VARIABLE SPEED SIMULATION FOR ACCELERATED INDUSTRIAL CONTROL SYSTEM CYBER TRAINING

THESIS

Luke M. Bradford, 2d Lt, USAF

AFIT-ENG-MS-18-M-014

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT-ENG-MS-18-M-014

VARIABLE SPEED SIMULATION FOR ACCELERATED INDUSTRIAL CONTROL SYSTEM CYBER TRAINING

THESIS

Presented to the Faculty

Department of Electrical and Computer Engineering

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

in Partial Fulfillment of the Requirements for the

Degree of Master of Science in Computer Science

Luke M. Bradford, 2d Lt, USAF, B.S.C.S

March 2018

AFIT-ENG-MS-18-M-014

VARIABLE SPEED SIMULATION FOR ACCELERATED INDUSTRIAL CONTROL SYSTEM CYBER TRAINING

THESIS

Luke M. Bradford, 2d Lt, USAF, B.S.C.S

Committee Membership:

Barry E. Mullins, Ph.D., P.E.

(Chairman)

Timothy H. Lacey, Ph.D., CISSP

(Member)

Stephen Dunlap

(Member)

Abstract

Industrial control systems are complex systems that employ a wide range of hardware, software, and network protocols to control physical processes critical to the smooth functioning of society. It is important for control system operators to receive quality training to respond to various cyber events such as operator actions and cyber attacks. Hands-on training exercises with real-world control systems allow operators to learn various defensive techniques and see the real-world impact of changes made to a control system. Cyber events can have effects that take a significant amount of time to manifest, making high-fidelity training exercises time-consuming. In addition, cyber events can have unforeseen effects that potentially cause physical harm to the system. The potential damage to exercise equipment threatens to make high-fidelity training exercises prohibitively expensive.

This thesis presents a method for accelerating training exercises. Specifically, the method entails simulating and predicting the effects of a cyber event on a partially-simulated control system that has the ability to speed up the simulated industrial process while allowing the control hardware to continue operating as intended. A hardware-in-the-loop system comprised of a software-modeled water tank controlled by a commercially-available programmable logic controller is used to demonstrate the feasibility of this method.

In order to verify the accuracy and consistency of the proposed method, experimentation includes validation testing with an actual water tank controlled by the same model programmable logic controller used in the simulated system. Together, the water tank and programmable logic controller represent a partial, real-world control system. The experiment requires the simulated system to replicate a cyber event

in the real-world system at real-time, two times faster than real-time, and ten times faster than real-time. An increase in the water level set point in the real-world system represents the cyber event that is used by the experiment. Data collected from the simulated control system when run at real-time and at higher speeds is compared to data obtained from the real control system in order to determine the accuracy and consistency of the proposed system. Specifically, the experiment compares the average difference between runs of the real-world system and the simulated system. Average difference represents the average distance between water level for the real-world system and water level for the simulated system at any point in time.

The results of the experiment demonstrate that the simulated system is able to replicate the change in water level set point in the real-world system when run at real-time, two times faster than real-time, and ten times faster than real-time. When run at real-time, the average difference between the simulation and the real-world system ranges from 0.198% to 0.21%. When run at two times faster than real-time, the average difference between the simulation and the real-world system ranges from 0.182% to 0.204%. When run at ten times faster than real-time, the average difference between the simulation and the real-world system ranges from 0.193% to 0.278%. The results also demonstrate that the simulation is consistent when run at each speed. However, as simulation speed increases, consistency decreases. When run at real-time, the standard deviation among simulation runs is 0.006%. When run at two times faster than real-time, the standard deviation among simulation runs is 0.009%. When run at ten times faster than real-time, standard deviation among simulation runs is 0.049%.

The method proposed by this research for accelerating industrial control system cyber training exercises allows operators to receive high-fidelity training in a practical amount of time. Rather than waiting for the effects of slow moving cyber events to

manifest, operators and exercise coordinators can speed up time in order to quickly see the results of a cyber event and devote more time to analysis and evaluation which represent higher levels of learning. Using systems similar to the one developed by this research in conjunction with a full-scale industrial control system enables operators to train in robust, high-fidelity environments while limiting the possibility of damage to control equipment caused by unforeseen effects from cyber events. Operators and exercise coordinators can use the system to see the future consequences of cyber events before they occur in order to prevent potential harm to exercise equipment.

To my Mom, Dad, and pets, for all of their love and support.

Acknowledgements

I would like to thank the members of my thesis committee, Dr. Barry Mullins, Dr. Timothy Lacey, and Stephen Dunlap as well as Dr. Mason Rice and Dr. Douglas Hodson, for sharing their time and knowledge with me. I am extremely grateful that I had to opportunity to work together and learn so much.

Luke M. Bradford, 2d Lt, USAF

Table of Contents

Page

Page

List of Figures

List of Tables

List of Acronyms

VARIABLE SPEED SIMULATION FOR ACCELERATED INDUSTRIAL CONTROL SYSTEM CYBER TRAINING

I. Introduction

1.1 Background

Most critical infrastructure owners and operators lack the training to prevent and properly respond to sophisticated cyber attacks against Industrial Control Systems (ICS) [1]. Additionally, information security operators lack an understanding of ICSs and cannot predict the impact of changes to a control system network. Adversaries know that a cyber attack launched against an ICS has the potential to cause significant physical harm to a nation's critical infrastructure. The ability to cause physical damage and the lack of well-trained operators make ICSs a lucrative target for potential adversaries. Thus, an increase in the quantity and sophistication of ICS cyber attacks is inevitable. Further exacerbating the lack of cyber-capable control system operators is the fact that most ICS cyber security training provides instruction at the basic or intermediate knowledge levels [2]. The absence of thorough, advanced training specifically designed for ICS cyber security is due primarily to the lack of robust training facilities that provide interaction with real-world control system components controlling physical processes. Consequently, there is a need for training environments that blend real control system components with physical processes so that students can understand the cyber-physical effects of cyber attacks and operator actions.

1.2 Problem Statement

It is important for control system operators to receive quality training to respond to various cyber events such as operator actions and cyber attacks. Hands-on training exercises with real-world control systems allow operators to learn various defensive techniques and see the real-world impact of changes made to a control system. Cyber events can have effects that take a significant amount of time to manifest. Therefore, replicating their impact in a learning environment may require an infeasible amount of time. Additionally, unforeseen consequences of cyber events have the potential to cause catastrophic damage to control system equipment. The potential damage to exercise equipment threatens to make high-fidelity training exercises prohibitively expensive. A critical component of any solution, then, is the ability to quickly model the effects of cyber events so that more time can be devoted to analysis and evaluation which represent higher levels of learning [3]. The ideal training environment is a full-scale, real-world facility with several interconnected processes [2]. Training in such a facility, however, is costly, time-consuming, and can damage the equipment. Thus, a variety of mobile, cost effective, realistic ICS training environments have been developed [2]. They utilize real process control hardware and software, which control a partially-simulated ICS. These environments are not full-scale ICSs, but do provide familiarization with real-world equipment, industrial networks, and process logic. They address many critical skills and maximize realism with hands-on training at a fraction of the cost of building a full-scale training environment. Unfortunately, these training environments lack the capability to replicate the impacts of slow moving cyber events in a feasible amount of time. This thesis seeks to answer the following question: *What alternative solution can be developed to provide realistic ICS cyber training that replicates the effects of cyber events in a practical amount of time?*

1.3 Research Goals and Hypothesis

The goal of this research is to develop a method that augments ICS cyber security training environments by enabling exercise coordinators to accelerate and predict the effects of a cyber event.

This research proposes simulating and predicting the effects of a cyber event on a mobile, cost effective, realistic ICS training environment as the method. Specifically, the training environment has the ability to speed up the simulated industrial process while allowing the control hardware to continue operating as intended. This research hypothesizes that the proposed method enables operators and exercise coordinators to accelerate the replication of cyber events in order to predict their effects and limit potential unforeseen damage to control system equipment.

1.4 Approach

1.4.1 Implementation.

Based upon the above goal, this research develops a Hardware-in-the-Loop (HiL) system comprised of a software-modeled water tank and a commercially-available Programmable Logic Controller (PLC) as a means for demonstrating the feasibility of the proposed method. In order to accelerate the operation of the system and maximize realism, a key component of development includes devising an approach for speeding up the simulated water tank while allowing the PLC to continue operating as intended.

1.4.2 Experimentation.

In order to verify the accuracy of the proposed method, experimentation includes validation testing with an actual water tank controlled by the same model PLC used

in the simulated system. Together, the water tank and PLC represent a partial, real-world control system. Data collected from the simulated control system when run at real-time and at higher speeds is compared to data obtained from the real control system in order to determine the accuracy and consistency of the proposed system.

1.5 Assumptions and Limitations

Due to time constraints and the wide breadth of this field of research, the following assumptions and limitations are required.

1.5.1 PLC Type.

Both the simulated control system and the real control system use an Allen-Bradley ControlLogix PLC as the control unit hardware. Allen-Bradley is one of the most widely deployed brands of PLCs but only represents one option from a variety of PLC vendors.

1.5.2 Process Selection.

A water tank provides the physical process for both the simulated control system and the real control system. Water tanks are found in many ICSs such as chemical mixing plants and wastewater treatment facilities. However, a water tank only represents one of many physical components found in ICSs.

1.6 Research Contributions

This thesis presents a method for accelerating ICS cyber training exercises by simulating and predicting the effects of a cyber event on a partially-simulated control system. The results of this research demonstrate the system's speedup capability which allows users to accurately simulate the effects of a cyber event at speeds faster

than real-time. The method proposed by this research allows operators to receive high-fidelity training in a practical amount of time. Rather than waiting for the effects of slow moving cyber events to manifest, operators and exercise coordinators can speed up time in order to quickly see the results of a cyber event and devote more time to analysis and evaluation. In addition, operators and exercise coordinators can use the method developed by this research to see the future consequences of cyber events before they occur in order to prevent potential harm to exercise equipment. Using systems similar to the one developed by this research in conjunction with a full-scale industrial control system enables operators to train in robust, high-fidelity environments while limiting the possibility of damage to control equipment caused by unforeseen effects from cyber events. In effect, operators are given the ability to speed up time and see the future consequences of their actions while limiting the possibility of physical damage to the exercise equipment.

1.7 Thesis Overview

Chapter II contains an overview of ICS technology and education as well as related work on modeling control systems. Chapter III describes the implementation of a partially-simulated control system that speeds up the effects of a cyber event. Chapter IV details the experimental design, and the results of the experiment are presented in Chapter V. Chapter VI discusses research conclusions as well as suggestions for future work.

II. Background and Related Research

2.1 Overview

This chapter discusses important background knowledge for ICSs as well as the need to provide high quality ICS cyber training for ICS operators. It then gives an overview of HiL simulation, an important concept used in this research effort. Finally, the chapter concludes with a discussion of related research for ICS cyber education and training environments developed with MATLAB Simulink, a key tool used in this research.

2.2 Background

2.2.1 Industrial Control Systems.

The United States describes critical infrastructure as systems and assets, both physical and virtual, so crucial to the country that their destruction or incapacity would threaten U.S. national security, economic areas, power supply, public health, public safety and many other areas [4]. Presidential Policy Directive 21 lists 16 critical infrastructure sectors in America including areas such as energy, transportation, water, and communications [5]. These sectors adopted industrial control technology in the form of ICSs in order to achieve increased automation, efficiency, and maintainability.

Operational Technology (OT) refers to the computing systems comprised of hardware and software dedicated to managing industrial operations and physical processes rather than administrative processes which are managed by Information Technology (IT) [6]. ICSs are a major component of OT as they monitor and control industrial processes and are composed of sensors, actuators, and control unit(s) [7]. In an ICS, one or more control units receive data from sensors. Based upon the data

received, the control unit(s) act through the actuators to control the process being monitored by the sensors in order to produce the desired output [7]. An ICS can be implemented locally, within the confines of a factory for example, or among several geographically remote sites such as a power grid. ICSs are often systems of systems that control several interconnected, mutually dependent processes that act together to achieve an industrial objective.

Figure 1 shows the layout of an ICS. An ICS has three critical parts, namely, the Human-Machine Interface (HMI), the Remote Diagnostics and Maintenance, and the Control Loop. Sensors in the Control Loop transmit data (e.g., temperature of a storage facility) to the controller which continuously polls the sensors for data. The controller then sends the data to the HMI where an operator monitoring the HMI can view the data. The operator can interpret the data in order to decide whether to take action (e.g., if the temperature is too high or too low). An example could be deciding whether to raise or lower the temperature in a storage facility. The operator could send this as a command to raise or lower the temperature back to the controller. When the controller receives a command, it sends it to the actuators which execute the command (e.g., turn on heater to raise temperature). Remote Diagnostic and Maintenance monitors the Control Loop in order to ensure the sensors, actuators, and controller are functioning properly. The PLC is the typical control hardware used in ICSs. PLCs are designed to be easily programmed and maintained. They are able to communicate with a central control system that may include several combined PLCs, each responsible for different tasks that collectively achieve a larger industrial objective [8].

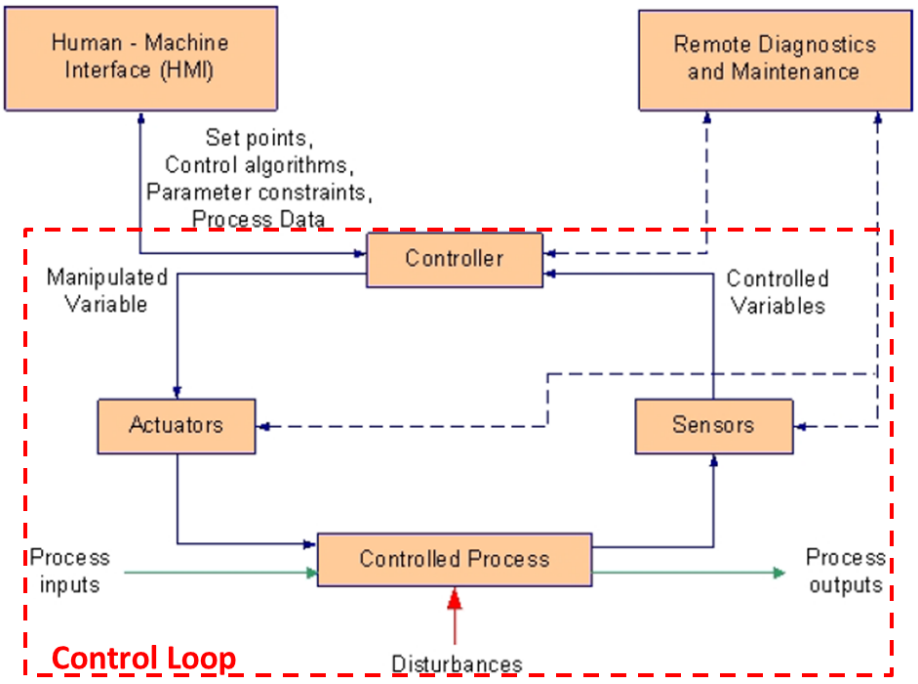

Figure 1. ICS Block Diagram. [7]

2.2.2 Need for Cost Efficient Cyber Training Environments for Industrial Control Systems.

Plumley et al. introduced levels of cognitive complexity for control system training environments based upon their respective capabilities [2]. Their goal was to create an ICS educational framework that offered training for varying organizational budgets and needs. There are four levels, each with a varying degree of capability. The primary delimiter of the level for a training environment is the realism it provides in the context of a real ICS. The complexity of training scenarios that can be accomplished in an ICS training environment depends upon the amount of realism provided by the environment. The level of cognitive complexity increases as training environment realism increases [2]. With this in mind, Plumley et al. created and mapped the four levels of ICS training environments to Bloom's Taxonomy in order to create a complete ICS cyber training educational framework. Training environments capable

of administering exercises at higher levels of thinking map to higher levels of Bloom's Taxonomy and access all lower levels of the taxonomy.

Bloom's Taxonomy is an educational framework created by the educational psychologist, Benjamin Bloom (1913-1999). The taxonomy classifies educational objectives based upon cognitive complexity [3]. Bloom's Taxonomy is widely used today by educators to structure courses that help students learn, apply knowledge, think critically, and create new ideas. Bloom's Taxonomy was revised in 2001 and consists of six categories of educational goals. [2]. The taxonomy progresses from the lowest cognitive complexity level of basic understanding to the highest cognitive complexity level which is creation of original ideas. Bloom's taxonomy offers a means for aligning educational tools to a specific level of cognitive complexity. Bloom emphasized acquiring concrete knowledge before increasing the complexity of training. In other words, one must master their current level in the taxonomy before proceeding to the next higher level. This explains why in many high risk or critical fields, training includes several levels of simulation where the complexity of each level increases so that students are well acquainted before attempting real tasks with real equipment.

A Level 1 training environment is totally software defined. It is uses software to simulate an industrial controller or control system. Level 1 environments reach the lowest two levels of the taxonomy, namely, Remembering and Understanding [2].

A Level 2 training environment includes an automated process which creates real physical effects. However, the environment is not constructed from vendor hardware and software used in industry. Rather, it is constructed from embedded devices (e.g., Arduino, Raspberry Pi) that can be programmed to monitor and control physical processes with common programming languages (e.g., C, Python). Level 2 environments reach the Applying and Analyzing levels of Bloom's Taxonomy [2].

A Level 3 environment uses real process control hardware and software. The hardware and software control a partial industrial control system [2]. Consider a prison for instance, an example Level 3 environment would be the controlling of the locks on a block of prison cell doors. These environments are not full-scale ICSs. However, they provide familiarization with real-world equipment, industrial networks, realistic process logic, and portability. Since they are not full-scale systems, they cannot provide an understanding of scale real-world systems where a malfunction in one process may affect several others due to the interconnection of the various system processes. This level reaches the Evaluating level of the taxonomy. Students can make evaluations by comparing data and observations with standard operation criteria and data of the control component. Realistic data allows students to make realistic evaluations that transfer to real-world systems. In order to maximize realism and minimize cost and space, Level 3 environments employ HiL simulation of industrial processes that are controlled by actual control hardware. HiL simulation removes the need to include large and expensive real-world physical processes in training environments by replacing them with accurate simulations that interface with real-world control hardware.

A Level 4 environment is a real-world, full-scale ICS facility. The training facility construction would be the same as the construction of a real-world facility [2]. A Level 4 environment reaches the highest level of thinking in the taxonomy, namely, Creating. For ICS cyber education, this type of cognitive complexity cannot be achieved without a full-scale system. Such an environment provides students with the ability to view and manipulate every possible component in an actual industrial environment. New methods and solutions can be devised, tested, and applied to actual industrial systems, and their effects can be observed.

Cyber attacks, whether against traditional IT systems or ICSs are often slow moving and thus difficult to detect. Stuxnet, a famous control system cyber attack, took months to complete [9]. Since cyber attacks often employ the low and slow attack paradigm, it takes a long time to observe their effects. Thus, a key component of providing realistic training to ICS operators in a reasonable amount of time is the ability to speed up the effects of slow moving cyber events.

2.2.3 Hardware-in-the-Loop Simulation.

The ICS training environment discussed in this thesis is a Level 3 environment designed specifically to speed up the simulation of physical processes so that users can quickly detect, observe, and predict the effects of cyber events. It employs HiL simulation of a physical, industrial process controlled by an actual PLC. HiL simulation is a common technique used in many industries for the development and evaluation of complex, real-time, embedded systems [10]. It provides a simulated process under control in order to create a realistic test environment for embedded systems. During testing, the embedded system interacts with the process simulation. The process simulation is often a software implementation of a mathematical representation of the dynamics of a real-world, complex process. A key component of HiL simulation is the electrical emulation of process sensors and actuators which serves as the interface between the process simulation and the embedded system under test. The process simulation determines the value of the electrically emulated sensors. These values are then read by the embedded system under test as feedback. The control algorithm running on the embedded system under test causes the embedded system to output actuator control signals based upon the feedback received. Changes to the output control signals result in changes to the variable values in the process simulation in-

cluding the values of the sensors. Thus, HiL simulation includes a complete control loop.

HiL simulation is commonly used to test embedded systems because in many cases, it is more efficient to use HiL simulation rather than connecting the embedded system to a real process. For example, HiL simulation can enhance the quality of testing by widening the scope of test scenarios and overcoming the testing limitations imposed by using a real process. Testing with a real process prohibits the embedded system from being tested at failure conditions. Furthermore, HiL simulation aids developing embedded systems under tight schedules that cannot allow testing to wait until a process prototype becomes available. Finally, it is usually more economical to test with a high-fidelity, real-time HiL simulator rather than a real process.

2.3 Related Research

2.3.1 MATLAB Simulink.

Saco et al. acknowledged that the ideal learning environment for control system operators is a real-world plant [10]. They also acknowledged that such environments are too dangerous, too large, and too expensive to bring to the classroom. Therefore, they highlighted the need for high-fidelity simulation tools in order to provide effective education to control system operators. They proposed an HiL real-time simulation system as a solution. MATLAB Simulink, a software tool for modeling dynamic systems, was used to model the control algorithm and the plant [11]. The plant was a water tank that included a water inflow pump and a pneumatic drainage valve. The control algorithm was converted to a C program with the Simulink Real Time Workshop. The C code was downloaded to real-time prototyping hardware and executed independent of MATLAB. They used a dSpace 1102 floating-point controller board as the prototyping hardware. Once the C code was downloaded to the board and ex-

ecuted, the controller hardware was then able to control the level of the water in the simulated water tank. The functionality provided by Simulink meant that students could implement and refine their own similar HiL systems in order obtain a better understanding of physical systems and control principles.

Thornton and Morris acknowledged that fundamental risks in ICSs can be identified by analyzing cyber attacks and their effects against control systems [12]. Similar to Saco et al., they affirmed that the ideal environment for studying cyber attacks against an ICS is a real system with real hardware, software, and communication technologies. They also concluded that such environments are prohibitively expensive. Thus, they proposed a virtual laboratory designed to be mobile, sharable, and expandable. They also employed Simulink for high-fidelity process modeling in their virtual ICS security research testbed. The laboratory included a gas pipeline simulation with sensors and actuators, a virtual PLC simulated with Python, and an HMI. The Simulink gas pipeline and the virtual logic controller communicated via JavaScript Object Notation (JSON) attribute-value pairs contained in User Datagram Protocol (UDP) packets. The virtual logic controller communicated with other devices such as the HMI and physical PLCs via Modbus/TCP, a standard ICS protocol. The communication between the virtual logic controller and other physical logic controllers allowed the virtual laboratory to produce accurate control system network traffic for analysis.

A major component missing from the two environments previously described in this section is the use of real-world control system hardware programmed to control the simulated physical processes. Using real-world control system hardware and software serves to increase the realism and utility of ICS learning environments. ICS-specific hardware brings simulated environments much closer to resembling the operation of actual control systems under both normal and hostile circumstances. Fi-

nally, both environments lack a way to speed up their operation so that students, researchers, control system operators, and white cell members of an ICS cyber training exercise can quickly model the effects of various control system cyber events in order to devote more time to analysis. For example, without a speedup capability, students, researchers, control system operators, and white cell members cannot quickly replicate low and slow attacks such as Stuxnet in order to see their effects and learn from them in an acceptable and useful time frame.

2.4 Chapter Summary

Chapter II began with an overview of ICSs and a discussion of the need for high-quality ICS cyber training for ICS operators. It then gave an overview of HiL simulation and described its advantages. Finally, the chapter concluded with a summary of related research for ICS cyber education and training environments developed with MATLAB Simulink, a key tool used in this research to develop an ICS cyber training environment.

III. Test Systems

3.1 Overview

This chapter describes the implementation of the test environment which includes two systems implementing a water tank control loop. A Lab-Volt 3531 Training System is used as a baseline, real-world control system [13]. A simulated water tank system is designed to replicate the operation of the Lab-Volt 3531 Training System. Both the Lab-Volt 3531 and the simulated water tank include the control loop shown in Figure 2. In this control loop, the PLC is programmed to keep the water level in the tank at a user defined set point. It polls the water level sensor in order to learn the current water level in the tank. Based upon the current water level, the controller sends commands to the drainage valve in order to increase or decrease the outflow rate.

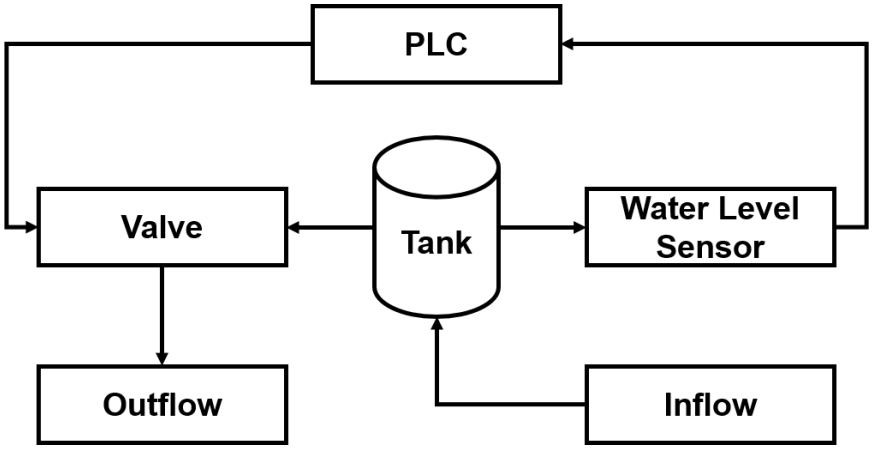

Figure 2. Test System Control Loop.

3.2 Lab-Volt 3531 Training System

Figure 3 lists the components included in the Lab-Volt 3531 system. An Allen-Bradley ControlLogix 1756-L55 PLC programmed with RSLogix 5000 from Rockwell

Automation serves as the primary control unit for the Lab-Volt system. The PLC is configured with the following firmware and modules:

- Firmware Version 5.001 Build 1
- **Slot 0** - 1756-L55 Controller with mode set to `REM` Run (remote Run)
- **Slot 1** - 1756-EWEB EtherNet/IP ENBT
- **Slot 2** - 1756-IB16/Digital Input - 24V DC Input, 16 Point
- **Slot 3** - 1756-OX8I/N.O./N.C. Isolated Relay Output - 8 Point
- **Slot 4** - 1756-IF8H/Analog Differential Input HART - Current/Voltage, 8 Point
- **Slot 5** - 1756-OF8H/Analog Output HART - Current/Voltage, 8 Point

The water tank used for the experiment is a 36 inch tall, 8 inch diameter cylindrical tank. The inflow rate of water into the tank is controlled by an AC pump driven by an Allen-Bradley PowerFlex 40 Variable Frequency Drive (VFD). A Differential Pressure Transmitter (DPT) is used to continuously monitor the inflow rate using the Venturi tube, which creates a differential pressure proportional to the rate of flow through a tube. A tap in the high pressure portion of the tube and a tap in the low pressure portion of the tube are connected to a DPT which measures the difference in pressure and calculates the flow rate. The DPT sends the flow rate to the PLC as a 4 mA to 20 mA analog signal. The PLC uses a Proportional-Integral-Derivative (PID) calculation to determine the appropriate pump speed to ensure that the water inflow rate remains constant at a given set point. The calculated pump speed is sent to the VFD as an analog value. Users can use the included Allen-Bradley PanelView Plus 600 HMI to monitor and configure the system.

PID controllers are control loop feedback tools commonly used in industrial systems [14]. A PID controller continuously calculates an error value as the difference

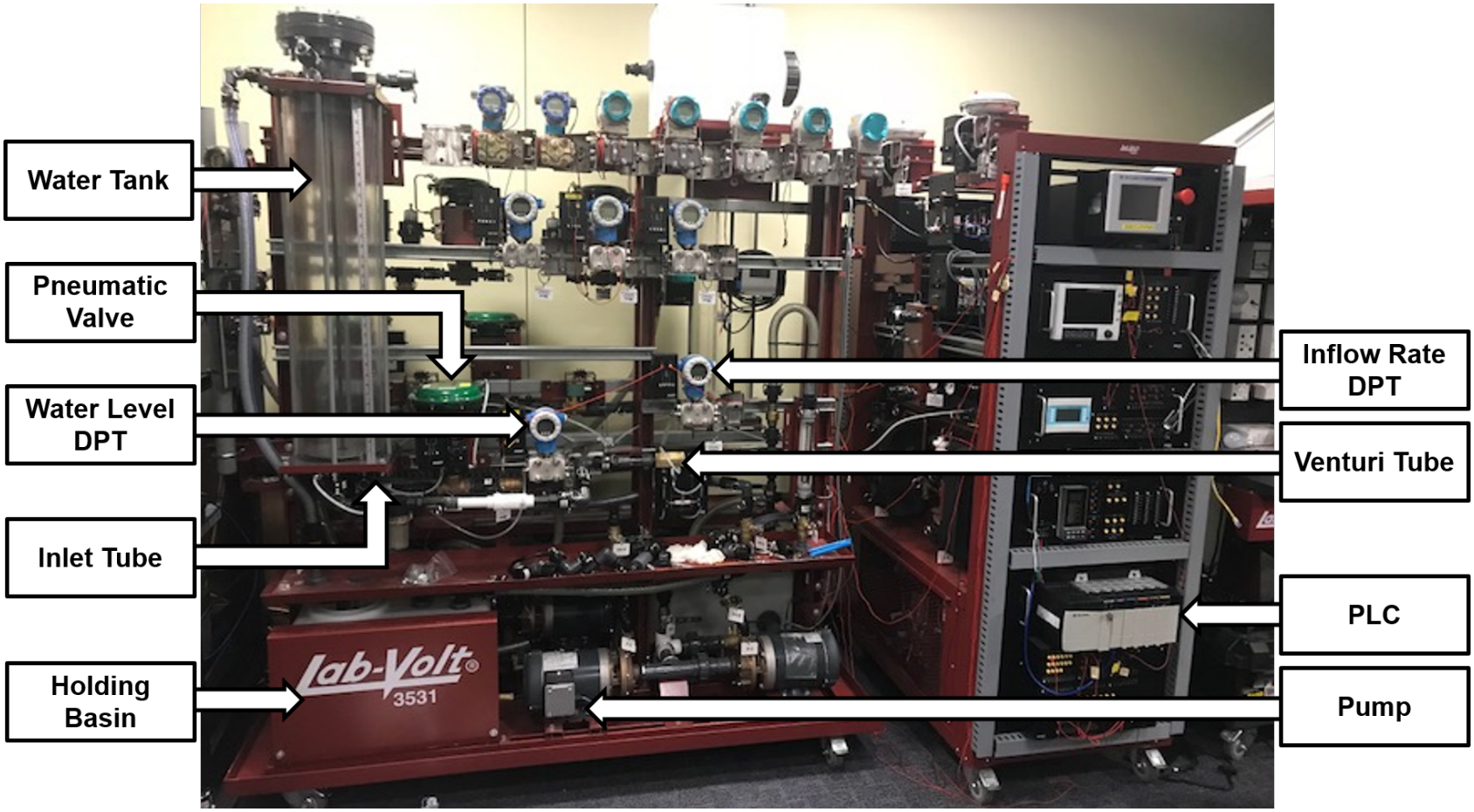

Figure 3. Lab-Volt 3531 Training System.

between a user-provided set point and a measured process variable. It applies correcting modifications based upon a balance of proportional, integral, and derivative terms which are denoted as P, I, and D respectively. The P term is proportional to the current error value. For example, if the current error value is large and negative, the control output of the P term is proportionally large and negative. Relying on the P term alone to control a process results in an error between the set point and the process variable because the P term requires an error value in order to generate its control response. Without an error, the P term cannot generate its corrective control response. The I term accounts for previous error values and integrates them over time in order to generate its corrective control response. For example, if there is a leftover error after the application of the corrective control response of the P term, the I term attempts to remove the leftover error by generating a corrective control response based upon the cumulative error value. The I term ceases to grow when the error is removed. As the error value decreases, the effect of the P term is lessened, but it is compensated by the growing effect of the I term. The D term is an estimate of the future trend of the error value based upon its current rate of change. Higher rates of change for the error value equate to an increased control effect from the D term.

In simple terms, a PID controller automatically applies accurate and quick modifications to correct a control function [14]. These corrections are then reflected by the control variable which is often a process actuator(s). Modification of the control variable by the PID controller causes the process variable to reach the set point. PID controllers ensure that the process variable reaches the set point in the optimal manner without lag or overshoot.

A pneumatic control valve located at the bottom of the tank allows water to exit the tank and drain into the holding basin. The valve fully or partially closes

in response to an analog signal received from the PLC. The valve is pneumatically operated and equipped with a spring-and-diaphragm actuator. The valve is a globe type valve which means it uses a plug to restrict the flow of water in the tank outlet. The plug has a fixed linear relationship with the distance traveled by the valve stem and the amount of flow allowed through the valve. When the valve receives an analog signal from the logic controller, the valve's current-to-pressure converter linearly converts the analog signal into a pneumatic pressure. The pneumatic pressure is applied to the surface of the valve diaphragm, producing a force that overcomes the spring force and moves the plug up or down. The plug restricts the flow of water through the valve from 0% to 100%. The percentage of flow allowed through the valve is referred to as the valve position.

A second DPT measures the pressure in the bottom tank to calculate the water level in the tank. The water level is transmitted to the PLC as a 4 mA to 20 mA analog signal. Another PID calculation in the PLC uses the water level supplied from the DPT to determine the valve position. The PLC sends the desired valve position to the valve as a 4 mA to 20 mA analog signal. The PID controller attempts to control the water level in the tank with minimal overshoot, undershoot, and set point deviation.

3.3 Simulated Water Tank

The process simulator for this experiment employs an HiL simulation of the physical process and a ControlLogix 1756-L55 PLC. The PLC is configured with the following firmware and modules:

- Firmware Version 5.001 Build 1
- **Slot 0** - 1756-L55 Controller with mode set to `REM` Run (remote Run)

- **Slot 1** - 1756-EWEB EtherNet/IP ENBT
- **Slot 2** - 1756-OF8/Analog Output - Current/Voltage, 8 Point
- **Slot 3** - 1756-IF16/Analog Input - Current/Voltage, 16 Point
- **Slot 4** - 1756-OF8/Analog Output - Current/Voltage, 8 Point
- **Slot 5** - 1756-OB8/Digital Output - DC Output, 8 Point

Figure 4 shows the simulated and real components of the test system. An HiL simulation allows the system to use real ICS hardware without the physical equipment such as pumps and tanks. The simulated physical process controlled by the PLC mirrors the Lab-Volt water tank system. A pump fills the tank with water at a constant inflow rate, and a drainage valve at the bottom of the tank allows water to exit. The simulation is intended to accurately replicate the behavior of the Lab-Volt system.

The simulated water tank is implemented as a MATLAB Simulink model. Simulink is a graphical programming environment for modeling systems. Users can model a variety of systems by selecting blocks from the various block libraries and connecting them with I/O arrows. Each block includes customizable features, and some blocks allow users to write custom supporting code. Simulink simplifies the modeling process because it removes the need to write tedious code for complex mathematical functions, and its graphical environment aids in visualizing complex systems. A MATLAB Function block in the model captures the tank dynamics. MATLAB Function blocks allow users to write custom functions and include them in their models.

The MATLAB Function block includes Equations (1), (2), (3), (4), and (5). These equations define the dynamics and physical characteristics of the water tank. The cross-sectional area of the tank is calculated as

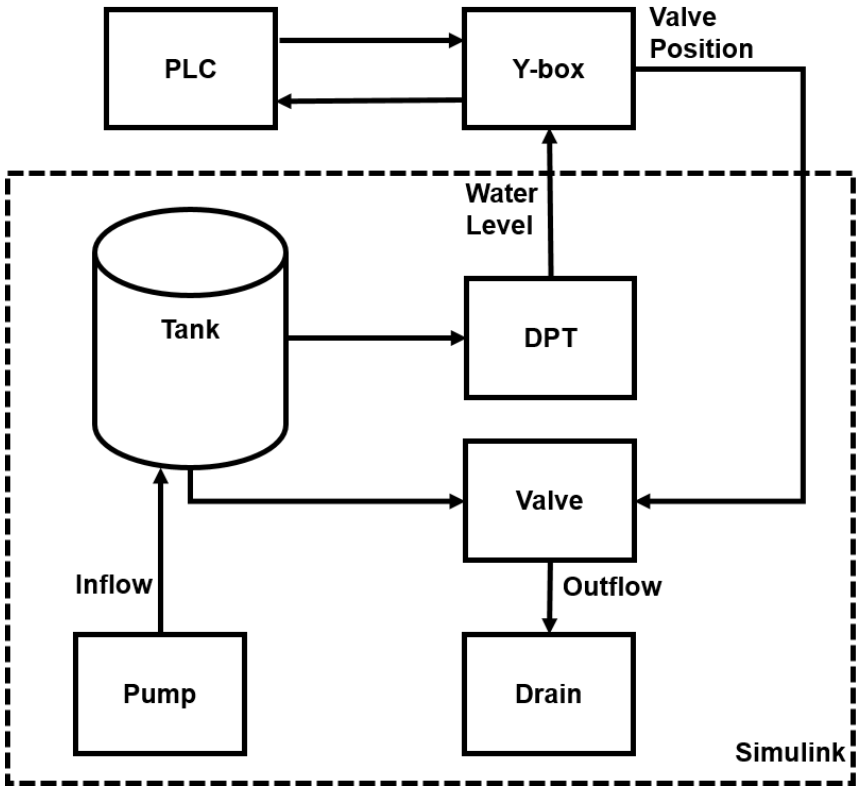

Figure 4. HiL Simulation of the Lab-Volt 3531.

$$A = \pi r^2 \tag{1}$$

where A represents the cross-sectional area of the tank in square inches and r represents the radius of the tank in inches. The height of the water in the tank is calculated as

$$Height = Volume/A \tag{2}$$

where *Height* represents the water level in inches and *Volume* represents the volume of water in the tank in cubic inches. The inflow rate is

$$Q_{in} = 588.9 \tag{3}$$

where Q_{in} represents the inflow rate of water in cubic inches per minute into the tank. Q_{in}, which is a is a user-provided argument that can be changed at any point in time during the simulation, is set to $588.9 in^3/min$. The outflow rate is calculated as

$$Q_{out} = ValvePosition * ValveConstant * \sqrt{water\ level\ psi} \tag{4}$$

where Q_{out} represents the outflow rate of water exiting the tank in cubic inches per minute. *Valve Position* is the position of the drainage valve which ranges from 0 to 1. If *Valve Position = 0.5*, for example, the valve is currently half closed in the model. Note that Q_{out} increases as the height of the water increases. This reflects the effect of water pressure on the outflow rate. The rate of change of water volume in the tank with respect to time is calculated as

$$\Delta Volume = Q_{in} - Q_{out} \tag{5}$$

where $\Delta\,Volume$ represents the rate of change of water volume in the tank with respect to time in cubic inches per minute. The signal generated by the $\Delta\,Volume$ output is fed to an integrator block which calculates the integral of the derivative with respect to time in order to calculate *Volume*. Initially *Volume = 0*, meaning the tank is empty. This is reflected as an initial condition in the configuration options of the integrator block.

The Simulink Dashboard library supports rapid development of Graphical User Interfaces (GUI) for monitoring and controlling simulated processes. Figure 6 shows the GUI for the simulated water tank. The water tank user interface includes a knob block for setting the inflow rate and gauge blocks for monitoring the water level, volume, current drainage valve configuration, and outflow rate. The user interface also has three light blocks which indicate if the current water level in the tank is at a critical level. By default, the lights shine green. When the changing water level reaches a critical height, the corresponding light starts to shine red. The lights correspond to whether the tank is less than ten percent full, greater than ninety percent full, or greater than ninety-five percent full respectively. As shown in Figure 7, a Speedup block from the Real-Time Pacer library allows the speed of the simulation to be increased or decreased. The user configures the Speedup block by entering a number that represents the speedup factor for the simulation. For example, a speedup factor of 2 doubles the speed of the simulation. A MATLAB Function block reports the current water level in the tank to the PLC every 0.1s. In order to maintain this report rate, the sample time of the MATLAB Function block must be adjusted according to the simulation speedup factor. When the simulation is run in real-time, the sample time is set to 0.1s as shown in Figure 5. A speedup factor of 2 requires the sample time to be set to 0.2s, and a speedup factor of 10 requires the sample time to be set to 1s. In general, the sample time is calculated as

$$SampleTime = SpeedupFactor/10 \tag{6}$$

A Y-box serves as the interface between the simulated water tank and the PLC by providing electrical emulation of the process sensors and actuators. The Y-box is a tool that aids in the development of HiL simulations and is designed to receive current and voltage as inputs and generate current and voltage as outputs based upon commands received over the Universal Serial Bus (USB) [15]. These inputs and outputs allow the Y-box to interface with a PLC in the same manner as regular sensors and actuators. The simulation and the Y-box communicate via serial communication. The test system utilizes the same model PLC and ladder logic used to control the Lab-Volt 3551 system. Note that the ladder logic used for the simulation does not include the PID controller that controls the inflow rate. In the Simulink model, inflow rate is set as a constant parameter. Minimal changes were made to the ladder logic to allow the PLC to operate in the simulated environment. First, the project path was updated with the Internet Protocol (IP) address of the PLC used in the simulated system. Second, the module numbers were updated to match the hardware present in the PLC used to control the simulated system. Third, the code section containing the PID controller that controls the inflow rate was not configured to activate since it is not used by the Simulink model. Figure 8 shows the setup for the simulated system.

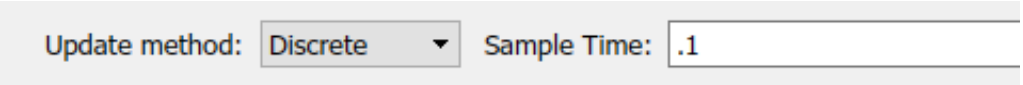

Figure 5. Setting Sample Time.

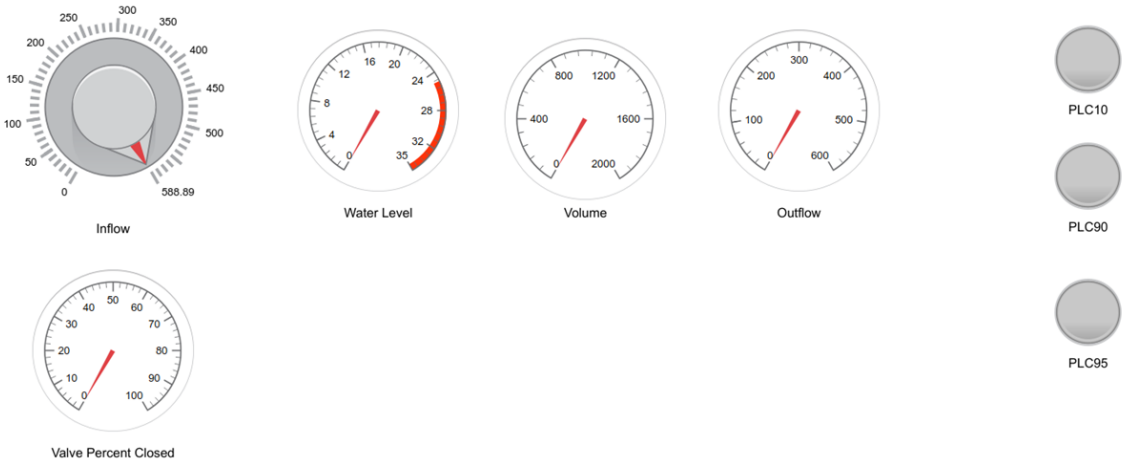

Figure 6. Simulink Water Tank GUI.

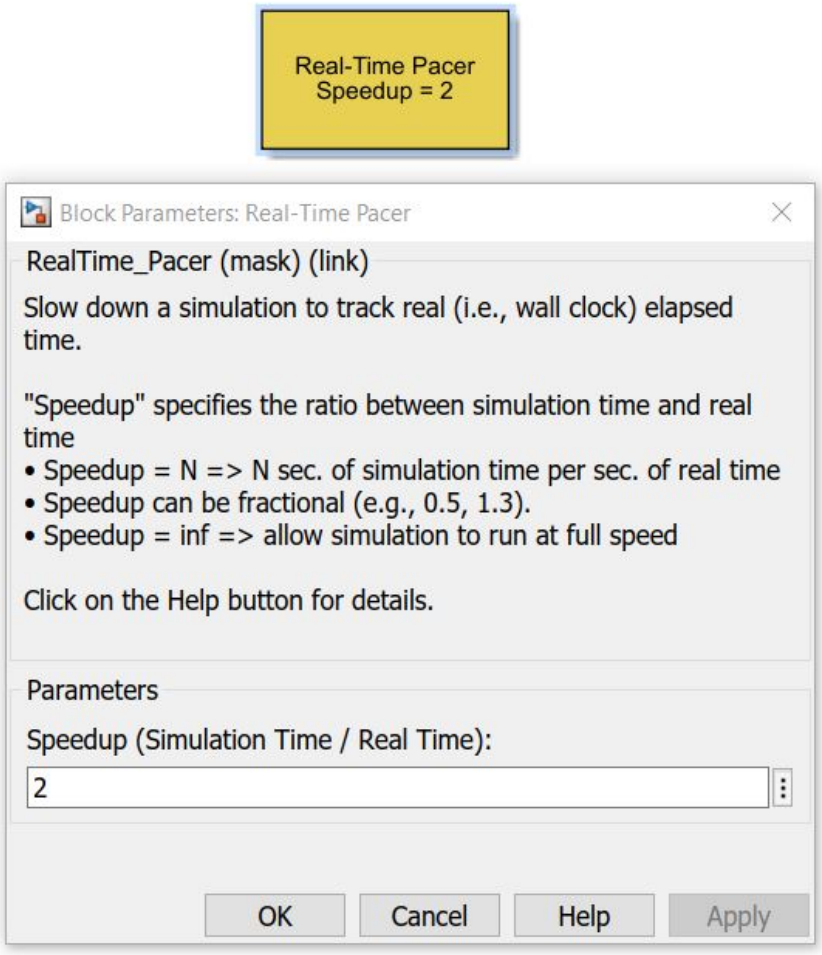

Figure 7. Simulink Speedup Block.

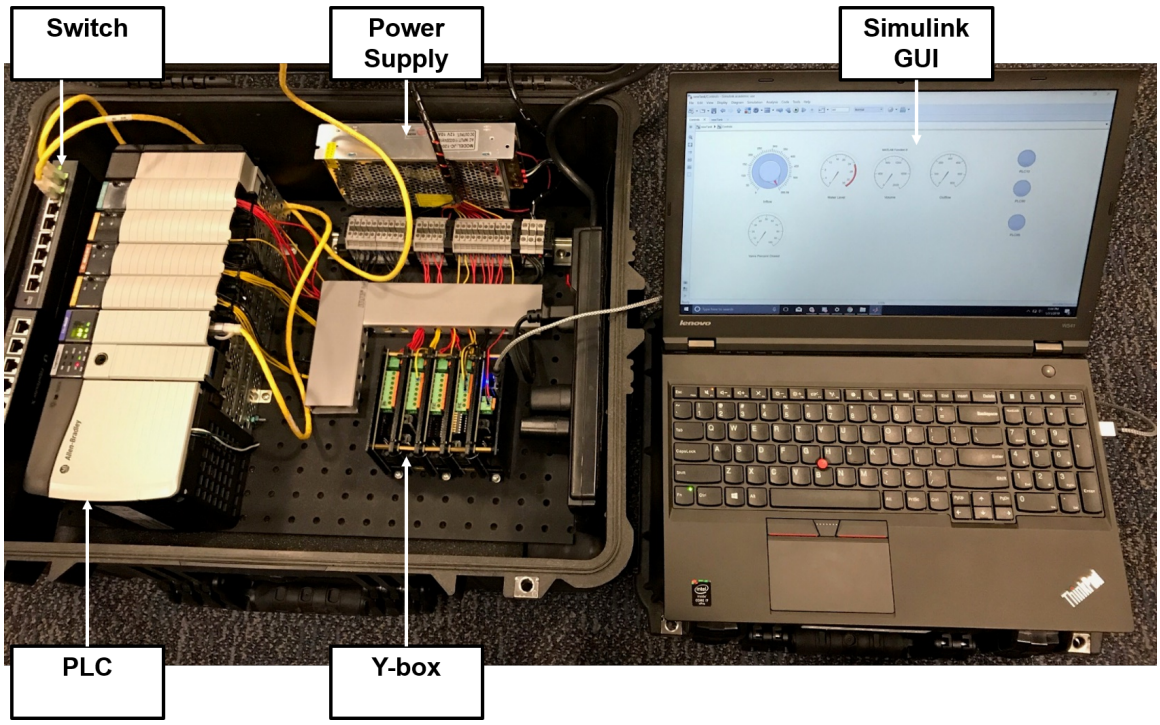

Figure 8. Simulated Water Tank Setup.

3.4 Chapter Summary

This chapter describes the implementation of the test environment which includes two systems implementing a water tank control loop. A Lab-Volt 3531 Training System is used as a baseline, real-world control system. An HiL water tank system simulated with MATLAB Simulink is designed to replicate the operation of the Lab-Volt 3531 Training System.

IV. Research Methodology

4.1 Overview

This chapter describes the methodology used to conduct an experiment involving the simulated water tank described in the previous chapter. The experiment supports research that seeks to develop a method for accelerating ICS cyber training exercises by simulating and predicting the effects of a cyber event on a partially-simulated control system. The overall goal of the experiment is determine if the proposed simulation can accurately and consistently model the effects of a cyber event at real-time and at speeds faster than real-time. In this experiment, the accuracy and consistency of the simulation are measured by comparing the water level of the simulated system to the water level of the Lab-Volt 3531 at various points in time throughout the operation of the simulation and the Lab-Volt system.

4.2 Experiment

4.2.1 Objectives.

The goal of the experiment is to demonstrate that the simulation reflects the normal operation of a real water tank when run at real-time and at faster simulation rates. In order to achieve this goal, water height is recorded at several points in time from multiple runs of the Lab-Volt 3531 and the simulated water tank. The data obtained from both systems is then compared to determine if the operation of the simulation reflects the operation of the Lab-Volt system.

4.2.2 Assumptions.

- The Allen-Bradley ControlLogix PLC is representative of other PLCs used in industry. This assumption allows the experiment to focus on measuring and

improving the simulated system's ability to accurately and consistently model the Lab-Volt system at real-time and at higher simulation rates rather than conducting the experiment with multiple PLCs.

- The Lab-Volt system is representative of an ICS. This assumption allows the experiment to focus on measuring and improving the accuracy and consistency of the simulated system rather than constructing a full-scale control system.
- The VFD and the HMI from the Lab-Volt system do not impact data collected from the Lab-Volt system even though they are entities in the same network as the Lab-Volt PLC.
- The analog I/O modules from the Lab-Volt system PLC and the simulated system PLC provide equivalent functionality even though their model numbers do not match.

4.2.3 Limitations.

Only the programs and software necessary for the operation of the Lab-Volt system are run when collecting data from the Lab-Volt system. Only the programs and software necessary for the operation of the simulated system are run when collecting data from the simulated system. These limitations reduce the number of external factors that could hinder the accuracy of water level recordings obtained from both systems.

4.3 Selection of the Response Variables

4.3.1 Response Variables.

The response variable for the experiment is the water level recorded at several points in time for each run of the Lab-Volt system and the simulated system. The

water level recordings acquired from each run of the Lab-Volt system and the simulated system are compared to determine the accuracy and consistency of the simulated system. Water level for both the Lab-Volt 3531 runs and the simulation runs is measured as percentage, with zero percent as empty and one hundred percent as full. Time is measured in seconds with the Python `time()` method. Both time and water level are measured with 0.001 precision.

4.4 Factors and Parameters

4.4.1 Factors.

The factor in this experiment is simulation speed, and it is listed in Table 1. Simulation speed is intentionally manipulated in order to produce differing output responses during the experiment and thus determine if the simulated system accurately and consistently models the Lab-Volt system. The factor is assigned the three levels shown in Table 1. The levels are chosen in order to test the accuracy and consistency of the system when configured with a small speedup factor and a large speedup factor.

4.4.2 Parameters.

The factors to be held constant are known as parameters and are the inflow rate to both the Lab-Volt system and the simulated system as well as the laptop used to collect data from both systems and run the simulated water tank. Pilot studies

Table 1. Factors.

Factor	**Level**
Simulation Speed	Real-time
	2*Real-time
	10*Real-time

showed that the PID controller in the Lab-Volt system responsible for maintaining a constant inflow rate produced minor fluctuations to the inflow rate each time it was forced to adjust the speed of the pump. The set point for the inflow rate was set to 10 liters per minute. However, due to the control delay associated with changing pump speed as well as the tuning of the PID controller, the inflow rate achieved by the pump in Lab-Volt system averaged 9.65 liters per minute. Thus, the inflow rate for the simulated system is also set to 9.65 liters per minute or 588.9 in^3/min. In this instance, control delay refers to the amount of time between the PLC sending a command to the VFD and the pump adjusting to the correct speed. The laptop used to collect data from both systems as well as run the simulated water tank is a Lenovo ThinkPad W541 with the Windows 10 Education (64) OS, Intel Core i7 processor, and 32GB RAM.

4.4.3 Test Environment.

Figure 9 portrays the Lab-Volt 3531 network. The PLC, VFD, HMI, and the laptop used for data collection communicate through a network switch. Figure 10 portrays the simulation network. It does not include the VFD and the HMI because they are not included in the simulation. The laptop communicates with the PLC through a network switch in order to collect data. The data collection laptop also runs the simulated water tank. The simulated water tank communicates with the PLC through a Y-box.

4.4.4 Experimental Design.

Validation testing is used to verify the accuracy and consistency of the simulation. The experiment compares the operation of the simulation to the operation of an actual water tank, specifically, the Lab-Volt 3531 Training System. The goal of the

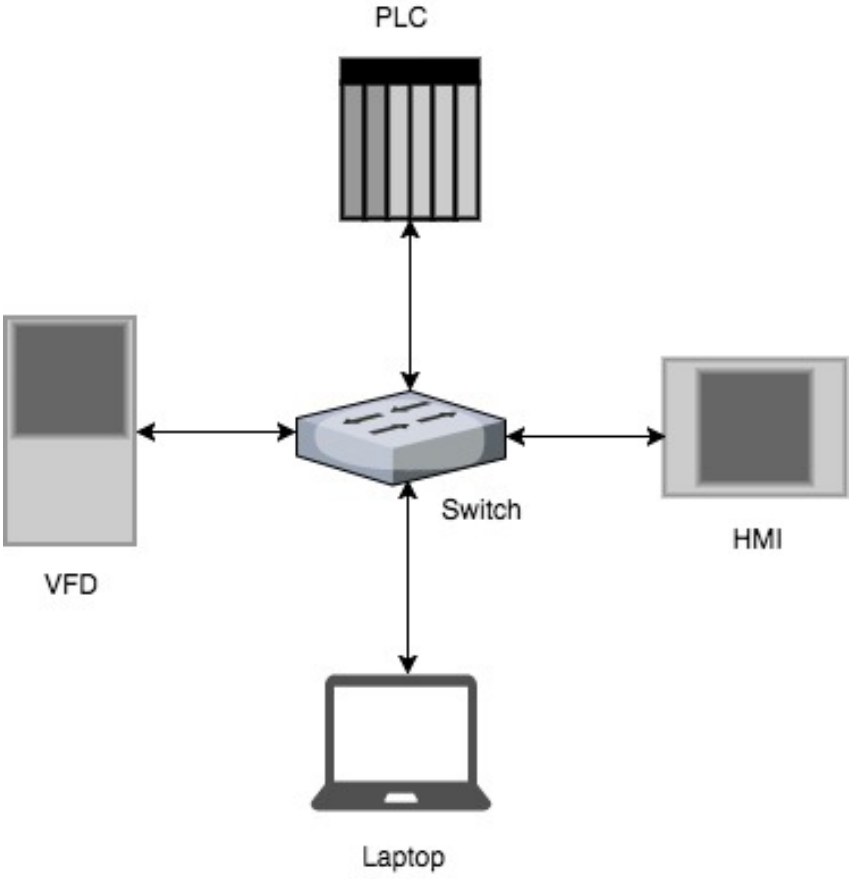

Figure 9. Lab-Volt 3531 Network.

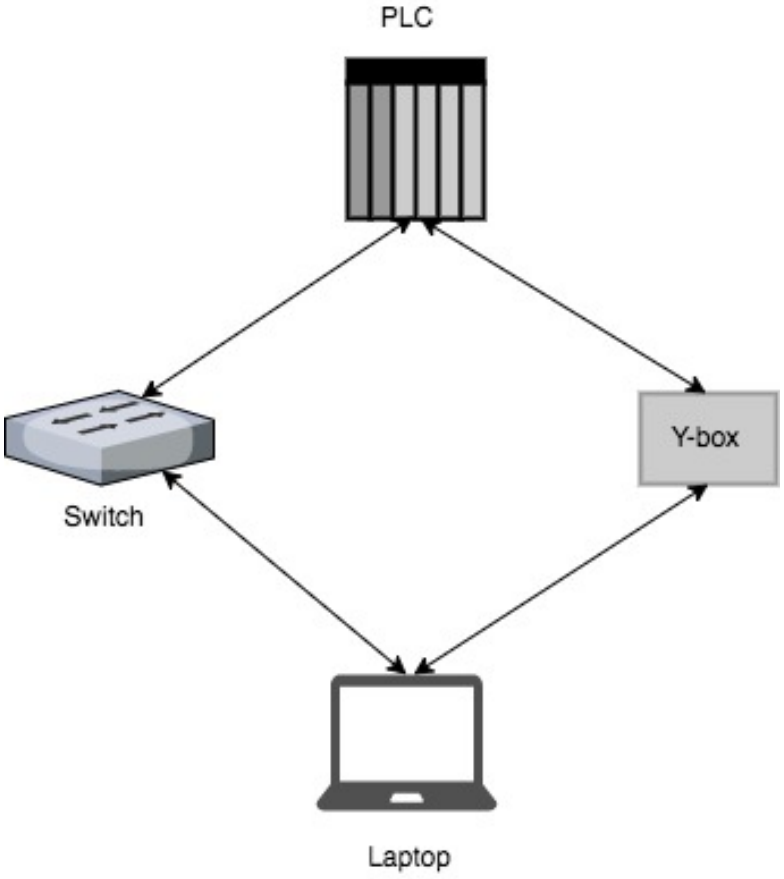

Figure 10. Simulation Network.

experiment is to show that the simulation reflects the normal operation of a real water tank when run at real-time and at faster simulation rates. Pilot studies showed the Lab-Volt system to be consistent from run to run. Specifically, the standard deviation between Lab-Volt runs was less than 0.01%. Thus, data from only three Lab-Volt runs is used in the experiment. The simulated water tank is also run three times at each speed specified in Table 1 for a total of nine runs. The data generated from the three Lab-Volt runs is compared with data obtained from the nine simulated water tank runs. Table 2 shows the names for each run in the experiment. In order to ensure accurate comparisons, all runs with the Lab-Volt 3531 and the simulation follow the procedure shown in Figure 11 which depicts a typical Lab-Volt 3531 run from the experiment. The procedure has the following steps.

- Start pump. The water level in the tank starts to rise.
- Execute Python script for reading and writing tags in the PLC ladder logic. The script sets the water level set point to thirty percent and begins to record the current water level and time at half second intervals starting at zero seconds. Water levels for both the Lab-Volt 3531 runs and the simulation runs are measured as percentage, with zero percent as empty and one hundred percent as full.
- As the water level in the tank approaches thirty percent, the Python script monitors the rising water level in order to detect water level steady state at thirty percent. Steady state occurs when the water level reaches the specified set point and remains at the set point with minimal deviations from the set point. This experiment considers deviations less than half of a percent above or below the set point to be minimal deviations. When using a well-tuned PID controller, the process variable reaches the set point with minimal overshoot. This experiment

considers overshoot less than three percent to be minimal overshoot. The PID controller then corrects the overshoot, and the process variable remains close to the set point with minimal deviations. The Python script detects steady state by checking if the current water level is within one tenth of a percent above or below the set point. After meeting this threshold, the Python script waits twenty-five seconds in order to let the PID controller correct any initial overshoot as shown in Figure 12. Afterwards, the Python script outputs confirmation that steady state has been reached. Pilot studies demonstrated that waiting twenty-five seconds is enough time for the PID controllers in both the Lab-Volt 3531 and the simulation to correct any initial overshoot and achieve water level steady state.

- Once the script detects steady state at thirty percent, it changes the water level set point to sixty percent, the water level in the tank starts to rise, and the script waits to detect water level steady state at sixty percent.

- Once the script detects water level steady state at sixty percent, it stops collecting data and ends the trial.

Table 2. Run Names.

Name	**Meaning**
LVTn	Lab-Volt 3531 Trial n
Simx1Tn	HiL System Real-time Trial n
Simx2Tn	HiL System 2*Real-time Trial n
Simx10Tn	HiL System 10*Real-time Trial n

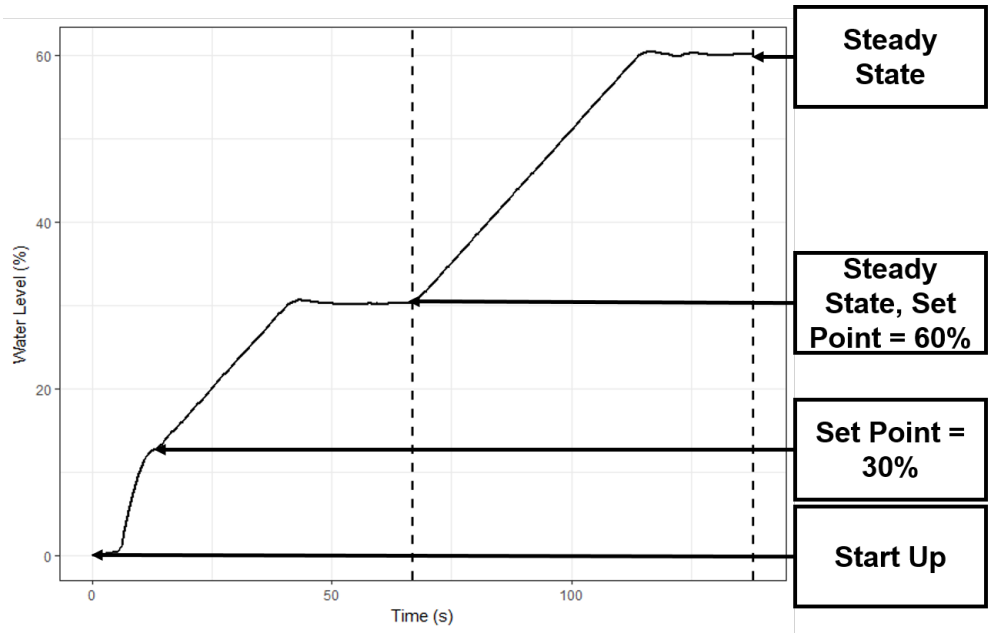

Figure 11. Experiment Procedure.

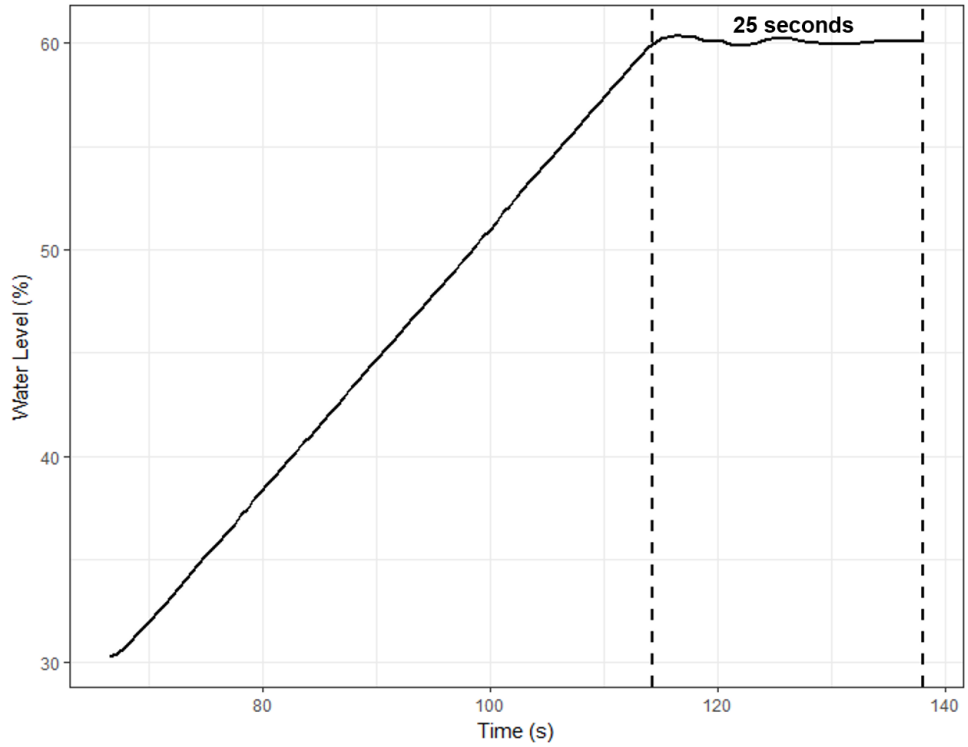

Figure 12. Correcting Set Point Overshoot.

4.4.5 Results and Analysis.

4.4.5.1 Raw Data.

Each run in the experiment generates a curve with water level on the y-axis and time on the x-axis. In order to ensure consistent analysis, the experiment considers an interval of time in which all three Lab-Volt 3531 runs achieve steady state at thirty percent, rise to sixty percent, and achieve steady state at sixty percent. After the three Lab-Volt runs are completed and the raw data is collected, another Python script adjusts all three curves so that they all reach forty-five percent full at the same time, specifically, zero seconds. In the first Lab-Volt curve, the Python script uses linear interpolation to calculate what time the curve reaches forty-five percent and subtracts this time from all other times in the curve [16]. The script repeats this process for the other two Lab-Volt curves. These adjustments shift the curves so that that they all center at zero seconds with a height of forty-five percent, the midpoint between the thirty percent and sixty percent set points. With all three curves adjusted, a one minute and twenty second time interval is selected. In this time interval, all three Lab-Volt runs complete the required behavior of achieving steady state at thirty percent, rising to sixty percent, and achieving steady state at sixty percent. When comparing curves from different runs, all curves are first adjusted so that they all reach forty-five percent full at zero seconds. Subsequent analysis compares only the portions of the curves containing the time interval previously described to ensure consistent comparisons. Having a method for consistent comparisons means that the analysis can accurately compare each run to every other run in the experiment.

4.4.5.2 Analysis.

Table 3 shows the matrix used to compare all of the runs. Each marked cell below the diagonal in Table 3 represents a comparison between the trial represented by the

cell's row and the trial represented by the cell's column. Note that the simulation run names are shortened in the matrix. Simx1Tn is denoted as Sx1Tn, Simx2Tn is denoted as Sx2Tn, and Simx10Tn is denoted as Sx10Tn. The experiment compares the curves from the trials by considering the average difference between them. Average difference represents the average distance between water level for the first curve and water level for the second curve at any point in time. In order to calculate the average difference between two curves, a Python script first adjusts the timestamps for each curve to account for speedup. For example, if the Sx1T1 run is being compared to the Sx10T1 run, all of the timestamps in the Sx10T1 run would be multiplied by ten to account for the speedup in the Sx10T1 run. Assume that the cell represented by this comparison has Sx10T1 as the row and Sx1T1 as the column. This cell can be seen in Table 3. Next, the Python script adjusts both curves so that they both reach forty-five percent full at the same time, specifically, zero seconds. In the Sx1T1 curve, the Python script uses linear interpolation to calculate what time the curve reaches forty-five percent and subtracts this time from all other times in the curve. The script repeats this process for the Sx10T1 curve. These adjustments shift both curves so that that they both center at zero seconds with a height of forty-five percent, the midpoint between the thirty percent and sixty percent set points. At this point, the script removes all portions from both of the curves that are not included in the one minute and twenty second time interval described earlier. Then the script iterates through the remaining timestamps for the Sx1T1 curve and performs linear interpolation to determine their corresponding heights in the Sx10T1 curve. Now, the Python script has three curves. The first curve is the Sx1T1 curve, and the second curve is the Sx10T1 curve. The third curve is a new Sx10T1 curve containing only the timestamps from the Sx1T1 curve and their associated heights calculated in the previous step with linear interpolation. With matching timestamps, the Sx1T1 and

Sx10T1 runs can be compared in a straightforward manner. Appendix A provides an example of the linear interpolation process used to generate matching timestamps for two different curves. Finally, the Python script iterates through the Sx1T1 curve and the new Sx10T1 curve. For each timestamp, the script calculates the absolute value of the difference between the water level in the Sx1T1 curve and the water level in the new Sx10T1 curve. After iterating through both curves, the Python script sums all of the absolute values, divides the sum by the number of timestamps, and outputs the quotient as the average difference between the two runs.

4.4.6 Evaluation Metrics.

The experiment has two evaluation metrics. The first metric measures whether or not the simulation completes the required behavior of achieving water level steady state at thirty percent, raising the water level to sixty percent, and achieving water steady state at sixty percent within the one minute and twenty second time interval. If a simulation run completes the required behavior within the time interval, the run passes the first metric. Otherwise, it fails the first metric. The second metric for the proposed simulation considers the average difference between two runs. The experiment uses the mean of the average differences between the three Lab-Volt runs as the threshold for the second evaluation metric. A high-fidelity simulation would have a similar average difference when compared to the Lab-Volt 3531. If the average difference between two runs is less than or equal to the mean of the average differences between the Lab-Volt runs, the runs would each pass the second metric. Otherwise, they fail the second metric.

Table 3. Test Matrix.

	LVT1	LVT2	LVT3	Sx1T1	Sx1T2	Sx1T3	Sx2T1	Sx2T2	Sx2T3	Sx10T1	Sx10T2	Sx10T3
LVT1												
LVT2	X											
LVT3	X	X										
Sx1T1	X	X	X									
Sx1T2	X	X	X	X								
Sx1T3	X	X	X	X	X							
Sx2T1	X	X	X	X	X	X						
Sx2T2	X	X	X	X	X	X	X					
Sx2T3	X	X	X	X	X	X	X	X				
Sx10T1	X	X	X	X	X	X	X	X	X			
Sx10T2	X	X	X	X	X	X	X	X	X	X		
Sx10T3	X	X	X	X	X	X	X	X	X	X	X	

4.4.7 System Boundaries.

Figure 13 depicts the System Under Test (SUT). The only factor is the simulation speed. The first metric is simply whether or not the simulation completes the required behavior. The second metric is the average difference associated with the particular simulation run. As mentioned in the previous section, the average difference is computed between all runs in the experiment. The main system parameter that affects the system output is the tuning of the PID controller.

In order to tune the simulation PID controller, the PID tuning from the Lab-Volt system is used as a baseline since it was properly tuned during initial set up. During set up of the Lab-Volt system, a trial and error method based upon knowledge of the effects of the P, I, and D terms respectively, was used to tune the PID controller for the Lab-Volt system. The P, I, and D terms were manipulated until the PID controller was able to limit the set point overshoot of the water level to less than three percent for both the thirty percent and sixty percent set points as well as well prevent set point deviations larger than half a percent after correcting the set point overshoot.

When run at real-time, the PID controller in the simulation uses the same tuning as the PID controller in the Lab-Volt system. In order to adjust the PID tuning for higher speed simulations, the PID was first adjusted for simulation runs with a

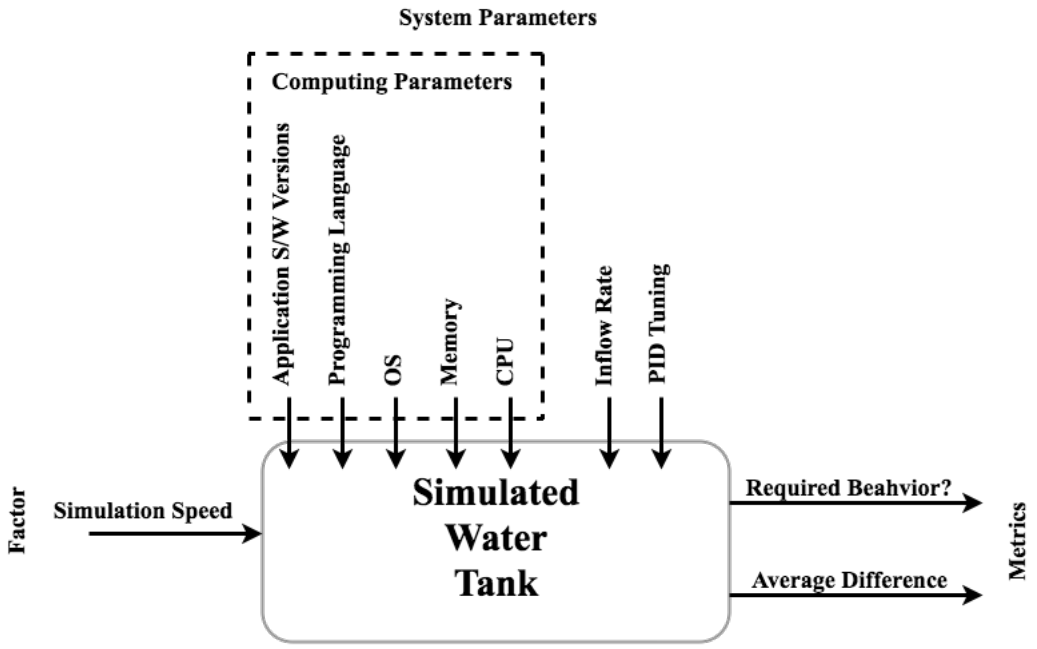

Figure 13. System Under Test.

speedup factor of ten. Pilot studies showed that the intuitive approach of dividing the PID parameters by the speedup factor was ineffective. Thus, a trial and error method based upon knowledge of the effects of the P, I, and D terms respectively, was again used to custom tune the PID controller for simulation runs with a speedup factor of ten. The P, I, and D terms were manipulated until the PID controller was able to limit the set point overshoot of the water level to less than three percent for both the thirty percent and sixty percent set points as well as well prevent set point deviations larger than half a percent after correcting the set point overshoot. The tuning for simulation runs with a speedup factor of two were then adjusted proportionally from the PID tuning for simulation runs with a speedup factor of ten. Table 4 shows the PID tunings the experiment uses for each simulation speed.

Table 4. PID Tunings.

	P	I	D	Sampling Time (ms)
Lab-Volt 3531	40	1.5	10	100
Simx1	40	1.5	10	100
Simx2	34	1.5	8.2	100
Simx10	10	1.5	1	100

4.5 Chapter Summary

This chapter details an experiment designed to evaluate the accuracy and consistency of the proposed control loop simulation. It describes the experiment design, data collection method, test environment, and approach for data analysis.

V. Results and Analysis

5.1 Overview

The following sections outline the results for the experiment described in the previous chapter. The chapter begins with a discussion of general results and then moves on to a detailed discussion of simulation consistency and the effects of simulation speedup.

5.2 Experiment Evaluation Metrics

5.2.1 Metric 1: Required Behavior.

All simulation runs completed the required behavior within the one minute and twenty second time interval. Regardless of simulation speed, all simulation runs passed the first evaluation metric of the experiment. In order to verify that each run passed the first metric, all runs were graphed. For each graph, the portion of the graph containing the one minute and twenty second time interval was visually inspected to ensure the simulation run represented by the graph achieved the required behavior.

Figures 14, 15, and 16 demonstrate the accuracy of the simulation in real-time, two times speed, and ten times speed, respectively. Note that the time scales for the Simx2 and the Simx10 lines were multiplied by their respective speedup factors to match real-time. Each of the three graphs represents a typical comparison of the simulation to the Lab-Volt system from the experiment. The slopes of the Lab-Volt system and Simx1, Simx2, and Simx10 lines are almost identical. The slight difference in slopes was most likely caused by variations to the inflow rate in the Lab-Volt system. As mentioned before, the PID controller in the Lab-Volt system responsible for maintaining a constant inflow rate produced minor fluctuations to the

inflow rate each time it was forced to adjust the speed of the pump. Another possible cause for the slight difference in slopes is the imprecise trial and error method used to tune the PID controller in the simulation. The main difference between the two lines occurs as they approach the sixty percent steady state. This difference resulted from the presence of control delay in the Lab-Volt system. In the Lab-Volt system, the control delay is the amount of time between the PLC sending a command to the valve and the valve adjusting the plug to the correct position. The simulation did not model this control delay and was therefore not affected by it.

Figure 17 shows how well the simulation run with speedup factors of two and ten matches the simulation when run at real-time. Note that the time scales for the Simx2 and the Simx10 lines were multiplied by their respective speedup factors to match real-time. The primary difference between the Simx1, Simx2, and Simx10 lines occurs as they approach the sixty percent steady state. This difference most likely resulted from the imprecise trial and error method used to tune the PID controller in the simulation.

5.2.2 Metric 2: Average Difference.

Figure 18 and Table 5 summarize the average differences between the Lab-Volt system and the simulation.

The average difference when comparing runs from the Lab-Volt 3531 is well below 0.1%. When run at real-time, the average difference between the simulation and

Table 5. Simulation Accuracy – Average Difference (%).

Comparison	Mean	Min	Max	StDev
Lab-Volt vs. Lab-Volt	0.055	0.045	0.061	0.009
Lab-Volt vs. Simx1	0.203	0.198	0.210	0.005
Lab-Volt vs. Simx2	0.194	0.182	0.204	0.007
Lab-Volt vs. Simx10	0.245	0.193	0.278	0.031

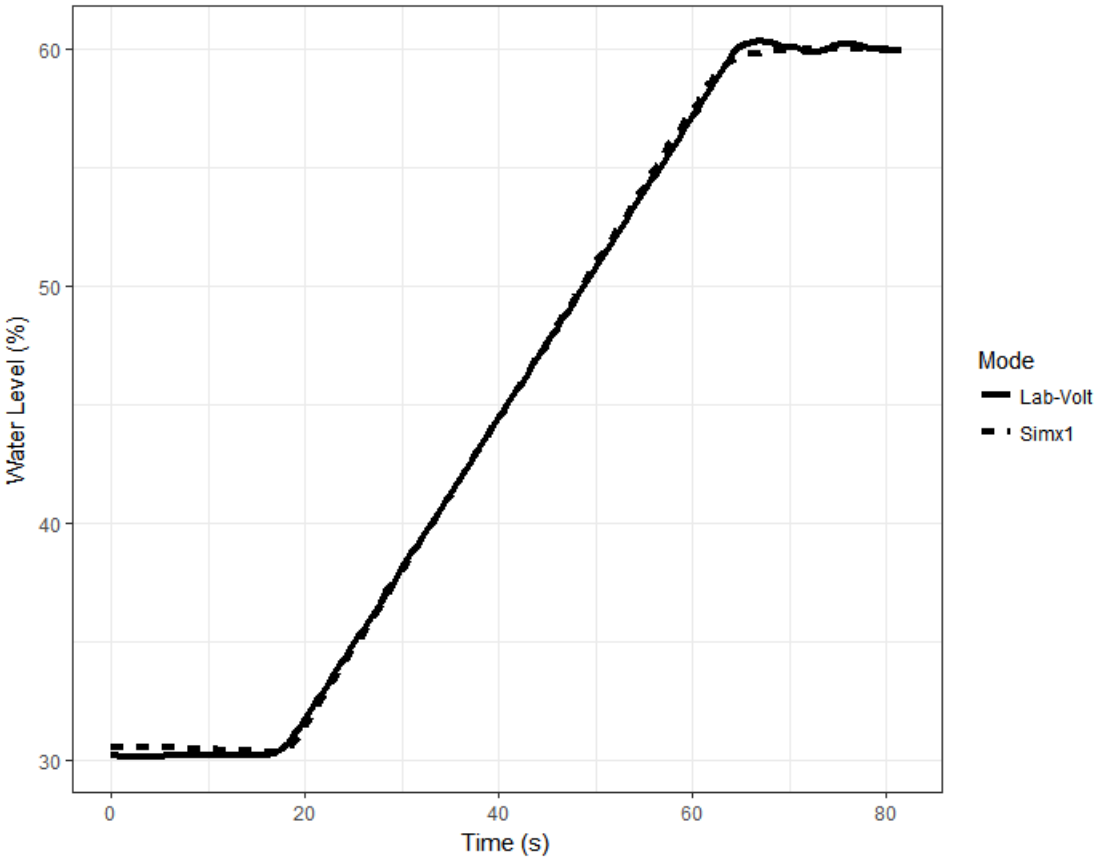

Figure 14. Lab-Volt vs. Simx1.

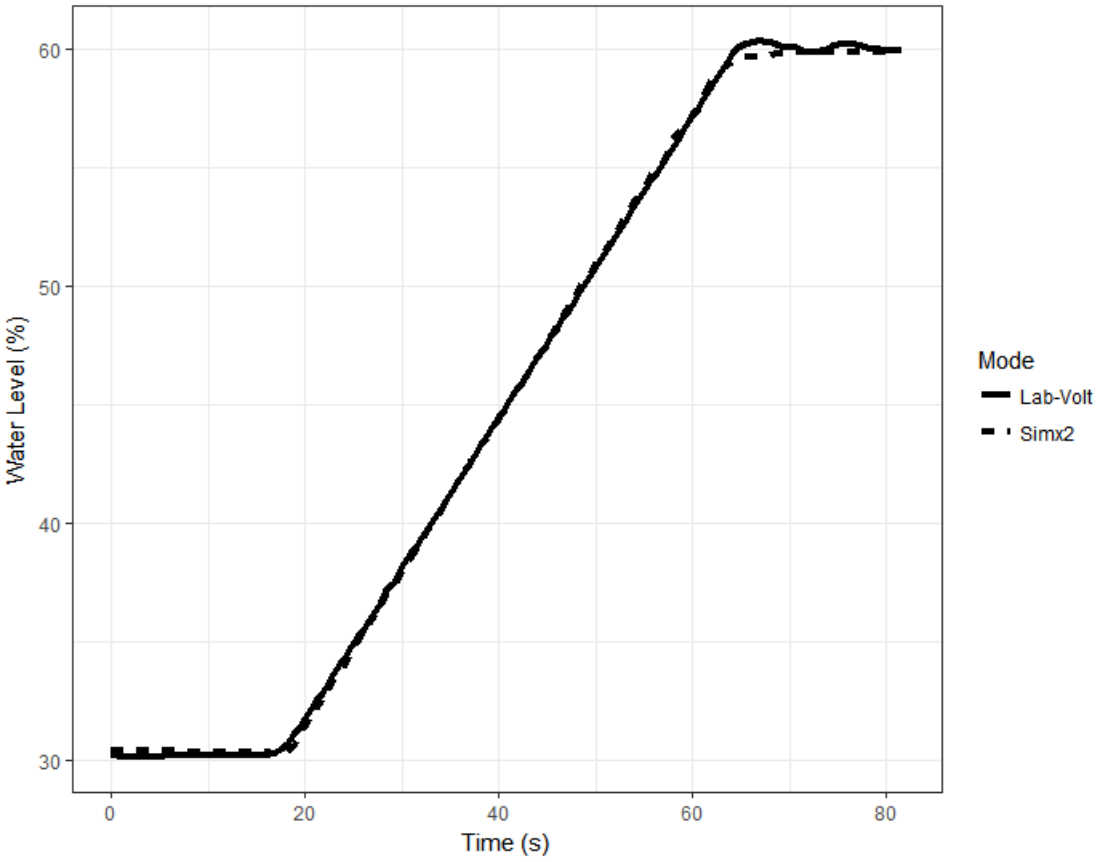

Figure 15. Lab-Volt vs. Simx2.

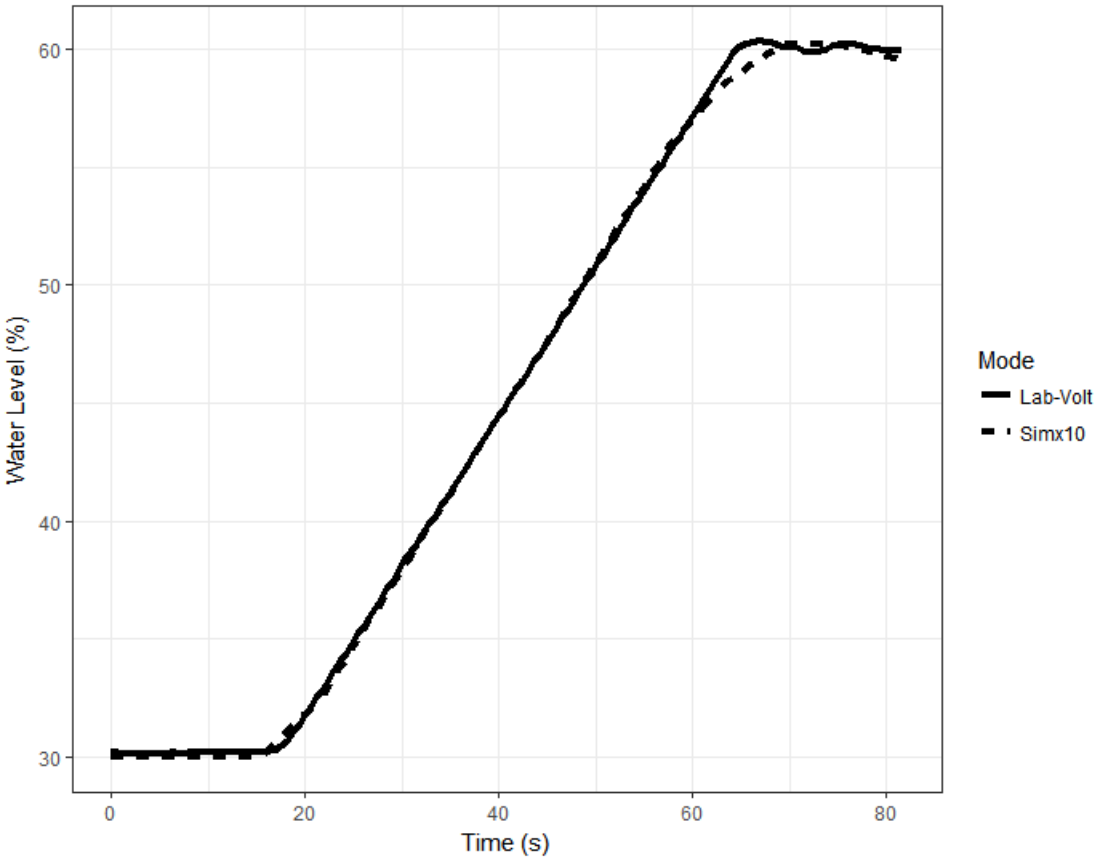

Figure 16. Lab-Volt vs. Simx10.

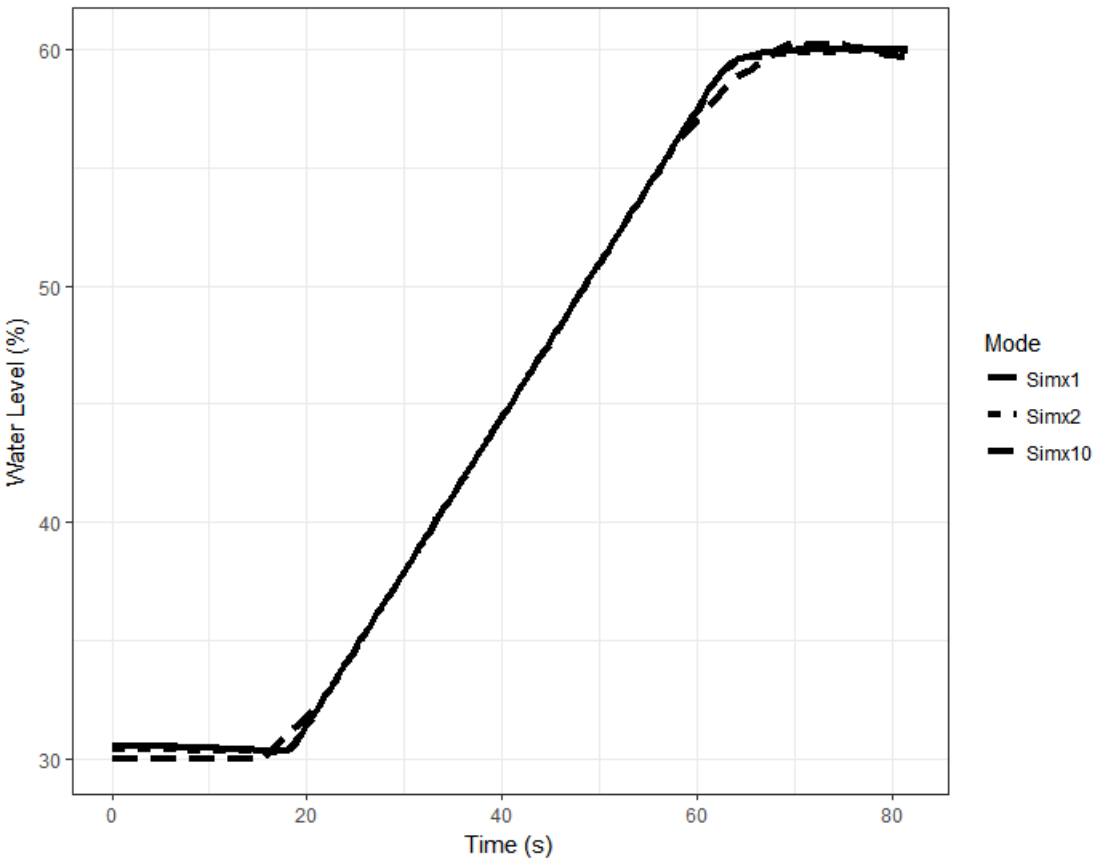

Figure 17. Simx1 vs. Simx2 vs. Simx10.

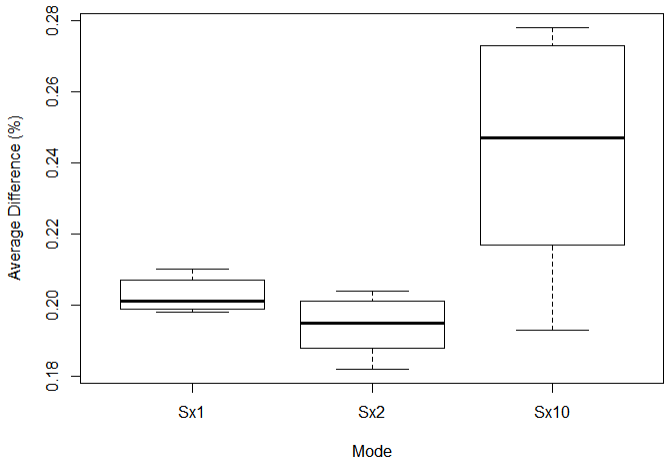

Figure 18. Lab-Volt vs. Simulation.

the Lab-Volt 3531 ranged from 0.198% to 0.21%. When run at two times faster than real-time, the average difference between the simulation and the Lab-Volt 3531 ranged from 0.182% to 0.204%. When run at ten times faster than real-time, the average difference between the simulation and the Lab-Volt 3531 ranged from 0.193% to 0.278%. These results show that the average difference between the simulation and the Lab-Volt 3531 was significantly higher than the average difference between Lab-Volt runs regardless of simulation speed. Thus, the proposed simulation did not pass the experiment's second evaluation metric.

Although the simulation did not pass the experiment's second evaluation metric, all of the simulation results are relatively consistent as shown by Figure 18. Despite not passing the evaluation metric, all of the simulation runs produced relatively low average differences that ranged from 0.182% to 0.278%. The mean of the average differences for the Simx2 runs was even slightly lower than the mean of the average differences for the Simx1 runs. In fact, a permutation test comparing the average differences from the Simx1 and Simx2 runs produced a p-value of 0.01354 which is greater than the 0.01 threshold for the 99% confidence level [17]. Thus, the permutation test shows that there is no significant difference between the mean of the average differences for the Simx2 runs and the mean of the average differences for the Simx1 runs at the 99% confidence level. The null hypothesis that the two means are equal cannot be rejected because the p-value is larger than 0.01. This supports that simulation accuracy is the same when the simulation is run at both real-time and two times real-time. Overall, the results of the experiment demonstrate that the proposed simulation consistently modeled the Lab-Volt 3531 with less than 0.28% error on average at any point in time.

5.3 Consistency

Table 6 demonstrates the consistency of multiple runs of the Lab-Volt system and the simulation system. For example, the first row of the table represents the consistency of the Lab-Volt system from run to run. The mean of the average differences between Lab-Volt runs was 0.055%, the minimum average difference was 0.045%, and the maximum average difference was 0.061%. The standard deviation between Lab-Volt runs was 0.009%. The other rows of the table represent the consistency of the simulation from run to run at each speed. When run at real-time, the average differences between the simulation runs are more consistent than the average differences between the Lab-Volt runs. The consistency in average differences decreases as simulation speed increases. The table shows that there is much less consistency from run to run when the simulation is run with a speedup factor of ten. The decrease in consistency may have resulted from a loss of precision when using linear interpolation to calculate curves from Simx10 runs. Since the simulation generated less points when it was run at higher speeds, there were less reference points for linear interpolation calculations. Consequently, Simx10 ten curves exhibit higher variability.

5.4 Simulation Speedup

Table 7 summarizes the average differences between the simulation run at real-time and the Lab-Volt system, the simulation run with a speedup factor of two, and the simulation run with a speedup factor of ten. The first row demonstrates that

Table 6. Run Consistency – Average Difference (%).

	Mean	**Min**	**Max**	**StDev**
Lab-Volt	0.055	0.045	0.061	0.009
Simx1	0.021	0.014	0.026	0.006
Simx2	0.037	0.031	0.048	0.009
Simx10	0.102	0.045	0.131	0.049

the simulation is consistent from run to run when run at real-time. The second and third rows demonstrate that when run at real-time, the simulation is consistent with the Lab-Volt system and the simulation when run with a speedup factor of two respectively. Finally, the fourth row demonstrates that the average difference between the simulation when run at real-time is consistent with the simulation when run with a speedup factor of ten. The average differences between the simulation when run at real-time and the simulation run with a speedup factor of two are much lower than the average differences between the simulation run at real-time and the simulation run with a speedup factor of ten. As mentioned earlier, this difference resulted from imprecise tuning of the PID controller in the simulation which had a greater effect when the simulation was run with a speedup factor of ten.

5.5 Chapter Summary

This chapter presents the results of the experiment described in Chapter IV. The simulation passed the first evaluation metric from the experiment, namely, completing the required behavior of achieving water level steady state at thirty percent, raising the water level to sixty percent, and achieving water steady state at sixty percent within the one minute and twenty second time interval. However, the simulation failed the second evaluation metric from the experiment, namely, average difference. Despite not passing the second evaluation metric, the results demonstrate that the simulation relatively consistently modeled the Lab-Volt system with a low average

Table 7. Effect of Speedup – Average Difference (%).

Comparison	Mean	Min	Max	StDev
Simx1 vs. Simx1	0.021	0.014	0.026	0.006
Simx1 vs. Lab-Volt	0.203	0.198	0.210	0.005
Simx1 vs. Simx2	0.070	0.056	0.081	0.008
Simx1 vs. Simx10	0.256	0.220	0.280	0.025

difference. The results show the feasibility of the proposed method for accelerating ICS cyber training exercises.

VI. Conclusions

6.1 Introduction

This chapter presents a summary of the research conclusions, impact, and future work for this research. It is important for ICS operators to receive quality training that provides hands-on exercises with real-world equipment in order to see the real-world impact of changes made to a control system and learn various techniques for defending against cyber attacks. Unfortunately, cyber events often have effects that take a significant amount of time to manifest, making high-fidelity training exercises require an infeasible amount of time. In addition, unforeseen effects from cyber events have the potential to damage the training equipment. This research sought to solve this problem by developing a method for accelerating ICS cyber security exercises.

6.2 Research Conclusions

This research developed a method for accelerating ICS cyber training exercises that involved modeling and predicting the impacts of a cyber event. The proposed method simulates the effects of a cyber event on a partially-simulated control system. Its speedup ability allows the system to replicate the effects of a cyber event at speeds faster than real-time, enabling exercise coordinators to predict the effects of a cyber event and conduct exercises in a reasonable time frame as well as prevent potential damage to equipment. Specifically, the method involved the creation of a test system that employed HiL simulation to speed up the dynamics of a simulated water tank while allowing a commercially-available PLC that controlled the tank to continue operating as intended. An experiment then compared the operation of the simulated system to the operation of an actual water tank regulated by the same model of PLC. The results of the experiment showed a significant difference between the operation of

the simulation and the real water tank. The average difference between the simulation and the real water tank ranged from 0.182% to 0.278%. The mean of the average differences for all comparisons between the simulation and the real water tank was 0.214% which is well above 0.055%, the mean of the average differences for the Lab-Volt runs. However, the experiment showed that the average difference between the simulation and the real water tank was consistently low, specifically, less than 0.28% with a standard deviation of 0.028%, even when the simulation was run a speeds much faster than real-time.

The goal of this research was to develop a method that augments ICS cyber security training environments by allowing exercise coordinators to model and predict the effects of a cyber event in rapid-time. While the experiment showed a significant difference between the simulated system and the real water tank, all tests showed that the physical process responded appropriately to the cyber event. In each test case, when the Python script changed the set point, the PLC was able to appropriately respond to the change and adjust the water level in the tank. This research achieved its goal because the results of the experiment show that the simulation would allow operators to predict the impact of the simulated cyber event in real-time, two times speed, and ten times speed. However, there is much room for improving the accuracy and consistency of the method especially when the system is run at ten times speed.

In order to fulfill its goal, this research developed a mobile, cost effective, realistic ICS training environment that has the ability to speed up the simulated industrial process while allowing the control hardware to continue operating as intended. This research proposed the use of the mobile training environment as the method. Finally, this research hypothesized that the proposed method enables operators and exercise coordinators to replicate cyber events in rapid-time in order to predict their effects and

limit potential unforeseen damage to control system equipment. Since the research goal was achieved, the research proves this hypothesis.

6.3 Research Contributions

This research developed a means for accelerating ICS cyber training exercises in the form of a system that speeds up a simulated physical process while allowing the PLC controlling it to continue operating as intended. Using this kind of simulated system in conjunction with a full-scale ICS enables operators to train in robust, high-fidelity environments efficiently and safely. In effect, operators are given the ability to speed up time and see the future consequences of their actions while limiting the possibility of physical damage to the control equipment.

6.4 Limitations of this Research

The main limitation of the method developed by this research is that it cannot replicate certain kinds of cyber events. For example, if malware that waits a specific period of time before acting is deployed, the method would only be able to replicate the impact of the malware in real-time, not at higher speeds. The speedup capability of the method would not impact the execution of the malware since it operates independently with its own means for tracking the passage time. Thus, exercise coordinators would not be able to predict the effects of such time based cyber events. In addition, the method would not be able to speed up the effects of a logic attack that waits for a specific trigger such as the physical pushing of a button by an operator. These limitations show that in some scenarios the method would not be able to replicate the effects of cyber events at speeds faster than real-time and help prevent potential damage to exercise equipment.

6.5 Recommendations for Future Work

6.5.1 Model Improvements.

While this research demonstrates the feasibility of the simulation speedup method, further work is needed to improve its accuracy and consistency especially when run at higher simulation speeds. One way to achieve this is to improve the model of the water tank. Specifically, the model could be expanded to incorporate features of the Lab-Volt system that it currently does not simulate such as control delay. The fidelity of the accelerated simulation heavily depends on the accuracy and fidelity of the physical process model.

6.5.2 Testing.

Additional testing is required to determine how well the simulation replicates a variety of operating conditions as well as the effects of various other cyber events. This research only demonstrated the feasibility of the proposed method for accelerating ICS cyber security exercises by developing a system that has only been shown to replicate one specific cyber event, namely, changing the water level set point. Further experimentation needs to determine how well the system can simulate other cyber events especially cyber attacks before using it to augment ICS cyber security training exercises.

6.5.3 Improvements to the Method.

The first improvement needed to enhance the method for accelerating ICS cyber security training exercises is a universal approach for adjusting PID tuning parameters. The trial and error method used in this research did not guarantee optimal tuning. It only guaranteed that the simulation would be able to complete the required behavior of achieving water level steady state at thirty percent, raising the water level

to sixty percent, and achieving water level steady state at sixty percent. Precise tuning of the PID would improve the method's accuracy and consistency especially at higher speeds.

The method can also be expanded to include additional interconnected processes and components. A water tank represents only one component of ICSs, and the ControlLogix PLC represents one of several PLCs used in industry. Including other interconnected processes and components would allow the method to replicate a wider variety of control systems and thus augment and accelerate a wider variety of ICS cyber security training exercises.

While speedup factors greater than 10 might work, the maximum speedup factor tested in this research was 10. Future work can test the upper bound of the speedup factor to determine how fast the proposed method can accurately and consistently accelerate ICS cyber security training exercises.

Finally, the proposed method for augmenting ICS cyber training environments needs a way to transfer events from full-scale, real-world testbeds such as the Level 4 environments described in Chapter II. Currently, if an ICS cyber exercise is using the proposed method to augment a Level 4 environment, the only way to duplicate an attack launched against a PLC in a Level 4 environment is to repeat the attack on the simulation PLC. A possible way to seamlessly and concurrently replicate cyber events from a Level 4 cyber environment in the simulation system is to mirror the traffic from the Level 4 environment in the simulation environment. This is important as the goal of the proposed method is to augment environments similar to Level 4 environments.

6.6 Chapter Summary

This chapter presents the conclusions and impacts of the research for developing a method for accelerating ICS cyber security training exercises. It concludes with several recommendations for future related research.

Appendix A. Example of Linear Interpolation Process

Tables 8 and 9 provide an example of the linear interpolation process used to generate matching timestamps for two different curves, namely, Sx1T1 and Sx10T1. In Table 8, columns one and two show the original heights (OH) and times (OT) collected from the Sx1T1 run respectively. Columns four and five show the original heights and times collected from the Sx10T1 respectively. Columns three and six show the times from the Sx1T1 and the Sx10T1 runs respectively after the times in the Sx10T1 have been multiplied by ten to account for the speedup and both curves have been adjusted (AT) so that they both reach forty-five percent full at zero seconds.

Table 8. Example Part 1.

Sx1T1-OH(%)	**Sx1T1-OT(s)**	**Sx1T1-AT(s)**	**Sx10T1-OH(%)**	**Sx10T1-OT(s)**	**Sx10T1-AT(s)**
0.502506256	0	-93.7166768	2.258136749	0	-541.2049998
0.863681793	0.599999905	-93.11667689	6.187103271	0.598999977	-535.215
1.22328949	1.199999809	-92.51667699	9.899368286	1.19900012	-529.2149986
1.645706177	1.799999952	-91.91667685	13.79221725	1.799000025	-523.2149995
2.123088837	2.399999857	-91.31667694	17.70234299	2.398000002	-517.2249998
2.48740387	3	-90.7166768	21.58576965	2.998000145	-511.2249983
2.853290558	3.599999905	-90.11667689	25.49903488	3.598999977	-505.215
3.21603775	4.200999975	-89.51567682	29.20815659	4.197999954	-499.2250002
3.674575806	4.80099988	-88.91567692	32.00020599	4.798000097	-493.2249988
4.015335083	5.400999784	-88.31567702	32.845047	5.397000074	-487.234999
4.41891098	6.000999928	-87.71567687	32.81991959	5.996999979	-481.235
4.780086517	6.601999998	-87.1146768	32.50585175	6.597000122	-475.2349986
5.150684357	7.20299983	-86.51367697	32.19335938	7.197000027	-469.2349995
5.617073059	7.802999973	-85.91367683	31.92797089	7.798000097	-463.2249988
5.987670898	8.403999805	-85.31267699	31.71126556	8.398000002	-457.2249998
6.356700897	9.003999949	-84.71267685	31.54480934	8.996999979	-451.235

Continued on next page

Table 8 – continued from previous page

Sx1T1-OH(%)	Sx1T1-OT(s)	Sx1T1-AT(s)	Sx10T1-OH(%)	Sx10T1-OT(s)	Sx10T1-AT(s)
6.799533844	9.603999853	-84.11267695	31.40033913	9.595999956	-445.2450002
7.11359787	10.20499992	-83.51167688	31.25272942	10.19500017	-439.2549981
7.558002472	10.80599999	-82.91067681	31.1804924	10.79500008	-433.254999
7.920749664	11.4059999	-82.3106769	31.08156204	11.39400005	-427.2649993
8.291347504	12.00699997	-81.70967683	30.98420143	11.99399996	-421.2650002
8.757736206	12.60699987	-81.10967693	30.88684082	12.59300017	-415.2749981
9.128334045	13.20799994	-80.50867686	30.79105186	13.19400001	-409.2649997
9.497364044	13.80799985	-79.90867695	30.7172451	13.79400015	-403.2649983
9.902507782	14.40899992	-79.30767688	30.64344025	14.39400005	-397.2649993
10.26211166	15.00899982	-78.70767698	30.59475899	14.99500012	-391.2549986
10.71122742	15.6079998	-78.108677	30.54607964	15.59500003	-385.2549995
11.06455231	16.20799994	-77.50867686	30.49739838	16.1960001	-379.2449988
11.43515015	16.80799985	-76.90867695	30.44714928	16.796	-373.2449998
11.9188118	17.40899992	-76.30767688	30.39846802	17.39700007	-367.2349991
12.28627014	18.00999999	-75.70667681	30.37491417	17.99699998	-361.235
12.71025848	18.61099982	-75.10567698	30.32623291	18.59800005	-355.2249993
13.05730438	19.21199989	-74.50467691	30.30110741	19.19799995	-349.2250003
13.40434647	19.8119998	-73.904677	30.27755356	19.7980001	-343.2249988
13.85817337	20.41099977	-73.30567703	30.25242805	20.39900017	-337.2149981
14.19893265	21.00999999	-72.70667681	30.2288723	20.99900007	-331.2149991
14.56796265	21.61099982	-72.10567698	30.2037468	21.59899998	-325.215
14.91971588	22.21099997	-71.50567683	30.18019295	22.19799995	-319.2250003
15.35155487	22.81099987	-70.90567693	30.17862129	22.7980001	-313.2249988
15.72215271	23.41199994	-70.30467686	30.15506744	23.39900017	-307.2149981
16.17126846	24.01099992	-69.70567688	30.13151169	24	-301.2049998
16.53087616	24.6099999	-69.1066769	30.12994194	24.60000014	-295.2049984
16.8889122	25.2099998	-68.506677	30.10638618	25.20000005	-289.2049993
17.34116745	25.80899978	-67.90767702	30.10638618	25.79900002	-283.2149996
17.66465378	26.40799999	-67.30867681	30.10638618	26.398	-277.2249998

Continued on next page

Table 8 – continued from previous page

Sx1T1-OH(%)	Sx1T1-OT(s)	Sx1T1-AT(s)	Sx10T1-OH(%)	Sx10T1-OT(s)	Sx10T1-AT(s)
18.12319183	27.0079999	-66.7086769	30.08126068	26.99699998	-271.235
18.48750687	27.6079998	-66.108677	30.08126068	27.59599996	-265.2450002
18.85496521	28.20799994	-65.50867686	30.05613709	28.19500017	-259.2549981
19.3056488	28.80799985	-64.90867695	30.05613709	28.79500008	-253.254999
19.66368485	29.40699983	-64.30967697	30.05613709	29.39400005	-247.2649993
20.02329063	30.00699997	-63.70967683	30.05770683	29.99399996	-241.2650002
20.42215538	30.60699987	-63.10967693	30.05770683	30.59300017	-235.2749981
20.77861977	31.20699978	-62.50967702	30.03258133	31.19300008	-229.274999
21.24814987	31.80699992	-61.90967688	30.03258133	31.79299998	-223.275
21.60775566	32.4059999	-61.3106769	30.03258133	32.39199996	-217.2850002
21.96736145	33.0059998	-60.710677	30.03258133	32.9920001	-211.2849988
22.43060875	33.60499978	-60.11167702	30.03258133	33.59100008	-205.294999
22.78079224	34.20499992	-59.51167688	30.03258133	34.19000006	-199.3049992
23.19064903	34.80499983	-58.91167697	30.03258133	34.78999996	-193.3050002
23.53455162	35.40399981	-58.31267699	30.03258133	35.38900018	-187.314998
23.90357971	36.00399995	-57.71267685	30.00745583	35.98900008	-181.314999
24.36682701	36.60399985	-57.11267695	30.00902557	36.59000015	-175.3049983
24.73428345	37.20299983	-56.51367697	30.00902557	37.19000006	-169.3049992
25.10331154	37.80199981	-55.91467699	30.00902557	37.79100013	-163.2949985
25.46291733	38.40099978	-55.31567702	30.00902557	38.39100003	-157.2949995
25.87120438	39	-54.7166768	30.00745583	38.9920001	-151.2849988
26.23394966	39.5999999	-54.1166769	30.00902557	39.59200001	-145.2849997
26.68777657	40.19999981	-53.51667699	30.00745583	40.19099998	-139.295
27.05366325	40.79899979	-52.91767701	30.00902557	40.79100013	-133.2949985
27.4211216	41.398	-52.3186768	30.00745583	41.39100003	-127.2949995
27.87965775	41.99799991	-51.71867689	30.00745583	41.99000001	-121.3049997
28.24554443	42.59899998	-51.11767682	30.00902557	42.59000015	-115.3049983
28.65225983	43.19999981	-50.51667699	30.00745583	43.19000006	-109.3049992
28.97574806	43.79899979	-49.91767701	30.00902557	43.78900003	-103.3149995

Continued on next page

Table 8 – continued from previous page

Sx1T1-OH(%)	Sx1T1-OT(s)	Sx1T1-AT(s)	Sx10T1-OH(%)	Sx10T1-OT(s)	Sx10T1-AT(s)
29.2678299	44.398	-49.3186768	30.00902557	44.38900018	-97.31499798
29.56776237	44.99799991	-48.71867689	30.00902557	44.98900008	-91.31499898
29.74521065	45.59799981	-48.11867699	30.00902557	45.58899999	-85.31499988
29.91951752	46.19799995	-47.51867685	30.00902557	46.18900013	-79.31499848
30.04828453	46.79799986	-46.91867694	30.00902557	46.78900003	-73.31499948
30.1597786	47.39899993	-46.31767687	30.00902557	47.38800001	-67.32499968
30.28226471	47.99899983	-45.71767697	30.00902557	47.98699999	-61.33499988
30.33722496	48.59799981	-45.11867699	30.00745583	48.58599997	-55.34500008
30.40632057	49.19899988	-44.51767692	30.00745583	49.18500018	-49.35499798
30.45028877	49.79899979	-43.91767701	29.98390198	49.78600001	-43.34499968
30.47384453	50.398	-43.3186768	29.98390198	50.38700008	-37.33499898
30.50682068	50.99899983	-42.71767697	30.00902557	51.58700013	-25.33499848
30.52095413	51.59799981	-42.11867699	32.43361664	52.18799996	-19.32500018
30.53037643	52.19699979	-41.51967701	36.32175827	52.78800011	-13.32499868
30.54607964	52.79699993	-40.91967687	40.23188019	53.38800001	-7.324999685
30.54607964	53.39599991	-40.32067689	44.13729477	53.98699999	-1.334999885
30.54450989	53.99599981	-39.72067699	48.00815964	54.58599997	4.654999915
30.54607964	54.59699988	-39.11967692	51.95597458	55.18500018	10.64500202
30.54607964	55.19799995	-38.51867685	55.84254456	55.78600001	16.65500032
30.54607964	55.79799986	-37.91867694	58.63145065	56.38499999	22.64500012
30.54293823	56.39899993	-37.31767687	60.25674438	56.98399997	28.63499992
30.52095413	56.99899983	-36.71767697	60.16095352	57.58400011	34.63500132
30.52095413	57.59899998	-36.11767682	59.60348511	58.18500018	40.64500202
30.51781273	58.19899988	-35.51767692	59.18106461	58.78600001	46.65500032
30.49739838	58.79799986	-34.91867694	59.08213425	59.38499999	52.64500012
30.49739838	59.39899993	-34.31767687	59.15594101	59.98500013	58.64500152
30.47227478	59.99899983	-33.71767697	59.33181763	60.58400011	64.63500132
30.47227478	60.59899998	-33.11767682	59.4558754	61.18500018	70.64500202
30.47227478	61.19899988	-32.51767692	59.53282166	61.78500009	76.64500112

Continued on next page

Table 8 – continued from previous page

Sx1T1-OH(%)	Sx1T1-OT(s)	Sx1T1-AT(s)	Sx10T1-OH(%)	Sx10T1-OT(s)	Sx10T1-AT(s)
30.44714928	61.79799986	-31.91867694	59.57992935	62.38600016	82.65500182
30.44714928	62.398	-31.3186768	59.60505676	62.98699999	88.66500012
30.44400787	62.99799991	-30.71867689	59.6286087	63.58700013	94.66500152
30.42359352	63.59799981	-30.11867699	59.65373611	64.18700004	100.6650006
30.42045403	64.19799995	-29.51867685	59.67729187	64.78600001	106.6550003
30.39846802	64.79799986	-28.91867694	59.72754288	65.38499999	112.6450001
30.39846802	65.39899993	-28.31767687	59.75109863	65.98500013	118.6450015
30.37491417	66	-27.7166768	59.77465057	66.58400011	124.6350013
30.37491417	66.60099983	-27.11567697	59.79977798	67.18400002	130.6350004
30.37491417	67.20099998	-26.51567682	59.82333374	67.78299999	136.6250001
30.34978867	67.80099988	-25.91567692	59.82333374	68.38300014	142.6250016
30.34978867	68.39999986	-25.31667694	59.84688568	68.98300004	148.6250006
30.34978867	68.99899983	-24.71767697	59.84688568	69.58300018	154.625002
30.32466316	69.59799981	-24.11867699	59.87044144	70.18300009	160.6250011
30.32466316	70.79699993	-22.91967687	59.87044144	70.78400016	166.6350018
30.56178284	71.39699984	-22.31967696	59.89556885	71.38400006	172.6350008
30.92453003	71.99699998	-21.71967682	59.91912079	71.98399997	178.6349999
31.28570557	72.59799981	-21.11867699	59.92069244	72.58300018	184.625002
31.64531136	73.19799995	-20.51867685	59.91912079	73.18400002	190.6350004
32.09285736	73.79799986	-19.91867694	59.92069244	73.78400016	196.6350018
32.45088959	74.39699984	-19.31967696	59.9442482	74.38400006	202.6350008
32.80735779	74.99699998	-18.71967682	59.9442482	74.98500013	208.6450015
33.22349548	75.59699988	-18.11967692	59.9442482	75.58599997	214.6549999
33.59095001	76.19799995	-17.51867685	59.9442482	76.18500018	220.645002
34.04948807	76.79899979	-16.91767701	59.9442482	76.78600001	226.6550003
34.41694641	77.39899993	-16.31767687	59.9442482	77.38600016	232.6550018
34.77655029	77.99899983	-15.71767697	59.96780396	77.98699999	238.6650001
35.22723389	78.59799981	-15.11867699	59.9693718	78.58599997	244.6549999
35.52246094	79.19799995	-14.51867685	59.9693718	79.18500018	250.645002

Continued on next page

Table 8 – continued from previous page

Sx1T1-OH(%)	Sx1T1-OT(s)	Sx1T1-AT(s)	Sx10T1-OH(%)	Sx10T1-OT(s)	Sx10T1-AT(s)
35.98727798	79.79799986	-13.91867694	59.96780396	79.78500009	256.6450011
36.35002136	80.39699984	-13.31967696	59.9693718	80.38499999	262.6450001
36.71276855	80.99699998	-12.71967682	59.9693718	80.98500013	268.6450015
37.18701172	81.59699988	-12.11967692	59.99292755	81.58400011	274.6350013
37.5497551	82.19799995	-11.51867685	59.99292755	82.18500018	280.645002
37.90465164	82.79899979	-10.91767701	59.99292755	82.78500009	286.6450011
38.30351257	83.398	-10.3186768	59.99292755	83.38600016	292.6550018
38.65998077	83.99799991	-9.718676889	59.9693718	83.98600006	298.6550008
39.10909271	84.59899998	-9.117676819	59.99292755	84.58599997	304.6549999
39.47341156	85.19899988	-8.517676919	59.99292755	85.18500018	310.645002
39.83929825	85.79899979	-7.917677009	59.99292755	85.78500009	316.6450011
40.29469299	86.39899993	-7.317676869	59.99292755	86.38499999	322.6450001
40.65901184	86.99899983	-6.717676969	59.99292755	86.98399997	328.6349999
41.07043839	87.59799981	-6.118676989	59.99292755	87.58300018	334.625002
41.43161392	88.19699979	-5.519677009	59.99292755	88.18300009	340.6250011
41.79436111	88.79599977	-4.920677029	59.99292755	88.78299999	346.6250001
42.24975586	89.39499998	-4.321676819	59.99292755	89.38400006	352.6350008
42.6172142	89.99499989	-3.721676909	59.99292755	89.98399997	358.6349999
42.9862442	90.59499979	-3.121677009	59.99292755	90.58400011	364.6350013
43.33171463	91.19499993	-2.521676869	59.99292755	91.18300009	370.6250011
43.75099182	91.79399991	-1.922676889	59.99292755	91.78299999	376.6250001
44.21738052	92.39399982	-1.322676979	59.99292755	92.38199997	382.6149999
44.57070541	92.99299979	-0.723677009	59.99292755	92.98100019	388.6050021
44.9271698	93.59299994	-0.123676859	59.99449921	93.58100009	394.6050011
45.28049469	94.19299984	0.476323041	59.99292755	94.18200016	400.6150018
45.73589325	94.79299998	1.076323181	59.99292755	94.78200006	406.6150008
46.10020828	95.39399982	1.677323021	59.99449921	95.38199997	412.6149999
46.51320648	95.99299979	2.276322991			
46.87752151	96.59299994	2.876323141			

Continued on next page

Table 8 – continued from previous page

Sx1T1-OH(%)	Sx1T1-OT(s)	Sx1T1-AT(s)	Sx10T1-OH(%)	Sx10T1-OT(s)	Sx10T1-AT(s)
47.24026871	97.19399977	3.477322971			
47.6909523	97.79399991	4.077323111			
48.05056	98.39399982	4.677323021			
48.41016388	98.99299979	5.276322991			
48.87498474	99.59299994	5.876323141			
49.20318222	100.1929998	6.476323001			
49.66799927	100.7919998	7.075323001			
50.04016876	101.392	7.675323201			
50.40448761	101.9919999	8.275323101			
50.85046005	102.5919998	8.875323001			
51.21006775	103.191	9.474323201			
51.54140472	103.7909999	10.0743231			
51.98266983	104.3899999	10.6733231			
52.33756256	104.9899998	11.273323			
52.80238342	105.589	11.8723232			
53.14942551	106.1899998	12.473323			
53.51060104	106.79	13.0733232			
53.9597168	107.3909998	13.674323			
54.27692413	107.9899998	14.273323			
54.72760773	108.5899999	14.8733231			
55.09663391	109.1899998	15.473323			
55.46880341	109.7889998	16.072323			
55.93205261	110.3889999	16.6723231			
56.30265045	110.9899998	17.273323			
56.66696548	111.589	17.8723232			
57.07996368	112.1889999	18.4723231			
57.42229462	112.79	19.0733232			
57.78504181	113.3899999	19.6733231			
58.25771332	113.9889998	20.272323			

Continued on next page

Table 8 – continued from previous page

Sx1T1-OH(%)	Sx1T1-OT(s)	Sx1T1-AT(s)	Sx10T1-OH(%)	Sx10T1-OT(s)	Sx10T1-AT(s)
58.5984726	114.589	20.8723232			
58.91254044	115.1899998	21.473323			
59.22346497	115.79	22.0733232			
59.40876389	116.3909998	22.674323			
59.57050705	116.9909999	23.2743231			
59.64274216	117.5919998	23.875323			
59.72911072	118.191	24.4743232			
59.79192352	118.7909999	25.0743231			
59.83746338	119.3909998	25.674323			
59.87201309	119.9899998	26.273323			
59.91440964	120.5899999	26.8733231			
59.92069244	121.1899998	27.473323			
59.94581604	121.79	28.0733232			
59.94581604	122.3899999	28.6733231			
59.9693718	122.9889998	29.272323			
59.9693718	123.5899999	29.8733231			
59.99449921	124.1889999	30.4723231			
59.99292755	124.79	31.0733232			
59.99449921	125.3909998	31.674323			
59.99449921	125.9899998	32.273323			
59.99292755	126.5899999	32.8733231			
59.99292755	127.1889999	33.4723231			
59.99449921	127.7889998	34.072323			
60.0196228	128.3889999	34.6723231			
60.01805115	128.9889998	35.272323			
59.99449921	129.5879998	35.871323			
60.0196228	130.1889999	36.4723231			
60.0196228	130.7889998	37.072323			
60.00234985	131.3879998	37.671323			

Continued on next page

Table 8 – continued from previous page

Sx1T1-OH(%)	Sx1T1-OT(s)	Sx1T1-AT(s)	Sx10T1-OH(%)	Sx10T1-OT(s)	Sx10T1-AT(s)
59.99449921	131.987	38.2703232			
60.0196228	132.5869999	38.8703231			
60.0196228	133.188	39.4713232			
59.9976387	133.7889998	40.072323			
59.9976387	134.3879998	40.671323			
59.99449921	134.987	41.2703232			
59.99449921	135.5869999	41.8703231			
59.99449921	136.1869998	42.470323			
60.01805115	136.7869999	43.0703231			
60.02119446	137.3859999	43.6693231			
59.99449921	137.9849999	44.2683231			
59.9976387	138.586	44.8693232			
59.9976387	139.1859999	45.4693231			
59.9976387	139.7849998	46.068323			
60.0227623	140.3839998	46.667323			
59.9976387	140.9829998	47.266323			
59.9976387	141.5829999	47.8663231			
59.9976387	142.1829998	48.466323			
59.9976387	142.7819998	49.065323			
59.9976387	143.3829999	49.6663231			
59.9976387	143.9829998	50.266323			

In Table 9, column three shows the timestamps from the one minute and twenty second time interval used for all curve comparisons. These timestamps are a subset of the timestamps in column three in Table 8. The timestamps range from approximately -41 seconds to 41 seconds. This interval represents the time period in which all three Lab-Volt runs completed the required behavior of achieving steady state at

thirty percent, rising to sixty percent, and achieving steady state at sixty percent. Thus, all curve comparisons in the experiment compare only the heights recorded from approximately -41 seconds to 41 seconds. Column one shows the associated heights in the Sx1T1 run which are a subset of the heights in column one in Table 8. Column two shows the associated heights in the Sx10T1 run which were calculated with linear interpolation based upon the times for the Sx1T1 run shown in column three of Table 9.

Table 9. Example Part 2.

Sx1T1 Final Height (%)	**Sx10T1 Final Height (%)**	**Final Time (s)**
30.54607964	29.98390198	-40.91967687
30.54607964	29.98390198	-40.32067689
30.54450989	29.98390198	-39.72067699
30.54607964	29.98390198	-39.11967692
30.54607964	29.98390198	-38.51867685
30.54607964	29.98390198	-37.91867694
30.54293823	29.98393825	-37.31767687
30.52095413	29.98519443	-36.71767697
30.52095413	29.98645061	-36.11767682
30.51781273	29.98770678	-35.51767692
30.49739838	29.98896087	-34.91867694
30.49739838	29.99021914	-34.31767687
30.47227478	29.99147532	-33.71767697
30.47227478	29.9927315	-33.11767682
30.47227478	29.99398768	-32.51767692
30.44714928	29.99524177	-31.91867694
30.44714928	29.99649795	-31.3186768
30.44400787	29.99775413	-30.71867689
30.42359352	29.99901031	-30.11867699

Continued on next page

Table 9 – continued from previous page

Sx1T1 Final Height (%)	Sx10T1 Final Height (%)	Final Time (s)
30.42045403	30.00026649	-29.51867685
30.39846802	30.00152266	-28.91867694
30.39846802	30.00278094	-28.31767687
30.37491417	30.00403921	-27.7166768
30.37491417	30.00529748	-27.11567697
30.37491417	30.00655366	-26.51567682
30.34978867	30.00780984	-25.91567692
30.34978867	30.01641696	-25.31667694
30.34978867	30.25806927	-24.71767697
30.32466316	30.49972159	-24.11867699
30.32466316	30.98342971	-22.91967687
30.56178284	31.22548542	-22.31967696
30.92453003	31.46754123	-21.71967682
31.28570557	31.71000034	-21.11867699
31.64531136	31.95205614	-20.51867685
32.09285736	32.19411185	-19.91867694
32.45088959	32.43706621	-19.31967696
32.80735779	32.82588037	-18.71967682
33.22349548	33.21469437	-18.11967692
33.59095001	33.60415651	-17.51867685
34.04948807	33.99361849	-16.91767701
34.41694641	34.38243265	-16.31767687
34.77655029	34.77124665	-15.71767697
35.22723389	35.15941268	-15.11867699
35.52246094	35.54822684	-14.51867685
35.98727798	35.93704084	-13.91867694
36.35002136	36.32522637	-13.31967696
36.71276855	36.71623872	-12.71967682
37.18701172	37.10725091	-12.11967692

Continued on next page

Table 9 – continued from previous page

Sx1T1 Final Height (%)	Sx10T1 Final Height (%)	Final Time (s)
37.5497551	37.4989149	-11.51867685
37.90465164	37.89057874	-10.91767701
38.30351257	38.28093945	-10.3186768
38.65998077	38.67195165	-9.718676889
39.10909271	39.06361564	-9.117676819
39.47341156	39.45462783	-8.517676919
39.83929825	39.84564003	-7.917677009
40.29469299	40.23665459	-7.317676869
40.65901184	40.62784798	-6.717676969
41.07043839	41.01838944	-6.118676989
41.43161392	41.4089309	-5.519677009
41.79436111	41.79947235	-4.920677029
42.24975586	42.19001396	-4.321676819
42.6172142	42.58120736	-3.721676909
42.9862442	42.97240076	-3.121677009
43.33171463	43.36359431	-2.521676869
43.75099182	43.75413577	-1.922676889
44.21738052	44.14525809	-1.322676979
44.57070541	44.53234457	-0.723677009
44.9271698	44.92007739	-0.123676859
45.28049469	45.30781005	0.476323041
45.73589325	45.69554286	1.076323181
46.10020828	46.0839217	1.677323021
46.51320648	46.47100818	2.276322991
46.87752151	46.858741	2.876323141
47.24026871	47.24711983	3.477322971
47.6909523	47.63485264	4.077323111
48.05056	48.02287207	4.677323021
48.41016388	48.41765341	5.276322991

Continued on next page

Table 9 – continued from previous page

Sx1T1 Final Height (%)	Sx10T1 Final Height (%)	Final Time (s)
48.87498474	48.81309393	5.876323141
49.20318222	49.20853426	6.476323001
49.66799927	49.60331561	7.075323001
50.04016876	49.99875617	7.675323201
50.40448761	50.39419653	8.275323101
50.85046005	50.78963688	8.875323001
51.21006775	51.18441837	9.474323201
51.54140472	51.57985873	10.0743231
51.98266983	51.97428937	10.6733231
52.33756256	52.36229973	11.273323
52.80238342	52.7496636	11.8723232
53.14942551	53.13832058	12.473323
53.51060104	53.52633113	13.0733232
53.9597168	53.91498811	13.674323
54.27692413	54.30235185	14.273323
54.72760773	54.69036234	14.8733231
55.09663391	55.0783727	15.473323
55.46880341	55.46573644	16.072323
55.93205261	55.85060994	16.6723231
56.30265045	56.1304317	17.273323
56.66696548	56.40932241	17.8723232
57.07996368	56.68867858	18.4723231
57.42229462	56.96850043	19.0733232
57.78504181	57.24785659	19.6733231
58.25771332	57.52674716	20.272323
58.5984726	57.80610347	20.8723232
58.91254044	58.08592518	21.473323
59.22346497	58.36528149	22.0733232
59.40876389	58.63940696	22.674323

Continued on next page

Table 9 – continued from previous page

Sx1T1 Final Height (%)	Sx10T1 Final Height (%)	Final Time (s)
59.57050705	58.8022077	23.2743231
59.64274216	58.96527972	23.875323
59.72911072	59.12780915	24.4743232
59.79192352	59.29060984	25.0743231
59.83746338	59.45341053	25.674323
59.87201309	59.6159399	26.273323
59.91440964	59.77874064	26.8733231
59.92069244	59.94154133	27.473323
59.94581604	60.1043421	28.0733232
59.94581604	60.25613255	28.6733231
59.9693718	60.24656943	29.272323
59.9693718	60.23697438	29.8733231
59.99449921	60.22741126	30.4723231
59.99292755	60.21781621	31.0733232
59.99449921	60.20822116	31.674323
59.99449921	60.19865804	32.273323
59.99292755	60.18907896	32.8733231
59.99292755	60.17951584	33.4723231
59.99449921	60.16993676	34.072323
60.0196228	60.15749167	34.6723231
60.01805115	60.1018376	35.272323
59.99449921	60.04627628	35.871323
60.0196228	59.99052944	36.4723231
60.0196228	59.93487537	37.072323
60.00234985	59.87931405	37.671323
59.99449921	59.82375271	38.2703232
60.0196228	59.76809864	38.8703231
60.0196228	59.7123518	39.4713232
59.9976387	59.65660498	40.072323

Continued on next page

Table 9 – continued from previous page

Sx1T1 Final Height (%)	Sx10T1 Final Height (%)	Final Time (s)
59.9976387	59.60163511	40.671323

Bibliography

1. Jonathan Butts and Michael Glover. How Industrial Control System Security Training is Falling Short. In *International Conference on Critical Infrastructure Protection IX*, pages 135–149. Springer, 2015.

2. Evan Plumley, Mason Rice, Stephen Dunlap, and John Pecarina. Categorization of Cyber Training Environments for Industrial Control Systems. In *International Conference on Critical Infrastructure Protection XI*, pages 243–271. Springer, 2017.

3. Patricia Armstrong. Bloom's Taxonomy. In *Center For Teaching*. Vanderbilt University. Retrieved 20 November 2017 from *https://cft.vanderbilt.edu/guides-sub-pages/blooms-taxonomy/*.

4. John Moteff and Paul Parfomak. Critical Infrastructure and Key Assets: Definition and Identification. Library of Congress Washington DC Congressional Research Service, 2004. Retrieved 18 January 2017 from *http://www.dtic.mil/docs/citations/ADA454016*.

5. Barack Obama. Presidential Policy Directive 21: Critical Infrastructure Security and Resilience. *Washington, DC*, 2013. Retrieved 18 January 2017 from *https://obamawhitehouse.archives.gov/the-press-office/2013/02/12/presidential-policy-directive-critical-infrastructure-security-and-resil*.

6. Graham Williamson. OT, ICS, SCADA – What's the Difference? *Kuppingercole Analysts*, 2015. Retrieved 18 January 2017 from *https://www.kuppingercole.com/blog/williamson/ot-ics-scada-whats-the-difference*.

7. Keith Stouffer, Joe Falco, and Karen Scarfone. Guide to Industrial Control Systems (ICS) Security. *NIST Special Publication*, 800(82):16–32, 2011. Retrieved 19 January from *http://www.gocs.com.de/pages/fachberichte/archiv/164-sp800_82_r2_draft.pdf*.

8. Galloway, Brendan and Hancke, Gerhard P and others. Introduction to industrial control networks. *IEEE Communications Surveys and Tutorials*, 15(2):860–880, 2013.

9. Kim Zetter. An Unprecedented Look at Stuxnet, the World's First Digital Weapon. *Wired*, 2014. Retrieved 15 July 2017 from *https://www.wired.com/2014/11/countdown-to-zero-day-stuxnet/*.

10. Roberto Saco, Eduardo Pires, and Carlos Godfrid. Real Time Controlled Laboratory Plant for Control Education. In *32nd Annual Frontiers in Education*, volume 1, pages pp. T2D–12–T2D–16. IEEE, 2002.

11. MathWorks Simulink. Retrieved 29 January 2018 from *https://www.mathworks.com/products/simulink.html*.

12. Zach Thornton and Thomas Morris. Enhancing a Virtual SCADA Laboratory Using Simulink. In *International Conference on Critical Infrastructure Protection IX*, pages 119–133. Springer, 2015.

13. Festo Didactic Ltd, Quebec, Canada. *Familiarization with the Training System Pressure, Flow, and Level User Guide 86004-E0*, 2010. Retrieved 31 October 2017 from *https://www.labvolt.com/downloads/datasheet_98-3531-0_en_120V_60Hz.pdf*.

14. Kiam Heong Ang, Gregory Chong, and Yun Li. PID Control System Analysis, Design, and Technology. In *IEEE Transactions on Control Systems Technology*, volume 13, pages 559–576. IEEE, 2005.

15. Andrew Chaves, Mason Rice, Stephen Dunlap, and John Pecarina. Improving the Cyber Resilience of Industrial Control Systems. In *International Journal of Critical Infrastructure Protection*, volume 17, pages 30–48. Elsevier, 2017.

16. Linear Interpolation. Florida A&M University – Florida State University College of Engineering. Retrieved 19 December from *url-http://www.eng.famu.fsu.edu/ dommelen/courses/eml3100/aids/intpol/*.

17. Salvatore S. Mangiafico. Summary and Analysis of Extension Program Evaluation in R: Introduction to Permutation Tests. *R Handbook*, 2016. Retrieved 19 February 2018 from *http://rcompanion.org/handbook/K_01.html*.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704–0188

1. REPORT DATE *(DD–MM–YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From — To)*
22-03-2018	Master's Thesis	Sept 2016 — Mar 2018

4. TITLE AND SUBTITLE

Variable Speed Simulation for Accelerated Industrial Control System Cyber Training

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Bradford, Luke, M., 2d Lt, USAF

5d. PROJECT NUMBER

17G310

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way
WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT-ENG-MS-18-M-014

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Department of Homeland Security ICS-CERT POC: Neil Hershfield, DHS ICS-CERT Technical Lead ATTN: NPPD/CS&C/NCSD/US-CERT Mailstop: 0635, 245 Murray Lane, SW, Bldg 410, Washington, DC 20528 Email: ics-cert@dhs.gov phone: 1-877-776-7585

10. SPONSOR/MONITOR'S ACRONYM(S)

DHS ICS CERT

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

DISTRIBUTION STATEMENT A:
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

It is important for industrial control system operators to receive quality training to defend against cyber attacks. Hands-on training exercises with real-world control systems allow operators to learn various defensive techniques and see the real-world impact of changes made to a control system. Cyber attacks and operator actions can have unforeseen effects that take a significant amount of time to manifest and potentially cause physical harm to the system, making high-fidelity training exercises time-consuming and costly. This thesis presents a method for accelerating training exercises by simulating and predicting the effects of a cyber event on a partially-simulated control system. A hardware-in-the-loop system comprised of a software-modeled water tank and a commercially-available programmable logic controller is used to demonstrate the feasibility of this method. The results demonstrate the system's speedup capability which allows users to accurately simulate the effects of a cyber event at speeds faster than real-time.

15. SUBJECT TERMS

Industrial control systems, cyber training exercises, hardware-in-the-loop simulation

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Dr. Barry E. Mullins (ENG)
U	U	U	U	90	19b. TELEPHONE NUMBER *(include area code)* (937) 255-3636 x7979 Barry.Mullins@afit.edu