2017

A Novel Approach to Determining Real-Time Risk Probabilities in Critical Infrastructure Industrial Control Systems

Michael Elrod
Nova Southeastern University, mike.elrod@opc.com

NSUWorks Citation

Michael Elrod. 2017. *A Novel Approach to Determining Real-Time Risk Probabilities in Critical Infrastructure Industrial Control Systems.* Doctoral dissertation. Nova Southeastern University. Retrieved from NSUWorks, College of Engineering and Computing. (1006) https://nsuworks.nova.edu/gscis_etd/1006.

A Novel Approach to Determining Real-Time Risk Probabilities in Critical Infrastructure Industrial Control Systems

by

Micheal T. Elrod

A dissertation submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in
Information Systems

College of Engineering and Computing
Nova Southeastern University

2017

We hereby certify that this dissertation, submitted by Michael Elrod, conforms to acceptable standards and is fully adequate in scope and quality to fulfill the dissertation requirements for the degree of Doctor of Philosophy.

James D. Cannady, Ph.D. Date
Chairperson of Dissertation Committee

Craig Miller, Ph.D. Date
Dissertation Committee Member

Peixiang Liu, Ph.D. Date
Dissertation Committee Member

Approved:

Yong X. Tao, Ph.D., P.E., FASME Date
Dean, College of Engineering and Computing

College of Engineering and Computing
Nova Southeastern University

2017

An Abstract of a Dissertation Submitted to Nova Southeastern University
in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy

A Novel Approach to Determining Real-Time Risk Probabilities in Critical Infrastructure Industrial Control Systems

by
Micheal T. Elrod
March 2017

Critical Infrastructure Industrial Control Systems are substantially different from their more common and ubiquitous information technology system counterparts. Industrial control systems, such as distributed control systems and supervisory control and data acquisition systems that are used for controlling the power grid, were not originally designed with security in mind. Geographically dispersed distribution, an unfortunate reliance on legacy systems and stringent availability requirements raise significant cybersecurity concerns regarding electric reliability while constricting the feasibility of many security controls. Recent North American Electric Reliability Corporation Critical Infrastructure Protection standards heavily emphasize cybersecurity concerns and specifically require entities to categorize and identify their Bulk Electric System cyber systems; and, have periodic vulnerability assessments performed on those systems. These concerns have produced an increase in the need for more Critical Infrastructure Industrial Control Systems specific cybersecurity research.

Industry stakeholders have embraced the development of a large-scale test environment through the Department of Energy's National Supervisory Control and Data Acquisition Test-bed program; however, few individuals have access to this program. This research developed a physical industrial control system test-bed on a smaller-scale that provided an environment for modeling a simulated critical infrastructure sector performing a set of automated processes for the purpose of exploring solutions and studying concepts related to compromising control systems by way of process-tampering through code exploitation, as well as, the ability to passively and subsequently identify any risks resulting from such an event.

Relative to the specific step being performed within a production cycle, at a moment in time when sensory data samples were captured and analyzed, it was possible to determine the probability of a real-time risk to a mock Critical Infrastructure Industrial Control System by comparing the sample values to those derived from a previously established baseline. This research achieved such a goal by implementing a passive, spatial and task-based segregated sensor network, running in parallel to the active control system process for monitoring and detecting risk, and effectively identified a real-time risk probability within a Critical Infrastructure Industrial Control System Test-bed. The practicality of this research ranges from determining on-demand real-time risk probabilities during an automated process, to employing baseline monitoring techniques for discovering systems, or components thereof, exploited along the supply chain.

Acknowledgements

I would like to thank my dissertation committee, chaired by Dr. James Cannady. Dr. Cannady has remained patient and steadfast throughout the entire dissertation process. His demeanor has been professional and unwavering, as has been my experience with him since beginning the program's course work at Nova Southeastern where he was the first instructor I had. His interest and passion in the area of AI and neural networks has guided my own interest into the area of immunological computation. It is my ambition to further this research into the area of adaptive machine learning, building off of his work and from that of his prior students. I am glad to have had the privilege of Dr. Cannady being a part of my committee. I thank Dr. Liu for his reviews among the many iterations of perfecting a quality product and his participation as a committee member. Much thanks and a tremendous amount of debt and gratitude of appreciation is extended to Dr. Craig Miller, as the National Rural Electric Cooperative Association's Chief Scientist, who, without hesitation, accepted my request to participate on my committee as an external subject matter expert. Dr. Miller recognized this research early on as an important contribution to the overall effort of securing the nation's electric grid, as well as, the additional layer of security it can provide for other critical infrastructure control systems. I want to thank my parents, Don and Sheila, for never doubting my ambitions. Lastly, I express a heartfelt thanks to my significant other, Joyce, and our four children, Alexis, Micheal, Austin, and Isabella, who have provided support and encouragement, along with copious amounts of patience and tolerance, during this long journey. This work would not have been possible without their continuous encouragement and support.

Table of Contents

List of Tables

Tables

List of Figures

Figures

Chapter 1
Introduction

Background

The economy of the Twenty-First Century is fueled in significant part by America's energy infrastructure. Without a stable energy supply, the U.S. economy could be destabilized and the health and welfare of the U.S. would be threatened. A heavy reliance exists on both rail and pipelines for the distribution of fuel products across America, and highlights the interdependencies between multiple critical infrastructure sectors, particularly the energy sector (Department of Homeland Security [DHS] & Department of Energy [DOE], 2010). As a result of this dependency, as well as the daily and unrelenting attacks against the electrical grid, the energy sector began a significant effort for increasing its planning and preparedness against vulnerabilities that confront it (DHS & DOE, 2010).

In 2003, Sandia National Laboratories (SNL) began conducting vulnerability assessments on information technology (IT) systems focusing on automation and control systems used in U.S. Critical Infrastructure (CI). SNL's report identified several reasons for security vulnerabilities in CI (Stamp, Dillinger, Young, & DePoy, 2003). The findings of SNL imposed consequences on CI sectors, particularly the energy industry, as legislation was enacted to define, identify, and secure the most critical Bulk Electric Systems (BES) in North America. The electric utility industry responded with Critical Infrastructure Protection (CIP) standards. (North American Electric Reliability Corporation [NERC], 2013; NERC, 2015).

Results from subsequent vulnerability assessments conducted by SNL during the development of the CIP standards, demonstrated that the potential for disrupting Critical Infrastructure Industrial Control Systems (CI2CS) devices could be caused by vulnerability scanners and should be used with caution on production CI2CS networks (Stouffer, Falco, &

Scarfone, 2013; as cited in Franz, 2003). Identifying vulnerabilities within a CI2CS requires a slightly different approach from that of typical information systems (IS) since they have real time potential considerations. (Stouffer et al., 2013). CI2CS include skid-mounted programmable logic controllers (PLC), supervisory control and data acquisition (SCADA) systems, and distributed control system (DCS) in control system configurations. The mutually dependent and often highly interconnected systems that these control systems operate are vital to the CI of this nation (Stouffer et al., 2013).

Traditional information processing system's characteristics differ from that of CI2CS because logic executing in CI2CS directly affects the physical world. (National Institute of Standards and Technology [NIST] SP800-82, 2015). However, there are some similar characteristics which includes health and safety risks, financial issues, damage to the environment, and security of confidential information. (Stouffer et al., 2013). Originally, CI2CS components were not attached to information technology (IT) systems or connected to networks and were located in physically secured areas, as a result, the security concerns pertained only to local threats. However, because today's CI2CS are less isolated, there is more concern about increased threats from remote, external actors and the necessity for greater system security. Additionally, since wireless networking is used more frequently, CI2CS implementations are exposed to more risk (Stouffer et al., 2013).

Control system threats can come from a number of sources. Security objectives for CI2CS are conventionally prioritized in the order of: availability, integrity, and confidentiality (Stouffer et al., 2013). The possible outcomes that might occur, as a result of a CI2CS incidents, are blocked or delayed flow of information through CI2CS networks, unauthorized changes to instructions, commands, or alarm thresholds, inaccurate information sent to system operators or to cause the operators to initiate inappropriate actions, CI2CS software or configuration settings modified, or CI2CS software infected with malware, or interference with the operation of safety systems, which could endanger human life. All of these are grave concerns and priorities for our nation.

Problem Statement

Because of their unique architecture, sensitive and volatile environments, and real-time physical and critical processes, CI2CS currently lack the capability for effectively monitoring real-time vulnerabilities across their critical cyber assets. Energy delivery control systems can be easily disrupted or broken by the cybersecurity technologies developed for protecting business networks and IT computer systems (Idaho National Laboratory [INL], 2011). The networks and computers that control our Nation's power grid and other critical processes are very different from those on our desks.

Real-time risk monitoring, network penetration testing, and vulnerability assessments conducted on live industrial control system (ICS) environments have resulted in systemic and/or operational failure causing both a safety and reliability concern (Duggan, 2005). Therefore, as a precaution, CI2CS vulnerability assessments are often conducted and performed in a laboratory environment with the resulting states compared to that of a live environment (Stamp et al., 2003). Unfortunately, this method fails to observe the various dynamic states and conditions that some ICS, such as generating facilities and critical manufacturing, continue to operate under due to the constraints and limitations of a simulated ICS environment, including, but not limited to, network noise, number and device types, varying temperatures, and networking schemes. A significant and necessary capability would include a passive real-time method or process for monitoring and alerting to a systemic threat within an operational CI2CS environment by considering the inclusion of stimuli emitted during its operation as part of the risk detection equation.

Dissertation Goal

The primary goal of this research was to develop an approach that takes into account the unique architecture; sensitive and volatile environments; and real-time physical and critical processes of a CI2CS environment in identifying and alerting an operator to a real-time suspected or actual risk across its network of critical cyber assets. Alternatively, such a goal increases the opportunity for discovering and mitigating any vulnerabilities or threats that may otherwise go unnoticed for a prolonged period of time until an off-line assessment can be conducted. It is crucial that threat and vulnerability detection is done in a timely fashion.

Relevance and Significance

The goal in developing SCADA protocols, so that task constraints on the network would be met, was to emphasize features that would ensure good performance (Stouffer, Falco, & Scarfone, 2013). At the time, network security was not much of a focus (Chen et al., 2013). The infection vectors that were initially most common include: exploitation of hardware/software vulnerabilities, unauthorized access to Internet facing devices, malware transfer via removable media, spear-phishing attacks, and probing and scanning of publicly accessible assets. The National Cybersecurity and Communications Integration Center (NCCIC) estimates that many more incidents are occurring than are reported on the basis that cyber incident reporting is done on a voluntary, and not required basis (National Cybersecurity and Communications Integration Center [NCCIC], 2015).

In 2014, the Industrial Control Systems Cyber Emergency Response Team (ICS-CERT) addressed 245 reported incidents of asset owners or related entities (NCCIC, 2015). CI organizations that operate ICS reported that the majority of these incidents were initially detected in their business networks. In each case, ICS-CERT evaluates the incident to determine the presence and extent of the intrusion with a focus on identifying lateral movement into the control environment or ex-filtration of sensitive CI2CS information from the business network (NCCIC, 2015).

There has been a misconception, until recently, regarding the security of SCADA networks. The most common misconception was that attackers could not access these networks because they were electronically isolated from all other networks (Carlson, 2005; Risley, Roberts, & LaDow, 2003). The security goal for industrial plants has typically been in placing more focus on heightening physical security. What used to be an isolated and simple control network has been altered into a member of a complex inter-network by increasing the connectivity between the corporate network and production floor. The demands for increased connectivity continues to grow.

Security concerns relating to these SCADA networks continue to be raised over this increase in network interconnectivity. SCADA networks, as with any network, can have multiple access points using current networking technology. There is also no guarantee that physical isolation equals network security. However, there are some occasions where it is necessary to have the local network connected to the external network, typically through a modem or corporate local area network (LAN). The two most common "external" connectivity scenarios involve either an engineer having to troubleshoot and/or configure a system remotely, or the original equipment manufacturer (OEM) requiring access as part of a service agreement. In either of these two scenarios, the connection could be exploited, thus enabling access to machines inside the factory network by a determined attacker (Cervin, Henriksson, &

Årzén,2003).

Although it was believed that attackers would have difficulty accessing SCADA network information, this is not the case (as cited in Byres & Lowe, 2004), because the proprietary SCADA communication protocol's standards, once used in the automation industry, have now been moved towards open international standards. Ironically, one of the down sides to open standards, however, is that the workings of SCADA networks are readily available to attackers. Attackers are able to get an in-depth knowledge of these standards and use it to their advantage (as cited in Byres & Lowe, 2004).

Commercial-off-the-shelf (COTS) software and hardware used in SCADA networks is another factor that contributes to CI2CS security vulnerabilities. Although costs and the time it takes to design can be reduced with COTS software, the security is compromised. (SNL, 2003). COTS software provides a target that tempts attackers because it is often weak on security. Fail-safe mechanisms that could be disabled, if exploited by an attacker, exist as security vulnerabilities in COTS software, unlike non-COTS devices. Non-COTS devices are usually designed to failsafe since they are intended to operate in safety-critical environments; however, this safety issue, caused by the result of the security issue, is another reason why designers of COTS devices should not just consider security issues, but safety ones too.

SCADA Events

There are a number of incidents that highlight CI2CS type events which have taken place where CI2CS or CI information systems have been exploited. There are also certain environments where control systems are not able to perform control and monitoring functions, thus resulting in some failures not being detected at all (Stouffer et al., 2013).

<u>Air Traffic Communications</u>

March 1997, (Worcester, Massachusetts) – Part of the public switched telephone network was disabled by a teenager using a dial-up modem (Stouffer et al., 2013). The weather service, control tower, carriers that used the airport, the airport fire department, and airport security all lost phone service. Also, a printer that controllers used to monitor flight progress and the tower's main radio transmitter were shut down, as well as, another transmitter that activated runway lights (Stouffer et al., 2013). Phone service to 600 homes and businesses in the nearby town of Rutland were also disabled by the attack (Stouffer et al., 2013; Teen Hacker Faces Federal Charges, 1998).

Maroochy Shire Sewage Spill

Spring 2000 –A former employee of an Australian organization (that developed manufacturing software) was rejected for a job by the local government (Stouffer et al., 2013; Smith, 2001). This employee supposedly became disgruntled, and reportedly, over a two-month period, using a radio transmitter, remotely broke into the controls of a sewage treatment system on as many as 46 occasions (Stouffer et al., 2013). The electronic data for particular sewerage pumping stations was altered causing their operations to malfunction which ultimately led to the release of raw sewage into nearby rivers and parks (Stouffer et al., 2013; Smith, 2001). In all, the release totaled about 264,000 gallons (Stouffer et al., 2013; Smith, 2001).

CSX Train Signaling System

August 2003 - The train signaling systems along the U.S. east coast were shut down by the Sobig computer virus. The signaling, dispatching, and other systems were all shut down by this virus as it infected the CSX Corporation's computer system located at its Jacksonville, Florida headquarters. There were ten Amtrak trains affected the morning of the event. Dark signals caused trains between Florence and Pittsburgh, South Carolina to be halted. There was also more than a two-hour delay of one regional Amtrak train from Richmond, Virginia to Washington D.C. and New York (Stouffer et al., 2013; Hancock, 1999). A delay between four and six hours also affected the long-distance trains (Hancock, 1999).

Davis-Besse

January 2003 - In August 2003, it was confirmed by the Nuclear Regulatory Commission, that a private computer network at the idle Davis-Besse nuclear power plant in Oak Harbor, Ohio was infected by the Microsoft structured query language (SQL) Server worm known as Slammer. This caused the safety monitoring system to be disabled for nearly five hours. It additionally took about six hours before the failed plant's process computer became available again. The Slammer virus propagated so quickly that it was able to affect communications on the control networks of at least five other utilities which resulted in successfully blocking control system traffic (Stouffer et al., 2013; Poulsen, 2003).

Northeast Power Blackout

August 2003 - First Energy's control room operators had inadequate situational awareness of critical operational changes to the electrical grid when their SCADA system prevented them from recognizing an alarm processor failure (Stouffer et al., 2013). Additionally, when the failure of the state estimator at the Midwest Independent System Operator occurred, due to incomplete information on topology changes preventing contingency analysis, it resulted in the prevention of effective reliability oversight (DOE, 2003; Stouffer et al., 2013). Several of the key 345kV transmission lines that came into contact with trees in Northern Ohio tripped. This led to an uncontrolled cascading failure of the grid, a loss of 61,800 megawatts (MW) load, which was caused by cascading overloads of additional 345kV and 138kV lines relating to 265 power plants. (DOE, 2003).

Zotob Worm

August 2005 - For almost an hour, 13 of DaimlerChrysler's U.S. automobile manufacturing plants were knocked offline by a round of Internet worm infections referred to as Zotob (DOE, 2003; Stouffer et al., 2013). During that time, workers were stranded as infected Microsoft Windows systems were patched (DOE, 2003). Plants in Indiana, Michigan, Illinois, Wisconsin, Delaware, and Ohio were knocked offline (DOE, 2003). Although the worm affected some early versions of Windows XP, it primarily affected Windows 2000 systems. The computer's repeated rebooting and shutdown were noted as part of the symptoms. Aircraft-maker Boeing, several large U.S. news organizations, and heavy-equipment maker Caterpillar, Inc. all suffered computer outages because of Zotob and its variation (Lemos, 2005).

Taum Sauk Water Storage Dam Failure

December 2005 – A catastrophic failure of the Taum Sauk Water Storage Dam caused the release of a billion gallons of water. The reservoir having been overtopped or simply filled to capacity may have caused the failure. The overtopped theory remains to be the current version, as it is speculated that the routine nightly pump-back operation failed to cease filling the reservoir which ultimately resulted in the reservoir's berm being overtopped (DOE, 2003; Stouffer et al., 2013). AmerenUE stated that the gauges at the Osage plant at the Lake of the Ozarks, which monitors and operates the Taum Sauk plant remotely, read differently than the gauges at the dam (DOE, 2003; Stouffer et al., 2013)). A network of microwave towers link the stations together. Taum Sauk had no operators on-site (FERC, 2005).

Stuxnet Worm

W32.Stuxnet was first categorized in July of 2010 and targeted ICS in order to take control of industrial facilities, such as power plants. It has been speculated that the most probable intent was industrial espionage. Stuxnet malware contained two strikingly different attack routines. Both attacks were aimed at damaging centrifuge rotors, but used different tactics. The first attack attempted to over-pressurize centrifuges. The second attack tried to over-speed centrifuge rotors and take them through their critical (resonance) speeds (Langner, 2013).

It is important to note that the electric segment's current CIP regulatory standards only require vulnerability assessments every 15 calendar months (NERC, 2013). In a situation, such as Stuxnet, where the attack operated periodically about once a month over the lifetime of the attack, vulnerability risk assessments every 15 calendar months are not sufficient (Langner, 2013).

Heartbleed

Heartbleed was a vulnerability with a proof-of-concept (PoC) exploit code discovered by Google Security and a team of security engineers at Codenomicon. Private secure sockets layer (SSL) keys used in OpenSSL Versions 1.0.1 through 1.0 were exposed due to a flaw in a "transport layer security/datagram transport layer security" (TLS/DTLS). There were twelve electric segment asset owner/operators that observed scanning and/or exploitation activity in the first 14 days after the Heartbleed vulnerability was announced. There have been over 200 source IP addresses known to be performing scanning and/or exploitation since the exploit was publicly announced.

Although not affecting the utility's control system, one utility confirmed their video teleconferencing system was successfully exploited. Initial detection of this activity was performed by network intrusion detection systems. This was escalated by their managed security services provider. An assessment confirmed that the device was vulnerable, and that network traffic suggested information was successfully accessed by the attacker (DHS, 2014).

Barriers and Issues

Test-Bed Design

This research attempted and succeeded in scaling a CI2CS environment, taking into account the dynamic swings from operational and environmental conditions that mimic a specific type of facility's operating environment for the express purpose of achieving better success in determining a more precise representation of the data. Essentially the scaled environment consisted of a minimal number of devices necessary for performing particular tasks and producing an expected and anticipated outcome within an enclosed and defined space. The scaled environment or test-bed, is explained in greater detail in chapter three.

Test-bed designs include both virtual and physical models performing simulations or actual functions, or sometimes a combination of the two, as in this research. An issue with any test-bed design is the ability to reconstruct or create an environment that mimics an actual system so that the results obtained in that environment are valid and reliable enough for representing the same or similar outcome in an actual system. Some physical design issues for consideration include the number of objects, the space occupied by the objects, the properties and attributes of those objects, and the thousands of operational and mechanical processes performed by that system. Since physical models typically exist in a controlled environment, as most often they are actual laboratory environments at universities or governmental facilities, dynamic swings in operational and environmental conditions are not always accounted for or even considered.

The test-bed used in this research was a scaled down model of a chemical processing facility. It included both a production and control system component, as well as the typical network infrastructure found in many CI processing industries. These are commonly referred to in the power generation and manufacturing industry as the corporate, automation, and application networks (relative to the control system vendor). The a*utomation* network is for the communication that occurs between the processes taking place on the production floor and the control system within the facility. The *application* network handles the communication between the control system and application services. Depending on the control system's vendor, it may simply and most commonly be referred to as the control system network. Automation and application network terms are used almost exclusively by Siemens. As in many cases with CI networks (especially the electricity sector), it is not an uncommon security practice to place an air-gap between the corporate and control system network, as will be the situation in this case.

The production facility was a near replica of an industrial type facility fabricated with steel construction, concrete floors and environmental conditioning-as determined by equipment performance. A dehumidifier was ultimately the only environmental conditioner required. This structure, although built to withstand the outdoors and being exposed to the same conditions as any other stand-alone facility could be subjected to, was located in a basement area that maintained an average temperature of 64 degrees Fahrenheit with a relative humidity of 47 percent, as shown in chapter four. Security controls were in place to prevent tampering or other malicious activities that could interfere with the validity of the research results. There were a sufficient number of operational assets, each with its own function, performing a variety of tasks; therefore, producing an ample number of stimuli to monitor while the automated process took place.

The control area, commonly referred to as a control room, was a typical environmentally conditioned and secure room. This was quite similar, if not an exact representation of a control room located at an industrial facility that is more suited for housing information systems, and not the more industrial environment of a shop floor, or mechanical building containing power generation equipment. The test-bed design combines the implementation of both the control room and production facility as a whole, as one cannot function without the other, and the test-bed would not be complete otherwise. This test-bed model, as described and further illustrated in chapter three, could pass industry verification, validation and/or certification and ultimately withstand industry scrutiny even if using an internationally recognized test-bed, such as Idaho National Lab's (INL) DHS Industrial Control Systems Cyber Security Training test-bed, Idaho Falls, Idaho, as a comparison. The INL test-bed was the inspiration behind the basic functionality for the test-bed used in this experiment.

Data Acquisition/Accuracy/Validity

Another potential issue in this research, and still relating to test-bed design, was acquiring an ample amount of data from enough actual objects and controller processes, so that a sufficient number of values could be used for comparative analysis. As the sensory data acquisition processing unit (SDAPU) is a passive system, it can be placed into many CI2CS environments with very low implementation overhead. The SDAPU itself takes up a miniscule amount of physical space. Significant effort went into fitting and tuning the test-bed, as it was one of the more crucial aspects in obtaining accurate and valid data. Because this research was designed using a linear approach in the type of processing cycle it performed, it is important to note that simultaneous, multi-function processing would require a somewhat different approach and method in data collection, analysis and formulation.

This research could potentially be validated against an actual commercial CI2CS, albeit using a smaller data set and not in real- time by evaluating logs, packet captures, and sensor data that would be matched and analyzed, off-line, using date-time stamps from various points in time to determine if object and operational attributes can be used to actually develop a modeling baseline before going live. However, as the overall test-bed results from this research proved to indicate accurate, reliable and conclusive findings there was no benefit to be realized in applying it within an actual commercial environment until a more robust solution using a non-linear method for placement in a closed-loop design is developed.

Live CI2CS Operational Volatility

Another issue is that commonalities among the suggested CI2CS actions do not generate traffic against production systems or on production operational networks; thereby, necessitating that this research be carried out in a test-bed environment. Less intrusive methods are preferred over more active methods for minimizing the risk of causing a failure during testing (Stouffer et al., 2013). In fact, non-real-time less intrusive methods, can gather all, or at least most, of the same information. However, this method would fail to accomplish the goal and achieve the intended purpose of this research (Duggan, 2005), as this research challenges the current static state and non-operational environmental process of determining risk to a CI2CS by introducing a novel and passive approach for identifying real-time risk probabilities in CI2CS.

It is imperative that CI2CS personnel understand the CI2CS being tested and related risks so they are aware and prepared to immediately address any problems that may arise while testing is being performed during a CI2CS assessment; especially in a production environment (Stouffer et al., 2013). Personnel able to perform manual control, if manual control of the system is possible, should be present during the testing (Stouffer et al., 2013). Personnel must be able to immediately address any unintentional stimulus or DoS that may affect the CI2CS (Duggan, 2005). Establishing a passive non-intrusive process for determining risks in a real-time operating environment eliminates the necessity for testing and additional personnel to perform manual controls. However, it does not exclude periodic audits, during facility outages, for verifying and validating the accuracy of real-time processes.

Another factor considered in choosing CI2CS testing methods for this research is that compared to IT systems, CI2CS have little spare capacity and are designed to have much greater longevity than their IT counterparts (Stouffer et al., 2013). CI2CS hardware and software is typically far from being state-of-the-art and can easily be overtaxed (Stouffer et al., 2013). If CI2CS systems are run on legacy networks, they may run at slow speeds. Active testing may generate a lot of traffic which may overwhelm the system. (Stouffer et al., 2013). A passive process used for determining real-time risk occurs as a result of both log and data analysis (i.e. a listening approach with “offline” processing), thereby eliminating any undue burden on the capacity limitations of either storage or bandwidth.

Assumptions, Limitations and Delimitations

Assumptions

As the focus of this research effort was centered around the cyber security domain, it can be assumed that there are potentially inherent risks with any device that has integrated chip(s) or micro controllers/processors, no matter their function or purpose, and could by design, be deliberately and intentionally embedded with malware. It should, therefore, stand to reason that an assumption was made that all of the production and cyber assets used in this research were free of deliberate or intentional manufacturer injected malicious exploits spanning the entire supply chain in order for the study to be effective and useful. To date, activities leading up and to the end of this research have included an investigation into the discovery of any knowledge base (KB) information that would indicate whether or not the assets used in this experiment had vulnerabilities introduced at any point in the supply chain. Throughout the research, no supply chain threat was ever discovered. An objective of this and future research is the ability to recognize peculiar behavior and occurrences from controller based objects along with their performance over time. This research demonstrated the capability for establishing and determining operational baseline normalcy, so that during an event when a device acted abnormally, the anomaly did not go unnoticed.

Limitations

The test-bed used in this experiment was a closely controlled environment explicitly used for the purpose of scaling an actual environment into a more manageable project. As a result, sensor and production/processing equipment quality varied from that of commercial and industrial sensor and production equipment. This limited the placement, purpose, and application of such sensors and equipment due to durability and lifecycle issues. This disparity in quality limits any kind of stress testing desired to help in differentiating major variations in the production environment with that of minor ones.

Spatial parameters limited the size of the overall test-bed. Although the size was sufficient to conduct and perform the research experiment with meaningful results, the size constraint limited the distances at which vibration and sound data could be captured. As it was, the sensors were located within inches and a few feet of the stimuli producing sources. Extending the distances between sources of stimuli and risk detection sensors could help determine sensor sensitivity.

Delimitations

In order to constrain the scope of the study in a way that made it more manageable, the research was conducted in a test-bed; however, there was an express intent of placing a SDAPU in a power block at a generating facility and evaluating its environmental spatial data to that of offline and historical log data generated from within the control system network. As not all power blocks or test-beds are created equal, the test-bed used in this research was designed and developed to model and control variables to the degree necessary for discovering whether or not this experiment would achieve its intended goal of identifying a potential risk introduced within a CI2CS. Because the processes in this research were performed in a linear fashion, the experiment did not garner the necessary qualifications applicable to a non-linear process; therefore, the research was not applied within an actual power block at a generating facility.

The testing schedule for the test-bed had to be carefully planned, so that while testing was taking place, the least number of exterior stimuli were present. Unwarranted external stimulus would undoubtedly affect the quality of the data. This was especially true and most critical during baselining. Anomalies or intentionally created stimuli were anticipated and even expected after baselining had been accomplished.

Definition of Terms

The following terms will be used in this research and are specific to the subject topics to be discussed in the study.

Amplitude—The maximum absolute value reached by a voltage or current waveform (InfoComm, 2006).

Bulk Electric System (BES)—Essentially consists of Transmission Elements operated at 100 kV or higher and Real Power and Reactive Power resources connected at 100 kV or higher. This does not include facilities used in the local distribution of electric energy (specific inclusions and exclusions have been omitted from this definition for brevity) (NERC, 2015).

Connected Components Workbench (CCW) software—a set of collaborative tools supporting Allen Bradley's safety relay, PLCs, drives and component operator interface products for small machines. It is based on Microsoft Visual Studio technology and offers controller programming, device configuration and integration with HMI editor (Rockwell, 2015). *Control System (CS)*—An interconnection of components (computers, sensors, actuators, communication pathways, etc.) connected or related in such a manner to command, direct, or regulate itself or another system, such as chemical process plant equipment/system, oil refinery equipment/systems, electric generation/distribution equipment/systems, water/waste water systems, or manufacturing control systems (MSISAC, 2016).

Data Acquisition—Sampling of the real world to acquire data that can be recorded and/or manipulated by a computer. Sometimes abbreviated DAQ, data acquisition typically involves acquisition of signals and waveforms and processing the signals to obtain desired information (MSISAC, 2016).

DNP3 (Distributed Network Protocol)—DNP3 is a set of communications' protocols used between components in process automation systems. Its main use is in utilities such as electric and water companies. Usage in other industries is not common. It was developed for communications between various types of data acquisition and control equipment. It plays a crucial role in SCADA systems, where it is used by SCADA Master Stations (aka Control Centers), RTUs, and Intelligent Electronic Devices (IEDs). It is primarily used for communications between a master station and RTUs or IEDs. ICCP, the Inter- Control Center Communications Protocol (a part of IEC 60870-6), is used for inter-master station communications (Triangle MicroWorks, 2016).

Data Historian—a centralized database for logging all process information within a CI2CS. Information stored in this database can be accessed to support various analyses, from statistical process control to enterprise level planning (Stouffer, Falco, & Scarfone, 2008).

Distributed Control Systems (DCS)—are used to control industrial processes such as electric power generation, oil refineries, water and wastewater treatment, and chemical, food, and automotive production. DCS are integrated as a control architecture containing a supervisory level of control overseeing multiple, integrated sub-systems that are responsible for controlling the details of a localized process. Product and process control are usually achieved by deploying feedback or feed forward control loops whereby key product and/or process conditions are automatically maintained around a desired set point. To accomplish the desired product and/or process tolerance around a specified set point, specific PLCs are employed in the field and proportional, integral, and/or derivative settings on the PLC are tuned to provide the desired tolerance as well as the rate of self-correction during process upsets. DCS are used extensively in process-based industries (NIST, (2015).

Factory Acceptance Test (FAT)—A test conducted at the Vendor's premise, usually by a third party, to verify operability of a system according to specifications (United States Computer Emergency Readiness Team, 2015).

Fieldbus—A digital, two-way, multi-drop communication link among intelligent measurement and control devices. It serves as a LAN for advanced process control, remote input/output and high speed factory automation applications (TradesInfo, 2015).

HART—is a bi-directional communication protocol that provides data access between intelligent field instruments and host systems. A host can be any software application from a technician's hand-held device or laptop to a plant's process control, asset management, safety or other system using any control platform (Greenfield, 2013).

Heartbeat Signals—A heartbeat can be described as regularly repeated signals generated by hardware, software, or firmware to indicate normal operation or for synchronization with other components within an energy delivery system. Also known as watchdog timer, keep-alive, health status. The signals indicate the communication's health of the system (DOE, 2014).

Human Machine Interface (HMI)—HMI is software or hardware that enables individuals to control a process under review, or to change controls or to effect an override manually if there is an emergency. An operator can use an HMI to configure set points or control algorithms and specifications as needed. Operators can obtain process information and historical data with an HMI. There are great variations in the type of platform or location that then affects the interactions. (SCADA Systems, 2014).

Metasploit—a penetration tool used in the process of identifying security gaps in an IT infrastructure by mimicking an attacker (Metasploit, 2014). It is different from port scanning in that:

- Port scanning identifies active services on hosts,
- Vulnerability management identifies potential vulnerabilities on systems based on the installed software version of the operating system or applications, and
- Penetration testing involves trying to take control over the systems and obtain data (Metasploit, 2014).

Modbus—is a serial communication protocol developed by Modicon in 1979 for use with its PLCs. It is a method used for transmitting information over serial lines between electronic devices (TradesInfo, 2015).

Nessus Vulnerability Scanner—provides patch, configuration, and compliance auditing; mobile, malware, and botnet discovery; and sensitive data identification (Nessus, 2014).

Nmap—a free and open source utility for network discovery and security auditing. It is useful for tasks such as network inventory, managing service upgrade schedules, and monitoring host or service uptime. Nmap uses raw IP packets in novel ways to determine what hosts are available on the network, what services (application name and version) those hosts are offering, what operating systems (and OS versions) they are running, what type of packet filters/firewalls are in use, and dozens of other characteristics (Fyodor, 2013).

Noise—Undesired or irrelevant elements in a visual image; a sound of any kind; an electric disturbance in a communications system that interferes with reception of a signal; a disturbance, especially a random and persistent disturbance that obscures or reduces the clarity of a signal; irrelevant or meaningless data (Copley, 2015).

Nonrepudiation—The sender cannot deny that he/she sent the data in question to ensure that a traceable legal record is kept and has not been changed by a malicious entity (Stouffer, Falco, & Scarfone, 2008).

Offline or Non Run-time—When the control system and network assets are at operational readiness, but the production equipment is not producing or operating (i.e., a generator is off-line, but ready for start-up if scheduled or dispatched).

Online or Run-Time—When the control system and network assets are at operational readiness and the production equipment is producing or operating.

OPC—is the interoperability standard for the secure and reliable exchange of data in the industrial automation space and in other industries. It is platform independent and ensures the seamless flow of information among devices from multiple vendors. Initially, the OPC standard was restricted to the Windows operating system. As such, the acronym OPC was borne from OLE (object linking and embedding) for Process Control. These specifications, which are now known as OPC Classic, have enjoyed widespread adoption across multiple industries, including manufacturing, building automation, oil and gas, renewable energy and utilities (OPC, 2015).

Outage-

Scheduled—Normally a pre-determined maintenance interval when the production system or individual assets, and control system network can be taken offline, and out of a scheduled operational or production state.

Unscheduled—Typically an emergency condition which has occurred and has taken the production system or individual assets, and/or control system network off line, although not scheduled for operation or production state.

Programmable Logic Controller (PLC)—A programmable microprocessor-based device designed to control and monitor various inputs and outputs used to automate industrial processes. PLCs are used in both SCADA and DCS systems as the control components of an overall hierarchical system to provide local management of processes through feedback control as described in the sections above. In the case of SCADA systems, they provide the same functionality as RTUs. When used in DCS, PLCs are implemented as local controllers within a supervisory control scheme. PLCs are also implemented as the primary components in smaller control system configurations. PLCs have a user-programmable memory for storing instructions for the purpose of implementing specific functions such as I/O control, logic, timing, counting, three mode proportional-integral-derivative (PID) control, communication, arithmetic, and data and file processing. The PLC is accessible via a programming interface located on an engineering workstation, and data is stored in a data historian, all connected on a LAN (Stouffer, Falco, & Scarfone, 2008).

Process Control—is an engineering discipline that deals with architectures, mechanisms and algorithms for maintaining the output of a specific process within a desired range. It is extensively used in industry and enables mass production of consistent products from continuously operated processes such as oil refining, paper manufacturing, chemicals, power plants and many others. It enables automation, by which a small staff of operating personnel can operate a complex process from a central control room (Söderqvist, 2014).

Process Field Bus (PROFIBUS)—is a standard for fieldbus communication in automation technology. It is standardized in IEC 61158 – the foundation has therefore been laid for interoperability and compatibility (Felser, 2001).

Radio Telemetry Unit (RTU), also called a *remote telemetry unit*—a special purpose data acquisition and control unit designed to support SCADA remote stations. RTUs are field devices often equipped with wireless radio interfaces to support remote situations where wire-based communications are unavailable. Sometimes PLCs are implemented as field devices to serve as RTUs; in this case, the PLC is often referred to as an RTU (Stouffer, Pillitteri, Lightman, Abrams, & Hahn, 2015).

Supervisory Control and Data Acquisition (SCADA)—systems are used in distribution systems such as water distribution and wastewater collection systems, oil and natural gas pipelines, electrical power grids, and railway transportation systems. They are highly distributed systems used to control geographically dispersed assets, often scattered over thousands of square kilometers, where centralized data acquisition and control are critical to system operation. A SCADA control center performs centralized monitoring and control for field sites over long-distance communications networks, including monitoring alarms and processing status data. Based on information received from remote stations, automated or operator-driven supervisory commands can be pushed to remote station control devices, which are often referred to as field devices. Field devices control local operations such as opening and closing valves and breakers, collecting data from sensor systems, and monitoring the local environment for alarm conditions (Campbell, 2015).

Sensory perception—occurs in organisms capable of performing neurophysiological processing of the stimuli in their environment, and covers the processes commonly called "the senses": hearing, vision, taste, smell and touch (Gene Ontology Consortium, 2016).

Site Acceptance Test (SAT)—A test conducted at the customer location, often by a third party, to verify operability of a system according to specification immediately prior to commissioning (Ameren, 2015).

Acronyms

The following acronyms will be used in this research and are specific to the subject topics to be discussed in the study.

ARP	Address Resolution Protocol
ADC	Analog-to-Digital Conversion
AC	Alternating Current
BES	Bulk Electric System
CCE	Common Configuration Enumeration
CERT	Computer Emergency Response Team
CIP	Critical Infrastructure Protection
CI2CS	Critical Infrastructure Industrial Control System
COTS	Commercial off the Shelf
CPES	Cyber Physical Energy System
CPU	Central Processing Unit
CSIRT	Cyber Security Incident Response Team
CSSP	Control System Security Program
CVE	Common Vulnerabilities and Exposures
DCOM	Distributed Common Object Model
DCS	Distributed Control System
DHS	U.S. Department of Homeland Security

DMZ	Demilitarized Zone
DNP	Distributed Network Protocol
DOD	Department of Defense
DOE	Department of Energy
EMI	Electromagnetic Interference
FERC	Federal Energy Regulatory Commission
FTP	File Transfer Protocol
HMI	Human Machine Interface
ICCP	Inter Control Center Communications Protocol
ICS	Industrial Control Systems
ICS-CERT	Industrial Control Systems Computer Emergency Response Team
IEEE	Institute of Electrical and Electronics Engineers
IDS	Intrusion Detection System
IO	Input/output
IP	Internet Protocol
IPS	Intrusion Prevention System
ISO	International Standards Organization
IS	Information Systems
IT	Information Technology
KB	Knowledge Base
LAN	Local Area Network
MitM	Man-in-the-Middle
NERC	North American Electrical Reliability Corporation

NIPP	National Infrastructure Protection Plan
NIST	National Institute of Standards and Technology
NSA	National Security Agency
OEM	Original Equipment Manufacturer
OLE	Object Linking and Embedding
OPSEC	Operational Security
OPC	OLE for Process Control
OSRAD	Operating and Sensory Risk Analysis Detection
OT	Operation Technology
PC	Personal Computer
PLC	Programmable Logic Controller
RPC	Remote Procedure Call
RMF	Risk Management Framework
RTR	Real-Time-Risk
RTU	Remote Telemetry Unit
SCADA	Supervisory Control and Data Acquisition
SEIG	Self-Excited Inductive Generator
SIEM	Security Incident Event Management
SDAPU	Sensory Data Acquisition Processing Unit
SQL	Structured Query Language
SSM	Spatial Sensory Module
TIA	Technology Industry Association
TCP	Transmission Control Protocol
US-CERT	U.S. Computer Emergency Readiness Team

Summary

Our nation’s CI2CS are constantly under attack by a number of rogue nation states determined to exploit one of our most vulnerable areas when it comes to the defense and security of the United States. In the not so recent past, these cyber vulnerabilities were largely contributed to legacy systems, some designed twenty years ago, or more, that were built with little to no security in mind. Now, more regularly, those legacy systems are being replaced by control systems that incorporate the most current COTS software integrated with the most current COTS networking and system hardware. This marrying of technology for the sake of streamlining protocols and standards throughout certain industries that are reliant and dependent upon automation; although, expedient and efficient, poses its own set of challenges and security risks. To this end, government has reacted to such a degree, that they have intervened and imposed extensive and burdensome regulatory compliance standards, particularly for that of a specific industry; that being the electricity sector. This includes the implementation of ten cyber security standards, with an additional one (regarding the supply chain) on the way, addressing several areas but not limited to: categorization of assets; change control; and, incident response and restoration.

Compounding the issue of regulated cyber security controls and policies are the requirements that vulnerability assessments on control system networks are performed periodically to ensure that vulnerabilities are discovered and mitigated to the extent possible, but not necessarily in a “timely” manner. Because CI2CS are more volatile than their IT system counterparts, vulnerability scans are not routinely conducted. Passive network devices such as IDS, IPS, file integrity monitoring and other such tools may be placed throughout a network, and user dependent change management integrity systems are beginning to emerge, such as, Plant Automation System's (PAS), Integrity, for example as a way of detecting potentially malicious activity; however so, a relatively closed network where IDS signatures for example are not and cannot be regularly updated, without knowing what an anomaly in a closed network may even look like, remain to offer little protection if devices or systems within the network have been internally compromised either through a malicious actor or exploited controller/processor that has made its way through the supply chain.

The aim of this research was to develop a method of passively monitoring a control system network, and randomly comparing its activity and that of the production or processing environment's actual spatial state to that of previously defined operational baselines of that same network and production environment during various operational states for identifying and determining whether or not potential risks have been introduced into the system.

Chapter 2

Review of the Literature

Context

General

This review walks through the literature beginning with a general and broad description of issues having direct implications to a number of security problems currently facing CI2CS and the causes leading up to them. Much of this relates to the intermingling of proprietary methods, devices, and protocols from proven legacy systems, with that of traditional IT systems and applications of today that are constantly plagued with security issues. The review continues by organizing not just the sections relating to specific CI2CS security issues and test-bed development, but to other areas, such as governance and standards, security, risk, and modeling. These areas will address CI2CS: risk assessments, standards and research development, sensory and environmental factors, and some probability methods which will be applied in determining risks.

The private sector owns a majority of the energy infrastructure. The energy sector supplies the transportation industry, provides electricity and energy vital to the nation (DHS & DOE, 2010). The (DHS) divides energy infrastructure into three interrelated segments consisting of petroleum, natural gas, and electricity. The electricity segment of the energy infrastructure contains in excess of 6,413 power plants (DHS & DOE, 2010). DHS estimates that 22 percent of electricity is produced by combusting natural gas, another 20 percent by nuclear power plants, and that the bulk of electricity production, 48 percent, is by combusting coal. The remaining ten percent of generation is provided by oil, renewable (wind, geothermal, and solar), hydroelectric plants and other sources. A heavy reliance exists on both rail and pipelines for the distribution of fuel products across America which highlights the interdependencies between multiple critical infrastructure sectors (DHS & DOE, 2010).

This interdependency between the transportation and energy system's sector further illustrates how all other sectors are in some way dependent upon the energy sector. As a result of this dependency, as well as the attacks against the electrical grid, the energy sector began an effort to increase its planning and preparedness against the vulnerabilities (DHS & DOE, 2010). It has been a benefit to industry, as a result of the cooperation among the various energy sector groups, to participate in substantial information sharing, especially relating to that of best practices. Although many sector owners and operators have had extensive infrastructure protection experience abroad, their recent attention has been in placing more emphasis and focus on domestic cybersecurity issues (DHS & DOE, 2010).

Power plants generate and transmit electricity over 203,930 miles of transmission lines, and electricity is subsequently distributed to millions of customers (DHS & DOE, 2010). DCS or supervisory control and data acquisition (SCADA) systems are used by regional grid operators and utilities to keep the electricity infrastructure system in balance (DHS & DOE, 2010). These are highly automated and sophisticated energy management systems.

SNL began conducting vulnerability assessments on IT systems in 2003, focusing on automation and control systems used in CI. Their report concluded that most security vulnerabilities in CI included failures to: sufficiently define the security sensitivity for data relating to automation systems; grant authenticated users the proper access control privileges to services and data based on operational requirements; identify and protect a security perimeter; and, establish comprehensive security through defense-in-depth (Stamp, Dillinger, Young, &

DePoy, 2003). It was determined that security vulnerabilities were due to deficiencies in security, budget pressures, loss of employees and administration concerns (Stamp et al., 2003). Also, during this time the industry was not focused on security and in large part unaware of the adversaries' capabilities and threat environment. Ultimately, much of the security deficiencies were due to the lack of adequate security education and training of the complex modern information technology equipment being used in control system automation (Stamp et al., 2003).

The findings of SNL imposed consequences on CI sectors resulted in legislation to define, identify, and secure the most critical BES. This legislation was enforced by the Federal Energy Regulatory Commission (FERC), as industry stakeholders were tasked through the North American Electric Reliability Corporation (NERC) to develop a set of comprehensive standards that would identify, assess, and correct those deficiencies identified by SNL's report.

At that time, the electric utility industry responded with the following nine CIP standards: Recovery Plans for Critical Cyber Assets, Critical Cyber Asset Identification, Security Management Controls, Incident Reporting and Response Planning; Personnel and Training, Physical Security of Cyber Assets, Systems Security Management, Electronic Security Perimeter, and Sabotage Reporting. While one standard, Sabotage Reporting, has been removed from the initial set of standards, two additional standards have been added. These are Protection of BES Cyber Information, and Incident Recovery. These eight revised CIP standards along with the two new standards, CIP-002 thru CIP-011, although originally slated for an April 2016 effective date, became enforceable July 2016 instead (North American Electric Reliability Corporation [NERC], 2013; NERC, 2015).

Results from subsequent vulnerability assessments conducted by SNL during the development of the CIP standards, and later included in the NIST Special Publication (SP) 800-

82 (2015), *Guide to Industrial Control Systems (ICS) Security*, demonstrated that the potential for disrupting CI2CS devices could be caused by vulnerability scanners and should be used with caution on production CI2CS networks (Stouffer, Falco, & Scarfone, 2013; as cited in Franz, 2003). A major concern cited was an accidental denial of service (DoS) to devices and networks. While attempting to verify vulnerabilities, the actual scanning process creates additional traffic that is otherwise not present on the network. This traffic is a result of the representative set of attacks and extensive probing conducted against those connected devices. CI2CS are built and designed to automate and control equipment or real-world processes. An interruption to this process and the delivery of incorrect instructions to the system may cause it to malfunction and perform improperly. This could result in damage to the equipment, a loss of product, injuries, or worse; death (Stamp et al., 2003).

Identifying vulnerabilities within a CI2CS requires a slightly different approach from that of typical information systems (IS) (Stouffer et al., 2013). Generally, devices of an IS can be replaced, restored, or just simply rebooted, leaving the end user with only a slight interruption of service. Since a CI2CS determines a physical process it has effects which can be extremely time sensitive. (Duggan, 2005).

CI2CS include skid-mounted PLC, SCADA systems, and DCS in control system configurations. Control systems are found in the industrial control sectors. SCADA systems can be used to distribute assets while maintaining central control. (Bobat, Gezgin, and Aslan, 2015). SCADA has been evolving with the changes in technology since they first started being used in the 1960s. SCADA systems have evolved from mainframe-based to client/server systems that use central communication protocols to send data from peripheral units to a master unit. (NCS, 2004). With the evolution of SCADA protocols, the once closed proprietary systems have now become open systems. Although this may be a benefit to designers, allowing them to select equipment that can assist them in monitoring their unique system using assets from a variety of vendors, it has likewise opened up a number of attack vectors (NCS, 2004).

Generally, DCS's are used within a local area, such as a factory using supervisory and regulatory control, to control production systems (Stouffer et al., 2013). PLCs, on the other hand, typically provide regulatory control and are used for discrete control for specific applications (Stouffer et al., 2013). Other types of devices used with CI2CS, such as field devices, include: remote telemetry units (RTU), used interchangeably with radio telemetry units; alternating current (AC) drives; solenoid valves, photo eyes, motors, lights and various sensors.

The mutually dependent and often highly interconnected systems that these control systems operate are vital to the CI of this nation (Stouffer et al., 2013). The majority (90 percent) of America's CI2CS mentioned above are owned and operated by private industry. There is a small percentage operated by federal agencies, such as materials handling and air traffic control (Stouffer et al., 2013).

CI2CS started out as isolated systems that ran proprietary control protocols using specialized software and hardware; unlike that of traditional IT systems with their publicly available protocols, and general software and hardware. Table 1 shows the earlier differences between IT and ICS. Because of the low-cost and widely available IP devices now replacing proprietary solutions, the potential and possibility for cybersecurity incidents and vulnerabilities have increased (Stouffer, Lightman, Pillitteri, Abrams, & Hahn, 2015). CI2CS are starting more and more to resemble IT systems, particularly as they begin adopting IT solutions to promote corporate remote access capabilities and business systems connectivity (Stouffer, Falco, &

Scarfone, 2013). CI2CS are now being developed with COTS computers, operating systems (OS), and network protocols (Stefanini et al., 2005).

Table 1. Summary of IT vs IC System Differences (NIST SP800-82, 2015)

Category	Information Technology System	Industrial Control System
Performance Requirements	Non-real-time Response must be consistent High throughput is demanded High delay and jitter may be acceptable Less critical emergency interaction Tightly restricted access control can be implemented to the degree necessary for security	Real-time Response is time-critical Modest throughput is acceptable High delay and/or jitter is not acceptable Response to human and other emergency interaction is critical Access to ICS should be strictly controlled, but should not hamper or interfere with human-machine interaction
Availability (Reliability) Requirements	Responses such as rebooting are acceptable Availability deficiencies can often be tolerated, depending on the system's operational requirements	Responses such as rebooting may not be acceptable because of process availability requirements Availability requirements may necessitate redundant systems Outages must be planned and scheduled days/weeks in advance High availability requires exhaustive pre-deployment testing
Risk Management Requirements	Control physical world Data confidentiality and integrity is paramount Fault tolerance is less important – momentary downtime is not a major risk Major risk impact is delay of business operations	Manage data Human safety is paramount, followed by protection of the process Fault tolerance is essential, even momentary downtime may not be acceptable Major risk impacts are regulatory non-compliance, environmental impacts, loss of life, equipment, or production
System Operation	Systems are designed for use with typical operating systems Upgrades are straightforward with the availability of automated deployment tools	Differing and possibly proprietary operating systems, often without security capabilities built in Software changes must be carefully made, usually by software vendors, because of the specialized control algorithms and perhaps modified hardware and software involved
Resource	Systems are specified with	Systems are designed to support the

Constraints	enough resources to support the addition of third-party applications such as security solutions	intended industrial process and may not have enough memory and computing resources to support the addition of security capabilities

Today’s CI2CS are significantly less isolated from the outside world than their legacy CI2CS were in the 1960’s; in large part, due to the new IT capabilities that are supported by CI2CS integration. This creates a greater need for securing these systems; however, there must be special precautions taken when introducing security solutions, especially those that were designed to deal with conventional IT systems, but are now being used for securing CI2CS. New security solutions are necessary in some cases, and others must be tailored to accommodate a particular CI2CS environment (DHS & DOE, 2010).

Traditional information processing system’s characteristics differ from that of CI2CS; although, there are some similar characteristics as shown in Table 1 (NIST SP800-82, 2015). The differences are because logic executing in CI2CS directly effects the environment. The characteristics are health and safety risks, damage to the environment, financial issues, and breaches of confidential information (Stouffer et al., 2013). General IT personnel may consider the use of some CI2CS applications and operating systems unconventional with their unique reliability and performance requirement. Sometimes control systems design and operation are conflicted with the goals of efficiency and safety (Stouffer et al., 2013).

Originally, CI2CS components were not attached to IT systems or connected to networks and were located in physically secured areas, as a result, the security concerns generally pertained only to local threats. However, because today’s CI2CS are significantly less isolated from the outside world than their legacy CI2CS were in the sixties, there is naturally more concern about increased threats from remote, external actors and the necessity for greater system security. Additionally, since wireless networking is used more frequently CI2CS implementations are exposed to more risk from equipment that is in close location even though it may not have direct physical access (Stouffer et al., 2013). Control system threats can come from a number of sources, such as disgruntled staff, natural disasters, terrorist organizations, complexities, accidental or malicious actions, malicious intruders, hostile governments, and accidents. Security objectives for CI2CS are conventionally prioritized in the order of: availability, integrity, and confidentiality (Stouffer et al., 2013).

Duggan (2005) lists some of the possible outcomes that might occur, as a result of a CI2CS incident as:

- Blocked or delayed flow of information through CI2CS networks, which could disrupt CI2CS operation.
- Unauthorized changes to instructions, commands, or alarm thresholds, which could damage, disable, or shut down equipment, create environmental impacts, and/or endanger human life.
- Inaccurate information sent to system operators, either to disguise unauthorized changes, or to cause the operators to initiate inappropriate actions, which could have various negative effects.
- CI2CS software or configuration settings modified, or CI2CS software infected with malware, which could have various negative effects.
- Interference with the operation of safety systems, which could endanger human life.

There are some CI2CS across America that are still reliant upon legacy components. Although they may be more prone to physical threats, they are less susceptible to cyber threats due to their proprietary nature. However, since the advent of IT systems and their overwhelming popularity and ubiquity in almost everything we do, they have also integrated their way into CI2CS.

The lack of change management and the application of patch management are arguably the most significant contributing factors to many of today's CI2CS security problems. Together, they have largely led to an increase in attack vectors, not only within the system itself, but also within their networks. Newer CI2CS are not solely built upon typical IT systems, and the fact that they are used quite differently, makes protecting these systems and networks somewhat more challenging, even so, Igure, Laughter, and Williams (2006) argue that CI2CS must be regularly maintained with the latest firmware, updates, and patches, as it applies to either its hardware and/or software extended across the entire network. However, Stouffer, Falco, and Scarfone (2013) go into great detail explaining the proper procedures for applying patches to CI2CS and the consequences that could result if those procedures are not followed. They unequivocally state that the "latest patches", as stated in Igure et al. (2006), cannot be applied without going through a proper vetting process. The proper vetting process would require that patches be tested off-line on a comparable CI2CS. Stouffer et al. (2013) demonstrate that other software can be adversely affected by patching. Patching can be used to eliminate a vulnerability. However, the use of patching can also result in production or safety risks. (Stouffer et al., 2013). It may additionally cause the functionality of the control application to be lost by altering the way it performs with the application or OS the patch was applied to (Stouffer et al., 2013).

Mainly citing NIST Special Publication (SP) 800-53, r.2, Security and Privacy Controls for Federal Information Systems and Organizations, Igure et al. (2006) simply draw attention to the fact that resources for securing the more recent generation of CI2CS do exist. NIST SP 800-

53 contains a set of comprehensive security management practices. In NIST (2013) many IT system security management practices have been developed and are routinely used across a number of industries and academia. Lastly, Igure et al. (2006) failed to consider that many CI2CS systems continue to use older operating systems. At some point, vendors stop supporting these OS, so patches for them either do not exist or may no longer be applicable (Stouffer et al., 2013). Since Igure et al. (2006) work, NIST SP 800-53, r4 (2015) has been published. NIST (2015) revision 4, unlike previous revisions, devotes much of the document to cross referencing IT system's security and privacy controls to that of their corresponding CI2CS where applicable and as appropriate. Not all IT system security and privacy controls correspond with CI2CS functions or capabilities.

NERC (2016) recognized and adopted Stouffer et al. (2013) advice, when it drafted CIP Standard CIP-007-6 and included a requirement in regards to "vetting" patches before they are permanently applied to a CI2CS.   NERC (2016) CIP-007-6 outlines a process for monitoring, analyzing and installing patches as appropriate for Cyber Assets. It includes identifying source(s) that a "Responsible Entity" tracks and determining whether a patching source exists. (NERC, 2016).

NERC (2015) takes into account that CI2CS patch management, for the purpose of enforcement, must be implemented according to Stouffer, Falco, & Scarfone (2013) and NIST (2014) using a systematic, accountable, and documented process for managing exposures to vulnerabilities. However, the enforcement process introduces an entirely different set of challenges that often requires the necessity of a third-party auditor to evaluate an organization's CI2CS infrastructure, up to and including network diagrams, physical layout, IP addresses, open ports and services, etc. This often creates a dilemma on a number of levels, as it exposes what NERC (2015) standard, CIP-011-2, refers to as BES Cyber System Information.

BES Cyber System Information is information about the system that poses a security risk because it can be used to access the system. The information on its own may not necessarily pose a risk (i.e. device names, ESP names). Procedures and information relating to BES Cyber Systems is usually confidential. It can be used to allow access or distribution that is not approved. NERC (2015)

Many sectors classified by the DHS as CI rely heavily on individual third-parties to provide their IT support (Igure, Laughter, and Williams, 2005). This is especially common for CI2CS that still have proprietary operational characteristics or strictly OEM contractual agreements (due to warranties), even if much of the system is comprised of other typical IT components. This arrangement for third-party CI2CS support creates somewhat of a paradox from a compliance perspective in that it cannot directly vet those that are part of the process, other than through non-disclosure agreements and contractual promises that these individuals are in compliance with NERC standard CIP-004-6 requirements pertaining to personnel risk assessments. This dilemma and argument can extend all the way back through the supply chain.

An entirely new effort is underway by FERC that directs NERC to develop and include supply chain standards to their already vast set of cyber and physical security compliance documents. The NIST (2013), DHS (2009), and Energy Sector Control Systems Working Group (ESCSWG), (2014) are addressing supply chain concerns, focusing on information and communications technology (ICT) issues. Although NIST continues to develop a national standard regarding the supply chain, the ESCSWG remains committed to developing their own reference pertaining to the energy sector. The ESCSWG was formed to take over the DHS (2009) product that first addressed supply chain concerns. ESCSWG members include a variety of expertise from the energy industry making it more conducive for developing a product the energy sector can benefit from. It stands to reason that NIST (2013) and ESCSWG (2014) could add some narrative to any future NERC supply chain standard to include establishing its foundation.

Other aspects contributing to CI2CS security shortfalls point directly to failures in an organization’s configuration management program. Most often the case is that there is no program. However, the change management process is an issue for a CI2CS network with numerous “distributed limited-functionality nodes” (Igure, Laughter, and Williams, 2005). Ironically, the problems that exist with the configuration management of many regular corporate networks are neither dissimilar nor irregular from those of CI2CS NERC’s (2016) new CIP-010-2 standard, which went into effect July 1st, 2016, requires a “configuration” change management program for utility entities registered with NERC, but does not go so far as to require a comprehensive change management program.

This new change management requirement found in NERC’s CIP-010-2 standard, will make configuration change management a regular maintenance process. Configuration changes alone are not necessarily effective if physical hardware modifications are made. This is especially troubling considering FERC’s (2016) latest interest concerning supply chain issues previously mentioned. Igure, Laughter, and Williams (2005) also assert that attacks against typical IT systems and networks are as, if not more, prevalent against CI2CS and networks. As a result, these systems should be constantly monitored for signs of intrusions and vulnerabilities.

Theory and Research Literature Specific to the Topic

Overview and Historical Perspective

As cited in Ralston Graham, and Patel (2006) and DHS (2013) we are reminded of how important the protection of America's CI is, and without its very existence the tremendous detriment it would cause to its citizens. There is no doubt it is essential to their physical and economic security. The National Commission on Terrorist Attacks upon the United States Commission Report (2004) made it apparent that America was very complacent with its then current CI schema. It illustrated the highly-interconnectedness and dependencies that existed between various industries and the reliance they had on each other. It also showed how little there was in a way of protection.

The protection of our nation's infrastructure is ultimately managed by the DHS. Although it may seem a daunting task, there are a variety of other agencies and groups that participate in this effort. Some of those agencies and groups include the: DHS National Cyber Security Division (NCSD) Control Systems Security Program (CSSP), US-CERT (Computer Emergency Readiness Team), and CERT® Coordination Center (CERT/CC). The national comprehensive initiative is led by the CSSP. Its goal is to identify, analyze and reduce the cyber risks associated with CI2CS. The US-CERT was established in 2003 to protect the nation's Internet infrastructure by coordinating responses to and defenses against cyber-attacks (as cited in Ralston Graham, & Patel, 2006). US-CERT is responsible for publishing documents that assist in improving control system security and determining vulnerabilities (as cited in Ralston et al., 2006; Nash, 2005; Nelson, 2005). Over 250 organizations related to cyber security response worldwide use the name "CERT". In 1988, CERT/CC was established at Carnegie Mellon

University. They work jointly with DHS to protect the nation's information infrastructure, and similar to US-CERT, contribute expertise in coordinating responses to and defenses against cyber-attacks (as cited in Ralston et al., 2006).

In order to meet a number of the Commission Report (2004) goals, the National Infrastructure Protection Plan (NIPP v.2, Jan. 2006) had to accomplish several objectives which included maximizing the use of resources, assessing risk and implementing risk reduction programs, and establishing and building security partnerships to foster the implementation of CI protection programs. As a way to unify the national cyber security structure and provide the best effort at protecting it, risk assessments for all CI2CS and other cyber systems became an integral part of fulfilling the purpose for the NIPP v.2 (2006) document. Academia, industry, and government are working together on issues addressing infrastructure security in a cooperative effort to provide those with a need-to-know with information that is necessary, and about events or situations that are occurring (NIPP v.2, Jan. 2006; as cited in Ralston et al., 2006).

Nicholson, Webber, Dyer, Patel, and Janicke (2012) focus on the three components paramount to CI2CS security. They include a) administration, b) platform security mechanisms, and c) architecture. The successful attacks against ICS worldwide, both real and simulated were the focus of this work. Specific examples were cited in illustrating some identified then patched vulnerabilities from real world systems.

A literature review and classification of the international journal articles, standards, and reports between the period from 1999 to 2010, provided Yusta, Correa, and Lacal-Arantegui (2011) the necessary tools, methodologies and applications for conducting studies in CI protection concepts based on the selection of their applicability and best-practice methodologies

(Yusta, Correa, & Lacal-Arantegui, 2011). A critical analysis of the specific considerations on electric infrastructures and methodological approaches just mentioned was also presented. Essential concepts around international strategies on infrastructure protection plans and energy security were included in the review.

Test-bed—System Design, Modeling, and Prototypes

Deveza and Martins' (2009) research suggests that the practical test of a control and automation process, controlled by PLCs, can be problematic. They recommend an approach using PLC Control and MATLAB/Simulink simulations along with the implementation of several solutions such as: batteries, LED's and switches, SCADA systems, a HMI, and simulation tools or scale models. However, the utilization of scale models performing real processes is difficult to adapt to different processes and very expensive (Deveza & Martins, 2009). They strongly support using simulations for teaching PLC controlled processes. They illustrate how doing so allows students to test their projects in an almost real environment; although, they also point out that the cost often prohibits its use (Deveza & Martins, 2009). They submit that the use of switch sets and LED's are uninteresting and extremely confusing, and that the student's motivation is severely reduced by this approach because they are only valid when small processes are considered (Deveza & Martins, 2009). Because of current microcontroller technology, PLCs that would once only work with a specific type of simulation tool will now work with any type (Deveza & Martins, 2009). SIMTSX, PSIM, and PC-SIM are also available, as other types of commercial PLC simulation tools; however, they are often not suitable to be integrated with other simulation tools (Deveza & Martins, 2009).

Kabilan and Manohar (2013) examine a power system scenario that is exposed to outages resulting from component malfunction and overdraw of power as a result of high demand. Farmers using pumps for supplying water, as a way of protecting the crops, contributed to this demand. The crops are threatened by a below par monsoon and extreme heat. Chaos, brought on by blackouts, erupts everywhere. The blackout causes essential services like traffic lights, metros, and trains to be halted (Kabilan & Manohar, 2013). The researchers are challenged by the policy of the country and the growing complexity of the power grid in this scenario due to the disruptions on the efficiency, availability and reliability of the power delivery system because it shows a high degree of uncertainty in accurately determining its overall impact (Kabilan & Manohar, 2013).

Uncertainty causes decision makers to hesitate before committing to managing the grid using smart systems. Kabilan and Manohar (2013) observe that the human element involved in near real time decision making is still limited, despite the aid of analysis and simulation tools to help with that process. Large amounts of data are used in representing the status of the grid at any given time. Considerable research remains to be done before fully automated control of the electrical grid is passed over to software agents, but the human limitation continues to be a factor in driving that research.

Kabilan and Manohar's (2013) blackout scenario above, and the paragraph following it, sets the stage for understanding how multi-agent systems can be applied in a power system. They recognize the tremendous challenge in trying to fully automate the grid with multi-agent systems. A significant amount of experimentation and research will be required to develop agents capable of functioning on par with human experts based on the variety of scenarios that can occur within the smart grid (Kabilan & Manohar, 2013). A number of security issues would be presented as a reliance on autonomous agents became more prevalent (Kabilan & Manohar, 2013). Communications and decisions of an agent could be manipulated by an attacker hacking in and then controlling an agent to perform malicious behavior.

As cited in Ralston et al., 2006; Schneider, Lima, Scherer, Camargo, and Franchi (2012) chose to simulate and emulate systems of a micro hydropower plant for performing a feasibility study in what they proposed as a new field of research. The research was in response to the restriction in the production of fossil fuels causing increased energy costs. Access to real-time data was available through a SCADA system. SCADA was connected to the emulated power plant over a Modbus network. Modbus has a faster speed and is more robust than OPC servers used in other tests, so it was naturally the preferred communication method at points where larger data streams were required (Schneider, Lima, Scherer, Camargo, & Franchi, 2012)

The primary machine used in the Schneider et al. (2012) model for process emulation consisted of a self-excited induction generator (SEIG) coupled to an electric motor driven by a frequency converter. An asynchronous electric induction machine, usually used in an electric motor application, performed as an electric generator. This type of unit was used for their project, not only because of its relative low cost and maintenance, but mainly for its reported robustness. (Ofualagba & Ubeku, 2011; Schneider et al., 2012).

Patel (1999); Ofualagba and Ubeku (2011) explain that except for the fact that a SEIG has capacitors connected across its stator, an electric motor or generator’s stationary portion, terminals for excitation, it performs in a manner similar to an electric machine operating by electrostatic induction in the “saturation region”. SEIGs have become an ideal choice where the grid’s reactive power is unavailable for generating electricity, such as stand-alone hydroelectric dams (Patel, 1999; Ofualagba & Ubeku, 2011). Reactive power is typically delineated for alternating current (AC) electrical systems" (Sauer, 2003). Without an adequate remanent magnetic field, excitation will not occur through the external capacitor. The load, the capacitance value in farads, and speed affect generator output voltage and frequency in the self-excited mode (Patel, 1999; Ofualagba & Ubeku, 2011).

Schneider et al. (2012); as cited in Ralston et al. (2006) also proposed using a SCADA system to collect data of the emulated dynamic response from the dynamic turbine simulation. The SCADA system made it possible for some parameters of the simulated hydropower to be changed which made analyzing a number of situations the system responded to possible. For the management of distributed generation, SCADA makes a great tool (Schneider et al., 2012). Without using a real turbine, Schneider et al. (2012) proposed system shows to be an important resource, particularly for testing the topology for new voltage control. Integrating the SCADA system with the hydropower plant simulation/emulation interface allowed monitoring in real-time with an added benefit of a pleasant interface. An analysis of the system's performance was allowed using the responses stored by the SCADA database (Schneider et al., 2012).

Schneider et al. (2012) research helps in the development of new technologies by eliminating the need for constructing a primary machine prototype. This results in lower costs and allows testing prior to implementing actual equipment. Lastly, it is easier to modify a system incorporating the supervisory control capability by exposing the parameters in a simple way.

Atlagic, Sagi, Milinkov, Bogovac, and Culaja (2011) also explore a model-based approach, as they develop and compare various SCADA applications. In explaining the need for modeling, they describe industrial plants as being more complex than their IT counterpart, both in the way they are operated, real-time by an IT-based control system, and their size. Their reference to size does not just pertain to the physical facilities where production and operations take place, but also to the geographical size where supervisory and control functions and capabilities exist. As it relates to their research, a control system is described as a SCADA system that performs supervisory control along with an underlying field installation of IO modules/controllers, control devices transmitters, cabinets, wiring, etc.

As a whole, both software and hardware design essentially make up SCADA application development. This includes the verification of a set of control and guided user interface (GUI) procedures (software design) and a field solution (hardware design) (Atlagic et al., 2011). In order to organize input/output (IO) communication to the field devices, a designed field solution is needed that produces a specification of protocols, IO signals and other parameters (Atlagic et al., 2011). The specification Atlagic et al. (2011) used for building an IO model for an industrial plant, and development of control application and a configuration model, is actually a good starting point.

Contemporary SCADA solutions have to provide a specialized tool set for designing and developing a system in order to meet time and budget constraints, user requirements, and generally coping with its overall complexities; which includes the verification of the control application with an emphasis on design and coding. The object of the SCADA program's final model will be to represent, in real-time, a particular type of industry plant; including information from each of the tools (Atlagic et al., 2011).

Part of the development process in Atlagic et al. (2011) work was to derive an application specific simulation tool, suitable for testing and debugging a control application before its final installation in a factory, from a process/configuration model. Likewise, the primary challenge confronting Atlagic et al. (2011) centered around designing such a general purpose SCADA model with those capabilities. It also raised a key question as to how SCADA solutions could be implemented more easily in an actual, yet complex, industrial plant using a chain of development tools. They develop and present a general purpose SCADA methodology for the express purpose of fulfilling the various requirements of an actual industrial application, to the extent it includes the most complex batch control (Atlagic et al., 2011). A well-defined framework, provided by a set of specialized tools, is used for the development and design of a control application (Atlagic et al., 2011).

The main concept used in Atlagic et al. (2011) was based on a previously designed system called GAUS. GAUS already included the long-term experience in design and implementation of a SCADA system (Atlagic et al., 2011). This SCADA solution had already, at least partially, implemented most of the issues relating to models already mentioned. GAUS was proven to be superior in memory consumption and data processing efficiency compared to other SCADA systems they accessed. This research led to the need for developing a new solution (Atlagic et al., 2011). The idea for their solution was to add important new features, especially for future applications, that would handle the most demanding processes while still preserving the level of efficiency GAUS demonstrated (Atlagic et al., 2011).

Chabukswar, Sino'poli, Karsai, Giani, Neema and Davis (2010) describe a typical SCADA system consisting of a network, actuators, and sensors, for providing the communication between the RTUs and the SCADA master, where the SCADA master provides overall control and monitoring of the system. The life spans of SCADA systems are designed to be quite long. They are usually measured in decades. Chabukswar et al. (2010) reiterate that unlike today, security was not part of the systems design for many of the SCADA systems still in use today. These legacy systems were subjected to physical threats. Today's systems are connected to the internet where they are both monitored and controlled. As a result of this internet connectivity, network security problems remain the primary threat of remotely connected systems.

Industry is well aware of these security risks, as they are quite evident; however, there is no simple task of merely "upgrading" SCADA systems. Upgrades are cumbersome for a number of reasons. The primary reason being that there is often an undesirable lengthy down time associated with the addition of security features, especially to traffic control systems and power plants. Secondly, new security protocols would have to be added to completely replace the embedded codes of existing SCADA devices. Lastly, SCADA system networks cannot typically be generalized because they are usually customized for that particular system. In order to make dependable and reliable security features that address security vulnerabilities for both legacy and in the design of future SCADA systems, their implementation must be assessed and rectified in realistic settings (Chabukswar et al., 2010).

In Chabukswar et al. (2010) a familiar security attack was simulated on a SCADA system and reasonable effects of the attacks were observed in the functioning of the system. This system was composed of models in different simulation environments and domains, implemented as a proof-of-concept at a chemical plant. Command and Control Wind Tunnel (C2WT) was used for facilitating the data transfer and interaction between the environments. C2WT is a visual space for designing and bringing into effect and action diverse C2 simulation states. It facilitates the expeditious growth of ‘integration models’, which it then uses during the entire cycle of the simulated space. (Hemingway, Neema, Nine, Sztipanovits & Karsai, 2011).

There has recently been a significant increase in the exploration and development of renewable energy resources. Because of their abundance and availability, solar power and wind systems are the more popular ones. However, these systems are not without implementation caveats such as sufficient wind speed and space. This location dependency must account for an ample amount of space for solar plants absorbing the sun’s energy and wind plant’s receiving not just wind flow, but wind speed. They are usually located separately and installed in rural areas. As such, SCADA systems are required to remotely control and monitor them. In another study consisting of a wind-PV-battery renewable energy system, Soetedjo, Lomi, Nakhoda and Tosadu (2013) proposed a SCADA system for monitoring the real-time electrical data using a remote controller and measurement devices by employing a PLC and digital power meters. The intranet was used for sending data to the monitoring center from remote devices (Soetedjo et al., 2013). The SCADA system was used for failure detection and monitoring the wind turbine. An observation of anomaly data taken from several measurements was used in detecting wind turbine failure (Soetedjo et al., 2013).

Morris, Srivastava, Reaves, Pavurapu, Abdelwahed, Vaughn, McGrew and Dandass (2010) introduces a cyber-physical energy system (CPES) that combines “computing, communication and control capabilities” and the physical world. In order for CPES to function properly, it must occur in real-time and provide a dependable, safe, secure, and efficient integration (Morris et al., 2010). Every physical component in CPES generally has its cyber capability embedded in it. Multiple scales are used in networking CPES components. Self-assembly, adaptation, learning, self-organization, and higher performance are integrated from both the cyber and physical components. An incomplete knowledge of the system operating state, time-varying utilization, physical degradation and malfunction, and hardware and software component failures are multiple factors that cause a dynamic and uncertain environment that these complex systems typically operate in (Morris et al., 2010).

The work of Morris et al. (2010) provides CPES research with: a set of challenges; a road map of research required to secure these systems; and lastly, a current survey of research results within this domain. There continues to be a significant challenge in securing CPES. As with many researchers in this field, Morris et al. (2010) also recognize that at the time when many process control systems were designed, there was not a security requirement in mind nor such a networked world used for controlling CPES. Recently, corporate networks having internet connectivity are connected to these same process control systems (Morris et al., 2010).

The security by the obscurity concept has traditionally been the basis for the protection of SCADA systems. (Giani, Karsai, Roosta, Shah, Sinopoli, & Wiley, 2008). Insufficient knowledge due to proprietary protocols once prevented an attacker from breaking into the system. Policies, recommendations, standards, and suggestions for possible countermeasures are mainly what much of today's protection relies upon (DOE, 2005; NERC, 2013). The development of a SCADA system test-bed is essential to gaining a better understanding of how SCADA systems can be protected. Security controls must be developed to protect SCADA systems from attacks. (Chabukswar et al., 2010).   Giani et al. (2008) describe the reference architecture, detailed implementation, attack scenarios and SCADA security test-bed status of that research.

Chen, Peng, and Wang (2013) note that there are many cybersecurity researchers on the CI2CS side of research, but that only a few of them are focused on the control equipment itself. More important is the realization of how few conclusions are made public from that research. Chen et al. (2013) present a design and development of a process control test-bed for control system security studies using several different PLCs and a DCS. Because their research focused on the cyber layer, and the advantages associated with simulated process, the simulated Tennessee-Eastman process was adopted in that study.

Because of the differences between IC systems and IT systems used in business applications, it is difficult to use IT cybersecurity techniques available directly for tackling problems within CI2CS. For example, CI2CS may work continuously for several years without incident, yet any modification or updates to the on-line control system may cause unexpected damages to the process and equipment. Countermeasures to mitigate the control system risk should be tested carefully before they are put into use (Chen, Peng, & Wang, 2013). Therefore, it is important to develop a test-bed, instead of using a live industrial processing control system, for the study of cybersecurity research.

The cybersecurity test-bed serves as a platform for validating research cybersecurity solutions serving both government and industry. It is used to identify common cybersecurity deficiencies in need of solutions development through the development of vulnerability taxonomies. It is also used in the identification of existing ICS vulnerabilities (DOE, 2008; as cited in Morris, Srivastava & Reaves, 2011). Test beds provide a low cost means for modeling the effects of cybersecurity attacks on ICSs (DOE, 2008). The Idaho National Labs (INL) National SCADA Test bed, supported by DOE, is a large-scale test-bed dedicated to control system cybersecurity training, assessment, outreach, and standards improvement (DOE, 2008).

A process control system test-bed known as the Industrial Instrumentation Process Laboratory is housed at the British Columbia Institute of Technology (BCIT). A power boiler, fully operational distillation column, evaporator, a chemical blending reaction process, and a batch pulp digester are included in the BCIT lab (DOE, 2008). There is also a variety of SCADA equipment in the lab that includes field devices, such as valves and measurement instrumentations, GE/Fanuc Series 90/70 and 90/30 PLCs with Genius I/O, Emerson Delta V and PROVOX DCS, Bailey Net90 DCS, Honeywell TDC 3000 DCS, Rockwell PLC-5s, Foxboro I/A DCS, Schneider 984 and Quantum PLCs, and F&P MC5000 controllers (BCIT, 2005).

CI2CS Governance and Standard Bodies

As cited in Ralston et al. (2006), although they are independent organizations, Presidential Directive 63 created the Information Sharing and Analysis Centers (ISAC's), with the assumption that important information, between industry sectors and the government relating to anomalies, threats, vulnerabilities and intrusions could be shared with one another. Information sharing is only as productive as the information that is shared. Some companies are hesitant to "share" information with government agencies. Likewise, government information often does little to benefit industry, especially relating to classified information.

The Center for SCADA Security was created by SNL where SCADA training, research, standards development, and red teams (hacking) take place. A functioning synergistic cyber and wireless test-bed, and power grid were created in a test-bed setting by the SNL in conjunction with the Idaho National Engineering and Environmental Laboratory (INEEL) (INL, 2011; as cited in Ralston et al., 2006). It is referred to as the National SCADA Test Bed. There they develop control system cyber security standards for industry. Other types of supporting work and pure research are performed at the National SCADA Test-bed Program. Reports that have emerged are Carlson (2002) and Carlson, Dagle, Shamsuddin, and Evans (2005) which summarizes some of the activities taking place there. Another report was the release of Kilman and Stamp (2005) SCADA Security Policy Framework (as cited in Ralston et al., 2006).

Singer and Weiss (2005) list a number of standard's bodies and industry groups that perform work addressing control system security needs. There are also additional activities being researched and taking place at the national labs. These include, but are not limited to: NERC, NIST, AGA (American Gas Association), CIGRE (International Council on Large Electric Systems), IEC (International Engineering Consortium), Chemical Sector Cyber Security Program organized by the Chemical Information Technology Council (ChemITC), and ISA (Instrumentation, Systems, and Automation Society) (as cited in Ralston et al., 2006). Documents relating to cyber security and risk assessment have been published by all of them (as cited in Ralston et al., 2006). AGA has prepared reports on communications encryption to various systems during different processes. (AGA 12, Part 1, 2006, final document, and ongoing work parts 2, 3 and 4). ISA published two technical reports addressing security technologies and their application to control systems (ANSI/ISA-TR99.00.01-2004, ANSI/ISA-TR99.00.02-2004) (as cited in Ralston et al., 2006).

Cyber security standards (CIP-002-6 – CIP-011-2, 2015) have been finalized by NERC (2015). These standards establish the requirements pertaining to recovery plans, security management programs, personnel, electronic and physical protection, and incident reporting for certain entities within the electric sector. However, since the time of their approval by FERC, the NIST has defined a baseline set of cohesive, cross industry common security requirements for existing and new control systems for various industries through its Process Control Security

Requirements Forum (PCSRF) (Melton, Fletcher, & Earley, 2004; Stouffer et al, 2004; Falco, Stouffer, Wavering, & Proctor, 2004).

FERC, DOE, and DHS all recognize and acknowledge NERC, as the energy sector coordinator. In early 2000, NERC developed a set of defined security requirements (NERC 1200), the "Urgent Action Cyber Security Standard". It specifically related to the electrical industry as a temporary standard for reducing risks, resulting from a compromise, of any critical cyber assets impacting the reliability of the BES. It was adopted for a one-year period beginning August 2003. That standard matured into more permanent Cybersecurity Standards, CIP-002-3 through CIP-009-3, that have been in place since 2007. The most recent version of CIP, version 6, consisting of ten standards, went into effect July 1, 2016 (NERC, 2015).

API 1164 is a SCADA security standard for operators of oil and gas liquid pipeline systems. It provides guidance for managing SCADA system security and integrity. However, the standard is not limited to pipelines and can be used for developing standards for SCADA systems as part of a best practice program. API 1164 does not cover refineries to date, and currently only applies to pipeline operators. Refineries continue to use cybersecurity guidelines, considered adequate by API, released prior to that time. Access control and cybersecurity are the primary emphasis of this standard, in fact, physical security is not addressed at all. API 1164 provides operators with descriptions of practices used by industry for securing SCADA systems along with a framework for developing sound security practices throughout the organization (American Petroleum Institute [API], 2005).

Similar to the NERC CIP standards, API 1164 includes such areas as: field devices configuration and local access, access control, network design, management systems, physical issues (including business continuity plans and disaster recovery), information distribution classification, data interchange between enterprise and third-party support/customers operating systems, and communication security (including encryption) (Radvanovsky & McDougall, 2009; API, 2005).

The Institute of Electrical and Electronics Engineers (IEEE) developed the Power Engineering Society/Substations 1402-2000 guide that discusses and identifies issues surrounding security related to human intrusion at electric power supply substations (IEEE, 2008).

In 1999 the International Electrotechnical Commission (IEC) developed and circulated report 62210, "Data and Communications Security," throughout the IEC. It included stakeholder identification, security definitions, threats and prioritization of threats, a "Common Criteria" protection profile, policy, attacks, and consequence analysis (DOE, 2003). The security of the TC 57 protocols was assessed using the consequence analysis security methodology. The IEC combined "Common Criteria" with consequence analysis in the development of a security specification that establishes a cryptographic communication channel, containing an example protection profile, between a master station and a substation.

As in many cases, the Transmission Control Protocol/Internet Protocol (TCP/IP) forms the basis for many TC 57 communication profiles. The industry continues to investigate a security solution common to that protocol. There is a consideration to use Transport Layer Security (TLS) as a way of securing IEC TC57 protocols and their derivatives. To ensure interoperability certificate revocation, roles in renegotiation of keys, certificate field, and certificate size issues will be standardized (International Electrotechnical Commission [IEC], 2013).

Spoofing, message replay, modification of packets, and to some extent denial of service are all threats addressed in the development of the IEC 62351-5 (2013) standard (DOE, 2003). Its purpose is to secure IEC 60870-5 and its derivatives. Distributed network protocol three (DNP3) is the derivative in this case. Therefore, supporting non-secure systems, conducting communication authentication at the application layer only, using default pre-shared keys when necessary, allowing bi-directional authentication and following a challenge/response model are all IEC 62351-5 primary design principles (IEC, 2013).

IEC 61850 (2013) standard entitled: “Communication networks and systems in Substations,” currently applies to substations; however, there are discussions within the IEC that 61850 should expand its scope and include the power system from the switchgear up to the master station in serving their control and communication needs. A completed set of IEC 61850 standards will essentially form one integrated system comprising substation equipment. This system includes HV (high voltage)-switchgear, control systems, protective relays, and instrument transformers. The communication network will be used for receiving trip commands from the protective relays, and measurements from the instrument transformers.

In IEC 61850, all substation objects that communicate with each other are defined by the object-oriented data model. The respective object’s data and attributes are contained in logical nodes. Each physical device’s properties and function allocation are contained in a device model within the node (IEC TC 57, 2003).

Security enhancements are required to implement and use IEC 61850 communication profiles in non-secure environments. IEC 62351-6 is packaging five IEC 61850 security profiles: Sample Measured Values (SMV), GOOSE (analogue and digital multicast primarily for protective relaying), GSSE Management, and Client/Server (using TLS and MMS) (DOE, 2003). "IEC 62351-6 references IEC 62351-5" (DOE, 2003; IEC TC 57, 2003).

The Process Control Systems Forum (PCSF) was created and funded by the Homeland Security/Homeland Security Advanced Research Projects Agency (DHS/HSARPA) (as cited in Ralston et al., 2006). For the interim, it is being managed by a private/public Governing Board since being established in February 2005. It was created with a focus on infrastructure control systems in response to the growing number of vulnerabilities in the increasingly automated, interdependent, and computerized operating environment (as cited in Ralston et al., 2006). The design, development, and deployment of more secure legacy and control systems, crucial for securing CI, require an accelerated implementation plan; this is the mission of the PCSF (Ravindranath, 2006). The PCSF interfaces with other organizations through working groups including international groups (DOE, 2003). There is still guidance needed on the actual analysis of the risk assessment that all of these groups' risk analysis and assessment reports and guidelines highlight the need for, because they are not always specific.

The Process Control Securities Requirements Forum (PCSRF) has formally stated the security requirements associated with CI2CS within a System Protection Profile (SPP). The capabilities and security issues relevant to the national critical information infrastructure are discussed within those requirements. ICSs comprised of electronic programmable components were defined as products where security capabilities would exist (National Information Assurance Partnership [NIAP], 2002).

A consortium of nonprofit, government, and academic organizations founded the Institute for Information Infrastructure Protection (I3P) in 2001 through the DHS (Trellue, 2005; as cited in Ralston et al., 2006). Their purpose was to coordinate fundamental development efforts and research in information infrastructure protection. The I3P funded a research endeavor "Unifying Stakeholders and Security Programs" for addressing infrastructure interdependencies and SCADA vulnerabilities (Trellue, 2005; as cited in Ralston et al., 2006). Developing a risk assessment tool and methodology for supporting the development of inherently secure SCADA and PCS systems continues to be the main task of the I3P (Kertzner, Bodeau, Nitschke, Watters, Young, & Stoddard, 2006). Another report relating to cyber security documents, Stoddard, Bodeau, Carlson, Glantz, Haimes, Lian, Santos, & Shaw (2005), included an extensive bibliography with a risk analysis overview. It also explained how some of the existing security metrics' tools identified applied to PCS (DOE, 2003).

Over the past few years there has been a significant increase in the number of SCADA and PCS articles and guides published for providing assistance to users and vendors of those type of systems (as cited in Ralston et al., 2006). The DOE, working with the President's Critical Infrastructure Protection Board, has made available DOE (2005) white paper, *21 Steps to Improve Cyber Security of SCADA Networks* (as cited in Ralston et al., 2006). The United Kingdom's National Infrastructure Security Coordination Centre (NISCC) has a similar guide entitled *Good Practice Guide on Firewall Deployment for SCADA and Process Control Networks* (2005) and has other security documents related to SCADA available (Chandia, Gonzalez, Kilpatrick, Papa & Shenoi, 2007; as cited in Ralston et al., 2006). There are also many PCS and SCADA white papers and guidance documents available from the Chemical Industry Data Exchange.

Risk Assessment Methods and Modelling

Miller and Byres (2005) notes that the relative risk of particular control system implementations has yet to be articulated in many of the papers on the topic of control system vulnerabilities. The procedures, policies, or technology used to protect cannot be determined until the vulnerabilities that can become threats and the resources needing protection are identified (as cited in Ralston et al., 2006).

RiskWatch is a commercial system that performs vulnerability assessments and quantitative or qualitative risk analyses provided by an automated tool. The tool includes proven risk analysis analytic techniques, predefined risk analysis templates, comprehensive knowledge databases, data linking functions, and user friendly interfaces (RiskWatch white paper, 2002).

OCTAVE (Operationally Critical Threat, Asset, and Vulnerability Evaluation) was developed at Carnegie Mellon University's CERT Coordination Center. Threats to critical assets are modeled using event/fault tree analyses. It also provides a framework for identifying and managing information security risks that are used for setting a security strategy, identifying the risks to critical assets, and defining the current state of security. Its greatest value is gained in the fact that it only requires a small team to lead the assessment, although it relies on the knowledge of many employees to complete it (Alberts, Dorofee, Stevens, & Woody, 2003).

There has been much work published relating to the assessment of risk, as such, an effort to categorize it has been somewhat challenging and difficult because in many instances the research is defined by several different aspects. The major aspect considered in categorizing this research depends on which area (when assessing risk) and how much of that particular process is addressed.

In a manner that allows users the ability to select the most appropriate risk assessment method(s), the level of detail and approach were determined to be two of the more important attributes chosen by Campbell and Stamp (2004) to be used as a way of classifying them. Their focus was on the overall availability of risk assessment tools. Risk assessment involves identification, analysis, evaluation and ranking of risk, and its management and treatment. (as cited in Ralston et al., 2006).

Qualitative risk assessment approaches have been described in some of the industry publications and government guidelines mentioned previously (as cited in Ralston et al., 2006). Georgia Institute of Technology (2003) researchers present a very systematic, but qualitative approach to assessing the general risk of information systems. A procedure describing the benefits of countermeasures and threats resulting in computing losses, and the problem of risk management is organized and presented using a three-axis view of the threat space (Georgia Institute of Technology, 2003).

CI have behavioral and operational characteristics that SNL have been able to provide valuable insight into by directly addressing these interdependencies through the development of several simulation and modeling approaches. As cited in Ralston et al. (2006), they created the following categories of detailed simulations and model interdependencies:

- Aggregate supply and demand tools which evaluate the total demand for an infrastructure service and the ability to provide it,
- Dynamic simulations to examine infrastructure operations, disruption effects, and downstream consequences,
- Agent based models which model physical components and their interactions and operational characteristics,

- Physics based models that analyze aspects of infrastructure with standard engineering techniques,

- Population mobility models primarily for transportation and social network study, and

- Leontif Input-Output models which provide an aggregated, time-independent analysis of generation, flow, and consumption of commodities among infrastructure sectors (Rinaldi, 2004).

An integral part to infrastructure risk analyses includes such simulation modeling and abilities.

Haimes (1981, 1998) declares hierarchical holographic modeling (HHM) to be the most comprehensive risk identification methodology in such cases. The diverse characteristics and attributes of a system are represented in this method. All conceivable sources of risk to any infrastructure that uses SCADA, as well as the SCADA systems themselves, can be identified by using this method. HHM is the ideal application for SCADA systems and their associated interconnected and interdependent infrastructures because risks in the total system, through the evaluation of subsystem risks and their corresponding contributions can be easily facilitated through its use (Ezell, 1998). The railroad sector identified sources of SCADA system risk using this method (Chittester and Haimes, 2004).

The risk filtering, ranking, and management method (RFRM), as described by Haimes, Kaplan, and Lambert (2002), identifies risks by building off HHM. RFRM takes HHM a number of steps further in risk prioritization by applying filters and assigning rankings. RFRM begins by using HHM to identify risk in the initial step of its eight-phase process. Once those risks are identified, they proceed through phases involving a variety of risk filtering scenarios where quantitative ranking is determined, before ultimately finishing with the management and feedback phase.

Many coupled CI are rendered to be at great risk from cyber-attacks due to their interdependencies where, most often, SCADA systems are used to remotely manage and control them. Although HHM can be used for identifying risk sources, inoperability input-output modeling (IIM) is needed to quantify the efficacy of risk management. Accounting for both the intra and interconnectedness with each infrastructure is enabled by using Leontief's-based IIM. An attack triggers the initial perturbation providing input to the system which returns the risks of inoperability results as outputs. The percentage of dysfunctionality and economic inoperability, measured in dollars lost, represent two different output metrics. This method is used by Haimes and Chittester (2005) for quantifying economic losses and their propagation through the various economic sectors where SCADA systems are used for controlling large scale civil infrastructures over IP communication networks. A cyber intrusion into the telecommunication sector and the perturbation that resulted is presented in a case study by Haimes and Chittester (2005). Crowther and Haimes (2005) describes IIM in more detail and provides additional case studies.

The methods of IIM, RFRM, and HHM were applied, by Crowther, Dicdican, Leung, Lian, Haimes, Lambert, Horowitz and Santos (2004), to Virginia's Interdependent transportation system as a way of assessing and managing the risk of terrorism. Interdependent sectors can be adversely affected through propagation of a failure in the transportation infrastructure. Crowther et al. (2004) assessed the consequences of those failures by applying a methodology and using a computer tool they developed.

Chance processes are studied in modern probability theory. This is when predictions for future experiments are influenced by the outcomes from knowledge gained by previous experiments. It is postulated that an event's collective past outcomes could have an influence in predicting the next experiment through an observation of a sequence of chance experiments. A.

A. Markov began studying an important new type of chance process in 1907 (Jurafsky & James, 2006). In this process, which is called a Markov chain, the outcome of the next experiment could be affected by the outcome of the given experiment. (Jurafsky & James, 2006).

The characterization of an infrastructure's performance and condition requires developing metrics that can be used in furthering the research of interdependent systems. Nozick, Turnquist, Jones, Davis & Lawton (2005) applied Markov and semi-Markov processes to network links as way of reflecting their uncertain capacity. An analysis of both steady-state and transient concerns regarding service availability is allowed using this Markov-based approach. They used a small-scale SCADA system for demonstrating this approach. Although difficult to obtain, empirical data is needed for parameter estimates in many model structures. It is imperative that these estimates are good since valid outcomes are dependent upon it.

As a way of observing real world events and determining the likelihood that those events may re-occur it is useful to utilize a Markov chain. Unfortunately, some of the events we have an interest in cannot be observed directly; they are hidden. As such, a Hidden Markov Model (HMM) considers events that can be viewed as well as hidden events all in the context of a probabilistic model (Jurafsky & James, 2006).

Pak (2011) argues that risk assessments conducted on IS today may not be valid tomorrow due to the dynamics of the computing environment. Pak (2011) further states that although manual risk assessment methods are good for evaluating threats and vulnerabilities, they are not adequate for operational networks. In order to respond to today's changing threat environment, risk assessments must be proactive, timely, and be able to predict future risk factors (Pak and Cannady, 2009). To cope with the dynamics of today's ISs, a risk assessment must be able to continuously calculate risk in a dynamically changing environment (Pak and Cannady, 2010). Pak (2011) used HMM's to develop a near real-time risk assessment methodology. Future risk levels of organizational assets were predicted by applying the HMM successfully; thereby, providing a prompt and current threat environment risk assessment in a near real-time manner. Using this method, organizational stakeholders were able to see near real-time mission-critical asset risk levels and plan countermeasures, mitigate the vulnerabilities, and justify their options based on a cost-benefit analysis (Houmb, Franqueira & Engum, 2009).

PRA is a method used to analyze risks relating to all aspects of a technological entity. It reviews the entire process, including potential design, building and operation, and end of use. (Stamatelatos, 2000). Unlike HMM, a probabilistic risk assessment (PRA) does not provide a guidance method; although the risk identification phase is technically included, it assumes the risk can be identified by the designer. PRA includes methods other than use logic diagrams and directed graphs; there is also event tree analysis (ETA), all fault/attack (FTA) tree analyses, cause/consequence analysis (CCA), and failure mode effect and criticality analysis (FMECA) or failure mode and effect analysis (FMEA) (Henley and Kumamoto, 1996). A combination or extension of these methods make up most of the other methods. Varying degrees of the methods mentioned earlier are incorporated into many of the tools (as cited in Ralston, et al., 2006).

The characterization of risk can be described as the likelihood of an action occurring resulting in an adverse consequence, as measured by its magnitude (or severity), and the given adverse consequence. Consequences are expressed numerically in PRAs, and frequencies or probabilities express their likelihood to occur. Answering just three questions is generally acceptable for determining risk: Can something go wrong - what? Is it likely – how? Are there consequences – what are they? (Kaplan and Barrick, 1981). PRA answers these by: initiating events or using a set of developed scenarios for discovering what can go wrong; the probability that any of the scenarios occur; and, lastly their consequential estimates (as cited in Ralston, et al., 2006). The PRA is helpful in making informed decisions through the development and presentation of a set of scenarios, frequencies, and associated consequences (Kaplan and Barrick, 1981). PRA evaluates “risk metrics”, which refers to the probability of an event (Stamatelalos, 2002) (as cited in Ralston, et al., 2006). Applying this technique is the most difficult task, as it requires determining needed probabilities of basic events.

What is needed as a way of reducing risks is a quantitative way for determining an attack’s particular probability impact and countermeasure outcome based on the determined risk reduction. As cited in Ralston, Graham, & Patel (2006), it is important that after applying modifications there is the ability for determining whether risk reduction is achieved or not. There are many published works containing simple risk reduction calculations (Tolbert, 2005). Tolbert (2005) used three factors: frequency product; occurrence likelihood; and severity (based on an arbitrarily selected scale of one to five) for calculating a risk metric. It is applied before and after a system is modified.

Ralston et al., 2006 analyzes the stages of a possible attack in conjunction with the attacker’s skill level (as cited in Ralston et al., 2006). McQueen, Boyer, Flynn and Beitel (2006) offers remedial actions for a set of control systems and a calculation for estimating risk reduction to a SCADA system.

Calculating risk reduction applicable to SCADA security can also be performed using PRA. The effect of an overall attack on the highest event probability can be computed by setting a specific threat probability of a lower event to zero simply by adding a security enhancement.

Other work by Graham, Patel and Ralston (2006) combines vulnerability tree and attack tree methods making use of augmented vulnerability trees. They have also developed a tool for quantifying SCADA systems risk that employs two indices for modeling risk.

Cyber & Physical Security/Threats

Attacks on SCADA and DCS are more prevalent than ever before due to the internet's widespread use, the attack vectors available, and the vulnerabilities they face. Researchers and plant managers have been alerted by industry, security experts, system vendors, and the government who have all recognized the vast amount of information available through the many discussions, reports, and papers. Because emphasis in the electricity sector has largely been on performance and reliability, and not security, since historically many of these systems stand alone and are isolated, the move to accept these SCADA and PCS vulnerabilities have been slow by many in the industrial community. Novak (2005) points out that typical network threats now make these systems vulnerable because of their connections to company networks and the internet, and that these threats are exacerbated by the business and economic processes that are tightly integrated with SCADA systems.

There has been a tremendous increase in security awareness for PCS and SCADA systems with an emphasis on the growing problem of being able to recognize threats, then using the vast amount of information available, learning how to find solutions to preventing them (Alper, 2005; Miller, 2005; Singer & Weiss, 2005; Byres & Lowe, 2005). Much of this security technology information is introduced and explained in detail. This information covers areas such as: hardening operating systems (Geer, 2006); system hardware hardening and network architecture (Creery and Byres, 2005); vulnerability testing and assessment (Strickles, Ozog &

Mohindra, 2003; Byres & Franz, 2006); and security monitoring of networks, encryption, and intrusion detection (Peterson, 2004; as cited in Ralston et al., 2006). It is pointed out in Geer's article that network access to systems where the proper functioning of the control application is required, could ultimately be closed as a result of hardening the operating system. It is further noted that employees will most likely circumvent security in situations where control systems become difficult to use as a result of the improper implementation of the controls used to secure them. Geer (2006) goes on to warn users about attempting to adopt an ideal security approach, all the while leaving a gap in search of the perfect solution. Instead, Geer (2016) advises that users should take interim steps to use what is effective and available now.

There is the potential an attacker may carry out a range of attacks against the network should they gain unauthorized access to the SCADA network. Literature from Carlson (2002); Risley et al. (2003), and Stouffer, Falco, & Scarfone (2013) describe many of these possible attacks. Some appliances used for defending against unauthorized network access, which work well in conjunction with one another, include IDSs and firewalls. Stamp et al. (2003) notes that it is many IDS are not able to review SCADA information for questionable activity; this is a similar problem with firewalls. However, the development problem with firewalls for SCADA networks are not as complicated as the development problems of IDS solutions. Simply by knowing a SCADA's protocol structure, firewalls can be developed. However, knowledge of the SCADA protocol's vulnerabilities are required for developing IDS rules to recognize an attack. A considerable number of vulnerability assessments of SCADA protocols were conducted to acquire this knowledge (Igure, Laughter, and Williams, 2005).

There has been a push by the federal government, particularly from the DOE and DHS, encouraging ICS vendors to have their products include built-in security (Carlson, 2005). There is also a need for investing more in securing private sector networks by CI owners. Many of these vendors are addressing this concern. One example is the Experion Process Knowledge System R300 by Honeywell. It protects the controller network, using embedded cyber security, against message flooding and denial of service attacks. Other vendors include Plantdata Technologies (Pollett, 2006) and Rockwell Automation. Plantdata boasts that the new type of firewall they recently developed delivers a higher level of network segmentation and defense due to the way it is distributed throughout the control system environment.

Communication and software vulnerabilities are as important as security vulnerabilities in control hardware (Byres & Franz, 2006). Byres and Franz (2006) state that software failures are not responsible for many of the ICS vulnerabilities, but instead have resulted from failed procedural or administrative security controls. Mis-configuration vulnerabilities, inherent protocol vulnerabilities, implementation vulnerabilities, and product design vulnerabilities are all part of a product’s lifecycle. Byres and Franz (2006) suggest that vulnerabilities be classified as to how and where in the lifecycle they entered. The last point they make is that ICS security will become an expected quality assurance issue as these controls are embedded into the products and cooperation between the users, vendors, and standards bodies result in proper security expectations that lead to more experience and testing.

As a way of improving business and government processes discussed repeatedly in the media, an exploitation of IS efficiencies are being explored by James, Mabry, St. Leger, Cook and Huggins (2012). Therefore, an estimation of the cyber-physical situation requires capabilities for the continuation of incremental fielding. James et al. (2012) proposes an approach that achieves a science and framework for subjective validation of compositions of components that comprise an approximation of the behaviors of objective experimentation and the domain of interest in order to achieve incremental fielding capabilities with tools used to estimate the cyber-physical situation (James et al, 2012).

There is a new emphasis in the response to physical breaches (break-ins) and unauthorized entries into unmanned CI locations and sites where these incidents have been historically viewed as traditional property crimes only (Henry, 2006). Today it is necessary to consider the possibility of dispatching the cyber security incident response team (CSIRT) to those facilities containing ICS after such an incident. Although, this might not seem to apply where it is apparent the incident was obviously a case of vandalism, theft or trespass and considered to be the intruder’s sole motive. However, when electrical power substations are involved, an emerging concern to consider is what was traditionally thought of as typical physical crimes may now have more nefarious implications, as those tactics become a ruse for shielding a more sinister cyber-crime, especially with an increase in the use of remote control and monitoring of unmanned facilities (Swartz and Assante, 2014). An example is breaching the security of an electrical substation or unmanned generator facility and stealing equipment or materials. Although this could genuinely be an actual property theft or burglary, in this day and age, an event such as this may be a decoy to a real, more insidious motive of distracting the asset owner’s investigation and attention elsewhere. Swartz and Assante (2014) inquire about the effect if the motive is access to the systems and devices within an area such that the individual can conduct future attacks, and misappropriate confidential and proprietary information.

Today’s open market offers literally thousands of readily available network surveillance products and high-capacity keystroke loggers. In fact, many remote-access Trojans (RAT) include key loggers. Although they are extremely easy to use, they can be difficult to detect. They can be wireless or wired, and/or software or hardware based. USB keyboard emulators also possess the potential for exploiting systems. Because emulators are used for initiating keystrokes versus simply recording them, some emulators could be designed for malicious purposes posing an even greater threat than key loggers (Swartz and Assante, 2014).

Swartz and Assante (2014) growing understanding of the Stuxnet worm, originally carried in on a USB stick, demonstrates the effectiveness of surreptitiously gaining physical access to private or isolated networks. Security professionals often employ such tactics when conducting sanctioned network penetration tests for corporate clients. Another timely article reported that an unmarked computer had been discovered running in a spare room of Iceland's Parliament (Swartz and Assante, 2014). It was seized by police, and its exact purpose has not yet been determined or revealed. However, the fact that it was an unauthorized device connected to their network and was devoid of fingerprints or identifying serial numbers suggests malicious intent (Swartz and Assante, 2014).

Production Facility Electrical, Construction, and Maintenance

Understanding the various sources of noise is essential in recognizing the interference of legitimate signals and data degradation transmitting across the network. The reference to noise is not confined to acoustics alone. Identifying and eliminating noise (Pivonka and Mazzuchelli, 2005) is a key part of continuous network troubleshooting and maintenance, and remains to be a major concern of LAN network certification, testing, and installation. Poor workmanship performed on cabling can create internally generated noise along with that emanating from external sources such as that of nearby transmission lines and equipment in the form of electromagnetic interference (EMI) (Pivonka and Mazzuchelli, 2005).

Managing noise becomes an increasingly important issue as network performance levels and bandwidths continue to escalate. There is a greater susceptibility to interference from shorter signal pulses, and more data is packed into shorter time cycles for achieving higher speed. Most often there is an unfortunate subtle degradation of performance before there is a hard failure of the network caused by the negative effects of noise. Noise interference can cause data to be improperly received resulting in retransmission of data which essentially creates network congestion all the while slowing down the completion of the data transfer (Pivonka and Mazzuchelli, 2005).

Normal mode and common mode network noise are just a few types of noise varieties Pivonka and Mazzuchelli (2005) explain in their literature. There is no relationship to the fundamental frequency bearing from the disturbance of a waveform representing noise at its basic level, and; thereby, interfering with its capabilities of carrying the waveform's signal. Since a dependency exists with signal integrity to maintain the differential relationship between the wiring pairs, common mode noise in twisted-pair networks is of particular importance.

The difference in potential between two physically remote grounds creates common mode noise. An antenna can be created from an ungrounded shielded cable or poor ground system causing an induced voltage to be gathered and applied to the input. This type of noise, particularly with today's high frequency networks, is becoming more problematic by increasing the frequency of the noise, thereby, making it hard to eliminate (Pivonka and Mazzuchelli, 2005). Pulses from individual signals become much less distinct when a data signal which contains a lot of noise is severely differentiated, or attenuated through natural capacitance, thus making data reaching the far end of the transmission less likely to be received correctly (Pivonka and Mazzuchelli, 2005).

With network speeds constantly increasing and circuit logic voltage levels simultaneously decreasing, the ability to maintain precise waveforms is becoming even more difficult. As described by Pivonka and Mazzuchelli (2005), major effects to a link's data-carrying capabilities can occur in the presence of noise levels at less than 1V and can have a major effect even when using 3V logic at 350 MHz rates for transmitting signals.

A major factor contributing to test failures in LAN links is caused by external noise, such as EMI sources. Unintentional radio frequency (RF) signals can be emitted from a variety of devices, such as office equipment, power lines, fluorescent lights, computers, stereo and television sets, power tools, and factory floor production equipment, and thus can radiate EMI (Pivonka and Mazzuchelli, 2005). Particularly difficult environments, causing cross-coupling across nearby cable links and providing routes to ground from many different signals, include wiring closets and patch panels.

Data cables in areas where the surrounding building is poorly grounded can suffer from the radiated effects imposed on them as a result of the prevailing neutral-to-ground voltage conditions. The use of shielded cabling may seem obvious for reducing external radiated energy; however, there is a side issue to consider. The building ground should be an important aspect of shielded cable worth remembering since the shielding is typically tied to it. Therefore, it follows that in a poorly grounded building, the cable shielding does not provide a benefit by cancelling or eliminating the noise, but in fact may actually contribute to it (Pivonka and Mazzuchelli, 2005).

The first and most critical step in troubleshooting noise-related failures, relating to noise detection and analysis, is the ability to distinguish between different sources and types of noise.

However, some test instruments see different types of noise the same. These testers may see what looks like radiated noise from external sources when in fact it is nothing more than a form of crosstalk caused by the energy radiating between pairs (Pivonka and Mazzuchelli, 2005). Technicians may discover a “false failure” in instances where cabling proximity is too close to external noise radiation; resulting in wasted time troubleshooting a non-existent problem.

In Pivonka and Mazzuchelli (2005) final analysis, field test instrument requirements are re-emphasized to ensure they have the capabilities for conducting noise testing throughout the entire network. They must be able to isolate, identify, analyze, and measure the noise in that environment within close precision to the specifications required or recommended by International Standards Organization (ISO) and Technology Industry Association (TIA) standards.

Sensors

The normal or typical PC has no “senses” per se, in that it does not “know” what is happening in its surroundings; as the typical PC configuration is not set up to distinguish or determine whether it is noisy or quiet, dark or light, or cold or hot (Copley, 2015). However, there are onboard components that can provide that information and make it interactive with the user. For example, modern tablets, laptops, notebooks, and cellphones use either an ambient light sensor or camera lens to adjust the screen brightness depending on the environment’s ambient light and user settings. Also, digital thermal sensors (DTS) used for measuring a PC’s central processing unit (CPU) temperature will activate and control the performance of that system’s cooling fan. Unfortunately, the DTS may not be able to reconcile the CPU temperature with that of the environmental temperature of the user. Unlike the DTS, some components are purpose built for more user specific functions, such as the camera functionality simply being there for taking pictures or making movies. PCs can, however, obtain additional functionality and utility by knowing its environment with the addition of specific types of sensors. A sensor can record real-world activity (i.e. temperature) into information that can be analyzed by a computer. temperature) (Copley, 2015). The senses we use as humans for knowing what is happening within our environment are our: skin, nose, mouth, ears, and eyes. Table 2 lists the types of sensors and their corresponding detection properties.

Computer and industrial control systems rely extensively on sensors for performing data logging, measuring, and monitoring tasks. A sensor is used to measure a specific property which is received and processed by the computer. An Analog-to-Digital Converter (ADC) is used for converting analog signals into digital data a PC can process (Copley, 2015).

Table 2. Sensor and Detection (Copley, 2015)

Sensor	What it Detects
Movement	Motion-Still/Active
Proximity	Distance-Far/Close
Switch or button	Contact-Open/Closed
Pressure	Resistance-Negative/Positive
Moisture	Wet/Dry
Temperature	Hot/Cold
Light	Bright/Dark
Water-level	Empty/Full

Summary

The literature review begins with an administrative overview to explain and provide a better understanding of the importance and expectations that the federal government, through its rules and regulations, has placed upon industries that make up our nation's CI, in an effort to make them more secure. The electricity sector has had mandates placed upon it by the federal government so that they must comply with a set of prescriptive measures in implementing cyber security requirements. In order to understand all the attention that industries reliant upon control systems are receiving, it is necessary to recognize their function and the role they play in our everyday life.

The literature review concentrates on the area pertaining to ICSs such as SCADA, PLC's, and DCS's and the security challenges surrounding it. Control systems are used in a variety of industries such as manufacturing, power systems, and water treatment. Some of these industries are critical to the safety and economic security of our nation. These industries have been categorized by the DHS as CI. The power industry particularly has not only come under tremendous regulatory scrutiny on environmental issues, but also of that pertaining to its cyber security state and the rigid security controls now placed upon it.

ICS used in our nation's CI are under constant attack. Early on there was a clear distinction between the particular functions and purposes that information and control systems performed; however, within the past ten years, control systems have evolved to integrate more with information systems. This makes for a less than desirable situation as aging control systems are being replaced with these hybrid systems. Whereas control systems had little in the way of security, information systems had and continue to place more focus on this area. However, unlike information systems, the reliability and dependability of a control system is vital, and the processes are more volatile; therefore, current security controls are often an impediment to those two factors.

The literature review continues by expanding the focus from CI2CS into other areas that elaborate on details relative to the proposed research. These other areas consider: differences between control and information systems; functions they perform; environments in which they operate; communication standards between them; vulnerabilities and how one might be introduced; types of sensor input and sensory receptors; automation concepts; network configurations; and risk mitigation methods. The review also includes and concludes with current literature addressing industry best practices and other research focused around control system security.

Chapter 3

Methodology

Overview

Building off previous research models and designs for determining near real-time risk assessments using HMM’s by Pak (2011), Pak and Cannady (2009), and Pak and Cannady (2010); and, SCADA test-beds by Moore, Ellison, and Linger (2001), Jung, Song, and Kim (2008), Hahn, Ashok, Sridhar, and Govindarasu (2013), Richardson and Chavez (2008) and Giani et. al. (2008), this research was conducted in a physical test-bed environment. By segregating networks and using independent sensor-based spatial supervision, as well as appropriately applying mean and variance equations, current risk levels of control system performance were determined; thereby, providing a prompt and current threat environment risk assessment of control system assets in a real-time manner.

The test-bed was designed in a manner similar to those from prior research models. Idaho, Argonne, Sandia, and Pacific Northwest National Labs, and the British Columbia Institute of Technology are all nationally recognized SCADA test-beds. On a smaller scale, Morris (2010), as previously referenced, has been conducting SCADA research since 2010, using a government funded test-bed at Mississippi State University. This research design closely resembled the Morris (2010) model, as there were certain requirements necessary for it to be recognized as a “legitimate” SCADA test-bed. The factors listed in Morris’ design were considered in choosing the control system components and designing the architecture used for this test-bed.

A test-bed permits the design of a custom fitted environment that allows more control over the variables; thereby, enabling the research to be conducted more efficiently and precisely by further decreasing the incremental frequency involved in fine tuning the production environment to that of its spatial sensors. This control allowed the processes' inputs and/or outputs (I/O), and reaction or response to the activity, to be properly scaled and calibrated before contemplating the application of applying it to a real-world application.

Research Methods Employed

This research considered a quantitative approach that took into account the actual real-time spatial, along with the physical, operating environment of a CI2CS, by using a variety of sensory inputs during random sampling, and evaluating that data against the system's operational assets and network activity (i.e., motors, pumps, real-time packet captures) and log data which arguably reflected the anticipated and expected condition of the operating environment it in essence created during that sampling instance. The functions and processes of a CI2CS environment, including its spatial conditioning, are accomplished through the control system (not withstanding any air-gapped safety controls such as halon fire suppression, safety disconnect switches, etc.); therefore, its entire environmental state can be considered a by-product of that space along with the objects it comprises and controls.

In an effort to effectively achieve detecting probabilities of a near to real-time risk (RTR), an actual CI2CS type of network environment (test-bed) comprised of a HMI, PLC, field device(s), and data historian along with access to electronic perimeter protection safeguards, such as a firewall, IDS, and IPS was assembled. As Pak (2011) suggests, it is more feasible to use a prototype network using a virtual environment such as Simulink and MATLAB applications, albeit in this case a physical control system versus a virtual one was employed as the test-bed, but a prototype none-the-less. All the necessary threat analysis and risk assessments were conducted in the prototype network (Pak, 2011).

This approach required several steps made up of tasks and sub-tasks, with the first step being the selection of a CI sector whose business process involved an operational or production environment that created a sufficient amount of production activity. The second step required determining the physical environment and defining the space(s) and objects that fulfilled the business process of that environment (test-bed development). The third step was to define or describe the function(s) and processes of that environment. In this case, the test-bed environment mimicked a chemical processing plant. The fourth step involved the development of a control system network and automating the processes. The fifth step involved developing a sensory programming module, known as a SDAPU. The sixth and fundamentally crucial step of the test-bed development was the establishment of an operational baseline to comparatively model data captured from the affected CI2CS and its environment, both in and out of operation/service. The baseline was used as the underlying foundation in recognizing and illustrating “normalcy”, and it essentially set the gold standard for future system modeling and initial algorithm design. The seventh and final step required evaluating the data with an algorithm for processing the data in a way to determine and alert to a suspected or known risk emanating from within the system’s internal virtual or external physical environment, but still within its confined space.

Other activities involved sub-tasks such as performing ongoing “tuning” (necessary in determining the threshold values required for comparing sensory to system data over a set period of time), acquiring and installing assets and sensors, and automating the production environment. Table 3 shows the steps and related tasks, as discussed above, that were used in carrying out the proposed research. The aim of the steps, tasks, and sub-tasks were to meet two control objectives. The first objective was to establish a suitable test-bed. The second objective, and dependent upon the first, was to ascertain quality and viable results.

Table 3. Steps, Tasks and Sub-Tasks Research Requirements

Steps

No.	Task	Sub-task
1	Selection of one of the 16 critical infrastructures industries as ranked by DHS (a pre-requisite to any selection is the necessity for operational/production noise)	Identifying a process or set of processes performed by an industry providing an environment rich with stimuli
2	Defining the physical environment of the selected mock industry (spatial parameters – closet, cabinet, room, building, warehouse, yard, etc.)	Providing noise abatement Ensuring rigidity and stability to structure Implementing physical security controls Enabling adequate access in and out and to and from test-bed
3	Equipping and defining the mock industries' functions and operational assets and processes	Installing assets
4	Develop control system network and automate processes	Starting operations Configuring processes Tuning production processes Evaluate assets and conditions for sensor installation (i.e., temperature, vapor, audible, vibration, light, moisture, pressure, movement)
5	Develop SDAPU	Identify various environmental noise (for use in sensory perception) baseline Collect data from both noise and operational/production processes
6	Establish operational processing baseline	Analyze and compare data Synchronize baselines (tune as necessary) Monitor and record operating activity for specified duration as required Confirm baseline values for consistency/deviation
7	Develop algorithm for comparing control system data to spatial data and evaluate the data	Introduce exploit into control system network Observe, record, and monitor behavior of processing environment Record, monitor, and observe reaction of the SDAPU Acknowledge SDAPU alarm Document and analyze findings

Selecting the Sector - Step One

The test-bed was comprised of a type of environment that generated various types of activity dynamic enough for producing the stimuli sufficient for stimulating sensor input; therefore, identifying an appropriate business, its functions, and type of environment was a pre-requisite and fundamental step in designing the experiment. Although there are sixteen CI sectors classified by DHS as mainstays to America's security and economic interest, not all are involved in manufacturing, and of the few that are, they do not necessarily have the conditions for generating sufficient "environmental noise".

Environmental noise in a control system environment can be compared to what some system manufacturers refer to as a heartbeat. A heartbeat is described as regular repeated signals resulting from systems, and evidencing normal activity within a system. These signals can be configured into the equipment. If the heartbeat is atypical of the norm, then this means that there is an issue. If there is corruption or unauthorized entry, problems will occur. (DOE, 2014)

As a way of emulating a legitimate CI facility, the test-bed was designed and built around the chemical sector in the form of a "specialty chemicals" processing facility.

Physical Environment - Step Two

The materials used in fabricating the test-bed manufacturing floor mimicked that of its real-world counterpart. This is an important detail when considering sensory tuning, especially as it relates to vibration, noise, and temperature. Vapor detection and visual indicators such as those that detect gases and liquids rely more on internal stimulus within a controlled and confined area than that in a well-ventilated or open exterior area. Vapor refers to non-visible gases for example such as carbon monoxide. Visual indicators are such things as movement, colors, fumes or moisture. For example, wetness detected on an exterior concrete pad might be more indicative of rain than a leaking pipe. However, wetness detected inside a facility should not be from rain and most likely is not, but instead is indicative of a leaking water line, tank, pump, or other component related to systems involving liquids. Therefore, relying on a moisture sensor outside a facility for the purpose of detecting moisture from tanks or leaking pipes may not be wise while it is raining, as it would be using a pressure sensor and flow meter. However, inside the facility a moisture/wetness sensor would be equally practical and efficient as a flow meter or pressure sensor.

All the assets were positioned and mounted horizontally on two, three by five foot Durock cement boards' butted side-by-side forming its base. Durock cement board was used to provide a foundation similar to that of concrete flooring, the material used in the majority of industrial commercial production facilities. There were two-foot-high walls, framed by metal studs, enclosing all four sides. The foundation was elevated two feet off the ground. Foam board insulation was used to fill the cavities between the studs. A flat metal roof, as used in most, if not all, industrial manufacturing facilities was affixed to the top of the scaled chemical processing structure (facility). The roof was insulated similarly to that of commercial buildings used for the same purpose. Insulation foam was used to seal the cracks between the foam board.

Design (Equipping Mock Industry with Operational Assets) - Step Three

The actual processing equipment, Table 4, consisted of COTS products that were not necessarily designed to be used in industrial applications, such as production environments. However, these products were designed and chosen to closely resemble those used for such an application in such an environment. The assets selected for this research withstood the rigors placed upon them.

The industry selected for this system design was modeled after a CI chemical manufacturing facility that processes extremely toxic chemical fluids. Since the fluids are highly toxic, processing is performed through a variety of automated processes without direct human interaction. These automated processes are accomplished using pumps for fluid transfer, a robotic arm for material handling, a centrifuge for fluid separation (in a mock quality control sampling process), and a SEIG that although functional, only simulated supplying power to the robotic arm.

Table 4. List of Automation Assets used in CI2CS Test-bed

Qty	Item	Automated Process	Purpose
1	Motor	Prime Mover	Simulates fuel source for generator
1	Generator	Powered mini-grid	Produces power to simulate grid
4	Pumps	Fluid Transfer	Transfers fluid from supply tanks to product tank then back to supply tanks to recycle batch process
1	Robotic Arm	Assigned Movement	Simulating the fill, alignment, placing, and in and out of centrifuge rotor
1	Centrifuge	Spin down fluids	Separating dense particles from less dense particles within fluids

The control system assets, also referred to as critical cyber assets or cyber systems (if made up of more than one cyber asset), would normally have had occasional human interaction, but in this case, were in fact continuously monitored during each production cycle. The control system cyber assets included a: firewall; network tap; two switches (one managed and one unmanaged); one notebook; and, three laptops. Table 5 contains an inventory of the control system cyber assets including its function and purpose.

As much of a limitation COTS products proved to be, it would equally contribute to the expected, albeit accelerated, degradation of performance that would occur from a compromised environment, especially when brought on suddenly or sporadically, such as that from a virus introduced into the control system or simply a mechanical failure. However, in an otherwise stable and normal operating environment, the sensor network baseline would naturally trend toward the declining performance in-between maintenance intervals, so that

Table 5. List of Cyber Assets used in CI2CS Test-bed

Manufacturer	Qty	Type	Function	Purpose	Model	OS/Firmware/Software
Dell	1	Laptop	Engineer Workstation	Programming Test-bed Control System (Connected Components Workbench)	Vostro 1200	Win 7
Dell	1	Laptop	Operator Workstation	Process monitoring	Inspirion	Win 7
Toshiba	1	Notebook	Security Platform	Software based IDS – Snort	NB310	Win 7
Dell	1	Laptop	Server	KepWare Log Server/Historian	Latitude	Win 7
Cisco	1	Firewall	Security Appliance	Network device and Port management	ASA 5505	Cisco
Netgear	1	Managed Switch	Networking	Managed Control System Switch	ProSafe GS108E	Cisco
Netgear	1	Switch	Networking	DMZ	ProSafe FS105NA	Cisco
US Robotics	1	Tap	Network Aggregator	Unidirectional communication to IDS	USR4503	None

false positives were eliminated or at least minimized. No decline in performance was noted during this experiment.

Develop Control System Network and Automate Processes - Step Four

The configuration of the application processes initially involved automating the functions of the production assets in order to accomplish their intended task of processing a chemical solution by: transferring, back-filling, centrifuging and separating. The substantive stimuli outputs focused on during this four-task process included the pumps used for general fluid transfer and the robotic arm used in the simulated sampling process. The functions for the robotic arm were controlled with an Arduino Uno.

The centrifuge, generator, motor, and pumps were all automated; however, a certain number of alterations or modifications were made so that the devices would perform as intended. This particularly related to the centrifuge being fitted with a solid-state relay for switching the PLC 12 VDC output to the 110 VAC line voltage powering the test-bed. The 220 VAC three-phase motor was connected to a 120 VAC motor controller, also known as a variable frequency drive, and was used in rotating the shaft of the SEIG. The robotic arm was the only device that functioned using an Arduino to independently control it. The remainder of the automated assets used in the production process were controlled with an Allen Bradley PLC.

The task specific functions differed between the Arduino Uno and PLC in that the Uno controlled the robotic arm performing the simulated sampling task, that involved the collection and placement of a vial from the tank area to the centrifuge area, while the PLC instructed and controlled the sequence of tasks performed throughout the entire process; including the initialization of the Uno. A SEIG was used to provide stimuli and simulate a grid; however, it did not apply power to the equipment performing any of the automated tasks. This is explained in chapter four.

After the robotic arm simulated loading the rotor of the centrifuge, the centrifuge operated for two minutes. Two minutes was chosen as a general time used in such a process, relative to the type of solution being spun down, and the consistency of the product being manufactured, but more importantly it enriched the environment with additional stimuli for an ample length of time.

A complete production cycle was considered finished once the centrifuge rotor came to a rest. However, this only applied before beginning the next cycle and not as it related to the data logging during that period of time. Data logging ended once the OPC server registered a zero value for the centrifuge process.

There were four pumps installed in the test-bed and they were used to move the fluid from two “supply” tanks into the “product” tank. One of the two pumps connected to the product tank pumped the fluid from a supply tank, labeled A, for a set period of time before pumping fluid from the supply tank, labeled B, for a set period of time. Once the transfer had been completed, a single pump connected to each of the supply tanks would replace the water taken by the product tank pumps. The entire “batch processing” design for transferring fluid resembled that of the DHS, INL test-bed.

There were only very brief and intermittent pauses throughout the process. As one process ended, another began. All of these processes, although performed in sequence and not simultaneously, provided a sufficient number of stimuli that created a dynamically rich environment and offered a large number of variations to the sensory baseline. A list of control system assets such as a PLC, motor controller and HMI are included in Table 6.

Table 6. List of CI2CS Process Control Assets used in Test-bed

Manufacturer	Qty	Type	Function	Purpose	Model	OS/Firmware/Software
Allen Bradley	1	Motor Controller	Primary Mover for Generator	Supplies power to mini-grid	PowerFlex	Win CE
Allen Bradley	1	PLC	Logic Controller	Controlling Test-bed Processes	Micrologix 850	Win CE
Allen Bradley	1	HMI	Visual interface with processing system	Monitoring and Interfacing with control system	T600	Win CE
Arduino	1	Micro-controller Device process controller	Robotic Arm	Simulate collection and placement of test tube samples	Uno	Sketch

The test-bed manufacturing processes were controlled by a PLC, sharing a similar system configuration to that of the PLC control system implementation example as illustrated in Figure 1 (NIST SP-800-82, 2015). Both SCADA and DCS systems use PLCs for managing local processes where they provide feedback control as described below (Stouffer et al., 2013).

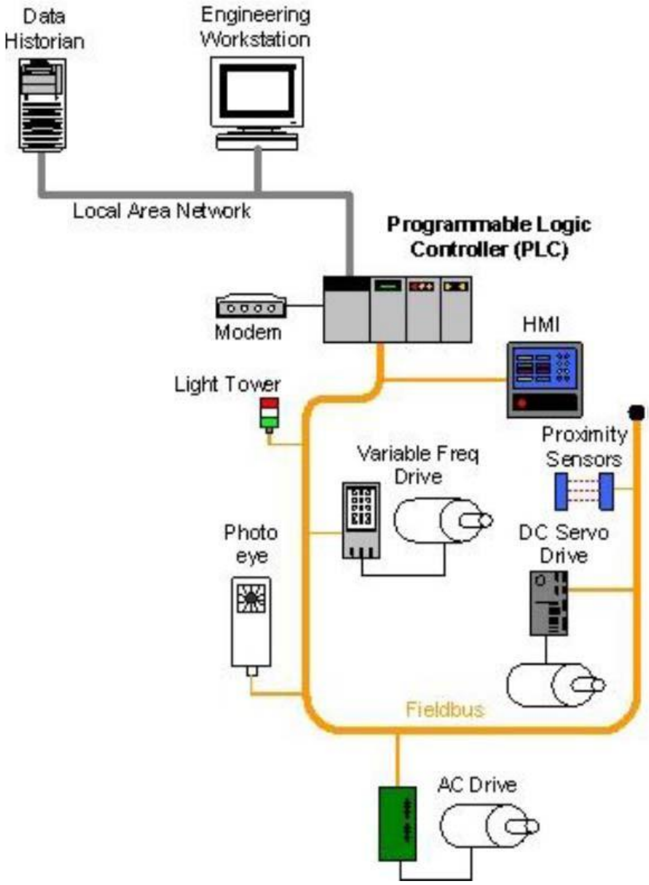


Figure 1. PLC Control System Implementation Example (NIST SP 800-82, 2015)

A PLC component provides control as part of an overall hierarchical system. Although the functionality it provides may be the same as RTUs, they are typically purposed quite differently. However, it is not uncommon for them to be used in SCADA systems to perform an RTU function. They are also used as part of a supervisory control scheme for providing local control in a DCS (NIST SP-800-82, 2015).

The operational control of discrete processes can typically be satisfied by a PLC, especially where requirements for smaller control system configurations are necessary, such as chemical manufacturing and power plant environmental emission monitoring controls (Stouffer et al., 2013). DCS and SCADA topologies differ from that of PLCs, Figure 1, in that the closed-loop control they provide does not directly involve a human; therefore, they normally do not have a central control server and HMI (Stouffer et al., 2013). However, as pointed out in *Step Three*, the test-bed in this study included a HMI. This test-bed was not a closed-loop control system, but instead an open-loop system.

A closed-loop control system differs from an open-loop system in that the control system receives signals from the field devices and reacts or adjusts according to the inputs it receives (ABB, 2016). For example, a ventilation fan may turn on based on the analog temperature input received by the PLC. Likewise, the PLC will send an output to turn the fan off once the temperature is within an acceptable range. In an open-loop system, the controller is not getting "feedback" per se, so functions are limited to an essentially on/off scenario. Some controllers may use a combination of both an open and closed-loop system (ABB, 2016).

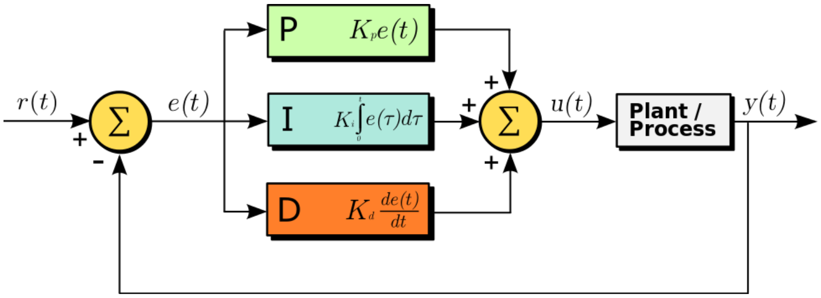


Figure 2. Three mode proportional-integral-derivative

The memory of a PLC permits storing of instructions for specific functions. (Stouffer et al., 2013), Figure 2. The PID controller is one of the most widely used feedback controllers (ABB, 2016). Using an error signal, the PID produces a control signal (ABB, 2016). Because the desired parameters/conditions for a closed-loop system are normally achieved by tuning the system to the inherent conditions without specific knowledge of a plant model, stability can often be ensured by using only the proportional term (ABB, 2016). A pure proportional model will result in a control offset; however, the integral term eliminates the control offset and derivative term provides damping or shaping of the response (ABB, 2016).

PID is called out because a closed-loop PID controlled system could potentially involve some, at the least - slight, modifications to the systems behavior, despite it being for the sake of improving performance. It is important to be aware of such parametric and conditional changes or modifications for no other reason than, if using a sensor-based parallel risk monitoring system on a PID controlled system, false positives are not produced as a result of the tuning taking place by the PID. Arguably the performance of a PID controlled system would be more stable and consistent than a non-PID controlled system; therefore, a deviation in a PID controlled process, as monitored by the approach used in this research, might suggest the PID itself has been compromised.

Figure 1 shows a fieldbus network for a manufacturing process using a PLC to perform its control processes. The test-bed used in this research did not have a fieldbus by definition, but instead the individual devices were connected directly to the PLC's I/O terminals. While the PLC controlled the operational processes for three of the four tasks of the production/manufacturing environment, the Arduino was used to automate the otherwise mechanical processes for one of the production assets. In this experiment, a HMI and engineering workstation, with a programming interface-Connected Components Workbench, was used for accessing and interfacing with the PLC. Kiwi's Syslog Server was used for process and file integrity monitoring. KepServer provided an OPC client/server that was used to communicate with the PLC, and served as the PLC historian and data logger.

Unlike a PLC with its limited processing capability, as depicted by a somewhat simplistic illustration in Figure 1, in a typical DCS configuration, Figure 3, one control box can execute from 1 to 256 regulatory control loops from geographically distributed digital controllers providing various functionality (Stouffer, Falco, Kent, 2011). The capabilities of a DCS were not necessary for this implementation, and clearly outside the scope of this research. A DCS is

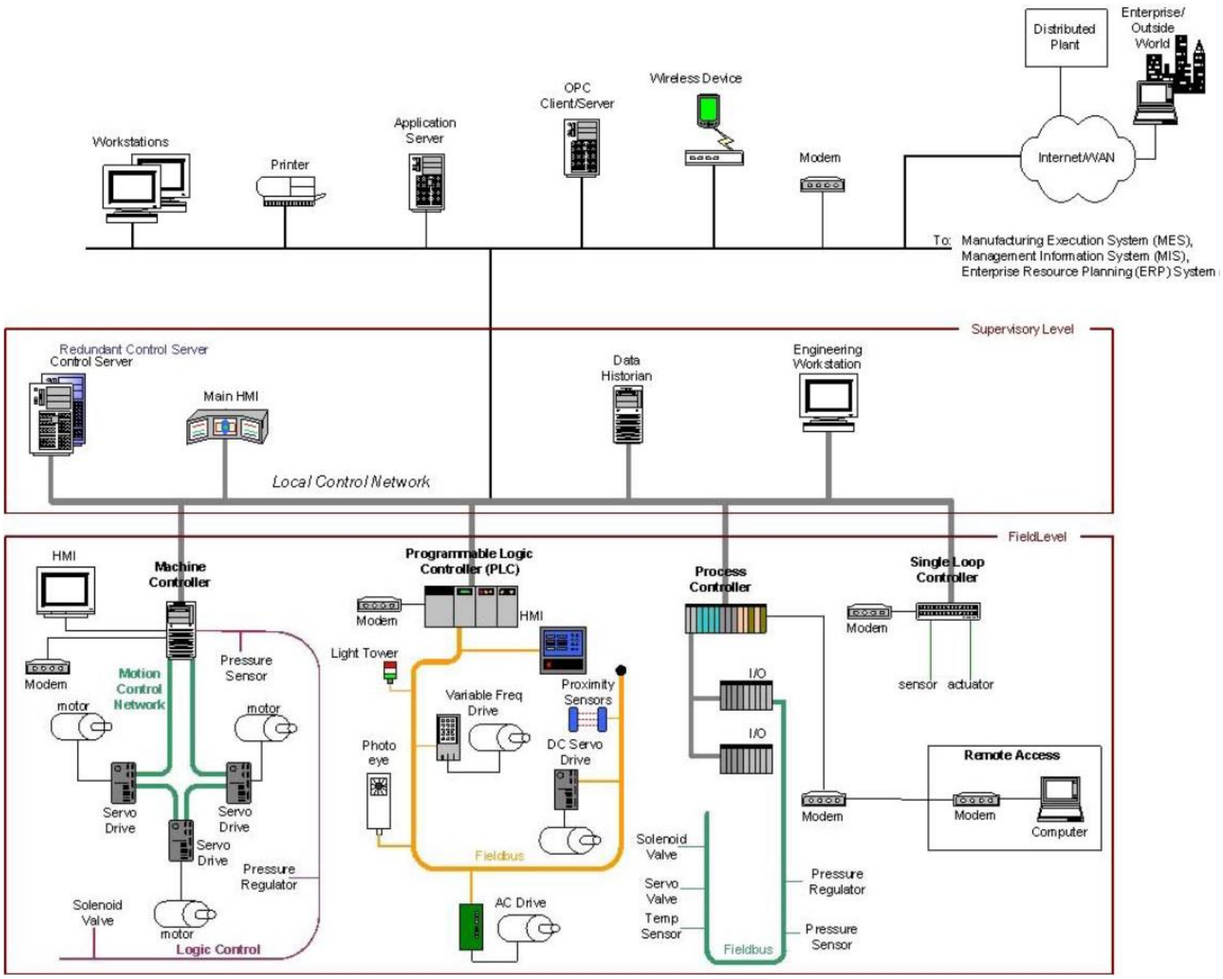


Figure 3. DCS Implementation Example (NIST SP-800-82, 2015)

essentially a master controller to more than one slave controller. Having less than a two-controller relationship with a DCS might arguably be considered a waste of functionality.

Although this experiment used two controllers within the control system process, they were not configured in a master/slave relationship. In fact, the Arduino had no communication with the PLC. Instead, it only provided an input signal, in the form of voltage, to the PLC that triggered, then maintained a two-minute output signal to the DC-AC relay being used as the on/off switch for the centrifuge.

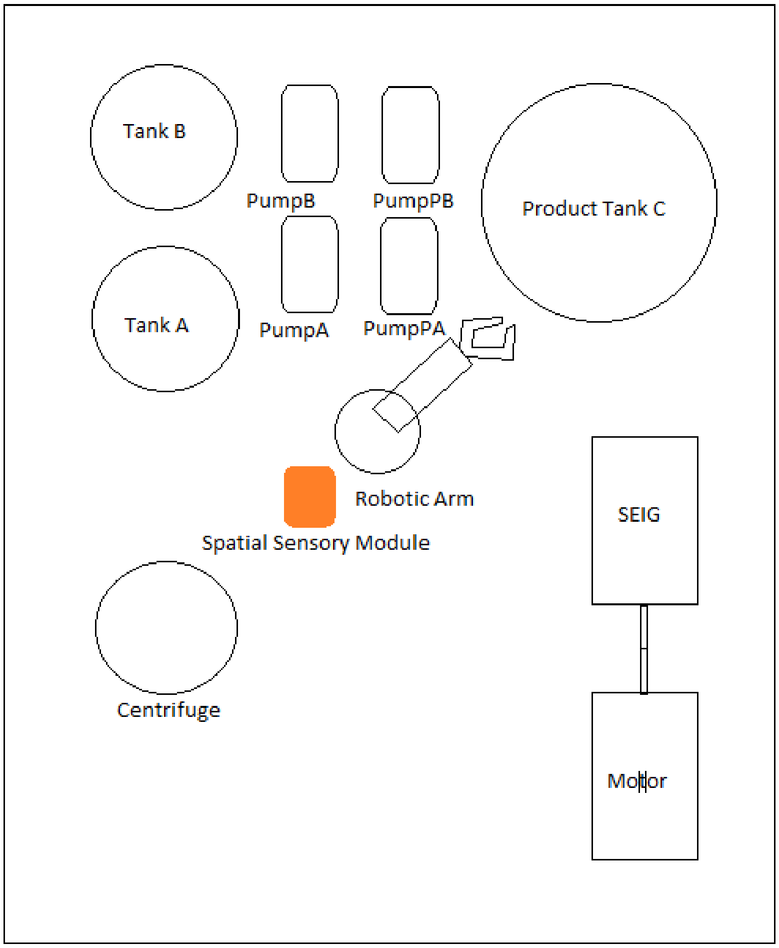


Figure 4. Test-bed components with spatial sensory module (SSM) in place

Develop SDAPU – Step Five

The SDAPU was used for logging and processing the sensory data acquired by the spatial sensory module (SSM). The SDAPU, although originally conceptualized as being a Raspberry Pi, ultimately ended up being a Wintel based platform (Dell laptop) running the Windows 7 operating system. The primary reason for the transition from a Raspberry Pi 3's Arm controller to that of an Intel based controller was in large part due to the numerous compatibility issues, caused by the collection of software assembled to provide the SDAPU its functionality, it had with the Pi's native Linux based operating system (Karvinen & Karvinen, 2014). Microsoft has a version of the Windows 10 operating system that is compatible with the Pi 3 model; however, some of the applications used in this research as part of the SDAPU were not compatible with Windows 10 either (Monk, 2013). Generally speaking, a number of SDAPU applications were not compatible with systems running Linux. Thus, the Wintel platform was used because it was compatible.

The SDAPU consisted of the following five core programs: CoolTerm, used for capturing and logging the sensor's input serial data; National Instrument's, SignalExpress, used for signal analysis; VMware's, Workstation, used for hosting Hortonworks; Apache's, NiFi, used for data ingestion; and, Hortonworks Sandbox, embedded with Ambari and Hive and used for real-time data processing.

The SSM was developed using an Arduino Uno as the controller which inherits the C programming language used in writing the controller functions. It resided in a central location as seen in Figure 4. This module works alongside the field devices as seen in the overlay in Figure 5 later. The SSM was comprised of three primary sensors which were a: DHT11, that measures both humidity and temperature; accelerometer, used to measure vibration; and, a sound recorder, used for monitoring sound. Initially a geophone was considered and used to measure spatial vibrations; however, it could not be sufficiently tuned to this environment. The automation spatial information was monitored during each production cycle.

Chapter four provides more detail in regards to how the data was captured and processed through the presentation of the data. As a preliminary explanation of the process, the SSM monitored the environment as the processes occurred. The data produced by the SSM was captured to the SDAPU and called on demand by the operator at various times during the production cycle. For this research, being a controlled experiment, the calls were only placed during the sampling cycle of the four-part process, albeit, at various times during that specific process. The real-time data was processed instantaneously using the Hortonwork's platform as described in more detail in chapter four.

The real-time physical environment is comprised of all sensory measurements very similar to that of a human; those being: touch, taste, smell, audio, and visual. The sensor network originally included sensors that essentially accounted for the five human senses a production technician might utilize if they themselves were performing a routine “walk down” or inspection of the production area. In a chemical processing facility for example, this might equate to picking up on certain odors, sounds, colors, activity, vibrations, and temperatures or pressures that produce a certain characteristic at certain or all times.

Two primary stimuli were present in the mock chemical facility test-bed. They were sound and vibration. This was naturally by design, as many production type processes involve a considerable amount of mechanical automation as opposed to mechatronics. Mechanical automation creates stimuli such as sound, vibration and heat.   Mechatronics tends to be more refined and although it may create the same stimuli, the stimuli and processes tend to be more subdued.

The temperature aspect was considered and tested during initial baselining. The ambient temperature stayed consistent throughout the course of testing, and the mechanized components when operational did not alter the test-bed’s overall climate. Two other sensors, specific to the environmental monitoring included a DHT11, which incorporates a humidity and temperature sensor. Although the temperature, humidity, and relative humidity was monitored and recorded during every operation, the environment changed very little throughout the course of this research.

A FLIR thermal camera was used for this experiment and directed toward the four pumps that performed the batch process or fluid transfer. The thermal image quality was good, relative to the part of the process and production cycle taking place. Unfortunately, the image did not normalize until after the first, and sometimes second, production cycle took place. This was caused by the displacement of water from the supply tanks into the product tank. As the 15-gallon product tank began to fill, the ambient air, during the initial cycle, within the tank began to mix and evacuate from the container. As this air turbulence occurred it would wash out

Table 7. List of Senses and Corresponding Sensor Type

Sense	Sensor
Hear	Microphone
Smell	Alcohol Gas
	Carbon Monoxide
	LPG Gas
	Hydrogen Gas
	Methane CNG
Taste	pH
Touch	Gas Pressure Monitor
	Vibration meter
	Thermometer
	Inertial Measurement Unit
	Pressure Gauge
	Load
	Humidity and Temperature
	Vacuum Gauge
See	Thermal Imaging
	Color Spectrum
	Video Camera
	Hall Effect
	Motion
	Optical Dust

the right one fourth of the image captured by the camera. This is illustrated and discussed in chapter four.

Table 7 lists the sensors along with their corresponding purpose that were considered in the design of the experiment. The final decision in determining the appropriate sensor(s) used in this research was made while observing and monitoring each device as it performed its intended task. The final sensors selected after making several observations is listed in Table 8.

Table 8. List of sensors used in test-bed

Qty	Sensor	Purpose
7	Temperature	Motor temp, pump temp, robotic arm servo-2 temp, spatial
3	Ultrasonic	Measuring tank fill status
1	Humidity	Measure relative humidity of internal test-bed environment
1	Thermal imager	Providing visual relation of pump temperatures and relay-valve monitoring
1	Voltmeter	Measuring the SEIG voltage output
4	Flow meters	Measuring flow rate between supply tanks and product tank
1	Sound recorder	Measuring process audio noise
1	Accelerometer	Measuring process vibrations and specific robotic arm movements

Other sensors used in this research were for the purpose of operational monitoring, such as flow meters for measuring flow rates and ultrasonic sensors used for measuring distance. In this case, the distance was between the tank water levels and the ultrasonic sensor. A complete list of sensors used in this research are listed in Table 8.

An initial concept introduced as part of this research involved comparing sensory inputs against that of CI2CS network traffic. As these systems were independent of one another and the goal of the research was to determine whether or not a system or component within the overall process was introduced to risk, this concept was not applied. The IDS, antivirus and firewall collectively played a role in preventing and or alerting to an intrusion or virus. As this was a controlled experiment, the test-bed control system was air-gapped. There was no internet connectivity. Therefore, no intrusions, viruses, or DDoS attacks occurred nor were deliberately introduced into this network.

The security controls were unable to detect the upload of altered code to the robotic arm's processor. These conventional security controls do not prevent or detect the type of potential threat used in this type of configuration. This reiterates the very point of this research. Although, these controls did exist, were in place and configured accordingly, the comparison of benign data proved to be pointless.

Ultimately, the control system network was air-gapped from any external connectivity and resided as an independent network to the sensory network. No control system data was needed in the analysis of a production cycle's performance. This is what makes this research unique. The only connectivity between the sensor network and the control system network was via a voltage signal for the purpose of synchronizing the real-time clock (RTC) of the sensory data captured with the start of the process. Arguably this is not necessary, as a snapshot of the sensory data, compared to that of the baseline, would indicate exactly what process was occurring at that point in the production cycle. In this instance, the RTC also validated the fact that the process had begun and established the time at which it started.

As previously mentioned, the purpose of segregating the data between the control system and the CI2CS sensory output system was to prevent an infection from one system into the other. Since these networks operate independently from one another, sniffer traffic and sensory data are explicitly unique and original to their domains. This means that the real-time state of an object, space, or both; at any point in time, as evaluated and indicated by sensory data; can be independently and randomly verifiably accurate (in essence "pure"), as it is derived through its own system; and not from the same network where a potentially compromised CI2CS might influence or affect the quality and integrity of that data.

Establishing Operational Baseline - Step Six

The initialization of the cyber security baseline established in this experiment was guided by a select set of core references that are recognized by the electricity industry as robust and sound. These references were developed by NIST, SANS, DHS, NERC and the NRECA. Although NERC and the NRECA develop guidance specific to the electricity sector, the cyber security requirements, accompanied by their technical basis and guidelines can be applied to many industries requiring the need for maintaining reliability, regardless of its nature. This set of references includes:

- *NIST SP 800-53, Revision 4, Security and Privacy Controls for Federal Information Systems and Organizations*
- *NIST SP 80-82 Guide to Industrial Control Systems (ICS) Security*
- *NIST Framework for Improving Critical Infrastructure Cybersecurity*
- *NERC CIP standards, version 5/6*
- *NRECA Guide to Developing a Cyber Security and Risk Mitigation Plan – Update 1*
- *SANS Top 20 Critical Security Controls, version 5*
- *DHS Recommended Practice: Improving Industrial Control Systems Cybersecurity with Defense-in-Depth Strategies*

All but one of the references listed provide recommendations and guidance for implementing industry cyber security best practices, some more specific to control systems, while one mandates compliance requirements specifically for entities within the electricity sector. These references embody a set of ideal security measures and controls aimed at reducing risks in corporate IT and production environments operated by industrial controllers. The rationale in considering industry mandates requiring entities within the electricity sector to comply with a rigid set of requirements, for the express purpose of reducing cyber risks, is to demonstrate the rigor and minimum number of measures and controls deemed necessary for achieving a particular industry's accepted level of risk to one of the DHS designated sectors listed as CI in the interest of national safety and economic security. Another purpose for using these requirements in this research was to evaluate systems in an already perceived mature state, and establish that the experiment will enhance and improve the security of a control environment by determining risks using independent measures versus those linked to the same and potentially affected infected control system network.

According to the DHS, an ideal control system network, if unable to operate as an "island" through air-gapping, would be configured using the defense-in-depth configuration shown in Figure 5. There are five zones depicted in Figure 5. They are the external, corporate, data, control and safety zones.

In the "Safety Zone", note that the Safety Instrument Systems operate in a state disconnected from any network to ensure a positive safety response in the event of an emergency. This might include responses to such hazards as fire, overpressure, flooding, and electrical. A positive safety reacts by a triggering mechanism set off from a sensor through an automated or manual process. It is a local safety feature that cannot be tampered with remotely.

As depicted in Figure 6, a modification to the DHS defense-in-depth model shows a completely independent spatial and device centric risk detection sensory system residing within the Control Zone. It is depicted as appearing on top of the sensors and actuators connected to the

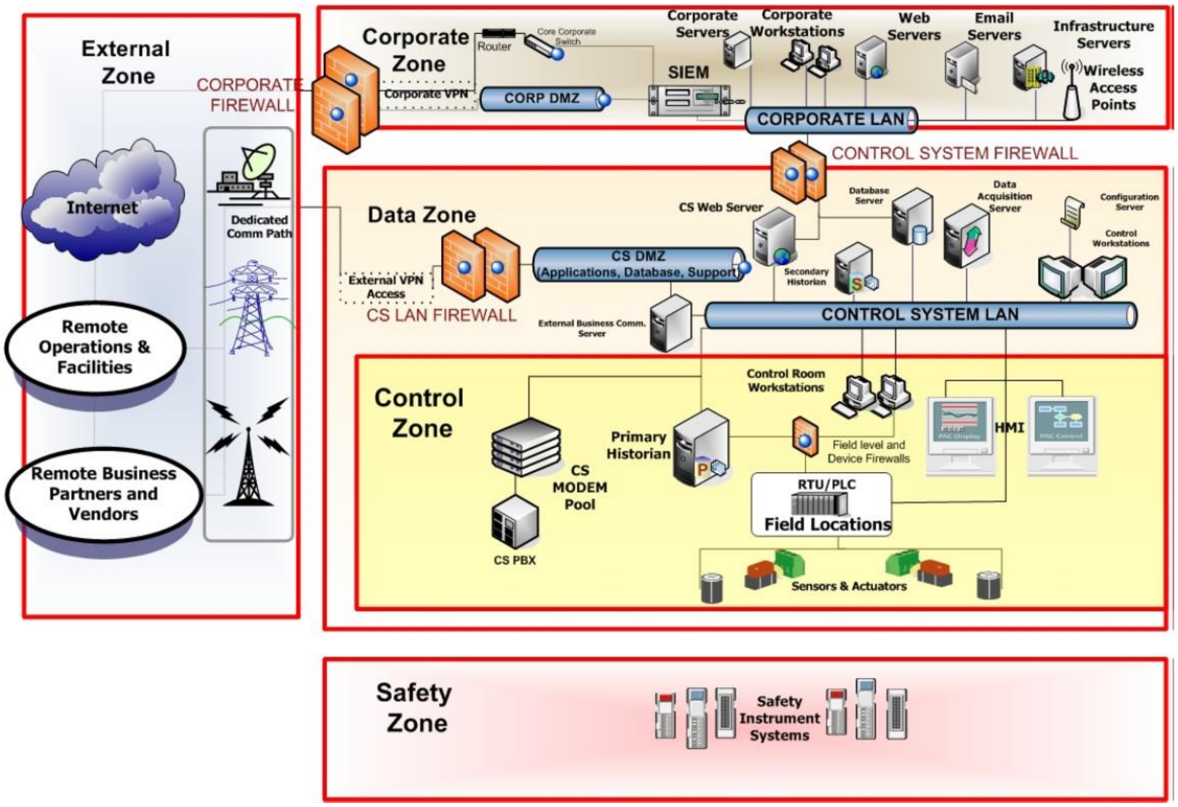


Figure 5. Recommended Defense-in-Depth network architecture for CI2CS (DHS, 2009)

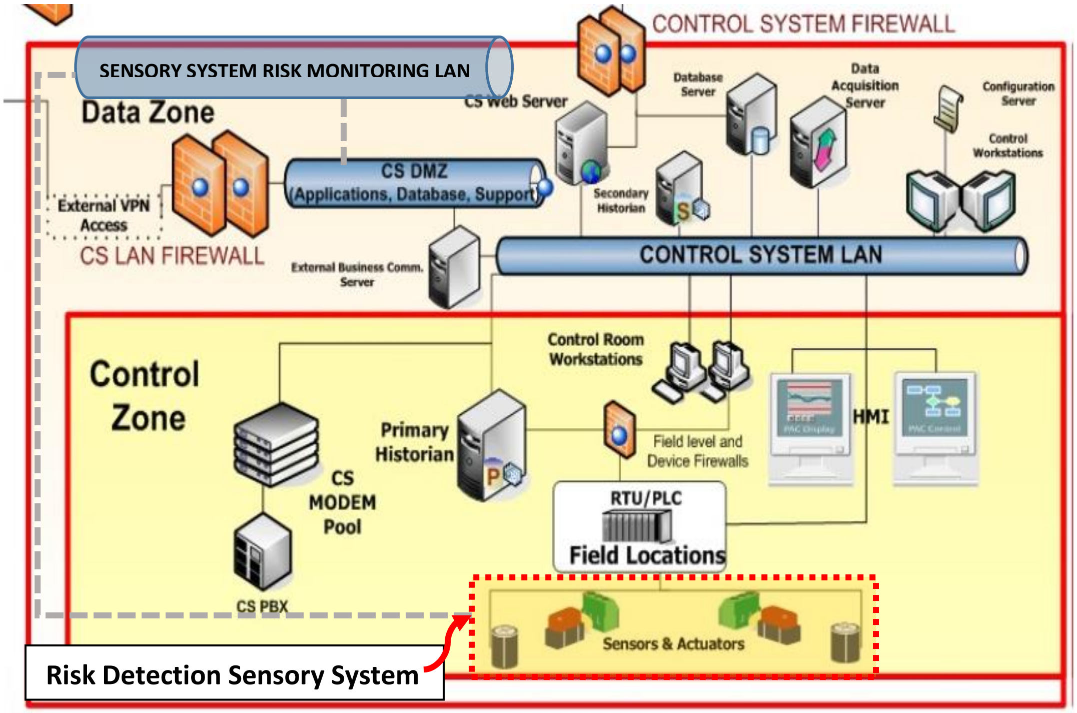


Figure 6. Close-up of recommended Defense-in-Depth network architecture for CI2CS showing integration of Risk Detection Sensory System within Control Zone (DHS, 2009)

RTU/PLC. For this research, it is a PLC. The risk detection sensory system was not connected nor was it intermingled with the control system communication network in any manner whatsoever. It does, however, require a single, specific, known output signal from a PLC or DCS, a PLC in this case, to initialize the SSM's RTC which synchronizes the start of the linear process used in this research and begins collecting data for the SDAPU to analyze and compare against the processing baseline. Although there are currently not as stringent governmental cybersecurity regulations in other CI sectors, this experiment will assume that there is a minimum set of standards employed across industries defined as CI, such as that of the electricity sector, which establishes a certain level of cyber security awareness and controls for ensuring the reliability of that industry. Some of the controls identified from the references listed above will take into account the "policy" aspect of cyber security and recognize that policy requirements and enforcement are crucial in achieving and maintaining a high-level of cyber security readiness. For the purpose of this research, it can be assumed that all policy related control(s) were considered and strictly adhered.

Baselining was performed over a sixteen-hour period with the initial goal being an effort to temper and normalize the spatial environment for conducting the remainder of the experiment. To further elaborate on this particular phase of the experiment, the test-bed in its entirety, meaning the structure and its internal apparatus were operated as designed throughout a number of processing cycles, at set intervals, as a means of determining operational normalcy. This was to develop the parameters for an initial functioning baseline and not of the more stringent performance requirements necessary for achieving an accurate and consistent operational baseline. Depending on the action or actions necessary and the average run time for a complete production cycle to occur, no more than four production cycles in an hour were possible. The first two hours were essentially spent discovering and researching the source of nuisance stimuli emanating during the production cycle, such as rattling, shaking, and vibrations. After determining the nuisance stimuli sources and then securing the various equipment, production cycles continued until a total of twelve complete cycles had been observed and monitored for significant and recognizable changes.

Initially, on average, the monitoring and supervision of the production cycles lasted over six hours a day with most testing being finished under nine hours. This schedule took place over a six-day period after the initial baseline had been established, and resulted in the acquisition of over 100 samples. Sample data acquired during the initial 50 production cycles were used for developing an operational baseline.

The operational baseline essentially formed the gold standard for all future processes. Any operational constraints were ultimately determined by the data quality and that of the processing equipment being used. The testing was accomplished without any significant incidents or notable observations outside of what was to be expected.

Sample Design

Algorithm Design, Data Collection and Software - Step Seven

This section describes the algorithm, data collection and software used for conducting the experiment used in this research. It also introduces the exploitation process, and how it was introduced into the control system network, through some explanation of the algorithm. This section covers the observation, recording, and monitoring of the test-bed’s environment and its characteristics. The automation equipment that was ultimately chosen to perform the processes throughout the product cycle provided a significant, satisfactory and dynamically rich set of stimuli which is described in the next section.

Data was collected from both the spatial sensory module and control system processes during the production cycle's run and non-run times (dynamic vs static). This validated the vast differences between the two modes. There was a minimum of three stimuli traits associated with each automated component. A temperature sensor could be applied to all the devices used in relation to either motor or fluid temperatures. However, the product cycles ran for a short time in general, such that temperature changes were only nominal. The spatial and robotic arm, servo-two, temperature data was recorded. Temperature data was not used in the risk-baseline comparison algorithm due to the nominal differences.

The most prevalent stimuli in this research were sound and vibration, both of these data sets were used as inputs in the algorithm. This is discussed in greater detail in chapter four. Data was collected during system processes for monitoring the system process, but not as part of the risk monitoring process. This included the flow rate and ultrasonic sensor that monitored the fluid transfer process. Arguably, flow rate and ultrasonic values could have been used as part of the existing algorithm. However, because of the reliability and accuracy found in the proposed algorithm using just the sound and vibration data, the flow rate and ultrasonic sensor data sets were not used for providing additional support for detecting risk.

The CI2CS baseline was the core dependency for performing a comparative analysis, with samples derived from all other production cycles. As such, it provided the known variables used in comparing real-time, random samples. The randomly acquired samples were compared to pre-defined baseline data points. The defined intervals were those intervals identified at specific data points along a linear pattern of a specific process. In this case, the robotic arm sampling process was used because of the deliberate manipulation and alteration of the code intended for insertion prior to the arm's actual task being performed. This was the part of the research design for the express purpose of disrupting the simulated sampling process and distinguishing a normal operation from an abnormal one. An abnormal one implied risk.

Software

The primary purpose of the software used in this experiment was for: logging, programming, monitoring, analysis, and configuration. The software provided a means to interact with cyber assets connected to the control system network and the micro controllers used for performing local tasks in the test-bed. This included applications for network monitoring, file integrity checking, malware detection, intrusion detection and equipment used for packet sniffing that sits on the CI2CS network that passed data to a syslog server located in the demilitarized zone (DMZ).

Hortonwork's platform was used to process the real-time data with that of a previously established baseline. The comparison was essentially a line-by-line relative time analysis of mean values derived from both the establishment of the baseline, and the snapshot of the recently captured data. Those two values were compared against the acceptable variance for that point (within hundredths of a second) in time and it was determined whether or not a potential process risk might exist based on the sum of valid data points used in the sample. This means that if *n* number of pre-determined data points were used for comparing samples, all *n* data points should indicate values within an acceptable range. A failure to match one or more data points was cause to render a graph of the data and study the deviation. A graph was used because it was observably clearer to evaluate the data and for identifying the discrepancy in the process. Based on the strategic placement of data points throughout the operating process, a snapshot of system sensor data was analyzed against both NULL and known acceptable values. Arguably, data points could be placed along the entire process and be continuously analyzed;

however, the massive amounts of data involved would distract from the stark contrasts seen between the random samples, discussed in chapter four, that illustrate this process.

Data Analysis

Sample stimuli assignment values corresponding to different assets are documented in Table 9. To further expound on the example of a motor starting, the SDAPU program would expect a response that includes an anticipated value derived from stimuli data, as processed by the SDAPU that directly corresponds to a motor being turned on. After start up, the motor operating stimuli values would naturally change in response to the dynamic conditions, along with the sensor input, as it adapts to the motor's new current "run" state, thus indicating a different conditional state. In an environment with simultaneous processes occurring, the sum of these states coupled with a value, derived from a quantitative assessment of each asset, would be used in processing, developing, and assigning a quantitative score (also referred to as a signature) to be used by the SDAPU for processing and ultimately determining the anticipated state of the environment. Table 10 shows a sample of production assets along with their conditional states of motion or rest where stimuli values were assigned and subsequently programmed into the SDAPU. Since this research was designed using a linear process model, it was not necessary to calculate multiple process values.

The independency between the CI2CS network and sensory network was critical for the purpose of this research as it is the fundamental premise to this argument: an actual state of either the object(s), space(s), or both, sensory data, is evaluated against an expected or supposed state of the CI2CS performance, as a comparable similarity should be observed between the two; should those established states be calculated (aligned) correctly between one another, if not; a dissimilarity should be observed. In the most basic sense, the I/O data of a

Table 9. Sample Stimuli Assignment Values

						Stimuli													
			Hear				See			Touch						Smell	Taste		Value
Production/Operational Asset	States	Code	Sound	Pitch	Volume	Thermal	Color	Brightness	Motion	Matter	Texture	Flexibility	Pressure	Weight	Temperature	Vapors	pH	Sample Response	Quantitative
Robotic Arm One	Rotate Left	RORL	x	x	x	x			x						x				6
Robotic Arm One	Rotate Right	RORR	x	x	x	x			x						x				6
Robotic Arm One	Shoulder Raise	ROSR	x	x	x	x			x						x				6
Robotic Arm One	Shoulder Lower	ROSL	x	x	x	x			x						x				6
Robotic Arm One	Elbow Raise	ROER	x	x	x	x			x						x				6
Robotic Arm One	Elbow Lower	ROEL	x	x	x	x			x						x				6
Robotic Arm One	Gripper Close	ROGC	x	x	x	x			x						x				6
Robotic Arm One	Gripper Open	ROGO	x	x	x	x			x						x				6
TankA Fluid	Fill	TAFF	x	x	x	x			x	x	x		x	x	x		x		11
TankA Fluid	Empty	TAFE	x	x	x	x			x	x	x		x	x	x		x		11
TankA Fluid	Maintain Volume	TAFV			x	x				x	x		x	x	x				7
TankB Fluid	Fill	TTFF	x	x	x	x			x	x	x		x	x	x		x		11
TankB Fluid	Empty	TTFE	x	x	x	x			x	x	x		x	x	x		x		11
TankB Fluid	Maintain Volume	TTFV			x	x				x	x		x	x	x				7
TankP Fluid	Empty	TTFE	x	x	x	x			x	x	x		x	x	x		x		11
TankP Fluid	Fill	TTFF	x	x	x	x			x	x	x		x	x	x		x		11
TankP Fluid	Maintain Volume	TPFV			x	x				x	x		x	x	x				7
Centrifuge Loaded	At Set Run Speed	CLASRS	x	x	x	x			x						x				6
Centrifuge Loaded	At Other Run Speed	CLAORS	x	x	x	x			x						x				6

CI2CS should agree with the stimuli it created in its environment. As an example, when a signal occurs that switches on a motor, that information or signal verification is sent to the SSM while the dynamic state of the spatial environment is simultaneously processed. There should be a correlation corresponding to the event that when a motor is turned on, stimuli such as sound,

Table 10. Example of Conditional States of Motion or Rest

Conditional States of Motion OR Rest (NULL)												
Production/Operational Asset	Conditions	Code (Variable)	One	Two	Three	Four	Five	Six	Seven	Eight	Nine	Cases
Motor	Energized	MotE										
Motor	Acceleration	MotA	MotAM									
Generator	Voltage	GenV										
TankA Fluid	Fill	TOFFF	TOFFF									
TankA Fluid	Empty	TOFE	TOFE									
TankA Fluid	Maintain Volume	TOFMP	TOFMP									
TankB Fluid	Fill	TTFF	TTFF									
TankB Fluid	Empty	TTFE	TTFE									
TankB Fluid	Maintain Volume	TTFE	TTFE									
TankP Fluid	Fill	TTFF	TTFF									
TankP Fluid	Temperature	TTFTT	TTFTT									
Robotic Arm One	Rotate Left	RAO RL	RAO RL+RAOSR	RAO RL+RAOSL	RAO RL+RAOER	RAO RL+RAOEL	RAO RL+RAOWR	RAO RL+RAOWL	RAO RL+RAOGC	RAO RL+RAOGO	NULL	RAO - Case ROT + Conditional State (one to many)
Robotic Arm One	Rotate Right	RAO RR	RAO RR+RAOSR	RAO RR+RAOSL	RAO RR+RAOER	RAO RR+RAOEL	RAO RR+RAOWR	RAO RR+RAOWL	RAO RR+RAOGC	RAO RR+RAOGO	NULL	
Robotic Arm One	Shoulder Raise	RAO SR	RAO SR+RAOER	RAO SR+RAOEL	RAO SR+RAOWR	RAO SR+RAOWL	RAO SR+RAOGC	RAO SR+RAOGO			NULL	RAO - Case SHO + Conditional State (one to many)
Robotic Arm One	Shoulder Lower	RAO SL	RAO SL+RAOER	RAO SL+RAOEL	RAO SL+RAOWR	RAO SL+RAOWL	RAO SL+RAOGC	RAO SL+RAOGO			NULL	
Robotic Arm One	Elbow Raise	RAO ER	RAO ER+RAOWR	RAO ER+RAOWL	RAO ER+RAOGC	RAO ER+RAOGO					NULL	RAO - Case ELB + Conditional State (one to many)
Robotic Arm One	Elbow Lower	RAO EL	RAO EL+RAOWR	RAO EL+RAOWL	RAO EL+RAOGC	RAO EL+RAOGO					NULL	
Robotic Arm One	Gripper Close	RAO GC	RAO GC								NULL	RAO - Case GRI + Conditional State (one to many)
Robotic Arm One	Gripper Open	RAO GO	RAO GO								NULL	
Centrifuge	Loaded at Set Run Speed	CLASRS	CLASRS									
Centrifuge	Loaded at Other Run Speed	CLAORS	CLAORS									

vibrations, and heat (for this particular example) are generated and recognized as legitimate attributes.

This research could be applied in an actual CI2CS environment with real system performance being legitimately validated by this design using an artificial external condition in lieu of introducing an actual virus into the system. An example would be tapping a metal object on a motor housing during operation, start-up or shut-down, or any other manually induced condition that could simulate an effect on a component caused as a result of a vulnerability being introduced into the system and producing the same or similar outcome. Essentially, anything out of the ordinary occurring within that environment should be cause for alarm, if the performance of its current state does not match its established baseline.

Formats for Presenting Results

Study results have been presented in several ways. Most of the results are recorded in tables or figures. The figures were either made up of screen shots or created using programs such as Google Sketch Up, Visio, or Adobe Photoshop. Screen shots from test instrumentation were produced from data displayed in that specific tool's native view, such as that from the Textronix MDO 3014 scope or Fluke 289 meter, and then transferred to a PC compliant application. National Instruments LabView Signal Express provides its own form of data rendering and it was captured via screen shots. Some SDAPU results were rendered using other non-native applications, in order to create a visual representation in graph form, of the sensory signals that were collected and analyzed with the aid of the MDO 3014 software. In particular, this included noise captured by the sound sensor/recorder. Results are also documented with photographs, where necessary, for capturing an asset's behavior in the production section of the test-bed environment at various points in time during its course of operation.

Resource requirements

Similar to Giani et. Al. (2008), this research incorporated a PLC, field devices, three laptops, one notebook, HMI and Cisco 5505 firewall as hardware, along with the list of other items included in Table 5. No other instrumentation was used for the purpose of this research. A complete list of resources is listed in with Tables 4, 5, 6 and 8 as introduced on page 84.

The test-bed with all of its fabrication, control, cyber, testing, and production components was a privately funded research endeavor. All of the aforementioned resources leading up to and throughout the course of the research were funded by the researcher. No grants or outside funding was received. Rockwell extended a ten percent academic discount for the three Allen-Bradley components through their local distributor, McNaughton-McKay. Some of the software applications previously mentioned were either purchased (at full or academic pricing), used during trial periods, or obtained as open source from the internet. Kepware granted a one year extension to their typical 45-day trial period for the use of their Kepware Server application. This research did not include any human participants as part of the study; therefore, no IRB approval was requested.

Summary

A control system performs real-time automated processes in real world situations and, therefore, is reliability-dependent during such time those automated processes and functions are being performed. An interruption or interference to such a process could, at the most, depending on its nature, be life- threatening. Because CI2CSs control real world automated processes and functions, it is imperative to prevent disturbances, deliberate or accidental, to volatile and sensitive control systems. One way of demonstrating changes to a CI2CS, without harming or damaging the system, is by developing a test-bed and then applying modifications and new implementations within the test-bed all the while studying the impact, if any, caused as a result of introducing those changes.

This research established a viable and legitimate test-bed that was used for just such a study by including assets found in similar test-beds used for research and performance analysis of control systems and the production processes which they control. The test-bed emulated that of a chemical manufacturing facility which is recognized as a critical sector to our national infrastructure. This test-bed provided a sufficient setting for the observation and monitoring of the production environment during its operational phase, so that adequate tuning, evaluation, testing and analysis could be conducted. The test-bed developed for this experiment was used for designing and verifying the "concept"; while real time control system data (logs and network activity) and corresponding sensory data could be evaluated and analyzed for determining potential risk, during off-line processing, in an attempt to validate, document, and demonstrate the results from an actual environment.

Chapter 4

Results

Introduction

This chapter provides an objective description and analysis of the findings with the results and outcomes from the research. Here, research contributions in the area of industrial control systems from an OT perspective, coupled with that of the real-time processing capabilities of “big data”, using data analytic platforms on the IT side of the business to assess risk are described in great detail. Lastly, any limitations and constraints encountered throughout the research are stated with recommendations provided for future studies.

Benefits of Continuous Real-time Independent Risk Monitoring of CI2CS

The dependency modern society has placed on electricity as the enabler of greater innovation and achievement for improving the progress of humankind has led to stringent measures in electric reliability oversight and importance. This has mostly resulted from the efficiencies and enhancements designed into information technology over the past 15 years that have contributed to providing a significant convenience factor for CI2CS implementers. However, this convenience factor has exposed these entire system(s) to an overall far greater security risk.

An unfortunate side effect of this convenience factor and increase in security risk lies in the inevitable and routine discovery of the sometimes-overwhelming number of vulnerabilities that exist, especially as some of these technologies mature, both in the exploitation capabilities and the manner in which the vulnerability can be exposed. These vulnerabilities have led to the need for continuous risk monitoring of not just the cyber or IT assets interconnected with a control system, but also of the automation and OT system's field devices they must interface with. This includes any type of automation devices having embedded controllers with programmability features in addition to programmatic dependent devices designed and used exclusively as master controllers such as a PLC or DCS.

Continuous risk monitoring in operational technology has traditionally existed for the benefit of the processes being performed and the equipment performing the process. For instance, an electric motor might have temperature and vibration sensors for the purpose of monitoring its performance throughout the process; however, that monitoring exists for the sole purpose of discovering and potentially determining a production mal-function and or establishing maintenance intervals for servicing. Alarms received from this type of scenario are only provided to alert the operator or control technician of a failure in the equipment which would subsequently suggest an immediate or pending interruption to that part of the production cycle; ultimately causing a cascade effect either upstream, downstream, or both, from that point in the process. Thus, there is a need for research to further develop the options relating to continuous real-time independent risk monitoring of CI2Cs.

Data Analysis

In keeping with the research design, the data analysis is covered in three sections. The first of these three sections describe the analysis of the test-bed development and the designing, creating, and establishing of an *initial* operational baseline of a CI2CS process from data gathered during the post-installation phase of each test-bed component. In this section, each of the four tasks in the process is discussed and data analyzed. Baselining in and of itself is a continuous process and is also discussed in the next section.

The next section reviews sensor selection, placement, and analysis of data for defining normalcy. Data analysis in this section, is derived after all the components of the system are installed and functioning properly, and a measure of normalcy is observed relative to the system's initial baseline, so that an *operational* baseline can be developed for establishing the performance standard that all subsequent production cycles will be evaluated and compared to for determining risk. Section three, discusses the introduction of the exploitation of the Arduino controller that causes the robotic arm to start behaving erratically and demonstrates the validity and success of this research by illustrating how that behavior is distinguished and recognized from that of the previously established operational baseline covered in the prior section. Findings and summary follow the data analysis.

Section One – Analysis of Test-bed Development and Initial Baselining

Data collection for this research took place over a three-month period. No set of experiments ever exceeded an eight-hour interval. This means that no more than 32 back-to-cycles were ever conducted in one sitting. This was not an intentional or deliberate effort, but arose more likely from traditional labor practices. During this period of time, none of the components ever failed nor was their performance noticeably degraded.

Although there were some climate variations over this period of time, none of the variations were extreme enough to alter the environment to an extent that there was any meaningful impact on the data. For example, the temperatures within the test-bed were never so low or high that it affected the sound (acoustic) acquisition of the operating processes. Air molecules are larger in cold air and thus have an impact on the speed at which sound travels. The colder the temperature, the slower sound travels, vice-versa with hotter temperatures. Essentially, the overall environmental conditions remained relatively constant.

Initially, data collection and analysis during the initial phase was intermittent and of irregular quality, as the components used in performing specific system processes were installed; activated, then aligned with the corresponding step that followed. Any issues were resolved early in the data gathering process by making adjustments as needed depending on the issue. With each complete production cycle completing in 14 minutes and one second, it was possible to run four complete cycles in an hour. However, more time was needed between each cycle for documenting and organizing the data captured throughout various processes. Early on during this phase, while making initial observations of the production cycle in operation, it was not atypical to spend from ten to fifteen minutes or even an hour documenting, analyzing, and interpreting the results obtained and recorded by various instruments and sensors, as well as, troubleshooting a number of performance characteristics directly affected by poor coding (programming). This was particularly noted in values obtained by the flow rate and humidity/temperature sensor, as it related to units of measures and their conversions, such as liters and Celsius, respectively, which the results obtained during their operation clearly showed.

Inconsistency in data presentation being represented consistently throughout the entire process was resolved early on in the data gathering phase. In most, if not all cases, be it due to a miscalculation in a unit of measure or garbage data in general, replacing a sensor, reviewing the code, or electrical grounding resolved the data inconsistency or anomaly, and eliminated certain sporadic equipment behavior during subsequent operations.

As described in the previous chapter, the test-bed imitates a chemical process facility that produces a *simulated* toxic compound. By virtue of its toxic nature, the production process is entirely automated. Although the process was described earlier in chapter three, the key process tasks will be analyzed in greater detail in this section in conjunction with the outcome of the research and the findings.

The simulated CI2CS chemical processing facility consists of four very specific and distinct tasks that must be performed in a precise linear fashion in order to manufacture the simulated chemical solution. As discussed in chapter 3, the four tasks are: (1) energizing the motor, (2) batch processing (which included four subtasks), (3) energizing the robotic arm and (4) energizing the centrifuge. It is essential that each task is completed in its proper sequence and form. Any deviation from these process tasks is reason for investigation and/or evaluation of its cause and would naturally be discovered during the manually initiated sensory comparison risk scan. The completion of each of these four tasks fulfill one complete process cycle. The complete cycle takes 14 minutes and one second to complete.

Task One - Energizing the Motor Control

The production cycle was initiated by manually interacting with a virtual (soft) button on a touch screen of a HMI or by pressing a physical, normally open button on a controller box, or by interacting with the Connected Components Workbench interface from the engineer's laptop. Any one of the three methods could start the production cycle through the PLC which would trigger the Timer on (TON) delay function, written into the ladder logic on rung one, Figure 7,

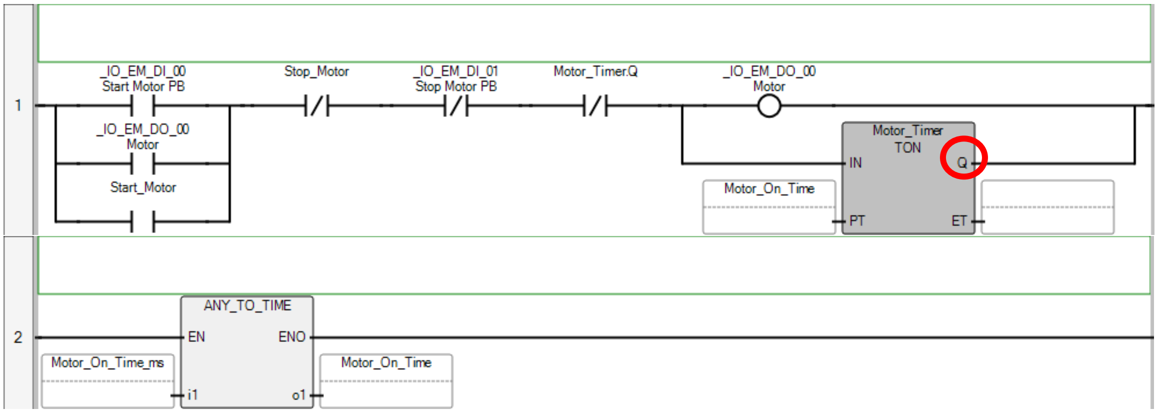
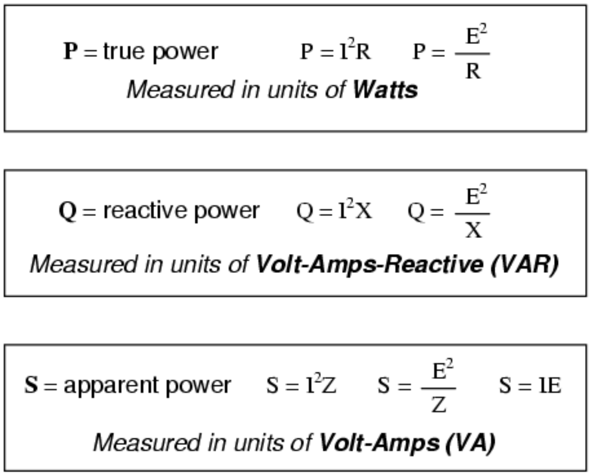


Figure 7. Controller.Micro850.CI2CS_circuit showing TON function (rung one) and variable conversion block (rung two)

that sent a signal to the110VAC motor controller which in turn energized a three phase, .5 hp, AC inductive motor, for the purpose of imitating a sustainable fuel source such as wind or hydro driven turbine, that turns the shaft of the SEIG. This is a 30 second process that is designed to imitate a mini-grid and could be used for powering the CI2CS test-bed facility.



P = true power $P = I^2R$ $P = \frac{E^2}{R}$

*Measured in units of **Watts***

Q = reactive power $Q = I^2X$ $Q = \frac{E^2}{X}$

*Measured in units of **Volt-Amps-Reactive (VAR)***

S = apparent power $S = I^2Z$ $S = \frac{E^2}{Z}$ $S = IE$

*Measured in units of **Volt-Amps (VA)***

Figure 8. Formulas for determining true, reactive, and apparent power (Kuphaldt, 2015)

Upon starting the electrical motor, the generator voltage increased within ten seconds from a floating .0289 VAC to 327.5 VAC and sustained a 327 VAC average for a twenty-second-time period. This can be seen starting at line 324 to 360 in Table 11, SEIG VAC data. The voltage then decreased as the motor wound down over a ten second interval. Although the *apparent power* (**S)**, if using the equation in Figure 8, indicated to average 327 VAC, the *true power* (**P**) proved to never get higher than 36 VAC of constant power; relative to the various loads. According to Kuphaldt (2015), relative to the resistance (**R**) of the applied load during various tests, **P** was determined by using the formula in the equations shown in Figure 9.

In determining **P**, Kuphaldt (2015) states that "as a rule, true power is a function of a circuit' dissipative elements, usually resistances." In the following formulas: **I** = amps; **R** = resistance; **X** is a circuit's resistance as the result of function of reactive power (**Q**); and, **Z** is a

Table 11. SEIG VAC data

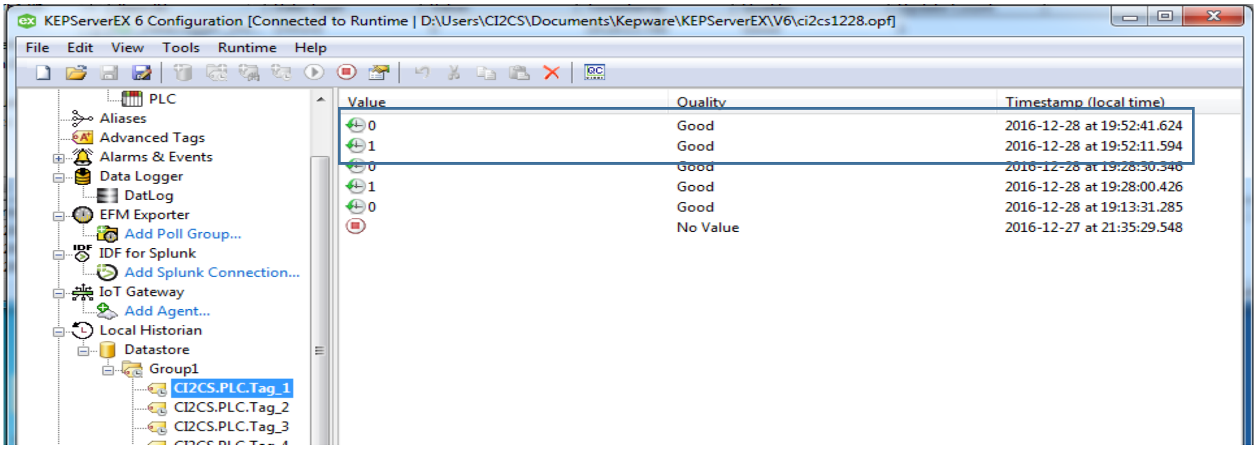
Line No.	Sample	Start Time	Duration	Max Time	Max	Average	Min Time	Min	Description	Stop Time
323	0.0289 V AC	23:18.8	00:00.9	23:19.6	0.0937 V AC	0.0461 V AC	23:18.8	0.0289 V AC	Unstable	23:19.7
324	0.1505 V AC	23:19.7	00:00.1	23:19.7	0.1505 V AC	0.1505 V AC	23:19.7	0.1505 V AC	Interval	23:19.8
325	0.1638 V AC	23:19.8	00:00.9	23:20.7	0.3761 V AC	0.2699 V AC	23:19.8	0.1638 V AC	Unstable	23:20.8
326	0.4032 V AC	23:20.8	00:00.1	23:20.8	0.4032 V AC	0.4032 V AC	23:20.8	0.4032 V AC	Interval	23:20.9
327	0.4295 V AC	23:20.9	00:00.9	23:21.7	0.6826 V AC	0.5522 V AC	23:20.9	0.4295 V AC	Unstable	23:21.8
335	5.0509 V AC	23:24.9	00:00.1	23:24.9	5.0509 V AC	5.0509 V AC	23:24.9	5.0509 V AC	Stable	23:25.0
336	OL V AC	23:25.0	00:01.7	23:25.0	OL V AC	V AC	23:25.0	OL V AC	Interval	23:26.6
337	OL V AC	23:26.6	00:01.7	23:26.6	OL V AC	V AC	23:26.6	OL V AC	Interval	23:28.4
338	292.99 V AC	23:28.4	00:00.4	23:28.7	305.64 V AC	299.14 V AC	23:28.4	292.99 V AC	Unstable	23:28.8
339	309.97 V AC	23:28.8	00:00.1	23:28.8	309.97 V AC	309.97 V AC	23:28.8	309.97 V AC	Interval	23:28.9
340	314.71 V AC	23:28.9	00:01.0	23:29.6	326.59 V AC	323.58 V AC	23:28.9	314.71 V AC	Interval	23:29.9
341	326.01 V AC	23:29.9	00:01.0	23:30.7	327.45 V AC	326.61 V AC	23:29.9	326.01 V AC	Interval	23:30.9
342	326.31 V AC	23:30.9	00:01.0	23:31.8	327.52 V AC	326.78 V AC	23:30.9	326.31 V AC	Interval	23:31.9
343	327.00 V AC	23:31.9	00:01.0	23:32.4	327.49 V AC	326.76 V AC	23:32.0	326.43 V AC	Interval	23:32.9
344	327.51 V AC	23:32.9	00:01.0	23:33.5	327.58 V AC	326.88 V AC	23:33.7	326.44 V AC	Interval	23:33.9
357	327.61 V AC	23:45.9	00:01.0	23:45.9	327.61 V AC	326.89 V AC	23:46.1	326.49 V AC	Interval	23:46.9
358	326.99 V AC	23:46.9	00:01.0	23:47.0	327.61 V AC	326.98 V AC	23:47.2	326.51 V AC	Interval	23:47.9
359	326.61 V AC	23:47.9	00:01.0	23:48.1	327.61 V AC	326.97 V AC	23:48.3	326.52 V AC	Interval	23:48.9
360	326.55 V AC	23:48.9	00:00.6	23:49.0	326.63 V AC	323.48 V AC	23:49.4	316.18 V AC	Unstable	23:49.5
361	312.11 V AC	23:49.5	00:00.4	23:49.5	312.11 V AC	305.88 V AC	23:49.8	299.93 V AC	Interval	23:49.9

Line No.	Sample	Start Time	Duration	Max Time	Max	Average	Min Time	Min	Description	Stop Time
362	295.37 V AC	23:49.9	00:00.9	23:49.9	295.37 V AC	275.56 V AC	23:50.7	255.95 V AC	Unstable	23:50.8
363	250.16 V AC	23:50.8	00:00.1	23:50.8	250.16 V AC	250.16 V AC	23:50.8	250.16 V AC	Interval	23:50.9
366	193.27 V AC	23:51.9	00:00.8	23:51.9	193.27 V AC	174.56 V AC	23:52.6	155.59 V AC	Unstable	23:52.7
367	150.03 V AC	23:52.7	00:00.2	23:52.7	150.03 V AC	147.22 V AC	23:52.8	144.40 V AC	Interval	23:52.9
368	138.56 V AC	23:52.9	00:00.9	23:52.9	138.56 V AC	112.01 V AC	23:53.7	82.81 V AC	Unstable	23:53.8
369	74.24 V AC	23:53.8	00:00.1	23:53.8	74.24 V AC	74.24 V AC	23:53.8	74.24 V AC	Interval	23:53.9
370	65.46 V AC	23:53.9	00:00.2	23:53.9	65.46 V AC	61.08 V AC	23:54.0	56.70 V AC	Unstable	23:54.1
371	48.22 V AC	23:54.1	00:00.1	23:54.1	48.22 V AC	48.22 V AC	23:54.1	48.22 V AC	Stable	23:54.2
382	0.0308 V AC	24:01.9	00:01.0	24:02.0	0.0312 V AC	0.0297 V AC	24:02.3	0.0289 V AC	Interval	24:02.9
383	0.0295 V AC	24:02.9	00:01.0	24:03.1	0.0302 V AC	0.0296 V AC	24:03.3	0.0289 V AC	Interval	24:03.9
384	0.0294 V AC	24:03.9	00:01.0	24:03.9	0.0294 V AC	0.0294 V AC	24:04.2	0.0293 V AC	Interval	24:04.9

circuit's total impedance as the result of a function of apparent power (Kuphaldt, 2015). **R** varied depending on the type of load that was applied.

The various loads tested and considered for performing some functional application as part of the research design included the following: 60 and 100-watt incandescent bulbs; Weller 100/140-watt soldering gun; Eastman 240V, 1500-watt water heating element; and, Wellman flat 120V, 1000 Watt, strip heater. Other than for the soldering gun, load measurements were extremely difficult to obtain due to the voltage instability caused as a result of power transference from the grid to the loads without some type of voltage regulator, transformer, or both. The power distribution aspect of the SEIG does not directly apply to the research problem being studied or the problem being solved. Stimuli coding for step one of the CI2CS process was (S;V;vAC). Temperature (T) was excluded in this process, as step one only lasted 30 seconds and ambient temperatures within the test-bed never exceeded a range of ± two degrees during any cycle. A two-degree shift in either direction from the normal ambient temperature was not compelling enough to warrant monitoring or logging temperature sensor data on either the electric motor (primary mover) or SEIG for the short interval in which it operated, especially without the application of a dedicated and continuous load. For the sake of electric reliability, scalability and safety, the SEIG did not power any of the production or process control devices while production cycles were performed and its use was solely as an additional component for enriching the environment with additional stimuli and providing variables to monitor as part of a holistic approach in considering rudimentary processing components typically included as part of this type of facility.

Table 11 illustrates the voltages generated between the motor start and stop times, as well as, several pre and post samples showing the voltage readings before and after the motor operation. “Motor” operation instead of “SEIG” operation is referenced here, so that it is recognized, although, the SEIG in fact produced the voltage output shown in Table 11 beginning with line 324 and ending with line 360, the motor’s function related directly to the performance of the generator during the times highlighted in blue and red in the table. However, the SEIG alone continued putting out high voltage even after the motor was decelerating. Several of the table line entries were removed between transition points as they were duplicative or substantially similar to other lines in that sequence in order to consolidate the information in the table. The brown highlighted color represents voltages less than 74.24VAC. The yellow highlight represents voltages above 74.24VAC and especially over 110VAC, the common voltage (110-120VAC) for residential distribution since the bulk of electrical appliances require it, although larger appliances, such as dryers and heating, ventilation and air condition (HVAC) components, operate on 220VAC. The 220VAC capability is accomplished by provisioning two 110VAC service lines to the demarcation point or “meter”. Shades of red, lines 341 through 360, represent an average of the maximum voltage sustaining between 326.01 and 326.55 VAC before which point the motor began decelerating, thus resulting in a decrease of voltage output by the generator.



Value	Quality	Timestamp (local time)
0	Good	2016-12-28 at 19:52:41.624
1	Good	2016-12-28 at 19:52:11.594
0	Good	2016-12-28 at 19:28:30.346
1	Good	2016-12-28 at 19:28:00.426
0	Good	2016-12-28 at 19:13:31.285
	No Value	2016-12-27 at 21:35:29.548

Figure 9. Tag 1 Motor Start/Stop Data, 00:00:30

Figure 9 above is a screenshot of the motor process start and stop time as displayed by the Kepware Server software used to communicate and interact with the PLC using OPC. This software was described in more detail in chapter three. Figure 9 shows the motor start time at 19:52:11.594 and end time at 19:52:41.624. The total time indicated here is 30 seconds. The data resolution for the timestamp showing milliseconds is constrained by the software since data logging of milliseconds is restricted to three places.

In Figure 10 below, a graph shows a visual interpretation of the amplitude (sound) readings obtained during a typical SEIG operation where the motor controller begins gradually increasing the frequency of the motor, until it reaches a maximum of 60 Hz, all the while increasing the voltage output being generated by the SEIG as indicated in Table 11.

The first notable state change can be observed by the increase in voltage (amplitude) from between -5 and 5 millivolts (mv) to between -20 and 20 mv as the ambient noise is joined, or replaced, with the sound genereated by the motor controller's fan, initiated by the output signal from the PLC, and the frequency output to the motor, as the motor RPMs begin to increase. This is observed at the onset of the motor operation and can be seen right before the 19:47:40 mark on the x-axis of the graph with the amplitude, represented as voltage (V), appearing just below the 20 mv vertical-axis line before it.

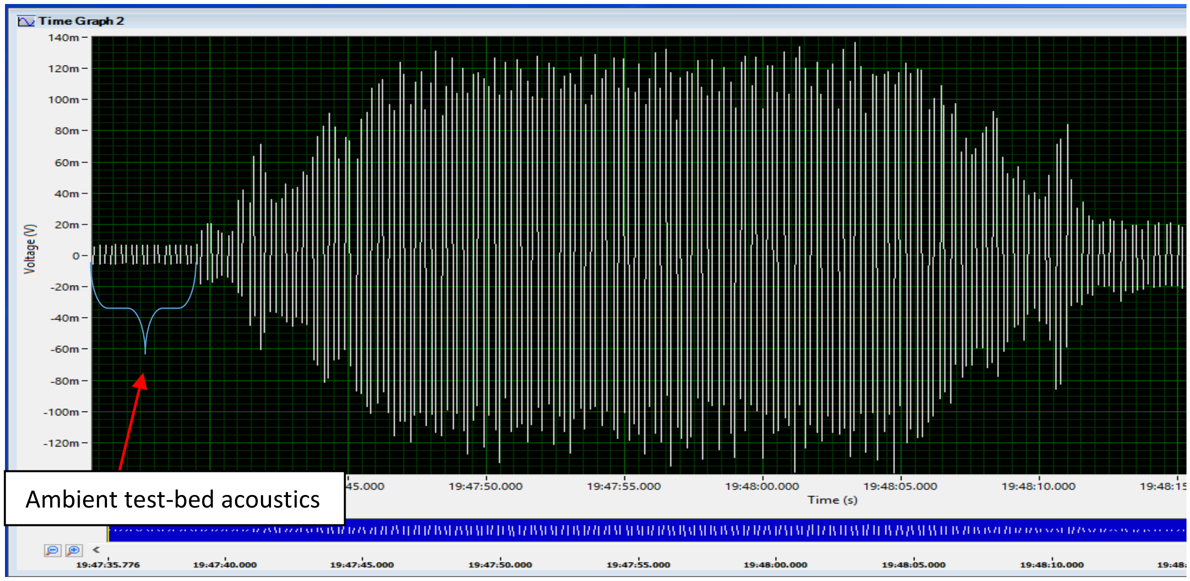


Figure 10. Motor sound data (Amplitude Readings of Motor Sounds)

This graph in Figure 10 was derived from the raw data logged during one of the many operating intervals. As such, the time line in this example was created by rendering the raw data into a visual representation, so that this particular part of the process could be interpreted more easily. Each process has its own vibration and sound graph as represented by the data obtained during the particular interval it was logged. A graphic, illustrating motor vibrations, at the far left of the image, can be seen in Figure 34 later in this chapter.

The millivolt decrease beginning right past 19:48:05 shows where the motor controller frequency is decreasing down to 0 Hz, and at 19:48:10 the motor controller signal from the PLC output has been terminated; thereby, de-energizing the motor. There is a trace of motor rotation occurring that exists from the residual momentum produced as a result of the generator winding down, along with the motor controller's fan, and the start of PumpPA that causes some moderate voltage activity for two seconds between time markers 19:48:10 and 19:48:12, peaking between -85 mv and 85 mv. The sound levels begin to smooth out at approximately

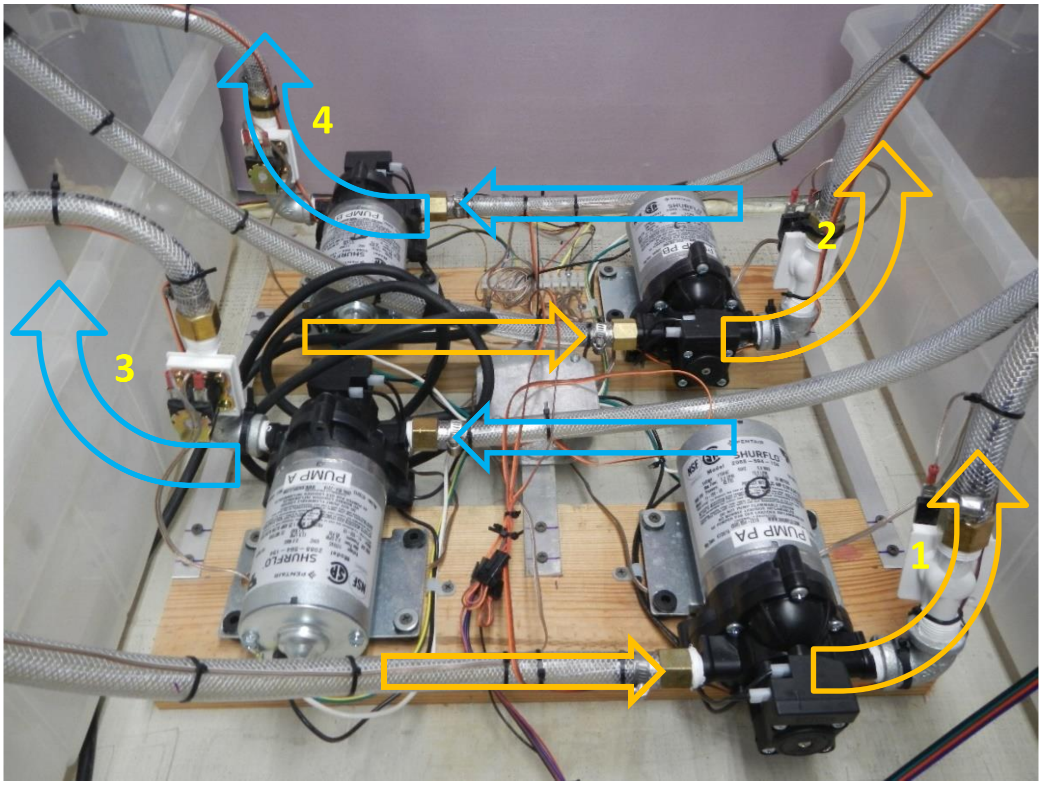


Figure 11. Pumping System (front-right PumpPA, front-left PumpA, rear-right PumpPB, rear- left PumpB)

19:48:13.500 for the remainder of PumpPA's operation as shown during the transition depicted in the graph in Figure 11.

Task Two- Batch Processes

Task Two is the batch process. Figure 11 above illustrates the flow process sequence for the making of one batch of product per production cycle. PumpPA began the batch process by partially filling product Tank C. It was then followed by PumpPB directly behind it. Once the product tank had been filled, PumpA, to the left of PumpPA in Figure 11, begins back-filling Tank A so that it can re-supply the simulated chemical solution (water) in preparation for the next batch to begin after the entire production cycle has been completed. PumpB follows suit after PumpA.

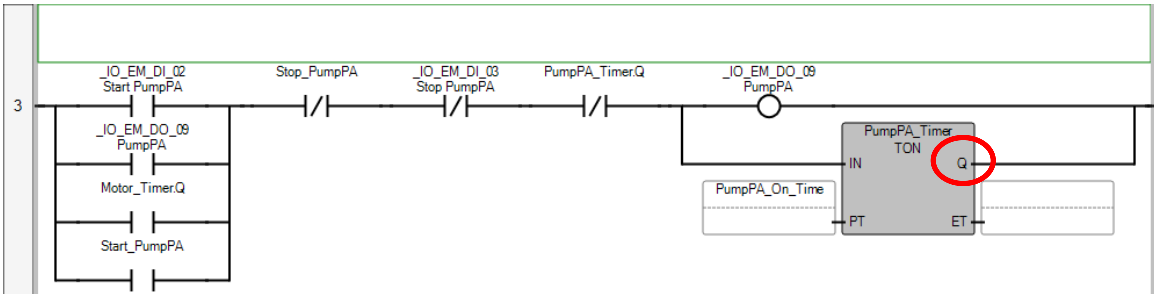


Figure 12. PumpPA ladder logic instruction started the "pumping" cycle

A complete batch means that PumpPA and PumpPB had expelled the contents from tank A and B, respectively, before returning the contents from the product tank back to tanks A and B. The term batches refer to the completion of one complete pumping process between the tanks. Rungs three through six, Figures 12 and 13, show the ladder logic used for sequencing the batch process.

On rung one of the PLC's motor timer TON function, Figure 7, an elapsed **Q** output, circled in red and similar to Figure 12, parameter triggered PumpPA that started the batch process for manufacturing the simulated chemical mixture. PumpPA along with the other three pumps used the PLC's **Q** parameter of the TON function within the ladder logic programming, set with specific times, for filling the product tank, tank C, and back-filling the "supply" tanks, tanks A and B. The pump run times were based on an average of the sampling results for fill times of five complete product batches as seen in Table 12.

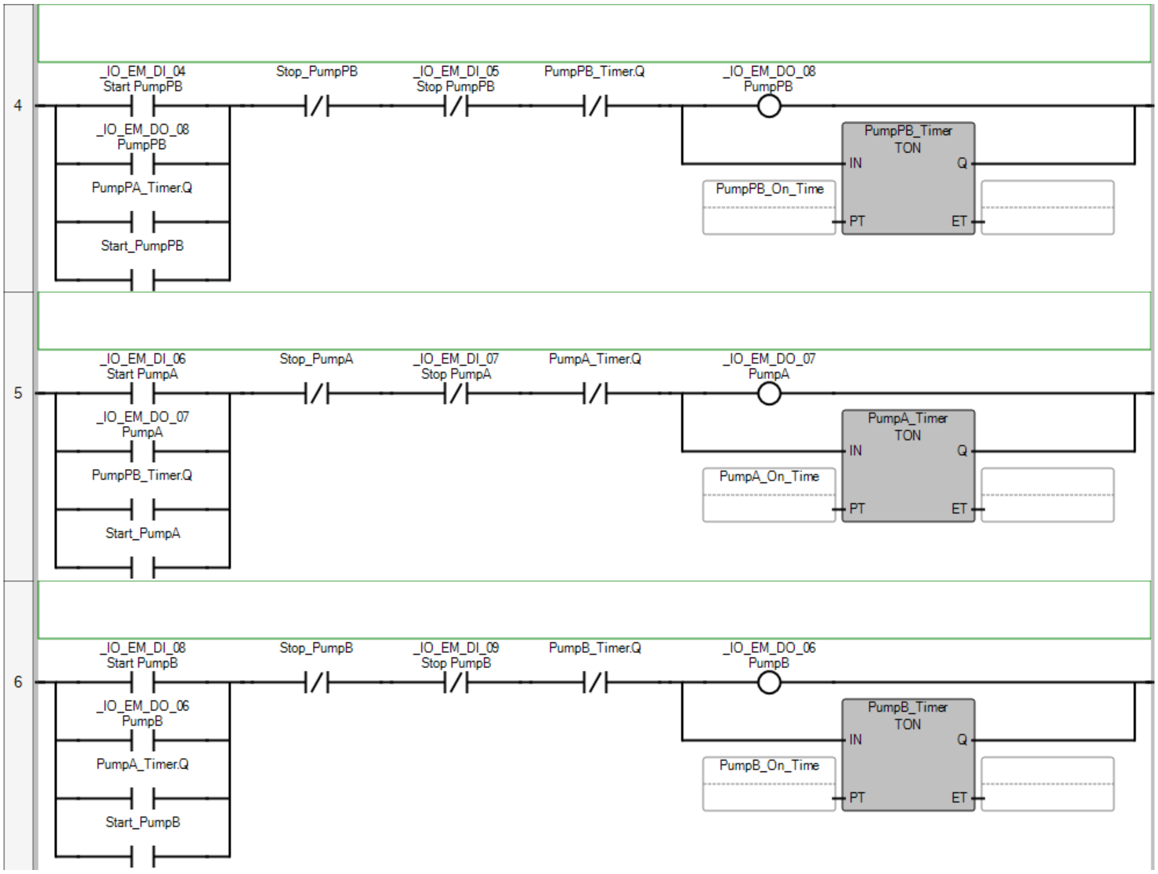


Figure 13. Pumps PB, A, and B ladder logic shown representing sequence of pumping process

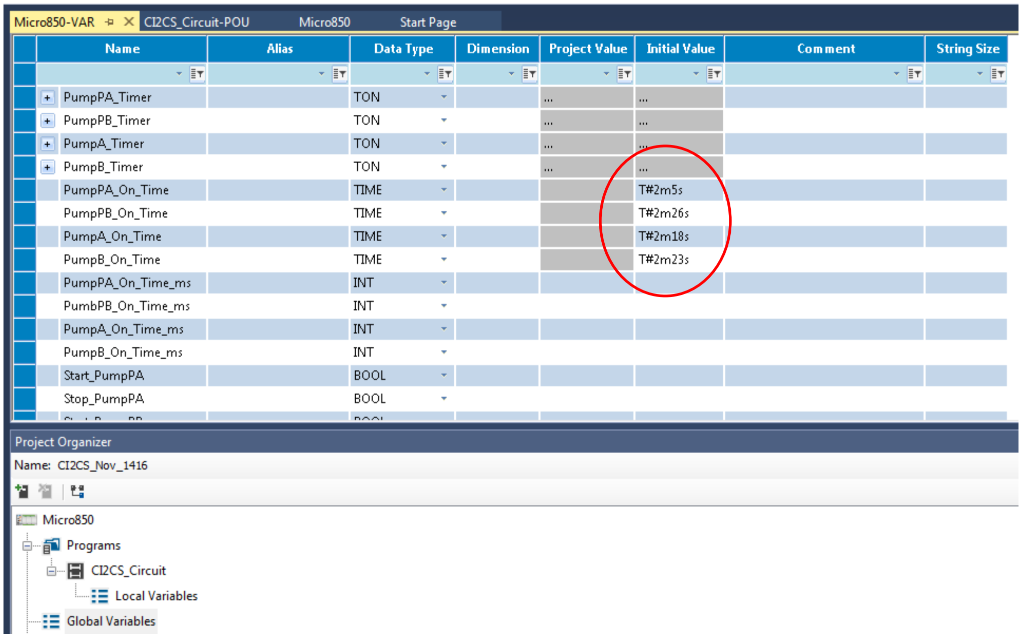
The batch process for making product was task two of the manufacturing process and essentially consisted of four sub-tasks within the overall four task production cycle. "Product" refers to the solution created as a result of combining TankA and TankB into the "Product" Tank. Table 12 identifies the pump and its average transfer rate from its respectable source tank and into its target tank. Fill times were determined after observing and recording the results from five separately timed iterations of consecutive pumping samples.

No sensors were used for obtaining fill times. Instead, the method employed consisted of a stop watch and recognition of fluid reaching a visual indicator (physical line) marked on the tank. This required backfilling the source (supply) tank before another iteration could be observed and recorded. The remainder of the tasks within this cycle finished upon the completion of TankB being back-filled. Stimuli coding for step-two was [S;V;D].

Figure 14 shows some of the global variables used in CCWs CI2CS project. The

Table 12. Average pumping times for batch processing

	PumpPA	PumpPB	PumpA	PumpB	Total
	2:06	2:25	2:19	2:25	09:15
	2:04	2:26	2:18	2:23	09:11
	2:05	2:27	2:18	2:23	09:13
	2:06	2:26	2:17	2:22	09:11
	2:04	2:26	2:18	2:22	09:10
Average	2:05	2:26	2:18	2:23	09:12

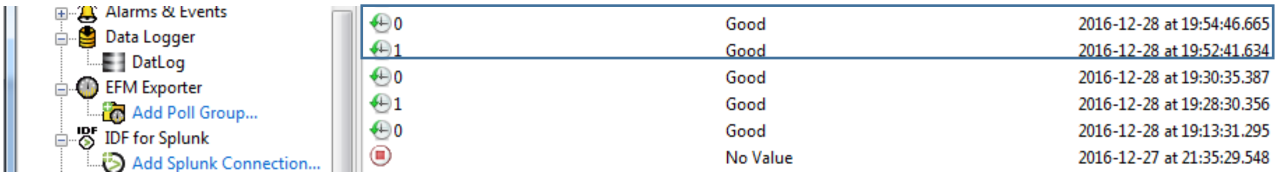


Name	Alias	Data Type	Dimension	Project Value	Initial Value	Comment	String Size
PumpPA_Timer		TON		...	...		
PumpPB_Timer		TON		...	...		
PumpA_Timer		TON		...	...		
PumpB_Timer		TON		...	...		
PumpPA_On_Time		TIME			T#2m5s		
PumpPB_On_Time		TIME			T#2m26s		
PumpA_On_Time		TIME			T#2m18s		
PumpB_On_Time		TIME			T#2m23s		
PumpPA_On_Time_ms		INT					
PumbPB_On_Time_ms		INT					
PumpA_On_Time_ms		INT					
PumpB_On_Time_ms		INT					
Start_PumpPA		BOOL					
Stop_PumpPA		BOOL					

Name: CI2CS_Nov_1416

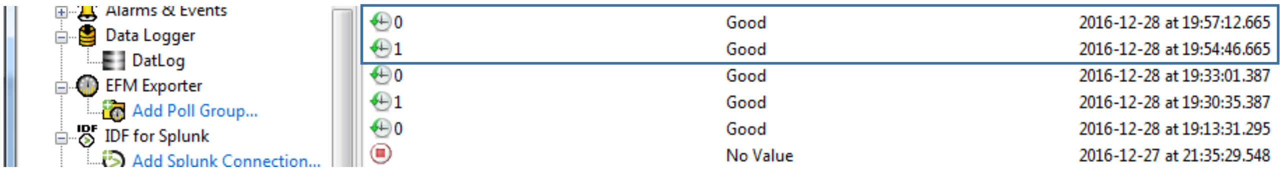
Figure 14. PLC ladder logic time values

variables named: PumpPA_On_Time, PumpPB_On_Time, PumpA_On_Time, and PumpB_On_Time can be seen with their initial values set to the time intervals established from their pumping times determined by averaging and in the prior step. Here the initial values, as seen circled in red, are called (used) when a project's program is initialized. After the program is initialized, other values, known as project values, can and will be used if they exist and are called. In this experiment, by running processes in a linear manner and supervising each production cycle, the pumping times for each cycle instance defaulted to their initial values.



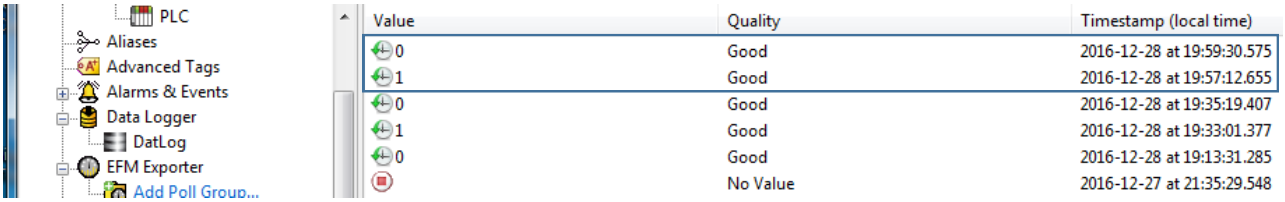
0	Good	2016-12-28 at 19:54:46.665
1	Good	2016-12-28 at 19:52:41.634
0	Good	2016-12-28 at 19:30:35.387
1	Good	2016-12-28 at 19:28:30.356
0	Good	2016-12-28 at 19:13:31.295
	No Value	2016-12-27 at 21:35:29.548

Figure 15. Tag 2 PumpPA Start/Stop Data, 00:02:05



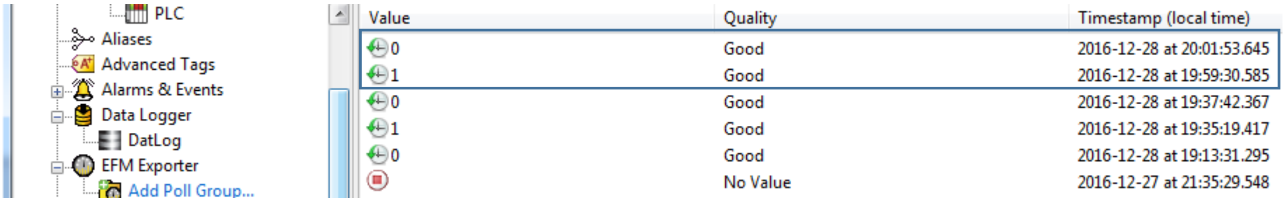
0	Good	2016-12-28 at 19:57:12.665
1	Good	2016-12-28 at 19:54:46.665
0	Good	2016-12-28 at 19:33:01.387
1	Good	2016-12-28 at 19:30:35.387
0	Good	2016-12-28 at 19:13:31.295
	No Value	2016-12-27 at 21:35:29.548

Figure 16. Tag 3 PumpPB Start/Stop Data, 00:02:26



Value	Quality	Timestamp (local time)
0	Good	2016-12-28 at 19:59:30.575
1	Good	2016-12-28 at 19:57:12.655
0	Good	2016-12-28 at 19:35:19.407
1	Good	2016-12-28 at 19:33:01.377
0	Good	2016-12-28 at 19:13:31.285
	No Value	2016-12-27 at 21:35:29.548

Figure 17. Tag 4 PumpA Start/Stop Data, 00:02:18



Value	Quality	Timestamp (local time)
0	Good	2016-12-28 at 20:01:53.645
1	Good	2016-12-28 at 19:59:30.585
0	Good	2016-12-28 at 19:37:42.367
1	Good	2016-12-28 at 19:35:19.417
0	Good	2016-12-28 at 19:13:31.295
	No Value	2016-12-27 at 21:35:29.548

Figure 18. Tag 5 PumpB Start/Stop Data, 00:02:23

Start and stop times for each of the four pumps are shown in Figures 16 through 18. In this instance, PumpPA began at 19:52:41.634 and ended at 19:54:46.665. The operating times for PumpPA and the other pumps all correlate to the times set as part of their initial CI2CS project program values; as close as what the program's data logging capabilities would support.

The timestamp column displays the time consistencies within a fraction of a millisecond. The complete batch process can be seen to take place between 19:52:41.634 and 20:01:53.645.

Task Three-Energizing the Robotic Arm

Once an entire batch process is completed, a robotic arm is energized and begins collecting simulated product samples from the direction of the product tank. The robotic arm process, task three, is triggered by the expiration of PumpB's **Q** parameter setting of the TON function on rung seven of the PLC ladder logic, Figure 19. As described in chapter three, the robotic arm is controlled by an Arduino controller. Although the PLC output to the robotic arm relay is a one second, 12vdc signal sent to "wake" the Arduino, the entire functional process of the robotic arm is controlled by the Arduino. This is important to remember because in section three, it provides the vector that is used to compromise this part of the process while other processes continue to run as normal within their respective order both before and after this task.

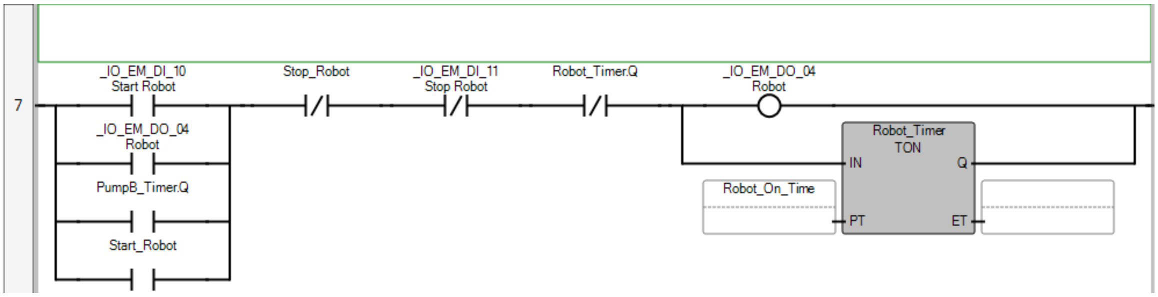


Figure 19. Robotic arm ladder logic

Although the product sample testing is simulated, the robotic arm functions in the exact manner it would, under normal conditions, if it were actually retrieving individual test tube samples from a test tube stand near the product tank before simulating the placement of them into the rotor of the centrifuge. The robotic arm, after waking and moving into an erect state, is positioned at a 90-degree angle to its first functional task of collecting a simulated product sample, Figure 20, as seen indicated by the dark blue arrow. The initialization of the robotic arm, as a result of powering the Arduino controller, places it in a posture and position referred to as the start position or ready-wait state. From there it moves through six complete cycles of simulating the retrieval and placement of six individual test tubes from the product tank and into the rotor before eventually pausing or resting at a 90° angle between the product tank and centrifuge. A complete functional cycle for the robotic arm means that from the ready-wait state, the arm rotates to the 180-degree position from the 90-degree position represented by the green arrow as seen in Figure 24. With the arm remaining in that direction, the elbow and hand (gripper) proceed to go through a series of movements indicating the actions that would be performed if it were to actually retrieve a test tube from a rack located near the product tank. Once that function is complete and a simulated collection has been taken, the arm will rotate 180-degrees from the 180-degree position to the 0-degree position, in alignment with the

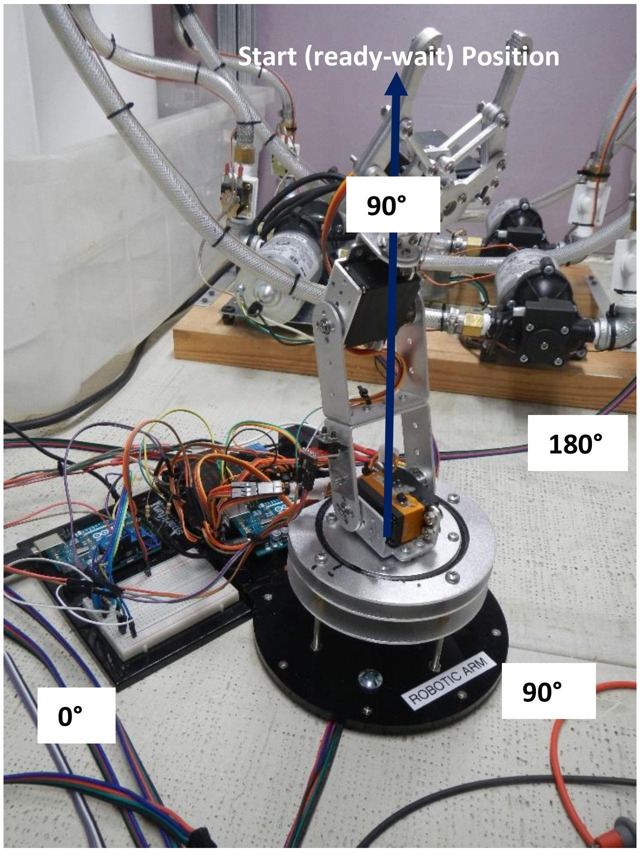
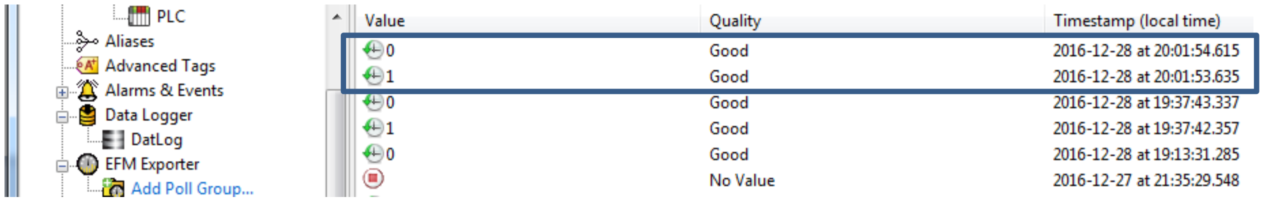


Figure 20. Robotic arm ready position in stand-by (start) state (y-axis avg .224)

centrifuge, and begin performing a similar motion of placing the simulated product sample into the centrifuge, as if actually having the test tubes there to place. This is represented by the yellow arrow in Figure 25.

While discussed in more detail in section three of this chapter, this independent Arduino controller created a vector that was used for introducing the threat into the environment. The threat was recognized by the erratic behavior and obvious deviation from the established baseline as a result of this behavior. Stimuli coding for task three was [S;V;T].

The threat actions ultimately achieved the intended purpose of mimicking the results, similarly produced by Stuxnet during the uranium enrichment process to the centrifuges at Iran's Natanz nuclear plant, albeit without the presence of uranium or an Arduino. Based on an extensive literature survey of publicly available information relating to the Stuxnet virus, it appears that no Arduino was used or involved in the Natanz event. Rather, the affected components were Siemens products.



Value	Quality	Timestamp (local time)
0	Good	2016-12-28 at 20:01:54.615
1	Good	2016-12-28 at 20:01:53.635
0	Good	2016-12-28 at 19:37:43.337
1	Good	2016-12-28 at 19:37:42.357
0	Good	2016-12-28 at 19:13:31.285
	No Value	2016-12-27 at 21:35:29.548

Figure 21. Robotic Arm Start/Stop Data, 00:00:01 (00:02:18 Arduino run-time)

Although Figure 21 shows a one second output signal between the 20:01:53.635 start and 20:01:54.615 stop time, the signal to the Arduino's input pin was used to instantiate the wake function which started running the code within the void setup function of the program. Arduino's IDE software, Sketch, pre-loads with two default functions whenever a new Arduino program is created. The default functions are the void setup and the void loop functions. The robotic arm normal baseline operation process, Figure 22, shows the void setup function containing the instructions that put the arm into the ready-wait state. The code within the void loop function, Figure 23, contained the commands for carrying out the robotic arm’s task specific process.

```
void setup() {
  pinMode(wakePin, INPUT_PULLUP);
  pinMode(led, OUTPUT);
  pinMode(centPin, OUTPUT);
  attachInterrupt(0, wakeUpNow, LOW); //interrupt 0 is used when wakeUpNow function tied to pin 2 goes LOW
  digitalWrite(centPin,LOW);
  myservo1.attach(servoPin1); // attaches the servo on pin 1 to the servo object. write degrees 0-180, speed 1-255, run in bckgnd
  myservo2.attach(servoPin2);
  myservo4.attach(servoPin4);
  myservo3.attach(servoPin3);
  myservo1.write(75,25,true); // elbow set the initial position of the servo, as fast as possible, run in background
  delay(1000);
  myservo2.write(90,25,true);  // shoulder set the initial position of the servo, as fast as possible, wait until done
  delay(1000);
  myservo4.write(90,25,true);
  delay(1000);
  myservo3.write(90,25,true);
  }
```

Figure 22. Arduino’s Sketch void setup function – normal (baseline) operation

Figure 23 below shows an abbreviated section of code containing the instructions sent to the Arduino servos. Duplicative commenting was removed from the Arduino code for presentation in Figure 23. Comments are preceded by two consecutive solidus or slashes (i.e., //) placed side-by-side. These instructions illustrate the servo’s position, speed in getting to the position, and whether or not the instructions are carried out in unison with the command given to the servo before it. There are several delay commands in between many of the instructions. This simply causes a pause (or delay) between commands. It was very useful for disrupting the robotic arm’s performance later in the experiment.

Figures 24 and 25 below show the robotic arm’s simulated sample collection and placement positions, respectively, as it was programmed into the Arduino. This is beneficial for

```
void loop() {
for(int i=0; i <= 5; i++){  // added for loop to operate arm six cycles to simulate the collection and placement of six
test tubes from product tank to centrifuge   i = i++;
//begin process
myservo1.write(75,25,true); // elbow set the initial position of the servo, as fast as possible, run in background
delay(1000);
myservo2.write(90,25,true);  // shoulder set the initial position of the servo, as fast as possible, wait until done
delay(1000);
myservo4.write(90,25,true);
delay(1000);
myservo3.write(90,25,true);                //FIGURE 20
delay(1000);
myservo3.write(180,35,true);              //FIGURE 24
delay(1000);
myservo2.write(155,35,false);
delay(1000);
myservo1.write(70,35,true);
delay(1000);                 // move the servo to 70, fast speed, run background, write(degrees 0-180, speed 1-255,
wait to complete true-false)
myservo4.write(145,75,true);
delay(1000);
myservo2.write(90,35,false);
myservo1.write(0,45,true);
delay(1000);
myservo3.write(0,35,true);                 //FIGURE 25
delay(1000);
myservo2.write(155,35,false);
myservo1.write(95,55,true);
delay(1000);
myservo1.write(60,55,true);
delay(1000);
myservo4.write(90,35,true);
delay(1000);
}
//after process performs six cycles of simulated sampling, the robotic arm returns to rest position, detaches servos
and goes to sleep
myservo1.write(75,25,true); // elbow set the initial position of the servo, as fast as possible, run in background
delay(1000);
myservo2.write(90,25,true);  // shoulder set the initial position of the servo, as fast as possible, wait until done
delay(1000);
myservo4.write(90,25,true);
delay(1000);
}
```

Figure 23. Arduino's Sketch void loop function – normal (baseline) operation

interpreting the following Figure 26, showing robotic arm vibration data, in that it helps visualize the movement of the arm. It is also relevant in illustrating the vivid contrasts between that of a normally appearing process, as compared to a previously established baseline, to that of an altered one. The exploited, or altered, robotic Arduino Sketch is presented in section three.

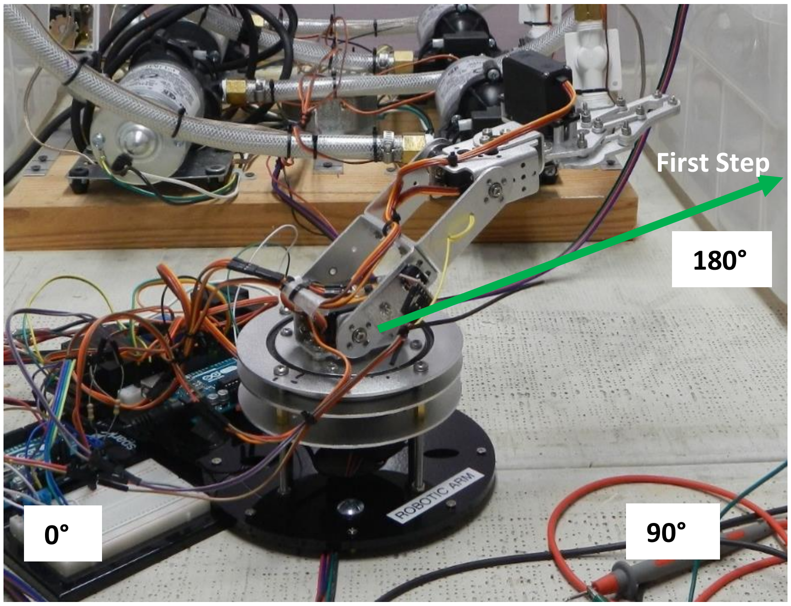


Figure 24. Robotic arm position for collecting product sample (product sample simulated)

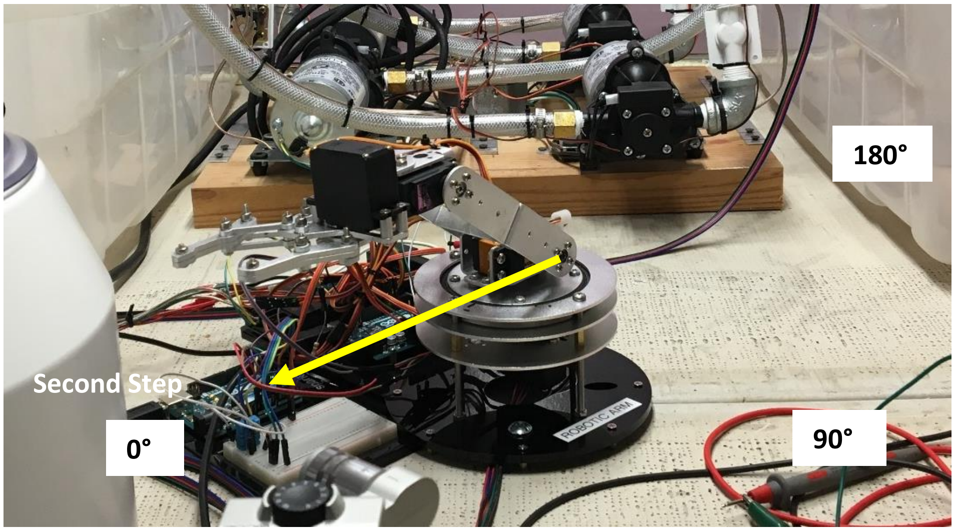


Figure 25. Robotic arm position for placing product sample into centrifuge rotor (product sample simulated)

In Figure 26, **Sa** represents the simulated sample collection activity and **C** represents the simulated centrifuge placement activity. The green arrow on the left of the graph indicates the

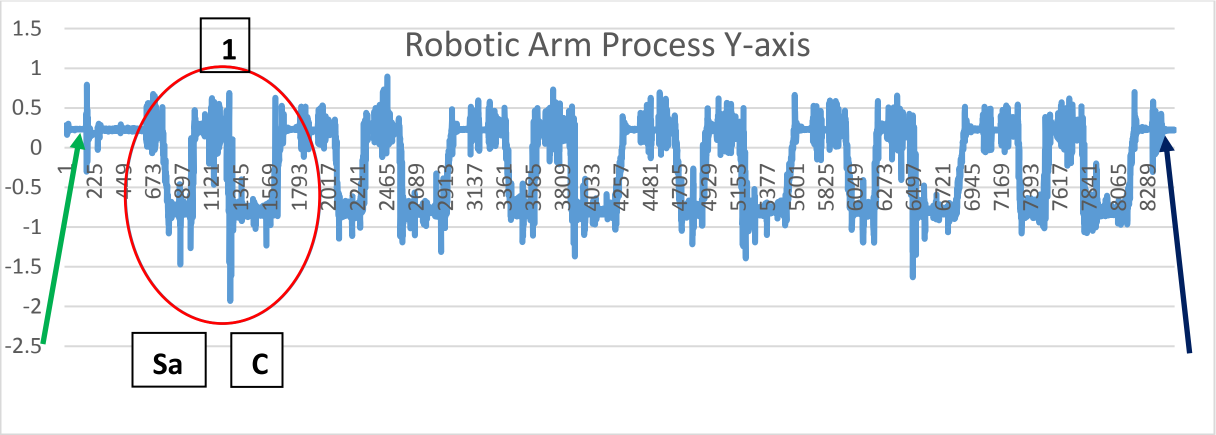


Figure 26. Robotic arm vibration (V) data, y-axis

process start time, and the dark blue arrow to the right of the signal represents when the sixth cycle completed and the arm returned to the ready-wait state. This graph is explained in more detail in section three when describing and illustrating the differences between a normal and potentially compromised abnormal state.

The variation of sound signals of the robotic arm can be clearly seen in Figure 27. A close look at the sound graph reveals a strong coorelation between the vibration and sound graph. The blue lines in the graph are inserted at cyclic intervals. The green line to the left signifies the process start and the red line to the right indicates the robotic arm's return to a ready-wait state. A visual observation of the process taking place corrobated the results and helped better understand the transitions between the process steps performed by the robotic arm. By following the write commands in the Arduino code above, Figure 23, and comparing both the sound and vibration graphs to each other, each movement can be accurately mapped to the baseline. This mapping ultimately enabled the creation of a mask that was used for the selection of different values from sampled data; both during an established baseline operation, and an intentionally manipulated code alteration operation. These values were then used to determine the probability of risk and to identify the affected component.

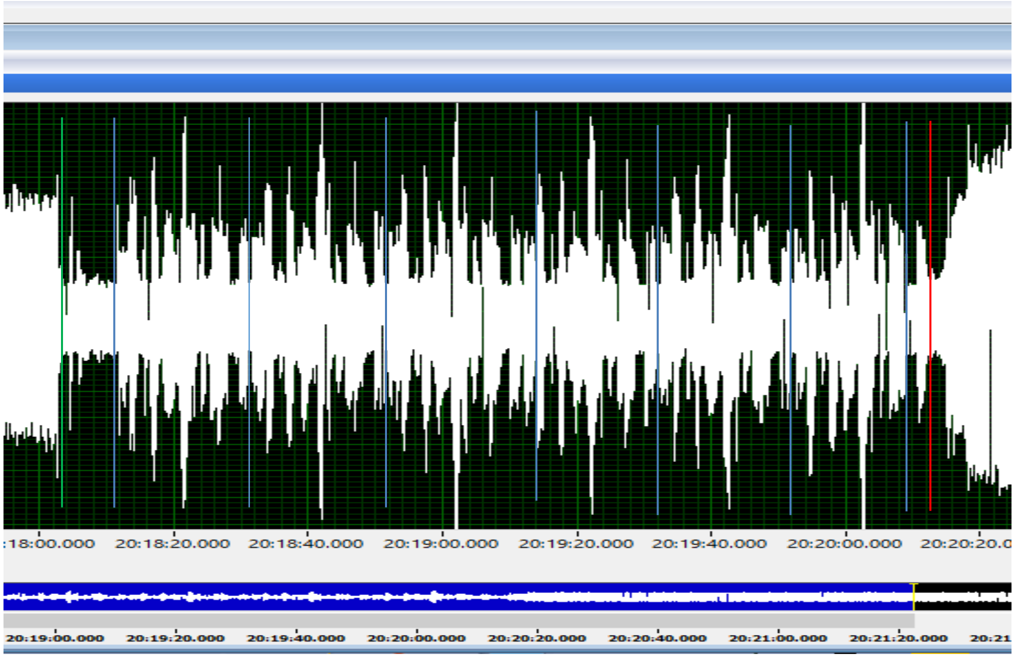


Figure 27. Robotic arm sound (S) data

Task Four-Energizing a Centrifuge

The last task in the faux chemical manufacturing process involves energizing a centrifuge that spins down the simulated product samples. Upon completion of task three, a relay on the robotic arm platform receives a one second 5vdc input, Figure 30, temporarily closing a circuit to one of the PLC inputs (this function is similar to the pressing of a physical normally open switch) which initializes an output that sends 12 VDC to the dc-to-ac solid-state relay, Figure 28; thereby, starting the centrifuge. The two-minute cycle is controlled by the centrifuge's PT (programmed time) input parameter of the PLC's TON function on rung eight, Figure 29. The centrifuge's rotor continues to spin after power has been removed, but eventually settles down naturally after a few moments.

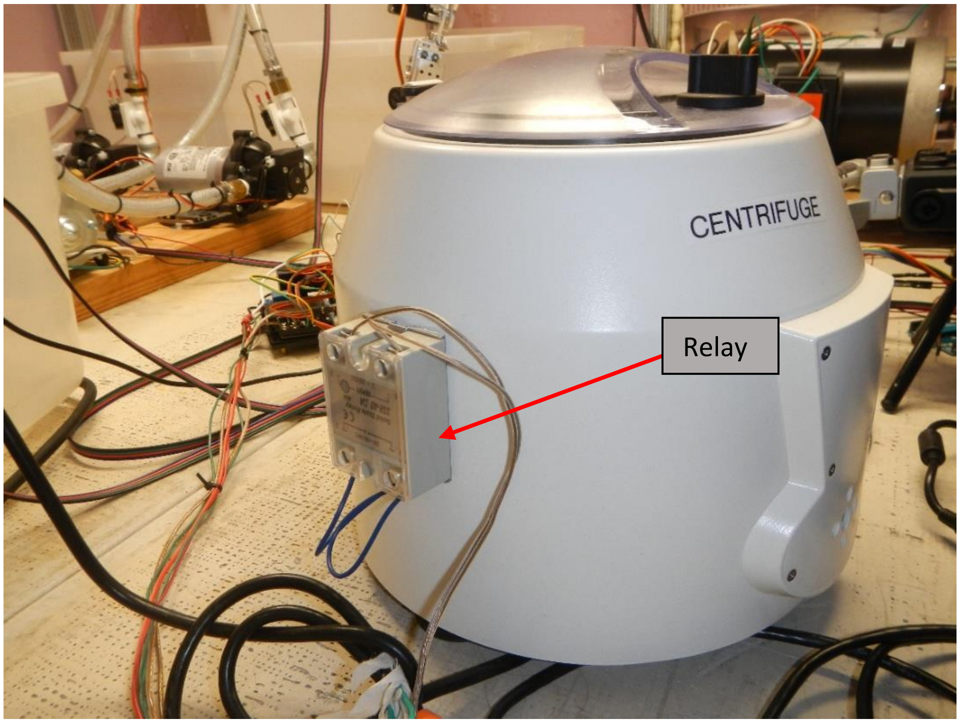


Figure 28. Centrifuge showing solid-state relay

The formula for determining risk within the four-task production cycle did not include the time it took for the centrifuge rotor to rest after power was terminated. Stimuli coding for task four was [S;V]. If the coasting (deceleration) state of the centrifuge had been included as part of the production cycle, a hall effect sensor for measuring rotor speed (sP) would have been included in the coding, thus resulting in the parameter [S;V;sP].

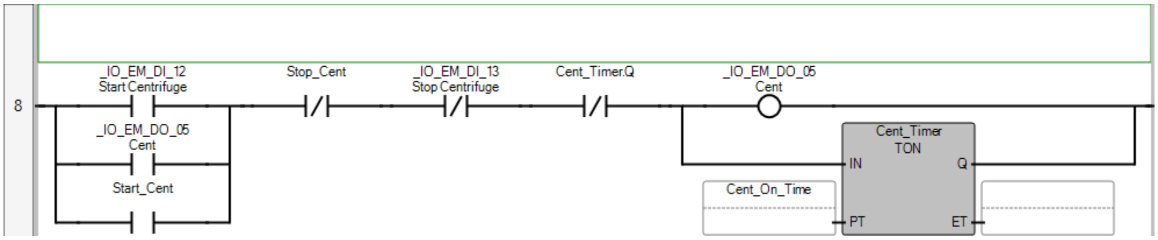
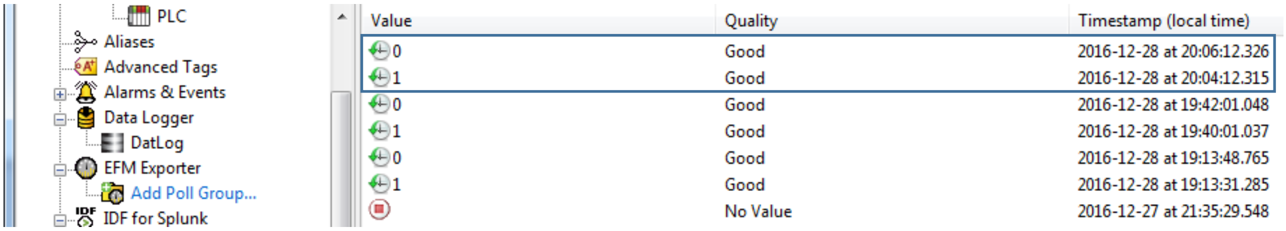


Figure 29. Centrifuge ladder logic



Value	Quality	Timestamp (local time)
0	Good	2016-12-28 at 20:06:12.326
1	Good	2016-12-28 at 20:04:12.315
0	Good	2016-12-28 at 19:42:01.048
1	Good	2016-12-28 at 19:40:01.037
0	Good	2016-12-28 at 19:13:48.765
1	Good	2016-12-28 at 19:13:31.285
	No Value	2016-12-27 at 21:35:29.548

Figure 30. Tag 7 Centrifuge Start/Stop Data, 00:02:00

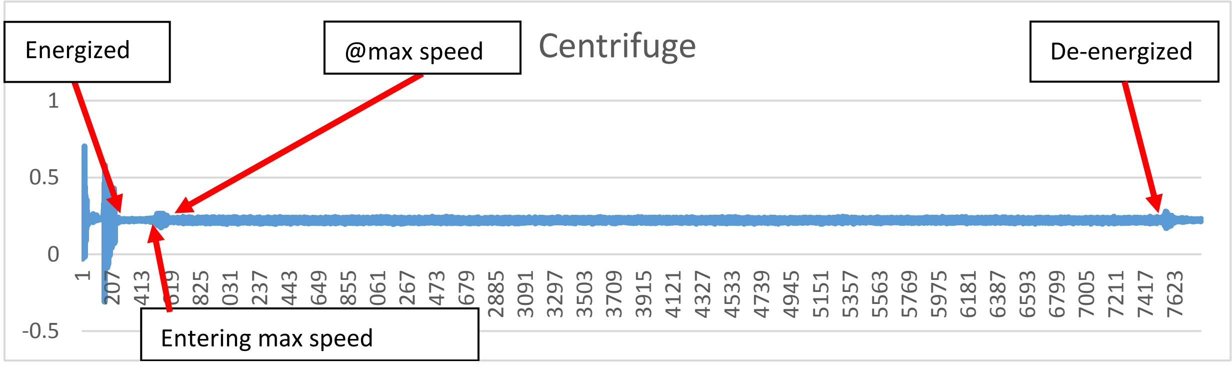


Figure 31. Centrifuge vibration data

The entire process cycle took 14 minutes and one second to complete. Even with the exploited asset, as introduced in this section, but explained in greater detail in section three, the time for the entire process cycle could have easily remained the same; however, the exploit example in the next section included a three second increase to the overall process cycle. Process timing is critical to the discovery and determination of systemic risks. Monitoring sensory input throughout each of the processing tasks in the production cycle at precise time intervals is key to discovering those risks. Therefore, baselining the processes after setting up the production environment's automated functions required a smooth, dependable, durable and reliable operation from start to finish.

The *initial* baselining was accomplished upon completion of the installation for each device. This process was an important first step for determining rudimentary performance measures and to learn whether the asset met the expectations of how that component interacted within the environment at large. Most importantly it was observed throughout its entire operation performing its intended task to ensure that it was adequate for the role it played and purpose it served.

Section Two – Sensor selection, placement and data results for defining normalcy

Baselining is a repetitive and iterative process for evaluating system performance over time. Baselines can change for a number of reasons. Baselining can be periodic or continuous, based on the implementation or purpose of the system or individual device. Establishing an operational baseline for this CI2CS environment took into consideration the number of controls in place and the fact that each process was performed in sequential and independent steps. This section discusses the approach taken in establishing the baseline method used in this research which is linear process baselining.

It is recognized and understood that an alteration to the baseline does not automatically constitute performance issues that result in failing components caused by a vicious exploit injected into the system, or for that matter, sabotage resulting from an insider threat. Some reasons for baseline changes, particularly as it relates to new construction or geographical location, stem from equipment and/or construction "settling in" or cross talk introduced by another industry's automated and/or mechanical processes. For example, when using sound and vibration sensors for monitoring a highly-automated production facility or coal power plant, an engineer may want to factor in rail car activity well ahead of arrival, during off-loading, and a certain time after departure. Although the sound from rail car activity may be canceled out by the operating commotion of the device in immediate proximity of its dedicated sensors, vibration sensors, especially geophones, tend to have a greater range of sensitivity, so that if a piece of equipment is not properly isolated or the sensor is not conditioned properly to its environment, false positives are likely to occur.

As most environments are unique, placement and implementation of different types of equipment varies in their operating characteristics. Some of these deviations can likely be attributed to supply chain involvement. This might include inconsistent quality control, various manufacturers making similar but not identical parts, and geographically dispersed assembly plants assembling the end-product using different standards and equipment. In this research for example, five of the same pumps (one being a spare) outwardly appeared to be identical, however, as shown in section one of this chapter, each pump (excluding the spare) had very different sound and vibration signatures. The baselining method used in the commissioning phase consisted of task specific operational functions meeting their expected and anticipated task objective. Table 12, introduced earlier, for example shows the measures used for establishing the initial operational baseline for the pumps. Many of these measures began as a temporary way of studying device specific characteristics until risk monitoring sensors were permanently installed. The sensors did allow for some adjustment in programming that permitted a way to condition the sensor more closely to its placement.

Type of Sensors and Installing

After commissioning the test-bed and establishing a rudimentary operational baseline, as described above, additional risk specific monitoring sensors were installed according to the coding stimuli assigned to the respective devices listed in the previous section for the functions they performed throughout those particular tasks. The coding stimuli was established during the commissioning phase, primarily through observation of a repeated process performing its corresponding task. The enablement of a segregated risk monitoring capability running in parallel to the system's programmed production cycle was realized at this phase in the experiment and led to the installation of a real-time clock (RTC) being integrated into the sensor network. The RTC remained separated from the control system network; however, it was initialized through the first output signal received from the PLC. Any compromise or deviation to the initialization process would have become evident during monitoring. Neither an unintended compromise nor a deviation ever occurred.

An evaluation of operational performance led to the selection of stimulus specific sensors, as identified by the coding in the previous section. There were three primary, four secondary and one tertiary input sensors utilized throughout this experiment. As listed in chapter three, the eight types of sensors used in this experiment, in their respective order of importance as described above, included a microphone, accelerometer, temperature, humidity, barometric, flow, ultrasonic and thermal. The data relating to each of the four processes relating to the particular sensor is listed and explained in the remainder of this section. The spatial environment for this experiment was coded [S;V;T;H).

This centralized sensor module provided the frame of reference that monitored all processes. As explained in chapter three, a frame of reference ensures that all processes were monitored consistently and equally relative to the sensor's position to that of the objects monitored throughout each production cycle. Figure 32 below shows a diagram of the spatial sensory module and sensors as coded. The sensors are labeled with their respective sensory designation.

Sensor placement is discussed in more detail in section two of this chapter. The sole accelerometer, although attached to the robotic arm, took the role of a centralized vibration

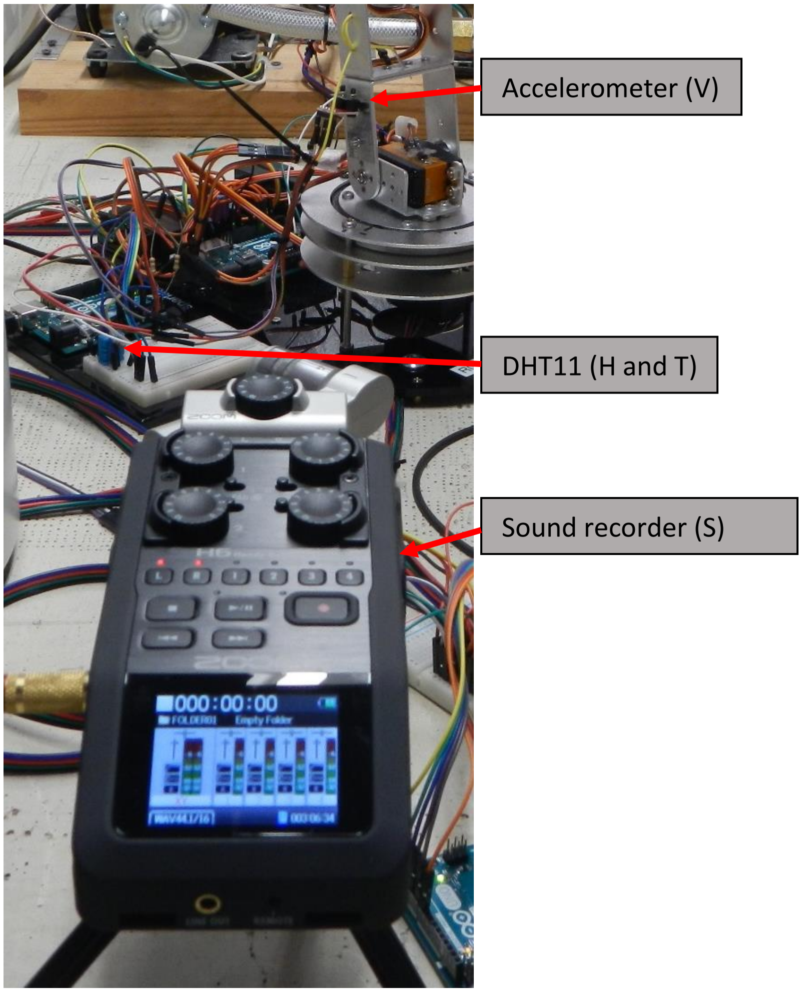


Figure 32. Sound recorder, vibration (accelerometer), temperature and humidity (DHT11) sensor for spatial environment

sensor, in large part because it was central to all of the processes, and also because the robotic arm provided a better placement of the sensor due to the characteristics of its configuration, especially in the ready-wait state. The robotic arm poised in a ready-wait state with the sensor mounted to the arm's shoulder actually improved, or at the least, maintained a certain quality of data with sufficient enough input to clearly identify start and stop points between processes, as well as, performance indicators during those processes. The robotic arm's inherent slack between its virtual anatomy such as the shoulder, elbow, hip, etc., created a mechanical amplifier in that it protracted the vibrations of not just the robotic arm functions, but those that were created by the other devices during their running process as well. This can be observed in any of the graphs originating from the data recorded during those processing cycles.

In order to establish an operational baseline, it was important that processes maintained their functional capabilities and operated within the parameters that were defined when documenting the initial baseline. The processes were run several times after the initial baseline to allow the test-bed to settle in. This was particularly important in one area of the test-bed where the supply tanks and product tank were located. Each supply tank had six gallons of water, each gallon weighing 8.34 lbs., and was located in one quadrant of the test-bed. The emptying of the supply tanks into the 15-gallon product tank shifted 100.8 lbs. volume of water to an essentially smaller footprint in another quadrant. Thus, it was necessary to study the weight redistribution through several cycles to learn whether a shifting load distribution interfered with the establishment of an operational baseline, particularly as it related to the vibration sensor.

Figure 33 displays the vibration data received during one of a number of production cycles conducted over the course of this research. Figure 34 shows sound data rendered in a graphic representation of a complete production cycle. Note transition points indicated by red arrows between processes in both graphs. Details of each task of the process are illustrated throughout the rest of this section. Both of these graphs establish and portray a vivid impression of the dynamic processes occurring throughout the entire production cycle. An early review of these two graphs will help guide the rest of the discussion regarding the data obtained not only by these two sensors, but also from the other sensors whose values are still useful for holistic risk monitoring.

A careful review of both graphs, Figure 33 and 34, shows that there is a strong correlation between the sound and vibration data and provides valuable feedback when compared to previous cycles that took place before this particular instance. With the exception of a few spikes during the sound and vibration recordings,

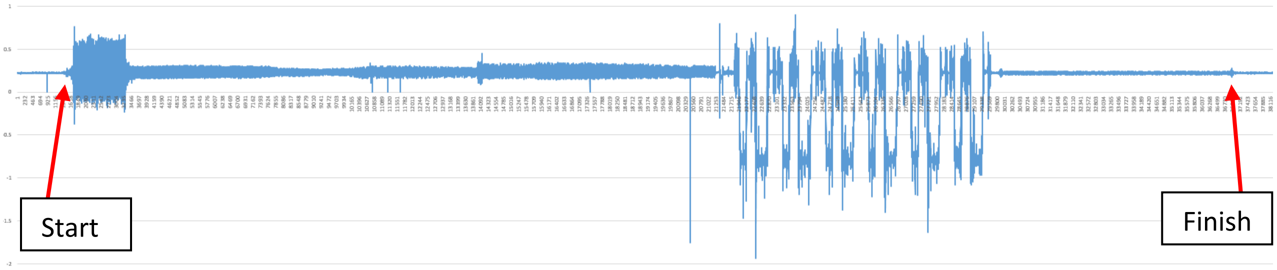


Figure 33. Vibration (V) graph, selected at random, of one complete production cycle

the graphs represent a fairly stable and fluid process. The sound and vibration sensors provided the most insightful, reliable and dependable data and complemented one another by the similarity in their functional characteristics. The signals obtained from both the vibration sensor and sound sensor consistently portrayed an image of tasks in the process taking place with better accuracy and an extremely high reliability rate compared to the other sensors used in this research. A comparative analysis between these two data sets ultimately determined whether a potential compromise occurred and the probability that some level of risk existed within the system.

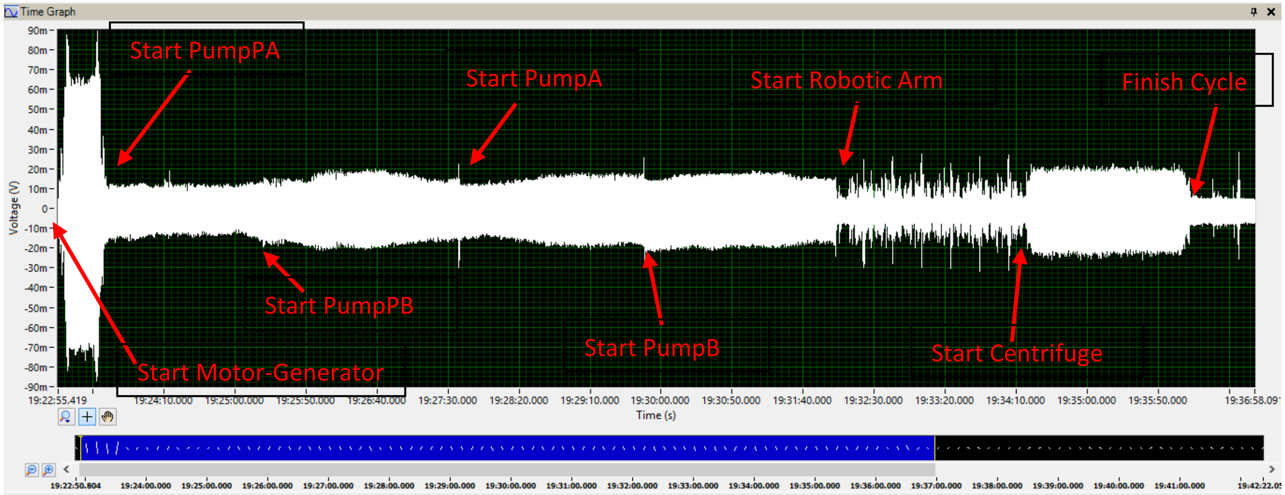


Figure 34. Sound (S) graph of one complete production cycle

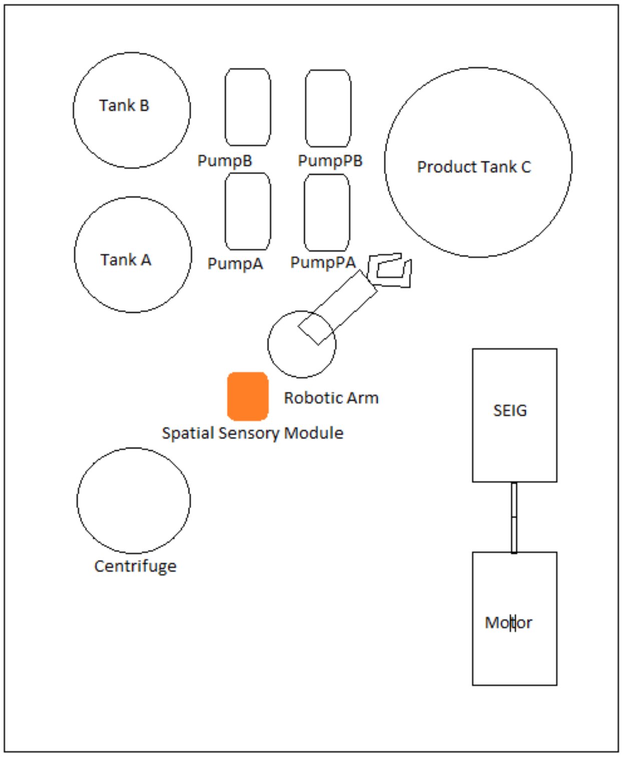


Figure 35. Relational frame of reference to spatial sensory module located adjacent to robotic arm

Figure 35 identifies the location of the SSM centrally located for monitoring the stimuli emanating around it. A frame of reference is essential when calculating relative changes to stimuli, especially temperature. Unlike sound and vibration data, where the amplitude can be artificially, physically, or programmatically increased or decreased for signal clarity, the input from a temperature sensor in relation to the producers of the climate altering stimuli affecting it cannot be as easily translated or enhanced. For example, the first production run on one particular day of monitoring started with the test-bed internal ambient temperature at 63-degrees. As the production cycles, incrementally increased, so too did the ambient temperature. This is to be expected. The more instances in close proximity of one another that each motor and pump operated, the more the temperature for each moving object individually increased with that of the ambient temperature. There was naturally a point of diminishing difference, as the test-bed environment was large enough to absorb much of the climate change that was occurring. Also, it is posited that the production cycles were operated for such short intervals that significant temperature variations and swings were not relevant enough to suggest an intentional or unintentional attempt to break, damage or compromise a particular object within the test-bed environment.

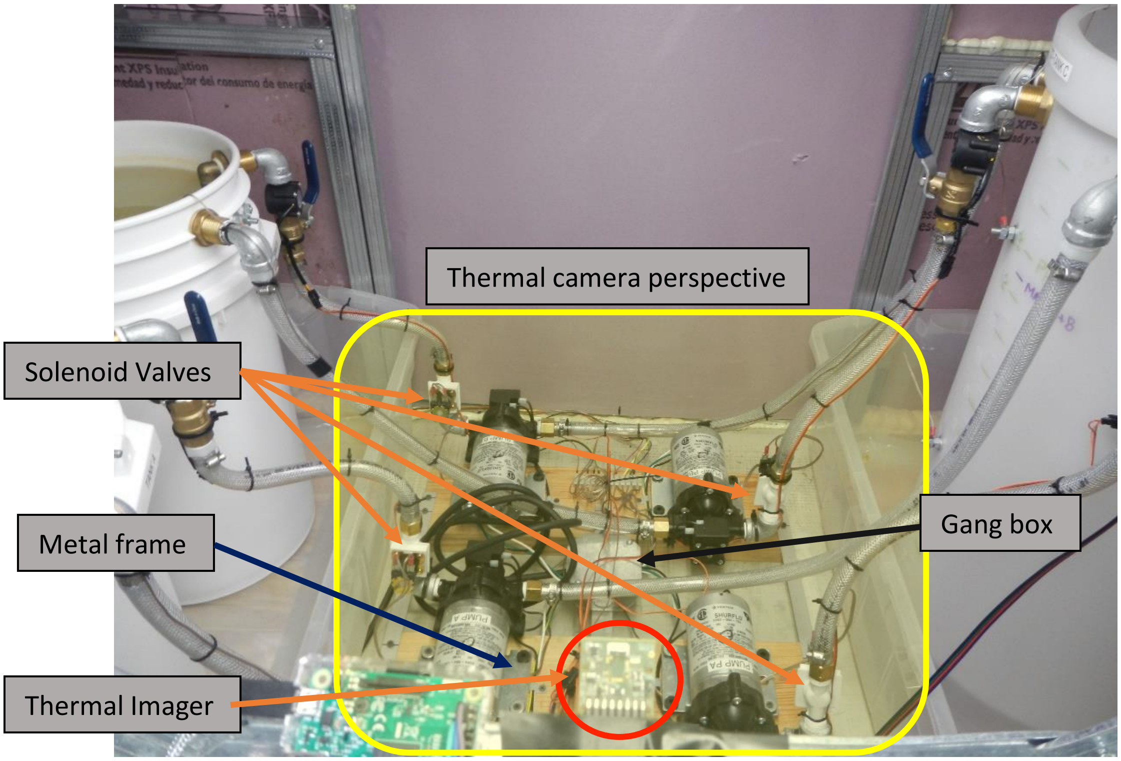
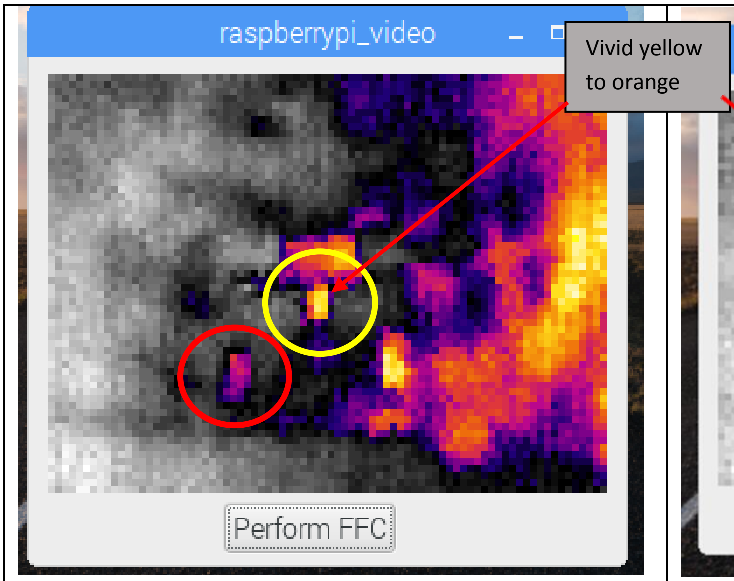


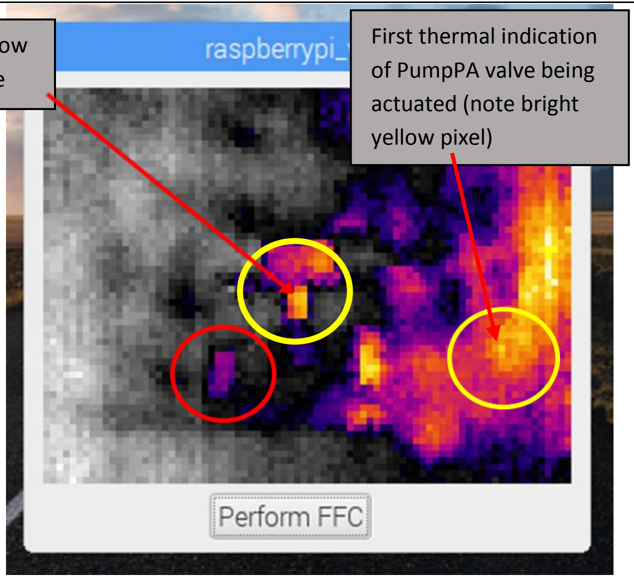
Figure 36. Perspective view of FLIR camera

A thermal imager was used to capture temperature characteristics occurring before, during, and following the batch process. Temperature signatures obtained from the thermal images were helpful in identifying and monitoring the sequence of the processes taking place during the pumping cycle. Figure 36 shows the mounting location of the thermal sensor (FLIR camera) used for studying and monitoring thermal activity created by the pump motors during their pumping interval. The imager, encircled in red, captures temperature signatures from the devices within the yellow rectangle. This includes the pump, its motor, and the relay valves, encircled in orange. The camera angle used to capture the view in Figure 36, provides a



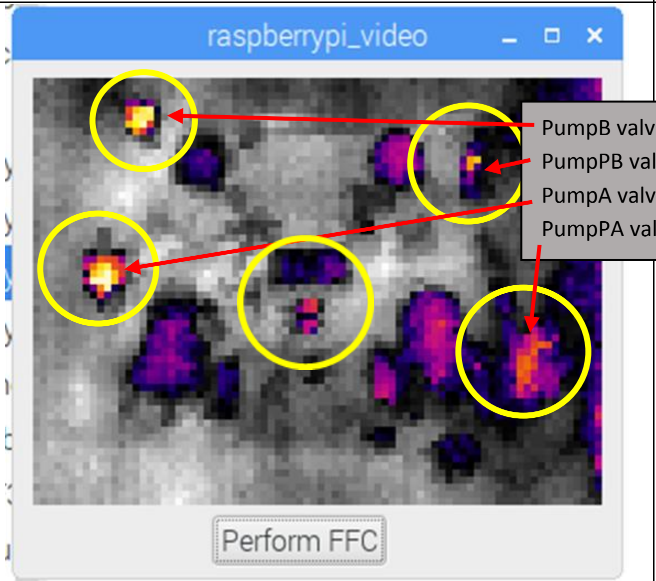
Vivid yellow to orange

Figure 37. Thermal image prior to start of batch process (screen shot 17:57)



First thermal indication of PumpPA valve being actuated (note bright yellow pixel)

Figure 38. Thermal image of same batch process just beginning to pump (screen shot 17:59)



PumpB valve
PumpPB valve
PumpA valve
PumpPA valve

Figure 39. Thermal image of same batch process finished, PumpB and PumpA motor and valve de-energized (screen shot 18:11)

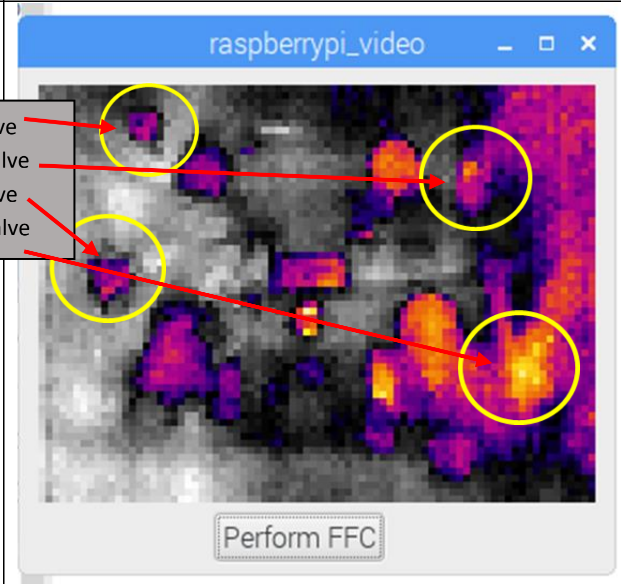


Figure 40. Thermal image of subsequent batch process with PumpPB motor and valve energized and PumpPA motor and valve de-energized (screen shot 18:18)

perspective for the preceding four images, Figures 37 - 40.

Some details were lost from view as external stimuli altered a portion of the image; thereby, rendering that particular portion of the results unusable. This can be seen in the Figure 37 above. Because the results demonstrated an inconsistency and untimely irregularity in the readings, the thermal camera was only used as an "event" and not a "risk" monitoring sensor. An event sensor was used for general purpose process monitoring. The red circle in Figure 37 captures the purple pixels in the lower left center of the figure above that illustrate the metal frame of pump A. This can be clearly seen, and is labeled, in Figure 36 as well. The orange and yellow pixels, yellow circle, mid-center of the figure show the metal electrical box, also labeled in Figure 36, housing the motor wiring from all four motors, and the aluminum strap it was mounted to. These colors, the purple, red, and yellow represent a slight temperature increase to other items around it.

In the Figure, 39, above, over a ten-minute period, the heat signature shown in Figure 38 above begins to fade ever so slightly. The heat dissipation is most notable in the mid-center area, encircled in yellow, in Figure 38, where the vivid yellow seen in the previous figure, 37, has faded from yellow to orange. Pump PA and pump PB's solenoids are not distinctly visible as the other two pump's A and B are in Figure 39. Although the thermal signature produced by the solenoids (valves) were not expected to be so prevalent; they were not unanticipated.

It was unknown that the solenoids heat signature would be as prevalent as they turned out to be. All four solenoids are clearly identified in Figures 39 and 40 above. The two orange/purple and orange/purple/yellow pixels in the right upper and lower quarter of Figure 39 show the residual heat still emanating from their earlier pumping interval when the solenoids were energized, while the two yellow/white and yellow/white/orage pixels in the upper left quarter on the left indicate that pump A has completed its pumping interval and pump B has recently finished. Figures 39 and 40, both show the stimulus emanating from all four of the motors as well. All solenoid valves in Figure 38 are encircled in yellow. Figure 39 illustrates a process, subsequent to the one that just completed and illustrated in Figures 38 and 39 above, where PumpPA has finished and PumpPB is begnning.

Table 13 Sample serial capture of DHT11 raw data using Cool Term

20:31:56	Humidity	Temp C	Temp F	Relative Humidity in C	Relative Humidity in F
388674	49.00%	17.00 *C	62.60 *F	16.03 *C	60.86 *F
388687	49.00%	17.00 *C	62.60 *F	16.03 *C	60.86 *F
388702	49.00%	17.00 *C	62.60 *F	16.03 *C	60.86 *F
388716	49.00%	17.00 *C	62.60 *F	16.03 *C	60.86 *F
388730	49.00%	17.00 *C	62.60 *F	16.03 *C	60.86 *F
388745	49.00%	17.00 *C	62.60 *F	16.03 *C	60.86 *F
388759	49.00%	17.00 *C	62.60 *F	16.03 *C	60.86 *F
20:31:56					
388788	49.00%	17.00 *C	62.60 *F	16.03 *C	60.86 *F
388802	49.00%	17.00 *C	62.60 *F	16.03 *C	60.86 *F
388816	49.00%	17.00 *C	62.60 *F	16.03 *C	60.86 *F
388831	49.00%	17.00 *C	62.60 *F	16.03 *C	60.86 *F
388845	49.00%	17.00 *C	62.60 *F	16.03 *C	60.86 *F
388859	49.00%	17.00 *C	62.60 *F	16.03 *C	60.86 *F
388874	49.00%	17.00 *C	62.60 *F	16.03 *C	60.86 *F
20:31:56					
388888	49.00%	17.00 *C	62.60 *F	16.03 *C	60.86 *F
388902	49.00%	17.00 *C	62.60 *F	16.03 *C	60.86 *F
388917	49.00%	17.00 *C	62.60 *F	16.03 *C	60.86 *F
388931	49.00%	17.00 *C	62.60 *F	16.03 *C	60.86 *F
388945	49.00%	17.00 *C	62.60 *F	16.03 *C	60.86 *F
388959	49.00%	17.00 *C	62.60 *F	16.03 *C	60.86 *F
388973	49.00%	17.00 *C	62.60 *F	16.03 *C	60.86 *F

Table 13 presents a sampling of DHT11 data obtained during one of the several dozen runs taken over the course of testing using CoolTerm. The DHT11 sensor was used to obtain the information recorded in Table 13. The data represents the stimuli of the internal test-bed environment at the time a production cycle was in progress. Although, seemingly insignificant, and for this research only used to memorialize the current atmospheric conditions of the space used by the automation assets at the time of the research, this data is critical in recognizing performance standards and characteristics when there are dramatic changes in temperature. The test-bed did not experience any dramatic temperature changes.

Table 14 Sample serial capture of flow rate raw data (PumpPA)

Liters per hour
64 L/hour
536 L/hour
552 L/hour
560 L/hour
552 L/hour
552 L/hour
560 L/hour
560 L/hour
552 L/hour
560 L/hour
560 L/hour
552 L/hour
560 L/hour
560 L/hour
560 L/hour
560 L/hour
552 L/hour
560 L/hour
560 L/hour
560 L/hour
560 L/hour
568 L/hour
560 L/hour
560 L/hour
560 L/hour

The data shown in Tables 14, 15, 16, and 17 appears in its raw state; however, a header row was appended to each of these tables in order to identify the data being presented. Table 14 lists a partial instance of flow rate data from pump PA. The leading data consisted of zeros until the pump began running. The zeros were removed from the table for brevity. The first value, 64 L/hr., displayed in the first line following the header row, accounts for the void in the supply line. In this sample the flow rate is seen to normalize between 552 and 568 liters. This is properly indicated by the majority of 560 L/hour entries. The difference between the minimum and maximum values would have to be noted here, and factored into any equation relying on this data for calculating risk.

Tables 15, 16, and 17 show fill states between tanks during the batch process. This begins with the data presented in Table 15. An ultrasonic sensor, which is used to measure distance, was used in acquiring this data. Initially an Arduino, in combination with physical conductors, acting as a switch, was used to sense tank water levels. The Arduino's limited controller capabilities and insufficient number of interrupt pins prevented an implementation of this type of switch, such that an alternative type of sensor was installed. In

Table 15 Sample serial capture of ultrasonic raw pre-fill data (in inches)

Tank A	Tank B	Product Tank
2	2	55
2	3	54
2	3	55
2	3	54
2	3	55
2	3	55
2	3	54
2	3	55
2	2	55
2	2	55
2	3	54
2	3	56
2	3	56
2	3	56
2	3	55
2	3	55
2	3	54
2	3	56

Table 15, it can be seen that the water level of tank A is two inches away from the sensor. The water level for tank B fluctuates between two and three inches from the sensor, and the level of the product tank fluctuates between 54 and 56 inches. The reason for this is because each tank maintained a small amount of water after moving its contents from one to the other, staying right above the opening of the supply line in order to prevent air from entering into the system.

Table 16. Sample serial capture of ultrasonic raw data (in inches) - begin product tank fill

Tank A	Tank B	Product Tank
3	3	55
3	3	55
3	3	55
4	3	55
3	3	55
4	3	56
4	3	56
4	3	56
4	3	56
4	3	54
4	3	55
4	3	55
4	2	54
4	3	55
4	3	55

In Table 16 above, PumpPA has started filling the product tank. It was noted in the previous table that the product tank averaged a 55-inch distance between the existing water level and the sensor. Since the product tank is much larger in volume than Tank A, fill values for the product tank will take longer before changes in the filling capacity are indicated.

Table 17 below illustrates the data anticipated while pump PB is operating. The data in this table indicates that pump PB is filling because the values in Tank A’s column is a consistent 36 inches from the water level to the sensor. This translates into Tank A being empty, less the small amount of water used for sealing the supply line. Unless there was a compromised or defective controller, faulty wiring or sensor, it would be unlikely that tank B would indicate a value less than its fill capacity, averaging between a two and three-inch reading, while Tank A displayed a 36-inch distance between its water level and sensor.

Table 17. Sample serial capture of ultrasonic raw data (in inches_ after PumpPA finished and PumpPB has started

Tank A	Tank B	Product Tank
36	13	29
36	13	28
36	14	29
36	13	29
36	13	28
36	14	29
36	13	29
36	14	28
36	13	29
36	13	29
36	13	28
36	13	29
36	13	29
36	14	28
36	13	29
36	14	29
36	13	28
36	14	28
36	14	29
36	14	28
36	14	28
36	14	29
36	14	28
36	14	28
36	14	29

The accelerometer raw data results are listed in the Table 18; however, they will be explained in greater detail in the next section. An accelerometer uses three axes for determining gyroscopic positioning. In this research, as described earlier, the accelerometer was placed on the robotic arm. Due to the orientation of the X and Y axis, the sensor could be X or Y dominant relative to the mounting position, or orientation, and the relationship it has with the device to which it is attached. The Y axis was dominant in this installation; therefore, data from the Y axis column was used in developing an algorithm for determining probability of risk.

The data presented in Table 18 shows the values for the X, Y, and Z axes as listed under their applicable columns. However, the first column indicates an absolute time the data was recorded interspersed with the relative time, displayed in milliseconds, that the sensors in the test-bed environment were being logged, whether the production cycle had started or not. The graph displayed earlier in this section, Figure 34, used the data derived from that listed on the Y-axis column.

The values in the Y-axis column of Table 18 are between 0.216 and 0.245. The fact that these values are positive indicate the position of the robotic arm is rearward or away from the working area. Negative values indicate that the arm is forward, or reaching out toward the work area. This could also be expressed as rear or front of center. If Y-axis values registered an average of one or negative one, it indicates that the robotic arm was either zero or 180-degrees, in other words, parallel to the base.

By examining the Arduino code in Figure 23 earlier, it is noted that the arm segment between the base and the elbow is set to 90 degrees. This is not a true 90-degrees in relation to the x, y, and z planes of the accelerometer, but instead, it is to the servo’s position in relation to the arm’s mount. There is also the introduction of slack (jitter) due to the poor quality in manufacturing of the robotic arm. Therefore, the specifications, quality and tolerances associated with the bearings, mounts, and servos used in the construction of the robotic arm all

Table 18. Sample serial capture of accelerometer raw data using Cool Term

Absolute Time in HH:MM:SS with Relative Time in Milliseconds	X-axis	Y-axis	Z-axis
20:31:56			
388674	-0.207	0.238	-0.952
388687	-0.214	0.217	-0.95
388702	-0.222	0.236	-0.948
388716	-0.229	0.245	-0.953
388730	-0.226	0.236	-0.948
388745	-0.221	0.224	-0.949
388759	-0.216	0.227	-0.948
20:31:56			
388788	-0.198	0.229	-0.952
388802	-0.21	0.229	-0.953
388816	-0.157	0.216	-0.944
388831	-0.213	0.225	-0.949
388845	-0.218	0.225	-0.956
388859	-0.217	0.229	-0.951
388874	-0.214	0.228	-0.959
20:31:56			
388888	-0.214	0.227	-0.951
388902	-0.223	0.222	-0.956
388917	-0.216	0.225	-0.954
388931	-0.213	0.228	-0.95
388945	-0.212	0.229	-0.953
388959	-0.216	0.228	-0.948
388973	-0.215	0.235	-0.948

contribute to the total amount of slack inherited by the system. As long as the asset's specifications and design nuances are recognized, understood, and properly factored into the process or set of processes, early in the implementation phase, it is possible to capture usable data. The design nuances were recognized and properly factored into the analysis such that valid usable data was obtained.

Section Three – Introducing the Exploit (Breaking the Robotic Arm)

Prior to performing any of the studies involving the automation processes taking place during the production cycles, it was necessary to energize the test-bed. Due to safety reasons and maintenance activities, the test-bed was de-energized after every set of experiments; not each production cycle. It was noted that during the energization, the relay for the robotic arm would receive voltage and wake the Arduino causing it to go through six-complete cycles of simulating the collection and placement of test tubes that imitated a quality control sampling process. These actions were to be expected since the Arduino was programmed to do this. However, after the initial energization, the Arduino controller remained energized which prevented the arm from functioning out of sequence despite the sleep mode function it received after performing its assigned task.

An Arduino controller reset function is different from the sleep and wake function. The most significant difference between the two functions is that a reset causes an interruption to the controller's power, whereas a sleep and wake function do not. This operational characteristic along with the way the robotic arm's functional process was configured, as part of the overall automation system helped identify a vector and created an opportunistic situation for introducing an alternative set of instructions to the Arduino controller while powering up, or in synchronization with its normal operating cycle. The following two graphs, Figures 41 and 42, illustrate what shape a normal sound and vibration signature, in comparison with the established baseline, should take while performing its programmed tasks of collecting and placing a simulated product sample from the tank to the centrifuge.

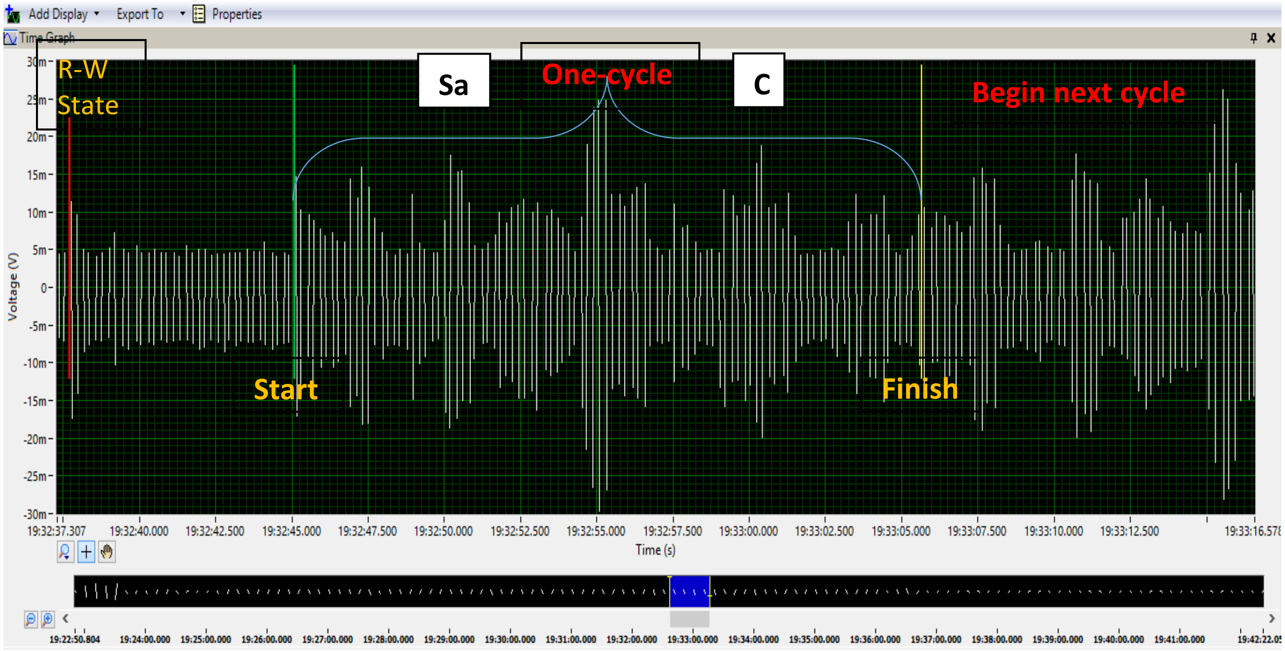


Figure 41. Sound graph showing intervals and transitions of the robotic arm process

In Figure 41, a sound graph provides a visual representation, creating a sound signature, illustrating the first cycle of a normally performing/functioning robotic arm. In other words, this particular graphic represents that this task instance is operating or performing to the established baseline. This graph focuses in on the appearance of a singular cyclic task being performed during this task in the process. By providing a vivid visual account of each complete cycle, a timely selection of the relative time and its expected corresponding voltage can be used as variables for determining risk. It is with these graphic representations that certain variables were chosen and used for that purpose.

Figure 41 incorporates both leading and trailing data to show the transitions between the actual start and finish of the collection and placement cycle. The red-line labeled as R-W State is the arm’s “home” position, referred to as the ready-wait state and holds there until the Arduino controller receives the wake command. There is a short pause before the robot rotates counter clockwise to the 180-degree position and begins moving through the motions described in the last section. The yellow vertical line in Figure 41 indicates where and when this cycle finished and the next one began. The significant voltage activity observed in this graph between 30 and -30 millivolts at an absolute time of 19:32:55.000, indicative of when the data was rendered into creating this graph for figurative purposes and not at the time the data was produced, represents the rotation from the 180-degree to the 0-degree position, relative to the arm’s attachment to the rotating base. Sa refers to the simulated sample collection and C refers to the simulated placement into the centrifuge.

Figure 42 is a graphic rendering of the vibration data collected during the robotic arm step of the production cycle. A sample of the vibration data was introduced in the last section; however, it is described in more detail here. Figure 42 shows the vibration characteristics from each of the six individual cycles performed at the time this particular process took place. The red lines between two adjoining circles delineates between where one cycle started and the other ended. As can be seen there are a few runts, not anomalies, but none that indicate an extreme or significant shift or alteration to the established baseline. Any vibration or sound data point can be exaggerated by even a slight disturbance to the functioning asset: however, an awareness of these occurrences can aid in filtering out garbage or extraneous noise.

A precisely-timed upload to the Arduino proved that a synchronized execution of a modified code, designed for the purpose of causing a malicious consequence, could be accomplished during the transition between the PumpB back-fill process and the robotic arm

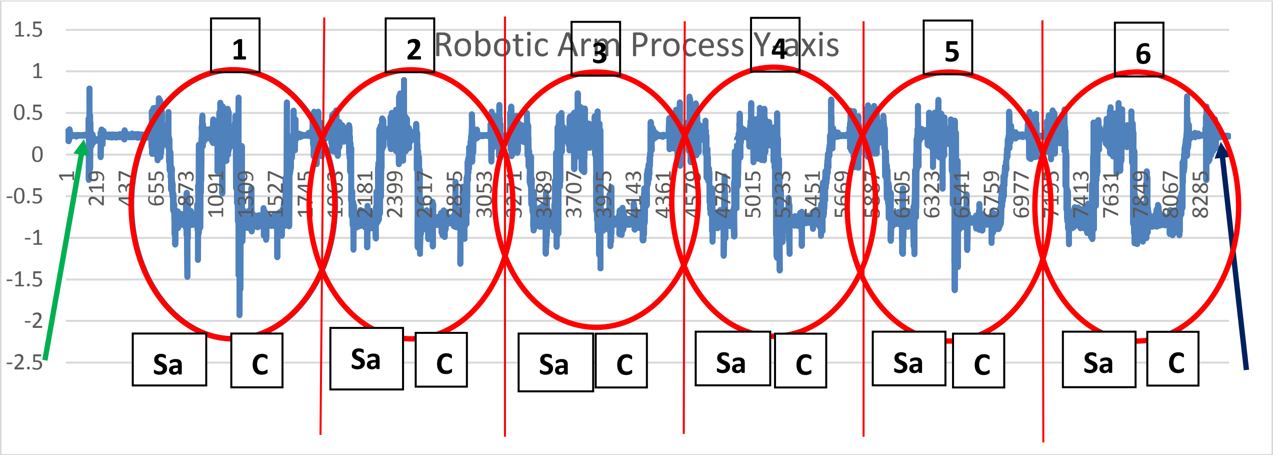


Figure 42. Robotic arm vibration data, y-axis

sampling process without detection; however, the first erratic action in response to the altered programming would trigger an alert. An alert was presented in the form of a graph in the same way all activity was rendered during the assessment process; however, in the instances where the Arduino's program had been modified to disrupt the process, the graph line resulting from the discovered deviation contrasted sharply with that of the baseline graph.

The timestamps, as recorded by the data historian during logging of the data used to create the graph above indicates a start time of 13:47:38.500 and end time of 14:01:36.832, thereby, the total time for this production cycle was 13 minutes, 58 seconds, and 332 milliseconds. This would indicate a difference of just under three seconds between the total time for this particular production cycle from that of others described in section two of this chapter. The reason for the time discrepancy lies in the modified code of the Arduino controller. In this instance, referring to Figure 43 below, the code was uploaded to the Arduino controller following the energization of the test-bed's main power strip.

The Arduino was powered by its own dedicated 12 VDC power supply and did not remain in a constant programming or monitoring capable state; however, a USB 2.0, type B cable was kept plugged into the USB port should monitoring or programming be necessary. A

USB connection provides the distinct capability of not only supplying power, but also of having the capacity to monitor and program the controller. This capacity includes the aspect of monitoring serial data transmitted to or received from the controller, and or uploading programming instructions (Blum, 2013). A controller with an independent power supply is unaffected by this arrangement. An Arduino will perform the last programming instructions received by the controller whenever power is applied.

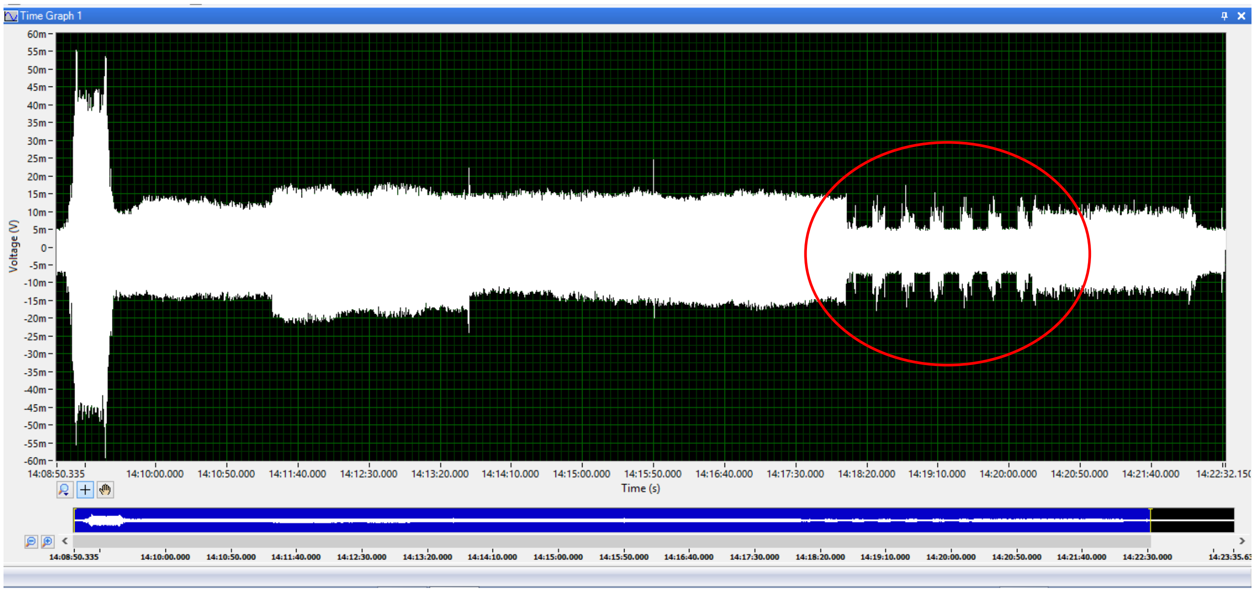


Figure 43. Sound graph of full production cycle showing deviant behavior caused by an altered robotic arm controller program

The application of power through either a USB connection or from a dedicated power supply will not impact an Arduino’s process as long as one form of continuous power exists. In other words, as long as the controller is in an energized state, attaching a USB cable will not disrupt the controller’s condition. Likewise, if a USB cable is attached, detaching a live power supply plug from an Arduino power jack will not have an impact on the processes taking place, with one exception. If an Arduino controlled peripheral device is powered by the Vin (voltage in) pin on the Arduino board, the reference voltage, if greater than five volts of direct current to that device will be affected; thereby causing an undesirable and possibly unpredictable response from that device (Blum, 2013). The Vin pin is marked as such because of its relation to the power input. This is not to be confused with Vout (voltage out), although it might seem more appropriate and in standing with conventional nomenclatures. This is simply not the case here. Peripheral devices can receive greater than five volts DC when powered by the Vin output pin.

Figure 43 shows a sound graph of a complete production cycle, from start to finish, including the section where the process indicated a deviation to the sound signature, or set of input signals, normally produced during the sampling step of the production cycle and that of the established baseline. Figure 44 shows how the normal sound signature appears compared to that of Figure 45 showing the sound signature as it appears after the robotic arm's programming instructions have been modified.

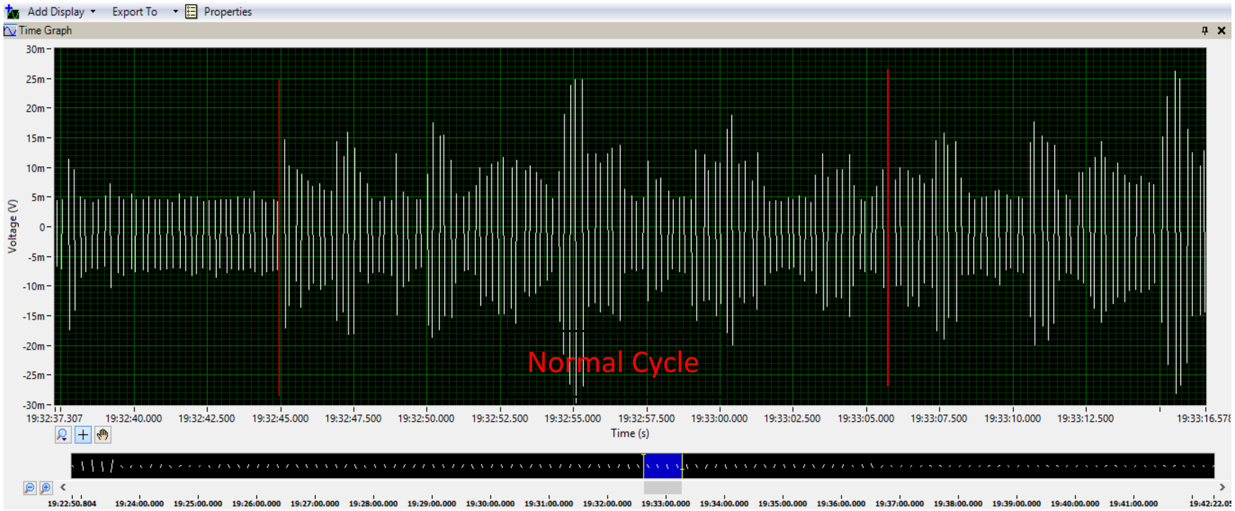


Figure 44. Normal sound signature from one cycle of the robotic arm process

The major difference between the two graphs, Figure 44 and 45, are shown by the stark contrasts involving the frequency and occurrence of the delay command appearing along each sampling cycle. In the previous, Figure 45, there is an incremental and sequential rythmic pattern to the delays placed between the different arm movements. These delays can be seen across one complete sampling cycle starting between timpstamp 19:32:45.000 and 19:33:05.500, marked by red lines, and the voltage axis between -7.5 and 5 mv. In the following Figure 45 there is only a singular drawn out delay between each sampling cycle.

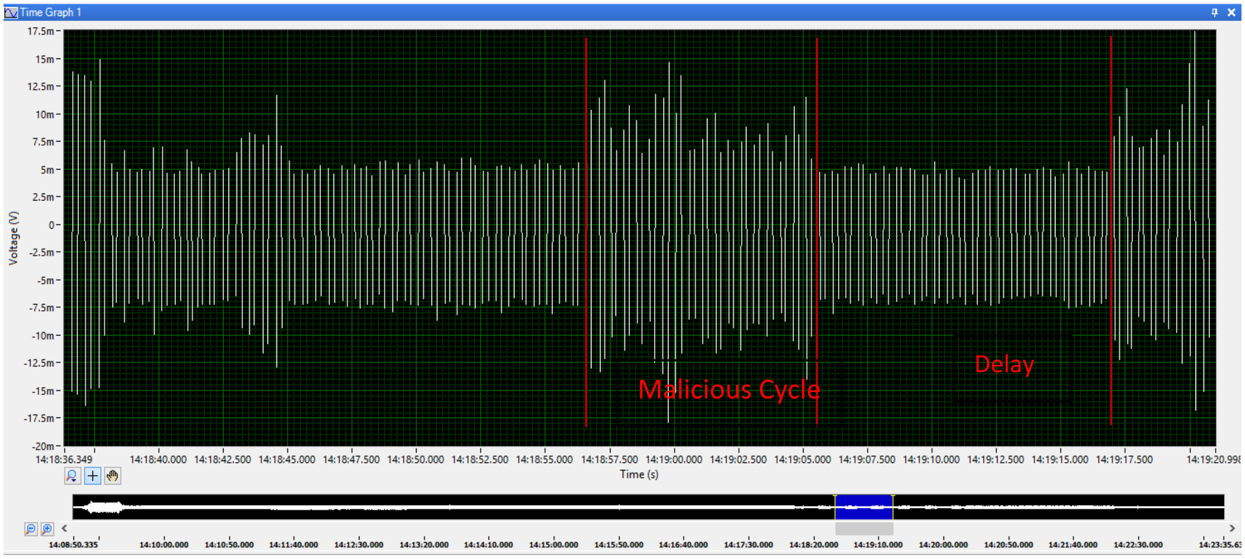


Figure 45. Sound signature of robotic arm process after malicious code upload

In Figure 45 the sampling process falls between 14:18:55.750 and 14:19:05.500. The volts still average between -7.5 and 5 mv during delays. Referring to the graphs Figures 44 and 45, the point at which the process deviates from the established baseline and a potential risk emerges is clear. The fact that a deviation has occurred in a process related function and the primary controller, the PLC in this case, rather than the Arduino, has not been compromised, largely based on the control system configuration and security controls in place, it is with high confidence that a systemic risk can be determined to have emerged from within or has been introduced into the system.

Figure 46 shows the vibration data sample from the very same process with the malicious code. Both a delay and cycle occurrence are indicated on the graph. As can be seen here, the arm was forward during

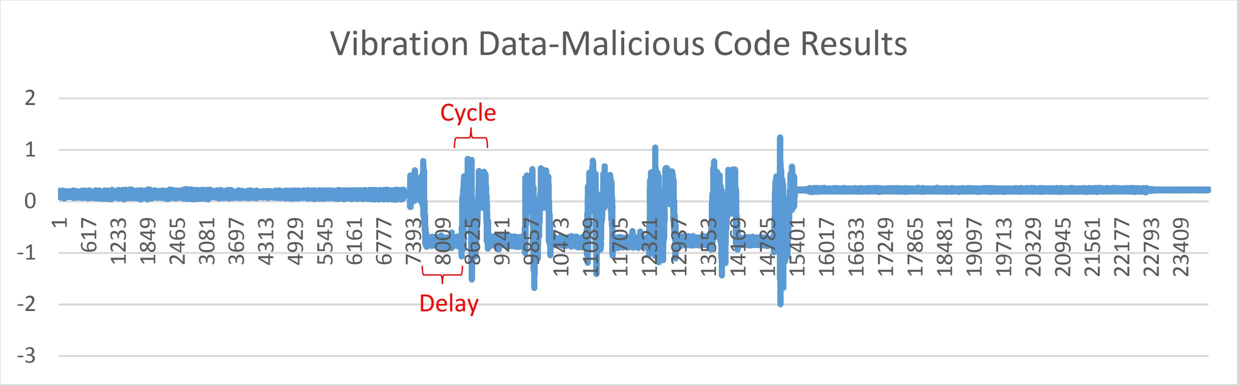


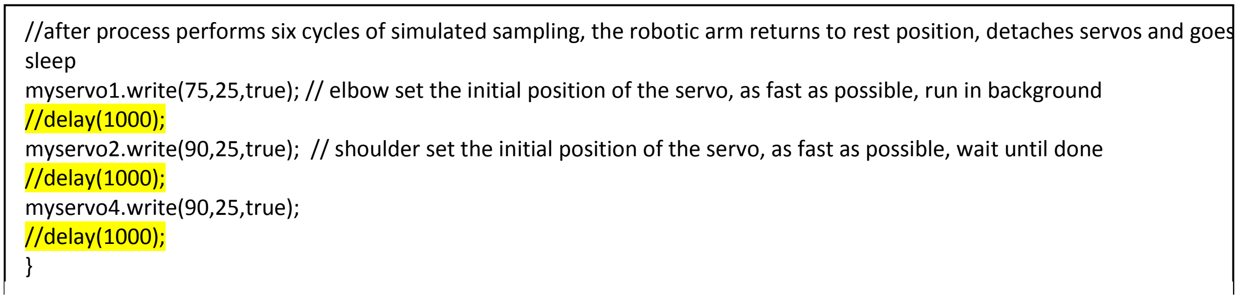
Figure 46. Vibration signature of robotic arm process after malicious code upload

```
void loop() {
for(int i=0; i <= 5; i++){  // added for loop to operate arm six cycles to simulate the collection and placement of six test
tubes from product tank to centrifuge   i = i++;
//begin process
myservo1.write(75,25,true); // elbow set the initial position of the servo, as fast as possible, run in background
//delay(1000);
myservo2.write(90,25,true);  // shoulder set the initial position of the servo, as fast as possible, wait until done
/delay(1000);
myservo4.write(90,25,true);
//delay(1000);
myservo3.write(90,25,true);
//delay(1000);
myservo3.write(180,35,true);
//delay(1000);
myservo2.write(155,35,false);
//delay(1000);
myservo1.write(70,35,true);
//delay(1000);                  // move the servo to 70, fast speed, run background, write(degrees 0-180, speed 1-255, wait to
complete true-false)
myservo4.write(145,75,true);
delay(13000);
myservo2.write(90,35,false);
myservo1.write(0,45,true);
//delay(1000);
myservo3.write(0,35,true);
//delay(1000);
myservo2.write(155,35,false);
myservo1.write(95,55,true);
//delay(1000);
myservo1.write(60,55,true);
//delay(1000);
myservo4.write(90,35,true);
//delay(1000);
}
```

Figure 47. Modified Arduino code having a direct impact on robotic arm's actions

the delay. The sharp spike down, following the delay, indicates the abrupt movements the arm was going through in between the delays.

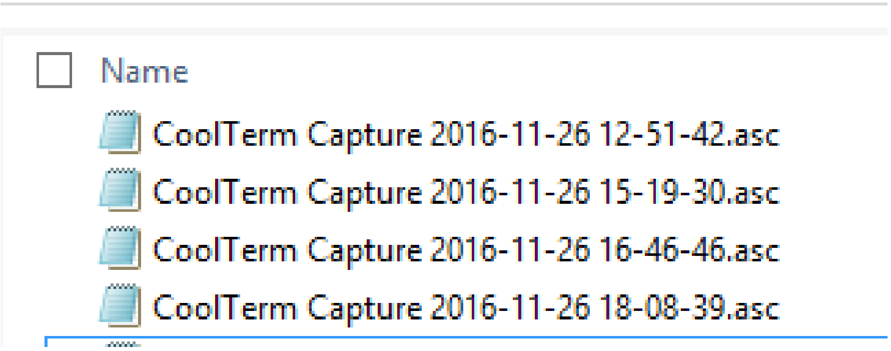
Figure 47 illustrates the code transformation used to produce the results as documented in Figures 45 and 46. The Arduino code in Figure 47 highlights the extended 13 second delay added between sampling cycles. It was determined that each individual delay could be consolidated into one strategic location within the code to maximize its effect. This kept the total time for the product sampling task in line with the baseline if it were not; however, as seen in Figure 48, affected by the three delays in sleep mode being commented out as well. The original delays that were placed between most commands were commented out. Commenting the command out prevents it from firing. This accounts for the three seconds lost from the overall production cycle, which could be accounted for from within the PLC's programming capability; however, it could also be independently discovered while sampling was performed as part of a real-time risk assessment process during the centrifuge task of the process. As it relates to sound and vibration monitoring, the assessment would still be based off time and voltage values.



```
//after process performs six cycles of simulated sampling, the robotic arm returns to rest position, detaches servos and goes
sleep
myservo1.write(75,25,true); // elbow set the initial position of the servo, as fast as possible, run in background
//delay(1000);
myservo2.write(90,25,true);  // shoulder set the initial position of the servo, as fast as possible, wait until done
//delay(1000);
myservo4.write(90,25,true);
//delay(1000);
}
```

Figure 48. Sleep instructions showing delay commands commented out

Since this was a controlled experiment, the sampling was conducted on demand. This means that each part of the process or tasks performed during the production cycle were supervised and monitored, and risk assessments were user initiated while the process was running. The raw data that was introduced in section two of this chapter populated a file, Figure 49, created at the start of each experiment, regardless of the number of cycles that ran. However, most of the files were created at the start of each cycle and not used to capture simultaneous, or



□ Name

CoolTerm Capture 2016-11-26 12-51-42.asc

CoolTerm Capture 2016-11-26 15-19-30.asc

CoolTerm Capture 2016-11-26 16-46-46.asc

CoolTerm Capture 2016-11-26 18-08-39.asc

Figure 49. Example of capture files and default file names generated by Cool Term

back-to-back, runs. Figures 50 and 51 show a screenshot of the logging program with its built in real- time logging feature used to capture serial data from the sensors used in this research.

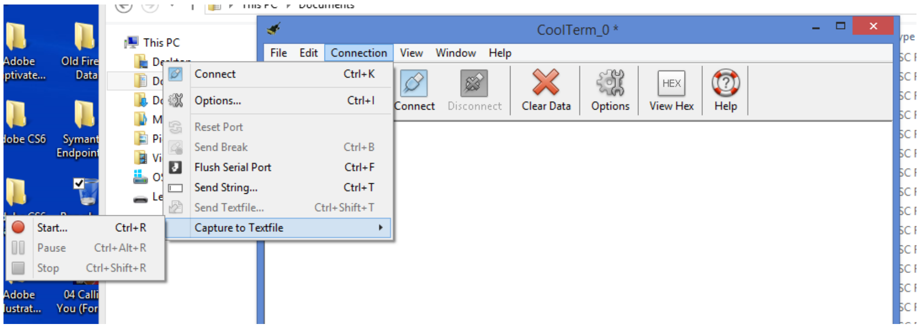
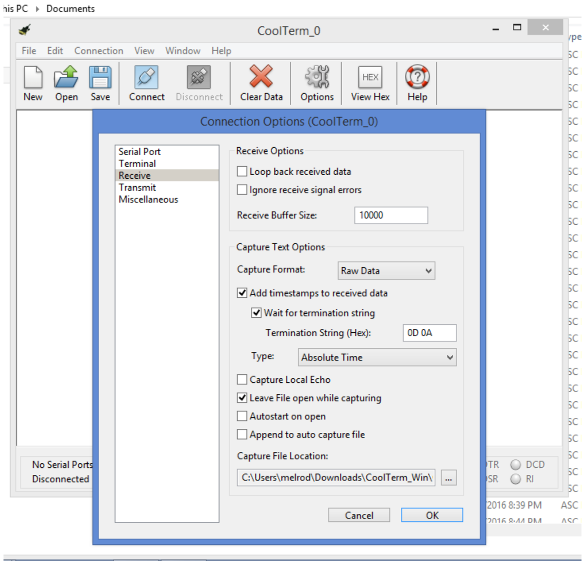


Figure 50. Cool Term data logger with real-time capture capability

There were three steps required in determining whether or not the system was placed at risk as a result of a cyber related incident versus a mechanical or performance related type of event:

- establishment of baseline for task evaluated,
- sampling of real-time data, and
- processing data with Ambari’s Hadoop Distributed File System Hive incorporated into Hortonworks Data Platform.

No device other than the robotic arm used an independent controller for controlling a process performing a specific task. All other devices were controlled by the PLC. The detection of a performance type of event based on a mechanical failure or operational quality issue was not applicable for this research.



Connection Options (CoolTerm_0)

Serial Port
Terminal
Receive
Transmit
Miscellaneous

Receive Options

□ Loop back received data
□ Ignore receive signal errors
Receive Buffer Size: 10000

Capture Text Options

Capture Format: Raw Data
☑ Add timestamps to received data
☑ Wait for termination string
Termination String (Hex): 0D 0A
Type: Absolute Time
□ Capture Local Echo
☑ Leave File open while capturing
□ Autostart on open
□ Append to auto capture file
Capture File Location:
C:\Users\melrod\Downloads\CoolTerm_Win\

Figure 51. Logging features of Cool Term showing timestamp feature

The first step in evaluating risk required the establishment of a baseline, especially for the task being evaluated. The data used in establishing the baseline was determined by averaging a collection of samples from both sound and vibration taken during fifty, closely supervised production cycles; after the test-bed had been officially commissioned and performance indicators registered consistently throughout each step of the process. The commissioning is discussed in section one above.

Appendix A lists a fraction of data, which is part of the baseline, and was used in creating Figure 52. This sample is used to illustrate the approach that was taken when testing real-time values and comparing them to the baseline values as a way of determining risk. The vibration sample was graphed to further illustrate the point. Figure 52 shows a portion of the baseline used for random sampling comparison. Figure 53 shows a random sample that was compared against the baseline shown in Figure 52.

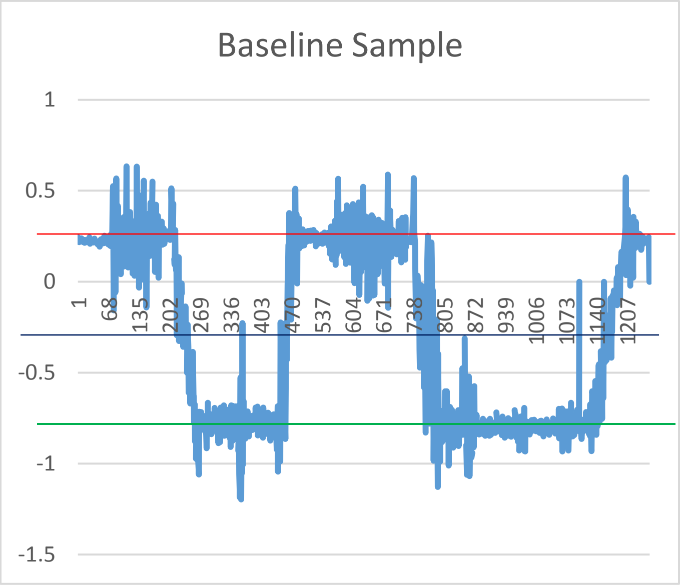


Figure 52. Baseline vibration sample

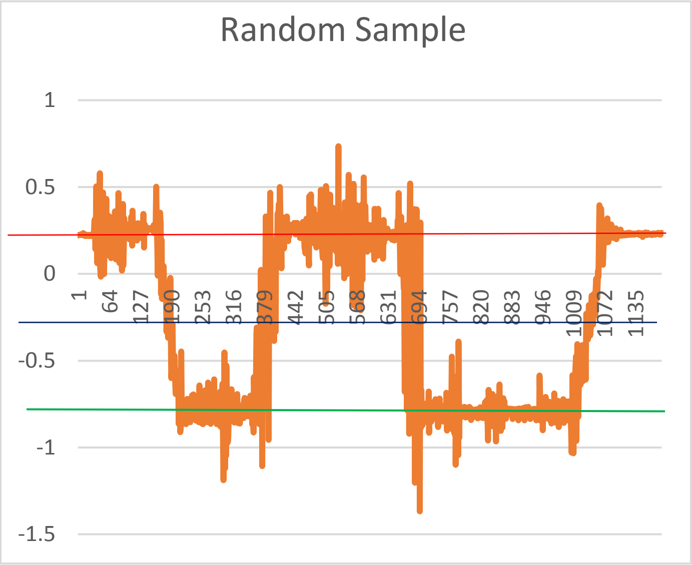


Figure 53. Random sample one - normal

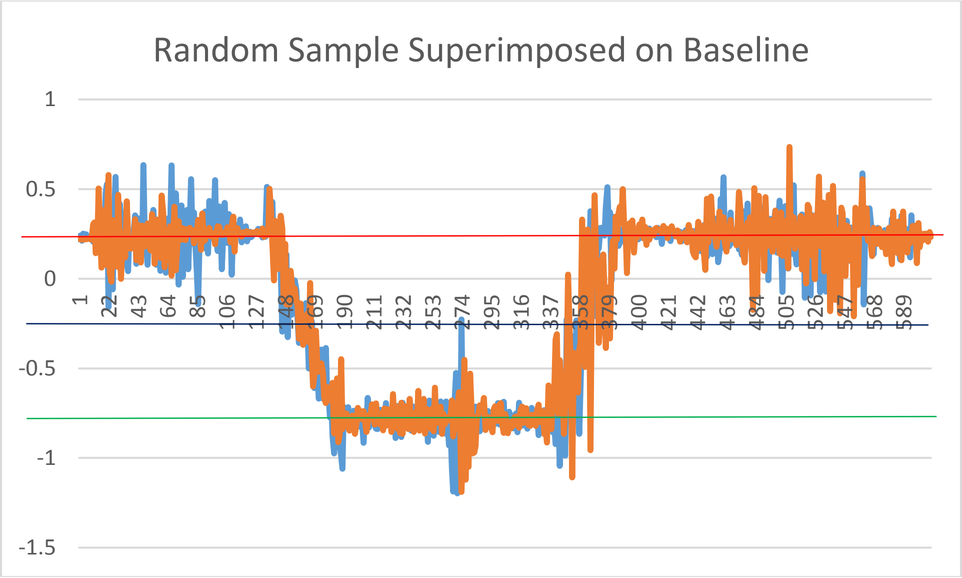


Figure 54. Random sample one data compared with baseline

The second step in calculating risk, required the sampling of real time data consisting of time and voltage (amplitude) values for sound, and or time and inertia values for vibration. Figures 53 shows a random data sample taken during the robotic arm product sampling cycle. This data was obtained on-demand by the user and compared against acceptable baseline values during a defined interval at a specific point in time, relative to the process taking place. Since

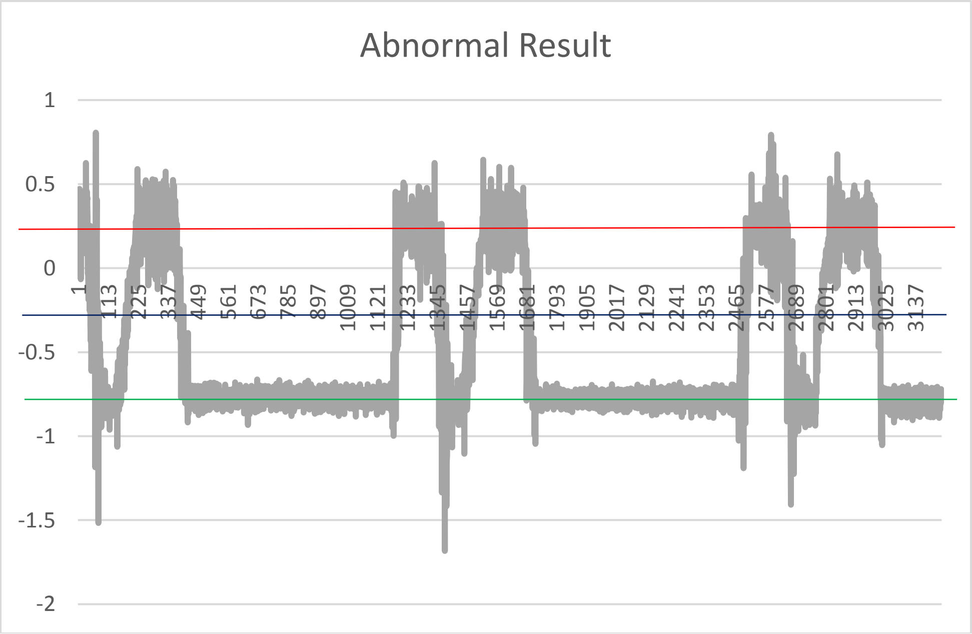


Figure 55. Random sample two – abnormal (malicious)

this was a controlled experiment and the affected asset had been defined, test samples were only taken during the step, or interval, at which point the simulated product sampling took place and a deviation from the baseline was expected to occur. This alleviated the production of extraneous data and the need to explain it as such.

Figures 54 and 56 show a stark contrast in baseline comparisons between sample one, Figure 53, and sample two, Figure 55. In order to make this comparison, a call to the data file, as introduced in Figure 49 earlier, was made while in the process of performing the data capture. This data was then ingested using Apache NiFi.

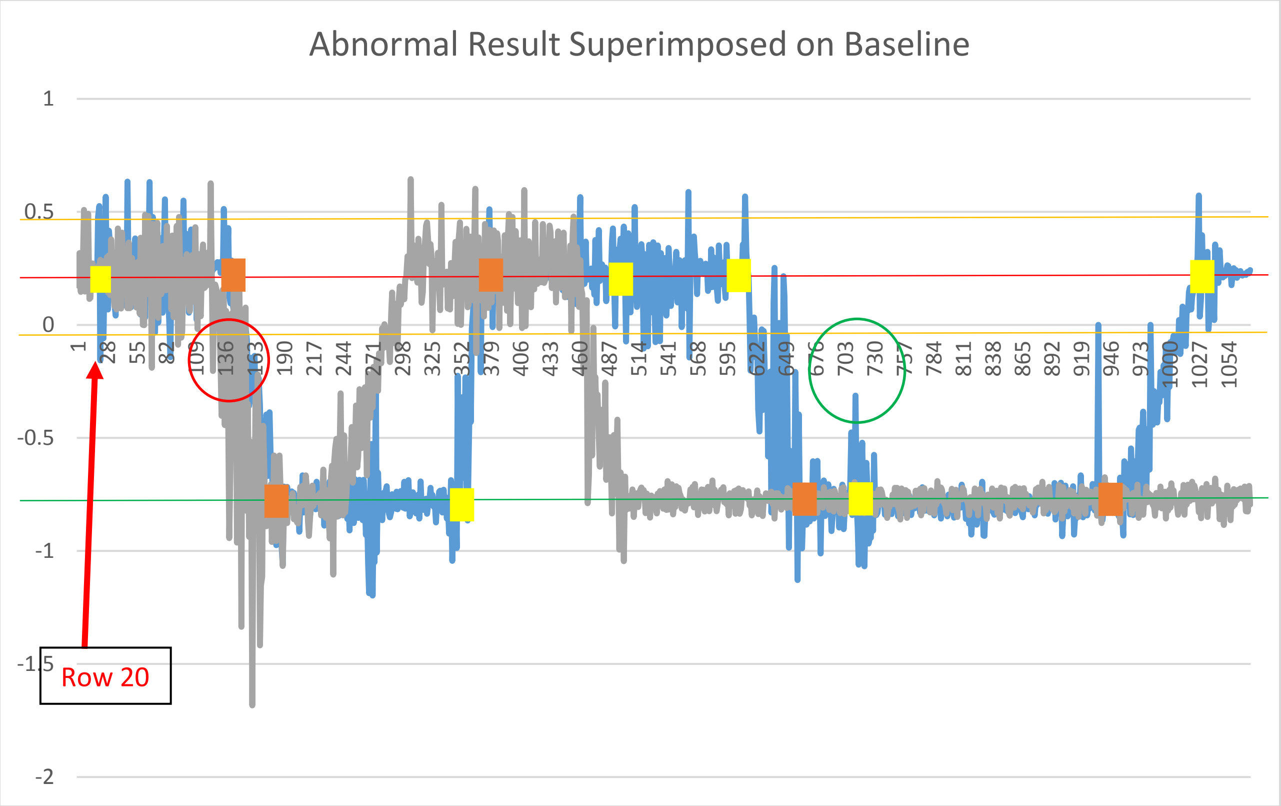


Figure 56. Vibration data showing sample two data deviating from baseline

The third and final step in evaluating and determining a potential risk was accomplished by taking the data ingested with NiFi and processing it using Ambari's Hadoop Distributed File System (HDFS), Hive, incorporated into the Hortonworks Data Platform. Although this platform is Apache based, it integrates with Microsoft Azure, and can be used with Azure's cloud-based capabilities. This operation was performed within the locally segregated sensor network; not in the cloud. A query created in Hive, compared the mean values of pre-defined data points, illustrated as orange and yellow squares, Figure 56, already established in the "baseline" table, to the data recorded in the "live" table at the point in time the data was captured. Comparison times between the baseline and the live data capture were based on relative processing times. This was an instantaneous process.

The mean was calculated using the general mean equation. The variance, likewise, was calculated using the general variance equation. Sample baseline, Figure 52, and random one and two data, Figures 53 and 55, provided in appendix A, B, and C, were used to determine the values for that point in time, as it relates to the data points shown in Figure 56. The data provided in the appendices were calculated, and applied, using the standard mean and variance equations.

Figure 56 identifies data points along the baselines of the X-axis. These data points were used in the query as a comparison reference to the live data. The intervals are noted by line numbers based on data points along both the upper and lower baseline, depicted in the graph as marker numbers, 136 for example, circled in red. These points are located at specific time intervals across the upper and lower baseline. The upper baseline is designated along the horizontal red line, between the upper and lower orange limit lines. The lower baseline limits are not illustrated, however, the lower baseline is, as illustrated by the horizontal green line in Figure 56. These data points represent a small portion of the process. However, they are sufficient for defining strategic points along the process and essentially create a sampling template that can be applied across identical processes.

For the event illustrated in Figure 56, the orange squares represent the match between the running process and the baseline. The yellow squares represent NULL data, or indicate extreme variances such as the one that can be seen just past marker (time reference) 703, circled in green. The first deviation, represented by a yellow square, actually occurs at marker 20, as pointed out by the red arrow. The second deviation identified occurs at marker 352. Although it is clear to see that other deviations appear to have occurred between marker 150 and 170, not represented, and most definitely at marker 243, not represented, there were no data points assigned to capture it.

Findings

Preliminary data analysis was conducted early in the initial phase of the experiment to establish the functional requirements necessary to carry out the automated processes. This initial analysis helped determine the most effective placement of the assets used to perform those processes, and to develop an understanding of the characteristics and types of stimuli each asset produced.

After an initial baseline was developed, the bulk of the data gathered throughout this experiment was generated during the repetitive production cycles that ran for several hours at a time. Several conditioning and commissioning runs were performed during the establishment of the operational baseline, after the development of an initial baseline, prior to capturing the production data used in the comparative analysis process of the risk detection phase.

The research demonstrated that a well-designed and carefully carried out control system production process can be independently monitored in a parallel fashion to its inherent monitoring capability, and that such monitoring can be instrumental in providing a reliable and accurate determination that a potential cyber induced risk to a CI2CS process has occurred. This research involved a CI2CS test-bed that performed a number of automated processes operating in a linear fashion. It was determined that an analysis of data from the random sampling, during the initial baselining, confirmed that multiple tasks performed simultaneously could also be processed and de-synthesized in such a way to enable threat detection and identification of a real-time risk. Such an analysis involved the use of a complex algorithm and sensor implementation scheme. An approach introducing machine learning could operate within the constructs similar to the methodology used in this research.

Summary of the Results and Outcomes of the Research

The entire functional concept of the CI2CSs final design should be understood from start to finish when considering the implementation of an independent parallel IT/OT risk monitoring solution. In today's IoT universe, many of the filters necessary to help define the anticipated spatial environment for CI2CS facilities can be ascertained with the vast amounts of data readily available, most of it accessible in real-time, or pre-scheduled, and they can be factored in ahead of time, prior to the production run-time. This would include information from a number of sources such as transportation, atmospheric activity, or local events. Specifically, it would take into consideration factors such as flight paths and schedules of commercial travel or logistical operations, including train and trucking schedules, weather reports, types of local events that would suggest higher than normal commuter activity along adjacent transportation corridors, agricultural activity (for facilities in rural areas), and other local industry activity that might have an impact on monitoring equipment situated within those settings. This is not unrealistic when comparing the orders of magnitude involved in tuning or conditioning an extremely stimuli-rich environment to one that is stimuli-poor, such as that created in a hydro facility during generation or pumping operations to that of a small pharmaceutical manufacturer respectively.

In summary, this research demonstrated that the security resilience of an IT/OT system cannot exclusively be comprised of a cyber-centric solution where the reliability and dependability of an entire security layer is placed in a potentially compromised IT/OT system's all-inclusive network. A compromised IT/OT network could be laden with vulnerabilities, as a result of defunct security controls, and/or corrupted processes. Stuxnet is a prime example of where the control system network was compromised, unbeknownst to operators, as the HMI did not indicate any issues with the centrifuge, yet a few of the centrifuges were self-destructing.

CI2CS IT/OT security is far different than conventional IT security alone; therefore, outside-the-box thinking of different defense-in-depth strategies are required in order to meet the challenges and nuances stemming from such an environment. The conventional COTS security controls deployed in an IT setting are not necessarily suited for an OT/IT environment as is. In fact, this research demonstrated that the automated asset(s) within the OT process itself can be deployed or used to complement a sensor within the system. An asset's very behavior and performance characteristics within an OT environment indicate a certain state of normalcy if consistent with previously established performance indicators; until it does not, thereby, requiring some type of investigation as to its cause. In this approach to assessing CI2CS risk, essentially each asset with a dynamic purpose composes a single element of the entire risk detection network. The community can benefit from this research by re-thinking the conventional IT/OT security implementation and looking past the limited IT capabilities surrounding the security implications present when marrying an IT security solution with an OT problem.

Chapter 5

Conclusions, Implications, Recommendations, and Summary

Introduction

Many of the industries designated as United States CI by the DHS are under constant cyber-attacks by threat actors from various nation states whose primary intent and centralized focus is on disrupting the American way-of-life, or as a means of testing our resiliency to such attacks should they be required toward a greater detriment in the future. In either case, the motive is sinister and the results could be catastrophic. The industries that are part of our CI, whether regulated or not, are expected to possess the capability of being able to protect themselves against the type of cyber-attacks anticipated, and in-fact are rapidly emerging, in the Twenty First Century. Internal bad-actors and supply chain integrity are two of today's more notable threats.

This research has developed and demonstrated a protection capability for defending against such an attack by taking an outside-the-box approach for detecting, assessing, and then determining the probability of real-time CI2CS cyber risk. Instead of subscribing to the conventional practices currently employed, and accepting the limitations they face, a novel and real-time method for performing CI2CS risk assessments was developed. This approach aims to confront those type of Twenty First Century threats; particularly as supply chain concerns persist and are treated with such high importance and current regulatory intervention.

Conclusions

The research conducted illustrates that linear processes functioning within a control system environment can benefit from having an independent and segregated sensory network performing parallel monitoring functions of those processes for the purpose of determining real-time risk probabilities by continuously evaluating those processes against previously established baseline data. It also supports that by adding additional monitoring tools and developing a more robust algorithm for determining risk probabilities, such as HMMs, this model can be expanded to take into account the often-simultaneous processes that occur during production cycles, as well as, the capability to recognize and filter out undesirable and unnecessary stimuli.

Research Problem

CI2CS currently lack the capability for effectively and independently monitoring real-time vulnerabilities across their critical cyber assets because of their unique architecture, sensitive and volatile environments, and real-time physical and critical processes. As such, systemic and/or operational failure, causing both a safety and reliability concern, exist while conducting vulnerability assessments. Therefore, as a precaution, CI2CS vulnerability assessments that have been conducted and performed in a laboratory environment, with the resulting states compared to that of a live environment, have failed to observe the various dynamic states and conditions that some ICS, such as generating facilities and critical manufacturing, continue to operate under due to the constraints and limitations of a simulated ICS environment. This includes, but is not limited to, network noise, number and device types, varying temperatures, sounds, vibrations, and networking schemes. This research took into account the various dynamic states of a control system environment running various processes and devices, and incorporated an independent and passive real-time method for monitoring and alerting to a systemic threat within an operational CI2CS environment by considering the inclusion of stimuli emitted during its operation as part of the risk detection equation.

Research Goal

The primary goal of this research was achieved by developing an approach that took into account the unique architecture, sensitive and volatile environments, and real-time physical and critical processes of a CI2CS environment in identifying and alerting an operator to a real-time suspected or actual risk across a network of critical cyber assets. Alternatively, because it is crucial that threat and vulnerability detection is done in a timely fashion, this research increased the ability to discover and mitigate vulnerabilities or threats that might have otherwise gone unnoticed for a prolonged period of time until an off-line assessment could have been conducted.

Research Contribution

The major contribution of this research pivots around the unique approach of separating a cyber risk monitoring network from the control system network and the security controls operating within the control system network. If a control system network becomes compromised, it could very well render the security controls affiliated with that network ineffective. Just as safety zones, or networks, operate in a completely disconnected and independent state from a control system network, a risk monitoring network running in parallel to the control system processes, but disconnected from the control system network, provides a viable and reliable solution for recognizing and identifying risks potentially associated with cyber related compromises. Another contribution of the research is in the area of ICS from an OT perspective, coupled with that of the real-time processing capabilities of "big data", and using data analytic platforms on the IT side of the business to assess and determine the potential presence of risk.

Implications

The intent and purpose of this research was not to focus on any one particular CI industry, but instead its aim was to include all industries utilizing automated processes where control systems could find themselves exposed to vulnerabilities. The steps set forth in this research can make an immediate impact on smaller industries engaging automation by offering a simple, yet cost effective, solution and using relatively inexpensive sensors, sensor controllers, and open source software that allows for real-time risk analysis, monitoring, and detection.

Of particular concern in today's threat landscape is the possibility that malware is purposefully and deliberately being introduced into control systems, and/or the components from which they are made, at some point through the supply chain as they evolve from a chip and into a system. This deliberate embedment of malware or device tampering, if such a thing were to occur, must be for the purpose of corrupting processes, permitting back door access, or modifying configuration settings that simply render the asset useless or unreliable either immediately upon implementation or at some other point in time. There is nothing to suggest that an act such as this has not already been accomplished prior to, or following, Juniper's firewall hack of 2015 (Wired, 2015).

Although the implication of such threats may be nothing more than a ruse to heighten industry's sensitivity to cyber security awareness, it has the potential for creating paranoia and lowering the level of confidence towards vendors who develop and build COTS IT solutions (designed and manufactured in good faith) for the express purpose of protecting and defending critical assets, by preventing such attacks from occurring. Between vendor uncertainty and increasing compliance requirements brought on by an overreaching regulatory agency's strict cyber-security standards, the ability to conduct an independent passive real-time risk assessment is valuable.

Some of the CI identified are involved and engaged in real-time processing, such as, the gas, electricity, and manufacturing sectors. All of the industries in these sectors are in need of a real-time risk detection solution that does not interfere with the processes taking place. This research designed just such a solution in that it provided real-time parallel monitoring of control system processes, so that any baseline deviations occurring during the production cycle were detected, analyzed and determined to be either potential risks or anomalies within the linear process.

Constraints and Limitation of the Study

Limitations

The study, albeit successful in the detection and determination of risk, had a few constraints worthy of mention prior to the start of any future research using this test-bed design as a template. The equipment used in this research, with exception of the Allen-Bradley products, were not explicitly rated for "industrial" use. This resulted in continuous physical monitoring throughout each production cycle to ensure that the functional dependability, capacity, and integrity required throughout the experiment was maintained during the process. Because of this limitation, the test-bed was designed to operate within specified time parameters.

Based upon off-line and component specific experimentation, as different devices were evaluated for test-bed use, this time constraint was set between 10 and 20 minute intervals per production cycle, so that certain mechanical parts were not stressed to failure. This was particularly important with the robotic arm where the initial installation and early operation of setting up the experiment, prior to baselining, resulted in the destruction of three servo motors. However, these issues helped formulate the rationale which decidedly led to determining the attack vector, and how such an attack might be accomplished.

The test-bed's actual, physical placement limited the more extreme temperature swings a facility would endure if it were located outside and exposed to the natural environment. This could include rain, a thunderstorm, and or lightning, which could contribute to the internal sound stimuli. Extreme heat or cold might involve cycling interior environmental conditioning systems such as a forced air furnace or air condition. This research observed peculiar thermal patterns at the onset of each experiment because of air currents caused as a result of fluid transfer. The air eventually normalized as subsequent processes were run.

Some process monitoring equipment, particularly the thermal imaging camera, lacked certain features that would have made it more practical for use as part of the spatial sensory module. One feature of interest would have been the capability to mask out certain zones within the test-bed, as to provide a clearer and more defined thermal signature. Surprisingly, the results from the thermal imager were not quite what was expected. Despite the thermal camera sensing the dynamic changes in temperatures from several sources, some details were lost from view as external stimuli altered a portion of the image; thereby, rendering that particular portion of the results unusable. This only occurred during the initial two production cycles and throughout approximately half of the third, and only affected a portion of the test-bed's climate. The result, however, demonstrated an inconsistency and untimely irregularity in the readings, and; therefore, was only used as an "event" and not a "risk" monitoring sensor. An event sensor was used for general purpose process monitoring.

Delimitations

The test-bed was designed as an open-loop versus a closed loop system. This allowed direct and dedicated monitoring of test-bed processes without interference from the controller's proportional, integral, and derivative (PID) functionality disrupting the logging of sensory data, as a result of applying process corrective measures. This was explained in more detail in chapter four. Essentially a closed loop system has the capability to make processing adjustments relative to sensory input. This capability would be useful and expected in a non-linear type of experiment.

The test-bed testing intervals for the product cycles were under 15 minutes each. Because the product cycles ran for a short time in general, the changes in temperature to the overall environment, as a result of machine activity, were only nominal. For example, the spatial and robotic arm, servo-two, temperature data was recorded, but the temperature data was not used in the risk-baseline comparison algorithm due to only observing a nominal difference between each production cycle. In other words, the automated activity was not demanding enough to have an effect on the test-bed's interior climate.

There were some delimitations resulting from software usage agreements. Testing was carefully coordinated, so that it was conducted at such a time where the KepServerEx, and NI Sound and Vibration Toolkit software provided maximum benefit. Both applications were trial versions that had 30 day limitations; although, NI Sound and Vibration Toolkit did offer an extension of up to 45 days, upon request, and KepServerEx still provided all of its features after the 30-day trial period; albeit at two hour intervals. Kepware eventually extended their trial version to one year. Overall, this had minimal impact on this experiment, as each production cycle was under 15 minutes; therefore, allowing the experiment to perform eight complete cycles prior to the two-hour expiration.

Hardware and software compatibility created another delimitation and prevented at least one preconceived system to be replaced. The Raspberry Pi, first mentioned in chapter three, which was intended to be used for capturing and processing SSM data, was not capable of hosting certain, ideally suited software preferred for data logging. This caused the need to replace the Raspberry Pi system with a system having an Intel based core processor.

Recommendations for Additional Research

Future research could include a similar configuration recreated in a real operating environment; minus the actual exploitation or introduction of malware in manipulating the control system. Then manually applying and deliberately modifying baseline data so that the deviations are used to simulate threats in demonstrating the sensory perception module, and corresponding comparative analysis application, recognize and alert to the anomaly or potential threat. This could cause an operating and performance condition similar to that of what the Stuxnet malware was designed to do, and in fact succeeded in doing, at Iran’s uranium enrichment facility.

Because this research focused on a linear-based processing environment, multi, and or simultaneous-process risk determination was not calculated, as there was one ongoing process at any one time during the production cycle. Future research could take into account multiple processes occurring simultaneously and the necessity for applying machine learning and pattern recognition in order to process and recognize anomalies or inconsistencies in such instances where they were discovered. A strong knowledgebase in CI industry remains a critical area of concern in performing and adhering to cyber-security best practices.

Summary

The ability to continuously monitor and recognize real-time risk(s) in ICS without compromising the performance or processing taking place at a facility/plant is a crucial cyber-security control we must consider in today’s geopolitical landscape where nation state bad-actors, alongside that of the common hacker attempt to take advantage of the vulnerabilities resulting from IT’s integration into the OT universe and then capitalize on those vulnerabilities by creating or delivering the exploits necessary to destroy it. Being unable to run vulnerability scans in a real-time processing environment has challenged industry to find other methods for discovering and determining real-time risks to CI2CS. One of these methods involves the creation of an independent sensory zone that sits beside the process automation sensors running throughout the control system, and then applying an independent parallel process for monitoring those sensors. This process then compares the asset’s functional data to that of the asset’s environmental sensory performance data, which is specifically conditioned to the spatial environment in which it and the devices it monitors reside; thereby, determining the probability of whether or not a real-time risk exists within that environment and, if so, indicating the affected asset.

Appendix A

Baseline Accelerometer Data Sample from Robotic Arm Interval

0.244	0.222	0.271	0.206	0.265	-0.669
0.228	0.234	0.354	0.033	0.278	-0.626
0.218	0.229	0.083	-0.142	0.271	-0.396
0.229	0.209	0.337	0.189	0.276	-0.668
0.229	0.229	0.325	0.27	0.23	-0.629
0.229	0.222	0.126	0.212	0.281	-0.386
0.225	0.24	0.147	0.171	0.234	-0.621
0.238	0.213	0.634	0.375	0.513	-0.633
0.221	0.252	0.14	0.193	0.354	-0.774
0.22	0.233	0.191	0.141	0.299	-0.675
0.23	0.248	0.078	0.434	0.275	-0.878
0.221	0.225	0.306	0.229	0.428	-0.974
0.226	0.226	0.215	0.254	0.125	-0.799
0.214	0.225	0.196	0.271	0.095	-0.847
0.229	0.222	0.383	0.55	0.285	-0.811
0.217	0.206	0.256	0.213	0.098	-0.92
0.22	0.213	0.078	0.155	0.063	-0.989
0.218	0.229	0.301	0.403	0.065	-1.06
0.228	0.202	0.214	0.281	-0.295	-0.715
0.221	0.197	0.045	0.054	0.061	-0.752
0.226	0.259	0.118	0.272	-0.052	-0.767
0.22	0.23	0.31	0.423	0.135	-0.716
0.22	0.264	0.242	0.209	-0.326	-0.813
0.243	0.194	0.033	0.2	-0.085	-0.776
0.209	0.478	0.335	0.361	-0.065	-0.791
0.227	0.526	0.332	0.335	-0.2	-0.736
0.221	-0.157	0.234	0.023	-0.121	-0.77
0.223	0.4	0.632	0.315	-0.009	-0.743
0.23	0.381	0.144	0.296	-0.063	-0.819
0.23	-0.059	0.242	0.289	-0.236	-0.765
0.217	0.311	0.477	0.242	-0.358	-0.828
0.215	0.567	0.365	0.244	-0.251	-0.791
0.232	0.102	-0.033	0.33	-0.145	-0.915
0.236	0.068	0.166	0.203	-0.338	-0.78
0.218	0.417	0.408	0.242	-0.279	-0.814
0.219	0.346	0.01	0.257	-0.137	-0.665
0.221	0.134	0.269	0.293	-0.307	-0.83
0.215	0.147	0.104	0.209	-0.5	-0.724
0.21	0.322	0.384	0.246	-0.241	-0.766
0.205	0.097	0.054	0.262	-0.236	-0.741
0.221	0.042	0.318	0.232	-0.359	-0.817
0.215	0.252	0.556	0.244	-0.61	-0.837
0.193	0.247	0.194	0.25	-0.425	-0.839
0.224	0.237	0.369	0.253	-0.499	-0.794

-0.731	-0.793	-0.687	-0.987	0.288	0.237
-0.756	-0.735	-0.846	-0.225	0.267	0.255
-0.736	-0.864	-0.811	-0.646	0.264	0.215
-0.815	-0.685	-0.827	-0.758	0.206	0.265
-0.694	-0.788	-0.741	-0.342	0.2	0.25
-0.796	-0.929	-0.782	-0.32	0.266	0.268
-0.749	-0.779	-0.767	-0.644	0.249	0.198
-0.818	-1.057	-0.834	-0.677	0.272	0.233
-0.775	-1.186	-0.76	-0.232	0.219	0.211
-0.835	-0.707	-0.751	-0.705	0.27	0.154
-0.749	-0.526	-0.857	-0.865	0.229	0.201
-0.888	-1.197	-0.692	-0.599	0.279	0.299
-0.71	-0.895	-0.798	-0.266	0.257	0.303
-0.779	-1.049	-0.805	-0.429	0.276	0.123
-0.708	-0.227	-0.819	-0.018	0.252	0.184
-0.883	-1.007	-0.777	0.062	0.224	0.209
-0.692	-0.654	-0.769	0.205	0.28	0.237
-0.763	-0.799	-0.836	-0.014	0.236	0.138
-0.814	-0.76	-0.791	0.377	0.268	0.175
-0.711	-0.808	-0.786	0.024	0.244	0.335
-0.806	-0.863	-0.721	0.265	0.263	0.321
-0.68	-0.743	-0.795	0.129	0.232	0.092
-0.847	-0.786	-0.752	-0.209	0.284	0.145
-0.711	-0.734	-0.804	0.182	0.27	0.444
-0.845	-0.714	-0.782	0.169	0.26	0.372
-0.75	-0.805	-0.873	0.283	0.194	0.566
-0.827	-0.834	-0.779	0.249	0.269	0.167
-0.793	-0.851	-0.796	0.288	0.263	0.237
-0.8	-0.828	-0.738	0.241	0.262	0.203
-0.802	-0.749	-0.733	0.398	0.236	0.181
-0.818	-0.799	-0.789	0.511	0.257	0.369
-0.703	-0.796	-0.829	0.289	0.21	0.199
-0.858	-0.813	-0.729	0.371	0.242	0.24
-0.76	-0.738	-0.799	-0.009	0.241	0.22
-0.909	-0.816	-0.761	0.347	0.247	0.229
-0.679	-0.761	-0.857	0.322	0.248	0.249
-0.869	-0.774	-0.786	0.249	0.243	0.164
-0.702	-0.735	-0.923	0.348	0.237	0.2
-0.876	-0.79	-0.862	0.184	0.262	0.134
-0.683	-0.808	-0.686	0.203	0.209	0.376
-0.841	-0.693	-1.044	0.274	0.227	0.097
-0.768	-0.813	-0.889	0.221	0.256	0.262
-0.854	-0.779	-0.706	0.247	0.271	0.42
-0.685	-0.787	-0.723	0.196	0.225	0.194

0.22	0.355	0.137	0.569	-0.783	-0.77
0.268	0.334	0.269	0.319	-0.799	-0.706
0.213	0.237	0.245	0.257	-0.208	-0.904
0.253	0.055	0.284	0.211	-0.627	-0.834
0.254	0.366	0.234	0.03	-1.128	-0.773
0.105	0.081	0.232	-0.202	-0.396	-0.816
0.223	-0.076	0.225	-0.016	-0.823	-0.742
0.201	0.203	0.247	0.026	-0.906	-0.818
0.2	0.352	0.271	0.033	-0.989	-0.739
0.158	0.12	0.253	-0.068	-0.847	-0.475
0.37	-0.072	0.151	-0.144	-0.711	-0.841
0.427	0.318	0.266	-0.095	-0.878	-0.821
0.217	0.064	0.166	-0.375	-0.883	-0.757
-0.007	0.202	0.334	-0.472	-0.793	-0.312
0.333	0.293	0.167	-0.195	-0.716	-0.809
0.265	0.297	0.299	-0.019	-0.87	-0.898
0.175	0.041	0.191	-0.377	-0.851	-1.061
0.319	0.166	0.278	-0.026	-0.817	-0.718
0.381	0.341	0.169	-0.237	-0.603	-0.734
0.147	0.222	0.106	-0.293	-0.857	-0.521
0.332	0.084	0.352	-0.257	-0.747	-0.767
0.435	0.132	0.201	-0.582	-0.714	-1.068
0.187	0.2	0.18	-0.341	-0.602	-0.839
-0.073	0.181	0.186	-0.207	-0.848	-0.61
0.259	0.245	0.276	-0.829	-0.748	-0.75
0.405	0.323	0.192	-0.67	-1.011	-0.967
0.206	0.099	0.12	0.095	-0.76	-0.94
0.056	0.264	0.354	0.252	-0.807	-0.942
0.383	0.286	0.187	-0.641	-0.688	-0.707
0.18	0.273	0.229	-0.727	-0.845	-0.91
0.139	-0.009	0.212	-0.781	-0.81	-0.574
0.521	0.221	0.229	0.009	-0.833	-0.831
0.161	0.277	0.228	-0.64	-0.705	-0.841
0.08	0.286	0.229	-0.521	-0.822	-0.829
0.329	0.284	0.222	-0.779	-0.791	-0.788
0.314	0.072	0.219	0.216	-0.912	-0.733
0.288	0.588	0.232	0.127	-0.836	-0.778
0.358	-0.144	0.236	-0.271	-0.725	-0.793
0.295	0.232	0.235	-0.045	-0.787	-0.795
-0.105	0.234	0.217	-0.498	-0.817	-0.778
0.313	0.291	0.255	-0.985	-0.827	-0.84
0.276	0.392	0.243	-0.556	-0.777	-0.796
0.296	0.318	0.356	-0.619	-0.896	-0.782
-0.094	0.25	0.283	-0.749	-0.833	-0.801

-0.78	-0.79	-0.779	-0.816	-0.761	-0.838
-0.81	-0.79	-0.778	-0.842	-0.851	-0.546
-0.837	-0.786	-0.77	-0.778	-0.797	-0.65
-0.754	-0.857	-0.78	-0.797	-0.737	-0.841
-0.794	-0.794	-0.804	-0.779	-0.803	-0.622
-0.793	-0.782	-0.779	-0.778	-0.843	-0.603
-0.779	-0.774	-0.823	-0.777	-0.793	-0.795
-0.788	-0.801	-0.83	-0.78	-0.813	-0.669
-0.847	-0.801	-0.84	-0.752	-0.738	-0.464
-0.806	-0.847	-0.791	-0.808	-0.794	-0.455
-0.791	-0.751	-0.76	-0.766	-0.704	-0.772
-0.78	-0.822	-0.77	-0.781	-0.771	-0.52
-0.801	-0.788	-0.783	-0.788	-0.821	-0.383
-0.776	-0.8	-0.782	-0.835	-0.811	-0.539
-0.834	-0.815	-0.773	-0.813	-0.794	-0.602
-0.762	-0.852	-0.818	-0.787	0	-0.753
-0.847	-0.834	-0.759	-0.773	-0.766	-0.461
-0.809	-0.773	-0.78	-0.762	-0.772	-0.468
-0.754	-0.792	-0.775	-0.813	-0.845	-0.64
-0.798	-0.773	-0.822	-0.7	-0.864	
-0.841	-0.757	-0.811	-0.861	-0.796	-0.441
-0.794	-0.809	-0.784	-0.753	-0.759	-0.442
-0.778	-0.707	-0.786	-0.713	-0.796	-0.293
-0.811	-0.76	-0.79	-0.722	-0.802	-0.306
-0.782	-0.729	-0.771	-0.833	-0.771	-0.42
-0.775	-0.789	-0.803	-0.698	-0.819	-0.432
-0.798	-0.858	-0.819	-0.934	-0.775	-0.34
-0.79	-0.785	-0.832	-0.852	-0.81	-0.343
-0.831	-0.771	-0.766	-0.784	-0.716	-0.208
-0.813	-0.879	-0.765	-0.711	-0.79	-0.312
-0.718	-0.831	-0.775	-0.808	-0.792	-0.269
-0.768	-0.928	-0.817	-0.847	-0.81	-0.178
-0.823	-0.839	-0.819	-0.747	-0.735	-0.253
-0.795	-0.797	-0.83	-0.836	-0.836	-0.091
-0.778	-0.742	-0.851	-0.778	-0.847	-0.418
-0.822	-0.82	-0.781	-0.77	-0.675	-0.207
-0.783	-0.884	-0.771	-0.714	-0.824	-0.275
-0.779	-0.739	-0.792	-0.928	-0.865	-0.076
-0.774	-0.856	-0.802	-0.834	-0.931	-0.124
-0.794	-0.765	-0.815	-0.855	-0.603	-0.075
-0.806	-0.85	-0.757	-0.754	-0.618	-0.048
-0.843	-0.694	-0.788	-0.813	-0.8	-0.124
-0.742	-0.931	-0.78	-0.814	-0.617	-0.092
-0.758	-0.933	-0.803	-0.666	-0.659	-0.11

-0.122	0.238
0.022	0.264
-0.129	0.261
-0.073	0.229
0.081	0.242
-0.032	0.176
-0.023	0.224
-0.093	0.251
0.075	0.22
0.063	0.242
0.146	0.209
0.165	0.219
0.069	0.236
0.125	0.239
0.152	0.222
0.187	0.218
0.22	0.225
0.24	0.224
0.3	0.221
0.573	0.233
0.037	0.236
0.123	0.229
0.397	0.243
0.22	
0.24	
0.262	
0.311	
0.324	
-0.021	
0.051	
0.131	
0.261	
0.185	
0.022	
0.356	
0.207	
0.292	
0.186	
0.33	
0.25	
0.217	
0.21	
0.226	
0.217	

Appendix B

Random Accelerometer Data Sample (Normal) from Robotic Arm Interval

0.222	0.579	0.292	0.296	-0.076	-0.865
0.224	0.032	0.018	0.346	-0.055	-0.805
0.223	-0.017	0.179	0.151	-0.123	-0.72
0.223	0.326	0.402	0.274	-0.156	-0.74
0.225	0.252	0.047	0.282	-0.334	-0.783
0.227	0.207	0.16	0.254	-0.138	-0.741
0.223	0.068	0.321	0.255	-0.283	-0.739
0.223	0.468	0.249	0.236	-0.368	-0.788
0.224	0.188	0.217	0.258	-0.267	-0.854
0.223	-0.001	0.282	0.25	-0.207	-0.805
0.234	0.229	0.244	0.263	-0.193	-0.759
0.226	0.258	0.216	0.264	-0.354	-0.762
0.225	0.12	0.276	0.257	-0.023	-0.7
0.22	0.432	0.263	0.257	-0.419	-0.764
0.218	0.246	0.284	0.258	-0.601	-0.814
0.228	0.17	0.253	0.257	-0.54	-0.695
0.22	0.167	0.267	0.258	-0.29	-0.802
0.219	0.228	0.198	0.258	-0.457	-0.758
0.221	0.193	0.273	0.266	-0.519	-0.814
0.22	0.332	0.237	0.255	-0.51	-0.749
0.218	0.103	0.328	0.247	-0.472	-0.846
0.224	0.301	0.288	0.256	-0.524	-0.724
0.22	0.234	0.163	0.28	-0.664	-0.821
0.224	0.091	0.364	0.234	-0.693	-0.728
0.22	0.296	0.21	0.258	-0.604	-0.746
0.218	0.205	0.215	0.251	-0.612	-0.748
0.225	0.18	0.266	0.245	-0.667	-0.854
0.219	0.196	0.273	0.502	-0.682	-0.643
0.223	0.308	0.284	0.37	-0.581	-0.811
0.222	0.227	0.237	0.194	-0.864	-0.712
0.226	0.158	0.238	-0.01	-0.66	-0.837
0.221	0.36	0.254	0.323	-0.556	-0.76
0.217	0.322	0.193	0.224	-0.912	-0.862
0.304	0.084	0.256	0.055	-0.875	-0.691
0.314	0.198	0.292	0.063	-0.448	-0.865
0.142	0.33	0.289	0.352	-0.838	-0.737
0.18	0.142	0.244	0.282	-0.781	-0.711
0.504	0.116	0.246	-0.116	-0.751	-0.761
0.268	0.465	0.247	0.194	-0.752	-0.838
0.062	0.364	0.258	-0.144	-0.826	-0.671
0.364	0.064	0.264	-0.086	-0.847	-0.824
0.383	0.15	0.287	-0.146	-0.783	-0.767
0.077	0.271	0.261	0.044	-0.763	-0.855
0.346	0.208	0.204	-0.027	-0.787	-0.745

-0.839	-0.757	-0.799	0.028	0.294	0.198
-0.626	-0.706	-0.786	0.136	0.256	0.152
-0.848	-0.734	-0.738	-0.078	0.28	0.231
-0.709	-0.717	-0.867	-0.386	0.227	0.272
-0.765	-0.665	-0.829	-0.049	0.267	0.329
-0.671	-0.843	-0.914	-0.208	0.279	0.22
-0.864	-0.793	-0.595	-0.333	0.293	0.272
-0.725	-0.771	-0.73	0.134	0.267	0.266
-0.824	-0.781	-0.838	0.212	0.263	0.188
-0.788	-0.822	-0.647	0.057	0.235	0.302
-0.836	-0.856	-0.687	0.347	0.269	0.483
-0.726	-0.727	-0.34	0.221	0.254	0.378
-0.85	-0.712	-0.31	0.367	0.206	0.191
-0.608	-0.78	-0.752	0.402	0.243	0.14
-0.807	-0.794	-0.454	0.311	0.25	0.091
-0.723	-0.703	-0.564	0.5	0.255	0.26
-0.766	-0.693	-0.589	0.311	0.258	0.314
-0.714	-0.808	-0.865	0.259	0.204	0.252
-0.866	-0.86	-0.73	0.032	0.299	0.342
-0.753	-0.755	-0.471	0.255	0.251	0.136
-0.811	-0.778	0.023	0.31	0.265	-0.173
-0.786	-0.862	-0.229	0.305	0.152	0.505
-0.792	-0.821	-0.524	0.324	0.132	0.325
-0.769	-0.835	-1.108	0.146	0.119	-0.087
-0.788	-0.796	-0.414	0.242	0.222	0.462
-0.679	-0.821	-0.53	0.226	0.318	0.185
-0.88	-0.77	-0.536	0.279	0.171	0.358
-0.83	-0.768	-0.634	0.303	0.22	0.351
-0.836	-0.773	-0.399	0.245	0.295	0.044
-0.732	-0.831	-0.354	0.33	0.131	0.455
-0.633	-0.736	0.331	0.225	0.049	0.181
-0.847	-0.714	-0.183	0.289	0.447	0.21
-1.189	-0.804	-0.491	0.189	0.202	0.269
-0.9	-0.751	0.062	0.274	0.27	0.327
-0.453	-0.723	0.275	0.217	0.458	0.151
-1.121	-0.797	-0.059	0.271	0.317	0.218
-0.653	-0.817	-0.956	0.263	0.263	0.381
-1.049	-0.82	0.099	0.242	0.225	0.283
-0.529	-0.759	0.049	0.278	0.178	0.295
-0.709	-0.793	0.466	0.283	0.25	0.141
-0.974	-0.758	0.098	0.272	0.374	0.369
-0.968	-0.821	0.106	0.268	0.232	0.129
-0.933	-0.729	-0.358	0.275	0.278	0.184
-0.772	-0.702	-0.249	0.258	0.342	0.323

0.204	0.287	0.198	0.003	-0.826	-0.774
0.058	0.104	0.225	0.278	-0.829	-0.799
0.735	0.244	0.295	0.055	-0.759	-0.686
0.197	0.233	0.222	-0.084	-0.747	-0.902
0.193	-0.21	0.296	0.036	-0.838	-0.767
0.342	0.397	0.087	-0.68	-0.783	-0.787
0.168	0.025	0.311	-0.257	-0.766	-0.79
0.217	-0.035	0.177	-0.41	-0.855	-0.78
0.209	0.386	0.251	-0.292	-0.854	-0.783
0.29	0.305	0.214	0.28	-0.784	-0.869
0.35	0.556	0.25	-0.785	-0.677	-0.722
0.052	0.251	0.252	-0.21	-0.915	-0.771
0.156	0.353	0.235	-0.249	-0.809	-0.879
0.336	0.392	0.206	0.286	-0.735	-0.477
-0.026	0.213	0.26	-0.921	-0.779	-0.682
0.412	0.261	0.234	0.279	-0.851	-0.899
0.396	0.2	0.237	0.52	-0.751	-0.784
0.13	0.127	0.209	-0.481	-0.811	-0.812
0.282	0.283	0.241	-0.398	-0.778	-0.918
0.349	0.271	0.236	-0.076	-0.803	-0.923
-0.002	0.252	0.217	-0.123	-0.77	-0.67
0.332	0.194	0.205	-0.913	-0.826	-1.1
0.313	0.281	0.22	-0.098	-0.835	-0.591
0.57	0.222	0.215	-0.271	-0.815	-0.762
0.182	0.177	0.196	-0.405	-0.723	-0.797
0.311	0.256	0.242	0.373	-0.809	-0.803
0.425	0.197	0.21	-1.204	-0.836	-1.045
0.041	0.137	0.218	-0.799	-0.766	-0.391
0.321	0.167	0.193	-1.2	-0.754	-0.946
0.138	0.283	0.244	0.371	-0.883	-0.825
0.467	0.178	0.225	-0.377	-0.797	-0.763
-0.181	0.082	0.242	-0.244	-0.742	-0.814
0.126	0.375	0.228	-1.001	-0.718	-0.785
0.518	0.158	0.203	-0.359	-0.847	-0.802
0.235	0.208	0.228	-0.649	-0.753	-0.823
-0.119	0.145	0.308	-0.538	-0.829	-0.776
0.273	0.319	0.302	-1.368	-0.822	-0.78
0.085	0.142	0.464	0.295	-0.787	-0.781
-0.194	0.103	0.199	-0.888	-0.764	-0.814
0.397	0.374	0.315	-0.67	-0.852	-0.766
0.366	0.2	0.02	-0.816	-0.822	-0.825
0.119	0.248	-0.001	-0.873	-0.813	-0.778
0.016	0.146	0.328	-0.777	-0.844	-0.828
0.297	0.25	0.144	-0.819	-0.795	-0.803

-0.773	-0.669	-0.778	-0.781	-0.837	-0.702
-0.779	-0.722	-0.79	-0.801	-0.812	-0.546
-0.819	-0.76	-0.778	-0.8	-0.689	-0.475
-0.776	-0.853	-0.782	-0.813	-0.838	-0.961
-0.794	-0.879	-0.779	-0.778	-0.857	-0.602
-0.766	-0.812	-0.837	-0.789	-0.78	-0.405
-0.792	-0.831	-0.782	-0.779	-0.814	-0.602
-0.79	-0.754	-0.78	-0.789	-0.798	-0.823
-0.842	-0.692	-0.792	-0.81	-0.791	-0.606
-0.778	-0.715	-0.791	-0.843	-0.731	-0.637
-0.832	-0.764	-0.798	-0.78	-0.758	-0.47
-0.797	-0.722	-0.811	-0.787	-0.883	-0.64
-0.783	-0.881	-0.779	-0.775	-0.797	-0.555
-0.799	-0.773	-0.794	-0.816	-0.748	-0.439
-0.813	-0.79	-0.782	-0.791	-0.794	-0.482
-0.785	-0.966	-0.779	-0.813	-0.815	-0.41
-0.79	-0.854	-0.797	-0.586	-0.757	-0.619
-0.781	-0.708	-0.844	-0.711	-0.826	-0.606
-0.797	-0.737	-0.775	-0.846	-0.813	-0.471
-0.788	-0.81	-0.777	-0.846	-0.79	-0.526
-0.784	-0.724	-0.778	-0.755	-0.722	-0.502
-0.779	-0.738	-0.797	-0.717	-0.858	-0.475
-0.833	-0.637	-0.792	-0.901	-0.829	-0.609
-0.807	-0.758	-0.805	-0.839	-0.795	-0.23
-0.781	-0.85	-0.777	-0.726	-0.762	-0.343
-0.776	-0.904	-0.802	-0.775	-0.82	-0.347
-0.828	-0.807	-0.774	-0.832	-0.774	-0.377
-0.768	-0.834	-0.783	-0.755	-0.735	-0.35
-0.781	-0.887	-0.793	-0.71	-0.8	-0.181
-0.776	-0.766	-0.844	-0.756	-0.843	-0.318
-0.774	-0.716	-0.795	-0.832	-0.744	-0.33
-0.776	-0.812	-0.768	-0.794	-0.802	-0.355
-0.831	-0.733	-0.768	-0.759	-0.819	-0.315
-0.8	-0.781	-0.808	-0.807	-0.839	-0.195
-0.782	-0.789	-0.796	-0.785	-0.734	-0.124
-0.815	-0.835	-0.812	-0.736	-0.797	-0.18
-0.823	-0.798	-0.776	-0.792	-0.933	-0.203
-0.739	-0.815	-0.767	-0.843	-1.03	-0.292
-0.802	-0.794	-0.781	-0.748	-0.872	-0.117
-0.745	-0.776	-0.784	-0.759	-0.583	-0.188
-0.728	-0.791	-0.808	-0.863	-0.718	-0.113
-0.813	-0.787	-0.837	-0.839	-1.033	-0.238
-0.869	-0.792	-0.786	-0.763	-0.59	-0.097
-0.959	-0.797	-0.787	-0.788	-0.719	-0.028

-0.066
-0.141
-0.081
0.018
-0.016
0.275
0.395
0.245
0.122
0.377
0.293
0.305
0.183
0.052
0.166
0.197
0.187
0.209
0.289
0.229
0.267
0.181
0.32
0.187
0.183
0.219
0.215
0.23
0.224
0.253
0.285
0.238
0.264
0.207
0.209
0.216
0.235
0.258
0.231
0.263
0.199
0.218
0.221
0.235
0.24
0.239
0.219
0.221
0.219
0.254
0.237
0.231
0.22
0.228
0.229
0.229
0.233
0.227
0.222
0.232
0.23
0.235
0.233
0.224
0.229
0.236
0.231
0.226
0.237
0.225
0.225
0.226
0.226
0.23
0.232
0.228
0.225
0.226
0.228
0.23
0.232
0.233
0.232
0.229
0.229
0.236
0.231
0.224
0.227
0.229
0.23
0.235
0.227
0.219
0.212
0.229
0.225
0.235
0.236
0.231
0.239
0.229
0.224
0.227
0.229
0.236
0.23
0.229
0.225
0.229
0.229
0.227
0.227
0.224
0.232
0.229
0.229
0.232
0.227
0.226
0.231
0.229
0.232
0.23
0.232
0.236
0.229
0.229
0.228
0.232
0.226
0.234
0.236

Appendix C

Random Accelerometer Data Sample (Exploited) from Robotic Arm Interval

.019	0.194	-0.726	-0.81	-0.359	0.211
0.236	-0.254	-0.947	-0.76	-0.724	0.184
0.289	0.252	-0.865	-0.879	-0.305	0.278
0.398	0.073	-0.9	-0.766	-0.52	0.24
0.169	-0.339	-0.827	-0.741	-0.577	0.311
0.277	-0.435	-0.728	-0.729	-0.312	0.278
0.472	-0.32	-0.75	-0.859	-0.302	-0.12
0.189	-0.611	-0.763	-0.821	-0.215	0.093
0.116	-0.459	-0.866	-0.745	-0.48	0.59
0.309	-0.332	-0.736	-0.813	-0.372	0.055
0.215	-0.173	-0.647	-0.846	-0.417	0.208
-0.067	-0.525	-0.68	-0.778	-0.258	0.292
0.298	-0.059	-0.633	-0.692	-0.412	0.325
0.251	-0.095	-0.774	-0.789	-0.073	0.018
0.397	-0.229	-0.61	-0.867	-0.196	0.356
0.209	-0.204	-0.652	-0.712	-0.377	0.479
0.309	-0.181	-0.829	-0.638	-0.146	0.252
0.251	-0.33	-0.893	-0.666	-0.112	0.192
0.396	0.199	-0.854	-1.063	-0.22	0.413
0.165	-0.447	-0.869	-0.679	-0.197	0.234
0.122	-0.316	-0.853	-0.556	-0.202	0.024
0.184	-1.185	-0.875	-0.742	-0.02	0.217
0.114	-0.082	-0.826	-0.678	-0.031	0.451
0.338	-0.038	-0.789	-0.618	-0.233	0.158
0.229	0.806	-0.744	-0.845	-0.104	0.142
0.089	-0.307	-0.831	-0.724	-0.138	0.349
0.461	0.113	-0.706	-0.481	-0.151	0.293
0.305	0.408	-0.775	-0.699	-0.005	0.182
0.304	-0.707	-0.823	-0.736	0.014	0.254
0.298	-0.866	-0.812	-0.816	0.014	0.3
0.625	-0.194	-0.812	-0.604	-0.041	0.31
0.381	-0.199	-0.865	-0.48	0.063	0.169
0.131	-0.598	-0.962	-0.722	0.047	0.127
0.455	-1.517	-0.811	-0.794	-0.093	0.308
0.406	-0.32	-0.788	-0.466	0.02	0.122
0.424	-0.26	-0.816	-0.576	0.111	0.081
0.299	-0.86	-0.816	-0.604	0.035	0.33
-0.032	-0.957	-0.772	-0.488	0.028	0.217
-0.019	-0.672	-0.732	-0.424	0.126	0.262
-0.189	-0.444	-0.769	-0.487	0.095	0.523
0.04	-0.665	-0.848	-0.512	0.085	0.187
0.023	-0.761	-0.799	-0.649	0.145	0.108
0.11	-0.545	-0.774	-0.491	0.181	0.433

0.26	0.063	0.085	-0.355	-0.726	-0.811
0.182	0.078	0.409	-0.387	-0.801	-0.777
0.165	0.215	0.328	-0.265	-0.793	-0.737
0.33	0.349	0.003	-0.443	-0.758	-0.688
0.464	0.013	0.192	-0.24	-0.783	-0.777
-0.103	0.264	0.423	-0.348	-0.787	-0.849
0.135	0.5	0.26	-0.479	-0.779	-0.777
0.502	-0.036	0.106	-0.577	-0.763	-0.83
0.129	0.095	0.275	-0.225	-0.738	-0.786
0.178	0.386	0.352	-0.463	-0.701	-0.741
0.301	0.286	0.05	-0.545	-0.852	-0.78
0.344	0.061	0.274	-0.578	-0.816	-0.806
0.028	0.481	0.524	-0.458	-0.75	-0.699
0.407	0.44	0.305	-0.628	-0.729	-0.751
0.489	0.014	0.011	-0.803	-0.847	-0.799
0.098	0.124	0.484	-0.603	-0.782	-0.687
0.159	0.47	0.493	-0.386	-0.797	-0.824
0.286	0.373	-0.012	-0.68	-0.739	-0.822
0.318	0	0.104	-0.526	-0.767	-0.74
0.035	0.271	0.341	-0.525	-0.771	-0.775
0.17	0.52	0.402	-0.585	-0.7	-0.844
0.529	-0.084	0.253	-0.709	-0.723	-0.823
0.088	0.098	0.193	-0.737	-0.874	-0.807
0.102	0.396	0.296	-0.824	-0.731	-0.681
0.406	0.264	-0.04	-0.919	-0.732	-0.763
0.338	0.077	0.082	-0.386	-0.842	-0.779
-0.043	0.392	0.263	-0.833	-0.83	-0.832
0.271	0.574	-0.223	-0.868	-0.774	-0.716
0.389	-0.011	-0.026	-0.761	-0.815	-0.757
0.065	0.166	-0.304	-0.781	-0.702	-0.766
-0.013	0.495	0	-0.815	-0.749	-0.802
0.525	0.131	0.036	-0.791	-0.822	-0.76
0.256	0.134	-0.205	-0.816	-0.743	-0.771
-0.068	0.47	0.063	-0.787	-0.836	-0.693
0.364	0.255	-0.093	-0.759	-0.831	-0.803
0.432	0.193	-0.206	-0.773	-0.773	-0.771
0.132	0.221	0.026	-0.839	-0.746	-0.761
0.177	0.309	0.112	-0.774	-0.718	-0.681
0.355	0.087	-0.126	-0.699	-0.863	-0.803
0.456	0.113	-0.421	-0.739	-0.741	-0.792
-0.125	0.331	-0.361	-0.76	-0.757	-0.776
0.319	0.227	-0.051	-0.8	-0.76	-0.83
0.451	0.312	-0.769	-0.7	-0.821	-0.782

-0.792	-0.796	-0.737	-0.78	-0.77	-0.779
-0.782	-0.768	-0.78	-0.761	-0.798	-0.792
-0.751	-0.733	-0.814	-0.83	-0.836	-0.704
-0.688	-0.725	-0.792	-0.763	-0.778	-0.729
-0.842	-0.84	-0.779	-0.814	-0.703	-0.824
-0.838	-0.748	-0.696	-0.7	-0.73	-0.827
-0.749	-0.724	-0.756	-0.728	-0.794	-0.745
-0.772	-0.801	-0.754	-0.772	-0.838	-0.823
-0.828	-0.799	-0.809	-0.724	-0.739	-0.767
-0.774	-0.784	-0.714	-0.706	-0.754	-0.8
-0.792	-0.8	-0.775	-0.792	-0.75	-0.805
-0.695	-0.723	-0.75	-0.76	-0.803	-0.755
-0.74	-0.743	-0.802	-0.774	-0.778	-0.687
-0.791	-0.817	-0.792	-0.811	-0.796	-0.846
-0.832	-0.85	-0.821	-0.751	-0.7	-0.704
-0.726	-0.733	-0.729	-0.739	-0.789	-0.745
-0.746	-0.745	-0.813	-0.791	-0.788	-0.792
-0.755	-0.816	-0.784	-0.752	-0.735	-0.817
-0.815	-0.786	-0.721	-0.718	-0.785	-0.768
-0.775	-0.777	-0.778	-0.859	-0.813	-0.795
-0.786	-0.72	-0.864	-0.805	-0.752	-0.701
-0.731	-0.729	-0.722	-0.741	-0.706	-0.821
-0.779	-0.782	-0.772	-0.784	-0.802	-0.762
-0.754	-0.732	-0.77	-0.828	-0.81	-0.666
-0.74	-0.735	-0.769	-0.719	-0.712	-0.761
-0.733	-0.76	-0.774	-0.792	-0.702	-0.813
-0.854	-0.775	-0.761	-0.663	-0.78	-0.752
-0.77	-0.714	-0.7	-0.763	-0.761	-0.786
-0.766	-0.791	-0.796	-0.758	-0.781	-0.836
-0.776	-0.814	-0.769	-0.852	-0.781	-0.791
-0.837	-0.679	-0.709	-0.696	-0.726	-0.786
-0.798	-0.747	-0.771	-0.767	-0.768	-0.782
-0.795	-0.828	-0.792	-0.758	-0.784	-0.745
-0.754	-0.729	-0.769	-0.787	-0.749	-0.703
-0.792	-0.791	-0.779	-0.779	-0.718	-0.834
-0.838	-0.758	-0.934	-0.804	-0.848	-0.829
-0.694	-0.796	-0.77	-0.699	-0.789	-0.744
-0.755	-0.797	-0.805	-0.803	-0.751	-0.778
-0.725	-0.78	-0.79	-0.784	-0.82	-0.825
-0.826	-0.758	-0.776	-0.701	-0.815	-0.78
-0.777	-0.727	-0.768	-0.842	-0.8	-0.783
-0.789	-0.831	-0.871	-0.829	-0.759	-0.721
-0.722	-0.818	-0.745	-0.75	-0.746	-0.748

-0.781	-0.81	-0.769	-0.771	-0.738	-0.779
-0.698	-0.727	-0.825	-0.699	-0.787	-0.789
-0.735	-0.748	-0.67	-0.804	-0.776	-0.775
-0.856	-0.76	-0.798	-0.759	-0.775	-0.843
-0.766	-0.806	-0.734	-0.73	-0.827	-0.727
-0.723	-0.797	-0.702	-0.738	-0.753	-0.785
-0.812	-0.798	-0.824	-0.858	-0.794	-0.791
-0.819	-0.704	-0.842	-0.754	-0.691	-0.746
-0.687	-0.809	-0.753	-0.781	-0.758	-0.718
-0.815	-0.808	-0.788	-0.811	-0.748	-0.868
-0.737	-0.74	-0.79	-0.838	-0.86	-0.712
-0.729	-0.823	-0.829	-0.782	-0.763	-0.764
-0.846	-0.813	-0.776	-0.79	-0.728	-0.792
-0.805	-0.813	-0.692	-0.733	-0.747	-0.83
-0.747	-0.699	-0.744	-0.714	-0.788	-0.768
-0.776	-0.756	-0.744	-0.811	-0.788	-0.812
-0.832	-0.784	-0.821	-0.845	-0.768	-0.692
-0.772	-0.729	-0.708	-0.731	-0.679	-0.805
-0.788	-0.713	-0.746	-0.773	-0.782	-0.763
-0.679	-0.81	-0.759	-0.809	-0.803	-0.701
-0.757	-0.759	-0.811	-0.771	-0.715	-0.703
-0.795	-0.783	-0.764	-0.78	-0.734	-0.828
-0.818	-0.767	-0.828	-0.694	-0.786	-0.746
-0.737	-0.779	-0.714	-0.76	-0.77	-0.788
-0.744	-0.72	-0.813	-0.77	-0.787	-0.801
-0.761	-0.795	-0.756	-0.797	-0.836	-0.824
-0.88	-0.749	-0.778	-0.724	-0.81	-0.798
-0.753	-0.727	-0.724	-0.751	-0.798	-0.8
-0.834	-0.813	-0.867	-0.76	-0.805	-0.735
-0.71	-0.859	-0.736	-0.773	-0.751	-0.707
-0.773	-0.752	-0.758	-0.778	-0.791	-0.836
-0.815	-0.768	-0.819	-0.818	-0.88	-0.818
-0.712	-0.831	-0.811	-0.705	-0.707	-0.762
-0.715	-0.78	-0.79	-0.806	-0.796	-0.75
-0.853	-0.791	-0.807	-0.771	-0.754	-0.82
-0.737	-0.682	-0.749	-0.724	-0.832	-0.772
-0.776	-0.75	-0.747	-0.775	-0.81	-0.781
-0.842	-0.758	-0.859	-0.852	-0.729	-0.718
-0.819	-0.835	-0.722	-0.723	-0.702	-0.744
-0.798	-0.714	-0.762	-0.79	-0.786	-0.761
-0.804	-0.763	-0.751	-0.798	-0.789	-0.699
-0.756	-0.744	-0.811	-0.806	-0.733	-0.724
-0.801	-0.796	-0.759	-0.772	-0.741	-0.838

-0.746	-0.723	-0.72	-0.703	-0.351	0.15
-0.783	-0.755	-0.709	-0.832	0.173	0.083
-0.77	-0.767	-0.873	-0.733	0.445	0.118
-0.802	-0.875	-0.729	-0.735	0.233	0.068
-0.795	-0.722	-0.763	-0.763	0.352	0.427
-0.797	-0.785	-0.75	-0.811	0.218	0.288
-0.732	-0.744	-0.838	-0.792	0.072	0.223
-0.728	-0.849	-0.782	-0.788	0.397	0.072
-0.813	-0.746	-0.783	-0.709	0.333	0.408
-0.883	-0.755	-0.677	-0.804	0.437	0.204
-0.759	-0.691	-0.799	-0.774	0.247	0.193
-0.76	-0.794	-0.775	-0.685	0.173	0.339
-0.814	-0.754	-0.714	-0.723	0.317	0.329
-0.778	-0.736	-0.711	-0.852	0.146	0.074
-0.795	-0.716	-0.789	-0.768	0.31	0.365
-0.717	-0.833	-0.743	-0.813	0.229	0.074
-0.709	-0.759	-0.782	-0.799	0.509	0.181
-0.767	-0.771	-0.822	-0.948	0.147	0.297
-0.813	-0.826	-0.774	-0.794	0.237	0.1
-0.755	-0.789	-0.81	-0.815	0.24	0.247
-0.745	-0.811	-0.809	-0.509	0.492	0.251
-0.747	-0.797	-0.755	-0.596	0.115	0.077
-0.765	-0.761	-0.73	-0.999	0.12	0.278
-0.764	-0.775	-0.858	-0.641	0.255	0.149
-0.824	-0.836	-0.708	-0.631	0.272	0.05
-0.692	-0.736	-0.742	-0.853	0.188	0.386
-0.803	-0.753	-0.746	-0.728	0.159	0.261
-0.778	-0.739	-0.835	-0.423	0.237	0.001
-0.718	-0.825	-0.765	-0.159	0.229	-0.001
-0.819	-0.802	-0.784	0.454	0.258	0.487
-0.807	-0.794	-0.699	-0.9	0.261	0.135
-0.741	-0.705	-0.81	0.13	0.288	0.146
-0.768	-0.818	-0.764	-0.261	0.263	0.482
-0.791	-0.773	-0.736	-0.054	0.282	0.325
-0.828	-0.747	-0.728	0.31	0.246	0.33
-0.781	-0.71	-0.83	-0.484	0.269	-0.189
-0.797	-0.823	-0.757	0.05	0.24	0.446
-0.723	-0.749	-0.776	0.445	0.301	0.138
-0.761	-0.79	-0.817	0.42	0.143	0.105
-0.779	-0.826	-0.815	-0.46	0.308	0.336
-0.817	-0.794	-0.791	-0.408	0.132	0.452
-0.786	-0.81	-0.808	0.12	0.379	0.032
-0.812	-0.812	-0.742	0.15	0.203	0.284

0.331	0.189	-0.89	-0.78	-0.662	0.094
0.137	0.322	-0.316	-0.797	-0.633	-0.047
0.349	0.176	-0.914	-0.709	-0.606	-0.107
0.262	0.626	-0.219	-0.783	-0.451	0.117
0.156	0.273	-0.565	-0.666	-0.698	0.17
0.153	-0.203	-1.418	-0.727	-0.578	0.09
0.438	0.059	-1.153	-0.87	-0.546	0.129
0.199	0.179	-1.113	-0.933	-0.356	0.143
0.319	0.163	-0.446	-0.841	-0.63	0.113
0.304	0.135	-0.602	-0.786	-0.58	0.203
0.068	0.016	-0.873	-0.71	-0.284	0.209
0.396	0.021	-0.891	-0.801	-0.347	0.303
0.035	-0.187	-0.848	-0.782	-0.545	0.238
0.202	-0.402	-0.785	-0.757	-0.504	0.297
0.316	0.156	-0.815	-0.768	-0.464	0.645
0.438	0.108	-0.705	-0.783	-0.276	0.207
-0.087	-0.443	-0.96	-0.751	-0.458	0.354
0.477	0.063	-0.83	-0.743	-0.666	0.325
0.298	-0.183	-0.578	-0.842	-0.216	0.232
-0.056	-0.45	-0.816	-0.776	-0.281	0.233
-0.029	-0.943	-0.956	-0.64	-0.41	0.187
0.262	-0.602	-0.649	-0.769	-0.258	0.381
0.289	0.05	-0.658	-0.852	-0.365	0.313
0.233	-0.585	-0.577	-0.796	-0.185	0.278
0.313	0.262	-0.756	-0.731	-0.278	0.098
0.238	-0.105	-0.941	-0.863	-0.22	0.146
0.082	-0.058	-1.066	-0.945	-0.153	0.302
0.061	-0.963	-0.836	-0.672	-0.291	0.349
0.242	-0.616	-0.778	-0.623	-0.126	0.44
0.309	0.264	-0.87	-1.105	-0.141	0.453
0.104	-0.159	-0.876	-0.604	-0.169	0.364
0.171	-1.336	-0.767	-0.545	-0.014	0.27
0.433	-0.009	-0.802	-0.504	-0.101	0.357
0.123	-0.82	-0.78	-0.848	0.003	0.305
-0.018	-0.54	-0.827	-0.534	-0.021	-0.024
0.312	0.077	-0.744	-0.549	-0.071	0.281
0.063	-0.822	-0.86	-0.303	-0.1	0.281
0.038	-0.349	-0.805	-0.791	-0.093	0.281
0.298	-0.937	-0.85	-0.843	-0.164	0.198
0.356	-0.515	-0.874	-0.488	0.048	0.319
0.163	-0.544	-0.878	-0.829	0.144	0.159
0.168	-1.683	-0.811	-0.757	-0.038	0.066
0.219	-0.888	-0.717	-0.658	-0.081	0.531

0.178	0.421	0.183	-0.206	-0.754	-0.787
0.233	0.225	0.182	-0.203	-0.765	-0.721
0.211	0.1	0.241	-0.081	-0.727	-0.743
0.19	0.253	0.29	0.11	-0.85	-0.758
0.235	0.418	0.189	-0.788	-0.785	-0.813
0.072	0.162	0.217	-0.664	-0.744	-0.772
0.196	0.149	0.446	-0.312	-0.788	-0.815
0.068	0.405	0.186	-0.191	-0.806	-0.821
0.274	0.373	0.197	-0.578	-0.763	-0.787
0.237	-0.014	0.367	-0.012	-0.731	-0.779
0.128	0.274	0.37	-0.374	-0.719	-0.79
0.27	0.3	0.271	-0.23	-0.751	-0.78
0.367	0.229	0.439	-0.411	-0.731	-0.728
0.333	0.098	0.412	-0.309	-0.752	-0.764
0.004	0.397	0.163	-0.503	-0.833	-0.716
0.381	0.332	0.24	-0.627	-0.746	-0.807
0.454	0.063	0.321	-0.375	-0.773	-0.703
0.001	0.395	0.37	-0.421	-0.779	-0.785
0.241	0.491	0.1	-0.428	-0.852	-0.769
0.277	0.003	0.314	-0.577	-0.754	-0.841
0.149	0.187	0.414	-0.255	-0.803	-0.752
0.21	0.478	0.195	-0.685	-0.766	-0.778
0.426	0.16	0.165	-0.597	-0.814	-0.727
0.078	0.191	0.241	-0.57	-0.762	-0.808
0.09	0.415	0.28	-0.497	-0.834	-0.729
0.391	0.317	0.408	-0.59	-0.746	-0.825
0.277	0.096	0.122	-0.834	-0.795	-0.765
0.127	0.099	0.158	-0.515	-0.735	-0.792
0.11	0.374	0.234	-0.487	-0.788	-0.716
0.337	0.287	0.207	-0.747	-0.749	-0.824
0.603	-0.014	0.177	-0.603	-0.812	-0.779
-0.124	0.192	0.352	-0.419	-0.728	-0.79
0.29	0.597	-0.028	-0.694	-0.788	-0.688
0.387	-0.005	0.396	-0.994	-0.818	-0.796
0.211	0.263	0.479	-0.653	-0.691	-0.729
0.109	0.323	0.205	-0.891	-0.817	-0.79
0.346	0.37	0.137	-0.706	-0.759	-0.824
0.288	0.098	0.343	-1.045	-0.806	-0.786
0.037	0.078	0.262	-0.671	-0.705	-0.751
0.447	0.331	0.079	-0.819	-0.838	-0.744
0.364	0.169	0.286	-0.773	-0.791	-0.729
0.096	0.158	0.16	-0.768	-0.805	-0.764
0.075	0.455	0.139	-0.813	-0.714	-0.718

-0.784	-0.737	-0.745	-0.764	-0.803	-0.773
-0.762	-0.793	-0.787	-0.792	-0.77	-0.748
-0.793	-0.77	-0.775	-0.767	-0.8	-0.771
-0.775	-0.857	-0.794	-0.803	-0.733	-0.72
-0.842	-0.744	-0.808	-0.713	-0.777	-0.734
-0.744	-0.791	-0.718	-0.821	-0.733	-0.768
-0.793	-0.744	-0.777	-0.8	-0.793	-0.77
-0.752	-0.801	-0.74	-0.784	-0.725	-0.78
-0.8	-0.758	-0.783	-0.746	-0.842	-0.796
-0.735	-0.817	-0.742	-0.839	-0.78	-0.841
-0.836	-0.778	-0.819	-0.802	-0.823	-0.762
-0.73	-0.784	-0.737	-0.793	-0.722	-0.799
-0.802	-0.761	-0.78	-0.729	-0.77	-0.772
-0.724	-0.763	-0.779	-0.797	-0.766	-0.793
-0.835	-0.81	-0.826	-0.727	-0.767	-0.723
-0.815	-0.722	-0.734	-0.738	-0.748	-0.812
-0.76	-0.729	-0.805	-0.768	-0.83	-0.732
-0.73	-0.772	-0.76	-0.783	-0.732	-0.842
-0.759	-0.808	-0.804	-0.799	-0.813	-0.77
-0.749	-0.736	-0.756	-0.773	-0.758	-0.725
-0.752	-0.816	-0.832	-0.854	-0.835	-0.784
-0.769	-0.774	-0.723	-0.81	-0.762	-0.745
-0.736	-0.803	-0.793	-0.807	-0.794	-0.787
-0.799	-0.776	-0.718	-0.78	-0.691	-0.792
-0.782	-0.831	-0.837	-0.792	-0.778	-0.798
-0.842	-0.779	-0.779	-0.751	-0.725	-0.858
-0.772	-0.818	-0.762	-0.793	-0.719	-0.751
-0.793	-0.727	-0.824	-0.721	-0.728	-0.802
-0.773	-0.793	-0.752	-0.776	-0.767	-0.774
-0.787	-0.739	-0.79	-0.721	-0.79	-0.799
-0.732	-0.771	-0.73	-0.806	-0.796	-0.723
-0.837	-0.736	-0.783	-0.825	-0.836	-0.808
-0.758	-0.812	-0.779	-0.861	-0.795	-0.797
-0.768	-0.72	-0.695	-0.723	-0.784	-0.788
-0.778	-0.777	-0.751	-0.805	-0.782	-0.733
-0.845	-0.777	-0.752	-0.732	-0.805	-0.834
-0.78	-0.817	-0.742	-0.771	-0.725	-0.793
-0.776	-0.743	-0.713	-0.731	-0.82	-0.857
-0.744	-0.791	-0.831	-0.854	-0.75	-0.719
-0.77	-0.731	-0.771	-0.737	-0.79	-0.758
-0.779	-0.749	-0.781	-0.767	-0.729	-0.751
-0.762	-0.764	-0.773	-0.806	-0.85	-0.723
-0.741	-0.831	-0.832	-0.806	-0.777	-0.777

-0.781	-0.78	-0.811	-0.784	-0.711	-0.745
-0.762	-0.828	-0.877	-0.745	-0.734	-0.758
-0.828	-0.821	-0.788	-0.755	-0.773	-0.771
-0.844	-0.787	-0.79	-0.752	-0.746	-0.715
-0.806	-0.796	-0.674	-0.784	-0.828	-0.824
-0.796	-0.708	-0.799	-0.795	-0.79	-0.726
-0.724	-0.807	-0.708	-0.804	-0.855	-0.825
-0.809	-0.745	-0.816	-0.844	-0.722	-0.712
-0.716	-0.857	-0.727	-0.743	-0.812	-0.794
-0.796	-0.718	-0.826	-0.808	-0.753	-0.717
-0.753	-0.826	-0.804	-0.788	-0.811	-0.813
-0.778	-0.755	-0.717	-0.798	-0.704	-0.737
-0.724	-0.804	-0.782	-0.715	-0.849	-0.807
-0.803	-0.757	-0.765	-0.811	-0.775	-0.706
-0.794	-0.746	-0.811	-0.765	-0.781	-0.813
-0.816	-0.73	-0.766	-0.794	-0.721	-0.787
-0.722	-0.752	-0.814	-0.737	-0.79	-0.786
-0.785	-0.807	-0.774	-0.857	-0.76	-0.794
-0.747	-0.738	-0.873	-0.75	-0.779	-0.853
-0.779	-0.834	-0.778	-0.821	-0.68	-0.889
-0.785	-0.757	-0.715	-0.729	-0.753	-0.779
-0.799	-0.788	-0.812	-0.809	-0.736	-0.812
-0.789	-0.776	-0.722	-0.701	-0.729	-0.776
-0.776	-0.85	-0.775	-0.761	-0.86	-0.775
-0.83	-0.785	-0.731	-0.732	-0.749	-0.764
-0.797	-0.793	-0.757	-0.787	-0.782	-0.778
-0.796	-0.713	-0.772	-0.778	-0.753	-0.715
-0.726	-0.804	-0.757	-0.788	-0.885	-0.821
-0.784	-0.746	-0.75	-0.84	-0.764	-0.755
-0.719	-0.836	-0.835	-0.817	-0.836	-0.803
-0.778	-0.729	-0.743	-0.796	-0.711	-0.73
-0.729	-0.765	-0.812	-0.743	-0.814	-0.799
-0.792	-0.767	-0.74	-0.803	-0.741	-0.753
-0.693	-0.786	-0.809	-0.708	-0.863	-0.855
-0.834	-0.737	-0.706	-0.84	-0.761	-0.726
-0.761	-0.746	-0.854	-0.729	-0.804	-0.797
-0.82	-0.766	-0.766	-0.794	-0.726	-0.796
-0.729	-0.757	-0.815	-0.707	-0.795	-0.736
-0.79	-0.808	-0.771	-0.774	-0.766	-0.813
-0.717	-0.728	-0.839	-0.778	-0.776	-0.779
-0.768	-0.844	-0.781	-0.771	-0.729	-0.778
-0.802	-0.74	-0.783	-0.694	-0.768	-0.824
-0.815	-0.806	-0.715	-0.771	-0.724	-0.82

-0.789	-0.749	-0.702	-0.853	-1.191	0.262
-0.782	-0.733	-0.818	-0.762	-0.294	0.211
-0.729	-0.745	-0.843	-0.783	-0.302	0.205
-0.803	-0.743	-0.765	-0.693	-0.738	0.125
-0.725	-0.771	-0.698	-0.874	0.058	0.373
-0.824	-0.718	-0.808	-0.71	-0.438	0.295
-0.808	-0.75	-0.761	-0.802	-0.307	0.228
-0.803	-0.788	-0.793	-0.777	-0.924	0.121
-0.731	-0.778	-0.789	-0.756	0.297	0.308
-0.787	-0.757	-0.731	-0.813	-0.626	0.179
-0.722	-0.884	-0.86	-0.84	-0.338	0.186
-0.876	-0.789	-0.841	-0.732	-0.371	0.294
-0.734	-0.777	-0.716	-0.774	0.065	0.253
-0.773	-0.747	-0.752	-0.872	0.057	0.186
-0.79	-0.76	-0.85	-0.761	-0.036	0.244
-0.776	-0.712	-0.745	-0.738	0.269	0.331
-0.774	-0.891	-0.754	-0.734	-0.175	0.16
-0.818	-0.694	-0.742	-0.773	-0.283	0.379
-0.753	-0.779	-0.756	-0.7	0.029	0.357
-0.729	-0.771	-0.717	-0.799	0.135	0.204
-0.839	-0.814	-0.83	-0.796	0.003	0.275
-0.765	-0.738	-0.721	-0.786	0.035	0.187
-0.771	-0.825	-0.817	-0.727	0.297	0.152
-0.726	-0.757	-0.764	-0.82	0.126	0.163
-0.805	-0.771	-0.812	-0.894	-0.059	0.03
-0.741	-0.8	-0.72	-1.027	-0.136	0.335
-0.778	-0.695	-0.793	-0.835	0.162	0.333
-0.745	-0.726	-0.752	-0.639	0.222	0.146
-0.847	-0.749	-0.756	-0.996	0.218	0.35
-0.708	-0.829	-0.795	-0.712	0.203	0.262
-0.768	-0.734	-0.767	-0.54	0.557	0.096
-0.832	-0.737	-0.839	-0.611	0.235	0.119
-0.861	-0.758	-0.782	-0.219	0.364	0.349
-0.739	-0.788	-0.76	-0.289	0.379	0.197
-0.799	-0.737	-0.748	-0.543	0.199	0.265
-0.766	-0.794	-0.86	-0.483	0.25	0.224
-0.751	-0.739	-0.738	-0.889	0.227	0.183
-0.777	-0.716	-0.747	-0.326	0.256	0.038
-0.859	-0.731	-0.746	-0.675	0.259	0.48
-0.714	-0.753	-0.792	-0.452	0.344	0.201
-0.777	-0.791	-0.725	-0.237	0.252	0.196
-0.813	-0.714	-0.841	-0.514	0.28	0.229
-0.885	-0.75	-0.693	0.058	0.249	0.057

0.32	0.139	-0.326	-0.73	-0.818	-0.137
0.013	0.211	-0.429	-0.745	-0.731	-0.111
0.042	0.273	-0.206	-0.773	-0.755	0.02
0.249	0.152	-0.095	-0.825	-0.767	-0.09
0.25	0.032	-0.986	-0.825	-0.853	0.049
0.336	0.408	-1.408	-0.749	-0.739	-0.045
0.15	0.297	-0.425	-0.716	-0.866	-0.004
0.539	0.124	-0.699	-0.785	-0.839	0.096
0.17	0.204	-0.705	-0.516	-0.791	0.007
0.104	0.345	-1.229	-0.852	-0.8	0.11
0.684	0.171	-0.602	-0.769	-0.898	0.099
0.182	0.145	0.161	-0.853	-0.937	0.074
0.284	0.357	-0.207	-0.721	-0.807	0.102
0.107	0.279	-0.888	-0.927	-0.862	-0.012
0.328	0.215	-0.569	-0.8	-0.295	0.151
0.332	0.224	-0.566	-0.796	-0.745	0.144
-0.145	0.312	-1.225	-0.946	-0.441	0.144
0.794	0.087	-0.469	-0.704	-0.387	0.177
0.165	-0.004	-0.893	-0.724	-0.485	0.137
0.01	0.2	-0.933	-0.808	-0.619	0.145
0.489	0.227	-0.997	-0.775	-0.318	0.116
0.164	-0.081	-0.866	-0.889	-0.425	0.248
0.641	0.107	-0.794	-0.881	-0.226	0.293
0.241	0.209	-0.883	-0.645	-0.447	0.533
-0.17	-0.085	-0.76	-0.756	-0.442	-0.077
0.737	0.042	-0.778	-0.807	-0.304	0.416
0.322	0.193	-0.76	-0.667	-0.368	0.023
0.113	-0.234	-0.912	-0.916	-0.279	0.257
0.133	0.537	-0.59	-0.854	-0.363	0.204
0.403	-0.209	-0.856	-0.788	-0.198	0.252
0.287	0.065	-0.776	-0.782	-0.323	0.301
0.187	-0.183	-0.657	-0.917	-0.313	0.083
0.36	0.337	-0.775	-0.934	-0.167	0.21
0.166	0.265	-0.969	-0.732	-0.281	0.213
0.549	-0.913	-0.64	-0.697	-0.374	0.244
0.543	-0.518	-0.705	-0.74	-0.278	0.207
0.14	-0.864	-0.782	-0.808	-0.279	0.324
0.162	0.26	-0.92	-0.773	-0.03	0.371
0.41	-0.1	-0.686	-0.762	-0.329	0.325
0.237	-0.71	-0.887	-0.835	-0.076	0.35
-0.193	-0.002	-0.879	-0.843	-0.063	0.393
0.316	-0.784	-0.887	-0.777	-0.126	0.274
0.323	0.106	-0.768	-0.773	-0.003	-0.078

0.463	0.199	0.08	0.296	-1.021	-0.749
0.482	0.038	0.313	0.438	-0.742	-0.875
-0.119	0.286	0.323	0.14	-0.749	-0.786
0.267	0.37	0.119	0.245	-1.054	-0.763
0.477	0.147	0.213	0.318	-0.712	-0.733
0.168	0.403	0.145	0.393	-0.709	-0.918
-0.098	0.162	0.194	0.126	-0.771	-0.7
0.677	0.398	0.264	0.408	-0.778	-0.836
0.201	0.126	0.269	0.399	-0.794	-0.742
-0.104	0.141	0.339	0.143	-0.774	-0.775
0.247	0.345	0.243	0.38	-0.769	-0.752
0.374	0.377	0.328	0.214	-0.835	-0.883
0.223	0.004	0.287	0.234	-0.85	-0.72
0.006	0.393	0.31	-0.193	-0.793	-0.768
0.508	0.405	0.244	0.26	-0.764	-0.753
0.141	0.271	0.259	0.127	-0.81	-0.804
0.03	0.003	0.181	0.369	-0.716	-0.748
0.282	0.3	0.405	-0.216	-0.794	-0.72
0.321	0.265	0.221	0.361	-0.728	-0.76
0.152	0.03	0.374	0.374	-0.743	-0.772
0.173	0.278	0.388	-0.35	-0.755	-0.821
0.305	0.354	0.202	0.011	-0.853	-0.738
0.189	0.175	0.119	-0.235	-0.735	-0.764
0.192	0.291	0.206	0.114	-0.856	-0.767
0.27	0.457	0.194	-0.16	-0.748	-0.751
0.448	0.259	0.232	-0.172	-0.737	-0.717
0.163	-0.021	0.252	0.031	-0.72	-0.814
0.174	0.311	0.344	-0.027	-0.786	-0.739
0.224	0.493	0.244	-0.19	-0.732	-0.852
0.191	0.083	0.157	-0.045	-0.843	-0.765
0.138	0.137	0.43	-0.061	-0.761	-0.773
0.249	0.465	0.293	-0.212	-0.748	-0.779
0.291	0.164	0.037	0.074	-0.772	-0.888
0.155	0.055	0.51	-0.47	-0.875	-0.719
0.18	0.343	0.51	-0.431	-0.746	-0.783
0.273	0.319	0.024	-0.368	-0.775	-0.73
0.181	0.205	0.233	-0.187	-0.775	-0.812
0.104	0.089	0.31	-0.085	-0.722	-0.705
0.224	0.445	0.014	-0.271	-0.772	-0.854
0.384	0.343	0.22	-0.315	-0.736	-0.75
0.2	-0.006	0.344	-0.409	-0.725	-0.759
0.135	0.391	0.22	-0.767	-0.773	-0.8
0.354	0.396	0.272	-0.797	-0.836	-0.743

-0.733	-0.729	-0.878	-0.787	-0.725	-0.878
-0.795	-0.724	-0.776	-0.711	-0.762	-0.743
-0.801	-0.738	-0.789	-0.832	-0.737	-0.78
-0.711	-0.736	-0.785	-0.718	-0.887	-0.747
-0.816	-0.793	-0.711	-0.765	-0.723	-0.707
-0.738	-0.747	-0.778	-0.741	-0.778	-0.729
-0.84	-0.724	-0.74	-0.864	-0.754	-0.718
-0.751	-0.904	-0.715	-0.735	-0.741	-0.726
-0.77	-0.816	-0.754	-0.866	-0.713	-0.797
-0.734	-0.742	-0.844	-0.752	-0.768	-0.761
-0.889	-0.87	-0.73	-0.76	-0.754	-0.74
-0.76	-0.743	-0.885	-0.708	-0.759	-0.891
-0.764	-0.838	-0.728	-0.774	-0.76	-0.829
-0.762	-0.733	-0.788	-0.739	-0.764	-0.759
-0.819	-0.856	-0.708	-0.86	-0.887	-0.766
-0.698	-0.749	-0.693	-0.726	-0.822	-0.843
-0.86	-0.758	-0.71	-0.72	-0.781	-0.754
-0.738	-0.713	-0.775	-0.883	-0.715	-0.805
-0.768	-0.789	-0.771	-0.777	-0.819	-0.72
-0.756	-0.736	-0.757	-0.777	-0.759	-0.744
-0.775	-0.848	-0.869	-0.731	-0.852	-0.743
-0.739	-0.777	-0.85	-0.815	-0.75	
-0.816	-0.791	-0.767	-0.748	-0.778	
-0.759	-0.847	-0.771	-0.828	-0.753	

References

Alberts, C., Dorofee, A., Stevens, J., & Woody, C. (2003). Introduction to the OCTAVE approach*.* *CERT-CC.*

American Gas Association (AGA). (2006). Cryptographic protection of SCADA communications, part 1: background, policies, and test plan*. AGA 12 Part 1*.

Ameren. (2015). IT Service General Conditions. *IT Services Agreement.* Illinois.

American Petroleum Institute. (2005). Standard 1164, *Pipeline SCADA Security, 3rd Edition*.

Atlagic, B., Sagi, M., Milinkov, D., Bogovac, B., & Culaja, S. (2012). A way towards efficiency of SCADA infrastructure. *Engineering of Computer Based Systems (ECBS).*

Rysavy, O., Rab, J., Halfar, P., & Sveda, M. (2012). A formal authorization framework for networked SCADA systems. In *Engineering of Computer Based Systems (ECBS), 2012 IEEE 19th International Conference and Workshops on* (pp. 298-302). IEEE.

BICSI/InfoComm. (2006) *AV Design Reference Manual, first edition*.

Blum, J. (2013). Tools and techniques for engineering wizardry. *Exploring Arduino, 2013*, John Wiley & Sons, Inc., 10475 Crosspoint Blvd. Indianapolis, IN., ISBN 978-1-118-54936-0.

Bobat, A., Aslan, H., & Gezgin, T. (2005). The use of SCADA system in dam management. *The SCADA System Applications in Management of Yuvacik Dam and Reservoir*. *Desalination and Water Treatment,* 54(8):1-12.

British Columbia Institute of Technology. (2005). Industrial instrumentation process lab.

Byres, E., & Lowe, J. (2004). The myths and facts behind cyber security risks for industrial control systems. In: *VDE Congress, VDE Association for Electrical, Electronic & Information Technologies,* Berlin; October 2004.

Campbell, R. (2015). Cybersecurity Issues for the bulk power system. *Congressional Research Service, 2015.*

Campbell, P., & Stamp, J. (2004). A classification scheme for risk assessment methods. *SNL Report, August 2004,* SAND2004-4233.

Carlson, R. (2002). Sandia SCADA program – high-security SCADA LDRD final report. *Sandia National Laboratories report, April 2002,* SAND2002-0729; pp. 20.

Carlson, C. (2005). DHS to state its case to business, *eweek, Issue 42, October 31, 2005.*

Carlson, R., Dagle, J., Shamsuddin, S., & Evans, R. (2005). A summary of control system security standards activities in the energy sector. *National SCADA Testbed, October 2005,* 48 page report by National Laboratories.

Cervin, A., Ohlin, M., & Henriksson, D. (2003). Simulation of networked control system using TrueTime. *IEEE Control Systems Magazine, June 2003,* 23:3, pp. 16–30.

Chabukswar, R., Sino'poli, B., Karsai, G., Giani, A., Neema, H., & Davis, A. (2010). Simulation of Network Security Attacks on SCADA Systems.

Chandia, R., Gonzalez, J., Kilpatrick, T., Papa, M., & Shenoi, S. (2007). Critical Infrastructure Protection. *Security Strategies for SCADA Networks*. DOI 10.1007/978-0-387-75462-8_15. Bookmetrix.

Chen, D., Peng, Y., & Wang, H. (2013). Development of a testbed for process control system cybersecurity research. *3rd International Conference on Electric and Electronics* (EEIC 2013).

Chittester, G., & Haimes, Y. (2004). Risks of terrorism to information technology and to critical interdependent infrastructures, *Journal of Homeland Security and Emergency Management, Vol.1, Issue 4, 2004,* article 402.

Copley, S. (2015). *IGCSE ICT Certification*.

Computer Virus Brings Down Train Signals. (2003, August). Information Week.

Creery, A. Byres, E. (2005). Industrial cybersecurity for power system and SCADA networks, *IEEE paper,* PCIC-2005-34, pp. 303-309.

Crowther, K., Dicdican, R., Leung, M., Lian, C., Haimes, Y., Lambert, J., …Horowitz, B. (2004). Assessing and managing risk of terrorism to Virginia's interdependent transportation systems, *Final Contract Report to Virginia Transportation Research Council* (VTRC 05-CR6), Center for Risk Management of Engineering Systems, University of Virginia, Charlottesville, VA (October 2004), 56 pages.

Deng, S., Meliopoulos, S., Mount, T., Sun, H., Yang, F., Stefopoulos, G., …Cai, X. (2007). Modeling market signals for transmission adequacy issues: valuation of transmission facilities and load participation contracts in restructured electric power systems, PSERC Publication, 2007.

Department of Energy. (2014). Cybersecurity procurement language for energy delivery systems. Energy Sector Control Systems Working Group (ESCSWG).

Department of Energy. (2008). Common cyber security vulnerabilities observed in control system assessments by the INL NSTB Program, Idaho National Laboratory, Idaho Falls, Idaho, 83415, November, 2008.

Department of Energy. (2005). 21 steps to improve cyber security of SCADA networks.

Department of Energy. (2003). Final report on the implementation of the task force recommendations.

Department of Homeland Security and Department of Energy. (2010). Energy specific sector plan: an annex to the national infrastructure protection plan.

Department of Homeland Security. (2014). Advisory (ICSA-14-105-03) Siemens industrial products openSSL HeartBleed vulnerability.

Deveza, T., & Martins, J. (2009). PLC control and Matlab/Simulink simulations. *A Translation Approach* .IEEE. DOI, 978-1-4244-2728-4, September, 2009.

Duggan, D. (2005). Penetration testing of industrial control systems, *Report SAND2005-2846P*, Sandia National Laboratories, 2005.

Ezell, B. (1988). *Risks of cyber attacks to supervisory control and data acquisition for water supply*, Thesis, University of Virginia, May 1988.

Falco, J., Stouffer, K., Wavering, A., & Proctor, F. (2004). IT security for industrial control systems. Intelligent Systems Division National Institute of Standards and Technology (NIST), in coordination with Process Control Security Requirements Forum (PCSRF).

Federal Energy Regulatory Commission. (2005). Commission conducts investigation of Taum Sauk dam breach. (No. P-2277).

Federal Information Processing Standards. (2004). Publication 199.

Felser, M. (2001). History and structures. *The Fieldbus Standards*, Research Gate.

Franz, M. (2003). Vulnerability testing of industrial network devices. *Critical Infrastructure Assurance Group*, Cisco Systems, 2003.

Fyodor, G. (2013). Nmap network scanning. *The Official Nmap Project Guide to Network Discovery and Security Scanning*, April 2013. Insecure.Com LLC.

General Accounting Office (GAO). (1999). Report GAO/AIMD-00-33, Information security risk assessment practices of leading organizations, *A Supplement to GAO's May 1998 Executive Guide on Information Security Management*, November 1999, 50 pages.

GE. (2015). GE Power & Water Distributed Power.

Geer, D. (2006). “Security of critical control systems sparks concern.” *Computer*, January, 2006, pp. 20-23.

Gene Ontology Consortium. (2016). Biological Process Ontology Guidelines.

Giani, A., Karsai, G., Roosta, T., Shah, A., Sinopoli, B., & Wiley, J. (2008). A testbed for secure and robust SCADA systems. *SIGBED Rev. 5, 2*, Article 4 (July 2008), 4 pages. DOI=10.1145/1399583.1399587. Science Direct.

Greenfield, David. (2013). Wired & wireless, know your options for deploying what’s best for your application. *Industrial Networks*, Automation World.

Hancock, D. (2003). Virus disrupts train signal. CBS News.

Haimes, Y. (1981). Hierarchical holographic modeling, *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 11, No. 9, pp. 606-617, 1981.

Haimes, Y. (1998). Risk modeling, assessment, and management. First Edition, New York: John Wiley and Sons, 1998.

Haimes, Y., & Chittester, C. (2005). A roadmap for quantifying the efficacy of risk management of information security and interdependent SCADA systems, *Journal of Homeland Security and Emergency Management*, Vol. 2, Issue 2, 2005, article 12.

Haimes, Y., Kaplan, S., & Lambert, J. (2002). Risk filtering, ranking, and management framework using hierarchical holographic modeling, *Risk Analysis*, Vol. 22, No. 2, pp. 381-395, 2002.

Haimes, Y., Lambert, J., Horowitz, B., & Santos, J. (2004). Assessing and Managing Risk of Terrorism to Virginia’s Interdependent Transportation Systems, Virginia Transportation Research Council Final Contract Report, VTRC 05-CR6, October 2004.

Henley, E., & Kumamoto, H. (1996). Probabilistic risk assessment, 2nd edition, IEEE Press, New York, 1996.

Henry, P. (2006). ICS cybersecurity response to physical breaches. *A How to Guide*.

Houmb, S., Franqueira, V., & Engum, E. (2009). Quantifying security risk level from CVSS estimates of frequency and impact. *The Journal of Systems and Software*, Article in Press, Corrected Proof, 2009, 1–31. Science Direct.

Idaho National Laboratory. (2011). vulnerability analysis of energy delivery control systems.

Igure, V., Laughter, S., & Williams, R. (2006). Security issues in SCADA networks. Science Direct, Elsevier.

International Electrotechnical Commission. (2003). IEC 62210 “Initial Report from IEC TC 57 ad-hoc WG06 on Data and Communication Security”.

International Electrotechnical Commission. (2013). IEC 61850 “Communication networks and systems in substations”.

International Electrotechnical Commission (2013). Power systems management and associated information exchange, data and communications security, Part 5: Security for IEC 60870-5 and derivatives.

Institute of Electrical and Electronics Engineers. (2008). IEEE 1402 *IEEE Guide for Electric Power Substation Physical and Electronic Security*.

International Standards for Organization. (2009). ISO/IEC 15408-1:2009, Information technology. *Security techniques-Evaluation criteria for IT security-Part 1: Introduction and general model*.

International Standards for Organization. (2005). Information technology-Security techniques. *Code of practice for information security management*.

James, J., Mabry, F., St. Leger, A., Cook, T., & Huggins, K. (2012). Cyber-physical situation awareness and decision support. Department of Electrical Engineering and Computer Science. United States Military Academy. West Point, NY, 10996.

Jurafsky, D., & Martin, J. (2006). An introduction to natural language processing, computational linguistics, and speech recognition. *Speech and language processing.*

Kabilan, D., & Manohar, S. (2013). MATLAB/Simulink based transmission line automation using multiagent system. *4th International Conference on Intelligent Systems, Modelling and Simulation*. 2166-0662/13. 2013 IEEE DOI 10.1109/ISMS.2013.75.

Kaplan, S., & Garrick, B. (1981). On the quantitative definition of risk. *Risk Analysis,* Vol. 1, No. 1, 1981, pp. 11-37.

Karvinen, K., & Karvinen, T. (2014). Measure the world with electronics, Arduino, and Raspberry Pi..*Make: Getting started with sensors,* MakerMedia. 1005 Gravenstein Hwy., Sebastapol, CA. 95472.

Kertzner, P. Bodeau, D. Nitschke, R. Watters, J Young, M., & Stoddard, M. (2006). P*rocess control system security technical risk assessment: analysis of problem domain.* Research Report No. 3, January 2006, I3P.

Kilman, D., & Stamp, J. (2005). Framework for SCADA security policy.

Kuphaldt, T. (2015). Lessons in electric circuits, Volume II, Chapter 11, Power Factor, true, reactive, and apparent power. Ibiblio. 2015.

Langner, R. (2013). A technical analysis of what Stuxnet's creators tried to achieve. *To Kill a Centrifuge.*

Lemos, D. (2005). Worm spreading through Microsoft plug-and-play flaw. *Security Focus*.

MathWorks. (2014). Simulink.

McQueen, M., Boyer, W., Flynn, M., & Beitel, G. (2006). Quantitative cyber risk reduction estimation methodology for a small SCADA control system, *Proceedings of the 39th Hawaii International Conference on System Sciences,* January, 2006.

Metasploit. (2014). Metasploit, world's most used penetration testing software. *Rapid7.*

Melton, R. Fletcher, T., & Earley, M. (2004). System protection profile-industrial control systems (SPP-ICS) Version 1.0, April 14, 2004, 151 pages,

Monk, S. (2013). Programming the Raspberry Pi, *Getting Started with Python*. McGraw-Hill, New York, New York. ISBN 978-0-07-180783-8.

Morris, T., Srivastava, A., & Reaves, B. (2011). A control system testbed to validate critical infrastructure protection concepts. *International journal of critical infrastructure protection,* vol. 4, pp. 88–104, 2011.

Morris, T., Srivastava, A., Reaves, B., Pavurapu, K., Abdelwahed, S., Vaughn, R.,… McGrew, W. (2010). Engineering future cyber-physical energy systems. *Challenges, Research Needs, and Roadmap*. Electrical and Computer Engineering Mississippi State University.

Multi-State Information Sharing & Analysis Center MSISAC. (2016). Center for internet security.

Nash, T. (2005). An undirected attack against critical infrastructure. *A Case Study for Improving Your Control System Security,* US-CERT Control Systems Security Center Document, September 2005, 11 pages.

Kean, T. H., & Hamilton, L. (2004). The 9/11 commission report: final report of the national commission on terrorist attacks upon the United States. Washington, D.C., National Cybersecurity and Communications Integration Center. (2013). *ICS-CERT Monitor*: October, November, December, 2013.

National Information Assurance Partnership. (2002). Process Control Security Requirements Forum (PCSRF).

National Institute of Standards and Technology (NIST). (2013). Security and privacy controls for federal information systems and organizations. *NIST Special Publication 800-53 Revision 4*. Joint Task Force Transformation Initiative.

National Institute of Standards and Technology (NIST). (2015). *Guide to Industrial Control Systems (ICS) Security.*

Nelson, T. (2005). Common control system vulnerability, US-CERT Document, November 2005, 7 pages.

Nessus. (2015). Tenable nessus scanner professional.

Nicholson, A., Webber, S., Dyer, S., Patel, T., & Janicke, H. (2012). SCADA security in the light of Cyber-Warfare. Science Direct, Elsevier.

North American Electric Reliability Corporation (2013). Critical infrastructure protection (CIP) standards. Revision 3.

North American Electric Reliability Corporation (2015). Critical infrastructure protection (CIP) standards. Revision 5.

Nozick, L., Turnquist, M., Jones, D., Davis, J., & Lawton, C. (2005). Assessing the Performance of Interdependent Infrastructures and Optimizing Investments, International Journal of Critical Infrastructures, Vol. 1, Nov. 2/3, 2005, pp. 144-154.

Ofualagba, G., & Ubeku, E. (2011). The analysis and modelling of a self-excited induction generator driven by a variable speed wind turbine. Intechopen.

Olympic Pipe Line accident in Bellingham kills three youths. (1999). Gas pipeline explodes. *The Bellingham Herald*. June 10, 1999. History Link.

OPC. (2015). Open platform communications. *The Interoperability Standard for Industrial Automation. What is OPC*?

Pak, C. (2008). The near real time statistical asset priority driven (nrtsapd) risk assessment. Proceedings of the 9th ACM SIGITE, Cincinnati, Ohio, 105–112.

Pak, C. (2011). *Near real-time risk assessment using hidden Markov models*. (Doctoral Dissertation). Retrieved from ProQuest Digital Dissertations. (UMI 3481329).

Pak, C., & Cannady, J. (2010). Risk forecast using hidden Markov models. Research in Information Technology (RIT), ACM SIGITE, 7(2), 4-15.

Pak, C., & Cannady, J. (2009). Asset priory risk assessment using hidden Markov models. Proceedings of the 10th ACM Conference on Sig-information Technology Education, Fairfax, Virginia, 65–73.

Patel, M. (1999), Wind and Solar Power Systems, Design, Analysis, and Operation. CRC Press, Taylor & Francis Grp. ch. 6.

Peterson, D. (2004). "Intrusion Detection and Cyber Security Monitoring of SCADA and DCS Networks, Presented at ISA Automation West, 2004.

Pivonka, E., & Mazzuchelli, E. (2005). Silence noise on your network. Ensure measurement accuracy for LAN testing by avoiding the negative effects of EMI and improper grounding. *Electrical Construction and Maintenance*. Jan 01, 2005.

Poulsen, D. (2003). Slammer worm crashed Ohio nuke plant network. Security Focus.

Radvanovsky, R., & McDougall, A. (2009). Critical Infrastructure. Homeland Security and Emergency Preparedness, Second Edition. CRC Press 2009. Pages 163–189. ISBN: 978-1-4200-9527-2. DOI: 10.1201/9781420095289-c8.

Ralston, P., Graham, J., & Patel, S. (2006). Literature review of security risk assessment of SCADA and DCS systems. Intelligent systems research laboratory. Technical report TR-ISRL-06-01.

Ralston, P., Graham, J., & Hieb, J. (2006). Cyber security risk assessment for SCADA and DCS networks. Elsevier, Science Direct. ISA Transactions 46 (2007) 583-594. October 2006.

Ravindranath, R. (2006). Smartgrid supervisory control and data acquisition (scada) system security issues and counter measures. Visveswaraiah Technological University, Karnataka, India.

Rinaldi, S. (2004). Modeling and simulating critical infrastructures and interdependencies, Proceedings of the 37th Hawaii International Conference on System Sciences.

Riskwatch. (2002). How to do a complete automated risk assessment: A Methodology Review, White Paper.

Risley, A., Roberts, J., & LaDow, P. (2003). Electronic security of real-time protection and SCADA communications. In: Fifth annual western power delivery automation conference, Spokane, Washington; April 1–3, 2003.

Rockwell. (2015). Rockwell Automation. Support.

Sauer, P. (2003). What is reactive power? Power Systems Engineering Research Center (PSERC). PSERC Background Power.

Sandia National Laboratories. (2003). The Center for SCADA Security Assets.

Schneider, G., Lima, V., Scherer, L., Camargo, R., & Franchi, S. (2012). Supervisory and control system development applied to distributed generation using Web tools. Research Gate. Conference Paper 2011.

Singer, B., & Weiss, J. (2006). Control Systems Cyber Security, Control Engineering.

Smith, T. (2001) Hacker jailed for revenge sewage attacks. The Register.

Söderqvist, M. (2104). Maintenance and safety of aging infrastructure. Edited by Dan Frangopol and Yiannis Tsompanakis. CRC Press, 2014. Pages 621–640. Print ISBN: 978-0-415-65942-0. DOI: 10.1201/b17073-21

Soetedjo, A., Lomi, A., Nakhoda, Y., & Tosadu, Y. (2013). Combining web SCADA software and Matlab-Simulink for studying wind-PV-battery power systems. International Journal of Computer Science Issues, Vol. 10, Issue 2, No 2, March 2013.

Stamatelalos, M. (2002). Probabilistic risk assessment procedure guide for NASA managers and practitioners, Report by NASA Office of Safety and Mission Assurance, August 2002, 323 pages.

Stamp, J., Dillinger, J., Young, W., & DePoy, J. (2003). Common vulnerabilities in critical infrastructure control systems.

Stefanini, A., Puppin, S., & Servida, A., (2005). Electric system vulnerabilities. *The crucial role of information & communication technologies in recent blackouts*.

Stoddard, M., Bodeau, D., Carlson, R., Glantz, C., Haimes, Y., Lian, C., …Santos, J. (2005). Process control system security metrics-state of practice. *I3P Research Report No.1*, August 2005, I3P.

Stouffer, K., Falco, J., & Scarfone, K. (2013). SP 800-82. Guide to industrial control systems (ICS) security. *Supervisory Control and Data Acquisition (SCADA) systems, Distributed Control Systems (DCS), and other control system configurations such as Programmable Logic Controllers (PLC)*, National Institute of Standards & Technology, Gaithersburg, MD, 2011.

Stouffer, K., Pillitteri, V., Lightman, S., Abrams, M., & Hahn, A. (2015). SP 800-82r.2. guide to industrial control systems (ICS) security. *Supervisory Control and Data Acquisition (SCADA)* *systems, Distributed Control Systems (DCS), and other control system configurations such as Programmable Logic Controllers (PLC)*, National Institute of Standards & Technology, Gaithersburg, MD, 2011.

Strickles, R., Ozog, H., & Mohindra, S. (2003). Security vulnerability assessment, ioMosaic Corporation White Paper, 10 pages.

Swartz, S., & Assante, M. (2014). Industrial control system (ICS) cybersecurity response to physical breaches of unmanned critical infrastructure sites. A SANS Analyst Whitepaper. January 2014

TradesInfo. (2015). Instrumentation and Control Technician Terminology.

Teen hacker faces federal charges. (1998). Caused computer crash that disabled Massachusetts airport. CNN.

Tolbert, G. (2005). Residual risk reduction, professional safety, *Modbus and DNP3 Communication Protocols.* November 2005, pp.25-33.

Triangle MicroWorks. (2016). Raleigh, North Carolina.

Trellue, R. (2005). I3P SCADA security research plan summary. *Control systems architecture analysis services,* May 20, 2005. United States Computer Emergency Readiness Team. (2015).

Wired. (2016). New discovery around Juniper backdoor raises more questions about the company. Wired Magazine. Kim Zetter. January 01, 2016.

Wonderware. (2007). Securing industrial control systems. *A guide for properly securing Industrial Control Systems operating in a Microsoft Windows environment. Revision 1.4,* Last Revision: 4/12/2007 Wonderware Invensys Systems, IncISA Transactions 46. (2007) 583–594

Yusta, J., Correa, G., & Lacal-Arantegui, R. (2011). Methodologies and applications for critical infrastructure protection: State-of-the-art. Science Direct, Elsevier.