Cyber Range for Industrial Control Systems (CR-ICS) for Simulating Attack Scenarios

Shaharyar Khan[1], Alberto Volpatto[2], Geet Kalra[1], Jonathan Esteban[1], Tommaso Pescanoce[2], Sabino Caporusso[2] & Michael Siegel[1]

[1] *Sloan School of Management, Massachusetts Institute of Technology*
[2] *BV TECH S.p.A.*

Abstract

Cyberattacks targeting industrial control systems (ICS) pose a particularly serious threat due to their potential to cause not only physical damage but also cascading disruptions to the supply of critical services (such as water, electricity, or gas). One way to address these threats is through training in a cyber range. Such training can bolster defensive capabilities by increasing cross-domain knowledge between IT and OT teams about real-world industrial processes and equipment on the one hand and attacker tactics, techniques, and procedures (TTPs) and cyber defense tools on the other hand. To that end, this paper presents the development of a Cyber Range for ICS (CR-ICS) that is based on a real-time attacker-defender gameplay model in conjunction with dynamic simulation models of typical industrial systems. As a proof of concept, we present an industrial gas turbine as one use-case of an archetypal industrial system. In addition to the architecture of the range and the building of the simulation model, this paper also provides a demonstration of a sample training exercise.

Keywords

Cyber Range, Industrial Control Systems, Simulation & Modeling, Gas Turbines

1. Introduction

Recent events, such as the attack on a Florida town's water treatment facility [1], have reignited fears about the vulnerability of our critical infrastructure to cyberattacks. Although this particular attack on the water treatment facility failed to cause any significant damage due to its early detection by an on-call operator, the attack serves as a wake-up call about the gravity and urgency of preparing for such threats. The potential for physical damage by remote manipulation of industrial control systems (ICS) is not a new phenomenon; in fact, as early as the year 2000, the remote attack on the Maroochy Shire sewage plant demonstrated the vulnerability of modern industrial systems and the destructive capability of such attacks [2].

However, for the most part, industrial systems (power grids, gas and water distribution plants, subway systems, etc.) have been considered relatively safe from cyberattacks because they have been traditionally segregated from the public internet i.e., *air-gapped*. With the advances in technology and connectivity, the infusion of Industrial Internet-of-Things (IIoT), and the integration of industrial control systems with management dashboards and other IT systems, the traditional air-gapped protections are no longer sufficient to provide adequate threat management. Not only that, but the functions that were previously performed by simple, easy to understand electro-mechanical components are now almost entirely implemented in software – hidden behind hundreds of millions of lines of code. The result has been an explosion in the *complexity* of ICS with many components interacting in many indirect ways. This increase in *complexity* has significantly expanded ICS *attack surface* and has made it extremely difficult to analyze these systems for vulnerabilities, trustworthiness, and mission-assurance.

[*] ITASEC21, April 07–09, 2021 – Online
EMAIL: shkhan@mit.edu (A. 1); a.volpatto@bv-tech.it (A. 2); geet@mit.edu (A. 3); jesteban@mit.edu (A. 4); t.pescanoce@bv-tech.it (A. 5); s.caporusso@bv-tech.it (A. 6); msiegel@mit.edu (A. 7)

CEUR Workshop Proceedings (CEUR-WS.org)

In addition to the increase in *complexity*, there is a fundamental disconnect between IT and OT in terms of language, culture, knowledge, and priorities. For one, IT is most concerned about ensuring data *confidentiality* and *integrity* whereas OT's primary concern is *availability* of equipment to ensure safe operation. Second, there are differences in technology, protocols, and equipment between IT and OT – from IT's perspective, the attack surface is comprised of computers, servers, printers, and network switches; for OT, the attack surface consists of field devices including Programmable Logic Controllers (PLCs), Remote Terminal Units (RTUs), Human-Machine Interfaces (HMIs), Supervisory Control and Data Acquisition (SCADA) systems, process actuators, and sensors. Finally, and perhaps most importantly, the *physics* of the system plays a crucial role in determining the success of an attack in the OT world while in the IT domain, the consequence of a cyberattack is measured in terms of loss of data, system disruption, loss of privacy, or financial loss.

Coupled with the increase in complexity, the disconnect between IT and OT has exacerbated the knowledge and skills gap which has significantly expanded the attack surface. This expanded attack surface, combined with the demonstrated capabilities and motivations of advanced cyber adversaries, such as nation states, has made modern-day critical infrastructure ICS extremely vulnerable. One approach to addressing these critical challenges is to upskill the workforce by exposing operators and information security teams to realistic cyberattack scenarios, practicing detection and response strategies using real-world tools and techniques, and improving coordination between IT and OT teams. However, training on live equipment is not only extremely costly but also inherently unsafe.

In this paper, we present the development of a next generation training platform for joint IT/OT cybersecurity training with a goal to predict, detect, and respond to cyberattacks. Our goal is to follow the 80/20 rule and disseminate cross-domain knowledge between IT and OT security professionals to as large a population as possible to improve cyber resilience of ICS. This training platform, referred to as a CR-ICS, combines a virtualized IT/OT network with simulation models of industrial processes developed in *Anylogic* [3], providing a safe, isolated approach for training without impacting business operations. The joint IT/OT training platform has three major objectives:

1. Enhance cyber awareness for OT operators; this includes increasing awareness about attacker tactics, techniques, and procedures (TTPs) in order to detect and respond to cyberattacks
2. Increase ICS awareness (equipment, protocols) for IT operators; this includes understanding system operation, interdependence of physical processes, and visualizing the effect of attacks
3. Bring IT and OT teams together to improve cyber resilience

We begin by providing a storyboard of a sample training exercise and expected learning outcomes using CR-ICS in Section 2 before describing the state-of-the-art for such training platforms in Section 3. This is followed by a description of the architecture of CR-ICS in Section 4. Next, we describe the development of an ICS simulation model along with its integration with the IT cyber range in Section 5. Finally, we preview future lines of research that would integrate with the training platform and augment its capability in Section 6.

2. Storyboard of a Training Exercise

The development of a CR-ICS simulation enables the extension of traditional IT cyberattack scenarios to include OT systems and protocols. Primarily, CR-ICS aids in developing an intuition about the *physics* of the system by providing a visualization of effects of attacker actions on industrial equipment and processes. It also provides Security Operations Center (SOC) operators new events based on industrial protocols to detect and respond to cyberattacks.

A high-level diagram of the simulated environment is presented in Figure 1. Note that the simulated environment consists of a traditional corporate IT network as well as an OT network. Specifically, the network consists of the following key components:

1. **External firewall** exposes corporate services to the external network
2. **DMZ** or demilitarized zone contains and exposes the company's external-facing services (such as data historian) to the untrusted public internet.

3. **Server network** hosts traditional IT systems (e.g., domain controller) and services
4. **SOC network** hosts workstations used by the blue team (defender team)
5. **SCADA network** hosts operator workstation and SCADA monitoring application
6. **OT network** consists of Modbus TCP servers which interact with the *Anylogic* simulation; note that the OT network is not routed to other networks

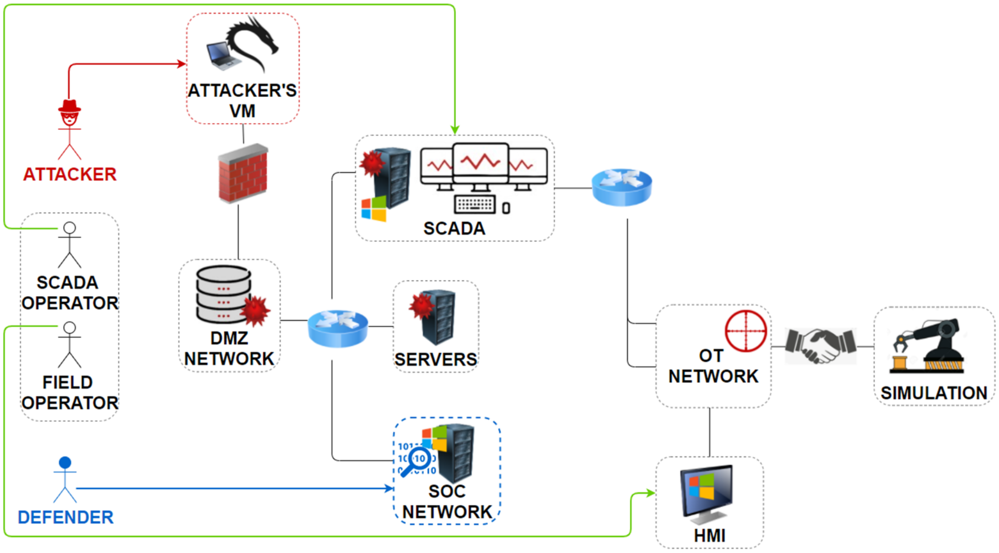


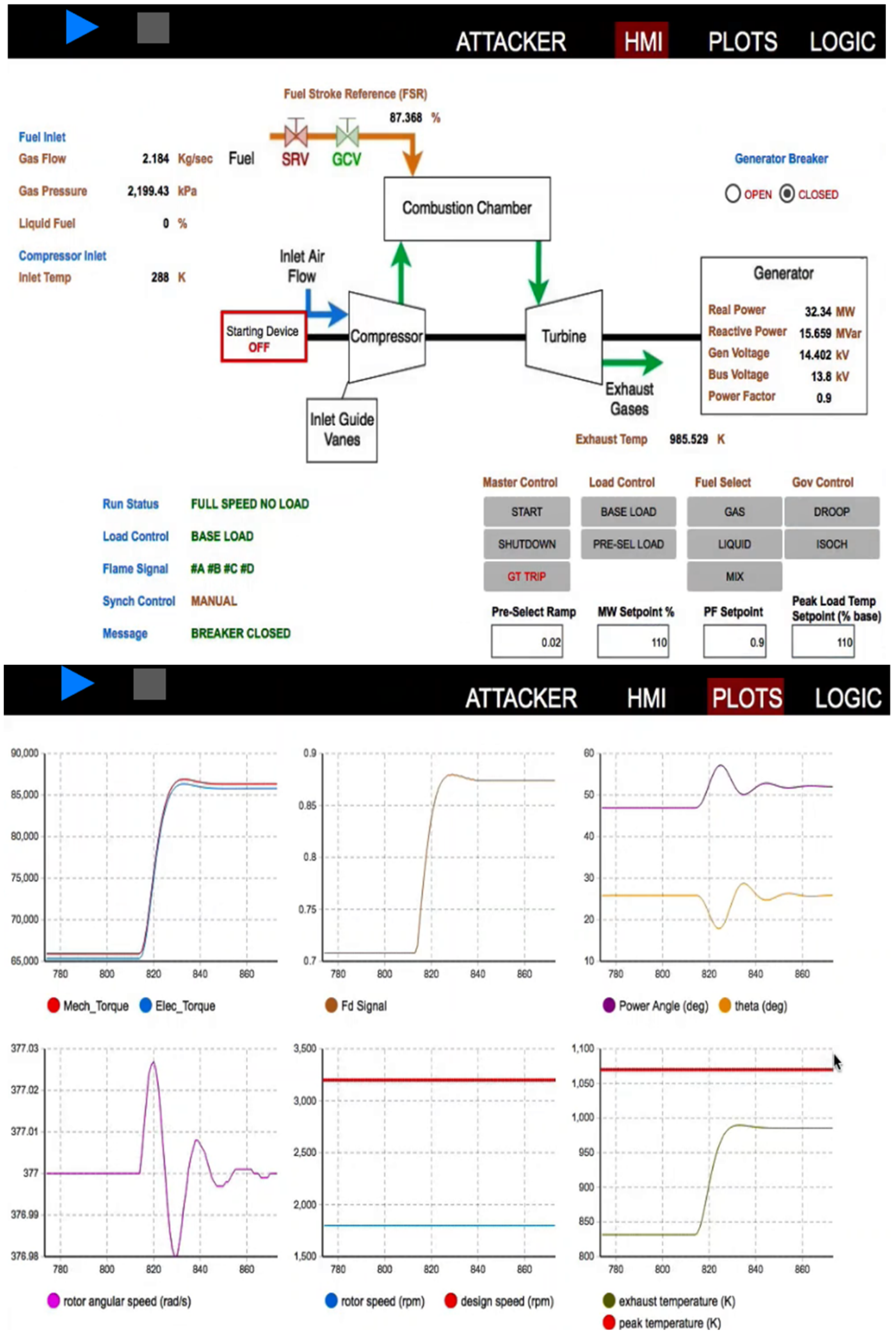
Figure 1: High-level representation of simulated environment

At the onset of the training exercise, the attacker has access to a standard *Kali Linux* virtual machine connected to the external network. The goal for the attacker is to detect and exploit vulnerabilities (that have been deliberately injected in selected systems in the network) in order to pivot to the OT network and interact with the Modbus TCP services. The expected attack steps are as follows:

1. The attacker detects an exposed service affected by a Remote Command Execution vulnerability. Successful exploitation leads to a compromised server.
2. From the server network, the attacker discovers the SCADA network and enumerates systems and services. The attacker exploits a system vulnerability gaining shell access to the host that can interact with the devices in the OT network.
3. The attacker detects the Modbus TCP server and performs information gathering activities to enumerate all available functions.
4. The attacker sends arbitrary Modbus TCP requests leading to system changes in the simulated *Anylogic* model.

For the purpose of this exercise, we utilize a gas turbine model that was developed for CR-ICS. Although this model is a fully functional dynamic model, we will discuss two specific scenarios that can be executed remotely by the attacker after gaining access to the OT network. Figure 2 shows the HMI screen for the gas turbine along with plots of some parameters that are of interest to the operator.

The first scenario is that of a remote command by an attacker to operate at peak load. Ordinarily, gas turbines operate at what is known as base load i.e., 100% power. However, some gas turbines have the additional capability to operate at higher than nominal base load temperatures for short periods of time to increase load and take advantage of higher electricity prices. This capability is known as *peak firing*. For short periods of time, peak firing is acceptable and within the design constraints of the gas turbine. However, if the gas turbine operates for extended periods of time at the peak load temperature, it can reduce its life and cause permanent damage [4]. An attacker may remotely target this functionality after gaining control of the OT network and operate the gas turbine at an elevated temperature unbeknownst to the operator. As shown in the plots in Figure 2, such an attack would cause not only the power output to change (which the attacker can realistically falsify on the HMI), but also several interdependent parameters would change accordingly. An operator may, however, still be able to detect the cyber intrusion by observing the changes in other parameters, such as the power angle, the gas control valve position (F_d), and the exhaust temperature curve, even if the power output is obfuscated or falsified.



Fuel Stroke Reference (FSR) 87.368 %

Fuel Inlet
Gas Flow 2.184 Kg/sec
Gas Pressure 2,199.43 kPa
Liquid Fuel 0 %

Compressor Inlet
Inlet Temp 288 K

Starting Device OFF

Generator Breaker ○ OPEN ● CLOSED

Real Power 32.34 MW
Reactive Power 15.659 MVar
Gen Voltage 14.402 kV
Bus Voltage 13.8 kV
Power Factor 0.9

Exhaust Temp 985.529 K

Run Status FULL SPEED NO LOAD
Load Control BASE LOAD
Flame Signal #A #B #C #D
Synch Control MANUAL
Message BREAKER CLOSED

Pre-Select Ramp 0.02
MW Setpoint % 110
PF Setpoint 0.9
Peak Load Temp Setpoint (% base) 110

Figure 2: Gas turbine HMI and parameters of interest – Scenario #1

The second scenario is that of a catastrophic failure of the turbine due to over-speeding. Turbine over-speeding is a dangerous and costly condition which can occur if there is a sudden loss of load without the immediate shut down of the fuel control valve [5]. Under normal circumstances, the shutdown of a gas turbine follows a sequence of steps to ensure a controlled safe shutdown that does not adversely affect service life due to thermal fatigue. On the other hand, gas turbine *trips* are much more onerous for the gas turbine as they require a sudden decrease in speed of the turbine following a loss of load condition. In our scenario, the attacker issues a *trip* command when operating the turbine at peak load conditions. Under these conditions, even a slight delay in the protection systems response could result in catastrophic destruction of the gas turbine due to over speeding. Time histories for various parameters of the gas turbine for this scenario are presented in Figure 3. Note that the rotor speed plot in Figure 3 shows a sudden spike in speed following a *trip* command to the control system.

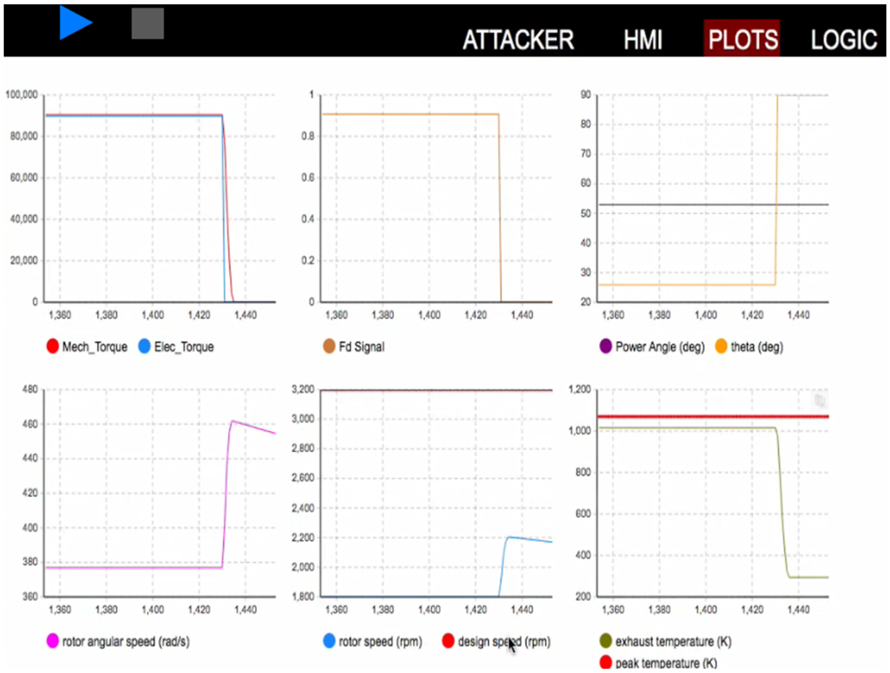


Figure 3: Simulation response due to trip command – Scenario #2

A variety of OT scenarios, similar to the ones illustrated above, can be simulated in CR-ICS. Importantly, note that practicing even with a limited number of training scenarios, several critical cybersecurity lessons can be learned by both IT and OT operators. For instance, OT operators can practice use of offensive and defensive tools, traditionally reserved for IT operators; meanwhile, IT operators can learn about OT protocols, equipment and processes and develop an intuition about the physics of the system. Next, we will explore the state-of-the-art in ICS cyber range development.

3. State-of-the-Art

Several instances of cyber ranges exist in the literature. We focus on cyber-ranges (alternatively called test-beds) built for industrial environments. One way to classify cyber ranges is to categorize them based on their composition. This type of classification consists of four main categories – *replication* (hardware and software infrastructure as found in industry), *simulation*, *virtualization,* and *hybrid* categories [6], [7].

The benefit of building a cyber range based on *physical replication* is that actual ICS hardware and software is used which helps to recreate an environment that is truly representative of real-life processes. However, the high cost associated with setting up and operating such a range is prohibitive for most players except for governmental bodies. A good example of such a cyber range is Idaho National Lab's *"61 mile, 138kV dual-fed power loop complete with seven substations and a control center"* electric grid test-bed which is used for cyber and physical testing of grid infrastructure [8]. Some universities also have hardware-based testbeds, albeit at a much smaller scale. For instance, the iTrust center at Singapore University of Technology and Design has setup a six-stage water treatment testbed, capable of producing 5 gallons/hour of filtered water [9].

An alternative to *physical replication* is to employ software-based *simulation* to develop a sandboxed environment that provides similar functions and behaviors of an actual industrial system. Simulation based cyber ranges offer several advantages including ease of reconfigurability, maintainability, and scalability. However, they do not provide high fidelity, especially when software exploits and cyberattacks need to be considered [6]. SCADASim [10] is an example of a software-based simulation cyber range.

The third class of cyber ranges found in the literature consists of *virtualization* of test beds. Virtualization allows for modeling of different components of the ICS environment as independent entities where the entities communicate with each other using actual communication protocols used in industry (such as Modbus, DNP3 etc.) [6]. Researchers proposed a modular approach for virtualization of a SCADA system [11]. In this approach, Simulink is used to create models of physical processes; the models are then connected to the virtual network where PLC's, HMI, and SCADA are virtualized using open-source software (e.g., OpenPLC and ScadaBR). Various industrial environments are modeled, such as a water tank system and a gas pipeline system. The virtualized environments are subject to various types of attacks such as ARP spoofing, MiTM attacks, etc. In addition, *Hussein et al.* [12] developed a virtual testbed that enables in-depth study of communication protocols and control authority concepts to validate false data injection attacks and their impact on power systems.

Recent trends in cyber range development include the building of *hybrid* cyber ranges. *Hybrid* cyber ranges combine *simulation*, *virtualization,* and physical device *replication* approaches in a single hybrid experiment [7]. Such hybrid approaches offer the promise to overcome the disadvantages associated with the other types of cyber ranges. Of note is the development of a small-scale hardware-in-the-loop (HIL) ICS testbed which is an example of a *hybrid* cyber range [13].

In addition to cyber ranges built by academic institutions and government labs, we also see development of cyber ranges in the private sector. For instance, *Accenture* has a cyber range in Houston where they provide end-to-end infrastructure for a typical ICS. This range includes equipment such as wellheads and pumps at a small scale [14]. Likewise, *Airbus* has developed an OT simulation platform which links to a cyber range for cybersecurity testing. It provides the ability to model realistic network data and employs common industrial protocols including Modbus, Profibus, DNP3, etc. [14]. Startups such as *Dragos* [14] and *Fortiphyd* [15] which provide a variety of OT related cybersecurity services have also developed small-scale ICS cyber ranges.

The CR-ICS cyber range that we present in this paper is a joint collaboration between industry and academia and can be classified as a *hybrid* cyber range. It consists of a *virtualized* IT/OT network that is integrated with a *simulation* of the physical industrial process and is intended to offer HIL capability in the future. In this work, we take a complex systems modeling approach to build the ICS simulation.

Modeling real-world *complex* systems is a challenging task. Complex systems are characterized not only by the number of components but also by the diversity of those components and their interactions and interdependencies. In these systems, the components produce *emergent* effects that are not easily predictable [16].

For most *industrial systems* (such as gas turbines, industrial chillers, pumps, etc.)*, Matlab* is the most widely used simulation program for modeling control system behavior; other commonly used programs include Maple, Mathematica, Octave, Scilab, etc. [17]. In each of these programs, numerical methods are applied to solve differential equations describing the *physics* of the system; this is known as *dynamic system modeling*. Such an approach is extremely powerful in optimizing control parameters for a dynamic model and is best suited to *'hierarchically controlled one-way systems'* [17].

However, real-world engineered cyber-physical systems are not limited to one-way hierarchical control. In fact, they are *socio-technical* systems where the technical system interacts with human behavior. In order to capture the dynamics of such a system holistically, we need to add human behavior models to the technical model; for this, dynamic system modeling approach in isolation is not adequate. In fact, what is a required is a multi-method approach that allows simulation of hybrid models allowing for discrete event, agent-based behavior as well as whole system dynamics simulations using the same model.

The *Anylogic* program [18] meets these requirements – it supports modeling using *system dynamics, discrete events,* and *agent-based models* as well as hybrid models containing any combination of these different modeling approaches in the same simulation. Although fairly well-known in the community of multi-approach simulations, *Anylogic* is not as well known in the areas of automation and control engineering [17]. The most relevant work is that by *Kremers* [17] where an energy system is modeled through a complex systems approach using an agent-based simulation model. *Kremers* [17] also provides a method to use a system dynamics approach to determine a numerical solution to a differential equation. Another related work is that by *El-Sefy et al.* [19] where the thermal dynamic processes in nuclear power plants are simulated using a system dynamics approach in *Anylogic*. Finally, *Mahmood et al.* [20] present an integrated modeling, simulation, and analysis approach for engineering complex systems and use a real-time adaptive cruise control system as a case study. To the best of our knowledge, our model is the first gas turbine simulation model implemented in Anylogic. Before providing details about the gas turbine model, we will first outline the architecture of CR-ICS in the next section.

4. CR-ICS Architecture

A cyber range aims at challenging trainees with real-world systems and scenarios. Therefore, software, protocols, and operating systems along with network diagrams and architecture are chosen to be representative of the target context. When architecting CR-ICS, we defined the following parameters as mandatory:

- **Sandboxed environments** – the underlying infrastructure and physical environment will provide a safe sandboxed environment to users. Due to the nature of the simulated environments where multiple vulnerable systems are deployed and attackers are instructed to abuse them with real techniques and working exploits, keeping the underlying infrastructure and physical network *safe* is mandatory.
- **Ease of access** – no special hardware or software shall be required to use CR-ICS. While providing a dedicated hardware (e.g., physical workstation) or software (e.g., VPN client) to access the environment is a common choice, doing so fails to provide a user-friendly experience in addition to introducing other challenges. For example, allowing users to work with their own laptops with a tunneled connection to the simulated environment could not only introduce security risks for their systems, but also make the monitoring process much harder.
- **Independent deployments** – the platform shall allow deployment of multiple environments guaranteeing full isolation between them.
- **Network mirroring** – the platform shall have the capability to define virtual span ports to mirror the virtual traffic. Mirror (or span) ports are a common feature of network switches to replicate the traffic transmitted on selected ports to a passive port mainly for feeding a network analyzer system (e.g., IDS). This capability is crucial for deploying real-world virtual network probes.

While the main public cloud providers offer different solutions to satisfy our requirements, we decided to work with an on-premise installation on top of a cluster of KVM hypervisors (i.e., Proxmox Virtual Environment). Figure 4 shows the high-level architecture of CR-ICS where we opted to split the system into services responsible for managing the platform and providing user access (*control plane*), and services that are responsible for executing the simulated environments (*execution plane*).

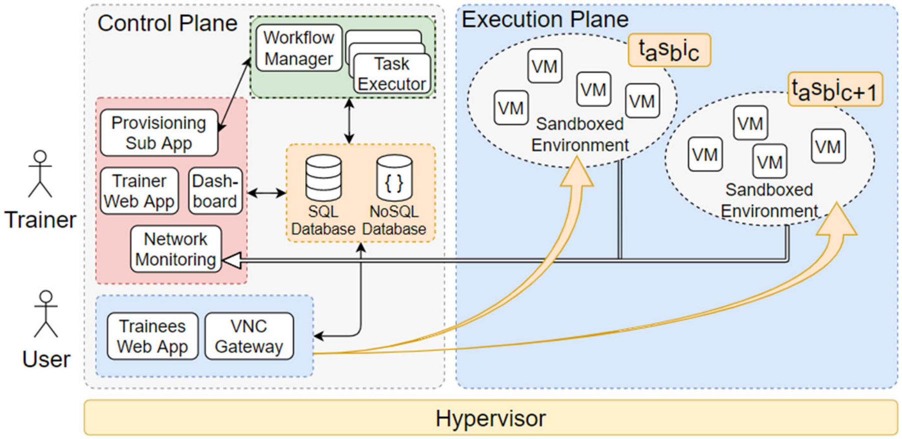


Figure 4: General Architecture of CR-ICS

Any instance of a simulated environment is defined by the triple $t_as_bi_c$, defined as:

- **t_a (theater *a*)**- *theater* defines the simulated infrastructure to be deployed in CR-ICS in terms of network topology, nodes, and services. It is the *context* within which the session is performed.

- ***s_b* (scenario *b*)**- *scenario* defines the actual *customization* of the selected theater for delivering a specific session. Differences in terms of positioning of attackers' nodes, configuration of services, and deployed resources may range from small configuration details (e.g., password policy) to more significant ones (e.g., internal threats, vulnerable software, etc.). A theater can have multiple scenarios (*1 to n*).
- ***i_c* (instance *c*)**– *instance* defines the deployment of a given scenario. Multiple instances of the same scenario are supported (*1 to n*).

Users access the platform through a web application whose features and contents change depending on the user's role. We defined three main types of users:

- **Red user** – plays the role of an *attacker* against the simulated environment. Only access to a dedicated Kali Linux virtual machine is initially granted, without restrictions on the actions that can be performed on it.
- **Blue user** – plays the role of a *defender* of the corporate system and reacts to the attacks performed by *red* users. Access to a dedicated domain joined virtual workstation is granted, from where the user can access all the security systems, services, and applications provided, such as IDS, IPS, and SIEM.
- **White user (trainer)** – plays the role of a *supervisor* of the session; manages the lifecycle of the environment, monitors the actions performed by both teams, and provides training to them. The *white user* does not interact directly with the simulated environment.

Red and blue users access the dedicated virtual machine (i.e., Kali Linux or workstation, respectively) via a web console leveraging the Virtual Network Computing (VNC) server provided by the hypervisor without requiring any additional software. No outbound traffic is allowed from the instances; users interact with the virtual environment directly from the dedicated virtual machine. Any other type of user involved in the training, such as SCADA operators, access the platform in the same way except that instead of using a virtual workstation connected to the SOC network, the *client* is a workstation belonging to the SCADA control network (i.e., operator workstation).

Virtual networks rely on OpenVSwitch which supports IPv4, IPv6, and 802.1Q standard, and connection of virtual environments with physical devices. While it is possible to interconnect OT devices (e.g., PLC) to the sandboxed environment to allow users to act on actual industrial components, machinery, and plants, it is unlikely to be connected in this way for many reasons, in particular safety.

To offer users the capability to interact with the simulated model using standard industrial protocols, we developed a Modbus TCP server in Python which acts as a middleware between the users and the *Anylogic* simulation, exposing a RESTful API on a dedicated and isolated network. More information about the interconnection of the two components is provided in the next Chapter.

5. Development of CR-ICS Simulation

As mentioned earlier, one of the key features of CR-ICS is the addition of a *dynamic interactive simulation* of an industrial process to the virtualized IT/OT cyber range network. The simulation enables prediction and visualization of the control system's response to malicious actions including its interactions with other interdependent systems and the environment. The following subsections present the implementation of the simulation model in *Anylogic* along with its connection to the cyber range. Although CR-ICS is intended to be *industrial process* agnostic (i.e., different industrial processes can be modeled and included in the range), we use an *industrial gas turbine* as a substantive example to illustrate the method used in building the model.

5.1 Simulator for an Industrial Gas Turbine

Building a simulation model of a gas turbine (or other industrial processes) for integration with a cyber range is a complex task because of the presence of non-linearities (such as discrete events), time delays, and emergent phenomena resulting from interaction of the various subcomponents and their autonomous controllers. As a proof of concept, we demonstrate an implementation of a *one-way* *hierarchical model* of a gas turbine in *Anylogic*. We begin by providing a description of the gas turbine and its control strategy. Further details can be found in various gas turbine reference manuals [21].

A gas turbine is an internal combustion engine, designed to accelerate a stream of gas to produce mechanical power to turn a load such as a synchronous generator that is coupled to the turbine shaft. The synchronous generator converts the rotational energy supplied by the turbine to 3-phase electrical power. Figure 5 presents a high-level view of the operation of a gas turbine.

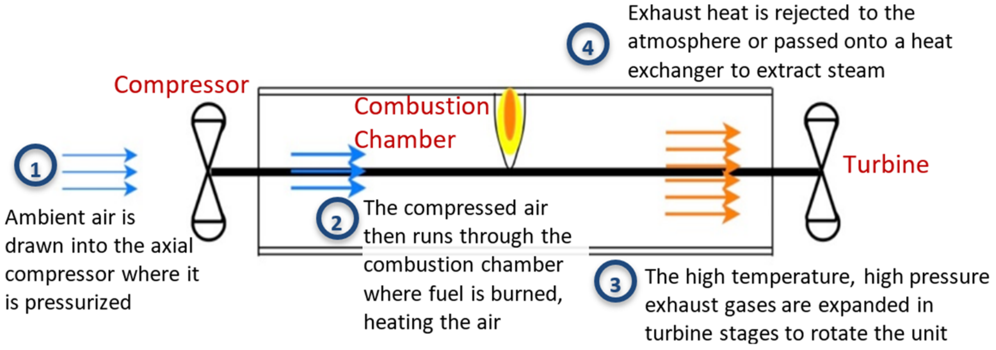

Exhaust heat is rejected to the atmosphere or passed onto a heat exchanger to extract steam
4
Compressor
Combustion Chamber
1
Turbine
Ambient air is drawn into the axial compressor where it is pressurized
2
The compressed air then runs through the combustion chamber where fuel is burned, heating the air
3
The high temperature, high pressure exhaust gases are expanded in turbine stages to rotate the unit

Figure 5: Simplified Diagram of a Gas Turbine

The primary control of the gas turbine is achieved by controlling the fuel input into the combustion chamber; as the amount of fuel injected into the combustion chamber is increased, the power output of the turbine increases. The amount of fuel that is injected into the combustion chamber is determined by the turbine controller based on feedback from various sensors. At any point in time, the fuel control valve is under control of one of three primary control loops: 1) power output control, 2) speed or frequency control, or 3) temperature control. Each control loop calculates a value for the fuel signal and passes it onto a logic gate which selects the lowest of the three values; this ensures the speed and exhaust gas temperature safety constraints are not exceeded. The fuel signal is then passed onto another logic gate which ensures that the lower limit for fuel flow is not violated which can cause flameout (a dangerous condition which can result in an explosion). The exhaust gas temperature is controlled by adjusting the input air flow; in certain configurations (such as Combined cycle) a proportional controller is used to ensure the exhaust temperature is as high as possible to maximize energy flow to downstream boilers [21], [22].

Figure A. 1 shows the simplified control strategy for the gas turbine as described above while Figure A. 2 shows its implementation in *Anylogic*. Note that we use a system dynamics modeling approach to model a Proportional-Integral-Derivative (PID) controller and nest it as an *agent* i.e., an autonomous entity. This enables reuse of the PID controller in the model by supplying it with the PID parameters along with a *normalized setpoint* and a *normalized feedback signal*. The *PID controller* calculates a *control signal* which is then passed on to the gas turbine model.

The gas turbine model consists of five main components as shown in Figure A. 3; these include the *fuel system, compressor, combustor, turbine,* and the *generator*. The *compressor*, *combustor,* and *turbine* are modeled as steady-state systems. Simple algebraic expressions, as presented by Ordys et al. [22] (pp. 153 to 156), are used to simulate the behavior of these components at steady-state, 100% power. The behavior of these components at any transient state is estimated using linear interpolation. The *fuel system* is modeled as a second-order system to campture the time delay dynamics of the valve and the actuator.

The *synchronous generator* is also modeled following the approach used by Ordys et al. [22]. Three first-order differential equations are used to capture the dynamics of *1) the change in shaft speed, 2) the change in power angle,* and *3) the transient voltage*. The generator's voltage (when operating in standalone mode) or reactive power output (when synchronized with the grid) is controlled by the *Automatic Voltage Regulator* (AVR) which is modeled using a *PID controller*. The generator equations are listed by Ordys et al. [22] (pp. 191-200).

The *Anylogic* model is validated by comparing the calculated values with the values presented by Ordys et al. [22] at 100% power, steady state conditions. The inner details of the simulation model, as presented above, are hidden from the users of CR-ICS. The user only sees the key controls for the gas turbine model laid out in the same format as that on a Human Machine Interface (HMI) that an operator in a real industrial plant would use to control the turbine.

In addition to using system dynamics, we employ state charts (Figure A. 3) to model *startup* and *shutdown* sequences as well as other discrete events that may be caused by attacker actions (such as lubrication pump shutoff, turbine trip command, etc.) which require a change in parameter value for the simulation. As a result, a variety of system behaviors (that are caused by attacker actions) can be easily predicted using the simulation model. These behaviors include interactions between components (such as between AVR and the turbine controller). For example, the simulation is capable of showing a sudden change in rotor speed (due to an imbalance in electrical and mechanical torque) as a result of a sudden loss of load. Likewise, it is also able to show turbine instability as a result of incorrect reactive power setpoints. By going through the training exercise and using the simulation model, the IT team can expand its understanding about the physics of the system while the OT team can practice its response strategies to emergent malicious conditions in a safe environment.

5.2 Connection to the Cyber Range

The connection of the simulation model to the cyber range is facilitated via a RESTful API. This allows for a standardized communication channel capable of handling multiple clients at once. Thus, the connection is agnostic to any client that would want to interact with the cyber range, on the condition that the client supports HTTP requests. In particular, the connection can handle POST requests that allow for a component's value to be modified and GET requests that allow for the retrieval of updates. These requests can be accepted from multiple clients from different addresses. Using the gas turbine as an example, a POST request could be made to modify the fuel valve value. Subsequently, a GET request could be sent to retrieve the power output of the turbine.

The API is structured based on the following requirements as summarized in Figure 6:

- There can be different simulated models running concurrently (e.g., a gas turbine model and a water tank simulation)
- Each model has its own distinct devices (e.g., gas turbine model has valves, pressure instruments, etc.) accessible via unique identifiers
- Each device has a series of parameters that can be retrieved or modified

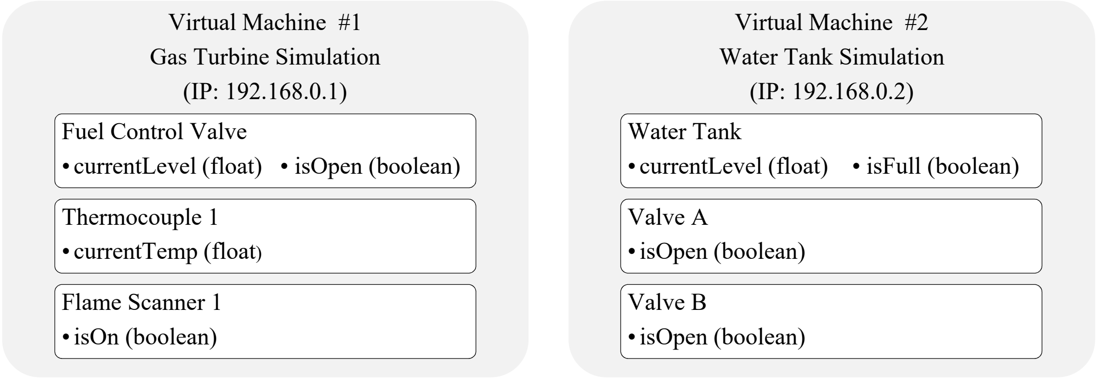


Figure 6: Layout of various simulation models

As such, any number of attacker-defender models can be simulated, each of which will be hosted in its own sandboxed virtual machine to keep each simulation's network disjointed from one another. One or more Modbus TCP servers are mapped to the sensors and/or actuators of the industrial model, translating the Modbus packet to a HTTP API request.

The sequence diagram depicted in Figure 7 provides an example of interaction between the Modbus TCP client and the *Anylogic* simulation; the Modbus request is parsed to determine which API call is mapped to the received function code and memory register.

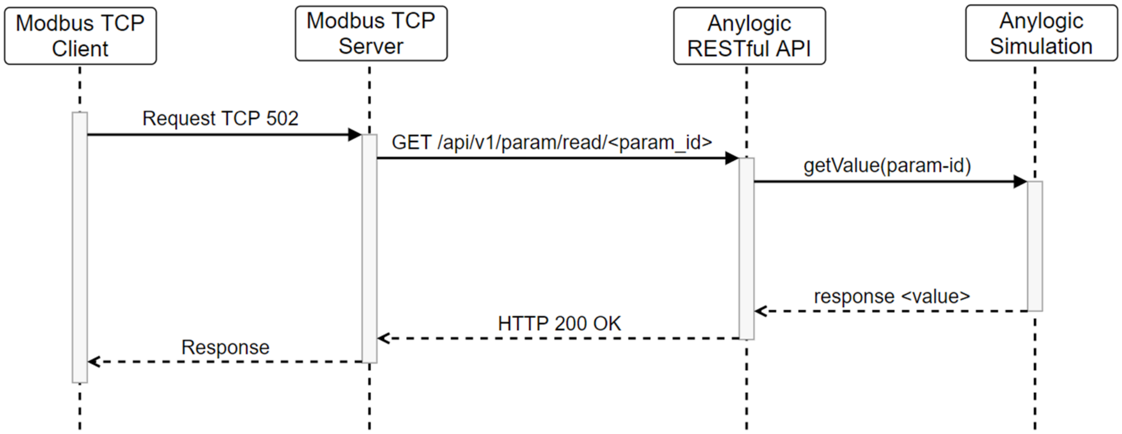


Figure 7: Example of interaction with the Anylogic model via Modbus TCP

6. Next Steps

The gas turbine model presented above can be used to demonstrate many different scenarios; however, the two simple scenarios presented earlier demonstrate the effectiveness of adding such a simulation to a traditional cyber range to bridge the gap between IT and OT and develop a joint cyber defense team.

For instance, in the scenarios presented above, the attacker team exploited vulnerabilities as they pivoted from one system to another. At each instance, the defender team had the opportunity to detect the intrusion using traditional IT tools. However, once the attackers navigated to the OT network and began to issue commands, in addition to network traffic, their actions impacted interdependent parameters of the physical system. If the operators were adequately cyber-trained, they would be able to identify the anomalous behavior by observing the system behavior and alert the information security team who can then zero in on the network traffic for those devices and thwart the attack while it is still in its early stages.

In the face of targeted attacks by advanced cyber-adversaries, we need to cross-train an army of people to defend our critical infrastructure ICS. The cyber range that we presented here would enable us to meet such an objective. As demonstrated above, it can enhance cyber awareness for OT operators, enable IT operators to visualize system behavior, develop an intuition about the physical system, and ultimately, bring IT and OT together to improve cyber resilience of our critical infrastructure industrial control systems.

As mentioned earlier, the addition of a *hierarchical one-way simulation model* to the cyber range is a first-step toward achieving our objectives for this cyber range. The use of system modeling tools (such as *Anylogic*) grants us the capability to model autonomous agents along with their interactive dynamic behavior, discrete events, and system dynamics in the same model. We intend to use this capability to realistically capture the complexity of real-world ICS. In addition to improving ICS simulation, we intend to integrate other research projects with CR-ICS including related work on anticipation of attacker moves using adversarial gameplay approach, AI/ML-based anomaly detection, and automatic scenario generation. Together, these projects will enable the creation of a next-generation training platform that would enhance ICS cyber resilience.

7. Acknowledgements

This work was co-funded by "Fondo Europeo di Sviluppo Regionale Puglia POR Puglia 2014 – 2020 – Asse I – Obiettivo specifico 1a – Azione 1.1 (R&S) - Titolo Progetto: Suite prodotti CyberSecurity e SOC" and BV TECH S.p.A.

8. References

[1] J. Peiser, "A hacker tried to poison the water supply in Oldsmar, Florida, police said - The Washington Post," *The Washington Post*, 2021.

[2] N. Sayfayn and S. Madnick, "Cybersafety Analysis of the Maroochy Shire Sewage Spill Cybersafety Analysis of the Maroochy Shire Sewage Spill (Preliminary Draft)," 2017.

[3] A. Borshchev and A. Filippov, "From System Dynamics and Discrete Event to Practical Agent Based Modeling: Reasons, Techniques, Tools – AnyLogic Simulation Software." [Online]. Available: https://www.anylogic.com/resources/articles/from-system-dynamics-and-discrete-event-to-practical-agent-based-modeling-reasons-techniques-tools/. [Accessed: 14-Feb-2021].

[4] J. Janawitz, J. Masso, and C. Childs, "GE Power & Water Heavy-Duty Gas Turbine Operating and Maintenance Considerations GER-3620M (02/15)."

[5] TMI Staff and Contributors, "Turbine overspeed testing ," *Turbomachinery Magazine*. [Online]. Available: https://www.turbomachinerymag.com/testing/. [Accessed: 02-Mar-2021].

[6] H. Holm, M. Karresand, A. Vidström, and E. Westring, "A survey of industrial control system testbeds," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2015, vol. 9417, pp. 11–26.

[7] Q. Qassim *et al.*, "A Survey of SCADA Testbed Implementation Approaches," *Indian J. Sci. Technol.*, vol. 10, no. 26, pp. 1–8, Jun. 2017.

[8] Idaho National Labs, "Securing the Electrical Grid from Cyber and Physical Threats." [Online]. Available: https://inl.gov/research-programs/grid-resilience/. [Accessed: 09-Mar-2021].

[9] A. P. Mathur and N. O. Tippenhauer, "SWaT: A water treatment testbed for research and training on ICS security," in *2016 International Workshop on Cyber-physical Systems for Smart Water Networks, CySWater 2016*, 2016, pp. 31–36.

[10] C. Queiroz, A. Mahmood, and Z. Tari, "SCADASimA framework for building SCADA simulations," *IEEE Trans. Smart Grid*, vol. 2, no. 4, pp. 589–597, Dec. 2011.

[11] T. Alves, R. Das, A. Werth, and T. Morris, "Virtualization of SCADA testbeds for cybersecurity research: A modular approach," *Comput. Secur.*, vol. 77, pp. 531–546, Aug. 2018.

[12] S. Hussain *et al.*, "A novel methodology to validate cyberattacks and evaluate their impact on power systems using real time digital simulation," pp. 1–6, 2021.

[13] T. Morris, R. Vaughn, and Y. S. Dandass, "A testbed for SCADA control system cybersecurity research and pedagogy," in *ACM International Conference Proceeding Series*, 2011, p. 1.

[14] Accenture, "Houston Cyber Fusion Center ." [Online]. Available: https://www.accenture.com/us-en/services/security/cyber-fusion-center-houston. [Accessed: 09-Mar-2021].

[15] Fortiphyd Logic, "Fortiphyd Training Grounds." [Online]. Available: https://fortiphyd.talentlms.com/index. [Accessed: 09-Mar-2021].

[16] R. B. Northrop, *Introduction to Complexity and Complex Systems - 1st Edition* . 2017.

[17] E. Kremers, *Modelling and Simulation of Electrical Energy Systems through a Complex Systems Approach using Agent-Based Models*. 2012.

[18] Anylogic, "AnyLogic: Simulation Modeling Software Tools & Solutions for Business." [Online]. Available: https://www.anylogic.com/. [Accessed: 15-Feb-2021].

[19] M. El-Sefy, M. Ezzeldin, W. El-Dakhakhni, L. Wiebe, and S. Nagasaki, "System dynamics simulation of the thermal dynamic processes in nuclear power plants," *Nucl. Eng. Technol.*, vol. 51, no. 6, pp. 1540–1553, Sep. 2019.

[20] I. Mahmood, T. Kausar, H. S. Sarjoughian, A. W. Malik, and N. Riaz, "An Integrated Modeling, Simulation and Analysis Framework for Engineering Complex Systems," *IEEE Access*, vol. 7, pp. 67497–67514, 2019.

[21] D. Johnson, W. Rowen, W. Coes, and H. Sechrist, "General Electric Speedtronic Control Systems ," *GE Gas Turbine Reference Library*. [Online]. Available: https://www.ge.com/content/dam/gepower-pgdp/global/en_US/documents/technical/ger/ger-3107-speedtronic-mark-ii.pdf. [Accessed: 17-Feb-2021].

[22] A. W. Ordys, A. W. Pike, M. A. Johnson, R. M. Katebi, and M. J. Grimble, *Modelling and Simulation of Power Generation Plants*. London: Springer London, 1994.

Appendix A – CR-ICS Figures

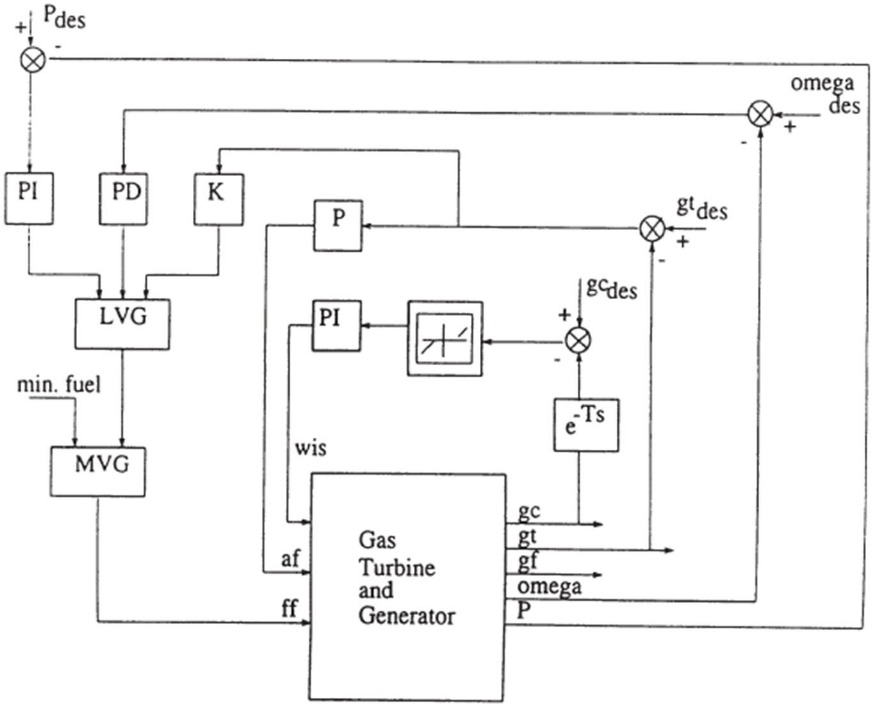


Figure A. 1: Simplified gas turbine control strategy

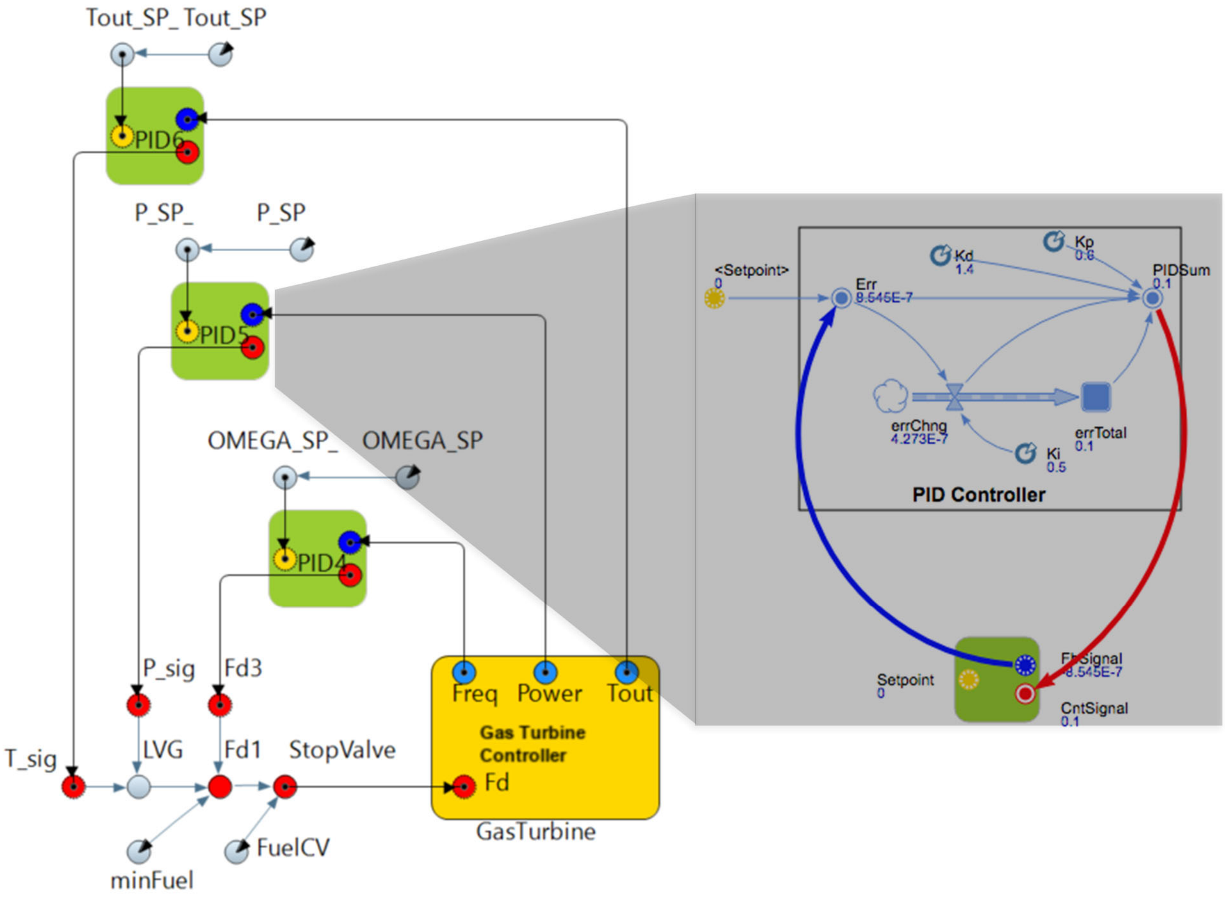


Figure A. 2: Overall control strategy for the gas turbine implemented in Anylogic

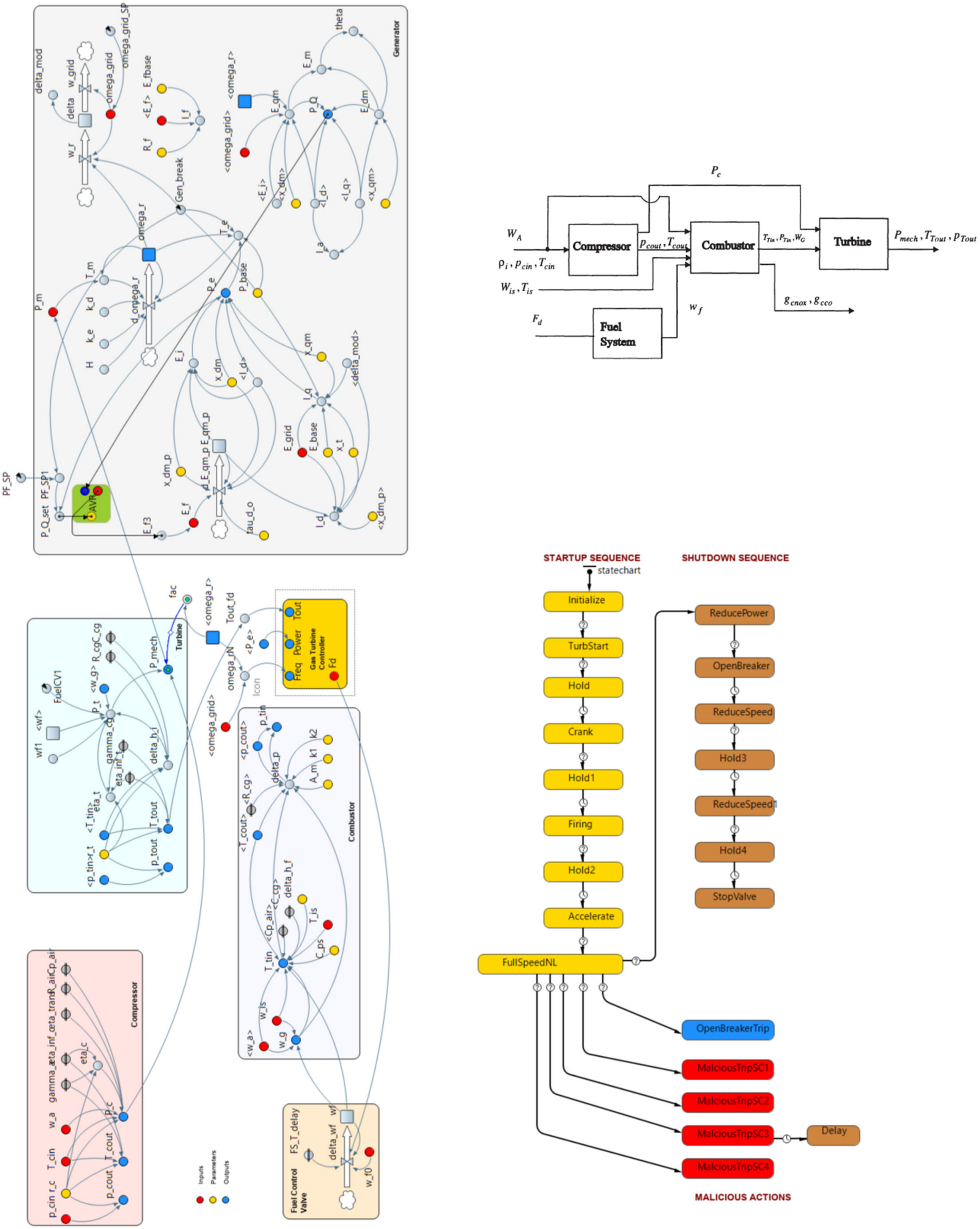


Figure A. 3: Main Components of the Gas Turbine Model