Towards Cyber-Physical Representation and Cyber-Resilience Against Attack and Failure within a Hydraulic Network Simulation Toolkit

Sean O'Toole
Department of Computer Scienceand
College of Engineering
Boise State University
Boise, Idaho, 83725
Email: SeanOtoole@u.boisestate.edu

Hoda Mehrpouyan
Department of Computer Scienceand
College of Engineering
Boise State University
Boise, Idaho, 83725
Email:hodamehrpouyan@boisestate.edu

***Abstract*—The ubiquitousness of water distribution and management systems is something often taken for granted in the modern world, as is the physical security and cybersecurity of these critical pieces of infrastructure. As cyberattacks in particular have become easier to perform with the development of security and hacking toolkits aimed at industrial control systems (ICS), effective resiliency modeling for water networks should involve representation of the cyber-physical systems (CPS) which permeate the systems. This paper provides some background on the role of CPS in ICS and the use of simulation in the evaluation of existing systems and the testing of proposed designs. It then provides a solution for some of the current functionality gaps in CPS network simulation in the form of a module add-on for the commonly used hydraulic simulator toolkit, the Water Network Tool for Resilience (WNTR). This module allows simulation of failure and attack scenarios involving CPS devices embedded in the network and is planned to provide an assessment of the criticality of each device from a network level, as well as options for more in-depth simulation of network traffic and device interaction in the future.**

I. Introduction

The modern world is supported by a complex web of physical and cyberphysical infrastructure, where electronic controllers and monitoring devices control switches, valves, lights, and other small physical components in order to safely automate aspects of our water, power, manufacturing, and communications systems. These cyber-physical devices are collectively referred to as Cyber-Physical Systems (CPS) and are considered to be part of the field of Operational Technology (OT), as compared to traditional Information Technology (IT) infrastructure. This distinction is important, as while both IT and OT are part of any critical infrastructure piece, as significantly taught in the discovery and deconstruction of STUXNET [1], the many pieces of technology implemented throughout the world of OT have not always been designed or installed with security and resilience in mind against cyber attacks. Although IT has struggled with the challenges of cyberattacks since the very beginning of the field, the potential to target and disrupt OT has only become fully apparent in recent years, both for those who maintain and protect the infrastructure [2], [3], [4], [5] and those who seek to disrupt it. CPS no longer reliably sit behind air gaps or operate purely on analog signals and pre-programmed parameters, but are part of an ecosystem of devices which are monitored and updated remotely. The benefits offered by this are significant in terms of improving efficiency and identifying problems quickly and theoretically can provide improved security and resilience.

In this work, we adapt cyber-physical system representation models used within the field of hydraulic network simulation and resiliency testing and apply them to an existing industry-standard toolkit for purely hydraulic simulation, allowing for the simulation of cyber-physical attack, failure, and disruption scenarios in a commonly used and well-understood platform. We demonstrate this functionality with an attack scenario in which unique disruptions corresponding to differing levels of manipulation and control of the CPS systems are tested, and assess the likely outcomes of this in a real-world hydraulic system, as well as how this can allow for identification of systemic weaknesses and potential improvements.

The rest of the paper is organized as follows. Section 2 provides a brief background on the use of simulation and modeling in improving resiliency and safety outcomes in systems with cyber-physical and human-interactive components, and then provides a more in-depth look at the strengths and weaknesses of two of the foremost toolkits in hydraulic systems simulation today. In Section 3, we present a proposed solution which builds on elements from both toolkits while aiming to make the solution one which has a broad set of use cases without requiring extensive user knowledge of both the hydraulic and cyber-physical domains involved. Section 4 presents a sample use case applied to a hydraulic network based on a real-world system, and demonstrates both a two-pronged attack scenario and that the module maintains accuracy to the original system with the added CPS components. Finally, in Section 5 we conclude by discussing the current contribution and ongoing work to expand module capabilities by providing more in-depth CPS network simulation and resiliency assessment metrics.

II. Providing Better Security and Resilience Through Simulation and Modeling

The abundance and critical roles of CPS within the infrastructure of every modern society, as well as the ability of these systems to protect against and respond to disaster, system failures or errors, and cyberattacks, are of paramount importance in the future. For this reason, simulation of CPS and digital twin work is a highly relevant field, as it allows for the modeling of existing and proposed CPS systems to estimate the impact known previous attacks might have on systems and the testing of novel attacks against systems without risking damage to the physical infrastructure of a testbed. Given that the threat to CPS systems is not limited to one particular private or public sector, this cost reduction allows a much wider variety of parties to conduct this security and resilience testing than would otherwise be able to afford it. This is not a new approach, as the simulation of safety-critical systems has been utilized by government and scientific organizations such as the National Aeronautics and Space Administration, which found in their 1978 disaster studies that most aviation accidents involved a lack of leadership coordination or decision making [6], and that the simulation of aviation scenarios during training significantly reduced such mistakes.

Projects more focused on the simulation of physical events and modern systems include the Simulation of Urban Mobility (SUMO) [7], Bonceur's CupCarbon [8], the US EPA and Sandia National Lab-backed WNTR [9], and the DHALSIM project developed through work from SUTD Critical Infrastructure Systems Lab and the TU Delft Department of Water Management [10] [11]. These last two projects, in particular, are of interest to the field, as the criticality of access to water and the resiliency of water systems have been highlighted in recent years as a result of natural disasters, government failures in maintaining water infrastructure, and in human conflict zones around the world. Accurate and easily usable simulation toolkits for hydraulic system simulation would allow estimations of resilience and security of existing systems and evaluation of planned systems prior to installation, saving material and labor costs across the board, and improving planning and likely outcomes for disaster scenarios.

A. WNTR

The WNTR toolkit is first and foremost a Python wrapper for the EPANET2 C-based hydraulic simulation engine [9]. It adds significant network modeling capability not present in the underlying hydraulic simulator, and has been expanded over the years to include a number of functions for assessing resilience and criticality of network components. The current release is capable of creating files that numerically model water distribution systems and allow for simulation of these systems over specified periods of time and demand on the network. This includes pressure-driven and demand-driven modeling options, which offer two different ways of estimating water flow through a network based on inputs and outputs, and control over the timestep spacing to allow for more detailed and small time-interval simulations or simulations covering extended periods of simulated time for longer scenarios. Most importantly, WNTR specifically looks to allow testing of the network against attack and disaster scenarios that modify the environment in which the network is operating and can affect the performance of elements of the system. This significantly expands the potential use of the original hydraulic engine, from allowing for water quality and flow performance estimations to allowing for estimations of damage during earthquakes or in the event of physical damage to pipes and nodes within the system.

1) Core Representational Approach: At present, the core of the WNTR toolkit and the underlying EPANET engine represent the hydraulic system that uses nodes and links to capture water sources, pipes, valves, and other key physical components, and a set of controls that define the behavior of these components at given times and conditions. Each of these is assigned to the registries corresponding to their type, and all controls are accessible from within all other components within the code. However, while controls themselves are represented as distinct entities within the toolkit, the SCADA, PLC, RTU, and other CPS devices to which these controls would realistically be written are abstracted entirely. This leaves the system currently unable to capture effects of the types of cyberattacks previously discussed, other than perhaps through roundabout implementation of controls or patterns of demand imitating malicious use.

2) Ongoing Work: Work on WNTR has recently added compatibility functions for the importation and use of Geographic Information System (GIS) files to improve the accuracy of EPANET models and to allow networks that have previously been mapped with GIS to be easily imported and tested within WNTR [12]. Work at the same time has also begun on allowing simulations to be interacted with in real-time, or by interrupting simulation to await human input or data ingestion as opposed to simply running a full simulation based on preset file inputs. Allowing for this stepwise functionality should even further expand the use cases for WNTR to allow for easier use in live training or in the integration of other programs with the simulation by pausing to await outputs from other sources before proceeding.

B. DHALSIM

The Digital Hydraulic Simulator (DHALSIM), first introduced by Murillo et al. [13] in 2020, aims to address many of the CPS representation gaps present in WNTR and other existing hydraulic system modeling tools. DHALSIM is based on a custom iteration of the Python epynet wrapper for EPANET 2.0 [14], which has been retrofitted to include WNTR as an optional basis for hydraulic modeling and simulation. It additionally utilizes network simulation in the form of miniCPS [15] and a custom YAML parser and SQLite database to show the physical processes, control logic, and network communication of a cyber-physical system while it is under attack. DHALSIM provides users with complete network capture of PLCs, SCADA systems, and network devices involved with the operation of the hydraulic system. This makes it an ideal tool for in-depth simulation of a pre-existing and documented hydraulic system and the corresponding CPS systems or for getting a thorough look at the potential function and resilience of a detailed hydraulic system proposal.

However, DHALSIM currently has two limitations that reduce its accessibility and usability for most potential users: Complexity of learning and its dependency on the computer platform.

1) Complexity: While the expansive functionality that DHALSIM can provide is excellent, the number of systems that the user must learn to access that functionality is correspondingly significant. Relying on two relatively advanced toolkits with custom code woven into them to function, as well as a custom YAML parser for all input files, getting a clear picture of how to design systems and scenarios for the project is a significant hurdle for anyone without experience in programming, water systems simulation, and knowledge of network protocols and designs. The project has already done some of the work to reduce this hurdle, as the YAML parser is intended to allow users to simply go through the five required YAML config files and punch in values they could draw from the corresponding hydraulic input and network configuration files. However, at the time of publication, DHALSIM provides little in the way of syntax documentation for the parser and requires the user to create five separate files defining network configuration and control logic before simulations can be run, which poses a significant barrier to use.

2) Platform Dependence: As a result of the project's use of the miniCPS module for network traffic simulation, DHALSIM is currently only usable within a Linux-based operating system distribution. Although this is not a significant hurdle for most in the research and academic spaces, it is a hurdle for many small government and utility organizations dealing with water distribution, as EPANET is platform-agnostic, but is most commonly used with Microsoft Windows or MACOS-based distributions, as these are the most common operating systems licensed by government and small utility systems across the developed world. While this is not to say that a Linux-focused platform is a negative, it should be said that to see the widest usage and provide the greatest benefit to the community of critical infrastructure professionals at large, a platform-agnostic toolkit would be preferred over any which must be limited to one distribution or another.

III. WNTR+CPS: A More Flexible and Usable Solution

To address the issues possessed by both excellent pre-existing projects, a module add-on to the WNTR project has been developed, which seeks to bring much of the CPS representation of DHALSIM into a more accessible and platform-agnostic toolkit, as well as to open the door to future additions to the main project in the same vein. This module, tentatively referred to as WNTR+CPS, adds representation of CPS elements such as Supervisory Control And Data Acquisition (SCADA) units, Programmable Logic Controllers (PLC), and Remote Terminal Units (RTU), as well as assignment of controls representing operational logics to individual CPS devices and modification of said controls. This allows for simulation of a fuller and more representative water systems network and of failure and attack scenarios involving CPS devices embedded in the network. Such scenarios could include power failure, denial-of-service attacks, and hijacking of the control system and malicious control manipulation.

Although representing the complete package of the interconnectivity of WNTR in an easily readable format is not feasible, we have captured the components of the hydraulic model core as well as the CPS node components added in Figure1. As of the time of publication, an implicit hierarchy of subclasses exists through function implementation, as SCADA units are capable of referencing a list of assigned PLCs in order to change or remove controls, but similar cross-device changes are not possible from within PLC or RTU devices. The basic logic checks implemented can be seen in Figure 3. PLCs are capable of changing their controls during operations, but RTUs are not. All of this reflects the general capability of their real-world counterpart device classes and is intended to allow for more in-depth and accurate modeling.

The CPS_node abstract class and implemented subclasses mirror the design used by WNTR's preexisting node classes, and any future implementation of communication or interaction between nodes will mirror the design of the corresponding link classes. This serves to aid in code consistency and ease of visualization alongside preexisting code and allows users to more easily learn how to create networks and scenarios, restricting the required file creation to one file and the syntax to the same Python-based approach as the rest of the WNTR toolkit. Additionally, this will allow better the use of the existing directed graph criticality analysis on both the hydraulic and the CPS sides of the system. However, this design choice does require bidirectional references, as maintaining preexisting class hierarchies while implementing control ownership requires that CPS devices contain a list referencing owned controls, and controls must also contain a pointer referencing the CPS device to which the control has been assigned, as shown in Figure 2. Nevertheless, the comparative reduction in required background knowledge, as well as availability of documentation for the connected toolkit, still makes for an overall simplification of the complexity to learn compared to DHALSIM at the time of writing.

This overall implementation at present represents a low-fidelity emulation of CPS device behavior, capturing control ownership and the relationships between CPS devices, but without simulation of firmware or emulation of hardware behavior. Work is ongoing to simulate the network-level behavior of CPS devices, representing the outputs of PLC and RTU operation, but emulation of other characteristics unique to hardware and firmware is not within the scope of this work.

Accurate emulation of internal firmware and hardware behavior represents a significantly greater challenge than traffic simulation and relationships between devices, given the proprietary nature of both in most real-world PLC and RTU devices.

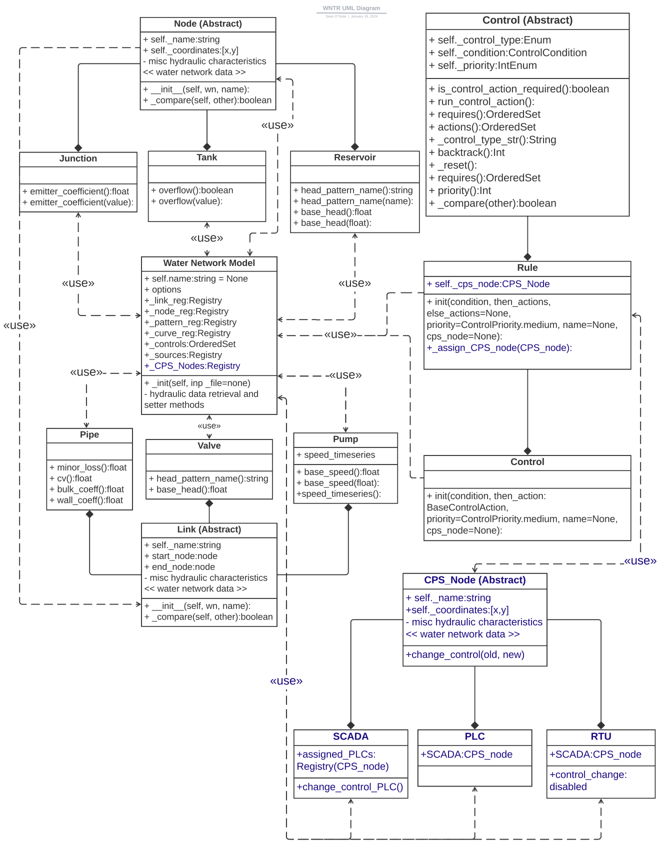

WNTR UML Diagram
Sean O'Toole | January 19, 2024
Node (Abstract)
+ self._name:string
+ self._coordinates:[x,y]
- misc hydraulic characteristics
<< water network data >>
+ __init__(self, wn, name):
+ _compare(self, other):boolean
Junction
+ emitter_coefficient():float
+ emitter_coefficient(value):
Tank
+ overflow():boolean
+ overflow(value):
Reservoir
+ head_pattern_name():string
+ head_pattern_name(name):
+ base_head():float
+ base_head(float):
Control (Abstract)
+ self._control_type:Enum
+ self._condition:ControlCondition
+ self._priority:IntEnum
+ is_control_action_required():boolean
+ run_control_action():
+ requires():OrderedSet
+ actions():OrderedSet
+ _control_type_str():String
+ backtrack():Int
+ _reset():
+ requires():OrderedSet
+ priority():Int
+ _compare(other):boolean
Water Network Model
+ self.name:string = None
+ options
+_link_reg:Registry
+_node_reg:Registry
+_pattern_reg:Registry
+_curve_reg:Registry
+_controls:OrderedSet
+_sources:Registry
+_CPS_Nodes:Registry
+ _init(self, inp _file=none)
- hydraulic data retrieval and setter methods
Rule
+ self._cps_node:CPS_Node
+ init(condition, then_actions, else_actions=None, priority=ControlPriority.medium, name=None, cps_node=None):
+_assign_CPS_node(CPS_node):
Pipe
+ minor_loss():float
+ cv():float
+ bulk_coeff():float
+ wall_coeff():float
Valve
+ head_pattern_name():string
+ base_head():float
Pump
+ speed_timeseries
+ base_speed():float
+ base_speed(float):
+speed_timeseries():
Control
+ init(condition, then_action: BaseControlAction, priority=ControlPriority.medium, name=None, cps_node=None):
Link (Abstract)
+ self._name:string
+ start_node:node
+ end_node:node
- misc hydraulic characteristics
<< water network data >>
+ __init__(self, wn, name):
+ _compare(self, other):boolean
CPS_Node (Abstract)
+ self._name:string
+self._coordinates:[x,y]
- misc hydraulic characteristics
<< water network data >>
+change_control(old, new)
SCADA
+assigned_PLCs: Registry(CPS_node)
+change_control_PLC()
PLC
+SCADA:CPS_node
RTU
+SCADA:CPS_node
+control_change: disabled
«use»

Fig. 1: WNTR Water Network Core Components (Black) + CPS Module Additions (Blue): Model, Nodes, Links, Controls, CPS_Nodes

```
Assign Controls and CPS_nodes
Requires: model_controls, CPS_network
$CPS\_network.SCADA\_list \rightarrow SCADA$
$SCADA.assigned \rightarrow PLC$
for $control$ in $model\_controls$ do
    $PLC.controls \leftarrow control$
    $control.cps\_node \leftarrow PLC$
end for
return CPS_network
```

Fig. 2: Assigning controls to CPS_nodes, and vice-versa, using a SCADA-assigned PLC as example

However, general characteristics and potential failure states could be represented fairly easily within this toolkit through the use of existing functions that modify controls and custom classes that represent specific PLC and RTU devices. Similar subclass-based specification has already been demonstrated and thoroughly tested in core WNTR within a variety of controls and condition checks representing unique categories of hydraulic valves and pumps and their associated hydraulic characteristics.

As an example, the representation of hardware characteristics of a PLC device could be left relatively abstract by using variables representing memory, processor data, and network connection capabilities, or could be made much more sophisticated through integration of Python modules for emulation of hardware behavior. Similarly, while emulation of commonly used firmware is made much more challenging due to their proprietary nature, some characteristics such as I/O count, average signal processing time delay, and permitted communication protocols could be specified as unique device properties.

```
SCADA Operation Modify Control
Requires: model_controls, SCADA, PLC, control, modified_control
if $control$ in $model\_controls$ then
    if $PLC$ not in $SCADA.assigned$ then
        return Reject
    else if $control$ not in $PLC.controls$ then
        return Reject
    else
        $PLC.controls[control] \leftarrow modified\_control$
        return Accept
    end if
else
    return Reject
end if
```

Fig. 3: Permission checking for changing commands

IV. Results

To show the functionality of the module, scenario testing was performed utilizing the commonly used hydraulic network file ’Net3.inp’, representing the operation of a hydraulic network for a small town over the course of one week. This simulation represents average daily consumption through demand patterns for water use and rules around when pumps and valves should be opened or closed. The system is dual-source, pulling from both a river and a town reservoir, theoretically giving the town some resilience against system failures. The pressures (in Pascals) of a handful of representative nodes at the main city pump, three water tanks, network junctions, and outlets in houses in the network for this baseline week can be seen in Figure5.

To demonstrate the use of control assignments and modifications, the scenario of a cyberattack is implemented in two parts. First, the attacker modifies the control parameters for the city’s lake-drawing pump, fully disabling scheduled on-off cycles. Additionally, the attacker removes the rule that triggers the city’s river-drawing pump to activate when the largest of the reserve water tanks drops below a certain level. This change takes effect immediately, but while the effects of disabling the lake draw pump rapidly manifest, it is only at 10 hours into the simulation that the lack of back-up controls for the river pump tank level monitors leads to a total flatline of the water tank levels.

The results of this two-pronged effort can be seen in Figure 7 and Figure 8 in which the pressures immediately drop at hour 20, the river pump is effectively shut off for the rest of the

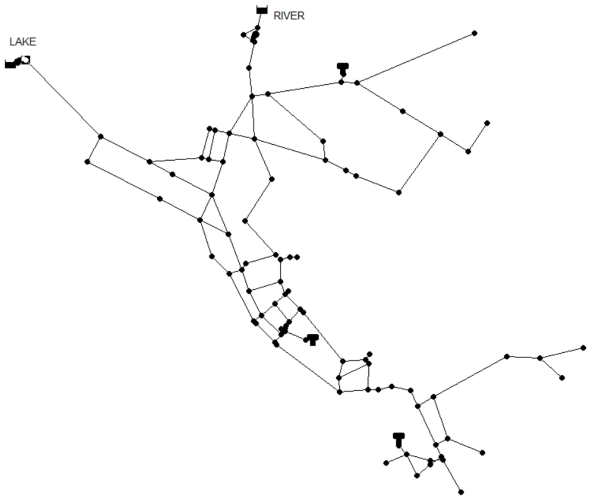


Fig. 4: Hydraulic Network Net3

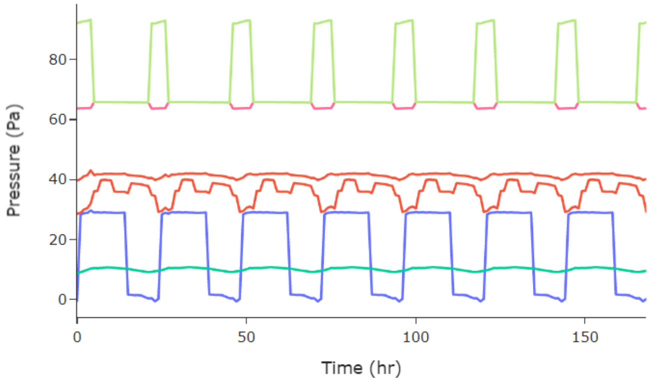


Fig. 5: Pressure Levels at Lake Pump 10 (Blue), River Pump 335 (Light Green) Water Reserve Tanks (Pink, Light Blue, Orange), and Key Junctions over 1 Week (Normal Operations)

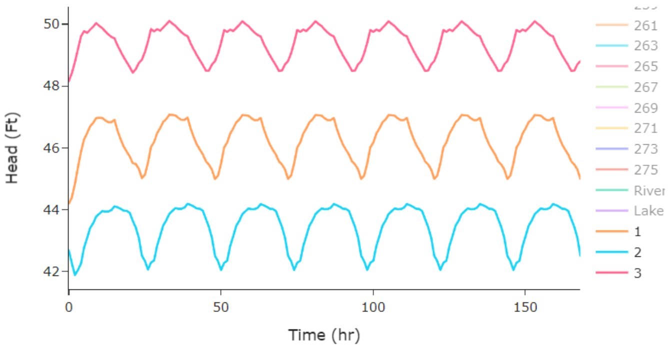


Fig. 6: Head Levels at Water Reserve Tanks over 1 Week (Normal Operations)

duration, and the tanks are rapidly depleted. The head values of each tank level off at minimums relative to their corresponding elevations, minimum levels, and tank diameters, but for all three tanks the result is a drawing-off to the mechanical cut-off point, leaving them effectively empty for the purposes of the town's needs. From then on, the lake pump only partially meets any and all demand during its normal operating hours, and since no controls were implemented for the lake pump to check tank levels or in any way serve as a backup for the river pump, this would realistically result in total loss of water for most residents until intervention by administrators. This should be an indicator for parties conducting this simulation that the system as is has several critical failure points with no identified fallbacks to ensure water availability during a pump outage or cyberattack scenario.

For a utility or city conducting a risk assessment on their water systems as recommended by the US EPA, uses like this would contribute significantly to understanding how, where, and why systems could fail, create more realistic training scenarios for emergency preparedness, and provide a platform for modeling and testing proposed solutions prior to costly real-world testing and deployment [16].

Additionally, to ensure that core functionality is not altered by the use of CPS_node-based functions, we have tested both baseline and the aforementioned attack scenarios using manually modified and CPS-modified controls. The WNTR simulation results for each dataframe, stored in a dataframe, were then compared using the Python *pandas* dataframe difference function. The empty dataframe that results indicates that there is no difference between the original and module-based implementations of both scenarios. However, it is important to note that while CPS vs. manual manipulation of the underlying controls and resulting changes to simulation outputs can be used to verify the integrity of the underlying water network model with the CPS node layer operating, there are scenarios which can be modeled in WNTR+CPS which cannot be easily integrity-tested against core WNTR at present. Particularly when performing interactions between CPS nodes involving authorization restrictions or performing actions which involve both modifying controls, the core WNTR framework has no way of replicating WNTR+CPS functions and validating their effects or lack thereof on the associate water network. This does not represent a gap in the function of the core library, but simply means that the evaluation of the functionality and output purely on the CPS side should be evaluated against the baseline WNTR + CPS control scenarios rather than trying to test against a core WNTR scenario for the purposes of verifying the integrity of the simulation.

As an example, we have created a demonstration scenario comparing two networks in which a number of control assignments and control modifications have been made, with the only difference being that in one an attempt to cause control change was carried out from an unauthorized PLC device, resulting in the change being rejected and preserving the underlying function of the CPS layer and the water network. This represents one of a number of attack, system error, and configuration failure scenarios which could not be represented in the core WNTR toolkit, including controller hijacking or failure and controller/SCADA misconfigurations, and whose results can therefore only be checked against other WNTR+CPS baselines or real-world testbeds the WNTR+CPS model is intended to digitally twin.

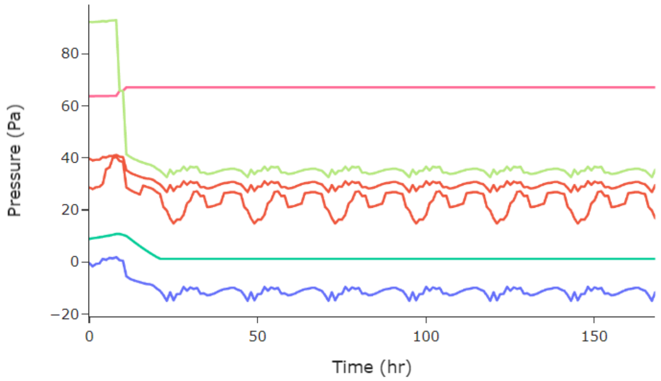


Fig. 7: Pressure Levels at Lake Pump 10 (Blue), River Pump 335 (Light Green) and River-to-Town (Pink), Water Reserve Tank 1 (Green), and Key Junctions (Red, Orange) over 1 Week (Pump Controller Sabotage)

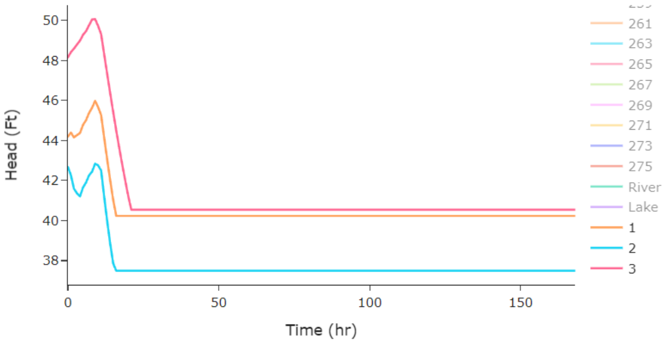
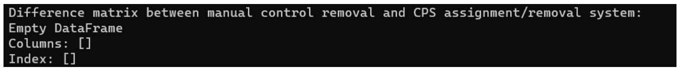
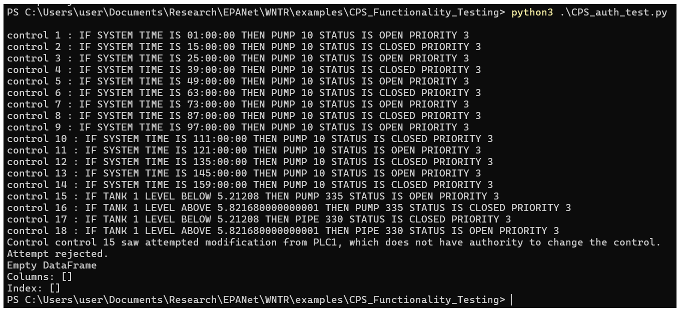


Fig. 8: Head Levels at Water Reserve Tanks over 1 Week (Pump Controller Sabotage)



```
Difference matrix between manual control removal and CPS assignment/removal system:
Empty DataFrame
Columns: []
Index: []
```

Fig. 9: Manual Control Implementation and CPS Implementation Dataframe Difference Matrix



```
PS C:\Users\user\Documents\Research\EPANet\WNTR\examples\CPS_Functionality_Testing> python3 .\CPS_auth_test.py

control 1 : IF SYSTEM TIME IS 01:00:00 THEN PUMP 10 STATUS IS OPEN PRIORITY 3
control 2 : IF SYSTEM TIME IS 15:00:00 THEN PUMP 10 STATUS IS CLOSED PRIORITY 3
control 3 : IF SYSTEM TIME IS 25:00:00 THEN PUMP 10 STATUS IS OPEN PRIORITY 3
control 4 : IF SYSTEM TIME IS 39:00:00 THEN PUMP 10 STATUS IS CLOSED PRIORITY 3
control 5 : IF SYSTEM TIME IS 49:00:00 THEN PUMP 10 STATUS IS OPEN PRIORITY 3
control 6 : IF SYSTEM TIME IS 63:00:00 THEN PUMP 10 STATUS IS CLOSED PRIORITY 3
control 7 : IF SYSTEM TIME IS 73:00:00 THEN PUMP 10 STATUS IS OPEN PRIORITY 3
control 8 : IF SYSTEM TIME IS 87:00:00 THEN PUMP 10 STATUS IS CLOSED PRIORITY 3
control 9 : IF SYSTEM TIME IS 97:00:00 THEN PUMP 10 STATUS IS OPEN PRIORITY 3
control 10 : IF SYSTEM TIME IS 111:00:00 THEN PUMP 10 STATUS IS CLOSED PRIORITY 3
control 11 : IF SYSTEM TIME IS 121:00:00 THEN PUMP 10 STATUS IS OPEN PRIORITY 3
control 12 : IF SYSTEM TIME IS 135:00:00 THEN PUMP 10 STATUS IS CLOSED PRIORITY 3
control 13 : IF SYSTEM TIME IS 145:00:00 THEN PUMP 10 STATUS IS OPEN PRIORITY 3
control 14 : IF SYSTEM TIME IS 159:00:00 THEN PUMP 10 STATUS IS CLOSED PRIORITY 3
control 15 : IF TANK 1 LEVEL BELOW 5.21208 THEN PUMP 335 STATUS IS OPEN PRIORITY 3
control 16 : IF TANK 1 LEVEL ABOVE 5.821680000000001 THEN PUMP 335 STATUS IS CLOSED PRIORITY 3
control 17 : IF TANK 1 LEVEL BELOW 5.21208 THEN PIPE 330 STATUS IS CLOSED PRIORITY 3
control 18 : IF TANK 1 LEVEL ABOVE 5.821680000000001 THEN PIPE 330 STATUS IS OPEN PRIORITY 3
Control control 15 saw attempted modification from PLC1, which does not have authority to change the control.
Attempt rejected.
Empty DataFrame
Columns: []
Index: []
PS C:\Users\user\Documents\Research\EPANet\WNTR\examples\CPS_Functionality_Testing> |
```

Fig. 10: Unauthorized Control Change Test

V. Conclusion and Future Works

The WNTR+CPS module at present brings core features for the representation of CPS devices in the WNTR project and allows the creation of CPS-focused failure or attack scenarios. This can be applied for an assessment of resilience in the water distribution system itself against failures or malicious disruption of CPS components. This constitutes a significant contribution to the utility of the overall WNTR project, and opens the door to further contributions regarding representation of CPS devices and communication within hydraulic network simulation, without constraint on user operating system.

Work is underway to flesh out the representation of network connections and relationships between CPS devices within WNTR + CPS, based on the commonly used graph theory format of *G(V,E)* representing CPS devices as vertices and the connections between them as edges, each with an appropriate set of properties. As CPS_Nodes as currently implemented represent vertices, the CPS_edges will be used to represent wired and wireless connections between them. This should allow for further scenario building and testing representing the strengths and vulnerabilities of different physical transmission mediums and the protocols used within them. Additionally, using link criticality assessment scenarios provided by recent WNTR GIS updates as templates, this implementation should allow the evaluation of the criticality of each device and its connections from a network level, providing both metrics for the overall resilience of the network and identifying devices and connections that should be prioritized for improvement or duplication.

This can optionally be combined with the use of the modules pymodbus [17], pycomm3[18], and pyserial [19] to generate simulation-accurate MODBUS, Ethernet over IP (Ethernet / IP) and serial traffic between CPS_nodes based on the scenario. This, once combined with ongoing core WNTR project work intended to allow interactive real-time and step-by-step simulation of network behavior, should enable both simulation of scenarios reliant on realistic network traffic and the use of the toolkit for real-time training scenarios and demonstrations. The version of WNTR+CPS at the time of publication represents the foundation required for those more advanced scenarios once interactive simulations are enabled in the core WNTR, but will continue to be developed and documented as work continues.

Acknowledgment

The authors would like to thank Sandia's Water Network Tool for Resilience (WNTR) team at Sandia National Lab and their partners at the US Environmental Protection Agency (EPA) for their help and support. We would also like to thank National Science Foundation Computer and Information Science and Engineering (CISE), award number 1846493 of the Secure and Trustworthy Cyberspace (SaTC) program: Formal TOols foR SafEty aNd. Security of Industrial Control Systems (FORENSICS).

References

[1] J. Slowik, "Evolution of ICS attacks and the prospects for future disruptive events," p. 15.

[2] C. Agbo and H. Mehrpouyan, "Resilience of industrial control systems using signal temporal logic and autotuning mechanism," in *2023 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech)*, 2023, pp. 0284–0293.

[3] ——, "Achieving cyber-informed engineering through bayesian belief network and sensitivity analysis," in *2023 10th International Conference on Dependable Systems and Their Applications (DSA)*, 2023, pp. 260–271.

[4] C. Ukegbu and H. Mehrpouyan, "Cooperative verification of plc programs using coveriteam: Towards a reliable and secure industrial control systems," in *Proceedings of Cyber-Physical Systems and Internet of Things Week 2023*, ser. CPS-IoT Week '23. New York, NY, USA: Association for Computing Machinery, 2023, p. 37–42. [Online]. Available: https://doi.org/10.1145/3576914.3587490

[5] C. Ukegbu, R. Neupane, and H. Mehrpouyan, "Ontology-based framework for boundary verification of safety and security properties in industrial control systems," in *Proceedings of the 2023 European Interdisciplinary Cybersecurity Conference*, ser. EICC '23. New York, NY, USA: Association for Computing Machinery, 2023, p. 47–52. [Online]. Available: https://doi.org/10.1145/3590777.3590785

[6] P.-N. Carron, L. Trueb, and B. Yersin, "High-fidelity simulation in the nonmedical domain: practices and potential transferable competencies for the medical field," *Advances in medical education and practice*, vol. 2, p. 149, 2011.

[7] K. Bisw, "The international journal on advances in systems and measurements is published by IARIA. ISSN: 1942-261x."

[8] A. Bounceur, "Cupcarbon: A new platform for designing and simulating smart-city and iot wireless sensor networks (sci-wsn)," in *Proceedings of the International Conference on Internet of Things and Cloud Computing*, ser. ICC '16. New York, NY, USA: Association for Computing Machinery, 2016. [Online]. Available: https://doi.org/10.1145/2896387.2900336

[9] K. A. Klise, R. Murray, and T. Haxton, "An overview of the water network tool for resilience (wntr)." 2018.

[10] A. Murillo, R. Taormina, N. O. Tippenhauer, D. Salaorni, R. v. Dijk, L. Jonker, S. Vos, M. Weyns, and S. Galelli, "High-Fidelity Cyber and Physical Simulation of Water Distribution Systems. I: Models and Data," *Journal of Water Resources Planning and Management*, vol. 149, no. 5, p. 04023009, 2023, _eprint: https://ascelibrary.org/doi/pdf/10.1061/JWRMD5.WRENG-5853. [Online]. Available: https://ascelibrary.org/doi/abs/10.1061/JWRMD5.WRENG-5853

[11] A. Murillo, R. Taormina, N. O. Tippenhauer, and S. Galelli, "High-Fidelity Cyber and Physical Simulation of Water Distribution Systems. II: Enabling Cyber-Physical Attack Localization," *Journal of Water Resources Planning and Management*, vol. 149, May 2023.

[12] T. Haxton, K. A. Klise, and D. Hart, "WNTR Capabilities to Support Data Integration and Co-simulation," Henderson, Nevada, May 2023.

[13] A. Murillo, R. Taormina, N. Tippenhauer, and S. Galelli, "Co-simulating physical processes and network data for high-fidelity cyber-security experiments," in *Sixth Annual Industrial Control System Security (ICSS) Workshop*, 2020, pp. 13–20.

[14] "Vitens/epynet," Dec. 2023, original-date: 2016-08-02T09:58:48Z. [Online]. Available: https://github.com/Vitens/epynet

[15] D. Antonioli, A. Agrawal, and N. O. Tippenhauer, "Towards High-Interaction Virtual ICS Honeypots-in-a-Box," in *Proceedings of the 2nd ACM Workshop on Cyber-Physical Systems Security and Privacy*, ser. CPS-SPC '16. New York, NY, USA: Association for Computing Machinery, Oct. 2016, pp. 13–22. [Online]. Available: https://doi.org/10.1145/2994487.2994493

[16] O. US EPA, "Basics of Water Resilience," Feb. 2015. [Online]. Available: https://www.epa.gov/waterresilience/basics-water-resilience

[17] "pymodbus-dev/pymodbus," Mar. 2024, original-date: 2011-12-05T01:30:08Z. [Online]. Available: https://github.com/pymodbus-dev/pymodbus

[18] I. Ottoway, "ottowayi/pycomm3," Mar. 2024, original-date: 2019-06-22T14:11:55Z. [Online]. Available: https://github.com/ottowayi/pycomm3

[19] "pyserial/pyserial," Mar. 2024, original-date: 2015-08-02T22:05:31Z. [Online]. Available: https://github.com/pyserial/pyserial