Programming an Automated pH and Flow Control System for Use in an Industrial Wastewater Pre-Treatment System

by

Kristen Fulk

A Report Submitted to the Faculty of the
Milwaukee School of Engineering
in Partial Fulfillment of the
Requirements for the Degree of
Master of Science in Engineering

Milwaukee, Wisconsin
November 2016

Abstract

This report explains a capstone project required for the completion of the Master of Science in Engineering (MSE) program at the Milwaukee School of Engineering (MSOE). The fundamental goal of the project was the programming of a complex automated pH control system in an industrial wastewater pre-treatment system associated with a metal plating plant. The project report serves as a case study of the implementation of a closed loop programmable logic controller (PLC)-based control system for a nonlinear process – in this case, pH levels in the industrial wastewater stream. The report focuses on the programming of the PLC, although each phase in the implementation of the pH control system – which entailed a large number of tasks – is described. The design phase focused on the analysis of power and electrical issues associated with the control system. The building phase covered issues associated with the control system's layout, wiring, and equipment. The programming phase, the focus of this report, featured activities associated with the programming of the system PLC and human-machine interface (HMI). In industrial wastewater treatment associated with plating facilities, a variety of technical solutions are available for treatment, including solutions for controlling pH levels. Moreover, a rich literature exists on pH control, which is a challenging control problem. This report includes a literature review that provides an overview of pH control solutions, along with an explanation of why a PLC-based solution was selected for this project, as well as the engineering principles that informed the programming. In the metal plating facility that is the focus of this report, both the implementation and the performance of the control system were successful. The PLC-based control system continues today to successfully maintain the metal plating plant's wastewater pH levels in a range that is compliant with state and federal environmental regulations.

Acknowledgments

The engineering staff at River's Bend Engineering were instrumental in this project. Michael Fulk, CAP (Certified Automation Professional), as the automation specialist, answered numerous questions and provided extensive instruction on the software used for the project. Richard Fulk, P.E., and Ryan Fulk, P.E., created the P&IDs, which were used in the project. Academic advisors for this project included Professor Gary Shimek, who provided significant help with literature research and with writing-related tasks, in addition to serving as an editor for the report, Dr. Kishore Acharya, who served as the primary advisor for the project, and Dr. Subha Kumpaty, who is the Master of Science in Engineering (MSE) program director at the Milwaukee School of Engineering. Without these gentlemen, this project could not have been completed.

Table of Contents

List of Figures

List of Tables

Nomenclature

CAD – Computer Aided Design

CFR – Code of Regulations

CWA – Clean Water Act

EPA – Environmental Protection Agency

FBD – Functional Block Diagram

HMI – Human-Machine-Interface

HOA – Hand-Off-Auto

IEC – International Electrotechnical Commission

IWWTP – Industrial Wastewater Treatment Plant

MSE – Master of Science in Engineering

MSOE – Milwaukee School of Engineering University

NEC – National Electric Code

NEMA – National Electric Manufacturing Association

P&ID – Process and Instrumentation Design

PID – Proportional-Integral-Derivative

PLC – Programmable Logic Controller

PO – Purchase Order

POTW – Publicly Operated (Owned) Treatment Works

WWTS – Wastewater Treatment System

Introduction

This Master of Science in Engineering (MSE) capstone design project concerns the designing, building, and programming of an automated control panel required to maintain the necessary pH levels and flows at different steps in a process of removing metals from the waste stream of a metal plating facility. This treatment is required in order for the plating facility to remain in compliance with the regulations for levels of nickel and pH set forth by the Environmental Protection Agency (EPA) and the associated local Publicly Operated Treatment Works (POTW). This project introduces the problems pertaining to adjusting and maintaining pH levels in wastewater as it is being treated in an industrial wastewater pre-treatment system of an electroplating facility.

The process of metal plating requires contact rinse water and cleaners. Metals and cleaners enter the water as it is sent through the plating process. If these metal-laden waters were to enter lakes, rivers or streams, or even a POTW, they could do harm to those environments. Regulations have been established by government agencies that set limits on the amount of metals that can be released into those environments. It is the responsibility of an industrial facility, then, to pre-treat wastewater and to maintain the proper levels of metals in their effluent flow, so none are accidently released. This is a challenging undertaking for many facilities, as their waste streams may change from time to time, introducing fluctuating levels of regulated metals into their wastewater. Also, in traditional flow-through, pH-adjusted systems, a clarifier – or settling tank -- is used as a way of removing solids from the water. This being the case, it is necessary to maintain a flow out of which the clarifier can successfully filter the solids. To solve this problem, a variety of technical solutions are available, but a common approach is to apply chemical concepts and technology to maintain pH levels and flow by way of a control system. It is the job of the electrical engineer to design, build, and program an automated control system that can respond to the complex and varying parameters which exist in such a treatment system.

The purpose of this report is to explain in particular the programming associated with an automated pH and flow control system that was implemented in an industrial wastewater pre-treatment system. The implementation of a pre-treatment industrial wastewater treatment system is a highly complex endeavor, typically approached in terms of three stages. In the first stage, using a variety of considerations -- such as the characteristics of the industrial wastewater that needs to be treated, the treatment requirements associated with the industrial plants, and so on -- designers design the system. The building of the system is the next stage, and it entails the consideration of a number of factors, such as cost data, regulatory requirements, and so forth. The third stage entails programming requirements. This capstone project report explains the programming activities associated with a pH and flow automated control system, which is part of an industrial wastewater pre-treatment system that was implemented in an industrial metal plating plant. The focus of this report is on the programming phase utilizing modern control theory. Control theory was used as a framework for the program. The control and program details are provided, along with descriptions and explanations of what was done, and why it was done. This project report also contains a background section that explains important concepts and topics, as well as a literature review that discusses relevant developments with respect to pH control.

Background

Industrial Wastewater

Wastewater is produced in a myriad of ways, but it is useful, first, to classify wastewater in terms of its point of origin. Domestic wastewater – or sewage – is "wastewater discharged from sanitary conveniences in residential, office, commercial, factories and various institutional properties" [1]. Domestic wastewater consists primarily of water along with organic (e.g., human wastes) and inorganic constituents (e.g., various forms of nitrogen) [1]. A significant problem associated with domestic wastewater is the presence of micro-organisms, which are in human wastes, and which can produce a number of harmful bacterial and viral diseases, such as cholera, typhoid, tuberculosis, and hepatitis [1].

Industrial wastewater, on the other hand, is wastewater associated with raw material processing and manufacturing [1]. Examples of industrial wastewaters include those occurring in association with chemical, pharmaceutical, electrochemical, electronics, petrochemical, and food processing industries [1]. The composition of industrial wastewater depends on the industry or materials associated with the wastewater, but such wastewaters can contain harmful or toxic chemicals, metals, and other dangerous pollutants and hazardous materials.

Industrial Wastewater and pH Levels

Moreover, industrial wastewaters can be associated with extreme pH levels. The pH level is a measure of the acidity or the alkalinity of a liquid solution. Generally, a liquid solution featuring a pH level less than 7 is an acid; a solution with a pH level greater than 7 is an alkaline or base [2]. Both acids and bases are classified as corrosive

[2]. Depending on the strength and concentration of a material's pH levels, the material – either an acid or a base – can be associated with five basic hazards, including (1) skin or tissue damage, (2) hazardous vapor inhalation, (3) dangerous reactions with other materials, (4) flammability, and (5) instability and toxicity [2]. Unlike sewage wastewater, which typically features pH levels in the range of 6 to 9 – which are suitable for biological processes – industrial wastewaters can be associated with significant acidity or alkalinity [1].

Environmental Regulations and Industrial Wastewater

In the United States, in compliance with the Clean Water Act (CWA), all wastewater must be treated before it is either returned to the environment or disposed of in some other manner so that the water does not pose a hazardous risk to human beings or to the environment [3]. The U.S. Environmental Protection Agency (EPA) is responsible for enforcing the laws and regulations associated with the CWA [3]. Each industry which could potentially pose risks to the environment via water disposal has specific regulations for permissible limits. These specific regulations are found in the Code of Federal Regulations (CFRs). Wikipedia [4] says "The Code of Federal Regulations (CFR) is the codifications of general and permanent rules and regulations (sometimes called administrative law) published in the Federal Register by the executive departments and agencies of the federal government of the United States. The CFR is divided into 50 titles that represent broad areas subject to federal regulation." Title 40 is devoted to protection of the environment, consisting mainly of environmental regulations promulgated by the EPA. Title 40 consists of five chapters, each chapter is devoted to the regulations promulgated by five different organizations, including the EPA and the

Department of Justice, the Council on Environmental Quality, the Chemical Safety and Hazard Investigation Board, and the EPA and the Department of Defense. Each chapter consists of subchapters, which in turn consist of parts and subparts. For example, Title 40, Chapter 1, is devoted to EPA regulations, and in Chapter 1, Subchapter N is devoted to 'Effluent Guidelines and Standards" associated with the Clean Water Act. Chapter N features Parts 400 through 471, covering general pre-treatment regulations, as well as, "point source categories" – that is, specific sources of effluent, such as dairy operations, grain mills, textile mills, iron and steel manufacturing, and so on. Part 413 is devoted to regulations concerning effluent that is produced in electroplating operations.

The Plating Industry and Wastewater

One type of industrial wastewater is associated with the plating industry. In the plating industry, two fundamental types of plating technology are employed. They include electroless plating and electroplating. Electroless plating is a chemical process that deposits a coating of metal on a surface without the use of an electrical current for the purposes of enhancing wear resistance, hardness, and corrosion protection, as well and decoration [5]. Electroplating, on the other hand, is employed to create a thin surface coating of one metal on another metal by a process known as electrodeposition [5]. Electroplating, additionally, is used in order "to provide corrosion protection, wear or erosion resistance, antifrictional characteristics, or for decorative purposes" [5]. Whether or not an electroless plating, or electroplating, process is employed depends on a number of factors, including the intended application of the treated material.

A significant amount of water is typically employed for bathing and rinsing purposes in an electroplating operation [5, 6, 7]. Cherry [5] points out that the water is employed “to remove the film (fluids and solids) that is deposited on the surfaces of the workpieces during the preceding process. As a result of this rinsing, the water becomes contaminated with the constituents of the film” [5]. Because of this contamination, the rinsing water must be treated by an industrial plating facility before it can be safely discharged. Industrial plating facilities, therefore, maintain industrial wastewater treatment plants (IWWTPs) or industrial wastewater treatment systems (WWTSs) – one such system is the focus of this capstone project report.

The wastewater generated by a plating operation is actually more properly characterized as a sludge – that is, a mixture of water and other solid matter [6, 7]. In a complex process – referred to as a train – the sludge, or wastewater stream, is required to travel through various unit processes, each with a number of refining purposes. For example, one unit process works to separate the solid matter and the water. Plating sludge is typically classified as a risky or hazardous waste, and it is dangerous and toxic [6, 7]. The reason for its toxicity is that it typically contains heavy and transitions metals, such as nickel and chromium [7]. Among other factors, the successful removal of these metals depends heavily on the surrounding environmental pH values [7]. Therefore, a key function of a plating operation WWTS entails the maintenance of proper pH levels during its various treatment processes [8].

The maintenance of proper pH levels in a plating IWTP is challenging, often requiring vigilant monitoring by trained personnel [8]. Many industrial WWTSs employ control systems to automate the process of maintaining proper pH levels, but depending on whether or not the system is “’well behaved’ (the pH of the wastewater entering the mixing tank does not change very often, or very much)” [8], the control process can be exceedingly complex, requiring, in fact, multiple complete control systems, each with their valves, pumps, motors, and power systems [8]. Many factors influence the behavior of a waste steam. NG [1] observes that "Automatic pH correction can be unexpectedly difficult activity to perform satisfactorily. This is, in part, because of the difficulty associated with mixing a small quantity of reagent uniformly with a large volume of wastewater. This is made even more difficult if the wastewater characteristics, such as flowrate, change rapidly."

Meeting the Challenge of Controlling Treatment System pH Levels

This capstone project describes the deployment of a programmable logic controller (PLC) in a plating WWTS in order to automate the control of the pH levels in the waste streams. A variety of technical solutions are available for treating industrial wastewater associated with electroplating, and including the control of pH levels. Weber [9] provides a review of conventional methods. However, a number of non-conventional methods have also been investigated. For example, Low, Lee, and Leo [10] report on the use of banana pith (*Musacea zingiberales*) to sorb metal ions from electroplating waste under both batch- and continuous flow conditions. Qin, Wai, Oo, and Lee [11] report on the successful use of an ultrafiltration (UF) reverse osmosis (RO) approach in a nickel-plating operation. Vijayaraghavan and Yun [12] review the use of biosorption for the removal of metals in plating industrial wastewater, and Suzuki *et al.* [13] report on the use of biological activated carbon treatment of effluent water from wastewater treatment processes in the plating industry. In the literature review of this project report, an overview is provided of technical solutions for pH control, from traditional approaches to more recent methods that employ artificial intelligence (AI). It is intended that this overview will contribute to the necessary background for an explanation of why a PLC-based solution was selected for this project, as well as the engineering principles that informed the programming phase. Among the reasons for the selection of a PLC-based solution is the level of control afforded by the technology. Control of the complex variables and interactions in a plating system – particularly, pH levels – is a significant challenge [14, 15].

As each industrial facility uses water in its own way (e.g., contact cooling water and contact rinse water), many different combinations of pollutants exist. And as each categorical waste metal must be removed from the wastewater, in accordance with 40 CFR PART 413 SUBPART 413.14 (c) [16], the WWTS must be designed, built, and commissioned for this purpose. The particular electroplating facility, which is the focus of this capstone project, discharges at least 72,000 gallons of wastewater per day to the local POTW. This number was calculated by multiplying the highest number of gallons per minute (GPM) the plate clarifier (a large tank that settles out sludge) can handle (in this case, 50 GPM) to 1440, which will result in the maximum gallons per day the system could discharge. Because the discharge is above 38,000 gallons per day, it is the responsibility of the facility to maintain the limits set forth in Table 1, which features the limits of each pollutant metal found in many electroplating facilities. One obvious justification for this project, therefore, is that it ultimately contributes to a healthier and safer environment.

Table 1: Limits of Common Metals Present in an Electroplating Facility's Effluent Flow for Dischargers over 10,000 Gallons per Day [16].

Subpart A—Common Metals Facilities Discharging 38,000 Liters or More Per Day PSES Limitations (mg/l) **Pollutant or pollutant property**	**Maximum for any 1 day** (mg/l)	**Average of daily values for 4 consecutive monitoring days shall not exceed** (mg/l)
Cyanide (Cn), T	1.9	1.0
Copper (Cu)	4.5	2.7
Nickel (Ni)	4.1	2.6
Chromium (Cr)	7.0	4.0
Zinc (Zn)	4.2	2.6
Lead (Pb)	.6	.4
Cadmium (Cd)	1.2	.7
Total metals	10.5	6.8

In addition, typically, in industrial wastewater treatment, each WWTS is designed specifically for a particular facility, as each waste stream is unique. The design takes into consideration not only the chemistry and flow required by the facility to maintain compliance, but other factors are also evaluated, such as the size of the space allotted and the cost of the system. Thus, the design challenges associated with an industrial WWTS are both unique and complex. Though there are many plating facilities like the one in this project, the design specifications in this project are unique. No other plating facility has exactly the same waste stream. Therefore, another justification for this project in one sense is straightforward: As there are no WWTSs which exist as off-the-shelf models, large enough or specific enough, to remove the particular pollutant metals from the wastewater produced in this facility, it is necessary for *this* WWTS to be designed and installed. That being said, if this WWTS is to be automated, there needs to be a made-to-order control system, also unique to the needs of the client, to maintain and monitor the pH and flow of the system, to remove the categorical metals from the waste stream in order to keep the effluent flow within compliance.

With respect to the PLC-based solution in this project, Cox [17] observes that a "PLC is a solid-state system designed to perform the logic functions previously accomplished by components such as electromechanical relays, drum switches, mechanical timers/counters, etc. for the control and operation of manufacturing process equipment and machinery." The National Electric Manufacturing Association (NEMA) additionally defines a PLC as "a digital electronic apparatus with a programmable memory for storing instructions to implement specific functions, such as logic, sequencing, timing, counting, and arithmetic to control machines and processes" [17]. Cox [17] notes that a PLC is designed to be operated by plant engineers and maintenance personnel with limited knowledge of computers, and that it is also designed to operate in the industrial environment with wide ranges of ambient temperature, vibration, and humidity, and is not usually affected by electrical noise that is inherent in most industrial locations. These characteristics make a PLC control system an ideal choice for the automated control of a WWTS.

Control Systems Used to Meet the Challenge of pH and Flow Control

In the metal plating plant that served as the context for this project, electrical engineers determined the function of the control system required to perform all the processes that civil engineers defined. A system, as described by Katsuhiko Ogata, “is a combination of components that act together and perform a certain objective” [18]. All of the equipment, the pH levels, and other specifications in the process are the components that have been put together for one purpose: to remove nickel from wastewater. Properties of that water, such as levels, pH values, and flow rate are all variables which need controlling in order to remove the nickel. Ogata explains, “The controlled variable is the quantity or condition that is measured and controlled. The control signal is the quantity or condition that is varied by the controller so as to effect the value of the controlled variable.” [18] Ogata goes on to observe that “Control means measuring the value of the controlled variable of the system and applying the control signal to the system to correct or limit deviation of the measured value from the desired value.” [18] This is the foundation on which the control system program for this project has been written.

In this project, since it is necessary to maintain a flow in gallons per minute (GPM) that the clarifier is rated for, the power and speed commands for the main feed pumps -- in this case, magnetically coupled centrifugal pumps – are routed through variable frequency drives (VFDs). An operator-selected flow rate is compared to the actual flow rate and the PLC program can adjust the frequency output sent to the VFD. The VFD then completes two power conversions, which in turn are sent to the induction motor of the pumps. This power determines at what revolutions per minute (RPMs) the impeller of the pump will turn, which is directly proportional to the flow rate out.

The pH control of the system works differently, as there are no analog control capabilities on the chosen chemical pumps. Digital controls and Boolean logic, therefore, are used to achieve the pH control necessary for the system to function.

In the control system in this project, control panels are electrical enclosures which "Employ materials and components that are determined to be usable in the application" [19], in this case, equipment needed for pH control. The enclosure is where the PLC and its ancillary equipment reside. The control panel also houses the Human-Machine Interface (HMI) touch screen. The HMI allows the operator to easily control the system from a touch screen in a centralized location.

The main goal of this project was to implement automated pH and flow control of a WWTS for the purpose of removing nickel from the wastewater streams associated with an electroplating facility. This goal was determined to be complete when evidence that the wastewater stream of the plating facility, after treatment, had reduced the amounts of nickel in the effluent to within the categorical pretreatment standard as set by the EPA, so that the associated publicly operated treatment works can safely accept the wastewater.

Review of Literature

Control of pH Values in Industrial Process Applications

In order to understand the significance of the work that was completed in this project, it is useful to review relevant literature concerning the control of pH values in various process applications. Rani [20] observes that pH "plays an important role in chemical and biochemical processes because the chemical and biochemical reactions yield desired products only at certain specified pH values or over a narrow pH range." Rani [20] and Singh, Bhanot, and Mohanta [21] all point out that pH control, accordingly, is an extremely important problem in many industrial process applications, such as wastewater treatment, boiler feedwater treatment in thermal power plants, food processing plants, pharmaceutical operations, bio-transformation processes, and various chemical processing plants. However, the pH control problem is notoriously challenging. Tan, Lu, Loh, and Tan [22] write that

> (t)he task of regulating the pH value in an acid-base titration process is a challenging one because the neutralization relationship, the "S-shape" curve, has regions of extreme sensitivity and insensitivity. Variations in the titration curve with changes in the feed conditions further complicate the control problem.

Salehi, Shahrokhi, and Nejati [23] additionally observe that because of "highly nonlinear characteristics, time-varying nature, variations in titration curve and inaccessible state measurements, pH control is a challenging problem." Finally, Jebarani and Rammohan [24] note that in pH control,

> (t)he selection of the control mechanism is fraught with difficulties because the control system design for pH neutralization is technically difficult to implement, owing to its non-linear responses, sensitive environment, uncertain results, and large number of requirements. Classical hold good only for linear, theory-based processes. These simple mechanisms do not work well when applied to constantly changing chemical systems with complex kinetic and thermodynamic reactions.

Because of the daunting challenges associated with the control of pH values in processes, the pH control problem has attracted significant research interest over the years in an effort to develop control solutions. In fact, the pH control continues to generate research in what is now a large and flourishing research field associated with an immense literature. In a review of the field, Rani [20] indicates that the pH control problem features four fundamental control approaches, including adaptive control, predictive control, nonlinear control, and artificial intelligence (AI)- based control.

Briefly, in an adaptive control system, a controller – such as a Proportional-Integral-Derivative (PID) controller, including a programmable logic controller (PLC)[1] -- must adapt to the changing parameters of a controlled system by making adjustments. In adapting to changing parameters, a controller employs various types of feedback data, which are obtained from system or process sensors. The controller effects adjustments based on its own control parameters, which include setpoints, or the stable parameters and values that are desired in a controlled system. The controller's control parameters essentially serve as instructions for how the controller needs to respond to the changing

[1] A PID control loop can be implemented in a PLC.

parameters in a controlled system. Tuning a controller entails the adjustments of its control parameters. Tuning can be complex. A variety of tuning methods have been developed, including manual and advanced self-tuning methods [25]

In predictive control, however, rather than responding in an adaptive manner, a control system employs a model "to predict the process behavior over a prediction horizon and this information is employed while minimizing an objective function subject to the specified constraints to find the control action over another time horizon called the control horizon" [20].

Rani [20] indicates that the two nonlinear control approaches employed in pH control include Generic Model Control (GMC) and Globally Linearizing Control (GLC). GMC employs a process dynamic model to derive an analytical control law and GLC

> is based on finding a nonlinear transformation for the process inputs and outputs with the help of Lie derivatives so that the resulting equations have a linear dynamic input-output relationship. A PI controller is then designed for the resultant linear system. [20]

Finally, Rani [20] indicates that artificial intelligence (AI)-based control strategies increasingly are being investigated for pH control. Singh, Bhanot, and Mohanta [21] report that these control methods include neural networks, fuzzy logic, genetic algorithms (GAs), "and their combinations."

Many factors must be weighed in selecting in the proper controls approach in any given pH control scenario, but in addition to financial and human resource constraints, as well as process-specific goals (e.g., remote control) and impacts (e.g., regulations), Rani

[20] reports that "the choice of a controller ultimately rests on the simplicity, ease of tuning, and reliability of the controller over a wide range of operating conditions." Because of its reliability and durability in rugged environments, and its ease of use, PLC control has been widely deployed for automatic process control in many wastewater treatment plants for a number of years [26].

pH Control with Programmable Logic Controllers (PLC)

A review of the patent literature reveals that programmable logic controllers (PLCs) have been deployed specifically for pH control purposes in industrial wastewater treatment systems, including treatment systems associated with plating applications. As such, the selection of a PLC-based solution is appropriate for the project described in this report.

For example, in "Methods for Treatment of Perchlorate Contaminated Water" (December 2012), Canzano, DelVecchio, Frisch, Watt, Loudon, and Webster [27] devised both systems and methods for removing perchlorate from water. Perchlorates are the salts derived from perchloric acid, and are associated with some industrial applications. Perchlorate has been classified as a contaminant. The systems in this patent consist of reactors comprising biomass for degrading perchlorate, and the operation of the systems can be controlled by logic specifications. As a result, the flow of water through the treatment systems – from the feed water source through the potable[2] effluent water – is controlled by PLCs. In addition, the pH of the feed water is also controlled by a PPLC at levels between 6 and 8.

---

[2] Potable water is safe drinking water.

In “Interlock-Control Modular Fenton Reagent Wastewater Treatment Method and Device” (December 2014), Wenping, Qiang, Qiang, Xiaodong, and Huadong [28] depict the design of a wastewater treatment system featuring adjusting tanks, pH value correction pools, and flocculation tanks, all controlled by PLCs.

In “Automatic Regulating Device of Paint Slag Wastewater in Painting Workshop,” (March 2013), Kun [29] depicts the use of a PLC, a touchscreen, a pH value online detector, and a pH value modifier metering pump as part of a device associated with the automatic control of pH in paint slag wastewater.

Yinghua, Ping, Hong, and Liang [30] in the 2004 patent, “Intermittent Automatic Process and Equipment for Treating Waste Water,” propose an automated batch wastewater treatment system. The system features a wastewater tank, a batch reactor, a dosing tank, a flocculant tank, and other components. PLCS are employed to control the entire treatment process. PLCs, additionally, are deployed to adjust the pH values of the wastewater – eventually to an emission standard range of 6 to 9. Yinghua *et al.* [30] point out that the system can be utilized in the electroplating industry.

Jun [31] in the 2013 patent, “Automatic pH Control Neutralization Device,” describes an automated process for the adjustment of pH values. The system is controlled by PLCs, and it is intended for use in a variety of industrial applications in which the pH value of a substance needs to be adjusted in order to match preset values, which can be determined by an operator.

Specifically with respect to wastewater treatment in the electroplating industry, Ran, Baosheng, and Cheng [32] describe such as system in their 2012 patent, “Comprehensive Electroplating Wastewater Treatment Method.” In the system, electroplating wastewater is divided into six classes, including ink wastewater, complex wastewater, pretreatment wastewater, cyanogen-containing wastewater, chromium-containing wastewater, and common electroplating wastewater. The centralized control system features both manual control as well as PLC-based control.

In his September 2009 patent, “Electroplating Reclaimed Water Recycling and Treating System,” Ng [33] proposes an electroplating reclaimed water recycling and treatment system, featuring a pH-adjusting device. The entire system is controlled by PLCs, with a pH value setpoint of 7.

Finally, in their March 2010 patent, “System for Removal of Metals from Aqueous Solutions,” Boren and Hammel [34] describe a system and method for the removal of metals from “aqueous solutions,” including “residential or industrial aqueous streams.” An aqueous solution is a solution in which the solvent is water. Electroplating entails immersion in an aqueous solution that contains the metal employed in the plating process. Electroplating wastewater, therefore, typically features toxic metals. In the system proposed by Boren and Hammel [34], a PLC controller is employed. Boren and Hammel [34] state that

> (t)he controller receives inputs from various probes and readers and converts them into ladder logic language that would be used by an internal control loop, such as a proportional-integral-derivative (PID) loop, to individually and simultaneously monitor system operational parameters and to reconcile the inputs with predetermined or computer-generated calculated setpoints for the operational

parameters, such as temperature, pressure, Eh, and pH levels, sorbent loading and target pollutant removal or capture rate.

pH Control with Artificial Intelligence (AI)

Rani [20] indicates that with respect to pH control, the two main Artificial Intelligence (AI)-based approaches include fuzzy logic control and control by neural networks. Singh, Bhanot, and Mohanta [21] also report that genetic algorithms increasingly are being deployed in pH control.

With respect to pH control, many fuzzy logic control approaches that are reported in the literature typically feature a combination of PID control and fuzzy logic. For example, Jebarani and Rammohan [24] describe a pH neutralization process featuring a classic PID control mechanism that is improved and optimized by the use of a fuzzy logic controller. Jebarani and Rammohan [24] experimentally verified the performance of the combined controller system, with good results between input and output. A minor delay in the output (i.e., controller response) with respect to changing value setpoints was traced to mechanical issues associated with flow valves.

Salehi, Shahrokhi, and Nejati [23], on the other hand, propose an adaptive nonlinear fuzzy logic control system for pH control that is not deployed in combination with PID control. Salehi *et al.* [23] developed their system to address challenges associated with previously proposed pH neutralization nonlinear controllers. For example, many previous designs require “state variables for implementation,” “which usually are not available in practice.” In addition, “some designs are practical but very complex and require time-consuming calculations,” and other systems feature “a large number of tuning parameters" that make their application "impractical" [23]. After presenting a mathematical model of pH processes, Salehi *et al.* [23] offer "a simple fuzzy adaptive pH measurement for implementation." Salehi *et al.* [23] successfully simulated their control design, demonstrating a satisfactory pH setpoint tracking response. Using a bench-scale pH setup, the researchers then experimentally verified the performance of their design, using a PI controller for comparison purposes. The experimental results demonstrated that their "proposed controller tracks the desired setpoint trajectory quite well" [23].

Recently, researchers have proposed a pH control system that combines fuzzy logic control with the use of genetic algorithms (GAs). Genetic algorithms can be described as computer programs that are designed to seek optimal solutions for problems with respect to variables associated with the problems [35, 36]. A genetic algorithm essentially leverages the well-established scientific theory of evolutionary natural selection to reach an optimal solution [35, 36]. By deploying computing power, a genetic algorithm executes thousands of trial-and-error runs of combinations of variables associated with a problem. With each iteration, the most successful combinations are retained, and then re-combined and re-executed until an optimal solution is reached. As an example of the power of this process, Walbridge [36] discusses the use of a genetic algorithm in the design of a truss, which is a frame of struts that supports a bridge, roof, or other structure. The problem entails producing "a design that can support the required load with a minimum of material" [36].

> In the program, a 10-member truss is represented by a string of 10 numbers, each number being the diameter of a particular strut. The program starts by randomly

> generating several hundred strings incorporating different strut diameters. An ordinary computer-aided design system then evaluates each string by calculating the weight of all the struts and the load they can carry. If the truss can support the required load, then the lower its weight, the higher the score. Of course, these first strings score poorly; some trusses are too weak, others too heavy. But as the genetic algorithm saves the better strings and recombines them, the scores rise. Eventually, they reach a plateau, signaling that the design is not likely to get better. Implicit parallelism allows the genetic algorithm to zero in on a good solution – a strong but light truss – in a reasonable time. [36]

In the same manner, genetic algorithms recently have been applied by researchers to the pH control problem. For example, Singh, Bhanot, and Mohanta [21] proposed a self-organized fuzzy logic controller for pH control. Their controller, however, is optimized by a performance correction table, whose elements in turn are optimized by a genetic algorithm. Tan, Lu, Loh, and Tan [22] also proposed the use of a genetic algorithm in their pH control system. In their design, a genetic algorithm is employed to develop a titration model, which in turn is used in a Wiener model controller that linearizes the pH process. A Wiener model is a framework for handling nonlinear problems, developed by the M.I.T. mathematician, Norbert Wiener, in his classic book, *Nonlinear Problems in Random Theory* [37]. Once the pH process is linearized, it can be controlled by a PID controller.

The application of genetic algorithms in pH control continues to be a rich and rapidly expanding research field. Altinten [38], for example, offers a detailed technical explanation of genetic algorithms in the context of pH control. In her work on predictive control of a pH neutralization process in a continuous flow tubular reactor, a model was developed that focused on pH and flow rate. The model's parameters were determined with genetic algorithms, with the goal of controlling the pH value at a given setpoint value when the process is subjected to variations in feed flow rate. Valarmathi, Devaraj, and Radhakrishnan [39] likewise successfully employed an enhanced genetic algorithm to tune a proportional-integral (PI) controller in a pH process. In their paper, Valarmathi *et al.* [39] provide a detailed technical description of the implementation of the genetic algorithm in their investigation. Finally, Mwembeshi, Kent, and Salhi [40] employed a genetic algorithmic strategy to optimize a model-based pH control methodology in a highly nonlinear pH neutralization reactor. Their approach offers an alternative to genetic algorithm-based PID tuning, with "much improved 'global' pH control over that using conventional PI controller" [40].

Description of Project

This project report principally describes the programming of a PLC-based pH control system control panel, enabling the implementation of the start-up phase of a WWTS in a metal plating facility. This panel provides power and control for all field equipment and will allow the operators an accessibility to that control via the HMI touch screen. The purpose of this WWTS is to remove waste metals from contact rinse water, which is collected from the plating process lines. This rinse water also includes cleaners, which makes this process more difficult. Therefore, a pH adjustment system is necessary, and it must be closely monitored at each step of the treatment process to ensure that the conditions are right for metals to be removed. The levels of the metals in the final effluent must be reduced to within the standards set forth by the EPA and the POTW, into whose system the wastewater will be released for final treatment.

Figure 1 shows a basic version of a general wastewater flow-through treatment system. The wastewater, which contains pollutant metals and cleaners held in solution, is fed into the system from the holding tank via the feed pump. Once in the first reaction tank, the pH of the water is lowered to break the hold the cleaners (chelators) have on the pollutant metals. This procedure is referred to as acid cracking. The water is gravity fed to reaction tank 2 where the pH is raised. At a high pH, the metals are able to come out solution, but they are very small and not very heavy as individual particles. When the water arrives at the flocculation tank, polymers are added to the stream, which attach to the metal particles and form large debris called floc. The water containing the floc flows into the clarifier, where it has a large surface area used to settle out the floc, as it is heavier and thus floats to the bottom. At the bottom of the clarifier, there is an auger that scrapes the bottom, moving the floc and a bit of water, now called sludge, which is pumped to a sludge holding tank. Meanwhile, the de-flocced water carries on to the final pH adjustment tank, where the pH is lowered to a neutral level. This water, free of large amounts of pollutant metals, is pumped out to the city. The sludge is sent through a filter press that creates the “cake” containing the pollutant metals. These are trucked off site to a landfill designed to handle such pollutants.

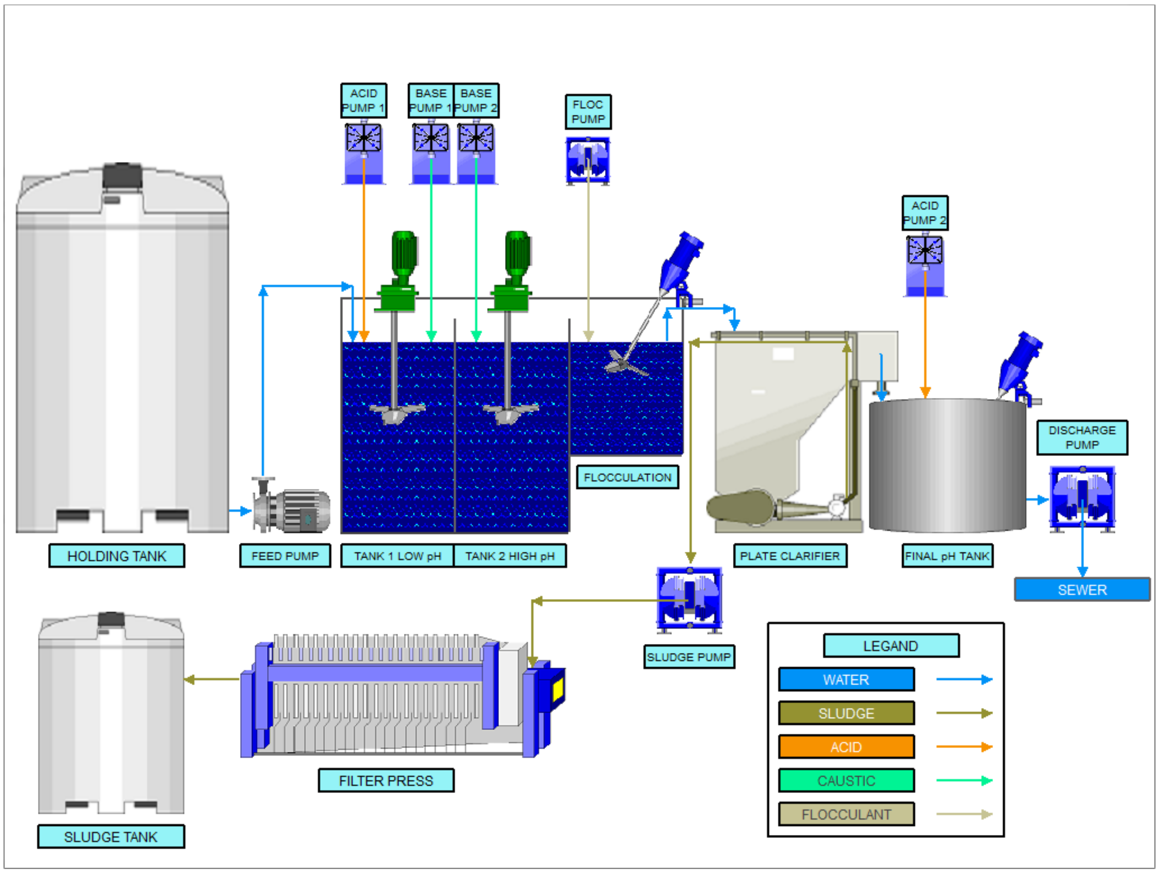

Figure 1: Complete Simplified Version of a Wastewater Treatment System.

The Process and Instrumentation Diagram (P&ID), as the name implies, diagrams the process (i.e., where the wastewater flows) and the instrumentation (i.e., all equipment required for the metal-removing system to work). No standard currently exists that defines and governs the content, symbolism, and format of a P&ID [41]. Every organization develops and uses its own P&IDs, and as a result, confusion and misunderstandings concerning the interpretation of P&IDs can occur [41]. For example, the “P” in the acronym sometimes stands for “piping” and sometimes for “process” [41]. However, conventionally, a P&ID is “widely understood to mean the principle document used to define the details of how a process works and how it is controlled” [41]. A P&ID, therefore, is a drawing or blueprint that shows “the components needed to run, monitor, and control specific processes” [42]. The P&ID typically includes the equipment, the piping that connects the equipment, and the lines and instruments used to monitor and control the process [42]. Conventionally, the P&ID features six parts, including the title block, the main diagram, the line schedules, the description of equipment, the description of issues or versions of the P&ID, and a notes area [42]. P&IDs are typically consulted in a number of situations, including troubleshooting, planning, and employee training scenarios [42]. Figure 2 shows an example of a simple P&ID with labels for the six standard components.

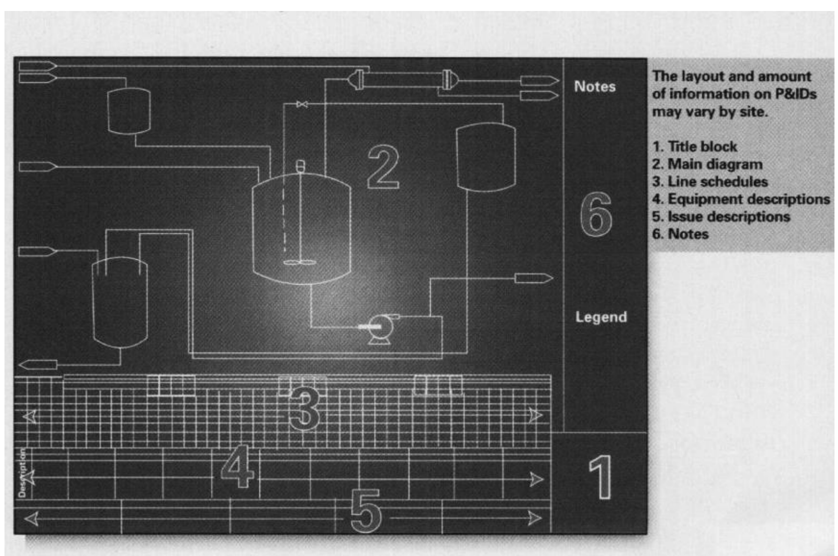

Figure 2: A Generic P&ID Diagram [42].

P&IDs typically are developed in an iterative fashion, and as such, version control is crucial [41]. Appendix A contains the P&ID drawings for this capstone project, which have been completed by the engineers at River's Bend Engineering, Richard J Fulk, P.E. and Ryan K Fulk, P.E., who employed the computer-aided design (CAD) program *Draft Sight* and the 3-D modeling software *Sketch-Up Pro* in the creation of the P&IDs.

The Design of the Control Panel

For the design of the control panel, the P&IDs were used to determine the equipment required inside the panel to control and power all of the field equipment (e.g., power supplies, fuses, PLC and I/O – input and output – cards). These diagrams are also analyzed to determine the input and output requirements for all field equipment (i.e., any equipment wired into the control panel from the outside). For example, a mixer would require a 3-Phase power source and an output from the PLC to tell it when to turn off and on. Also, most mixers employ overload protection circuitry that would inform the PLC if there is a problem with the mixer. Therefore, this mixer would also require an input to the PLC. These inputs and outputs are examples of digital signals, either on or off, to be handled by the PLC. Some field equipment utilizes analog signals, signals which have a level associated with them, not just on or off. The controlling field devices (e.g., pH meters, level elements and flowmeters) will input analog signals to the PLC.

All of these design considerations are taken into account as the electrical drawings are produced. The electrical wiring drawings for this project were created using the *Promis-e* electrical CAD software, the result of which is a bound PDF – all pages in one PDF – of the 11" X 17" electrical schematic (see Appendix B for copy).

The Build Phase of the Control Panel

In this project, the electrical schematics were used to build the control panel. The build entailed the purchasing of parts, the layout, and wiring of the control panel. The build begins with the layout, or the placement, of each piece of equipment that resides on the backplane of the panel. Figure 3 displays an example of a layout of equipment on the backplane of a panel.

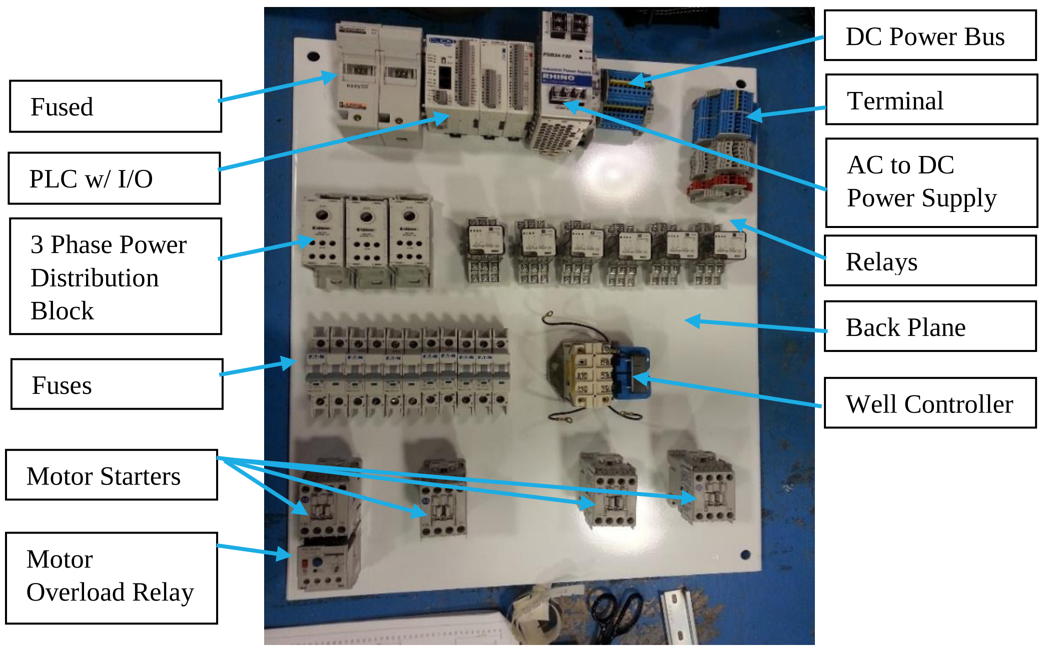

Figure 3: An Example of a Panel Layout on the Back Plane.

Once all components are placed, they are secured to the back plane using din rail or they are secured directly to the back plane. The placement of wire runs is also considered; each run is secured directly to the back plane. The placement of wires carrying different voltages is taken into account, as each should be placed with respect to the National Electrical Code (NEC) [43]. The last pieces of equipment placed on the panel are secured to the door, as opposed to the back plane. When wiring is concluded, the panel's power flow and emergency stop function are ready to be tested and to have the program loaded into the PLC.

The Programming of the Control System

The electrical schematics detail which piece of equipment is connected to each I/O, so the PLC can address where to retrieve required inputs and where to send required outputs. The programming begins by entering the designated I/O into the program. Rockwell Automation's *Studio 5000* programming software was used in this project because of the utilization of an Allen-Bradley PLC and its accompanying I/O cards.

The main program incorporates several routines, or smaller programs, that act to break the main program into dedicated parts. For example, there is a routine that only deals with how the PLC handles digital float signals (i.e., digital signals from a device that turns on or off at the presence of water). Another example of a routine is one that was written for this project – it only details what conditions must be present for the alarms to occur.

According to Morriss [44], "IEC 1131-3 [a standard produced by the International Electrotechnical Commission (IEC)] classifies PLC programming languages as ladder logic, functional block [diagram], instruction list, structured text and sequential flow chart." Each routine for this project was written using one of the following PLC programming languages.

- *Ladder Logic*, which Morriss [44] describes as a "graphics based" program language that "look[s] like a drawing of a stepladder and a little like the relay logic circuit diagrams that industrial electricians use." Each rung of the ladder represents a logical expression that can be either true or false, which turns the associated output on or off [44].

- *Functional Block Diagrams (FBDs)* act as a preprogrammed instruction, or set of instructions, which is applied to one or more inputs to result in one or more outputs. Inputs and outputs are displayed as tags and the main instructions contained inside the block. Figure 4 is an example of a simple FBD – in this case, for controlling a Hand-Off-Auto switch (HOA), which is a set of push buttons that when pressed, will put the associated device into a manual mode (i.e., Hand), turn the device off, or put it into an automatic mode where the device is controlled by the PLC program.

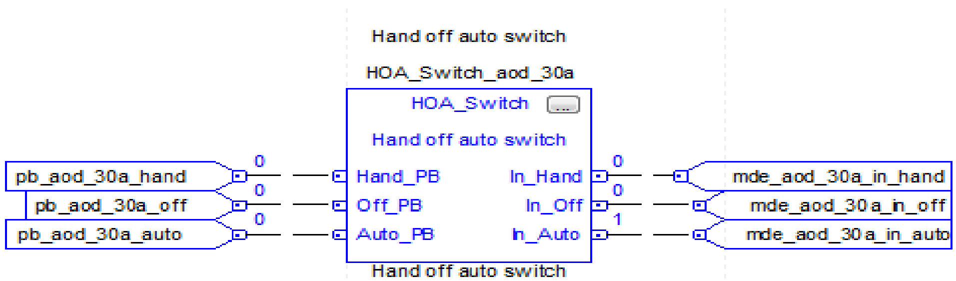

Figure 4: Example of an FBD Used for Controlling Hand-Off-Auto of an AOD Pump.

- *Structured Text* is a programming language that "…is based on [Pascal] [which] includes instructions for logical operations which work fine on data elements of type Boolean, and offer the other data types and data manipulation instructions" [44]. IEC 1131-3 dictates the set of instructions used for structured text. Figure 5 is the logic structured text associated with the FBD used to control Hand-Off-Auto switches.

```
if Hand_PB & (NOT Off_PB) then
    In_Hand := 1;
    In_Off := 0;
    In_Auto := 0;
end_if;

if Off_PB then
    In_Hand := 0;
    In_Off := 1;
    In_Auto := 0;
end_if;

if Auto_PB & (NOT Off_PB) then
    In_Hand := 0;
    In_Off := 0;
    In_Auto := 1;
end_if;
```

Figure 5: An Example of Structured Text Used for Controlling Hand-Off-Auto of an AOD Pump.

To program the control system, in this case the pH control system, it is necessary to begin with a diagram of the system. Figure 6 shows a basic system diagram.

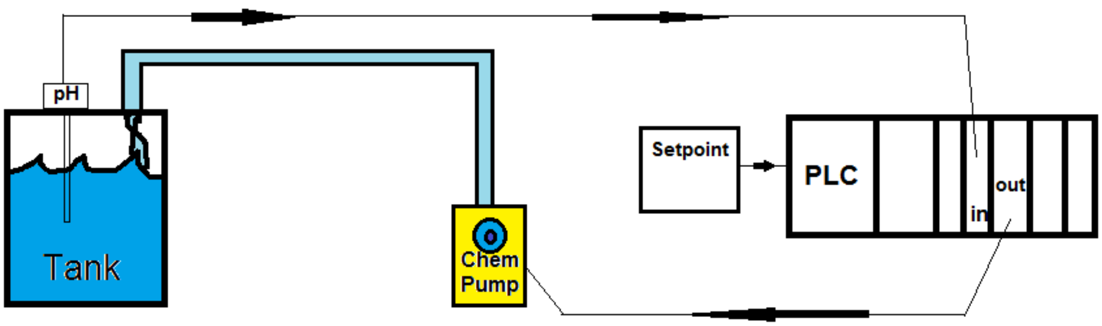

Figure 6: Basic Diagram of a pH System.

The system in Figure 6 is a closed loop system, where the PLC is getting information about the desired pH from a setpoint. In the PLC program, that setpoint is compared with the signal from the pH probe (the controlled variable). The comparison is analyzed by the program, which determines when to turn on the chemical pump (the control signal). As the chemical changes the pH in the tank, the probe sends that signal back (i.e., feedback) to the PLC to again be compared to the setpoint. This is just a very basic example of a pH control system, though its theory will be used in the programming section. The flow control system for maintaining the required flow through the clarifier will be described in the programming section.

The Start-Up Phase

When the design, building, and programming phases were complete in this project, the program was loaded into the PLC and the logic of the program was tested. The panel was then shipped to the facility where the electricians installed the control panel and wired all associated field equipment to the panel. The full functionality of the panel was tested in steps. First, the power was tested (i.e., Is power reaching each piece of equipment?). Then, as each part of the system (e.g., feed pumps, tank levels), were brought on line, the logic of the program was verified. Verification in this case entailed troubleshooting, fixing, or modifying the program to meet the demands of the directors, operators, and other personnel associated with the facility. Verification additionally ensured that the properties of the effluent water flow would comply with the standards required by the EPA and the POTW. The tuning of the proportional-integral-derivative (PID) feedback loop, used in the program to control the flow, occurred during start-up.

This project report also includes an explanation of the of human-machine interface (HMI) touch screen programming, as the use of the touchscreen is necessary in the project. See Appendix C for screenshots of a number of touchscreens. The functions on each screen are explained in detail in this report. Pictures and descriptions of each pushbutton and setpoint that are available to the operator are given in this project. The programming required to make such interfaces function, however, is not described, as there isn't a programming "language" associated with the software used.

Product Specifications

In order for the programming of a fully automated control system in this project to succeed, several aspects needed to be addressed. The aspects that require control by the PLC program include the influent flow rate, pH adjustment in the first and second reaction tanks, and the final pH adjustment tank, and the transfer pumps that move the wastewater throughout the system.

Influent Flow Rate

The influent flow rate into the system must be maintained to a limit below that of the maximum flow allowed by the clarifier -- in this case, 50 gallons per minute. This was accomplished by utilizing a tuned Proportional-Integral-Derivative (PID) loop that can control the speed of the influent pumps. By controlling the speed of the pumps, the flow is directly controlled, which is monitored using a flow sensor installed on the influent line.

pH Levels

The pH setpoints are selected based on the chemical requirements at each stage of the treatment. The pH is maintained at a low level in the first reaction tank, at a high level in the second reaction tank, and at a neutral level in the final pH adjustment tank. The pH levels of three tanks are monitored and maintained by using sensor inputs, which are compared to operator-selected setpoints and are adjusted based on the differences. The PLC program turns chemical feed pumps on for a setpoint-controlled amount of time, and then as a result of the information from the differences between the selected setpoints and the pH sensors in each tank.

Pump Controls Non-Analog

Pumps are used throughout the system to move water from one tank to another or out to the sewer. The on-and-off control for these pumps is based on the PLC program and digital (on/off) and analog inputs. Level elements measure the level of wastewater in a tank. Based on this input, an operator can select setpoint levels at which the pumps can operate. The program collects all the data from level switches (devices that will detect the presence of water at their position in the tank) and the operator's requirements to determine the state of each of the pumps in the system. It is crucial for the pump control be such that tanks do not overflow and wastewater is present at the right places at the appropriate time.

Each aspect of the control system is considered completed when each is tested and operating as programmed, and all levels are within compliance. That is, the flow is maintained at the operator selected setpoint, all pH levels are tracking as programmed, and pumps are moving water as required by the operator and system. This verification is typically determined during the commissioning phase.

Methods

This section features more detailed information about the design, build, and programming phases of the pH control project that is the focus of this report.

The Design Phase

River's Bend Engineering is an Engineering consulting firm, which means it receives purchase orders (POs) from clients to design wastewater treatment or pretreatment systems. If required by the client, the firm can source tanks, equipment, and subcontractors – mechanical and electrical – and provide services during construction. For example, the company can provide a construction manager on site throughout the build. The job of the electrical engineers at the firm is to design the control panel that houses the programmable controller, I/O cards, power supplies, and the other electrical components that enable the system to be fully automated. To begin any project, the Civil Engineers provide the electrical engineers with a P&ID. The engineering problem to be solved by the electrical engineers is how to make the control system autonomous.

To begin the design in this project, the PLC and I/O cards had to be selected. As each I/O is defined, they must appear in an electrical schematic, which details how the control panel will be wired. The result of the design phase is the electrical schematic of the control panel, shown in Appendix B of this paper. This schematic is what was used to proceed on to the build phase of the project.

During the design phase, the equipment detailed in the P&ID was analyzed to determine what PLC would be used in the project. The number of inputs, outputs, and whether the PLC is Ethernet communications capable are some of the considerations taken into account in the selection of a PLC.

The controller of the control panel cannot work on its own. The controller must be powered and it must be connectable to the field equipment (any equipment controlled that is not inside the panel). The controller requires other devices to provide physical -- as opposed to programmed -- safeties, and devices that help get the information to and from the controller. The panel itself required large voltages for powering motors on pumps and mixers in the field, and those devices required their own safeties, which are required by codes and standards. In a typical project, the electrical engineer accomplishes this task for everything within the panel, but the installing electrician provides any required safeties in the field. In this project, an electrical engineer and electrician addressed these challenges.

The large voltage associated with this project needed to be transformed into a smaller voltage for powering other electrical devices in the panel. And even that smaller voltage needed to be converted into yet another smaller voltage for different devices. The devices which transform the voltages at each step have limits to the power they can handle. Therefore, the analysis of the P&ID was done to determine how much power each device requires. How much power the components that are drawing power from those smaller voltage devices additionally needed to be taken into account. And lastly, relevant codes and standards required that safety measures be put in place, as powering certain field devices from a control panel can be dangerous. Not being right next to the control panel where the power can be easily shut off in case of an emergency made it essential for there to be power disconnects within line of site and no more than three feet away from the motor. The codes require that the safety of people and the safety of the devices and the voltage supplier line into the panel also be taken into account as well.

Therefore, safeties were put in place to keep the voltage supply line safe from any shorts that may, inadvertently, occur within the panel.

For the build phase, the equipment must be selected, sourced, and wired together to create the physical control panel itself, using the electrical schematics produced in the design phase. Figures 7, 8, and 9 are pages from the project P&ID given to the author of this report by the Civil Engineers. A set of these drawings appears in Appendix A.

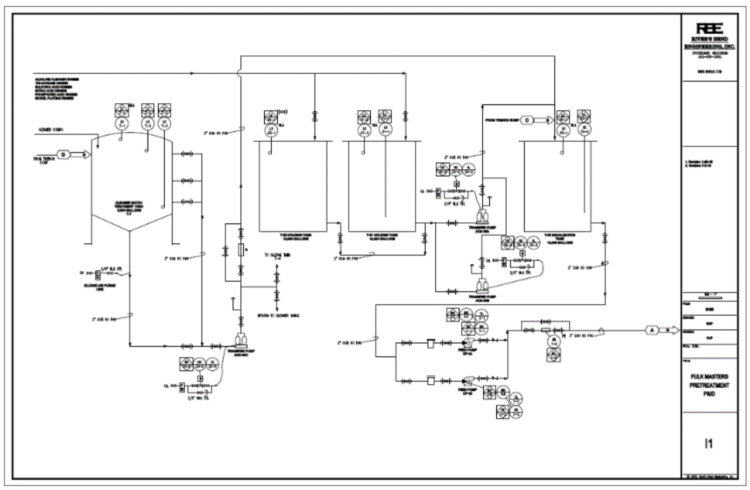

Figure 7: Page I1 of the Project P&ID.

Figure 7 shows the front end of the system. It includes the tanks that hold the different wastewaters from the plating process. These waste streams are brought together into the equalization tank via the transfer pumps. And the feed pumps move the water from the equalization tank to the treatment part of the system, shown in Figure 8. There are switches and level elements in each tank. These are shown with symbols that the engineers use to tell the people who will refer to this drawing what each piece of equipment is. The symbols have boxes and circles, which allow technicians to know how the equipment is to be used and where it should be placed.   Figure 8 also shows that there are pumps that will move the water through the system. The engineers have specified the use of two different types of pumps in Figure 8, including Air Operated

Diaphragm (AOD) pumps and Centrifugal pumps. These pumps are indicated by the symbols used for each – AOD and CP, respectively. Figure 8 actually contains a significant amount of information, but not all necessary information is included, such as the horsepower of the centrifugal pumps. This information must be obtained from the project engineers before a project design phase is undertaken.

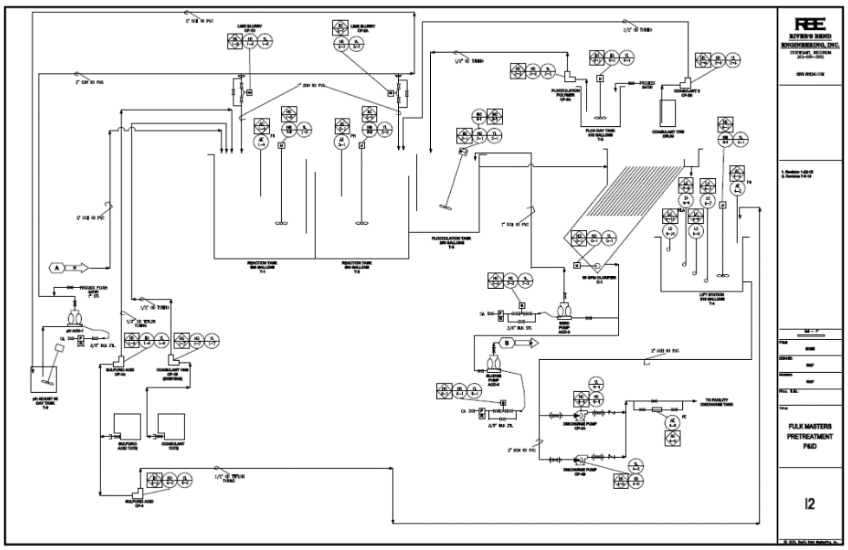

Figure 8: Page I2 of the Project P&ID.

Figure 8 shows the main part of the system. In Figure 8, mixers and chemical feed pumps are introduced, as well as a plate clarifier and its auger. Figure 8 also features the pH sensors (indicated by PH) in the system. For example, these sensors are shown in the reaction tanks and the lift station. In Figure 8, contaminated water is brought in from the equalization tank via the feed pumps at point a seen in the left, center and goes into the reaction tanks, where all the chemical adjustments take place. The water travels by gravity through each of the reaction tanks and the flocculent tank before it is filtered through the inclined plate clarifier and then sent to the final pH adjustment tank. From the plate clarifier, an AOD seed pump and an AOD sludge pump can be seen. The seed pump puts some water from the clarifier back into the flocculation tank to aid in the production of flocculent. The sludge pump sends the filtered-out solids from the bottom of the clarifier and moves them to sludge-holding tanks, shown in Figure 9. Filtered water from the top of the clarifier is sent on to the final pH adjustment tank. From there, two centrifugal pumps take turns sending the metal-free water out to the sewers. Figure 8 additionally shows which chemicals will be sent to which tanks during the process. Figure 8 does not contain information on how much horsepower is required for the mixers, pumps, and auger. Again, this information needed to be obtained from the project engineers before the project design phase could commence.

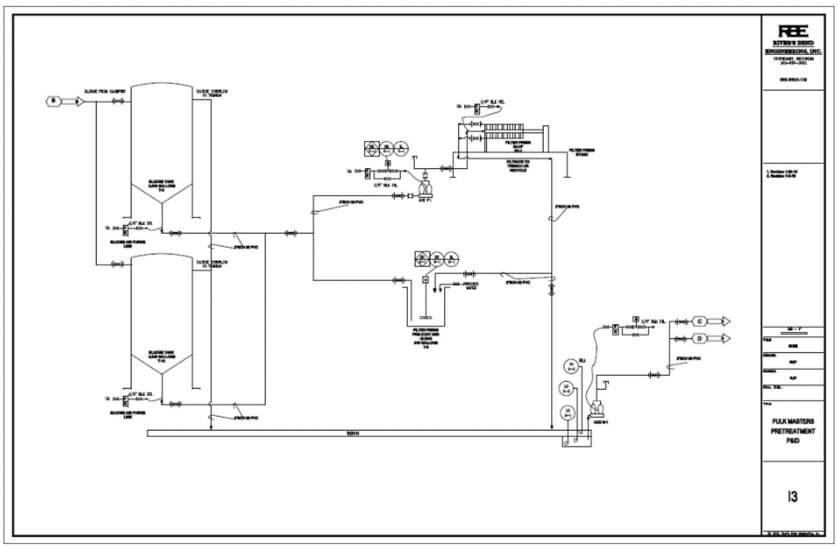

Figure 9: Page I3 of the Project P&ID.

Figure 9 shows the sludge-holding tanks, which receive the water via the sludge pump shown in Figure 8. The sludge is a combination of solids and liquids and needs to be pressed, or dewatered, to finish the process of removing the metals from the water. The filter press AOD pump moves the water from the holding tanks to the filter press. The filter press removes the solids from the water by pressure and filtering. What is left is a dry “cake” of the metals and other solids. The water is fed back through the system via the sump, or trench, which holds the water, and the sump pump, which moves the water to the equalization tank at the front of the system. The cake is taken off site and sequestered at a regulated dump facility.

In this project, the civil engineers described the functional requirements of the field equipment and the process of the system. They defined which levels in the tanks should activate the pumps, at which pH levels the reaction tanks should be maintained, as well as the speed control on the feed pumps, so that the flow of the system can be accommodated by the capability of the clarifier. They further stipulated how the switches associated with the tank levels would control the transfer and discharge pumps.

This project information, as well as the information in the P&IDs, was employed to generate a set of electrical drawings. These drawings were the responsibility of the project electrical engineers and were used by the panel builder and the electricians to install the system. The electrical drawings can be seen in Appendix B.

The Building Phase

To build the control panel associated with this project, the electrical engineers used the electrical drawing set to purchase the required electronics, the enclosure, and the touchscreen that are employed to make the system function. Once the required parts arrived, the building phase began. The layout of the electronics was planned with ease of use for the wiring and for the electricians. Space requirements around the variable frequency drives (VFDs) needed to be maintained and high voltage traces were kept separate from the low voltage ones. Figure 10 shows the right-hand side of the completed panel as it stands in the facility.

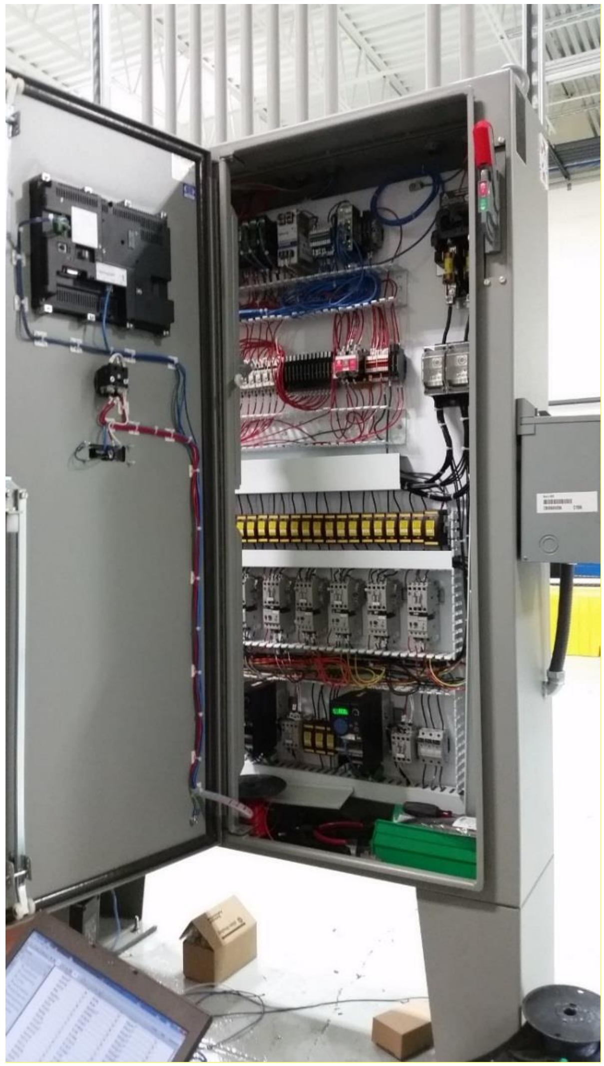

Figure 10: Control Panel, Right-Hand Side -- Installed.

Figure 11 is the left-hand side of the completed panel.

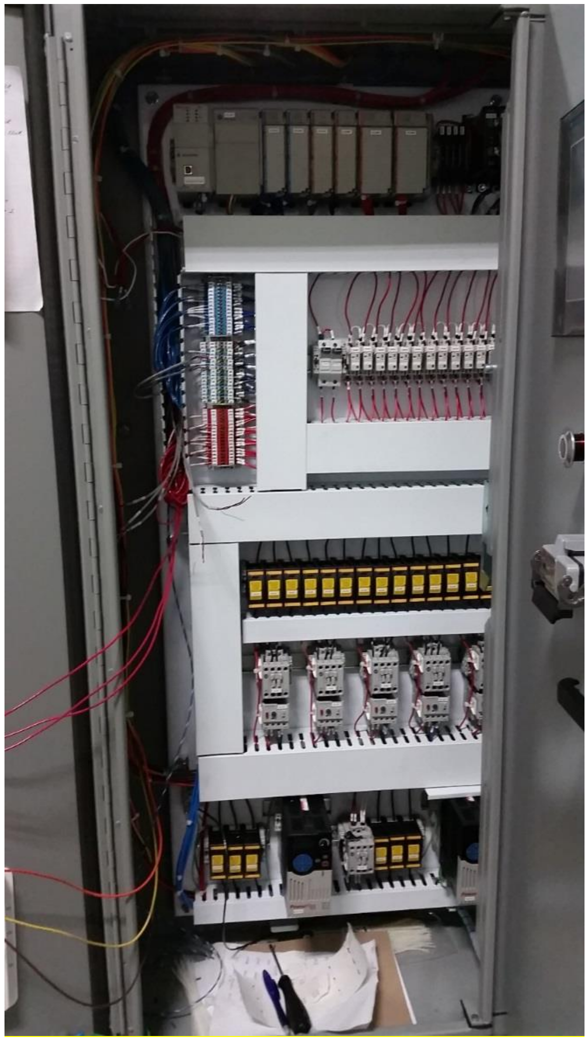

Figure 11: Control Panel, Left-Hand Side, During Construction.

The wire colors identify the voltage levels passing through them. Black is used for 480 volts of alternating current (VAC), red is used for 120 VAC, and blue is used for 24 VDC. There are also gray cables that contain a red wire and a black wire and a shield wire. These are used for sending and receiving analog signals from the sensors in the field. The sizes of the wires are dependent on the power requirements and current-carrying capacity requirements of the equipment. The rest of the building phase is not a focus of this paper and will not be described further.

The Programming Phase

The Allen Bradley PLC was specified by counting inputs and outputs (I/Os), and by determining how many functions it was capable of performing, as well as by its communications capabilities. The PLC chosen for this project was the AB 1769-L30ER.

Whether a signal is an input or an output is always in reference to the PLC. If the PLC is receiving information from a field device, it is an input. If the field device is receiving information from the PLC, it is considered an output. Inputs and outputs can be either digital or analog signals. All devices that communicate with the PLC via Ethernet count as nodes, and travel through an Ethernet switch. They are considered neither inputs nor outputs.

Figure 12 shows the batch tank T- 7 and the transfer pump AOD-30C that pumps water from the batch tank to T-30.

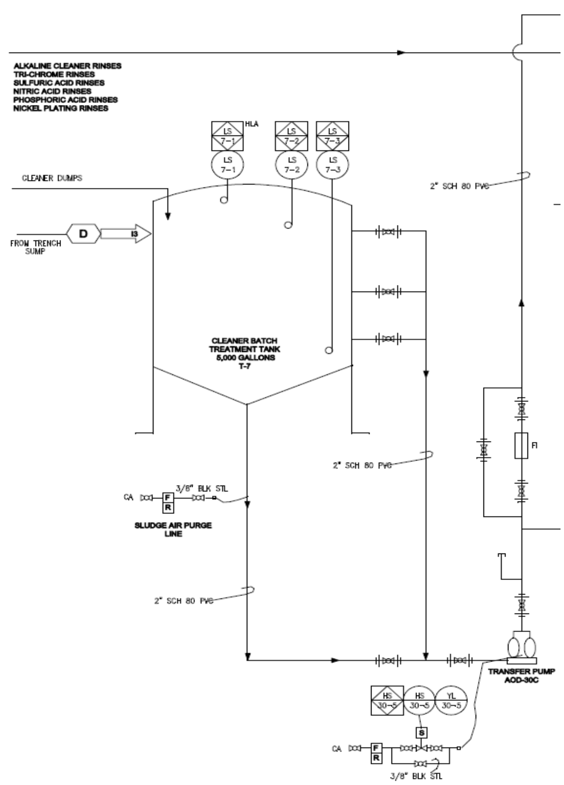

Figure 12: Cleaner Batch Tank T-7 and Transfer Pump AOD-30C P&ID.

Level switches (LSs) in tanks are digital devices. They are either on or off. They are indicated on the P&ID by a vertical line with a circle at the bottom. Each LS is assigned a number that references the tank it is associated with and the device type that it is. Additional information can be found in the boxes above the name or annotated to the side. Figure 12 indicates that LS 7-1 needs a High Level Alarm (HLA) associated with it. This configuration informs the technician that this LS is a normally closed (NC) switch that opens the contact when the water reaches that high. HLA LSs are always NC; all the others are normally open (NO). A normally open level switch closes the contact when water is present.

The type of I/O a level switch should be counted as is decided by whether the information from the device is needed by the PLC to react, or whether the PLC needs to send commands to the device to perform its task, or both. Continuing with the example of LS7-1, the level switch needs to tell the PLC that the water has reached the level of the switch. The programming in the PLC then decides what to do about the situation. Either way, this level switch is treated as one (1) digital input. For this project, it was decided that all digital inputs would be powered by 24 volts of dedicated current (VDC), as the smallest voltage needed to do a job should be used. In addition, it is important to point out that all of the digital inputs in the project will be sinking rather than sourcing, as the load is being provided a connection to ground through the PLC I/O card. This design adds a measure of safety to the system, as it will stop the possibility of alarming on false positives if there is ever a short to ground.

LS 7-2 is a NO level switch. Like LS 7-1, it is also one (1) digital input. Unlike LS 7-2, a normally closed (NC) switch, LS 7-1 is a normally open (NO) switch. This difference is emphasized in the electrical system schematic by means of different symbols. LS 7-3 is also a NO level switch and similarly counts as one (1) digital input. These three switches constitute all of the level switches for batch tank T-7.

The transfer pump AOD-30C in Figure 12, as an air operated diaphragm pump, requires an air supply to function. The turning on and off of this supply is accomplished using a solenoid valve. There are only two states for the valve to be in, open or closed. When the valve receives power, it will open. When the power goes away, the valve will close. This is an example of a device requiring a digital output from the PLC. The PLC doesn’t receive any information from the solenoid; the PLC only needs to tell the valve whether to be open or closed. Thus, all AODs count as one (1) digital output.

Solenoids require power to be sent in order to function. All solenoids used in this project require 120 VAC. That voltage is supplied by the control panel. Therefore, all digital outputs must be 120 VAC. Relay output cards are employed because to send AC voltage to the devices.

Figure 13 is the P&ID of the holding tanks T-32 and T-31 and the transfer pumps that pump water from the holding tank to T-30.

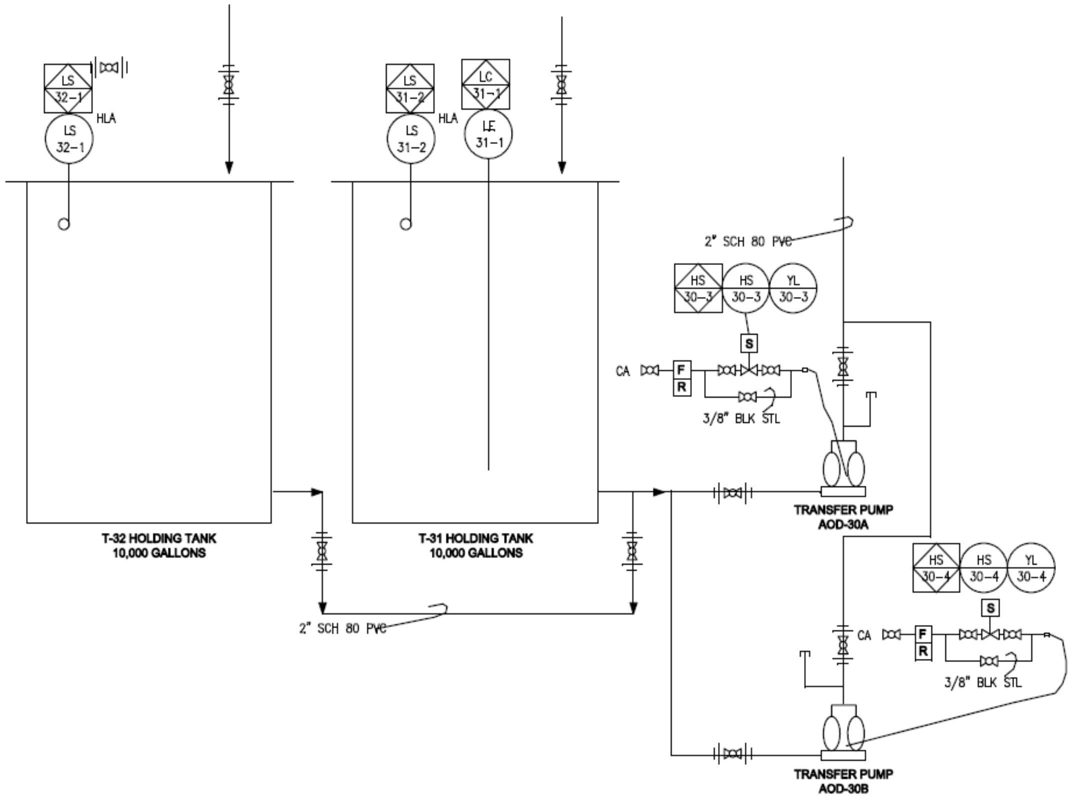

Figure 13: Holding Tanks T-31 and T-32 and the Transfer Pumps AOD-30A and AOD-30B P&ID.

Holding tank T-32 only has one level switch associated with it (LS 32-1). It is designated as a HLA, and it counts as one (1) digital input. Holding tank T-31 also has a level switch (LS 31-2). It additionally is designated as an HLA, and it counts as one (1) digital input.

All sensor elements are symbolized as a vertical line. The nomenclature for each depends on which type of sensor it is. Sensors send an analog signal to the PLC, which enables the program scaling to provide a level, a pH value, a flow rate, or any real number type of data. Tank T-31 has a level element (LE) that measures the differential pressure in the tank (LE 31-1). The signal is sent to the PLC as a 4-20 mA signal. This signal is scaled in order to indicate the level of the water in the tank. Therefore, LE is counted as one (1) analog input. These devices are loop powered, which means they are configured as a two-wire hook-up. These devices generally require 24 VDC to power them. The lower DC voltage is good to use when such small signals are being transmitted alongside the powered wires.

Each of the transfer pumps, AOD-30A and AOD-30B, require one (1) digital output. These outputs also require 120 VAC.

Figure 14 is the P&ID of the equalization tank T-30 and the feed centrifugal pumps (CP), CP-1A and CP-1B. This figure also includes the analytic element (AE), which measures the flow, AE-1-3.

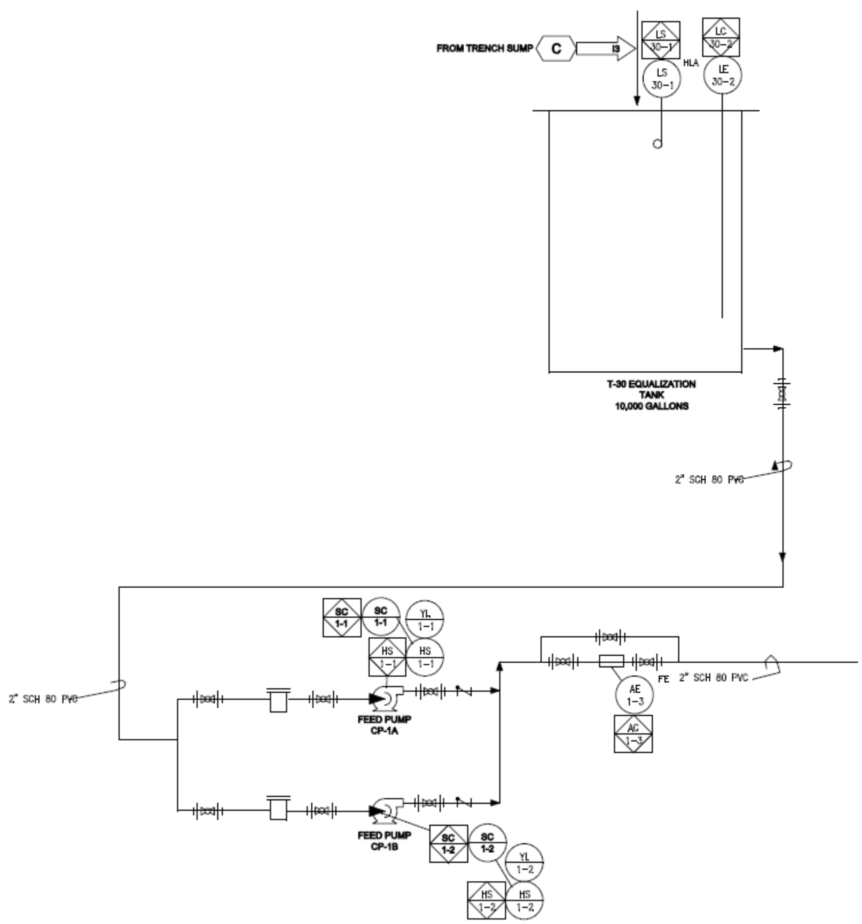

Figure 14: Equalization Tank T-30 and Feed Pumps CP-1A and CP-1B P&ID.

The equalization tank T-30 contains one (1) digital input (LS 30-1) and one (1) analog input, (LE 30-2). The HLA associated with this tank indicates that it is a normally closed (NC) level switch. The analytic element is the flow element, AE 1-3. This device measures the main flow in number of gallons per minute of the water into the system, which needs to be controlled with the feed pumps. AE 1-3 counts as one (1) analog input, as the PLC doesn't need to send controls to it.

The centrifugal pumps are designated with the SC symbols on the P&ID, which indicate speed control. Because it was decided to use VFDs in the system, as well as Ethernet communications between the VFDs and the PLC, the VFDs require no inputs or outputs into I/O cards. Instead, an Ethernet configuration is required for the VFDs so that they can communicate with the program.

Figure 15 shows the reaction tanks T-1 and T-2 and the flocculation tank T-3. There are several devices associated these tanks, because it is in these tanks that the chemistry takes place to get the metals into a form that can then be removed from the water.

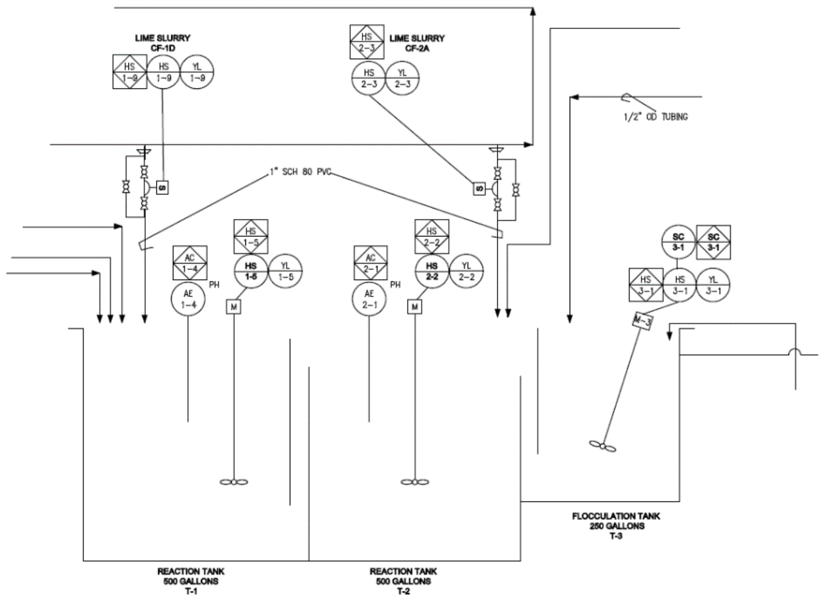

Figure 15: Reaction Tanks T-1 and T-2, Flocculation Tank T-3 and Associated Equipment.

The solenoid pinch valves (CF-1D and CF-2A) are labeled as chemical feed pumps, because in the project, a revision occurred that changed the control of the caustic chemical entering T-1 and T-2. Accordingly, they are treated as the previous solenoid valves: Both CF-1D and CF-2D are counted as one digital (1) output each.

There are analytic elements in each of the reaction tanks, and their notation indicates they are measuring pH in their respective tanks. Each pH sensor is treated as one (1) analog input.

The symbol for a mixer is a vertical line with a propeller at the bottom and a box with M in it at the top. Each tank in Figure 15 has a mixer, but some are treated differently than others. The mixers in T-1 and T-2 have no speed control specified and therefore will need no VFD. However, this does mean that they will be controlled directly from the PLC. The mixer’s motor is what needs to be turned on and off. As it doesn’t have a VFD to provide alerts about any faults or overloads, it is necessary to provide another method to enable operators to become aware of problems. This task is accomplished by an overload circuit protector with the motor contactor inside the control panel. The overload protector is capable of providing a digital signal. These overload protectors count as one (1) digital input each for M-1 and M-2 in Figure 15.

These mixers also need to be turned on and off. This will happen through the motor contactor, which acts as a special relay. The relay output sends a 120 VAC to the coil of the contactor, which then passes the three phase 480 VAC to the mixer's motor. The mixers, therefore, also require one (1) digital output each.

Mixer M-3 in the flocculation tank requires speed control. This requirement, again, means that it is necessary to use a VFD to control the mixer. Therefore, mixer M-3 requires neither an input to, nor an output from, the PLC. All commands will be received and sent via Ethernet.

Figure 16 contains some of the chemical day tanks and the associated chemical feed pumps.

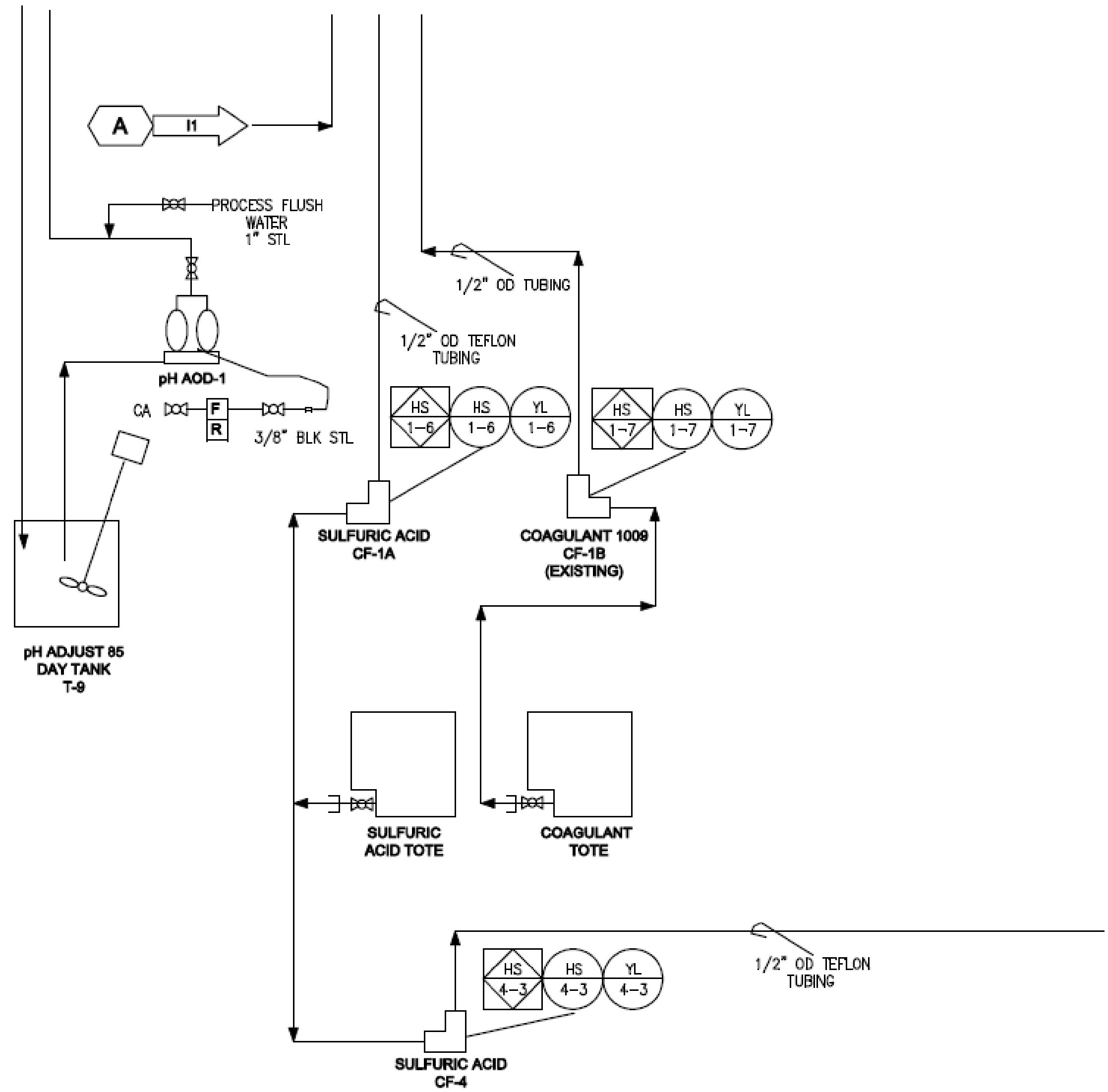

Figure 16: Chemical Totes and Day Tanks and Associated Chemical Feed Pumps.

PH Adjust 85 is a lime slurry used in the system as the caustic that raises the pH in the tank to which it is added. A day tank is the name given to a tank that holds the chemicals. Some chemicals are kept in the totes they are transported in, as that is the safest way to use the chemicals. The totes reduce the handling time by limiting when operators can be in direct contact with the chemicals. As Figure 16 shows, there are a sulfuric acid tote and a coagulant tote.

The chemical feed pumps are named for the tank they pump to, except in the case of AOD-1. This pump only pumps the lime slurry as far as the two pinch valves described in Figure 8, CF-1D and CF-2A. Since AOD-1 is an AOD and uses a solenoid valve to open the air required to make it function, it is counted as one (1) digital output. The other chemical feed pumps -- CF-1A, CF-B and CF-4A -- each count as digital outputs as well, because they will only need to be turned on or off. Engineers could choose more expensive chemical feed pumps that can be speed-controlled through the PLC, and therefore, would count as analog output. To keep cost down, the engineers in the project decided to use chemical feed pumps that have manual adjustments for stroke length and speed. The pumps are adjusted during the start-up phase and remain set when the system is running.

After all digital and analog inputs and outputs are accounted for, they must be programmed into the PLC and given tag names. A tag name describes the signal as it is used in the program. This is the part of programming referred to as the infrastructure. It is the basis for the rest of the programming to come. Digital and analog inputs and output and analog scaling are all parts of the infrastructure.

Defining inputs is done in the input routine using functional block diagrams. Figure 17 is the input routine of this program.

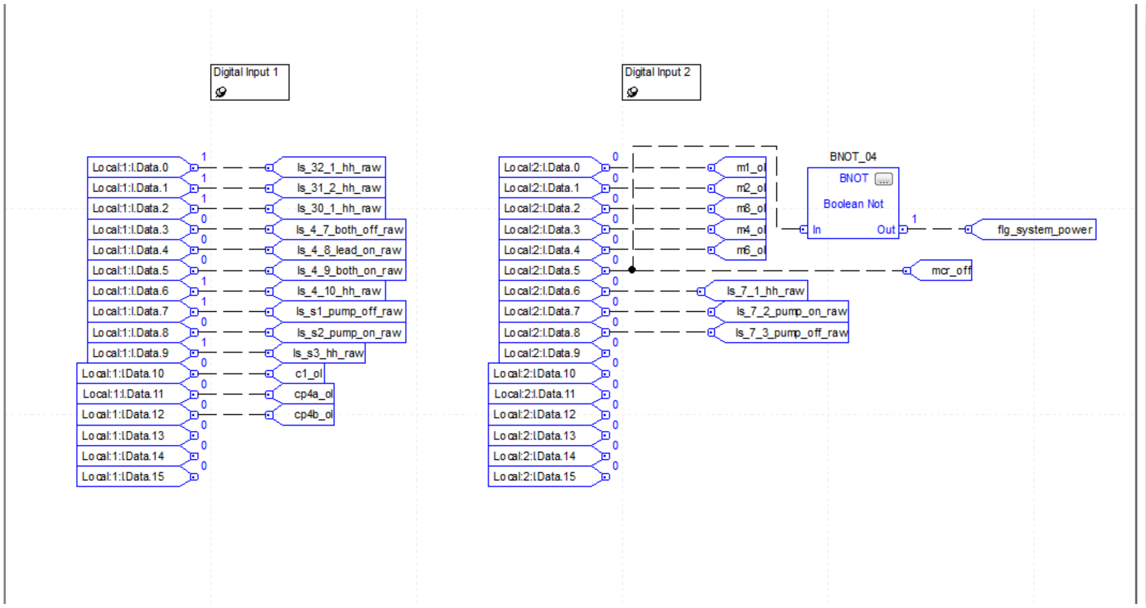

Figure 17: Digital Input FBD.

Digital inputs are hooked into the I/O cards, which are given a location as to where the data can be found. For example, *Local1:Data 0* means that these data are coming from the first input card and are connected to data port 0. To use the data from the given inputs in the program, the locations are given tags. Each tag carries with it a name, and sometimes a description, a data type, and scoping. The names given the tags are taken from the electrical schematics. For example, *ls_32_t_hh_raw* refers to the raw signal from the high-level level switch in tank 32. The data types are based on how the signal enters the PLC I/O. For these inputs, they are digital signals, and therefore, the data types are Boolean; in Figure 17, they are displayed as a dotted line. That means Boolean logic will be employed in the program wherever these tags are needed. The scoping of a tag is dependent on where and how the PLC program calls the tag. If the tag will only be used within a subroutine, it is locally scoped, which aids with data integrity and troubleshooting. If the tag is globally scoped, the tag is visible from anywhere in the program and can be read directly by outside devices, like the HMI or other data recording devices. These inputs are globally scoped. Because the inputs are Boolean, they are data requiring only one-bit to pass information. To save memory space in the PLC, an array -- which in this case is a 32-bit word called *arr_input1* – is used, to which all input tags are aliased. This means that instead of using a whole 32-bit word for one 1-bit piece of information, thirty-two 1-bit pieces of information can be stored, maximizing memory usage. Maximizing processor times and saving memory space are more important with larger programs with many more instructions, but it is nevertheless good programming practice.

Figure 18 shows the digital output routine of the program.

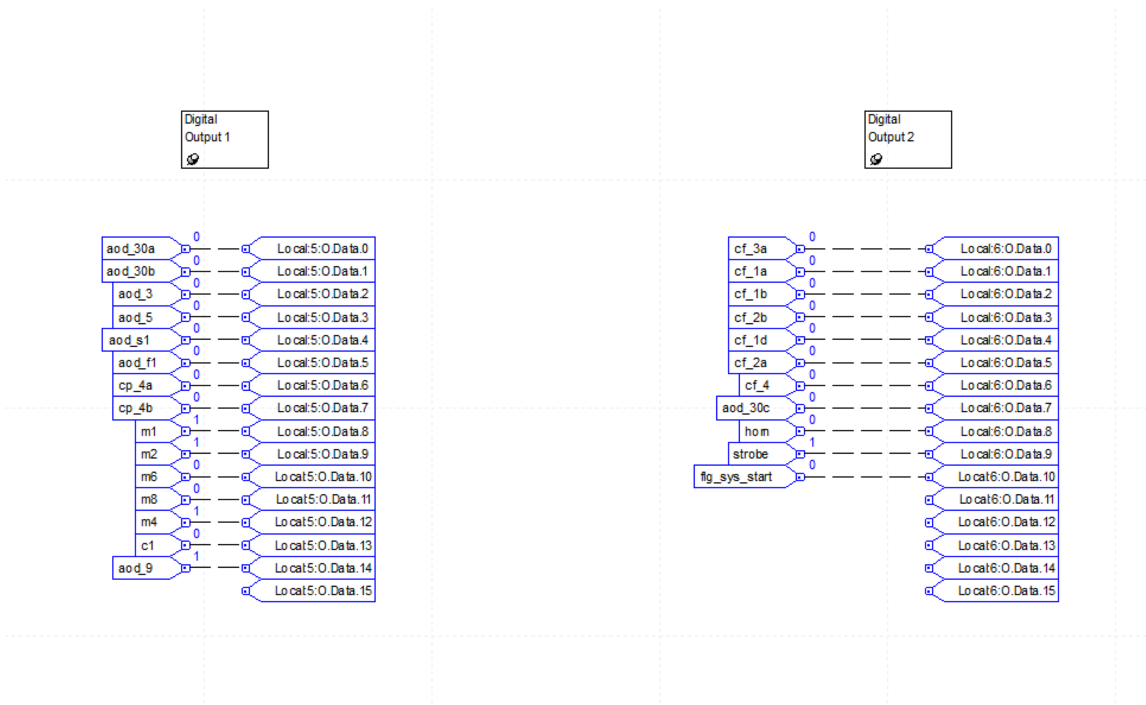

Figure 18: Output Routine FBD.

The output routine functions in nearly the opposite manner as the input routine. As the purpose of the input routine is to take information from the outside world and give it a name to be referred to in the program, the output routine similarly takes a name used in the program and sends it to the outside world. Just like the inputs, the tag names are aliased to an array to save memory, as the data types are Boolean.

Figure 19 is the Analog Input routine used in the program.

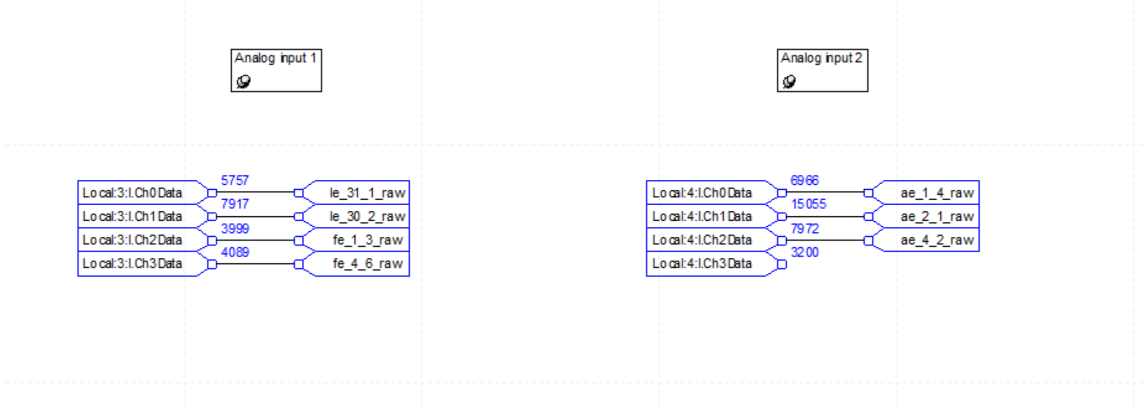

Figure 19: Analog Input Routine FBD.

The analog data types can vary and are dependent on what type of information is being conveyed. For analytical sensor elements passing information about pH, the data type used is real, as real numbers include decimals. Real numbers will allow a maximum value of +/- 3.402823 x 10^38 to a minimum value of +/- 1.1754944x10^-38. Real numbers do take up more memory, but are used nonetheless to maintain a high fidelity with the signal.

Information being sent to the PLC from level elements (such as level switches) use 32-bit, signed integers, as the information being passed is a whole number, requiring no decimals. Thirty-two-bit signed integers also allow for a maximum value of 2147483647 to be passed, which is good when working with large-volume tanks.

Another input pre-process is the scaling of the analog input. This task is necessary because the operators need to know level, pH, and flow in the units they are meant to be read in and the analog devices are reading the current through the device. The scaling block instruction is part of the Allen-Bradley software and does not need to have an add-on instruction written for it. Within the instruction, the maximum and minimum input signals are entered, as well as the maximum and minimum engineering units (EUs) (the units the operators are familiar with). The input units used are 4000 to 20000, representing the 4-20 mA current limits. The EUs are dependent on what the signal is measuring and perhaps the sensor's configuration. For pH, the EUs are 2-14. For the level, the EUs are the limits of the sensor based on the length of the cord submerged in the tank, minus the distance from the bottom. The net length of the sensor and the radius of the tank are used to calculate the number of gallons present in a disc described by the dimensions of the tank with a thickness of one inch. Once the gallons per one inch are calculated, the maximum and minimum EUs can be entered. For flow, the EUs again are based on the sensor configuration. The sensor can read flows in the range of 0-150 GPMs, so these are the EUs entered.

Allen-Bradley scaling uses Equation (1) and the parameter limits to determine the output of the scaling block:

$$Out = (In - InRawMin)x\left(\frac{InEUMax - InEUMin}{InRawMax - InRawMin}\right) + InEUMin. \tag{1}$$

Figure 20 is the raw signal entering into the scaling function block. The output is then passed through an adder that adds it to an offset value. The offset is used only for calibration and signal corrections. If the correction is not needed, the output of the scaling block is added to 0 and thus remains the same.

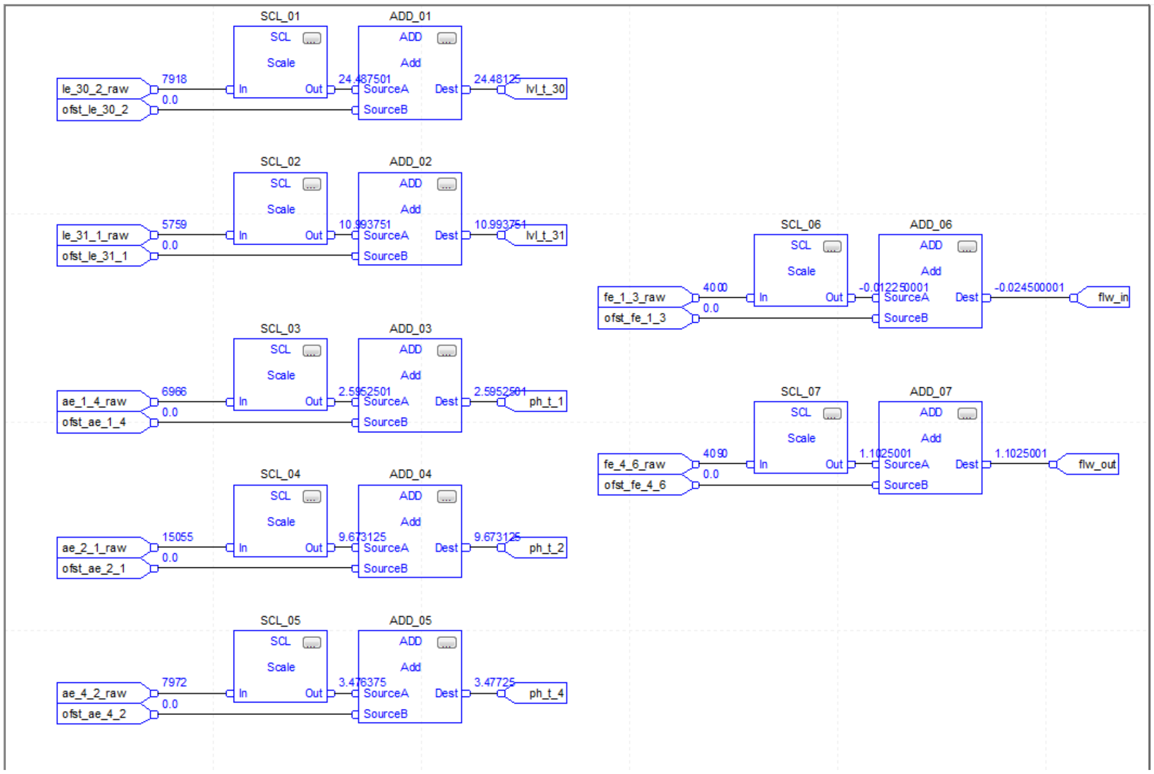

Figure 20: Analog Scaling Routine FBD.

The information coming from flow elements are defined as real numbers, like those from the pH sensors, because of the need to pass a decimal number.

All inputs -- digital or analog -- with a tag including the word “raw” are so named because their signal needs to be preprocessed before it will be used in the program. Figure 21 and 22 show the float routines.

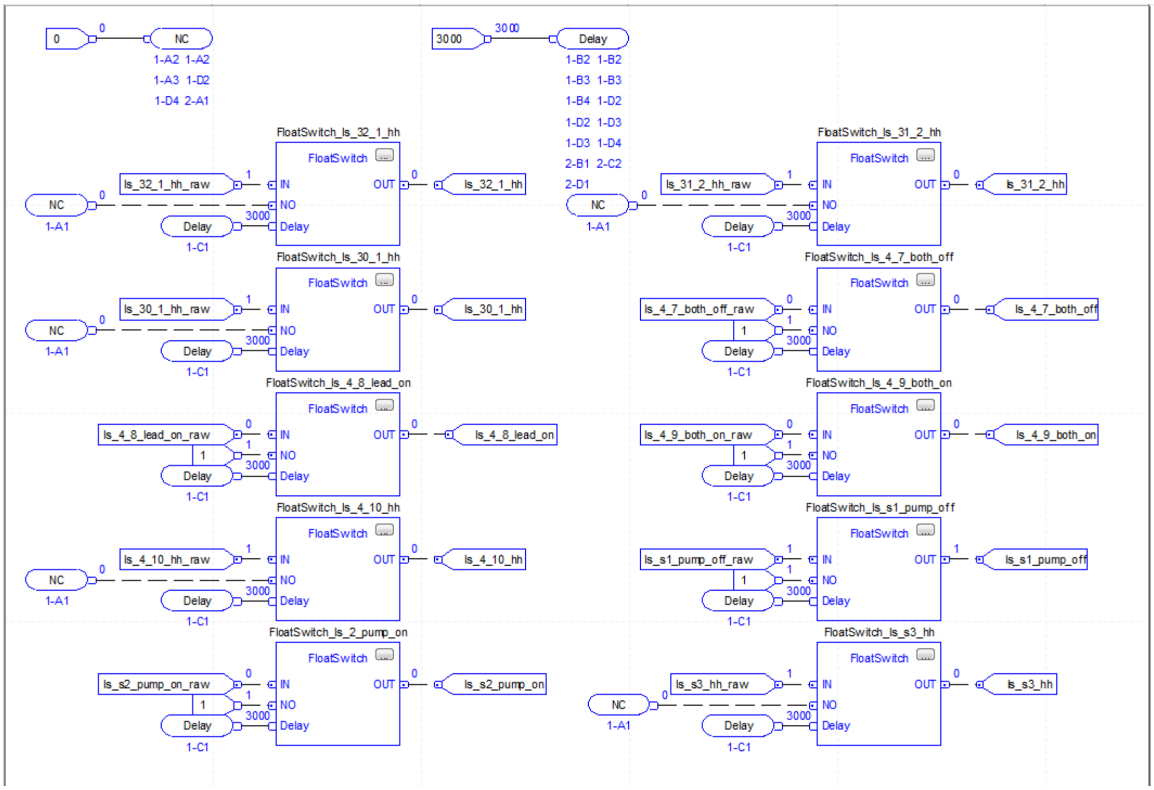

Figure 21: Float Routine 1 FBD.

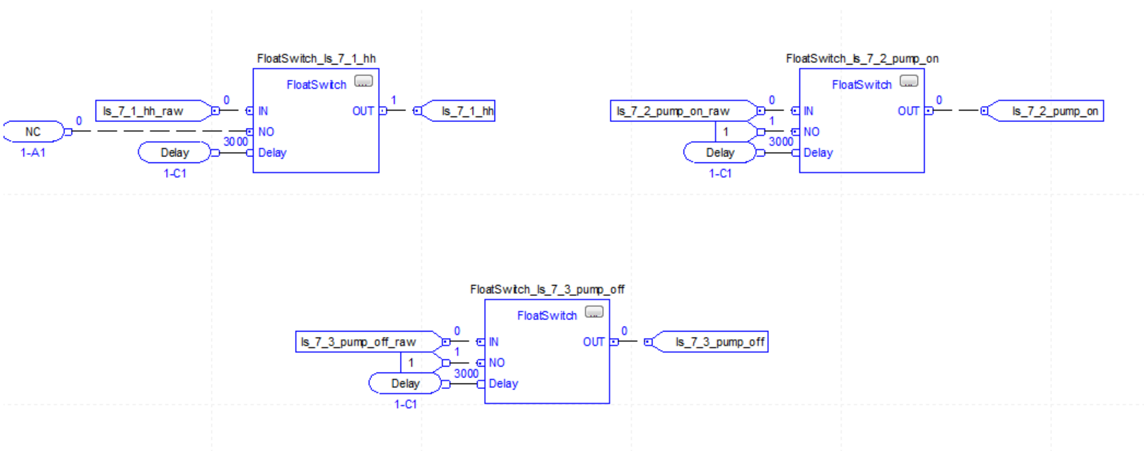

Figure 22: Float Routine 2 FBD.

Each float is either normally open (NO) or normally closed (NC). When the water reaches a normally open float, the float switch closes and passes a high digital signal to the PLC I/O. When the water reaches a normally closed float, the switch is opened and the once high signal goes low. This scenario can lead to some confusion for the programmer. Because both cases mean that water is present, but each results in a different signal level, an add-on instruction is used to make the output of the instruction high only when water is present. Therefore, the FBDs in these routines employ the FloatSwitch add-on instruction. FloatSwitch is an instruction that was written to be used over and over again. Figure 23 is the logic used in the add-on instruction.

```
abt_delay.PRE := Delay;
if NO then
    if IN then
        abt_delay.TimerEnable := 1;
    else
        abt_delay.TimerEnable := 0;
    end_if;
else
    if NOT IN then
        abt_delay.TimerEnable := 1;
    else
        abt_delay.TimerEnable := 0;
    end_if;
end_if;
TONR(abt_delay);
OUT := abt_delay.DN;
```

Figure 23: FloatSwitch Add-On Instruction Logic.

This logic describes how the signal from the raw digital input from a float switch will be manipulated to output a digital signal that is high when water is present at that particular float switch. Each parameter used in the instruction is called by linking it to tags used in the program. The parameters in this instruction are Delay, NO, IN, and OUT. Each parameter is linked to tags from the main program in the FBD.

The abt_delay is an anti-bounce timer. This is a timer employed to prevent quick successions of on-off behavior in the switch. Abt_delay is used internally and is not linked to the rest of the program. The timer abt_delay.PRE is defined first. It calls the parameter Delay from the routine, which has been defined as 3000 ms at the top of float routine 1. Delay is the amount of time the timer is set for. The instruction then calls for the NO parameter. If this value is a 1, referring to a NO float, as set by a tag linked to the NO parameter, then the logic continues to look at the input parameter, IN, the raw data from the float. If it is a normally open float and water is present, the input goes high, and the anti-bounce timer starts timing. Then, when it is done, the output signal goes high. The output parameter is linked to the tag in the float routine and is used in the program. The logic also describes what happens when the NO parameter is 0, declaring a normally closed float, in the else section of the logic. The instruction looks for a low signal to start the timer timing (NOT IN). And again, the output goes high to show water is present at the normally closed float. Each float of the system goes through this process.

Alarm add-on instructions function the same as the other add-ons. The logic is more complicated because of the many styles of alarms that are present in such a system. This add-on needs to account for setpoints, high and low alarms, anti-bounce timers, alarm resets, and the function of the alarm horn. Figures 24 and 25 show the add-on instructions logic used for level alarms.

```
/* do input parameters exist or are we using defaults? */
if delay = 0 then
    abt_delay := 3000;
else
    abt_delay := delay*1000;
end_if;

if PctRst = 0 then
    Percent_reset := .05;
else
    Percent_reset := PctRst/100;
end_if;

/* Set up parameters */
reset_high := AlarmLevel-(AlarmLevel*Percent_reset);
reset_low := AlarmLevel+(AlarmLevel*Percent_reset);
abt_low_level_off.PRE := abt_delay;
abt_low_level_on.PRE := abt_delay;
abt_high_level_off.PRE := abt_delay;
abt_high_level_on.PRE := abt_delay;

/* If this is a low level alarm */
if NOT HiLo then
    if CurLevel <= AlarmLevel then
        abt_low_level_on.TimerEnable := 1;
    else
        abt_low_level_on.TimerEnable := 0;
    end_if;
    if abt_low_level_on.DN then
        InAlarm := 1;
    end_if;
    if InAlarm
        AND (CurLevel >= reset_low)
    then
        abt_low_level_off.TimerEnable := 1;
    else
        abt_low_level_off.TimerEnable := 0;
    end_if;
    if abt_low_level_off.DN then
        InAlarm := 0;
    end_if;
end_if;
```

Figure 24: Level Alarm Add-On Instruction Logic.

```
/* If this is a high level alarm */
if HiLo then
    if CurLevel >= AlarmLevel then
        abt_high_level_on.TimerEnable := 1;
    else
        abt_high_level_on.TimerEnable := 0;
    end_if;
    if abt_high_level_on.DN then
        InAlarm := 1;
    end_if;
    if InAlarm
        AND (CurLevel <= reset_high)
    then
        abt_high_level_off.TimerEnable := 1;
    else
        abt_high_level_off.TimerEnable := 0;
    end_if;
    if abt_high_level_off.DN then
        InAlarm := 0;
    end_if;
end_if;

TONR(abt_low_level_on);
TONR(abt_low_level_off);
TONR(abt_high_level_on);
TONR(abt_high_level_off);
```

Figure 25: Level Alarm Add-on Instruction Logic, Continued.

Figures 26 and 27 shows the part of the logic that activates the alarm horn and strobe.

```
/* Check if an alarm or warning exists in the system */
if (alarm_word_0 <> 0) OR (alarm_word_1 <> 0) OR (alarm_word_2 <> 0) OR (alarm_word_3 <> 0) OR (alarm_word_4 <> 0) OR (alarm_word_5 <> 0) then
    alarm_request := 1;
else
    alarm_request := 0;
end_if;

if (warn_word_0 <> 0) OR (warn_word_1 <> 0) OR (warn_word_2 <> 0) OR (warn_word_3 <> 0) OR (warn_word_4 <> 0) OR (warn_word_5 <> 0) then
    warn_request := 1;
else
    warn_request := 0;
end_if;

/* Determine if horn has to be reacivated due to new alarm condition */
// Freeze current state when alarm silence button is pushed
if horn_silence OR alarm_clear OR re_take_snapshot then
    last_state_0 := alarm_word_0;
    last_state_1 := alarm_word_1;
    last_state_2 := alarm_word_2;
    last_state_3 := alarm_word_3;
    last_state_4 := alarm_word_4;
    last_state_5 := alarm_word_5;
end_if;

// compare current state with saved state, if different state_change_x will not be zero
state_change_0 := last_state_0 XOR alarm_word_0;
state_change_1 := last_state_1 XOR alarm_word_1;
state_change_2 := last_state_2 XOR alarm_word_2;
state_change_3 := last_state_3 XOR alarm_word_3;
state_change_4 := last_state_4 XOR alarm_word_4;
state_change_5 := last_state_4 XOR alarm_word_5;

// Make sure alarm is a current one and not a change from alarm to cleared (for self-clearing alarms
new_alarm_0 := state_change_0 AND alarm_word_0;
new_alarm_1 := state_change_1 AND alarm_word_1;
new_alarm_2 := state_change_2 AND alarm_word_2;
new_alarm_3 := state_change_3 AND alarm_word_3;
new_alarm_4 := state_change_4 AND alarm_word_4;
new_alarm_5 := state_change_5 AND alarm_word_5;
```

Figure 26: Alarm and Warning Add-On Instruction.

```
// If an alarm clears itself reprime the horn so it will sound if the same alarm occurs again
alarm_cleared_0 := state_change_0 AND NOT alarm_word_0;
alarm_cleared_1 := state_change_1 AND NOT alarm_word_1;
alarm_cleared_2 := state_change_2 AND NOT alarm_word_2;
alarm_cleared_3 := state_change_3 AND NOT alarm_word_3;
alarm_cleared_4 := state_change_4 AND NOT alarm_word_4;
alarm_cleared_5 := state_change_5 AND NOT alarm_word_5;

// if an alarm has cleared retake alarm state snapshot
if (alarm_cleared_0 <> 0) OR (alarm_cleared_1 <> 0) OR (alarm_cleared_2 <> 0) OR (alarm_cleared_3 <> 0) OR (alarm_cleared_4 <> 0) OR (alarm_cleared_5 <> 0) then
    re_take_snapshot := 1;
else
    re_take_snapshot := 0;
end_if;

// if the state has changed signal horn to come back on
if (new_alarm_0 <> 0) OR (new_alarm_1 <> 0) OR (new_alarm_2 <> 0) OR (new_alarm_3 <> 0) OR (new_alarm_4 <> 0) OR (new_alarm_5 <> 0) then
    re_enable_horn := 1;
else
    re_enable_horn := 0;
end_if;

// Mute horn if horn silence is pressed
if horn_silence AND NOT re_enable_horn then
    horn_silence_request := 1;
end_if;
if re_enable_horn then
    horn_silence_request := 0;
end_if;

/* If an alarm or warning exists turn on lamp */
if alarm_request OR warn_request then
    Lamp_on := 1;
else
    Lamp_on := 0;
end_if;

/* If an alarm exists and the silence horn button is not active turn on horn */
if alarm_request AND NOT horn_silence_request then
    Horn_on := 1;
else
    Horn_on := 0;
end_if;
```

Figure 27: Alarm and Warning Add-On Instruction, Continued.

As each alarm is a Boolean data type, it requires only one bit to command it on or off. Therefore, like digital inputs and outputs, they are stored in a 32-bit word called the alarm array (arr_alm). Each alarm is aliased to a bit. This logic makes controlling the horn and strobe much easier. The program looks to the alarm words used to see if the binary number formed by the word has changed, checking each pass of the program. If the program registers a change in any word, the strobe and the horn will be activated. This technique is a very clever way to accomplish this alarm function, as it takes care to only notify the operator of any new alarm conditions. If the operator knows of an alarm and turns the horn off (which does not clear the alarm or turn off the strobe), they do not have to do so again on the next program sweep when the same alarm is read again, as it will be the same value as the last sweep, so the horn stays off. If the alarm horn is stopped and the alarm condition has cleared, the word will be set back to 0, so if any new alarm condition is present, the horn and the strobe will activate anew and the operator knows the conditions have changed.

The feed pumps are two centrifugal pumps that move the water from the holding tanks into reaction tank 1. From there, the water flows by gravity through the pH adjustment processes, the inclined plate clarifier, and then the final pH tank. The activation of these pumps is controlled in the program through a number of logical steps. Figure 28 shows the feed pump power control block diagram, beginning when the water level reaches the "pump-on float" and ending with the pump being energized.

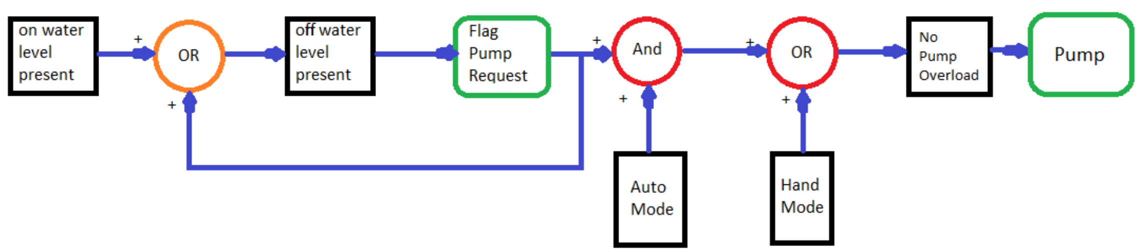

Figure 28: Feed Pump Power Control Block Diagram.

The power is sent to the pump if a high digital signal reaches the pump block. Several factors are taken into account. The level of the water, if the pump is being requested to run, is the pump in auto mode or hand mode, and if there is an overload condition at the pump. Figure 29 shows how this control is implemented in the program logic.

```
//*  Feed Pumps Run Requests and CP1A Active Conditions  *//

/* CP1X Run Request  */
pc_cp1x.level_in := lvl_t_30;
pc_cp1x.sp_start_level := sp_t_30_pump_on;
pc_cp1x.sp_stop_level := sp_t_30_pump_off;
pc_cp1x.low_alarm := alm_t30_soft_low;
PumpControl(pc_cp1x);
flg_cp1x_pump_run_req := pc_cp1x.pump_request;

if flg_cp1x_pump_run_req
    AND (NOT alm_t4_hh)
    AND (NOT alm_ph_stuck_op_int)
    AND flg_mixers_ok_for_feed
    AND flg_pH_meter_ok
then
    flg_system_run_ok := 1;
else
    flg_system_run_ok := 0;
end_if;


/*  Feed Pump Duplexing Comands  */
dp_cp1a_active.Pump_A_Running := cp1a_running;
dp_cp1a_active.Pump_B_Running := cp1b_running;
dp_cp1a_active.Pump_A_Mode_Auto := mde_cp_1a_in_auto;
dp_cp1a_active.Pump_B_Mode_Auto := mde_cp_1b_in_auto;
dp_cp1a_active.Pump_A_Mode_Off := mde_cp_1a_in_off;
dp_cp1a_active.Pump_B_Mode_Off := mde_cp_1b_in_off;
dp_cp1a_active.Pump_A_Overload := alm_cp1a_fault;
dp_cp1a_active.Pump_B_Overload := alm_cp1b_fault;
Duplex_Pumps(dp_cp1a_active);
flg_cp1a_active := dp_cp1a_active.Pump_A_Active_Pump;
```

Figure 29: Feed Pump Logic Using the PumpControl and Duplex_Pump Add-On Instruction.

The PumpControl add-on instruction uses floats and level parameters to determine when it is necessary to energize a feed pump. This logic is an example of how add-on instructions are called for using structured text: all input parameters are linked to the program tags, the add-on instruction is defined, and then the program tags for the output of the instruction are linked to the output parameter, which in this case is the pump run request. Figure 30 is the logic of the PumpControl add-on instruction.

```
/* Setup */
if pump_off_delay = 0 then
    pump_off_delay := 3;
end_if;
if pump_on_delay = 0 then
    pump_on_delay := 3;
end_if;

abt_pump_off.PRE := pump_off_delay * 1000;
abt_pump_on.PRE := pump_on_delay * 1000;

/* Pump start */
if level_in > sp_start_level then
    abt_pump_on.TimerEnable := 1;
else
    abt_pump_on.TimerEnable := 0;
end_if;
TONR(abt_pump_on);

if abt_pump_on.DN then
    pump_request := 1;
end_if;

/* Pump Stop */
if level_in < sp_stop_level then
    abt_pump_off.TimerEnable := 1;
else
    abt_pump_off.TimerEnable := 0;
end_if;
TONR(abt_pump_off);

if abt_pump_off.DN
    OR low_alarm
then
    pump_request := 0;
end_if;
```

Figure 30: PumpControl Add-On Instruction Logic.

The next bit of logic checks to see if there is anything stopping the feed pumps from being energized, such as a pH level that is out of range, or the final pH adjust tank is in high level. If these conditions are met, then the system is flagged as being acceptable to run the feed pumps, and the logic is passed to the Duplex_Pump add-on instruction. Duplexing is logic put in place when there are two feed pumps. The pumps are designated as either Pump A or Pump B. The logic requests Pump A to run first. If it is not in off mode or in some fault condition, Pump A will be energized when in auto mode. This will be the case for five requests. The sixth request goes to Pump B, and as long as it is not off, or in some fault condition, Pump B will be energized. This means that Pump A will run five times more often than Pump B. Duplexing is used to ensure that the two feed pumps will not require maintenance or replacing at the same time.

The request to turn on, having been established, activates the feedback loop that determines the frequency to be sent to the pump's VFD. The system is programmed to run at a flow rate selected by the operator and produced by the feed pumps. The flow rate, measured in gallons per minute, is a setpoint selected between 0 and 50, limited by the flow capabilities of the inclined plate clarifier. Exceeding the capabilities of the inclined plate clarifier will result in solids in the effluent, and this is to be avoided. The rate the operator chooses is dependent on several factors, such as how much water needs to be treated each day, and how much chemical is required to treat that much water. For the first month or so, the operator can adjust the flow rate to determine which is preferred to maximize gallons of treated water and minimize the cost of chemicals used.

The feed pumps, the pumps that move water from the holding tanks to the first reaction tank, are controlled via a variable frequency drive (VFD). By varying the frequency sent to the VFD, the revolutions per minute of the pump can be controlled, and thus, the flow rate out of the pump and into the system, as well. If the flow of the water into the system needs to increase, the frequency sent to the VFD would be increased.

The use of VFDs not only give the operators control of the flow, but they are also a cost effective way of running the large centrifugal pumps used in feeding the system. Instead of the pump being all the way on all the time, only the power necessary to run at a given flow rate is used.

The use of the VFDs leads to the use of a proportional-integral-derivative feed-back loop, an operation utilized in control theory. The PID flow control diagram, shown in Figure 31, depicts the basic flow control loop system. The PID flow control diagram is a functional block diagram, which was briefly discussed earlier in the project description section of this report. To further clarify, as Harrison and Bollinger [45] observe,

> Block diagrams provide a convenient way means for visualizing and analyzing control systems. These diagrams are obtained by first writing the equations which describe the behavior of each of the elements which make up the system. Then, the information contained in each of the equations is put in the form of a ratio of some output quantity to some input quantity. The relationship obtained is called a *transfer function* and is the mathematical representation of the particular element which is placed in the block. When all the elements in a system are represented in suitably related blocks, the over-all system equation can be obtained by a manipulation of the block diagram rather than by the simultaneous solution of the system equations by the usual mathematical methods. The transfer function for a linear element, component, or system can be defined as the ratio of the transform of the output to the transform of the input with the assumption that all initial conditions are zero. The transfer function for a particular element can be obtained by using the following three steps.
>
> 1. Write the appropriate equation which defines the behavior of the element.
> 2. Transform this equation assuming all initial conditions to be zero.
> 3. Form the ratio of out $O(s)$ to $I(s)$ as

$$\frac{O(s)}{I(s)} = F(s).$$

The transfer function is *F(s).*

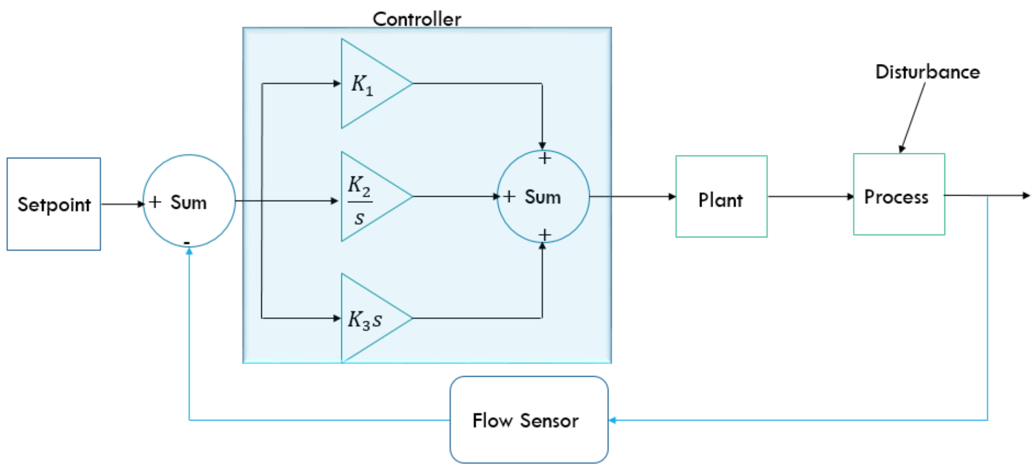

Figure 31: PID Flow Control Diagram [18].

In Figure 31, as Ogata [18] explains, the flow read by the sensor – which goes through the change of units in the scaling part of the program – is subtracted from the setpoint. The result is called the error. This error is sent through three separate operations, including (1) direct multiplication of K_1 (the gain of the proportional leg), (2) the integrating operation in state space multiplied by K_2 (the gain of the integral leg), and (3) the derivative operation multiplied by K_3 (the gain of the derivative leg). This is the PID part of the control system. The resulting transfer function is shown in Equation (2) – from Ogata [18] -- which is the sum of each branch:

$$G_c(s) = K_1 + \frac{K_2}{s} + K_3 s. \quad (2)$$

Here, $G_c(s)$ represents the controller.

Figure 32 shows the actual controller section of the system block diagram used by the Allen-Bradley PID operation.

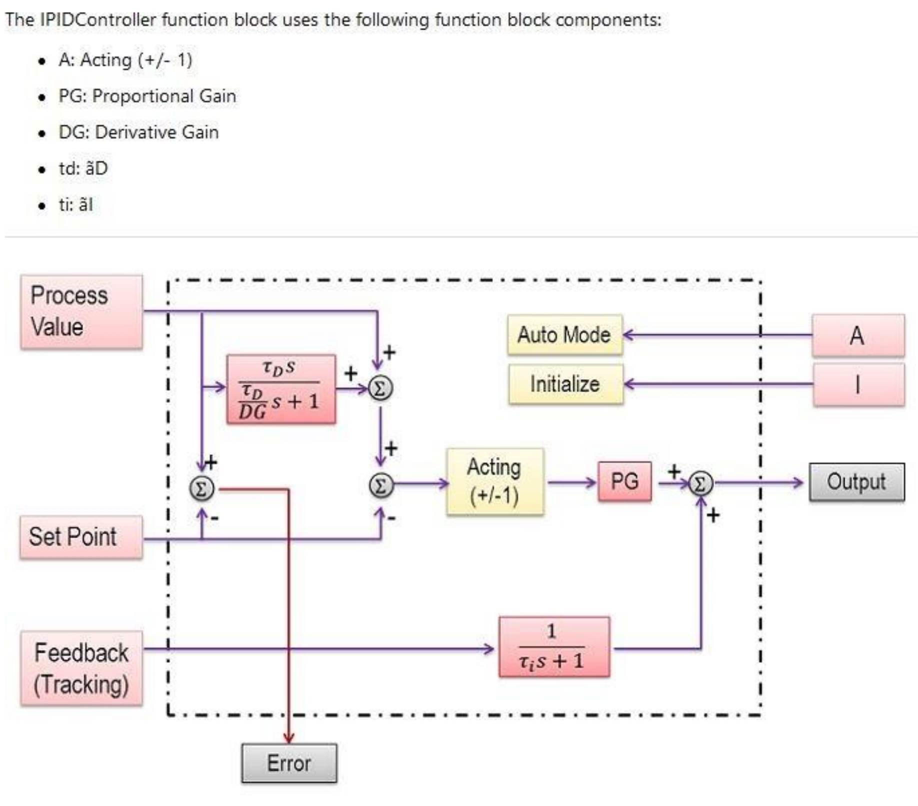

Figure 32: PIDE Controller Functional Block [46].

The setpoint the operator entered is input into the PLC, which then compares it to the signal from the flowmeter. In this operation, the setpoint is subtracted from the process variable -- in this case, the flow. That process variable, or flow, goes through the derivative transfer function, which is summed with the process variable itself. The original setpoint is also subtracted from this sum and is passed on to a decision block. Here the program determines if the frequency is to be raised or lowered. Once decided, the signal now proceeds through the proportional transfer function, a gain multiplier. The resultant signal is added to the feedback (this comes from the output of the PID block itself), which has gone through the Integral transfer function. This sum is the output frequency, which is sent to the VFD.

Each of these gains are calculated within the Allen-Bradley PID function block using the auto-tune function. Figure 33 is an excerpt from the Help menu of the Allen Bradley Studio 5000 software. It is a description of how the particular gains are calculated while tuning the enhanced PIDE loop (i.e., PIDE).

Independent Gains Form

$$CV_n = CV_{n-1} + K_P \Delta E + \frac{K_I}{60} E\Delta t + 60K_D \frac{E_n - 2E_{n-1} + E_{n-2}}{\Delta t}$$

PIDE term:	Description:
CV	control variable
E	error in percent of span
Δt	update time in seconds used by the loop
K_P	proportional gain
K_I	integral gain in min^{-1} a larger value of K_I causes a faster integral response.
K_D	derivative gain in minutes

Figure 33: Excerpt from Allen Bradly Help Section of Studio 5000, Search: PIDE [46].

The Allen-Bradley software comes with its own PID control function, so using a model to predict behavior and calculate gains ahead of time is not necessary, though the use of models in pH control is discussed in the literature review section of this report.

Figure 34 is the Allen Bradley PID routine used in the Feed Pump A and B speed control.

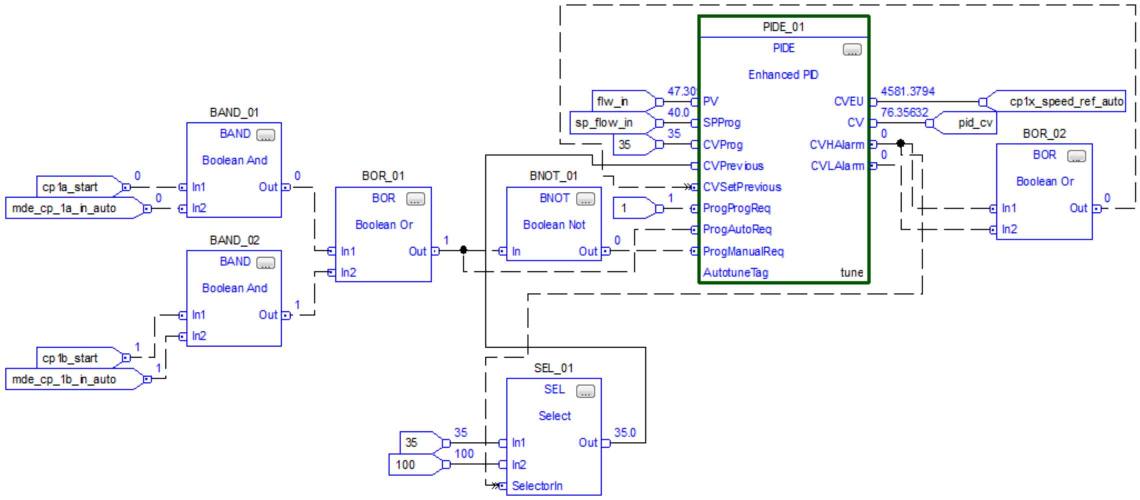

Figure 34: PIDE Routine for Feed Pump A and B.

The initializer is the Boolean signal that results from the pump request when in auto, or when the pump is in manual control, as previously described. These signals are referred to as *ProgAutoReq* and *ProgManualReq*. The *flow_in*, the scaled analog input, is the Process Variable (PV) in GPMs. The *sp_flow_in* is the setpoint for the process variable (SPProg). The PIDE instruction detects the difference and proceeds through the operations to output a new frequency to the VFD. One output is the Control Variable (CV), which is the percentage of the maximum frequency allowed (maximum being 60 Hz from the facility), shown in Figure 34 to be 76.36%. The Control Variable Engineering Units (CVEU) is the actual frequency output used as the pump's speed reference, shown as 4581 (the actual frequency recognized by the VFD is 45.81 Hz, as it knows where to put the decimal).

This is only the control section of the system. The Plant and the Process sections remain. The Plant transfer function refers to the transfer function of the VFD. Figure 35 is the block diagram describing how the VFD converts the frequency input to the signal output required by the motor to spin at the required RPM.

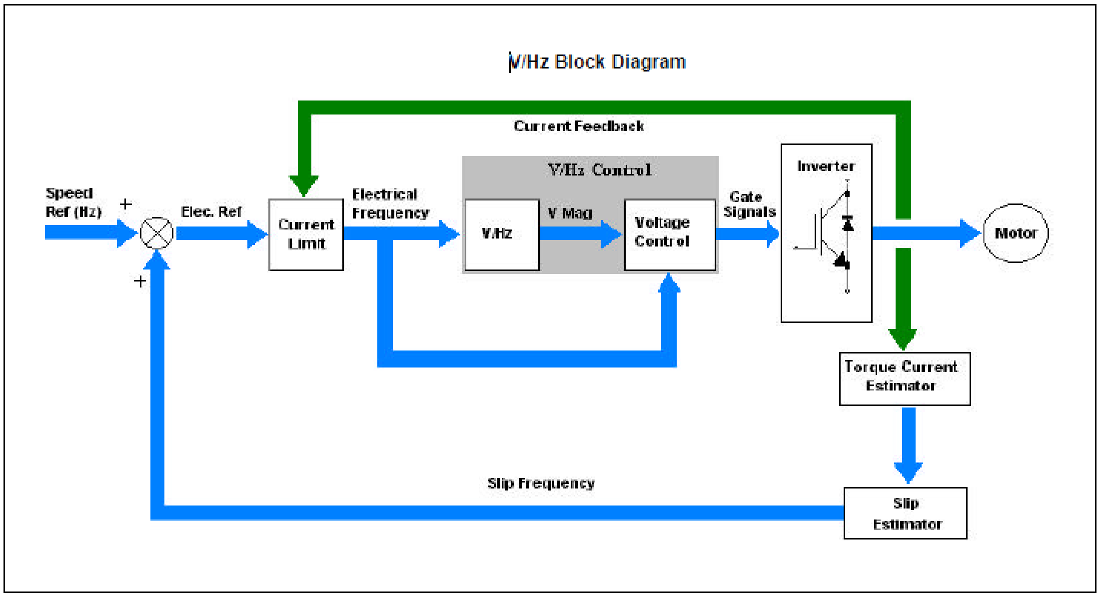

Figure 35: V/Hz Block Diagram Used by Allen-Bradley VFDs [47].

A transfer function is not associated with this process, as the RPMs of the impeller in the pump are proportional to the frequency into the VFD. Equation (3) – as recommended by the Diesel Service & Supply website [48] -- describes this relationship[3]:

$$RPM = \frac{120 * frequency}{Number\ of\ Poles\ in\ the\ Motor}. \qquad (3)$$

[3] The two factors are related as per the following formula:
Generator Frequency (f) = Number of revolutions per minute of the engine (N) * Number of magnetic poles (P) / 120. Conversely, P = 120*f/N. Thus, a 2-pole generator producing an output frequency of 60 Hz has an engine speed of 3,600 rpm.

The RPMs of the impeller of the motor determine the flow, which is based on the size of the impeller, the head loss of the pump, the horsepower of the motor, and the pipe diameter. For this project, 1800 RPMs was calculated to be the maximum speed using a 4-pole motor. At 75%, the impeller velocity is 1350 RPMs, which produces a flow in this system of 40 GPMs. The flow rate is proportional to the RPM of the pump's impeller [48].

This change is the disturbance that takes place in the Process block of the system. The sensor senses the flow, which is compared to the operator-selected setpoint. If there is a difference, the program sends the request through the PID loop to change the frequency sent to the pump. The pump reads the frequency and spins the impeller to accomplish the flow rate change. This change is the new disturbance that the flowmeter senses and sends back to the PLC, and so on.

There is a delay before the flow sensor feels that change, but it is slight. The result of this loop is a gradual increase or decrease of the flow rate, with very little overshoot of the setpoint, and a quickly acquired steady-state. Then, speed of these pumps, or the frequency the VFD is sending to the pumps, is dependent on the setpoint the operator sets on the HMI screen.

Once the request for the feed pump to run is set, and the VFDs are given their frequency, the request is sent to the equipment routine. The equipment routine is the part of the programing that looks to see what mode a device is in, and whether there is a request to energize it – if so, the program then passes the command to the device. This happens directly for all equipment not requiring a VFD. The feed pumps do require a VFD, so there is that extra step between the command to run and the command that actually turns on the pump. Figures 36 and 37 show the equipment logic for the feed pumps. Figures 38 and Figure 39 show the VFD routine FBDs that address pump control signals, such as activation and speed, from the program tags to the VFD. Also, the fault and running conditions are linked to the program tags within the VFD routine.

```
//* Feed Pumps *//

/* CP-1A */
if ((mde_cp_1a_in_auto AND flg_system_run_ok AND flg_cp1a_active)
    OR mde_cp_1a_in_hand)
    AND (NOT alm_cp1a_fault)
    AND flg_system_power
    AND (NOT flg_sys_start)
then
    cp1a_start := 1;
else
    cp1a_start := 0;
end_if;

if mde_cp_1a_in_auto
    AND cp1a_start
then
    cp1a_speed := cp1x_speed_ref_auto;
elsif mde_cp_1a_in_hand
    AND cp1a_start
then
    cp1a_speed := sp_vfd_cp1a_man_speed*100;
else
    cp1a_speed := 0;
end_if;

cp1a_speed_hmi := (cp1a_speed/60);
```

Figure 36: Feed Pump A Logic from the Equipment Routine.

```
/* CP-1B */
if ((mde_cp_1b_in_auto AND flg_system_run_ok AND (NOT flg_cp1a_active))
    OR mde_cp_1b_in_hand)
    AND (NOT alm_cp1b_fault)
    AND flg_system_power
    AND (NOT flg_sys_start)
then
    cp1b_start := 1;
else
    cp1b_start := 0;
end_if;

if mde_cp_1b_in_auto
    AND cp1b_start
then
    cp1b_speed := cp1x_speed_ref_auto;
elsif mde_cp_1b_in_hand
    AND cp1b_start
then
    cp1b_speed := sp_vfd_cp1b_man_speed*100;
else
    cp1b_speed := 0;
end_if;

cp1b_speed_hmi := (cp1b_speed/60);
```

Figure 37: Feed Pump B Logic from the Equipment Routine.

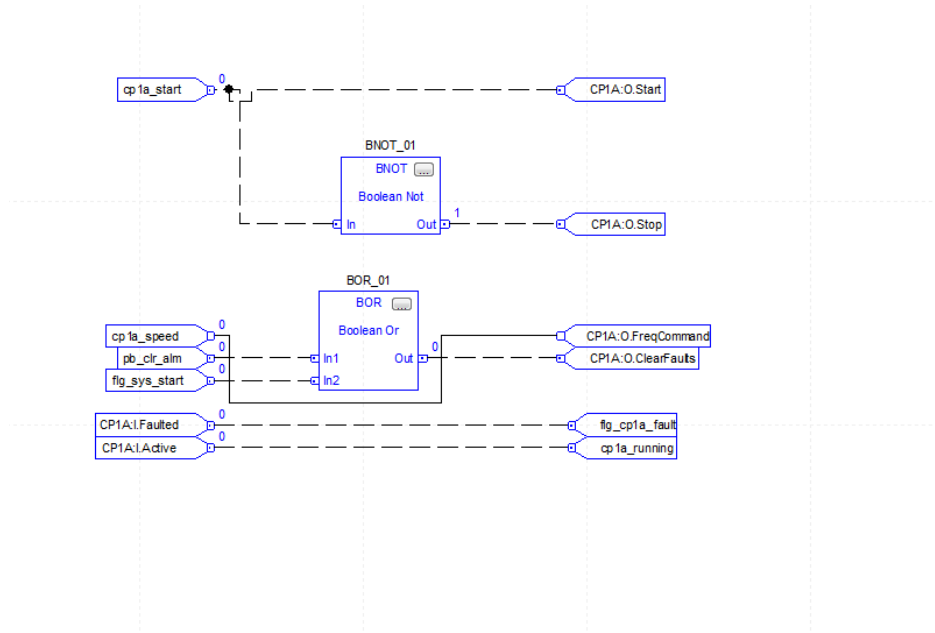

Figure 38: Feed Pump A Program Tags Linked to Pump A VFD.

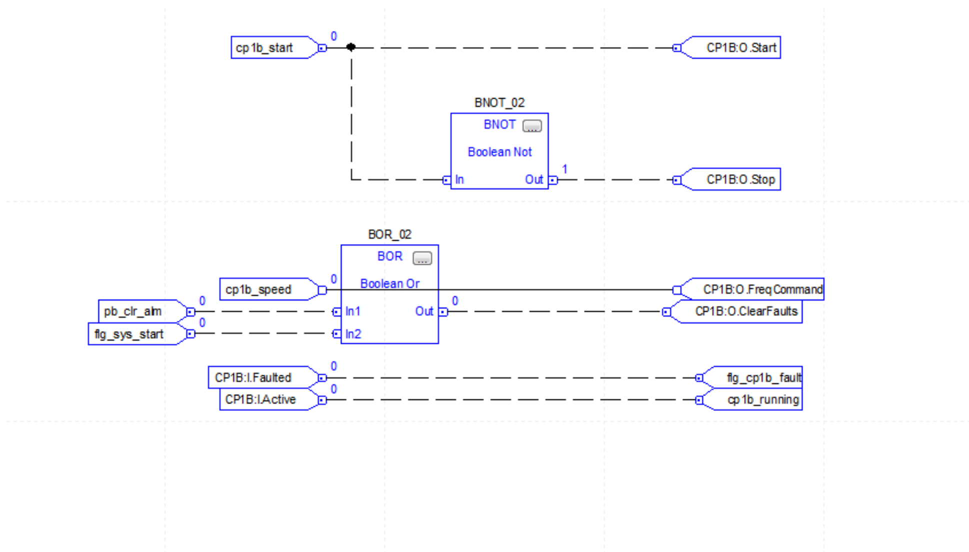

Figure 39: Feed Pump B Program Tags Linked to Pump B VFD.

When the water reaches reaction tank 1, the pH adjustment system takes over. The first step in removing nickel is to break the chelator's (cleaner's) hold on it. The chelator holds the nickel in solution and in order to get it out of solution (remove it from the water), the chelator needs to let go. This is accomplished by lowering the pH in the first reaction tank (making it acidic) to around 2.1. This process is known as acid cracking. Automating the pH adjustment system requires a decision procedure to take place. To illustrate how the PLC controls the pH adjustment, Figure 40 has been provided.

Acid cracking can only be achieved at very specific pH levels, determined by which chelators are holding which metals. Because of the necessity for this fine tuning, a double-sided decision control is used. That means that the pH can be adjusted both up (by adding caustic) and down (by adding acid). Figure 40 shows the two-sided pH decision diagram.

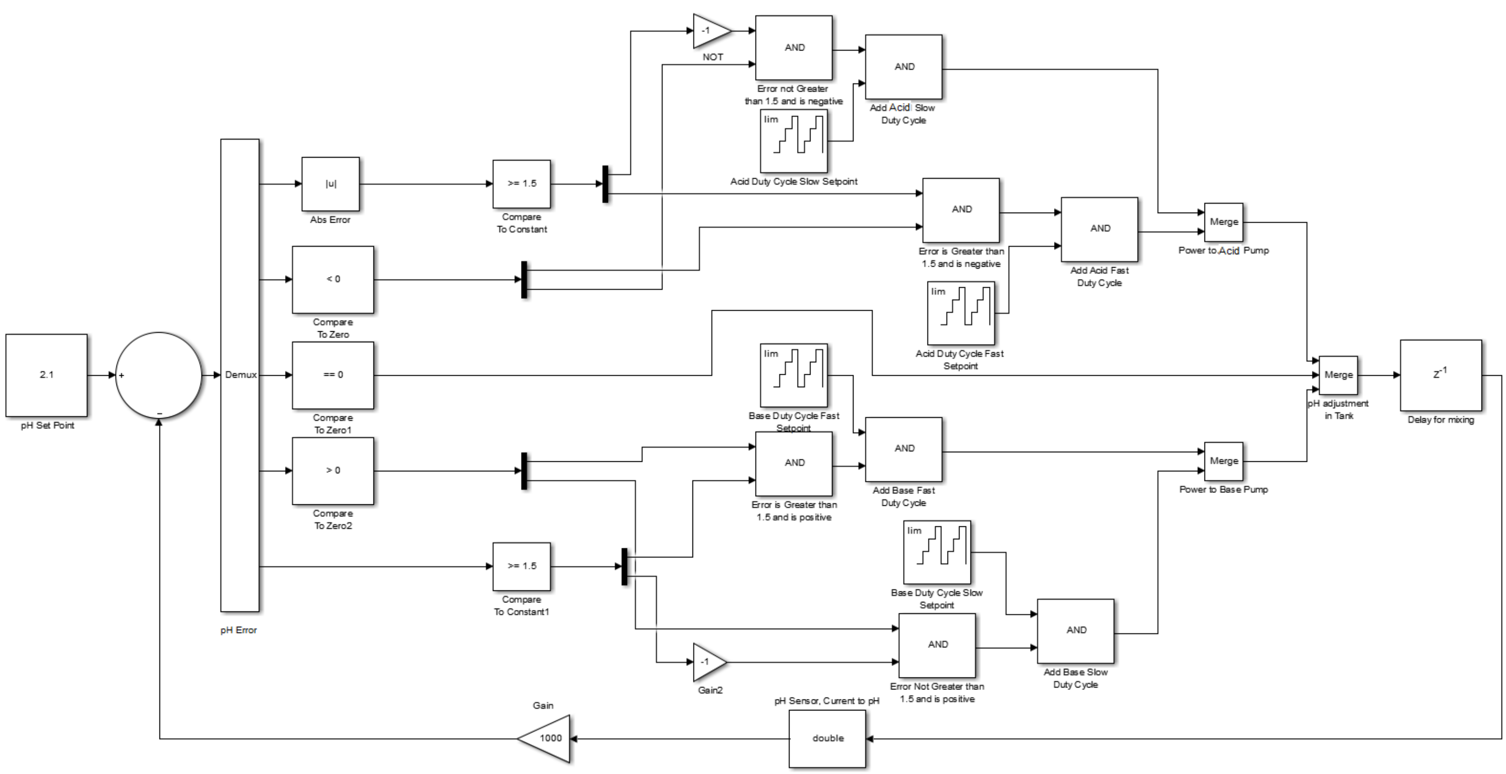

Figure 40: Two Sided pH Decision Diagram.

Figure 40 shows the PLC's decision process in adjusting the pH. The setpoint is compared to the sensor input. If the absolute value of the difference (the error) is greater than 1.5 and the error is negative, the fast Acid Duty Cycle is called and the pump will turn on for as many seconds (out of ten) that the operator has set as the fast acid duty cycle. If the absolute value of the error is less than 1.5 and it is negative, then the slow acid duty cycle is called and the pump will turn on for as many seconds that the operator has set for the slow acid duty cycle. If the error is greater than 1.5 and is positive, then the fast caustic duty cycle is called and the caustic pump will turn on for as many seconds (out of ten) that the operator has set as the fast caustic duty cycle. If the error is less than 1.5 and it is positive, then the slow caustic duty cycle is called. And finally, if the error is equal to zero, there is no chemical addition to the tank.

PH adjustment also takes place in reaction tank 2 and in the final pH adjustment tank, though each of these use one-sided adjustments. In reaction tank 2, there is a minimum limit the pH can be in order for the nickel removal process to continue. This is set around 10.1. If the pH reads below that limit, the caustic chemical feed pump will be requested to run. The water that comes from reaction tank 1 is set so low that the natural drift in pH is upward. The control diagram is the same as the caustic side of the two-sided diagram, including the case where the error is equal to zero. The pH in the final tank needs to be maintained no higher than 8.5. This makes it an upper limit, one-sided pH control. The pH control diagram for the final pH adjustment tank is the same as the acid side of the two-sided diagram, including the case where the error is zero. Figures 41a and 41b show the logic used for each of the tanks containing pH adjustment and the chemical feed pump run requests.

```
/* Defaults */
if ph_stuck_time <= 0 then
    ph_stuck_time := 5;
end_if;
if acid_normal_duty_cycle <= 0 then
    acid_normal_duty_cycle := 25;
end_if;
if acid_fast_duty_cycle <= 0 then
    acid_fast_duty_cycle := 60;
end_if;
if base_normal_duty_cycle <= 0 then
    base_normal_duty_cycle := 25;
end_if;
if base_fast_duty_cycle <= 0 then
    base_fast_duty_cycle := 60;
end_if;

/* pH levels for acid addition */
if (pH_reading < sp_high_ph_limit)
then
    acid_adjust := 0;
end_if;

if (pH_reading > sp_high_ph_limit)
    AND (pH_reading <= (sp_high_ph_limit + 1.5))
then
    acid_adjust := 1;
end_if;

if (pH_reading > (sp_high_ph_limit + 1.5))
then
    acid_adjust := 2;
end_if;

/* pH levels for base addition */
if (pH_reading > sp_low_ph_limit)
then
    base_adjust := 0;
end_if;

if (pH_reading < sp_low_ph_limit)
    AND (pH_reading >= (sp_low_ph_limit - 1.5))
then
    base_adjust := 1;
end_if;

if (pH_reading < (sp_low_ph_limit - 1.5))
then
    base_adjust := 2;
end_if;
```

```
/* duty cycle settings */
internal_acid_dc[1] := acid_normal_duty_cycle * 100;
internal_acid_dc[2] := acid_fast_duty_cycle * 100;
internal_base_dc[1] := base_normal_duty_cycle * 100;
internal_base_dc[2] := base_fast_duty_cycle * 100;
tmr_acid_off.PRE := 10000 - internal_acid_dc[acid_adjust];
tmr_base_off.PRE := 10000 - internal_base_dc[base_adjust];

if acid_adjust = 1 then
    tmr_acid_on.PRE := internal_acid_dc[1];
    flag_run_acid := 1;
elsif acid_adjust = 2 then
    tmr_acid_on.PRE := internal_acid_dc[2];
    flag_run_acid := 1;
else
    flag_run_acid := 0;
end_if;

if base_adjust = 1 then
    tmr_base_on.PRE := internal_base_dc[1];
    flag_run_base := 1;
elsif base_adjust = 2 then
    tmr_base_on.PRE := internal_base_dc[2];
    flag_run_base := 1;
else
    flag_run_base := 0;
end_if;
```

Figure 41a: pH Control Logic Chemical Feed Pump Run Requests.

```
/* Chemical feed pump timers */
if (NOT tmr_acid_off.DN)
    AND flag_run_acid
then
    tmr_acid_on.TimerEnable := 1;
else
    tmr_acid_on.TimerEnable := 0;
end_if;
if (tmr_acid_on.DN)
    AND flag_run_acid
then
    tmr_acid_off.TimerEnable := 1;
else
    tmr_acid_off.TimerEnable := 0;
end_if;

if (NOT tmr_base_off.DN)
    AND flag_run_base
then
    tmr_base_on.TimerEnable := 1;
else
    tmr_base_on.TimerEnable := 0;
end_if;
if (tmr_base_on.DN)
    AND flag_run_base
then
    tmr_base_off.TimerEnable := 1;
else
    tmr_base_off.TimerEnable := 0;
end_if;

TONR(tmr_acid_on);
TONR(tmr_acid_off);
TONR(tmr_base_on);
TONR(tmr_base_off);
```

```
/* Chemical feed pump control */
if tmr_acid_on.TT
    AND treat_enable
then
    acid_pump_request := 1;
else
    acid_pump_request := 0;
end_if;

if tmr_base_on.TT
    AND treat_enable
then
    base_pump_request := 1;
else
    base_pump_request := 0;
end_if;

/* pH stuck alarms */
tmr_stuck_high.PRE := ph_stuck_time * 60 * 1000;
tmr_stuck_low.PRE := ph_stuck_time * 60 * 1000;
if flag_run_acid
    AND treat_enable
then
    tmr_stuck_high.TimerEnable := 1;
else
    tmr_stuck_high.TimerEnable := 0;
end_if;
if flag_run_base
    AND treat_enable
then
    tmr_stuck_low.TimerEnable := 1;
else
    tmr_stuck_low.TimerEnable := 0;
end_if;
TONR(tmr_stuck_low);
TONR(tmr_stuck_high);

alarm_ph_stuck_high := tmr_stuck_high.DN;
alarm_ph_stuck_low := tmr_stuck_low.DN;
```

Figure 41b: pH Control Logic Chemical Feed Pump Run Requests.

As mentioned with the feed pumps, the equipment routine contains the logic that sends the power command to a device's relay. This is required for the chemical feed pumps, AOD pumps, and mixers, as well. Figures 42 through 45 show the program logic for the equipment routine for each device relating to reaction tank 1, reaction tank 2, flocculation tank 3, tank 8 and the final pH adjustment tank 4.

```
//*   T-1   *//

/* M-1 */
if (mde_m1_in_auto OR mde_m1_in_hand)
    AND (NOT alm_m1_ol)
    AND flg_system_power
then
    m1 := 1;
else
    m1 := 0;
end_if;

/*  CF-1A  */
if ((mde_cf_1a_in_auto AND flg_cf1a_req)
    OR mde_cf_1a_in_hand)
    AND flg_system_power
then
    cf_1a := 1;
else
    cf_1a := 0;
end_if;

/*  CF-1B  */
if ((mde_cf_1b_in_auto AND flg_cf1b_run_req)
    OR mde_cf_1b_in_hand)
    AND flg_system_power
then
    cf_1b := 1;
else
    cf_1b := 0;
end_if;

/*  CF-1D  */
if ((mde_cf_1d_in_auto AND flg_cf1d_req)
    OR mde_cf_1d_in_hand)
    AND flg_system_power
then
    cf_1d := 1;
else
    cf_1d := 0;
end_if;
```

Figure 42: Equipment Routine Logic for Devices Related to Reaction Tank T-1.

```
//*  T-2  *//

/* M-2 */
if (mde_m2_in_auto OR mde_m2_in_hand)
    AND (NOT alm_m2_ol)
    AND flg_system_power
then
    m2 := 1;
else
    m2 := 0;
end_if;

/*  CF-2A  */
if ((mde_cf_2a_in_auto AND flg_cf2a_req)
    OR mde_cf_2a_in_hand)
    AND flg_system_power
then
    cf_2a := 1;
else
    cf_2a := 0;
end_if;

/*  CF-2B  */
if ((mde_cf_2b_in_auto AND flg_cf2b_run_req)
    OR mde_cf_2b_in_hand)
    AND flg_system_power
then
    cf_2b := 1;
else
    cf_2b := 0;
end_if;
```

Figure 43: Equipment Routine Logic for Devices Related to Reaction Tank T-2.

```
//* T-3  *//

/* M-3 */
if (mde_m3_in_auto OR mde_m3_in_hand)
    AND (NOT alm_m3_fault)
    AND flg_system_power
    AND (NOT flg_sys_start)
then
    m3_start := 1;
else
    m3_start := 0;
end_if;

m3_speed := sp_vfd_m3_man_speed*100;
m3_speed_hmi := m3_speed / 60;

/*  CF-3A  */
if ((mde_cf_3a_in_auto AND flg_cf3a_run_req)
    OR mde_cf_3a_in_hand)
    AND flg_system_power
then
    cf_3a := 1;
else
    cf_3a := 0;
end_if;


/* AOD-3 Seed  */
if ((mde_aod_3_in_auto AND flg_seed_pump_req)
    OR mde_aod_3_in_hand)
    AND flg_system_power
then
    aod_3 := 1;
else
    aod_3 := 0;
end_if;
```

Figure 44: Equipment Routine Logic for Devices Related to Flocculation Tank T-3.

```
//* T-8 *//

/* M-8 */
if (mde_m8_in_auto OR mde_m8_in_hand)
    AND (NOT alm_m8_ol)
    AND flg_system_power
then
    m8 := 1;
else
    m8 := 0;
end_if;


//* T-4 *//

/* M-4 */
if (mde_m4_in_auto OR mde_m4_in_hand)
    AND (NOT alm_m4_ol)
    AND flg_system_power
then
    m4 := 1;
else
    m4 := 0;
end_if;

/*  CF-4  */
if ((mde_cf_4_in_auto AND flg_cf4_req)
    OR mde_cf_4_in_hand)
    AND flg_system_power
then
    cf_4 := 1;
else
    cf_4 := 0;
end_if;
```

Figure 45: Equipment Routine Logic for Devices Related to Tank T-8 and Final pH Adjustment Tank T-4.

Each piece of equipment features request logic that must be passed before it can be energized. Figure 46 shows the request logic.

```
/* Treatment running indication with runon */
tmr_system_run_on.PRE := (1000*60) * 5;
if cp1a_running
    OR cp1b_running
then
    tmr_system_run_on.TimerEnable := 1;
else
    tmr_system_run_on.TimerEnable := 0;
end_if;
TOFR(tmr_system_run_on);
flg_treat_run_on := tmr_system_run_on.DN;

/* t-1 pH Adjustment */
phadj_t1.pH_reading := ph_t_1;
phadj_t1.sp_high_ph_limit := sp_t_1_upper_pH;
phadj_t1.sp_low_ph_limit := sp_t_1_lower_pH;
phadj_t1.acid_normal_duty_cycle := sp_t_1_acid_norm;
phadj_t1.acid_fast_duty_cycle := sp_t_1_acid_fast;
phadj_t1.base_normal_duty_cycle := sp_t_1_base_norm;
phadj_t1.base_fast_duty_cycle := sp_t_1_base_fast;
phadj_t1.treat_enable := flg_treat_run_on;
FT_pH_Maintain(phadj_t1);
flg_cf1a_req := phadj_t1.acid_pump_request;
flg_cf1d_req := phadj_t1.base_pump_request;

/* t-2 pH Adjustment */
phadj_t2.pH_reading := ph_t_2;
phadj_t2.sp_low_ph_limit := sp_t_2_lower_pH;
phadj_t2.base_normal_duty_cycle := sp_t_2_base_norm;
phadj_t2.base_fast_duty_cycle := sp_t_2_base_fast;
phadj_t2.treat_enable := flg_treat_run_on;
FT_pH_Maintain(phadj_t2);
flg_cf2a_req := phadj_t2.base_pump_request;
```

```
/* clarifier, sludge and auger */
tmr_sldg_on.PRE := (sp_sldg_on_min * (1000*60)) + (sp_sldg_on_sec * 1000);
tmr_sldg_off.PRE := (sp_sldg_off_min * (1000*60)) + (sp_sldg_off_sec * 1000);
if flg_treat_run_on
    AND NOT tmr_sldg_off.DN
then
    tmr_sldg_on.TimerEnable := 1;
else
    tmr_sldg_on.TimerEnable := 0;
end_if;
if flg_treat_run_on
    AND tmr_sldg_on.DN
then
    tmr_sldg_off.TimerEnable := 1;
else
    tmr_sldg_off.TimerEnable := 0;
end_if;
TONR(tmr_sldg_on);
TONR(tmr_sldg_off);
flg_sldg_run_req := tmr_sldg_on.TT;

/* seed pump aod-3 */
tmr_seed_on.PRE := (sp_seed_on_min * (1000*60)) + (sp_seed_on_sec * 1000);
tmr_seed_off.PRE := (sp_seed_off_min * (1000*60)) + (sp_seed_off_sec * 1000);
if flg_treat_run_on
    AND NOT tmr_seed_off.DN
then
    tmr_seed_on.TimerEnable := 1;
else
    tmr_seed_on.TimerEnable := 0;
end_if;
if flg_treat_run_on
    AND tmr_seed_on.DN
then
    tmr_seed_off.TimerEnable := 1;
else
    tmr_seed_off.TimerEnable := 0;
end_if;
TONR(tmr_seed_on);
TONR(tmr_seed_off);
flg_seed_pump_req := tmr_seed_on.TT;

/* t-4 pH adjustment */
phadj_t4.pH_reading := ph_t_4;
phadj_t4.sp_high_ph_limit := sp_t_4_upper_pH;
phadj_t4.acid_normal_duty_cycle := sp_t_4_acid_norm;
phadj_t4.acid_fast_duty_cycle := sp_t_4_acid_fast;
phadj_t4.treat_enable := flg_treat_run_on;
FT_pH_Maintain(phadj_t4);
flg_cf4_req := phadj_t4.acid_pump_request;
```

Figure 46: Equipment Request to Run Logic.

The *FT_pH_Maintenance* add-on instruction was used to great effect. It contains logic for two-sided pH adjustment, but as it is up to the programmer which parameters are called on, the logic works for all three scenarios.

Conclusions and Recommendations

This report describes and explains the resolution of an actual pH control problem associated with an industrial wastewater pre-treatment system in a metal plating plant. Metal plating is achieved through the use of two fundamental technologies, including electroless plating and electroplating. The processes associated with both technologies produce contaminated wastewater, which typically features toxic and hazardous materials, such as metals. The contaminated wastewater must be pre-treated by the metal plating plant so that the toxic and hazardous materials are removed before the wastewater can be safely pumped to a public wastewater treatment plant, where it is further treated so that it can be safely released to the environment.

The characteristics of metal plating wastewater depend almost entirely on the metal plating process itself, including the metals and materials employed in the plating process. In the metal plating wastewater that is the focus of this report, it was necessary in the pre-treatment process to control pH levels at various stages so that the metal nickel – which was employed in the plating process – could be removed. This report details the successful implementation of a closed-loop, PLC-based control system, which was designed, programmed, and built in order to automatically control the pH levels. This report particularly focuses on the programming of the PLC.

As Figure 47 demonstrates, after the pH control system was implemented, it successfully controlled the pH levels in the wastewater pre-treatment stages, so that the nickel was removed, eventually resulting in wastewater effluent that meets regulatory requirements, and which could therefore be released to the local public wastewater treatment plant. Figure 47 specifically shows trend lines of pH levels in pH treatment tanks over a period of time of roughly six minutes. The trend lines are color-coded, and they include both a tank's setpoint (i.e., the desired pH value) and the tank's actual pH level (as measured by a sensor). In each case, the actual pH level performance complies with its corresponding setpoint. For example, the two trend lines at the top of Figure 47 represent the setpoint and the actual performance of tank 2. Tank 2 features a lower pH value of just under 10 for its setpoint. It is desired that the actual pH level not drop below this level, and in fact, the actual pH level performance does not drop below the setpoint level. The pH control is successful.

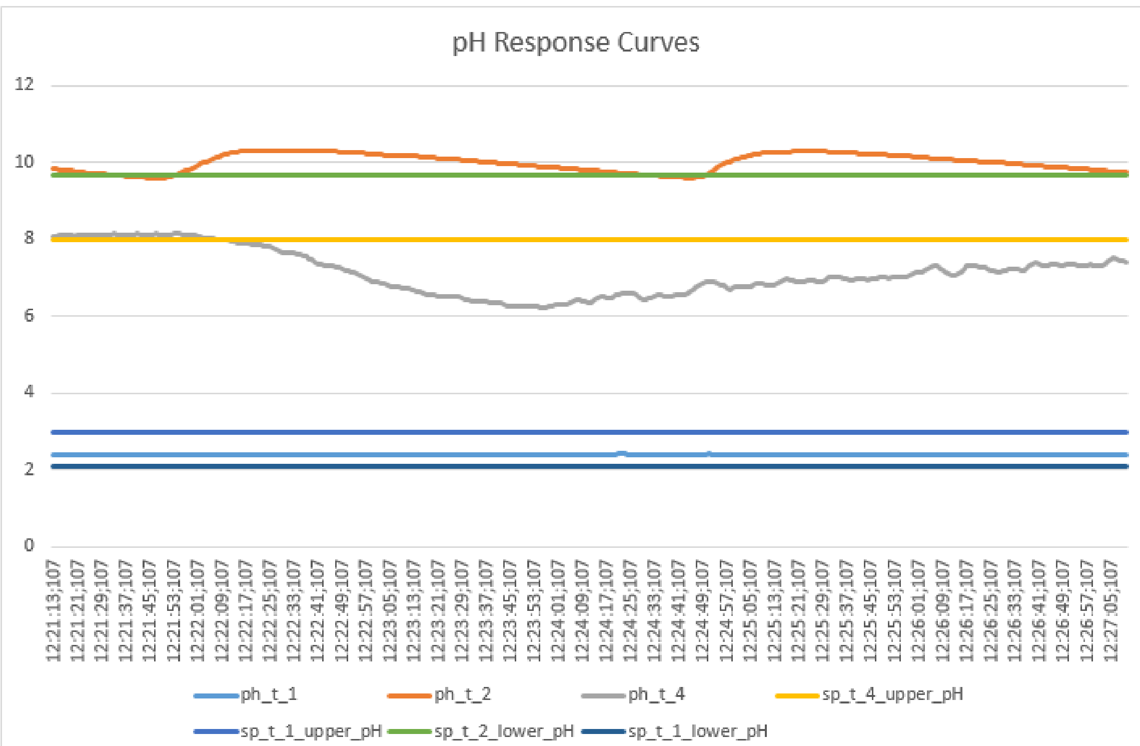

Figure 47: Chart of pH Levels and Setpoints in Each Tank.

In this project, a PID control loop was implemented with an off-the-shelf Allen-Bradley PLC microcontroller in order to successfully control pH levels in a wastewater treatment process. Although the PLC microcontroller was easily obtained, its programming was a complex and demanding endeavor, dependent on unique parameters, requirements, and characteristics associated with the wastewater treatment process. The literature review in this report demonstrates that pH control is challenging because of its nonlinear characteristics. As a result, a variety of pH control approaches have been developed and are discussed in the relevant literature. In this project, other pH control methods could have been designed and deployed. Moreover, once the decision was made to move forward with a PID-enabled, PLC-based, closed-loop system, numerous other strategies could have been explored in the design of the system, including strategies specifically focusing on PID tuning [49, 50]. However, every project features constraints. The investigation and implementation of other approaches, methods, and strategies in this project would have required significantly more resources than the ones that were available. As it is, the design, build, and programming of a PID-enabled, PLC-based control system for this project – although a conventional pH control approach – nonetheless served, and continues to serve, the needs of the customer in the project, and additionally served as a challenging professional endeavor for the author.

References

[1] NG, Wun Jern. 2006. *Industrial Wastewater Treatment*. London: Imperial College Press.

[2] Varela, Joe. 2015. *Hazardous Materials: Handbook for Emergency Responders.* New York, New York: Wiley. [Internet, WWW, Database]. *Available:* Available from the Ebrary e-book database; ADDRESS: http://site.ebrary.com/.

[3] Carruth, Russelyn S. and Bernard D. Goldstein. 2013. *Environmental Health Law: An Introduction.* New York: John Wiley & Sons, Inc. [Internet, WWW, Database]. *Available:* Available from the Ebrary e-book database; ADDRESS: http://site.ebrary.com/.

[4] Atlantic306. Last revision: 27 October 2016. "Code of Federal Regulations." [Internet, WWW]. *Available:* Available from the Wikipedia Free Encyclopedia website; ADDRESS: https://en.wikipedia.org/wiki/Code_of_Federal_Regulations.

[5] Cherry, Kenneth. 1982. *Plating Waste Treatment*. Ann Arbor, MI: Ann Arbor Science Publishers, Inc.

[6] Viguri, J., A. Andres, R. Ibanez, C. Ruiz Puente, and A. Irabien. 2000. "Characterization of Metal Finishing Sludges: Influence of pH." *Journal of Hazardous Materials*, A79, pp. 63-75. [Internet, WWW, Database]. *Available*: Available from the ScienceDirect Database; ADDRESS: http://www.sciencedirect.com/.

[7] Magalhaes, Jorge M., Joao E. Silva, Fernando P. Castro, Joao A. Labrincha. 2005. "Physical and Chemical Characterisation of Metal Finishing Industrial Wastes." *Journal of Environmental Management*, Vol. 75, pp. 157-166. [Internet, WWW, Database]. *Available*: Available from ScienceDirect Database; ADDRESS: http://www.sciencedirect.com/.

[8] Woodard, Frank. 2005. *Industrial Wastewater Treatment Handbook.* Butterworth-Heineman. [Internet, WWW, Database]. *Available:* Available from the Ebrary e-book database; ADDRESS: http://site.ebrary.com/.

[9] Weber, Thomas J. 2007. "Finishing Plant Engineering Wastewater Treatment." *Metal Finishing*, Vol. 105(10), pp. 699-714. [Internet, WWW, Database]. *Available*: Available from ScienceDirect Database; ADDRESS: http://www.sciencedirect.com/.

[10] Low, K.S., C.K. Lee, and A.C. Leo. 1995. "Removal of Metals from Electroplating Wastes Using Banana Pith." *Bioresource Technology*, Vol. 51(2-3), pp. 227-231. [Internet, WWW, Database]. *Available*: Available from ScienceDirect Database; ADDRESS: http://www.sciencedirect.com/.

[11] Qin, Jian-Jun, Maung Nyunt Wai, Maung Htun Oo, and Hsiaowan Lee. 20 February 2004. *Desalination*, Vol. 161(2), pp. 155-167. [Internet, WWW, Database]. *Available*: Available from ScienceDirect Database; ADDRESS: http://www.sciencedirect.com/.

[12] Vijayaraghavan, K. and Yeoung-Sang Yun. May-June 2008. "Bacterial Biosorbents and Biosorption." *Biotechnology Advances*, Vol. 26(3), pp. 266-291.

[Internet, WWW, Database]. *Available*: Available from ScienceDirect Database; ADDRESS: http://www.sciencedirect.com/.

[13] Suzuki, Yoshitake, Kazuchiro Mochidzuke, Yasushi Takeuchi, Yoshiteru Yashishita, Tad Fukuda, Hideo Amakusa, and Hiroshi Abe. April 1996. "Biological Activated Carbon Treatment of Effluent from Wastewater Treatment Processes of Plating Industries." *Separations Technology,* Vol. 6(2), pp. 147-153. [Internet, WWW, Database]. *Available*: Available from ScienceDirect Database; ADDRESS: http://www.sciencedirect.com/.

[14] Szafnicki, Konrad, Cécile Narcé, and Jacques Bourgois. June 2005. "Towards an Integrated Tool for Control, Supervision, and Operator Training – Application to Industrial Wastewater Detoxification Plants." *Control Engineering Practice,* Vol. 13(6), pp. 729-738. [Internet, WWW, Database]. *Available*: Available from ScienceDirect Database; ADDRESS: http://www.sciencedirect.com/.

[15] Joseph, David. July-August 2003. "The ABCs of pH Measurement in Metal Finishing." *Metal Finishing,* Vol. 101(7-8), pp. 36-43. [Internet, WWW, Database]. *Available*: Available from ScienceDirect Database; ADDRESS: http://www.sciencedirect.com/.

[16] U.S. Environmental Protection Agency. 14 December 2015. Code of Federal Regulations. Title 40: *Protection of the Environment*. Part 413: *Electroplating Point Source Category*. [Internet, WWW]. *Available:* Available from the U.S. Government Publishing Office; ADDRESS: http://www.ecfr.gov/cgi-bin/text-

idx?tpl=/ecfrbrowse/Title40/40cfr413_main_02.tpl. [Accessed: 16 December 2015].

[17] Cox, Richard. 1995. *Technician's Guide to Programmable Controllers*. Albany: Demar.

[18] Ogata, Katsuhiko. 2010. *Modern Control Systems.* Upper Saddle River, New Jersey: Prentice Hall.

[19] Underwriters Laboratory (UL). 20 December 2013. *Standards for Safety for Industrial Control Panels*. UL Standard 508A.

[20] Rani, K. Yamuna. 30 September 2013. "pH Control Approaches – An Overview." *Chemical Industry Digest.* [Internet, WWW, Database]. *Available:* Available from the Proquest SciTech Collection Database; ADDRESS: http://www.proquest.com/.

[21] Singh, Parikshit Kishor, Surekha Bhanot, and Hare Krishna Mohanta. November 2013. "Optimized and Self-Organized Fuzzy Logic Controller for pH Neutralization Process." *International Journal of Intelligent Systems and Applications*, Vol. 12, pp. 99-112. [Internet, WWW, Database]. *Available*: Available from ScienceDirect Database; ADDRESS: http://www.sciencedirect.com/.

[22] Tan, W.W., F. Lu, A.P. Loh, and K.C. Tan. 2005. "Modeling and Control of a Pilot pH Plant Using Genetic Algorithm." *Engineering Applications of Artificial Intelligence*, Vol. 18, pp. 485-494. [Internet, WWW, Database]. *Available*:

Available from ScienceDirect Database; ADDRESS: http://www.sciencedirect.com/.

[23] Salehi, Shahin, Mohammad Shahrokhi, and Ali Nejati. 2009. "Adaptive Nonlinear Control of pH Neutralization Processes Using Fuzzy Approximators." *Control Engineering Practice,* Vol. 17, pp. 1329-1337. [Internet, WWW, Database]. *Available*: Available from ScienceDirect Database; ADDRESS: http://www.sciencedirect.com/.

[24] Jebarani, Irene Martina and T. Rammohan. June 2014. "Fuzzy Logic Based PID Controller for pH Neutralization Process." *International Journal of Computer Applications*, Vol. 95(6), pp. 36-40. [Internet, WWW, Database]. *Available*: Available from ScienceDirect Database; ADDRESS: http://www.sciencedirect.com/.

[25] Gerkšič, S. *et al.* August 2006. "Advanced Control Algorithms Embedded in a Programmable Logic Controller." *Control Engineering Practice*, Vol. 14(8), pp. 935-948. [Internet, WWW, Database]. *Available*: Available from ScienceDirect Database; ADDRESS: http://www.sciencedirect.com/.

[26] Batorski, J. January 1996. "Wastewater Plant Automates Facilities with PCs, MMI and PLC Control." *Water Engineering & Management*, Vol. 143(1), pp. 18-20. [Internet, WWW, Database]. *Available:* Available in the Proquest SciTech Collection database; ADDRESS: http://www.proquest.com/.

[27] Canzano, Ken [Inventor], Michael A. Del Vecchio [Inventor], Samuel Frisch [Inventor], Douglas Watt [Inventor], and Todd Webster [Inventor]. 4 December

2012. "Methods for Treatment of Perchlorate Contaminated Water." United States Patent 8323496B2. [Internet, WWW]. *Available:* Available from Google Patents; ADDRESS: https://patents.google.com/patent/US8323496B2.

[28] Wenping, D. [Inventor], S. Qiang [Inventor], L. Qiang [Inventor], G. Xiaodong [Inventor], and C. Huadong [Inventor]. 3 December 2014. "Interlock-Control Modular Fenton Reagent Wastewater Treatment Method and Device." Chinese Patent CN103601317B. [Internet, WWW]. *Available:* Available from Google Patents; ADDRESS: https://patents.google.com/patent/CN103601317B/en.

[29] Kun, Y. [Inventor]. 20 March 2013. "Automatic Regulating Device of Paint Slag Wastewater in Painting Workshop." Chinese Patent CN202808461U. [Internet, WWW]. *Available:* Available from Google Patents; ADDRESS: https://patents.google.com/patent/CN202808461U/en.

[30] Yinghua, W. [Inventor], W. Ping [Inventor], S. Hong [Inventor], and L. Liang [Inventor]. 28 April 2004. "Intermittent Automatic Process and Equipment for Treating Waste Water." Chinese Patent CN1147436C. [Internet, WWW]. *Available:* Available from Google Patents; ADDRESS: https://patents.google.com/patent/CN1147436C/en.

[31] Jun, H. [Inventor]. 19 June 2013. "Automatic Control pH Neutralization Device." Chinese Patent CN203012536U. [Internet, WWW]. *Available:* Available from Google Patents; ADDRESS: https://patents.google.com/patent/CN203012536U/en.

[32] Ran, L. [Inventor], S. Baosheng [Inventor], and L. Cheng [Inventor]. 7 April 2012. "Comprehensive Electroplating Wastewater Treatment Method." Chinese Patent CN102531296A. [Internet, WWW]. *Available:* Available from Google Patents; ADDRESS: https://patents.google.com/patent/CN102531296A/en.

[33] Ng. [Inventor]. 22 September 2009. "Electroplating Reclaimed Water Recycling and Treating System." Chinese Patent CN101838071A. [Internet, WWW]. *Available:* Available from Google Patents; ADDRESS: https://patents.google.com/patent/CN101838071A/en.

[34] Boren, R.M. [Inventor] and C.F. Hammel [Inventor]. 11 March 2010. "System for Removal of Metals from Aqueous Solutions." United States Patent 20100059428A1. [Internet, WWW]. *Available:* Available from Google Patents; ADDRESS: https://patents.google.com/patent/US20100059428A1.

[35] Ronik, Wolf von. December 1997. "Cutting to the Chase with Genetic Algorithms." *Futures*, Vol. 26(14), pp. 54-58. [Internet, WWW, Database]. *Available:* Available from the ABI/Inform Collection Database; ADDRESS: http://www.proquest.com/.

[36] Walbridge, Charles T. January 1989. "Genetic Algorithms: What Computers Can Learn from Darwin." *Technology Review*, Vol. 92(1), pp. 46-53. [Internet, WWW, Database]. *Available:* Available from the Proquest SciTech Database; ADDRESS: http://www.proquest.com/.

[37] Wiener, Norbert. 1958. *Nonlinear Problems in Random Theory*. Cambridge, Massachusetts: The M.I.T. Press.

[38] Altinten, Ayla. 2007. “Generalized Predictive Control Applied to a pH Neutralization Process.” *Computers and Chemical Engineering*, Vol. 31, pp. 1199-1204. [Internet, WWW, Database]. *Available*: Available from ScienceDirect Database; ADDRESS: http://www.sciencedirect.com/.

[39] Valarmathi, K., D. Devaraj, and T.K. Radhakrishnan. 2007. “Adaptive Enhanced Genetic Algorithm-Based Proportional Integral Controller Tuning for pH Process.” *Instrumentation Science and Technology*, Vol. 35, pp. 619-635. [Internet, WWW, Database]. *Available*: Available from ScienceDirect Database; ADDRESS: http://www.sciencedirect.com/.

[40] Mwembeshi, M.M., C.A. Kent, and S. Salhi. 2004. “A Genetic Algorithm Based Approach to Intelligent Modelling and Control of pH in Reactors.” *Computers and Chemical Engineering*, Vol. 28, pp. 1743-1757. [Internet, WWW, Database]. *Available*: Available from ScienceDirect Database; ADDRESS: http://www.sciencedirect.com/.

[41] Meier, Frederick A. 2002. “A P&ID Standard: What, Why, How?” *ISA Transactions*, Vol. 41, pp. 389-394. [Internet, WWW, Database]. *Available:* Available from the ScienceDirect Database; ADDRESS: http://www.sciencedirect.com/.

[42] “P&ID is the Plant Floor of Record.” January 2001. *InTech*, Vol. 48(1), pp. 73-74. [Internet, WWW, Database]. *Available:* Available from the Proquest Database Service; ADDRESS: http://www.proquest.com/.

[43] National Fire Protection Association. 2008. *National Electrical Code®.* NFPA Code Number 70. [Internet, WWW]. *Available:* Free read-only access available from the NFPA website; ADDRESS: http://www.nfpa.org/codes-and-standards/all-codes-and-standards/list-of-codes-and-standards?mode=code&code=70.

[44] Morriss, Brian S. 2000. *Programmable Logic Controllers.* Upper Saddle River, New Jersey: Prentice Hall.

[45] Harrison, Howard L. and John G. Bollinger. 1969. *Introduction to Automatic Controls*, Second Edition. New York: International Textbook Company.

[46] Allen-Bradley and Rockwell Automation. November 2015. *Reference Manual: Micro800 Programmable Controllers General Instructions*. [Internet, WWW, PDF]. *Available:* Available from the Rockwell Automation Literature Library; ADDRESS: http://literature.rockwellautomation.com/idc/groups/literature/documents/rm/2080-rm001_-en-e.pdf.

[47] Allen-Bradley and Rockwell Automation. June 2000. *Pulse Width Modulated (PWM) Drives: AC Drives Using PWM Techniques*. [Internet, WWW, PDF]. *Available:* Available from the Rockwell Automation Literature Library; ADDRESS: http://literature.rockwellautomation.com/idc/groups/literature/documents/wp/drives-wp002_-en-e.pdf.

[48] Diesel Service & Supply. 1999-2016. “Converting Generator Frequencies to Supply 60Hz and 50Hz Power.” [Internet, WWW]. *Available:* Available from the Diesel Service & Supply website; ADDRESS: http:// www.dieselserviceandsupply.com/Generator_Frequency_Conversion.aspx

[49] Shamsuzzoha, Mohammed and Sigurd Skogestad. December 2010. “The Setpoint Overshoot Method: A Simple and Fast Closed-Loop Approach for PID Tuning.” *Journal of Process Control*, Vol. 20(10), pp. 1220-1234. [Internet, WWW, Database]. *Available*: Available from ScienceDirect Database; ADDRESS: http://www.sciencedirect.com/.

[50] Leva, A. and A.M. Colombo. 2001. “Implementation of a Robust PID Autotuner in a Control Design Environment.” *Transaction of the Institute of Measurement & Control*, Vol. 23(1), pp. 1-20. [Internet, WWW, Database]. *Available*: Available from ScienceDirect Database; ADDRESS: http://www.sciencedirect.com/.

Appendix A:

Project Process and Instrumentation Design (P&ID) Drawings

Appendix A contains the process and instrumentation design (P&ID) drawings associated with the project that is the focus of this capstone project report.

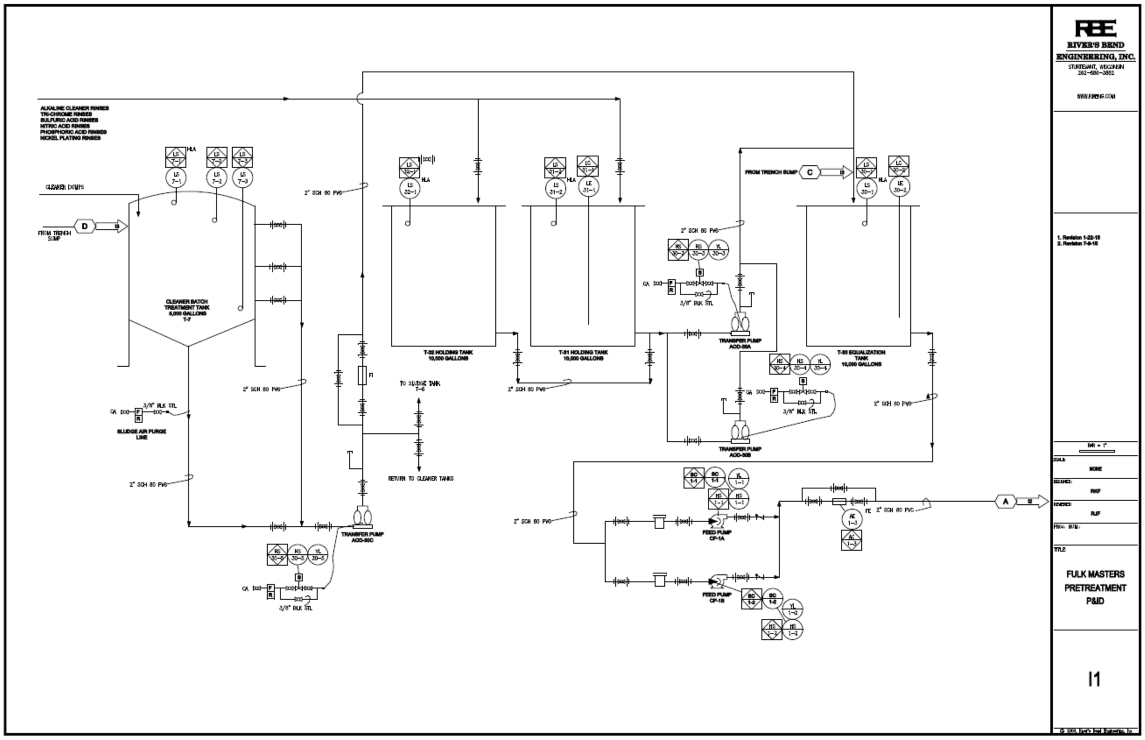

Figure A-1: Page I1 of the Project P&ID.

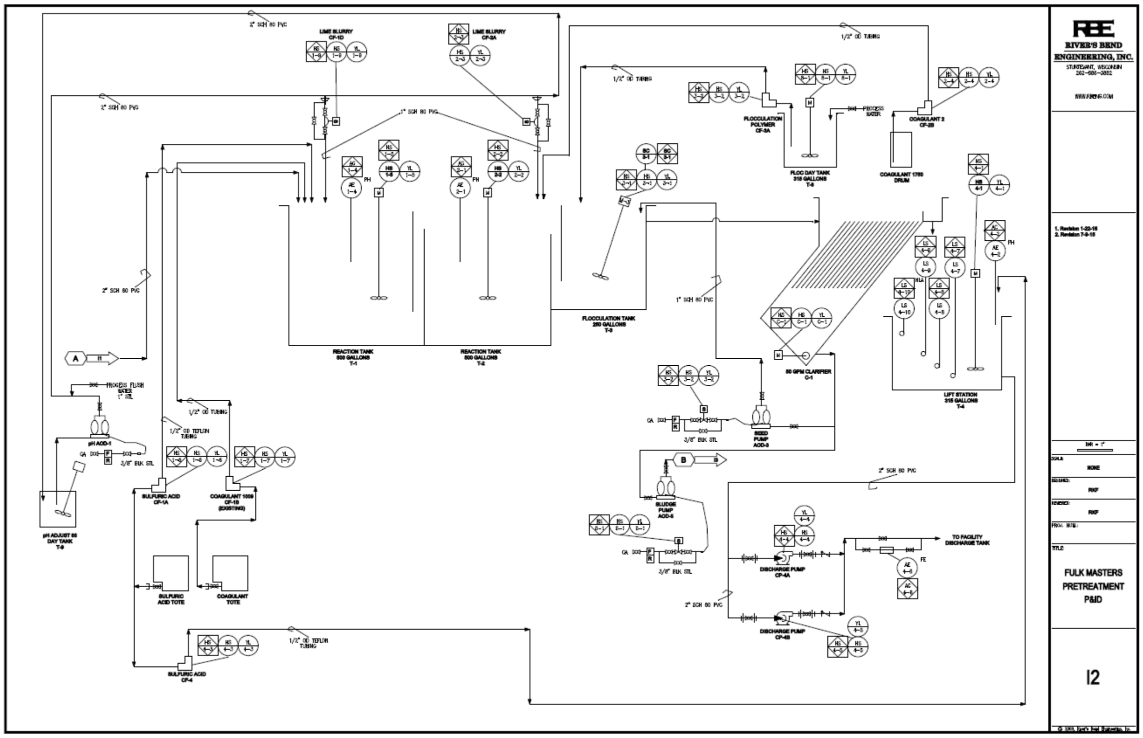

Figure A-2: Page I2 of the Project P&ID.

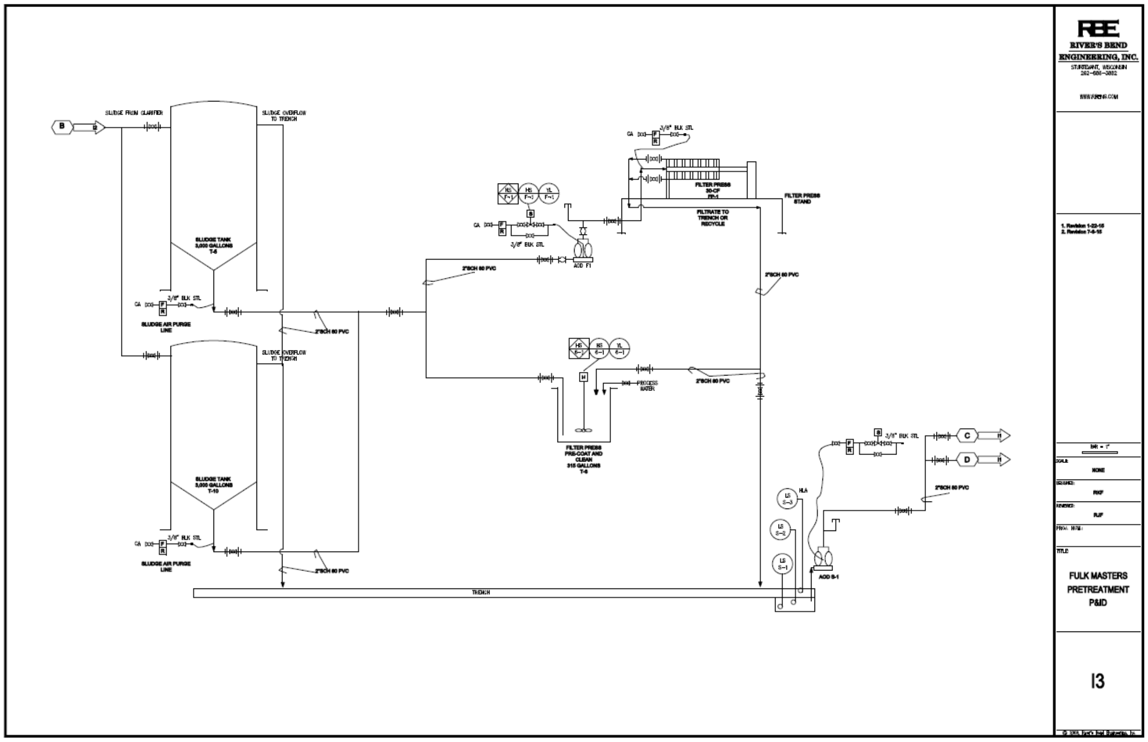

Figure A-3: Page I3 of the Project P&ID.

Appendix B:

Project Electrical Schematics

Appendix B contains the electrical schematic drawings associated with the project that is the focus of this capstone project report.

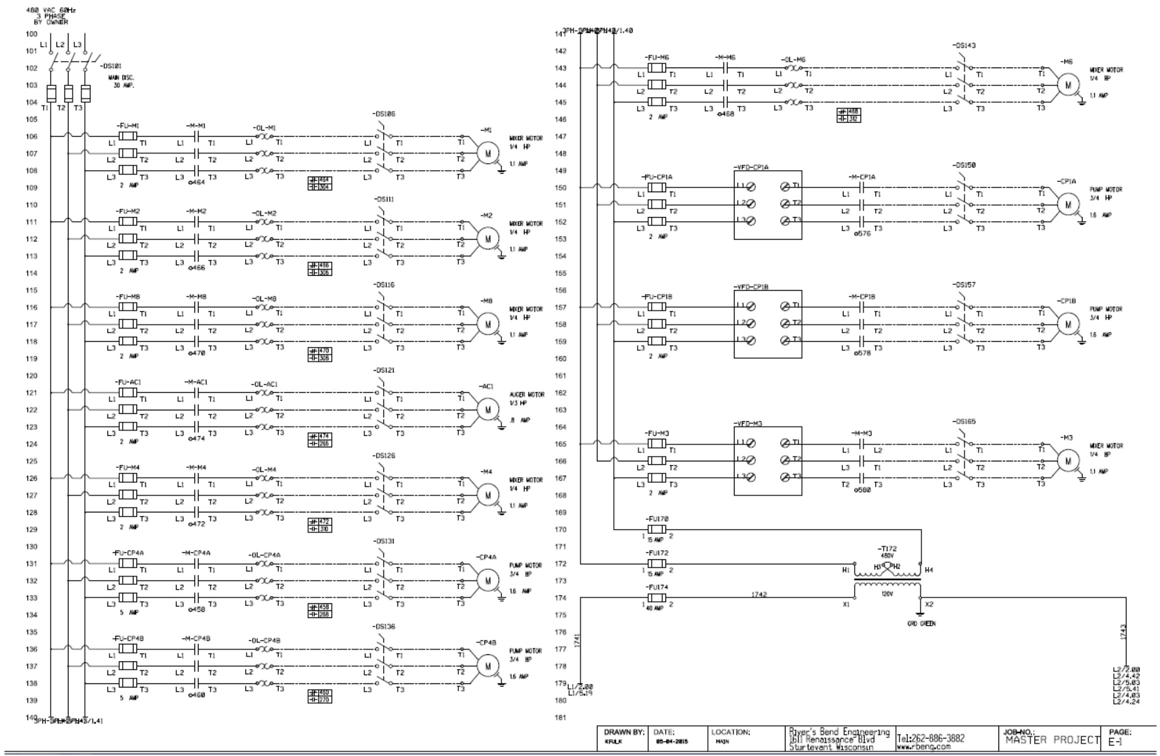

Figure B-1: Project Electrical Schematic E1.

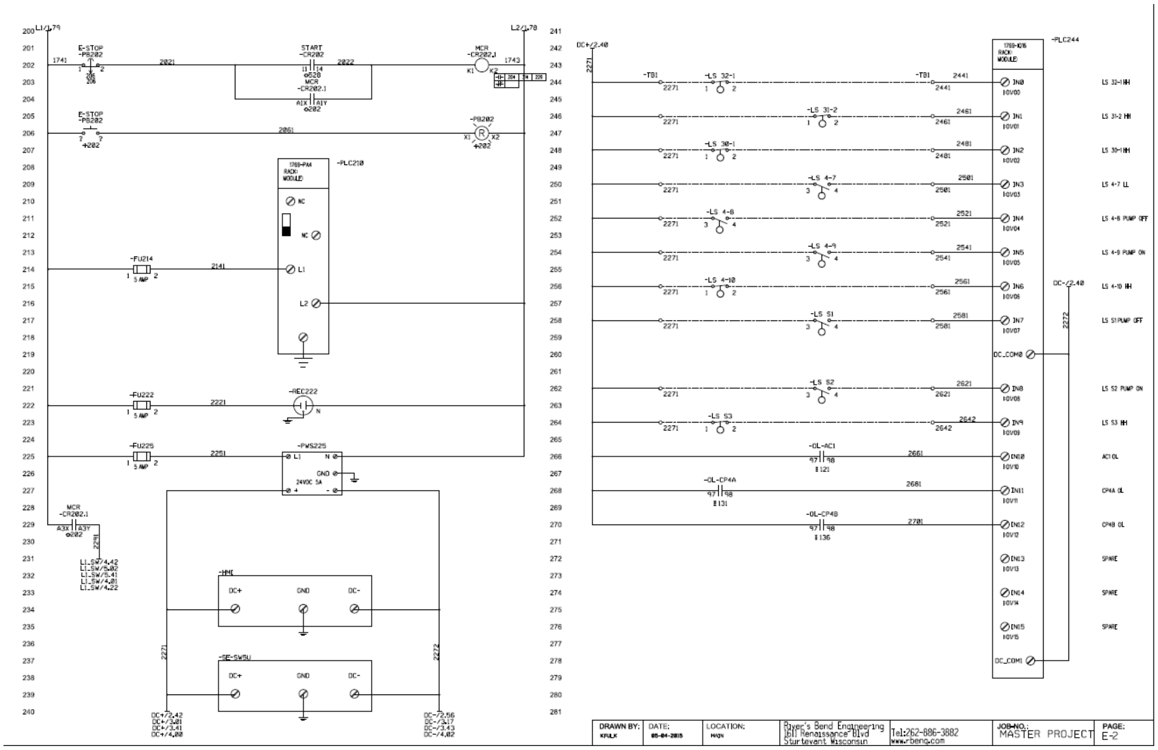

Figure B-2: Project Electrical Schematic E2.

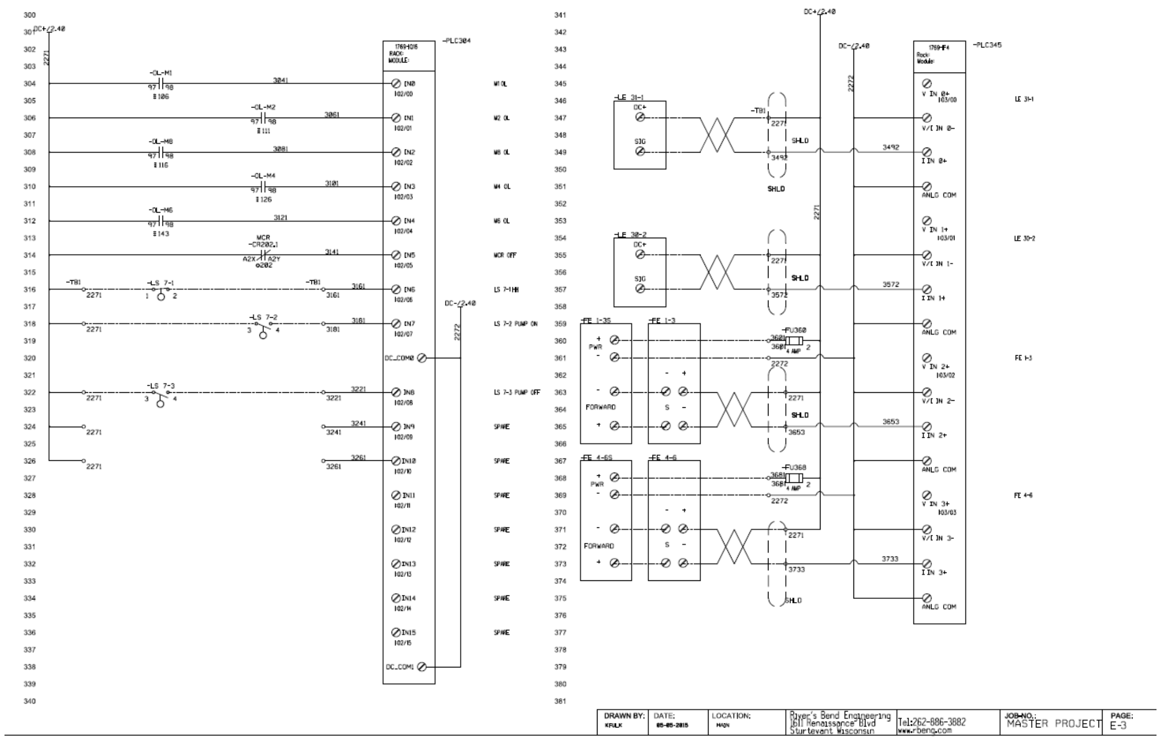

Figure B-3: Project Electrical Schematic E3.

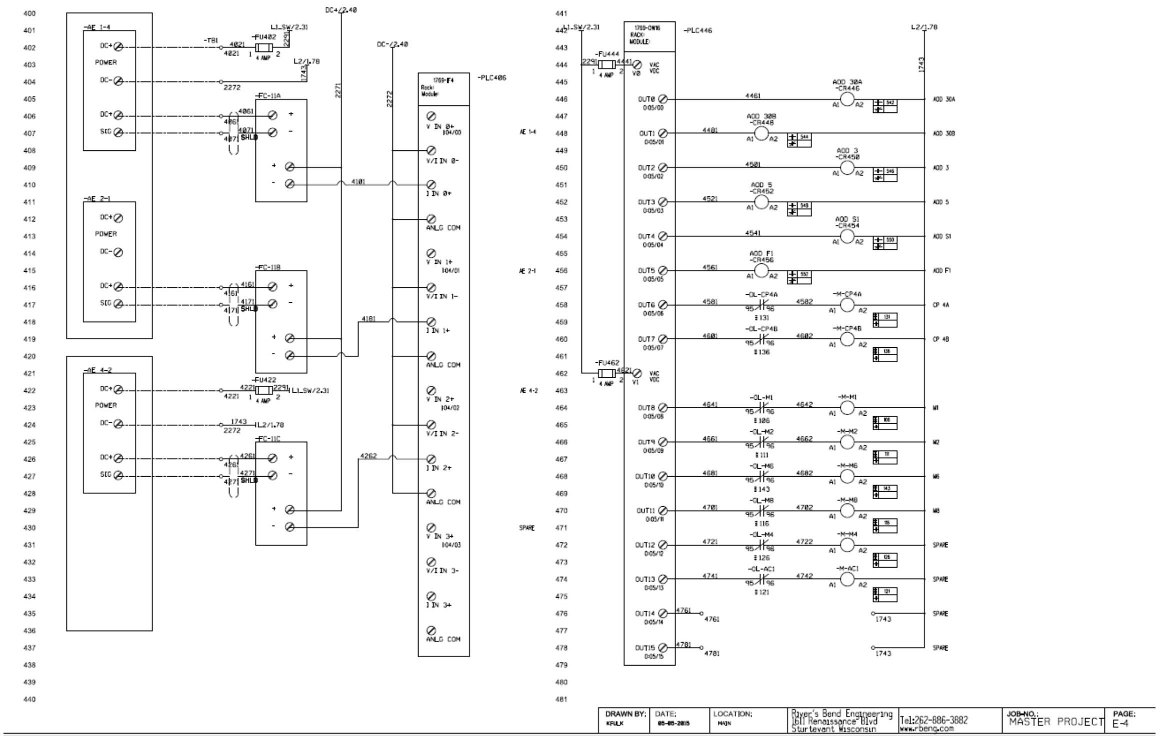

Figure B-4: Project Electrical Schematic E4.

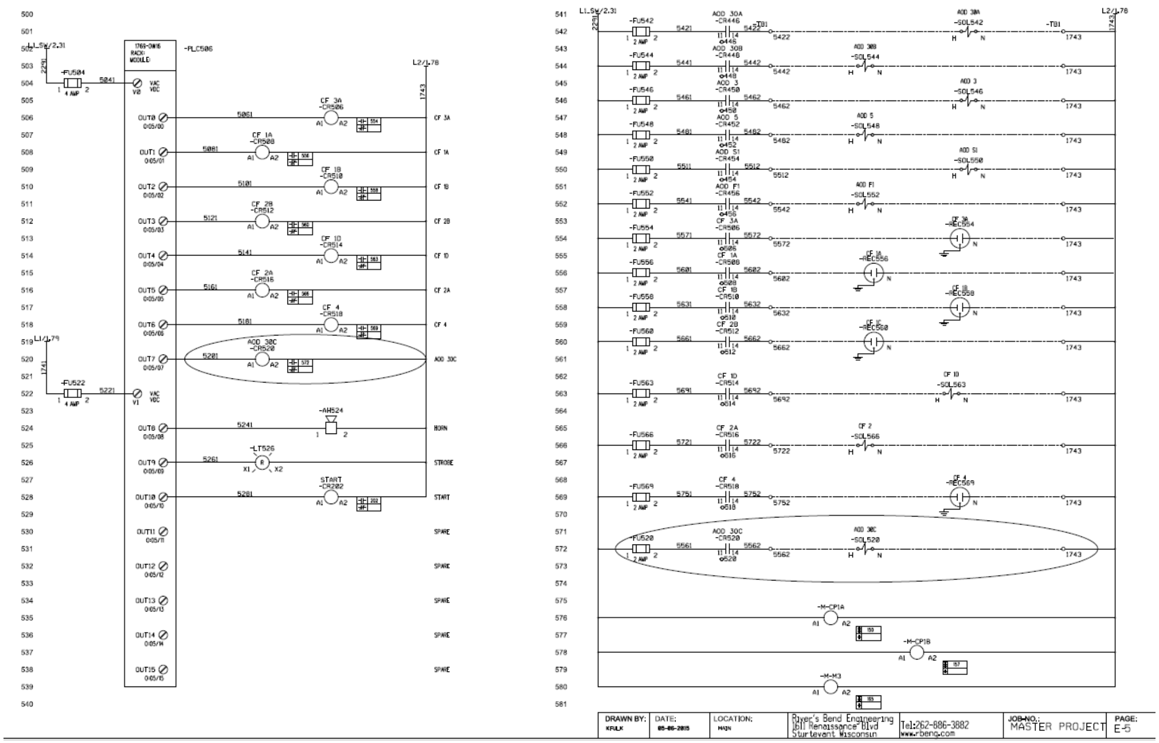

Figure B-5: Project Electrical Schematic E5.

Appendix C:

Project Control System Human Machine Interface (HMI) Screens

Appendix C features screenshots of operator touchscreens associated with the wastewater treatment control system that is the focus of this capstone project report. Screenshots are included for the top menu, as well as for chemical process control, feed process control, filter press control, final pH adjustment control, clarifier and sludge control, transfer process control, and treatment process control.

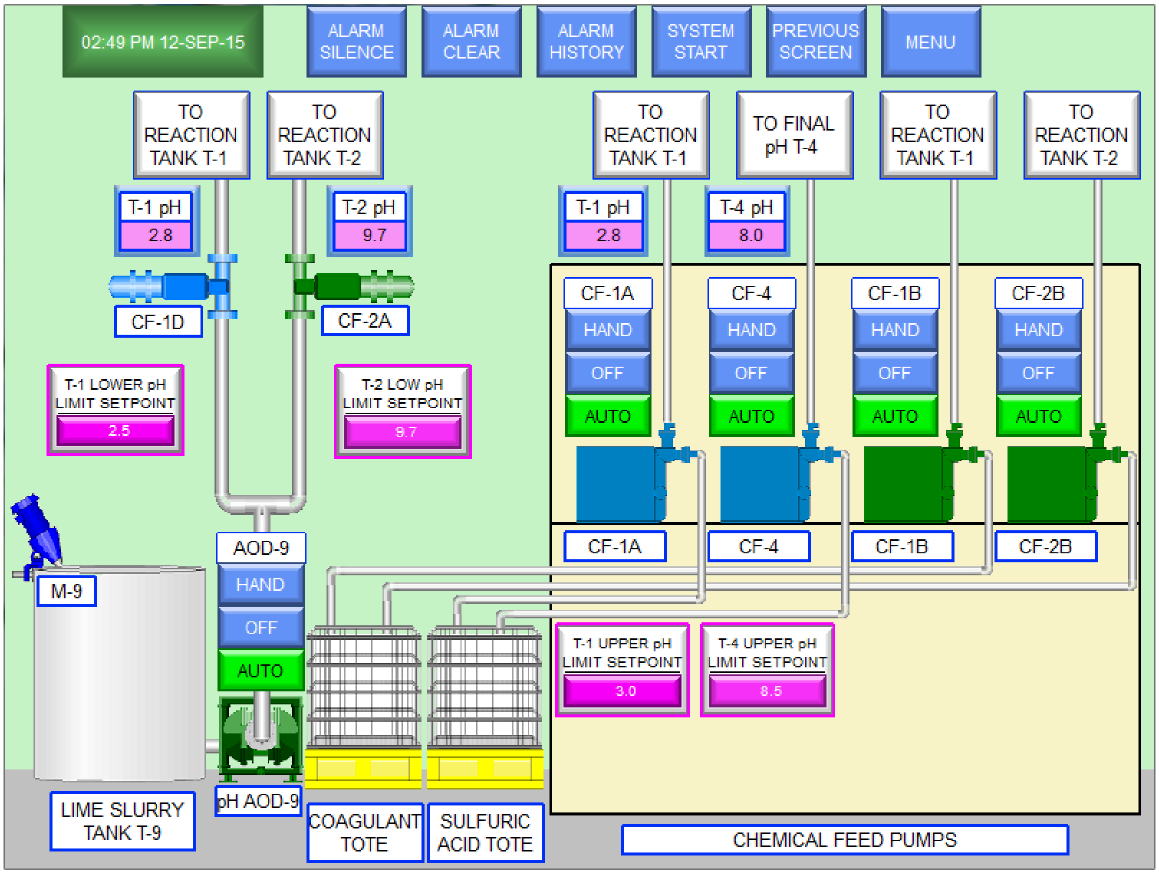

Figure C-1: Chemical Process Control System HMI Screen.

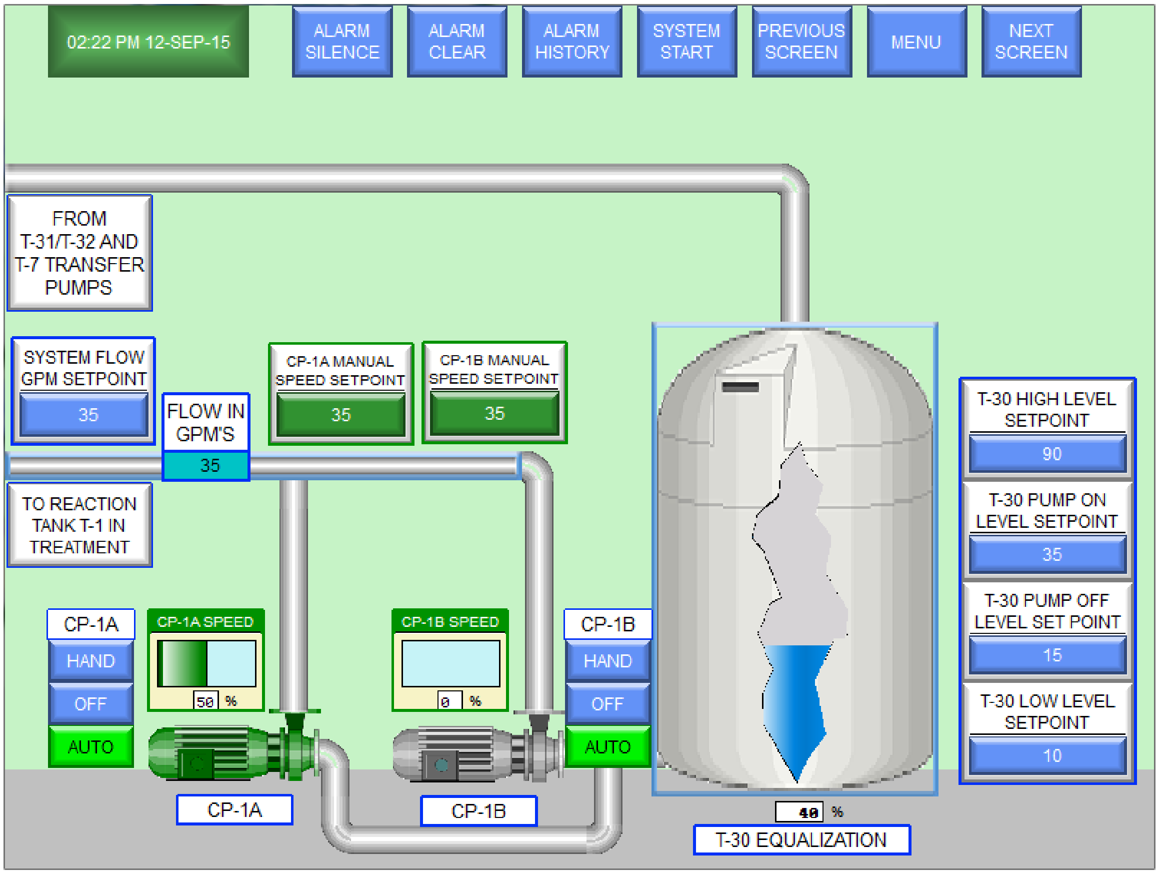

Figure C-2: Feed Process Control System HMI Screen.

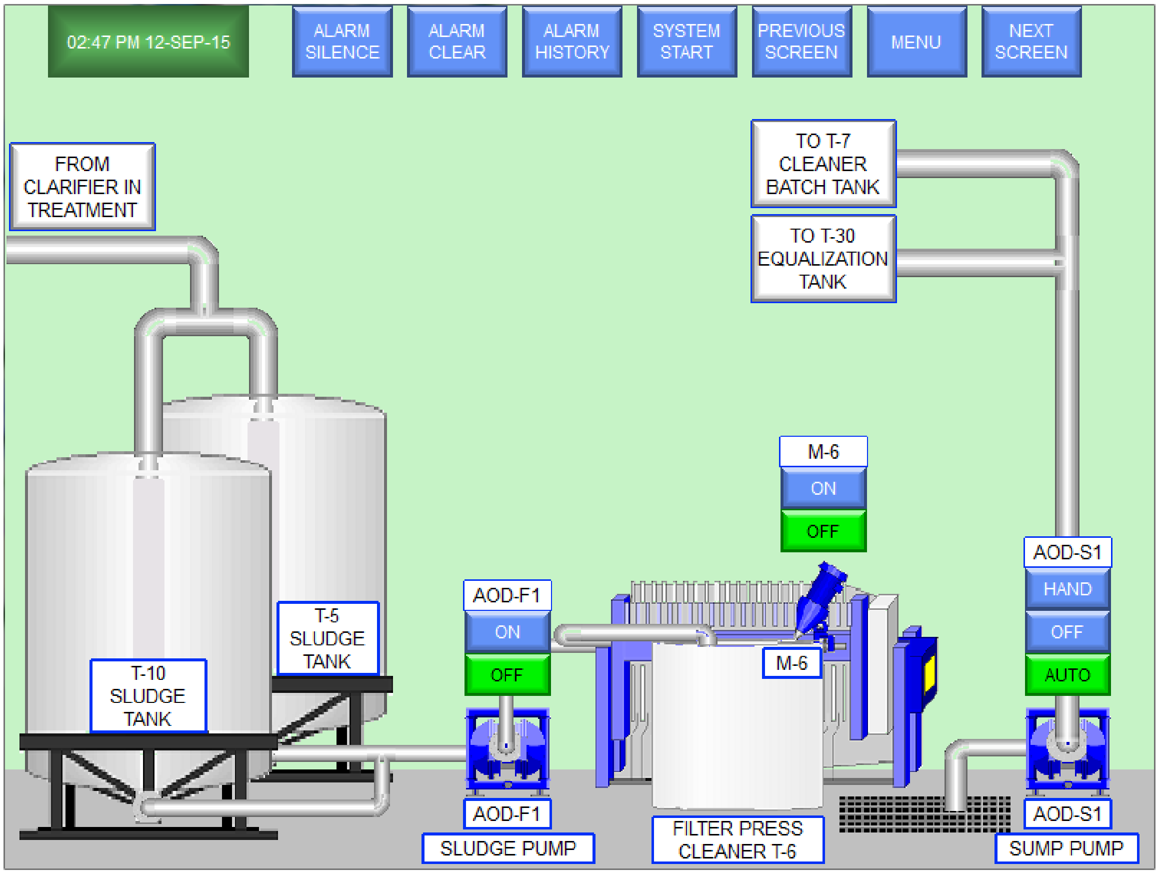

Figure C-3: Filter Press Process Control System HMI Screen.

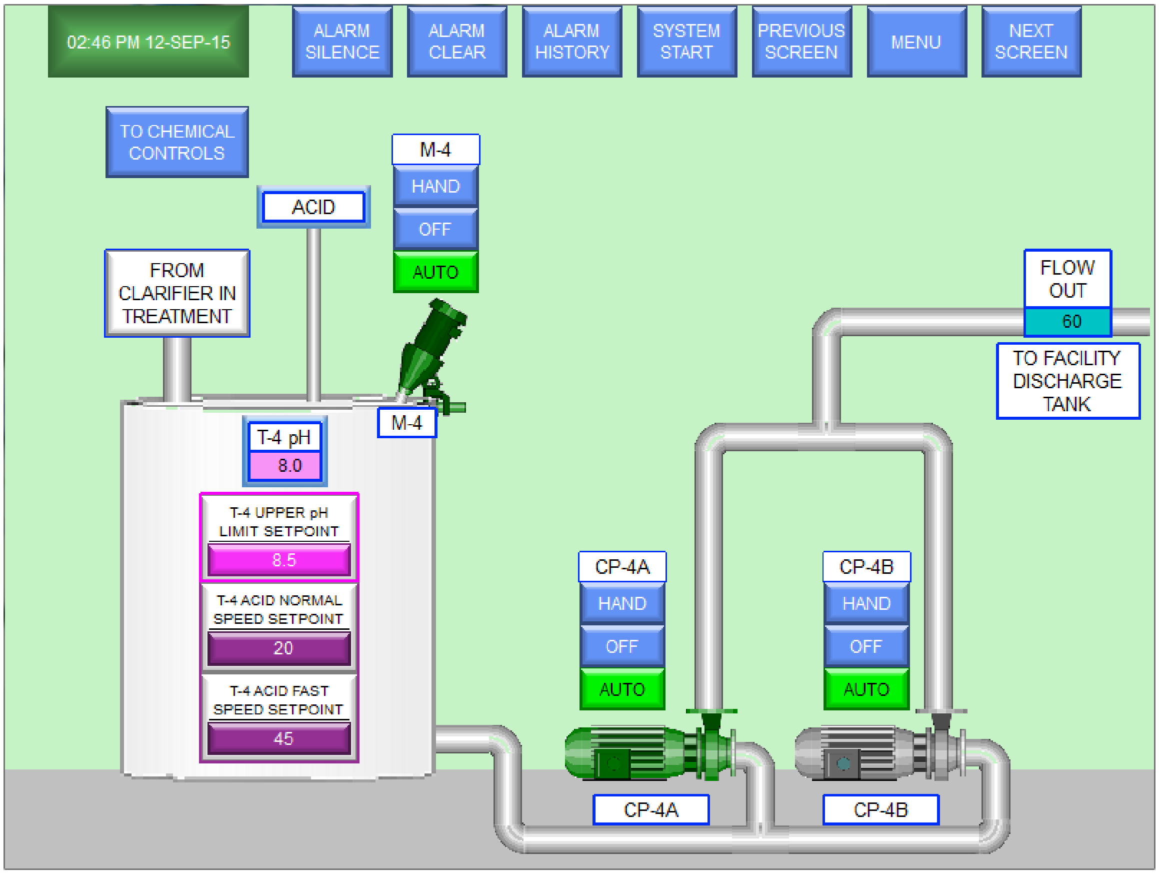

Figure C-4: Final pH Adjustment Process Control System HMI Screen.

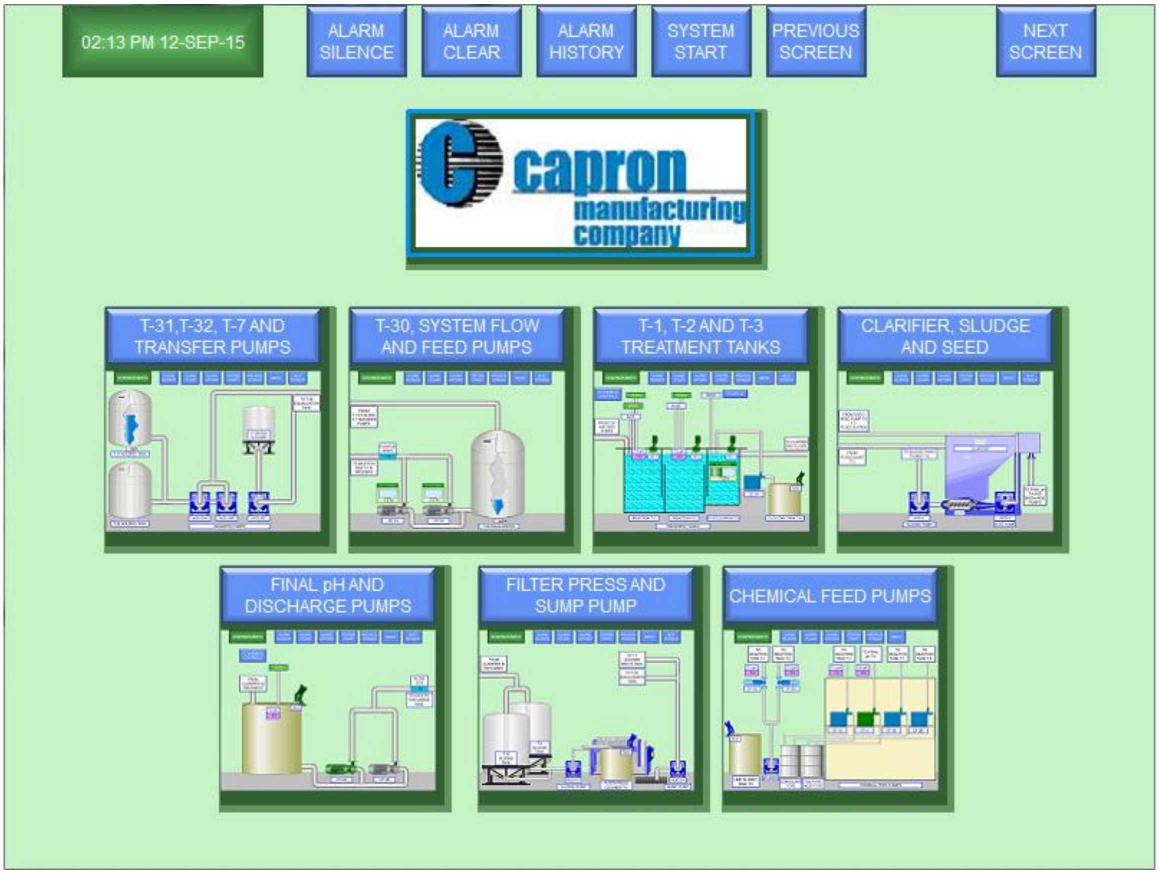

Figure C-5: Top-Level Menu Control System HMI Screen.

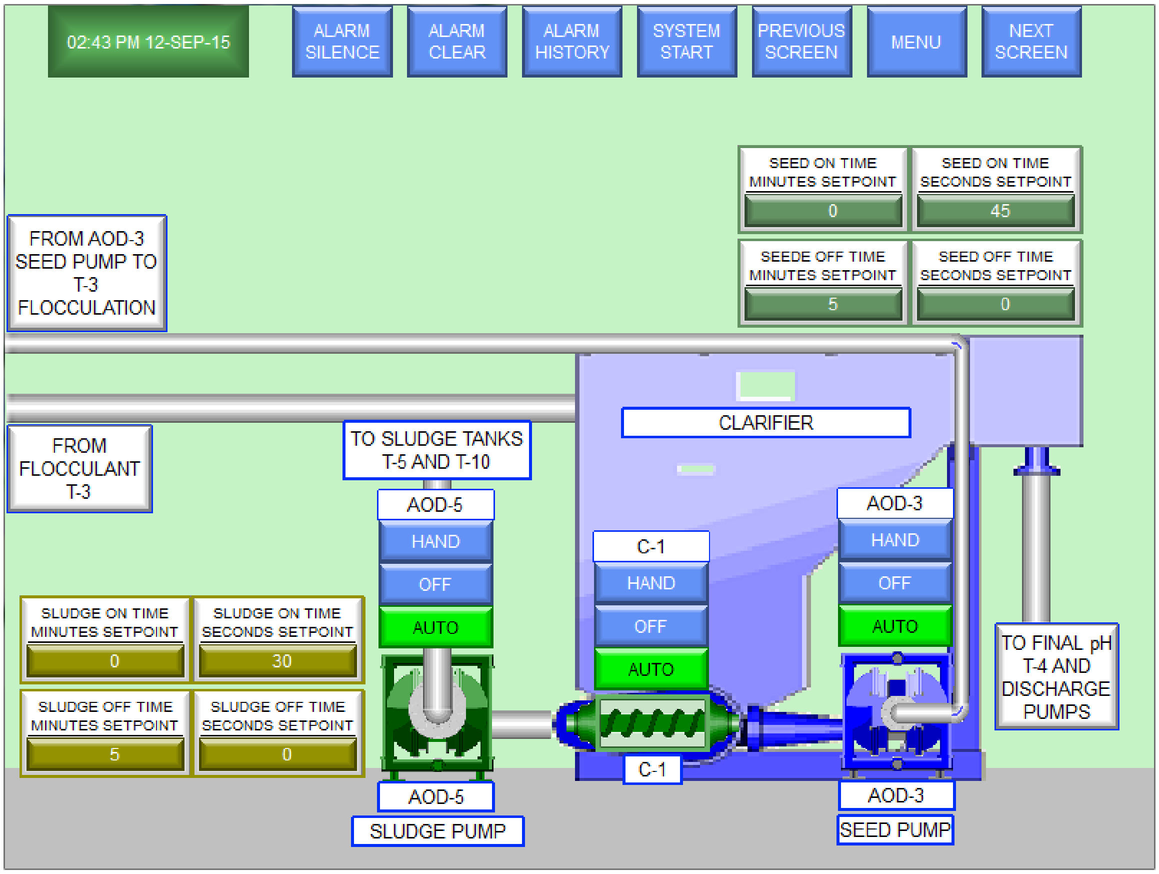

Figure C-6: Clarifier and Sludge Process Control System HMI Screen.

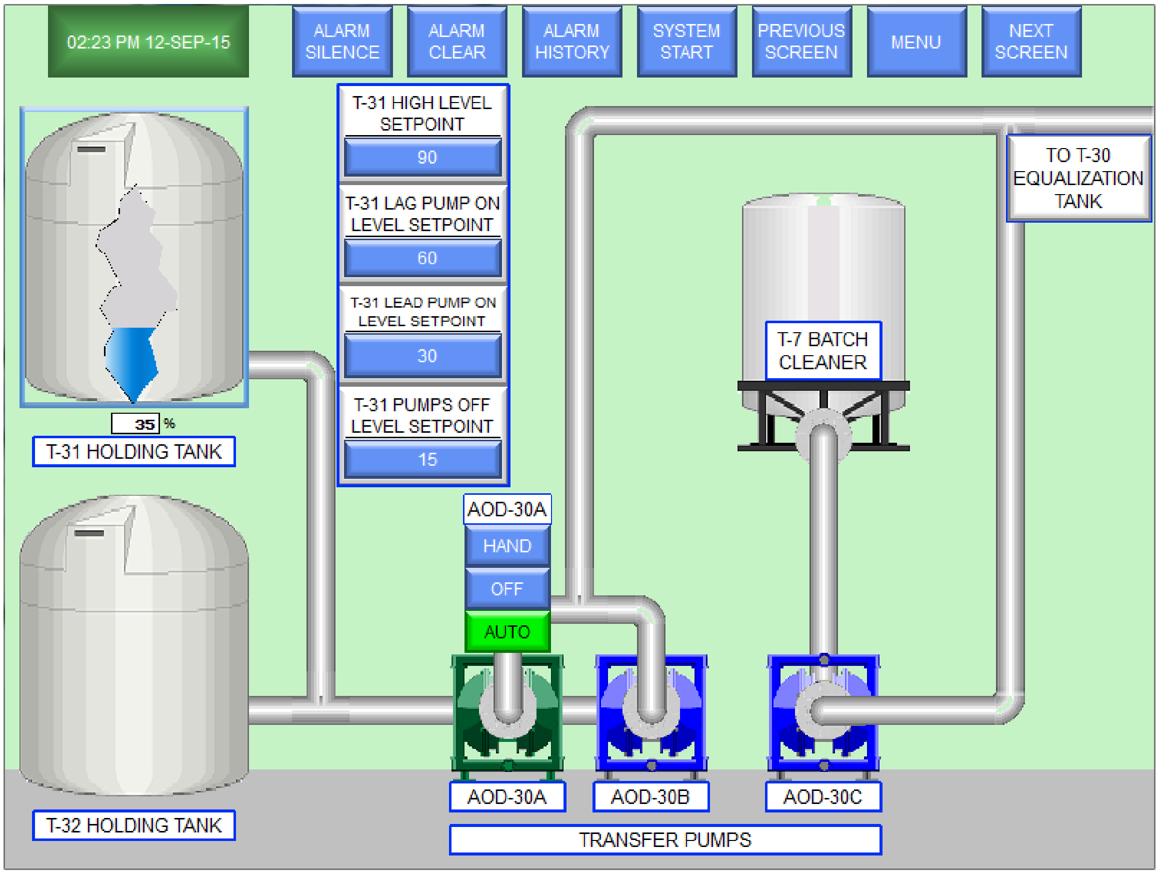

Figure C-7: Transfer Process Control System HMI Screen.

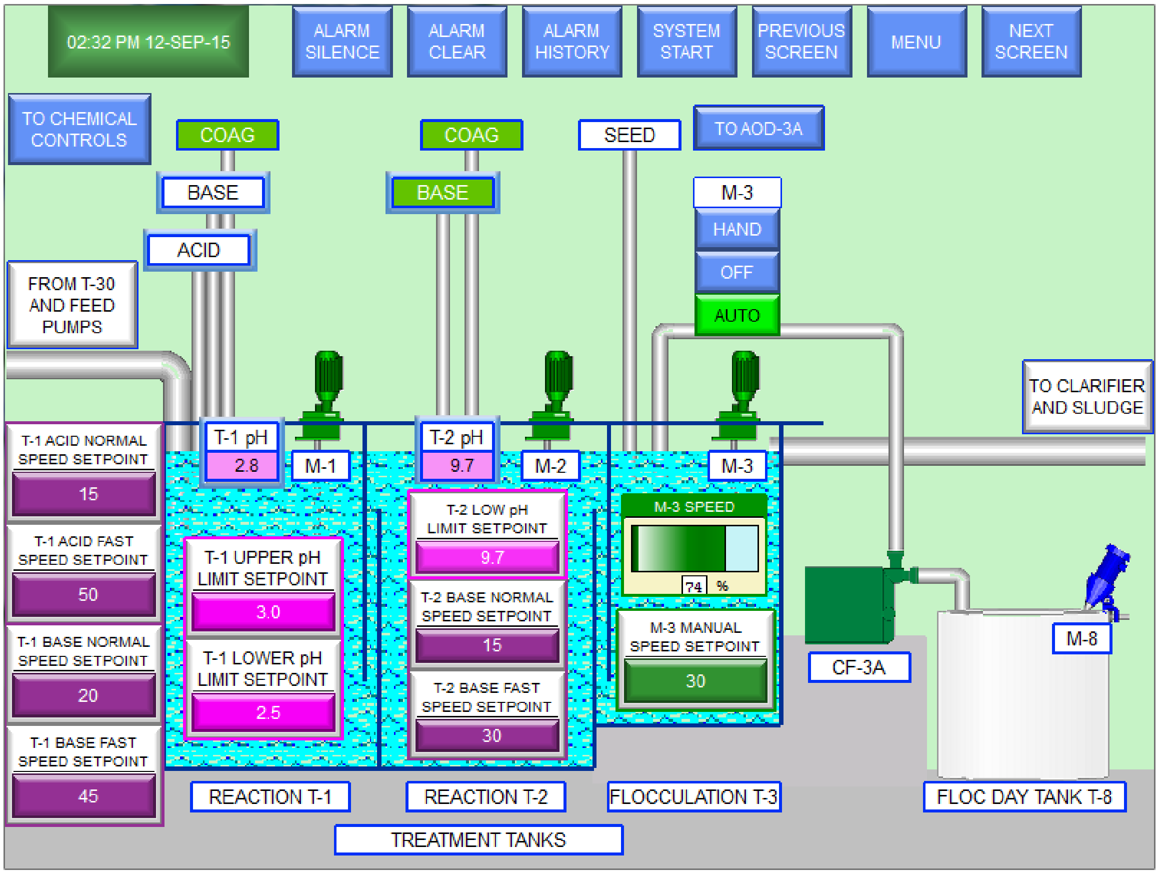

Figure C-8: Treatment Process Control System HMI Screen.

Engineering

Capstone Report Approval Form

Master of Science in Engineering – MSE

Milwaukee School of Engineering

This capstone report, titled "Programming an Automated pH and Flow Control System for Use in an Industrial Wastewater Pre-Treatment System," submitted by the student Kristen Fulk, has been approved by the following committee:

Faculty Advisor: ___________________________ Date: ______________________

Dr. Kishore Acharya, Ph.D.

Faculty Member: ___________________________ Date: _______________________

Dr. Subha Kumpaty, Ph.D.

Faculty Member: ___________________________ Date: _______________________

Professor Gary Shimek, M.L.I.S.