Pacific Northwest
NATIONAL LABORATORY

PNNL-37048

Scalable Control Co-design for Resilient-by-Design Cyber Physical Systems

December 2024

Thiagarajan Ramachandran
Aowabin Rahman
Soumya Vasisht
Ramij Raja Hossain

U.S. DEPARTMENT OF ENERGY

Prepared for the U.S. Department of Energy
Under contract DE-AC05-76RL01830

DISCLAIMER

PACIFIC NORTHWEST NATIONAL LABORATORY
operated by
BATTELLE
for the
UNITED STATES DEPARTMENT OF ENERGY
under Contract DE-AC05-76RL01830

Printed in the United States of America

Available to DOE and DOE contractors from
the Office of Scientific and Technical Information,
P.O. Box 62, Oak Ridge, TN 37831-0062
www.osti.gov
ph: (865) 576-8401
fox: (865) 576-5728
email: reports@osti.gov

Available to the public from the National Technical Information Service
5301 Shawnee Rd., Alexandria, VA 22312
ph: (800) 553-NTIS (6847)
or (703) 605-6000
email: info@ntis.gov
Online ordering: http://www.ntis.gov

PNNL-37048

Scalable Control Co-design for Resilient-by-Design Cyber Physical Systems

December 2024

Thiagarajan Ramachandran
Aowabin Rahman
Soumya Vasisht
Ramij Raja Hossain

Prepared for
the U.S. Department of Energy
Under Contract DE-AC05-76RL01830

Pacific Northwest National Laboratory
Richland, Washington 99352

Abstract

Critical infrastructure networks, such as power and transportation networks, are often modelled as cyber-physical systems. With ever increasing complexity of these systems, there is a need for newer and more relevant metrics and design tools that will co-optimize the physical system components and control policies to guarantee resilience against cyber and natural threats. To this end, a simulation-based control co-design computational framework that will concurrently determine the system and control parameters of a cyber-physical system to meet pre-specified resilience, operational and economic objectives has been developed. The capabilities of the developed co-design engine are demonstrated by designing the physical components and control parameters of a microgrid system that will meet its resiliency objectives when subjected to various cyber and physical threats.

Executive Summary

Cyber-physical systems are characterized by tight coordination between cyber communication components, physical equipment, control algorithms, and associated hardware/software platforms to implement the controls. These multiple layers with well-defined interfaces are usually designed independently by domain experts with different skillsets, such as equipment vendors, computer-network architects, control engineers, and embedded system designers. In many cases, the overall CPS design and integration process is sequential (or iterative at best) with little to no collaboration between control engineers, equipment manufacturers, and system integrators early in the design phase. Control design is effectively an independent step in a sequential process. However, there exists strong coupling between physical equipment designs and control performance that can significantly affect operational efficiency in critical infrastructures, such as the power grid, buildings, and transportation. Consequently, a sequential design process - that ignores such coupling effects - often results in sub-optimal designs and poor system performance. There is a need to jointly designing plant and control parameters for resiliency planning and mechanisms while accounting for constraints on available system resources (e.g., limited number of sensor locations, equipment size constraints, communication bandwidth limits etc.). This report will detail a state-of-the-art in control co-design framework to jointly optimize system design and control parameters using a unified optimization framework and guarantee system resiliency under adverse conditions and operational uncertainties. The simulation framework is utilized to conduct a vulnerability assessment on the IEEE 123-node test system. Finally, the report also details a data-driven control methodolgy for ensuring resiliency of the IEEE 123-node test system in the presene of communication failures.

Acknowledgments

This work was carried out (contract DE-AC05-76RL01830) under the support from the U.S. Department of Energy Office of Electricity as part of the PNNL-Resilience Through Data-Driven, Intelligently Designed Control (RD2C) Initiative.

Contents

Figures

1.0 Introduction

Cyber-physical systems (CPS) are dynamical systems characterized by tight coordination between a physical system and controllers interacting over a communication network. Design of such a system involves sizing and siting of actuators and sensors, tuning of various control parameters that dictate the behavior of the individual components and determining the topology of the overall system. Currently, each of these tasks are performed by various domain experts, such as equipment vendors, computer-network architects, control engineers, and embedded system designers. In most cases, the overall CPS design and integration process is sequential or iterative with minimal interaction between the design teams (see for e.g [9], [11], [19]). In critical infrastructure networks, such as power grid [35], buildings [6] and transportation [14], there is a strong coupling between physical equipment design and controller performance that can significantly affect the resilience and operational efficiency of such systems. Consequently, a sequential design process - that ignores such coupling effects - often results in sub-optimal designs and poor system performance [26]. Furthermore, with rapid penetration of public-facing IoT devices, wireless sensor networks and point-to-point communications ([1], [16], [21]), critical infrastructure networks are much more vulnerable to targeted cyber-attacks due to the increased attack surface [5]. As such, there is an increased need to adopt a *concurrent design approach* to designing cyber-physical systems in order to improve system resiliency to adversarial disruption along with traditional performance objectives such as economic efficiency and constraint satisfaction.

1.1 Literature Review

Control co-design approaches aim to determine system and control parameters concurrently in order to maximize certain pre-specified system level objectives [18] and have been applied to a wide variety of system design problems (see for e.g [10], [3], [13], [36]). Typically, model-based co-design approaches fall into three categories: iterative, simultaneous and nested. Iterative approaches alternate between optimizing the control approach with a fixed plant design and then optimizing the plant given a fixed control approach repeatedly until convergence. The iterative approach works well when the coupling between the plant and control design is weak, but fails to provide system-wide reliability when those conditions are not met [17]. The simultaneous approach solves the control co-design problem by accounting for all the plant and control interactions in a single optimization model. The nested approach formulates the co-design problem as a bilevel optimizaton problem where the outer loop determines the plant configuration and the inner loop determines the control parameters as a function of the plant configuration. Even if the generated design is a local minimum to the co-design optimization problem, these approaches are able to provide system level guarantees as they are expressed as constraints in the co-design optimization problem ([28], [29]). The main drawback of traditional co-design approach is the need for physics-based models of the target system. While this is feasible for systems operating in a controlled environment, complex cyber-physical systems consists of a large number of interacting components and operate in highly dynamic and uncertain environments. As such, there is a need to adapt the existing model-based co-design approaches to a simulation-based co-design framework.

1.2 Structure of the Report

The report details the work done under the SCOREDEC: Scalable Control Co-design for Resilient-by-Design Cyber Physical systems in the fiscal year 2024. The report is split into following sections: Section 2 details the co-design software development effort, Section 3 provides a detailed application of the framework to an red-teaming usecase. Section 4 details the development of a control methodology for cyber-physical systems in the presence of communication failure. Finally, Section 5 concludes the report.

2.0 Software Development

This section discusses the software development of a computational tool (which we will refer to as the SCOREDEC tool) for co-design of cyber-physical systems (CPS). The technical challenges in developing this tools were in making it modular with respect to the CPS system of interest, the definition of adversarial scenarios, and the optimization algorithm; as well as providing flexibility to the user for deploying the tool on a local machine or on high-performance computing (HPC) platform in a computationally-efficient manner. Figure 1 presents a schematic of the computational architecture developed in *Python*. The architecture allows the user to interface with the co-design engine with user-specific selection of algorithm and domain-specific simulator in a "plug-and-play" manner. Key modules of the architecture are described as follows:

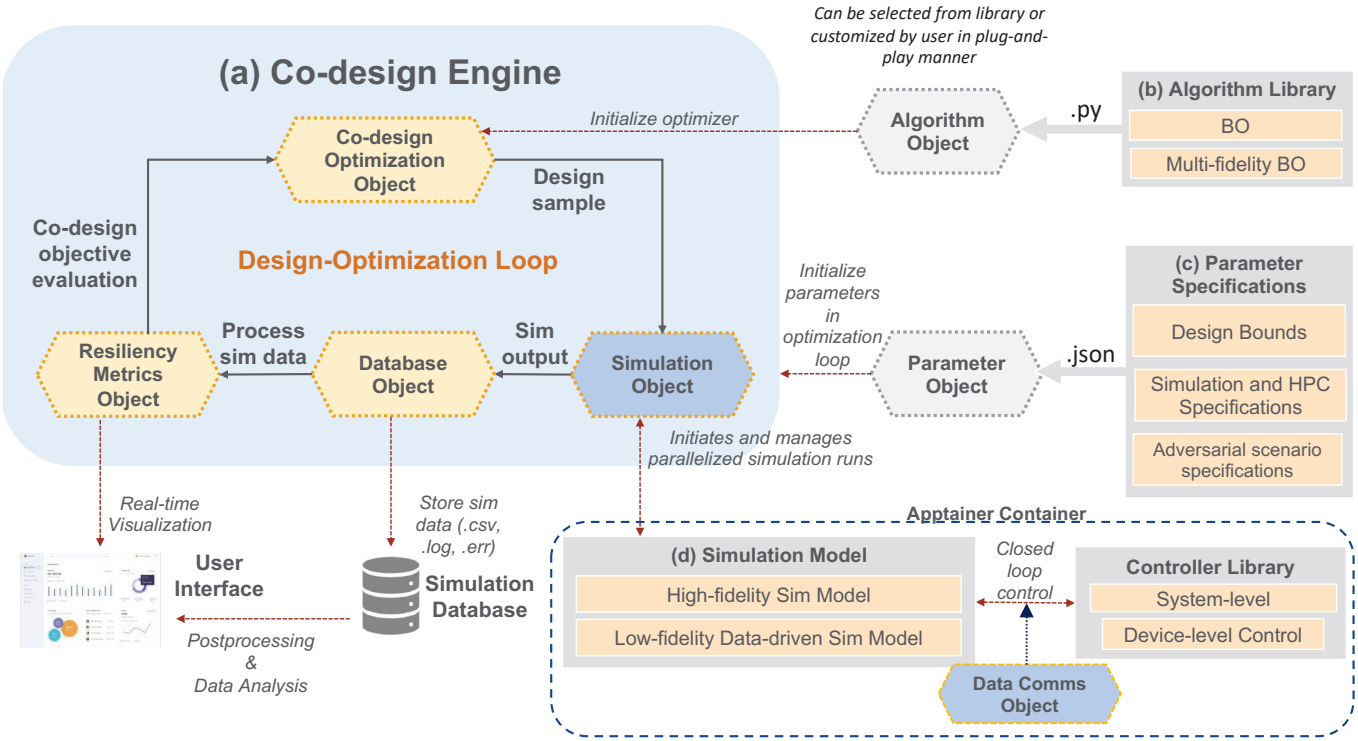

Figure 1. Representation of the computational architecture of the SCOREDEC computational tool. The key components of the comptuational tool are: (a) Simulation-based codesign engine; (b) Algorithm for codesign optimization; (c) Parameters module that allows the user to define the set of allowable design choices and (d) Simulation module (inside a container), which can include low-level controllers-in-the-lop. Each component is modular, and as such, the SCOREDEC computational tool is agnostic with respect to the selection of simulators and codesign optimization algorithms.

- *Co-design Engine*: The co-design engine is designed to easily interface with the domain-specific simulator and the optimization algorithm of interest, allowing the algorithm to compute the next set of design parameters for each iteration. Specific sub-modules within the co-design engine are:
  - *Co-design Optimization sub-module* allows the user to define a problem-specific objective

function and couple it with the optimization algorithm (i.e. *Simulator* module) and the set of design spaces (i.e. *Parameter* module). Currently, the objective can be entered as a closed-loop expression (i.e. a string input parsed by the sub-module) by the user that can be linear or non-linear; or it can be a user-specific function.
  - *Simulations sub-module* interfaces with the domain-specific simulator using sampled values from the *Algorithm module* to run parallel instantiations of the simulator, provided that the simulator is wrapped inside a container. In this context, a "container" will refer to a software package/unit containing the CPS simulator and all required dependencies. The simulations sub-module is designed to be flexible with respect to the availability of compute resources, i.e. they can be deployed on a local machine using *docker* (which is suited to run a container locally) or in an HPC cluster using *Apptainer*. Apptainer (formerly called Singularity) is an open-source platform that can deploy containers on HPC environments, such that they are portable and reproducible [15]. Leveraging existing *Python* libraries *docker* and *SPython*, the sub-module can build an existing container image given a user-specfic *dockerfile* (for *Docker*) or *recipe file* (for Apptainer) and run multiple independent instances of the container. Note that *dockerfile* and *recipe file* are configutation files for *Docker* and *Apptainer* respectively, which creates container with a virtual environment to run the simulator and all its associated dependencies, thereby allowing for modularity with respect to the simulator. To make the workflow computationally-efficient, the sub-module allows for parallel instantiation, where each instantiation corresponds to a single adversarial scenario.
  - *Database sub-module* allows for managing the outputs from the simulation model (which could be .csvs, .log or .txt files)
  - *Resiliency Metrics sub-module* (or Evaluation sub-module) allows for computing the metrics of interest specific-to-the problem, and aggregating the values of these metrics across multiple adversarial scenarios.

- *Algorithm:* The *Algorithm* module computes the design values for each iteration based on the objective defined and parsed in *Co-design Engine module*. These are passed on to the domain-specific simulator via the *Simulations* module. Currently, the SCOREDEC tool comes with the Bayesian Optimization library. However, since the *Algorithm* module is decoupled from the simulator and user-specific objectives, the SCOREDEC tool can support a user-specific optimization library without modifications to the domain-specific simulator, the *Co-design module* or the *Parameters* module.

- *Parameters:* This module consists of user-specific set of co-design variables with the allowable set of values they can take, and the set of adversarial scenarios, which is defined as a set of simulation parameters that can be changed by the user. The co-deisgn variables can include design parameters as well as parameters pertaining to the low-level controllers, and can either be discrete or continuous. Both the co-design variables and the adversarial scenarios can be defined using a user-specific JSON file, and subsequently modified using the SCOREDEC *Python* interface.

- *Domain-specific Simulator:* This module consists of the containerized simulation models specific to the use-case that can evaluate the co-design parameters. The simulation can come with built-in low-level controllers; and the SCOREDEC tool can optimize for parameters associated with these controllers.

The SCOREDEC tool has been demonstrated with multiple use-cases/domains, as discussed below:

- Microgrid simulator with controller-in-the-loop: The SCOREDEC tool has been used to optimize for system design (i.e. capacities of generator, battery and storage) and control parameters (prediction horizon for built-in low-level controller, parameters associated with battery usage, sampling interval for controller) for a microgrid simulator with the goal of minimizing load curtailment, and operating and capital costs; while accounting for multiple types of adversarial attacks (e.g. generator and inverter set-points) The domain-specific simulator in this case was Gridlab-D with a Julia-based low-level controller for optimal dispatch. Results demonstrating co-design using the SCOREDEC tool are presented in [27].

- Commercial chiller plant simulator: The goal for this use-case was to optimize design (i.e chiller capacities) for a commercial chiller plant. The simulator developed by Bhattacharya et al. [7] was containerized, and subsequently SCOREDEC tool was used to interface with the simulator to solve the optimization problem presented in [7]. Note that the goal for using this simulator was for development and demonstration of SCOREDEC only, the results and methods pertaining to the simulator and optimization problem have been published in prior literature [7].

- Autommous red-teaming for NATIG Simulator: Network Attack Testbed in [Power] Grid (NATIG) is a containerized virtual environment for simulating cybersecurity and performance scenarios pertaining to the powergrid [4]. The NATIG simulator contains a power-flow simulator (Gridlab-D) and a network simulator (NS3); and can simulate multiple types of adversarial attacks such as denial of service and man-in-the-middle attacks. The SCOREDEC tool is being leveraged to develop an autonomous red-teaming agent for the NATIG simulator, which can help the power-grid take contingency measures against adversarial attack. The problem formulation and preliminary results pertaining to this task are presented in a later section.

- Optimal Sizing and Integrated Heat Pump Systems for Deep Decarbonization and Energy Resilience: The SCOREDEC tool is being leveraged as part of the Rehoboth project, with the goal of optimal design of heat pump systems $> 20\%$ energy savings. Design parameters include: water tank height, inner diameters, sensor location, compressor target pressure and hydrokit subcooling setpoint.

While the first three use-cases were demonstrated on HPC cluster (i.e. using Apptainer-based container), the fourth use-case (i.e. heat pump system design) was done on a local laptop using a docker container. Currently, the SCOREDEC tool is available at the PNNL stash repo and is undergoing software release. In the following section, we will discuss the application of this tool to conduct an automated vulnerability assessment exercise on a power grid example.

3.0 Automated Red Teaming Agent for Networked Microgrids

Reliable operation of cyber-physical systems (CPSs) such as microgrids require secure operation of the underlying network communication; however CPSs are vulnerable to cyber-attacks that are increasingly becoming more complex [4]. For instance, man-in-the-middle (MITM) attacks can damage the integrity of data sent between two endpoints within a network [12], subsequently affecting grid perfmance [4]. Network attack simulators such as NATIG (Network Attack Testbed in [Power] Grid) that integrate power-flow simulators and network simulators can allow us to understand the impact of adversarial attacks on performance of both the communication layer and the power grid. However, these attack scnearios are often *dynamic*, i.e. the attack values are often dependent on the state of the power grid. As such, there is a need for an automated red-teaming agent that is dynamic, in order to evaluate the performance of power grids during these attacks and provide contingency measures. In this task, we use the SCOREDEC tool to formulate such an automated red-teaming agent leveraging the NATIG test-bed. Methods proposed in this task can support improved design of the physical layer (e.g. capacities of inverters, generators and battery storage) as well as the cyber layer (e.g. network topology)

In the sections that follow, we will describe provide a high-level system description of the NATIG simulator, before describing the controller used for optimal dispatch. Subsequently we present our problem formulation for optimizing parameters corresponding to the red-teaming agent, as well as some preliminary results.

3.1 System Description

3.1.1 NATIG Simulation and Microgrid Configurations

NATIG is a co-simulation environment for distribution power grid network, combinding a power-flow simulation model (Gridlab-D) with a network simulator (NS3-simulator) [4]. NS3 is a network simulator often used for developing and analysing network architecture, and allows for flexibility in simulating different types of attack scenarios [30]; whereas Gridlab-D is a power-flow simulator that incorporates modern grid components (e.g. generators, storage and inverters) and models interactions between them [30]. The co-ordination between NS3 and Gridlab-D is done using HELICS (Hierarchical Engine for Large-scale Infrastructure Co-Simulation) [20]. NATIG accommodates multiple configurations for communication network, as well as different attack types (e.g. denial-of-service, MITM, etc.). In this work, we will consider a 3G network and MITM-type attacks. In terms of the simulation model, we will use the IEEE-123 bus system that can operate in islanded mode.

3.1.2 Controller for Optimal Dispatch

The NATIG simulator has a version of the IEEE-123 bus system, however we made two necessary modifications before leveraging it for developing an automated red-teaming agent. To start with, we added a battery energy storage system (BESS) to one of the inverters, as well as converted one of the loads to be a ”controllable load” (CL). The BESS and the CL can be dispatched whenever there is a load-generation imbalance. Since microgrids often have low system inertia, this can stabilise the system and prevent large frequency deviations [8]. Secondly, we implemented a centralized controller, taken from Bhattarai et al. [8], to compute

the dispatch points for the BESS and the CL when there is a load/generation imbalance. The optimization problem solved to compute the BESS and CL setpoints is detailed in [8], and summarized here for completeness:

$$\max \sum_{k=1}^{N_{Bt}} \alpha_k \Delta P_{Bt}^k + \rho \sum_{m=1}^{N_{CL}} \beta_m \Delta P_{CL}^m \tag{1}$$

$$s.t. \sum_{k=1}^{N_{Bt}} \Delta P_{Bt}^k + \sum_{m=1}^{N_{CL}} \Delta P_{CL}^m = \Delta P_{imb} \tag{2}$$

$$\Delta P_{Bt}^k + P_{Bt}^k \leq P_{Bt}^{k,R}, \forall k \in 1, 2, ...N_{Bt} \tag{3}$$

$$\Delta P_{CL}^m + P_{CL}^m \leq P_{CL}^{m,R}, \forall m \in 1, 2, ...N_{CL} \tag{4}$$

In the equations 1-4, P_{Bt}^k and P_{CL}^m are the monitored powers of the k^{th} battery storage and m^{th} controllable loads respectively; ΔP_{Bt} and ΔP_{CL} are the incremental changes made at a given time to BESS and CL, ΔP_{imb} represents the load/generation imbalance; $P_{Bt}^{k,R}$ and $P_{CL}^{m,R}$ represent the rated BESS and CL powers. The optimization variables are incremental change in setpoints ΔP_{Bt}^k and ΔP_{CL}, i.e. $P_{Bt}^k(t+1) = P_{Bt}^k(t) + \Delta P_{Bt}^k(t)$ and $P_{CL}^m(t+1) = P_{CL}^m(t) + \Delta P_{CL}^m(t)$. The coefficients α_k and β_m can be expressed as:

$$\alpha_k = \Delta \bar{P}_{Bt}^k / \sum_{k=1}^{N_{Bt}} \Delta P_{Bt}^k, \beta_m = \Delta \bar{P}_{CL}^k / \sum_{k=1}^{N_{CL}} \Delta P_{CL}^k \tag{5}$$

Here ΔP_{Bt} and $\Delta \bar{P}_{CL}^k$ denote the available headroom at a given time for BESS and CL respectively. Intuitively this means that a given BESS or CL is incentivized to have larger deviations in its setpoint, if it has a higher headroom.

It should be noted that the controller is agnostic towards the presence of an adversarial red-teaming agent; however, since the primary goal of the red-teaming agent is to increase the load/generation imbalance, the controller (with inclusions of BESS and CL) serves as a mitigation measure to minimize the impact of the red-teaming agent.

3.2 Problem Formulation and Preliminary Results

We formulate the problem by considering that the red-teaming agent will perform MITM attacks on one or more distributed energy resources (DERs) within the microgrid. The goal of the red-teaming agent is to maximize the load/generation imbalance, while avoiding detection. However, the likelihood of being detected can increase with the duration of attack, as well as the magnitude of deviation from the original value [34]. Due to this tradeoff, the MITM attacker needs to target their attacks in a dynamic manner. Therefore, we parameterize our attack parameters as:

$$\gamma = [\gamma_{thresh}, \gamma_{mag,n}], \forall n \in 1, 2...n_{att} \tag{6}$$

Here, γ_{thresh} denotes the threshold fraction of the total generation capacity, with respect to which the red-teaming agent performs the MITM attacks (i.e. it only attacks when the current generation, $P_{gen}(t) \geq \gamma_{thresh} P_{gen,total}$. $\gamma_{mag,n}$ denotes the attack magnitude for a given DER n: $V_{att,n} = \gamma_{mag,n} V_{0,n}$, where V_{att} and V_0 denote the attack and original values respectively, and n_{att} denotes the number of DERs being attacked. Thus, the MITM only occurs when it is likely

to inflict most damage (i.e. likely to achieve a high load/generation imbalance). The design optimization of the MITM attack can be expressed as:

The first term C_1 is associated with load curtailment (i.e. power imabalance), whereas the second term C_2 denotes the penalty term associated with attacking for longer duration and with higher attack values. Here, $\Delta\tilde{P}_{imb}$ represents the normalized load/generation imbalance, i.e. $\Delta\tilde{P}_{imb} = \frac{\Delta P_{imb}}{P_{nc,load}}$, where $P_{nc,load}$ denote the non-controllable loads within the microgrid. The indicator function in C_2 is defined as follows:

$$\mathbb{1}(P_{gen}, \gamma_{thresh}) = \begin{cases} 1, & \text{if } P_{gen}(t) > \gamma_{thresh} P_{gen,total} \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

k_1 and k_2 are weights for each one of the two cost terms C_1 and C_2 respectively. We utilize the SCOREDEC co-design framework (detailed in the previous section) to determine the optimal attack parameters γ_{thresh} and $\gamma_{mag,n}$. For these preliminary results, we consider $n_{att} = 1$, i.e. a single inverter under MITM attack and that the real component of inverter power is compromised under the MITM attack (note that this is a different inverter that is connected to BESS described in section 3.1). We also consider $k_1 = 1$ and $k = 0.05$. Figure 2 presents the preliminary results. The optimal parameters were found to be $\gamma_{thresh} = 0.98$ and $\gamma_{thresh} = 0.02$, which indicates that MITM attack is being done for a short duration with significant reduction. As mentioned in section 3.1, the BESS dispatch can, to an extent, mitigate the power imbalance; so from the perspective of the red-teaming agent, it might not be beneficial to have a sustained attack duration. Furthermore, a sustained attack duration will also result in a higher penalty in C_2.

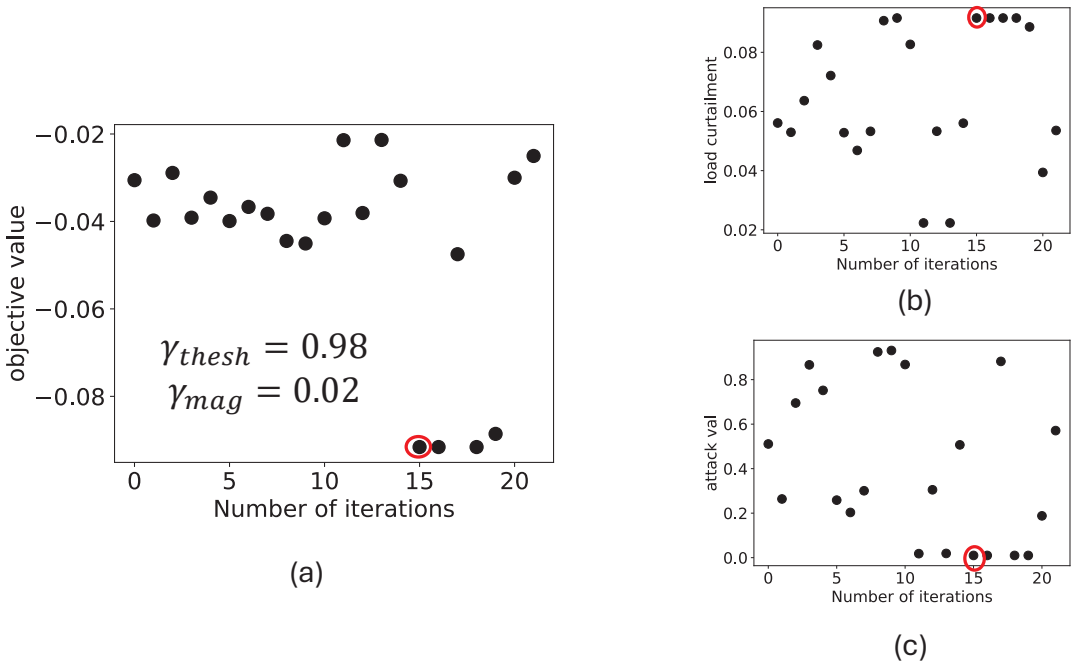

Figure 2. Preliminary results showing of attack parameters γ_{thresh} and γ_{mag}: (a) the value of the objective function; (b) the load curtailment term and (c) the term describing the "strength" of the attack, which accounts for both the duration and magnitude of the attack. The plot shows that at $\gamma_{thresh} = 0.98$ and $\gamma_{mag} = 0.02$ the objective value is at its minimum. At these attack parameter values, the MITM attack values are reduced significantly while not sustaining the attack for lengthy duration.

4.0 Decentralized Control of Cyber-Physical System under Control failures

This section provides a generalized problem formulation and solution methodology to tackle the issues related to communication failure in Networked Dynamical Systems.

4.1 Description: Networked Dynamical Systems (NDSs)

We consider a nonlinear networked dynamical system with n nodes, where the node dynamics for the i^{th} node are represented as:

$$\Sigma_1 : \begin{cases} x_i^{k+1} = f_i(x_i^k, u_i^k) + \sum_{j \in \mathcal{N}_i} f_{ij}(x_j^k), \\ y_i^k = h_i(x_i^k),\ x_i^0 = \zeta_i. \end{cases} \tag{8}$$

where $x_i^k \in \mathbb{R}^{n_i}$, $u_i^k \in \mathbb{R}^{m_i}$, and $y_i^k \in \mathbb{R}^{p_i}$ denote the states, control inputs, and output, respectively, for the i^{th} dynamical node at time instant k. The set $\mathcal{N}_i$ contains all the neighboring nodes of i. $f_i(\cdot), f_{ij}(\cdot)$, and $h_i(\cdot)$ represents the self dynamics of i^{th} node, the coupling between node i and its neighbor node j, and the output function for node i, respectively. The concatenated state and control for the coupled NDS are compactly denoted by $x^k = [x_1^k, \cdots, x_n^k]$ and $u^k = [u_1^k, \cdots, u_n^k]$.

Assumption 1: The mapping $f_i(,) : \mathbb{R}^{n^i} \times \mathbb{R}^{m^i} \to \mathbb{R}^{n^i}$ is assumed to be locally Lipschitz in its arguments, and $f_i(0,0) = 0$.
Assumption 2: We assume that the dynamical system is equipped with a centralized controller $u^k = \pi_{nom}(y^k)$ that solves a systems level optimization problem to achieve the optimal control performance by minimizing the following objective,

$$\min_{u^k} J(y^0, u^k) = \min_{u^k} \sum_{k=0}^{\infty} l(y^k, u^k), \tag{9}$$

where $l(\cdot)$ captures any system-level control design objective of the NDS. In practice, the centralized controller receives the measurement from all the nodes and sends the computed optimal control inputs to individual nodes utilizing a dedicated communication infrastructure as shown in Fig. 3.

4.2 Communication Failures in NDSs — implications and proposed solutions

The communication infrastructure can encounter link failures. As shown in Fig 3, the communication link from a node to controller *(measurement link)* carries the measurement signals, while the link from the controller to node *(actuation link)* is used to send control inputs. If the *measurement link* fails, the centralized controller can use the measurement history or an observer to estimate the measurement signals for the failed node. Following this, we are interested in investigating the scenarios of failure in *(actuation link)*, in which case the centralized controller fails to send the node control signals. Therefore, we intend to design a resilient auxiliary control layer that has partial observability of the system dynamics and only relies on the local information from the individual node. Let us assume that the i^{th} node has

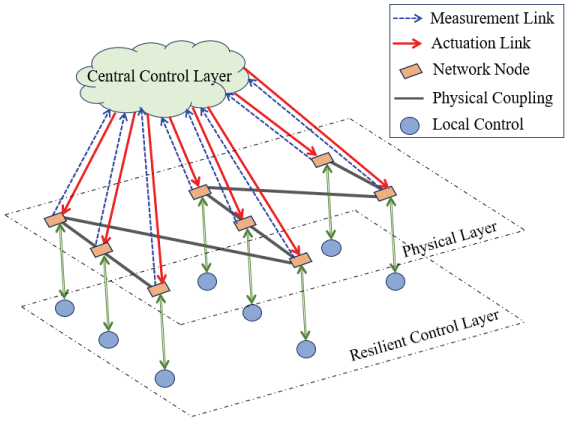

Figure 3. Networked dynamical system with hierarchical controls.

been under link failure (definitely, this can be extended to multiple nodes under failure), and we intend to develop resilient controls for that node. Here, we denote this resilient control input as $u_k^i = u_{res,k}^i$, and express it in a parameterized form as follows.

$$u_{res,i}^k = \mathcal{K}_\theta^i(y_i^k) \tag{10}$$

Please note that the functional form $\mathcal{K}_\theta^i(\cdot)$ represents any generic feedback control function that can be parameterized with $\theta \in \mathbf{R}^w$. For example, in case of the linear output feedback law $u_{res,i}^k = K_{res,i}\, y_i^k$, the elements of the matrix $K_{res,i}$ represents parameter θ. In case of nonlinear resilient control laws $\mathcal{K}_\theta^i(\cdot)$ can be represented by a neural network (NN), where constitute θ constitutes the parameters (weights and biases) of the NN. Next, we formalize the problem statement of finding resilient control law $\mathcal{K}_\theta^i(\cdot)$.

(P1) Resilient control problem: In an event of *actuation link* failure at node i, find a resilient local control law $u_{res,k}^i = \mathcal{K}_\theta^i(y_k^i)$ for node i considering the coupled NDS (8) and minimizing the resilient recovery objective (11).

$$\min_{u_{res,k}} J'(y_0, u_{res,k}) = \min_{u_{res,k}} \sum_{k=0}^{\infty} l'(y_k, u_{res,k}) \tag{11}$$

Note $l'(\cdot,\cdot)$ — resilient recovery objective for node i is specific to that particular node and design requirement related to the underlying problem.

However, there are some challenges in solving (P1) using conventional control design. The local controller can access only the local node information, therefore the complete system dynamics integrated with the centralized control is unknown to the local controller. This motivates us to explore learning-based methods, and we present a methodology for computing the local resilient control law using zeroth order policy gradient (ZOPG)-based reinforcement learning (RL).

4.3 Zeroth-Order Policy Gradient (ZOPG)-based RL

The problem formulation given in (11) solves the communication resilient control problem for each node i in a decentralized fashion. Now, considering nonlinear dynamics we aim to minimize the cost function directly optimizing over the policy parameters θ using a zeroth-order policy gradient (ZOPG) approach. This methods explore in the parameter space rather than the action space in the optimization process and utilizes gradient-free exploration strategies of

Algorithm 1: ZOPG-based Resilient RL

Inputs: A feasible policy parameter θ^0, threshold ϵ, number of ZOPG samples M, step-size η, and number of iterations N.
for $j = 0, 1, \ldots, N-1$ **do**
  for $s = 1, \ldots, M$ **do**
    Sample the random $U_s \in \mathcal{S}_{\mathcal{K}}$;
    Use (12) to return $\hat{\nabla}_\theta J'(\theta^j; U_s)$ using trajectories under actuation link failure.
  end
  Update $\theta^{j+1} \leftarrow \theta^j - \eta \left(\frac{1}{M} \sum_{s=1}^{M} \hat{\nabla}_\theta J'(\theta^j; U_s) \right)$.
end
Return: the final iterate θ^j.

reinforcement learning. The methodology generally uses the networked dynamics with the centralized controller implemented as the oracle model for learning. The gradient is computed using a trajectory-driven cost computation framework. The simulation oracle is excited with control policy $u^i_{res,k} = \mathcal{K}^i_\theta(y^i_k)$ starting from parameters θ, and gathering trajectory data $\{y_k, u_k, J'_k\}$ for sufficient numbers of time-steps under actuation link failures. We can estimate the zeroth-order policy gradient using:

$$\hat{\nabla}_\theta J'(\theta; U) = \frac{1}{r}(J'(\theta + rU) - J'(\theta))U, \tag{12}$$

where U represents a random perturbation following the same structure as θ, with $\|U\| = 1$, while r indicates the smoothing radius. From the initial policy parameter θ, we perform iterative gradient descent updates on θ. To mitigate estimation variance, we consider average of N_i number of ZOPG estimates. Algorithm 1 captures the detailed steps to learn the resilient control parameters θ.

A specific extension to this approach for linear systems and control laws are described next. Let us consider a linear NDS with n nodes each having their own dynamics represented by (13) which is equivalent to (8), and without loss of generality, we assume $y_i = x_i \, \forall i$.

$$x_i^{k+1} = A_{ii} x_i^k + \sum_{j \in \mathcal{N}_i} A_{ij} x_j^k + B_i u_i^k \tag{13}$$

The above system can compactly be represented as $\dot{x} = Ax + Bu$. We assume existence of a centralized LQR control gain $K_{lqr} \in \mathbb{R}^{\sum_i m_i \times \sum_i n_i}$ solving the control problem: $\int_0^\infty (x^\top Q x + u^\top R u) dt$ with $u = -K_{lqr} x$. Now, for i^{th} actuation link failure, the corresponding row of K_{lqr} is all *zero*. The dynamics with failed link — $\dot{x} = (A - K^{fail}_{lqr} B)x$ cannot achieve the desired performance, therefore, we need to add the local resilient control corresponding to failure of i^{th} input. Therefore, each node i has a resilient control gain $K^i_{res} \in \mathbb{R}^{\sum_i m_i \times \sum_i n_i}$ which has structural constraints as: $K^i_{res} = \{K : K(a,b) = 0 \text{ if } a \notin \mathcal{E}_1, b \notin \mathcal{E}_2,)\}$. Note, for i^{th} node, $\mathcal{E}_1 = \{m_i \times (i-1) + 1, \cdots, m_i \times i\}$ and $\mathcal{E}_2 = \{n_i \times (i-1) + 1, \cdots, n_i \times i\}$. The ZOPG algorithm can be extended to compute K^i_{res} with the bounds on the learning rate, smoothing radius, and iterations resulting in convergence to the stationary point of the optimization problem with high probability [23]. The learned K^i_{res} is utilized for online implementation to address communication fails in *actuation links*. The online implementation is done following $\dot{x} = (A - K^{fail}_{lqr} B)x + \sum_i W_i K^i_{res}$, where, $W_i = 1$ if i^{th} *actuation link* fails, otherwise $W_i = 0$.

4.4 Applications to Networked Microgrids

The proposed communication resilient control architecture is integrated with the cyber-physical networked microgrid. With increased penetration of renewable-based distributed energy resources (DERs), the necessity of forming smaller localized microgrid is gaining importance. Usually, these microgrids are self-sufficient and are equipped of renewable and nonrenewable energy resources to supply loads within a clearly defined electrical boundary. For past decades, the operation of islanded microgrids at the distribution level of power systems operations have been proven effective in isolating and supporting critical loads during extreme events. However, the collaborative interconnection among isolated microgrids can help improving the overall system resiliency — introduces the concept of 'networked microgrid' operation [31] in power systems. Networked microgrids involve connecting multiple microgrids to exchange power with the distribution system at a common coupling point (PCC) [2]. This setup optimally utilizes distributed energy resources (DERs), thereby enhancing system resiliency and reliability.

The renewable-based DERs are interfaced to the network via power-electronic inverters, therefore inverter control technologies, particularly grid-forming (GFM) inverters, have emerged as a critical component for improving operational resiliency in networked microgrid operations. GFM controls are hierarchical in nature — the primary controls of GFMs come from P-ω and Q-V actions and following any disturbances this primary control helps GFM to achieve behavior similar to synchronous generators. However, the primary control actions cannot change set-point values, resulting in post-disturbance steady-state deviations in frequency/voltage. This necessitates the introduction of secondary control, which acts as a supervisory control layer —- receives the measurement data and sends optimal set-points for individual GFMs of the networked microgrid.

GFM dynamics with well-tested CERTS [24] power-frequency (P-ω) droop and var-voltage (Q-V) droop controls as primary GFM controls are as follows [33, 22]:

$$\dot{\delta}_i = \omega_i - \omega_0, \tag{14a}$$

$$\dot{\omega}_i = \frac{1}{\tau_i}\left[\omega_0 - \omega_i + m_{p_i}(P_i^{set} - P_i)\right], \tag{14b}$$

$$\dot{V}_i^e = \frac{1}{\tau_i}\left[V_i^{set} - V_i - V_i^e + m_{q_i}(Q_i^{nom} - Q_i)\right], \tag{14c}$$

$$\dot{E}_i = k_i^{pv}\dot{V}_i^e + k_i^{iv}V_i^e. \tag{14d}$$

where, δ_i and ω_i are respectively the voltage angle and frequency of the internal node, V_i, and E_i are respectively the voltages of the external and the internal nodes. m_{p_i}, m_{q_i} are the droop coefficients of P-ω and Q-V droops, respectively. τ_i, k_j^{pv} and k_j^{iv} are respectively the measurement time constants, proportional and integral gains in the Q-V droop control. V_i^{set} and V_i^e represent the voltage set-point and voltage error, respectively. P_i^{set}, P_i, Q_i are, respectively, the active power set-points, active power, and reactive power injection. Q_i^{set} nominal reactive power. From (14) it is important to note that the active and reactive power injection P_i and Q_i are the network interaction variable for individual GFM and can be obtained from standard nonlinear power flow equations. For balanced network operation, P_i and Q_i at node i are:

$$P_i = V_i \sum_{k=1}^{N} V_k \Big[G_{ik}\cos(\delta_i - \delta_k) + B_{ik}\sin(\delta_i - \delta_k)\Big], \tag{15a}$$

$$Q_i = V_i \sum_{k=1}^{N} V_k \Big[G_{ik}\sin(\delta_i - \delta_k) - B_{ik}\cos(\delta_i - \delta_k)\Big] \tag{15b}$$

where G_{ik} and B_{ik} are, respectively, the transfer conductance and susceptance of the line connecting the nodes i and k. N is the number of node in the networked microgrid. Note that, if two nodes i and k are not connected in a power network, $G_{ik} = 0$ and $B_{ik} = 0$; therefore, from (15) it is evident that P_i and Q_i are function of neighboring node variables only. This can easily be extended for unbalanced operation of power network. If GFM is not present in a node, that node does not contain the dynamics given in (14), and can only contain the algebraic equations of P_i and Q_i of (15). Overall, the combined operation of networked microgrids can be represented by the system dynamics Σ_1 given in (8).

Without loss of generality, here, we assume frequency regulation problem of networked microgrid that is managed by a centralized control center — implementing specific control objective through the secondary control layer and dispatches the active power set-points $P^{set} = [P_1^{set}, \cdots, P_n^{set}]^\top$ to mitigate the frequency deviations after any credible contingencies, where $n :=$ number of GFM. In line with our generic formulation, it is assumed that the centralized control formulation of the networked microgrid is already in operation. Now, in presence of communication failure in the *actuation link*, the centralized control center fails to update the set-point of the corresponding GFM. To this end, we are interested in designing a decentralized control law for each GFM, acting as an auxiliary layer that comes into effect upon detection of communication link failure and dispatches local control set-points based on local measurement data. This resilient local control is represented as: $P_i^{set} = \mathcal{K}_\theta^i(\omega_i)$, which minimizes the deviation of local frequency ω_i with respect to nominal frequency ω_0. Next, utilizing the concept of feedback optimization-based control law, we can define the control law in continuous time:

$$\frac{dP_i^{set}}{dt} = -\bar{\theta}_i(\omega_i - \omega_0) \tag{16}$$

consequently the discrete time version is:

$$P_{i,k+1}^{set} = P_{i,k}^{set} - \widehat{\theta}_i(\omega_i - \omega_0). \tag{17}$$

Therefore, to achieve the optimal performance of the local resilient controller, the designer needs to find an optimal $\widehat{\theta}_i$. It is important to note that from local resilient control design perspective, the combined microgrid dynamics given by (14)-(15) is unknown — this necessitates the utilization of data-driven learning based method, as described in Section II, for finding optimal $\widehat{\theta}_i$.

4.5 Numerical Experiments

This section discusses the simulation studies to validate the problem formulation and proposed solution method. The standard IEEE-123 node test system is modified to create three microgrids (Microgrid 1, 2 and 3) and a “region” (Region 4) which can potentially be energized from one of the microgrids but cannot form a self-operating stable microgrid. This test system, which was first introduced in [32] and later adopted in [25], consists of two diesel generators (DGs) with total capacity of 1.6 MW, and five grid forming (GFM) inverters with total capacity of 2 MW. The locations of DGs and GFMs are shown in Fig. 4. The primary controls of GFM inverters come from P-ω and Q-V , and secondary control is designed to appropriately update the control set-points of the inverters to achieve optimal performance based on frequency deviations and power sharing. The networked microgrid is simulated in GridLAB-D where controllers are implemented through HELICS-based co-simulation platform. The *delta* mode in GridLAB-D is enabled to consider the transients that arise due to islanding of microgrids,

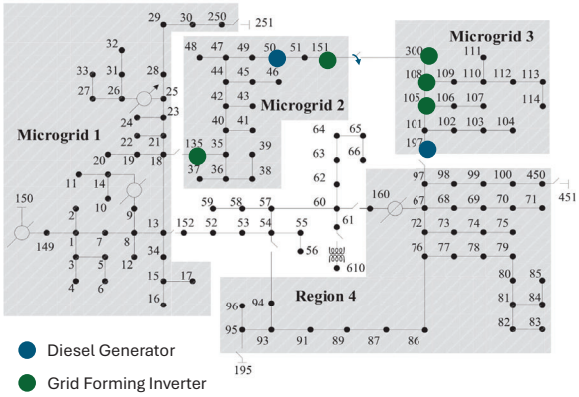

Figure 4. Networked Microgrid using IEEE 123-node distribution system.

energization and de-energization of Region 4. The simulation time step for *delta* mode is $1\,ms$ and the secondary control dispatch happens in every $0.5\,s$.

Scenario 1. We consider a large disturbance scenario with (a) islanding of Microgrid 1 and de-energization of Region 4 at $t = 1\,s$. Microgrids 2 and 3 are operated as a networked microgrid to achieve stable operation, (b) energization of Region 4 at $t = 5\,s$. It is important to note that Region 4, with no generating resources, is added as a 'bulk load' to the networked micorgrid formed by Microgrid 2 and 3, creating a significant frequency deviations. In normal operation with healthy communication, the designed secondary control layer can mitigate the frequency deviation, but in presence of communication failure at *actuation link*, the secondary controller fails to send the updated set-point signal, resulting in undesirable behaviors.

4.5.1 Offline Learning of $\widehat{\theta}_i$

The link failure may happen at any of the GFM, therefore the proposed local resilient controller needs to be integrated at individual GFM, necessitating learning of the optimal $\widehat{\theta}_i$ of (17) for $i = 1, \cdots, 5$. To learn $\widehat{\theta}_i$, we simulated *actuation link* failure for inverter i at $t = 6\,s$ considering the disturbance mentioned in Scenario 1, which is assumed to be the largest disturbance for the system mentioned above. Here, during the learning process, we assume (designers' choice), upon detection of link failure at t', the corresponding set-point $P_i^{set}(t')$ sets to 0, and resilient controller picks after a delay of 1 step of secondary control. The starting P_i^{set} of the learning process is set to 0, because based on the operating condition and disturbance the secondary control-based P_i^{set} for GFM i will vary, and it is prohibitive to learn optimal $\widehat{\theta}_i$ for each such cases. Also, 1 step delay is used to achieve the learning under more pronounced effects of the link failure.

Next, following Algorithm 1, the learning of $\widehat{\theta}_i$ is conducted for each i considering failure of the *actuation link* of i^{th} GFM. Note that $\widehat{\theta}_i$ is scalar quantity, hence, during the iterative process of RL, at j^{th} iteration the ZOPG based gradient computation is done selecting two $\widehat{\theta}_i^j + \Delta\widehat{\theta}_i^j$ and $\widehat{\theta}_i^j - \Delta\widehat{\theta}_i^j$. The objective function is defined as $J_i' := \sum_{t \geq t'} ||\omega_i(t) - \omega_0||^2$. The learning for 5 GFMs are done separately; the corresponding learning curves are shown in Fig. 5. The learned $\widehat{\theta}_i$ s can be utilized for online implementation to validate the efficacy during *actuation link* failure. Here are the learned $\widehat{\theta}_i$ s for different GFMs:
These learned $\widehat{\theta}_i$ s can be utilized for online implementation to validate the efficacy during *actuation link* failure.

Actuation Link Failure at GFM #				
GFM 1	GFM 2	GFM 3	GFM 4	GFM 5
1.29	1.39	1.44	1.15	1.02

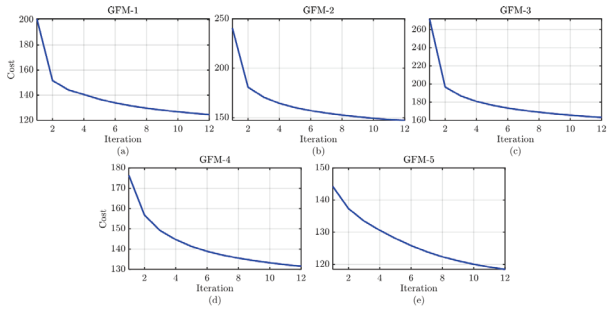

Figure 5. Learning curves: (a) GFM 1, (b) GFM 2, (c) GFM 3, (d) GFM 4, and (e) GFM 5.

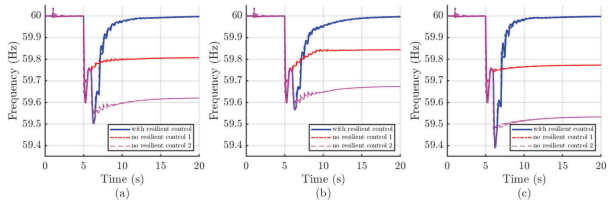

Figure 6. Frequency plots for Scenario 1 under actuation link failure at (a) {GFM 1, GFM 3}, (b) {GFM 2, GFM 5}, and (c) {GFM 1, GFM 2, GFM 3, GFM 4, GFM 5}.

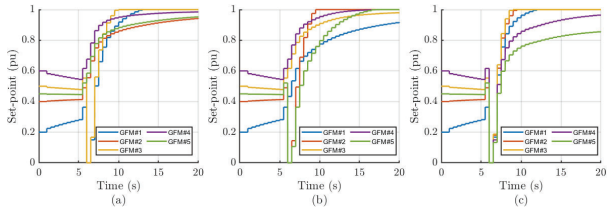

Figure 7. Set-points plots 'with resilient control' for Scenario 1 under actuation link failure at (a) {GFM 1, GFM 3}, (b) {GFM 2, GFM 5}, and (c) {GFM 1, GFM 2, GFM 3, GFM 4, GFM 5}.

In Scenario 1, 3 link failures are tested: combined link failure of (a) {GFM 1, GFM 3}, (b) {GFM 2, GFM 5}, and (c) {GFM 1, GFM 2, GFM 3, GFM 4, GFM 5}. For each of the test cases, we provide the GFM frequency plots with the proposed resilient control, referred as 'with resilient control', and compare them with 2 baselines: (i) 'no resilient control 1' — the set-points of affected GFMs are kept at the last values sent by secondary control layer, and (ii) 'no resilient control 2' — the set-points of affected GFMs are made 0. The plots provided in Fig. 6 clearly shows that under communication failure the proposed method – 'with resilient control' mitigates the finite frequency deviations of baseline cases appeared due to link failures. Fig. 6(c) also shows that under extreme condition of communication failures of all GFMs, the local resilient control can completely minimize the steady state frequency deviations. The set-point plots 'with resilient control' for three cases are given in Fig. 7.

Remark: It is important to note that under communication failure the proposed resilient control can improve the system performance and enhance operational resiliency but it might fail to achieve some system level control objective, e.g. equal active power sharing — evident from Fig. 7. This shows the requirement of system level secondary control layers with robust communication network. But, under failure with deteriorating system performance, local communication resilient control layer can safeguard the system and improve operational resiliency.

To show the proposed mechanism can be effectively handle link failures related issues at other operating condition, we create a separate scenario with relatively smaller magnitude load disturbance.

Scenario 2. Similar to the Scenario 1, at $t = 1\,s$, Microgrid 1 is islanded, Region 4 is deenergized. However, at $t = 5\,s$, instead of reenergization of Region 4, a small amount of load gets connected to the networked microgrid formed by Microgrid 2 and 3. Now, we consider two cases combined link failure of (a) {GFM 3, GFM 5}, (b) {GFM 1, GFM 2, GFM 3, GFM 4, GFM 5}. Fig. 8 provides the frequency plots in comparisons to the baseline cases. Here also, the proposed control proves superior in solving the frequency deviation problem arises from link failure.

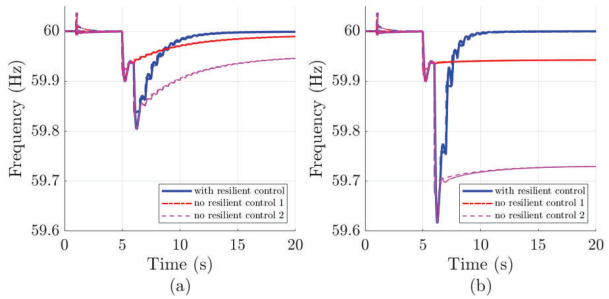

Figure 8. Frequency plots for Scenario 1 under actuation link failure at (a) {GFM 3, GFM 5} and (b) {GFM 1, GFM 2, GFM 3, GFM 4, GFM 5}.

5.0 Conclusion

This report presented a computational framework for concurrent design of plant and control parameters for resilient cyber-physical systems. The applicability of the framework was explored by addressing IEEE 123 microgrid red teaming problem to determine the optimal timing and magnitude of the false data injected as part of a man-in-the-middle type attack. Future work will address scalability challenges associated with co-design computations and aim to provide theoretical guarantees on the optimality of generated designs.

The report also presents the development of an auxiliary control layer to tackle actuation link failures in network dynamical system. The method provides a decentralized, learning-based parametric solution for the proposed control layer. The proposed approach was demonstrated on IEEE 123 networked microgrid with applications to optimal frequency restoration in the presence of wide-spread communication failure.

References

[1] Anders Ahlén, Johan Akerberg, Markus Eriksson, Alf J Isaksson, Takuya Iwaki, Karl Henrik Johansson, Steffi Knorn, Thomas Lindh, and Henrik Sandberg. Toward wireless control in industrial process automation: A case study at a paper mill. *IEEE Control Systems Magazine*, 39(5):36–57, 2019.

[2] Mahamad Nabab Alam, Saikat Chakrabarti, and Arindam Ghosh. Networked microgrids: State-of-the-art and future perspectives. *IEEE Transactions on Industrial Informatics*, 15(3):1238–1250, 2019.

[3] Ali Baheri and Chris Vermillion. Combined plant and controller design using batch bayesian optimization: a case study in airborne wind energy systems. *Journal of Dynamic Systems, Measurement, and Control*, 141(9), 2019.

[4] Oceane Bel, Joonseok Kim, William J Hofer, Manisha Maharjan, Sumit Purohit, and Shwetha Niddodi. Co-simulation framework for network attack generation and monitoring. *arXiv preprint arXiv:2307.09633*, 2023.

[5] Arnab Bhattacharya, Thiagarajan Ramachandran, Sandeep Banik, Chase P Dowling, and Shaunak D Bopardikar. Automated adversary emulation for cyber-physical systems via reinforcement learning. In *2020 IEEE International Conference on Intelligence and Security Informatics (ISI)*, pages 1–6. IEEE, 2020.

[6] Arnab Bhattacharya, Soumya Vasisht, Veronica Adetola, Sen Huang, Himanshu Sharma, and Draguna L. Vrabie. Control co-design of commercial building chiller plant using bayesian optimization. *Energy and Buildings*, 246:111077, 2021.

[7] Arnab Bhattacharya, Soumya Vasisht, Veronica Adetola, Sen Huang, Himanshu Sharma, and Draguna L Vrabie. Control co-design of commercial building chiller plant using bayesian optimization. *Energy and Buildings*, 246:111077, 2021.

[8] Bishnu Bhattarai, Laurentiu Marinovici, Francis Tuffner, Kevin Schneider, Xiaoyuan Fan, Frederick Rutz, and Gowtham Kandaperumal. Prototypical communication systems for electrical distribution system analysis: Design basis and exemplification through co-simulation. *IET Smart Grid*, 5(5):363–379, 2022.

[9] Samuel S Booth, James Reilly, Robert S Butt, Mick Wasco, and Randy Monohan. Microgrids for energy resilience: A guide to conceptual design and lessons from defense projects. Technical report, National Renewable Energy Laboratory (NREL), Golden, CO (United States): US, 2020.

[10] Luca Carlone and Carlo Pinciroli. Robot co-design: beyond the monotone case. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 3024–3030. IEEE, 2019.

[11] Yixing Chen, Chuhao Yang, Xiao Pan, and Da Yan. Design and operation optimization of multi-chiller plants based on energy performance simulation. *Energy and Buildings*, 222:110100, 2020.

[12] Mauro Conti, Nicola Dragoni, and Viktor Lesyk. A survey of man in the middle attacks. *IEEE Communications Surveys and Tutorials*, 18(3):2027–2051, 2016.

[13] Joe Deese, Nihar Deodhar, and Chris Vermillion. Nested plant/controller co-design using g-optimal design and extremum seeking: Theoretical framework and application to an airborne wind energy system**this work was supported by nsf grant number 1453912, entitled career: Efficient experimental optimization for high performance airborne wind energy systems. *IFAC-PapersOnLine*, 50(1):11965–11971, 2017. 20th IFAC World Congress.

[14] Donald J. Docimo, Ziliang Kang, Kai A. James, and Andrew G. Alleyne. Plant and controller optimization for power and energy systems with model predictive control. *Journal of Dynamic Systems, Measurement, and Control*, 143(8), apr 2021.

[15] Dave Dykstra. Apptainer without setuid. In *EPJ Web of Conferences*, volume 295, page 07005. EDP Sciences, 2024.

[16] Etimad Fadel, Vehbi C Gungor, Laila Nassef, Nadine Akkari, MG Abbas Malik, Suleiman Almasri, and Ian F Akyildiz. A survey on wireless sensor networks for smart grid. *Computer Communications*, 71:22–33, 2015.

[17] Hosam K Fathy, Julie A Reyer, Panos Y Papalambros, and AG Ulsov. On the coupling between the plant and controller optimization problems. In *Proceedings of the 2001 American Control Conference.(Cat. No. 01CH37148)*, volume 3, pages 1864–1869. IEEE, 2001.

[18] Mario Garcia-Sanz. Control co-design: an engineering game changer. *Advanced Control for Applications: Engineering and Industrial Systems*, 1(1):e18, 2019.

[19] Djamal Eddine Ghersi, Meriem Amoura, Khaled Loubar, Umberto Desideri, and Mohand Tazerout. Multi-objective optimization of cchp system with hybrid chiller under new electric load following operation strategy. *Energy*, 219:119574, 2021.

[20] Trevor D. Hardy, Bryan Palmintier, Philip L. Top, Dheepak Krishnamurthy, and Jason C. Fuller. Helics: A co-simulation framework for scalable multi-domain modeling and analysis. *IEEE Access*, 12:24325–24347, 2024.

[21] Fahad Khan, Muhammad Abu Bakar Siddiqui, Ateeq Ur Rehman, Jawad Khan, Muhammad Tariq Sadiq Adeel Asad, and Adeel Asad. Iot based power monitoring system for smart grid applications. In *2020 International Conference on Engineering and Emerging Technologies (ICEET)*, pages 1–5. IEEE, 2020.

[22] Kyung-Bin Kwon, Ramij Raja Hossain, Sayak Mukherjee, Kaustav Chatterjee, Soumya Kundu, Sameer Nekkalapu, and Marcelo Elizondo. Coherency-aware learning control of inverter-dominated grids: A distributed risk-constrained approach. *IEEE Control Systems Letters*, 2024.

[23] Kyung-bin Kwon, Lintao Ye, Vijay Gupta, and Hao Zhu. Model-free learning for risk-constrained linear quadratic regulator with structured feedback in networked systems. In *2022 IEEE 61st Conference on Decision and Control (CDC)*, pages 7260–7265. IEEE, 2022.

[24] Bob Lasseter. Microgrids [distributed power generation]. In *2001 IEEE power engineering society winter meeting. Conference proceedings (Cat. No. 01CH37194)*, volume 1, pages 146–149. IEEE, 2001.

[25] Sai Pushpak Nandanoori, Alok Kumar Bharati, Subhrajit Sinha, Avijit Das, Soumya Kundu, Veronica Adetola, and Kevin Paul Schneider. Empowering the grid: Decentralized autonomous control for effective utilization and resilience. *Under Review*, 2024.

[26] Diane L Peters. *Coupling and controllability in optimal design and control*. University of Michigan, 2010.

[27] Thiagarajan Ramachandran, Soumya Vasisht, Aowabin Rahman, Arnab Bhattacharya, and Veronica Adetola. A computational framework for control co-design of resilient cyber–physical systems with applications to microgrids. *IEEE Transactions on Control Systems Technology*, 2023.

[28] Julie A Reyer, Hosam K Fathy, Panos Y Papalambros, and A Galip Ulsoy. Comparison of combined embodiment design and control optimization strategies using optimality conditions. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, volume 80234, pages 1023–1032. American Society of Mechanical Engineers, 2001.

[29] Julie A Reyer and Panos Y Papalambros. Combined optimal design and control with application to an electric dc motor. *J. Mech. Des.*, 124(2):183–191, 2002.

[30] George F Riley and Thomas R Henderson. The ns-3 network simulator. In *Modeling and tools for network simulation*, pages 15–34. Springer, 2010.

[31] Kevin P Schneider, Craig Miller, Stuart Laval, Wei Du, and Dan Ton. Networked microgrid operations: Supporting a resilient electric power infrastructure. *IEEE Electrification Magazine*, 8(4):70–79, 2020.

[32] Kevin Paul Schneider, Xueqing Sun, and Frank Tuffner. Adaptive load shedding as part of primary frequency response to support networked microgrid operations. *IEEE Transactions on Power Systems*, 39(1):287–298, 2023.

[33] Ankit Singhal, Thanh Long Vu, and Wei Du. Consensus control for coordinating grid-forming and grid-following inverters in microgrids. *IEEE Transactions on Smart Grid*, 2022.

[34] Cheng Wang, Christopher Redino, Ryan Clark, Abdul Rahman, Sal Aguinaga, Sathvik Murli, Dhruv Nandakumar, Roland Rao, Lanxiao Huang, Daniel Radke, et al. Leveraging reinforcement learning in red teaming for advanced ransomware attack simulations. *arXiv preprint arXiv:2406.17576*, 2024.

[35] Tarek Youssef. Co-design of security aware power system distribution architecture as cyber physical system. 2017.

[36] Gioele Zardini, Nicolas Lanzetti, Mauro Salazar, Andrea Censi, Emilio Frazzoli, and Marco Pavone. Towards a co-design framework for future mobility systems. *arXiv preprint arXiv:1910.07714*, 2019.

Pacific Northwest National Laboratory

902 Battelle Boulevard
P.O. Box 999
Richland, WA 99352
1-888-375-PNNL (7675)

www.pnnl.gov