Connectivity and Functionality of the Virtual Cybersecurity Testbed of a Navigational Lock

The University of Alabama in Huntsville

Center for Cybersecurity Research and Education

SueAnne Griffith, Shelton Wright, Thomas H. Morris

31 July, 2019

Introduction

This document outlines the construction and internal workings of the virtual model of a navigational lock based upon the Poe Lock at Sault St. Marie, Michigan. The model was designed by researchers at The University of Alabama in Huntsville (UAH) for the US Army Corps of Engineers (USACE) Engineering Research and Development Center (ERDC). An outline and diagram of the full system can be found in Appendix A.

Overview of Cyber-Physical Systems

Cyber-physical systems (CPSs) can be broken down into five primary components: the physical system, the cyber-physical link connecting this system to a digital control system (DCS), a programmable logic controller (PLC) or other type of DCS, the network between the DCS and the control center, and the human-machine interface (HMI) in the control center [2]. These five components are depicted in Figure 1.

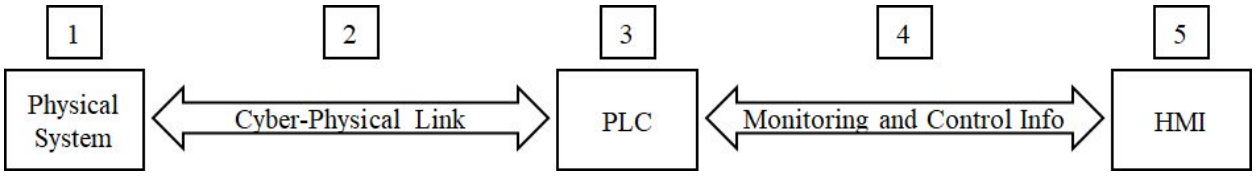

Figure 1: The five basic components of a CPS

The physical system, the component labeled with a 1 in Figure 1, is the portion of the CPS with moving, mechanical parts, generally including actuators such as valves, fans, and hydraulic pumps. Additionally, any sensors used to read data from the mechanical components are considered part of the physical system [3].

The physical system connects to the PLC via the second component cyber-physical link, a wired or wireless connection used to send and receive data. Sensor data from the physical system is relayed to the PLC, and actuator commands are sent from the PLC back to the physical system's devices. PLCs are used to control the physical system, as well as relay information to the human operators. Register values in the PLC are modified and the program logic is executed with the sensor data received from the physical system and the commands received from the human operator.

The communication network between the PLC and HMI in this research uses TCP Modbus protocol on a simulated wired network. Modbus is the most commonly used protocol for PLC communication, due to its simplicity and availability as an open-source protocol without licensing fees [4]. The human operators of the CPS often control and monitor the system(s) under their supervision remotely, rather than being physically near the controlled system at all times. HMIs are used to display data and provide input sources; this is often accomplished through software, with the operator viewing information on a graphical user interface (GUI).

In order to develop a virtual model of a navigational lock, the five basic CPS components from Figure A1 were mapped to the software and techniques used for virtual modeling. These are shown below in Figure 2.

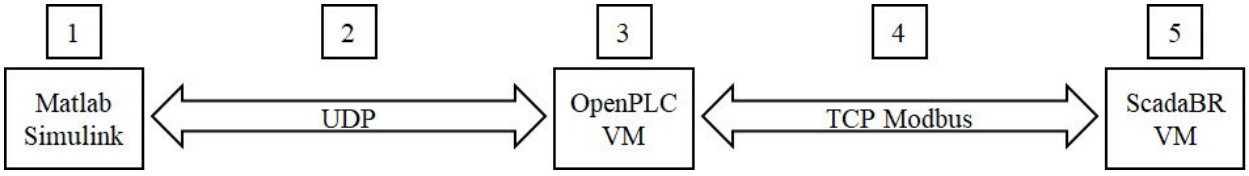

Figure 2: How the five CPS components are modeled in this research

The physical system, labeled with a 1 in Figures 1 and 2, was created in Matlab Simulink on the host computer and is further described in 3.1 below [5, 6]. Next in Figure 1 is the cyber-physical link; it is modeled using user datagram protocol (UDP), as its fire-and-forget property emulates a wire connection. This portion is labeled with a 2 in Figures 1 and 2. To model a PLC, labeled with a 3 in Figure 1, the open-source OpenPLC software was installed on a Linux operating system, with the firmware written in ladder logic in the PLCOpen Editor and uploaded to the PLC using the structured text (ST) format [4]. Monitoring and control information passed between the PLC and HMI in this research's model is sent using TCP Modbus, as this is a common and easy to emulate protocol. Lastly, the HMI component was implemented using ScadaBR, an open-source HMI software. In this model, ScadaBR is run in a virtual machine (VM) and viewed using an internet browser; see component 5 in Figure 2 [7].

Physical System

The physical system is modeled in Matlab Simulink. To aid in the understanding of the terminology used in this document to describe the physical system, Figure 3 has been included. This is a variation of the graphic found in [1].

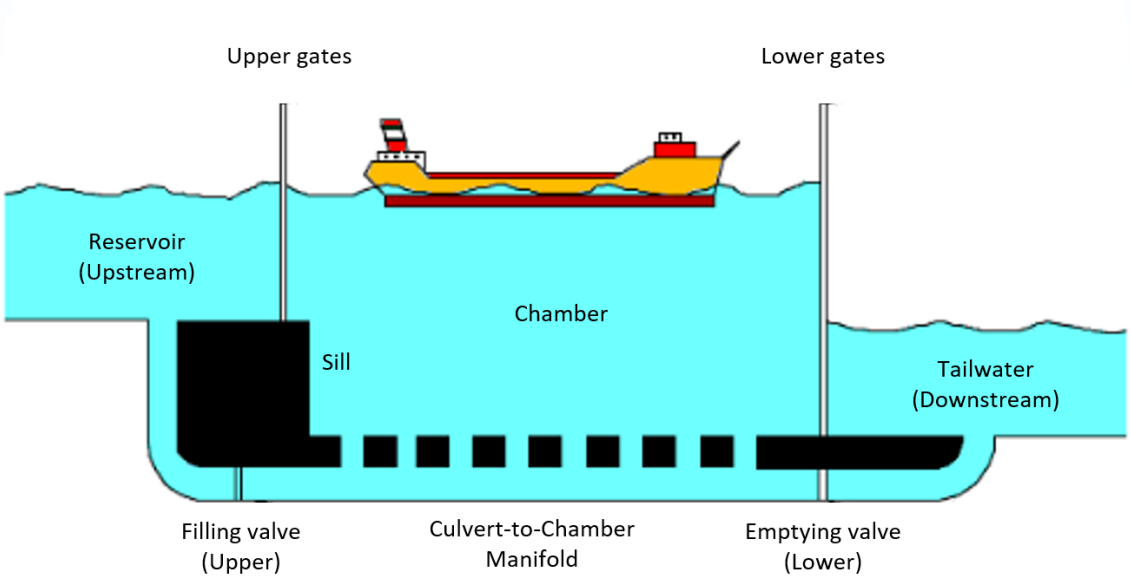

Figure 3:Terminology used to describe lock

The entire model is shown on the next page in Figure 4 just to demonstrate the large scale of the project. There are five function blocks: Upper Gate Position/Movement, Lower Gate Position/Movement, Fill Valve Position/Movement, Empty Valve Position/Movement, and Change in Depth; these are each discussed in their own sections of this document. Additionally, this paper describes the manual emergency controls and the warning horn.

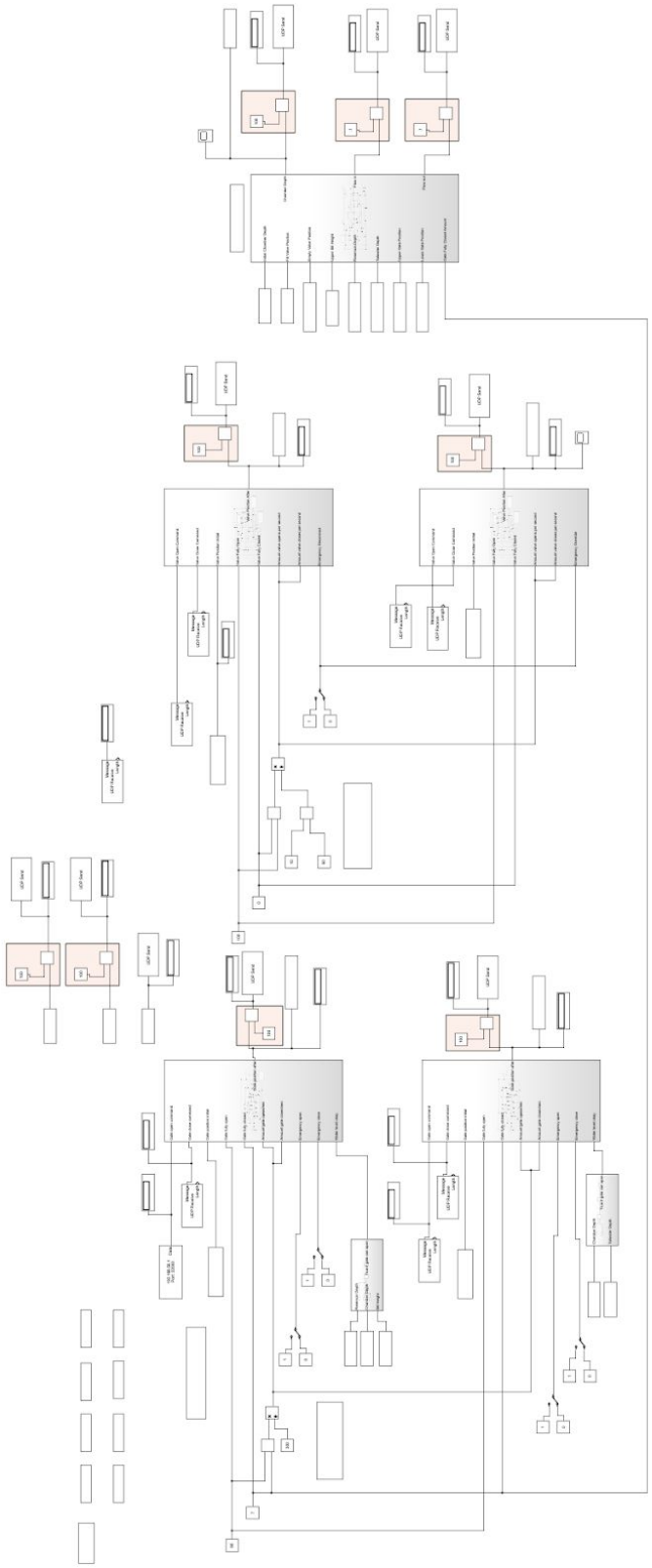

Figure 4: The full Simulink model

Upper Gate

The upper gate is located on the upstream side of the lock and used to allow ships in and out of the chamber from the reservoir side. The upper and lower gate function boxes currently share

the same input values for Gate Fully Open, Gate Fully Closed, Amount Gate Opens per Second, and Amount Gate Closes per Second. The open command is received on port 22000, the close command is received on port 22001, and the gate position (upper_gate_position) is sent out on port 23002.

The function block and connections for this upper gate, entitled Upper Gate Position/Movement, is shown in Figure 5, with the functions contained shown in Figure 6 on the next page. Like many other variables sent from Simulink to the PLC, the upper_gate_position is multiplied by a constant, in this case 100, and later divided by this same constant after it is received by the control code. This was found to be the easiest way to preserve significant figures throughout data type changes in the system.

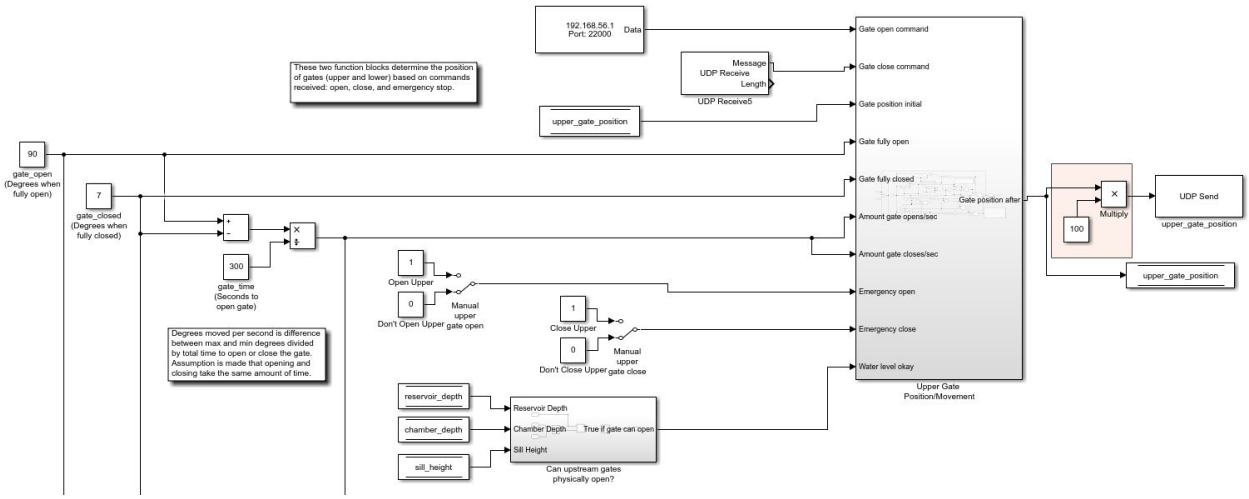

Figure 5: The UpperGate Position/Movement function block

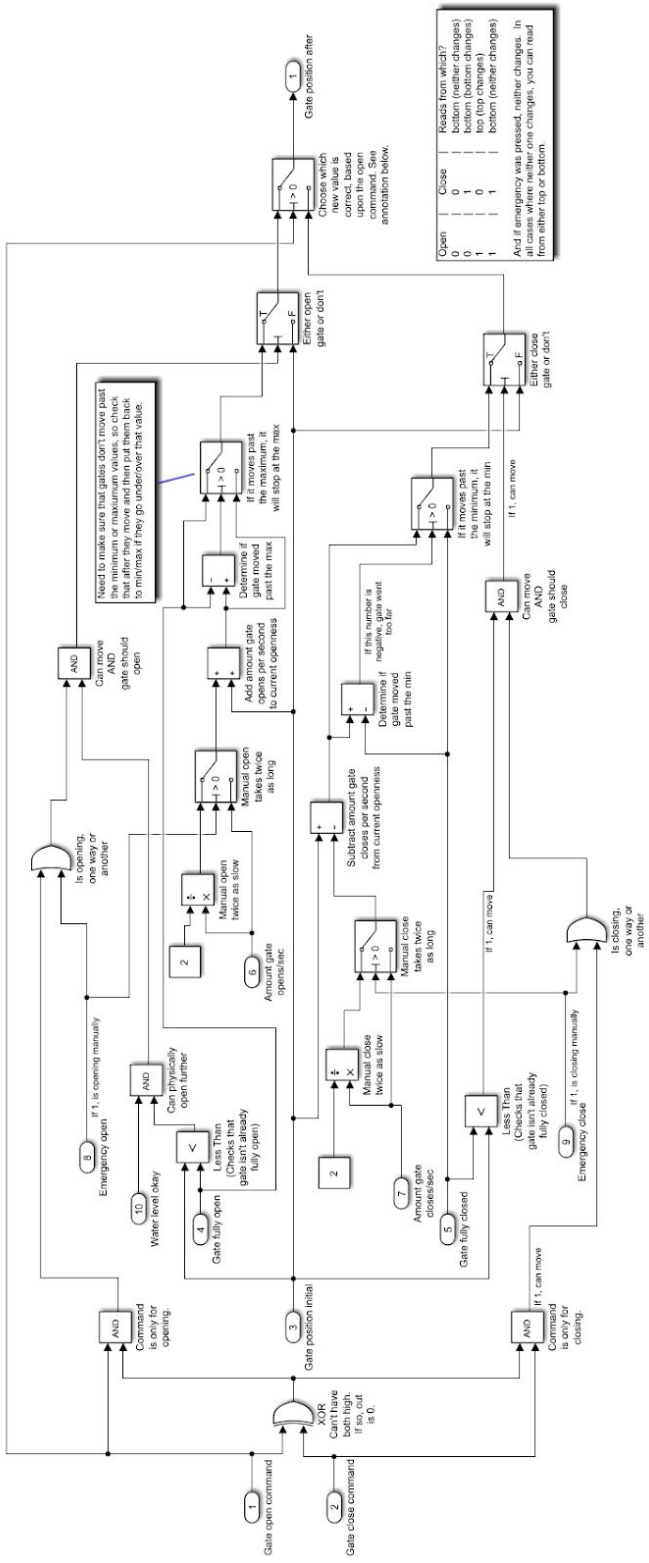

Figure 6: Contents of the Upper Gate Position/Movement function block

The program first checks that the gate motor is not receiving both open and close signals at the same time, as this would result in the motor not moving at all. The program also checks that the gate is not already fully open, in the case of an "open" signal being received, or fully closed, in the case of a "close" signal, as these scenarios would also result in the motor not moving. Additionally, the gates cannot move past these boundaries. The speeds at which the gate opens and closes are based upon values found in USACE studies of the Soo Locks, and those amounts are appropriately added or subtracted to the upper_gate_position value (the "Gate position initial" input) when the gate is opening or closing, respectively.

Lower Gate

The lower gate, which is located on the downstream side of the lock and used to allow ships in and out of the chamber from the tailwater side, is programmed in the same manner as the upper gate. The two gate function boxes currently share the same input values for Gate Fully Open, Gate Fully Closed, Amount Gate Opens per Second, and Amount Gate Closes per Second. The open command is received on port 22002, the close command is received on port 22003, and the gate position (lower_gate_position) is sent out on port 23003.

The function block and connections for this lower gate, entitled Lower Gate Position/Movement, is shown in Figure 7, with the functions contained shown in Figure 8 on the next page.

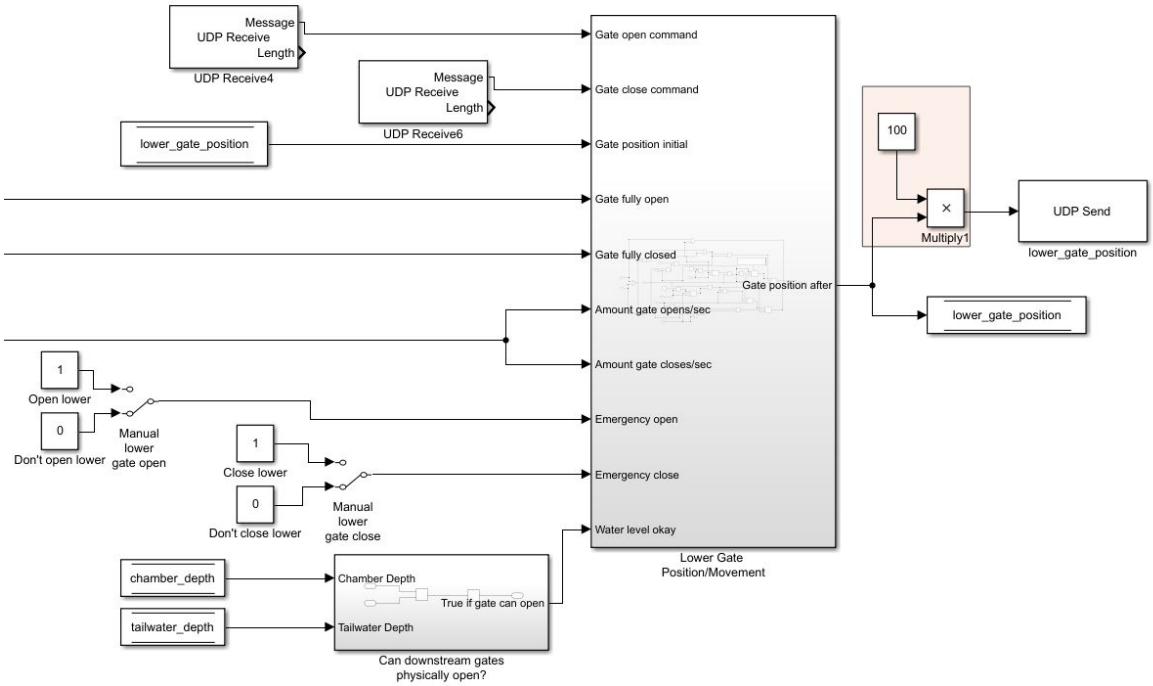

Figure 7: The Lower Gate Position/Movement function block

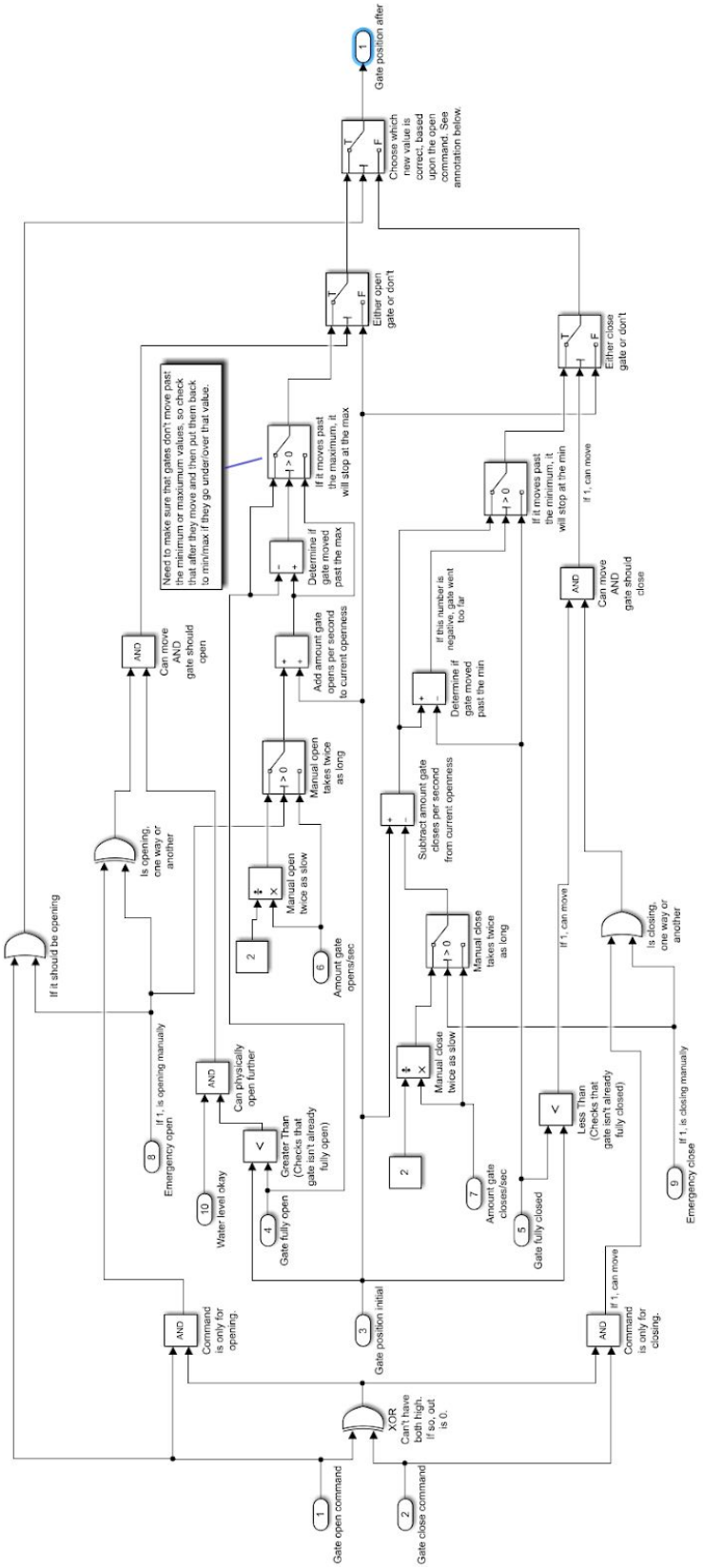

Figure 8: Contents of the Lower Gate Position/Movement function block

Filling Valve

The filling valve, which is located on the upstream side of the lock and used for filling the chamber, contains the math used to determine valve movement and timing. The function block for this upper valve, entitled Fill Valve Position/Movement in the Simulink program, is shown in Figure 9, with the functions contained shown in Figure 10 on the following page.

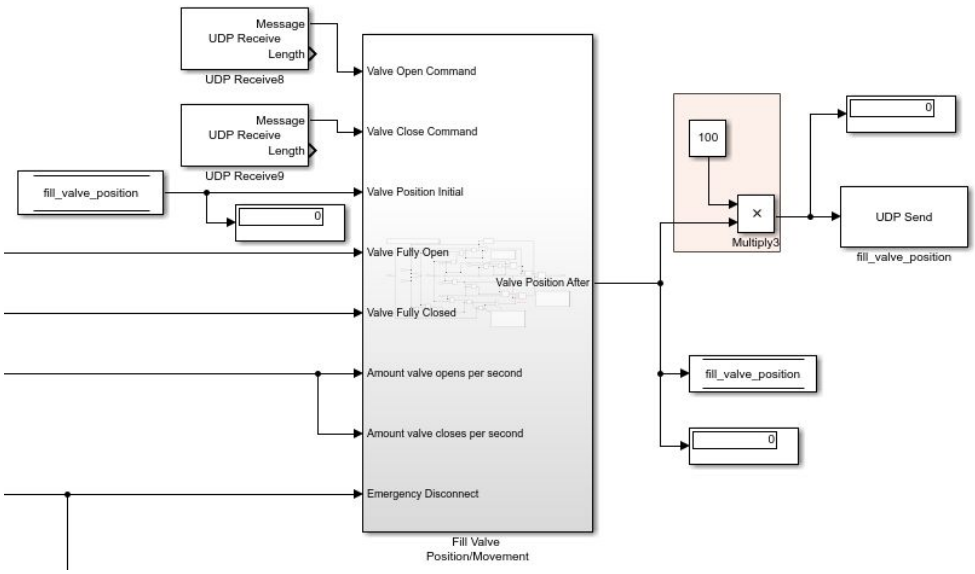

Figure 9:The Fill Valve Position/Movement function block

The program first checks that the motor is not receiving both open and close signals at the same time, as this would result in the motor not moving at all. The program also checks that the valve is not already fully open, in the case of an "open" signal being received, or fully closed, in the case of a "close" signal, as these scenarios would also result in the motor not moving. Additionally, the valves cannot move past these boundaries. The speeds at which the valve opens and closes are based upon values found in USACE studies of the Soo Locks, and those amounts are appropriately added or subtracted to the fill_valve_position value (the "Gate position initial" input) when the valve is opening or closing, respectively.

The emergency disconnect functionality is discussed later in this document, in the Emergency section. The filling valve shares input values with the Emptying Valve for Valve Fully Open, Valve Fully Close, Amount Valve Opens per Second, Amount Valve Closes per Second, and Emergency Override. The open command is received on port 22004, the close command is received on port 22005, and the valve position (empty_valve_position) is sent out on port 23004 after being multiplied by 100.

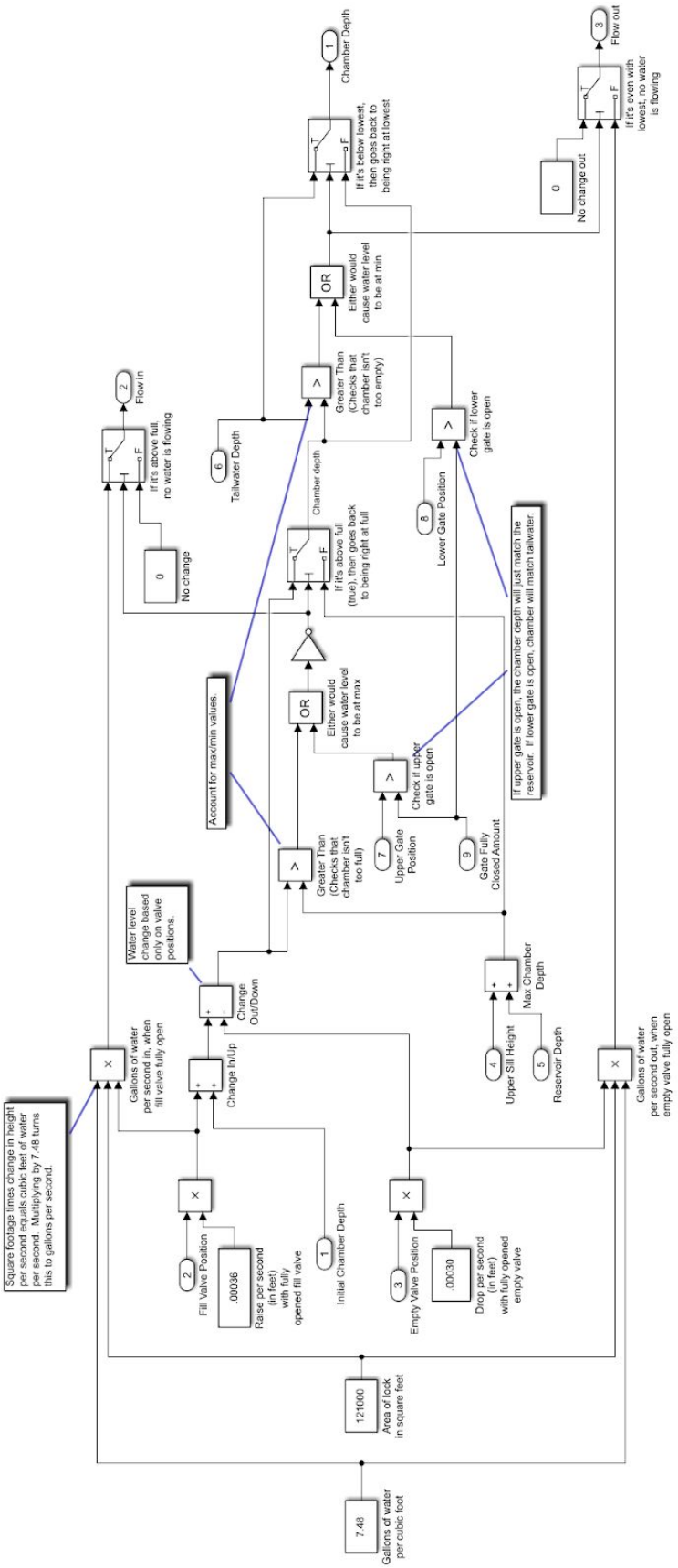

Figure 10:Contents of the Fill Valve Position/Movement function box

Emptying Valve

The emptying valve, which is located on the downstream side of the lock and used for emptying the chamber, is programmed in the same manner as the filling valve. The two valve function boxes currently share the same input values for Valve Fully Open, Valve Fully Close, Amount Valve Opens per Second, Amount Valve Closes per Second, and Emergency Override. The open command is received on port 22006, the close command is received on port 22007, and the valve position (empty_valve_position) is sent out on port 23006.

The function block for this lower valve, entitled Empty Valve Position/Movement, is shown in Figure 11, with the functions contained shown in Figure 12 on the next page.

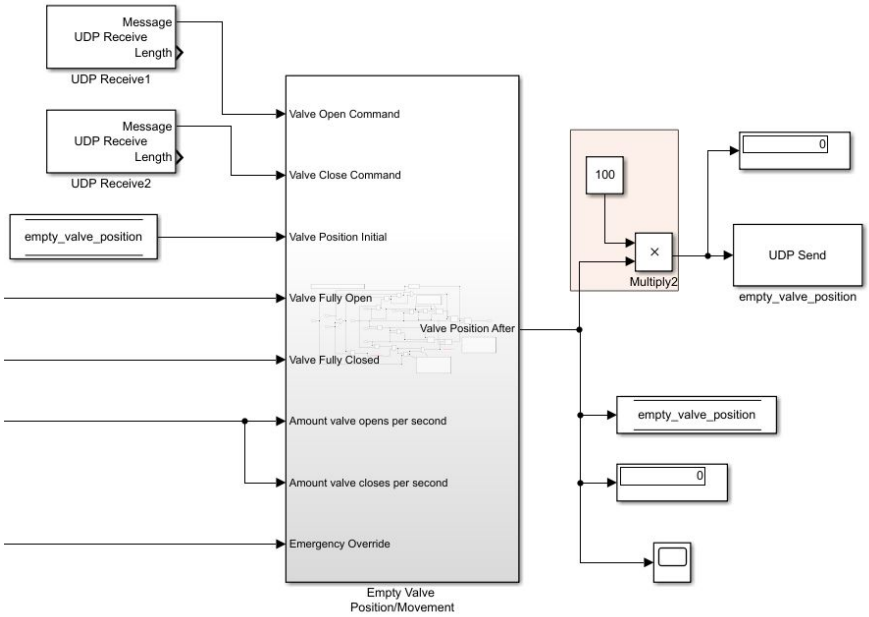

Figure 11: The Empty Valve Position/Movement function block

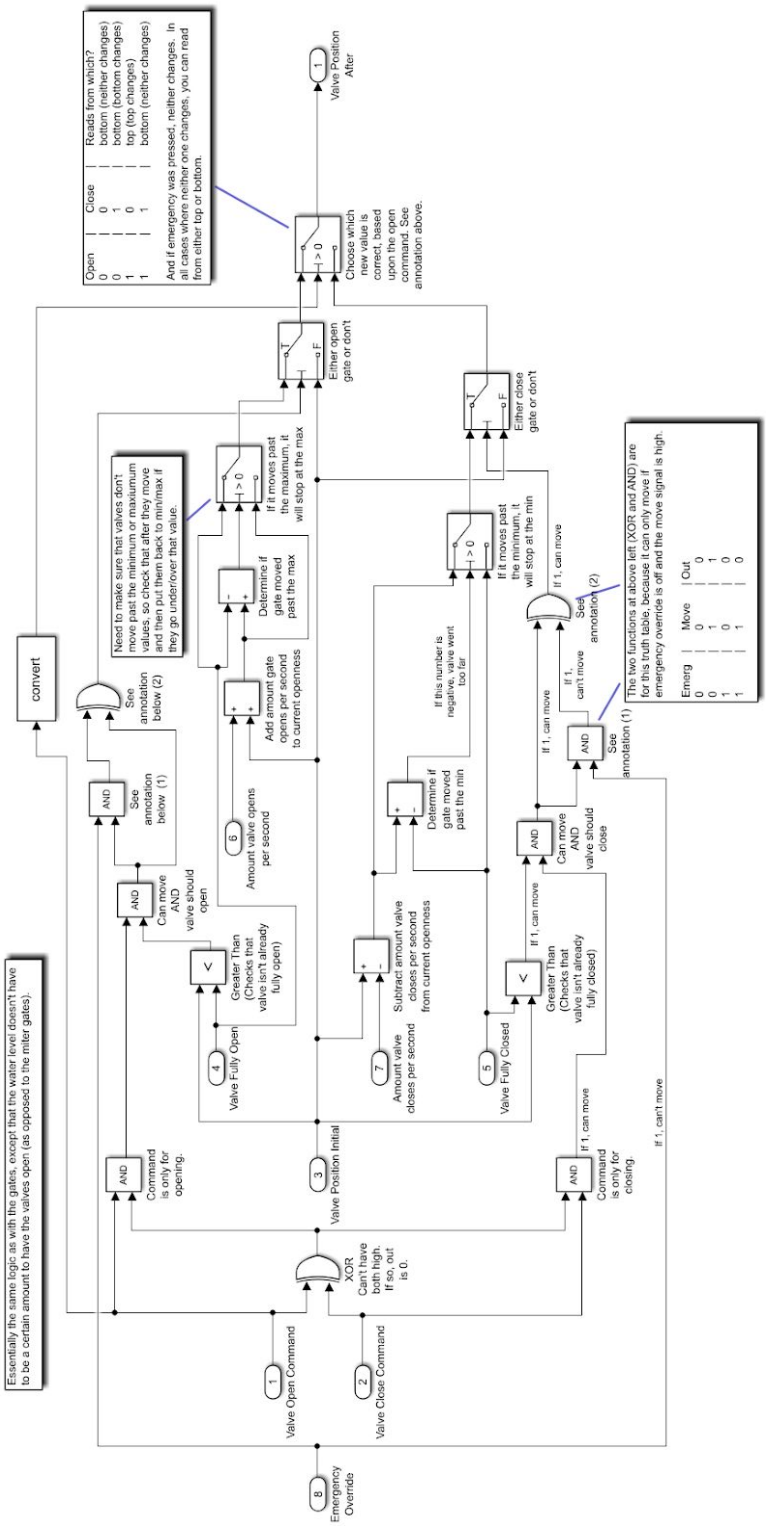

Figure 12: The contents of the Empty Valve Position/Movement block

Chamber

The water depths are calculated in Simulink and sent to the PLC. The reservoir and tailwater depths are currently constants, but these may be modified to vary randomly within a reasonable bounds to emulate flooding and other natural occurrences. At present, the constants that are used in the Simulink model and then relayed to the PLC and HMI are the values for the reservoir depth, tailwater depth, and sill height. The sill height is the difference in height between the surfaces of the tailwater and reservoir; the sill height may also be described as the amount which the chamber depth changes to go from its highest and lowest points.

Figure 13 shows the three aforementioned constants, stored as reservoir_depth, tailwater_depth, and sill_height, connected to UDP Send and Display blocks. These are sent on ports 23007, 23008, and 21000. The sill height value of 21.5 feet was chosen based upon readings on the Poe Lock in the Soo Locks System. Note that these output values are multiplied by constants prior to being sent, just as the aforementioned gate and valve position values.

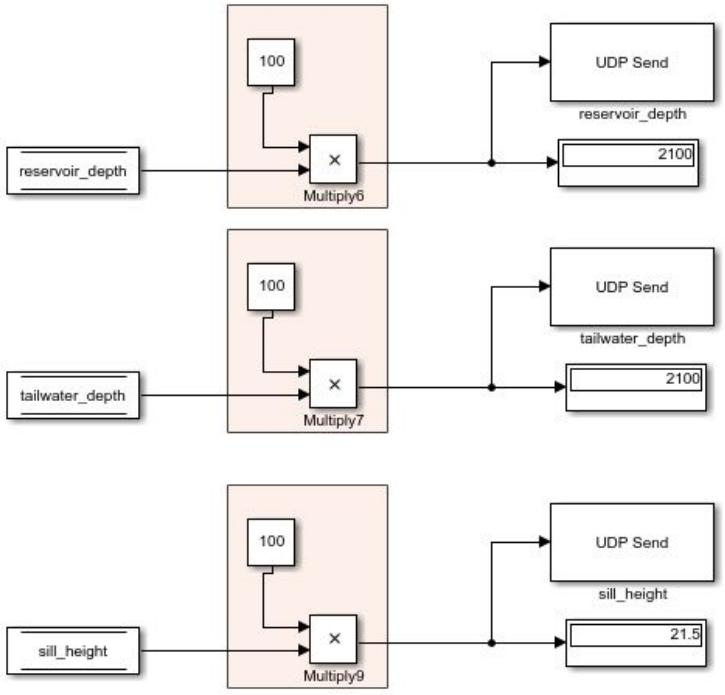

Figure 13: The reservoir depth, tailwater depth, and sill height constants

A function block was created in Simulink to calculate the flow rates in and out of the chamber, as well as the change in chamber depth, based upon the gate and valve positions. The chamber depth is sent to port 23006, with the flow rates in and out of the chamber sent on ports 23000 and 23001, respectively. The function block itself, as well as its inputs and outputs, can be seen in Figure 14. The contents of this Change in Depth block are shown on the next page in Figure 15.

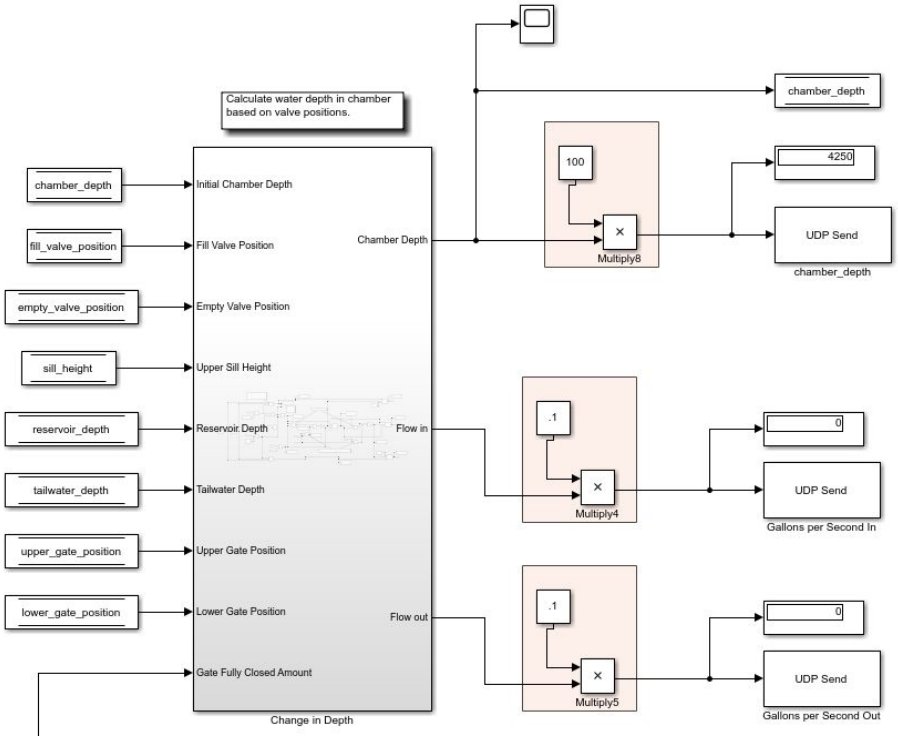

Figure 14:The Change in Depth function block

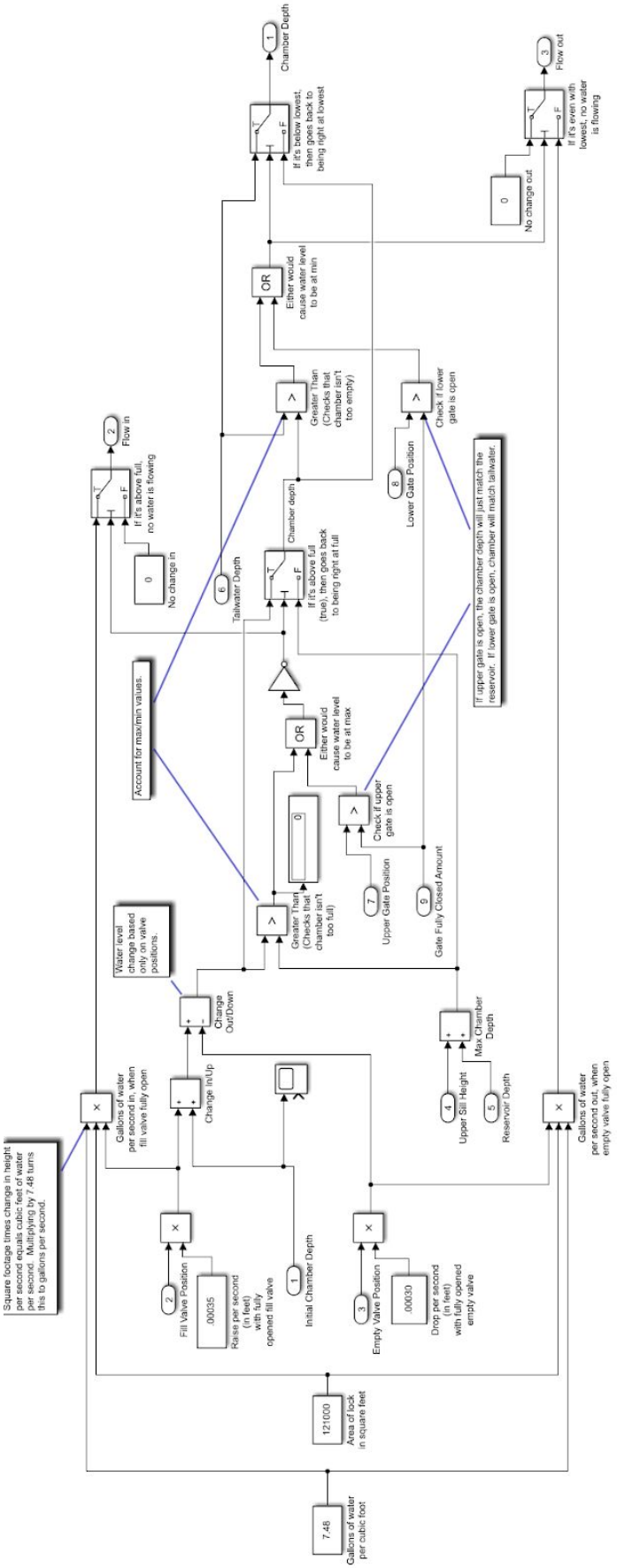

Figure 15: The contents of the Change in Depth block

To calculate the change in chamber depth, the program multiplies a constant raise or drop in feet per second by the percent the valve is open. Currently, these values are .00036 ft/sec for raising and .0003 for lowering. The products of these multiplications are appropriately added and subtracted from the initial chamber depth. If the resulting depth is above the maximum height, where the sill height plus reservoir depth is equal to the chamber depth, the chamber depth is set to exactly the maximum. The same is true if the chamber reaches its minimum value of equivalent to the tailwater depth.

The flow rate is calculated by taking the aforementioned product of the valve openness percentage and the amount the water level rises or falls and then multiplying that by the area of the lock's water surface to get the change in water volume in cubic feet per second. This amount is multiplied by 7.48, the conversion of gallons of water per cubic foot, to get the flow rate in gallons per second. Equation 1 shows this calculation.

$$Flow\ (gal/sec) = \%\ valve\ open * \frac{water\ level\ change\ (ft)}{1\ sec} * lock\ area\ (sq.\ ft) * \frac{7.48\ gal\ of\ water}{1\ cu.\ ft}$$

Equation 1: Calculation of flow in gallons per second through the valves.

When calculating flow rates, this function block must also take into account when the system reaches equilibrium; at this point, no water will be flowing. The system is at equilibrium when the height of the chamber water is equivalent to that of the tailwater or reservoir, depending on whether the locks were filling or emptying. This is true whether the gates are open or closed. Recall that the lower gate cannot open if the chamber and reservoir water levels are equal and that the upper gate cannot open if the chamber and tailwater are equal. For example, if the chamber water height is even with that of the tailwater, no water will flow through the emptying valve, regardless of if the lower gate is open.

Emergency

Manual controls are included within the Simulink model to emulate the physical valves and levels lock operators may use to manually open, close, or stop the valves and gates. These are represented as switches in Simulink, allowing the model's operator to toggle them when the testbed is running. Because physical overrides are not digitally controlled, and therefore not hackable, these toggle switches receive no networked input.

Activating the emergency control for the valves causes the valve to freeze in place; the switch used in the Simulink model is shown in Figure 16. One switch is used for both valves, though this is easily changeable if the user decides to implement separate shutoff overrides for each valve individually.

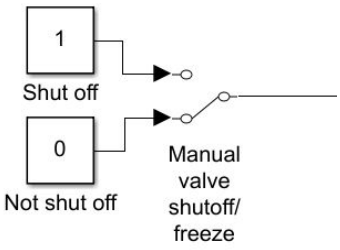

Figure 16: Toggle switch to manually freeze valve movement

Having the valves freeze when the emergency override is triggered means that the valves can only move if the emergency switch is not triggered (and thus outputting a 0) and the move command received digitally is a on (outputting a 1). The truth table for this arrangement is shown in Table 1, with an output of 0 meaning the valve does not move and an output of 1 meaning it does. This logic is produced by performing an AND of the override and move commands, then XORing this result with the move command.

Table 1: Truth table for valve movement

Emergency Override	Digital Move Command	Output
0	0	0
0	1	1
1	0	0
1	1	0

There are also two toggle switches for the gates: one to open and one to close, because in some real-life systems, a large wheel can be turned. The switches for the upper gate are shown in Figure 17; the switches on the lower gate are identical. The output from each switch (either 0 or 1) is ORed with the digital command received to open or close the valve, respectively. This way, if either the digital command or the emergency override are 1 (to move), the gate will move. The speed of the gate is halved when it is moved manually, as a person turning a crank would be less powerful than a motor.

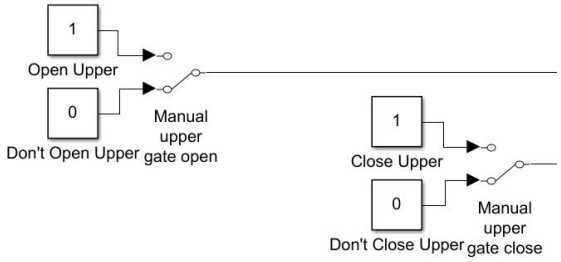

Figure 17: The upper gate manual override switches

Manual Warning

Prior to opening either the filling or emptying valves, the lock operator will activate the warning horn to alert boaters that water will begin flowing. This warning horn is represented in the model by a display block that shows a one or zero based upon if the horn is sounding or not; this is shown in Figure 18. The signal to trigger the horn is received on port 22010.

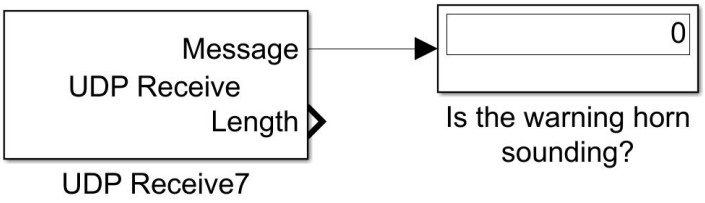

Figure 18: Code for the warning horn

PLC

Modbus addressing is used to store variables in the PLC. The four basic types of variables are digital output, digital input, analog input, and analog output. The digital inputs and outputs function as a binary signal with values of either a 0 or 1. In the ladder logic, digital values can be used as contacts or coils. A contact determines the "electrical connection" of the circuit. If the value is true, then power is allowed to flow through the contact. Coils are used to assign values to digital variables. In the ladder logic diagram, if power is reaching the coil, the bit is assigned 1. If power is not able to flow to the coil, the bit is set to 0.

Analog inputs and outputs are 16 bit registers. These 16 bits allow integer values from 0 to 65,535 to be stored in the registers. These registers can be used for math and comparison operators as well as storing values from sensors. Modbus address types and their properties is seen below in Table 2.

Table 2: Modbus register types

Type	PLC Address	Register Size	Value Range
Digital Outputs	%QX0.0 - %QX99.7	1 bit	0 or 1 / OFF or ON
Digital Inputs	%IX0.0 - %IX99.7	1 bit	0 or 1 / OFF or ON
Analog Inputs	%IW0 - %IW99	16 bits	0 to 65,535
Analog Outputs	%QW0 - %QW99	16 bits	0 to 65,535

Upper Gate Command Check

The Open and Close commands for the upper gate are binary values stored in digital output registers on the PLC. For both commands, a signal of value "1" is active and "0" is inactive. The Open variable is stored in Modbus register %QX0.0 (Digital Output, offset 0) while the Close variable is in register %QX0.1 (Digital Output, offset 1). The ladder logic for the Upper

Gate Command Check is below in Figure 19. The Upper Gate Command Check checks both open and close commands for the upper gate to ensure only one signal is being sent to the physical system i.e. both open and close can not be sent at the same time. This is done by applying an exclusive-or (XOR) logical operation to both the open and close commands. The result of this XOR is then fed as an input to independent AND logical operators for the open and close signals. The result of the AND writes to the corresponding command's coil. Therefore, if both commands are active the XOR result is false leading to both AND gates being false which in turn clears both coils.

If the XOR is true, only one of the AND blocks will be true, so one of the two command signals is active. In practice, this means if both open and close commands are set active at the same time, the PLC will clear both commands and wait for the next command to be given. For safety, there are also negated contacts for the emergency gate command present behind the open and close command contacts. If the emergency signal is active, both open and close commands will be cleared. This emergency signal is stored in register %QX1.1.

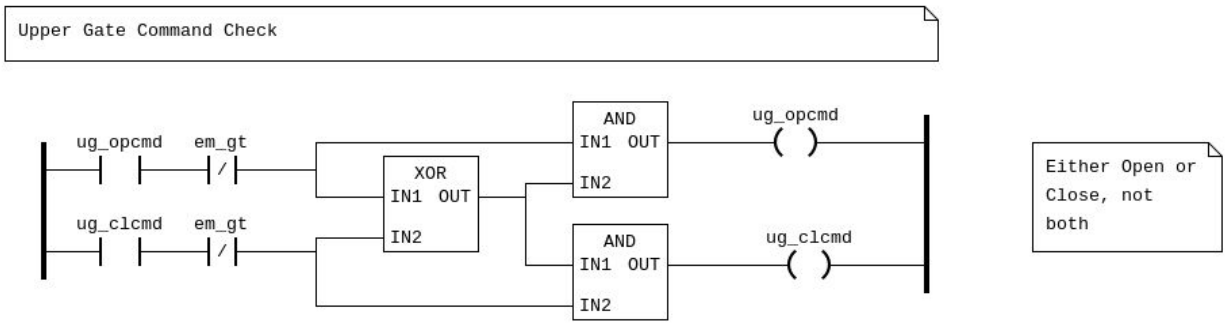

Figure 19: Command check for upper gate

Upper Gate Change Position

The Upper Gate Change Position uses sensor feedback from the physical system to check the current position of the gate versus the position of fully open (90°) or fully closed (7°). The value for the current gate position is stored in Modbus register %IW3 (Analog Input, offset 3). The value stored in the register is divided by 100 before any operation occurs. This is due to the fact the value was multiplied by 100 before being sent by the physical model, as mentioned above in the Physical System description. When the value of the gate's current position becomes equal to or beyond the limit values, the corresponding command is cleared. If the gate is in between the two limits, no action is taken and the command coils remain in their current state. To determine if the gate position is at one of the limits, the position is logically compared to the values of the limits. For fully open, the gate position is checked if it is less-than 90°. If this is true, the gate is not fully open and the open command remains at its current value. If the logic is false, the gate is fully open and the open command coil is cleared. The process is similar for closing the gate. The value of the gate is checked if it is greater than 7°. When this limit is reached, the close command coil is cleared. This logic allows the operator to give the desired open or close command and not worry about issuing a stop command as the command will automatically be cleared when the open or close limit is reached. The ladder logic is seen in Figure 20.

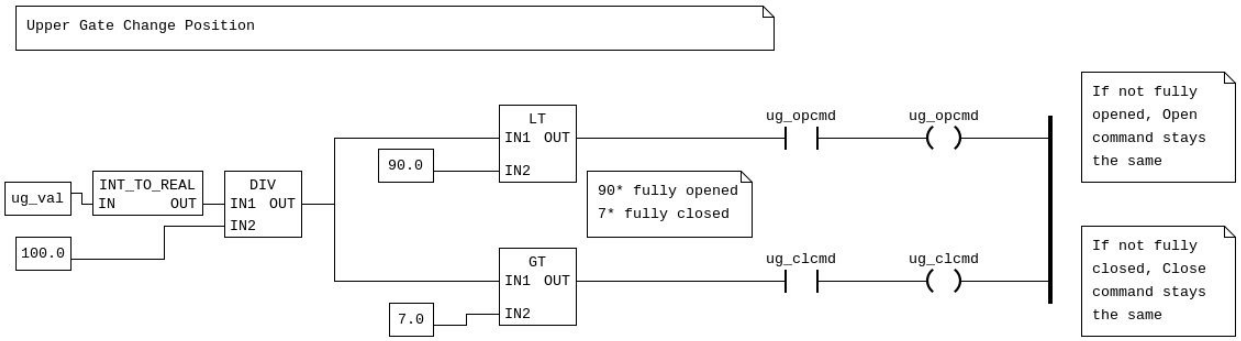

Figure 20: Logic for upper gate position change

Lower Gate Command Check

The Lower Gate Command Check is identical in logic to the upper gate check. The Open command is stored in %QX0.2 while the Close command is %QX0.3. The same emergency signal is used for both the Upper and Lower gate checks. Figure 21 shows the command check for the lower gate.

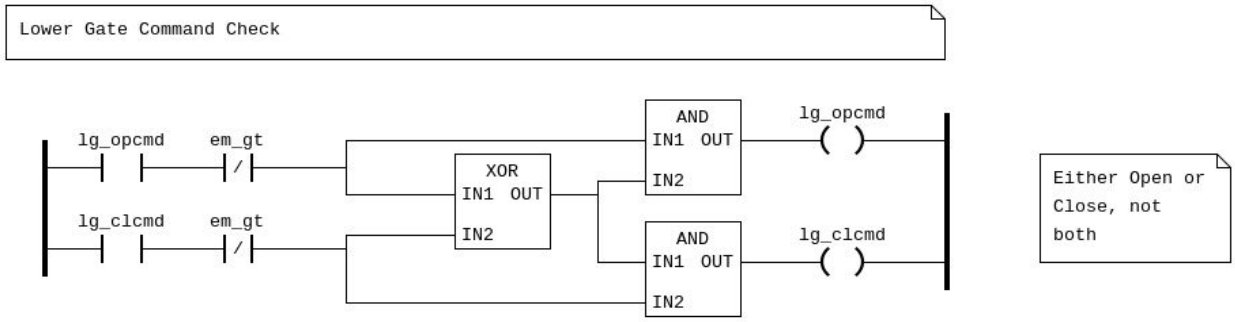

Figure 21: Command check for lower gate

Lower Gate Change Position

The Lower Gate Change Position is identical in logic ot the upper gate position. The logic for the lower gate position change is shown in Figure 22. The value for the current lower gate position is stored in register %IW3. This value is divided by 100 before being used in any logic comparison.

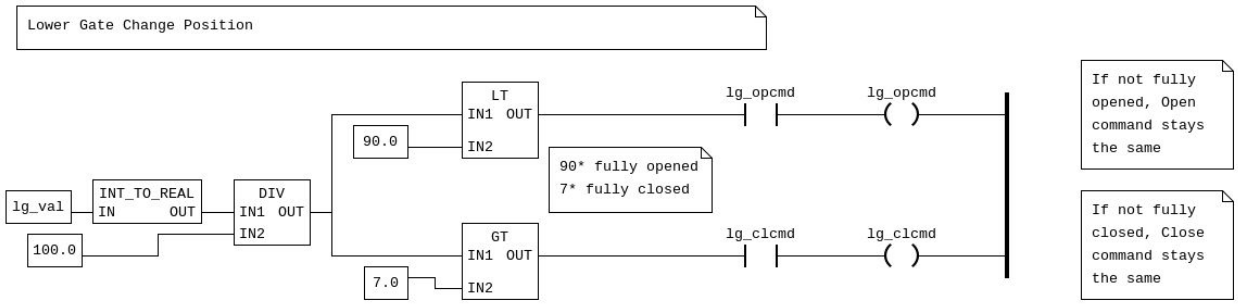

Figure 22: Logic for lower gate position change

Filling Valve Command Check

The core of the Filling Valve Command Check is the same as the gate command checks. The difference is an additional OR logical operator that has inputs of both the result of the AND block for the close command as well as the emergency valve command, as seen in Figure 23. If either of these are true, the close command will be issued. This additional logic will close the valve if the emergency valve command is issued, regardless of what other commands are currently active. The filling valve Open and Close commands are in registers %QX0.4 and %QX0.5, respectively. The binary emergency valve signal is in %QX1.2

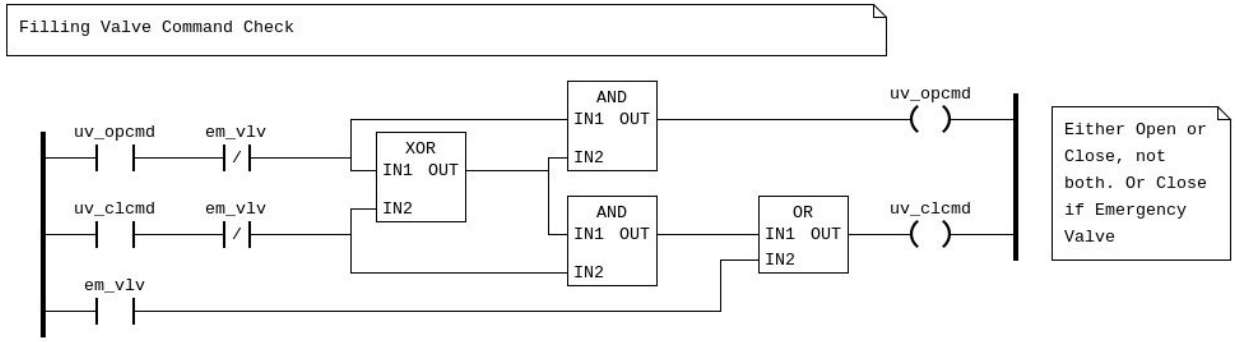

Figure 23: Command check for filling valve

Filling Valve Change Position

The Filling Valve Change Position logic is similar to the position changes of the gates. The only difference from the gates being the valve position values are in percentages rather than degrees, with 100% being fully opened and 0% being fully closed. The current value for the valve position is stored in register %IW4 and is divided by 100 before being used in comparison. The ladder logic for this is seen below in Figure 24.

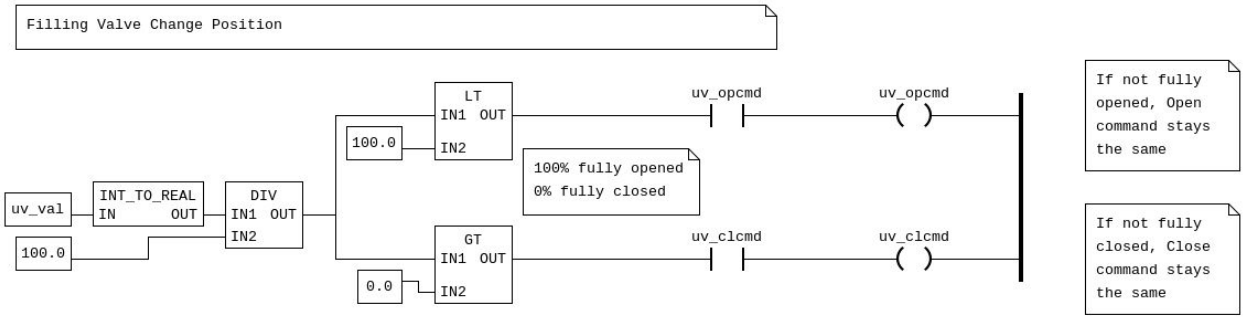

Figure 24: Logic for filling valve position change

Emptying Valve Command Check

The Emptying Valve Command Check, seen in Figure 25, is identical in logic to the upper valve check. The Open command for the emptying valve is stored in register %QX0.6 while the Close

command is in %QX0.7. The same emergency valve signal is used for the filling and emptying valves.

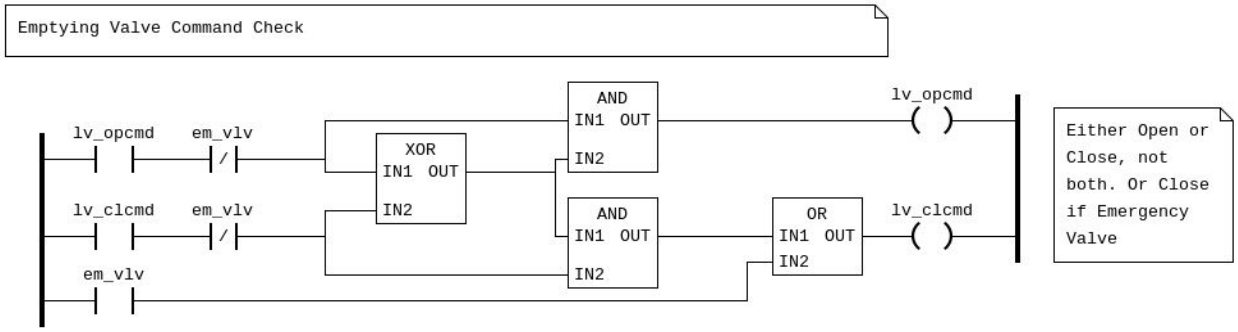

Figure 25: Command check for emptying valve

Emptying Valve Change Position

The Emptying Valve Change Position is identical in logic to the upper valve position. The current emptying valve position is stored in register %IW5 and is divided by 100 before comparison. Figure 26 shows the logic.

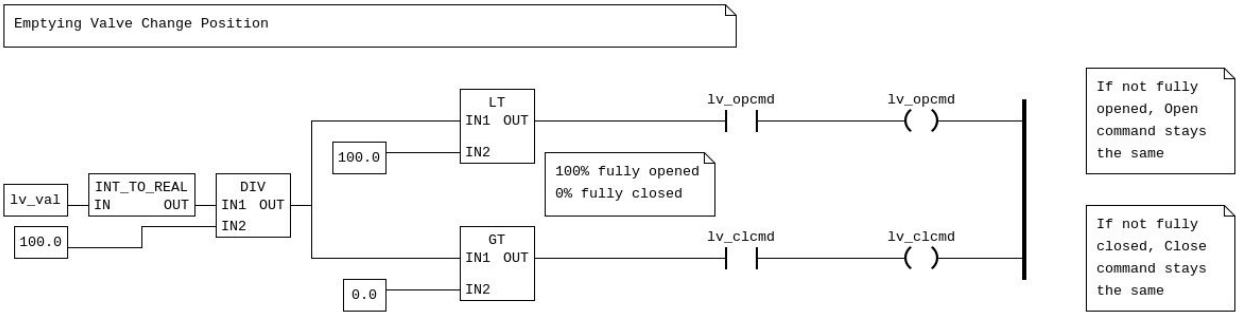

Figure 26: Logic for emptying valve position change

Reservoir Relative Height

The Reservoir Relative Height calculates the depth of the reservoir relative to the floor of the chamber. Due to different floor elevations in the reservoir and chamber, the water depths in the two areas are not the same when the water surface is the same. The sill height represents this difference in elevation. Therefore, the sill height is added to the reservoir depth to get the relative depth of the reservoir. When the sum of these two values is equal to the chamber depth, the water surface of the two areas are equal and the upper gate may be opened.

The sill height is stored in an analog input register %IW9. The real reservoir depth from the sensor is stored in %IW7. The relative depth of the reservoir to the chamber is stored in an analog output register %QW0. The ladder logic is seen below in Figure 27.

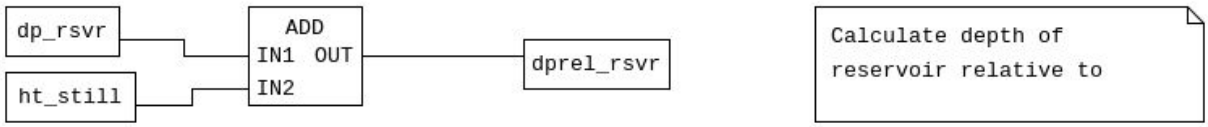

Figure 27: Reservoir relative depth calculation

Depth Compare

The Depth Compare simply compares the depth of the chamber to both the tailwater depth and the reservoir's relative depth. This allows the operator to know when the upper or lower gates can physically open. For both the relative reservoir depth and the tailwater depth, the value is compared with an Equal operator to the chamber depth. If either value is equal, the corresponding coil will be set to true. The logic for this is seen in Figure 28. The chamber depth is stored in register %IW6 and the tailwater depth in %IW8. The calculated relative reservoir depth is in register %QW0, as noted above. The digital output indicating the reservoir and chamber depths are equal is in digital output register %QX1.3. The register for the chamber and tailwater comparison is %QX1.4.

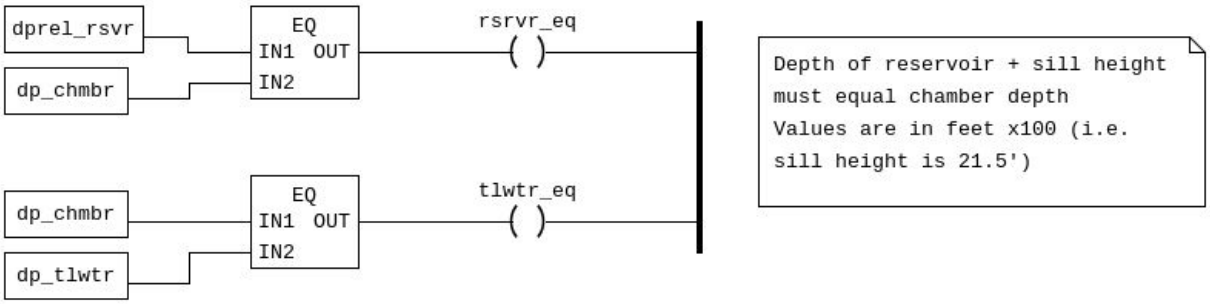

Figure 28: Depth comparison logic

Warning Timer

The Warning Timer takes the input of the manual warning and passes it along to a physical system such as a horn. When the signal goes active, it passess through a pulse timer where it remains active for 3 seconds. After the 3 seconds, the warning command is cleared. This allows the operator to press a button and have a warning horn sound and/or lights flash for three seconds when needed to alert ship operators or nearby boaters. The digital output register for the warning signal is %QX1.0. The logic is seen below in Figure 29.

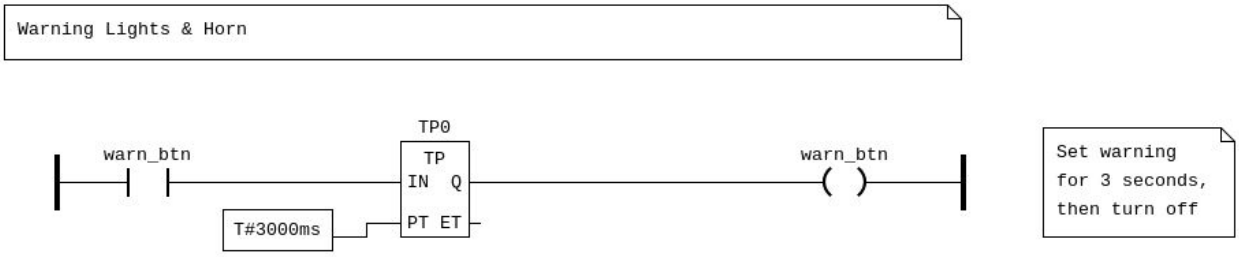

Figure 29: Manual warning logic

HMI

The Human Machine Interface (HMI) is the view the operator has of the system. It contains displays and commands that allow the operator to control and view feedback from the physical model and PLC. A view of the entire HMI is seen below in Figure 30. To preserve their significant figures, all depth values as well as gate and valve positions were multiplied by 100 when leaving the physical model. To account for this, all of these values are divided by 100 on the HMI server-side scripts. Similarly, the filling and emptying flow rates are multiplied by 10 in the HMI in order to display the true value.

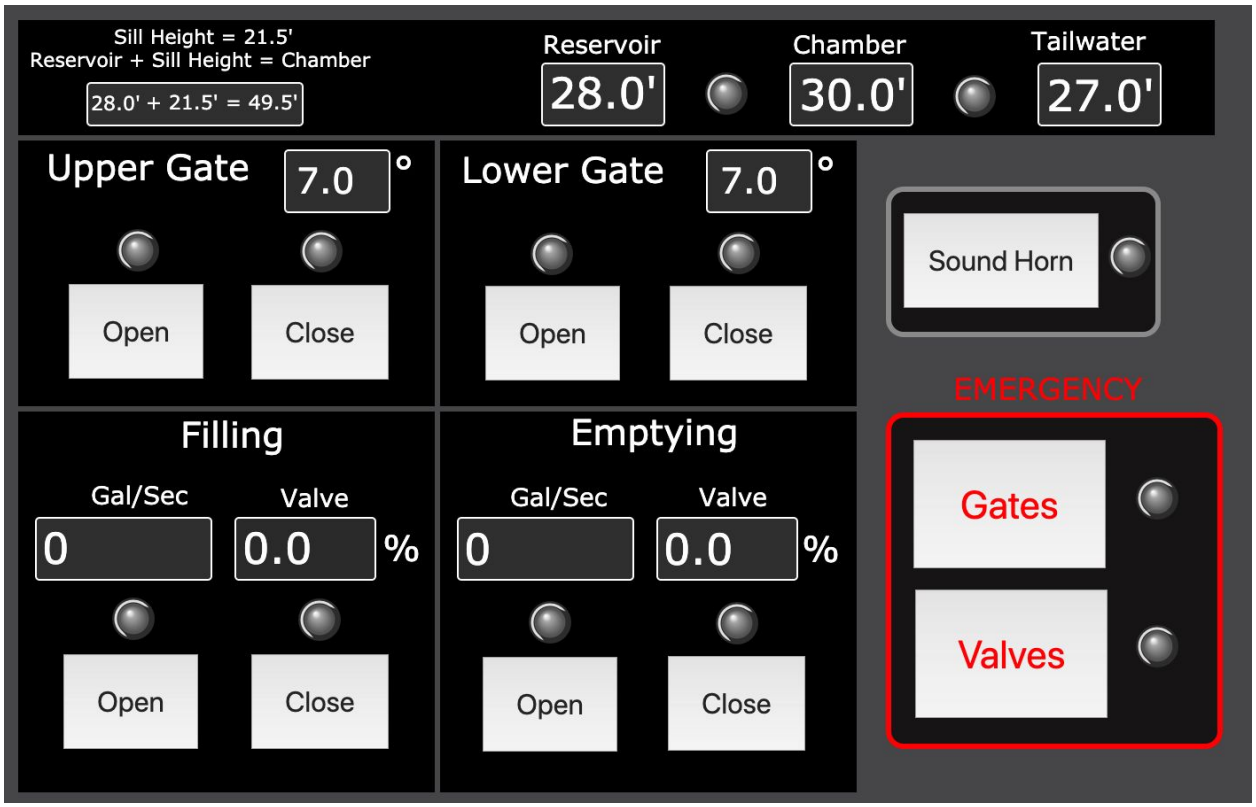

Figure 30: Full HMI operator's view

Chamber

The Fill Rate is the flow rate, in gallons per second, of water entering into the chamber of the lock. This value is calculated in the physical model and available to the operator through the HMI as an analog display. The Empty Rate is the flow rate of water exiting the chamber and is also shown as an analog display. The filling and emptying rates can be seen in their corresponding sections as shown in Figure 31.

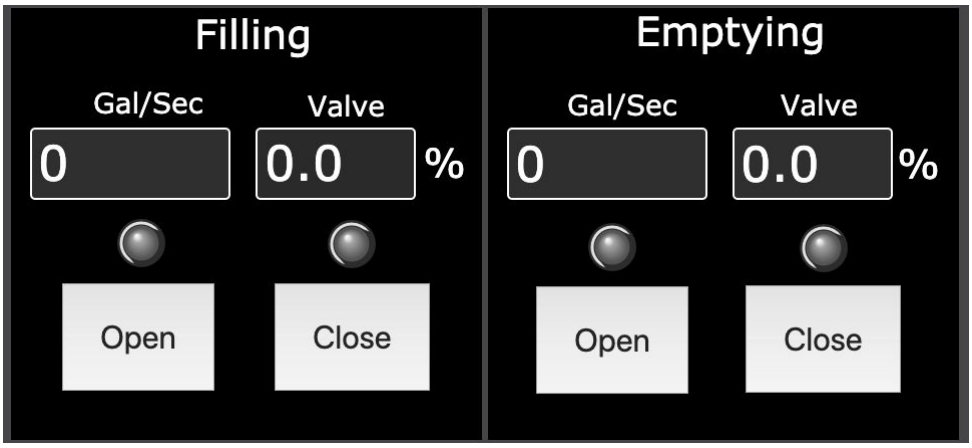

Figure 31: Filling and emptying rates

Upper Gate

The Open command is represented as a binary button on the HMI. When the operator presses the button, the open command is sent to the PLC. Similarly, the Close command is a binary button on the HMI. The Upper Gate Value is the value of the current position of the gates. The value is calculated and stored in the physical model and passed on to the PLC, for use in control logic, and HMI. The value is shown as an analog display on the HMI with units in degrees, 90° being fully open and 7° fully closed.

Lower Gate

The Lower Gate is identical to the Upper Gate controls and displays on the HMI. Figure 32 shows the portion of the HMI containing the upper and lower gate commands.

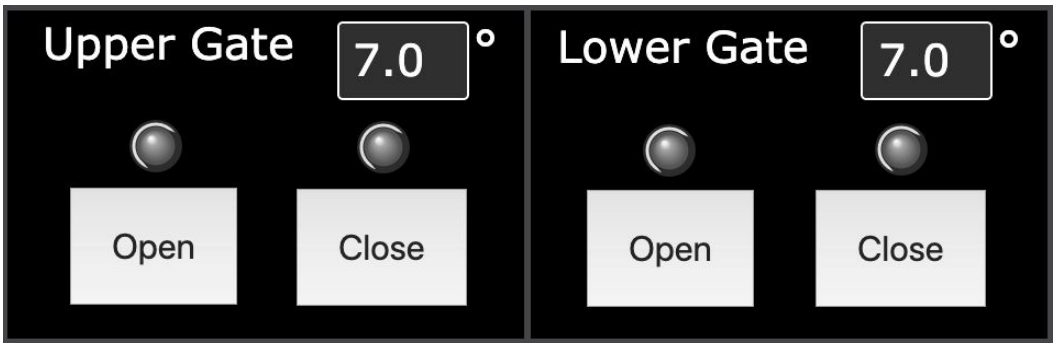

Figure 32: Upper and lower gate commands and values

Filling Valve

Similar to the gates, the Filling Valve Open and Close commands are shown as binary buttons on the HMI. The Filling Gate Value is calculated and stored in the physical model and passed on the PLC and HMI. It is shown on the HMI in an analog display as a percentage from 0% (fully closed) to 100% (fully open). The previous Figure 31 shows the commands and value for the filling valve.

Emptying Valve

The Emptying Valve is identical to the Filling Valve controls and displays on the HMI. The HMI section for the emptying valve is also seen in the previous Figure 31.

Water Depths

The water depths are represented as analog displays on the HMI. The Chamber is the middle portion of the locks, the Reservoir is the upstream portion, and the Tailwater is the downstream portion. All three values come from sensor readings in the physical model and are passed through the PLC. The values are displayed with units of feet as seen in Figure 33.

Figure 33: Water depths displays on the HMI

Depth Compare

The Sill Height is a constant in the physical model and passed on to the PLC and HMI. It is the difference in floor elevation from the reservoir to the chamber. This value is shown as an analog display on the HMI and is needed for calculations determining the equality of water surface levels between the reservoir and chamber.

The Reservoir Relative is the relative, or adjusted, depth of the reservoir compared to the chamber. This value is the sum of the sill height and the reservoir depth that is calculated in the PLC. This value is shown as an analog display on the HMI, seen in Figure 34 below.

The Reservoir Equal and Tailwater Equal are both binary values that are shown as binary displays in the form of LEDs on the HMI. These LEDs are located in between the two values being compared as Figure 33 above shows. These values are the results of the Depth Compare block in the PLC. When the reservoir relative depth is equal to the chamber depths, the Reservoir Equal LED will be turned on. When they are not equal, the LED will be off. The Tailwater Equal LED will be on when the tailwater and chamber are the same depth and off when they are not.

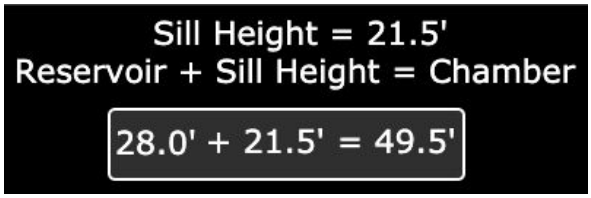

Figure 34: Reservoir relative depth calculation

Emergency

Two emergency controls are present on the HMI. Both the Gates and Valves emergency signals are triggered by separate binary buttons on the HMI. As discussed in the PLC section, the emergency Gates signal will cause both the upper and lower gates to stop in their current position. The emergency Valves command will send the Close command to both the filling and emptying valves, cancelling any previous commands either valve may have. Both signals in the HMI can be seen below in Figure 35.

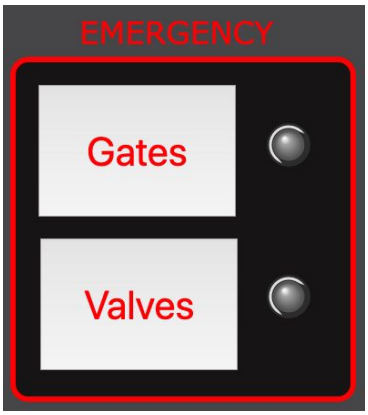

Figure 35: Emergency buttons for gates and valves

Manual Warning

The Warning Horn is shown as a binary button on the HMI, pictured in Figure 36.. When activated by the operator, this signal will be active for three seconds before being cancelled automatically. This allows the operator to warn ships and people in the area before gates or valves are moved. In the physical world, this signal would be tied to a horn and/or lights.

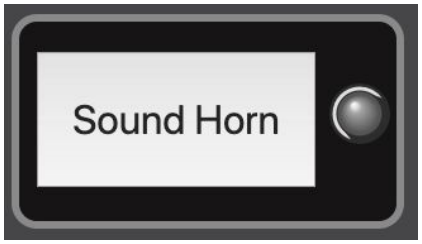

Figure 36: Manual warning button

References

[1] U.S. Army Corps of Engineers, https://www.lre.usace.army.mil/Missions/Recreation/Soo-Locks-Visitor-Center/Soo-Locks-Animation/ (Accessed 3 April, 2019).

[2] V. Igure, S. Laughter, R. Williams, "Security issues in SCADA networks," *Computers & Security 25*, pp. 498-506, 2006.

[3] T. Morris et al. "A control system testbed to validate critical infrastructure protection concepts," *International Journal of Critical Infrastructure Protection 4*, pp. 88-103, 2011.

[4] T. Alves, "OpenPLC: Towards a Fully Open and Secure Programmable Logic Controller," PhD. dissertation, ECE, UAH, Huntsville, AL, 2019.

[5] T. Alves and T. Morris. "OpenPLC: An IEC 61131-3 Compliant Open Source Industrial Controller for Cyber Security Research," *Computers and Security*, 2018.

[6] ScadaBR, "Principle Functionalities." (In Portuguese) http://www.scadabr.com.br/ (Accessed 5 Mar., 2019).

[7] U.S. Army Corps of Engineers, "Event Study of the August 2004 Mcalpine Lock Closure," IWR Report 05-NETS-R-07, 1 September, 2005.

Appendix A: Full System Diagram