loop pump, pressurizer, and steam generator. The primary loop is intended to be a single phase liquid. The most common PWRs are called light water reactors, since they use demineralized water as the moderator and coolant. At atmospheric pressure the primary coolant (water) would boil, because of the high temperatures generated in the reactor during normal operation. However, boiling must be avoided within the reactor of a PWR. Therefore, the primary loop is controlled under pressure to maintain liquid phase of the coolant. To maintain the pressure and act as a thermal-hydraulic damper, a pressurizer is added to the primary loop. Additional detail about the design and function of a typical pressurizer is provided in Section 1.2.

Steam is created in the secondary loop inside the steam generator, as heat is transferred from the primary loop to the secondary loop. This steam travels through high and low pressure turbines to extract the thermal energy and transform it into mechanical rotational energy. The turbines are connected to a shaft that rotates a generator, which converts the mechanical rotational energy into electricity that is transmitted and distributed via the power grid to load sources, such as homes and businesses, with the balance of plant components (e.g., turbine, condenser, moisture separator and reheater "MSR", etc.).

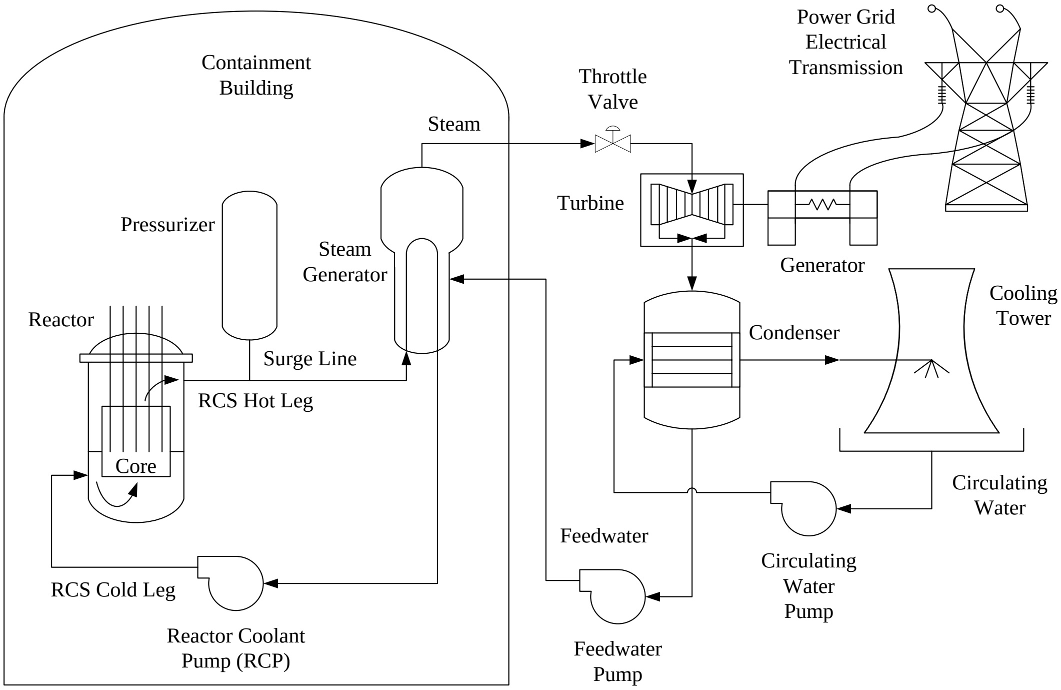


FIGURE 2: PWR NPP functional schematic

1.2 Pressure Control for a Pressurized Cooling System

Section 1.2 provides a background overview of the typical pressure control strategy commonly used in pressurized water cooling systems (e.g., PWR NPP)

A pressurizer is a large vessel partially filled with water at the bottom and a steam bubble at the top. FIGURE 3 provides a schematic of a typical pressurizer [15] and [74]. The PZR is maintained at saturation [14]. As the cooling system temperature increases, the density of the water decreases causing it to expand (for all other parts of the fluid system being in equilibrium) forcing liquid into the PZR. Then, the PZR liquid level increases, compressing the steam and increasing the pressure. Therefore, an increase in temperature results in an increase in pressure and a rise in level. Oppositely, as the cooling system temperature decreases, the density of the water increases causing it to contract; this results in a decrease of pressure and a reduction in level within the pressurizer. If the system temperature changes result in pressure changes that are faster than the pressurizer's normal saturation response can handle, heaters (providing added heat at the bottom of the pressurizer) and sprayers (providing sub-cooled water at the top of the pressures) are added to decrease the pressurizer's response time to a system change. This allows the pressure in the vessel to be regulated by adding heat through the heaters into the water or by condensing steam through spraying sub-cooled water into the steam bubble. The bottom of the pressurizer is attached to a fluid system, and maintains the fluid system at the equilibrium pressure in the pressurizer vessel.

There are two important process variables that are typically controlled in a PZR [14]:

1) The pressure of the steam/water system
2) The level of the water in the vessel

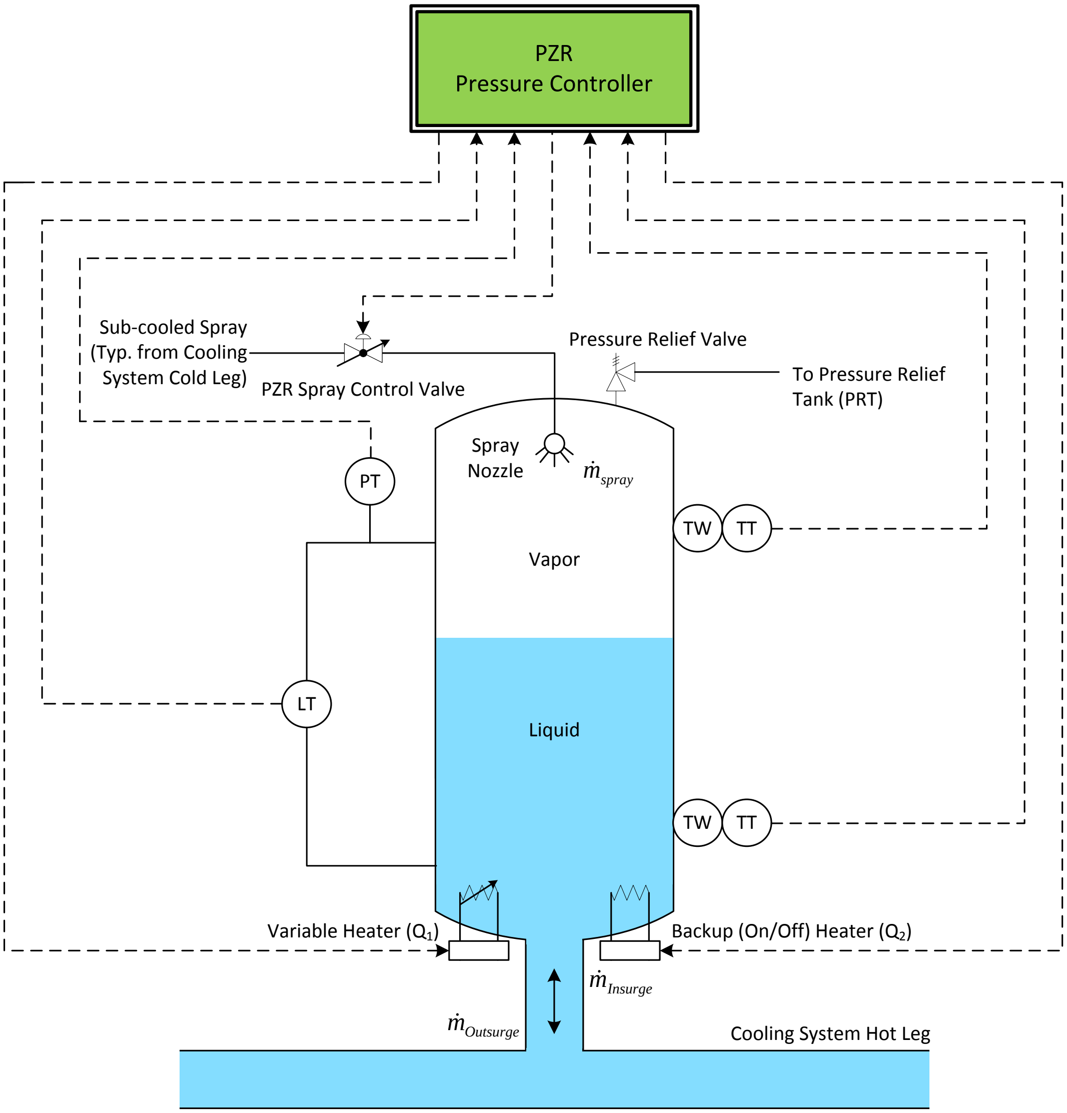


FIGURE 3: Typical pressurizer schematic

However, all other parts of the fluid system are seldom in equilibrium. Changes in temperature of the larger fluid system to which the pressurizer is attached will cause the liquid in the system to shrink and swell, varying the amount of fluid in the pressurizer and thus the water level. Sudden breaks in the fluid system piping can cause large, quick losses of fluid inventory, resulting in pressurizer level dropping quickly. Smaller piping leaks can result in slower drops in pressurizer level, but can also result in pressurizer drops that could cause more boiling in the pressurizer, compensating for or actually reversing pressurizer level changes.

Pressurizer pressure control is important, since the pressure derived from the pressurizer prevents liquid in the larger fluid system from boiling while operating at temperatures that would be above the fluid boiling point at atmospheric pressure. Therefore, sudden drops in pressurizer pressure can indicate a large break in the fluid system pipe. Excursions in the pressurizer pressure can result in loss of fluid inventory because of the necessity of venting the pressurizer to a relief tank or the atmosphere.

1.3 Recent Advances in Pressurized Cooling System Pressure Control

Section 1.3 provides a survey of the current research advances for PWR PZR Pressure Control (e.g., conference proceedings, journals publications, etc.), in order to identify what areas have been investigated in the field of PZR control and what areas might benefit from additional research. Included below is a summary of the research that has recently been conducted for investigating PWR PZR pressure control and closely related topics (e.g., PZR level control, general NPP system control, and NPP and PZR system modeling). As observed from the list below, much of the PZR pressure control research has focused on application of intelligent techniques (e.g., ANN) with adaptive controllers (e.g., self-tuning PID controller). However, there is less research identified for implementation of optimal controllers or dynamic controllers (e.g., MPC). Details of the proposed research to expand understanding of PZR pressure control, specifically for a pulsed cooling system, are discussed in CHAPTER 6.

1) PZR Pressure Control:

    - Adaptive fuzzy control of PZR pressure, 2005 [48]
    - Fuzzy logic control of PZR system in PWR NPP, 2010 [77] and 2014 [50]
    - Artificial intelligence techniques (e.g., artificial neural network) for modeling and control of NPP PZR system pressure, 2013 [47]
    - Artificial Neural Network (ANN) based PID control of PZR pressure in PWR NPP, 2013 [49]
    - Back-propagation (BP) Neural Network Control of Self-Adjusted PID Parameters for PZR pressure, 2013 [76]

2) PZR Level Control:

    - Radial Basis Function (RFB) Neural Network and PID control of PZR level, 2010 [28]
    - Fuzzy-PID control of PZR level, 2011 [27]

3) General Nuclear Power Plant (NPP) System Control:

    - H_∞ control of a PWR NPP plant using a lower order MIMO state space system model obtained via identification from experimental data, 1994 [44]
    - Optimal linear parameter-varying (LPV) control for a PWR, 1995 [45]
    - Linear parameter-varying (LPV) control for a PWR, 1996 [46]
    - Model Predictive Control (MPC) of dynamics in a PWR NPP primary cooling system, 2009 [26]
    - Controller for Supervision, Control, and Protection Systems in PWR NPPs, 2011 [54]

4) NPP and PZR System Modeling:

- PWR PZR modeling and response during transients, 1982-1987 [60], [61], [62], [63], [64], and [65]
- Dynamic modeling of PZR surge tank transients, 2005 [59]
- Non-equilibrium PZR model with non-condensable gas, 2006 [58]
- PZR model identification of an identified NPP system model, 2006 [56]
- PZR benchmark models using TRACE code, 2009 [53]
- Coupling RELAP5 and MATLAB-Simulink for NPP modeling and simulation, 2010 [29]
- ANN model for PZR in sensitivity studies, 2011 [51] and 2013 [52]
- PZR pressure control system mathematical modeling, 2012 [55]
- Control-oriented model for PZR transient dynamics, 2013 [39]
- Two-region and four-region PZR models with simulations, 2014 [57]

1.4 Fusion Reactor (with a Pressurized Cooling System) Overview

A comprehensive understanding of a process is necessary to properly design and implement control of that process. The focus of this research involves pressure control for a pulsed cooling system. Therefore, in order to best control pressure of a pulsed cooling system, an understanding of the process that typically creates this behavior is necessary. Pressurizers have commonly been used to control pressure in PWR NPPs. This basic pressurized cooling system design is also utilized in the primary cooling water system of a tokamak style nuclear fusion reactor. Therefore, this section provides an introduction to nuclear fusion technology, as it relates to power generation.

Nuclear fusion involves the joining of atoms (see EQUATION 2), instead of splitting atoms as in a nuclear fission reaction [1], [2], [3]. Fusion is actually very common in the universe, since it occurs in the core of the Sun and stars [4]. Nuclear fusion of hydrogen isotopes deuterium ($^{2}_{1}$H) and tritium ($^{3}_{1}$H) requires temperatures around 10 times higher than the temperature required for the hydrogen-hydrogen fusion reaction occuring at the Sun's core [5], as seen in FIGURE 5 (which is borrowed from [6]). At such high temperatures plasma is formed. The $^{2}_{1}$H-$^{3}_{1}$H fusion reaction is shown in FIGURE 4.

$$^{2}_{1}H + ^{3}_{1}H \rightarrow ^{4}_{2}He(3.5MeV) + ^{1}_{0}n(14.1MeV)$$

EQUATION 2

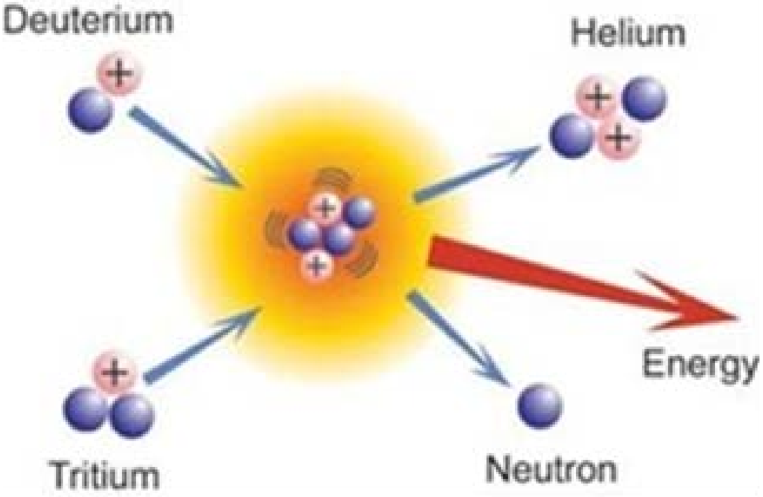
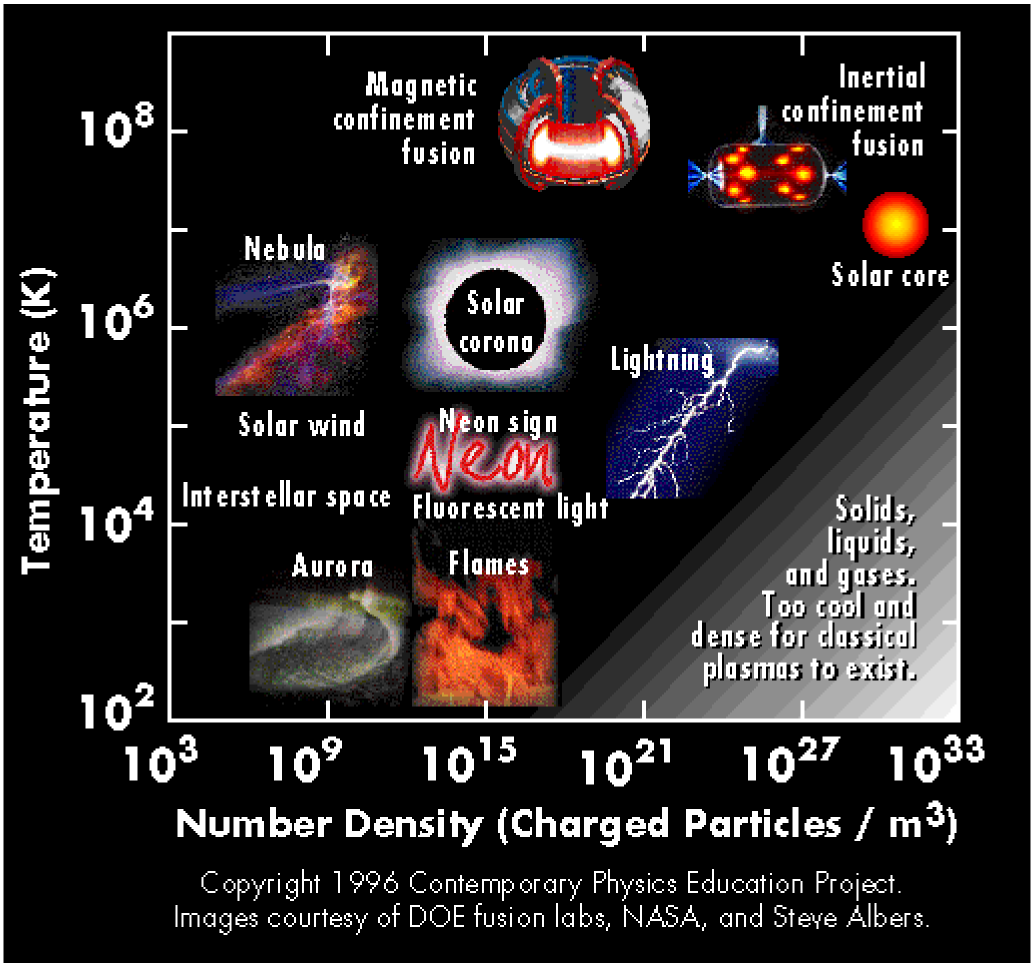


FIGURE 4: Deuterium-tritium fusion reaction [5]



Copyright 1996 Contemporary Physics Education Project.
Images courtesy of DOE fusion labs, NASA, and Steve Albers.

FIGURE 5: Plasma plot of temperature vs. number density [6]

Deuterium and Tritium nuclei can be collided directly in an accelerator or randomly in high temperature plasma. There are three primary methods of plasma confinement (see FIGURE 6): gravitational confinement, inertial confinement, and magnetic confinement [4]. The Sun uses gravity to contain plasma. However, it is not possible to use gravitational confinement of plasma on Earth. Inertial confinement involves the compression of a hydrogen pellet with lasers of very high power. Plasma is produced for a very short time (μs), but many reactions are produced quickly [4]. “In magnetic confinement the particles and energy of a hot plasma are held in place using magnetic fields. A charged particle in a magnetic field experiences a Lorentz force” [8]. The equation for the Electromagnetic Force “Lorentz Force Law” is shown below (see EQUATION 3) [4] and [10]:

$$\vec{F} = q\vec{V} \times \vec{B}$$

EQUATION 3

Where,

F is the force (in Newtons)

B is the magnetic field (in Teslas)

q is the electric charge (in Coulombs)

V is the instantaneous velocity (in m/s)

× is the vector cross product

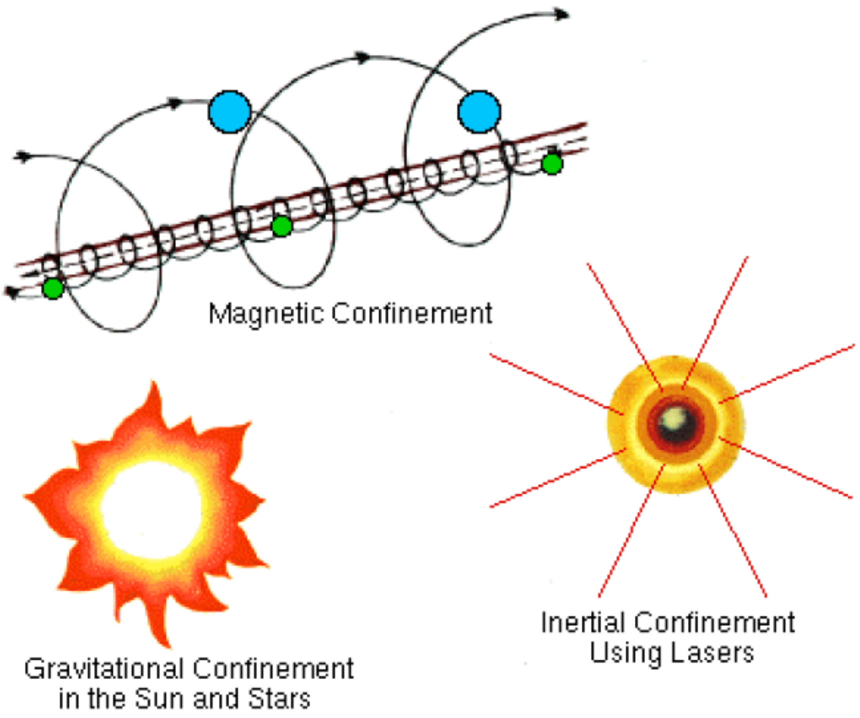


FIGURE 6: Three methods of plasma confinement [7]

To further understand magnetic confinement, plasma is charged particles [3]. The charged particles are forced to follow helical paths around magnetic field lines [5]. Compared with inertial confinement, fewer reactions per second are achieved with magnetic confinement. However, a greater amount of energy is released, since it occurs over a longer period of time [4]. For example, the ITER toroidal field magnets produce an 11.8 Tesla magnetic field and the central solenoid is designed to achieve a magnetic field of 13 Tesla at a coil temperature of 4 Kelvin [5].

A device called a tokamak is used to confine and control the plasma. The tokamak accomplishes confinement and control of the plasma by the use of magnetic fields [11]. A typical magnet configuration for tokamak is shown in FIGURE 7 and in FIGURE 8.

FIGURE 7 schematically displays the inner poloidal field coils (central solenoid), outer poloidal field coils, poloidal magnetic field, toroidal field coils, plasma electric current, and resulting helical magnetic field. FIGURE 8 displays an image from the JET Tokamak before (on the left side) and during (on the right side) operation [5]. ITER also utilizes a tokamak for magnetic confinement of the high temperature plasma [5].

Most of the magnetic field produced by the tokamak is contained within the intended magnetic circuit. However, there are stray magnetic fields produced that spread throughout the Tokamak building complex, as seen in FIGURE 9 and FIGURE 10. The stray magnetic fields produced are low frequency (0 to 50 Hz) and could be as large as 150 mT to 200 mT in some areas of the Tokamak building [13]. This becomes relevant when considering environmental capability of process components and instruments, which may inhibit some technologies or necessitate appropriate magnetic shielding, which would not be a constraint in typical nuclear fission power plants.

Tokamak type fusion reactors "generate their magnetic field cage partly with external magnet coils that encompass the plasma vessel. The rest is produced by an electric current flowing in the plasma that is induced there in pulsed mode by a transformer. Without auxiliary facilities tokamaks can therefore only operate in pulsed mode." [102] "The plasma current is normally induced by a transformer coil. This is why a tokamak does not operate in the steady state, but in a pulsed mode. In a transformer it is only for a limited time that an increasing current can be generated in the primary winding so that a current can be driven in the plasma. The transformer must then be discharged and the current started up afresh." [103] It is possible for other types of fusion reactors (e.g., stellarator) to operate in continuous mode, since they function without plasma current, but instead use magnetic fields generated exclusively with external coils. However, the stellarator magnetic coils are significantly more complex than those used in tokamaks. [102] Therefore, tokamaks are the primary focus for this investigation.

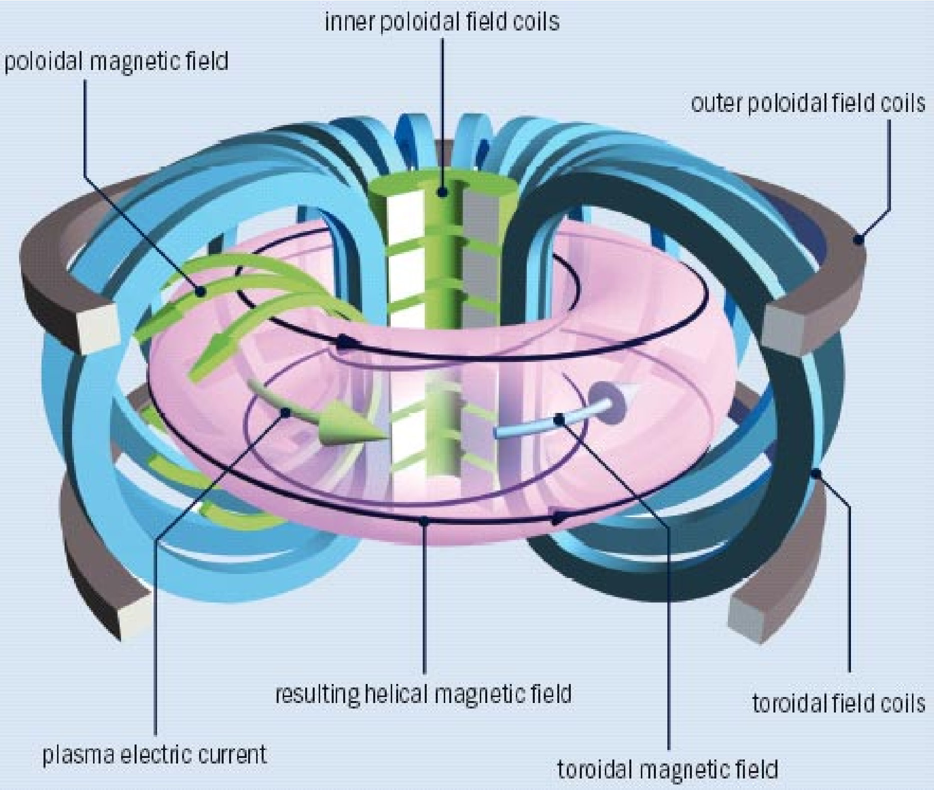


FIGURE 7: Tokamak magnetic field [11]

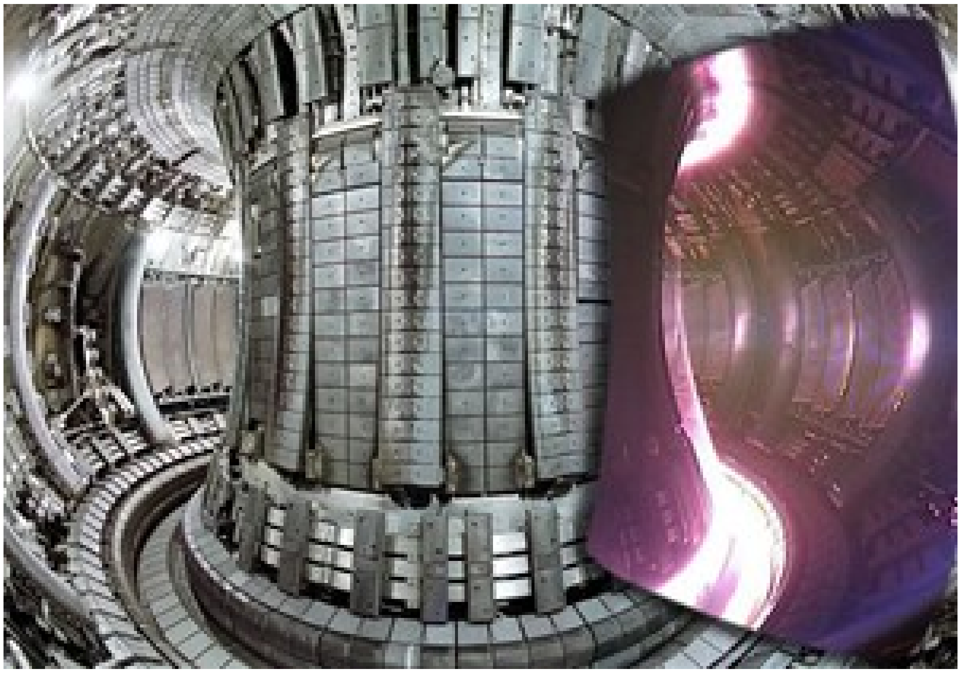


FIGURE 8: JET tokamak before and during operation [5]

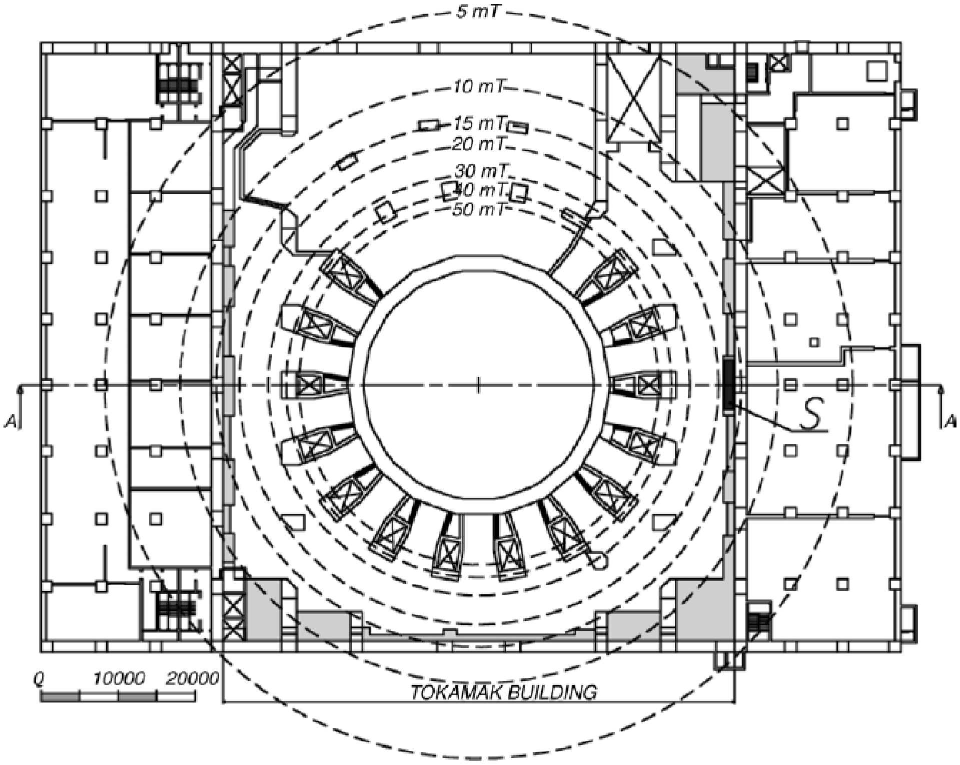


FIGURE 9: Magnetic field map in the tokamak building and main axis [13]

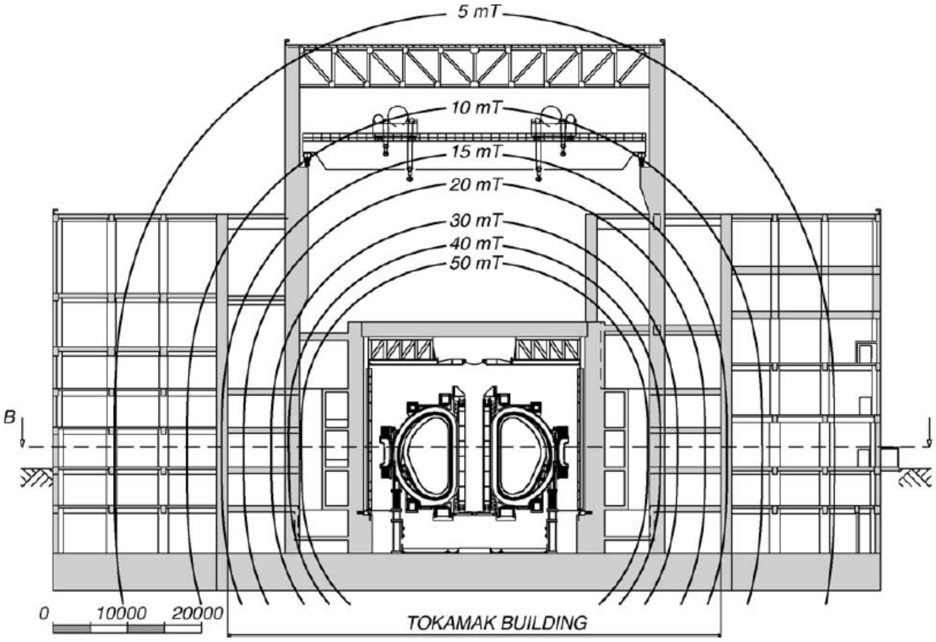


FIGURE 10: Magnetic field map at the equatorial level of the tokamak [13]

1.5 Pressure Control Challenges for Pulsed Pressurized Cooling System

Pressurizer behavior is well understood in typical pressurized water reactor (PWR) nuclear power plants (NPP) [1], [2], [15], [16], [17], [18]. Pressure and level control are relatively straightforward, because the larger fluid system of which the pressurizer is a part operates normally in a very steady state.

However, the cooling system for a tokamak style fusion reactor [2], [4] such as ITER is different [22], [23], [24], [25]. During normal operation a fusion reactor's plasma is pulsed, instead of staying at a constant value, which results in temperature swings between the plasma pulses, as depicted in FIGURE 11. This design characteristic means that:

1) Coolant temperatures fluctuate over a larger range during steady state operation than typical PWRs experience [17], [22]
2) Fluid volume also varies as the coolant temperature changes, since fluid (water) density is a function of temperature [22], [14]

These changes cause wide swings in pressurizer level and pressure [22] , as shown in FIGURE 12. Letdown and makeup of the fluid system are important for maintaining chemistry, and they are also affected by pressurizer level. Because of the shrink and swell of the coolant, the temperature of the water in the pressurizer varies as water enters and exits the bottom of the pressurizer. This temperature change in the pressurizer further complicates level and pressure control in the vessel.

There are both (a) safety and (b) operational control functions typically associated with the pressurizer (e.g., pressure and level). When the coolant is at a steady state, the operational control and safety trip points can be kept sufficiently far apart to avoid interactions. However, the shrink and swell effects of the coolant may force interaction between the pressurizer (a) safety and (b) control functions. Therefore, the amount of primary coolant volume fluctuation resulting from the pulsed power temperature changes must be accommodated by the pressurizer while maintaining pressure within an acceptable tolerance band [22]. Keeping the operating pressure below the setpoint limits is critical (see Sections 2.2 and 3.3 for the setpoints used on the test case PZR for this research).

Additionally, as mentioned, the cyclic pulses of the cooling system result in insurge and outsurge of coolant from the pressurizer. These pressurizer surge pulses cause an increase in cyclic stresses, which result in fatigue of all associated materials and components. Fatigue will shorten component life and possibly result in component failure, if not addressed. In fact, fatigue is one of the predominant failure mechanisms in nuclear power plants. Consequently, plants usually have fatigue monitoring systems or programs to identify when a component should be replaced before failure occurs. For components in a tokamak cooling system, the pressure pulses can potentially be one of the primary drivers of the cyclic fatigue. The cooling system pulses cannot be eliminated in this system, since the plasma pulses are intrinsic to the tokamak system design (i.e., the cyclic frequency is fixed). However, it may be possible to reduce the magnitude of the pressure pulses, which will reduce cyclic fatigue of components (i.e., reducing the cyclic stress magnitude will allow the component to endure more cycles before failure [66]).

The costs associated with replacing a typical PZR are significant in relation to value of most plants. Replacement of a PZR requires careful planning and execution to ensure safe and reliable operation and maintain commercial profitability. Failure to replace major components properly can result in a utility deciding to retire or close a plant, because of safety and financial profitability reasons, as was the case with the faulty steam generator replacement for San Onofre Nuclear Generating Station (SONGS) owned by Southern California Edison [104]. As NPPs are aging, the market demand for reactor PZR replacement is increasing. It is anticipated that "year-on-year demand for the reactor pressurizer around the globe will increase from 19 units in 2010 to 26 units in 2020, showing an average annual growth rate (AAGR) of 3.1% during this period. This market is expected to reach $338m in 2020, increasing at an AAGR of 7.9% between 2010 and 2020." [105] The cost to replace other large components, such as steam generators, is similar in magnitude to that for PZR replacement. For example, to build four steam generators to replace those in the Watts Bar Nuclear Plant Unit 2 reactor (which is located near Spring City, Tennessee), TVA directors voted to pay $160 million to Westinghouse [107]. Therefore, each steam generator costs approximately $40 million.

However, the cost associated with smaller components on the PZR (e.g., PZR spray valve) is considerably smaller in comparison. Therefore, it might be prudent to allow the smaller components to be used sacrificially, if doing so reduces overall the costs. For example, increasing the modulation frequency on a PZR spray value will reduce its operating life (i.e., wear it out sooner). However, increasing the modulation of the PZR spray valve may reduce the operating costs of the PZR (e.g., avoid interaction with safety functions) and possibly increase the PZR life (e.g., reduced stresses on the PZR).

As an example cost comparison, the Target Rock Modulating Pressurizer Spray Valve from Curtiss-Wright is approximately $460,000 per unit [108]. The PZR spray valves are typically low maintenance. For example, they are typically tested for over

100,000 cycles with borated water ([108]). The tokamak cooling system will experience one plasma pulse cycle every 1800 seconds with an expected plant operating life of 60 years. For this pulsed cooling system, a maximum of 17520 cycles are anticipated per year. Considering the stated product testing, the PZR spray value should provide maintenance free service for over 5 years. Accordingly, the PZR cost is approximately 28 times greater than the PZR spray valve cost, as shown in TABLE 1. The cumulative cost to replace the PZR spray valve over the life of the plant would still be less than the cost to replace the PZR (i.e., for 60 year plant life the PZR spray value would be replaced about 12 times, which would cost approximately $5.5M). Therefore, it would be prudent to use the PZR spray valve sacrificially to reduce the magnitude of the pressure pulses in the cooling system.

TABLE 1: Range of magnitude cost estimates for NPP components

Item	**Cost / One Item**
Pressurizer	$13M
Steam Generator	$40M
Pressurizer Spray Valve	$0.46M

While not reasonably measurable for this simulation research, since there are many specific factors that influence PZR design life (which are unique to each plant design), the reduction in pressure spike during cyclic transient cooling system pulses will result in less fatigue of the PZR vessel and components (i.e., less stress). Reducing the cyclic stress will increase the number of cycles the PZR can withstand and hence, could extend the design life, as shown in FIGURE 13. A specific quantitative measure of design life for each plant is not reasonably achievable, since it will vary between plants. However, the fatigue has been qualitatively assessed with an assumed general quantitative evaluation. Based on the design life criteria provided in Regulatory Guide 1.207 [80] and NUREG/CR-6909 [81] (using the generic PZR parameters for this research and the target control performance compared with the conventional control performance, see Sections 2.2 and 3.3), it is estimated that the number of cycles permitted at maximum pressure (4.28E+9 cycles) is much greater than the anticipated number of cycles in the plant life of 60 years (1.05E+6 cycles). Therefore, fatigue from plasma cycles is not considered a limiting criterion with the test case PZR design. While fatigue is a concern for the stated reasons, it is fortunately not as strong a concern for the typical simulation PZR design used with this research, since the anticipated number of thermal cycles is much less than the allowable number of cycles at the current pressure fluctuation imposed.

As well, the U.S. Nuclear Regulatory Commission (NRC) uses PRA models to “look at the frequency and consequences of a nuclear power plant not achieving a safe, stable end-state” [78]. A plant that transitions to a safe and stable end-state for a specified period of time is the desired response [78]. PRA models are complex and include numerous inputs (e.g., plant design, operation data, thermal-hydraulic analysis, etc.) [78]. As such, plant PRA might be improved by keeping the pressure during normal operating conditions further away from the safety set-points (i.e., avoiding possible interaction of safety and non-safety functions) and by reducing the amount of fatigue seen by the PZR and associated components from the cyclic loading stresses of the pulsed cooling system. A quantitative value of this cost savings (to relate an incremental reduction in pressure to a corresponding reduction in cost) is not realistically feasible, because of the numerous input factors and complex (non-linear) relationships. However, qualitatively it is anticipated that a reduction in pressure would correlate to a reduction in cost for the reasons mentioned in this section.

Therefore, a pressurizer control design that reduces the magnitude of pressure pulses is desirable (i.e., avoid exceeding safety setpoints and reduction in fatigue). Consequently, a suitable design solution for pressurizer control is necessary to support development of this safe, clean, CO2-free, and inexhaustible energy source for the future.

This project evaluated and analyzed the selected control design approach for this problem. In PWRs, pressurizer control is typically accomplished with an on/off and proportional-integral-derivative (PID) control strategy [17]. However, this approach alone may not meet the desired control performance for a tokamak cooling system, considering the varying process conditions described. Advanced control strategies (e.g., adaptive control, intelligent control such as neural networks, optimal control with state estimation or model predictive control, etc.) [19], [20], [21], [26], [27], [28] are suitable to investigate and analyze for applicability to this problem.

Therefore, it is important to investigate:

1) If traditional PID feedback control (i.e., reacting to something that already occurred) will provide sufficient control performance.
2) If employing a single advanced control method or combination/hybridization (e.g., multiple adaptive control methods) can improve control performance.

The pressurizer control design should satisfy the operational control requirements and fulfill the safety control functions without the two interfering with each other. There are different ways to approach a solution for this problem. Physical testing is not feasible. One applicable alternative (apart from physical testing) is to model the pressurizer using bounding conditions with the expected coolant fluctuations and run various control scenarios, as discussed above. As such, appropriate modeling and analysis tools (e.g., MATLAB/Simulink with thermal-hydraulic safety analysis software [29]) are necessary to research this problem.

Therefore, a dynamic model based control design approach that permits modification online as the process dynamics change is proposed to explore this control problem. A new method (i.e., called LiMeRICK) is presented that uniquely combines a hybrid control technique (i.e., called LiMe) with a technique to improve system knowledge (i.e., called RICK). The RICK method (used to improve the pressurizer system knowledge) is based on Recursive Least Squares Identification (RLS ID), a Kalman filter, and a model corrector. The LiMe technique (used to improve pressurizer pressure control) is a hybrid control architecture that combines a Linear Quadratic Regulator (LQR) and Minimum Variance (MV) controller. Combining these two new methods (i.e., LiMe and RICK) enables enhanced control during transient conditions.

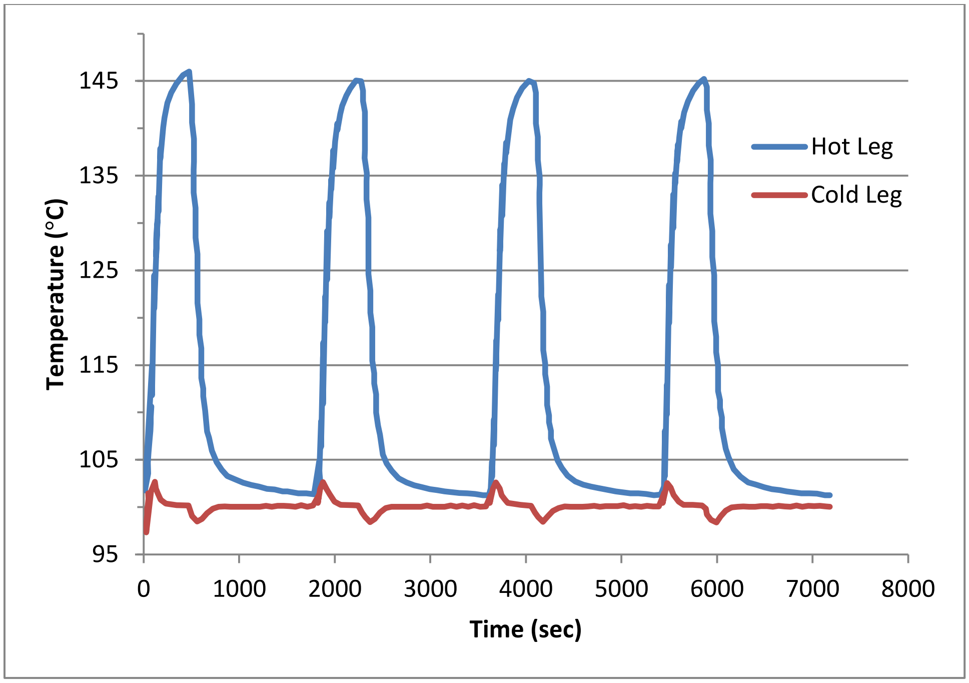


FIGURE 11: Typical primary coolant temperatures during plasma pulses [22]

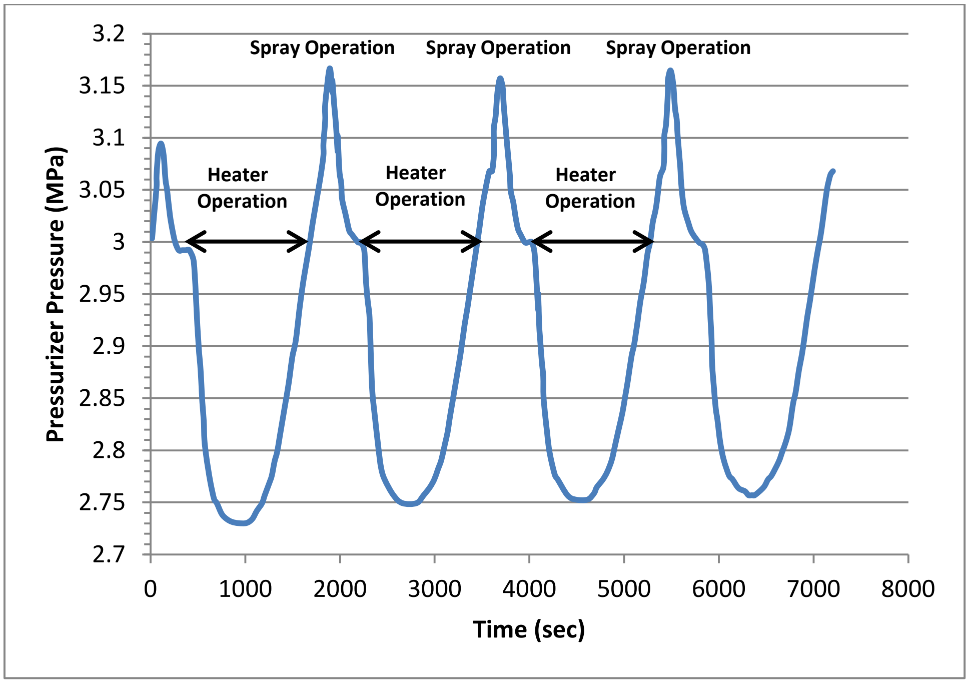


FIGURE 12: Typical primary coolant system pressures during plasma pulses [22]

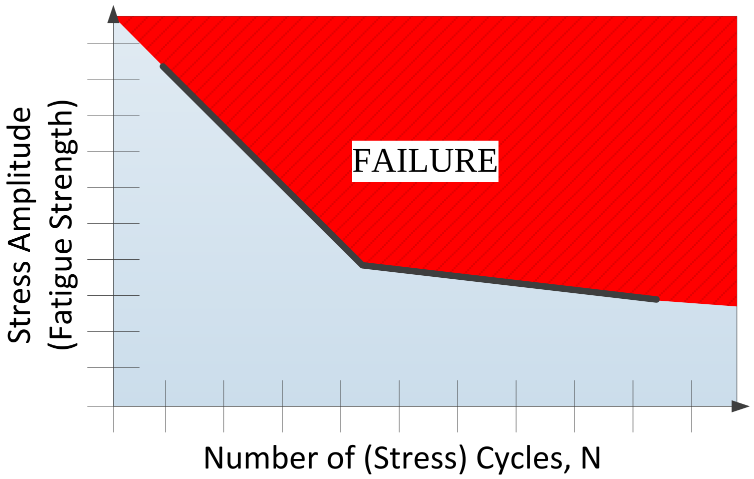


FIGURE 13: Example S-N fatigue diagram of fatigue test

1.6 Research Activities to Address Identified Challenges

FIGURE 14 provides an activity flow chart for the research plan that was used to address the challenges identified with PZR pressure control in a pulsed system. The initial activity was to survey the literature of what research has been conducted in the topic areas related to PZR pressure control. Next, system modeling was conducted to identify both simplified and dynamic models of the PZR thermal-hydraulic system. Then, existing control methods were applied for PZR pressure control with both the reduced order models (see CHAPTER 4) and dynamic PZR model (see CHAPTER 5). Finally, a proposed new control method was developed, which is presented in CHAPTER 6.

In summary, the proposed research intends to demonstrate (via modeling and simulation) that advanced control methods can provide improved control performance over conventional control methods for control of PZR pressure in a pulsed cooling system.

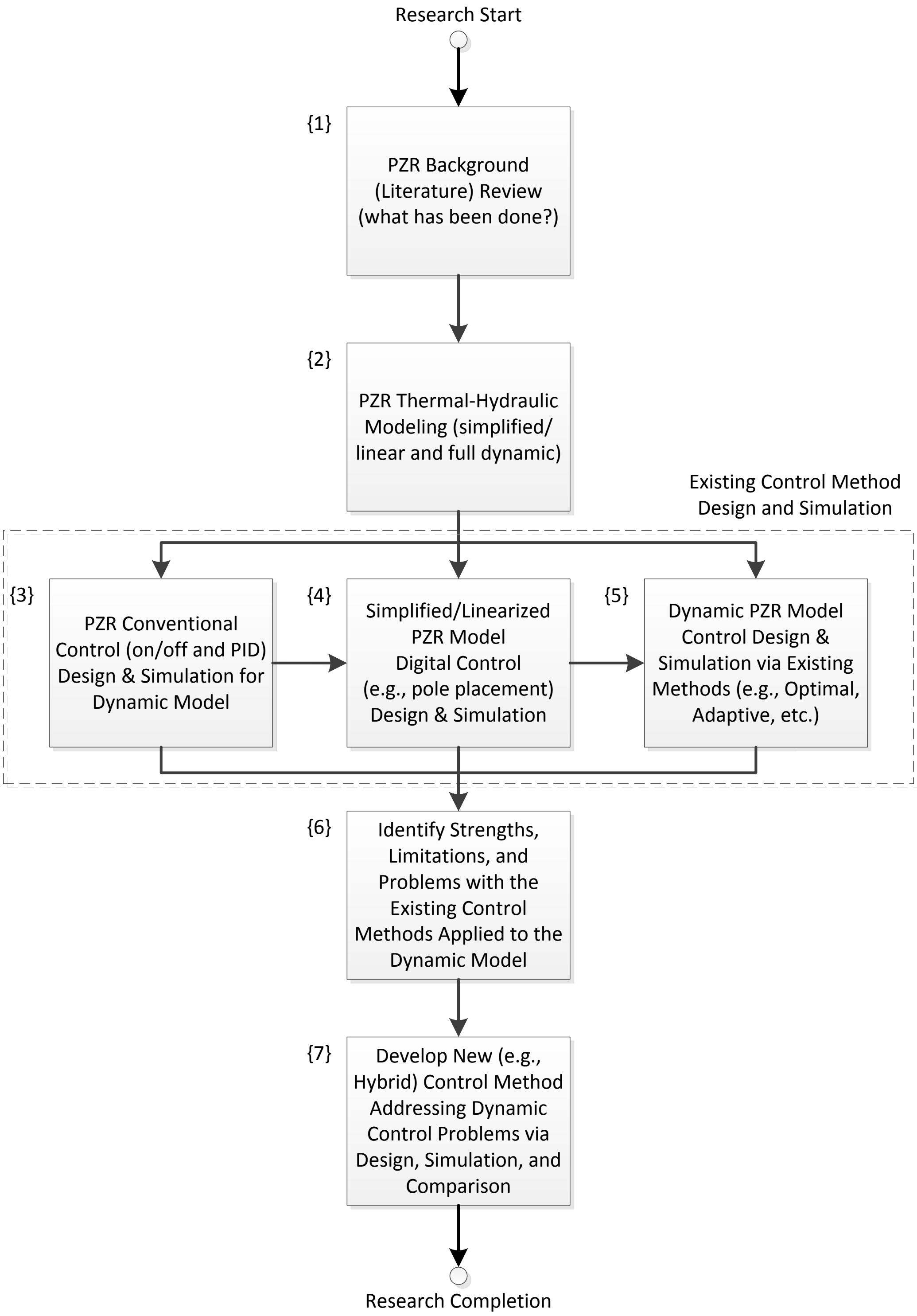

Research Start
{1}
PZR Background (Literature) Review (what has been done?)
{2}
PZR Thermal-Hydraulic Modeling (simplified/ linear and full dynamic)
Existing Control Method Design and Simulation
{3}
PZR Conventional Control (on/off and PID) Design & Simulation for Dynamic Model
{4}
Simplified/Linearized PZR Model Digital Control (e.g., pole placement) Design & Simulation
{5}
Dynamic PZR Model Control Design & Simulation via Existing Methods (e.g., Optimal, Adaptive, etc.)
{6}
Identify Strengths, Limitations, and Problems with the Existing Control Methods Applied to the Dynamic Model
{7}
Develop New (e.g., Hybrid) Control Method Addressing Dynamic Control Problems via Design, Simulation, and Comparison
Research Completion

FIGURE 14: Pressurizer control research plan

1.7 Research Contributions

Section 1.7 (TABLE 2) describes the contributions of this research, highlighting the value added to each topic area.

TABLE 2: Research contributions

Contribution Topic Area	**Method Name and Description**	**Application Examples**	**Sec.**
Pressurizer Modeling for Control Simulation	PZR Models Identified (SISO and MIMO) for Adaptive Control Simulation	Simulate the dynamic PZR behavior with advanced (adaptive) control methods using the identified existing PZR models.	2.1-2.5
PZR Dynamic Simulation Disturbance Inputs	PZR Inputs for Mass Surge Flow Rate	Simulate the PZR behavior with different disturbance inputs for PZR mass surge flow rate.	2.2
Adaptive-Optimal Control	Dynamic "K" (Gain Matrix) and "B" (State Space Input Matrix) Optimal Control	Improve optimal (LQR) control of PZR pressure over that of LQR with a static "K" matrix.	5.3.1-5.3.4
	Dynamic "K" Optimal Control with RSL ID	Improve optimal (LQR) control of PZR pressure over that of LQR with a static "K" matrix.	5.3.5
Adaptive-Dynamic Control	LiMe (LiMeRICK): Hybrid control technique that uniquely combines LQR and MV Control	1) Reduce the controlled variable (e.g., pressure) fluctuation range for improved regulator performance to maintain the reference value. 2) Reduce the controller/regulator response time to minimize duration of the process value exceeding the allowable reference range.	6.3, 6.5, and 6.6
System Knowledge (signal processing and model identification)	RICK: Technique to improve system knowledge that uniquely combines an RLS ID, a state space based model corrector, and a Kalman filer.	1) Produce a more accurate adaptive system model. 2) Produce a better estimate of process values.	6.4, 6.5, and 6.6

CHAPTER 2: PRESSURIZER MODELING

A mathematical model of a PZR must be established before any control methods can be applied. Therefore, this chapter focuses on (1) development of simplified PZR models that will be used to understand basic system behavior and provide a suitable (state space or transfer function) form of an estimated PZR system model to facilitate implementation of some control methods (e.g., optimal control) and (2) introduction of a suitable control oriented dynamic PZR model that was previously developed [39].

First, this chapter introduces the governing differential equations to model pressurizer behavior, as discussed in section 2.1. Second, the benchmark PZR design (e.g., PZR diameter and height, etc.) and process conditions (e.g., experimental insurge/outsurge data to accurately simulate system behavior) utilized for this research are provided in section 2.2. Then, two simplified generic pressurizer models are presented that are used to develop a generalized solution: section 2.3 introduces a reduced order SISO PZR model and section 2.4 introduces a reduced order MISO PZR model. Last, the prospect for a suitable control-oriented dynamic thermal-hydraulic pressurizer model to simulate control strategies is discussed in section 2.5.

2.1 Differential Equations Governing PZR Behavior

There has been much research conducted on understanding PZR behavior and developing PZR models [56], [57], [58], [59], [60], [61], [62], [63], [64], [65]. During the 1980s several research projects were conducted to develop analytic models [60], [61], [62], [63], [64], [65]. This research has continued as an active area of research [39], [56],

[57], [58], [59], as more accurate models are desired to describe the behavior during normal operation and transient conditions.

The PZR model for transient analysis uses the controlled volume approach, as discussed in Todreas and Kazimi [15]. To setup the energy equations for the PZR control volume analysis:

- The PZR is considered the control volume
- There are two “gates” for flow into or out of the PZR, which are (1) the surge line and (2) the spray nozzle
- Heat flow (dQ/dt) occurs via (1) electric heaters and (2) conduction through the PZR vessel walls
- The PZR vessel volume is considered non-deformable and stationary.

As previously discussed, the PZR spray nozzle injects subcooled water (typically from the cold leg of a cooling system) into the upper steam region of the PZR upon an increase in pressure above a predetermined set-point, which is typically the result of an insurge. The subcooled water acts to condense some of the steam, resulting in a reduction in pressure. Similarly, heaters in the lower region of the PZR are used to add heat upon a decrease in pressure below a predetermined set-point, which is typically the result of an outsurge. The added heat to the PZR results in some of the saturated water in the PZR flashing to steam, which acts to increase the pressure (since steam occupies more volume than water for a given mass).

For an equilibrium single-region PZR model, as seen in FIGURE 15, the general transient mass, energy, and volume conservation equations are presented below. A single region model assumes that the system can be reasonable modeled as a single volume (i.e., the entire PZR volume is treated as a single unit) with two thermodynamic states (i.e., liquid and vapor).

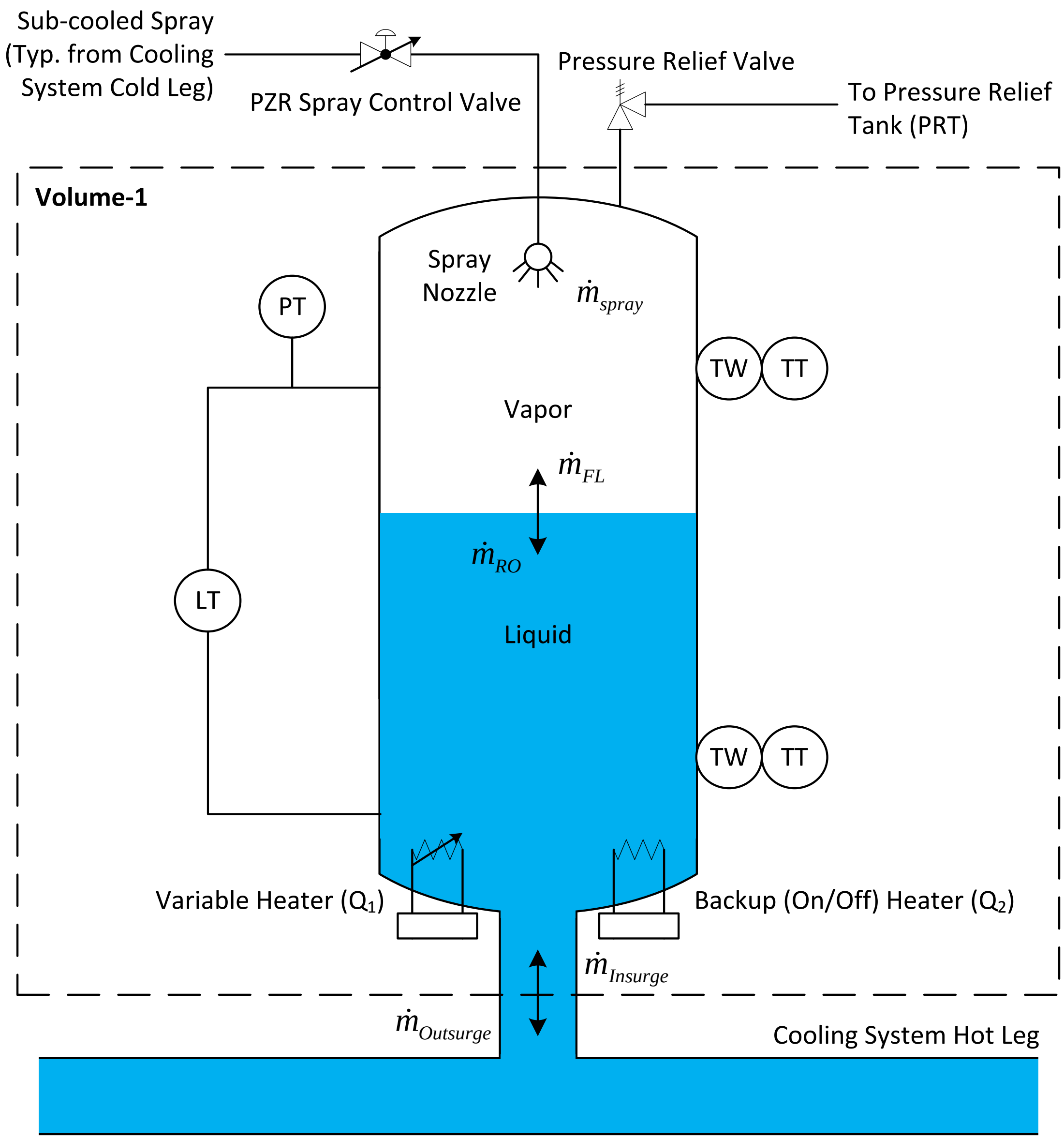


FIGURE 15: Typical 2-region (liquid & vapor) 1-volume PZR [39]

$$\frac{d}{dt}m = \dot{m}_{surge} + \dot{m}_{spray}$$

EQUATION 4

$$\frac{d}{dt}(mu) = \dot{m}_{surge}h_{surge} + \dot{m}_{spray}h_{spray} + \dot{Q}_h - p\frac{d}{dt}(mv)$$

EQUATION 5

$$\frac{d}{dt}(mv) = \frac{d}{dt}(m_v v_v + m_l v_l) = 0$$

EQUATION 6

EQUATION 4 provides the conservation of mass. EQUATION 5 provides the conservation of energy. EQUATION 6 provides the conservation of volume.

Where,

$$m = m_v + m_l$$

EQUATION 7

$$mu = m_v u_v + m_l u_l$$

EQUATION 8

$$mv = m_v v_v + m_l v_l$$

EQUATION 9

Therefore, from these equations $\dot{m}_{spray}$, h_{spray}, $\dot{m}_{surge}$, h_{surge}, and $\dot{Q}_h$ are the five input variables. However, there are seven unknown variables (p, m_v, u_v, v_v, m_l, u_l, and v_l) and only three equations relating these parameters. As such, the following relationships are included to close the problem solution. These added relationships assume that all vapor and liquid conditions are maintained at saturation.

$u_l \equiv u_f = f(p)$

EQUATION 10

$v_l \equiv v_f = f(p)$

EQUATION 11

$u_v \equiv u_g = f(p)$

EQUATION 12

$v_v \equiv v_g = f(p)$

EQUATION 13

The PZR transient model terms and subscripts are identified in TABLE 3 and TABLE 4. However, there is some overlap between the standard PZR model notations and control terms. TABLE 7 provides a list of standard control terms to avoid confusion.

TABLE 3: PZR model terms

Symbol	**Description**	**Typical Units**
$\dot{m}$	Mass Flow Rate	kg/sec
$\dot{Q}$	Heater Power (i.e., heat flow rate), $\dot{Q} = p\dot{V}$	kJ/s = kW
h	Specific Enthalpy	kJ/kg
m	Mass	kg
p	Pressure	kPa or bar
u	Internal (Mass Specific) Energy	kJ/kg
U	Internal Energy	kJ
v	Mass Specific Volume	m^3/kg
V	Volume	m^3
X	Vapor (Steam) Quality	No units or %
ρ	Density	kg/m^3

TABLE 4: PZR model subscripts

Symbol	**Description**
f	Liquid at saturation condition
g	Vapor at saturation condition
h	Heater
l	Liquid (i.e., Water)
out1	flow out of Volume 1 into 2
out2_up	flow out of Volume 2 into 3
out2_down	flow out of Volume 2 into 1
out3	flow out of Volume 3 into 2
p	Pressurizer
RO	Rainout
SC	Spray Condensate
spray	Spray Line
surge	Surge Line
v	Vapor (i.e., Steam)
WC	Wall Condensate
FL	Flashing

In addition to the terms in the model above, another mass flow rate term ($\dot{m}_{vlv}$) is sometime included to consider the flow rate exiting the PZR if the PZR relieve valve is opened, which would be a third "gate" for the control volume model. However, it was neglected from the above model, since the relief valve only opens if the pressure exceeds a safety set-point, which is outside the normal operating rage (i.e., PZR spray controller failed). The PZR model differential equations can be combined in matrix form as follows [39]:

$$\psi \cdot \dot{z} = \eta$$

EQUATION 14

The matrix form can be rearranged as follows:

$$\dot{z} = \psi^{-1} \cdot \eta = f(z, u)$$

EQUATION 15

Where, η is a vector that contains the system inputs “u” (e.g., mass flow rate, heater power, etc.), ψ is the state matrix, and z is a vector that contains the system state variables (e.g., pressure, steam quality, volume, etc.). The system input is commonly notated as “u” in most control literature and should not be confused with specific internal energy, which is also commonly notated as “u” in thermodynamic literature.

The PZR model equation set is in the form of a differential algebraic equation (DAE), instead of ordinary differential equations (ODE), with a constraint on volume as seen in EQUATION 6 [39].

Expanded, with inclusion of parameter functional dependence (e.g., h = *f*(p, x), ρ = *f*(p, x), etc.) and application of the chain rule for solution of thermodynamic variables with respect to time (see EQUATION 18), the PZR single region model is expressed as follows, where pressure (p) and vapor quality (X) are chosen as the state variables [39]:

$$\dot{z} = \psi^{-1} \cdot \eta = f(z, u)$$

$$\psi = \begin{bmatrix} \mathrm{V_p} \dfrac{\partial \rho_\mathrm{p}}{\partial p} & \mathrm{V_p} \dfrac{\partial \rho_\mathrm{p}}{\partial X_p} \\ \mathrm{V_p}\left(\rho_\mathrm{p} \dfrac{\partial h_\mathrm{p}}{\partial p} - 1 \right) & \mathrm{V_p} \rho_\mathrm{p} \dfrac{\partial h_\mathrm{p}}{\partial p} \end{bmatrix}$$

$$\eta = \begin{bmatrix} \dot{m}_{surge} + \dot{m}_{spray} \\ \dot{m}_{surge}\left(h_f - h_p\right) + \dot{m}_{spray}\left(h_{spray} - h_p\right) + Q \end{bmatrix}$$

$$\dot{z} = \frac{\mathrm{d}}{\mathrm{dt}} \begin{bmatrix} p \\ X_p \end{bmatrix}$$

EQUATION 16

Similar to EQUATION 16, the single-region dynamic model of a pressurize (PZR) with insurge can also be stated in summarized matrix form, where pressure (p) and volume (V) are chosen as the state variables, as shown in EQUATION 17.

$$\eta = \psi \cdot \dot{z} \Rightarrow$$

$$\frac{\mathrm{d}}{\mathrm{dt}}\begin{bmatrix} m_{total} \\ U_{total} \end{bmatrix} = \begin{bmatrix} (\rho_l' V_l + \rho_v' V_v) & (\rho_l - \rho_v) \\ (\rho_l' V_l h_l + \rho_v' V_v h_v + \rho_l V_l h_l' + \rho_v V_v h_v') & (\rho_l h_l - \rho_v h_v) \end{bmatrix} \begin{bmatrix} \dot{p} \\ \dot{V}_l \end{bmatrix}$$

EQUATION 17

The PZR system model is in the form of a nonlinear differential algebraic equation (DAE), since the density and enthalpy are a nonlinear function of the pressure. The standard form of an ordinary differential equation is $\dot{x}$ = *f*(t,x) [71]. However, the standard form of a DAE is F($\dot{x}$,x,t) = 0 [71]. Therefore, a full solution necessitates an iterative numerical type of approach. The matrix ψ contains partial derivatives of density (ρ) and enthalpy (h), where the chain rule is applied to find the relational change of each variable with respect to time and pressure (see EQUATION 18). Therefore, it is beneficial to see if this equation form is similar to any other models with a similar equation form to understand how it will behave and how to evaluate the mathematical model(s). Consequently, per observation, ψ functions similar to a Jacobian matrix (J) of the system pressure (e.g., this form is similar to what is utilized in solving power flow equations via the Newton-Raphson technique) [70], as shown in EQUATION 18.

$$\eta = \psi \cdot \dot{\mathrm{z}} = \mathrm{J} \cdot \dot{\mathrm{u}} \Rightarrow \dot{\mathrm{y}} = \mathrm{J}^{-1}\dot{\mathrm{u}} = \mathrm{B}\dot{\mathrm{u}} \;\; (\psi^{-1} = \mathrm{J}^{-1}) \Rightarrow (\approx \mathrm{y} = \mathrm{Gu})$$

$$\psi^{-1} = \begin{bmatrix} (\rho_l' V_l + \rho_v' V_v) & (\rho_l - \rho_v) \\ (\rho_l' V_l h_l + \rho_v' V_v h_v + \rho_l V_l h_l' + \rho_v V_v h_v') & (\rho_l h_l - \rho_v h_v) \end{bmatrix}^{-1}$$

$$\rho'(p,t) = \frac{\partial \rho}{\partial t} = \frac{\partial \rho}{\partial p}\frac{\partial p}{\partial t} \qquad \text{and} \qquad h'(p,t) = \frac{\partial h}{\partial t} = \frac{\partial h}{\partial p}\frac{\partial p}{\partial t}$$

EQUATION 18

The eigenvalues of "ψ^{-1}" in EQUATION 18 are the nontrivial solutions to the equation form $A\Phi = \lambda\Phi$, which leads to the nontrivial solutions of $\det(A - \lambda I) = 0$. The eigenvalues can be either real or complex. Shown below are some sample stability analysis values for an example PZR model fit to a prototype 2nd order system, based on EQUATION 36.

Example simplified PZR Model (2nd Order System) Stability Analysis:

```
p = 30          (PZR Initial Pressure at equilibrium, bar)
Vtot = 16.5     (PZR Total Volume, m^3)
V1 = 8.8        (PZR Initial Liquid Volume at Equilibrium, m^3)

eigen = [-0.0140, 0.0000] (Eigenvalues)

Wn = [0.0140, 0.0000]     (Natural Frequencies, rad/sec))

zeta = 1        (Damping Ratio)

tau = 71.6      (Time Constant, sec)

Right side eigenvector (phi) and eigenvalues (lamda):
phi =
   [-0.9978   -0.0010]
   [ 0.0657   -1.0000]

lamda =
   [-0.0140         0]
   [      0    0.0000]

Left side eigenvector (psi):
psi =
   [-1.0021    0.0010]
   [-0.0658   -0.9999]

Participation Matrix (Part):
Part =
    [0.9999    0.0001]
    [0.0001    0.9999]

Open Loop Transfer function:
         0.000195
--------------------------
s^2 + 0.02793 s + 0.000195

Closed Loop Transfer function:
        0.000195
-------------------------
s^2 + 0.02793 s + 0.00039
```

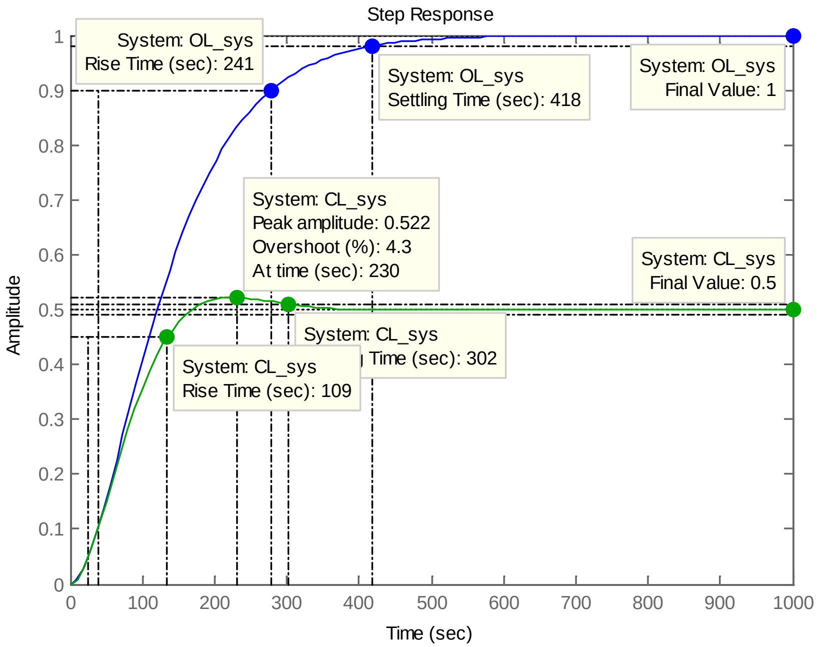


FIGURE 16: Example simplified (2nd order) PZR model step response

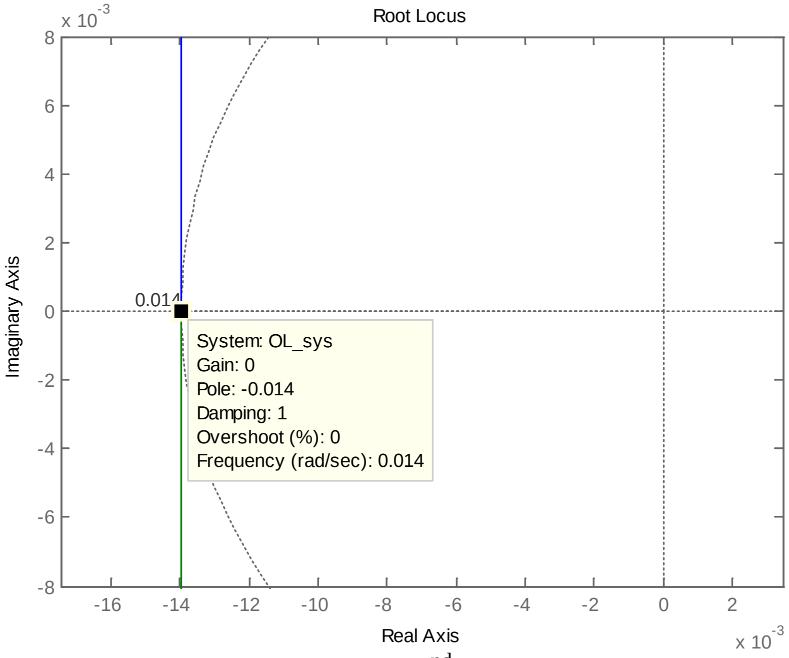


FIGURE 17: Example simplified (2nd order) PZR model root locus

As seen in the example, there are two eigenvalues for this system. Both eigenvalues only contain negative real components, which correspond to a stable decaying non-oscillatory mode [70], as further observed in FIGURE 16 and FIGURE 17.

The behavior of the system (e.g., eigenvectors, eigenvalues, participation matrix, etc.) can be obtained via the state (transfer function) matrix, so it becomes a critical starting point in this analysis.

The right eigenvector provides the mode shape (i.e., relative activity of the state variables when the particular mode is excited). For example, phi(1,1) = -0.9978.

The left eigenvector provides a measure of contribution (weighting) of a modes activity with respect to states (i.e., it identifies which combination of original state variables displays only a particular mode). For example, psi(1,1) = -1.0021.

The participation matrix allows a normalized view of the relationship between states and modes that is independent of units and scaling associated with the state variables, which is not possible when only viewing right and left eigenvectors (i.e., the participation matrix combines the right and left eigenvectors into a single matrix) [70]. The participation matrix consists of matrix elements called participation factors (i.e., P(k,i) = phi(k,i)*psi(i,k), which provide a relative participation of the k^{th} state variable in the i^{th} mode (e.g., Part(1,1) = 0.9999 is the participation of the first state variable in the first mode).

The natural frequency is the frequency at which the system most desires to oscillate (i.e., the frequency that requires the least amount of energy input to induce oscillation).

The damping ratio describes the ability of the system to damp (i.e., eliminate) oscillatory behavior (e.g., act as a shock absorber). For example, a damping ratio of 1 (critically damped) indicates the system is non-oscillatory, since all eigenvalues contain only real components.

Together, these parameters (i.e., natural frequency, damping ratio, eigenvalues, eigenvectors, and participation matrix) provide the necessary information to understand the characteristics of the system behavior.

2.2 Test Case Process and Pressurizer Parameters

TABLE 5 provides a description of the characteristic PZR parameters required to define the PZR for simulation and includes test case sizing values [39] that were used for the PZR control research simulations. PZR surge test data, as discussed in [39] and [67], is depicted in FIGURE 18. The Shippingport PZR performance data during loss-of-load tests [67] were used as the transient surge data to disturb the system for the PZR control research (see FIGURE 18). In addition to the simulation surge mass flow rate provided in FIGURE 18, five additional inputs were used with each of the pressurizer control methods investigated (i.e., a total of six inputs were used). These inputs and the simulation results with the control methods investigated are provided in APPENDIX C.

The simulation time period was selected as 600 seconds, since this was the time period of available empirical data that was used as the benchmark for simulation testing. Only one time period was used for the simulation, since the typical pressure response of the system is faster (see FIGURE 30 and FIGURE 50). Plasma pulses might occur on the order of once every 30 minutes [22], which decouples the effects of most transient dynamics from one pulse to the next (i.e., the system will have mostly settled out by the next plasma pulse). Hence, simulating for multiples pulses is not necessary. However, if a pressurized cooling system experiences thermal-hydraulic (temperature and pressure) pulses on a period faster than the pressurizer response time constant (see stability example in Section 2.1), then the above assumption is no longer valid, which would necessitate analysis over multiple pulses to see the coupled effect of pulses on the PZR pressure control response.

Per [106], the pressure control setpoint for a typical PWR PZR is 2250psia, with the first (high pressure) alarm occurring at 2350psia. Therefore, the first high alarm occurs at a pressure value 4.44 percent greater than the control setpoint. For the PZR used in pulsed cooling system research, the control setpoint is 140bar. Therefore, the first high pressure alarm (using the same alarm logic) would occur around 146.2bar. As such, to avoid initiating high pressure alarms during normal operation, a target maximum pressure for control was selected as 146bar.

TABLE 5: PZR parameter descriptions and test case values [39]

Parameter	**Value**	**Units**	**Comment**
Temperature of PZR spray	260	ºC	A constant nominal value is expected in the model
Mass flow rate of PZR surge	varies	kg/sec	A vector of time series data is expected in the model (see FIGURE 18)
Temperature of PZR surge fluid	280	ºC	A vector of time series data is desired to properly compute the specific surge enthalpy at each surge fluid temperature and system pressure, however, the model can accept a constant/nominal value
Internal Diameter of PZR	1.371	m	A constant value
Internal Cross-sectional Area of PZR	1.4763	m^2	A constant value
Internal Volume of PZR	7.419	m^3	A constant value
Internal Volume of PZR Region 1	0.9274	m^3	A constant value (liquid)
Internal Volume of PZR Region 2	0.9274	m^3	A constant value (liquid)
Internal Volume of PZR Region 3	5.5643	m^3	A constant value (liquid + vapor)

TABLE 5 (continued)

Parameter	**Value**	**Units**	**Comment**
Nominal Pressure of System	140	bar	A constant nominal or initial value is expected in the model
Maximum PZR Pressure	150	bar	A constant value. In the model, this is currently used as the set-point pressure to open the safety relief value (i.e., this is not necessarily the maximum design pressure).
Power output of Heater 1	40	kW	A constant maximum value for each on/off and an operating range for proportional heaters
Power output of Heater 2	80	kW	A constant maximum value for each on/off and an operating range for proportional heaters
Power output of Heater 3	250	kW	A constant maximum value for each on/off and an operating range for proportional heaters
Heater 1 (Temperature) Set-point - On	332.2	ºC	The model currently uses the liquid temperature in the PZR to turn on/off one bank of PZR heaters.
Heater 1 (Temperature) Set-point - Off	335.6	ºC	The model currently uses the liquid temperature in the PZR to turn on/off one bank of PZR heaters.
Heater 2 (Pressure) Set-point - On	136.5	bar	The model currently uses the PZR Pressure to turn on/off some of the PZR heaters
Heater 2 (Pressure) Set-point - Off	137.9	bar	The model currently uses the PZR Pressure to turn on/off some of the PZR heaters
Heater 3 (Pressure) Set-points - On	133.4	bar	The model currently uses the PZR Pressure to turn on/off some of the PZR heaters
Heater 3 (Pressure) Set-points - Off	138.9	bar	The model currently uses the PZR Pressure to turn on/off some of the PZR heaters
Mass flow rate of PZR spray	*f*(p,T), (e.g., 1.9 to 4.0)	kg/sec	For variable control, the operating range of values is desired for the model (min. and max. flow rate of PZR spray), but the maximum value can be used for on/off control. $\dot{m}_{\text{spray}} = f$(nominal pressure, spray temperature)
PZR Spray Pressure Set-point - On	142.7	bar	A constant value (or range for variable spray control) is expected in the model to turn on/off or vary the PZR spray.

TABLE 5 (continued)

Parameter	Value	Units	Comment
PZR Spray Pressure Set-point - Off	139.6	bar	A constant value (or range for variable spray control) is expected in the model to turn on/off or vary the PZR spray.
PZR High Pressure Alarm	146.2	bar	A constant value is specified for the first alarm on high pressure. Actions above this pressure are for safety function outside the scope of normal operational control (i.e., the focus of this research).

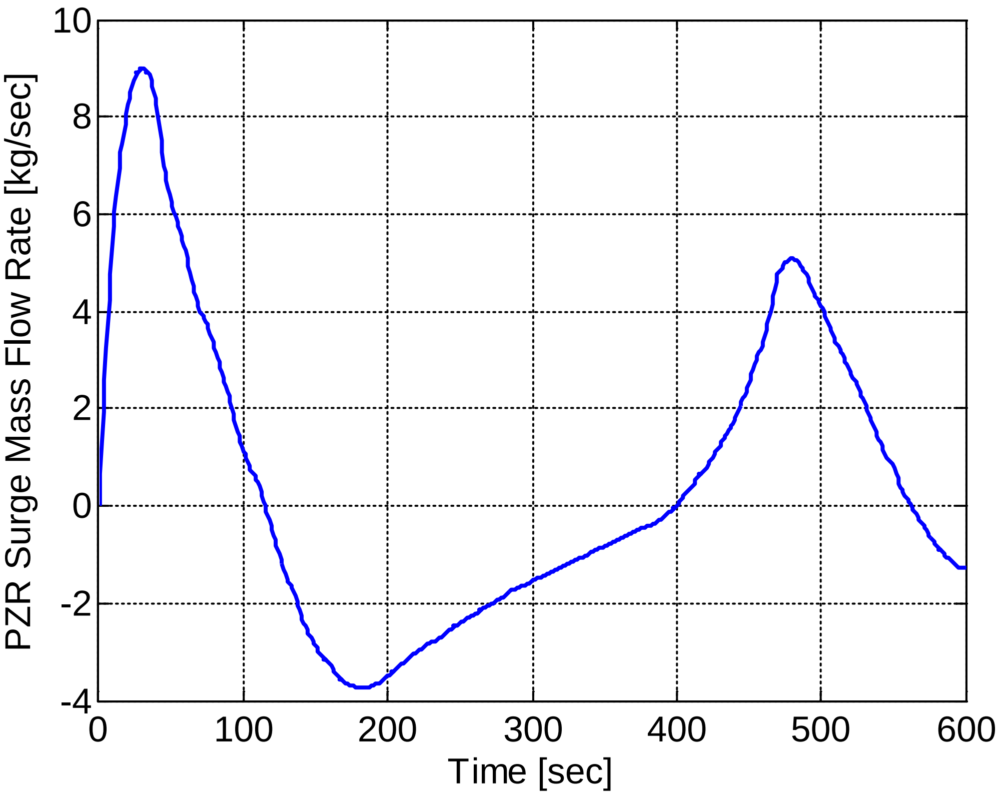


FIGURE 18: Test case PZR surge data [39] and [67]

2.3 Reduced Order SISO PZR Model

PZR models (e.g., linearized and reduced order) for thermal-hydraulic analysis have been developed [14], [39]. Reference [39] introduces a 4th order linearized state space PZR model. However, it is specific to the PZR discussed in [39], rather than a generic model. Additionally, the model in [39] is developed as multi-input (which is realistic of the physical system). However, a multi-input system is beyond the design scope for a SISO PZR model.

Therefore, a 4th order model was developed via MATLAB functions “iddata” and “ssest”, which produce a simulated model of a system based on actual input and output data. For this exercise, the Shippingport Pressurizer Test data discussed in [39] and [67] was utilized. The PZR mass flow rate was selected as the input and the PZR pressure was selected as the output.

Shown below are additional details regarding the development of the fourth order PZR model utilized in this project. The simulated PZR model exhibits a 91.5 percent fit with the experimental data. Therefore, while close to the actual model data, the model does not completely capture the physical system. Additionally, it does not incorporate the other model inputs that exist in the physical system. Therefore, additional modeling work is necessary to obtain a more accurate 4th order model for final control design solution. However, the model developed is believed sufficient to demonstrate the digital control design techniques discussed in the following sections. Shown below (at the end of this section) are figures for the input and output variables, open loop root locus plot, and open loop step response of the system.

MATLAB Script “PZR_SS_Estimate.m” was utilized to obtain the fourth order pressurizer (PZR) model based on the Shippingport Test data [39]. The MATLAB command line output is shown below:

```
sys =
  Continuous-time identified state-space model:
      dx/dt = A x(t) + B u(t) + K e(t)
       y(t) = C x(t) + D u(t) + e(t)
  A = 
              x1         x2         x3         x4
   x1  -0.003106    0.05404   -0.05883    -0.1773
   x2    -0.2246      4.028     -4.678     -13.45
   x3     -0.388      7.324     -7.508     -22.72
   x4  -0.004663    0.06582     0.2743     -0.348


  B = 
             u1
   x1  0.003074
   x2    0.2312
   x3    0.3925
   x4   0.00767


  C = 
            x1       x2       x3       x4
   y1     2251   0.5812   0.3125  -0.1467


  D = 
       u1
   y1   0


  K = 
              y1
   x1  -0.003454
   x2    -0.3062
   x3    -0.4716
   x4    -0.0225



Parameterization:
   FREE form (all coefficients in A, B, C free).
   Feedthrough: none
   Disturbance component: estimate
   Number of free coefficients: 28
   Use "idssdata", "getpvec", "getcov" for parameters and their 
uncertainties.

Status:
Estimated using SSEST on time domain data "PZR_Data".
Fit to estimation data: 91.51% (prediction focus)
FPE: 0.2535, MSE: 0.2098

sys_tf =
 
  From input "u1" to output "y1":
    7.175 s^3 - 0.5092 s^2 + 0.8674 s + 0.08123
  ------------------------------------------------
  s^4 + 3.83 s^3 + 12.34 s^2 + 2.956 s + 0.0003883
 
Continuous-time transfer function.
```

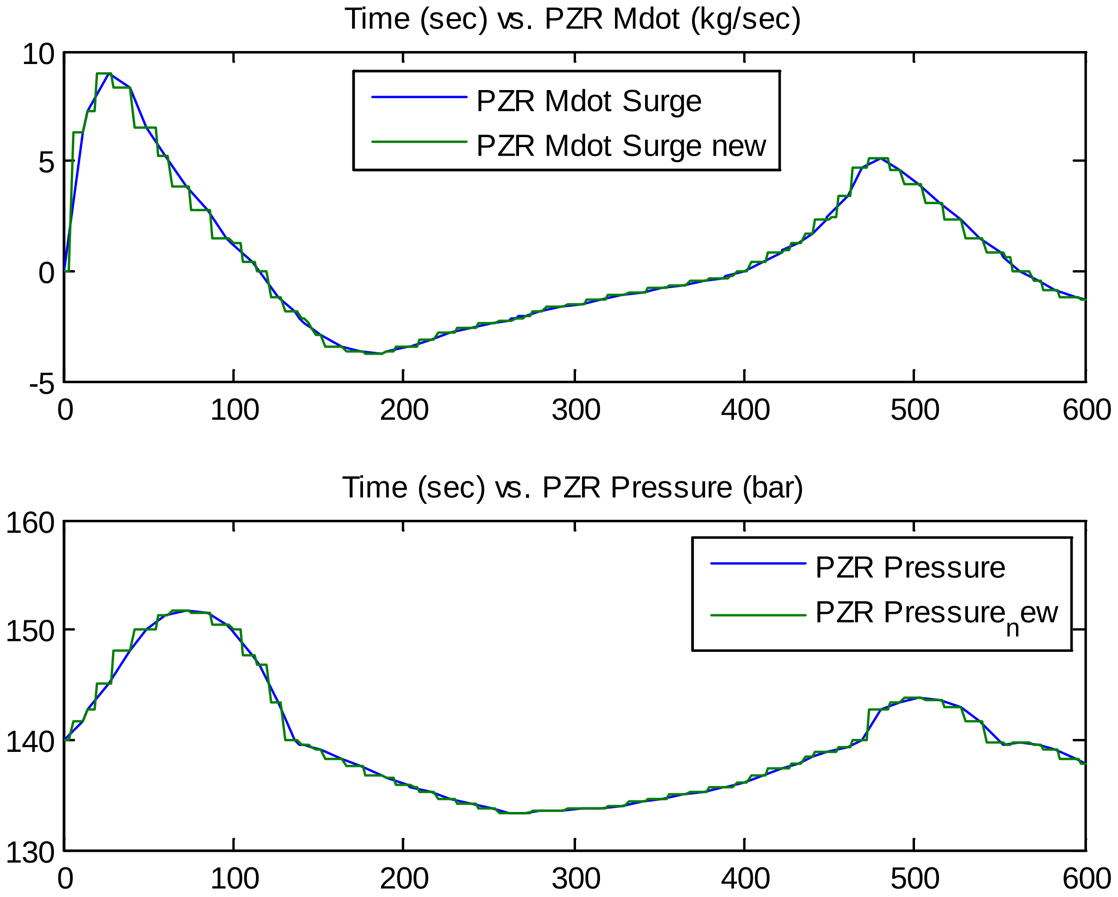


FIGURE 19: PZR fata (raw and normalized) utilized to construct PZR model "PZR_SS_Estimate.m" (data obtained from [39] and [67])

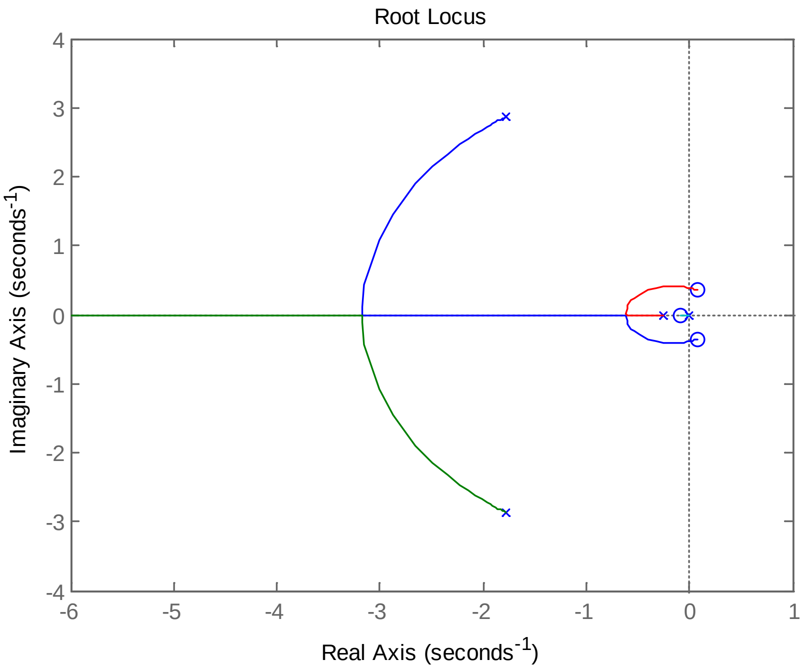


FIGURE 20: Root locus plot of PZR model (“PZR_SS_Estimate.m”)

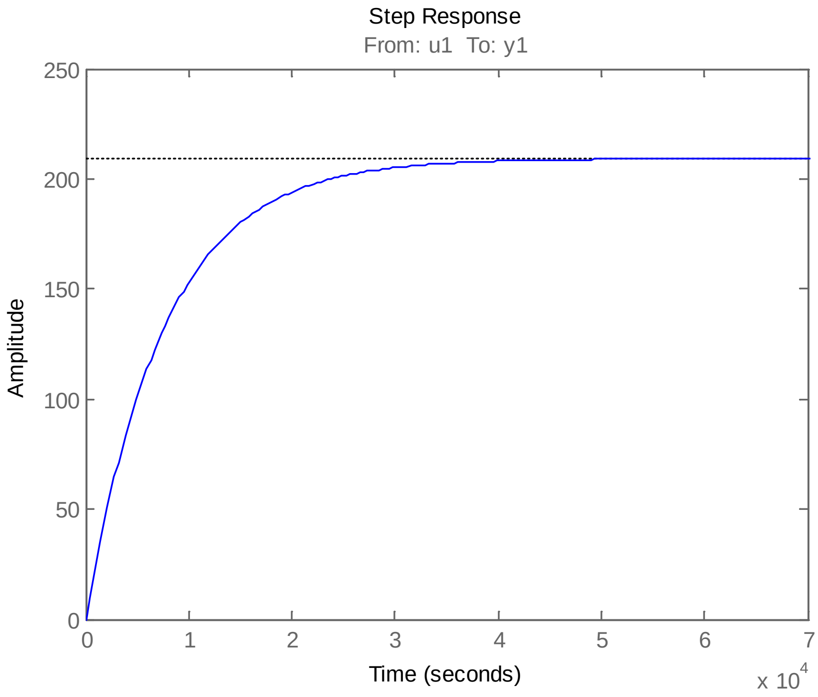


FIGURE 21: Unit step response of PZR model (“PZR_SS_Estimate.m”)

2.4 Reduced Order MISO PZR Model

A numerical linearized fourth order multi-input single-output (MISO) state space PZR model is provided in [39], see EQUATION 19. The PZR dynamic equilibrium point occurs when the PZR is at thermal equilibrium, which permits linearization of a dynamic PZR model (similar to that presented in Section 2.5) around the equilibrium point. However, the linearized MISO PZR model is only applicable for small transients near the thermal equilibrium point, since it does not capture the dynamic behavior of the physical system. Additionally, the model presented is in numerical form, which is specific to the PZR system presented in [39]. Therefore, the linearized MISO model is not useful to model large dynamic behavior or other PZR systems (since it is a specific numerical model). However, it can be helpful to understand how the different system inputs (e.g., $\dot{m}_{spray}$, $\dot{m}_{surge}$, etc.) affect the output of interest (i.e., pressure). This model uses the PZR pressure as the only state variable (i.e., the others are zeroed out, as seen in the state space dynamics “A” matrix), with five inputs (as seen in the state space input “B” matrix). A root locus plot from each input is provided in FIGURE 22, which shows all the poles are within the unit circle indicating system stability.

To evaluate the output (i.e., PZR pressure) sensitivity to each input, a step input was provided to each input variable to see the output response (see FIGURE 23). A linear quadratic regulator (LQR) with a linear quadratic estimator (LQE) was placed in the system to simulate the behavior with a controller. As expected, the step response from the $\dot{m}_{surge}$ had the greatest impact to the pressure (i.e., pressure changed the most from a $\dot{m}_{surge}$ step input). The other inputs do affect the output, but not nearly to the same magnitude (i.e., the next largest impacting input produced an output that is an order of magnitude smaller). This affirms that the mass surge flow rate ($\dot{m}_{surge}$) has the greatest impact to pressure. However, as discussed in CHAPTER 4, $\dot{m}_{surge}$ is not a controllable input (i.e., it is a system disturbance). Therefore, the PZR pressure controller inputs (e.g., $\dot{m}_{spray}$, Q_{heater}, and $\dot{m}_{spray}$) act to mitigate the pressure effect of a $\dot{m}_{surge}$ disturbance.

$$
\begin{aligned}
&\dot{x} = Ax + Bu \\
&y = Cx + Du \\
&A = \begin{bmatrix} -4.3506e^{-4} & 0 & 0 & 0 \\ 2.0295e^{-6} & 0 & 0 & 0 \\ 2.1675e^{-3} & 0 & 0 & 0 \\ 7.8033e^{-4} & 0 & 0 & 0 \end{bmatrix} \\
&B = \begin{bmatrix} -1.4810e^{-2} & -1.5719e^{-2} & -2.3785e^{-2} & -1.1219e^{-1} & 9.0397e^{-1} \\ -1.8029e^{-3} & 1.9453e^{-3} & -1.7610e^{-3} & 2.3954e^{-3} & -4.2168e^{-3} \\ 1.2276e^{-2} & 1.3030e^{-2} & 1.9716e^{-2} & 9.2995e^{-2} & -7.4929e^{-2} \\ -1.9789e^{-1} & 1.2585e^{-1} & 1.9042e^{-1} & 8.9820e^{-1} & 2.1331e^{-1} \end{bmatrix} \\
&C = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \\
&D = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \end{bmatrix}
\end{aligned}
$$

EQUATION 19

In addition to the linearized MISO PZR model presented in this chapter, a different empirically based fourth order state space PZR model is provided in Section 5.3.2 for use with an optimal controller design that uses measurable process values (e.g., pressure, level, temperature, and volume) as state variables.

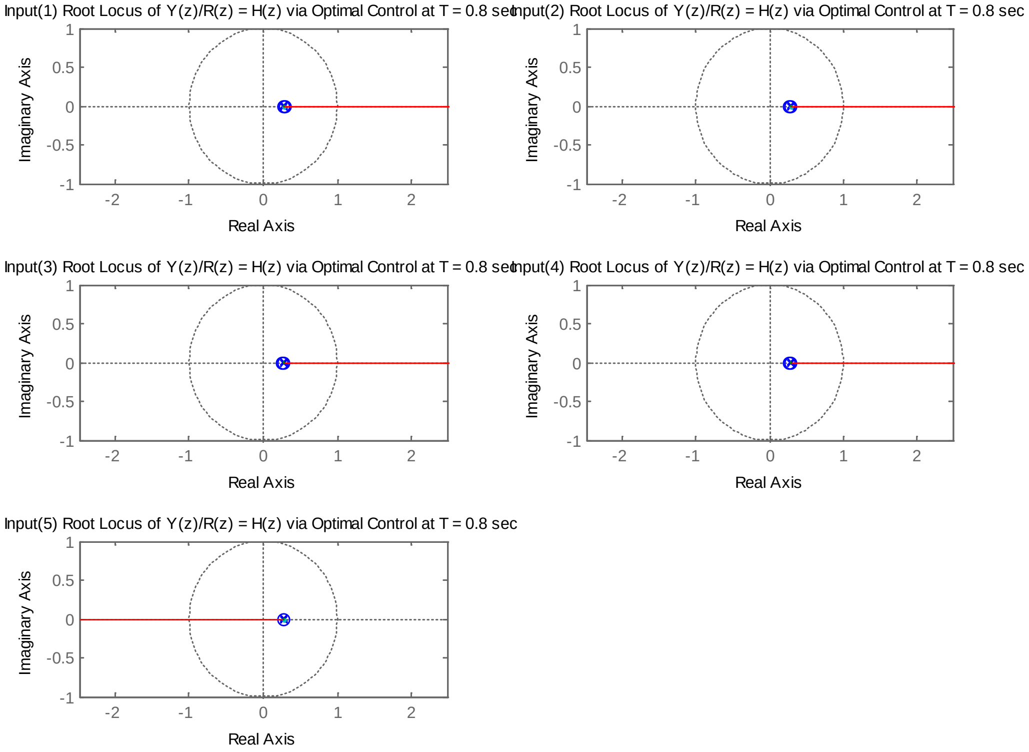


FIGURE 22: Linearized MISO PZR model [39] root locus (each input)

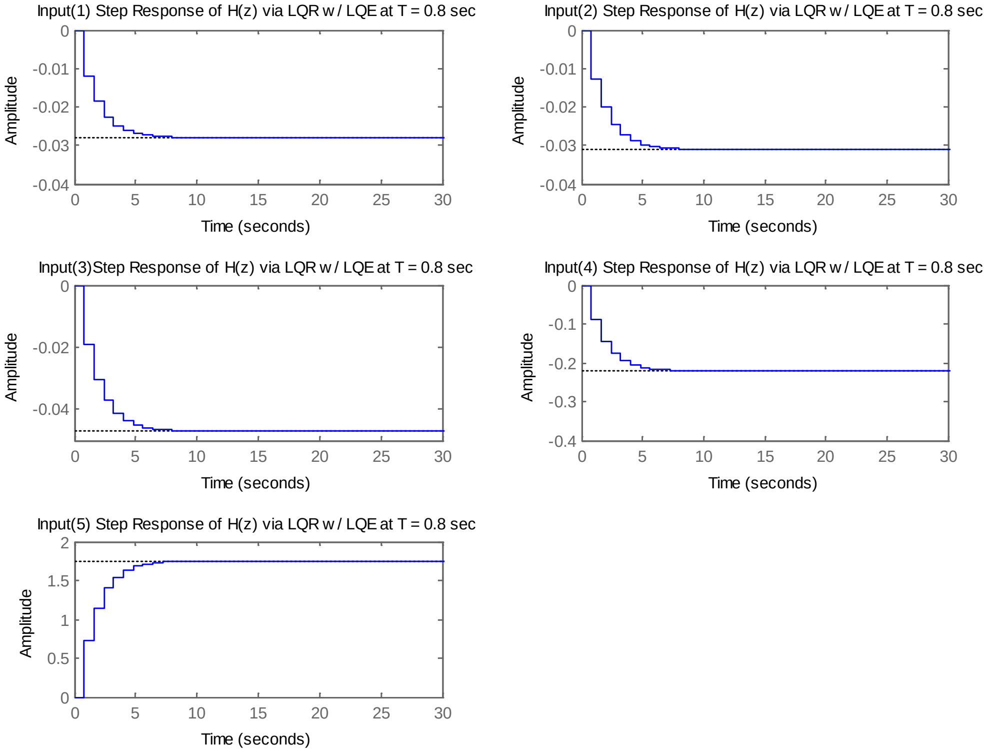


FIGURE 23: Linearized MISO PZR model [39] input step response

2.5 Dynamic MIMO PZR Model

For transient analysis a dynamic PZR model is required. Several different model approaches are available. Empirically based (e.g., RELAP5 or TRACE [72], [73]) and analytical (e.g., Dymola®) software modeling tools are commonly employed to analyze transient dynamics in thermal-hydraulic systems (e.g., a cooling system with a PZR), but are sometimes limited in advanced control modeling capability [29]. Therefore, ongoing research has focused on developing analytical PZR models for dynamic transient analysis that permit implementation of advanced control strategies [39]. Some research has focused on coupling thermal-hydraulic modeling tools (e.g., RELAP5) with control oriented modeling tools (e.g., MATLAB) [29]. However, this necessitates two separate software tools that were not initially designed to interface. Therefore, a model that combines the thermal-hydraulic and control features into one model software is desired. The dynamic PZR model developed by A. Pini [39], which performed well compared to other PZR models and experimental data, is implemented in MATLAB and Simulink. Therefore, advanced control strategies can be easily implemented with the thermal-hydraulic PZR model, since MATLAB and Simulink provide a flexible platform to implement commands and include many generic functions related to controls.

Accordingly, the dynamic PZR model developed by A. Pini [39] was chosen as the basis model to implement the PZR control research. The dynamic PZR model is constructed with two volumes (liquid and vapor) and three regions (upper, middle, and bottom), as shown in FIGURE 24. The two-volume (liquid and vapor) three-region (upper, middle, and bottom) dynamic PZR model considers four possible thermodynamic conditions, referred to as “states” in [39]:

TABLE 6: Description of PZR region thermodynamic “states” [39]

State	Description
1	Top Super-Heated, Bottom Subcooled
2	Top Super-Heated, Bottom Boiling
3	Top Condensing, Bottom Subcooled
4	Top Condensing, Bottom Boiling

Each thermodynamic state has a unique model equation set (8th order model) and it in an expanded form of EQUATION 16 that includes additional terms to describe the interaction between “regions” [39]. An example of the (two-region, three-volume) dynamic PZR model for “state-1” is provided below in EQUATION 20 [39]. Similar models are developed for the other thermal-hydraulic “states”.

A top level view of the PZR Dynamic Model constructed in Simulink is provided in FIGURE 25, as adapted from [39]. The functions under the “Pressurizer” block were developed in [39]. A few minor modifications were made to the “Pressurizer” block (e.g., variables passed out to the “Controller” block), but is mainly unchanged, since it provides the validated PZR model. However, advancing the “Controller” is the key focus of the ongoing research. Therefore, much of the later sections (e.g., CHAPTER 5 and CHAPTER 6) focus on implementing various control methods in the “Controller” block of FIGURE 25 to achieve the desired control performance. FIGURE 26 (adapted from [39]) provides an example of the conventional controller constructed in Simulink.

Therefore, for PZR simulations during transients, this PZR Dynamic Model (Section 2.5) is preferred over the other PZR models presented (i.e., reduced order MIMO and SISO PZR model), since it is capable of capturing the dynamic behavior more accurately. Section 2.2 provides the test case PZR parameters and inputs for simulations.

$$\dot{z} = \psi^{-1} \cdot \eta$$

$$\psi = \begin{bmatrix} \mathrm{V}_{3\mathrm{V}} \dfrac{\partial \rho_{3\mathrm{V}}}{\partial p} & \rho_{3\mathrm{V}} & \mathrm{V}_{3\mathrm{V}} \dfrac{\partial \rho_{3\mathrm{V}}}{\partial h_{3\mathrm{V}}} & 0 & 0 & 0 & 0 & 0 \\ \left(\mathrm{V}_3 - \mathrm{V}_{3\mathrm{V}}\right) \dfrac{\partial \rho_{3\mathrm{L}}}{\partial p} & -\rho_{3\mathrm{V}} & 0 & \left(\mathrm{V}_3 - \mathrm{V}_{3\mathrm{V}}\right) \dfrac{\partial \rho_{3\mathrm{L}}}{\partial h_{3\mathrm{V}}} & 0 & 0 & -1 & 0 \\ \mathrm{V}_2 \dfrac{\partial \rho_2}{\partial p} & 0 & 0 & 0 & \mathrm{V}_2 \dfrac{\partial \rho_2}{\partial h_2} & 0 & 1 & -1 \\ \mathrm{V}_1 \dfrac{\partial \rho_1}{\partial p} & 0 & 0 & 0 & 0 & \mathrm{V}_1 \dfrac{\partial \rho_1}{\partial h_1} & 0 & 1 \\ -\mathrm{V}_{3\mathrm{V}} & 0 & \rho_{3\mathrm{L}} V_{3V} & 0 & 0 & 0 & 0 & 0 \\ -\left(\mathrm{V}_3 - \mathrm{V}_{3\mathrm{V}}\right) & 0 & 0 & \rho_{3\mathrm{L}}\left(\mathrm{V}_3 - \mathrm{V}_{3\mathrm{V}}\right) & 0 & 0 & -\left(h_2 - h_{3L}\right) & 0 \\ -\mathrm{V}_2 & 0 & 0 & 0 & \rho_1 V_2 & 0 & 0 & -\left(h_1 - h_2\right) \\ -\mathrm{V}_1 & 0 & 0 & 0 & 0 & \rho_1 V_1 & 0 & 0 \end{bmatrix}$$

$$\eta = \begin{bmatrix} -\dot{m}_{SC} - \dot{m}_{VLVy} \\ \dot{m}_{SC} + \dot{m}_{spray} \\ 0 \\ \dot{m}_{surge\,(\mathrm{in})} \\ \dot{m}_{SC}\left(h_{3V} - h_g\right) + Q \\ \dot{m}_{SC}\left(h_g - h_{3L}\right) + \dot{m}_{spray}\left(h_{spray} - h_{3L}\right) + Q \\ Q \\ \dot{m}_{surge\,(\mathrm{in})}\left(h_{surge\,(\mathrm{in})} - h_1\right) + Q \end{bmatrix}$$

$$\dot{z} = \begin{bmatrix} \dot{p} \\ \dot{\mathrm{V}}_{3\mathrm{V}} \\ \dot{\mathrm{h}}_{3\mathrm{V}} \\ \dot{\mathrm{h}}_{3\mathrm{L}} \\ \dot{\mathrm{h}}_2 \\ \dot{\mathrm{h}}_1 \\ \dot{m}_{out\,2\,(up)} \\ \dot{m}_{out1} \end{bmatrix}$$

EQUATION 20

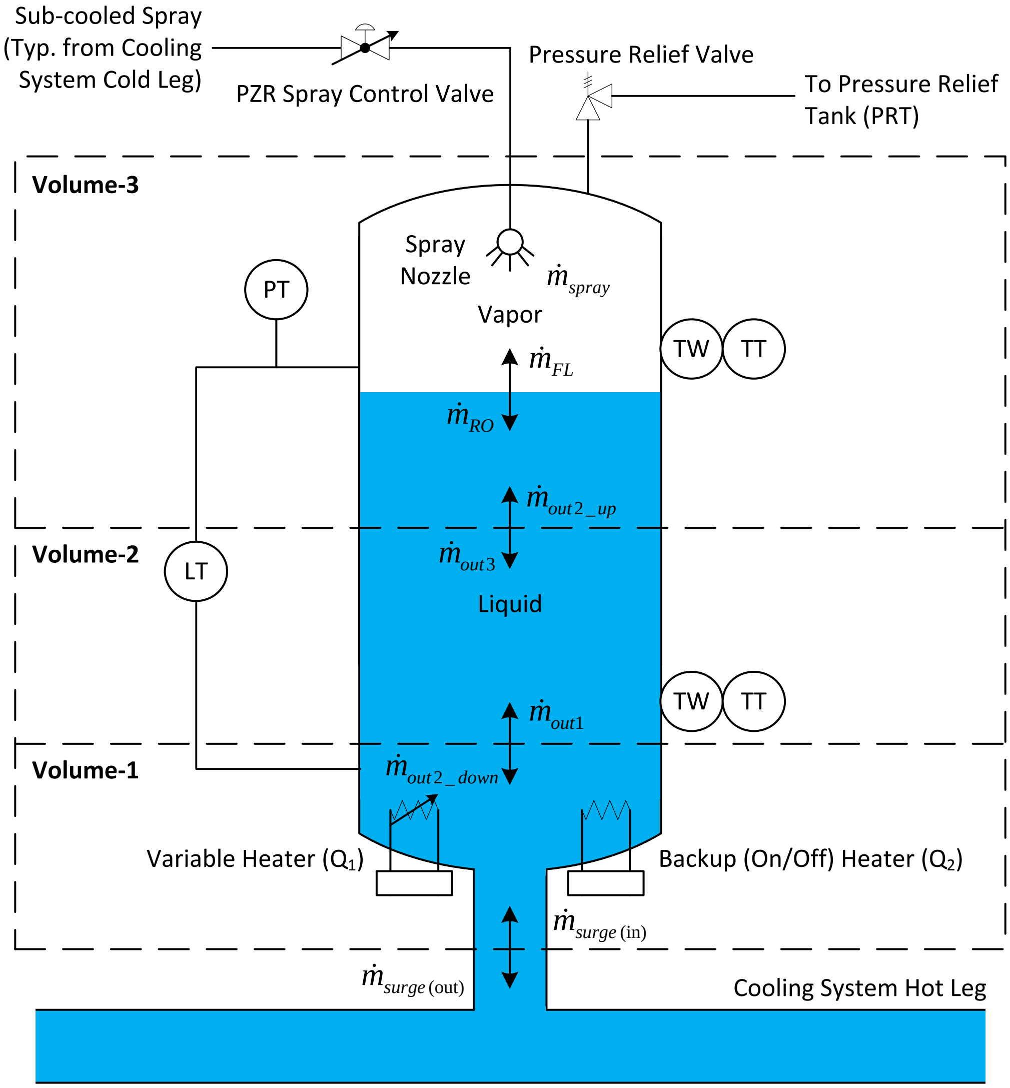


FIGURE 24: Typical 2-region (liquid & vapor) 3-volume PZR [39]

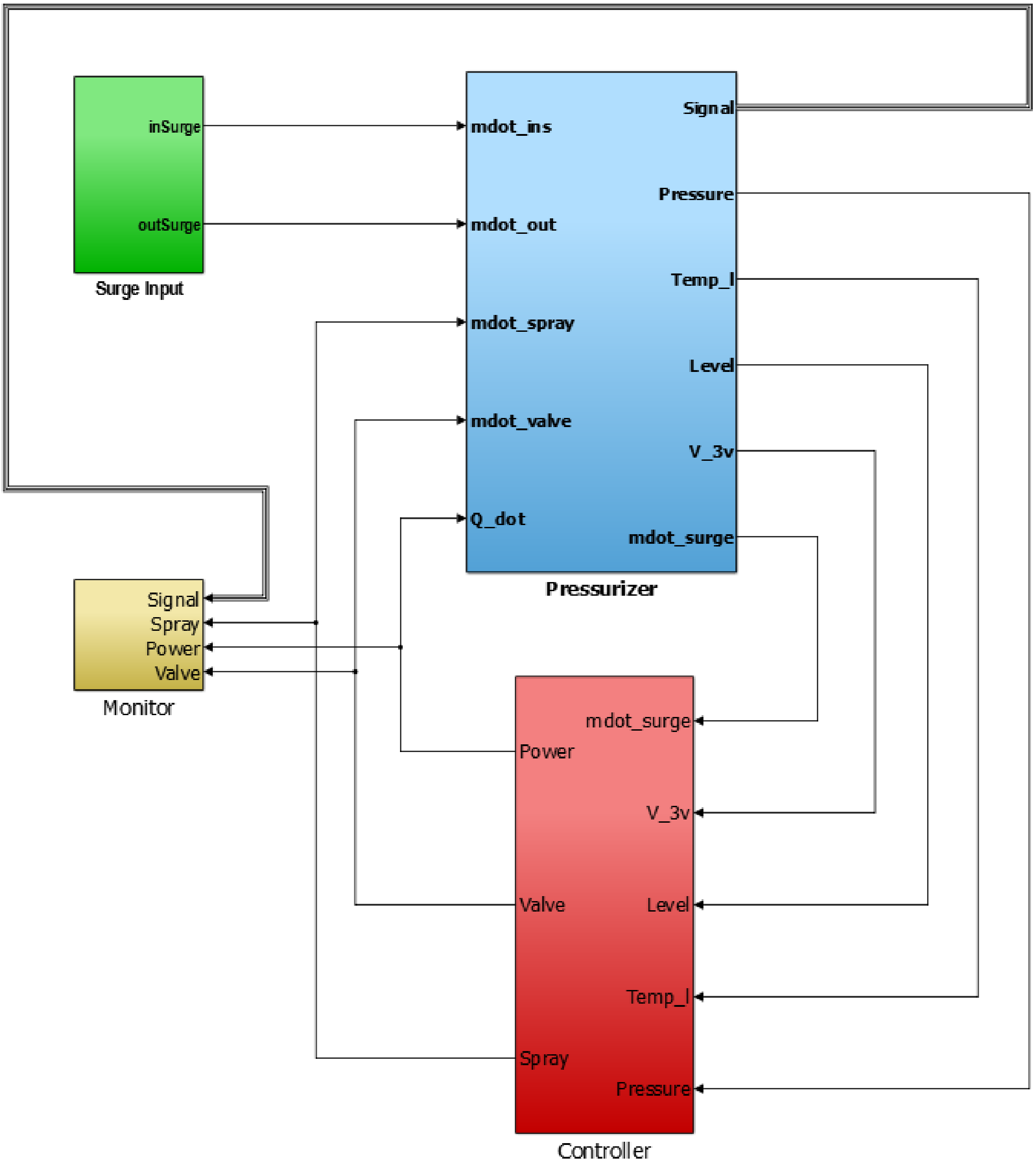


FIGURE 25: Simulink PZR dynamic model –top level

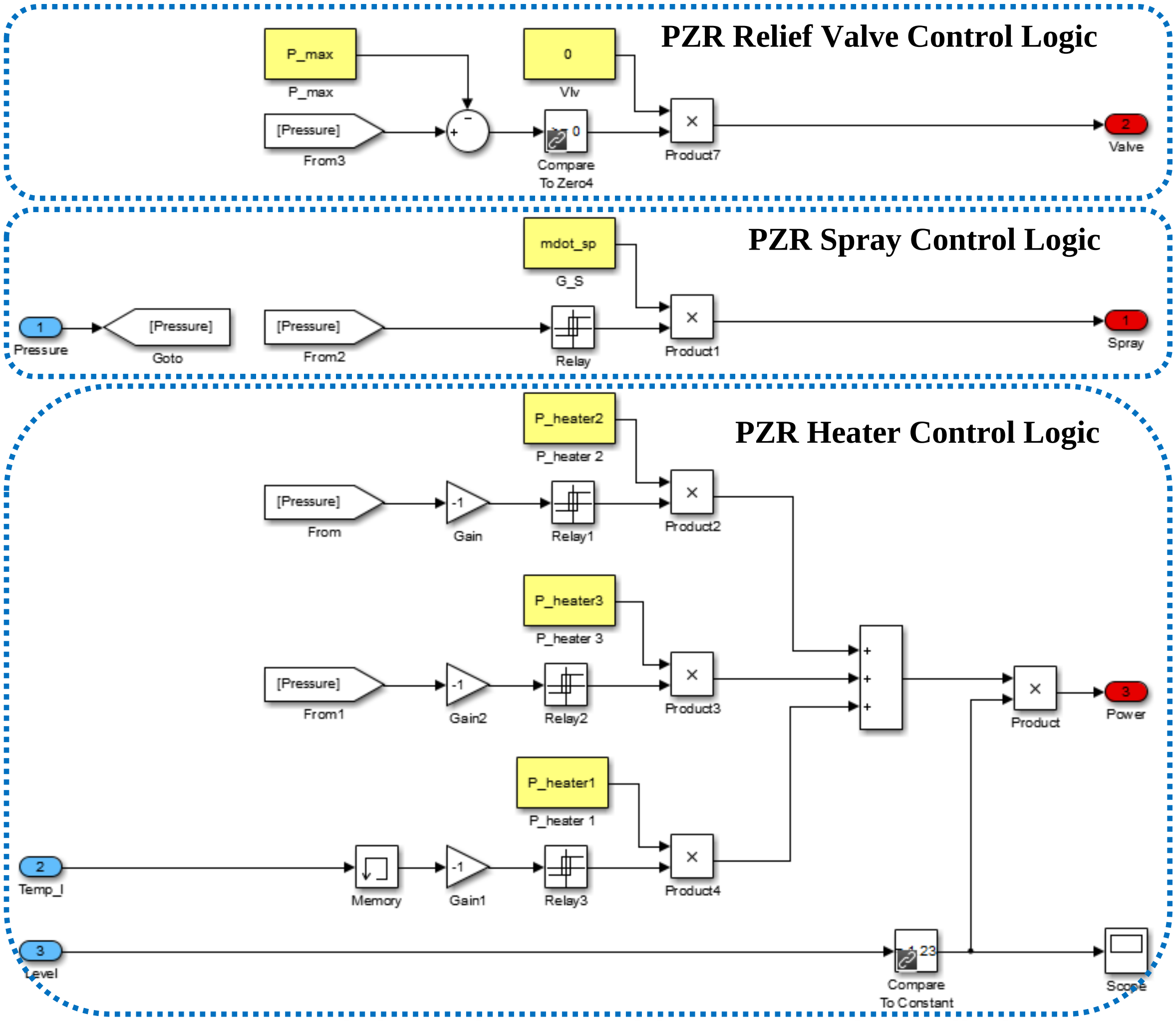


FIGURE 26: Simulink PZR dynamic model – conventional controller logic

CHAPTER 3: CONVENTIONAL PRESSURIZER CONTROLS

This chapter reviews the conventional on/off and PID pressurizer control strategy, including typical PZR pressure control logic and conventional control performance data.

3.1 Conventional (On/Off and PID) Control Summary

Normally, no control action is required, since a PZR's normal saturation response will regulate small pressure changes within the system (e.g., pressure increase from an increase in temperature). An increase in pressure results in some of the steam within the PZR condensing; this reduces the pressure, since water is denser than steam. Oppositely, a decrease in pressure results in some of the water within the PZR flashing to steam; this increases the pressure, since steam is less dense than water. However, if the pressure changes are greater than this normal response can manage (i.e., the pressure changes are greater than the allowed range), additional control actions are added to maintain pressure within the desired range. FIGURE 27 provides a typical block diagram for PZR pressure control functions.

3.2 Typical (Conventional) PZR Pressure Control Logic

The PZR pressure control function can be categorized into two control conditions: (1) control an increase in pressure (e.g., insurge) above the desired range and (2) control a decrease in pressure (e.g., outsurge) below the desired range, as described below. A typical PZR pressure control set-point diagram is provided in FIGURE 29, which was adapted from [75].

The PZR pressure control logic flow sequence is depicted graphically (i.e., simplified logic diagram) in FIGURE 28. This logic pressure control logic is applicable to both conventional controllers (e.g., on/off and PID) and advanced controllers (e.g., optimal). The selected appropriate controller needs to be placed in the associated “controller” block on the diagram (e.g., controller for PZR spray). Otherwise the control philosophy remains the same (e.g., the controlled process variable is always pressure, the manipulated process variables for control are the pressurizer spray and heaters, and the major driving system disturbance is PZR surge mass flow rate). This logic assumes there is active of the pressure relief valve. However, this is typically not the case, since the pressure relief valves are typically mechanically operated relief valves that actuate passively (e.g., PORV), with limit switches to notify the operator if the valve opens or closes.

PZR Control Functional Description:

1) Actions to control an increase in pressure:

    a) If the PZR pressure increases above HIGH-3, then the variable (e.g., proportional) spray is turned on by modulating the PZR spray control valve (with control of the spray valve position as a function of the PZR pressure). The PZR spray nozzle injects sub-cooled water into the top of the PZR, which collapses some of the steam back into water. Since the same mass of water occupies less volume than the same mass of steam, the mass volume decreases. This causes the pressure to decrease within the PZR. Once the desired PZR pressure is reached, the PZR spray is turned off (i.e., the PZR spray control valve is closed).

b) If the PZR pressure increases above HIGH-2, then the PZR safety relief valve automatically opens to the PRT, until the pressure sufficiently decreases. This function is accomplished automatically without any control necessary (i.e., mechanical relief valve). However, there are limit switches on the valve to notify the operator that it has opened.

2) Actions to control a decrease in pressure:

a) If the PZR pressure decreases below LOW-2, then the variable (e.g., proportional) heater is turned on (with control of heat as a function of the pressure). Once the desired PZR pressure is reached, the PZR variable heater is turned off.

b) If the PZR pressure decreases below LOW-1 (i.e., the variable heater alone was unable to add enough heat to sufficiently increase the pressure), then the backup heater is also turned on (with on/off control) in addition to the variable heater (which is set to maximum heat output) until the pressure is > LOW-1.

c) If the PZR water level decreases below the minimum water level setpoint, the heaters are turned off to protect the heaters.

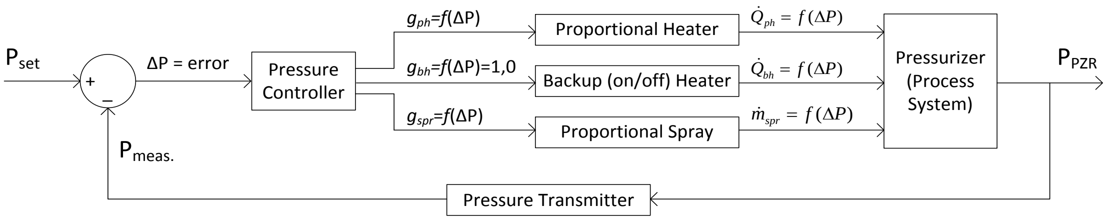


FIGURE 27: Typical block diagram for conventional PZR pressure control [97]

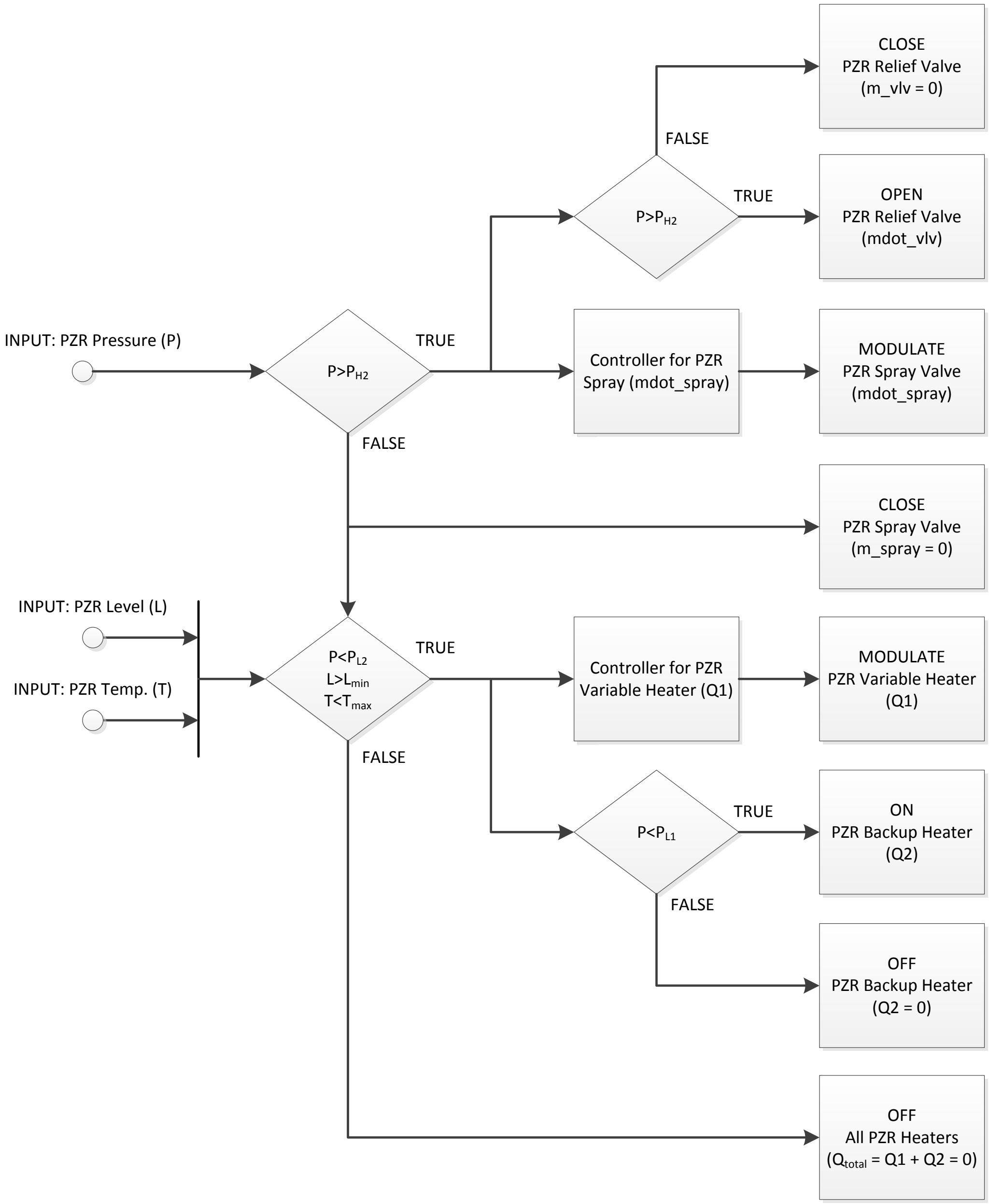


FIGURE 28: Typical control logic flow sequence for PZR pressure control

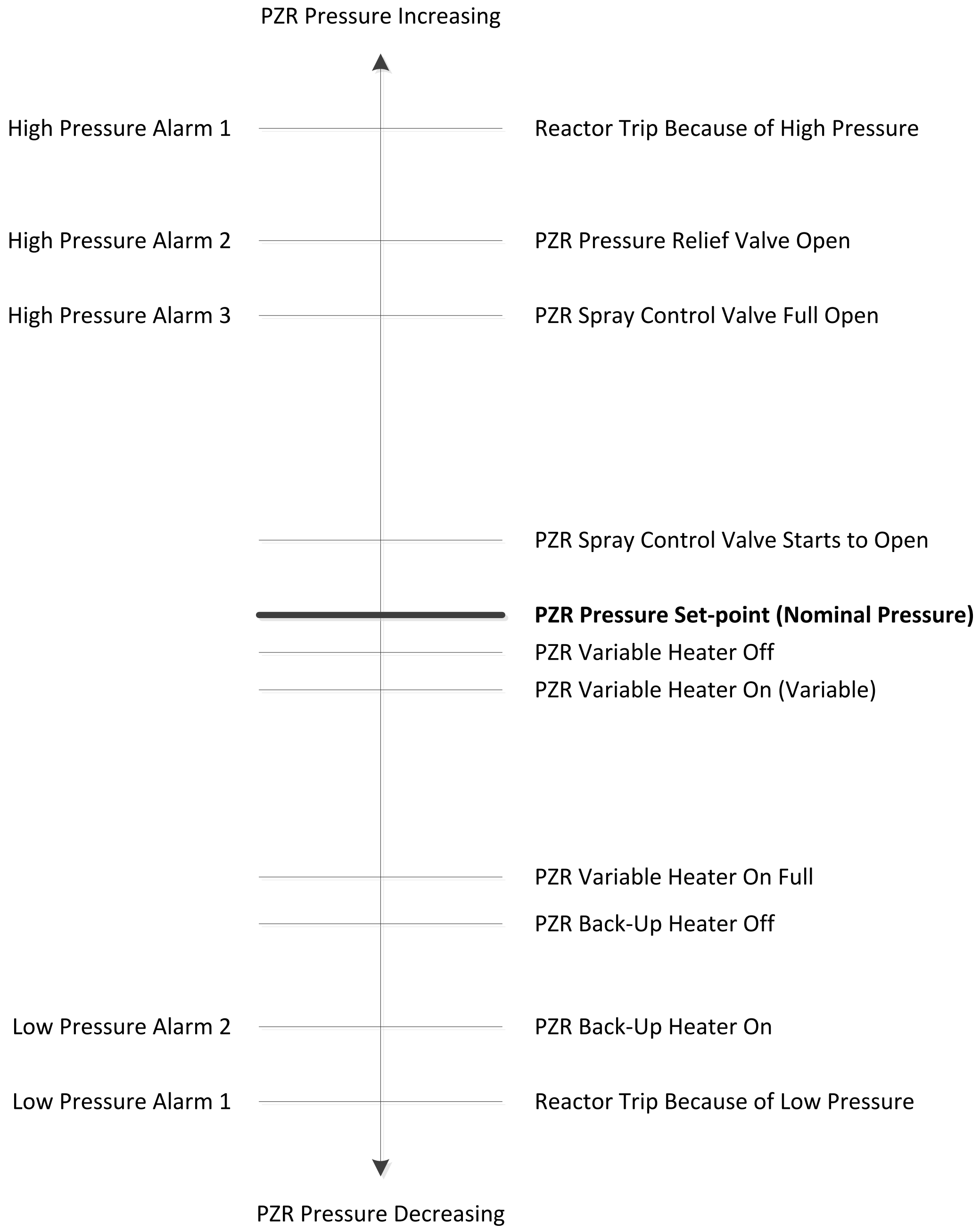

PZR Pressure Increasing
High Pressure Alarm 1
Reactor Trip Because of High Pressure
High Pressure Alarm 2
PZR Pressure Relief Valve Open
High Pressure Alarm 3
PZR Spray Control Valve Full Open
PZR Spray Control Valve Starts to Open
PZR Pressure Set-point (Nominal Pressure)
PZR Variable Heater Off
PZR Variable Heater On (Variable)
PZR Variable Heater On Full
PZR Back-Up Heater Off
Low Pressure Alarm 2
PZR Back-Up Heater On
Low Pressure Alarm 1
Reactor Trip Because of Low Pressure
PZR Pressure Decreasing

FIGURE 29: Typical PZR pressure control set-point diagram [75], [106]

3.3 Test Case Conventional Control Performance Data

The conventional control performance for the dynamic MIMO PZR model, discussed in Section 2.5, is presented in this chapter. The test case process and PZR parameters are provided in Section 2.2. The system is modeled as shown in FIGURE 52. See TABLE 7 for a definition of the control terms and symbols. The mass surge flow rate this simulation is provided in FIGURE 18.

This performance data was used as a benchmark for comparison to the other control techniques investigated (e.g., see CHAPTER 5 and CHAPTER 6). FIGURE 30 provides the conventional control performance for PZR pressure and level. FIGURE 31 provides the control (spray and heater) inputs commanded for the system. The PZR system was disturbed by the PZR surge mass flow rate shown in FIGURE 18. As discussed in Section 2.2, a maximum target pressure of 146bar was selected for control operation to avoid engagement of safety functions for normal operation (i.e., avoid first high pressure alarm). The maximum pressure observed with the conventional control method is approximately 152bar (at 75 seconds), which exceeds the safety setpoint limits imposed. Therefore, alternative control methods are necessary during this type of transient.

In summary, CHAPTER 3 included a summary of conventional control, provided the pressure control logic for a typical PZR, and presented test case data for conventional control performance. Next, CHAPTER 4 introduces several digital control methods and presents PZR pressure control performance with those digital control methods using the reduced order PZR model provided in CHAPTER 2.

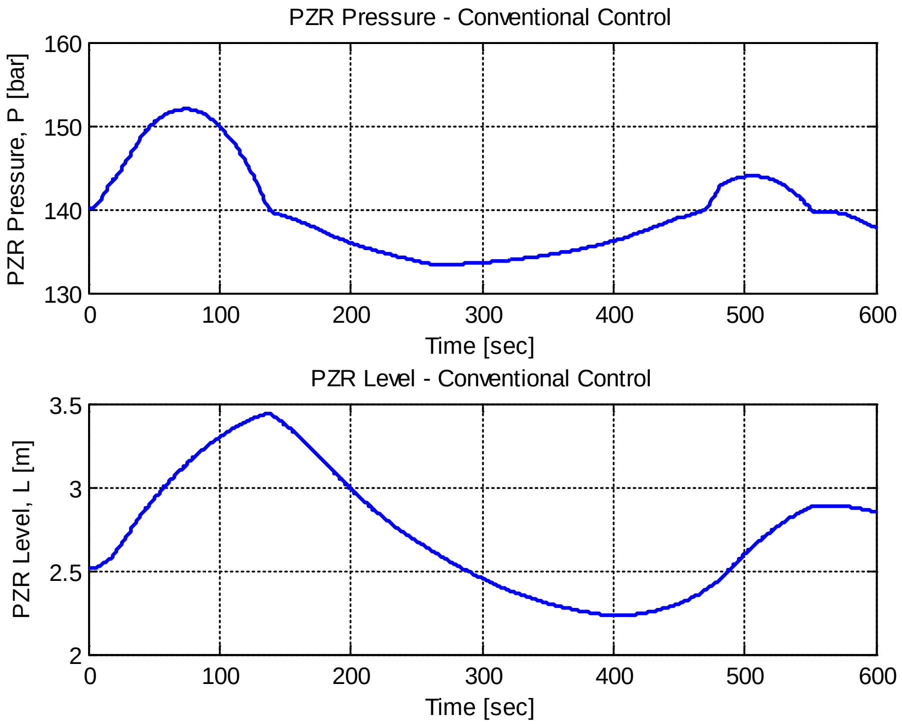


FIGURE 30: Test case PZR pressure and level data [39] and [67]

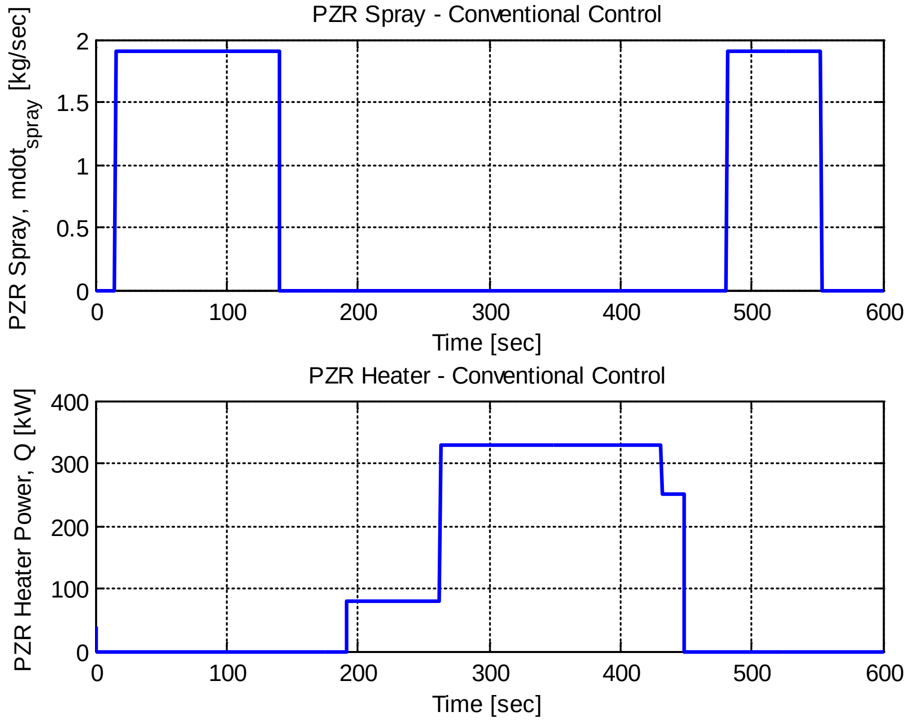


FIGURE 31: Test case PZR spray and heater data [39] and [67]

CHAPTER 4: DIGITAL CONTROLS FOR REDUCED ORDER PZR MODEL

This chapter investigates four digital control methods that are applied to the reduced order PZR models introduced in CHAPTER 2: (1) Indirect Digital Control, (2) Direct Digital Control, (3) Pole Placement Control, and (4) Optimal Control with State Estimator. The results of simulating these control methodologies are presented.

Descriptions and simulation results are provided for the basic digital control methods utilized with the reduced order PZR model presented Section 2.3. This control modeling and simulation proved useful in understanding the basic system behavior and control interaction and served as a basis for selecting which control method(s) to investigate for application with the dynamic PZR model, as discussed in CHAPTER 5. The control methods presented in this chapter are primarily presented as offline control design methods for linear systems (i.e., once designed, the controller does not change as the system changes, for example the optimal gain “k” is a constant). However, CHAPTER 5 places additional focus on making the controller adaptive to dynamic system behavior.

Furthermore, the PZR pressure controller (in CHAPTER 4) directly manipulates the PZR surge mass flow rate, since the system model is SISO, where the input is the surge mass flow rate (i.e., the simplified model does not include disturbances). However, this is not realistic of the true system (i.e., the controlled process variable is pressure, the manipulated process variables for control are the pressurizer spray and heaters, and the major driving system disturbance is PZR surge mass flow rate). Control could be placed on the PZR surge line to restrict surge flow into or out of the PZR, but this would be at the expense of the cooling system and negate the intended function of the PZR. However, for the investigation in CHAPTER 4, the PZR surge mass flow rate was used as the controlled input variable, so its impact to the system can be understood. Hence, the system response times indicated (e.g., settling time of 15 seconds from step input to PZR, as seen in FIGURE 34) are not necessarily realistic of the true pressure control capability for a physical PZR system with spray and heater controls, but rather are intended to demonstrate how the PZR surge flow rate impact the system pressure. This analysis was helpful in demonstrating the importance of mitigating the effects of PZR mass surge flow rate to controlling the pressure. While the spray and heaters are not directly controlling the PZR surge flow rate, they can act to mitigate the effects, which have the desired net effect of controlling the pressure.

4.1 Indirect Digital Control

Section 4.1 provides an overview of the indirect digital control method and presents simulation results of applying this method to the reduced order SISO PZR model that was presented in Section 2.3.

4.1.1 Indirect Digital Control Modeling

An overview of the Indirect Solution Approach (i.e., design controller in "s" and then transform the solution to "z") is presented below, as discussed per [36], [37], [38], [40], [41], [42]:

Recall the continuous time transfer function G(s) can be obtained from the state space model, as shown in EQUATION 21:

$$G(s) = C\Phi B + D$$
$$\Phi = \left[sI - A\right]^{-1}$$

EQUATION 21

The discrete equivalent of G(s) is obtained via trapezoidal integration:

$$s \approx \frac{2}{T}\frac{z-1}{\mathrm{z}+1}$$

$$\mathrm{G(z)} = \mathrm{G}\left(\mathrm{s} \approx \frac{2}{T}\frac{\mathrm{z}-1}{\mathrm{z}+1}\right)$$

EQUATION 22

The desired discrete characteristic equation “CE(z)” of the overall transfer function H(z), as specified from desired roots in problem statement, can be expressed as follows:

$$\begin{aligned}&\text{Desired Polynomial Roots of } \mathrm{CE(z)}, \mathrm{r}_1, \mathrm{r}_2, ..., \mathrm{r}_\mathrm{n} \Rightarrow \mathrm{CE(z)} \\ &\text{e.g.,} \quad \mathrm{CE(z)} = e\mathrm{z}^2 + fz + g = 0 \Rightarrow\end{aligned}$$

The desired continuous CE(s) of the overall transfer function H(s) via inverse z-transform is then state as shown in EQUATION 23:

$$\mathrm{CE(s)} = Z^{-1}[\mathrm{CE(z)}] \Rightarrow e.g., \quad \mathrm{CE(s)} = a\mathrm{s}^2 + bs + c = 0$$

EQUATION 23

Therefore, the desired continuous overall transfer function H(s):

$$\mathrm{H(s)} = \frac{\mathrm{D(s)G(s)}}{1+\mathrm{D(s)G(s)}} \Rightarrow e.g., \quad \mathrm{H(s)} = \frac{gs+h}{a\mathrm{s}^2 + bs + c}$$

EQUATION 24

This leads to computing the continuous controller transfer function D(s):

$$\mathrm{D(s)} = \frac{1}{\mathrm{G(s)}}\frac{\mathrm{H(s)}}{1-\mathrm{H(s)}}$$

EQUATION 25

Where, the controller transfer function D(s) serves to negate the undesired poles and zeroes (as shown in the root locus figure plots in FIGURE 32 and FIGURE 33), so that the overall transfer function H(s) is obtained (as shown by the step response plots in

FIGURE 34). Then, obtain the required discrete controller transfer function D(z) via Trapezoidal Integration:

$$\mathrm{D(z)} = Z[D(s)]$$
$$s \approx \frac{2}{T}\frac{z-1}{\mathrm{z}+1}$$
$$\mathrm{D(z)} = \mathrm{D}\left(\mathrm{s} \approx \frac{2}{T}\frac{z-1}{\mathrm{z}+1}\right)$$

EQUATION 26

Therefore, as a check of the controller design (for unity), where the poles = zeroes

$$\mathrm{Unity} = 1 = \frac{\mathrm{H(z)}}{\frac{\mathrm{D(z)G(z)}}{1+\mathrm{D(z)G(z)}}} = \frac{a+bz^{-1}+cz^{-2}+...+mz^{-n}}{a+bz^{-1}+cz^{-2}+...+mz^{-n}}$$

EQUATION 27

4.1.2 Indirect Digital Control Simulation Results

MATLAB Script was utilized to obtain the indirect solution via trapezoidal integration. Shown below are the “indirect design” figure plots for z = 0.8 +/- j0.4:

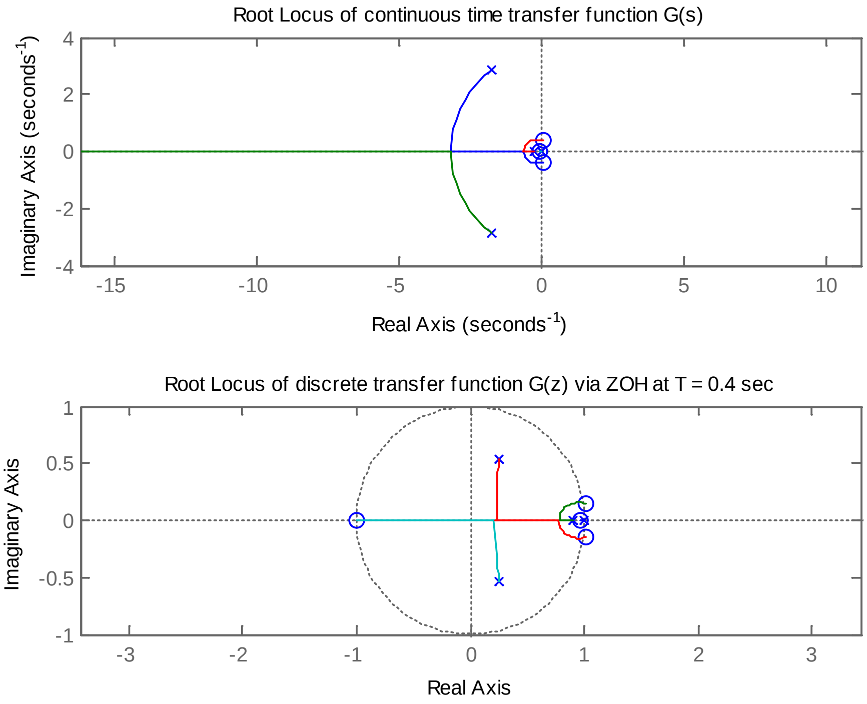


FIGURE 32: Root locus of G(s) and G(z) for indirect method

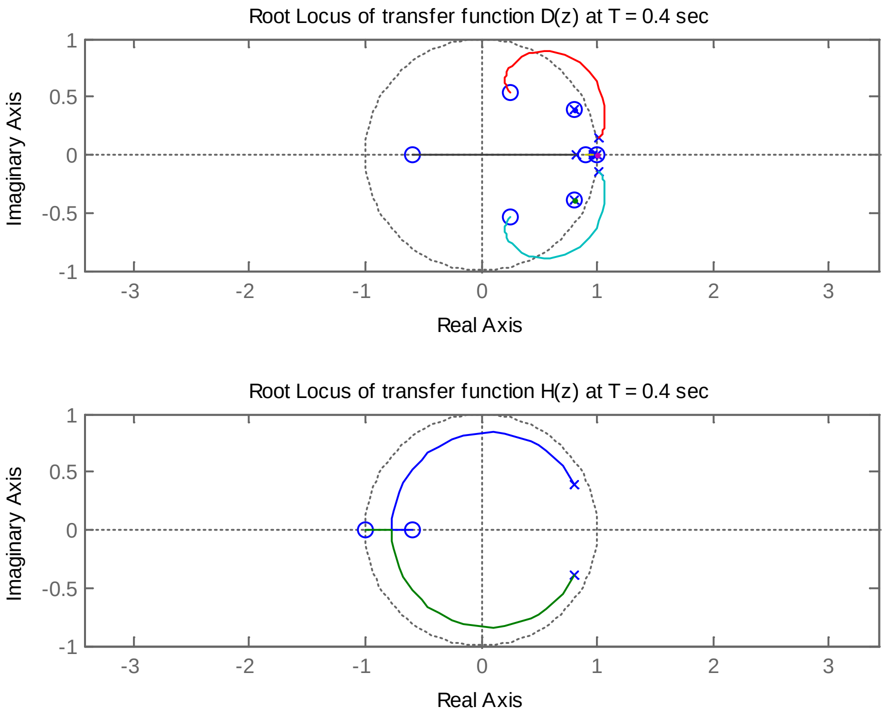


FIGURE 33: Root locus of D(z) and H(z) for indirect method

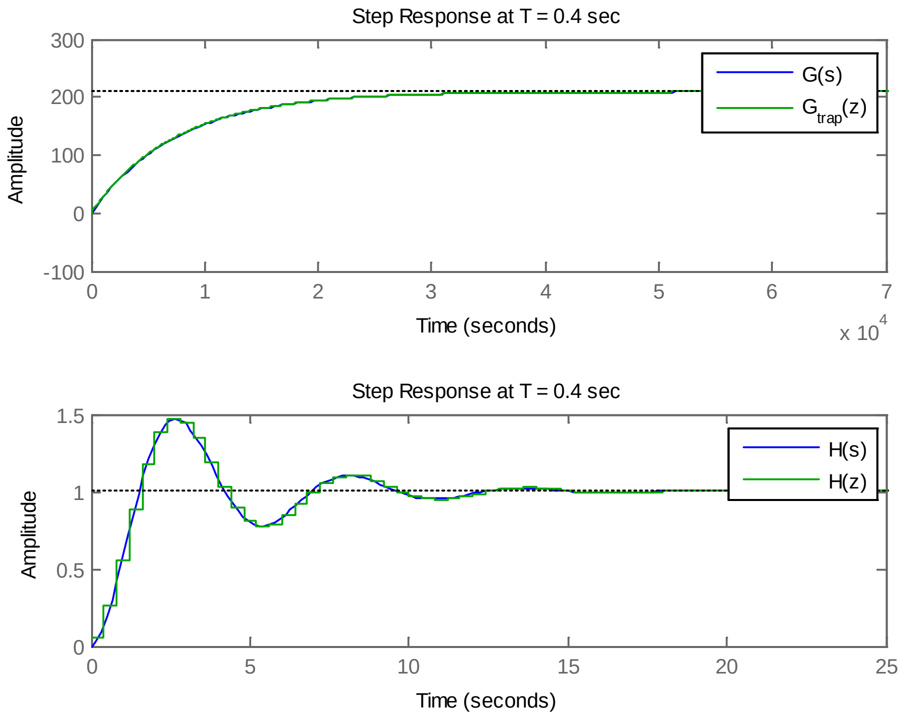


FIGURE 34: Step response of G(s) & G(z) and H(s) & H(z) via indirect method

4.2 Direct Digital Control

Section 4.2 provides an overview of the direct digital control method and presents simulation results of applying this method to the reduced order SISO PZR model that was presented in Section 2.3.

4.2.1 Direct Digital Control Modeling

An overview of the Direct Solution Approach (i.e., G is transformed from “s” to “z” domain and the controller D designed in “z” domain) is presented below, as discussed per [36], [37], [38], [40], [41], [42]:

The continuous time transfer function G(s) can be obtained from the state space representation as follows:

$$\mathrm{G(s)} = \mathrm{C\Phi B} + \mathrm{D}$$
$$\Phi = [\mathrm{sI - A}]^{-1}$$

EQUATION 28

Discrete equivalent of G(s) via Zero Order Hold (ZOH):

$$\mathrm{G(z)} = \left(1 - z^{-1}\right) Z\left[\frac{\mathrm{G(s)}}{\mathrm{s}}\right]$$

EQUATION 29

The desired continuous time overall transfer function, H(s), characteristic equation:

$$\mathrm{CE(s)} = as^2 + bs + s = 0$$

EQUATION 30

Where, the desired roots of H(s) are denoted as $r_1(s)$ and $r_2(s)$.

The discrete equivalent CE(z) of the overall transfer function CE(s) via mapping:

$$\mathrm{CE(z)} = d\mathrm{z}^2 + ez + f = 0 \Rightarrow$$
$$\mathrm{CE(z)} = 1 + \frac{e}{d}\mathrm{z}^{-1} + \frac{f}{d}\mathrm{z}^{-2}$$

EQUATION 31

Therefore, the discrete overall transfer function, H(z), for stability with $K_v = 1$:

$$\mathrm{H(z)} = \frac{b_0 + b_1\mathrm{z}^{-1} + b_2\mathrm{z}^{-2} + b_3\mathrm{z}^{-3} \ldots + b_n\mathrm{z}^{-n}}{\mathrm{CE(z)}}$$

EQUATION 32

For the causality design constraint:

$$\mathrm{H}(z)\big|_{z=\infty} = 0 \Rightarrow$$
$$\therefore b_0 = 0$$

EQUATION 33

For a Step Input (to obtain Simultaneous Equation 1):

$$\mathrm{H}(z)\big|_{z=1} = 1 \Rightarrow$$
$$\mathrm{H}(1) = \frac{b_1 + b_2 + b_3 ... + b_n}{\mathrm{CE}(1)} = 1 \Rightarrow$$
$$\therefore b_1 + b_2 + b_3 ... + b_n = \mathrm{CE}(1)$$

EQUATION 34

For a Ramp Input (to obtain Simultaneous Equation 2):

$$\text{-}\mathrm{T}\frac{\mathrm{dH}}{\mathrm{dz}}\bigg|_{z=1} = \frac{1}{K_v} \Rightarrow \qquad T = 1 \quad \text{and} \quad K_v = 1$$
$$\frac{\mathrm{dH}}{\mathrm{dz}^{\text{-}1}}\bigg|_{z=1} = 1 = \frac{\mathrm{CE}(1)\left(b_1 + 2b_2 + 3b_3 ... + nb_n\right) - \mathrm{CE}(1)\left(\frac{e}{d} + 2\frac{f}{d}\right)}{\mathrm{CE}(1) \times \mathrm{CE}(1)} \Rightarrow$$

$$\frac{\mathrm{dH}}{\mathrm{dz}^{\text{-}1}}\bigg|_{z=1} = 1 = \frac{\left(b_1 + 2b_2 + 3b_3 ... + nb_n\right) - \left(\frac{e}{d} + 2\frac{f}{d}\right)}{\mathrm{CE}(1)} \Rightarrow$$
$$\therefore b_1 + 2b_2 + 3b_3 ... + nb_n = \mathrm{CE}(1) + \left(\frac{e}{d} + 2\frac{f}{d}\right)$$

EQUATION 35

For this problem, there are only two equations to satisfy (EQUATION 34 and EQUATION 35 for step and ramp input). Therefore, the unknowns for H(z) are b_1 and b_2. These unknowns can be obtained by solving the following simultaneous equations:

$$b_1 + b_2 = \mathrm{CE}(1)$$
$$b_1 + 2b_2 = \mathrm{CE}(1) + \left(\frac{e}{d} + 2\frac{f}{d}\right)$$
$$\begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}\begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} \mathrm{CE}(1) \\ \mathrm{CE}(1) + \left(\frac{e}{d} + 2\frac{f}{d}\right) \end{bmatrix} \Rightarrow \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}^{-1}\begin{bmatrix} \mathrm{CE}(1) \\ \mathrm{CE}(1) + \left(\frac{e}{d} + 2\frac{f}{d}\right) \end{bmatrix}$$

EQUATION 36

Therefore, the overall transfer function, H(z):

$$\mathrm{H(z)} = \frac{b_1\mathrm{z}^{-1} + b_2\mathrm{z}^{-2}}{1 + \frac{e}{d}\mathrm{z}^{-1} + \frac{f}{d}\mathrm{z}^{-2}}$$

$$\mathrm{H(z)} = \frac{b_1\mathrm{z} + b_2}{\mathrm{z}^2 + \frac{e}{d}\mathrm{z} + \frac{f}{d}}$$

EQUATION 37

Finally, the direct design solution for the discrete control transfer function D(z) is:

$$\mathrm{H(z)} = \frac{\mathrm{D(z)G(z)}}{1 + \mathrm{D(z)G(z)}} \Rightarrow$$
$$\mathrm{D(z)} = \frac{1}{\mathrm{G(z)}}\frac{\mathrm{H(z)}}{1 - \mathrm{H(z)}}$$

EQUATION 38

The controller transfer function D(z), in EQUATION 38, serves to negate the undesired poles and zeroes, which results in the desired overall transfer function H(z) (as shown by the step response plot in FIGURE 37). Therefore, as a check of the controller design (for unity) to verify that the poles equal the zeroes, the following expression (in EQUATION 39) is applied:

$$\mathrm{Unity} = 1 = \frac{\mathrm{H(z)}}{\frac{\mathrm{D(z)G(z)}}{1 + \mathrm{D(z)G(z)}}} = \frac{a + bz^{-1} + cz^{-2} + ... + mz^{-n}}{a + bz^{-1} + cz^{-2} + ... + mz^{-n}}$$

EQUATION 39

4.2.2 Direct Digital Control Simulation Results

MATLAB Script was utilized to obtain the indirect solution via trapezoidal integration. Shown below are the “direct design” figure plots for z = 0.8 +/- j0.4:

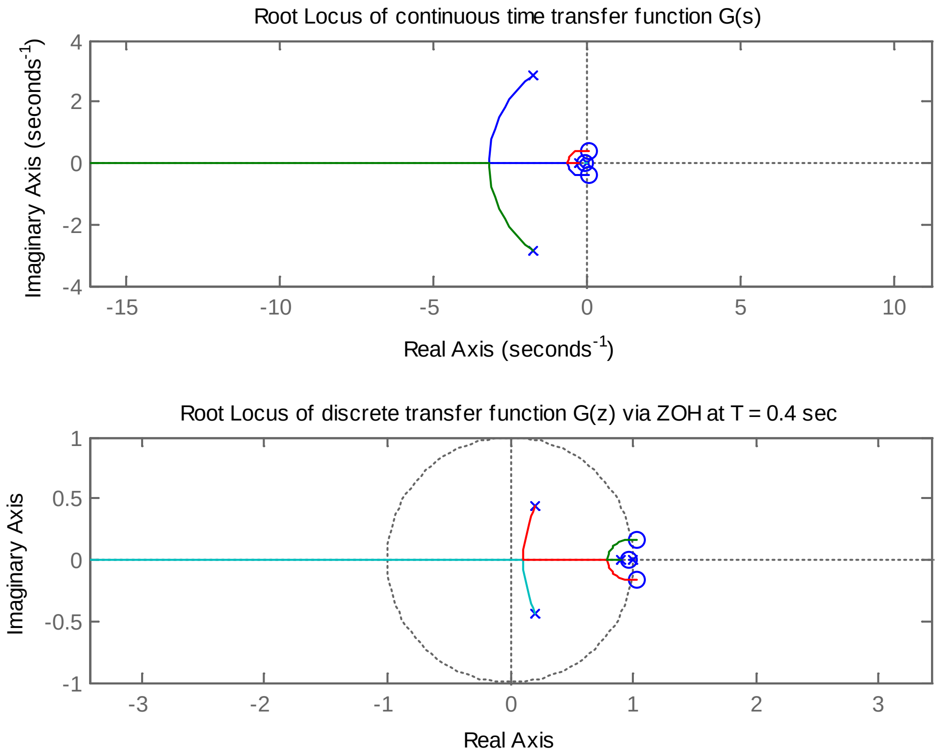


FIGURE 35: Root Locus Plot of G(s) and G(z) for Direct Method

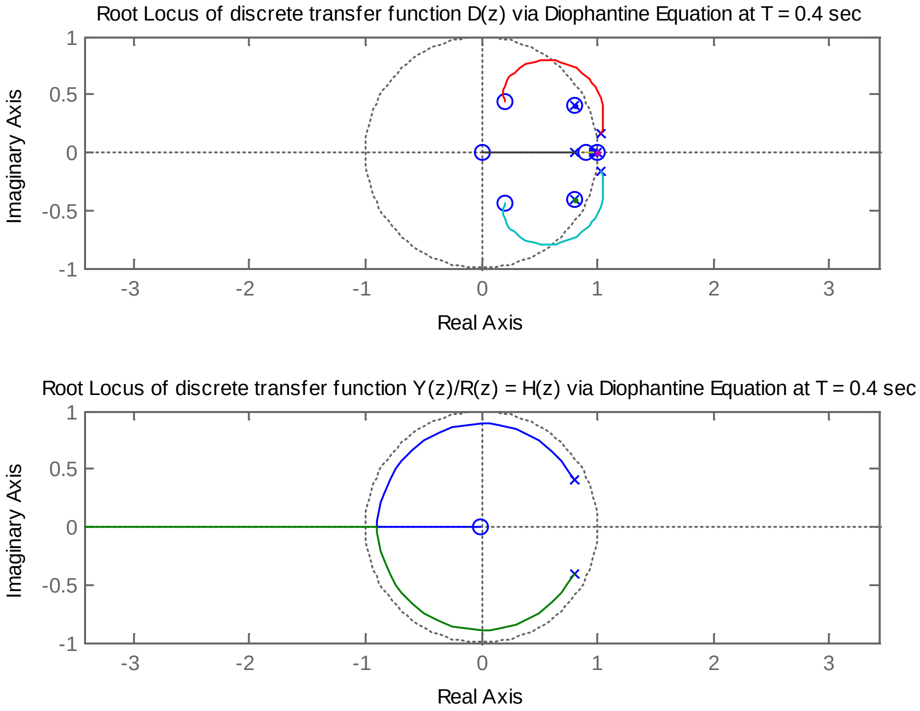


FIGURE 36: Root Locus Plot of D(z) and H(z) for Direct Method

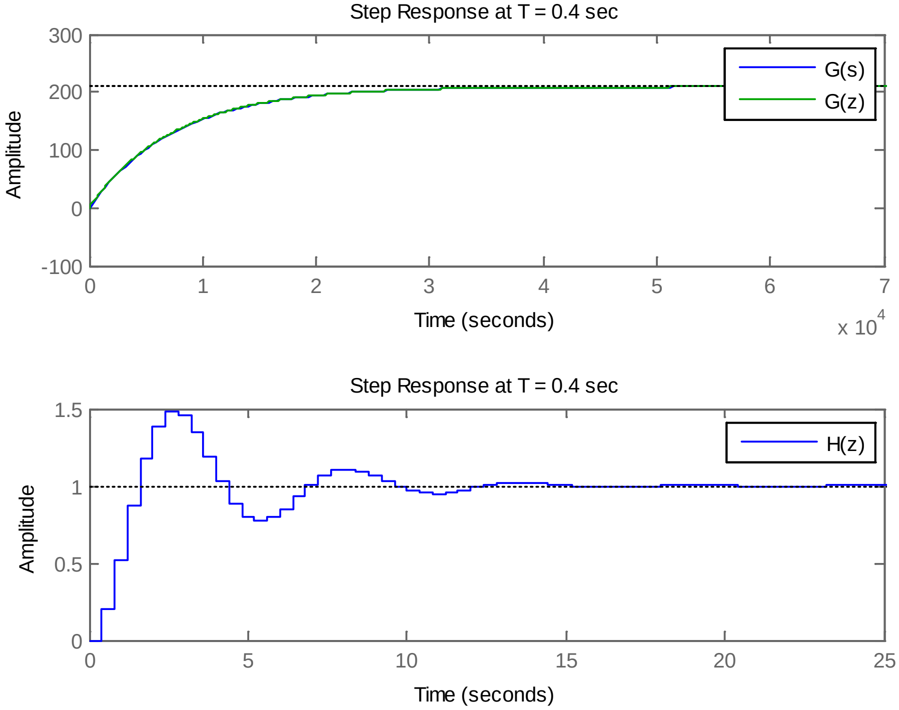


FIGURE 37: Step Response of G(s) & G(z) and H(z) for the Direct Method

4.3 Pole Placement (Ackermann's Formula) Control Design

Section 4.3 provides an overview of the pole placement (Ackermann's Formula) digital control method and presents simulation results of applying this method to the reduced order SISO PZR model that was presented in Section 2.3.

4.3.1 Pole Placement Control Modeling

1) Pole Placement via Ackermann's Formula (discrete state space control):

An overview of the pole placement method (Ackermann's Formula) is provided below, as developed per [36], [37], [38], [40], [41], [42]:

Start with the discrete state space model:

$x(k+1) = Ax(k) + Bu(k)$

$y(k) = Cx(k) + Du(k)$

EQUATION 40

Sometime the matrices “A”, “B”, “C”, and “D” are denoted as follows to differentiate from the continuous time model, since the matrices of the continuous time state space model are indeed different values than the discrete state space model respectively:

$\mathrm{A} = \Phi, \quad \mathrm{B} = \Gamma, \quad C = H, \quad \text{and } \mathrm{D} = \mathrm{J}$

EQUATION 41

However, in this document, they will be denoted as A, B, C, and D, since the discrete state space model is the only model discussed.

The control input “u(k)” is chosen as a linear combination of the states:

$u(k) = -K_1 \cdot x_1(k) - K_2 \cdot x_2(k) - \dots - K_n \cdot x_n(k) = -Kx(k)$

EQUATION 42

Where, “K” is the “gain matrix”.

$K = [K_1\ K_2\ K_3\ \dots\ K_n]$

EQUATION 43

Therefore, EQUATION 40 can be written as follows:

$x(k+1) = (A - BK)x(k) = A_f x(k)$

EQUATION 44

Where,

$A_f = (A - BK)$

EQUATION 45

Let the desired “z-domain” poles occur as follows:

$z = \lambda_1, \lambda_2, \lambda_2, \ldots, \lambda_n$

EQUATION 46

Therefore, the characteristic equation and polynomials of the closed loop system:

$0 = |zI - A_f| = |zI - (A - BK)| = |zI - A + BK| \Rightarrow$

$\alpha_c(z) = |zI - A + BK| = (z - \lambda_1)(z - \lambda_2)(z - \lambda_3)\ldots(z - \lambda_n)$

EQUATION 47

The system model can be expressed in control conical form as follows:

$$A = \begin{bmatrix} 0 & 1 & 0 & \ldots & 0 \\ 0 & 0 & 1 & \ldots & 0 \\ & & & \ddots & \\ -a_0 & -a_1 & -a_2 & \ldots & -a_{n-1} \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}$$

EQUATION 48

Therefore, the characteristic equation can be written as follows:

$|zI - A + BK| = z^n + (a_{n-1} + K_n)z^{n-1} + \ldots + (a_1 + K_2)z + (a_0 + K_1) = 0$

EQUATION 49

The desired characteristic equation can be stated as follows:

$\alpha_c(z) = z^n + \alpha_{n-1} z^{n-1} + \ldots + \alpha_1 z + \alpha_0$

EQUATION 50

Therefore, the gain coefficients are as follows:

$K_{i+1} = \alpha_i - a_i$, for i = 0, 1, 2, …, n-1

EQUATION 51

This leads to a simpler general solution, which is “Ackermann’s Formula” (as derived in [38]), where the plant model is as shown in EQUATION 40.

$x(k+1) = Ax(k) + Bu(k)$

EQUATION 52

The desired characteristic equation is:

$\alpha_c(A) = A^n + \alpha_{n-1} A^{n-1} + \ldots + \alpha_1 A + \alpha_0 I$

EQUATION 53

Therefore, the gain matrix K is computed as follows via “Ackermann’s Formula”:

$K = [0 \quad 0 \ldots 0 \quad 1][B \quad AB \ldots A^{n-2}B \quad A^{n-1}B]^{-1} \alpha_c(A)$

EQUATION 54

Per [41], the controller transfer function matrix “D_d” (not to be confused with the state space matrix D mentioned previously) for the pole placement method gain “K” without state feedback:

$D_d = -K\,[zI - A + BK]^{-1}B$

EQUATION 55

The controller transfer function:

$G_c = D_d\,G$

EQUATION 56

Where, G is the transfer function of the system plant. The overall closed loop transfer function of the system **(**H) with the controller for pole placement via gain “K” (obtained via Ackermann’s Formula):

$H = G_c / (1 + G_c)$

EQUATION 57

2) State Estimator Design – Predictor Estimator ([36] and [38]):

The method described above requires that all states be available for feedback. However, this is not always the case. In fact, a typical system may have several states that are not physically measurable. In this case, such states must be estimated. Two basic kinds of state estimators exist: (1) current estimators $\hat{x}(k)$ based on measurements y(k) and (2) predictor estimators $\bar{x}(k)$ based on measurements up to y(k-1). These can be implemented as full estimators where all states are estimated or in cases where some of the states are measureable, a reduced-order estimator is possible, which uses the measured states and estimates the unmeasured states. The full order predictor estimator is discussed below.

Recall from EQUATION 42 u(k) = -Kx(k). Therefore, for the predictor estimator, u(k) will be as follows, which allows the state feedback to be replaced with the predictor estimate:

$u(k) = -K\bar{x}(k)$

EQUATION 58

The state estimates could be obtained via a constructed model of the plant dynamics, since A, B, and u(k) are known, provided the initial state values are known (i.e., $x(0) = \bar{x}(0)$).

$\bar{x}(k+1) = A\bar{x}(k) + Bu(k)$

EQUATION 59

The state estimate error ($\tilde{x}$) is defined as follows. In some texts $\tilde{x}$ is denoted as ε.

$\tilde{x} = x - \bar{x}$

EQUATION 60

Therefore, substitution of EQUATION 40 and EQUATION 59 into EQUATION 60 produces the estimator error equation:

$\tilde{x}(k+1) = A\tilde{x}(k)$

EQUATION 61

Unfortunately, the estimator is running as an open loop without utilizing the continuous measurement so the system’s behavior, which is acceptable for an asymptotically stable system. However, for a marginally stable or unstable system, the error will never decrease from the initial error value. Therefore, as the system behavior changes, it diverges. As a solution, the measured output can be fed back and compared with the estimated output to correct the model with the error signal. This feedback system is constructed around the open-loop estimator and provides the output error as feedback via a feedback gain matrix, L_p, as shown in FIGURE 38 below adapted from [36].

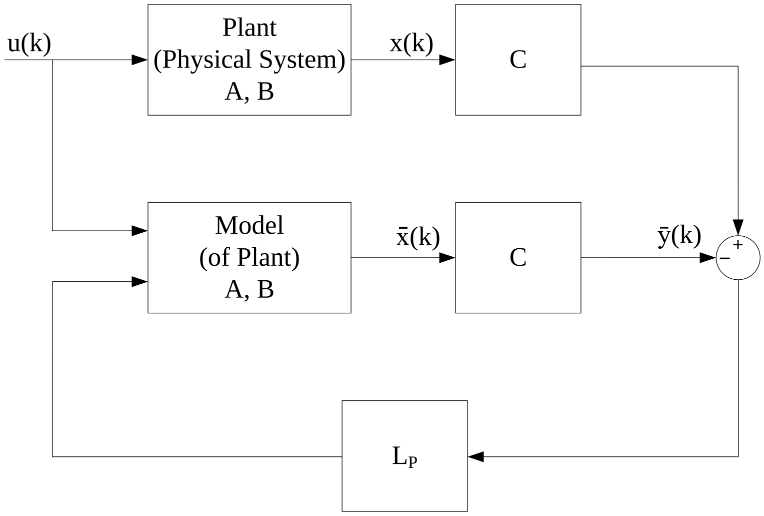


FIGURE 38: Close loop predictor estimator (adapted from [36])

Therefore, the predictor state estimation equation with measured output feedback included:

$\bar{x}(k+1) = A\bar{x}(k) + Bu(k) + L_p[y(k) - C\bar{x}(k)]$

EQUATION 62

The predictor estimator error equation is obtained via subtracting EQUATION 62 from EQUATION 40:

$\tilde{x}(k+1) = [A - L_pC]\tilde{x}(k)$

EQUATION 63

To choose quantities that produce a stable system with sufficiently small error, specify the desired estimator pole location in the z-plane to obtain the required estimator characteristic equation, similar to that shown in EQUATION 47, except B's are the desired estimator pole locations.

$(z - \lambda_1)\ (z - \lambda_2)\ (z - \lambda_3)\ldots\ (z - \lambda_n)$

EQUATION 64

Therefore, the predictor state estimation characteristic equation from the estimator error equation shown in EQUATION 63:

$|zI - A + L_pC| = 0$

EQUATION 65

Consequently, values for the output feedback gain matrix L_p must be obtained. It is observed that EQUATION 62 is in a similar form to the pole placement technique form shown earlier, with a solution technique via Ackermann's Formula (see EQUATION 54).

Therefore, Ackermann's formula can also be applied here to obtain the feedback gain matrix L_p, since the term $|zI - A - L_pC|$ in EQUATION 65 resembles $|zI - A - BK|$ in

EQUATION 49, with the exception of L_pC being in reverse order of the BK term. However, this is easily remedied, since the system dynamics of $(A-L_pC)$ are the same as $(A-L_pC)^T$.

$$L_p^T = [0 \;\; 0 \ldots 0 \;\; 1][C^T \;\; A^T C^T \ldots (A^T)^{(n-2)} C^T \;\; (A^T)^{(n-1)} C^T]^{-1} \alpha_c(A^T) \Rightarrow$$

$$L_p^T = [0 \;\; 0 \ldots 0 \;\; 1][C^T \;\; (CA)^T \ldots (CA^{(n-2)})^T \;\; (CA^{(n-1)})^T]^{-1} \alpha_c^T(A) \Rightarrow$$

$$L_p = \alpha_c\,(A)\,\{[C^T \;\; (CA)^T \ldots (CA^{(n-2)})^T \;\; (CA^{(n-1)})^T]^T\}^{-1} [0 \;\; 0 \ldots 0 \;\; 1]^T \Rightarrow$$

$$L_p = \alpha_e\,(A) \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-2} \\ CA^{n-1} \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}$$

EQUATION 66

In EQUATION 66, the desired characteristic polynomial for the estimator dynamics is denoted as α_e (A), which is similar to EQUATION 50. Per [40], the controller transfer function matrix "D_d" (not to be confused with the state space matrix D mentioned previously) for the pole placement method with full predictor estimator state feedback:

$$D_d = K\,[zI - A + BK + L_pC]^{-1}L_p$$

EQUATION 67

Then, as before, the controller transfer function:

$$G_c = D_d\,G$$

EQUATION 68

Where, G is the transfer function of the system/plant. Additionally, the overall closed loop transfer function of the system (H) with the controller for pole placement via gain "K" (obtained via Ackermann's Formula) with full predictor estimator state feedback:

$H = G_c / (1 + G_c)$

EQUATION 69

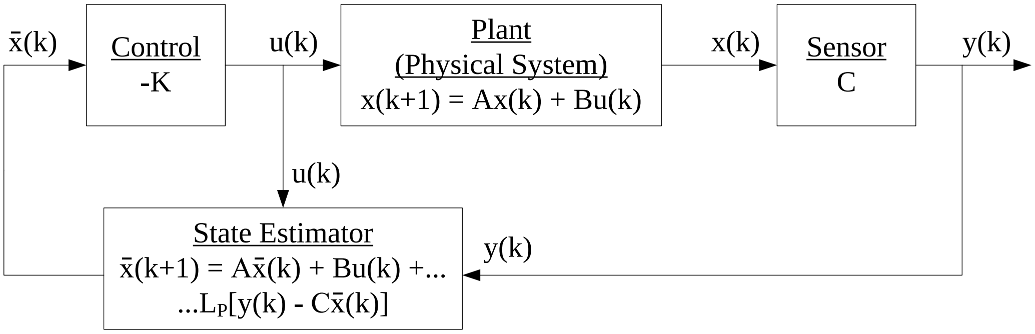


FIGURE 39: Close loop system with predictor estimator (adapted from [36])

3) State Estimator Design – Current Estimator ([36] and [38]):

This section covers current estimators $\hat{x}(k)$ based on measurements y(k), as mentioned in the previous section. The predictive estimator is not the most accurate technique, since the current control value is not dependent on the current error [38]. The “current” estimator attempts to resolve this issue, when the computation time for EQUATION 62 is relatively short in comparison to the sample period.

Recall from before with the predictor estimator that the state estimates could be obtained. The same can be obtained for the “current” state estimates via a constructed model of the plant dynamics, since A, B, and u(k) are known, provided the initial state values are known (i.e., $x(0) = \hat{x}(0)$).

FIGURE 40 provides a graphic illustration of how the “current” estimator functions. The new portion identified is the “current” feedback gain matrix, L_C, as shown in FIGURE 40 below, which was adapted from [36].

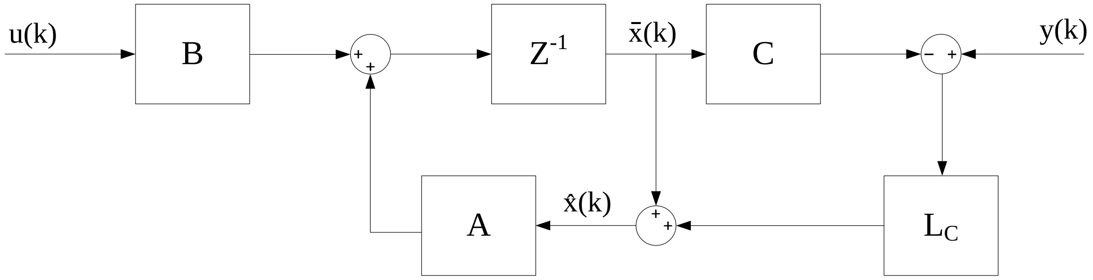


FIGURE 40: Close loop current estimator (adapted from [36])

Therefore, modifying EQUATION 62 with the current estimates “y(k)”, results in the following. The current state estimation equation with measured output feedback included:

$\hat{x}(k) = A\bar{x}(k) + L_C[y(k) - C\bar{x}(k)]$

EQUATION 70

Where,

$\bar{x}(k) = A\hat{x}(k-1) + Bu(k-1)$

EQUATION 71

However, this exact estimator is not physically possible, since it requires sampling, calculation, and output to occur simultaneously without dely. Therefore, the following estimate approximation is performed to provide an implementable solution.

Substitution of EQUATION 70 into EQUATION 71 yields the following for the current state estimation equation with measured output feedback included:

$\bar{x}(k+1) = A\bar{x}(k) + Bu(k) + AL_C[y(k) - C\bar{x}(k)]$

EQUATION 72

The current estimator error equation is obtained via subtracting EQUATION 72 from EQUATION 40:

$\tilde{x}(k+1) = [A - AL_pC]\bar{x}(k)$

EQUATION 73

By observation of the above equations with the comparable “predictor” estimator equations reveals that the “current” estimator feedback gain matrix L_C and “predictor” estimator feedback gain matrix “L_p” are related by the following relationship:

$L_p = AL_C$

EQUATION 74

By further reduction, the estimator error can be stated as the following expression:

$\tilde{x}(k+1) = [A - L_CCA]\tilde{x}(k)$

EQUATION 75

Therefore, as with the predictor estimator, Ackermann’s formula can also be applied here to obtain the feedback gain matrix L_C, with the primary difference between EQUATION 66 and EQUATION 76 being that the current estimator uses CA, instead of just A in the first term, as shown below:

$$L_C = \alpha_e\,(A)\begin{bmatrix} CA \\ CA^2 \\ \vdots \\ CA^{n-2} \\ CA^{n-1} \end{bmatrix}^{-1}\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}$$

EQUATION 76

In EQUATION 76, the desired characteristic polynomial for the estimator dynamics is denoted as α_e (A), which is similar to EQUATION 50. Per [40], the controller transfer function matrix “D_d” (not to be confused with the state space matrix D mentioned previously) for the pole placement method with full current estimator state feedback:

$D_d = zK\,[zI – A + L_CCA + BK – L_CCBK]^{-1}L_C$

EQUATION 77

Then, as before, the controller transfer function:

$G_c = D_d\,G$

EQUATION 78

Where, G is the transfer function of the system/plant.

Additionally, the overall closed loop transfer function of the system (H) with the controller for pole placement via gain “K” (obtained via Ackermann’s Formula) with full current estimator state feedback:

$H = G_c / (1 + G_c)$

EQUATION 79

4.3.2 Pole Placement Control Simulation Results

MATLAB Script was utilized to obtain the pole placement (state space “Ackermann” Formula) solution with and without (predictor and current) state estimation. Shown below are the performance plots from the simulations. FIGURE 41 provides the root locus plots of G(s) and G(z). FIGURE 42 provides the root locus plots of D(z) and H(z) via pole placement without state estimation. FIGURE 43 provides the root locus plots of D(z) and H(z) via pole placement with predictor state estimation. FIGURE 44 provides the root locus plots of D(z) and H(z) via pole placement with current state estimation. FIGURE 45 provides the step response of G(s) and G(z) (top plot), the step response of H(z) for pole placement without state feedback (middle plot), and the step response of H(z) with predictor and current state estimation (bottom plot).

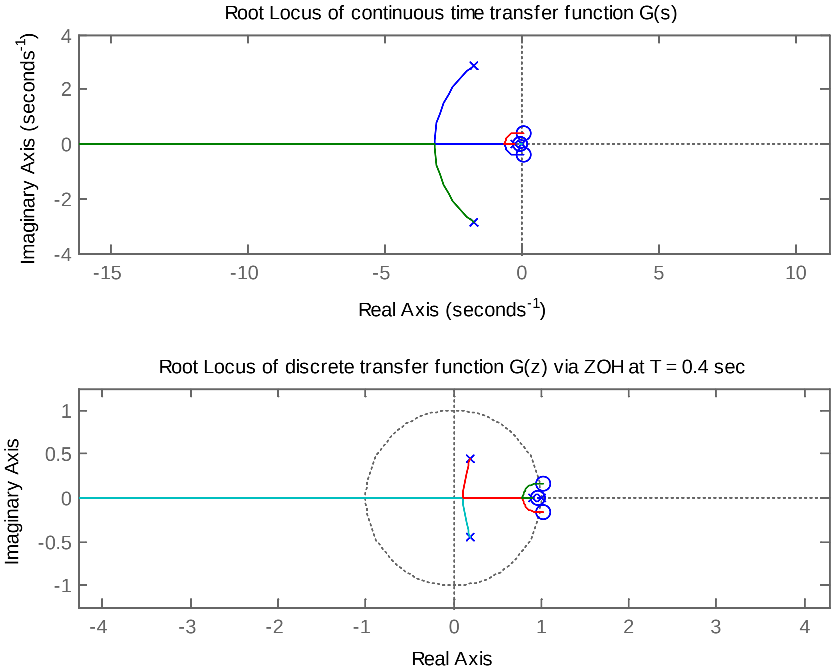


FIGURE 41: Root locus of G(s) and G(z)

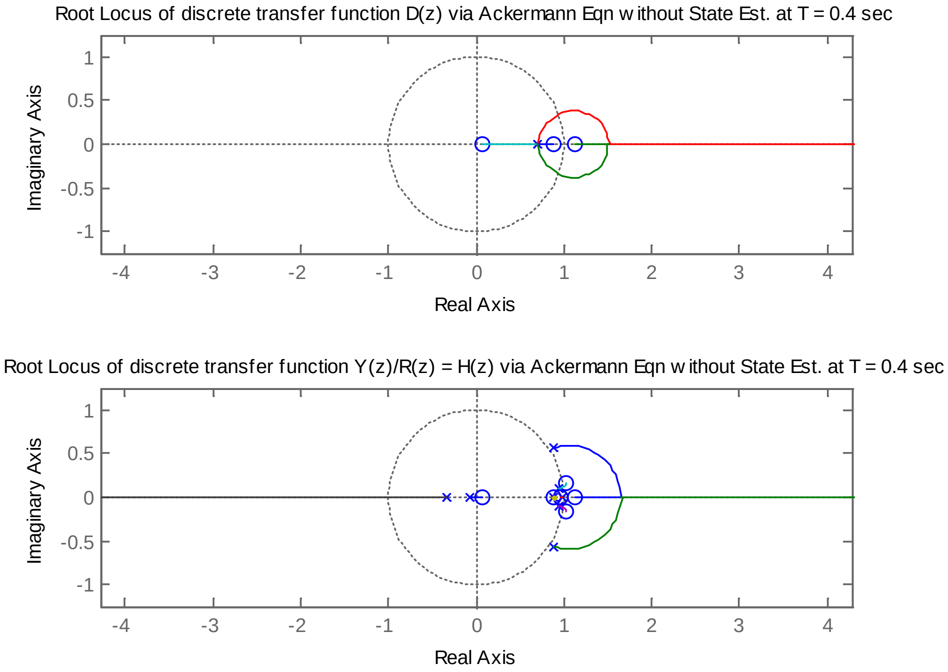
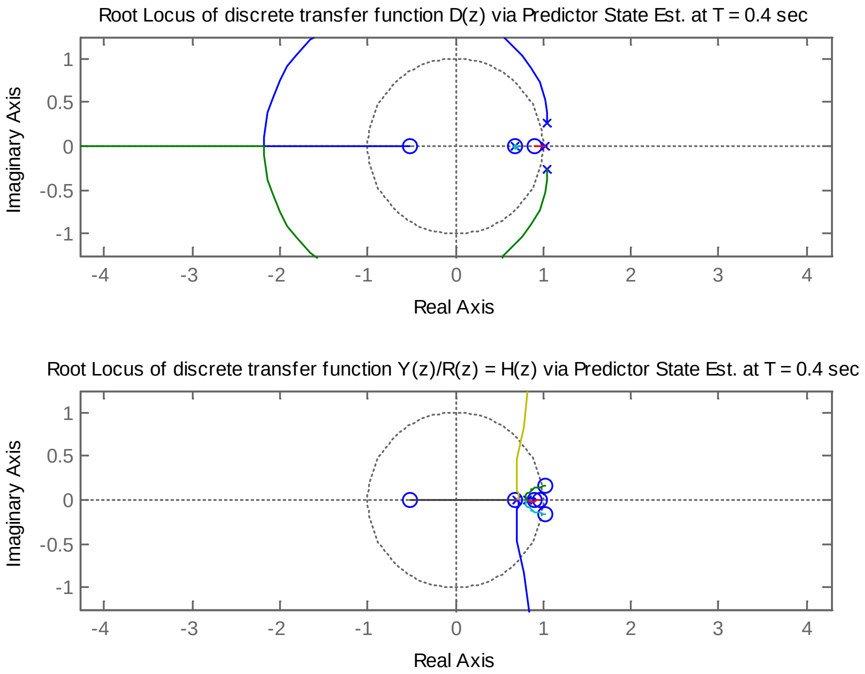
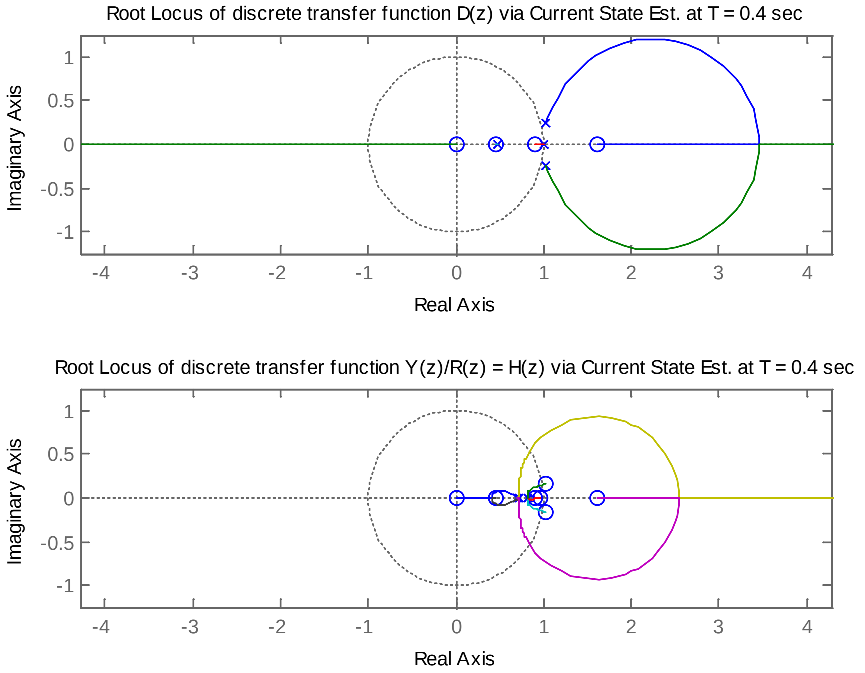


FIGURE 42: Root locus of D(z) & H(z) via pole placement without state estimation


Root Locus of discrete transfer function D(z) via Predictor State Est. at T = 0.4 sec

Root Locus of discrete transfer function Y(z)/R(z) = H(z) via Predictor State Est. at T = 0.4 sec

FIGURE 43: Root locus of D(z) & H(z) via pole placement with predictor state est.


Root Locus of discrete transfer function D(z) via Current State Est. at T = 0.4 sec

Root Locus of discrete transfer function Y(z)/R(z) = H(z) via Current State Est. at T = 0.4 sec

FIGURE 44: Root locus of D(z) & H(z) via pole placement with current state est.

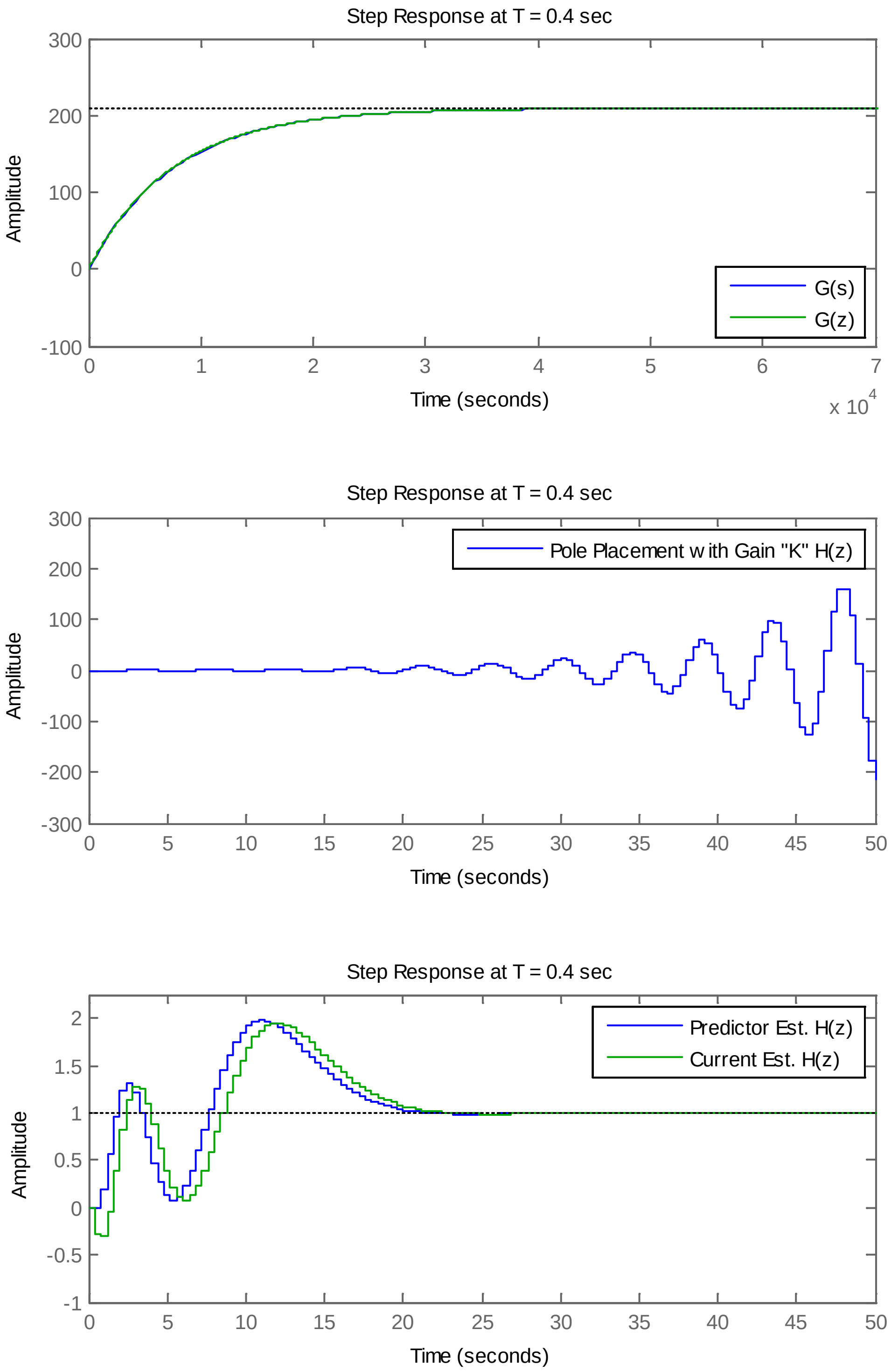


FIGURE 45: Step response of G(s) & G(z) (top), H(z) for pole placement without state feedback (middle), and H(z) with predictor est. & current est. (bottom)

4.4 Optimal Control with State Estimator

Section 4.4 provides an overview of the optimal control method and presents simulation results of applying this method to the reduced order SISO PZR model that was presented in Section 2.3.

4.4.1 Optimal Control with State Estimator Modeling

1) Optimal Control (state space design method for digital control in time domain):

The Optimal Control method is presented below, as developed per [36], [37], [38], [40], [41], [42]:

Start with the discrete state space model:

$x(k+1) = Ax(k) + Bu(k)$

$y(k) = Cx(k) + Du(k)$

EQUATION 80

Sometime the matrices "A" (transition matrix), "B" (control input matrix), "C" (measurement matrix), and "D" are denoted as follows to differentiate from the continuous time model, since the matrices of the continuous time state space model are indeed different values than the discrete state space model respectively:

$A = \Phi, \quad B = \Gamma, \quad C = H, \quad \text{and } D = J$

EQUATION 81

However, in this document, they will be denoted as A, B, C, and D, since the discrete state space model is the only model discussed.

The control input "u(k)" is chosen as a linear combination of the states "x(k)":

$u(k) = -K_1 \cdot x_1(k) – K_2 \cdot x_2(k) – \ldots – K_n \cdot x_n(k) = -K \cdot x(k)$

EQUATION 82

Where, “K” is the “gain matrix”.

$K = [K_1\ K_2\ K_3\ \ldots\ K_n]$

EQUATION 83

For optimal control the goal is to minimize a “cost function” (J), as shown in discrete form in EQUATION 84:

$$J = f\left(x(k), u(k)\right) = \frac{1}{2}\sum_{k=0}^{N}\left[x^{T}(k)\cdot Q\cdot x(k) + u^{T}(k)\cdot R\cdot u(k)\right] + \frac{1}{2}x_{n}^{T}\cdot H\cdot x_{n}$$

EQUATION 84

Where, “Q” and “R” are symmetric weighting matrices chosen by the designer:

Q = state-cost weighted matrix

R = control weighted matrix

H = final state-cost weighted matrix, which can be neglected if the final cost is zero

For convenience, Q can be chosen as a function of C:

$Q = C^{T}\cdot C$

EQUATION 85

R is chosen by the designer to obtain the desired performance characteristics.

The minimization of “*J*” is obtained via the following constraint:

$-x(k+1) + Ax(k) + Bu(k) = 0, \quad \text{for } k = 0, 1, \ldots, N$

EQUATION 86

The Lagrange multipliers method is utilized to solve this standard constrained-minima problem. The steps for this solution are summarized below:

$$J = \sum_{k=0}^{N} \begin{bmatrix} \frac{1}{2} x^T(k) \cdot Q \cdot x(k) + \frac{1}{2} u^T(k) \cdot R \cdot u(k). + .. \\ ...\lambda^T(k+1) \cdot \left(-x(k+1) + Ax(k) + Bu(k)\right) \end{bmatrix}$$

EQUATION 87

The control equations:

$$\frac{\partial J'}{\partial u(k)} = u^T(k) \cdot R \cdot u(k) + \lambda^T(k+1) \cdot B = 0$$

EQUATION 88

The state equations:

$$\frac{\partial J'}{\partial \lambda(k+1)} = -x^T(k+1) + A \cdot x(k) + B \cdot u(k) = 0$$

EQUATION 89

The adjoint equations:

$$\frac{\partial J'}{\partial x(k)} = x^T(k)Q - \lambda^T(k) + \lambda^T(k+1) \cdot A = 0$$

EQUATION 90

The adjoint equations can be stated as the following expression (backward difference form):

$$\lambda(k) = A^T \lambda(k+1) + Q \cdot x(k)$$

EQUATION 91

Therefore, via Equations (7), (9), and (12):

$$x(k+1) = Ax(k) + Bu(k)$$

EQUATION 92

$$u(k) = -R^{-1}B^T\lambda(k+1)$$

EQUATION 93

$$\lambda(k+1) = A^{-T}\lambda(k) - A^{-T}Q \cdot x(k)$$

EQUATION 94

These equations present a set of coupled difference equations required for the optimal solution.

2) The Linear Quadratic Regulator (LQR) steady-state optimal control technique:

The discrete Linear Quadratic Regulator (LQR) produces an optimal control solution. The LQR solution is applicable to a linear system (EQUATION 80), with quadratic cost (EQUATION 84), where a regulator control is used (FIGURE 27). A steady state solution of the discrete LQR can be obtained as follows, which permits a solution of the algebraic Riccati equation via Hamilton's equations through eigenvector decomposition.

First, substitution of EQUATION 93 into EQUATION 89 produces:

$$x(k+1) = Ax(k) + B\left[-R^{-1}B^{T}\lambda(k+1)\right]$$

EQUATION 95

Second, simplifying EQUATION 94 produces:

$$\lambda(k+1) = A^{-T}\left[\lambda(k) - Q \cdot x(k)\right]$$

EQUATION 96

Last, Substitution of EQUATION 96 into EQUATION 95 produces:

$$x(k+1) = Ax(k) + B\left[-R^{-1}B^{T}A^{-T}\left[\lambda(k) - Q \cdot x(k)\right]\right]$$

EQUATION 97

In matrix form, EQUATION 95 and EQUATION 96 can be stated as the following:

$$\begin{bmatrix} x(k+1) \\ \lambda(k+1) \end{bmatrix} = \begin{bmatrix} \mathrm{A} + \mathrm{BR}^{-1}\mathrm{B}^{\mathrm{T}}\mathrm{A}^{-T}\mathrm{Q} & -\mathrm{R}^{-1}\mathrm{B}^{\mathrm{T}}\mathrm{A}^{-T} \\ \text{-}\,\mathrm{A}^{-T}\mathrm{Q} & \mathrm{A}^{-T} \end{bmatrix} \begin{bmatrix} x(k) \\ \lambda(k) \end{bmatrix}$$

EQUATION 98

EQUATION 98 is typically identified as the “Hamilton’s equations” or “Euler-Lagrange equations”, which can be express on compact form as:

$$\begin{bmatrix} x(k+1) \\ \lambda(k+1) \end{bmatrix} = H \begin{bmatrix} x(k) \\ \lambda(k) \end{bmatrix}$$

EQUATION 99

Where, “*H*” is the “control Hamiltonian” matrix:

$$H = \begin{bmatrix} \mathrm{A} + \mathrm{BR}^{-1}\mathrm{B}^{\mathrm{T}}\mathrm{A}^{-T}\mathrm{Q} & -\mathrm{R}^{-1}\mathrm{B}^{\mathrm{T}}\mathrm{A}^{-T} \\ \text{-}\,\mathrm{A}^{-T}\mathrm{Q} & \mathrm{A}^{-T} \end{bmatrix}$$

EQUATION 100

Therefore, since EQUATION 98 is linear and EQUATION 100 is constant, the eigenvalues in EQUATION 100 can be obtained via standard techniques (e.g., MATLAB function eig.m), which for a SISO system (as considered in this problem) permits the optimal values of “K” to be identified. Specifically, the optimal roots (i.e., eigenvalues of EQUATION 100) can be applied to the Ackermann’s formula pole placement method to obtain the corresponding gain values (K). However, for a MIMO system, the solution is more complicated, but is not addressed here, since it is not utilized in this problem. See Section 9.3 of [36] for details on addressing a MIMO system. Next, the properties of reciprocal roots of EQUATION 98 are discussed. EQUATION 91, EQUATION 92, and EQUATION 93 may be expressed as follows via the z-transform:

$$z\mathrm{X}(z) = \mathrm{AX}(z) + \mathrm{BU}(z)$$

EQUATION 101

$$\mathrm{U}(z) = -z\mathrm{R}^{-1}\mathrm{B}^{\mathrm{T}}\Lambda(z)$$

EQUATION 102

$$\Lambda(z) = \mathrm{Q} \cdot \mathrm{X}(z) + z\mathrm{A}^{T}\Lambda(z)$$

EQUATION 103

Substituting EQUATION 102 into EQUATION 101 and combining with EQUATION 103 for matrix form:

$$[0] = \begin{bmatrix} z\mathrm{I} - \mathrm{A} & \mathrm{BR}^{-1}\mathrm{B}^{\mathrm{T}} \\ -\mathrm{Q} & z^{-1}\mathrm{I} - \mathrm{A}^{-T} \end{bmatrix} \begin{bmatrix} \mathrm{X}(z) \\ z\Lambda(z) \end{bmatrix}$$

EQUATION 104

This is a similar form to EQUATION 98. Therefore, the Hamiltonian system roots satisfy EQUATION 105:

$$0 = \det \begin{bmatrix} z\mathrm{I} - \mathrm{A} & \mathrm{BR}^{-1}\mathrm{B}^{\mathrm{T}} \\ -\mathrm{Q} & z^{-1}\mathrm{I} - \mathrm{A}^{-T} \end{bmatrix}$$

EQUATION 105

Rearrangement of EQUATION 105 via row operation produces:

$$0 = \det \begin{bmatrix} z\mathrm{I} - \mathrm{A} & \mathrm{BR}^{-1}\mathrm{B}^{\mathrm{T}} \\ 0 & z^{-1}\mathrm{I} - \mathrm{A}^{-T} + \mathrm{Q}(z\mathrm{I} - \mathrm{A})^{-1}\mathrm{BR}^{-1}\mathrm{B}^{\mathrm{T}} \end{bmatrix}$$

EQUATION 106

EQUATION 106 contains a block-wise triangular matrix, which can be expressed as:

$$0 = \det[z\mathrm{I} - \mathrm{A}]\det\left[z^{-1}\mathrm{I} - \mathrm{A}^{-T} + \mathrm{Q}(z\mathrm{I} - \mathrm{A})^{-1}\mathrm{BR}^{-1}\mathrm{B}^{\mathrm{T}}\right]$$

EQUATION 107

EQUATION 107 can be used to demonstrate that det[A^T] = det[A], which means if a system has characteristic root (z_i), then its reciprocal (z_i^{-1}) is also a characteristic root. These extra corresponding roots are the coupled roots in EQUATION 98.

Therefore, returning to EQUATION 98 and EQUATION 100

$$H' = \begin{bmatrix} \mathrm{E}^{-1} & 0 \\ 0 & \mathrm{E} \end{bmatrix}$$

EQUATION 108

Where, E^{-1} corresponds to the stable eigenvalues (i.e., |z| < 1) and E the unstable eigenvalues (i.e., |z| > 1). The corresponding eigenvector matrix associated with the unstable eigenvalues (i.e., roots outside the unit circle).

$$\begin{bmatrix} X_0 \\ \Lambda_0 \end{bmatrix}$$

EQUATION 109

The corresponding eigenvector matrix associated with the stable eigenvalues (i.e., roots inside the unit circle).

$$\begin{bmatrix} X_l \\ \Lambda_l \end{bmatrix}$$

EQUATION 110

Or, as a combined matrix:

$$\mathrm{W} = \begin{bmatrix} X_l & X_0 \\ \Lambda_l & \Lambda_0 \end{bmatrix}$$

EQUATION 111

Therefore, *H*’ (normal mode) can be stated as:

$$H' = \mathrm{W}^{-1} H \mathrm{W}$$

EQUATION 112

$$\begin{bmatrix} x \\ \lambda \end{bmatrix} = \mathrm{W} \begin{bmatrix} x' \\ \lambda' \end{bmatrix} = \begin{bmatrix} X_l & X_0 \\ \Lambda_l & \Lambda_0 \end{bmatrix} \begin{bmatrix} x' \\ \lambda' \end{bmatrix}$$

EQUATION 113

$$\begin{bmatrix} x' \\ \lambda' \end{bmatrix}_N = \begin{bmatrix} \mathrm{E}^{-N} & 0 \\ 0 & \mathrm{E}^{\mathrm{N}} \end{bmatrix} \begin{bmatrix} x' \\ \lambda' \end{bmatrix}_0$$

EQUATION 114

EQUATION 113 and EQUATION 114 lead to the following:

$$\lambda(k) = \Lambda_1 \mathrm{X}_1^{-1}\mathrm{x(k)} = \mathrm{S}_\infty \mathrm{x(k)}$$

EQUATION 115

Therefore, the steady state solution of the “Riccati equation” (S_∞)

$$\mathrm{S}_\infty = \Lambda_1 \mathrm{X}_1^{-1}$$

EQUATION 116

Recall from EQUATION 82

$u(k) = -K_\infty \cdot x(k)$

Where, K_∞ is computed via the solution of the algebraic Riccati equation (S_∞):

$K_\infty = [R + B^T S_\infty B]^{-1} B^T S_\infty A$

EQUATION 117

Therefore, to summarize discrete LQR steady-state optimal controls technique:

1) Compute the eigenvalues of “*H*” in EQUATION 100
2) Compute the eigenvectors associated with the stable eigenvalues of *H* (i.e., $|z| < 1$) in EQUATION 110
3) Compute the control gain “K_∞” from EQUATION 117 with S_∞ per EQUATION 116

3) The LQR optimal control technique (numerical solution):

The above LQR technique is valid for the steady state solution, assuming the system is linear time invariant (LTI) (i.e., A, B, C, Q, and R matrices in EQUATION 80 and

EQUATION 84 are constant) and stable. The non-steady state discrete LQR solution is summarized below per [20], [36], [95], [97], and [98]. The discrete LQR algorithm for solution of the discrete algebraic Riccati equation (DARE) is provided, which permits a recursive numerical solution approach (see APPENDIX A, “dlqr_fun.m”). This method can also be used as an alternative solution approach for the steady state solution of the DARE, but can be more computationally demanding [36]. As before, the LQR control law is shown in EQUATION 82 ($u_{LQR}(k) = -K(k)\cdot x(k)$), where “K(k)” is the discrete LQR feedback gain matrix, as shown in EQUATION 121. Therefore, in order to find the discrete LQR (optimal) control input (u_{LQR}), it is necessary to compute “K(k)”, as described in EQUATION 121 and EQUATION 122.

$$K(k) = \left[B^T \cdot P(k+1) \cdot B + R\right]^{-1} B^T \cdot P(k+1) \cdot A$$

EQUATION 118

$$P(k) = Q + K(k)^T RK(k) + \left[A - B \cdot K(k)\right]^T \cdot P(k+1) \cdot \left[A - B \cdot K(k)\right]$$

EQUATION 119

Where, “P(k)” in EQUATION 122 represents the solution of the DARE. P(k) is comparable to S_∞ in the steady state solution, but is obtain numerically by computing the solution backward in time from “N” time steps until a final value of is reached. The discrete LQR gain solution requires an iterative approach. The following steps can be used to implement the discrete LQR, where the iterations start at the end and go backward to the beginning, so that the current optimal gain value required in reaching an end state are determined (i.e., discrete LQR algorithm):

1) Initial Estimates (let k = N, P(N) = H)
2) Apply EQUATION 121 to compute the gain value “K(k)”

3) Apply EQUATION 122 to compute “P(k)”

4) Repeat steps “2” and “3” for N-1 → 0

4.4.2 Optimal Control with State Estimator Simulation Results

MATLAB Script was utilized to obtain the optimal control solution. Shown below are the “optimal design” figure plots:

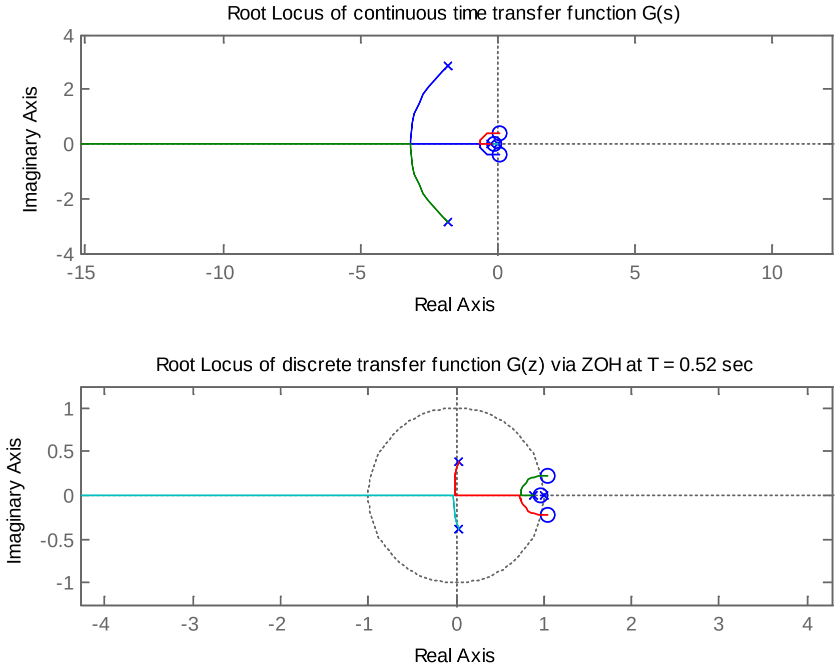


FIGURE 46: Root locus of G(s) and G(z)

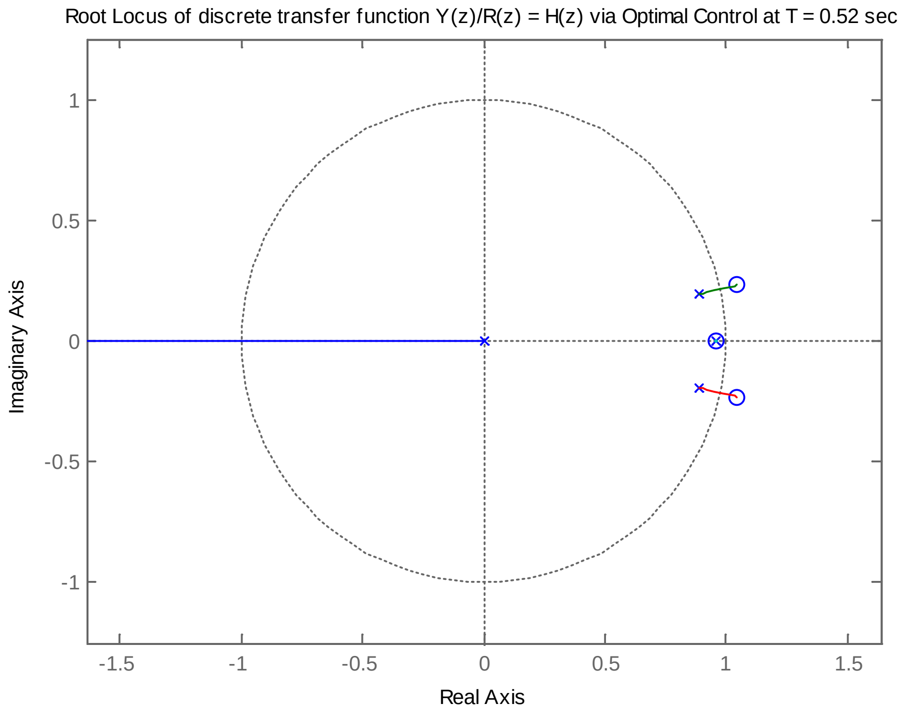


FIGURE 47: Root locus of H(z) via optimal control

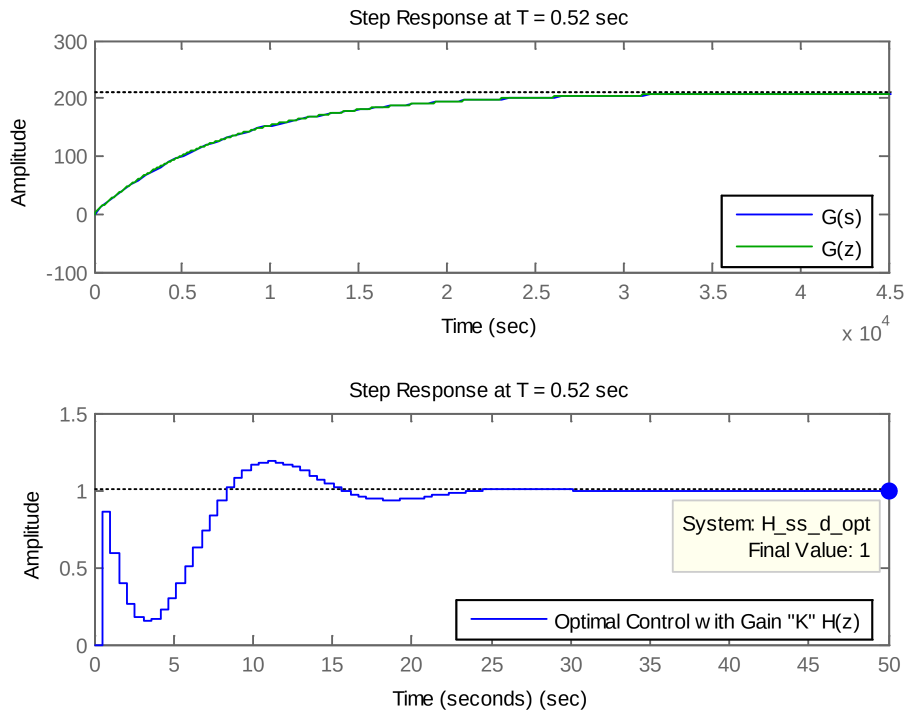


FIGURE 48: Step response of G(s) & G(z) and H(z) via optimal control

4.5 Comparison of Digital Control Method for Reduced Order PZR Model

A simulated fourth order PZR model was developed that has a 91.5 percent fit with the experimental data utilized (see Section 2.3). This reduced order model was used for the digital control simulations presented in CHAPTER 4. The open loop system plant (i.e., no feedback control) exhibits a stable, non-oscillatory step response, with slow rise time to reach steady state (i.e., around 5 hours). The design for four types of digital control methods was presented (i.e., indirect, direct, Ackermann’s pole placement with estimated state feedback, and optimal control), with a summary for each provided below.

1) Indirect:

The indirect design method produced a controller that enabled a stable closed loop transfer function H(z) unit step response, as shown in FIGURE 33 and FIGURE 34.

2) Direct:

Similar to the indirect method, the direct design method produced a controller that enabled a stable closed loop transfer function H(z) unit step response, as shown in FIGURE 36 and FIGURE 37.

The closed loop system performance characteristics were similar for both the indirect and direct method designed controllers, with comparable rise time, percent overshoot, settling time, and steady state values.

3) Pole Placement with Estimator:

FIGURE 45 (produced from MATLAB Script) provides the step response of:

- The system plant without control “G(s) and G(z)” (i.e., open loop response),

- The closed loop transfer function H(z) for control using the pole placement technique with “Ackermann’s Formula” applied to find the feedback gain “K”, and
- The closed loop transfer function H(z) for the pole placement technique with (a) a “predictor” state estimator feedback controller and (b) a “current” state estimator feedback controller.

A reduced order estimator was not utilized in this design, since the PZR model is simulated with physically undefined state variables, which may not be physically measureable. In a physical PZR system, most (if not all) state variables typically utilized in a model would all be physically measureable. Therefore, a reduced order estimator would be applicable.

As seen in FIGURE 45, the system plant “G(s)” exhibits a stable, non-oscillatory behavior, but has very slow rise time to reach steady state, which is consistent with the pole placement in the “s-domain”, as shown in FIGURE 41.

When the pole placement technique for control was applied without state estimation feedback (i.e., gain “K”), the closed loop transfer function H(z) was unstable, with the error increasing over time and exhibiting oscillatory behavior, as shown in the middle plot in FIGURE 45, which is consistent with the pole placement shown in FIGURE 42 (i.e., two poles are outside the unit circle).

However, when the “predictor” estimator was included in the controller, the system response became stable, with less desirable rise time, overshoot, and settling time in comparison to the other methods as shown in the bottom plot in FIGURE 45, which is consistent with the pole placement shown in FIGURE 43 (i.e., all “z-domain” poles for H(z) are within the unit circle).

Last, the “current” estimator was implemented in the controller, which resulted in slightly slower system response over the “predictor” estimator, but it did have slightly less overshoot and approximately equal settling time, as shown in the bottom plot of FIGURE 45, which is consistent with the pole placement shown in FIGURE 44. All “z-domain” poles are within the unit circle. Additionally, there is one more “zero” added to the forward path of the “current” estimator, than occurs in the “predictor” estimator controller. This added “zero” tends to lead to the system behaving more stably with greater bandwidth, faster rise time, and less overshoot [42], which is generally exhibited in this case, as shown in FIGURE 45, with the exception of rise time. Of additional interest, are the zeroes that occur outside the unit circle (non-minimum phase), which are responsible for the system behavior of oscillating a negative response before settling to the unit step value of one, as shown in FIGURE 45. This non-minimum phase behavior is further discussed in the “Optimal Control Performance Summary” below.

4) Optimal Control Performance Summary:

The discrete time optimal control method was presented. The optimal design method produced a controller that enabled a stable closed loop transfer function H(z) unit step response, as shown in FIGURE 47 and FIGURE 48.

As was the case with the indirect, direct, and Ackermann’s formula pole placement methods, the optimal control method produced improved closed loop system performance characteristics (i.e., rise time, percent overshoot, settling time, and steady state value). Of additional interest, are the zeroes that occur outside the unit circle (non-minimum phase), as shown in FIGURE 47, which are responsible for the system behavior response (from 0 to 3 seconds) before settling to the unit step value of one, as shown in FIGURE 48. This non-minimum phase behavior is common in these types of thermal-hydraulic systems. For example, power plant steam generators also exhibit non-minimum behavior, as discussed in [43]. Of additional interest for PZR performance is the magnitude of the overshoot. The optimal method produced the least amount of overshoot (even though it exhibited non-minimum phase behavior for this system). For comparison, the overshoot for the direct and indirect methods are more than twice the amount of that for optimal method.

5) Comparison of Methods:

All four methods considered produced generally acceptable system performance characteristics, especially regarding stability. If additional performance constraints were imposed, a particular method may become more desirable (i.e., the direct controller design method produces a controller that does not need to sample as fast as the indirect method controller to maintain performance and stability). Additionally, some of the controllers may be more technically feasible or less expensive to implement than another, which would drive the design solution. Therefore, baring other constraints, the “optimal” controller produced the best overall system performance characteristics for overshoot. As such, the “optimal” control design was investigated further for application to the dynamic MIMO PZR model (see Section 2.5 for the dynamic MIMO PZR model and Section 5.3 the optimal controller application to this model).

CHAPTER 5: CONTROLS FOR DYNAMIC PZR MODEL

Section 2.5 presented a suitable control-oriented dynamic thermal-hydraulic pressurizer model to simulate control strategies (e.g., dynamic MIMO PZR model). In CHAPTER 5, the pressure behavior of the dynamic PZR model without any control is reviewed. This chapter also reviews and analyzes the applicability of the following existing (i.e., textbook) control methodologies for a pressurizer in a pulsed system using the dynamic MIMO PZR model: (1) optimal control and (2) adaptive control. As seen in Section 1.3, intelligent model identification and control (e.g., adaptive neural network) has been previously investigated with documented results; therefore, it is not considered further in CHAPTER 5. However, this chapter places additional focus on making the controller adaptive to dynamic system behavior. TABLE 7 provides a list and definition of the control terms and symbols used.

The sections in this chapter are detailed as follows:

- Section 5.1 – PZR Behavior with Open Loop
- Section 5.2 – PZR Behavior with Conventional (On/Off or PID) Control
- Section 5.3 – PZR Behavior with Adaptive Optimal (LQR) Control
- Section 5.4 – PZR Behavior with Adaptive Self-Tuning (STR) PID
- Section 5.5 – PZR Behavior with Adaptive Minimum Variance (MV) Control
- Section 5.6 – Primary Issues Associated with Existing PZR Control Methods

TABLE 7: Control terms

Symbol	**Description**
r	command or reference input(s) (e.g., desired pressure)
u	input(s) to the system (e.g., $\dot{m}_{surge}$, $\dot{m}_{spray}$, etc.)
u_c	actuator or control input(s) to the system (e.g., $\dot{m}_{spray}$)
y	outputs of the system (e.g., PZR pressure)
$\hat{y}$	estimated output of the system
y_m	identified model (e.g., RLS ID) estimated system output
ε	error of system output to reference (ε = r - y)
ε_{ym}	error of identified model output (y_m) to estimated system output ($\hat{y}$) ($\varepsilon_{ym} = \hat{y} - y_m$)
w	disturbance to the system (e.g., $\dot{m}_{surge}$)
x	state variable of the system (e.g., PZR pressure, PZR level , PZR temperature, etc.)
$\dot{x}$	first derivative of the state variables with respect to time (dx/dt)

5.1 Dynamic PZR Model Pressure Behavior with No Control (Open Loop)

In this section the results of dynamic PZR model without any controls are considered. The system is modeled as shown in FIGURE 49, where the commanded input (u_c) is zero for the simulation. See TABLE 7 for a definition of the control terms and symbols. Results of the simulation with no control (open loop) are provided in FIGURE 50 (PZR pressure and level output) and FIGURE 51 (PZR spray and heater input). The $\dot{m}_{surge}$ (system disturbance "w") for the simulation is provided in FIGURE 18.

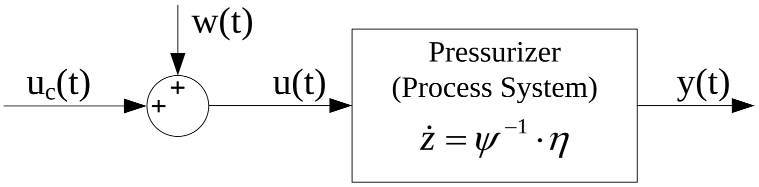


FIGURE 49: Dynamic PZR model – no control (open loop)

The results from the dynamic PZR model pressure behavior with no control:

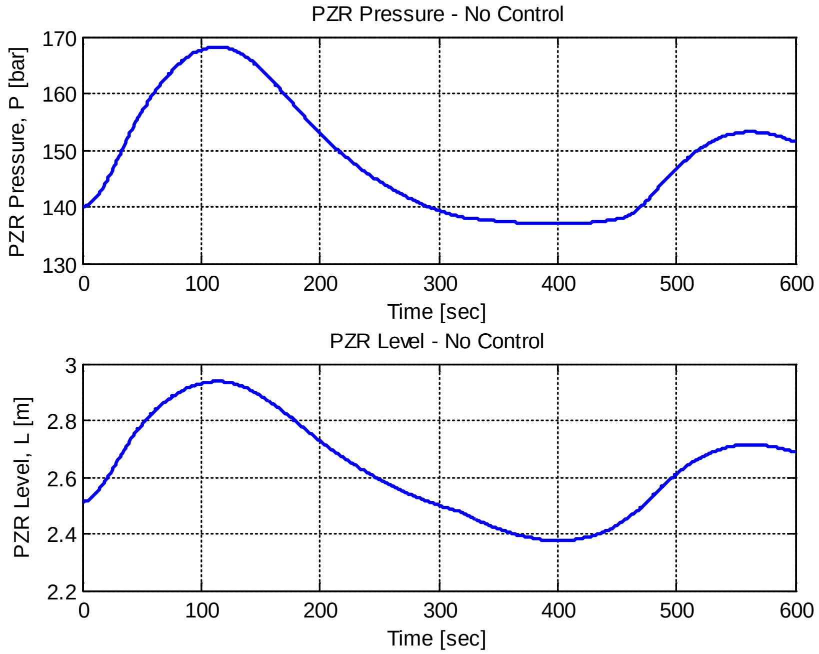


FIGURE 50: Dynamic PZR model pressure and level – no control

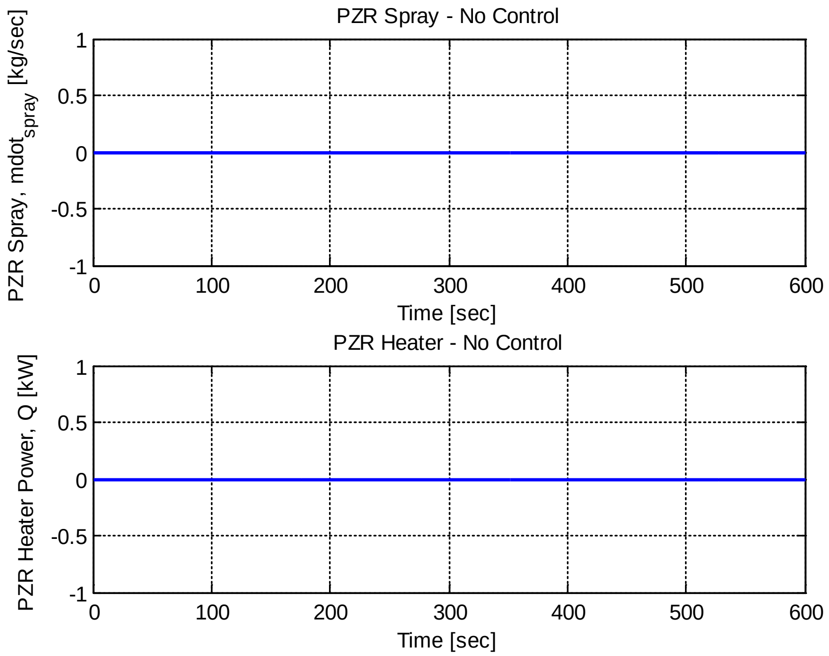


FIGURE 51: Dynamic PZR model spray and heater input – no control

5.2 Dynamic PZR Model Pressure Behavior with Conventional Control

In this section the results of dynamic PZR model with conventional controls are considered. The system is modeled as shown in FIGURE 52. The Simulink model is provided in FIGURE 26 of Section 2.5. See TABLE 7 for a definition of the control terms and symbols.

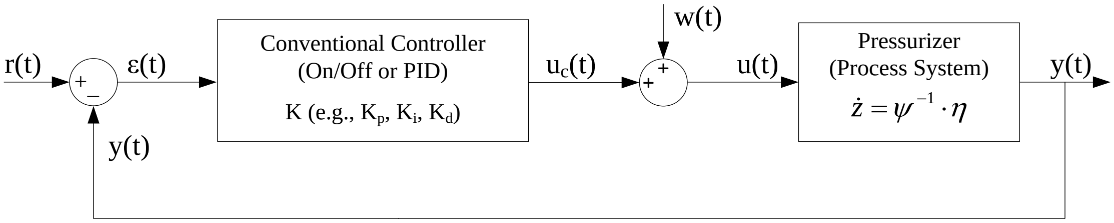


FIGURE 52: Dynamic PZR model – conventional control

See Section 3.3 for results of the Dynamic PZR Model Pressure Behavior with Conventional Controls.

5.3 Optimal Control for Dynamic PZR Model

In this section and subsections the modeling and simulation results for the dynamic PZR model with several versions of optimal controls are considered:

- Section 5.3.1 considers a traditional optimal controller for PZR spray, where the optimal gain “K” is a constant value.
- Section 5.3.2 considers updating the optimal gain “K” for PZR spray at each time step via computing a new state space input matrix “B” at each time step based on measured system values.
- Section 5.3.3 expands on the method discussed in Section 5.3.2 by adding the same method of optimal control used on the PZR spray to the PZR heaters.

- Section 5.3.4 summarizes the challenges in implementing optimal control for a dynamic PZR model.
- Section 5.3.5 considers an adaptive optimal controller that updates the state space system model via recursive least squares identification.
- Finally, Section 5.3.6 provides a performance review and conclusion of the optimal control methods considered in Section 5.3.

5.3.1 Optimal Control of Spray for Dynamic PZR Model: Constant "K" (Based on a Fixed State Space Model)

In this section the modeling and simulation results of the dynamic PZR model with traditional optimal control of PZR spray are considered. The optimal gain matrix "K" (EQUATION 83) is computed using EQUATION 117. The 4th order state space model of the PZR system (i.e., A, B, C, and D matrices of the system state space model) used to compute the optimal gain matrix "K" was obtained empirically, where the state variables of the system are physically measurable process variables (i.e., $[x_1, x_2, x_3, x_4]$ = [pressure, level, temperature, volume]). Details of this empirical 4th order state space model are discussed in Section 5.3.2. Therefore, since the state space model is held constant, the optimal gain "K" stays constant (i.e., this controller does not act adaptively with changes to the system). The system with optimal control is modeled as shown in FIGURE 53. The Simulink model is similar to that provided in FIGURE 68 and FIGURE 70, except it does not have the system identification (i.e., "RLS ID"), associated "THETA" input to the optimal controller, or $\dot{x}(t)$ (state variable derivative) input to the optimal controller (see Section 5.3.5). See TABLE 7 for a definition of the control terms and symbols.

Results of the simulation with optimal control are provided in FIGURE 54 (PZR pressure and level output) and FIGURE 55 (PZR spray and heater input). The $\dot{m}_{surge}$ (system disturbance “w”) for the simulation is provided in FIGURE 18. Discussion of the results and comparison with the other control methods is provided in Section 5.3.6.

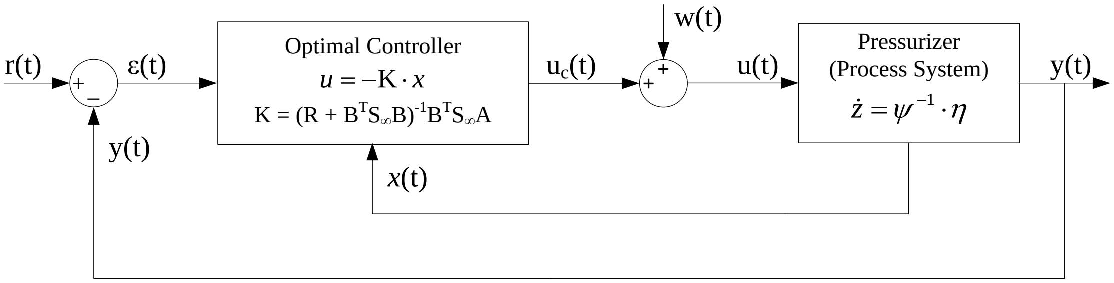


FIGURE 53: Dynamic PZR model – optimal control

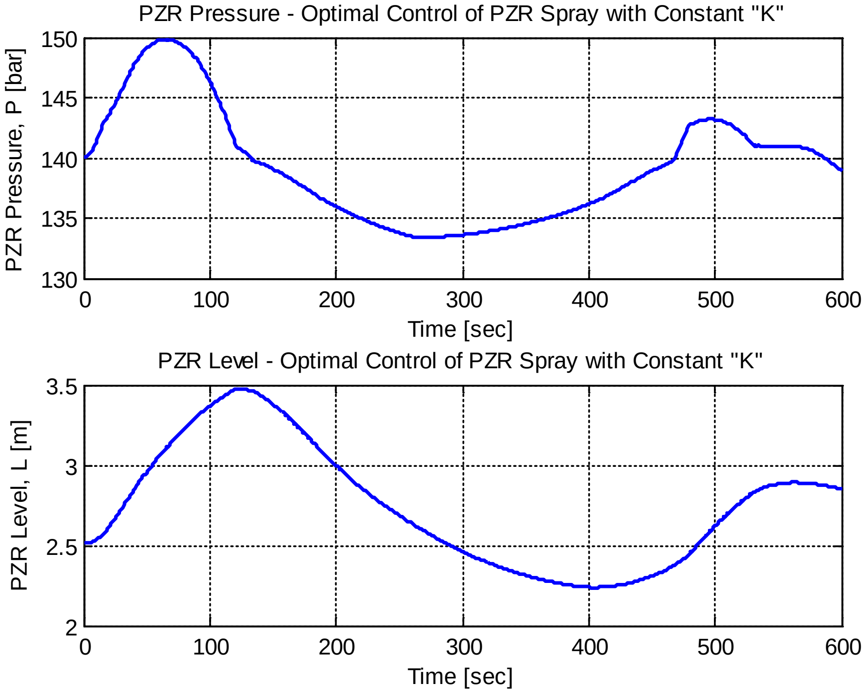


FIGURE 54: Dynamic PZR model P & L – LQR of PZR spray with constant “K”

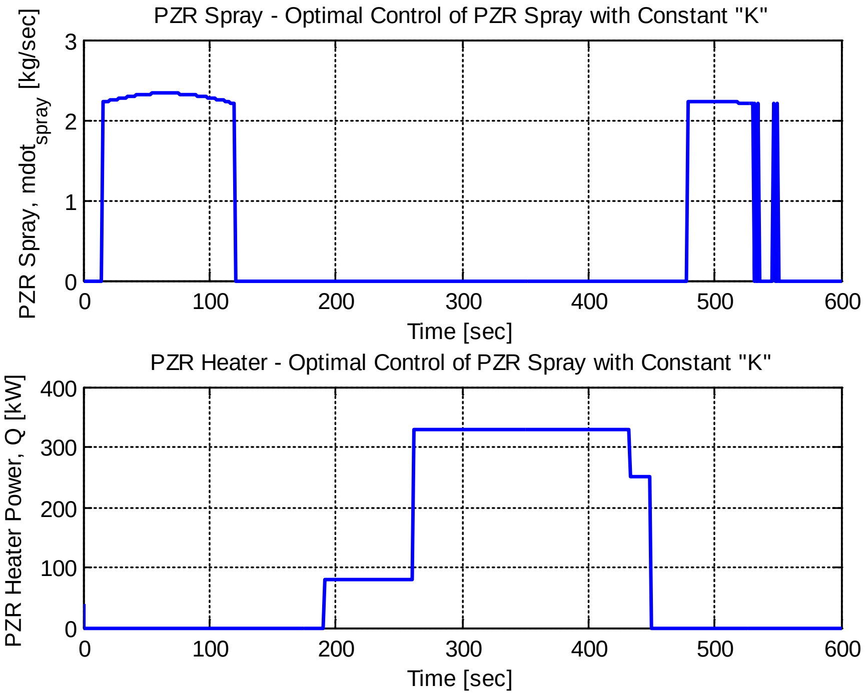


FIGURE 55: Dynamic PZR model spray and heater input – LQR of PZR spray with constant “K”

5.3.2 Optimal Control of Spray for Dynamic PZR Model: New “K” at each time step via Constant “A” and New “B” at Each Time Step, where B = *f*(A, x_i)

One of the challenges with the optimal controller presented in Section 5.3.1 is that it does not adapt as the system dynamic change during a transient, since the physical system is non-linear. However, it is assumed that treating the system as linear for a finite time interval (i.e., the time step) will adequately capture the system dynamics. In this section the modeling and simulation results of the dynamic PZR model with an optimal controller for PZR spray that updates the optimal gain matrix “K” at each time step by computing a new “B” state space matrix at each time step are considered. The system is modeled as shown in FIGURE 56. The Simulink model is similar to that provided in FIGURE 68 and FIGURE 70, except it does not have the system identification (i.e., “RLS

ID”) or associated “THETA” input to the optimal controller (see Section 5.3.5). See TABLE 7 for a definition of the control terms and symbols.

In order to help the state space (SS) model of the system adapt to the changing dynamics of the PZR system during a transient, the SS input “B” matrix at each time step is updated based on measurable state variable. The SS dynamic “A” matrix is held constant (as obtained in the SS model estimate below). Therefore, a new optimal gain “K” can be computed at each time step that more closely reflects the actual non-linear dynamics of the PZR physical system, as shown in EQUATION 121 by rearranging EQUATION 120. The system pressure output (u) is a physically measurable quantity and is also one of the state variables. Additionally, the system state variables selected (x_i) are physically measurable process variables (i.e., [x_1, x_2, x_3, x_4] = [pressure, level, temperature, volume]). Therefore, by measuring the PZR pressure, level, temperature, and volume at each time step and using the estimated SS dynamic “A” matrix, a new SS input “B” matrix can be computed via EQUATION 121.

$$\dot{x}(t) = Ax(t) + Bu(t)$$

$$y(t) = Cx(t) + Du(t)$$

EQUATION 120

$$B = [\dot{x}(t)-Ax(t)]u(t)^{-1}$$

EQUATION 121

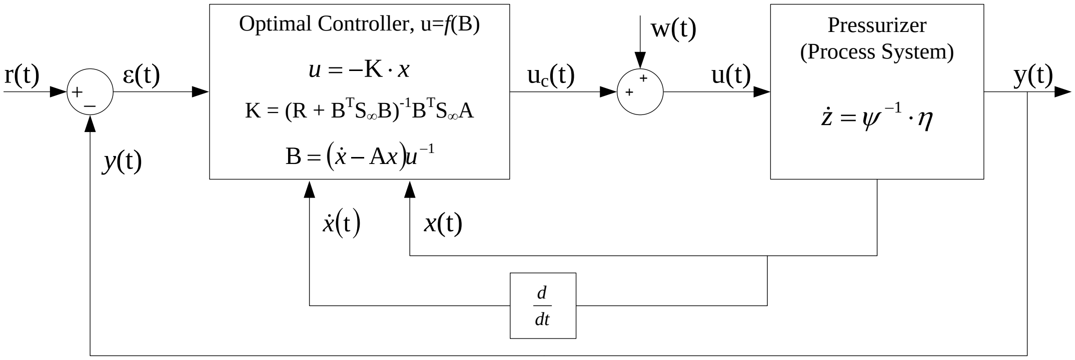

r(t)
+
−
ε(t)
y(t)
Optimal Controller, u=f(B)
$u = -K \cdot x$
$K = (R + B^T S_\infty B)^{-1} B^T S_\infty A$
$B = (\dot{x} - Ax)u^{-1}$
$\dot{x}(t)$
$x(t)$
$u_c(t)$
w(t)
+
+
u(t)
Pressurizer
(Process System)
$\dot{z} = \psi^{-1} \cdot \eta$
y(t)
$\frac{d}{dt}$

FIGURE 56: Dynamic PZR– LQR with new “K” via new “B” each cycle

1) Empirical Fourth Order PZR State Space Model:

Shown below is an empirically based 4th order state space (SS) PZR model (see APPENDIX B). This SS model serves to provide an estimate of the SS “A” (dynamic) matrix and “B” (input) matrix. This 4th order SS model is based on (simulated) empirical data from four PZR state variables (pressure, level, temperature, and volume) and exhibits a greater than 99 percent fit to the data. The eigenvalues of the SS “A” (dynamic) matrix are all less than or equal to zero (i.e., left half of “s” plane). This indicates the empirically based SS system is marginally stable, which is consistent with the properties exhibited by the other reduced order SS models discussed in Sections 2.3 and 2.4. Therefore, it is sufficiently accuracy to capture the system behavior.

```
Time [sec]
x_State_Var_data = [1, 2, 3, 4]
1) Press,   [bar]
2) level,   [m]
3) T_3l = Temp_l, [deg C]
4) V_3v  [m^3]
u_Input_Ctrl_Var = [mdot_spray (kg/sec), Q_dot(kJ/sec),
mdot_valve(kg/sec)]
Mdot_Surge [kg/sec]

dT = 0.1000
x_size = 4
u_size = 1
sys_order = 4
```

```
sys =
  Continuous-time identified state-space model:
      dx/dt = A x(t) + B u(t) + K e(t)
       y(t) = C x(t) + D u(t) + e(t)

  A =
                x1          x2          x3          x4
   x1  -1.362e-05   2.523e-05   3.384e-05  -8.447e-06
   x2    0.005459   -0.001737   -0.001599  -0.0001972
   x3   -0.004414    0.001948    0.003162    0.003184
   x4    0.005199    -0.01009    -0.02774     -0.0165

  B =
                u1
   x1  -9.718e-08
   x2  -2.541e-05
   x3  -6.214e-05
   x4  -0.0001655

  C =
               x1         x2         x3         x4
   y1  1.044e+04       -155     -121.3       -112
   y2      203.8     -17.49     -14.21      1.578
   y3  2.519e+04      3.586     -27.12      3.747
   y4      228.4      4.729      18.66     0.1887

  D =
       u1
   y1   0
   y2   0
   y3   0
   y4   0

  K =
               y1          y2          y3          y4
   x1  6.394e-05  -0.002407   0.001372  -0.000559
   x2   -0.04107      -1.08    0.03244    -0.8558
   x3    0.01666     0.6503   -0.02945      1.661
   x4    -0.2754      2.756     0.1412      1.721

Parameterization:
   FREE form (all coefficients in A, B, C free).
   Feedthrough: none
   Disturbance component: estimate
   Number of free coefficients: 52
   Use "idssdata", "getpvec", "getcov" for parameters and their
uncertainties.

Status:
Estimated using SSEST on time domain data "PZR_Data".
Fit to estimation data: [99.89;99.99;99.94;99.98]% (prediction
focus)
FPE: 2.664e-29, MSE: 8.776e-06

Eigenvalues of A [eig(sys.a)]:

Eigenvalues_A = [0.0000, -0.0015, -0.0044, -0.0092]
```

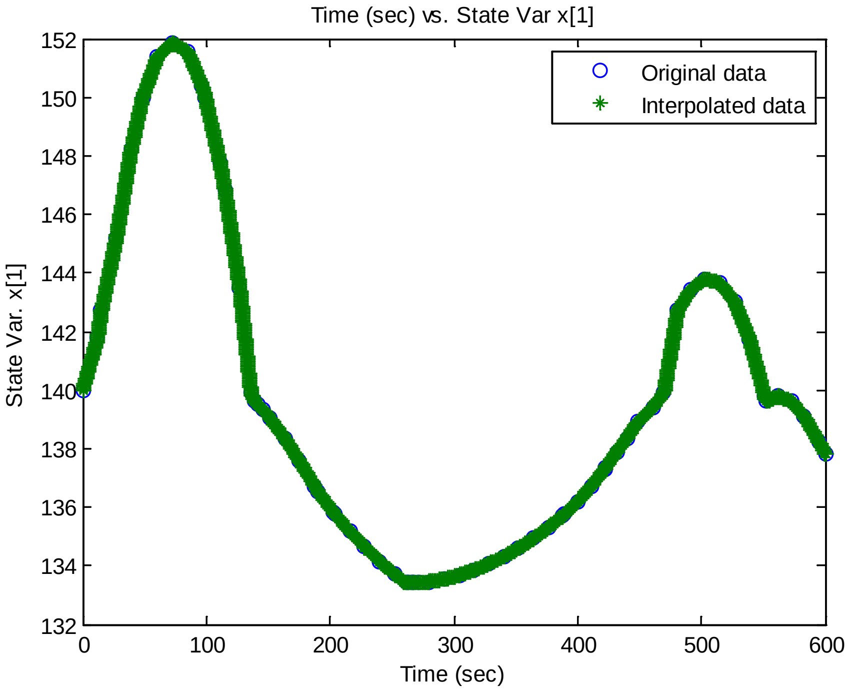


FIGURE 57: PZR pressure (x[1]) data used to construct PZR 4th order model

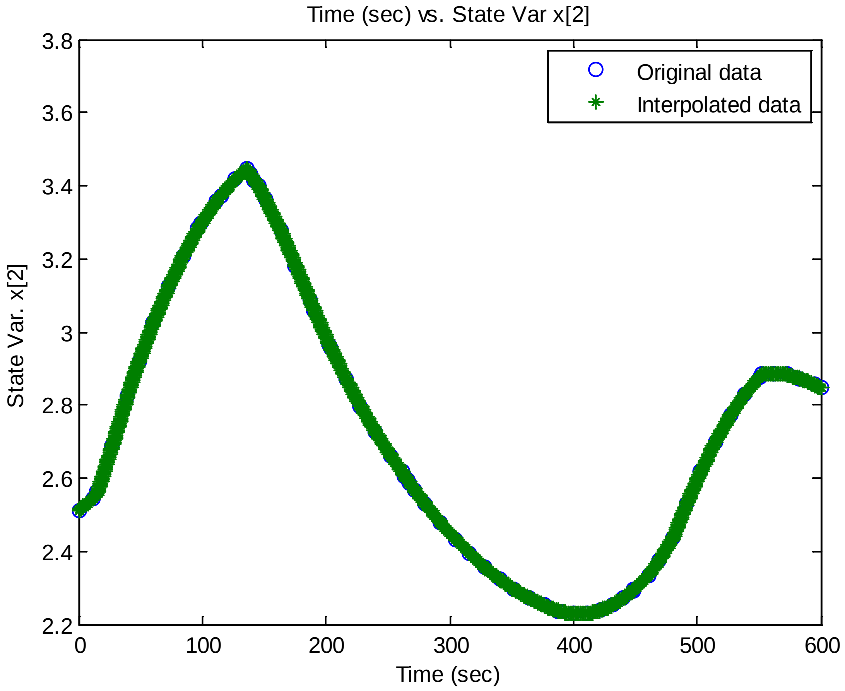


FIGURE 58: PZR level (x[2]) data used to construct PZR 4th model

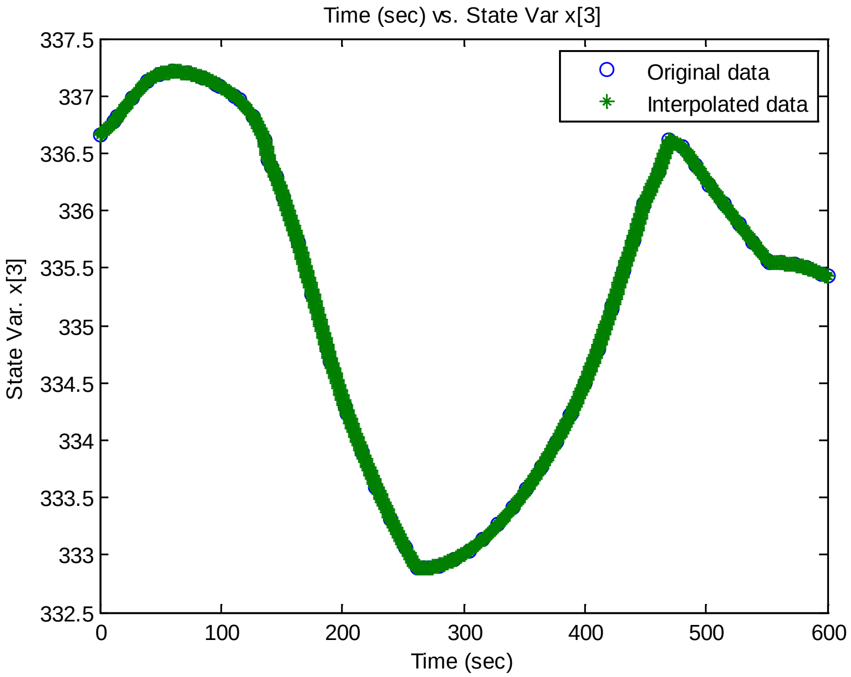


FIGURE 59: PZR temperature (x[3]) data used to construct PZR 4th order model

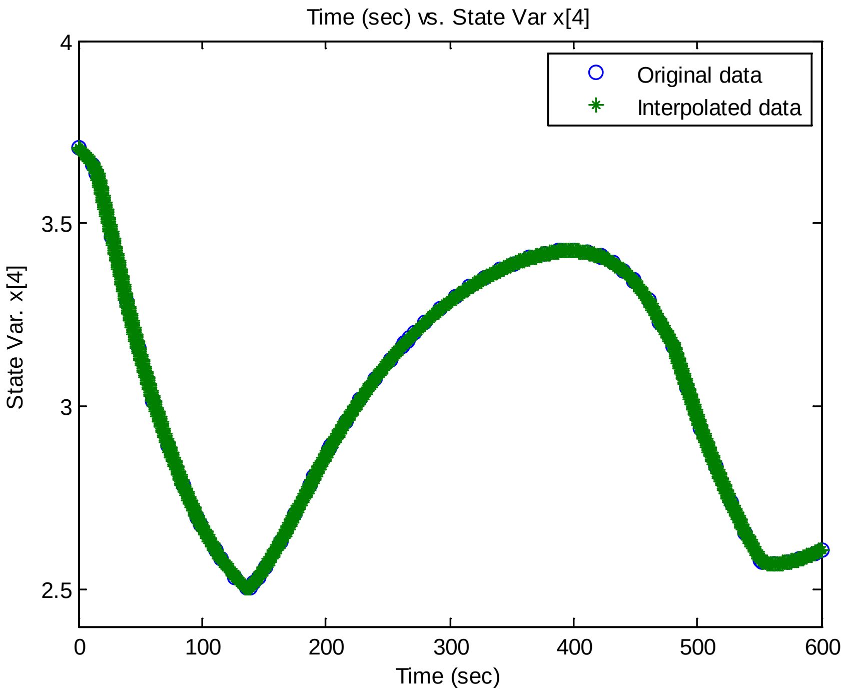


FIGURE 60: PZR volume (x[4]) data utilized to construct PZR 4th order model

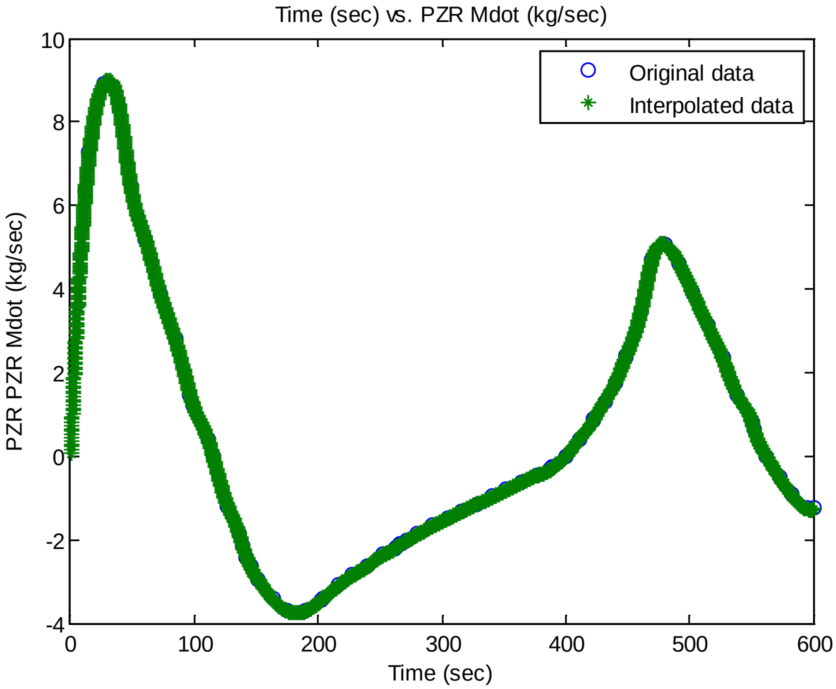


FIGURE 61: PZR surge mass flow rate data used to construct PZR 4th order model

Simulation Results – Optimal Control of Spray with New SS “B” Each Time Step:

Results of simulating the dynamic PZR model using optimal control of PZR spray that computes a new “K” and “B” matrix at each time step are provided in FIGURE 62 (PZR pressure and level output) and FIGURE 63 (PZR spray and heater input). The $\dot{m}_{surge}$ (system disturbance “w”) for the simulation is provided in FIGURE 18. Discussion of the results and comparison with the other control methods is provided in Section 5.3.6.

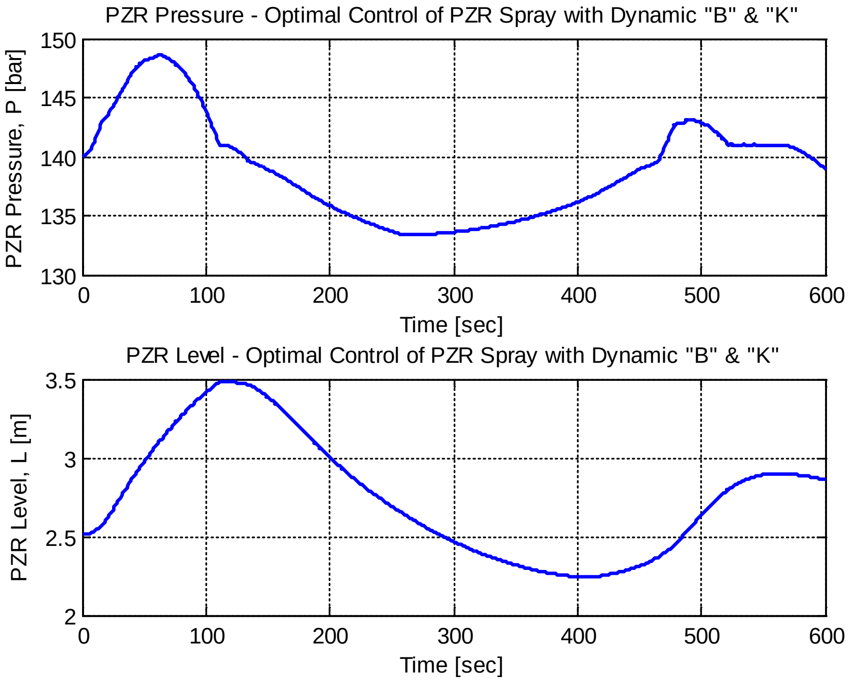


FIGURE 62: Dynamic PZR P & L – LQR of PZR spray with dynamic “B” & “K”

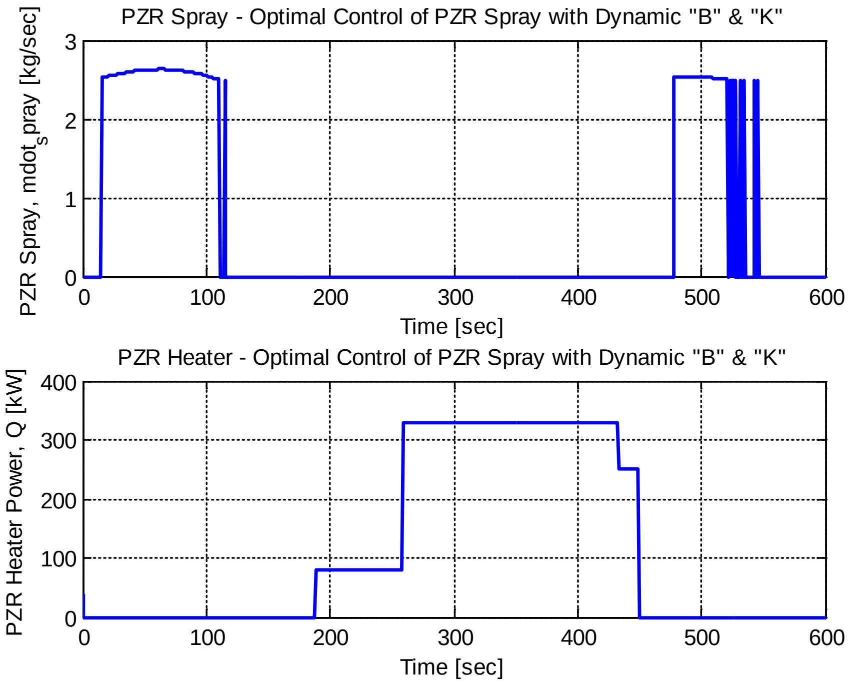


FIGURE 63: Dynamic PZR spray and heater input – LQR of PZR spray with dynamic “B” & “K”

5.3.3 Optimal Control of Spray and Heaters for Dynamic PZR Model: New “K” at each time step via Constant “A” and New “B” at Each Time Step, B = f(A, x_i)

In this section the modeling and simulation results of the dynamic PZR model with an optimal controller for PZR spray and PZR heaters that updates the optimal gain matrix “K” at each time step by computing a new “B” state space matrix at each time step are considered. The system is modeled as shown in FIGURE 56 using the same method presented in Section 5.3.2 (see EQUATION 121). The Simulink model is similar to that provided in FIGURE 68 and FIGURE 70, except it does not have the system identification (i.e., “RLS ID”) or associated “THETA” input to the optimal controller (see Section 5.3.5). See TABLE 7 for a definition of the control terms and symbols.

Results of simulating the dynamic PZR model using optimal control of PZR spray and PZR heaters that computes a new “K” and “B” matrix at each time step are provided in FIGURE 64 (PZR pressure and level output) and FIGURE 65 (PZR spray and heater input). The $\dot{m}_{surge}$ (system disturbance “w”) for the simulation is provided in FIGURE 18. Discussion of the results and comparison with the other control methods is provided in Section 5.3.6.

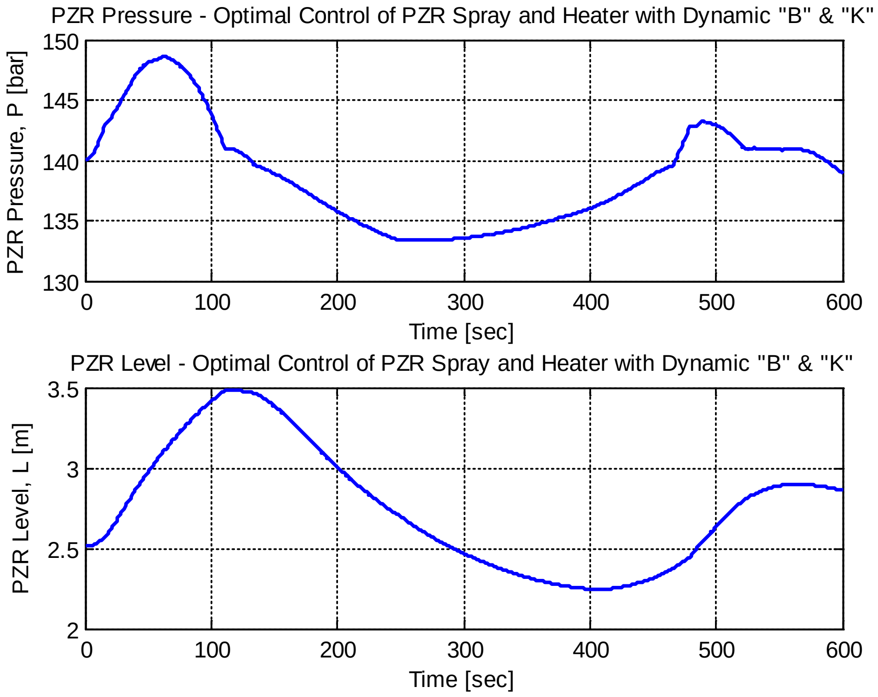


FIGURE 64: Dynamic PZR model P & L – LQR of PZR spray and heater with dynamic "B" & "K"

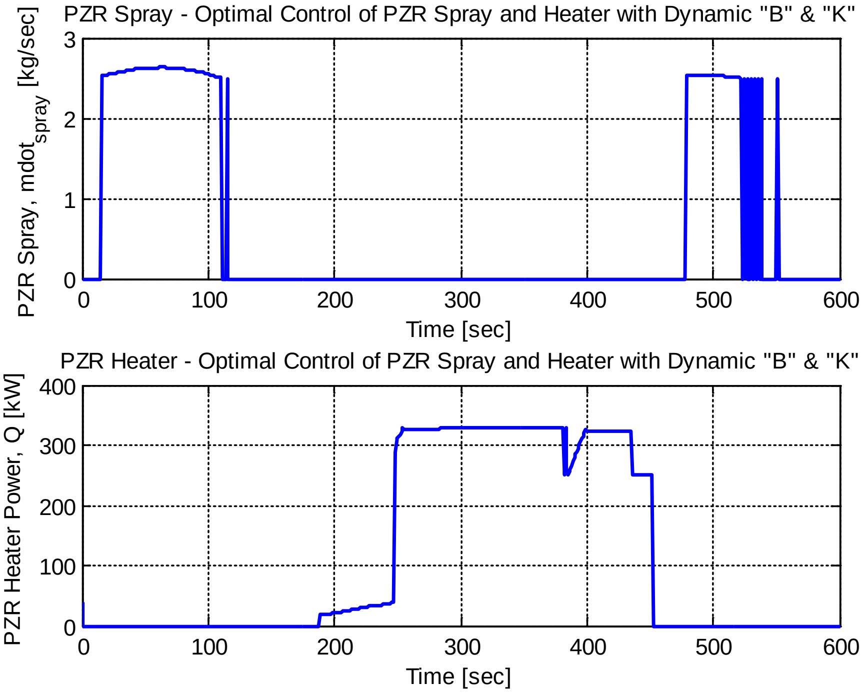


FIGURE 65: Dynamic PZR model spray and heater input – LQR of PZR spray and heater with dynamic "B" & "K"

5.3.4 Challenges in Implementing an Optimal Control for the Dynamic PZR Model

While the optimal control methods considered thus far for the dynamic PZR model (Sections 5.3.1, 5.3.2, and 5.3.3) show an improved performance compared to the conventional controller, there are still several challenges (and opportunities for improvement) with implementing this type of controller for the PZR dynamic conditions:

1) All the optimal control methods considered, as seen with EQUATION 117, assume a state space model of the system is available and that the state space model adequately captures the behavior.
2) The optimal controller with a constant gain “K” (Section 5.3.1) is not able to change as the dynamics of the physical system change. Instead, the optimal gain “K” is computed using a constant state space model of the PZR system. Unfortunately, the PZR is a non-linear dynamic system (as is the case with many physical systems) whose behavior is not adequately captured during transient conditions with a linear state space model. Consequently, the optimal gain “K” computed from the constant state space model is not always the best optimal solution for any given point, since the system is changing during transient (when control is needed most). Therefore, it is desirable to obtain a state space model of the system than can change adaptively as the dynamics of the system change.
3) The optimal controller with an new gain “K” computed via an adaptive state space “B” matrix (see Sections 5.3.2 and 5.3.3) exhibit some challenges in developing a suitable state space model for some time steps. Holding the “B” matrix constant and changing “A” was investigated. However, the computed

"A" matrix was unstable, which presents problems for implementing optimal control. Hence, "A" was held constant, with "B" updated, since it general produced a system that had an optimal solution. Furthermore, this method is limited in that it necessitates measurement of all the relevant state variables to capture dynamic behavior, but this is not always possible. Finally, updating the state space input matrix "B" at each time step (see EQUATION 121) does not capture all the dynamics of the physical system at all moments, since the state space dynamic matrix "A" is treated as a constant. Therefore, a system identification method that updates both the "A" and "B" matrix is desirable. Such an identification method is presented in Section 5.3.5.

5.3.5 Optimal Control of Spray for Dynamic PZR Model: New "K" at each time step via RLS Identification of the System to find New "A" and "B" at Each Iteration

In this section the modeling and simulation results of the dynamic PZR model with an optimal controller for PZR spray that updates the optimal gain matrix "K" at each time step by computing a new state space model of the system via Recursive Least Squares Identification (RLS ID) at each time step are considered. The system is modeled as shown in FIGURE 66. See TABLE 7 for a definition of the control terms and symbols. The RLS ID method can be implemented using the following algorithm (adapted from [36], [69], [96]), as seen in MATLAB function file "RLS_ID_PZR_Function_R0.m" provided in APPENDIX B.

1) Specify values for "α" (typically a large number), "γ" (the weighting factor for window), and "N" (the window length). A large "γ" (i.e., $\gamma = 1$) produces a large filter memory, which reduces noise, whereas a smaller "γ" can better

track changes that occur in the system. The window length “N” is generally greater than or equal twice “n” (N≥ 2n), where “n” is the model order of the system identification desired.

2) Select initial values for P(N) and $\hat{\theta}$(N). The initial value for P(N) can be defined as P_0(N) = α·I. It is recommended that the initial $\hat{\theta}$(N) be set to a non-zero number, since setting it to zero can produce computation error.

3) Acquire the physical system input data history (i.e., u(0), u(1),…, u(N)) and output data history (i.e., y(0), y(1),…, y(N)) and form the following structure:

$$\varphi^{\mathrm{T}}(N+1)=\left[y(N) \quad y(N-1) \quad \ldots \quad y(N+1-n) \quad u(N) \quad u(N-1) \quad \ldots \quad u(N+1-n)\right]$$

4) To start, allow sample “k” ← “N”

5) Compute the following:

$$\mathrm{L}(\mathrm{k}+1) \leftarrow \frac{\mathrm{P(k)}\varphi(\mathrm{k}+1)}{\gamma+\varphi^{\mathrm{T}}(\mathrm{k}+1)\mathrm{P(k)}\varphi(\mathrm{k}+1)}$$

6) Acquire the process input and output data for the next step y(k+1) and u(k+1)

7) Compute the identified process model for “k+1”:

$$\theta(\mathrm{k}+1) \leftarrow \theta(\mathrm{k})+\mathrm{L}(\mathrm{k}+1)\left[\mathrm{y}(\mathrm{k}+1)-\varphi^{\mathrm{T}}(\mathrm{k}+1)\theta(\mathrm{k})\right]$$

8) Compute the following to update for the next iteration:

$$\mathrm{P}(\mathrm{k}+1)=\frac{1}{\gamma}\left[\mathrm{I}-\mathrm{L}(\mathrm{k}+1)\cdot\varphi^{\mathrm{T}}(\mathrm{k}+1)\right]\mathrm{P}(\mathrm{k})$$

9) Construct a new $\varphi^{\mathrm{T}}(k+2)$ from the new process history, shown in Step 3.

10) Update “k” ← “k+1”

11) Return to Step 5 of the RLS identification algorithm process and repeat for the entire data series length.

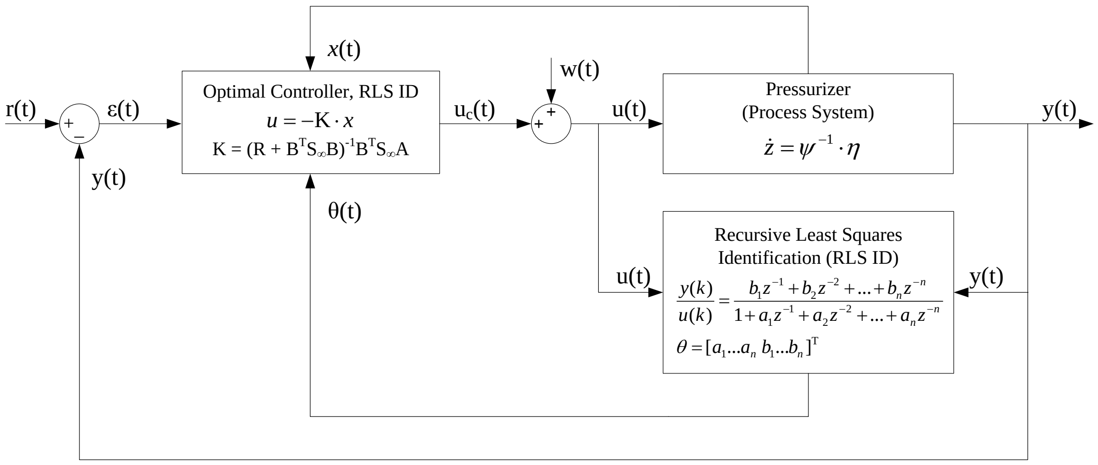

x(t)
w(t)
r(t)
ε(t)
Optimal Controller, RLS ID
$u = -\mathrm{K} \cdot x$
$\mathrm{K} = (\mathrm{R} + \mathrm{B}^{\mathrm{T}}\mathrm{S}_{\infty}\mathrm{B})^{-1}\mathrm{B}^{\mathrm{T}}\mathrm{S}_{\infty}\mathrm{A}$
$u_c(t)$
u(t)
Pressurizer
(Process System)
$\dot{z} = \psi^{-1} \cdot \eta$
y(t)
+
+
+
–
y(t)
θ(t)
Recursive Least Squares
Identification (RLS ID)
u(t)
$\frac{y(k)}{u(k)} = \frac{b_1 z^{-1} + b_2 z^{-2} + ... + b_n z^{-n}}{1 + a_1 z^{-1} + a_2 z^{-2} + ... + a_n z^{-n}}$
$\theta = [a_1 ... a_n \; b_1 ... b_n]^{\mathrm{T}}$
y(t)

FIGURE 66: Dynamic PZR model – LQR with new “K” via RLS ID every time step

An overview of the dynamic PZR Simulink model optimal controller with RLS ID of the system is provided in FIGURE 68. Details of the RLS ID and Optimal Controller are provided in FIGURE 69, FIGURE 70, and APPENDIX B.

Before implementing the RLS ID with the optimal controller, it was tested in the dynamic system without any control (open loop) in order to demonstrate the RLS ID that model could successfully capture the system behavior (i.e., pressure output). The RLS ID model was selected to identify a fourth order model, since there are four state variable used as inputs to the optimal controller (i.e., the state variables “x” used in the optimal controller are: x = [Pressure, Temperature, Level, Volume]$^{\mathrm{T}}$, see Section 5.3.2).

The largest error observed between the RLS ID output and the actual pressure output occurs at the inflection points during the transient simulation (e.g., approximately 100 seconds and 400 seconds). However, the error between “y” and “y_m” was typically much less, as observed from FIGURE 67. In general, the forth order RLS ID model is able to track the actual behavior of the system.

Therefore, since the test case verified the performance over a larger dynamic fluctuation, the fourth order RLS ID was assumed acceptable for identifying the system model for implementation with the optimal controller, since the dynamic pressure fluctuation range with the optimal controller should be much less than without control.

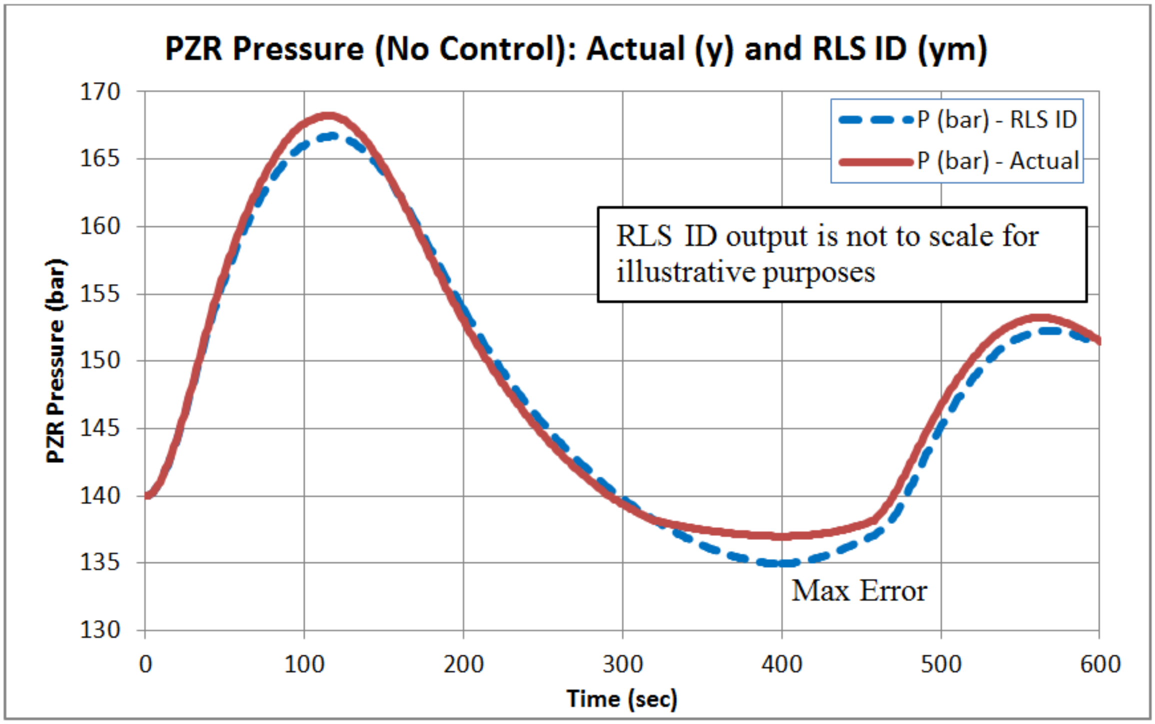


FIGURE 67: Dynamic PZR model – pressure actual (y) vs. RLS ID output (y_m)

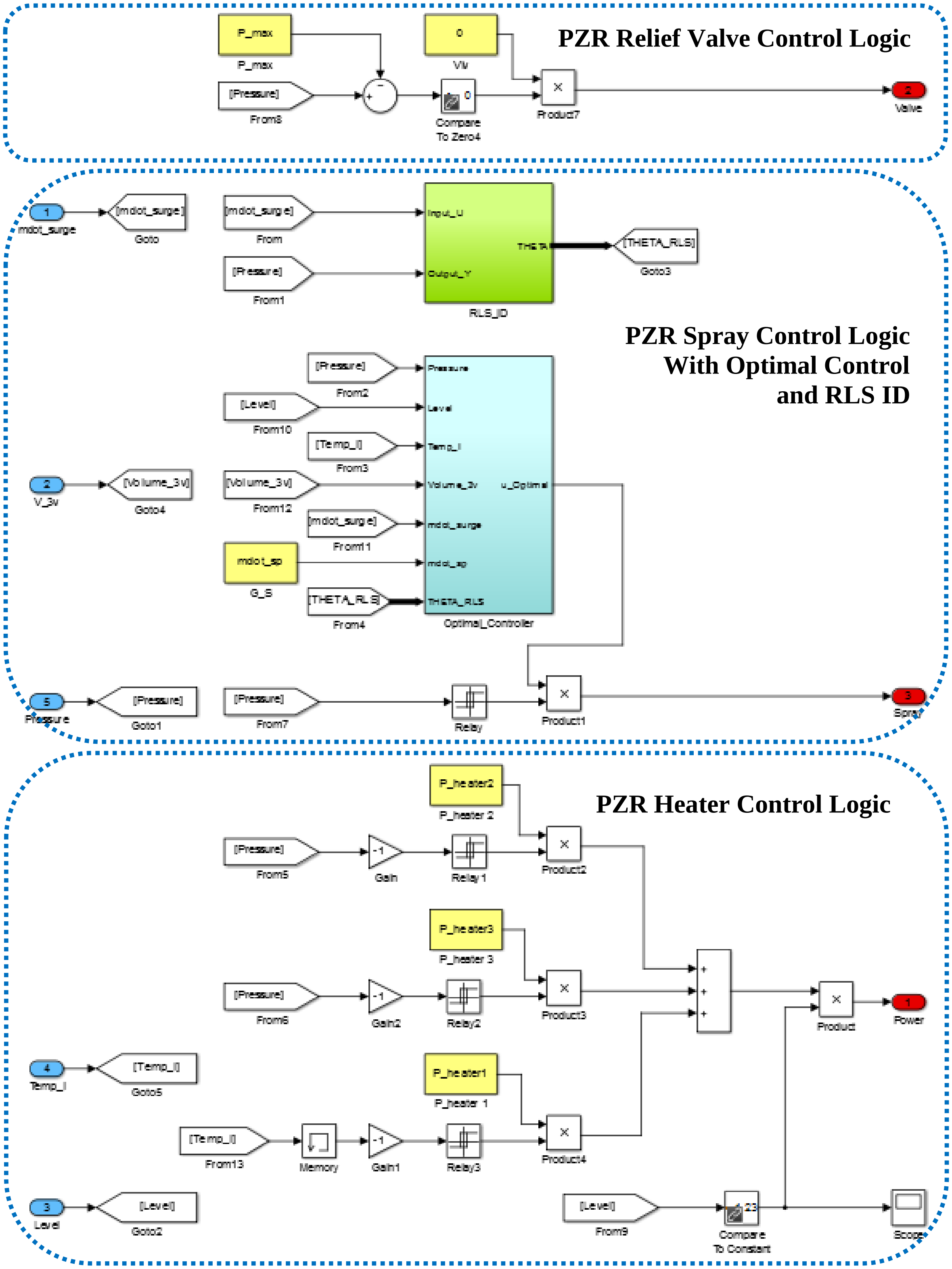


FIGURE 68: Dynamic PZR Simulink model of optimal controller with RLS ID

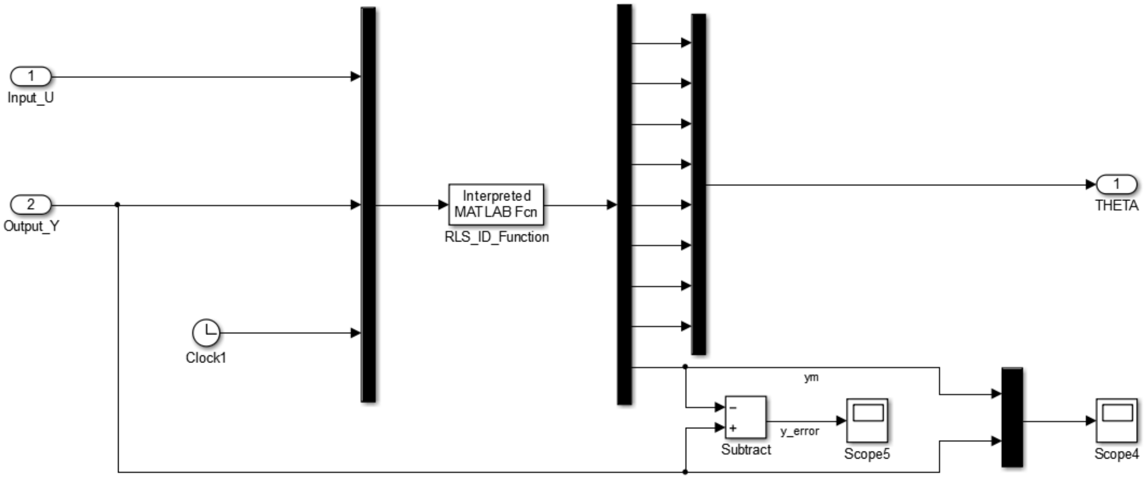


FIGURE 69: Simulink model of RLS ID

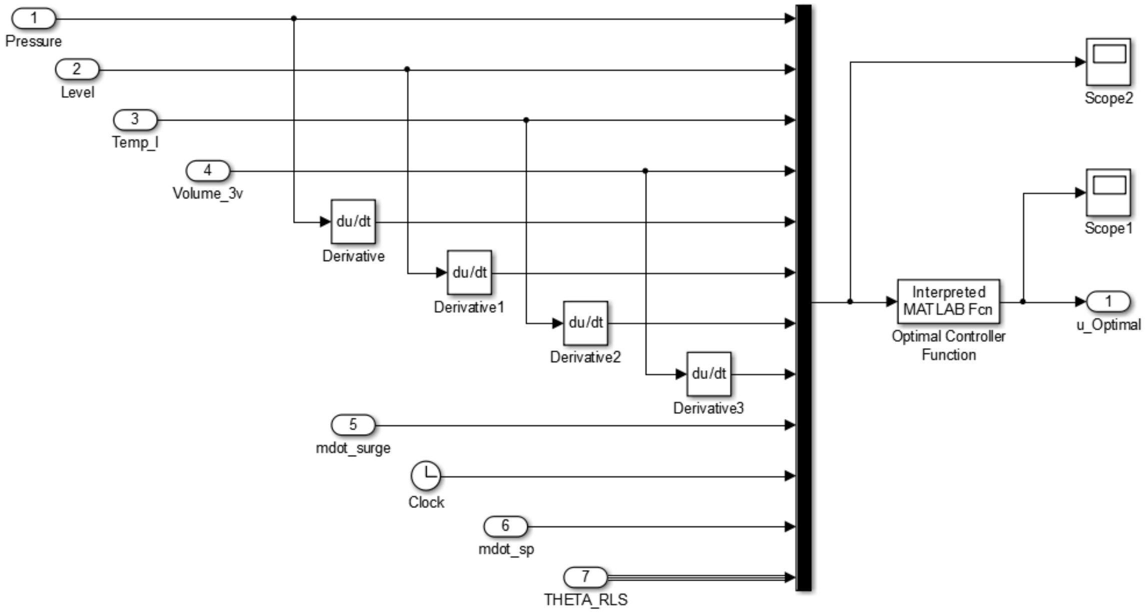


FIGURE 70: Simulink model of optimal controller with RLS ID

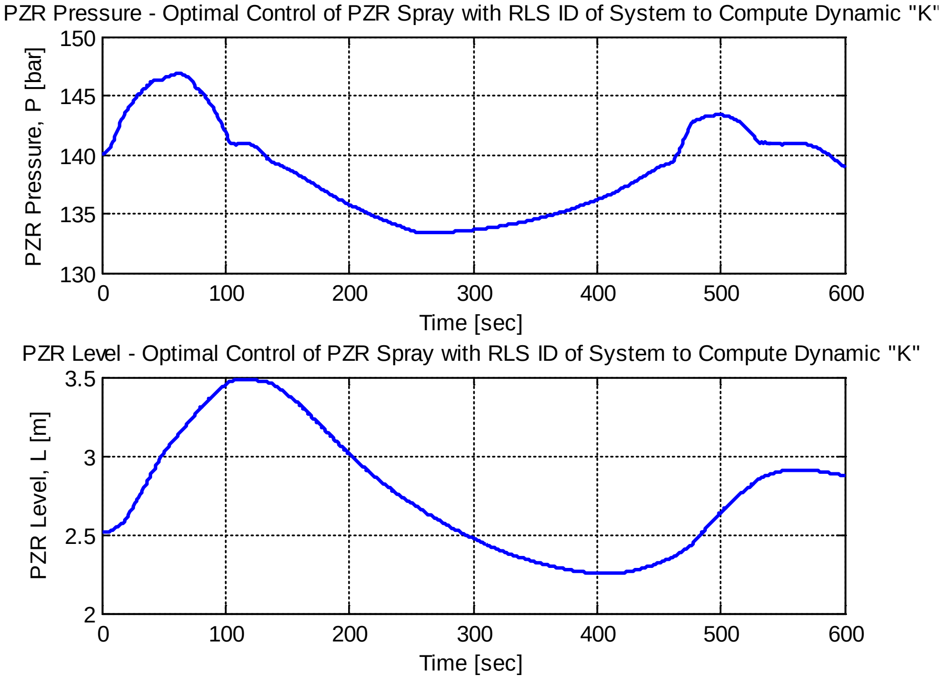


FIGURE 71: Dynamic PZR model P & L – LQR of PZR spray with RLS ID of system to compute Dynamic “K”

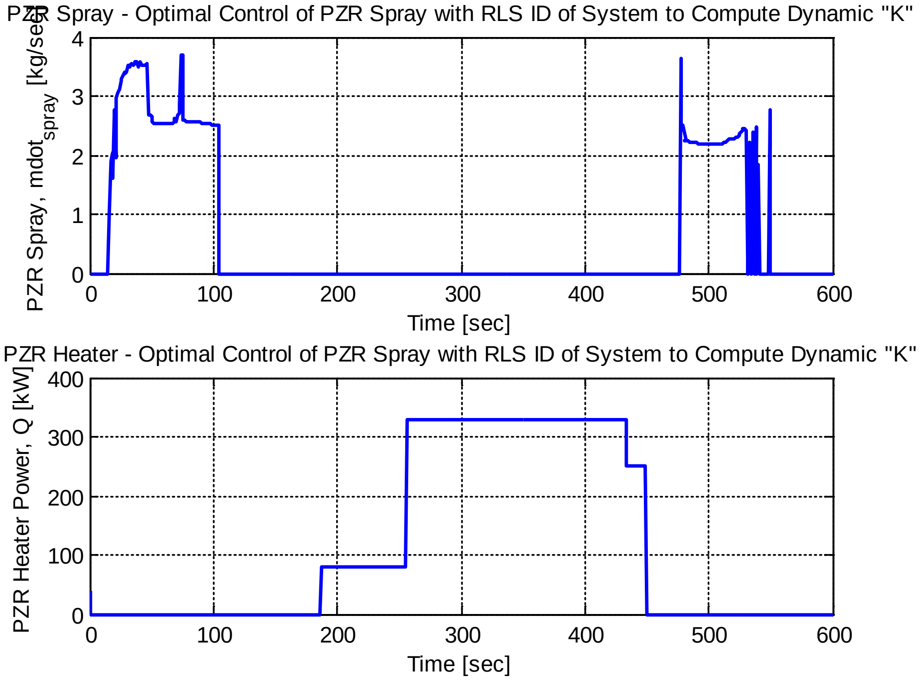


FIGURE 72: Dynamic PZR model spray and heater input – LQR of PZR spray with RLS ID of system to compute dynamic “K”

Results of simulating the dynamic PZR model using optimal control of PZR spray that computes a new “K” via a new state space model of the system via RLS ID at each time step are provided in FIGURE 71 (PZR pressure and level output) and FIGURE 72 (PZR spray and heater input). The $\dot{m}_{surge}$ (system disturbance “w”) for the simulation is provided in FIGURE 18. Discussion of the results and comparison with the other control methods are provided in Section 5.3.6.

5.3.6 Performance Review of Optimal Control Methods for the Dynamic PZR Model

Section 5.3.6 provides a performance review and conclusion of the optimal control methods considered in Section 5.3.

1) Optimal Control of Spray with Constant “K” (Section 5.3.1):

The optimal controller was successfully implemented for the dynamic PZR model. The first PZR pressure spike resulting from the first surge transient (see FIGURE 18) was reduced by approximately 2 bar in comparison with the conventional controller, as seen in FIGURE 73, FIGURE 74, and TABLE 8.

Of additional interest, the PZR pressure and level exhibit an inverse relationship when the PZR pressure controls are acting upon the system. PZR pressure and level both increase from an insurge transient or decrease from an outsurge transient. In general, the PZR pressure controls act to minimize the pressure spike or drop. However, the PZR pressure controls exhibit an opposite effect on PZR level by increasing the variation compared with no control. As such, PZR level is impacted by the PZR pressure control.

However, control of PZR level is handled differently (e.g., different controller, actuators, etc.) than PZR pressure control and therefore, is not considered in this document. FIGURE 75 shows that the change in level, while initially large in comparison to the open loop level change, is much less in magnitude compared with the reduction in pressure achieved by implementing more advanced pressure control. Additionally, provided the level stays within the acceptable range, it is more desirable from a fatigue standpoint to have the level fluctuate in the PZR rather than the pressure fluctuate.

If level fluctuation is too severe, further inventory control (i.e., which controls PZR level) can be accomplished by controlling (1) the charging rate of the cooling system from the chemical volume and control system (via a charging control valve) and (2) the cooling system letdown (via a letdown control valve control). Therefore, the fluctuation in level observed in FIGURE 75 is not a concern. Furthermore, the change in level between the different PZR pressure control methods is very small in comparison to the amount of change they can impart on minimizing the pressure fluctuation.

2) Optimal Control of Spray with New “K” via New SS “B” (Section 5.3.2):

Section 5.3.2 presents a simple adaptive method of computing a new optimal gain matrix “K” at each time step by updating the state space input matrix “B” at each time step via measured state variables. This control method was applied to the PZR spray controller. While it does not capture all the system dynamics, this method does show an improved control performance over the conventional controller and the optimal controller with a constant gain “K” (which was presented Section 5.3.1), as seen in FIGURE 73. Limitation or challenges associated with this method of updating the “B” matrix are discussed in Section 5.3.4.

3) Optimal Control of Spray and Heaters with New “K” via New “B” (Section 5.3.3):

Section 5.3.3 builds on the method presented in Section 5.3.2 by applying it to the PZR heater control in addition to the PZR spray control. As shown by comparing the

PZR pressure in FIGURE 62 and FIGURE 64, the addition of optimal control for the PZR heater resulted in minimal impact of heater performance (compared to conventional heater control) to mitigate a pressure decrease (see FIGURE 73). It is possible that the minimal impact of PZR heater optimal control on regulating PZR pressure is the result of (1) poor scaling for adequate control (e.g., computed optimal gain "K" is not adequately scaled) and (2) physical limitations, such as (2a) the amount of heat that can be added to the system (e.g., control saturation) and (2b) response of the system to the added heat (e.g., delay between control input and response of the system).

4) Optimal Control of Spray with New "K" via RLS ID model (Section 5.3.5):

Section 5.3.5 introduces an adaptive method of computing a new optimal gain matrix "K" at each time step by updating the state space model (both "A" and "B" state space matrix) at each time step via recursive least squares identification (RLS ID). The RLS ID looks at the PZR input and output to develop an approximation of the physical system at that specific point based on input and output data history. As shown in FIGURE 67, the fourth order RLS ID performed reasonably to estimate the system output (i.e., $y \approx y_m$ for most time steps). The maximum error was six percent between the RLS ID estimate output and the actual output. However, there is room for system identification improvement. It is desirable that the RLS ID estimated output (y_m) equal the actual output (y) within in a reasonable tolerance (e.g., ±3%). Therefore, an RLS ID corrector might be able to help improve the system identification accuracy, as discussed in CHAPTER 6. In comparison, this method of adaptive optimal control via RLS ID performed better at controlling the pressure than any of the other control methods considered. It was able to reduce the pressure spike resulting from the first surge transient the more that the other control method (see FIGURE 73).

5) Comparison of Optimal Control Methods:

A (PZR pressure and level) performance comparison of the different optimal controllers considered in Section 5.3 is presented below.

- FIGURE 73 provides a PZR pressure plot for all the control methods, including no control (open loop).
- FIGURE 74 provides a PZR pressure plot for all the control methods, excluding no control (open loop), so the deviation between the control methods can be more easily observed.
- FIGURE 75 provides a PZR level plot for all the control methods, including no control (open loop).
- FIGURE 76 provides a graphical depiction of the tabulated information in TABLE 8.
- TABLE 8 provides tabulation for each control method of the time required to reach the peak pressure at the first pressure spike and the pressure value at that peak.

As expected, the more a controller can adapt to the current dynamics of the system, the more it is successful in controlling the pressure. The smallest pressure spike (for the controllers presented in these figures) was achieved by the optimal controller for PZR spray using a fourth order RLS ID that was used to compute a new optimal gain “K” at each time step (see 5.3.5), as seen from FIGURE 75. However, as observed in FIGURE

67, there is still room for improvement in the system identification (RLS ID), which should equate to an improvement in controller performance.

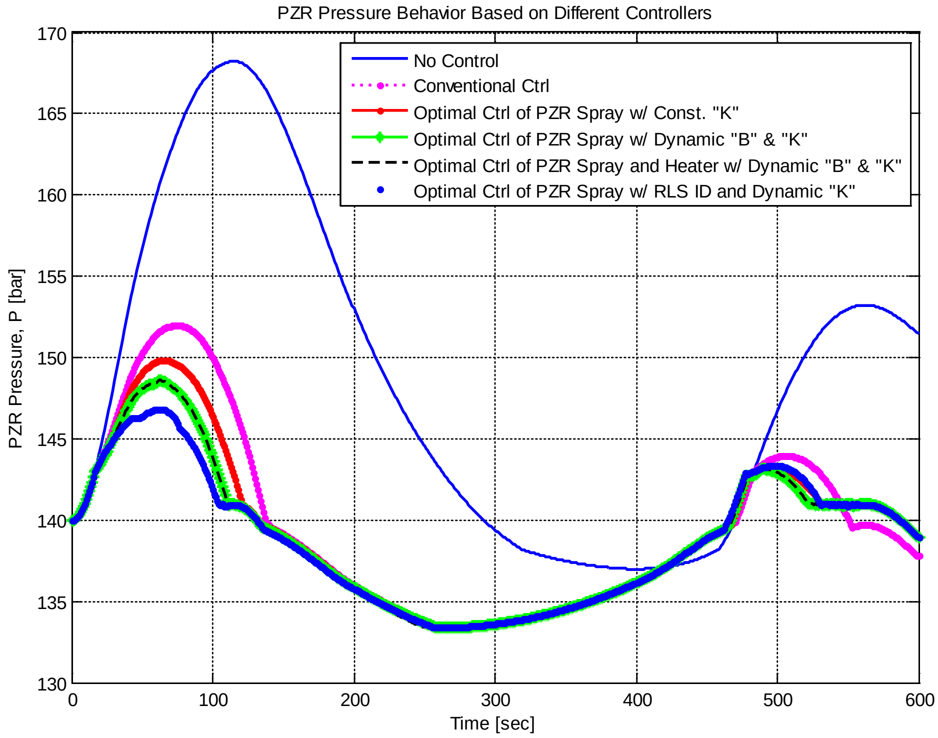


FIGURE 73: Dynamic PZR model pressure – optimal control comparison 1

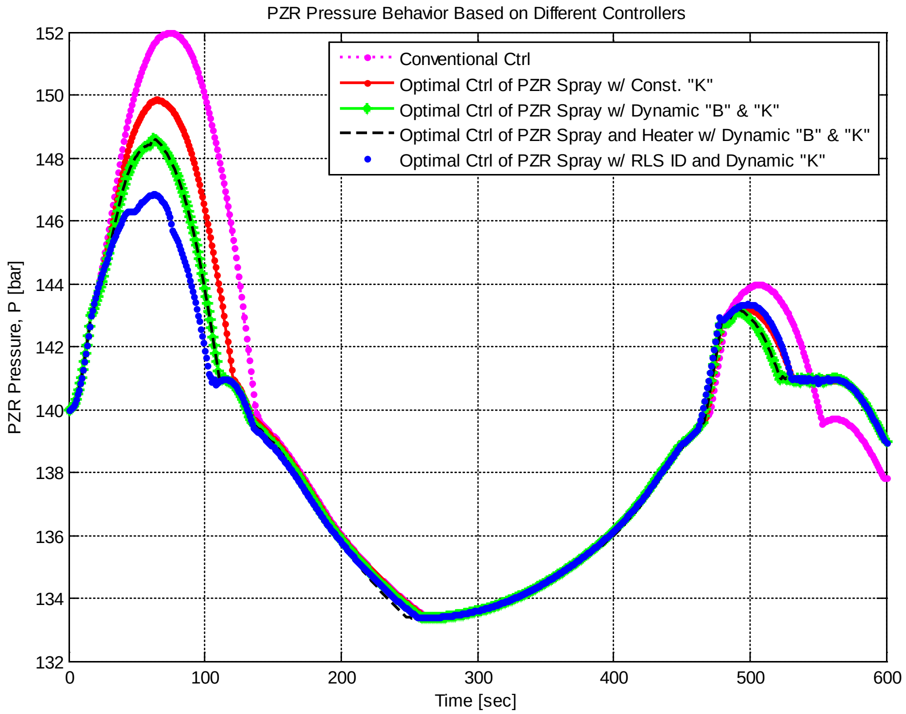


FIGURE 74: Dynamic PZR model pressure – optimal control comparison 2

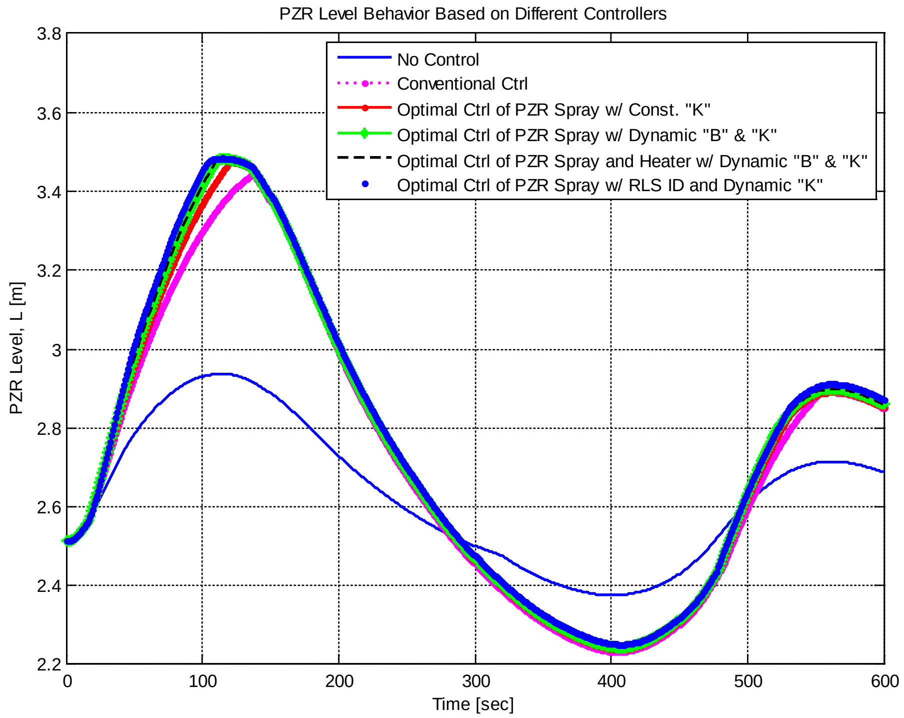


FIGURE 75: Dynamic PZR model level – optimal control comparison

TABLE 8: PZR optimal controller performance results

Control Method	Time to Peak (first surge), sec	Pressure at Peak (first surge), bar
No Control	114	168.2
Conventional	75	152
Optimal (Constant K)	65	149.9
Optimal (Dynamic K) Hybrid	62	148.6
Optimal (RLS ID w/ Dynamic K)	60	146.8

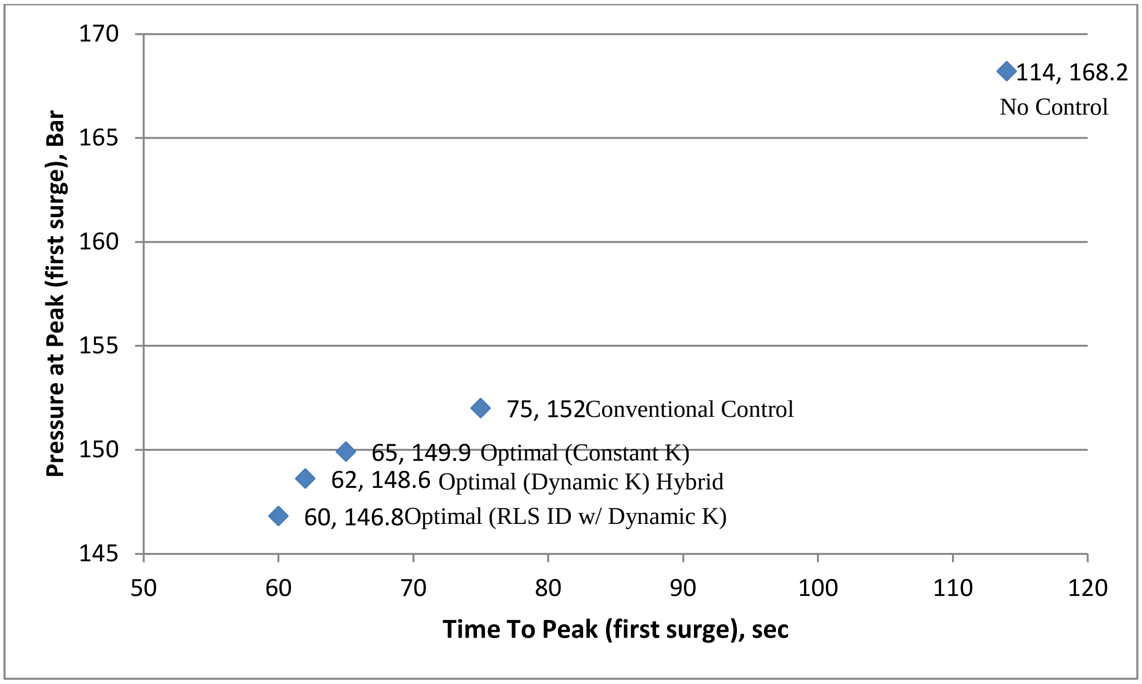


FIGURE 76: PZR optimal controller performance results

5.4 Adaptive Control (STR PID) for Dynamic PZR Model

Section 5.3 considered a few optimal controller versions that can be considered adaptive (e.g., optimal controller that computes new optimal gain matrix at each time step from an identified model of the system, as discussed in Section 5.3.5). In this section and subsections the modeling and simulation results for the dynamic PZR model with another version of an adaptive controller (self-tuning PID with RLS ID) is considered for comparison.

5.4.1 Adaptive Control (Self-Tuning PID Controller with RLD ID) Modeling

This section presents the model for a self-tuning regulator (STR) Proportional-Integral-Derivative (PID) controller for PZR spray with RLS ID and shows the implementation in Simulink. Section 5.3.5 details modeling of the RLS ID. However, for the STR PID, the RLS ID is constrained to identification of a second order system [69]. The system is modeled as shown in FIGURE 77. See TABLE 7 for a definition of the control terms and symbols.

The PID controller is a feedback type controller that is located in the forward loop before the system, as shown in FIGURE 77.

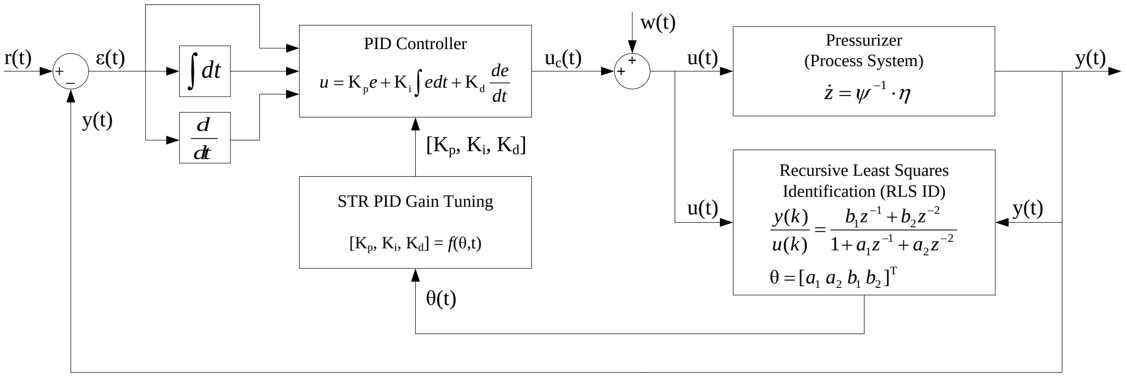

r(t)
+
−
ε(t)
$\int dt$
$\frac{d}{dt}$
y(t)
PID Controller
$u = K_p e + K_i \int edt + K_d \frac{de}{dt}$
$u_c(t)$
w(t)
+
+
u(t)
Pressurizer
(Process System)
$\dot{z} = \psi^{-1} \cdot \eta$
y(t)
$[K_p, K_i, K_d]$
STR PID Gain Tuning
$[K_p, K_i, K_d] = f(\theta,t)$
u(t)
Recursive Least Squares
Identification (RLS ID)
$\frac{y(k)}{u(k)} = \frac{b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}}$
$\theta = [a_1\ a_2\ b_1\ b_2]^T$
y(t)
θ(t)

FIGURE 77: Dynamic PZR model – STR PID control with new [K_p, K_i, K_d] tuned gains computed via RLS ID at each time step

This type of controller "chases" the error of the system. The general control equation for a PID controller is provided in EQUATION 122 [82], where the objective of the controller is to drive the error between the system output (y) and the reference or commanded value (r) to zero. This is achieved by some or all of the elements in

EQUATION 122 (i.e., P, PI, and PID controllers) by selecting appropriate gain values (i.e., K_p, K_i, and K_d), where K_p is the proportional control gain, K_i is the integral control gain, and K_d is the derivative control gain.

$$u(t) = \mathrm{K_p}\varepsilon(t) + \mathrm{K_i}\int \varepsilon(t)dt + \mathrm{K_d}\frac{d}{dt}\varepsilon(t)$$

EQUATION 122

For a second order system (see EQUATION 123) the closed loop response with a PID controller is impacted by each gain factor uniquely [36], [42], [83], [84], [86]:

- K_p generally acts to decrease rise time and steady state error, with an increase in overshoot and no change to settling time.
- K_i generally acts to eliminate steady state error and can decrease rise time, with an increase in overshoot and settling time (i.e., less damping).
- K_d generally acts to decrease overshoot and settling time, with no change to rise time or steady state error.

$$G(s) = \frac{Y(s)}{U(s)} = \frac{K_0}{s^2 + 2\zeta\omega_n + \omega_n^2}$$

EQUATION 123

There are several classical methods (e.g., Ziegler-Nichols) for selecting the gain values (as discussed in [83] and [84]), which statically tune the gains (i.e., once the gain value is selected, it does not change). However, to make the controller adaptive, an online tuning method is considered in this section. Detail of the STR PID controller MATLAB function file used in the Simulink model are provided in APPENDIX B. The STR PID modeling is described below (EQUATION 124 through EQUATION 133), developed per [68] and [69].

The STR PID method assumes the system can be reasonably modeled as a second order system [69], as seen from the integral and derivative in EQUATION 122. To develop the STR PID model, consider a process system in the following open loop form (EQUATION 124):

$$\frac{y}{u} = \frac{b_1 q^{-1} + b_2 q^{-2}}{1 + a_1 q^{-1} + a_2 q^{-2}} = \frac{\mathrm{A}}{\mathrm{B}}$$

EQUATION 124

The characteristic equation for this second order system is described as shown in EQUATION 125:

$$c.c. = 1 + a_1 q^{-1} + a_2 q^{-2} = 0$$

EQUATION 125

The characteristic equation can be stated as follows (EQUATION 126) using pole shifting (see CHAPTER 4) by a factor of α (alpha), where $0 \le \alpha \le 1$:

$$c.c. = \left(1 + \alpha q^{-1}\right)\left(1 + a_1 \alpha q^{-1} + a_2 \alpha^2 q^{-2}\right) = 0$$

EQUATION 126

If the control structure is stated as follows (see EQUATION 127 and EQUATION 128), a relationship with EQUATION 124 can be developed.

$$u(k) = \frac{T\left(q^{-1}\right)}{R\left(q^{-1}\right)} y_r(k) - \frac{S\left(q^{-1}\right)}{R\left(q^{-1}\right)} y(k)$$

EQUATION 127

Where, $R(q^{-1})$ and $S(q^{-1})$ are more fully expressed as shown EQUATION 128:

$$R\left(q^{-1}\right)=\left(1-q^{-1}\right)\left(1-r_1q^{-1}\right)$$
$$S\left(q^{-1}\right)=s_0+s_1q^{-1}+s_2q^{-2}$$
$$s_0=T_sK_i+\frac{K_d}{T_s}+K_p$$
$$s_1=\frac{-2K_d}{T_s}-K_p+r_1K_p$$
$$s_2=\frac{K_d}{T_s}-r_1K_p$$

EQUATION 128

Additionally, for T(q^{-1}):

$$T\left(q^{-1}\right)=-\left(s_0+s_1+s_2\right)=-T_sK_i$$

EQUATION 129

Therefore, combining these equations, the closed loop system can be expressed as shown in EQUATION 130 and EQUATION 131.

$$Ay=Bu=B\left[\frac{T}{R}y_r-\frac{S}{R}y\right]\Rightarrow$$
$$\frac{y}{y_r}=\frac{BT}{AR+BS}$$

EQUATION 130

Where,

$$AR+BS=\left(1+\alpha q^{-1}\right)\left(1+a_1\alpha q^{-1}+a_2\alpha^2q^{-2}\right)$$

EQUATION 131

Expanding EQUATION 130 and comparing coefficients yields the following four linear relationships (EQUATION 132), which are expressed in matrix form for ease of solution (i.e., four equations with four unknowns: r_1, s_0, s_1, s_2).

$$\begin{bmatrix} X \\ Y \\ Z \\ 0 \end{bmatrix} = \begin{bmatrix} 1 & b_1 & 0 & 0 \\ a_1 - 1 & b_2 & b_1 & 0 \\ a_2 - a_1 & 0 & b_2 & b_1 \\ -a_2 & 0 & 0 & b_2 \end{bmatrix} \begin{bmatrix} r_1 \\ s_0 \\ s_1 \\ s_2 \end{bmatrix} = \begin{bmatrix} 1 - a_1 + \alpha + a_1\alpha \\ a_1 - a_2 + a_1\alpha^2 + a_2\alpha^2 \\ a_2 + a_2\alpha^3 \\ 0 \end{bmatrix}$$

EQUATION 132

Once the set of linear algebraic equations in EQUATION 132 have been solved for the unknowns (i.e., r_1, s_0, s_1, and s_2), the PID gains can be computed as follows (see EQUATION 133) per EQUATION 127, EQUATION 128, and EQUATION 129.

$$K_p = \frac{s_1 + 2s_2}{1 + r_1}$$
$$K_i = \frac{(s_0 + s_1 + s_2)}{T_s}$$
$$K_d = T_s \left[\frac{-r_1 s_1 + (1 - r_1) s_2}{1 + r_1} \right]$$

EQUATION 133

These gains are then inserted into the PID controller (see EQUATION 122) as new values at each time step to adaptively tune the PID gains based on the behavior of the identified second order system (via RLS ID).

An overview of the dynamic PZR Simulink model STR PID controller with RLS ID of the system is provided in FIGURE 78. Details of the RLS ID and Optimal Controller are provided in FIGURE 79, FIGURE 80, and APPENDIX B.

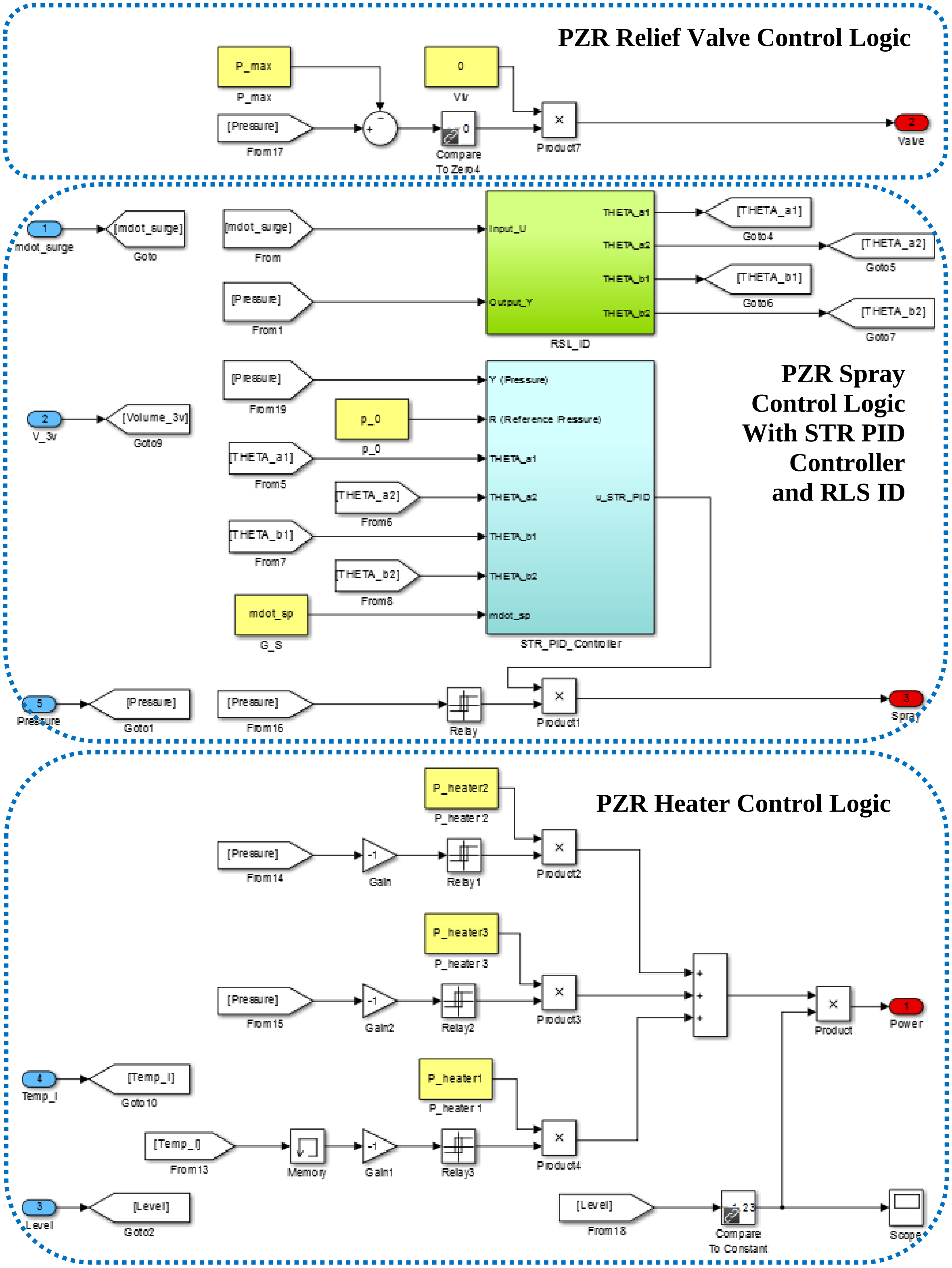


FIGURE 78: Dynamic PZR Simulink model of STR PID controller with RLS ID

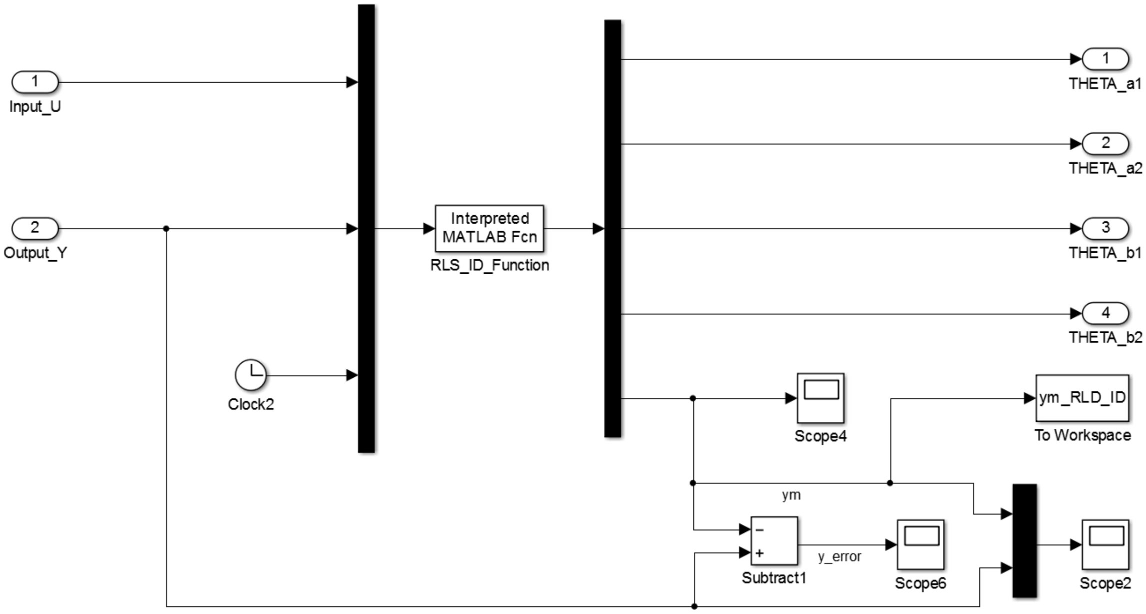


FIGURE 79: Simulink model of RLS ID (second order system identification)

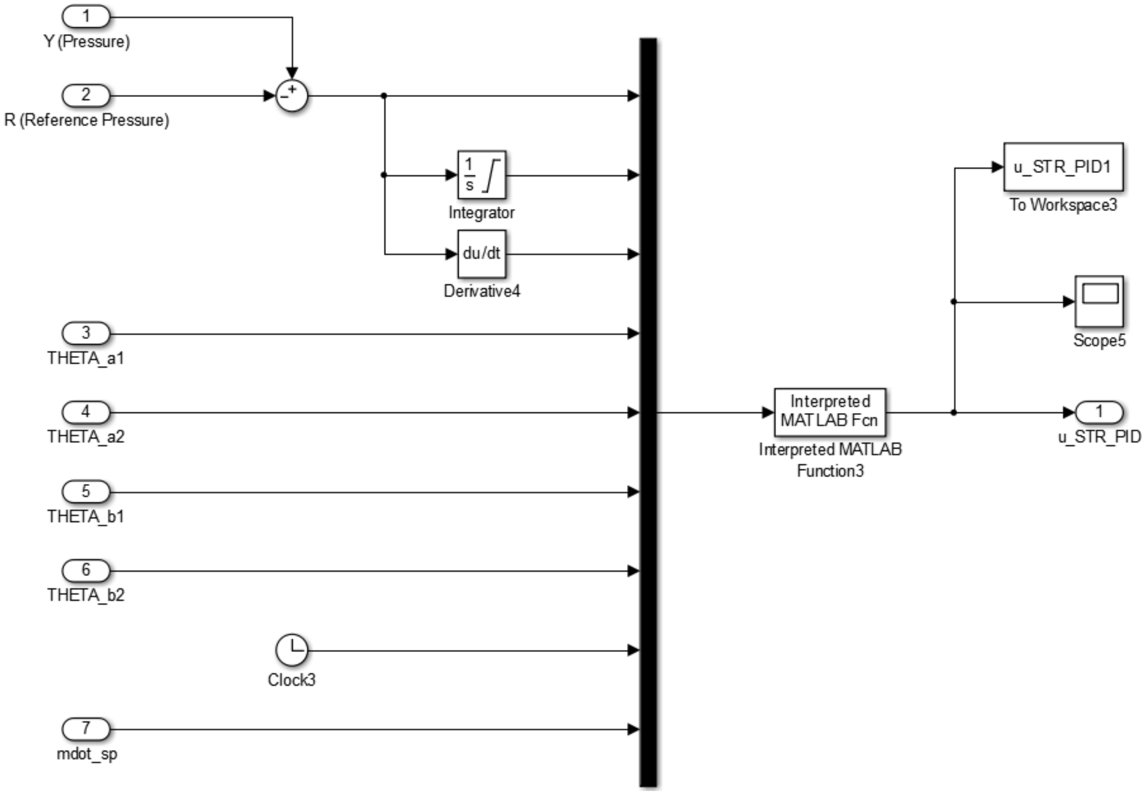


FIGURE 80: Simulink model of STR PID controller

5.4.2 Adaptive Control (Self-Tuning PID Controller with RLD ID) Results

Results of simulating the dynamic PZR model using STR PID control of PZR spray that computes new [K_p, K_i, K_d] for the PID controller via a new state space model of the system from RLS ID at each time step are provided in FIGURE 82 (PZR pressure and level output) and FIGURE 83 (PZR spray and heater input). The $\dot{m}_{surge}$ (system disturbance "w") for the simulation is provided in FIGURE 18. Discussion of the results and comparison with the other control methods is provided in Section 5.4.3.

The RLS ID identified a second order model of the true system at each time step for input to the STR PID controller (i.e., to compute new gains based on the current system model). Consequently, if the RLS ID is unable to produce a model of the system with sufficient accuracy, the STR PID will not perform properly. Therefore, it is beneficial to check the model output against the actual (measured) output.

FIGURE 81 provides a plot for comparison of the system pressure output obtained from the RLS ID model (y_m) and the actual (measured) pressure output of the system (y). RLS ID model output error (as compared to the actual output of the system) reduces the effectiveness of the STR PID controller. The STR PID model identifier output error can be attributed to the RLS ID model being second order for the STR PID (by necessity). For this situation, the true system (simulated with the dynamic PZR model) needs a higher order identification to capture all the dynamic behavior occurring (e.g., in comparison, a fourth order identified model exhibited improved performance for the optimal controller, as seen in FIGURE 67).

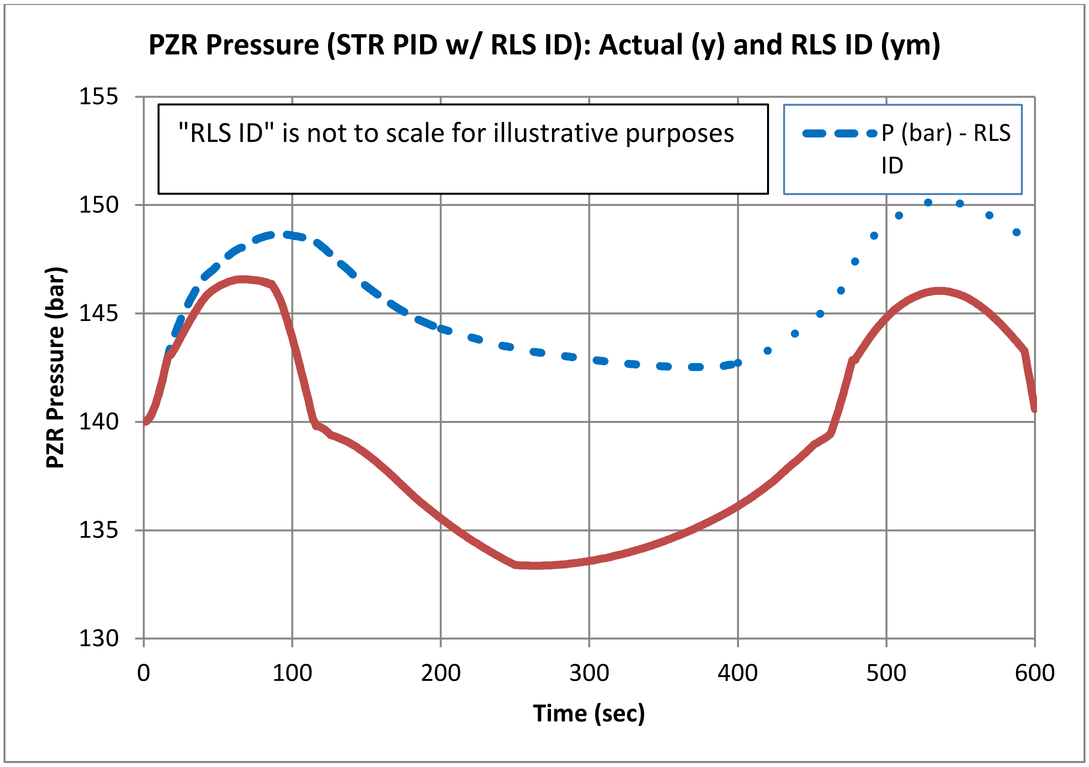


FIGURE 81: Dynamic PZR model – STR PID with RLS ID – pressure “y” vs. “y_m”

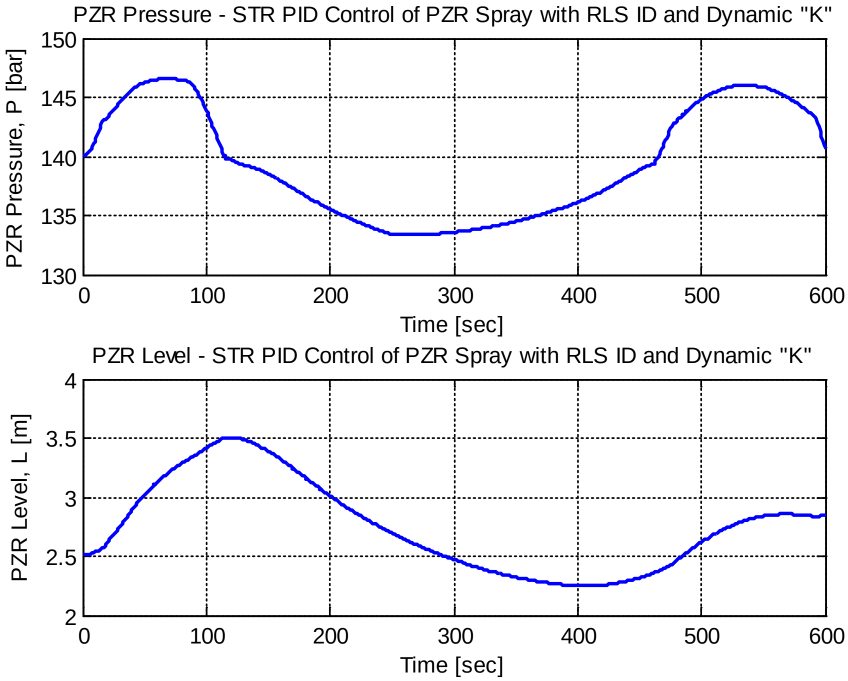


FIGURE 82: Dynamic PZR model pressure and level – STR PID of PZR spray with RLS ID of system to compute dynamic “K”

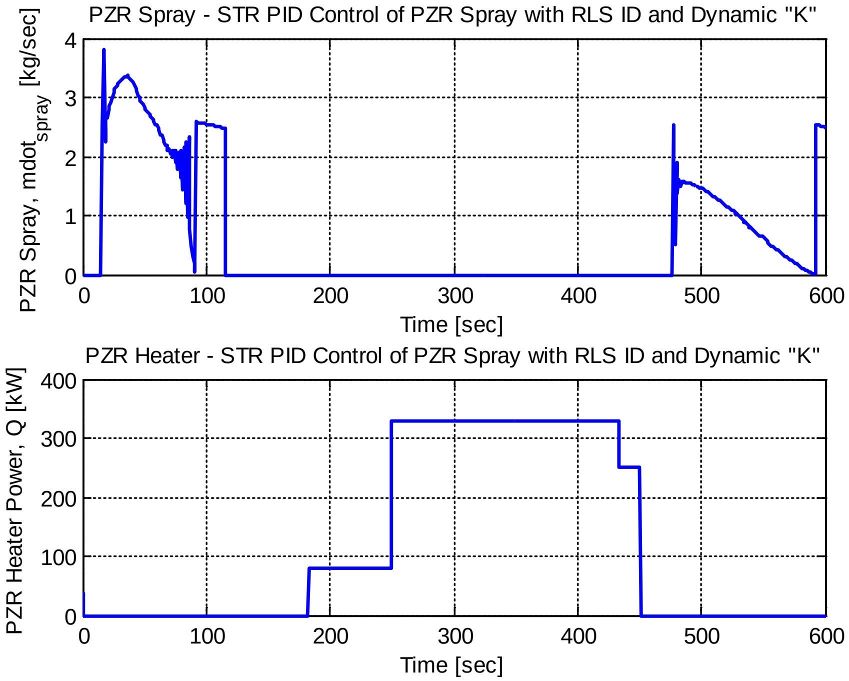


FIGURE 83: Dynamic PZR model spray and heater input – STR PID of PZR spray with RLS ID of system to compute dynamic “K”

5.4.3 Adaptive Control (Self-Tuning PID Controller with RLD ID) Comparison

A (PZR pressure and level) performance comparison for the PZR with (1) no control, (2) conventional control, (3) adaptive optimal control with RLS ID, and (4) STR PID with RLS ID is presented below.

- FIGURE 84 provides a PZR pressure plot for all the control methods, including no control (open loop).
- FIGURE 85 provides a PZR pressure plot for all the control methods, excluding no control (open loop), so the deviation between the control methods can be more easily observed.
- FIGURE 86 provides a PZR level plot for all the control methods, including no control (open loop).

In summary, the STR PID controller performed better than the conventional controller and almost as well as the adaptive optimal controller with RLS ID, which is noteworthy. The STR PID controller is limited in the fact that it requires a second order model of the system capture the relevant dynamics. However, the PZR dynamic model can exhibit higher order dynamics during a transient (e.g., the dynamic PZR model in EQUATION 20 is 8th order). Therefore, the second order RLS ID model for STR PID can deviate from the actual system, as seen in FIGURE 81, which reduces the effectiveness of the STR PID controller.

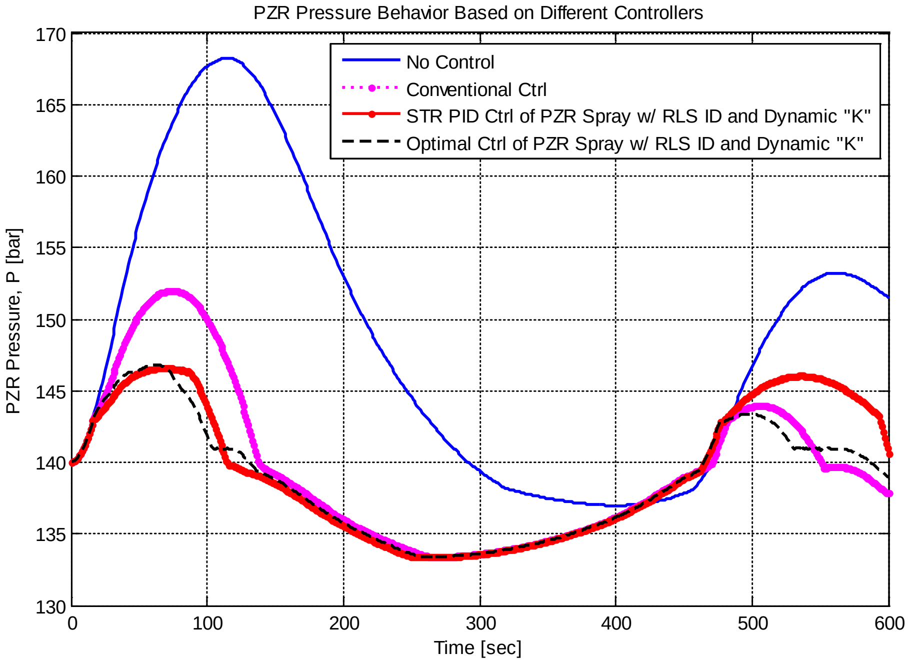


FIGURE 84: Dynamic PZR model pressure – STR PID and optimal control, 1

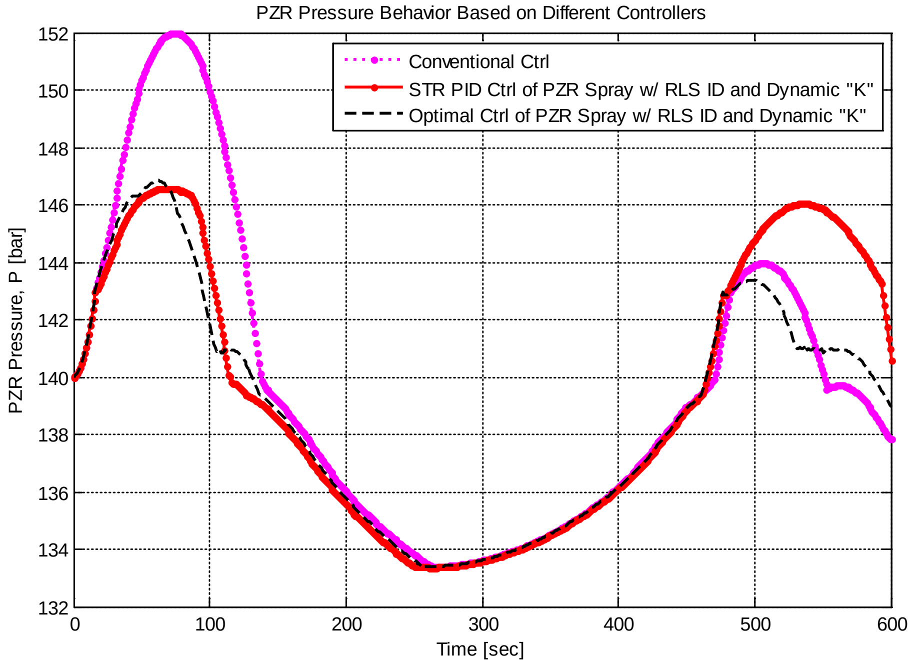


FIGURE 85: Dynamic PZR model pressure – STR PID and optimal control, 2

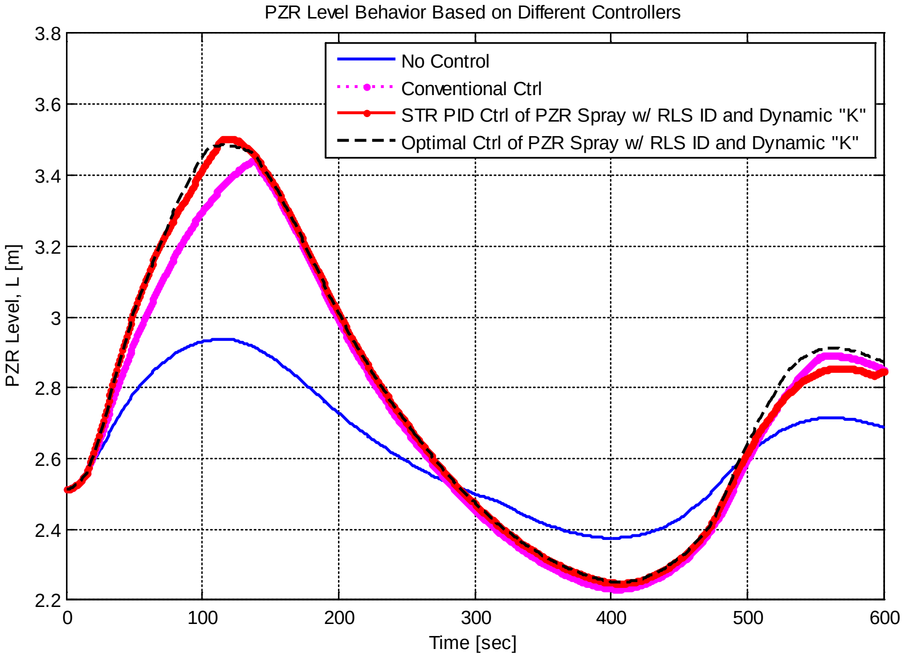


FIGURE 86: Dynamic PZR model level – STR PID and optimal control

5.5 Adaptive Control (MV Control) for Dynamic PZR Model

Section 5.3 considered a few optimal controller versions that can be considered adaptive (e.g., optimal controller that computes new optimal gain matrix at each time step from an identified model of the system, as discussed in Section 5.3.5). In this section and subsections the modeling and simulation results for the dynamic PZR model with another version of an adaptive controller (Minimum Variance Control with RLS ID) is considered for comparison. The information and results in this section is adapted from M. Smith and S. Kamalasadan [95].

5.5.1 Adaptive Control (Minimum Variance Controller) Modeling

An overview of the minimum variance (MV) control algorithm is described below for a generic feedback control application. MV theory and implementation algorithms are well documented in publications (as discussed in [93] and [94]). An ARMAX model of the system is used with the MV function. The ARMAX model of a system is shown in EQUATION 80, where ε(k) is the unmeasured disturbance treated as zero-mean white noise with variance σ_ϵ^2, "A(q)" and "C(q)" are monic polynomials with orders "n" and "p", "B(q)" is a polynomial with order "m", "τ" is the integer number of sample periods (τ ≥ 1), and "q" is the shift operator (as discussed in [93] and [94]).

$$A(q) \cdot y(k) = q^{-\tau} \cdot B(q) \cdot u(k) + C(q) \cdot \varepsilon(k)$$

EQUATION 134

$$A(q) = 1 + a_1 q^{-1} + a_2 q^{-2} + ... + a_n q^{-n}$$

EQUATION 135

$$B(q) = b_1 q^{-1} + b_2 q^{-2} + ... + b_m q^{-m}$$

EQUATION 136

$$C(q) = 1 + c_1 q^{-1} + c_2 q^{-2} + ... + c_p q^{-p}$$

EQUATION 137

Therefore, the discrete transfer function of the system input and output “G(q)” is the ratio of the “B(q)” and “A(q)” polynomials, as shown in EQUATION 138. The numerator and denominator are a series of polynomials, where the denominator exhibits a monic polynomial structure. The model parameters (i.e., polynomial coefficients) can be captured in a vector “θ”, as shown in EQUATION 139. The physical system input data history (i.e., u(0), u(1),…, u(N)) and output data history (i.e., y(0), y(1),…, y(N)) can be capture in the following structure shown in EQUATION 140. For a linear time-invariant (LTI) system, “q” is equivalent to “z”, as defined with the z-transform. Therefore, The “G(q)” polynomial coefficients in “θ” are equal those in “G(z)”. As such, estimate of the model parameters can be obtained using techniques such as RLS ID (see Section 5.3.5 for additional information regarding RLS ID).

$$G(q) = \frac{B(q)}{A(q)} = \frac{b_1 z^{-1} + b_2 z^{-2} + ... + b_n z^{-m}}{1 + a_1 z^{-1} + a_2 z^{-2} + ... + a_m z^{-n}}$$

EQUATION 138

$$\theta = [a_1 ... a_n \ b_1 ... b_n]^T$$

EQUATION 139

$$\varphi(N+1) = [y(N) \quad y(N-1) \quad ... \quad y(N+1-n) \quad u(N) \quad u(N-1) \quad ... \quad u(N+1-n)]^T$$

EQUATION 140

The MV controller focuses on minimizing a cost function “J_{MV}” ([94]), as shown in EQUATION 141, where “r” is the reference value (e.g., 140 bar). Therefore, MV control targets minimizing the variance of the error between the reference value (set point) and the actual output value.

$$J = f\left(y(k)\right) = E\left\{\left[r - y(k+\tau)\right]^2\right\}$$

EQUATION 141

To achieve the cost function minimization goal, an expression is needed to relate the input “u” with the output “y” for MV control implementation. EQUATION 141 enables prediction of the output “τ” steps ahead, which is obtained by manipulating EQUATION 134 via multiplication with “E_τ” and substitution using the Diophantine equation.

$$y(k+\tau) = \frac{F_\tau(q)}{C(q)} y(k) + \frac{E_\tau(q) \cdot B(q)}{C(q)} u(k) + E_\tau(q) \cdot \varepsilon(k+\tau)$$

EQUATION 142

$$E_\tau(q) = e_0 + e_1 q^{-1} + e_2 q^{-2} + \ldots + e_{\tau\text{-}1} q^{-(\tau\text{-}1)}$$

EQUATION 143

$$F_\tau(q) = f_0 + f_1 q^{-1} + f_2 q^{-2} + \ldots + f_{n\text{-}1} q^{-(n\text{-}1)}$$

EQUATION 144

As such, minimization is attained when the first two components in EQUATION 142 are equal to each other (i.e., this can also be expressed as their sum equaling zero). The MV control law is shown in EQUATION 146, which is obtained by manipulating EQUATION 145 for “u_{MV}”.

$$\frac{F_\tau(q)}{C(q)} y(k) = \frac{E_\tau(q) \cdot B(q)}{C(q)} u(k)$$

EQUATION 145

$$u_{MV}(k) = \frac{F_\tau(q)}{E_\tau(q) \cdot B(q)} y(k)$$

EQUATION 146

The following steps can be used to implement MV control (FIGURE 87):

1. Measure “y” and “u” data and form “φ”, as shown in EQUATION 140
2. Use RLS ID to identify model parameters “θ”, as shown in EQUATION 139
3. Determine the MV parameters for “F_τ”, “E_τ” and “B” from “θ”
4. Compute the MV control input “u_{MV}”, as shown in EQUATION 146
5. Repeat steps 1 – 4 for each sample period.

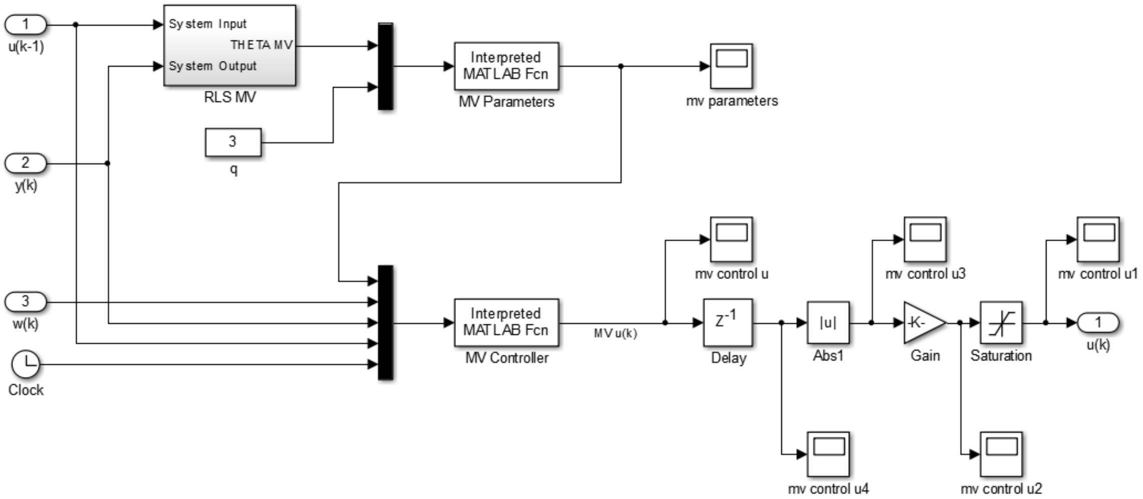


FIGURE 87: MV control implemented in Simulink

5.5.2 Adaptive Control (MV Control) Results

MV control simulation results are discussed in this section, where the system input is PZR mass surge flow rate “$\dot{m}_{surge}$” and the system output is the PZR pressure “P”. The MV control methodology presented in Section 5.5.1 was used to obtain a MV control input for the PZR spray (to reduce pressure). The PZR spray was selected for the same reasons discussed in Section 5.3.6. The focus is on the control input for the PZR spray to reduce the magnitude of pressure pulses. The PZR heater input can also be driven by an LQR (instead of on/off or proportional control). However, in the simulations presented, the difference in control performance between the conventional and LQR control of the heaters was negligible, because of the PZR and heater sizing constraints imposed. The PZR heaters exhibit control saturation, so the only way to improve control performance would be to increase the heater size. Therefore, the control input for the PZR heaters is not addressed in this paper, since the heater sizes are considered fixed for the simulation to maintain representation of the basis PZR design (i.e., the focus of this research is not to redesign the PZR or actuators, but improve performance of the existing actuators via advanced control strategies).

Plots of the MV control performance are provided in FIGURE 88 and FIGURE 89. The mass surge flow rate “$\dot{m}_{\mathrm{surge}}$” shown in FIGURE 18 is the system input for the RLS ID and the PZR pressure “P” shown in FIGURE 88 that results from the MV control is the system output for the RLS ID. The PZR pressure “P” (i.e., resulting output process variable of the PZR) and the PZR spray flow rate “$\dot{m}_{\mathrm{spray}}$” (i.e., control input to the PZR) are shown in FIGURE 88 for the simulation duration (600 seconds).

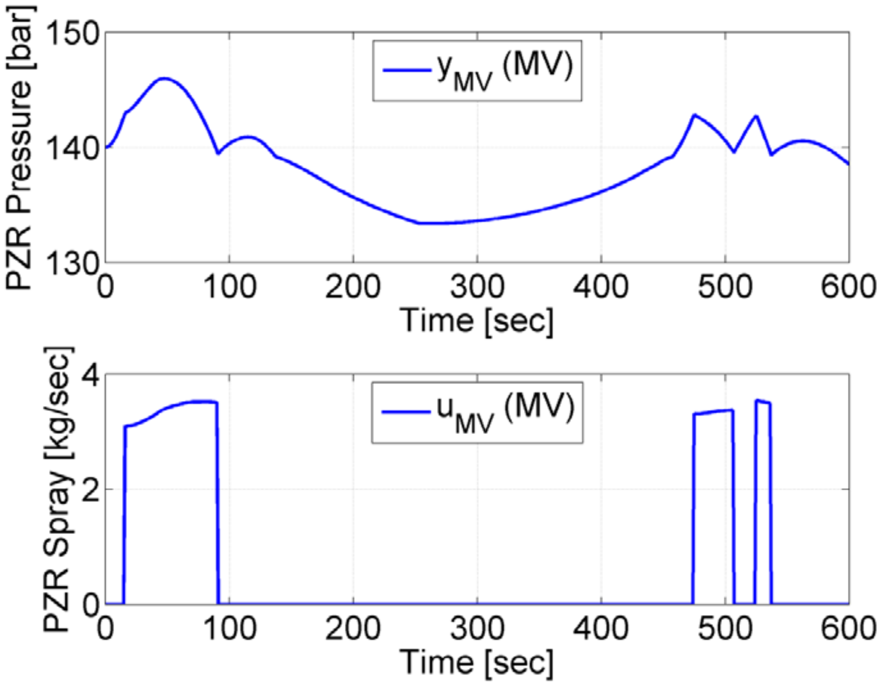


FIGURE 88: PZR pressure (output) and PZR spray (control input) with MV [95]

5.5.3 Adaptive Control (MV Control) Comparison

A (PZR pressure) performance comparison for the PZR with conventional control, optimal (LQR) control, and MV control are presented in TABLE 9 and FIGURE 89.

The maximum PZR pressure with the MV feedback controller is approximately 145.97bar, which occurs at 48 seconds. A reduction in both the maximum pressure and time required to reach the maximum pressure was achieved with MV control (as compared with the conventional control performance).

TABLE 9: MV statistical performance measures for PZR application [95]

	Conventional	**Optimal**	**MV Control**
Max Pressure (bar)	151.98	146.84	145.97
Time (sec)	75	62	48

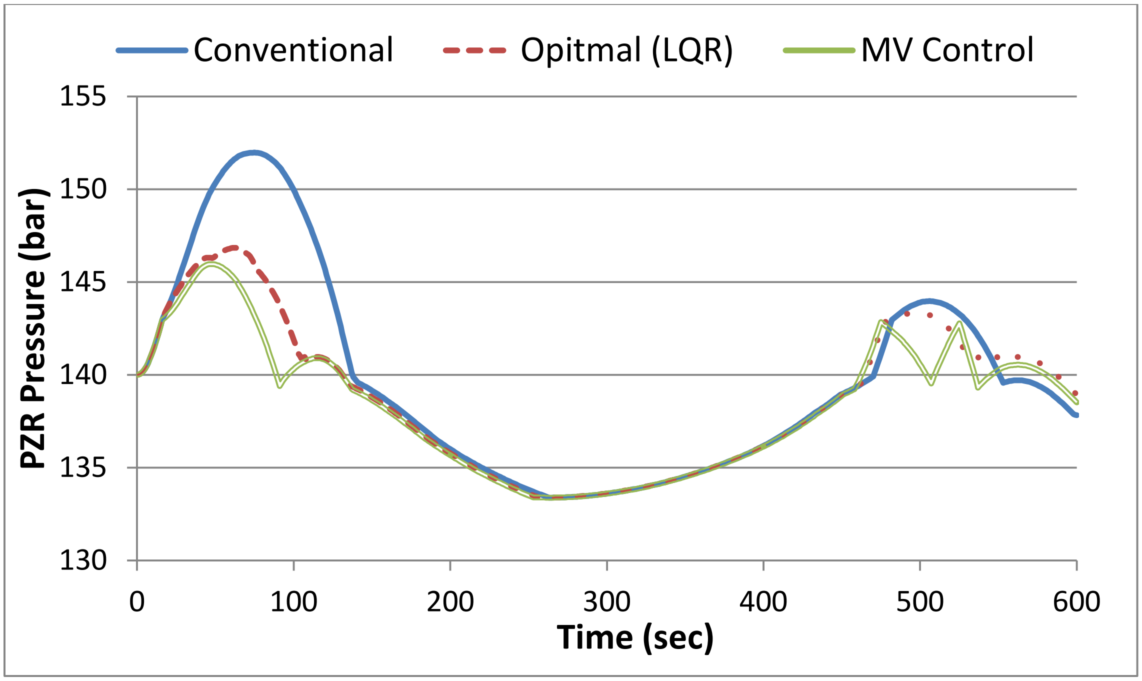


FIGURE 89: Dynamic PZR model pressure – conv., LQR, and MV control

5.6 Primary Issues Associated with Existing Control Methods

This section identifies several of the overall strengths and limitations with the existing control methods applied to the dynamic PZR model. Strengths and limitations specific to each individual control method (considered in CHAPTER 5) are described in Sections 5.3.4, 5.3.6, 5.4.3, and 5.5.3. However, there are some common controller attributes and challenges that are useful to identify. CHAPTER 6 presents some proposed methods to address these challenges.

1) The advanced controllers (e.g., optimal, STR PID, and MV) investigated in CHAPTER 5 exhibited a noticeable performance improvement over the conventional controller, especially with adaptive control techniques, as seen in FIGURE 84 and FIGURE 89. In general, as the control method used became more advanced, a greater reduction in the pressure spike and

response time were achieved. However, these controllers may not act dynamically enough to respond to the transient dynamics of the physical system. For example, the PZR spray setpoint for these simulations was 142.7bar (i.e., in the best case the pressure will increase to at least this value, since the PZR spray actuator does not act until this pressure value is reached), with a desired reference value of 140bar. Still, even with the adaptive optimal (LQR) controller with RLS ID, the pressure during the first pressure spike increased to 146.8bar. With adaptive MV control the maximum pressure spike occurred at 146.0bar. While it may not be realistic to achieve a controller that holds to pressure from exceeding 142.7bar, additional reduction in the pressure overshoot may be possible.

2) The RLS ID is a useful tool to adaptively identify the changing dynamics of a system. However, as seen with FIGURE 67 and FIGURE 81, it would be beneficial to reduce error in the system identification (i.e., $y_m \rightarrow y$).

3) The PZR control actuators (e.g., PZR spray and heaters) do not respond instantaneously, but are physically limited in response time. However, typical response times are available per vendor documentation (e.g., PZR spray valve and actuator). The control commanded for the electric heaters did not change quickly and in general the system response to the heaters was slower than the time required for initiating a command. Electric heaters can be actuated much faster than the system can respond; therefore, response time is not an issue. However, the PZR spray is controlled via modulation of a control value (e.g., Fisher® SS-84PSV4 Rotary Control Valve [88]). A typical PZR spray value

actuator (e.g., the Fisher® 1052PSV Rotary Actuator is a pneumatic spring/diaphragm type actuator that can be used on use on a Fisher® SS-84PSV4 valve for PZR spray control [89]) might exhibit a total actuation stroke time as fast as one second to open [89]. This mean the PZR spray changes occurring every second (i.e., the sampling time of the system) can marginally be achieved by the actuator (provided the commanded change is not full closed to full open repeatedly), since the changes commanded from one time step to the next are generally small compared with the total actuation. The PZR spray control valve and actuator may wear out sooner, but it is much less expensive than the cost of a PZR. Therefore, no additional action is anticipated regarding the PZR spray control valve response time.

4) As with any physical system, the control inputs (PZR spray and heaters) can only alter the physical system (PZR pressure) by a finite amount. With an infinite amount of energy, it would be possible to control the physical system as desired. However, with finite control inputs, the control capabilities are limited. Specifically, there is a maximum amount of heat the PZR heaters can add and there is a maximum mass flow rate for the PZR spray. For the controller, this limitation appears as a gain limit (i.e., saturation limit). Therefore, the controller (e.g., optimal) may command a certain gain to drive the system to the desired operating condition, but may not be able to achieve the commanded gain, if it is beyond the limits for that control input.

CHAPTER 6: NEW CONTROL METHOD FOR PZR IN A PULSED SYSTEM

This chapter presents a new control methodology that is proposed for control of PZR pressure in a pulsed system. Since the control methods discussed in CHAPTER 5 are considered in the available literature for similar problems (e.g., PWR NPP control), additional control methodologies are investigated that function as a unique hybridization of these techniques to achieve an improved PZR pressure control performance in a pulsed thermal-hydraulic system. This new pressure control methodology is focused on minimizing response time, settling time, and overshoot to maintain the controlled variable (pressure) as constant as possible in the pulsed (transient) system.

In summary the assertion is that advanced control methods (e.g., dynamic control with RLS system identification including a model estimator/corrector of the system output) provide improved control performance over conventional control methods for control of PZR pressure in a pulsed cooling system. Potential benefits of this improved control include:

1) Decoupling of safety and non-safety functions (e.g., pressure in normal operation does not approach the safety set-points)
2) Increase in component life by reducing fatigue and stress corrosion cracking (SCC), which are major concerns in NPPs. This is accomplished by reducing the magnitude of each PZR pressure spike.
3) Improved plant PRA by reducing plant risk via (a) the cooling system pressure spike magnitudes being farther away from the safety set-points (i.e.,

reduce chance of a safety event during normal operation), (b) improved ability of the pant to restore the cooling system pressure to a stable state following a transient event [78], and (c) reduced fatigue smaller pressure spikes at each plasma pulse.

The PZR pressure control research plan that was used is provided in FIGURE 14. Step 7 focuses on developing a new control method to address the dynamic control problems. FIGURE 90 provides further detail of this step.

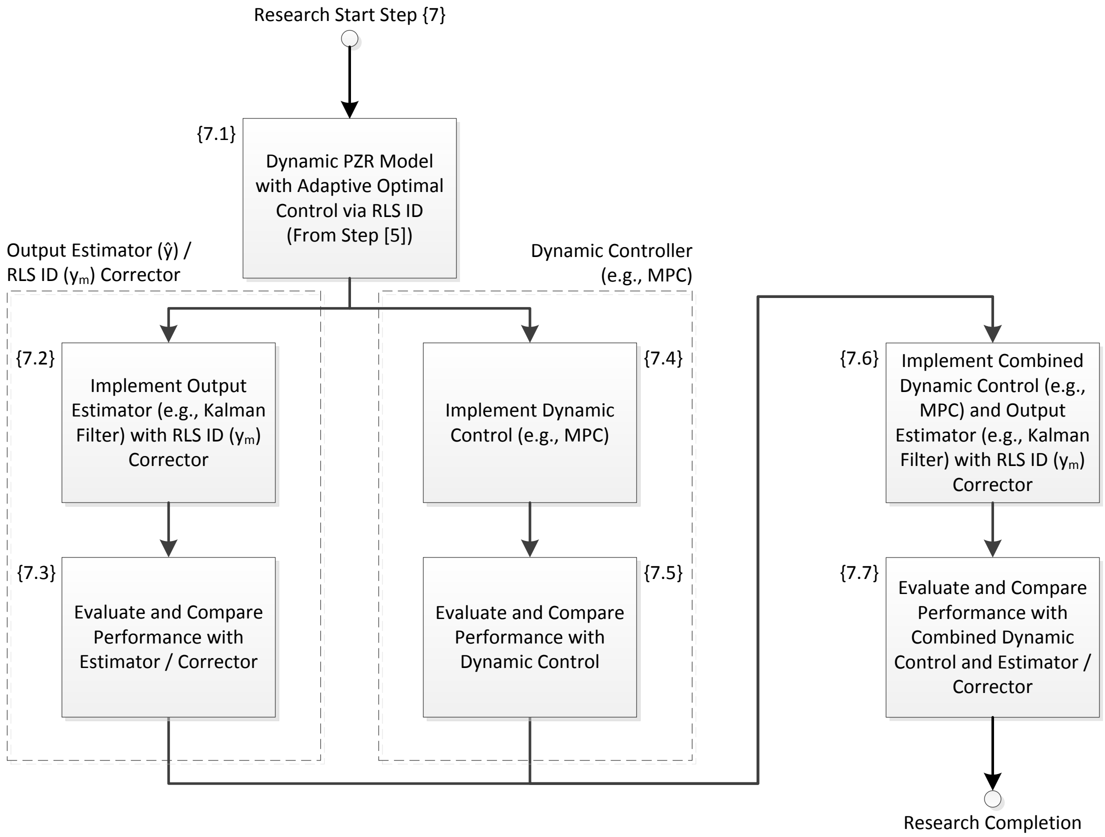

Research Start Step {7}
{7.1}
Dynamic PZR Model with Adaptive Optimal Control via RLS ID (From Step [5])
Output Estimator (ŷ) / RLS ID (y_m) Corrector
Dynamic Controller (e.g., MPC)
{7.2}
Implement Output Estimator (e.g., Kalman Filter) with RLS ID (y_m) Corrector
{7.3}
Evaluate and Compare Performance with Estimator / Corrector
{7.4}
Implement Dynamic Control (e.g., MPC)
{7.5}
Evaluate and Compare Performance with Dynamic Control
{7.6}
Implement Combined Dynamic Control (e.g., MPC) and Output Estimator (e.g., Kalman Filter) with RLS ID (y_m) Corrector
{7.7}
Evaluate and Compare Performance with Combined Dynamic Control and Estimator / Corrector
Research Completion

FIGURE 90: PZR pressure control research plan – step 7 details

6.1 Proposed Control Architecture

A proposed new PZR pressure control architecture is provided in FIGURE 91. Specifically, the new control architecture focuses on two aspects: (1) using a dynamic-adaptive controller and (2) coupling an estimator/corrector with the system identification to produce a more accurate system model.

1) Dynamic-Adaptive Controller:

To address the first limitation item listed in Section 5.6, additional pressure control performance might be obtained from a controller that behaves more dynamically as the system changes during a transient (see [31] and [93]), so that additional reduction in the pressure overshoot may be achieved. Therefore, the new architecture uses a dynamic-adaptive controller (e.g., hybrid LQR and MV controller with RLS ID system at each time-step), since the conventional controller alone does not act dynamically. For additional detail on the dynamic-adaptive controller (i.e., LiMe) see Section 6.3.

2) Estimator/Corrector for Improved System Identification:

As discussed in the second limitation item listed in Section 5.6, it would be beneficial to reduce error in the system identification (i.e., $y_m \rightarrow y$). It is not possible to eliminate all error between the RLS ID estimated system output (y_m) and the actual system output (y) for several reasons. One of the main reasons is that the actual system output is not known. Rather, the pressure of the physical system is obtained via an instrument measurement, which also has errors captured in a measurement uncertainty budget. The usual uncertainties associated with safety instruments in a NPP are well documented and analyzed to ensure safe operation of the plant. Typically, PZR pressure has three redundant transmitters for safety. Therefore, various statistical methods can be applied to estimate the system output.

These statistical methods tend to produce an estimate that is more accurate than any one of the measured process values alone. Consequently, it would be beneficial to correct the RLS ID based on an estimate of the actual process output, so that the identified system model (THETA) would have greater accuracy. Therefore, to address the second limitation item listed in 5.6, a state estimator (e.g., Kalman Filter [87]) can be used to estimate the PZR pressure (i.e., system output denoted as $\hat{y}$) and then that $\hat{y}$ estimate used to iteratively correct the identified system error, until the error between y_m and $\hat{y}$ is below an acceptable value, as shown in FIGURE 91. Accordingly, the new architecture contains an estimator and then uses the $\hat{y}$ estimate to correct the RLS ID for an improved model of the system (i.e., Estimator/Corrector) [79], since the fourth order RLS ID model output (y_m) exhibited some error in comparison to the actual output (y). Therefore, reducing the error should improve the pressure control (i.e., a more accurate model of the system at any given time).

Also, using a state estimator (e.g., Kalman filter) is desired, since the system output (pressure) is a measured variable with noise. In a noisy system, the Kalman Filter tends to produce a more accurate estimate of the true process value (e.g., pressure) than any single measurement alone, as discussed in [90], [91] and [99], especially when there are multiple redundant measurements available (i.e., as is the case with PZR pressure). Kalman filtering theory and implementation algorithms are well documented in publications (e.g., [93], [98], [100], and [101]). Measured input and output data for a system are provided as inputs to the Kalman filter, which then produces state estimates, as shown in FIGURE 91 and FIGURE 92. An overview of the Kalman filter algorithm is provided in FIGURE 92, which includes the associated equations, where L_K is the Kalman gain, P is the error covariance, and the other terms are as described in Section 4.4.1 with the current estimator. FIGURE 93 provides an example plot demonstrating how the state estimates from a Kalman filter exhibit less error than the measured process value alone. This demonstration example used the simplified fourth order state space PZR model presented in Section 5.3.2 to represent the system and the $\dot{m}_{surge}$ data in FIGURE 18 as the input, with assumed instrument noise. See Section 6.4 for additional information on the technique to improve system knowledge (i.e., RICK).

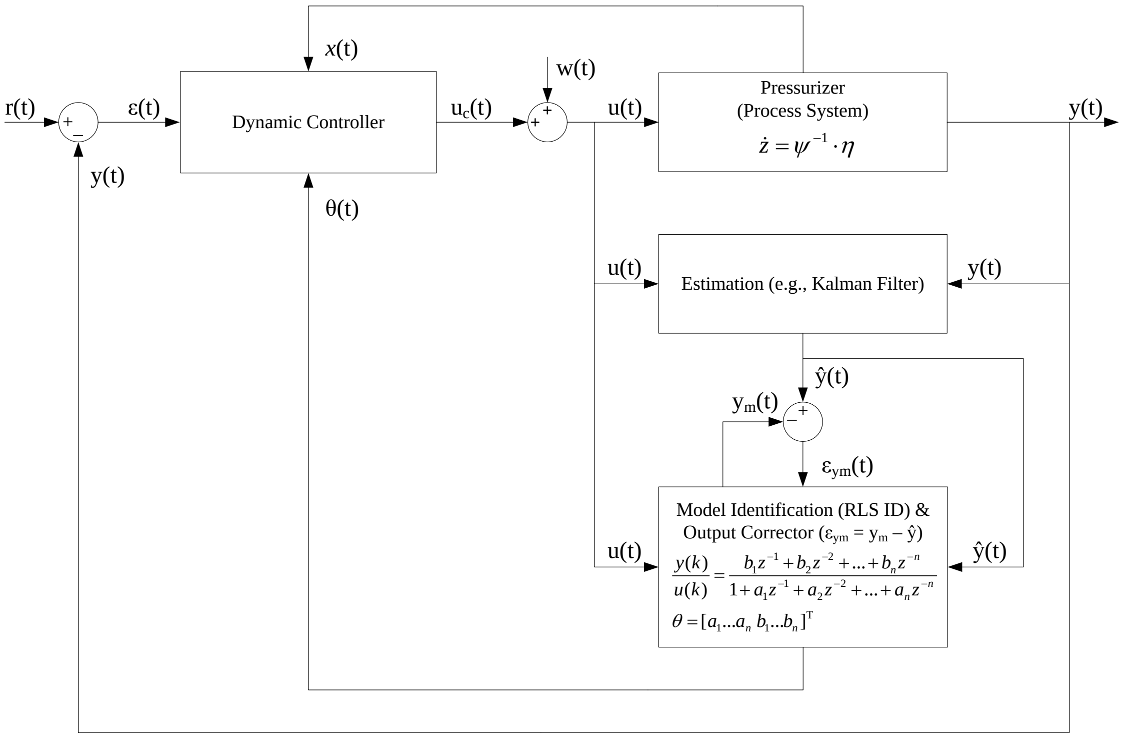

x(t)
w(t)
r(t)
+
–
ε(t)
Dynamic Controller
u_c(t)
+
+
u(t)
Pressurizer
(Process System)
$\dot{z} = \psi^{-1} \cdot \eta$
y(t)
y(t)
θ(t)
u(t)
Estimation (e.g., Kalman Filter)
y(t)
ŷ(t)
y_m(t)
+
–
ε_ym(t)
Model Identification (RLS ID) &
Output Corrector (ε_ym = y_m – ŷ)
$\frac{y(k)}{u(k)} = \frac{b_1 z^{-1} + b_2 z^{-2} + ... + b_n z^{-n}}{1 + a_1 z^{-1} + a_2 z^{-2} + ... + a_n z^{-n}}$
$\theta = [a_1 ... a_n \; b_1 ... b_n]^T$
u(t)
ŷ(t)

FIGURE 91: Dynamic PZR model – dynamic control with new “K” gain matrix via RLS ID model with estimator corrector at each time step

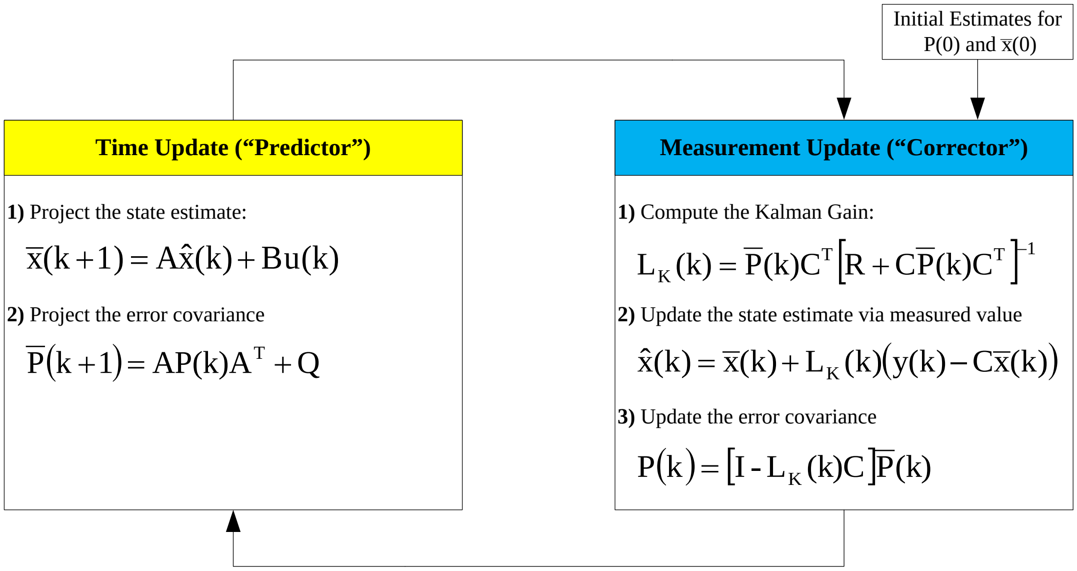

Initial Estimates for P(0) and $\overline{x}(0)$
Time Update ("Predictor")
1) Project the state estimate:
$\overline{x}(k+1) = A\hat{x}(k) + Bu(k)$
2) Project the error covariance
$\overline{P}(k+1) = AP(k)A^T + Q$
Measurement Update ("Corrector")
1) Compute the Kalman Gain:
$L_K(k) = \overline{P}(k)C^T\left[R + C\overline{P}(k)C^T\right]^{-1}$
2) Update the state estimate via measured value
$\hat{x}(k) = \overline{x}(k) + L_K(k)\left(y(k) - C\overline{x}(k)\right)$
3) Update the error covariance
$P(k) = \left[I - L_K(k)C\right]\overline{P}(k)$

FIGURE 92: Kalman filter algorithm overview (adapted per [36] and [40])

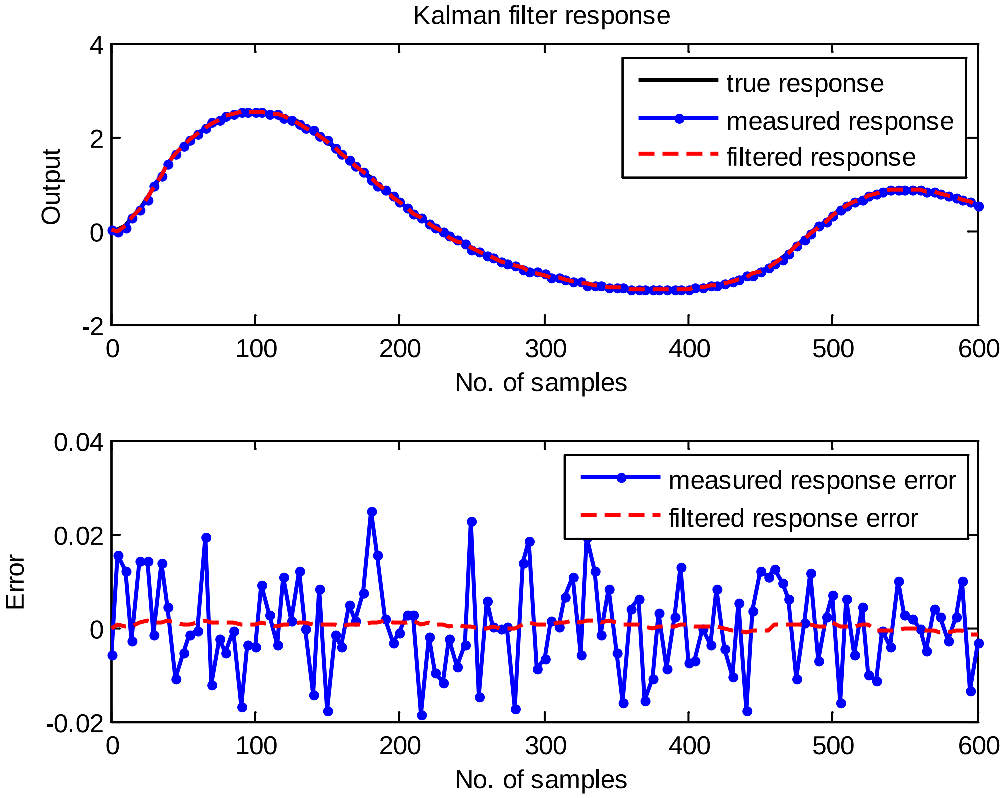


FIGURE 93: Example PZR Kalman filter estimate comparison to measured data

6.2 Research Plan for Proposed PZR Control Method

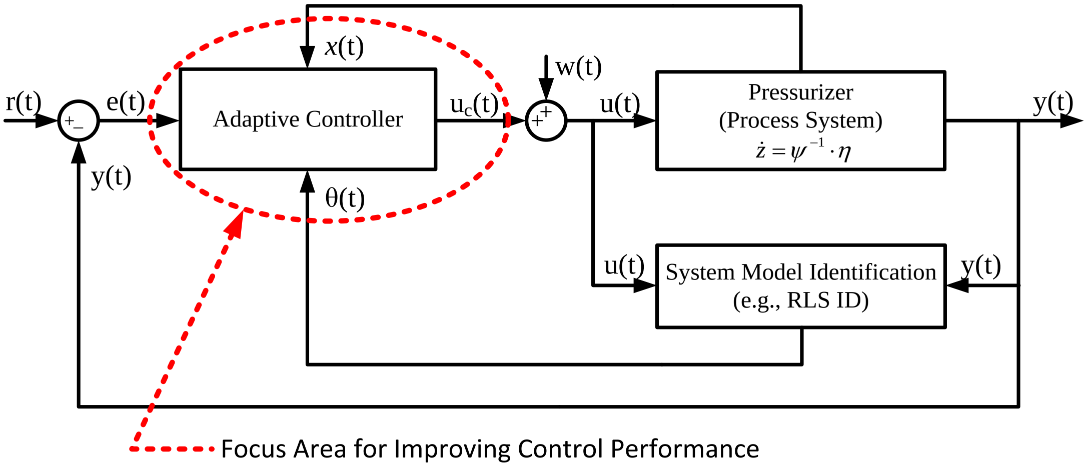
Planning is essential for success of any endeavor (i.e., poor planning produces poor results). Therefore, TABLE 10 provides a SMART plan [92] that was used to investigate the proposed PZR pressure control method for a pulsed system.

TABLE 10: S.M.A.R.T plan for PZR pressure control research

Planning Criteria	**Descriptions**
Specific	In order to improve PZR pressure control in a pulsed cooling system, this research simulated the new PZR pressure architecture (see FIGURE 91) that focuses on two aspects: (1) dynamic-adaptive controller (LiMe) and (2) estimator/corrector for improved system identification (RICK). Following simulation of the new control architecture, the performance results were evaluated and compared with the other control methods simulated for the dynamic PZR model previously (see CHAPTER 5).
Measurable	Target additional reduction of the maximum pressure spike (e.g., 146.8bar for optimal control) to be less than the first high pressure alarm setpoint provided in Section 2.2 (i.e., maximum pressure ≤ 146bar).
Agreeable	There are several motivations for research of this topic (see Section 1.5 and CHAPTER 6) and the proposed methods are aimed to address the currently identified primary challenges with PZR pressure control in a pulsed cooling system (see Sections 5.6 and 6.1).
Realistic	The new control architecture proposed for PZR pressure control has valid theoretical basis and each proposed new part has noted success individually in other applications (see Section 6.1). However, these method(s) have not been applied to this specific problem in this arrangement (see Section 1.3). Therefore, it is valid and realistic to investigate these methods for this problem to further the understanding of this topic area.
Timely	The targeted completion for this research was fall 2015, which provided sufficient time to investigate these specific items.

6.3 LiMe (Hybrid LQR and MV Control)

A method for improved pressurizer pressure control via a hybrid control architecture that combines a Linear Quadratic Regulator (LQR) and Minimum Variance (MV) controller is proposed. The LQR provides “an optimal control method that focuses on minimizing the control cost associated with the inputs and states of the system. The MV controller, conversely, focuses on minimizing the control costs associated with the output of the system. Combining these two control methods enables enhanced control during transient conditions over either of these control methods alone.” [95] FIGURE 94 graphically identifies this focus area for improved control performance. The information and results in this section is adapted from M. Smith and S. Kamalasadan ([95]).



Focus Area for Improving Control Performance

FIGURE 94: LiMe focus for adaptive PZR pressure control [95]

6.3.1 LiMe Method

The information in this section is adapted from Smith and Kamalasadan ([95]), which provides greater detail of the LiMe method. The LiMe (Linear Quadratic Regulator and Minimum Variance Control) method combines LQR ([98]; [36]; and [20]) with MV control ([93] and [94]), as shown in FIGURE 95, to improve system control. The LiMe method is aimed at improving control accurate (i.e., better control performance than either the LQR and MV control methods), while remaining computationally acceptable for online implementation. The proposed architecture combining these techniques is realistically achievable, since both LQR and MV control are capable of online implementation. At each time step LQR and MV control both produce values for the control input "u(k)" (i.e., PZR spray mass flow rate). The RLS ID method is used to obtain a new model of the system at each time step (see Section 5.3.5 for additional information regarding RLS ID).

The LiMe control input "u_{LiMe}" is computed by weighting the LQR and MV control input values in accordance with their sensitivity to the system output "y" (i.e., PZR pressure) via a LiMe weighting sensitivity factor "α". The LQR control input (uLQR) is obtained via the method provided in Section 4.4.1 (also see APPENDIX B, "dlqr_fun.m"). The MV control input (uMV) is obtained via the method provided in Section 5.5.1 (also see APPENDIX B, "mv_param_R0.m" and "mv_ctrl_R0.m"). The process steps listed below are used to implement the LiMe technique (see FIGURE 96 and FIGURE 97) [95]:

1) Compute "u_{LiMe}" via the LiMe control law per EQUATION 147:

$$u_{LiMe} = \alpha \cdot u_{LQR} + (1 - \alpha) \cdot u_{MV}$$

EQUATION 147

2) The LiMe weighting sensitivity factor "α" is computed per EQUATION 148:

$$\alpha = \frac{R_{LQR}}{R_{LQR} + R_{MV}}$$

EQUATION 148

3) The sensitivity “R” of the output error with respect to the LQR and MV inputs is computed per EQUATION 149, EQUATION 150, and EQUATION 151:

$$R_{LQR} = \left| \frac{\partial(e_{PZR})}{\partial(u_{LQR})} \right| \approx \left| \frac{e_{PZR}(k) - e_{PZR}(k-1)}{u_{LQR}(k) - u_{LQR}(k-1)} \right|$$

EQUATION 149

$$R_{MV} = \left| \frac{\partial(e_{PZR})}{\partial(u_{MV})} \right| \approx \left| \frac{e_{PZR}(k) - e_{PZR}(k-1)}{u_{MV}(k) - u_{MV}(k-1)} \right|$$

EQUATION 150

$$e_{PZR} = r - y_{actual} \approx r - y_{meas.}$$

EQUATION 151

4) The constraints in EQUATION 152, EQUATION 153, EQUATION 154 must be satisfied to apply EQUATION 148, EQUATION 149, and EQUATION 150:

$|e_{PZR}(k) - e_{PZR}(k-1)| > 0$

EQUATION 152

$|u_{LQR}(k) - u_{LQR}(k-1)| > 0$

EQUATION 153

$|u_{MV}(k) - u_{MV}(k-1)| > 0$

EQUATION 154

5) If any of the constraints in EQUATION 155, EQUATION 156, or EQUATION 157 are not satisfied, then “α” is obtained per EQUATION 155, EQUATION 156, or EQUATION 157:

If $|e_{PZR}(k) - e_{PZR}(k-1)| = 0$, Then $\alpha = 0.5$ (i.e., equal weighting)

EQUATION 155

If $|u_{LQR}(k) - u_{LQR}(k-1)| = 0$, Then $\alpha = 0$ (i.e., let MV drive system)

EQUATION 156

If $|u_{MV}(k) - u_{MV}(k-1)| = 0$, Then $\alpha = 1$ (i.e., let LQR drive system)

EQUATION 157

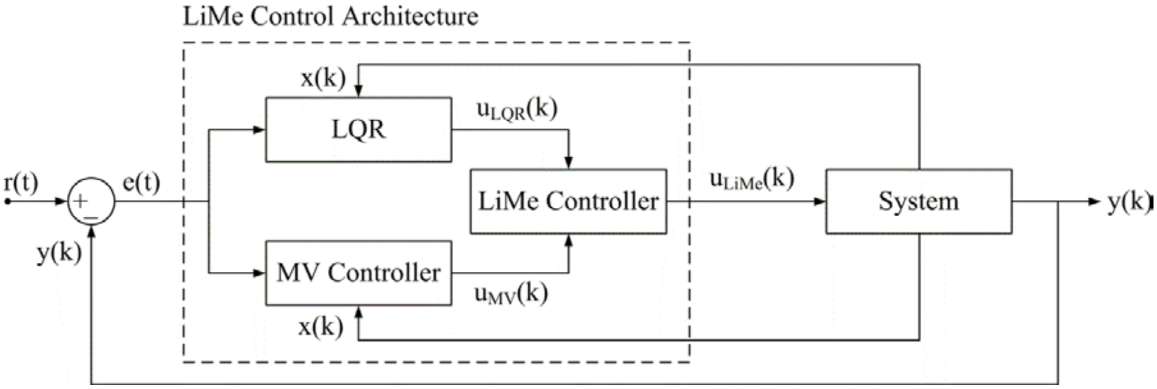


FIGURE 95: LiMe overview block diagram [95]

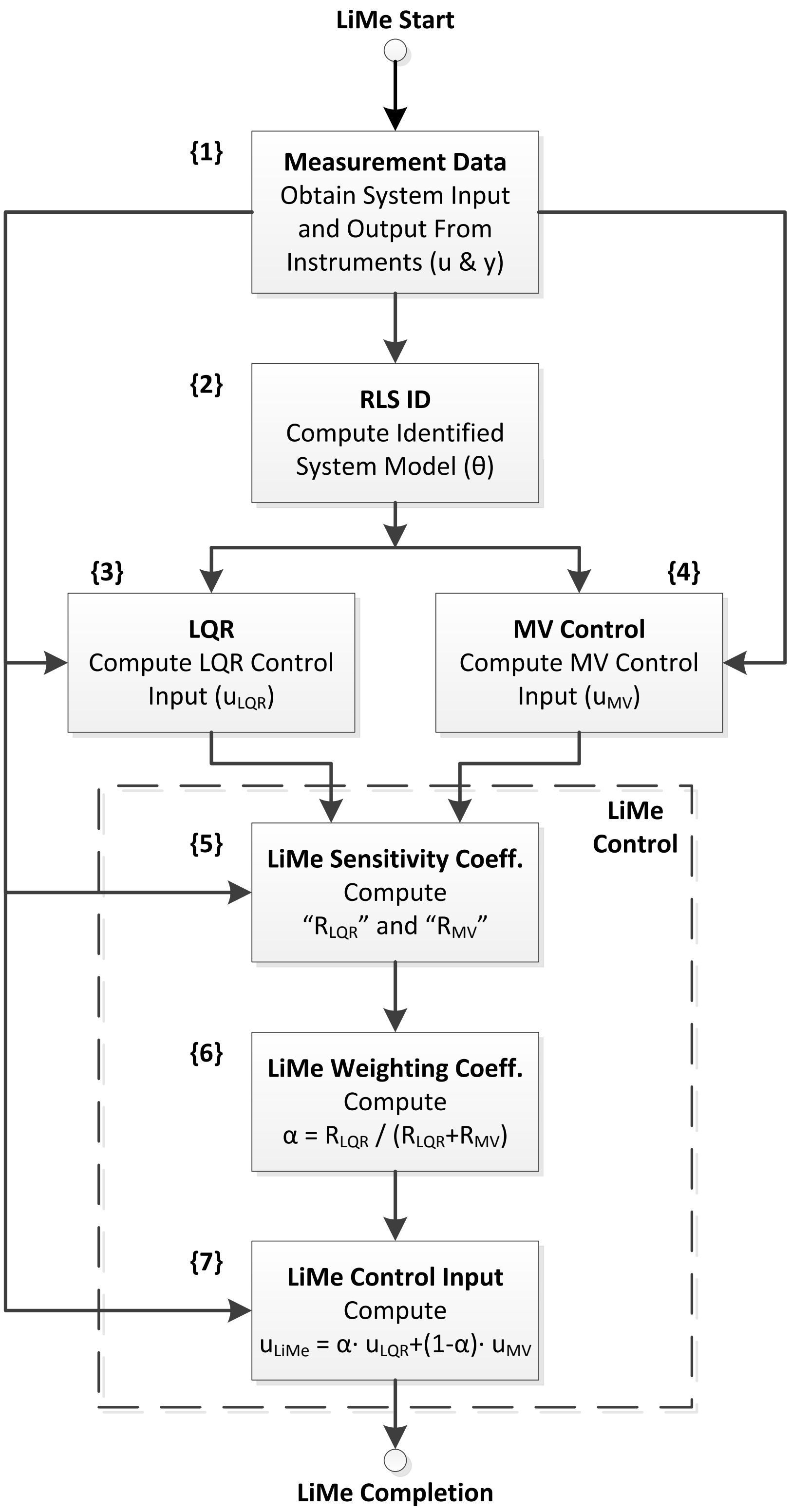

LiMe Start
{1}
Measurement Data
Obtain System Input and Output From Instruments (u & y)
{2}
RLS ID
Compute Identified System Model (θ)
{3}
LQR
Compute LQR Control Input (u_{LQR})
{4}
MV Control
Compute MV Control Input (u_{MV})
LiMe Control
{5}
LiMe Sensitivity Coeff.
Compute "R_{LQR}" and "R_{MV}"
{6}
LiMe Weighting Coeff.
Compute
$\alpha = R_{LQR} / (R_{LQR}+R_{MV})$
{7}
LiMe Control Input
Compute
$u_{LiMe} = \alpha \cdot u_{LQR}+(1-\alpha) \cdot u_{MV}$
LiMe Completion

FIGURE 96: LiMe process steps [95]

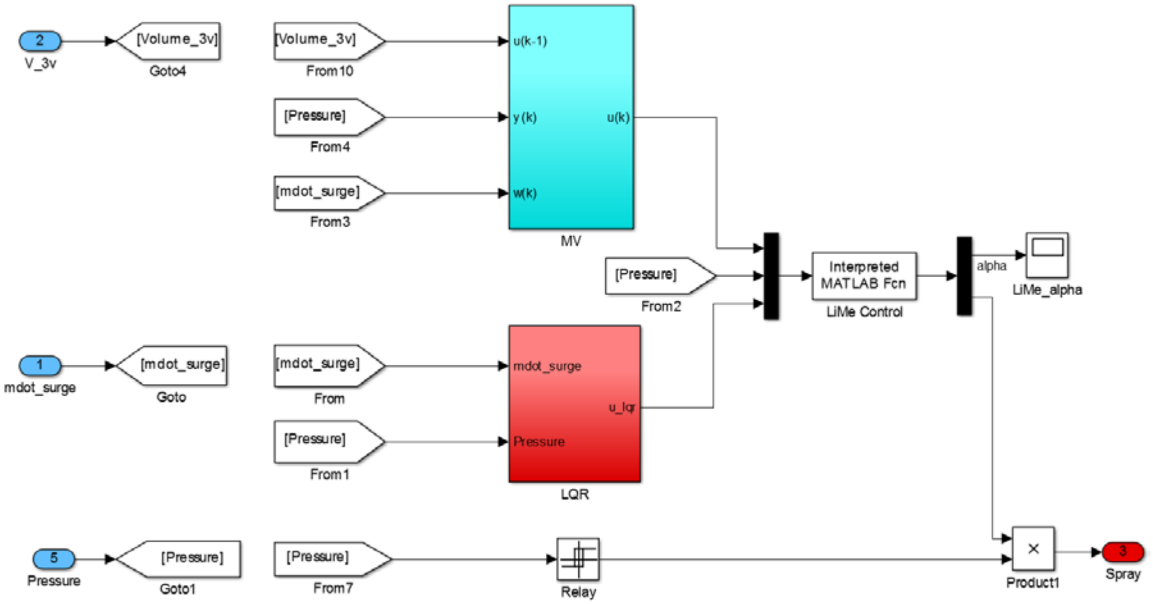


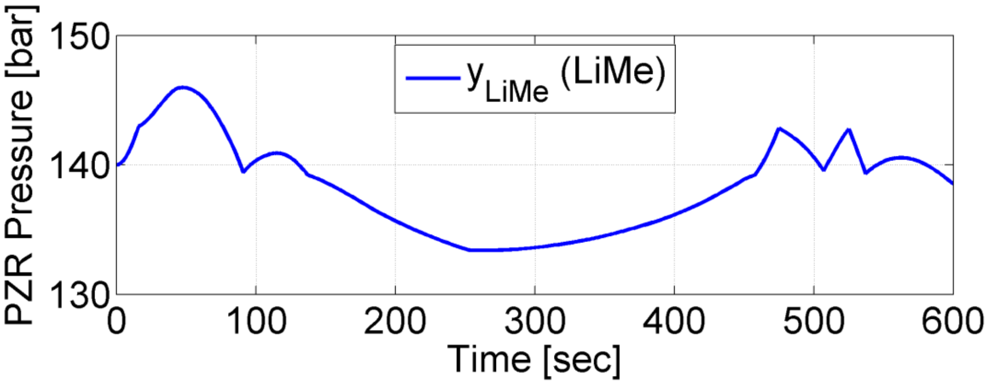
FIGURE 97: LiMe implemented in Simulink

6.3.2 LiMe Simulation Results

PZR simulation results for LiMe are provided in FIGURE 98, FIGURE 99, FIGURE 100, and FIGURE 101. TABLE 11 provides statistical performance measures. FIGURE 98 shows the PZR pressure (output) and PZR spray (control input). As shown in FIGURE 99, improved control performance was achieved with the LiMe control method (e.g., fastest overall response time with the least overshoot) in comparison with not only conventional control, but also LQR and MV control. As seen in FIGURE 101, the LiMe weighting sensitivity factor “α” varies in value as the system dynamics change. The LQR and MV controller each provided the best control at a few discrete point during the simulation (i.e., when “α” equals 1 or 0). A combination of LQR and MV control inputs, however, resulted in the best LiMe input (i.e., $0 < \alpha < 1$) for the majority of the simulation. The LiMe method desirably resulted in the smallest maximum pressure spike of all three control methods (see FIGURE 99). The LiMe method also commanded the smallest maximum magnitude for the control input compared with the other control methods and exhibited less fluctuation than the other control methods (see FIGURE 100). As a result, the PZR pressure also exhibited reduced fluctuation (see FIGURE 99). In comparison with the LQR and MV control methods or conventional control, the PZR pressure was more predictable and controllable using the LiMe control method.

TABLE 11: LiMe statistical performance measures for PZR application [95]

	Conventional	**LQR**	**MV Control**	**LiMe**
Max Pressure (bar)	151.97	146.84	145.97	145.76
Time (sec)	75	62	48	52



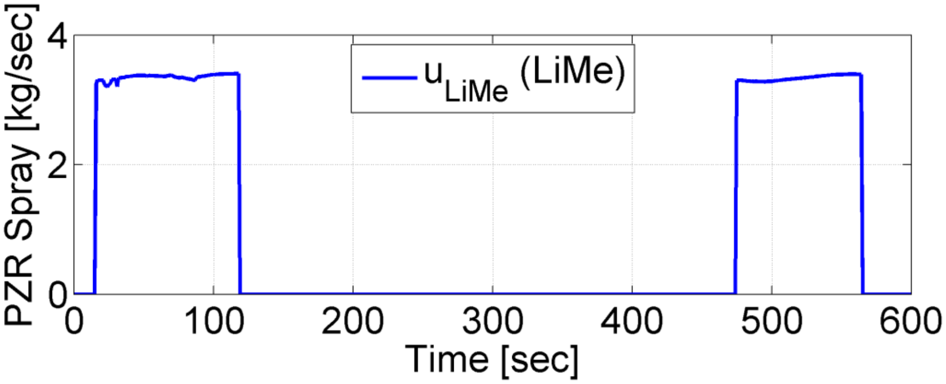


FIGURE 98: PZR pressure and PZR spray (control input) with LiMe [95]

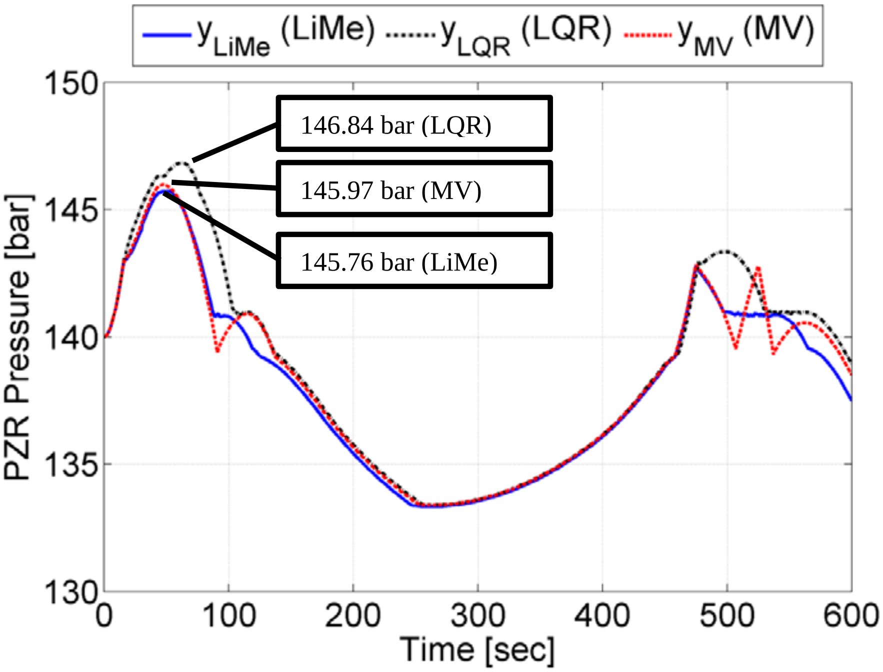


FIGURE 99: PZR pressure comparison for LQR, MV, and LiMe [95]

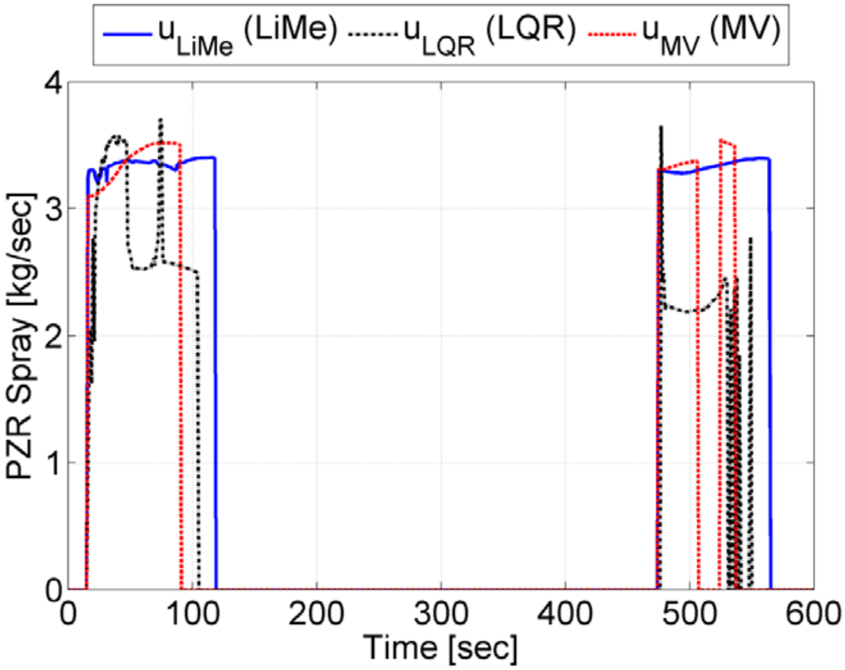


FIGURE 100: PZR spray comparison for LQR, MV, and LiMe [95]

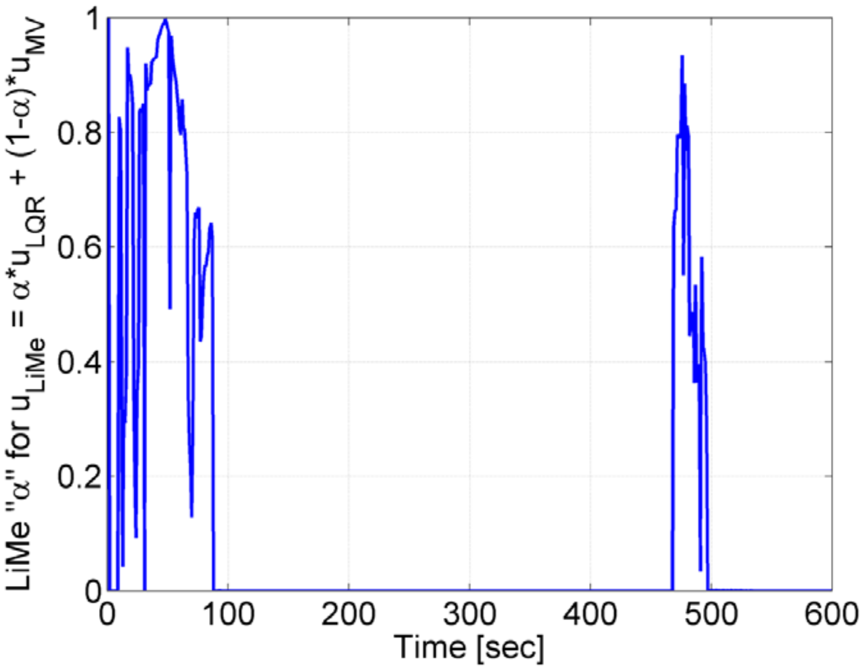


FIGURE 101: LiMe control weighting coefficient “α” [95]

6.4 RICK (System Knowledge Improvement)

The information and results in this section is adapted from M. Smith and S. Kamalasadan ([95] and [96]). A method is proposed for improved pressurizer system knowledge. FIGURE 102 graphically identifies this focus area for improved pressurizer system knowledge. This method is “based on a hybrid knowledge base using Recursive Least Squares Identification (RLS ID), a Kalman filter, and a model corrector. RLS ID enables empirical identification of system model parameters based on measured data, so the model updates dynamically as the system changes. A Kalman filter (sometime called a Linear Quadratic Estimator) produces estimates of state variables (e.g., pressure) that tend to be more accurate than the measured variables alone, considering typical uncertainties, which improves system knowledge. The model corrector is used to improve the model accuracy with the Kalman filter state estimates.” [96]

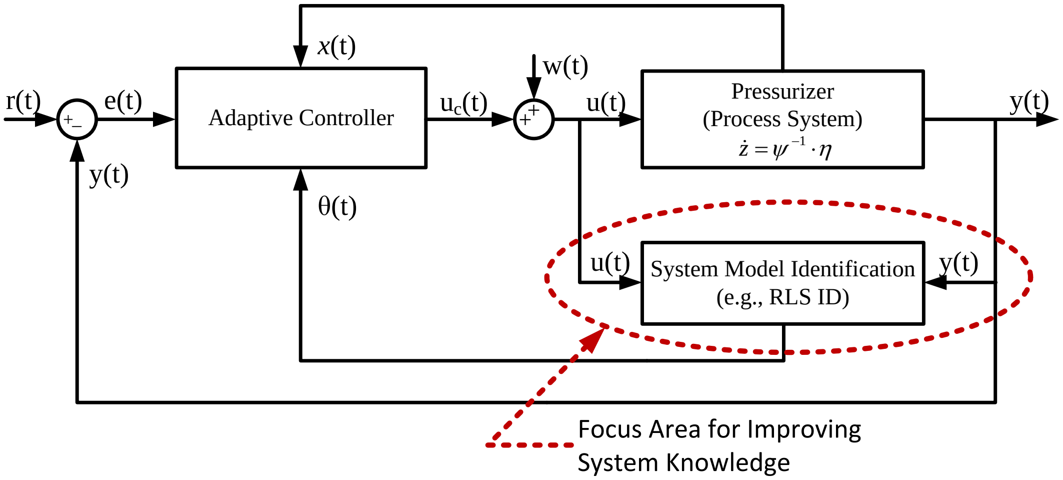


FIGURE 102: RICK focus area for adaptive PZR pressure control [95], [96]

6.4.1 RICK Method

The information in this section is adapted from Smith and Kamalasadan ([95] and [96]), which provides greater detail of the RICK method. As shown in FIGURE 104, RICK (Recursive Least Squares Identification with Corrector and Kalman Filter) combines a Recursive Least Squares Identification (RLS ID) system model ([36] and [93]) with a Kalman filter ([36]; [93]; [98]; [99], [100]; and [101]) and adds a model corrector. The RLS ID method is used to obtain a new model of the system at each time step (see Section 5.3.5 for additional information regarding RLS ID). The RICK method is focused on achieving a more accurate identified system model (i.e., the output of the model more closely equals the actual output of the system). RLS ID and Kalman filtering are both capable of online implementation (i.e., these techniques are computationally achievable with existing technology).

To improve system knowledge, the system output from the Kalman filter output estimate "$\hat{y}(k) = H\bar{x}(k)$" is sent to the RLS ID, instead of the measured output, since $\hat{y}(k)$

tends to be more accurate than the single measurement. In support of using a Kalman filter in this manner, Kalman filters were used elsewhere to provide corrected sensor readings to update empirical system models ([79]).

The RLS ID model output without correction is represented with the symbol "$\hat{\theta}(k)$". This model parameter estimation "$\hat{\theta}(k)$" can be improved by applying knowledge of the system output from the Kalman filter "$\hat{y}(k)$" with a relationship of the state space model, where "y(k) = Hx(k)", to update the model parameters. This model corrector technique focuses on updating the identified state space measurement matrix "H_θ", so that the corrected model output "$y_{m,cor}(k)$" equals the Kalman filter estimated output "$\hat{y}(k)$". "H_θ" is obtained from converting the discrete transfer function model "$\hat{\theta}(k)$" to state space format. This model corrector technique does not update the system dynamics in the identified model, since the "F_θ" and "G_θ" matrices are not changed. The RICK technique can be implemented as follows (see FIGURE 103 and FIGURE 105) [95], [96]:

1) Compute the RLS ID model output estimate "$y_m(k)$" per EQUATION 158:

$$y_m(k) = \varphi^T(k+1)\hat{\theta}(k)$$

EQUATION 158

2) Construct the transfer function model from the RLS ID parameters.
3) Convert the RLS ID discrete transfer function model to discrete state space.
4) Extract the discrete state space model matrices. These are denoted with a subscript "θ", since they are obtained via RLS ID (i.e., "F_θ", "G_θ", and "H_θ").
5) Find the RLS ID model estimated states "$x_m(k)$" (EQUATION 159), where "$y_m(k)$" is the RLS ID model output estimate in EQUATION 158:

$$x_m(k) = H_\theta^{-1} y_m(k)$$

EQUATION 159

6) Compute a corrected measurement matrix “$H_{\theta,cor}$” via EQUATION 160:

$$H_{\theta,cor} = \hat{y}(k) \cdot x_m(k)^{-1}$$

EQUATION 160

7) Compute a corrected model output “$y_{m,cor}(k)$” via EQUATION 161:

$$y_{m,cor}(k) = H_{\theta,cor} x_m(k)$$

EQUATION 161

8) Reconstruct the discrete state space system model with the corrected measurement matrix “$H_{\theta,cor}$”.

9) Transform the corrected system model from state space to transfer function.

10) Extract the corrected transfer function parameters and form a corrected “$\hat{\theta}_c(k)$”.

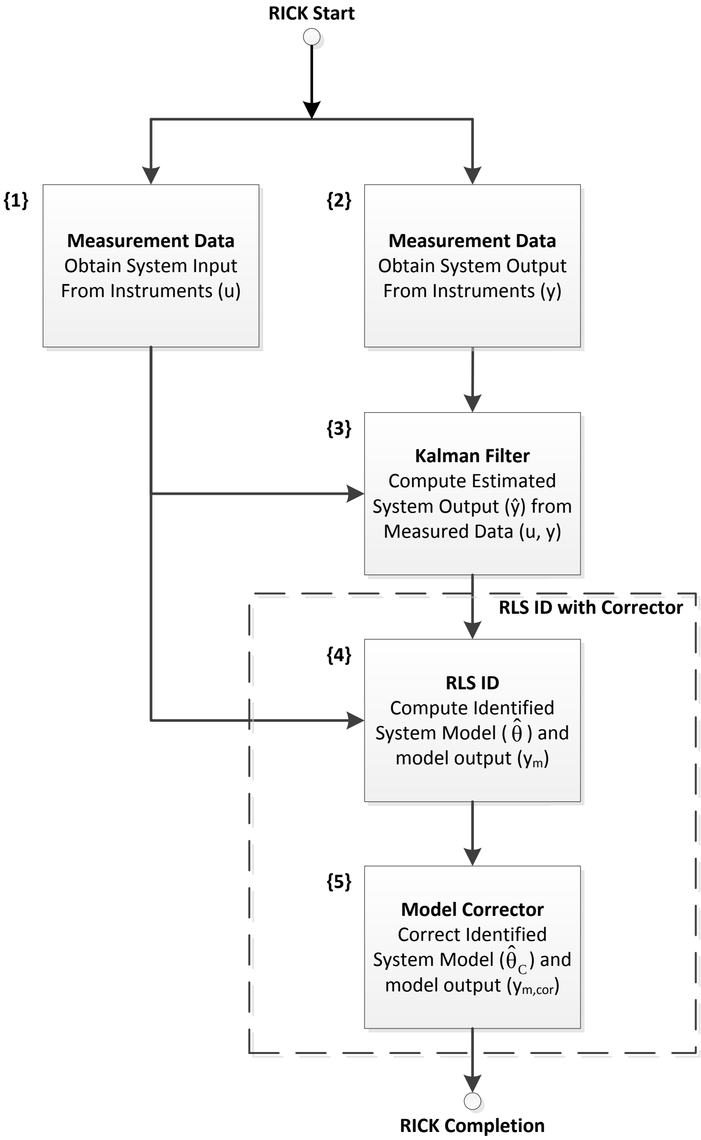


FIGURE 103: RICK process steps ([95] and [96])

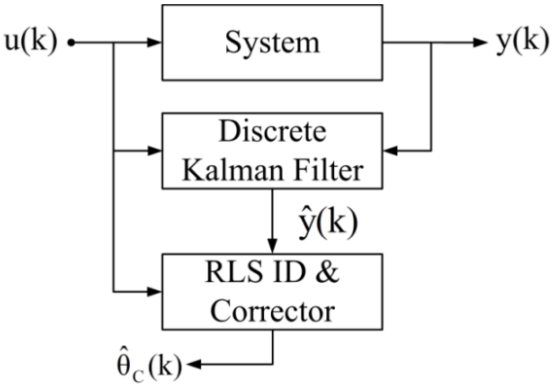


FIGURE 104: RICK overview block diagram ([95] and [96])

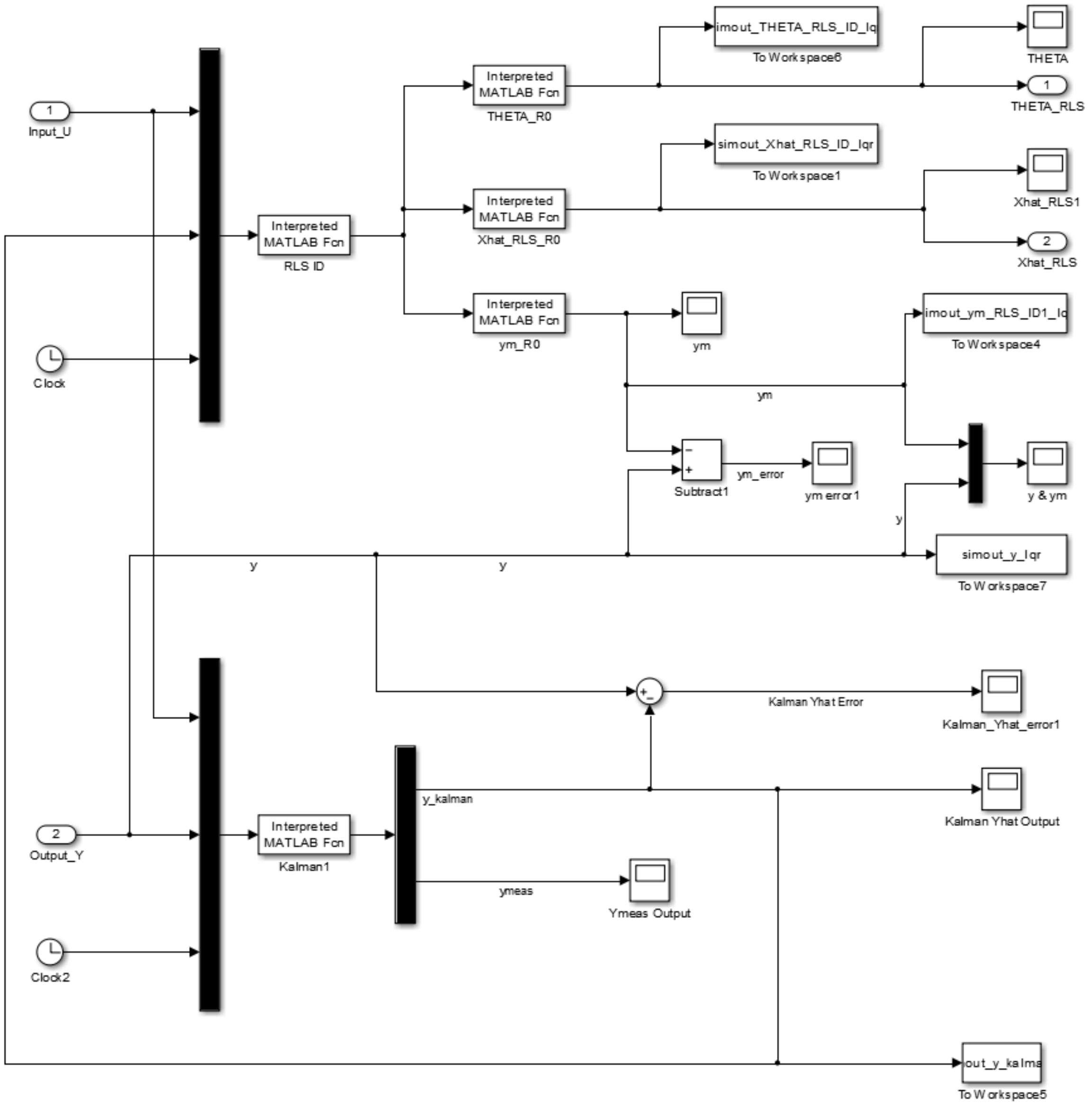


FIGURE 105: RICK implemented in Simulink

6.4.2 RLS ID Simulation Results

For comparison with RICK, the RLS ID method (see Section 5.3.5) was applied to the conventional control data provided in FIGURE 18 and FIGURE 30. The PZR simulation results using RLS ID are provided in FIGURE 106, FIGURE 107, and

FIGURE 108. TABLE 12 provides statistical performance measures. The RLS ID method was used to obtain an adaptive fourth order model of the PZR system, where $\dot{m}_{surge}$ was the input and the PZR pressure was the output. Typical noise terms were added to the data and process to simulate a physical system with measurement noise. The output performance of the online RLS ID model during transients is more accurate than the performance of the static system model, with the same model order. The RLS ID model error is approximately ten times smaller than that of the static model. ([95] and [96])

TABLE 12: RLS ID "y_m" statistical performance for PZR application [95], [96]

Error Max. (bar)	**Error Mean (bar)**	**Error Standard Deviation (bar)**
0.5967	0.0053	0.1537

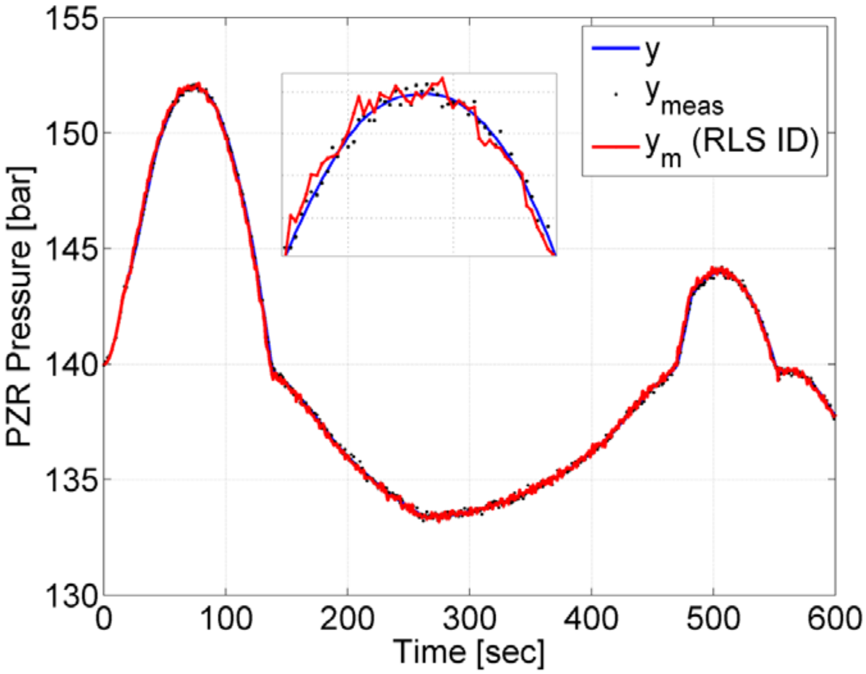


FIGURE 106: PZR pressure - actual, measured, and RLS ID ([95] and [96])

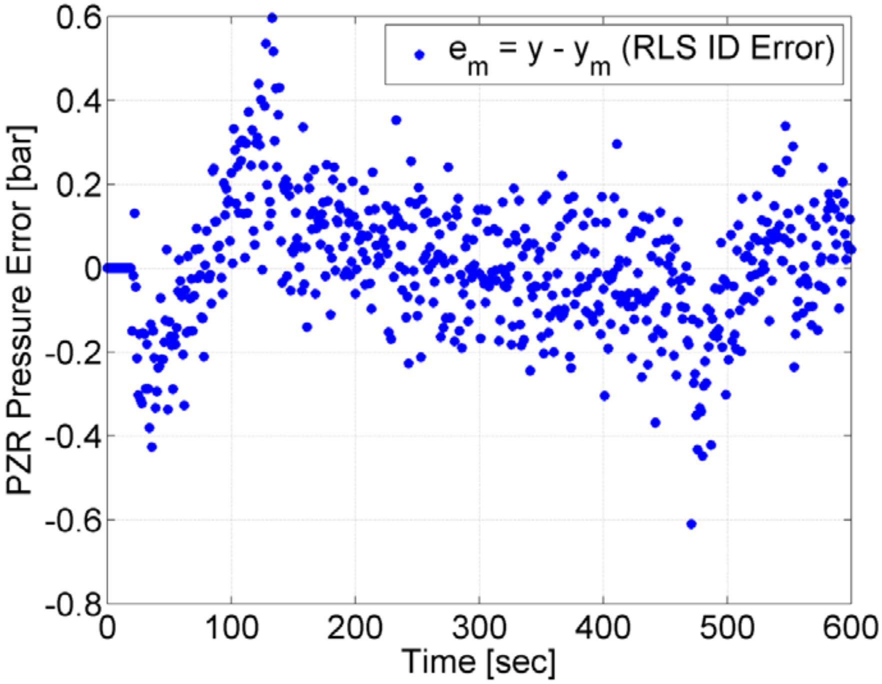


FIGURE 107: PZR pressure RLS ID error ([95] and [96])

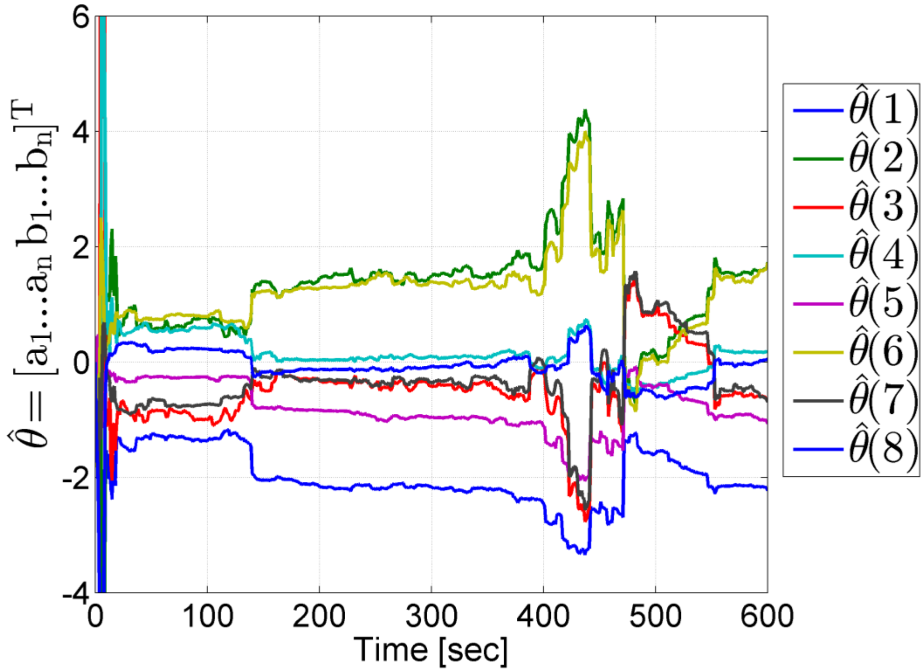


FIGURE 108: PZR pressure model coefficients via RLS ID ([95] and [96])

6.4.3 Kalman Filter Simulation Results

For comparison with RICK, the Kalman filter (see FIGURE 92) was applied to the conventional control data provided in FIGURE 18 and FIGURE 30. The PZR simulation results using the Kalman filter are provided in FIGURE 111 and FIGURE 112. TABLE 13 provides statistical performance measures for the Kalman filter and measured date. As with the RLS ID discussed in Section 6.4.2, typical noise terms were added to the data and process to simulate a physical system with measurement noise. The Kalman filter error is smaller than the measurement error, demonstrating that the Kalman filter estimates are more accurate than the measured values alone. As anticipated, the Kalman filter estimates exhibited a smaller maximum error, a smaller mean error, and a smaller error standard deviation than the measured values alone. ([96])

TABLE 13: Kalman filter statistical performance for PZR application [96]

	Error Max (bar)	**Error Mean (bar)**	**Error Standard Deviation (bar)**
y_{meas}	0.2673	-0.0087	0.0959
$\hat{y}$	0.1959	-0.0061	0.0754

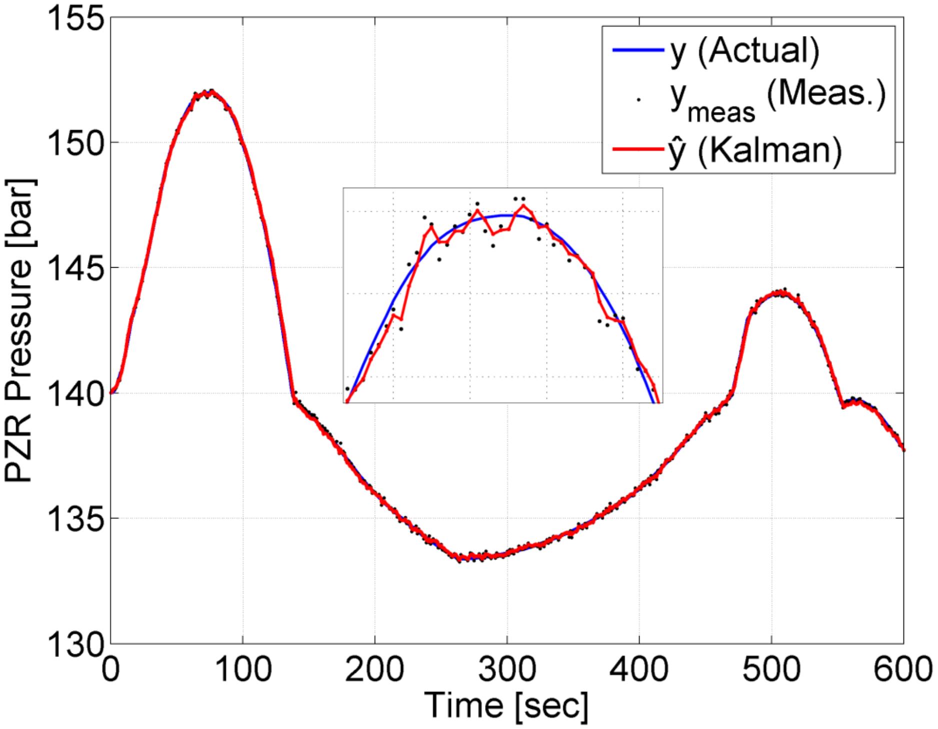


FIGURE 109: PZR pressure - actual, measured, and Kalman filter ([96])

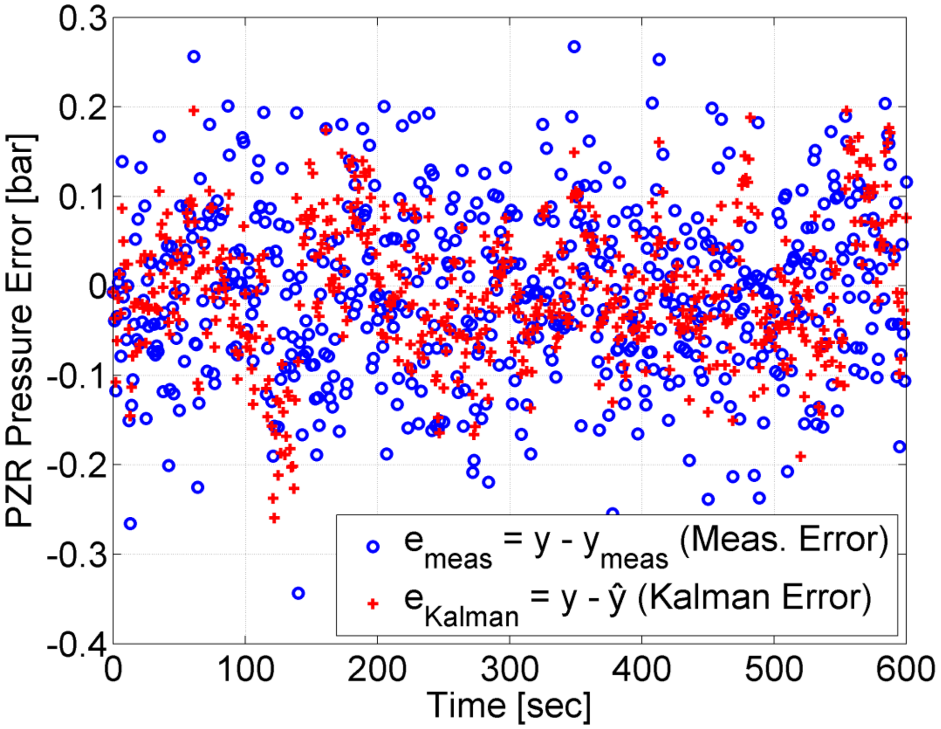


FIGURE 110: PZR pressure RLS ID error ([96])

6.4.4 RICK Simulation Results

The RICK method was applied to the conventional control data provided in FIGURE 18 and FIGURE 30. The PZR simulation results using RICK are provided in FIGURE 111, FIGURE 112, and FIGURE 113. TABLE 14 and TABLE 15 provide statistical performance measures. The RICK estimated model output "y_{RICK}" closely matches the actual process values for pressure, with less deviation from the true value than the measured data "y_{meas}", as seen in the zoomed inset of FIGURE 111. The magnitude of the RLS ID error was greater than that of the RICK model output error, as seen in FIGURE 112.

Spikes in the RLS ID bias appear systematic (e.g., around 40 seconds, 130 seconds, 480 seconds, and 550 seconds), as shown in FIGURE 112. However, the RICK bias does to exhibit these spikes. The RLS ID model output error is less normally distributed and centered about zero than the RICK model output error, as seen in FIGURE 112. Therefore, the RLS ID errors appear to be more systematic than the RICK errors. The RLS ID system model parameters exhibit greater deviation than those for the RICK, which would indicate that the RICK model has greater stability than that of the RLS ID.

A comparison of the percentage change in the output between (a) the Kalman filter pressure estimate and the measured pressure and (b) the RICK estimated model pressure and the RLS ID estimated model pressure is shown in TABLE 16. Compared to the measured output with noise "y_{meas}", the Kalman filter pressure estimate "$\hat{y}$" was able to achieve a 26.7% reduction in maximum error, a 29.9% reduction in mean error, and a 21.4% reduction in error standard deviation. Compared to the RLS ID estimated model pressure output "y_m", the RICK estimated model pressure output exhibited a 67.2%

reduction in the maximum error, a 20.8% increase in mean error, and a 50% reduction in error standard deviation. While, the mean error magnitude of "y_{RICK}" was slightly larger than the mean error magnitude of "y_m", the increase is not considered significant, since both mean error values are very close to zero.

In conclusion, the chief advantage of the newly proposed RICK method is that it provides improved knowledge of the PZR system (i.e., more accurate system knowledge), as compared to RLS ID method alone. However, there is opportunity for additional improvement, since the RICK error has not been eliminated (e.g., see FIGURE 112).

TABLE 14: RICK "y_{RICK}" statistical performance for PZR application [95], [96]

"y_{RICK}" Error Max. (bar)	**"y_{RICK}" Error Mean (bar)**	**"y_{RICK}" Error Standard Deviation (bar)**
0.1959	-0.0064	0.0760

TABLE 15: RICK statistical performance comparison [96]

	Error Max. (bar)	**% "y_{meas}" Error Max.**	**Error Mean (bar)**	**% "$\|y_{meas}\|$" Error Mean**	**Error Standard Deviation (bar)**	**% "y_{meas}" Error Std. Dev.**
y_{meas}	0.2673	100%	-0.0087	100%	0.0959	100%
y_m	0.5967	223%	0.0053	61%	0.1537	160%
$\hat{y}$	0.1959	73%	-0.0061	70%	0.0754	79%
y_{RICK}	0.1959	73%	-0.0064	74%	0.0760	79%

TABLE 16: RICK percentage difference comparison [96]

A % of B via (A-B)/B	**Error Max.**	**Error |Mean|**	**Error Std. Dev.**
Error $\hat{y}$ % of Error y_{meas}	-26.7%	-29.9%	-21.4%
Error y_{RICK} % of Error y_m	-67.2%	20.8%	-50.6%

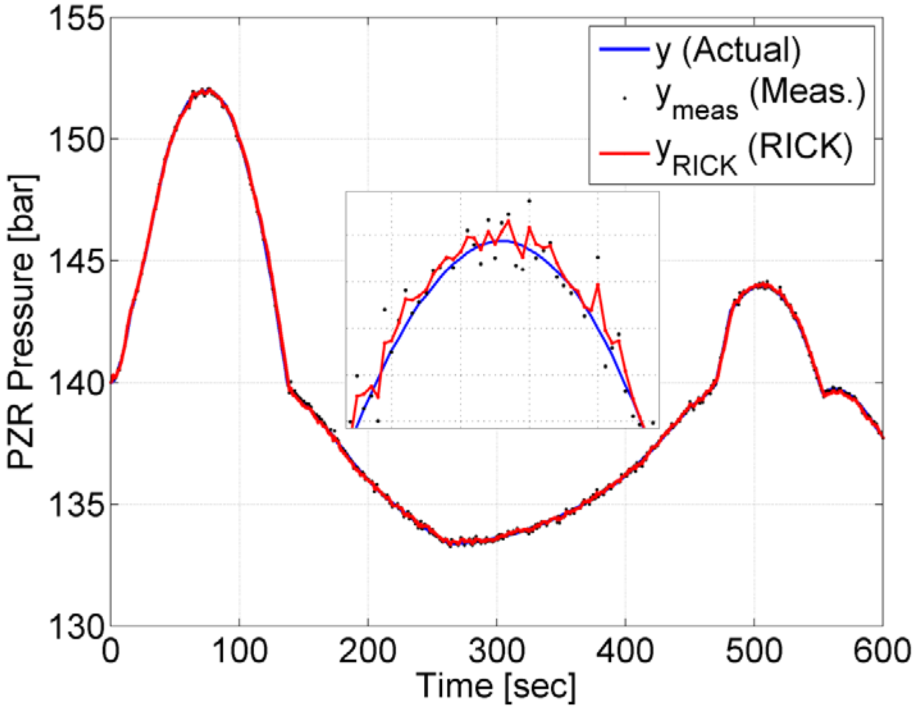


FIGURE 111: PZR pressure - RICK ([95] and [96])

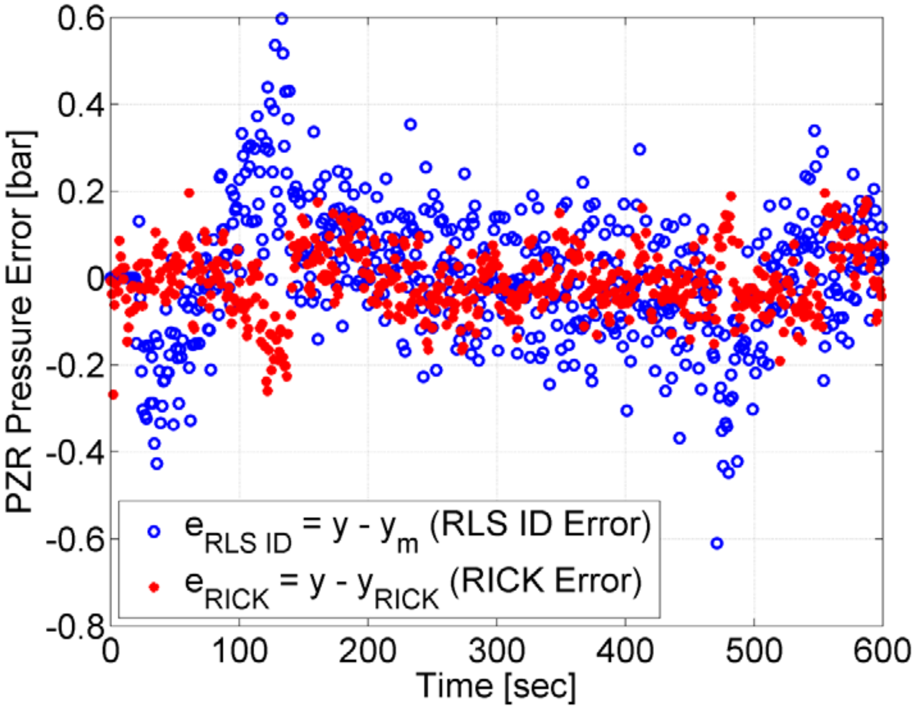


FIGURE 112: PZR pressure error – (1) RLS ID and (2) RICK ([95] and [96])

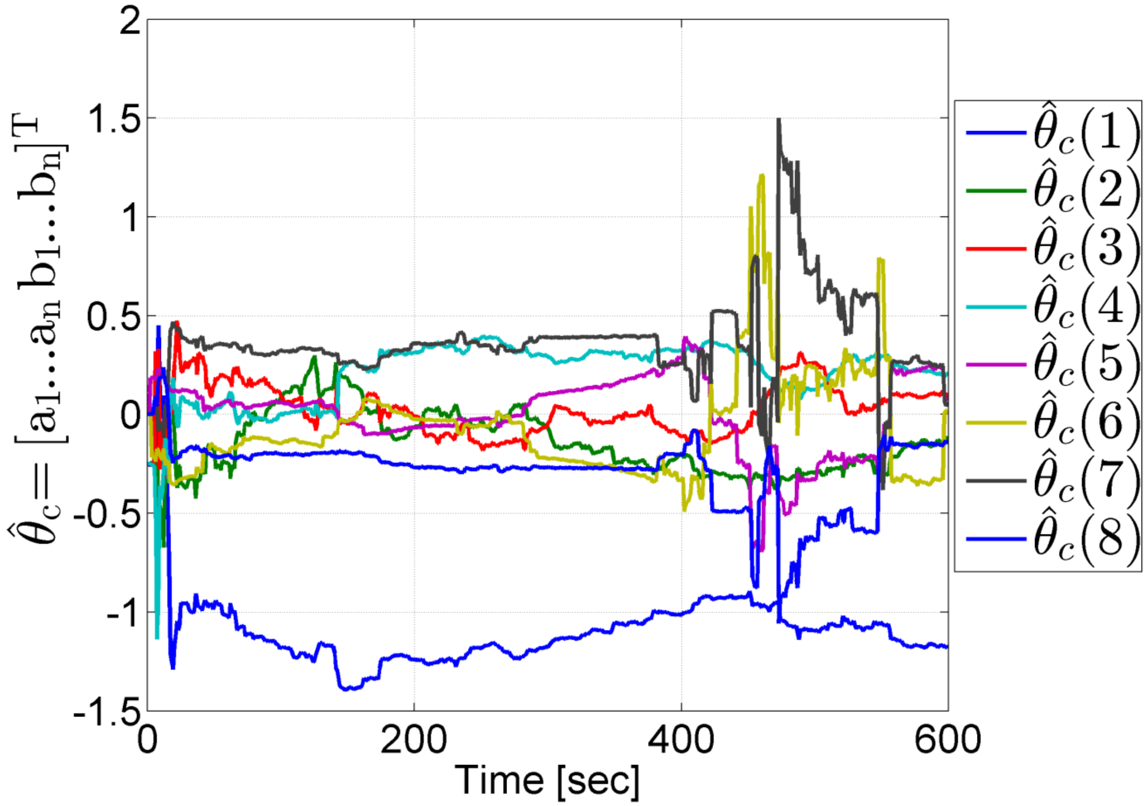


FIGURE 113: PZR pressure model coefficients via RICK ([95] and [96])

6.5 LiMeRICK Control (Enhanced Control Method)

The information and results in this section is adapted from M. Smith and S. Kamalasadan [95]. A method (i.e., called LiMeRICK) is proposed “that combines a hybrid control technique (i.e., called LiMe, see Section 6.3) with a technique to improve system knowledge (i.e., called RICK, see Section 6.4)” [95]. FIGURE 114 graphically identifies the LiMeRICK focus area in a typical block diagram of a feedback control system. “Combining these two methods (i.e., LiMe and RICK) enables enhanced control during transient conditions. The proposed control architecture (LiMeRICK) that combines LiMe and RICK is presented. Results from simulations are presented that demonstrate improved pressure control is achieved with this method.” [95]

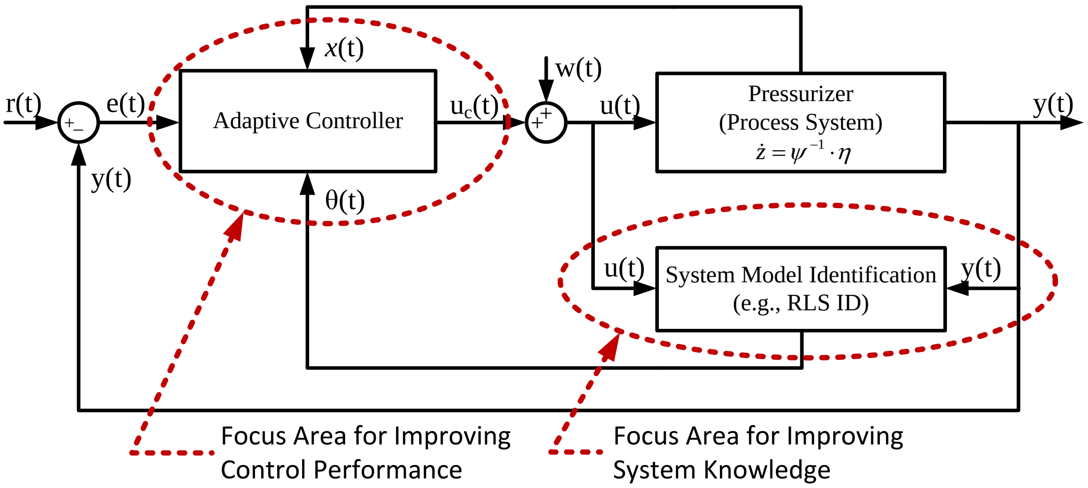


FIGURE 114: LiMeRICK focus for adaptive PZR pressure control [95]

6.5.1 LiMeRICK Method

To improve system knowledge and control of the observed system, LiMeRICK (LiMe with RICK) combines the LiMe method (see Section 6.3) with the RICK method (see Section 6.4), as shown in FIGURE 116. The goal of the LiMeRICK method is to achieve a more accurate controller for pressure control in a pulsed cooling system. This is achieved by (1) using a combination of advanced control strategies (i.e., LiMe is a hybrid controller consisting of both LQR and MV control) and (2) using digital signal processing techniques (e.g., RLS ID and Kalman filter) to obtain more accurate system information (i.e., the RICK method improves model accuracy and system knowledge). Both LiMe and RICK are computationally capable of being implemented online with existing technology, so the proposed architecture combining these techniques is realistically achievable. The process steps for implementing LiMeRICK are described sequentially in FIGURE 117 and detailed in the list below.

Implementation of the proposed LiMeRICK method involves RICK producing an RLS ID system model with correction ($\hat{\theta}_C$) and the estimated system states and output via the Kalman filter (e.g., $\hat{y}$), which are used as inputs the LiMe control method (i.e., system feedback and adaptive system model). Then, the proposed LiMe method computes the optimum "u" value by weighting the LQR and MV control input values in accordance with their sensitivity to the system output "y" (i.e., PZR pressure) via a LiMe weighting sensitivity factor "α". The following method (see list below, FIGURE 115, and FIGURE 117) can be used to implement the LiMeRICK technique for the PZR:

1) Acquire the measured input (u) and output (y) data of the system from instruments
2) Using the RICK method (see Section 6.4), compute the RLS ID model output with correction ($\hat{\theta}_C$) and the estimated system states and output via the Kalman filter (e.g., $\hat{y}$)
3) Using the RICK model output with correction ($\hat{\theta}_C$) and Kalman filter estimates (e.g., $\hat{y}$) as inputs the LiMe controller, compute "$u_{LiMeRICK}$" via the LiMe control law (see Section 6.3).

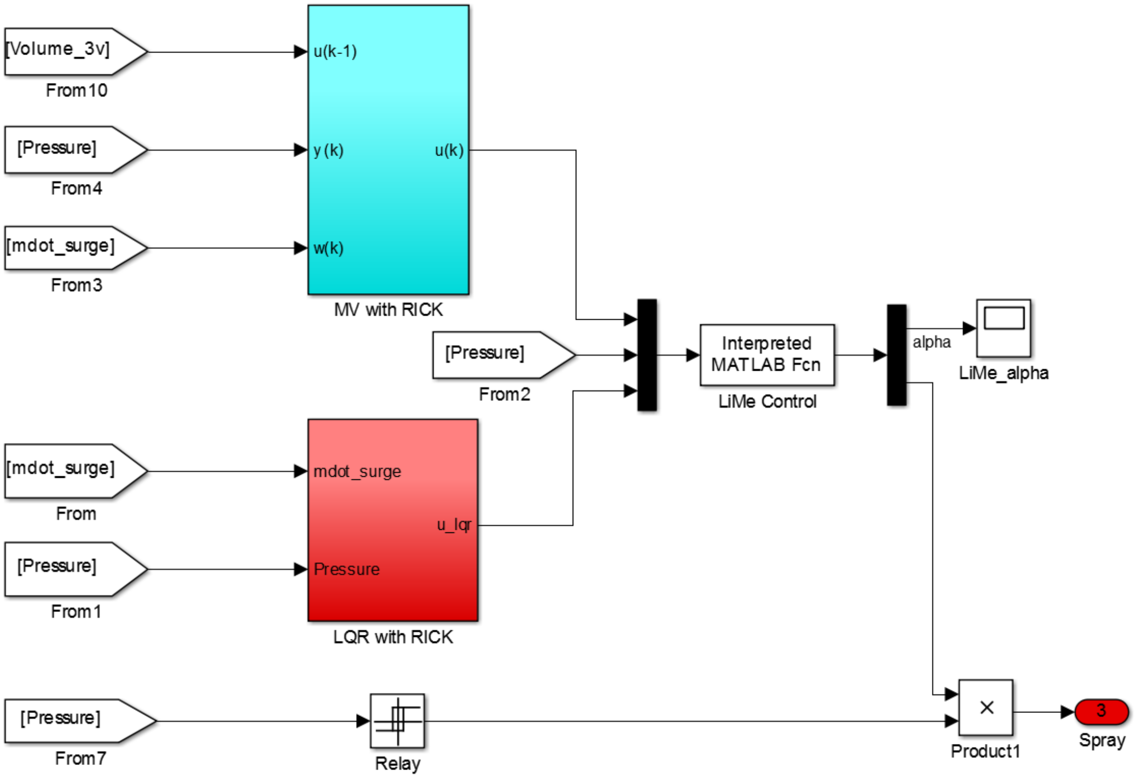


FIGURE 115: LiMeRICK implemented in Simulink

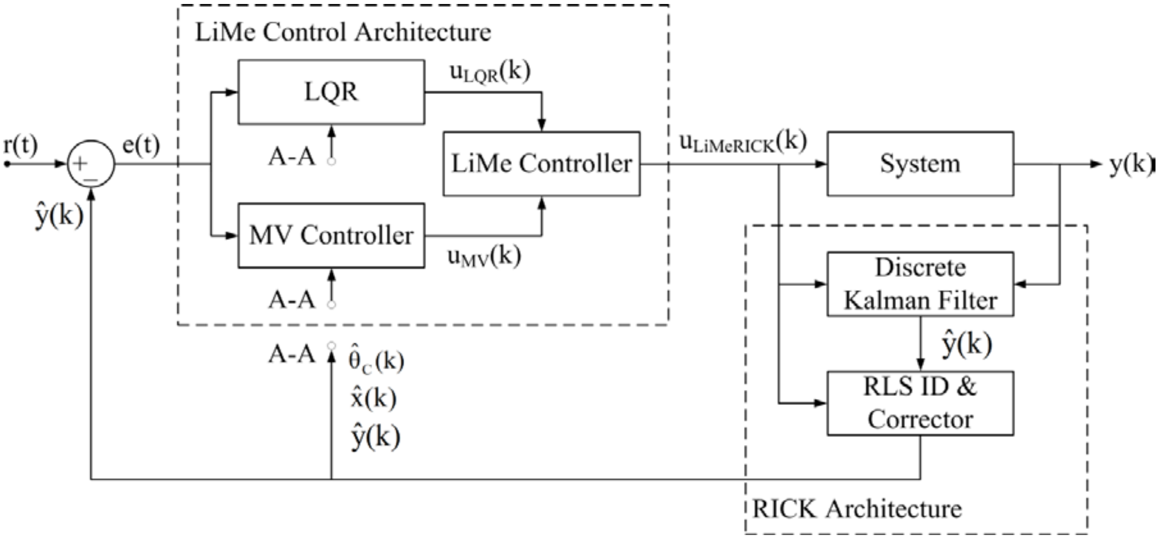


FIGURE 116: LiMeRICK overview block diagram [95]

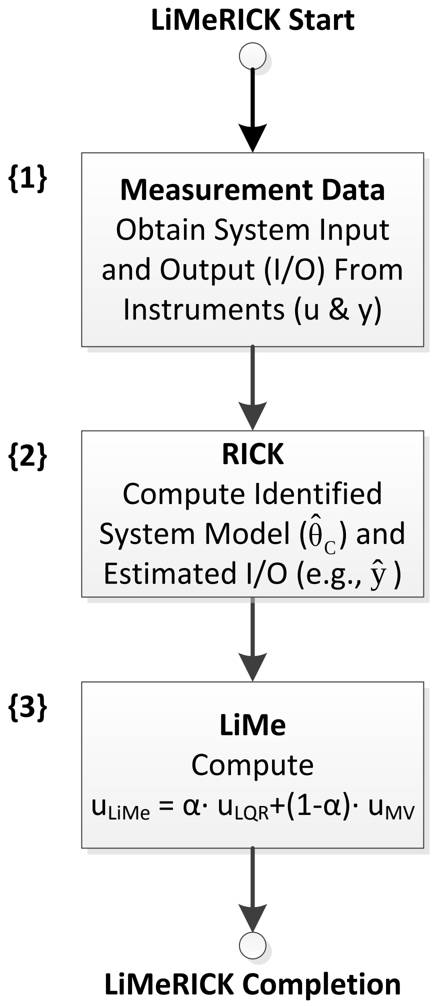

LiMeRICK Start
{1}
Measurement Data
Obtain System Input and Output (I/O) From Instruments (u & y)
{2}
RICK
Compute Identified System Model ($\hat{\theta}_C$) and Estimated I/O (e.g., $\hat{y}$)
{3}
LiMe
Compute
$u_{LiMe} = \alpha \cdot u_{LQR} + (1-\alpha) \cdot u_{MV}$
LiMeRICK Completion

FIGURE 117: LiMeRICK process steps [95]

6.5.2 LiMeRICK Simulation Results

Simulation results for the PZR using the LiMeRICK control method are provided in FIGURE 118, FIGURE 119, FIGURE 120, and FIGURE 121. Statistical performance measures are provided in TABLE 17 that compares the peak PZR pressure and time required to reach the peak pressure using conventional control, LiMe, LQR with RICK, MV with RICK, and LiMeRICK. As seen in FIGURE 118, the LiMeRICK method achieved a lower peak pressure and faster response than that of the LiMe controller, which illustrates the importance of having accurate system knowledge in achieving improved control performance. A PZR pressure plot for LiMeRICK, LQR with RICK, and MV control with RICK is provided in FIGURE 120.

The LiMeRICK method commanded a slightly greater control input that that of the LiMe, but the general shape of the control input was approximately the same for LiMe and LiMeRICK. However, as seen in FIGURE 121, the LiMeRICK control input was similar in magnitude to that of the LQR with RICK and MV with RICK. The LQR with RICK control input exhibited some fluttering (e.g., see FIGURE 121 around 85 to 120 seconds) that could prematurely wear the PZR spray control valve and would probably not be achievable considering typical response times required for actuation of such a valve. However, this fluttering was not present in LiMeRICK control input.

For comparison, additional simulation data plots with the LiMeRICK, LiMe, LQR with RICK, and MV control with RICK are provided in FIGURE 122 and FIGURE 123, which show the PZR pressure response based on mass surge flow rates that are different than the one provided in FIGURE 18. FIGURE 122 plots the pressure response for a single large PZR insurge. FIGURE 123 plots the pressure response for a PZR insurge followed immediately by a PZR outsurge. The pressure response trends in FIGURE 122 and FIGURE 123 are consistent with the trends shown in FIGURE 120 (e.g., the LiMeRICK control method exhibited superior performance characteristics in comparison with the LQR and MV control methods). For the single insurge (FIGURE 122), the LiMe exhibited the greatest pressure spike (i.e., 150.2 bar, 167 seconds), the MV with RICK maximum pressure (i.e., 149.2 bar, 163 seconds) was less than the LQR with RICK maximum pressure (149.6 bar, 166 seconds), but both of these were notably larger than the LiMeRICK maximum pressure (i.e., 148.7 bar, 165 seconds). For the insurge followed immediately by an outsurge (FIGURE 123), the MV control with RICK exhibited the greatest pressure spike (i.e., 145.8 bar, 64 seconds), the LQR with RICK maximum pressure (i.e., 145.1 bar, 68 seconds) was less than the MV control with RICK or LiMe (145.4 bar, 64 seconds), but all of these were larger than the LiMeRICK maximum pressure (i.e., 144.8 bar, 63 seconds).

Based on the simulation plots and statistical performance measures, the pressure is more predictable and controllable using the LiMeRICK control method, as compared with the LQR with RICK, the MV control with RICK, the LiMe, or the conventional control method. The LiMeRICK method produced the best overall performance with the smallest maximum peak pressure and fastest response, which illustrates the benefit of combining improved control (LiMe) and system knowledge techniques (RICK).

TABLE 17: LiMeRICK statistical performance measures for PZR application [95]

	Conv.	**LiMe**	**LQR with RICK**	**MV with RICK**	**LiMeRICK**
Max Pressure (bar)	151.98	145.76	145.34	145.14	145.01
Max Pressure above Reference Pressure (bar)	11.98	5.76	5.34	5.14	5.01
Pressure % above Reference Pressure	8.56%	4.11%	3.81%	3.67%	3.58%
Time for Max Pressure(sec)	75	52	47	45	45
Time % of Conventional Peak Time	0	-31%	-37%	-40%	-40%

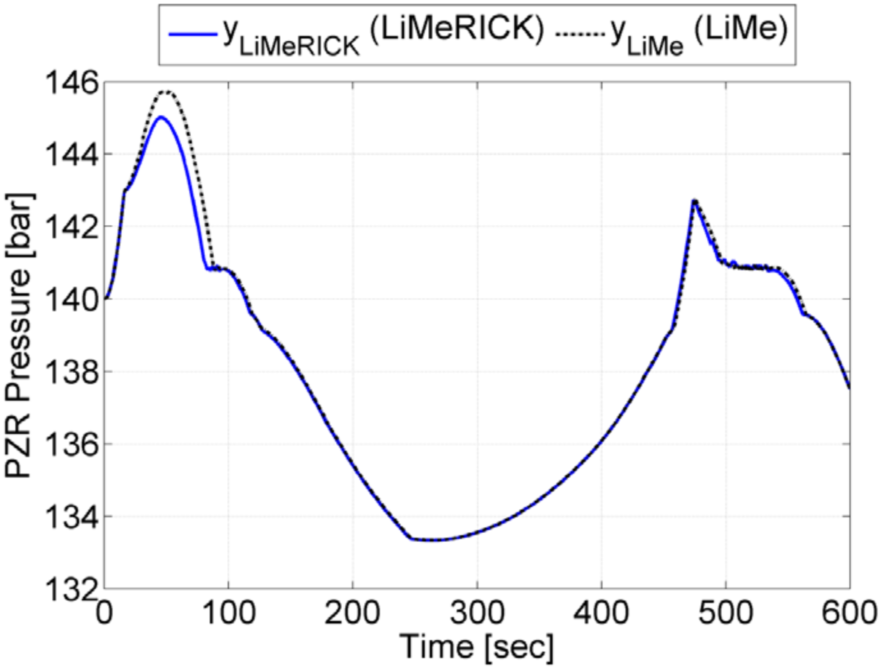


FIGURE 118: PZR pressure comparison for LiMeRICK and LiMe [95]

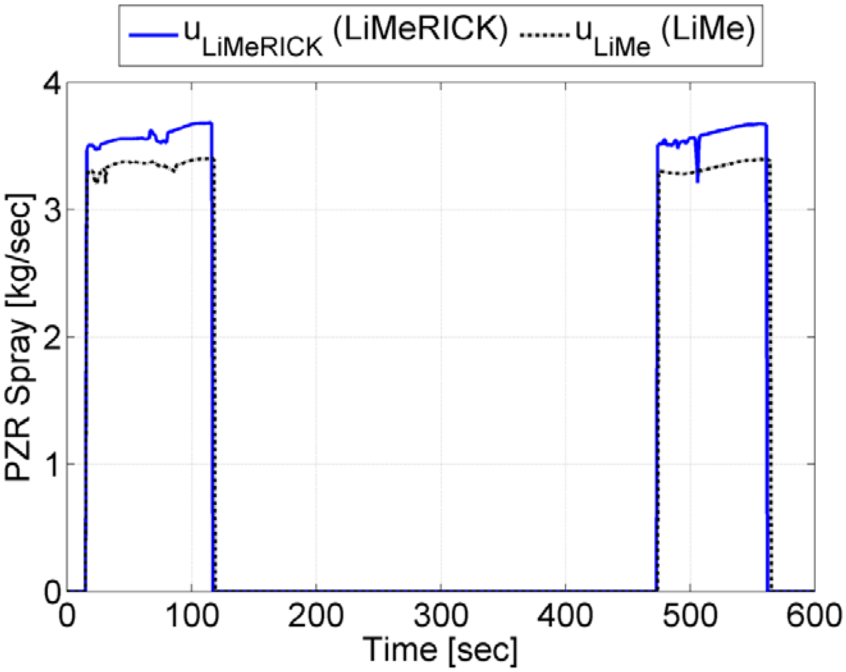


FIGURE 119: PZR spray comparison for LiMeRICK and LiMe [95]

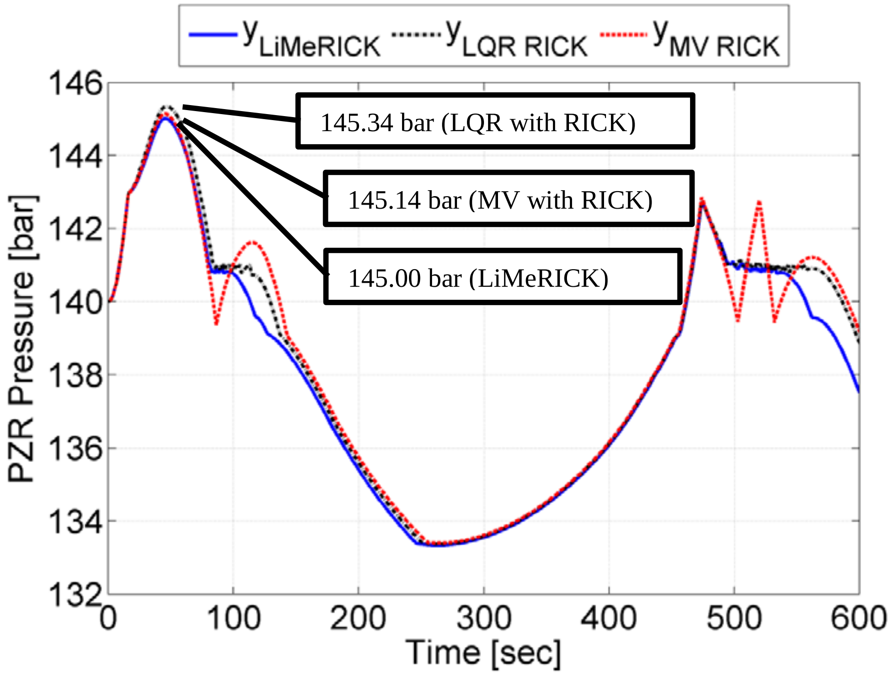


FIGURE 120: PZR pressure for LiMeRICK, LQR RICK, and MV RICK [95]

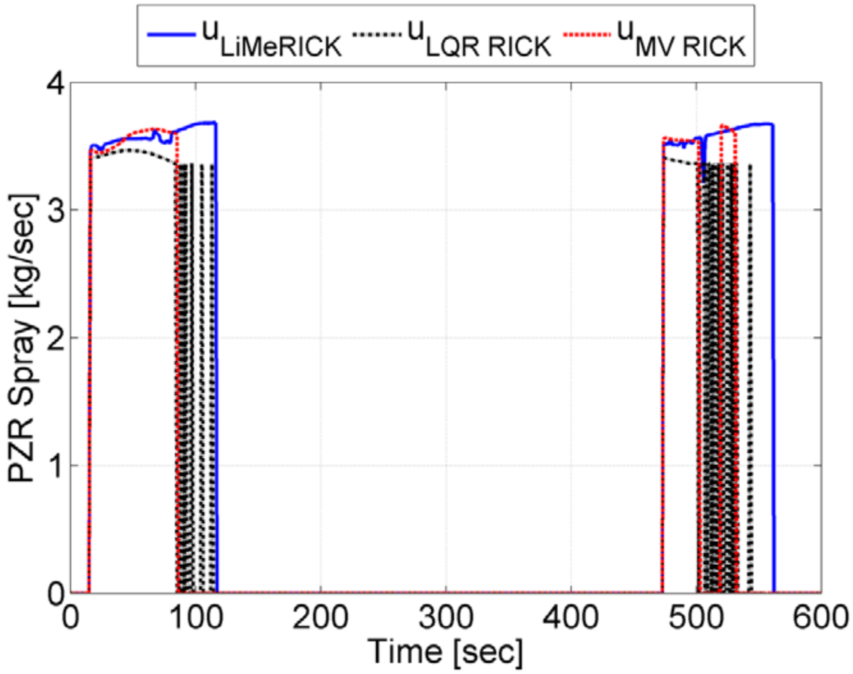


FIGURE 121: PZR spray for LiMeRICK, LQR RICK, and MV RICK [95]

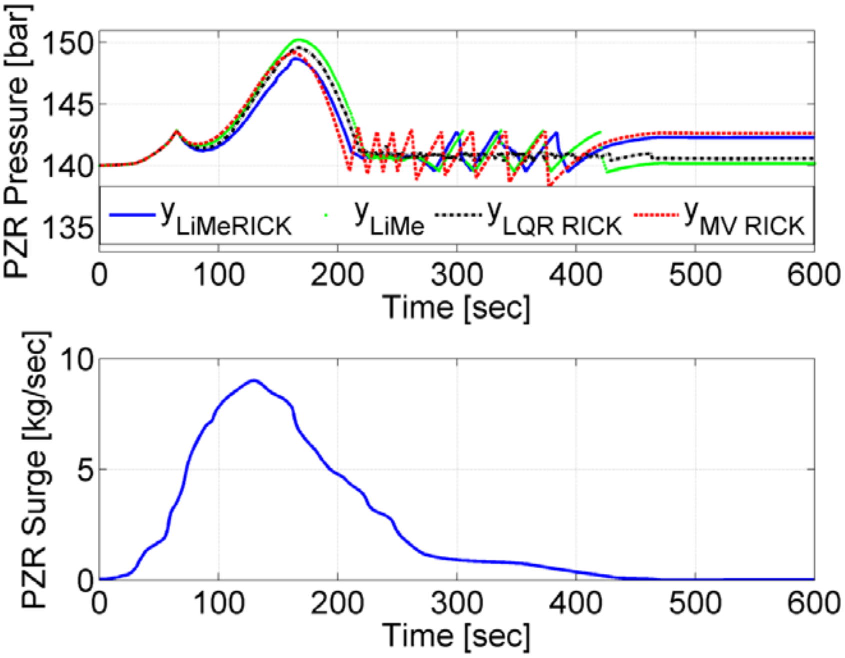


FIGURE 122: PZR input ($\dot{m}_{surge}$) and output (P) data with LiMeRICK, LiMe, LQR with RICK, and MV control with RICK – single insurge [95]

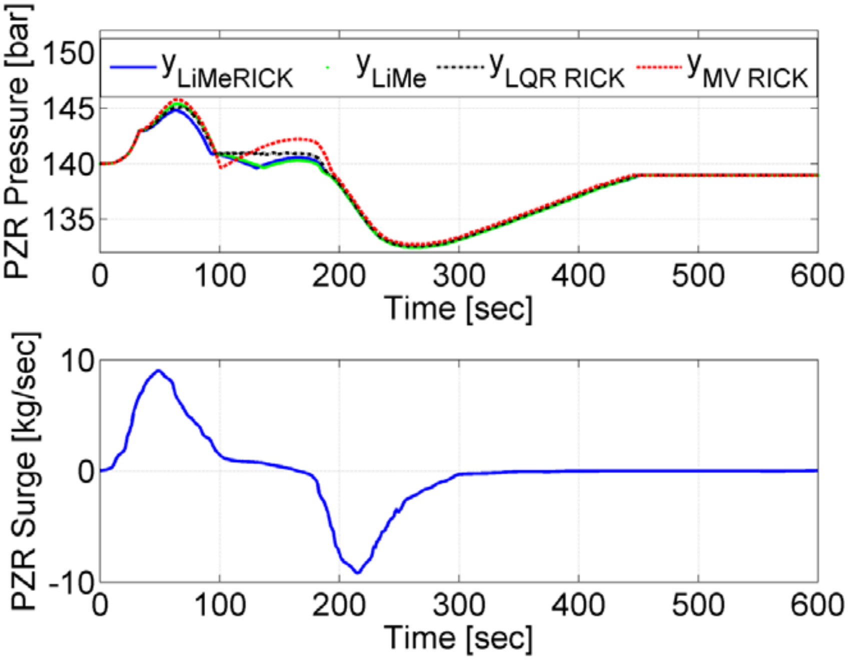


FIGURE 123: PZR input ($\dot{m}_{surge}$) and output (P) data with LiMeRICK, LiMe, LQR with RICK, and MV control with RICK – insurge/outsurge [95]

6.6 LiMeRICK Control with Predictive Horizon

In this section the LiMeRICK method (see Section 6.5) is expanded to include control based on a predictive horizon, rather than acting on the current time step information. The LiMe controller uses predicted values for the states and associated model of the system, so that it is acting on the predicted characteristic of the system “n” time steps ahead. Section 6.6.1 describes the LiMeRICK Predictive Horizon Method. Section 6.6.2 provides simulation results with comparison to LiMeRICK and LiMe.

6.6.1 LiMeRICK Predictive Horizon Method

In a dynamic system the states and model of the system can change rapidly. If the controller cannot respond to those changes fast enough, reduced control performance occurs. Methods have been presented to enhance control performance via hybrid control (i.e., LiMe, see Section 6.3) and improve system knowledge (i.e., RICK, see Section 6.4). If future values of the system states can be obtained, it is possible to achieve improved control performance by the controller acting preemptively in response to an anticipated behavior in the system (e.g., turn on the PZR spray sooner to reduce the pressure spike from a PZR insurge), rather than waiting until the event has already occurred (i.e., proactive control rather than reactive control).

Predictive estimators provide one method to look into the future. The standard Kalman filter (see FIGURE 92 in Section 6.1) uses “Predictor” and “Corrector” steps. The “Predictor” step typically estimates the states one time step ahead (i.e., k+1, where k is the current time step). However, it is possible for the predictor to estimate the states “n” time steps ahead. These estimates provides sufficient accuracy provided that the system model has not changed significantly in the future from the currently identified model of the system (i.e., the system is still reasonably linear for the predictive horizon). In general, the uncertainty associated with the predicted state estimates increases as the size of “n” increases, since it is not known if the system will continue linearly. Understanding these limitations is necessary to effectively use this technique. While an infinite horizon predictor would be useful for control, intuitively it is not realistically achievable, because of these limitations. However, for a finite predictive horizon this approach can be effective in predicting future values to achieve improved control performance.

The proposed method modifies the RICK method by using the predictive horizon output of the Kalman filter “n” steps ahead as input to the RLS ID. Therefore the RLS ID produces an identified model of the system that is anticipated “n” steps ahead (i.e., it predicts what the model will be “n” time step ahead). The predicted system model and predicted states are then used as input to the LiMe controller, so the controller is currently acting on the predicted system that “n” steps ahead, which enables faster response to transients. The predicted output of the Kalman filter assumes that the system model (i.e., A, B, C, and D state space matrices) are still valid “n” steps ahead (i.e., the system is sufficiently linear to produce the desired control performance) and the system input can be treated as constant for “n” steps (since the true value is not known). Additionally, to help smooth the predicted value, the root mean square (RMS) value of the output estimate is used (i.e., this helps reduce the impact of changing system dynamic on the predicted values). Listed below (and in FIGURE 124) are the steps used to implement the LiMeRICK Predictive Horizon method. Also see the MATLAB function file provided in APPENDIX B for the modified Kalman filter with predictive horizon (i.e., "PZR_Kalman_Function_MPC_R2_lqr.m"). The rest of the controller behaves the same as with the LiMeRICK method, with the exception that the input to the RLS ID and LiMe is the predicted values for the states and system model.

Modified LiMeRICK method steps (Kalman filter algorithm) for predictive horizon output that is "n" steps ahead:

1) Kalman Filter Predictor step (i.e., modify the steps in FIGURE 92)

    a) $\bar{x}(k+1) = A\hat{x}(k) + Bu(k)$

    b) $\bar{P}\left(k+1\right) = AP(k)A^{T} + Q$

    c) Estimate the states "n" steps ahead (use $\bar{y}(k+n)$ as input to RLS ID)

    For m = 1:n

    $\bar{x}(k+m) = A\hat{x}(k+m-1) + Bu(k)$

    $\bar{y}(k+m) = C\bar{x}(k+m)$

    End

$$\bar{y}_{RMS}(k+n) = \sqrt{\frac{\bar{y}(k+1)^2 + \bar{y}(k+2)^2 + ... + \bar{y}(k+n)^2}{n}}$$

2) Kalman Filter Correct (same steps provided in FIGURE 92)
3) Use predicted output value "$\bar{y}_{RMS}(k+n)$" as input to the RLS ID, which will produce a predicted model of the system at "k+n".
4) Use the predicted values as input to the LiMe controller to predict control needed at the "k+n" time step.

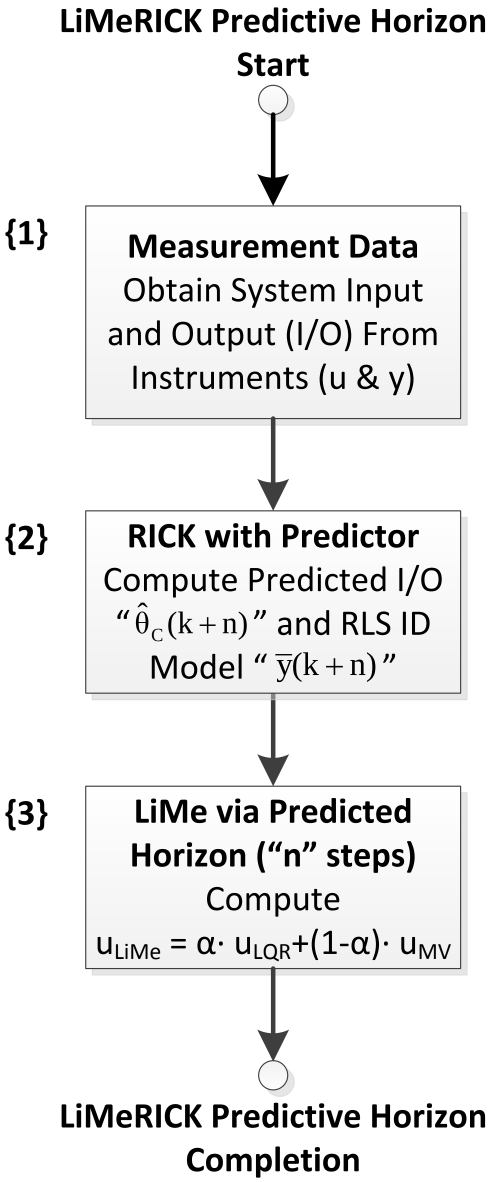

LiMeRICK Predictive Horizon
Start
{1}
Measurement Data
Obtain System Input and Output (I/O) From Instruments (u & y)
{2}
RICK with Predictor
Compute Predicted I/O "$\hat{\theta}_C(k+n)$" and RLS ID Model "$\overline{y}(k+n)$"
{3}
LiMe via Predicted Horizon ("n" steps)
Compute
$u_{LiMe} = \alpha \cdot u_{LQR} + (1-\alpha) \cdot u_{MV}$
LiMeRICK Predictive Horizon
Completion

FIGURE 124: LiMeRICK predictive horizon (P.H.) process steps

6.6.2 LiMeRICK Predictive Horizon Simulation Results

Simulation results for the PZR using the LiMeRICK Predictive Horizon control method are provided in FIGURE 125 and FIGURE 126. Statistical performance measures are provided in TABLE 18 that compares the peak PZR pressure and time required to reach the peak pressure using LiMe, LiMeRICK, and LiMeRICK with Predictive

Horizon. As seen in FIGURE 125, the LiMeRICK Predictive Horizon method achieved a lower peak pressure than that of the LiMe or LiMeRICK controllers, which illustrates the benefit of predictive control. The response time of the LiMeRICK Predictive Horizon was 46 seconds, which is faster than the LiMe by six seconds, but slower than the LiMeRICK by one second. On average the LiMeRICK with Predictive Horizon produced an output pressure that was 0.27 bar less than that achieved by the LiMeRICK. The PZR spray turned on at 16 seconds for the LiMeRICK method and 14 seconds for the LiMeRICK method with Predictive Horizon, which demonstrates how the predictor enables the system to respond sooner to an insurge. However, the LiMeRICK with Predictive Horizon did not perform as well as the LiMe or LiMeRICK during the transient occurring between 450 seconds and 600 seconds. During this time period, the spray switched off and on multiple times (as seen in FIGURE 126), which resulted in undesirable spikes in the pressure. The magnitude of the LiMeRICK Predictive horizon spray for this time period was also greater than that commanded by the LiMe or LiMeRICK. This behavior is resembles that observed with the MV control (see FIGURE 88 in Section 5.5), which would indicate that the MV control is undesirably driving the system during this period.

TABLE 18: LiMeRICK predictive horizon statistical performance measures

	LiMe	**LiMeRICK**	**LiMeRICK with Predictive Horizon**
Max P (bar)	145.76	145.01	144.52
Time (sec)	52	45	46

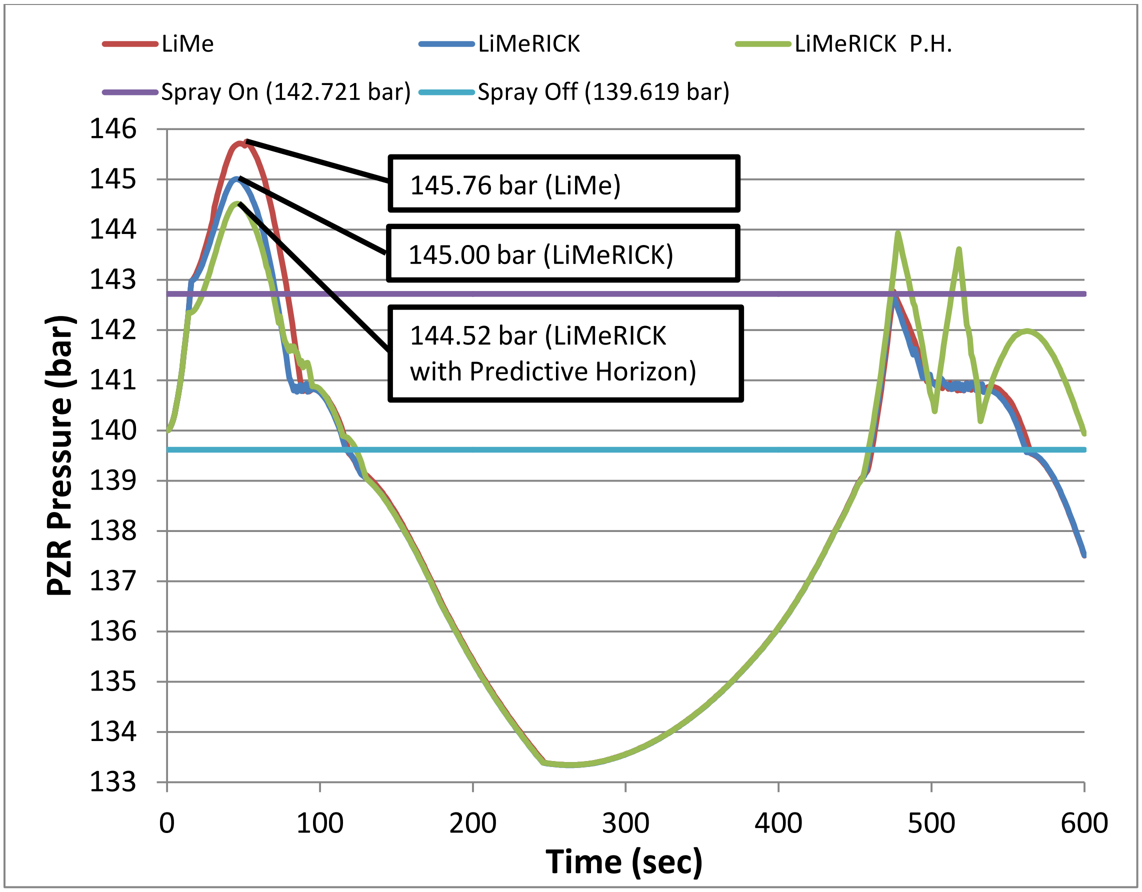


FIGURE 125: PZR pressure comparison for LiMe, LiMeRICK, & LiMeRICK P.H.

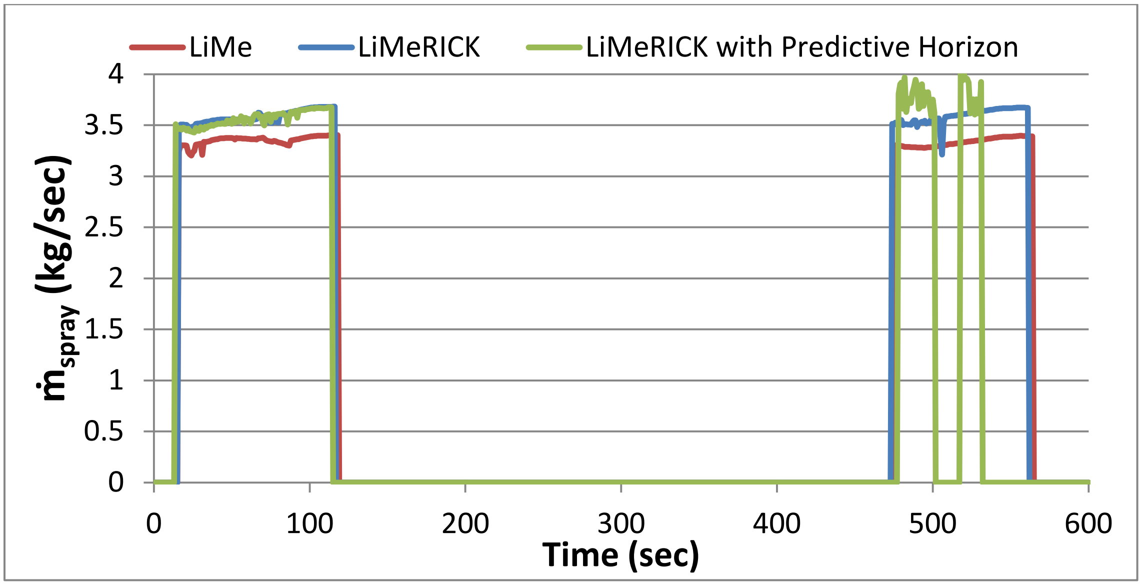


FIGURE 126: PZR spray comparison for LiMe, LiMeRICK, & LiMeRICK P.H.

CHAPTER 7: CONCLUSION AND FUTURE RESEARCH

The challenges of pressurizer (PZR) pressure control in a pulsed cooling system (e.g., cooling system of a tokamak fusion reactor) were introduced in CHAPTER 1. To simulate the thermal-hydraulic pressurizer behavior of a pulsed cooling system, two reduced order PZR models (SISO and MISO) and a dynamic PZR model were presented (see CHAPTER 2). Several control methods (e.g., conventional, digital, optimal, and adaptive) were applied to PZR pressure control using both reduced order and dynamic PZR models (see CHAPTER 3, CHAPTER 4, CHAPTER 5). Generic cooling system and pressurizer values were selected to provide a general solution to this type of pressurizer control problem (see Section 2.2). Therefore, the control methodology presented could be applied and adapted to project specific pressurizer control designs involving pulsed cooling systems. For the reduced order models, the following control methods were applied:

1) Indirect Digital Control (Section 4.1)
2) Direct Digital Control (Section 4.2)
3) Pole Placement Control (Section 4.3)
4) Optimal Control (Section 4.4)

Optimal control with state estimation produced the best performance with the reduced order model. Therefore, it was selected as the initial control method to implement on the dynamic PZR model. For control of the dynamic PZR model exhibiting pulsed cooling system behavior, several control methods were investigated:

1) Baseline of System Performance with (open loop) No Control (Section 5.1)
2) Conventional Control (Section 5.2)
3) Optimal Control of PZR Spray with Constant Gain “K” (Section 5.3.1)
4) Adaptive Optimal Control of PZR Spray via a New State Space “B” matrix computed at Each Time Step (Section 5.3.2)
5) Adaptive Optimal Control of PZR Spray and Heater via a New State Space “B” matrix computed at Each Time Step (Section 5.3.3)
6) Adaptive Optimal Control of PZR Spray via Recursive Least Squares Identification (RLS ID) based on the PZR system input and output (Section 5.3.5)
7) Adaptive STR PID Control of PZR Spray via RLS ID (Section 5.4)
8) MV Control of PZR Spray (Section 5.5)

The results of these control simulations were analyzed for applicability and compared for performance to identify the most appropriate control methodology for this application. The adaptive optimal control of PZR spray via RLS ID and MV control of PZR spray exhibited the best overall pressure control performance of the methods considered.

Finally, based on the system behavior and controller performance of these advanced methods, a new control methodology was presented (see CHAPTER 6) that focused on improving the control performance (e.g., minimize response time and overshoot) for pressure control of a pulsed cooling water system. This improvement focused on two aspects: (1) implementing an adaptive dynamic controller that can respond to the transient behavior of the pulsed PZR system (i.e., LiMe, see Section 6.3) and (2) implementing an estimator/corrector to the RLS ID to reduce system identification errors for improved system knowledge (i.e., RICK, see Section 6.4). These two improvement aspects (i.e., LiMe and RICK) were coupled to produce the LiMeRICK control architecture for enhanced control performance (i.e., see Section 6.5). The LiMeRICK control method exceeded the target reduction in maximum pressures and offers noted potential for the application of PZR pressure control in a pulsed system.

The contributions of this research (as discussed in Section 1.7) include:

1) PZR models identified (SISO and MIMO) for adaptive control simulation (Sections 2.1-2.5)
2) PZR dynamic simulation disturbance inputs for mass surge flow rate (Section 2.2)
3) Adaptive-optimal control techniques to dynamically update the LQR "K" matrix via:
    a. A new state space "B" matrix at each time step (Sections 5.3.1-5.3.4)
    b. RLS ID to make a new system model at each time step (Section 5.3.5)
4) An adaptive-dynamic control approach called LiMeRICK, where LiMe is a hybrid control technique that uniquely combines LQR and MV control (Sections 6.3, 6.5, and 6.6)
5) A method to improve system knowledge called RICK that uniquely combines an RLS ID, a state space based model corrector, and a Kalman filer (Sections 6.4, 6.5, and 6.6)

7.1 Future Research

Some potential topics for future research related to PZR pressure control in a pulsed cooling system are introduced in this section, but are beyond the scope of what is addressed in this research effort.

Future investigation could consider ways to further improve performance of the RICK technique [96]. For example, it might be possible to obtain more accurate measurement estimates using a different state estimator. Currently, the model correction technique only updates the measurement matrix “H”. However, to capture dynamic errors in the system model, it might be possible to also update the transition matrix “F” and control input matrix “G” with techniques similar to those describe in Section 6.4. This improvement to RICK will provide a more accurate model of the system (i.e., with an improved model of the system, the LiMe controller can achieve improved performance). Also, it might be possible to achieve additional control performance by incorporating a predictive element with the control scheme (e.g., model predictive control).

The advanced control strategies investigated in this research effort (e.g., LiMeRICK) are based on a SISO identified model of the PZR, where PZR surge mass flow rate was the input and the PZR pressure was the output. However, these are several other state variables that would be beneficial to include in the PZR control model (e.g., level) that are included in the dynamic PZR model. Therefore, it would be useful to investigate MIMO control for this application. The same RICK method could be used to develop SISO models for the various inputs and outputs desired. Then, the multiple SISO models could be combined into a single MIMO model for control implementation.

In addition to investigating control methods for pressure control of a pulsed cooling system (e.g., tokamak cooling system), the instrumentation [30] [31] used for pressurizer control in this application should be analyzed for accuracy and range. Furthermore, a total combined (control loop) uncertainty budget [32] should be developed to evaluate if instruments that have the capability to measure the large fluctuation range during normal operation can also satisfy the resolution and accuracy requirements necessary for the safety and control functions. If not, an alternative design using multiple ranges of sensors may be necessary. However, such an approach will further complicate the control design, since appropriate logic will need to be developed (which is outside the scope of this research project) to specify which instrument input is to be used for which range of the process variations. Therefore, it is recommended that future research initially focus on instrument accuracy, range, and uncertainty, as pertaining to performing the pressurizer control functions for a pulsed cooling system.

Further research not directly pertaining to, but potentially impacting instrumentation and control is needed in material/component life. The cooling system pulses result in pressurizer insurge and outsurge transients. This causes increased fatigue of all associated materials and components. Therefore, additional investigation is needed to determine if the added fatigue necessitates any design changes or modifications. If PZR design changes are necessary, they could potentially impact the behavior and control of the PZR. Consequently, following sufficient inquiry of this area, the control research presented in this document should be reevaluated to assess potential impact.

REFERENCES

[1] R. L. Murray, *Nuclear Energy: An Introduction to the Concepts, Systems, and Application of Nuclear Processes*, 5th ed., Butterworth-Heinemann, USA, 2000, pp 174-193.

[2] A. A. Harms, K. F. Schoepf, G. H. Miley, and D. R. Kingdon, *Principles of Fusion Energy: An Introduction to Fusion Energy for Students of Science and Engineering*, World Scientific Publishing Company, USA, 2000.

[3] S. Cowley, "Hot Fusion," *Physic World*, vol. 23, pp. 46-53, Oct. 2010.

[4] G. M. McCracken, P. Stott, *Fusion: The Energy of the Universe*, Elsevier Academic Press, USA, 2005.

[5] *iter*, ITER Organization, 2015.

[6] N. Anne Davies, U.S. Department of Energy, Fusion Energy Sciences Program, Nuclear Task Force of PCAST Energy R&D Panel.

[7] "Introduction to Fusion via Plasma Magnetic Confinement," *Universidad Carlos III de Madrid, Departamento de Física*.

[8] "Magnetic Confinement," *Encyclopedia Britannica*, 2015.

[9] J. V. Stewart, *Intermediate Electromagnetic Theory*, World Scientific Publishing Company, USA, 2001, pp. 277.

[10] R. Bansal, *Handbook of Engineering Electromagnetics*, Marcel Dekker, USA, 2004.

[11] W. M. Stacey, *Fusion: An Introduction to the Physics and Technology of Magnetic Confinement Fusion*, 2nd ed., Wiley-VCH, Germany, 2010.

[12] *Laboratory of Theoretical Physics*, University of Latvia, 2015.

[13] I. Benfatto, P. Bettini, M. Cavinato, A. De Lorenzi, J. Hourtoule, E. Serra, "Magnetic compatibility of standard components for electrical installations: Computation of the background field and consequences on the design of the electrical distribution boards and control boards for the ITER Tokamak building," *Fusion Engineering and Design*, vol. 75-79, Pages 235-239, Nov. 2005.

[14] US NRC Technical Training Center, 4-1, 0603, "Reactor Concepts Manual - Pressurized Water Reactor Systems".

[15] N. E. Todreas, M. S. Kazimi, *Nuclear Systems Volume I: Thermal Hydraulic Fundamentals*, 2nd ed., Taylor & Francis, USA, 2010.

[16] N. I. Kolev, *Multiphase Flow Dynamics 5: Nuclear Thermal Hydraulics*, 2nd ed., Springer, Germany, 2012.

[17] K. D. Kok, *Nuclear Engineering Handbook*, CRC Press, USA, 2009.

[18] M. W. Hibbell, *The Fundamentals of Nuclear Power Generation: Questions and Answers*, AuthorHouse, USA, 2011.

[19] L. Wang, *Model Predictive Control System Design and Implementation Using MATLAB®*, 1st ed., Springer, USA, 2009.

[20] D. E. Kirk, *Optimal Control Theory: An Introduction*, Dover Publications, USA, 2004.

[21] C. J. Harris, C. G. Moore, M. Brown, *Intelligent Control: Aspects of Fuzzy Logic and Neural Nets*, World Scientific Pub Co Inc, Singapore, 1993.

[22] E. Popov, A. Ying, "Modeling and simulation of the ITER first wall/blanket primary heat transfer system," *ANS Fusion Science and Technology*, vol. 60, no. 1, pp. 128-133, Jul. 2011.

[23] D. Carlonia, G. Dell'Orcob, G. Babulalb, F. Sombolib, L. Seriob, S. Paci, "Thermal hydraulic feasibility analysis of the IBED PHTS for ITER," *Fusion Engineering Design*, vol. 88, pp. 1709-1713, Jun. 2013.

[24] J. J. Carbajo, G. L. Yoder, S. H. Kim, US ITER 12103-TD0002-R00, "RELAP5 model of the vacuum vessel primary heat transfer system," ORNL, Oak Ridge, TN, May 2010.

[25] E. Popov, G. L. Yoder, S. H. Kim, US ITER 12102-TD0002-R00, "RELAP5 model of the divertor primary heat transfer system," ORNL, Oak Ridge, TN, Aug. 2010.

[26] T. Péni, G. Szederkényi, "Model predictive control for the hybrid primary circuit dynamics of a pressurized water nuclear power plant," *Periodica Polytechnica*, vol. 53, no. 1-2, pp. 37-44, 2009.

[27] J. Guo, X. Sun, H. Wu, "Research of pressurizer water level control system based on fuzzy-PID control," in *IEEE Fourth International Workshop on Advanced Computational Intelligence*, Wuhan, Hubei, China, Oct. 19-21, 2011, pp. 706-709.

[28] J. Ye, J. Yi, H. Ji, “Research on pressurizer water level control of nuclear reactor based on RBF neural network and PID controller,” in *Proceedings of the Ninth International Conference on Machine Learning and Cybernetic*s, Qingdao, Jul. 11-14, 2010, pp. 1486-1489.

[29] M. Lin, D. Hou, Z. Xu, Y. Yang, R. Zhang , “System simulation of NPP by coupling RELAP5 and MATLAB-Simulink.” in *Proceedings of ASME ICONE14*, Miami, FL, Jul. 17–20, 2006, vol. 2, pp. 337-344.

[30] B. G. Liptak, *Instrument Engineers’ Handbook, Volume I: Process Measurement and Analysis*, 4th ed., CRC Press, USA, 2003.

[31] B. G. Liptak, *Instrument Engineers’ Handbook, Volume II: Process Control and Optimization*, 4th ed., CRC Press, USA, 2006.

[32] R. H. Dieck, *Measurement Uncertainty: Methods and Applications*, 4th ed., ISA, USA, 2007.

[33] *Electronic Reading Room, Basic References, Students’ Corner, Nuclear Reactors, Pressurized Water Reactor*, U.S. Nuclear Regulatory Commission, 2015.

[34] J. Duderstadt, L. J. Hamilton, *Nuclear Reactor Analysis*, 1st ed., John Wiley & Sons Inc., USA, 1976.

[35] *DOE Fundamentals Handbook – Nuclear Physics and Reactor Theory*, DOE-HDBK-1019/1-93, vol. 1 of 2, U.S. Department of Energy, 1993.

[36] G. F. Franklin, J. D. Powell, M. L. Workman, *Digital Control of Dynamic Systems*, 3rd ed., Addison Wesley Longman. Inc., USA, 1998.

[37] K. Ogata, *Discrete-Time Control Systems*, 2nd ed., Prentice-Hall International Inc., USA, 1995.

[38] S. Kamalasadan, “ECGR 8890 Digital Control Systems Individual Study Class Notes,” University of North Carolina at Charlotte, Spring 2014.

[39] A. Pini, “A control-oriented model for pressurizer transient dynamics,” M.S. thesis, Energy Dept., Politecnico Di Milano, Milan, Italy, 2013.

[40] C. L. Phillips, H. T. Nagle, *Digital Control System Analysis and Design*, 3rd ed., Prentice Hall, USA, 1995.

[41] B. Kuo, *Digital Control Systems*, 2nd ed., Oxford University Press, USA, 1995.

[42] B. Kuo, F. Golnaraghi, *Automatic Control Systems*, 8th ed., John Wiley & Sons Inc., USA, 2003.

[43] M. V. Kothare, B. Mettler, M. Morari, P. Bendotti, C. M. Falinower, "Level control in the steam generator of a nuclear power plant," *IEEE Transactions on Control Systems Technology*, vol.8, no.1, pp. 55-69, Jan. 2000.

[44] P. Bendotti, B. Bodenheimer, "Identification and H_∞ control design for a pressurized water reactor," in *Proceedings of the 33rd IEEE Conference on Decision and Control*, Dec. 14-16, 1994, vol. 2, pp. 1072-1077.

[45] B. Bodenheimer, P. Bendotti, "Optimal linear parameter-varying control design for a pressurized water reactor," in *Proceedings of the 34th IEEE Conference on Decision and Control*, Dec. 13-15, 1995, vol.1, pp. 182-187.

[46] G. Becker, P. Bendotti, P. Gahinet, C. M. Falinower, "Analysis and controller synthesis for a pressurized water reactor using linear parameter varying systems," in *Proceedings of the 35th IEEE Conference on Decision and Control*, Dec. 11-13, 1996, vol. 2, pp. 1676-1681. doi: 10.1109/CDC.1996.572791.

[47] M.V. Oliveira, J.C.S. Almeida, "Application of artificial intelligence techniques in modeling and control of a nuclear power plant pressurizer system," *Progress in Nuclear Energy*, vol. 63, pp. 71-85, 2013. doi: 10.1016/j.pnucene.2012.11.005.

[48] B. Yang, X. Bian, W. Guo, "Application of adaptive fuzzy control technology to pressure control of a pressurizer," *Journal of Marine Science and Application*, vol. 4, issue 1, pp. 39-43, Mar. 2005.

[49] G. D. Zhang, X. H. Yang, S. Y. Qiao, Y. J. Wu, "Research on pressurizer pressure control system of 900MW pressurized water reactor nuclear power plant," *Advanced Materials Research*, vol. 718-720, pp. 1215-1220, 2013. doi: 10.4028/www.scientific.net/AMR.718-720.1215.

[50] C. Brown, H. A. Gabbar, "Fuzzy logic control for improved pressurizer systems in nuclear power," *Annals of Nuclear Energy*, vol. 72, pp. 461-466, 2014. doi: 10.1016/j.anucene.2014.05.024.

[51] S. P. Costa, F. R. de Andrade Lima, C. M. F. Lapa, C. A. B. de Oliveira Lira, "Artificial neural network used in the study of sensitivities in IRIS reactor pressurizer," in *International Nuclear Atlantic Conference - INAC 2011*, Brazil, Oct.24-28, 2011.

[52] S. P. Costa, F. R. de Andrade Lima, C. M. F. Lapa, A. C. de Abreu Mol, C. B. A. de Oliveira Lira, “The artificial neural network used in the study of sensitivities in the IRIS reactor pressurizer,” *Progress in Nuclear Energy*, vol. 69, pp. 64-70, 2013.

[53] Y. H. Cheng, J. R. Wang, H. T. Lin, C. Shih, “Benchmark calculations of pressurizer model for Maanshan nuclear power plant using TRACE code,” *Nuclear Engineering and Design*, vol. 239, pp. 2343–2348, 2009.

[54] S. Di Gennaro, B. Casillo-Toledo, “Comparative study of controllers for the supervision, control and protection systems in pressurized water reactors of evolutive generation,” Center of Excellence DEWS - Department of Electrical and Information Engineering, University of L’Aquila, Italy, 2011.

[55] M. Jin, C. Afang, L. Lixia, “Mathematical modeling and simulation of pressurizer pressure control system,” presented at the 2nd Int. Conf. on Computer Application and System Modeling, Taiyuan, Shanxi, China, Jul. 27-29, 2012.

[56] G. S. Varga, K. M. Hangos, J. Bokor, “Modeling and model identification of a pressurizer at the Paks nuclear power plant system identification,” in *14th IFAC Symposium on System Identification*, Newcastle, Australia, vol. 14, Part 1, Mar. 29-31, 2006, pp. 678-683. doi: 10.3182/20060329-3-AU-2901.00105.

[57] S. K. Moghanaki, M. Rahgoshay, “Simulation of two-region and four-region models for typical PWR pressurizer and benchmark obtained results using available results,” *Annals of Nuclear Energy*, vol. 63, pp. 302–308, 2014. doi: 10.1016/j.anucene.2013.08.014.

[58] T. W. Kim., J. W. Kim, G. C. Park, “Development of nonequilibrium pressurizer model with noncondensable gas,” *Nuclear Engineering and Design*, vol. 236, pp. 375-384, 2006. doi: 10.1016/j.nucengdes.2005.09.003.

[59] R. Zarghami, F. Jalali, N. Mostoufi, R. Sotudeh, K., Sepanloo, F. Dastjerdi, N., Ahmari, “The dynamic modeling of the pressurizer surge tank transients in light water reactor nuclear power plants,” *Iranian Journal of Science & Technology*, Transaction B, Engineering, vol. 29, no. B5, pp. 483-491, 2005.

[60] S. N. Kim, P. Griffith, “PWR pressurizer modeling,” *Nuclear Engineering and Design*, vol. 102, pp. 199-209, 1987. doi: 10.1016/0029-5493(87)90253-6.

[61] H. S. Saedi, P. Griffith, “The pressure response of a PWR pressurizer during an insurge transient,” *Transactions of The American Nuclear Society 1983 Annual Meeting, Thermal-Hydraulics Modeling*, vol. 44, Jun. 12-16, 1983.

[62] M. T. Leonard, P. Griffith, “The effects of a noncondensable gas on pressurizer insurge transients,” *LWR Thermal Hydraulics: II, Part 3, Transactions of The American Nuclear Society 1984 Annual Meeting*, vol. 46, Jun. 3-7, 1984.

[63] H. R. Saedi, “Insurge pressure response and heat transfer for PWR pressurizer,” M.S. thesis, Dept. Mech. Eng., MIT, Cambridge, MA, 1982.

[64] S. N. Kim, “An experimental and analytical model of a PWR pressurizer during transients,” P.h.D. dissertation, Dept. Nuclear Eng., MIT, Cambridge, MA, 1984.

[65] M. T. Leonard, “The effects of a non-condensable gas on pressurizer insurge transients,” M.S. thesis, Dept. Nuclear Eng., MIT, Cambridge, MA, 1983.

[66] R. G. Budynas, J.K. Nisbett, *Shigley's Mechanical Engineering Design*, 8th ed., McGraw-Hill, USA, 2008.

[67] J. A. Redfield, V. Prescop, S. G. Margolis, “Pressurizer Performance During Loss-of-Load Tests at Shippingport Analysis and Test,” *Nuclear Technology*, vol. 4, no. 3, pp. 173-181, Mar. 1968.

[68] V. Bobál, J. Böhm, J. Fessl, J. Machácek, *Digital Self-tuning Controllers: Algorithms, Implementation and Applications*, Springer, 2005.

[69] S. Kamalasadan, “ECGR 8090 Adaptive Control Systems Class Notes,” University of North Carolina at Charlotte, Fall 2014.

[70] P. Kundur, *Power System Stability and Control*, EPRI/ McGraw-Hill, USA, 1994.

[71] U. M. Ascher, L. R. Petzold, *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*, SIAM, USA, 1998.

[72] *RELAP5-3D Code Manual – Volume I: Code Structure, System Models, and Solution Methods*, Idaho National Laboratory INEEL-EXT-98-00834, Rev. 2.3, 2005.

[73] *TRACE 5.0 Theory Manual – Field Equations, Solution Methods, and Physical Models*, ML071000097, U.S. Nuclear Regulatory Commission, Draft.

[74] *FPL Energy Seabrook Station – Reactor Coolant System Pressurizer*, PID-1-RC-D20846, Rev. 14, ML100060476, U.S. Nuclear Regulatory Commission, 2006.

[75] *Westinghouse Technology Systems Manual, Section 10.2, Pressurizer Pressure Control System*, ML11223A287, U.S. Nuclear Regulatory Commission.

[76] G. D. Zhang, X. H. Yang, D. Q. Lu, Y. X. Liu, “Research on pressurizer pressure control system based on BP neural network control of self-adjusted PID parameters,” *Applied Mechanics and Materials*, vol. 291-294, pp. 2416-2423, 2013. doi: 10.4028/www.scientific.net/AMM.291-294.2416.

[77] L. Dai, X. Yang, G. Liu, J. Ye, H. Qian, Y. Xue, “Research on pressure control of pressurizer in pressurized water reactor,” *SPIE Proceedings from International Conference on Display and Photonics*, vol. 7749, 22 Jul. 2010.

[78] *Tutorial on Probabilistic Risk Assessment (PRA) - P-101: Risk-Informed Regulation for Technical Staff*, U.S. Nuclear Regulatory Commission, 2015.

[79] A. A. Boechat, U. F. Moreno, D. Haramura, “On-line calibration monitoring system based on data-driven model for oil well sensors,” *Proceedings of the 2012 IFAC Workshop on Automatic Control in Offshore Oil and Gas Production*, vol. 1, part 1, pp. 269-274, 2012. doi: 10.3182/20120531-2-NO-4020.00037.

[80] *Guidelines for Evaluating Fatigue Analyses Incorporating the Life Reduction of Metal Components Due to the Effects of the Light-Water Reactor Environment for New Reactors*, Regulatory Guide 1.207, March 2007.

[81] *Effect of LWR Coolant Environments on the Fatigue Life of Reactor Materials*, NUREG/CR-6909, Rev. 1.

[82] R. C. Dorf, R. H. Bishop, *Modern Control Systems*, 11th ed., Pearson Prentice Hall, Singapore, 2008.

[83] A. Visioli, *Practical PID Control*, Springer, London, 2006.

[84] D. A. Coggan, *Fundamentals of Industrial Control: Practical Guides for Measurement and Control*, 2nd ed., ISA, USA, 2005.

[85] R. G. Budynas, J. K. Nisbett, *Shigley's Mechanical Engineering Design*, 8th ed., McGraw-Hill, USA, 2008.

[86] R. S. Esfandiari, B. Lu, *Modeling and Analysis of Dynamic Systems*, 2nd ed., CRC Press, USA, 2014.

[87] M. S. Grewal, A. P. Andrews, *Kalman Filtering: Theory and Practice Using MATLAB*, 2nd ed., John Wiley & Sons, USA, 2001.

[88] Fisher® Instruction Manual, D103416X012, SS-84PSV4 Valve, “Fisher® Vee-Ball_ SS-84PSV4 NPS 4x3 Rotary Control Valve,” November 2010.

[89] Fisher® Instruction Manual, D103407X012, 1052PSV Actuator, “Fisher® 1052PSV Size 60 Rotary Actuator,” May 2011.

[90] D. R. Omstead, *Computer Control of Fermentation Processes*, CRC Press, USA, 1989.

[91] C. Dunis, S. Likothanassis, A. Karathanasopoulos, G. Sermpinis, K. Theofilatos, *Computational Intelligence Techniques for Trading and Investment*, Routledge, USA, 2014.

[92] K. F. Hughey, D. Nelson, J. K. Damminger, B. McCalla-Wriggins, *The Handbook of Career Advising*, Jossey-Bass, 2009.

[93] K. J. Astrom, B. Wittenmark, *Adaptive Control*, Addison-Wesley Publishing Company, USA, 1995.

[94] M. Jelali, *Control Performance Management in Industrial Automation: Assessment Based on Minimum-Variance Principles*, Advances in Industrial Control, Springer-Verlag, London, 2013, doi: 10.1007/978-1-4471-4546-2_2.

[95] M. Smith, S. Kamalasadan, “Hybrid State Output Feedback Controller for Pressurizer System,” *Manuscript submitted for publication*, 2015.

[96] M. Smith, S. Kamalasadan, “Predictive Knowledge Based Pressurizer Control via Kalman Filter Inspired System State Identification,” *Manuscript submitted for publication*, 2015.

[97] M. Smith., S. Kamalasadan, W. Van Hove, “Pressure Control for a Pressurizer in a Pulsed System,” *Proceedings of the ANS 9th International Conference on NPIC & HMIT*, Charlotte, NC, 23-26 February 2015.

[98] E. Bryson, *Applied Linear Optimal Control: Examples and Algorithms*, Cambridge University Press, United Kingdom, 2002.

[99] D. W. Franklin, D. M. Wolpert, “Computational Mechanisms of Sensorimotor Control,” *Neuron*, vol. 72, issue 3, pp. 425-442, Nov. 2011, doi: 10.1016/j.neuron.2011.10.006.

[100] R. E., Kalman, “A New Approach to Linear Filtering and Prediction Problems,” *Transactions of the ASME - Journal of Basic Engineering*, Volume 82, pp. 35-45, 1960, doi: 10.1115/1.3662552.

[101] D. Simon, *Optimal State Estimation: Kalman, H_∞, and Nonlinear Approaches*, John Wiley and Sons, USA, 2006.

[102] *Fusion Basics - Status and Prospects*, Max-Planck-Institutfür Plasmaphysik (IPP), Oct. 2015.

[103] M. Ragheb, “Magnetic Confinement Fusion,” University of Illinois at Urbana-Champaign, Nuclear Power Engineering (NPRE 402), USA, 4 Nov. 2013.

[104] T. Mento, A. St. John, “San Onofre To Be Permanently Closed,” KPBS, 7 June 2013.

[105] “Nuclear Pressurizer- Global Market Size, Competitive Landscape, Trends and Analysis to 2020,” Market Research Report, January 2011.

[106] *0520 - R325C - CE Technology Cross Training R325C, Chapter 6.1, Pressurizer Pressure Control*, ML11251A021, U.S. Nuclear Regulatory Commission.

[107] D. Flessner, “TVA orders new generator before unit is completed,” Times Free Press, 14 Feb. 2014.

[108] Curtiss-Wright Flow Control, “Target Rock - Nuclear Power Generation - Solenoid Operated Valves,” Literature Number 295.

APPENDIX A: MATLAB FILES FOR DYNAMIC SYSTEM CONTROLS

Appendix A.1: MATLAB Files for RLS ID

MATLAB Function File “RLS_ID_PZR_Function_R0.m”:

```
% Pressurizer Pressure Control
% Department of Electrical and Computer Engineering
% UNC Charlotte
% Professor: Sukumar Kamalasadan
% Function Description:
% RLS Identification without Output Corrector Implemented for PZR System
% Created by: Michael Smith
% 22 April 2015
% References:
% [1] S. Kamalasadan, “ECGR 8090 Adaptive Control Systems Class
%     Notes,” University of North Carolina at Charlotte, Fall 2014.

function [RLS_output] = RLS_ID_PZR_Function_R0(input_output)
persistent  P THETA X u y ym k

n=4; %Model Order
gamma= 0.99; %0.985; %Weighting factor;
alpha=1E9;
n_zero_delay = 20;
Ts = 1; %sampling time

Time_k = input_output(3);
%================================================================
% Recursive Least Squares Identification
%================================================================
if Time_k == 0;
    u(1:n_zero_delay) = input_output(1);
    y(1:n_zero_delay) = input_output(2);
    ym(1:n_zero_delay+1) = input_output(2);
    k = n_zero_delay;
    P=alpha*eye(2*n,2*n);
    THETA=zeros(2*n,1);

    %Form initial X vector
    for i=1:n;
        X(i,1)=-y(k+1-i);
        X(i+n,1)=u(k+1-i);
    end
end

k = k+1;

%=====================================================================
%  Add noise to system input and output
%=====================================================================
% Use sqare of the rms noise for the diagonal element E[v^2(t)], per
% Franklin, Powell, "Digital Control of Dyanmic Systems," 3rd Ed,
% Addison-Wiley, 1998, pp399.
% Measurement Noise (v) Covariance
Rk=(1.25e-03)^2;
V=0.316581291^2*randn;
Meas_Noise=V+sqrt(Rk)*randn;

u(k) = input_output(1) + Meas_Noise;
```

```
y(k) = input_output(2) + Meas_Noise;
ym(k) = ym(k-1);


%Begin on-line iteration for model identification
for kk=(k-n):k;
    K=(P*X)/(gamma+(X'*P*X));
    THETA=THETA+(K*(y(kk)-X'*THETA));

    for rr=1:n;
        ym(kk)=ym(kk)-THETA(rr)*ym(kk-rr)+THETA(rr+n)*u(kk-rr);
    end;

    % New variables
    P=(1/gamma)*(eye(2*n,2*n)-K*X')*P;
    for i=1:n;
        X(i,1)=-y(kk+1-i);
        X(i+n,1)=u(kk+1-i);
    end
end

% Final THETA
THETA_RLS=THETA';

%=================================================================
% RLS ID Function Output
%=================================================================
if k <= n_zero_delay+2
    ymout = input_output(2);
else
    ymout = X'*THETA;
end

x_state = [0, 0, 0, 0]';
%=================================================================
% State Estimate
%=================================================================
if Time_k > 2
    % Create discrete transfer function of THETA "G_tf_d"
    G_tf_d = tf([THETA_RLS(n+1:2*n)],[1, THETA_RLS(1:n)], Ts);
    % Convert "G_tf_d" to state space model
    G_ss_d = ss(G_tf_d);
    % Extract state space model matrices
    [Ad, Bd, Cd, Dd, Ts] = ssdata(G_ss_d);
    % Compute the state vector via y = C*x => x = inv(C_theta)*ym
    x_state = pinv(Cd)*ymout;
end

RLS_output =
[THETA_RLS(1),THETA_RLS(2),THETA_RLS(3),THETA_RLS(4),THETA_RLS(5),THETA_RLS(
6),THETA_RLS(7),THETA_RLS(8), x_state(1), x_state(2), x_state(3),
x_state(4), ymout];
```

Appendix A.2: MATLAB Files for Optimal Control

MATLAB Function File “Optimal_Controller_PZR_Spray_R3_linear.m”:

```
% Pressurizer Pressure Control
% Department of Electrical and Computer Engineering
% UNC Charlotte
% Professor: Sukumar Kamalasadan
% Dynamic PZR Model - Optimal Control for PZR Spray – Constant “K”
% Date: 5 August 2014

% Created by: Michael Smith

% References:
% [1]   G. F. Franklin, J. D. Powell, M. L. Workman, “Digital Control of
%       Dynamic Systems,” 3rd Ed., Addison Wesley Longman. Inc., 1998.
% [2]   K. Ogata, “Discrete-Time Control Systems,” 2nd Ed., Prentice-Hall
%       International Inc., 1995.
% [3]   S. Kamalasadan, “ECGR 8890 Digital Control Systems Class
%       Notes,” University of North Carolina at Charlotte, Spring 2014.
% [4]   C. L. Phillips, H. T. Nagle, "Digital Control System Analysis and
%       Design," 3rd Ed., 1995.
% [5]   B. Kuo, "Digital Control Systems," 2nd Edition, 1992.


function [mdot_spray_out, time_out] = Optimal_Controller_PZR_Spray_R3_linear
(ctrl_input)


% x_state, x_state_dot, u_input, mdot_spray_max
x_state(1) = ctrl_input(1);
x_state(2) = ctrl_input(2);
x_state(3) = ctrl_input(3);
x_state(4) = ctrl_input(4);
x_state = x_state';

x_state_dot(1) = ctrl_input(5);
x_state_dot(2) = ctrl_input(6);
x_state_dot(3) = ctrl_input(7);
x_state_dot(4) = ctrl_input(8);
x_state_dot = x_state_dot';

u_input = ctrl_input(9);

time = ctrl_input(10);

mdot_spray_max = ctrl_input(11);


P_max=150;
P_0 = 140;
P_s_on= 141; %142.721;  %bar
P_s_off=139.619; %bar

if x_state(1) >= P_s_on

    if time >= 0

        % display('INPUT ')
        Tsample = 1; %Sample Time, sec

        % Input System Information:
        % Time [sec]
        % x_State_Var_data = [1, 2, 3, 4]
        % 1) Press, [bar]
        % 2) level, [m]
        % 3) T_3l = Temp_l, [deg C]
        % 4) V_3v  [m^3]
        % u_Input_Ctrl_Var = [mdot_surge (kg/sec)]
        % Mdot_Surge [kg/sec]
        % dT = .1
        % x_size = 4
        % u_size = 1
        % sys_order = 4


        % Bc per "Shippingport_PZR_MIMO_SS_Estimate_R2.m"
        % Bc for SS est model with u = mdot_surge at dT = .1 sec
        Bc_est = [
        -9.718e-08
        -2.541e-05
        -6.214e-05
        -0.0001655];

        Bc = Bc_est;

        %Ac per "Shippingport_PZR_MIMO_SS_Estimate_R2.m"
        %Ac for SS est model with u = mdot_surge at dT = .1 sec
        Ac = [
        -1.362e-05   2.523e-05   3.384e-05  -8.447e-06
         0.005459   -0.001737   -0.001599  -0.0001972
        -0.004414    0.001948    0.003162    0.003184
         0.005199    -0.01009    -0.02774     -0.0165];

        Cc = [1, 1, 1, 1];
        Dc = [0];

        %============= Continuous Time System, G(s) ===============
        G_ss = ss(Ac, Bc, Cc, Dc);

        % ============= Discrete Time System, G[z] ===============
        G_ss_d = c2d(G_ss, Tsample, 'zoh');
        [Ad, Bd, Cd, Dd] = ssdata(G_ss_d);

        % [Wn,zeta] = damp(G_ss); % mathworks.com/help/control/ref/damp.html
        % Wn = min(Wn);
        % Tsys = 1/(2*Wn);     % Sampling Time "Tsample" must be <= Tsys
        %
        co = ctrb(G_ss_d);
        controllability = rank(co);
        ob = obsv(G_ss_d);
        observability = rank(ob);
        poles_Ad = eig(Ad);
        order = length(Ad);

        G_ss_d_minreal = minreal(G_ss_d);
        order_minreal = length(G_ss_d_minreal.a);

        if order == order_minreal

% display('Compute Optimal "K" gain coefficients')
            % mathworks.com/help/control/ref/lqr.html
            Q = abs(Ad'*Ad);
            R = eye(length(u_input));
            K = dlqr(Ad,Bd,Q,R);

            u_spray = -K*x_state/4E3; %*(1/10)*mdot_spray_max

            if (u_spray > 0) && (u_spray < 2*mdot_spray_max)
                mdot_spray_out = u_spray;
            end

            if u_spray > 2*mdot_spray_max
                mdot_spray_out = mdot_spray_max;
            end

            if u_spray <= 0
                mdot_spray_out = 0;
            end

        end

        if order > order_minreal
             mdot_spray_out = mdot_spray_max;
        end

    end
end

if x_state(1) < P_s_on
    mdot_spray_out = 0; %mdot_spray_max;
end
```

MATLAB Script File “Dynamic_PZR_SS_Estimate_R0.m”:

```
% Shippingport pressurizer tests were conducted in 1967
% Shown below is the partial output array from running
% "press_1v_2v_3v_PINI.slx"
% PINI_PZR_Surge_Data_Fig4-1.xlsx
% FIG. 4.1 Mass flow rate during 74 MW loss-of-load transient [kg/s vs. S]
% Data for input obtaind from PINI Fig. 4-1
% (Shippingport Mass Flow Rate Insurge Data)
% Extrapolated via WebPlotDigitizer (arohatgi.info/WebPlotDigitizer/)
% Removal of duplicate time series input to "press_1v_2v_3v_PINI.slx" via
% "PINI_PZR_Surge_Data_Fig4-1.xlsx"
% Removeal of duplicate output time series from "press_1v_2v_3v_PINI.slx"
% via "PINI_PZR_Surge_Data_Output_Simulink.xlsx"
% Created by: Michael Smith
% Date: 15 April 2014

% Note: "dT" and "sys_order" require input values specified

clc; clear all; close all;

% Time [sec]
Time = [...];

% PZR_Pressure [bar]
PZR_Pressure = [...];
```

```
% PZR_Mdot_Surge [kg/sec]
PZR_Mdot_Surge = [...];

dT = 1 %Desired new time step, (e.g., dT = 1 sec) REQUIRED INPUT...
Time_new = (min(Time):dT:max(Time));
% Time = min(Time):max(Time)/length(Time):max(Time)-1;
PZR_Pressure_new = interp1(Time, PZR_Pressure, Time_new, 'spline');
figure(1)
plot(Time, PZR_Pressure, 'o', Time_new, PZR_Pressure_new,'*')
legend('Original data','Interpolated data')
title('Time (sec) vs. PZR Pressure (bar)')
xlabel('Time (sec)')
ylabel('PZR Pressure (bar)')

PZR_Mdot_Surge_new = interp1(Time, PZR_Mdot_Surge, Time_new, 'spline');
figure(2)
plot(Time, PZR_Mdot_Surge, 'o', Time_new, PZR_Mdot_Surge_new,'*')
legend('Original data','Interpolated data')
title('Time (sec) vs. PZR Mdot (kg/sec)')
xlabel('Time (sec)')
ylabel('PZR PZR Mdot (kg/sec)')

%==================================================================
PZR_Data = iddata(PZR_Pressure_new', PZR_Mdot_Surge_new', [],
'SamplingInstants',Time_new')

%======================================================================
sys_order = 4 %order of desired system
sys = ssest(PZR_Data, sys_order)
sys_tf = tf(sys)

figure(3)
rlocus(sys_tf)

figure(4)
step(sys_tf)
```

MATLAB Function File “Optimal_Controller_PZR_Spray_R0.m”:

```
% Pressurizer Pressure Control
% Department of Electrical and Computer Engineering
% UNC Charlotte
% Professor: Sukumar Kamalasadan
% Dynamic PZR Model - Optimal Control for PZR Spray – Dynamic “B” and “K”
% Date: 5 August 2014

% Created by: Michael Smith

% References:
% [1]   G. F. Franklin, J. D. Powell, M. L. Workman, “Digital Control of
%       Dynamic Systems,” 3rd Ed., Addison Wesley Longman. Inc., 1998.
% [2]   K. Ogata, “Discrete-Time Control Systems,” 2nd Ed.,
%       Prentice-Hall International Inc., 1995.
% [3]   S. Kamalasadan, “ECGR 8890 Digital Control Systems Class
%       Notes,” University of North Carolina at Charlotte, Spring 2014.
% [4]   C. L. Phillips, H. T. Nagle, "Digital Control System Analysis and
%       Design," 3rd Ed., 1995.
% [5]   B. Kuo, "Digital Control Systems," 2nd Edition, 1992.
```

```
function [mdot_spray_out, time_out] =
Optimal_Controller_PZR_Spray_R0(ctrl_input)

% x_state, x_state_dot, u_input, mdot_spray_max
x_state(1) = ctrl_input(1);
x_state(2) = ctrl_input(2);
x_state(3) = ctrl_input(3);
x_state(4) = ctrl_input(4);
x_state = x_state';

x_state_dot(1) = ctrl_input(5);
x_state_dot(2) = ctrl_input(6);
x_state_dot(3) = ctrl_input(7);
x_state_dot(4) = ctrl_input(8);
x_state_dot = x_state_dot';

u_input = ctrl_input(9);

time = ctrl_input(10);

mdot_spray_max = ctrl_input(11);

P_max=150;
P_0 = 140;
P_s_on= 141; %142.721;  %bar
P_s_off=139.619; %bar

if x_state(1) >= P_s_on

    if time >= 0

        % display('INPUT ')

        Tsample = 1; %Sample Time, sec

        % Input System Information:
        % Time [sec]
        % x_State_Var_data = [1, 2, 3, 4]
        % 1) Press, [bar]
        % 2) level, [m]
        % 3) T_3l = Temp_l, [deg C]
        % 4) V_3v  [m^3]
        % u_Input_Ctrl_Var = [mdot_surge (kg/sec)]
        % Mdot_Surge [kg/sec]
        % dT = .1
        % x_size = 4
        % u_size = 1
        % sys_order = 4

        %Ac per "Shippingport_PZR_MIMO_SS_Estimate_R2.m"
        %Ac for SS est model with u = mdot_surge at dT = .1 sec
        Ac = [
        -1.362e-05   2.523e-05   3.384e-05  -8.447e-06
         0.005459   -0.001737   -0.001599  -0.0001972
        -0.004414    0.001948    0.003162    0.003184
         0.005199    -0.01009    -0.02774     -0.0165];

        % Construct Bc matrix
        Bc = (x_state_dot - Ac*x_state)*pinv(u_input);
        Cc = [1, 1, 1, 1];
        Dc = [0];
        %============= Continuous Time System, G(s) ===============
        G_ss = ss(Ac, Bc, Cc, Dc);

        % ============= Discrete Time System, G[z] ===============
        G_ss_d = c2d(G_ss, Tsample, 'zoh');
        [Ad, Bd, Cd, Dd] = ssdata(G_ss_d);

        % [Wn,zeta] = damp(G_ss); % mathworks.com/help/control/ref/damp.html
        % Wn = min(Wn);
        % Tsys = 1/(2*Wn);     % Sampling Time "Tsample" must be <= Tsys
        %
        co = ctrb(G_ss_d);
        controllability = rank(co);
        ob = obsv(G_ss_d);
        observability = rank(ob);
        poles_Ad = eig(Ad);
        order = length(Ad);

        G_ss_d_minreal = minreal(G_ss_d);
        order_minreal = length(G_ss_d_minreal.a);

        if order == order_minreal

% display('Compute Optimal "K" gain coefficients')
            % mathworks.com/help/control/ref/lqr.html
            Q = abs(Ad'*Ad);
            R = eye(length(u_input));
            K = dlqr(Ad,Bd,Q,R);

            u_spray = K*x_state;

            if (u_spray > 0) && (u_spray < 2*mdot_spray_max)
                mdot_spray_out = u_spray;
            end

            if u_spray > 2*mdot_spray_max
                mdot_spray_out = 1.3*(x_state(1)/P_0)*mdot_spray_max;
            end

            if u_spray <= 0
                mdot_spray_out = 0;
            end

        end

        if order > order_minreal
             mdot_spray_out = 1.3*(x_state(1)/P_0)*mdot_spray_max;
        end

    end
end

if x_state(1) < P_s_on
    mdot_spray_out = 0; %mdot_spray_max;
end
```

MATLAB Function File “Optimal_Controller_PZR_Spray_R4_RLS_ID.m”:

```
% Pressurizer Pressure Control
% Department of Electrical and Computer Engineering
% UNC Charlotte
% Professor: Sukumar Kamalasadan
% Dynamic PZR Model - Optimal Control for PZR Spray
% Date: 2 September 2014

% Created by: Michael Smith

% References:
% [1]   G. F. Franklin, J. D. Powell, M. L. Workman, “Digital Control of
%       Dynamic Systems,” 3rd Ed., Addison Wesley Longman. Inc., 1998.
% [2]   K. Ogata, “Discrete-Time Control Systems,” 2nd Ed.,
%       Prentice-Hall International Inc., 1995.
% [3]   S. Kamalasadan, “ECGR 8890 Digital Control Systems Class
%       Notes,” University of North Carolina at Charlotte, Spring 2014.
% [4]   C. L. Phillips, H. T. Nagle, "Digital Control System Analysis and
%       Design," 3rd Ed., 1995.
% [5]   Kuo, Benjamin, "Digital Control Systems," 2nd Edition, 1992.

function [mdot_spray_out, time_out] =
Optimal_Controller_PZR_Spray_R4_RLS_ID(ctrl_input)


% x_state, x_state_dot, u_input, mdot_spray_max
x_state(1) = ctrl_input(1);
x_state(2) = ctrl_input(2);
x_state(3) = ctrl_input(3);
x_state(4) = ctrl_input(4);
x_state = x_state';

x_state_dot(1) = ctrl_input(5);
x_state_dot(2) = ctrl_input(6);
x_state_dot(3) = ctrl_input(7);
x_state_dot(4) = ctrl_input(8);
x_state_dot = x_state_dot';

u_input = ctrl_input(9);

time = ctrl_input(10);

mdot_spray_max = ctrl_input(11);

THETA_RLS(1:8) = ctrl_input(11+1:11+8);

P_max=150;
P_0 = 140;
P_s_on= 141; %142.721;  %bar
P_s_off=139.619; %bar

if x_state(1) >= P_s_on


    if time >= 0

        % display('INPUT ')

        Ts = 1; %Sample Time, sec
        n = 4;
        % Input System Information:
```

```
        G_tf_d = tf([THETA_RLS(n+1:2*n)],[1, THETA_RLS(1:n)], Ts);

        G_ss_d = ss(G_tf_d);

        [Ad, Bd, Cd, Dd, Ts] = ssdata(G_ss_d);

        % [Wn,zeta] = damp(G_ss); % mathworks.com/help/control/ref/damp.html
        % Wn = min(Wn);
        % Tsys = 1/(2*Wn);     % Sampling Time "Tsample" must be <= Tsys
        %
        co = ctrb(G_ss_d);
        controllability = rank(co);
        ob = obsv(G_ss_d);
        observability = rank(ob);
        poles_Ad = eig(Ad);
        order = length(Ad);

        G_ss_d_minreal = minreal(G_ss_d);
        order_minreal = length(G_ss_d_minreal.a);

        if order == order_minreal

            % display('Compute Optimal "K" gain coefficients')
            % mathworks.com/help/control/ref/lqr.html
            Q = abs(Ad'*Ad);
            R = eye(length(u_input));
            K = dlqr(Ad,Bd,Q,R);
            u_spray = K*x_state*(2/125)*mdot_spray_max;

            if (u_spray > 0) && (u_spray < 2*mdot_spray_max)
                mdot_spray_out = u_spray;
            end

            if u_spray > 2*mdot_spray_max
                mdot_spray_out = 1.3*(x_state(1)/P_0)*mdot_spray_max;
            end

            if u_spray <= 0
                mdot_spray_out = 0;
            end

        end

        if order > order_minreal
            mdot_spray_out = 1.3*(x_state(1)/P_0)*mdot_spray_max;
        end

    end
end

if x_state(1) < P_s_on
    mdot_spray_out = 0; %mdot_spray_max;
end
```

MATLAB Function File “dlqr_fun.m”:

```
%================================================================
% Department of Electrical and Computer Engineering
% UNC Charlotte
% Professor: Sukumar Kamalasadan
% Pressurizer Control
% Discrete Linear Quadratic Regulator (LQR) Function
% Date: 20 May 2015
% Created by: Michael Smith
% References:
% [1] S. Kamalasadan, “ECGR 8090 Power System Control Class
%     Notes,” University of North Carolina at Charlotte, Spring 2015.
%================================================================

function Uout = dlqr_fun(dqlr_input)

Theta_lqr = dqlr_input(1:end-1);
ym = dqlr_input(end);

Ts = 0.1;
nn = length(Theta_lqr)/2;
G_tf_d = tf([Theta_lqr(nn+1:2*nn)'],[1, Theta_lqr(1:nn)'], Ts);
% Convert "G_tf_d" to state space model
G_ss_d = ss(G_tf_d);
% Extract state space model matrices
[Ak, Bk, Ck, Dk, Ts] = ssdata(G_ss_d);

% These values for Q, R, and H are specific to the Pressurizer Problem
Q = Ck'*Ck;
R=1;
H = Q;
Tf = 10;
N=Tf/Ts;
Plqr=H; % define initial value for Plqr (DARE solution)
for j=N-1:-1:0
    i=j+1; % shift indices to avoid index of 0
    F=inv(R+Bk'*Plqr*Bk)*Bk'*Plqr*Ak;
    Ff(i,:,:) = F;
    Plqr=(Ak-Bk*F)'*Plqr*(Ak-Bk*F)+F'*R*F+Q;
end

% Define final LQR gain for the PZR
K_pzr = F;

% Compute the estimated value of the states for the PZR THETA model
x_state = pinv(Ck)*ym;

% Compute the LQR control output (u = -K*x)
Uout = -K_pzr*x_state;

% time = N-1:-1:0;
% figure(1);
% plot(time,Ff(:,1,1),time,Ff(:,1,2),'MarkerSize',2)
% legend(['Optimal F(1,1)=',num2str(F(1,1))],['Optimal
F(1,2)=',num2str(F(1,2))],'Location','NorthEast')
% xlabel('Time')
% ylabel('F')
```

Appendix A.3: MATLAB Files for STR PID

MATLAB Function File “STR_PID_Controller_R1.m”:

```
% Self-tuning PID Controller
% Created by: Michael Smith
% Date: 4 September 2014
% References:
% [1]   V. Bobál, J. Böhm, J. Fessl, J. Machácek, "Digital Self-tuning
%       Controllers: Algorithms, Implementation and Applications," Springer,
%       2005.

function [mdot_spray_out] = STR_PID_Controller_R1(error_PID)
Ts = 1; %sec, set in simulation settings
alpha = 1; %closed loop pole shifting factor

%================================================================
% Self Tuning PID - Error Input and RLD ID System (THETA)
%================================================================
error_P = error_PID(1);
error_I = error_PID(2);
error_D = error_PID(3);

THETA_RLS = error_PID(4:7);

a1 = THETA_RLS(1);
a2 = THETA_RLS(2);
b1 = THETA_RLS(3);
b2 = THETA_RLS(4);

time_K = error_PID(8);
mdot_spray_max = error_PID(9);

if time_K < 2
    u_pid = mdot_spray_max;
    mdot_spray_out = u_pid;

elseif time_K >=2

%================================================================
% Self Tuning PID - Construct Self Tuning Matrix
%================================================================
X = 1-a1+alpha+a1*alpha; %X = r1+b1*s0
Y = a1-a2+a1*alpha^2+a2*alpha^2; %Y = r1*(a1-a)+b1*s1+b2*s0
Z = a2+a2*alpha^3; %Z = r1*(a2-a2)+b2*s1+b2*s2
D = 0; %D = -a2*r1+b2*s2

% [X]   [1          b1     0      0] [r1]
% [Y] = [a1-1       b2    b1      0] [s0]
% [Z]   [a2-a1       0    b2     b1] [s1]
% [0]   [-a2         0     0     b2] [s2]

Coeff_Matrix = [
    1,     b1,  0,   0;
    a1-1,  b2,  b1,  0;
    a2-a1, 0,   b2,  b1;
    -a2,   0,   0,   b2];

XYZD = [X;Y;Z;D];

Coeff_Vector = Coeff_Matrix\XYZD; %b = A*x => x = inv(A)*b = A\b
```

```
r1 = Coeff_Vector(1);
s0 = Coeff_Vector(2);
s1 = Coeff_Vector(3);
s2 = Coeff_Vector(4);

%=================================================================
% Self Tuning PID - Compute New Controller Output
%=================================================================
Kp = (s1+2*s2)/(1+r1);
Ki = (s0+s1+s2)/Ts;
Kd = Ts*(-r1*s1+(1-r1)*s2)/(1+r1);

u_pid = (error_P*Kp + error_I*Ki + error_D*Kd);
u_spray = u_pid*(2/50)*mdot_spray_max;
mdot_spray_out = u_spray;

    if (u_spray > 0) && (u_spray < 2*mdot_spray_max)
        mdot_spray_out = u_spray;
    end

    if u_spray > 2*mdot_spray_max
        mdot_spray_out = mdot_spray_max;
    end

    if u_spray <= 0
        if error_P <= 0
            mdot_spray_out = 0;
        end
        if error_P > 0
            P_0 = 140;
            P_k = P_0+error_P;
            mdot_spray_out = 1.3*(P_k/P_0)*mdot_spray_max;
        end
    end
end
```

Appendix A.4: MATLAB Files for MV Control

MATLAB Function File “mv_param_R0.m”:

```
function [parameters]=mv_param_R0(input)
% The code in “mv_param_R0.m” was adapted from [1].
% References:
% [1] V. Bobal, P. Chalupa, "Self-Tuning Controllers Simulink Library,"
%     Tomas Bata University in Zlín, Department of Control Theory, Faculty
%     of Applied Informatics, March 2008.
%
% [parameters] = mv_param_R0(input)
% Minimum Variance (MV) controller (2nd order example)
% This function computes MV controller parameters (r0, q0, q1, q2, p1, p2).
% The MV controller output is calculated as follows:
%
% u(k) = 1/q*(a1*y(k-1) + a2*y(k-2) - b1*u(k-1) - b2*u(k-2) + w(k)) + u(k-1)
%
%                    r0                      q0+q1*z^-1 + q2*z^-2
% U(z^-1) = ---------------------*W(z^-1)- -----------------------*Y(z^-1)
%            1+p1*z^-1 + p2*z^-2            1 + p1*z^-1 + p2*z^-2
%
% where q0=0
%
% Transfer function of the controlled system is:
%
%               b1*z^-1 + b2*z^-2
% Gs(z^-1) = -----------------------
%             1 + a1*z^-1 + a2*z^-2
%
% Input parameters:
% input(1) = a1, input(2) = b1, input(3) = a2, input(4) = b2, input(5) = q
%
% Output (MV controller) parameters:
% parameters (1) = r0, parameters(2) = q0, parameters(3) = q1,
% parameters (4) = q2, parameters (5) = p1, parameters (6) = p2

a1 = input(1);
a2 = input(2);
b1 = input(3);
b2 = input(4);
q = input(5); %q - penalization factor

% check q
if (q == 0)
   % penalization factor (q) > 0
   % q updated to 1e-6
   q = 1e-6;
end

r0 = 1/q;
q0 = -1/q * a1;
q1 = -1/q * a2;
q2 = 0;
p1 = 1/q * b1 - 1;
p2 = 1/q * b2;

parameters = [r0; q0; q1; q2; p1; p2];
```

MATLAB Function File “mv_ctrl_R0.m”:

```
function [u_mv] = mv_ctrl_R0(input)
% The code in “mv_ctrl_R0.m” was adapted from [1].
% References:
% [1] V. Bobal, P. Chalupa, "Self-Tuning Controllers Simulink Library,"
%     Tomas Bata University in Zlín, Department of Control Theory, Faculty
%     of Applied Informatics, March 2008.
%
% MV Controller via RQP feedforward feedback controller
% Output of the controller is calculated follows:
%
%                  r0                        q0 + q1*z^-1 + q2*z^-2
% U(z^-1) = --------------------*W(z^-1) - ------------------------*Y(z^-1)
%            1+p1*z^-1 +p2*z^-2              1 + p1*z^-1 + p2*z^-2
%
% u(k) = r0*w(k) - q0*y(k) - q1*y(k-1) - q2*y(k-2) - p1*u(k-1) - p2*u(k-2)
%
% input(1) = r0
% input(2) = q0
% input(3) = q1
% input(4) = q2
% input(5) = p1
% input(6) = p2
% input(7) = w(k)
% input(8) = y(k)
% input(9) = u(k)
% input(10) = Time(k)

persistent  uk_full yk_full
Time_k = input(10);
n_zero_delay = 5;

if Time_k == 0;    %initialization
    yk_full(1:n_zero_delay) = input(8);
    uk_full(1:n_zero_delay) = input(9);
    u_mv = 0;  %Number of continuous states.
end

if Time_k > 0;
    yk_full(end+1) = input(8);
    uk_full(end+1) = input(9);
    yk1 = yk_full(end-1);  %y(k-1)
    yk2 = yk_full(end-2);  %y(k-2)
    uk2 = uk_full(end-1);  %u(k-2)
    wk = input(7);   %w(k)
    yk = input(8);   %y(k)
    uk1 = input(9);  %u(k-1)
    r0 = input(1);
    q0 = input(2);
    q1 = input(3);
    q2 = input(4);
    p1 = input(5);
    p2 = input(6);

    u = r0*wk - q0*yk - q1*yk1 - q2*yk2 - p1*uk1 - p2*uk2;
    u_mv = u;
end
```

APPENDIX B: MATLAB FILES FOR PROPOSED CONTROL METHOD

Appendix B.1: MATLAB Files for LiMe Method

MATLAB Function File “LiMe_Ctrl_Weight_R1.m”:

```
function [ fun_output ] = LiMe_Ctrl_Weight_R1( LiMe_inputs )
% Created by: Michael Smith
% 15 April 2015
% LiMe_Ctrl_Weight provides the control cost function for LiMe
% Linear quadratic regulator (LQR) with Minimum variancE (MV) control (LiMe)
% Sensitivity of output error with respect to the LQR and MV inputs
% R_lqr = |d(e_pzr)/d(u_lqr)| =>
% R_lqr = |{e_pzr(k) - e_pzr(k-1)}/{u_lqr(k) - u_lqr(k-1)}|
% R_mv = |d(e_pzr)/d(u_mv)| =>
% R_mv = |{e_pzr(k) - e_pzr(k-1)}/{u_mv(k) - u_mv(k-1)}|
% e_pzr = P_ref - P_act
% Cost Function for LiMe:
% min{J_lime = alpha*J_lqr + (1-alpha)*J_mv}
% u_LiMe = alpha*u_lqr + (1-alpha)*u_mv
% alpha = R_lqr / (R_lqr + R_mv), subject to the following:
% |e_pzr(k) - e_pzr(k-1)| > 0
% |u_lqr(k) - u_lqr(k-1)| > 0
% |u_mv(k) - u_mv(k-1)| > 0
% if |e_pzr(k) - e_pzr(k-1)| = 0, alpha = 0.5 (equal weighting)
% if |u_lqr(k) - u_lqr(k-1)| = 0, alpha = 0 (let MV drive system)
% if |u_mv(k) - u_mv(k-1)| = 0, alpha = 1 (let LQR drive system)

persistent e_pzr_old u_lqr_old u_mv_old
u_mv = LiMe_inputs(1); %MV control input
Pact = LiMe_inputs(2); %actual pressure (measured) in bar
Pref = 140; %reference pressure in bar
u_lqr = LiMe_inputs(3); %LQR control input


if isempty(e_pzr_old)  %initialization of u_lqr
    e_pzr_old = 0;
end

if isempty(u_lqr)  %initialization of u_lqr
    u_lqr_old = 0;
end

if isempty(u_mv)  %initialization of u_mv
    u_mv_old = 0;
end

e_pzr = Pref - Pact;


if abs(e_pzr - e_pzr_old)>0 && abs(u_lqr - u_lqr_old)>0 && abs(u_mv -
u_mv_old)>0
    R_lqr = abs((e_pzr - e_pzr_old)/(u_lqr - u_lqr_old));
    R_mv = abs((e_pzr - e_pzr_old)/(u_mv - u_mv_old));
    alpha = R_lqr / (R_lqr + R_mv);
end

if abs(e_pzr - e_pzr_old) == 0
```

```
    alpha = 0.5; %(equal weighting)
end

if abs(u_lqr - u_lqr_old) == 0
    alpha = 0; %(let MV drive system)
    if abs(u_mv - u_mv_old) == 0
        alpha = 0.5; %(equal weighting)
    end
end

if abs(u_mv - u_mv_old) == 0
    alpha = 1; %(let lqr drive system)
end

u_LiMe = alpha*u_lqr + (1-alpha)*u_mv;

fun_output = [alpha, u_LiMe];

u_lqr_old = u_lqr;
u_mv_old = u_mv;
e_pzr_old = e_pzr;

end
```

Appendix B.2: MATLAB Files for RICK Method

MATLAB Function File “PZR_Kalman_Function_R1_lqr.m”:

```
% Function: Kalman Filer Implemented for PZR Pressure
% Created by: Michael Smith
% 16 March 2015
% References:
% [1] D. Simon, "Optimal State Estimation Kalman, Hinf, and Nonlinear
%     Approaches," John Wiley & Sons, USA, 2006.

function [fun_output] = PZR_Kalman_Function_R1_lqr(input_output)

persistent  Pkp Xhat

%=====================================================================
%  Define Input and Output of System
%=====================================================================
U = input_output(1);
Yact = input_output(2);
Time_k = input_output(3);


%=====================================================================
%  Construct the System Model
%=====================================================================
% Continuous Time State Space "A" Matrix
Ac = [
   -1.362e-05   2.523e-05   3.384e-05  -8.447e-06;
   0.005459   -0.001737   -0.001599  -0.0001972;
   -0.004414    0.001948    0.003162    0.003184;
   0.005199    -0.01009    -0.02774     -0.0165];

% Continuous Time State Space "C" Matrix
Bc = [
   -9.718e-08;
   -2.541e-05;
   -6.214e-05;
   -0.0001655];

% Continuous Time State Space "C" Matrix
C_full = [
    1.044e+04       -155     -121.3       -112;
    203.8     -17.49     -14.21      1.578;
    2.519e+04      3.586     -27.12      3.747;
    228.4      4.729      18.66     0.1887];

Cc = C_full(1,:);

% Continuous Time State Space "D" Matrix
D_full = [
        0;
        0;
        0;
        0];

Dc = 0;

% Sample Time, sec
Ts = 1;

% Discrete Time State Space Model
```

```
sys_model = c2d(ss(Ac,Bc,Cc,Dc),Ts);
[Ak,Bk,Ck,Dk,Ts] = ssdata(sys_model);

% ------------------------------------------------------------------
% The system model is constructed as follows for the Kalman filter:
%   x(k+1) = F*x(k) + w(k)
%   y(k) = H*x(k) + v(k)
%
% Define "F", "H", and "x" in terms of state space model
% "A", "B", "C" , "D", "x" and "u":
%   x(k+1) = A*x(k) + B*u(k) + w(k)
%   y(k) = C*x(k) + D*u(k) + v(k)
%
% Therefore, by observation:
%   xhat(k+1) = [A B; 0 I]*[x u+v]
%   yhat(k) = [C D]*[x u+v]

% [M,N] = size(X) for matrix X, returns the number of rows and columns in
% X as separate output variables.
% ------------------------------------------------------------------

[Bk_row, Bk_col] = size(Bk);

Fk=[Ak, Bk;...
   zeros(1,length(Ak)), eye(Bk_col)];

Hk=[Ck, Dk];

[Hk_row, Hk_col] = size(Hk);

% Process Noise (w) Covariance
Qk=(9e-8)^2*eye(Hk_col);

% Measurement Noise (v) Covariance
Rk=(1.25e-03)^2;


%=====================================================================
%  Add noise to system input and output
%=====================================================================
% Use square of the rms noise for the diagonal element E[v^2(t)], per
% Franklin, Powell, "Digital Control of Dyanmic Systems," 3rd Ed,
% Addison-Wiley, 1998, pp. 399.
V=0.316581291^2*randn;
Ymeas=Yact+V+sqrt(Rk)*randn;
Umeas=(U+V)+sqrt(Rk)*randn;
Uk=Umeas;
Yk=Ymeas;


%=====================================================================
%  Form initial Xhat (state estimation)
%=====================================================================
if Time_k == 0
    % Xhat = State Estimates
    Xhat=zeros(length(Fk),length(U));

    % Pk = "a posteriori" (empirical) error covariance matrix
    Pkp = (1.25e-03)*eye(Hk_col);
end


%=====================================================================
% Kalman Filtering of System
%=====================================================================
% A matrix inversion is not required for a SISO, since "Hk*Pkm*Hk'+Rk"
% is scalar.

for qq = 1:3
Pkm=Fk*Pkp*Fk'+Qk;
Kk=Pkm*Hk'*(Hk*Pkm*Hk'+Rk)^-1;
Xhat=Fk*[Xhat(1:length(Ak));Uk];
Xhat=Xhat+Kk*(Yk-Hk*Xhat);
Yhat=Hk*Xhat;
Pkp=(eye(Hk_col)-Kk*Hk)*Pkm*(eye(Hk_col)-Kk*Hk)'+Kk*Rk*Kk';
end

%=====================================================================
% Send System Output "Yhat" as Output of Function
%=====================================================================
yout = Yhat;
Ymeas_out = Ymeas;
fun_output = [yout, Ymeas_out];
```

MATLAB Function File “RLS_ID_PZR_Function_R2_lqr.m”:

```
% Function Description:
% RLS Identification with Output Corrector Implemented for PZR System
% Created by: Michael Smith
% 16 March 2015
% References:
% [1] S. Kamalasadan, “ECGR 8090 Adaptive Control Systems Class Notes,”
%     University of North Carolina at Charlotte, Fall 2014.

function [RLS_output] = RLS_ID_PZR_Function_R2_lqr(input_output)

persistent  P THETA X u y ym k

n=4; %Model Order
gamma= 0.99; %0.985; %Weighting factor;
alpha=1E9;
n_zero_delay = 20;
Ts = 1; %sampling time

Time_k = input_output(3);
%================================================================
% Recursive Least Squares Identification
%================================================================
if Time_k == 0;
    u(1:n_zero_delay) = input_output(1);
    y(1:n_zero_delay) = input_output(2);
    ym(1:n_zero_delay+1) = input_output(2);
    k = n_zero_delay;
    P=alpha*eye(2*n,2*n);
    THETA=zeros(2*n,1);

    %Form initial X vector
    for i=1:n;
        X(i,1)=-y(k+1-i);
        X(i+n,1)=u(k+1-i);
    end
end

k = k+1;
```

```
u(k) = input_output(1);
y(k) = input_output(2);
ym(k) = ym(k-1);


%Begin on-line iteration for model identification
for kk=(k-n):k;
    K=(P*X)/(gamma+(X'*P*X));
    THETA=THETA+(K*(y(kk)-X'*THETA));

    for rr=1:n;
        ym(kk)=ym(kk)-THETA(rr)*ym(kk-rr)+THETA(rr+n)*u(kk-rr);
    end;

    % New variables
    P=(1/gamma)*(eye(2*n,2*n)-K*X')*P;
    for i=1:n;
        X(i,1)=-y(kk+1-i);
        X(i+n,1)=u(kk+1-i);
    end
end

% Final THETA
THETA_RLS=THETA';

%================================================================
% RLS ID Function Output
%================================================================
if k <= n_zero_delay+2
    ymout = input_output(2);
else
    ymout = X'*THETA;
end

x_state = [0, 0, 0, 0]';
%================================================================
% Model Output Corrector
%================================================================
if Time_k > 2
    % Create discrete transfer function of THETA "G_tf_d"
    G_tf_d = tf([THETA_RLS(n+1:2*n)],[1, THETA_RLS(1:n)], Ts);
    % Convert "G_tf_d" to state space model
    G_ss_d = ss(G_tf_d);
    % Extract state space model matrices
    [Ad, Bd, Cd, Dd, Ts] = ssdata(G_ss_d);
    % Compute the state vector via y = C*x => x = inv(C_theta)*ym
    x_state = pinv(Cd)*ymout;
    % Update the "C" matrix with actual "y" via y = C*x => C_new = y*inv(x)
    Cd = y(end)*pinv(x_state);
    % Compute the model output "Ym" using the new "C" matrix
    ymout = Cd*x_state;
    % Constuct the new transfer function model of the system
    [G_tf_NUM, G_tf_DEN] = ss2tf(Ad, Bd, Cd, Dd);
    THETA_RLS= [G_tf_DEN(2:end), G_tf_NUM(2:end)];
end

RLS_output =
[THETA_RLS(1),THETA_RLS(2),THETA_RLS(3),THETA_RLS(4),THETA_RLS(5),THETA_RLS(
6),THETA_RLS(7),THETA_RLS(8), x_state(1), x_state(2), x_state(3),
x_state(4), ymout];
```

MATLAB Function File “THETA_R0_lqr.m”:

```
function [THETA_out] = THETA_R0_lqr(ym_THETA)
   % nn = length(ym_THETA);  %length of [THETA, Xhat, ym]
   % mm = nn-1;              %length of [THETA, Xhat]
   % pp = mm/3;              %length([THETA, Xhat]) = 3*length(Xhat)
   % qq = 2*pp:              %length of THETA = 2*length(Xhat)
   % THETA_out = ym_THETA(1:qq);
   THETA_out = ym_THETA(1:(length(ym_THETA)-1)*2/3);
end
```

MATLAB Function File “ym_R0_lqr.m”:

```
function [ymout] = ym_R0_lqr(ym_THETA)
      ymout = ym_THETA(length(ym_THETA));
end
```

Appendix B.3: MATLAB Files for LiMeRICK Method

See APPENDIX B sections for LiMe and RICK MATLAB files. The LiMeRICK method combined the LiMe and RICK methods. As such, it does not have any MATLAB unique function files.

Appendix B.4: MATLAB Files for LiMeRICK with Predictive Horizon

MATLAB Function File “PZR_Kalman_Function_MPC_R2_lqr.m”:

```
% Function: Kalman Filer Implemented for PZR Pressure
% Created by: Michael Smith
% 16 March 2015
% References:
% [1] D. Simon, "Optimal State Estimation Kalman, Hinf, and Nonlinear
%     Approaches," John Wiley & Sons, USA, 2006.

function [fun_output] = PZR_Kalman_Function_MPC_R2_lqr(input_output)

persistent  Pkp Xhat Yhatmpc_rms

%=====================================================================
%  Define Input and Output of System
%=====================================================================
U = input_output(1);
Yact = input_output(2);
Time_k = input_output(3);


%=====================================================================
%  Construct the System Model
%=====================================================================
% Continuous Time State Space "A" Matrix
Ac = [
   -1.362e-05   2.523e-05   3.384e-05  -8.447e-06;
   0.005459   -0.001737   -0.001599  -0.0001972;
   -0.004414    0.001948    0.003162    0.003184;
   0.005199    -0.01009    -0.02774     -0.0165];

% Continuous Time State Space "C" Matrix
Bc = [
   -9.718e-08;
   -2.541e-05;
   -6.214e-05;
   -0.0001655];

% Continuous Time State Space "C" Matrix
C_full = [
    1.044e+04       -155     -121.3       -112;
    203.8     -17.49     -14.21      1.578;
    2.519e+04      3.586     -27.12      3.747;
    228.4      4.729      18.66     0.1887];

Cc = C_full(1,:);

% Continuous Time State Space "D" Matrix
D_full = [
        0;
        0;
        0;
        0];

Dc = 0;

% Sample Time, sec
Ts = 1;

% Discrete Time State Space Model
```

```
sys_model = c2d(ss(Ac,Bc,Cc,Dc),Ts);
[Ak,Bk,Ck,Dk,Ts] = ssdata(sys_model);

% -----------------------------------------------------------------
% The system model is constructed as follows for the Kalman filter:
%   x(k+1) = F*x(k) + w(k)
%   y(k) = H*x(k) + v(k)
%
% Define "F", "H", and "x" in terms of state space model
% "A", "B", "C" , "D", "x" and "u":
%   x(k+1) = A*x(k) + B*u(k) + w(k)
%   y(k) = C*x(k) + D*u(k) + v(k)
%
% Therefore, by observation:
%   xhat(k+1) = [A B; 0 I]*[x u+v]
%   yhat(k) = [C D]*[x u+v]

% [M,N] = size(X) for matrix X, returns the number of rows and columns in
% X as separate output variables.
% -----------------------------------------------------------------

[Bk_row, Bk_col] = size(Bk);

Fk=[Ak, Bk;...
   zeros(1,length(Ak)), eye(Bk_col)];

Hk=[Ck, Dk];

[Hk_row, Hk_col] = size(Hk);

% Process Noise (w) Covariance
Qk=(9e-8)^2*eye(Hk_col);

% Measurement Noise (v) Covariance
Rk=(1.25e-03)^2;


%=====================================================================
%  Add noise to system input and output
%=====================================================================
% Use square of the rms noise for the diagonal element E[v^2(t)], per
% Franklin, Powell, "Digital Control of Dyanmic Systems," 3rd Ed,
% Addison-Wiley, 1998, pp. 399.
V=0.316581291^2*randn;
Ymeas=Yact+V+sqrt(Rk)*randn;
Umeas=(U+V)+sqrt(Rk)*randn;
Uk=Umeas;
Yk=Ymeas;


%=====================================================================
% Form initial Xhat (state estimation)
%=====================================================================
if Time_k == 0
    % Xhat = State Estimates
    Xhat=zeros(length(Fk),length(U));

    % Pk = "a posteriori" (empirical) error covariance matrix
    Pkp = (1.25e-03)*eye(Hk_col);

    % Initial value for predicted Yhat RMS
    Yhatmpc_rms = 140;
end
```

```
%=====================================================================
% Kalman Filtering of System
%=====================================================================
% A matrix inversion is not required for a SISO, since "Hk*Pkm*Hk'+Rk"
% is scalar.

for qq = 1:1
    Pkm=Fk*Pkp*Fk'+Qk;
    Kk=Pkm*Hk'*(Hk*Pkm*Hk'+Rk)^-1;
    Xhat=Fk*[Xhat(1:length(Ak));Uk];
    Xhat=Xhat+Kk*(Yk-Hk*Xhat);
    Yhat=Hk*Xhat;
    Pkp=(eye(Hk_col)-Kk*Hk)*Pkm*(eye(Hk_col)-Kk*Hk)'+Kk*Rk*Kk';

    % Predictive Estimate for input to RLS ID to create MPC
    n_mpc = 40; % MPC predictive steps (horizon) "n"
    Xhatmpc =  Xhat; %set initial X value for predictive step
    Yhatmpc(1)=Yhatmpc_rms; %set initial Y value for predictive step
    for ff = 1:n_mpc
        %Predict "X" ahead "n" steps
        Xhatmpc = Fk*[Xhatmpc(1:length(Ak));Uk];
        %Predict "Y" ahead "n" steps
        Yhatmpc(ff+1)=Hk*Xhatmpc;
    end
    % Predicted Y Root Mean Square (RMS) for "n" steps
    Yhatmpc_rms = sqrt(sum(Yhatmpc.^2)/(n_mpc+1));
end

%=====================================================================
% Send System Output "Yhat" as Output of Function
%=====================================================================
yout = Yhatmpc_rms;
Ymeas_out = Ymeas;
fun_output = [yout, Ymeas_out];
```

APPENDIX C: SIMULATION RESULTS FOR DIFFERENT INPUTS

Appendix C.1: Sample Data Output

This appendix section provides a typical sample data output file from the dynamic pressurizer model simulations. This sample data output is for the LiMeRICK method.

TABLE 19: Sample data output for dynamic PZR simulation (LiMeRICK)

Time (sec)	Level (m)	Pressure (bar)	T1 (ºC)	T2 (ºC)	T3v (ºC)	T3l (ºC)	$\dot{m}$ Surge (kg/sec)	$\dot{m}$ Spray (kg/sec)	$\dot{Q}$ (kW)
0	2.5128	140.00	336.66	336.66	336.68	336.66	0.0000	0.0000	40
1	2.5130	140.01	336.64	336.66	336.69	336.67	0.6500	0.0000	0
2	2.5138	140.05	336.57	336.67	336.72	336.67	1.3000	0.0000	0
3	2.5150	140.12	336.45	336.67	336.77	336.67	1.9500	0.0000	0
4	2.5166	140.21	336.29	336.67	336.85	336.68	2.6000	0.0000	0
5	2.5188	140.33	336.07	336.68	336.94	336.68	3.2500	0.0000	0
6	2.5214	140.47	335.82	336.68	337.05	336.69	3.7500	0.0000	0
7	2.5244	140.64	335.54	336.69	337.19	336.70	4.2500	0.0000	0
8	2.5277	140.82	335.21	336.69	337.33	336.71	4.7500	0.0000	0
9	2.5314	141.03	334.86	336.70	337.50	336.73	5.2500	0.0000	0
10	2.5356	141.26	334.46	336.70	337.68	336.74	5.7500	0.0000	0
11	2.5400	141.51	334.05	336.69	337.88	336.76	6.0500	0.0000	0
12	2.5447	141.77	333.61	336.69	338.09	336.77	6.3500	0.0000	0
13	2.5496	142.05	333.15	336.68	338.30	336.79	6.6500	0.0000	0
14	2.5548	142.34	332.68	336.66	338.54	336.81	6.9500	0.0000	0
15	2.5602	142.65	332.19	336.64	338.78	336.82	7.2500	0.0000	0
16	2.5658	142.97	331.68	336.62	339.03	336.84	7.4500	3.4668	0
17	2.5791	142.99	331.15	336.57	339.05	336.85	7.6500	3.5067	0
18	2.5926	143.02	330.61	336.51	339.07	336.85	7.8500	3.5070	0
19	2.6062	143.06	330.06	336.45	339.11	336.86	8.0500	3.5067	0
20	2.6200	143.11	329.50	336.39	339.15	336.87	8.2500	3.5052	0
21	2.6340	143.17	328.94	336.31	339.20	336.88	8.3750	3.5056	0
22	2.6481	143.23	328.37	336.23	339.25	336.88	8.5000	3.4887	0
23	2.6623	143.30	327.80	336.15	339.31	336.89	8.6250	3.4705	0
24	2.6766	143.38	327.23	336.06	339.38	336.90	8.7500	3.4774	0
25	2.6911	143.46	326.66	335.96	339.45	336.90	8.8750	3.4899	0
26	2.7056	143.55	326.08	335.85	339.52	336.91	8.9000	3.4720	0
27	2.7202	143.64	325.52	335.74	339.60	336.92	8.9250	3.5135	0
28	2.7348	143.74	324.95	335.62	339.68	336.92	8.9500	3.5163	0
29	2.7495	143.83	324.39	335.50	339.76	336.93	8.9750	3.5181	0
30	2.7642	143.93	323.84	335.37	339.84	336.94	9.0000	3.5214	0
31	2.7789	144.03	323.29	335.24	339.92	336.94	8.9750	3.5239	0
32	2.7937	144.13	322.75	335.10	340.00	336.94	8.9500	3.5291	0
33	2.8084	144.23	322.22	334.96	340.08	336.95	8.9250	3.5320	0
34	2.8231	144.32	321.70	334.81	340.16	336.95	8.9000	3.5345	0
35	2.8379	144.42	321.18	334.66	340.25	336.96	8.8750	3.5373	0
36	2.8526	144.51	320.67	334.51	340.32	336.96	8.7500	3.5397	0

TABLE 19 (continued)

Time (sec)	Level (m)	Pressure (bar)	T1 (ºC)	T2 (ºC)	T3v (ºC)	T3l (ºC)	$\dot{m}$ Surge (kg/sec)	$\dot{m}$ Spray (kg/sec)	$\dot{Q}$ (kW)
37	2.8672	144.60	320.18	334.35	340.40	336.96	8.6250	3.5443	0
38	2.8817	144.68	319.70	334.19	340.47	336.96	8.5000	3.5470	0
39	2.8962	144.76	319.23	334.03	340.54	336.96	8.3750	3.5491	0
40	2.9105	144.83	318.77	333.87	340.60	336.96	8.2500	3.5513	0
41	2.9248	144.89	318.33	333.71	340.65	336.96	8.0000	3.5533	0
42	2.9388	144.94	317.90	333.55	340.70	336.96	7.7500	3.5553	0
43	2.9527	144.98	317.49	333.39	340.73	336.95	7.5000	3.5565	0
44	2.9663	145.00	317.10	333.23	340.76	336.95	7.2500	3.5572	0
45	2.9798	145.01	316.73	333.07	340.77	336.95	7.0000	3.5575	0
46	2.9931	145.01	316.37	332.92	340.78	336.94	6.8500	3.5574	0
47	3.0063	145.00	316.02	332.77	340.78	336.93	6.7000	3.5574	0
48	3.0194	144.98	315.68	332.62	340.77	336.93	6.5500	3.5573	0
49	3.0324	144.95	315.35	332.47	340.76	336.92	6.4000	3.5573	0
50	3.0452	144.92	315.04	332.32	340.74	336.91	6.2500	3.5571	0
51	3.0579	144.88	314.73	332.17	340.72	336.90	6.1500	3.5567	0
52	3.0706	144.83	314.42	332.02	340.69	336.90	6.0500	3.5575	0
53	3.0832	144.78	314.13	331.88	340.66	336.89	5.9500	3.5576	0
54	3.0957	144.73	313.84	331.73	340.63	336.88	5.8500	3.5579	0
55	3.1081	144.66	313.56	331.59	340.58	336.87	5.7500	3.5569	0
56	3.1204	144.60	313.29	331.44	340.54	336.85	5.6500	3.5582	0
57	3.1326	144.52	313.02	331.30	340.49	336.84	5.5500	3.5579	0
58	3.1448	144.44	312.76	331.16	340.44	336.83	5.4500	3.5598	0
59	3.1568	144.36	312.51	331.02	340.38	336.82	5.3500	3.5588	0
60	3.1688	144.27	312.26	330.88	340.31	336.81	5.2500	3.5617	0
61	3.1806	144.17	312.02	330.74	340.25	336.79	5.1000	3.5620	0
62	3.1924	144.06	311.79	330.61	340.17	336.78	4.9500	3.5540	0
63	3.2040	143.94	311.57	330.48	340.09	336.76	4.8000	3.5535	0
64	3.2154	143.82	311.35	330.35	340.00	336.75	4.6500	3.5550	0
65	3.2268	143.69	311.14	330.22	339.90	336.73	4.5000	3.5580	0
66	3.2380	143.54	310.94	330.10	339.80	336.72	4.4000	3.5618	0
67	3.2491	143.40	310.75	329.98	339.69	336.70	4.3000	3.6225	0
68	3.2602	143.24	310.56	329.86	339.58	336.68	4.2000	3.6217	0
69	3.2711	143.08	310.37	329.74	339.46	336.67	4.1000	3.5954	0
70	3.2820	142.92	310.19	329.63	339.33	336.65	4.0000	3.5891	0
71	3.2928	142.75	310.02	329.51	339.21	336.63	3.9500	3.5866	0
72	3.3035	142.57	309.85	329.40	339.08	336.61	3.9000	3.5398	0
73	3.3141	142.40	309.68	329.29	338.94	336.59	3.8500	3.5341	0
74	3.3247	142.22	309.51	329.17	338.81	336.58	3.8000	3.5322	0
75	3.3352	142.04	309.35	329.06	338.67	336.56	3.7500	3.5261	0
76	3.3456	141.85	309.19	328.95	338.53	336.54	3.6500	3.5232	0
77	3.3559	141.66	309.04	328.85	338.39	336.52	3.5500	3.5451	0
78	3.3662	141.47	308.89	328.74	338.24	336.49	3.4500	3.5390	0
79	3.3763	141.26	308.74	328.64	338.08	336.47	3.3500	3.5293	0
80	3.3863	141.05	308.60	328.54	337.92	336.45	3.2500	3.5260	0
81	3.3938	140.97	308.47	328.45	337.86	336.44	3.1500	3.6092	0
82	3.4013	140.89	308.34	328.36	337.79	336.42	3.0500	3.6112	0
83	3.4086	140.79	308.22	328.27	337.72	336.41	2.9500	3.6137	0
84	3.4127	140.88	308.10	328.19	337.79	336.41	2.8500	3.6161	0
85	3.4199	140.77	307.99	328.11	337.71	336.39	2.7500	3.6159	0
86	3.4237	140.85	307.88	328.04	337.77	336.39	2.6500	3.6220	0

TABLE 19 (continued)

Time (sec)	Level (m)	Pressure (bar)	T1 (ºC)	T2 (ºC)	T3v (ºC)	T3l (ºC)	$\dot{m}$ Surge (kg/sec)	$\dot{m}$ Spray (kg/sec)	$\dot{Q}$ (kW)
87	3.4275	140.91	307.78	327.97	337.82	336.38	2.5500	3.6266	0
88	3.4345	140.79	307.68	327.89	337.72	336.37	2.4500	3.6263	0
89	3.4381	140.84	307.58	327.83	337.77	336.36	2.3500	3.6327	0
90	3.4416	140.89	307.50	327.77	337.81	336.36	2.2500	3.6366	0
91	3.4451	140.93	307.41	327.71	337.84	336.35	2.1250	3.6412	0
92	3.4517	140.77	307.33	327.64	337.72	336.34	2.0000	3.6409	0
93	3.4550	140.80	307.25	327.59	337.74	336.33	1.8750	3.6471	0
94	3.4581	140.81	307.18	327.54	337.75	336.33	1.7500	3.6507	0
95	3.4612	140.82	307.12	327.49	337.76	336.32	1.6250	3.6551	0
96	3.4642	140.82	307.05	327.45	337.76	336.32	1.5250	3.6590	0
97	3.4671	140.82	307.00	327.41	337.76	336.31	1.4250	3.6623	0
98	3.4699	140.80	306.94	327.37	337.75	336.31	1.3250	3.6653	0
99	3.4727	140.79	306.89	327.33	337.74	336.30	1.2250	3.6681	0
100	3.4753	140.76	306.85	327.30	337.72	336.30	1.1250	3.6706	0
101	3.4779	140.73	306.81	327.27	337.69	336.29	1.0500	3.6730	0
102	3.4805	140.69	306.77	327.24	337.67	336.29	0.9750	3.6745	0
103	3.4829	140.65	306.73	327.21	337.63	336.28	0.9000	3.6759	0
104	3.4854	140.61	306.70	327.18	337.60	336.28	0.8250	3.6772	0
105	3.4877	140.56	306.66	327.16	337.56	336.27	0.7500	3.6782	0
106	3.4900	140.50	306.64	327.13	337.52	336.26	0.7000	3.6792	0
107	3.4923	140.45	306.61	327.11	337.47	336.26	0.6500	3.6793	0
108	3.4945	140.39	306.58	327.09	337.43	336.25	0.6000	3.6794	0
109	3.4967	140.32	306.56	327.07	337.38	336.25	0.5500	3.6793	0
110	3.4989	140.25	306.54	327.05	337.32	336.24	0.5000	3.6792	0
111	3.5010	140.18	306.52	327.04	337.27	336.24	0.4000	3.6789	0
112	3.5030	140.10	306.51	327.02	337.20	336.23	0.3000	3.6799	0
113	3.5049	140.02	306.49	327.01	337.14	336.22	0.2000	3.6807	0
114	3.5067	139.93	306.49	327.00	337.06	336.22	0.1000	3.6813	0
115	3.5085	139.83	306.48	327.00	336.99	336.21	0.0000	3.6818	0
116	3.5102	139.72	306.48	326.99	336.90	336.21	-0.1000	3.6820	0
117	3.5118	139.61	306.48	326.99	336.81	336.20	-0.2000	0.0000	0
118	3.5115	139.59	306.49	326.99	336.80	336.20	-0.3000	0.0000	0
119	3.5112	139.57	306.50	327.00	336.78	336.20	-0.4000	0.0000	0
120	3.5108	139.53	306.51	327.00	336.75	336.19	-0.5000	0.0000	0
121	3.5103	139.49	306.53	327.01	336.72	336.19	-0.6000	0.0000	0
122	3.5098	139.45	306.55	327.01	336.68	336.19	-0.7000	0.0000	0
123	3.5091	139.39	306.57	327.02	336.64	336.19	-0.8000	0.0000	0
124	3.5083	139.33	306.60	327.03	336.59	336.18	-0.9000	0.0000	0
125	3.5075	139.26	306.62	327.04	336.53	336.18	-1.0000	0.0000	0
126	3.5065	139.18	306.66	327.05	336.47	336.17	-1.1000	0.0000	0
127	3.5048	139.13	306.69	327.07	336.43	336.17	-1.2000	0.0000	0
128	3.5021	139.11	306.73	327.08	336.41	336.16	-1.3000	0.0000	0
129	3.4991	139.08	306.77	327.10	336.39	336.14	-1.4000	0.0000	0
130	3.4959	139.06	306.81	327.12	336.37	336.13	-1.5000	0.0000	0
131	3.4925	139.03	306.86	327.14	336.35	336.11	-1.5500	0.0000	0
132	3.4891	139.00	306.91	327.16	336.33	336.10	-1.6000	0.0000	0
133	3.4855	138.97	306.96	327.18	336.30	336.08	-1.6500	0.0000	0
134	3.4818	138.94	307.01	327.21	336.28	336.07	-1.7000	0.0000	0
135	3.4780	138.91	307.06	327.23	336.25	336.05	-1.7500	0.0000	0
136	3.4740	138.88	307.12	327.25	336.23	336.03	-1.8500	0.0000	0

TABLE 19 (continued)

Time (sec)	Level (m)	Pressure (bar)	T1 (ºC)	T2 (ºC)	T3v (ºC)	T3l (ºC)	$\dot{m}$ Surge (kg/sec)	$\dot{m}$ Spray (kg/sec)	$\dot{Q}$ (kW)
137	3.4698	138.85	307.17	327.28	336.20	336.01	-1.9500	0.0000	0
138	3.4654	138.81	307.24	327.30	336.17	335.99	-2.0500	0.0000	0
139	3.4608	138.78	307.30	327.33	336.14	335.97	-2.1500	0.0000	0
140	3.4559	138.74	307.37	327.36	336.11	335.95	-2.2500	0.0000	0
141	3.4509	138.70	307.43	327.38	336.08	335.92	-2.3250	0.0000	0
142	3.4457	138.65	307.51	327.41	336.04	335.90	-2.4000	0.0000	0
143	3.4403	138.61	307.58	327.44	336.01	335.88	-2.4750	0.0000	0
144	3.4348	138.57	307.65	327.48	335.97	335.85	-2.5500	0.0000	0
145	3.4291	138.52	307.73	327.51	335.94	335.82	-2.6250	0.0000	0
146	3.4233	138.47	307.81	327.54	335.90	335.80	-2.6750	0.0000	0
147	3.4173	138.42	307.89	327.57	335.86	335.77	-2.7250	0.0000	0
148	3.4113	138.38	307.97	327.60	335.82	335.74	-2.7750	0.0000	0
149	3.4051	138.33	308.06	327.64	335.78	335.71	-2.8250	0.0000	0
150	3.3988	138.28	308.14	327.67	335.74	335.69	-2.8750	0.0000	0
151	3.3925	138.22	308.23	327.71	335.70	335.66	-2.9250	0.0000	0
152	3.3860	138.17	308.31	327.74	335.65	335.63	-2.9750	0.0000	0
153	3.3794	138.12	308.40	327.77	335.61	335.60	-3.0250	0.0000	0
154	3.3728	138.06	308.49	327.81	335.58	335.57	-3.0750	0.0000	0
155	3.3662	138.01	308.58	327.84	335.55	335.54	-3.1250	0.0000	0
156	3.3595	137.96	308.67	327.88	335.52	335.50	-3.1500	0.0000	0
157	3.3528	137.90	308.77	327.91	335.49	335.47	-3.1750	0.0000	0
158	3.3460	137.85	308.86	327.95	335.46	335.44	-3.2000	0.0000	0
159	3.3391	137.79	308.95	327.98	335.42	335.41	-3.2250	0.0000	0
160	3.3322	137.74	309.05	328.02	335.39	335.38	-3.2500	0.0000	0
161	3.3253	137.68	309.14	328.05	335.36	335.35	-3.3000	0.0000	0
162	3.3182	137.62	309.24	328.09	335.33	335.31	-3.3500	0.0000	0
163	3.3110	137.56	309.33	328.13	335.29	335.28	-3.4000	0.0000	0
164	3.3037	137.51	309.43	328.16	335.26	335.25	-3.4500	0.0000	0
165	3.2963	137.45	309.53	328.20	335.23	335.21	-3.5000	0.0000	0
166	3.2888	137.39	309.63	328.23	335.19	335.18	-3.5250	0.0000	0
167	3.2813	137.32	309.73	328.27	335.16	335.14	-3.5500	0.0000	0
168	3.2737	137.26	309.83	328.30	335.12	335.11	-3.5750	0.0000	0
169	3.2661	137.20	309.93	328.34	335.09	335.07	-3.6000	0.0000	0
170	3.2584	137.14	310.03	328.37	335.05	335.04	-3.6250	0.0000	0
171	3.2507	137.07	310.13	328.41	335.01	335.00	-3.6400	0.0000	0
172	3.2430	137.01	310.24	328.44	334.98	334.96	-3.6550	0.0000	0
173	3.2352	136.95	310.34	328.48	334.94	334.93	-3.6700	0.0000	0
174	3.2274	136.88	310.44	328.51	334.90	334.89	-3.6850	0.0000	0
175	3.2196	136.82	310.54	328.55	334.87	334.85	-3.7000	0.0000	0
176	3.2117	136.76	310.64	328.58	334.83	334.82	-3.7100	0.0000	0
177	3.2038	136.69	310.74	328.61	334.79	334.78	-3.7200	0.0000	0
178	3.1959	136.63	310.85	328.64	334.76	334.74	-3.7300	0.0000	0
179	3.1880	136.56	310.95	328.68	334.72	334.71	-3.7400	0.0000	0
180	3.1801	136.50	311.05	328.71	334.68	334.67	-3.7500	0.0000	80
181	3.1720	136.44	311.15	328.74	334.65	334.64	-3.7500	0.0000	80
182	3.1640	136.38	311.25	328.77	334.62	334.60	-3.7500	0.0000	80
183	3.1560	136.33	311.35	328.81	334.58	334.57	-3.7500	0.0000	80
184	3.1480	136.27	311.45	328.84	334.55	334.54	-3.7500	0.0000	80
185	3.1400	136.22	311.55	328.87	334.52	334.51	-3.7500	0.0000	80
186	3.1320	136.16	311.65	328.90	334.49	334.47	-3.7400	0.0000	80

TABLE 19 (continued)

Time (sec)	Level (m)	Pressure (bar)	T1 (ºC)	T2 (ºC)	T3v (ºC)	T3l (ºC)	$\dot{m}$ Surge (kg/sec)	$\dot{m}$ Spray (kg/sec)	$\dot{Q}$ (kW)
187	3.1240	136.10	311.75	328.93	334.45	334.44	-3.7300	0.0000	80
188	3.1160	136.05	311.85	328.96	334.42	334.41	-3.7200	0.0000	80
189	3.1081	135.99	311.95	328.99	334.39	334.38	-3.7100	0.0000	80
190	3.1002	135.94	312.04	329.02	334.36	334.35	-3.7000	0.0000	80
191	3.0923	135.88	312.14	329.04	334.33	334.31	-3.6850	0.0000	80
192	3.0844	135.82	312.24	329.07	334.29	334.28	-3.6700	0.0000	80
193	3.0766	135.77	312.33	329.10	334.26	334.25	-3.6550	0.0000	80
194	3.0688	135.71	312.43	329.13	334.23	334.22	-3.6400	0.0000	80
195	3.0611	135.66	312.52	329.15	334.20	334.19	-3.6250	0.0000	80
196	3.0534	135.61	312.61	329.18	334.17	334.16	-3.6000	0.0000	80
197	3.0457	135.55	312.70	329.20	334.14	334.12	-3.5750	0.0000	80
198	3.0381	135.50	312.79	329.23	334.10	334.09	-3.5500	0.0000	80
199	3.0306	135.44	312.88	329.25	334.07	334.06	-3.5250	0.0000	80
200	3.0231	135.39	312.97	329.28	334.04	334.03	-3.5000	0.0000	80
201	3.0156	135.34	313.06	329.30	334.01	334.00	-3.4750	0.0000	80
202	3.0083	135.29	313.15	329.32	333.98	333.97	-3.4500	0.0000	80
203	3.0009	135.24	313.23	329.35	333.95	333.94	-3.4250	0.0000	80
204	2.9936	135.19	313.32	329.37	333.92	333.91	-3.4000	0.0000	80
205	2.9864	135.14	313.40	329.39	333.89	333.88	-3.3750	0.0000	80
206	2.9793	135.08	313.48	329.41	333.86	333.85	-3.3500	0.0000	80
207	2.9721	135.04	313.57	329.43	333.84	333.83	-3.3250	0.0000	80
208	2.9651	134.99	313.65	329.45	333.81	333.80	-3.3000	0.0000	80
209	2.9581	134.94	313.73	329.47	333.78	333.77	-3.2750	0.0000	80
210	2.9511	134.89	313.81	329.49	333.75	333.74	-3.2500	0.0000	80
211	2.9442	134.84	313.89	329.51	333.72	333.71	-3.2250	0.0000	80
212	2.9374	134.79	313.96	329.53	333.69	333.68	-3.2000	0.0000	80
213	2.9306	134.75	314.04	329.55	333.67	333.66	-3.1750	0.0000	80
214	2.9238	134.70	314.12	329.57	333.64	333.63	-3.1500	0.0000	80
215	2.9172	134.65	314.19	329.59	333.61	333.60	-3.1250	0.0000	80
216	2.9105	134.61	314.27	329.60	333.59	333.58	-3.1000	0.0000	80
217	2.9040	134.56	314.34	329.62	333.56	333.55	-3.0750	0.0000	80
218	2.8974	134.52	314.41	329.64	333.53	333.52	-3.0500	0.0000	80
219	2.8910	134.47	314.48	329.65	333.51	333.50	-3.0250	0.0000	80
220	2.8845	134.43	314.55	329.67	333.48	333.47	-3.0000	0.0000	80
221	2.8782	134.38	314.62	329.68	333.46	333.45	-2.9800	0.0000	80
222	2.8718	134.34	314.69	329.70	333.43	333.42	-2.9600	0.0000	80
223	2.8656	134.29	314.76	329.72	333.40	333.40	-2.9400	0.0000	80
224	2.8593	134.25	314.83	329.73	333.38	333.37	-2.9200	0.0000	80
225	2.8531	134.21	314.90	329.74	333.36	333.35	-2.9000	0.0000	80
226	2.8470	134.17	314.96	329.76	333.33	333.32	-2.8800	0.0000	80
227	2.8409	134.13	315.03	329.77	333.31	333.30	-2.8600	0.0000	80
228	2.8348	134.08	315.09	329.79	333.28	333.27	-2.8400	0.0000	80
229	2.8288	134.04	315.16	329.80	333.26	333.25	-2.8200	0.0000	80
230	2.8228	134.00	315.22	329.81	333.23	333.22	-2.8000	0.0000	80
231	2.8168	133.96	315.29	329.83	333.21	333.20	-2.7900	0.0000	80
232	2.8109	133.92	315.35	329.84	333.19	333.18	-2.7800	0.0000	80
233	2.8050	133.88	315.41	329.85	333.16	333.15	-2.7700	0.0000	80
234	2.7991	133.84	315.47	329.86	333.14	333.13	-2.7600	0.0000	80
235	2.7932	133.80	315.54	329.88	333.12	333.11	-2.7500	0.0000	80
236	2.7874	133.76	315.60	329.89	333.09	333.08	-2.7250	0.0000	80

TABLE 19 (continued)

Time (sec)	Level (m)	Pressure (bar)	T1 (ºC)	T2 (ºC)	T3v (ºC)	T3l (ºC)	$\dot{m}$ Surge (kg/sec)	$\dot{m}$ Spray (kg/sec)	$\dot{Q}$ (kW)
237	2.7816	133.72	315.66	329.90	333.07	333.06	-2.7000	0.0000	80
238	2.7759	133.68	315.72	329.91	333.05	333.04	-2.6750	0.0000	80
239	2.7702	133.64	315.78	329.92	333.02	333.02	-2.6500	0.0000	80
240	2.7646	133.60	315.83	329.93	333.00	332.99	-2.6250	0.0000	80
241	2.7591	133.57	315.89	329.95	332.98	332.97	-2.6000	0.0000	80
242	2.7536	133.53	315.95	329.96	332.96	332.95	-2.5750	0.0000	80
243	2.7481	133.49	316.00	329.97	332.94	332.93	-2.5500	0.0000	80
244	2.7427	133.46	316.06	329.98	332.91	332.91	-2.5250	0.0000	80
245	2.7374	133.42	316.11	329.99	332.89	332.89	-2.5000	0.0000	80
246	2.7320	133.39	316.17	330.00	332.87	332.86	-2.4800	0.0000	330
247	2.7265	133.38	316.22	330.01	332.87	332.86	-2.4600	0.0000	330
248	2.7209	133.38	316.28	330.02	332.87	332.86	-2.4400	0.0000	330
249	2.7154	133.37	316.33	330.03	332.86	332.86	-2.4200	0.0000	330
250	2.7099	133.37	316.38	330.04	332.86	332.85	-2.4000	0.0000	330
251	2.7045	133.36	316.43	330.05	332.86	332.85	-2.3800	0.0000	330
252	2.6991	133.36	316.48	330.06	332.86	332.85	-2.3600	0.0000	330
253	2.6938	133.36	316.53	330.07	332.86	332.85	-2.3400	0.0000	330
254	2.6885	133.35	316.59	330.08	332.85	332.85	-2.3200	0.0000	330
255	2.6832	133.35	316.63	330.09	332.85	332.84	-2.3000	0.0000	330
256	2.6780	133.35	316.68	330.10	332.85	332.84	-2.2900	0.0000	330
257	2.6728	133.35	316.73	330.11	332.85	332.84	-2.2800	0.0000	330
258	2.6676	133.34	316.78	330.12	332.85	332.84	-2.2700	0.0000	330
259	2.6624	133.34	316.83	330.13	332.85	332.84	-2.2600	0.0000	330
260	2.6572	133.34	316.88	330.14	332.85	332.84	-2.2500	0.0000	330
261	2.6521	133.34	316.93	330.15	332.85	332.84	-2.2250	0.0000	330
262	2.6471	133.34	316.97	330.16	332.84	332.84	-2.2000	0.0000	330
263	2.6420	133.34	317.02	330.17	332.84	332.84	-2.1750	0.0000	330
264	2.6371	133.34	317.06	330.18	332.84	332.84	-2.1500	0.0000	330
265	2.6322	133.34	317.11	330.19	332.84	332.84	-2.1250	0.0000	330
266	2.6273	133.34	317.15	330.20	332.85	332.84	-2.1100	0.0000	330
267	2.6225	133.34	317.20	330.20	332.85	332.84	-2.0950	0.0000	330
268	2.6176	133.34	317.24	330.21	332.85	332.84	-2.0800	0.0000	330
269	2.6129	133.34	317.28	330.22	332.85	332.84	-2.0650	0.0000	330
270	2.6081	133.34	317.33	330.23	332.85	332.84	-2.0500	0.0000	330
271	2.6034	133.35	317.37	330.24	332.85	332.84	-2.0300	0.0000	330
272	2.5987	133.35	317.41	330.25	332.85	332.84	-2.0100	0.0000	330
273	2.5941	133.35	317.45	330.26	332.85	332.85	-1.9900	0.0000	330
274	2.5895	133.36	317.49	330.26	332.86	332.85	-1.9700	0.0000	330
275	2.5850	133.36	317.53	330.27	332.86	332.85	-1.9500	0.0000	330
276	2.5805	133.36	317.57	330.28	332.86	332.85	-1.9350	0.0000	330
277	2.5760	133.37	317.61	330.29	332.86	332.85	-1.9200	0.0000	330
278	2.5715	133.37	317.65	330.30	332.87	332.86	-1.9050	0.0000	330
279	2.5671	133.38	317.69	330.31	332.87	332.86	-1.8900	0.0000	330
280	2.5627	133.38	317.73	330.31	332.87	332.86	-1.8750	0.0000	330
281	2.5584	133.39	317.77	330.32	332.87	332.87	-1.8500	0.0000	330
282	2.5541	133.39	317.80	330.33	332.88	332.87	-1.8250	0.0000	330
283	2.5499	133.40	317.84	330.34	332.88	332.87	-1.8000	0.0000	330
284	2.5457	133.41	317.88	330.34	332.89	332.88	-1.7750	0.0000	330
285	2.5415	133.41	317.91	330.35	332.89	332.88	-1.7500	0.0000	330
286	2.5374	133.42	317.95	330.36	332.89	332.89	-1.7400	0.0000	330

TABLE 19 (continued)

Time (sec)	Level (m)	Pressure (bar)	T1 (ºC)	T2 (ºC)	T3v (ºC)	T3l (ºC)	$\dot{m}$ Surge (kg/sec)	$\dot{m}$ Spray (kg/sec)	$\dot{Q}$ (kW)
287	2.5333	133.43	317.98	330.37	332.90	332.89	-1.7300	0.0000	330
288	2.5293	133.44	318.02	330.37	332.90	332.90	-1.7200	0.0000	330
289	2.5252	133.45	318.05	330.38	332.91	332.90	-1.7100	0.0000	330
290	2.5212	133.45	318.08	330.39	332.91	332.91	-1.7000	0.0000	330
291	2.5172	133.46	318.12	330.40	332.92	332.91	-1.6850	0.0000	330
292	2.5133	133.47	318.15	330.40	332.92	332.92	-1.6700	0.0000	330
293	2.5093	133.48	318.18	330.41	332.93	332.92	-1.6550	0.0000	330
294	2.5054	133.49	318.22	330.42	332.93	332.93	-1.6400	0.0000	330
295	2.5016	133.50	318.25	330.43	332.94	332.93	-1.6250	0.0000	330
296	2.4977	133.51	318.28	330.43	332.95	332.94	-1.6100	0.0000	330
297	2.4939	133.52	318.31	330.44	332.95	332.94	-1.5950	0.0000	330
298	2.4901	133.53	318.34	330.45	332.96	332.95	-1.5800	0.0000	330
299	2.4864	133.54	318.38	330.45	332.96	332.96	-1.5650	0.0000	330
300	2.4827	133.55	318.41	330.46	332.97	332.96	-1.5500	0.0000	330
301	2.4790	133.56	318.44	330.47	332.98	332.97	-1.5356	0.0000	330
302	2.4753	133.58	318.47	330.47	332.98	332.98	-1.5213	0.0000	330
303	2.4717	133.59	318.50	330.48	332.99	332.98	-1.5069	0.0000	330
304	2.4681	133.60	318.53	330.49	333.00	332.99	-1.4925	0.0000	330
305	2.4645	133.61	318.55	330.49	333.01	333.00	-1.4781	0.0000	330
306	2.4610	133.63	318.58	330.50	333.01	333.01	-1.4638	0.0000	330
307	2.4575	133.64	318.61	330.51	333.02	333.01	-1.4494	0.0000	330
308	2.4540	133.65	318.64	330.51	333.03	333.02	-1.4350	0.0000	330
309	2.4505	133.67	318.67	330.52	333.04	333.03	-1.4206	0.0000	330
310	2.4471	133.68	318.70	330.53	333.05	333.04	-1.4063	0.0000	330
311	2.4437	133.69	318.72	330.53	333.05	333.04	-1.3919	0.0000	330
312	2.4403	133.71	318.75	330.54	333.06	333.05	-1.3775	0.0000	330
313	2.4370	133.72	318.78	330.55	333.07	333.06	-1.3631	0.0000	330
314	2.4337	133.74	318.80	330.55	333.08	333.07	-1.3488	0.0000	330
315	2.4304	133.75	318.83	330.56	333.09	333.08	-1.3344	0.0000	330
316	2.4272	133.77	318.85	330.56	333.10	333.09	-1.3200	0.0000	330
317	2.4240	133.78	318.88	330.57	333.11	333.10	-1.3056	0.0000	330
318	2.4208	133.80	318.90	330.58	333.12	333.11	-1.2913	0.0000	330
319	2.4176	133.82	318.93	330.58	333.13	333.12	-1.2769	0.0000	330
320	2.4145	133.83	318.95	330.59	333.14	333.13	-1.2625	0.0000	330
321	2.4114	133.85	318.98	330.59	333.15	333.14	-1.2481	0.0000	330
322	2.4083	133.87	319.00	330.60	333.16	333.15	-1.2338	0.0000	330
323	2.4053	133.88	319.03	330.61	333.17	333.16	-1.2194	0.0000	330
324	2.4023	133.90	319.05	330.61	333.18	333.17	-1.2050	0.0000	330
325	2.3993	133.92	319.07	330.62	333.19	333.18	-1.1906	0.0000	330
326	2.3964	133.94	319.09	330.62	333.20	333.19	-1.1763	0.0000	330
327	2.3934	133.96	319.12	330.63	333.21	333.20	-1.1619	0.0000	330
328	2.3906	133.98	319.14	330.64	333.22	333.21	-1.1475	0.0000	330
329	2.3877	134.00	319.16	330.64	333.23	333.22	-1.1331	0.0000	330
330	2.3849	134.01	319.18	330.65	333.24	333.23	-1.1188	0.0000	330
331	2.3821	134.03	319.20	330.65	333.25	333.24	-1.1044	0.0000	330
332	2.3793	134.05	319.22	330.66	333.26	333.26	-1.0900	0.0000	330
333	2.3765	134.07	319.25	330.66	333.28	333.27	-1.0756	0.0000	330
334	2.3738	134.09	319.27	330.67	333.29	333.28	-1.0613	0.0000	330
335	2.3712	134.12	319.29	330.67	333.30	333.29	-1.0469	0.0000	330
336	2.3685	134.14	319.31	330.68	333.31	333.30	-1.0325	0.0000	330

TABLE 19 (continued)

Time (sec)	Level (m)	Pressure (bar)	T1 (ºC)	T2 (ºC)	T3v (ºC)	T3l (ºC)	$\dot{m}$ Surge (kg/sec)	$\dot{m}$ Spray (kg/sec)	$\dot{Q}$ (kW)
337	2.3659	134.16	319.33	330.69	333.32	333.32	-1.0181	0.0000	330
338	2.3633	134.18	319.34	330.69	333.34	333.33	-1.0038	0.0000	330
339	2.3607	134.20	319.36	330.70	333.35	333.34	-0.9894	0.0000	330
340	2.3582	134.22	319.38	330.70	333.36	333.35	-0.9750	0.0000	330
341	2.3557	134.25	319.40	330.71	333.38	333.37	-0.9606	0.0000	330
342	2.3532	134.27	319.42	330.71	333.39	333.38	-0.9463	0.0000	330
343	2.3508	134.29	319.44	330.72	333.40	333.39	-0.9319	0.0000	330
344	2.3483	134.31	319.46	330.72	333.42	333.41	-0.9175	0.0000	330
345	2.3460	134.34	319.47	330.73	333.43	333.42	-0.9031	0.0000	330
346	2.3436	134.36	319.49	330.73	333.44	333.43	-0.8888	0.0000	330
347	2.3413	134.39	319.51	330.74	333.46	333.45	-0.8744	0.0000	330
348	2.3390	134.41	319.52	330.74	333.47	333.46	-0.8600	0.0000	330
349	2.3367	134.44	319.54	330.75	333.49	333.48	-0.8456	0.0000	330
350	2.3345	134.46	319.56	330.75	333.50	333.49	-0.8313	0.0000	330
351	2.3323	134.49	319.57	330.76	333.52	333.51	-0.8169	0.0000	330
352	2.3301	134.51	319.59	330.76	333.53	333.52	-0.8025	0.0000	330
353	2.3280	134.54	319.60	330.77	333.55	333.54	-0.7881	0.0000	330
354	2.3258	134.56	319.62	330.77	333.56	333.55	-0.7738	0.0000	330
355	2.3238	134.59	319.63	330.78	333.58	333.57	-0.7594	0.0000	330
356	2.3217	134.62	319.65	330.78	333.59	333.58	-0.7450	0.0000	330
357	2.3197	134.64	319.66	330.79	333.61	333.60	-0.7306	0.0000	330
358	2.3177	134.67	319.68	330.79	333.62	333.61	-0.7163	0.0000	330
359	2.3157	134.70	319.69	330.80	333.64	333.63	-0.7019	0.0000	330
360	2.3138	134.73	319.70	330.80	333.66	333.65	-0.6875	0.0000	330
361	2.3119	134.75	319.72	330.81	333.67	333.66	-0.6731	0.0000	330
362	2.3100	134.78	319.73	330.81	333.69	333.68	-0.6588	0.0000	330
363	2.3081	134.81	319.74	330.82	333.71	333.70	-0.6444	0.0000	330
364	2.3063	134.84	319.75	330.82	333.72	333.71	-0.6300	0.0000	330
365	2.3045	134.87	319.77	330.82	333.74	333.73	-0.6156	0.0000	330
366	2.3028	134.90	319.78	330.83	333.76	333.75	-0.6013	0.0000	330
367	2.3010	134.93	319.79	330.83	333.77	333.76	-0.5869	0.0000	330
368	2.2993	134.96	319.80	330.84	333.79	333.78	-0.5725	0.0000	330
369	2.2977	134.99	319.81	330.84	333.81	333.80	-0.5581	0.0000	330
370	2.2960	135.02	319.82	330.85	333.83	333.82	-0.5438	0.0000	330
371	2.2944	135.05	319.83	330.85	333.84	333.83	-0.5294	0.0000	330
372	2.2929	135.08	319.85	330.85	333.86	333.85	-0.5150	0.0000	330
373	2.2913	135.11	319.86	330.86	333.88	333.87	-0.5006	0.0000	330
374	2.2898	135.14	319.87	330.86	333.90	333.89	-0.4863	0.0000	330
375	2.2883	135.18	319.88	330.87	333.92	333.91	-0.4719	0.0000	330
376	2.2868	135.21	319.88	330.87	333.94	333.93	-0.4575	0.0000	330
377	2.2854	135.24	319.89	330.87	333.96	333.94	-0.4431	0.0000	330
378	2.2840	135.27	319.90	330.88	333.97	333.96	-0.4288	0.0000	330
379	2.2826	135.31	319.91	330.88	333.99	333.98	-0.4144	0.0000	330
380	2.2813	135.34	319.92	330.89	334.01	334.00	-0.4000	0.0000	330
381	2.2800	135.37	319.93	330.89	334.03	334.02	-0.3950	0.0000	330
382	2.2787	135.41	319.94	330.89	334.05	334.04	-0.3900	0.0000	330
383	2.2774	135.44	319.94	330.90	334.07	334.06	-0.3850	0.0000	330
384	2.2761	135.48	319.95	330.90	334.09	334.08	-0.3800	0.0000	330
385	2.2748	135.51	319.96	330.90	334.11	334.10	-0.3750	0.0000	330
386	2.2736	135.54	319.97	330.91	334.13	334.12	-0.3500	0.0000	330

TABLE 19 (continued)

Time (sec)	Level (m)	Pressure (bar)	T1 (ºC)	T2 (ºC)	T3v (ºC)	T3l (ºC)	$\dot{m}$ Surge (kg/sec)	$\dot{m}$ Spray (kg/sec)	$\dot{Q}$ (kW)
387	2.2724	135.58	319.98	330.91	334.15	334.14	-0.3250	0.0000	330
388	2.2713	135.61	319.98	330.92	334.17	334.16	-0.3000	0.0000	330
389	2.2702	135.65	319.99	330.92	334.19	334.18	-0.2750	0.0000	330
390	2.2691	135.69	320.00	330.92	334.21	334.20	-0.2500	0.0000	330
391	2.2682	135.72	320.00	330.93	334.23	334.22	-0.2250	0.0000	330
392	2.2672	135.76	320.01	330.93	334.26	334.25	-0.2000	0.0000	330
393	2.2664	135.80	320.01	330.93	334.28	334.27	-0.1750	0.0000	330
394	2.2655	135.84	320.01	330.93	334.30	334.29	-0.1500	0.0000	330
395	2.2648	135.88	320.02	330.94	334.32	334.31	-0.1250	0.0000	330
396	2.2640	135.92	320.02	330.94	334.35	334.33	-0.1000	0.0000	330
397	2.2634	135.96	320.03	330.94	334.37	334.36	-0.0750	0.0000	330
398	2.2628	136.00	320.03	330.94	334.39	334.38	-0.0500	0.0000	330
399	2.2622	136.04	320.03	330.95	334.42	334.40	-0.0250	0.0000	330
400	2.2617	136.08	320.03	330.95	334.44	334.43	0.0000	0.0000	330
401	2.2613	136.12	320.03	330.95	334.46	334.45	0.0400	0.0000	330
402	2.2609	136.16	320.03	330.95	334.49	334.48	0.0800	0.0000	330
403	2.2606	136.21	320.03	330.95	334.51	334.50	0.1200	0.0000	330
404	2.2604	136.25	320.02	330.95	334.54	334.53	0.1600	0.0000	330
405	2.2602	136.29	320.01	330.95	334.56	334.55	0.2000	0.0000	330
406	2.2601	136.34	320.00	330.95	334.59	334.58	0.2350	0.0000	330
407	2.2601	136.38	319.99	330.95	334.62	334.60	0.2700	0.0000	330
408	2.2602	136.43	319.97	330.95	334.64	334.63	0.3050	0.0000	330
409	2.2603	136.48	319.96	330.95	334.67	334.66	0.3400	0.0000	330
410	2.2605	136.52	319.94	330.94	334.70	334.68	0.3750	0.0000	330
411	2.2607	136.57	319.92	330.94	334.72	334.71	0.4250	0.0000	330
412	2.2611	136.62	319.89	330.94	334.75	334.74	0.4750	0.0000	330
413	2.2616	136.67	319.87	330.93	334.78	334.77	0.5250	0.0000	330
414	2.2621	136.72	319.84	330.93	334.81	334.80	0.5750	0.0000	330
415	2.2627	136.77	319.80	330.92	334.84	334.82	0.6250	0.0000	330
416	2.2634	136.82	319.77	330.91	334.87	334.85	0.6500	0.0000	330
417	2.2642	136.87	319.73	330.91	334.89	334.88	0.6750	0.0000	330
418	2.2650	136.92	319.70	330.90	334.92	334.91	0.7000	0.0000	330
419	2.2659	136.97	319.66	330.89	334.95	334.94	0.7250	0.0000	330
420	2.2668	137.02	319.62	330.88	334.98	334.97	0.7500	0.0000	330
421	2.2677	137.08	319.58	330.87	335.01	335.00	0.8000	0.0000	330
422	2.2688	137.13	319.53	330.86	335.05	335.03	0.8500	0.0000	330
423	2.2700	137.18	319.49	330.85	335.08	335.06	0.9000	0.0000	330
424	2.2712	137.24	319.44	330.84	335.11	335.09	0.9500	0.0000	330
425	2.2726	137.29	319.38	330.83	335.14	335.13	1.0000	0.0000	330
426	2.2740	137.35	319.33	330.82	335.17	335.16	1.0500	0.0000	330
427	2.2755	137.41	319.27	330.80	335.20	335.19	1.1000	0.0000	330
428	2.2771	137.46	319.21	330.79	335.24	335.22	1.1500	0.0000	330
429	2.2787	137.52	319.15	330.77	335.27	335.26	1.2000	0.0000	330
430	2.2803	137.58	319.08	330.76	335.30	335.29	1.2500	0.0000	330
431	2.2821	137.64	319.01	330.74	335.34	335.32	1.3000	0.0000	330
432	2.2838	137.70	318.94	330.72	335.37	335.36	1.3500	0.0000	330
433	2.2857	137.76	318.87	330.70	335.41	335.39	1.4000	0.0000	330
434	2.2876	137.82	318.79	330.68	335.44	335.43	1.4500	0.0000	330
435	2.2895	137.89	318.71	330.66	335.48	335.46	1.5000	0.0000	330
436	2.2915	137.95	318.63	330.64	335.51	335.50	1.5500	0.0000	250

TABLE 19 (continued)

Time (sec)	Level (m)	Pressure (bar)	T1 (ºC)	T2 (ºC)	T3v (ºC)	T3l (ºC)	$\dot{m}$ Surge (kg/sec)	$\dot{m}$ Spray (kg/sec)	$\dot{Q}$ (kW)
437	2.2935	138.00	318.54	330.62	335.54	335.53	1.6000	0.0000	250
438	2.2956	138.06	318.46	330.60	335.57	335.56	1.6500	0.0000	250
439	2.2977	138.11	318.37	330.57	335.61	335.59	1.7000	0.0000	250
440	2.2999	138.17	318.28	330.55	335.65	335.62	1.7500	0.0000	250
441	2.3022	138.23	318.18	330.52	335.70	335.66	1.8250	0.0000	250
442	2.3046	138.29	318.08	330.49	335.75	335.69	1.9000	0.0000	250
443	2.3071	138.35	317.98	330.46	335.80	335.73	1.9750	0.0000	250
444	2.3097	138.41	317.87	330.43	335.85	335.76	2.0500	0.0000	250
445	2.3124	138.47	317.76	330.40	335.90	335.80	2.1250	0.0000	250
446	2.3151	138.54	317.65	330.37	335.95	335.83	2.2000	0.0000	250
447	2.3180	138.60	317.53	330.33	336.00	335.87	2.2750	0.0000	250
448	2.3209	138.67	317.41	330.29	336.05	335.91	2.3500	0.0000	250
449	2.3239	138.73	317.28	330.26	336.11	335.95	2.4250	0.0000	250
450	2.3270	138.80	317.16	330.22	336.16	335.99	2.5000	0.0000	250
451	2.3303	138.87	317.02	330.17	336.22	336.02	2.6000	0.0000	250
452	2.3336	138.94	316.89	330.13	336.28	336.06	2.7000	0.0000	0
453	2.3367	138.98	316.74	330.08	336.31	336.09	2.8000	0.0000	0
454	2.3400	139.02	316.60	330.03	336.34	336.11	2.9000	0.0000	0
455	2.3434	139.06	316.44	329.98	336.37	336.13	3.0000	0.0000	0
456	2.3469	139.10	316.29	329.93	336.40	336.15	3.0750	0.0000	0
457	2.3504	139.19	316.13	329.87	336.48	336.17	3.1500	0.0000	0
458	2.3540	139.35	315.97	329.82	336.61	336.17	3.2250	0.0000	0
459	2.3577	139.51	315.81	329.77	336.74	336.17	3.3000	0.0000	0
460	2.3615	139.68	315.64	329.71	336.87	336.17	3.3750	0.0000	0
461	2.3654	139.85	315.48	329.66	337.01	336.16	3.5000	0.0000	0
462	2.3694	140.03	315.30	329.60	337.15	336.16	3.6250	0.0000	0
463	2.3736	140.22	315.12	329.54	337.30	336.16	3.7500	0.0000	0
464	2.3779	140.41	314.94	329.47	337.45	336.16	3.8750	0.0000	0
465	2.3823	140.61	314.75	329.40	337.61	336.16	4.0000	0.0000	0
466	2.3869	140.82	314.55	329.33	337.77	336.16	4.1500	0.0000	0
467	2.3917	141.03	314.35	329.26	337.94	336.15	4.3000	0.0000	0
468	2.3966	141.25	314.14	329.18	338.12	336.15	4.4500	0.0000	0
469	2.4018	141.49	313.93	329.10	338.30	336.15	4.6000	0.0000	0
470	2.4071	141.73	313.71	329.01	338.49	336.15	4.7500	0.0000	0
471	2.4125	141.97	313.49	328.92	338.69	336.14	4.8000	0.0000	0
472	2.4179	142.22	313.26	328.83	338.89	336.14	4.8500	0.0000	0
473	2.4235	142.48	313.04	328.74	339.08	336.14	4.9000	0.0000	0
474	2.4290	142.73	312.81	328.65	339.29	336.13	4.9500	3.5134	0
475	2.4399	142.64	312.58	328.53	339.22	336.11	5.0000	3.5049	0
476	2.4500	142.61	312.35	328.42	339.20	336.10	5.0200	3.5272	0
477	2.4610	142.52	312.11	328.31	339.13	336.08	5.0400	3.5221	0
478	2.4719	142.43	311.88	328.19	339.07	336.06	5.0600	3.5103	0
479	2.4829	142.34	311.65	328.07	339.00	336.04	5.0800	3.5552	0
480	2.4939	142.25	311.42	327.95	338.94	336.02	5.1000	3.5527	0
481	2.5050	142.16	311.19	327.83	338.88	336.00	5.0800	3.5029	0
482	2.5159	142.07	310.96	327.71	338.81	335.98	5.0600	3.5093	0
483	2.5268	141.98	310.74	327.59	338.75	335.96	5.0400	3.5109	0
484	2.5377	141.89	310.51	327.47	338.68	335.94	5.0200	3.5083	0
485	2.5486	141.80	310.29	327.35	338.61	335.92	5.0000	3.5081	0
486	2.5594	141.70	310.08	327.23	338.54	335.90	4.9500	3.5044	0

TABLE 19 (continued)

Time (sec)	Level (m)	Pressure (bar)	T1 (ºC)	T2 (ºC)	T3v (ºC)	T3l (ºC)	ṁ Surge (kg/sec)	ṁ Spray (kg/sec)	Q̇ (kW)
487	2.5701	141.61	309.86	327.11	338.47	335.88	4.9000	3.5282	0
488	2.5808	141.51	309.65	326.99	338.40	335.85	4.8500	3.5480	0
489	2.5884	141.63	309.45	326.88	338.49	335.84	4.8000	3.5479	0
490	2.5990	141.52	309.25	326.76	338.41	335.82	4.7500	3.4811	0
491	2.6094	141.41	309.05	326.64	338.33	335.80	4.6750	3.5183	0
492	2.6198	141.30	308.86	326.52	338.25	335.78	4.6000	3.5250	0
493	2.6301	141.18	308.67	326.40	338.16	335.75	4.5250	3.5371	0
494	2.6403	141.05	308.48	326.29	338.07	335.73	4.4500	3.5301	0
495	2.6487	141.05	308.30	326.18	338.07	335.71	4.3750	3.5456	0
496	2.6571	141.05	308.12	326.07	338.06	335.70	4.3250	3.5186	0
497	2.6654	141.04	307.95	325.96	338.06	335.68	4.2750	3.5262	0
498	2.6737	141.02	307.78	325.86	338.05	335.67	4.2250	3.5290	0
499	2.6806	141.11	307.62	325.75	338.13	335.65	4.1750	3.5620	0
500	2.6904	140.97	307.45	325.64	338.02	335.63	4.1250	3.5620	0
501	2.6984	140.95	307.29	325.54	338.00	335.61	4.0500	3.5628	0
502	2.7064	140.92	307.13	325.44	337.98	335.60	3.9750	3.5644	0
503	2.7143	140.89	306.98	325.34	337.96	335.58	3.9000	3.5675	0
504	2.7208	140.96	306.83	325.24	338.02	335.57	3.8250	3.5664	0
505	2.7274	141.01	306.69	325.15	338.06	335.56	3.7500	3.3815	0
506	2.7340	141.05	306.55	325.05	338.10	335.54	3.6750	3.2129	0
507	2.7415	141.01	306.41	324.96	338.07	335.53	3.6000	3.5281	0
508	2.7489	140.96	306.27	324.86	338.03	335.51	3.5250	3.5844	0
509	2.7563	140.90	306.14	324.77	337.99	335.49	3.4500	3.5865	0
510	2.7625	140.92	306.02	324.68	338.01	335.48	3.3750	3.5865	0
511	2.7697	140.86	305.89	324.59	337.96	335.46	3.3250	3.5896	0
512	2.7755	140.90	305.77	324.51	337.99	335.45	3.2750	3.5912	0
513	2.7826	140.82	305.65	324.42	337.94	335.44	3.2250	3.5934	0
514	2.7875	140.92	305.53	324.34	338.02	335.43	3.1750	3.5966	0
515	2.7945	140.84	305.42	324.26	337.96	335.41	3.1250	3.5965	0
516	2.7992	140.94	305.31	324.18	338.03	335.40	3.0500	3.6012	0
517	2.8060	140.85	305.20	324.10	337.97	335.38	2.9750	3.6011	0
518	2.8106	140.93	305.09	324.02	338.04	335.38	2.9000	3.6069	0
519	2.8173	140.83	304.99	323.94	337.96	335.36	2.8250	3.6067	0
520	2.8217	140.91	304.89	323.87	338.02	335.35	2.7500	3.6120	0
521	2.8281	140.80	304.79	323.80	337.94	335.34	2.7000	3.6118	0
522	2.8324	140.87	304.70	323.73	338.00	335.33	2.6500	3.6158	0
523	2.8367	140.94	304.61	323.66	338.05	335.32	2.6000	3.6188	0
524	2.8429	140.82	304.51	323.59	337.96	335.30	2.5500	3.6186	0
525	2.8471	140.88	304.43	323.53	338.01	335.30	2.5000	3.6225	0
526	2.8511	140.94	304.34	323.46	338.06	335.29	2.4250	3.6249	0
527	2.8562	140.89	304.26	323.40	338.03	335.28	2.3500	3.6211	0
528	2.8601	140.94	304.18	323.34	338.06	335.27	2.2750	3.6310	0
529	2.8660	140.80	304.10	323.27	337.96	335.25	2.2000	3.6308	0
530	2.8696	140.84	304.02	323.21	337.99	335.24	2.1250	3.6345	0
531	2.8732	140.87	303.95	323.16	338.02	335.24	2.0500	3.6367	0
532	2.8768	140.90	303.88	323.11	338.04	335.23	1.9750	3.6402	0
533	2.8802	140.93	303.82	323.06	338.06	335.22	1.9000	3.6435	0
534	2.8856	140.76	303.75	323.00	337.94	335.21	1.8250	3.6433	0
535	2.8888	140.78	303.69	322.95	337.95	335.20	1.7500	3.6465	0
536	2.8920	140.79	303.63	322.91	337.96	335.19	1.6750	3.6477	0

TABLE 19 (continued)

Time (sec)	Level (m)	Pressure (bar)	T1 (ºC)	T2 (ºC)	T3v (ºC)	T3l (ºC)	ṁ Surge (kg/sec)	ṁ Spray (kg/sec)	Q̇ (kW)
537	2.8951	140.79	303.58	322.86	337.96	335.19	1.6000	3.6505	0
538	2.8980	140.79	303.53	322.82	337.97	335.18	1.5250	3.6531	0
539	2.9009	140.79	303.48	322.78	337.96	335.17	1.4500	3.6555	0
540	2.9037	140.78	303.43	322.74	337.96	335.17	1.3750	3.6578	0
541	2.9064	140.77	303.38	322.71	337.95	335.16	1.3000	3.6600	0
542	2.9090	140.75	303.34	322.67	337.94	335.15	1.2250	3.6620	0
543	2.9116	140.73	303.30	322.64	337.92	335.15	1.1500	3.6638	0
544	2.9140	140.70	303.26	322.61	337.90	335.14	1.0750	3.6656	0
545	2.9163	140.66	303.23	322.58	337.87	335.13	1.0000	3.6671	0
546	2.9186	140.63	303.19	322.55	337.84	335.13	0.9750	3.6686	0
547	2.9209	140.59	303.16	322.52	337.82	335.12	0.9500	3.6687	0
548	2.9231	140.55	303.13	322.50	337.79	335.12	0.9250	3.6686	0
549	2.9253	140.51	303.10	322.47	337.75	335.11	0.9000	3.6685	0
550	2.9274	140.47	303.07	322.44	337.72	335.10	0.8750	3.6683	0
551	2.9295	140.42	303.04	322.42	337.69	335.10	0.7750	3.6681	0
552	2.9315	140.37	303.02	322.40	337.65	335.09	0.6750	3.6698	0
553	2.9333	140.31	302.99	322.38	337.60	335.09	0.5750	3.6713	0
554	2.9350	140.24	302.98	322.36	337.55	335.08	0.4750	3.6728	0
555	2.9366	140.17	302.96	322.35	337.49	335.07	0.3750	3.6740	0
556	2.9381	140.09	302.95	322.33	337.43	335.07	0.3250	3.6750	0
557	2.9396	140.01	302.94	322.32	337.37	335.06	0.2750	3.6743	0
558	2.9409	139.93	302.93	322.31	337.30	335.06	0.2250	3.6737	0
559	2.9423	139.84	302.92	322.30	337.24	335.05	0.1750	3.6729	0
560	2.9435	139.75	302.91	322.30	337.16	335.05	0.1250	3.6721	0
561	2.9447	139.66	302.91	322.29	337.09	335.04	0.0750	3.6709	0
562	2.9458	139.57	302.90	322.28	337.02	335.04	0.0250	0.0000	0
563	2.9458	139.57	302.90	322.28	337.02	335.04	-0.0250	0.0000	0
564	2.9458	139.56	302.90	322.28	337.01	335.04	-0.0750	0.0000	0
565	2.9456	139.55	302.91	322.29	337.01	335.04	-0.1250	0.0000	0
566	2.9454	139.54	302.91	322.29	337.00	335.04	-0.1750	0.0000	0
567	2.9452	139.53	302.92	322.29	336.99	335.04	-0.2250	0.0000	0
568	2.9448	139.51	302.92	322.30	336.97	335.04	-0.2750	0.0000	0
569	2.9444	139.49	302.93	322.30	336.96	335.03	-0.3250	0.0000	0
570	2.9439	139.47	302.94	322.31	336.94	335.03	-0.3750	0.0000	0
571	2.9434	139.44	302.95	322.31	336.91	335.03	-0.4200	0.0000	0
572	2.9428	139.41	302.96	322.32	336.89	335.03	-0.4650	0.0000	0
573	2.9422	139.37	302.98	322.33	336.86	335.03	-0.5100	0.0000	0
574	2.9414	139.34	302.99	322.34	336.83	335.02	-0.5550	0.0000	0
575	2.9407	139.29	303.01	322.35	336.80	335.02	-0.6000	0.0000	0
576	2.9398	139.25	303.03	322.36	336.76	335.02	-0.6400	0.0000	0
577	2.9389	139.20	303.04	322.37	336.73	335.02	-0.6800	0.0000	0
578	2.9380	139.16	303.06	322.38	336.69	335.01	-0.7200	0.0000	0
579	2.9370	139.10	303.08	322.39	336.65	335.01	-0.7600	0.0000	0
580	2.9359	139.05	303.10	322.41	336.60	335.01	-0.8000	0.0000	0
581	2.9348	138.99	303.13	322.42	336.56	335.01	-0.8400	0.0000	0
582	2.9337	138.93	303.15	322.43	336.51	335.00	-0.8800	0.0000	0
583	2.9324	138.87	303.18	322.45	336.46	335.00	-0.9200	0.0000	0
584	2.9312	138.80	303.20	322.46	336.40	334.99	-0.9600	0.0000	0
585	2.9299	138.73	303.23	322.48	336.35	334.99	-1.0000	0.0000	0
586	2.9285	138.66	303.26	322.50	336.29	334.99	-1.0250	0.0000	0

TABLE 19 (continued)

Time (sec)	Level (m)	Pressure (bar)	T1 (ºC)	T2 (ºC)	T3v (ºC)	T3l (ºC)	$\dot{m}$ Surge (kg/sec)	$\dot{m}$ Spray (kg/sec)	$\dot{Q}$ (kW)
587	2.9271	138.59	303.28	322.51	336.23	334.98	-1.0500	0.0000	0
588	2.9257	138.52	303.31	322.53	336.17	334.98	-1.0750	0.0000	0
589	2.9242	138.44	303.34	322.55	336.11	334.97	-1.1000	0.0000	0
590	2.9227	138.36	303.37	322.57	336.05	334.97	-1.1250	0.0000	0
591	2.9211	138.28	303.40	322.58	335.99	334.96	-1.1500	0.0000	0
592	2.9196	138.20	303.44	322.60	335.92	334.96	-1.1750	0.0000	0
593	2.9180	138.12	303.47	322.62	335.86	334.95	-1.2000	0.0000	0
594	2.9163	138.04	303.50	322.64	335.79	334.95	-1.2250	0.0000	0
595	2.9147	137.95	303.53	322.66	335.72	334.95	-1.2500	0.0000	0
596	2.9130	137.86	303.57	322.68	335.65	334.94	-1.2500	0.0000	0
597	2.9113	137.78	303.60	322.70	335.58	334.94	-1.2500	0.0000	0
598	2.9096	137.69	303.64	322.72	335.51	334.93	-1.2500	0.0000	0
599	2.9079	137.60	303.67	322.74	335.44	334.93	-1.2500	0.0000	0
600	2.9062	137.52	303.70	322.76	335.37	334.92	-1.2500	0.0000	0

Appendix C.2: System Inputs for Pressurizer

In addition to the simulation surge mass flow rate provided in FIGURE 18, five additional inputs were used with each of the pressurizer control methods investigated (i.e., a total of six inputs were used), as shown below:

- Input 1 – Shippingport PZR Surge Mass Flow Rate – FIGURE 127
- Input 2 – One Large Insurge Mass Flow Rate – FIGURE 128
- Input 3 – One Large Outsurge Mass Flow Rate – FIGURE 129
- Input 4 – One Insurge/Outsurge Mass Flow Rate – FIGURE 130
- Input 5 – Three Insurge/Outsurge Mass Flow Rate – FIGURE 131
- Input 6 – Shippingport PZR Surge with Noise – FIGURE 132

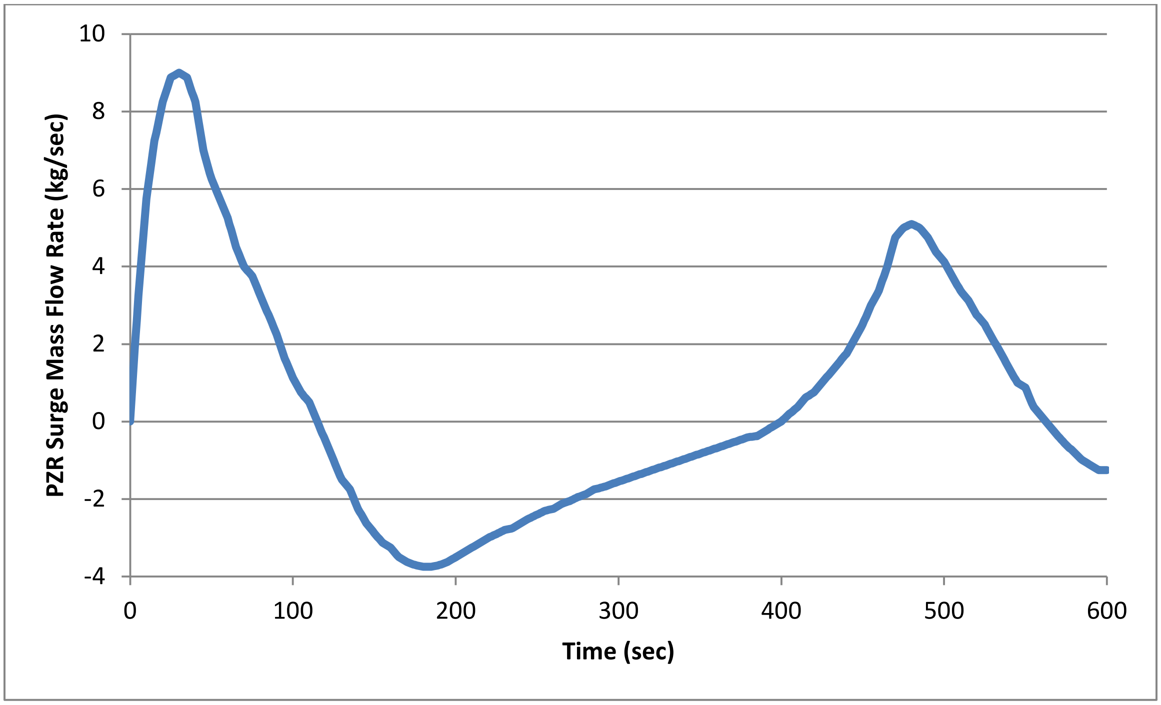


FIGURE 127: Test case PZR surge data – input 1 (Shippingport PZR surge)

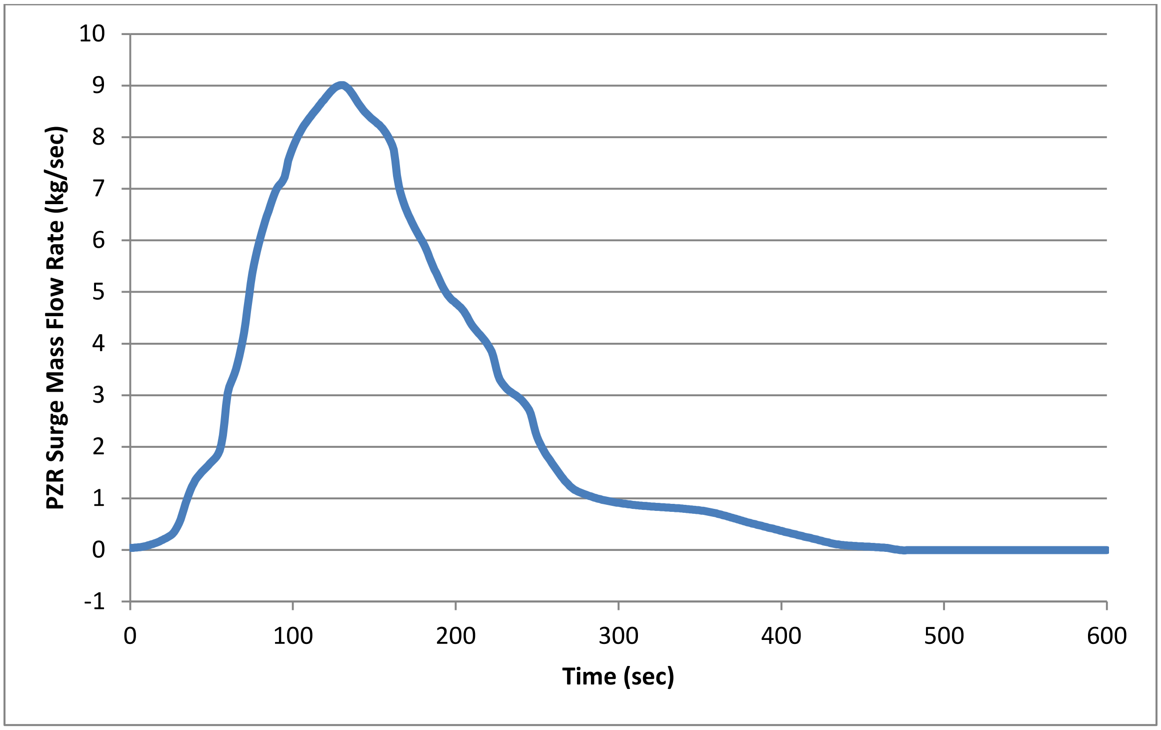


FIGURE 128: Test case PZR surge data – input 2 (one insurge)

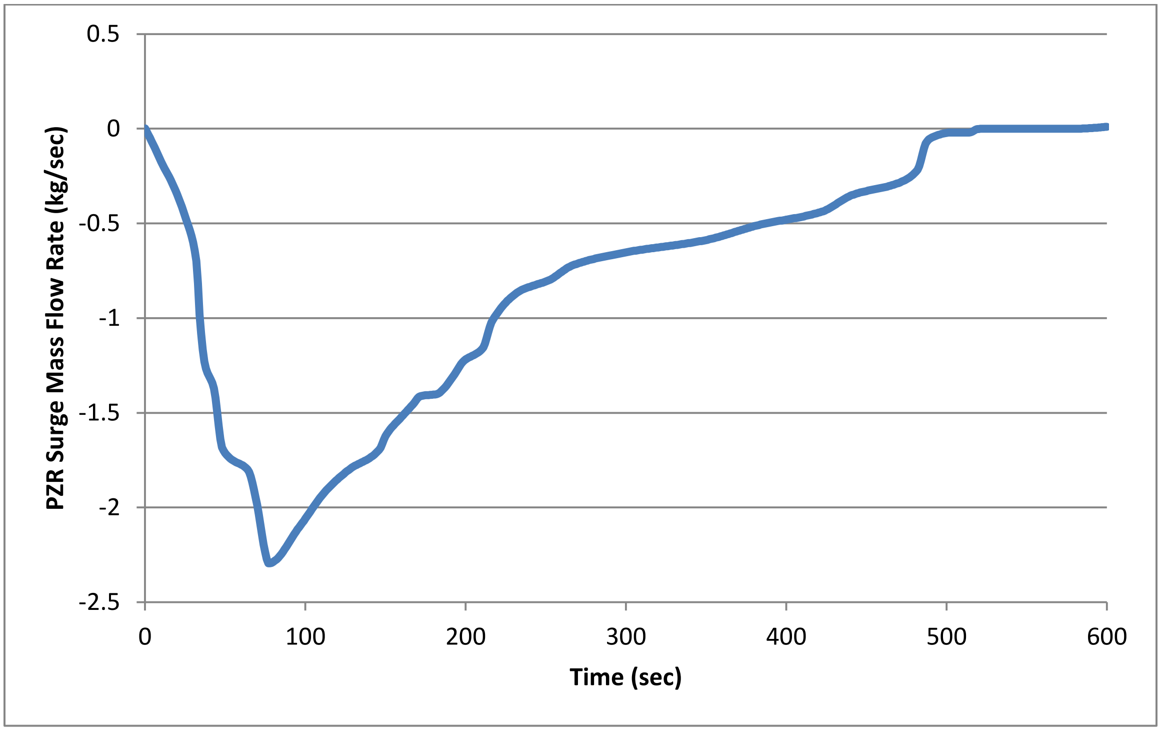


FIGURE 129: Test case PZR surge data – Input 3 (one outsurge)

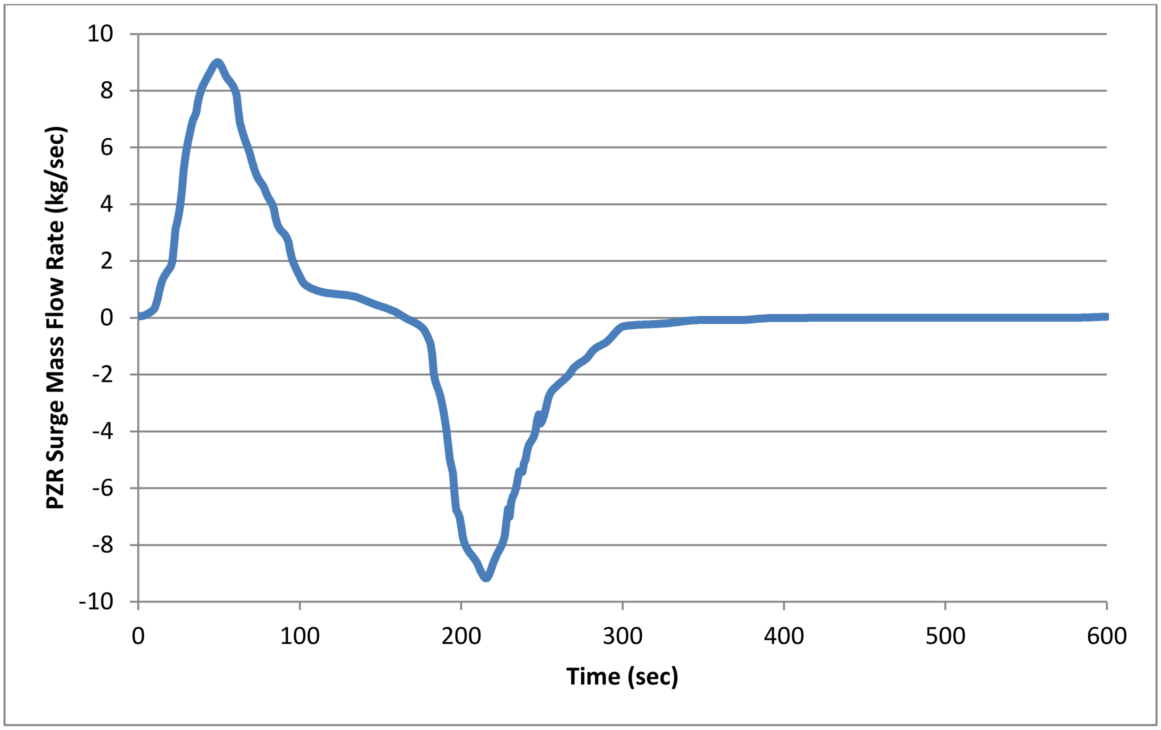


FIGURE 130: Test case PZR surge data – input 4 (one in/out surge)

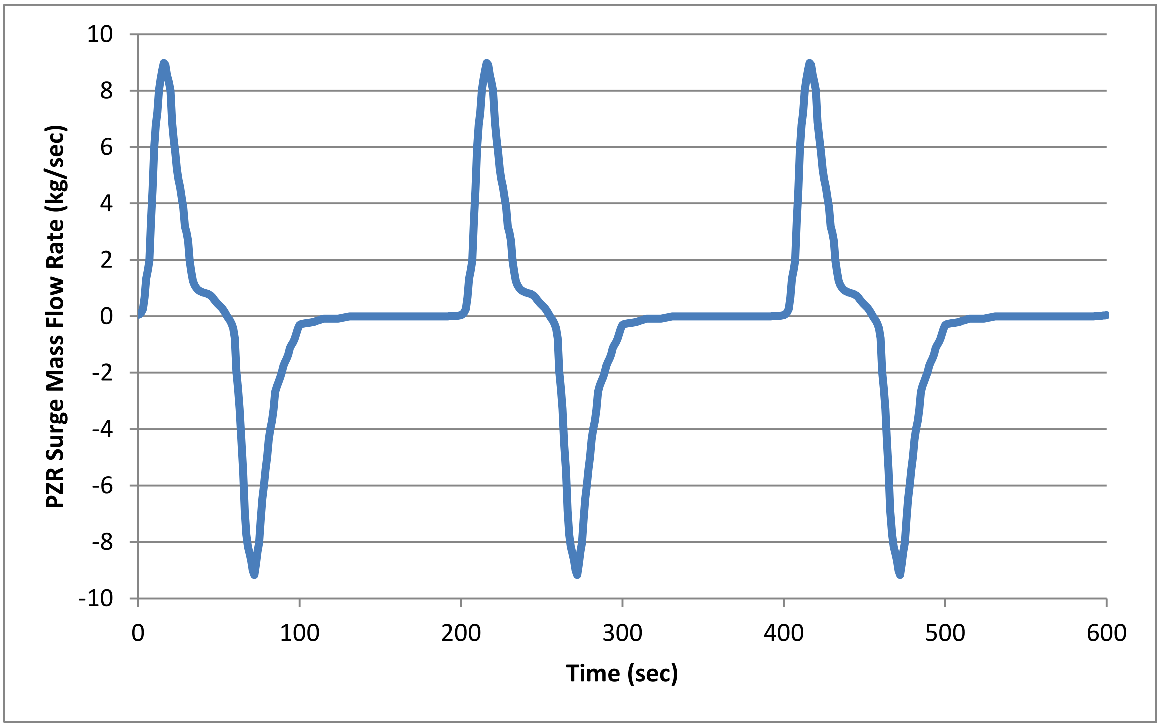


FIGURE 131: Test case PZR surge data – input 5 (three in/out surges)

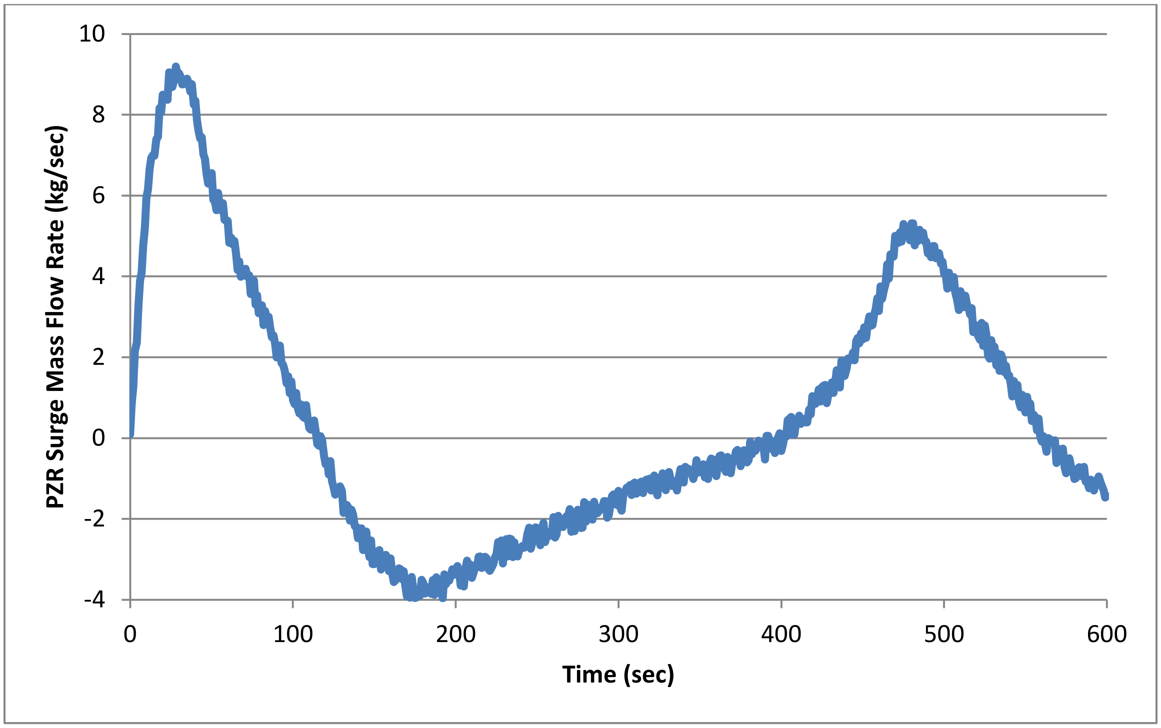


FIGURE 132: Test case PZR surge data – input 6 (Shippingport surge with noise)

Appendix C.3: Simulation Result Plots

This appendix section provides the simulation output results for the dynamic PZR model using the six PZR surge mass flow rate inputs (see APPENDIX C). For comparison, this data demonstrates the behavior of the PZR using each of the control methods. Performance plots for PZR pressure and PZR spray mass flow rate are provided for the following control methods (see the list shown below). The open loop and conventional control plots are on the same figures and the other adaptive control techniques (i.e., LQR, MV, LQR with RICK, MV with RICK, LiMe, LiMeRICK, and LiMeRICK P.H.) are on the same plots.

- Open Loop
- Conventional Control
- LQR
- MV
- LQR with RICK
- MV with RICK
- LiMe
- LiMeRICK
- LiMeRICK with Predictive Horizon (LiMeRICK P.H.)

Open Loop and Conventional Control Performance Plots:

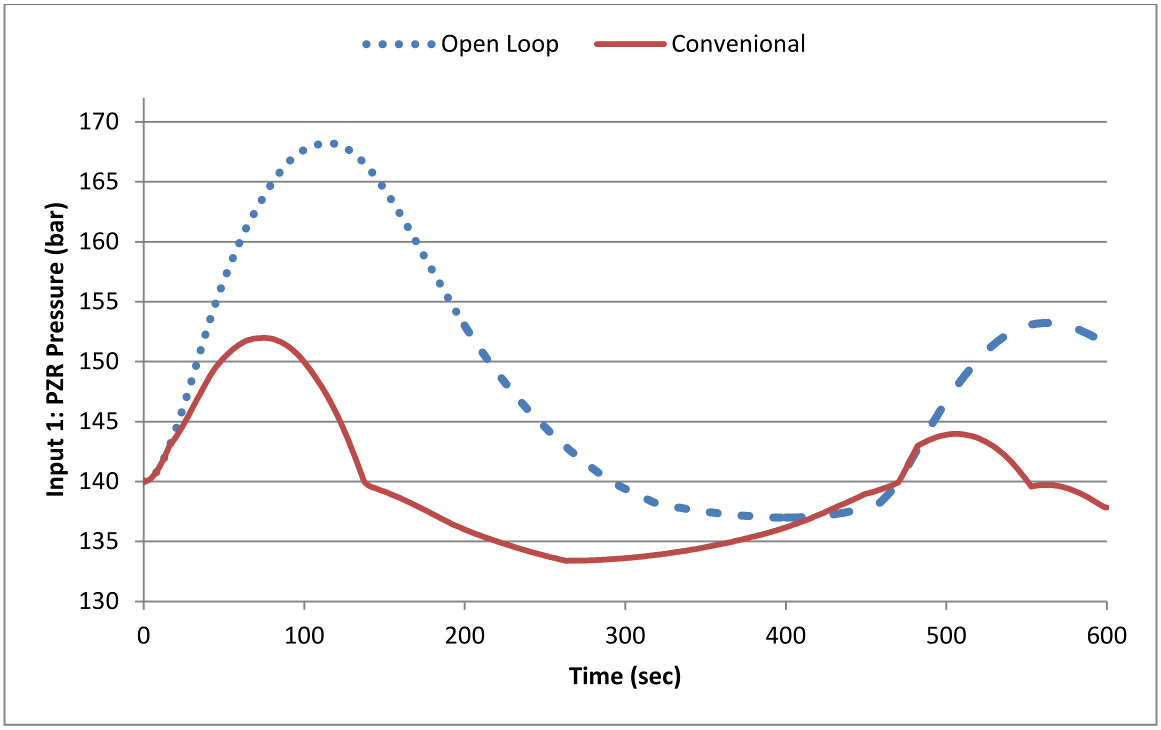


FIGURE 133: Input 1 – PZR pressure (open loop and conventional control)

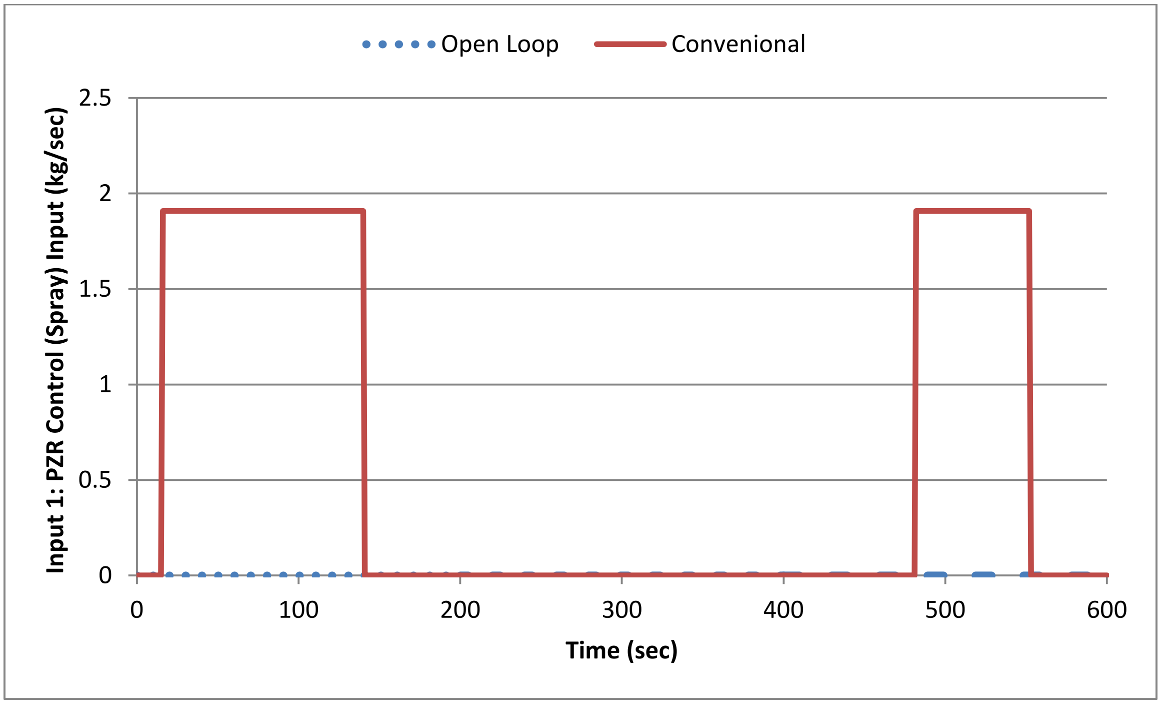


FIGURE 134: Input 1 – PZR spray (open loop and conventional control)

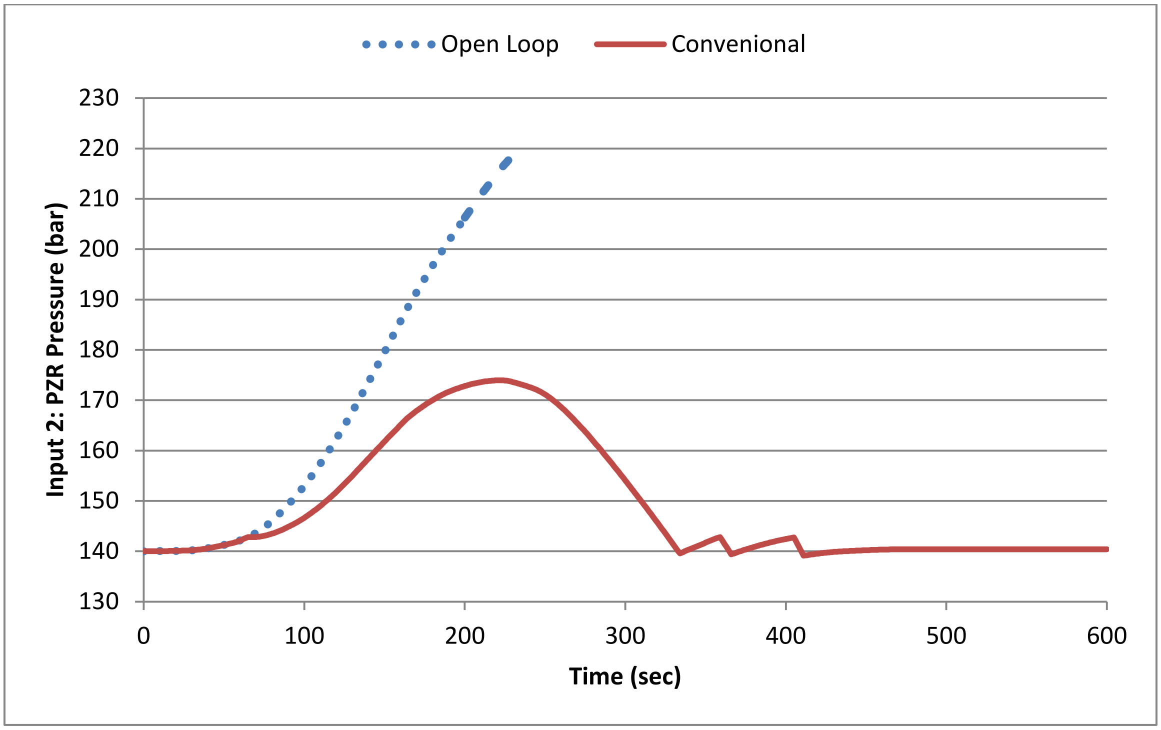


FIGURE 135: Input 2 – PZR pressure (open loop and conventional control)

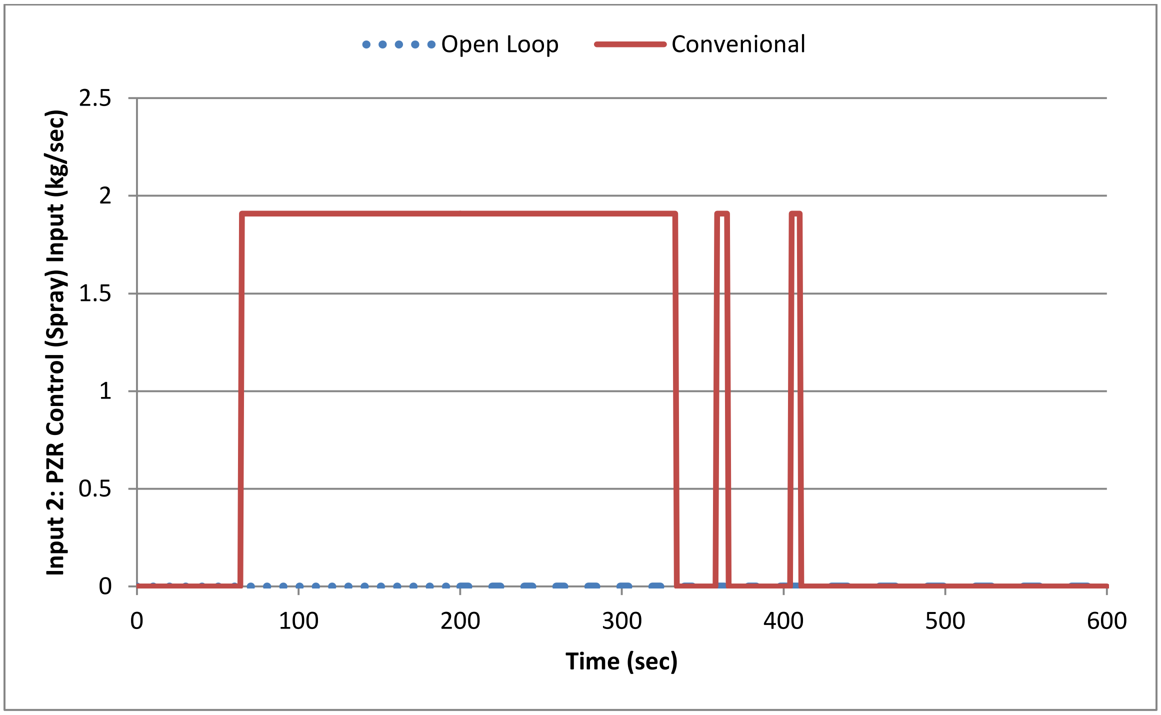


FIGURE 136: Input 2 – PZR spray (open loop and conventional control)

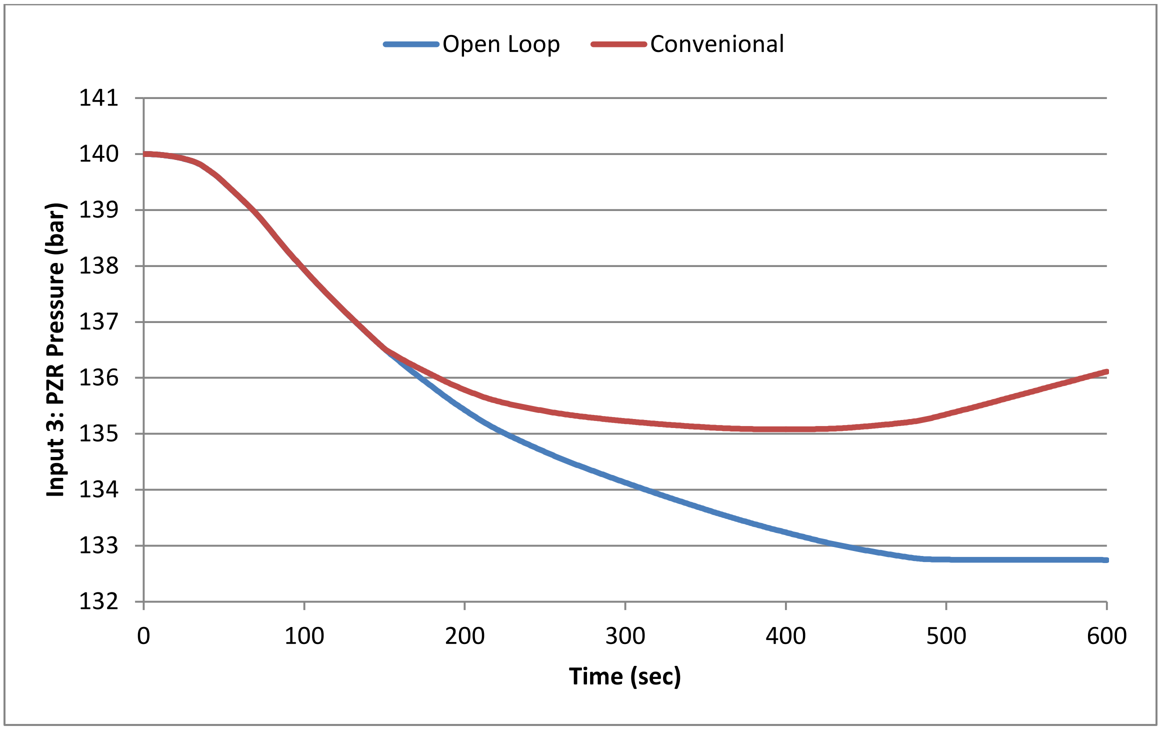


FIGURE 137: Input 3 – PZR pressure (open loop and conventional control)

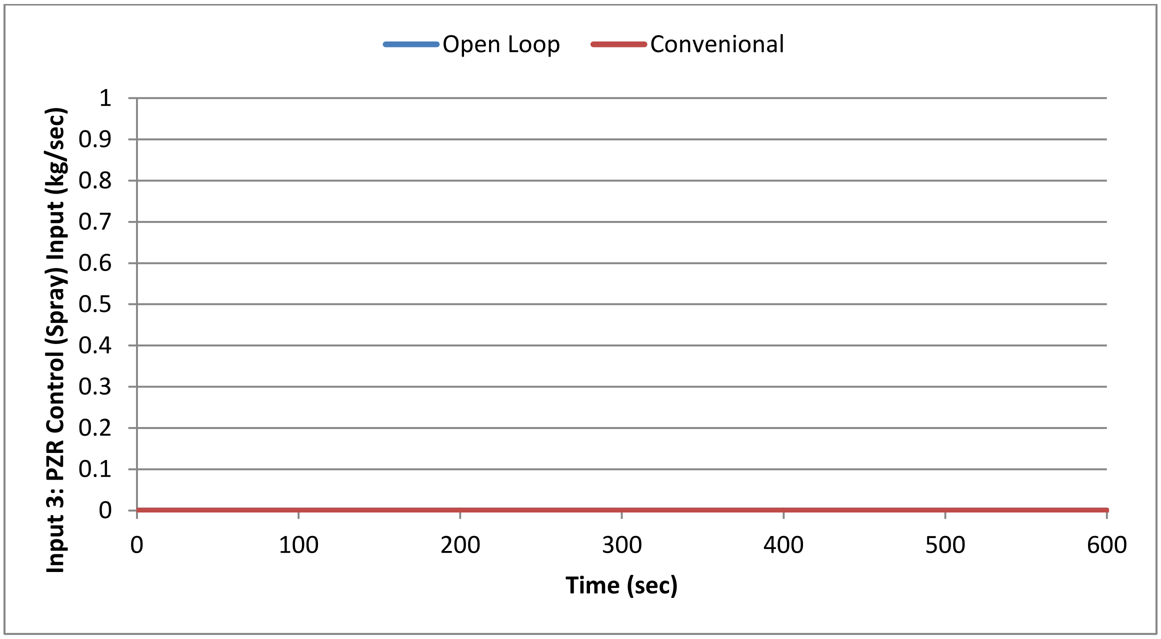


FIGURE 138: Input 3 – PZR spray (open loop and conventional control)

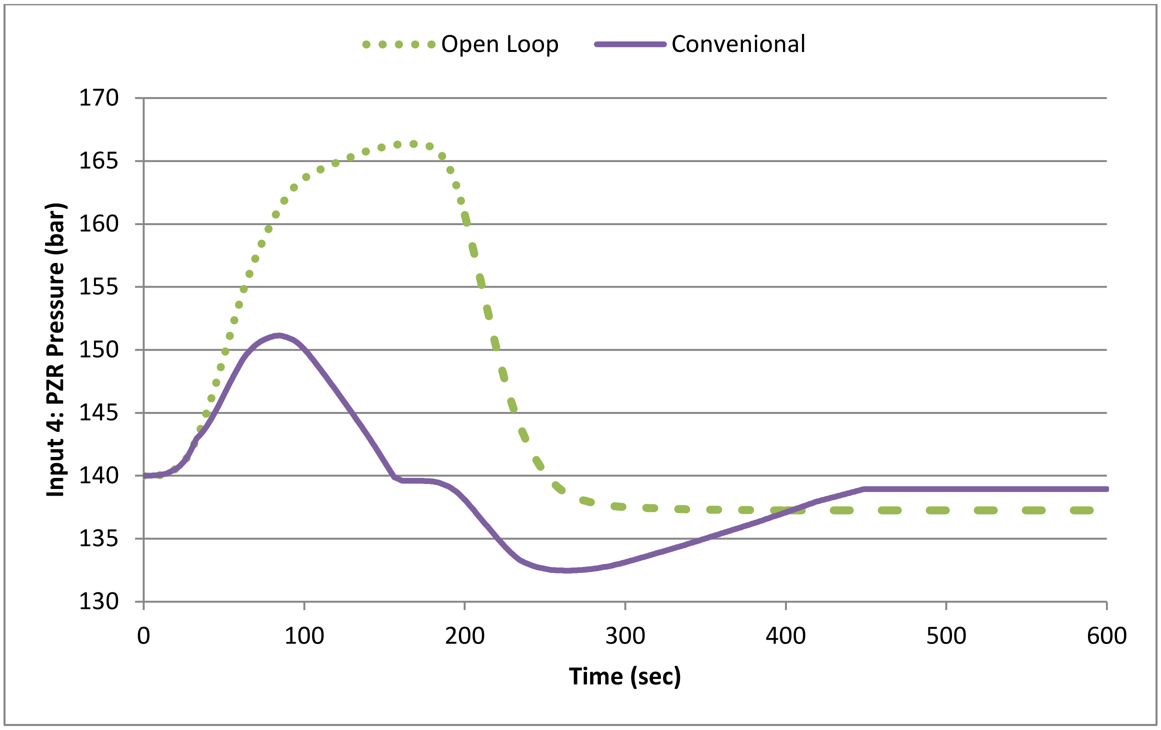


FIGURE 139: Input 4 – PZR pressure (open loop and conventional control)

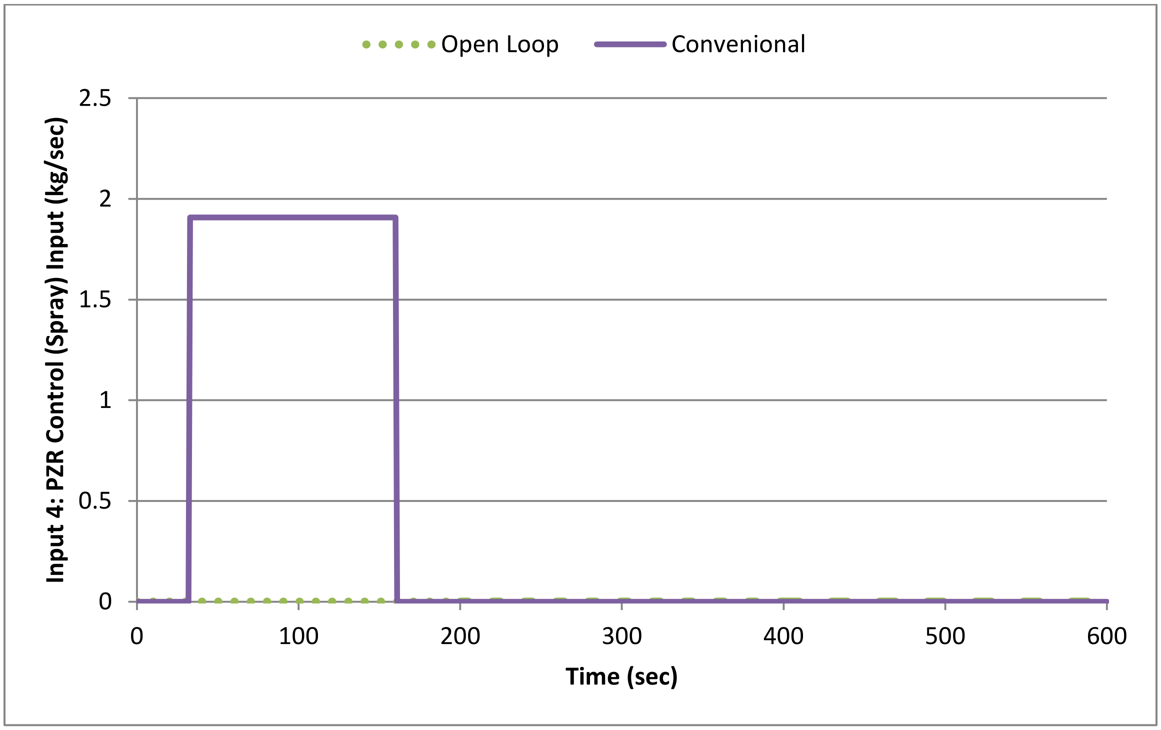


FIGURE 140: Input 4 – PZR spray (open loop and conventional control)

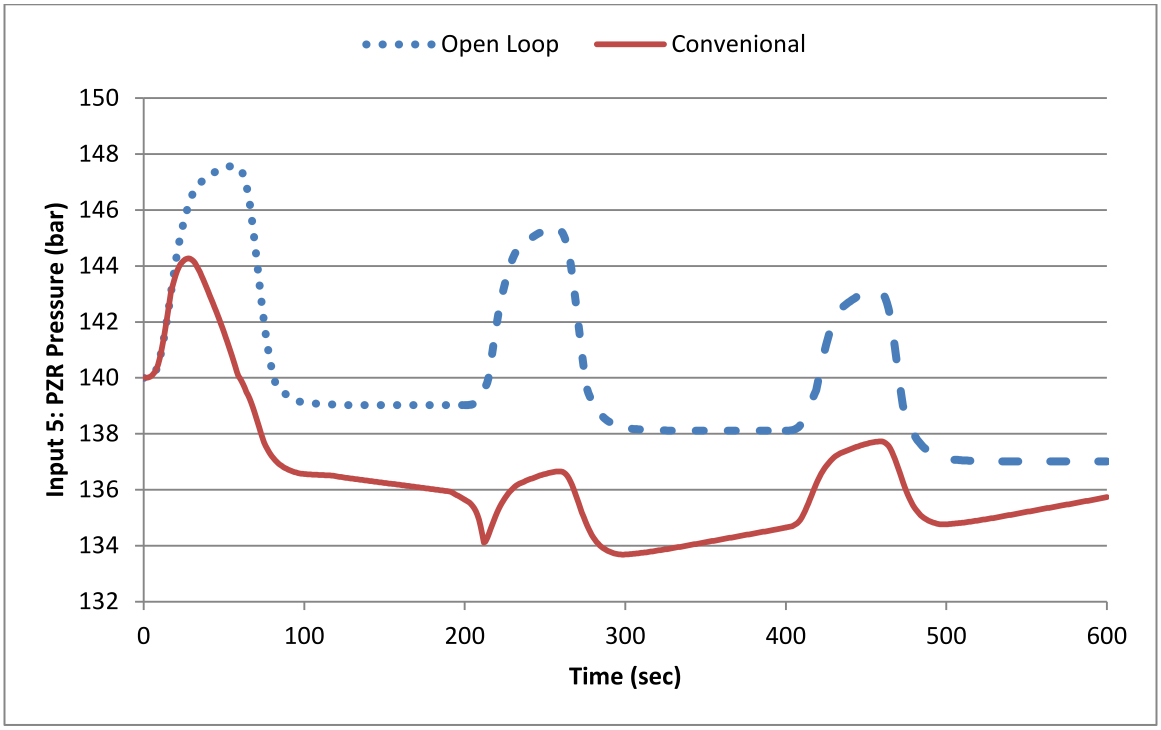


FIGURE 141: Input 5 – PZR pressure (open loop and conventional control)

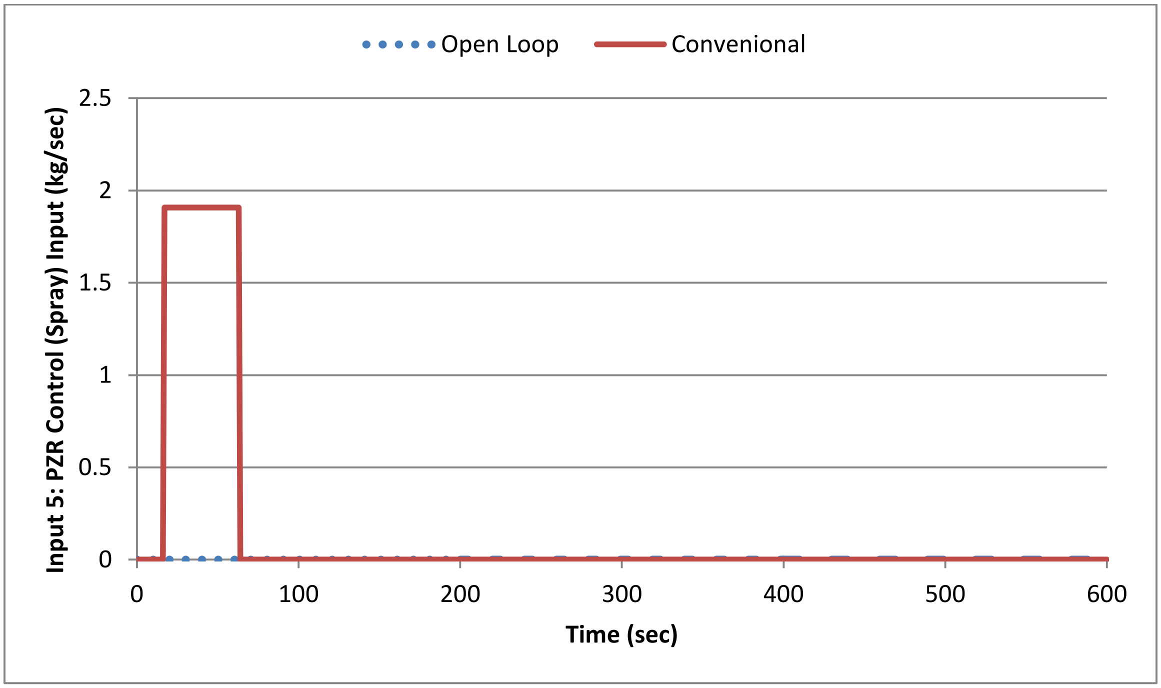


FIGURE 142: Input 5 – PZR spray (open loop and conventional control)

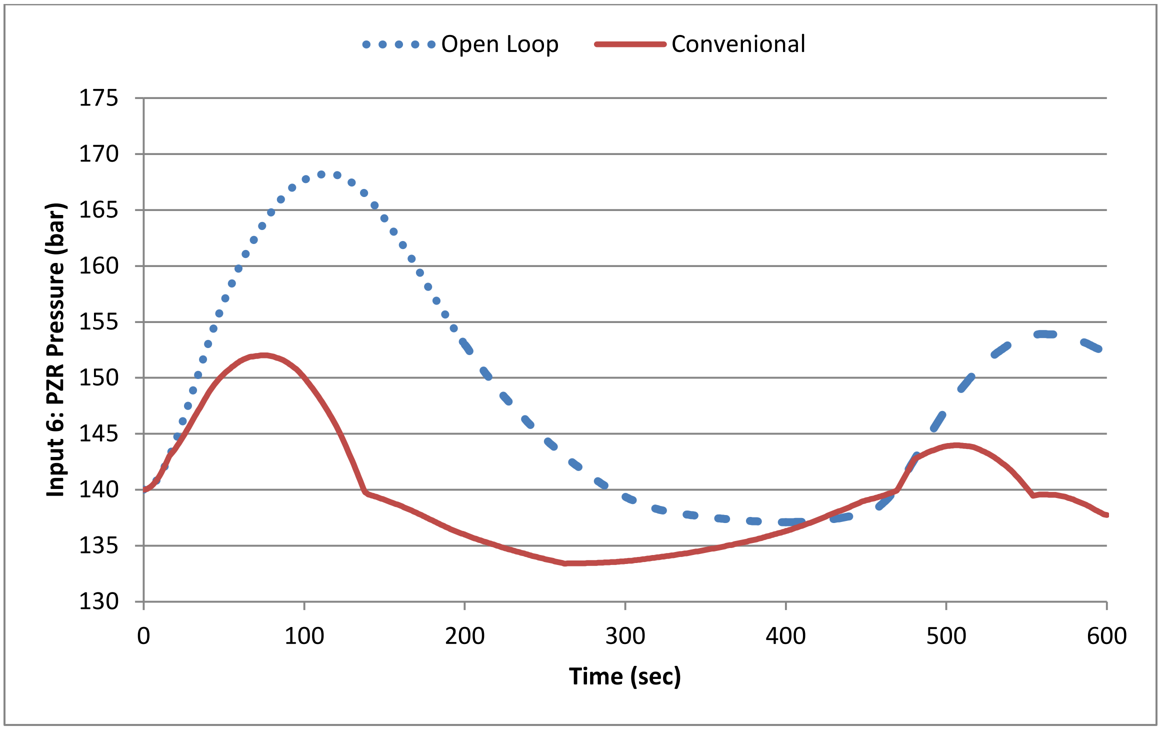


FIGURE 143: Input 6 – PZR pressure (open loop and conventional control)

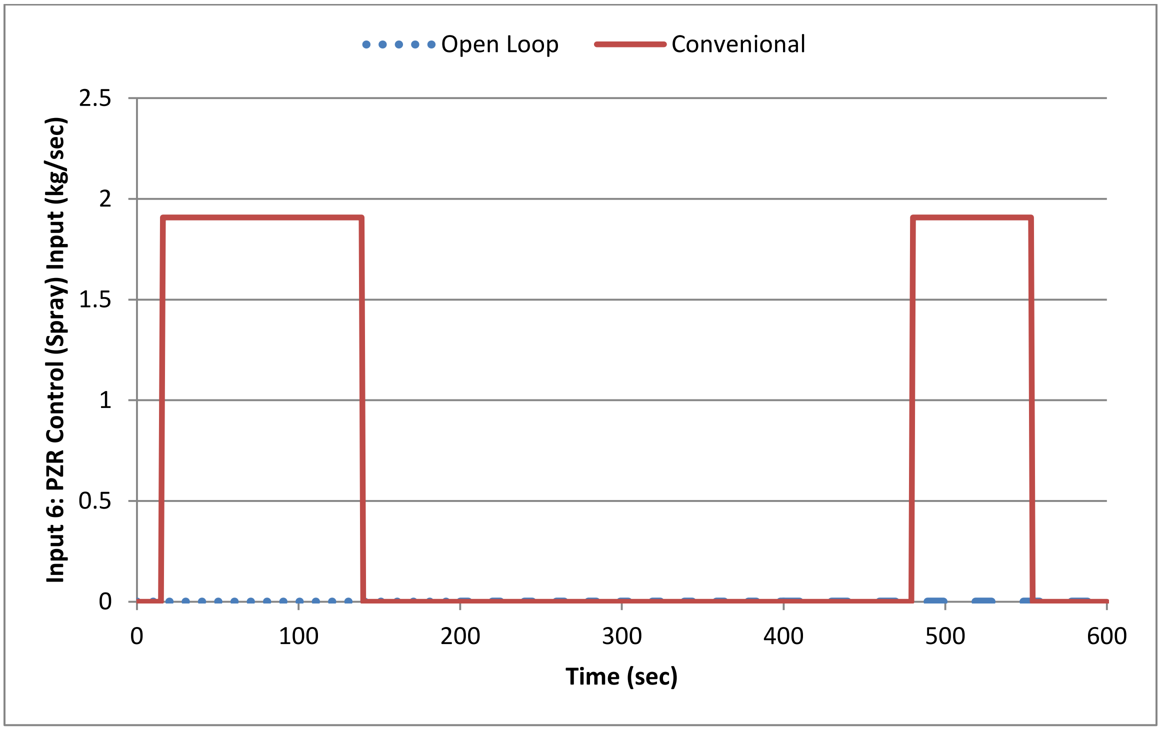


FIGURE 144: Input 6 – PZR spray (open loop and conventional control)

Adaptive Control Performance Plots:

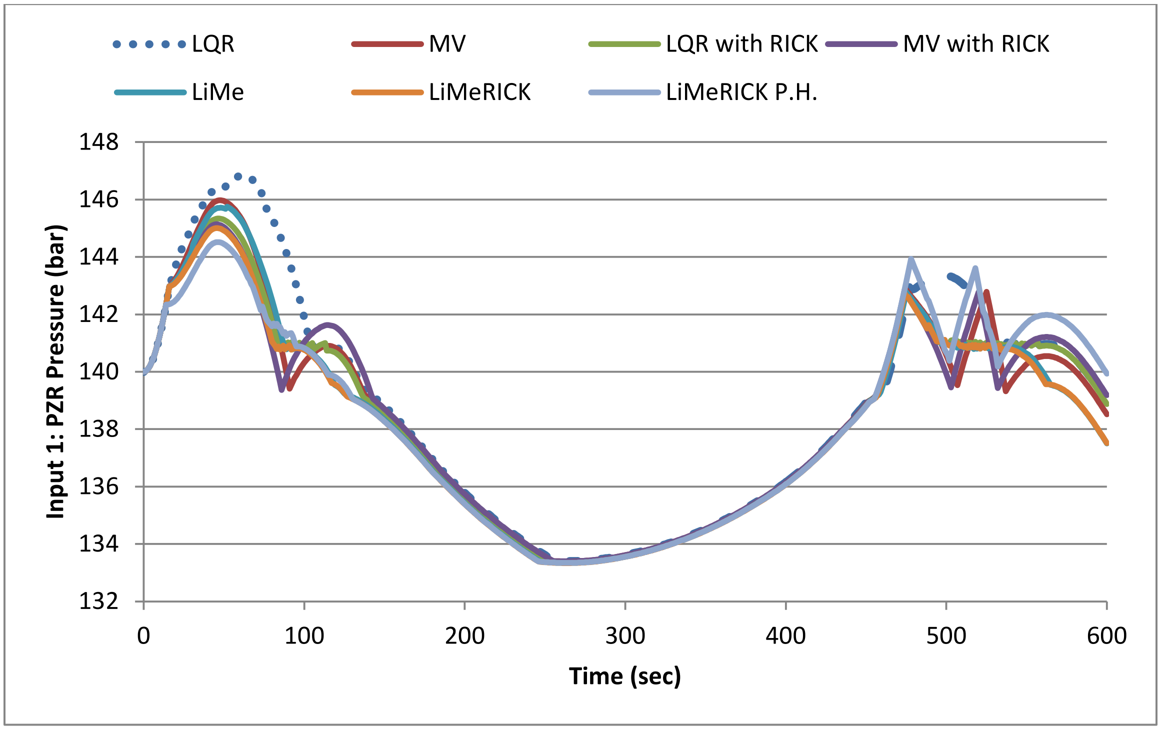


FIGURE 145: Input 1 – PZR pressure (adaptive control)

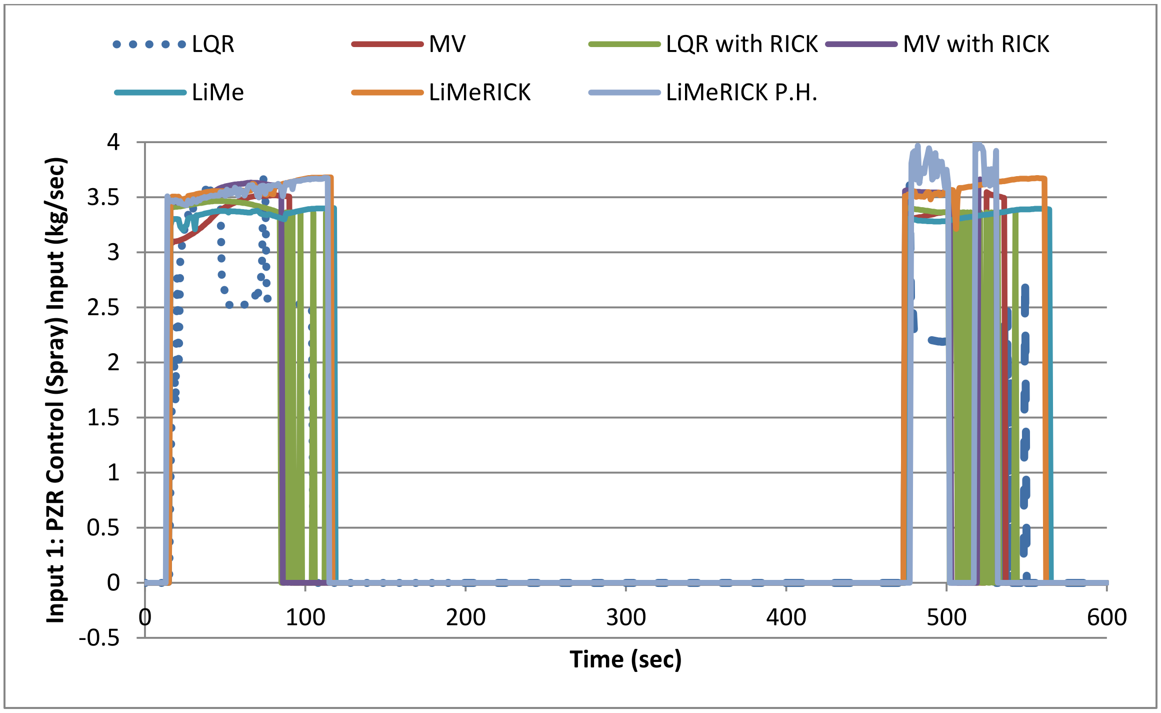


FIGURE 146: Input 1 – PZR spray (adaptive control)

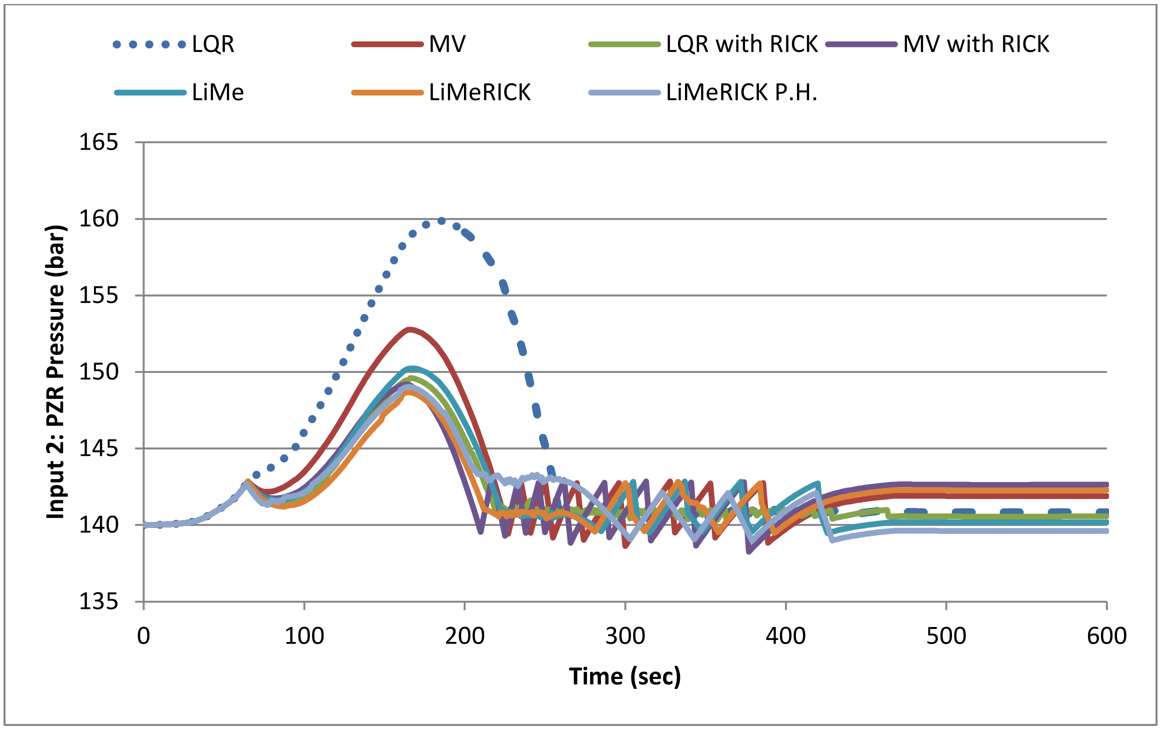


FIGURE 147: Input 2 – PZR pressure (adaptive control)

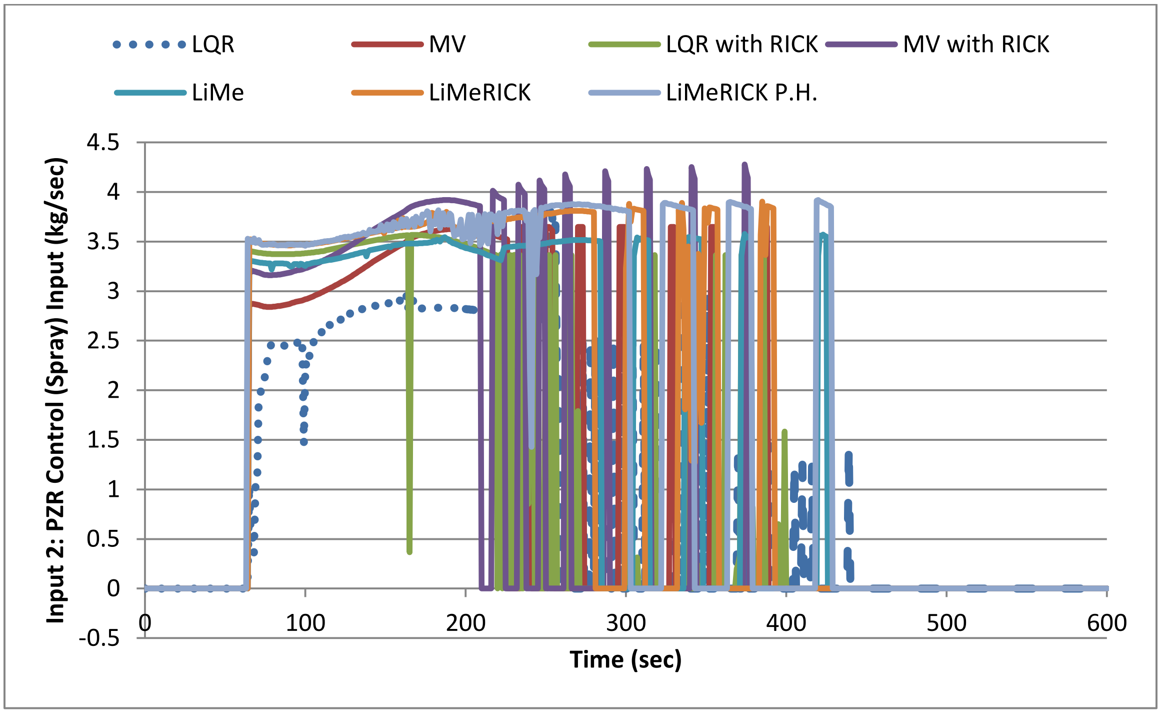


FIGURE 148: Input 2 – PZR spray (adaptive control)

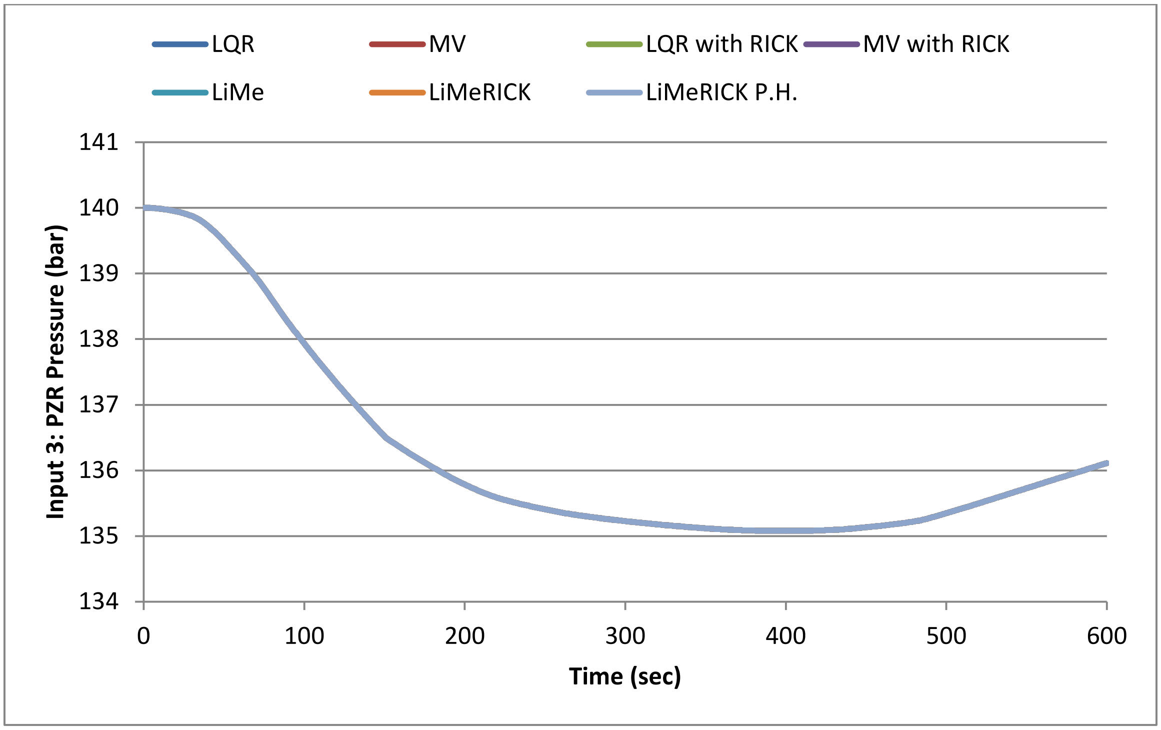


FIGURE 149: Input 3 – PZR pressure (adaptive control)

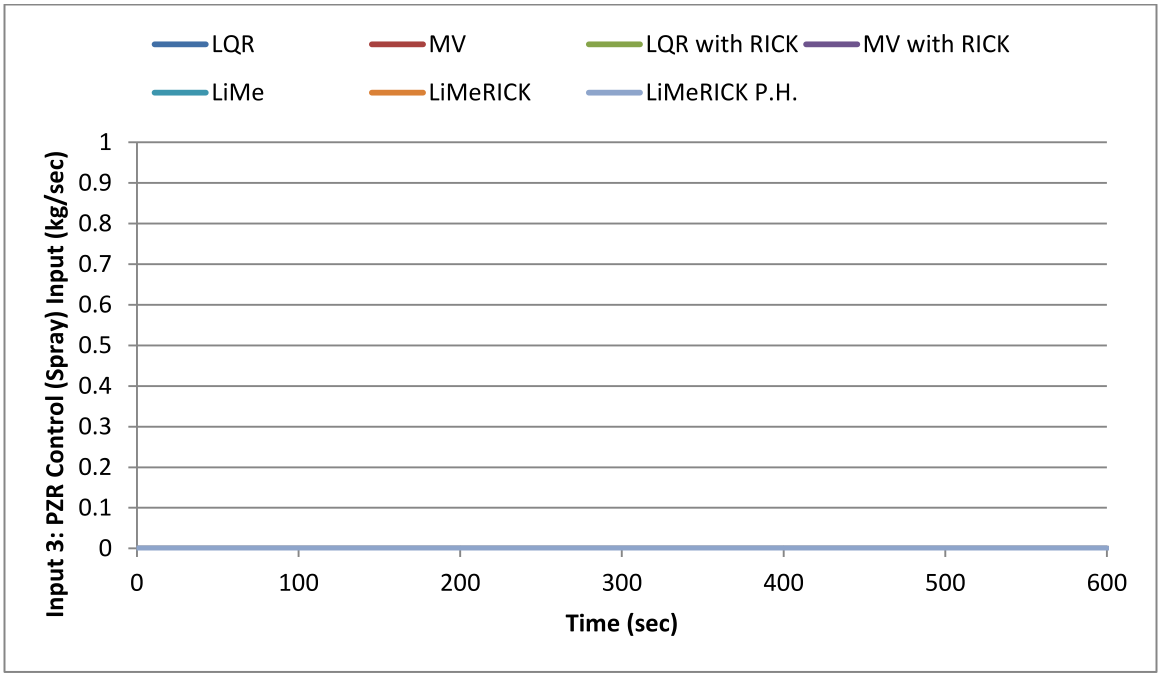


FIGURE 150: Input 3 – PZR spray (adaptive control)

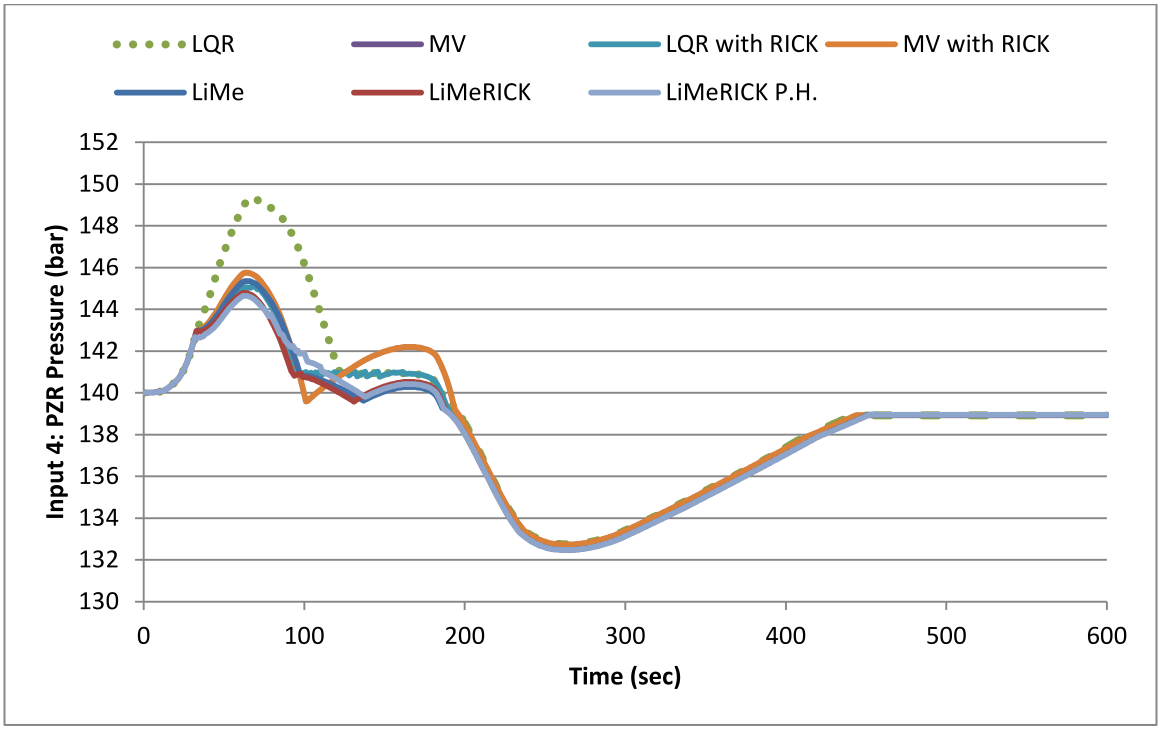


FIGURE 151: Input 4 – PZR pressure (adaptive control)

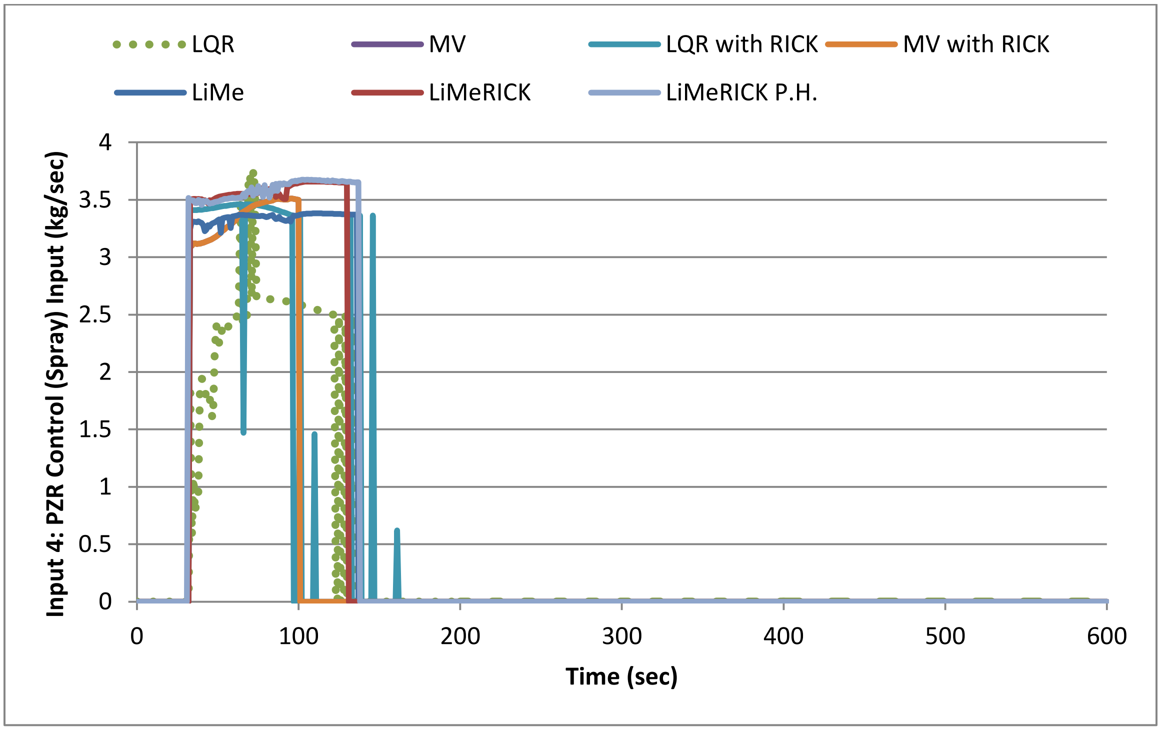


FIGURE 152: Input 4 – PZR spray (adaptive control)

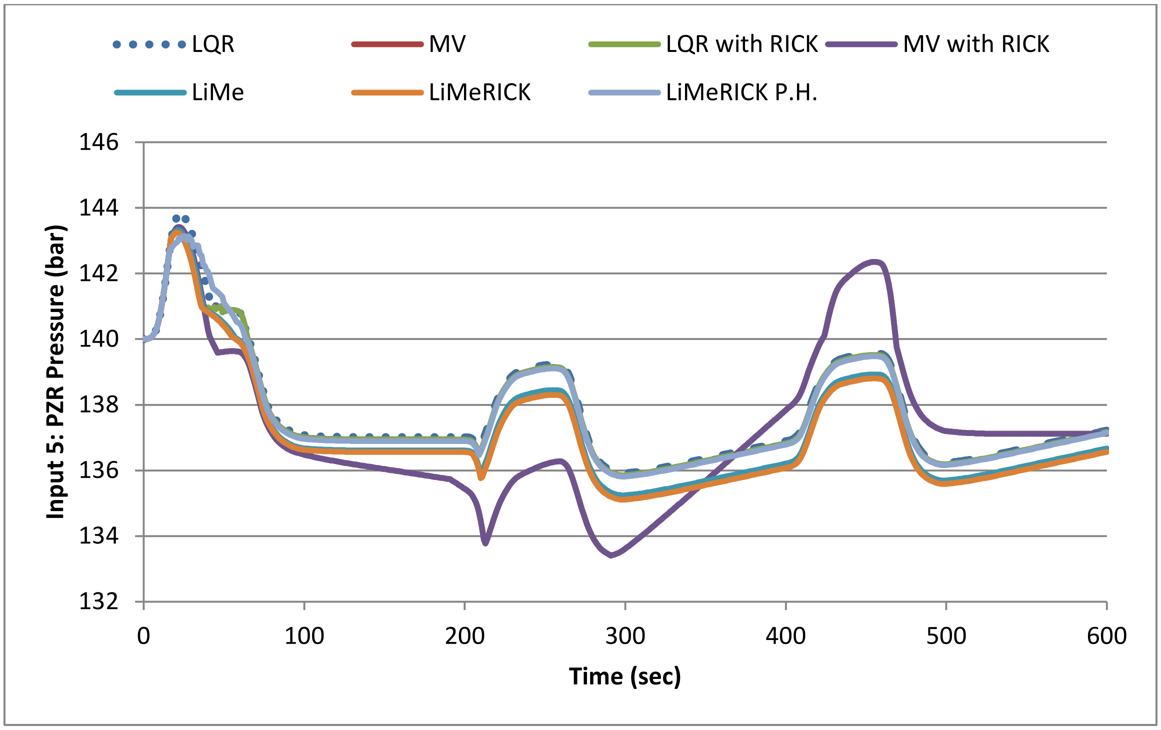


FIGURE 153: Input 5 – PZR pressure (adaptive control)

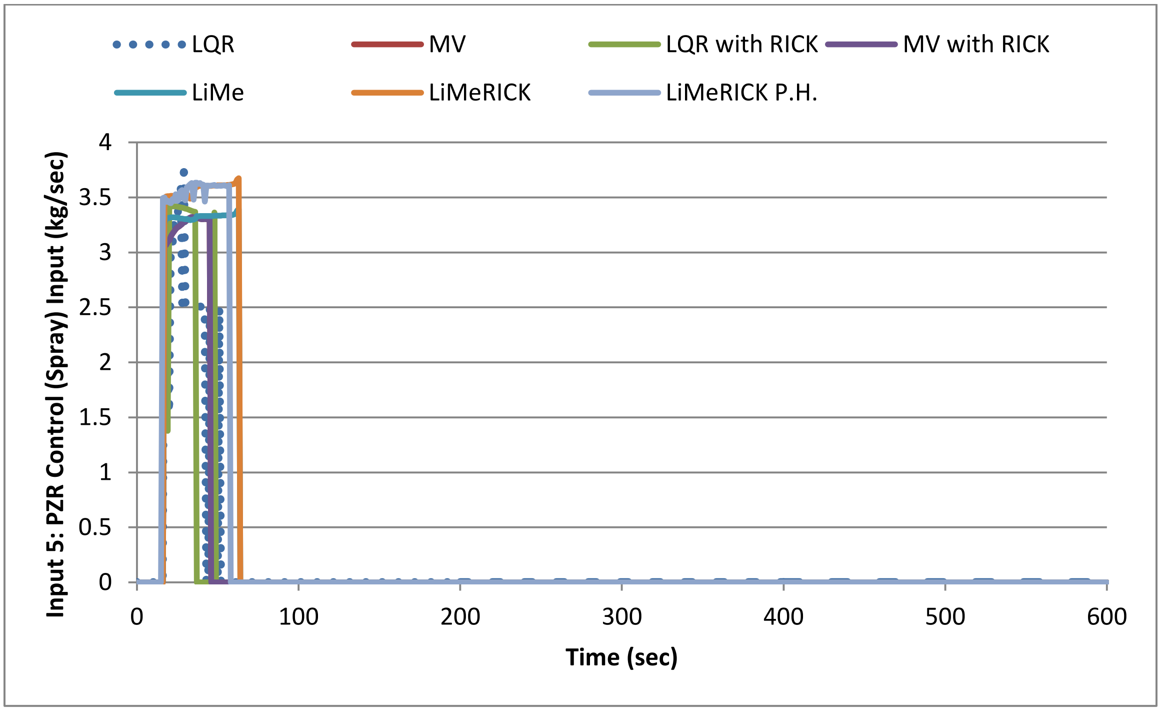


FIGURE 154: Input 5 – PZR spray (adaptive control)

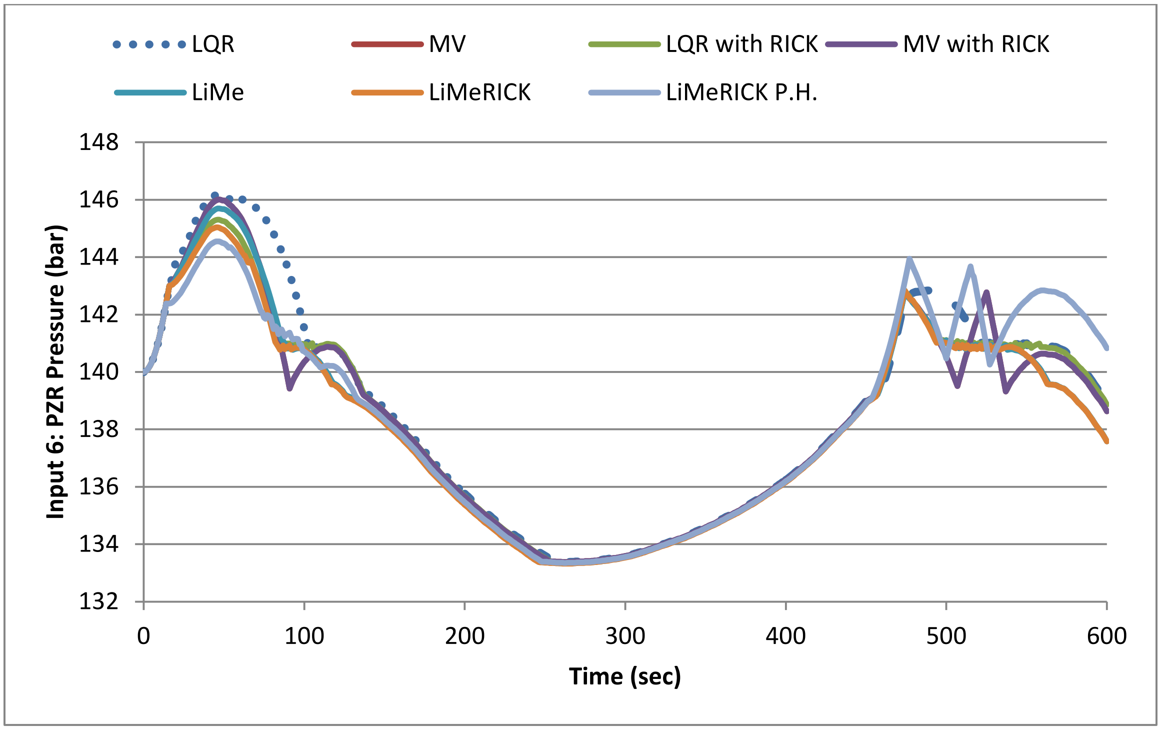


FIGURE 155: Input 6 – PZR pressure (adaptive control)

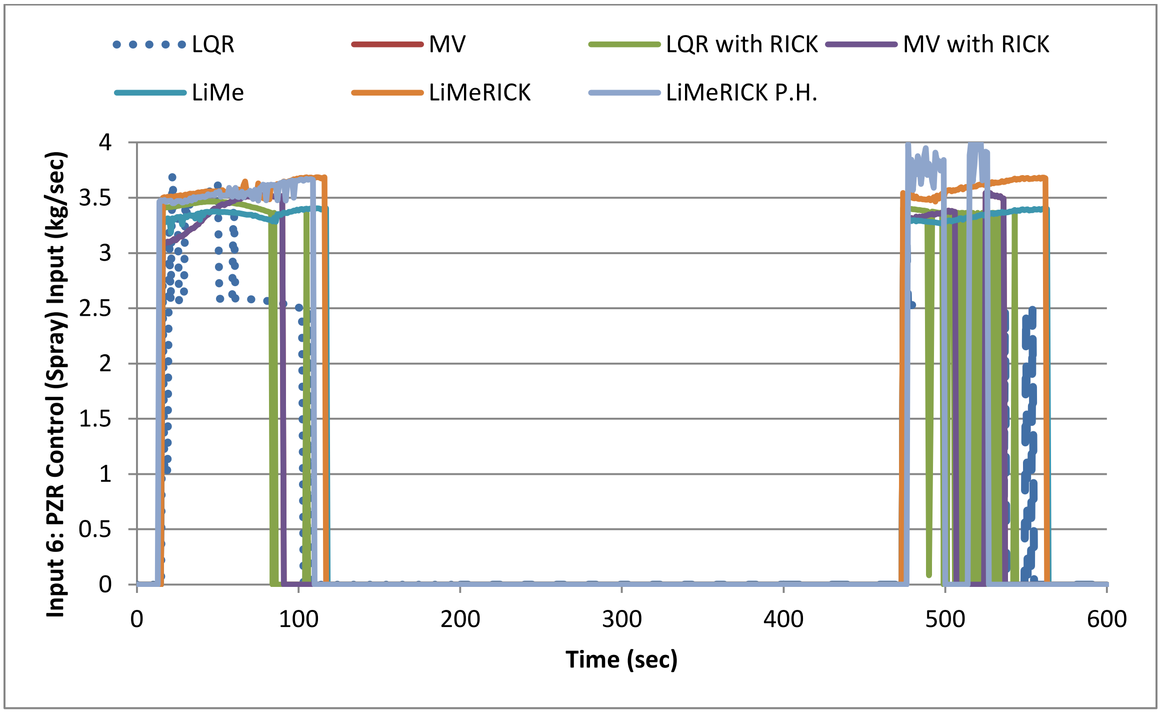


FIGURE 156: Input 6 – PZR spray (adaptive control)