A Framework for Testing Automated Detection, Diagnosis, and Remediation Systems on the Smart Grid

SURF Final Progress Report
Shing-hon Lau
Carnegie Mellon University, PA, USA
Mentors: Bryan Johnson and Kymie Tan

This research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, and was sponsored by the Space Grant program and the National Aeronautics and Space Administration.

1 Introduction

America's electrical grid is currently undergoing a multi-billion dollar modernization effort aimed at producing a highly reliable critical national infrastructure for power – a Smart Grid. While the goals for the Smart Grid include upgrades to accommodate large quantities of clean, but transient, renewable energy and upgrades to provide customers with real-time pricing information, perhaps the most important objective is to create an electrical grid with a greatly increased robustness.

We will formally discuss the concept of robustness later in this paper, but at a high level a robust grid is one architected to prevent failures from occurring, to the maximum cost-effective degree possible, and which is resilient to failures when they are not stopped by preventative measures. It is critical to have an electrical grid which experiences failures infrequently and which continues to deliver power even when failures occur. Failures can range from the traditional, such as downed power lines or a blown transformer, to the exceptional, such as Stuxnet type cyber-attacks or large-scale natural disasters. Robustness stems from passive sources, like a defense-in-depth design, and active sources, like the ability to autonomously detect, diagnose, and remediate failures.

In this work, we intend to lay out a framework for testing the robustness of the Smart Grid. Indeed, the critical importance of the electrical grid dictates that rigorous testing must be performed to ensure a high degree of confidence in its correct operation in virtually every imaginable scenario. We will be focused primarily on the testing of active measures designed to ensure resilient operation in the event of a failure. A particular emphasis will be given to testing autonomous detection, diagnosis, and remediation (DDR) systems. In our view, these are the least understood, but most powerful, resilience mechanisms due to their ability to learn with experience and rapidly identify faults so that repairs can be carried out as soon as possible. Unfortunately, performing rigorous testing is complicated by the fact that the Smart Grid is a massive, complex, and federated cyber-physical system.

The massive size of the electrical grid, which stretches across the entire width of the United States, renders it impractical to replicate physically. As a result, we must resort to simulation techniques. However, simulating such a large structure requires an enormous amount of processing power; a full fidelity simulation is far beyond the reach of even the most advanced supercomputers. A partial fidelity simulation is all that is possible, but if it is done improperly, we risk missing emergent behaviors that arise due to the complexity of the grid. Moreover, the highly federated nature of the electrical grid dictates that thousands of different utilities and governing bodies are involved. Obtaining the correct and sufficiently detailed specifications of the components being tested, as is required for an accurate simulation, is a difficult task in and of itself. Finally, the cyber-physical nature of the Smart Grid poses a challenge because it meshes an inherently analog system, the electrical grid, with an inherently digital systems, communication networks and processing systems.

Given all of these challenges, performing rigorous testing of automated DDR systems is a daunting task. It will be critical to utilize a principled and highly-focused testing regime to validate the properties of the DDR systems. An unprincipled, haphazard testing approach will not work in the Smart Grid domain as it will almost certainly miss significant problems, leading to a false sense of security. Moreover, it will result in duplicated testing efforts, resulting in excessively high costs. Rather, we believe that a guiding framework must be established for testing the Smart Grid. This will involve establishing working definitions for key terms (such as resilience), mapping out the threats to nominal operation in the Smart Grid, and highlighting relevant evaluation techniques and issues.

The rest of this paper is devoted to creating such a guiding framework. We will discuss related works in Section 2, precisely state the objectives of the project in Section 3, identify the approach in Section 4, summarize

our results in Section 5, note design requirements for the Smart Grid that will ensure accurate testing in Section 6, and conclude with future work directions in Section 7.

2 Related Works

In our survey of the Smart Grid field, we have found three different types of related work. The first type contains work that seeks to use simulation testing to validate algorithms or techniques developed by the authors. The second type contains works that wish to lay out frameworks for topics related to testing on the Smart Grid. The third and final type contains works that describe testbeds that can be used for testing Smart Grid elements or have even been designed explicitly for that purpose.

2.1 Simulation Testing

Simulation techniques have been used by many different authors for a variety of purposes. These techniques range from the extremely simple, with only a small handful of elements, to highly complex simulations of thousands of electrical components. For those authors that provided information about the execution of their simulations, most utilized either the PowerWorld software [3] or Matlab's Simulink package [6] (or a derivative thereof).

One of the simplest uses of simulation the work done by Alizadeh, Scaglione, and Wang. The authors compare their stochastic optimal power flow algorithm (S-OPF) to a traditional OPF algorithm when EVs are plugged into the electric grid. They note that their algorithm causes far fewer line violations within the electrical grid; this improvement over existing algorithms comes from the stochastic modelling of EV charging. To obtain their results, the authors use a 3-element system consisting of a generator, a line, and a load that varies as the number of EVs charging changes [7].

Other authors have examined more sophisticated simulations. For example, Ouyang and Due˜nas-Osorio used simulation of the power transmission grid in Harris County, Texas to validate their resilience modeling technique. While the authors mention the source of their data and their general simulation approach, they do not provide any implementation details. Moreover, their resilience metric is not fine-grained, and it would not have required a high-fidelity simulation [15].

Kowli and Cross examined how the use of Demand Response Resources could help reduce congestion rates and fees. They did so by simulating a network with 241 buses and 555 transmission lines. Loads from the Midwest ISO system (in 2006) were used for the simulation. However, few implementation details are presented in their work [13].

There have also been several theses done which focus on simulation results. Aquino-Lugo focused his PhD thesis on developing a distributed OPF algorithm. Proving that the algorithm worked required simulations run with JADE, a Java-based agent simulation tool, and Matlab's Simulink software for power simulation. Simulations were run on systems with 34 and 69 distribution lines [9].

Within his PhD thesis, Anwar examined how to optimally harden the IEEE-118 bus test case. Under some assumptions about the cost for attackers to compromise specific nodes in the network, and costs for securing those nodes against different types of attacks, he derived the optimal hardening under a specific budget. That is, the single deployment of hardware that most reduced the potential damages that could be caused by an attacker [8].

2.2 Frameworks

Berthier et al. created a requirements and architectural framework for intrusion detection on the Smart Grid's Advanced Metering Infrastructure (AMI) [12]. They note several different types of attacks previously investigated by other researchers and indicate what type of monitoring activities are be required to detect these attacks. They also suggest that isolated sensors, with high-computational power and increased physical security, should be used to run intrusion detection systems.

Viswanathan et al. have detailed a semantic framework for analyzing data in networked systems [16]. Their framework utilizes behaviors as the fundamental unit. Simple behaviors can be melded into more complex behaviors. Within the Smart Grid, it may be possible to define the behavior of normal operations, and then examine the remaining traffic for intrusions or other anomalies.

2.3 Smart Grid Testbeds

While some researchers have focused on performing simulations to evaluate a system or algorithm, others have focused on actually developing the testbeds that can be used to perform these evaluations. These testbeds range from the incredibly simple ones, involving only a single computer, to highly complex ones that involve hundreds of computers.

The simplest testbeds available are single machines running either PowerWorld [3] or Matlab's Simulink software [6], occasionally in conjunction with extensions like SimPowerSystems [5]. Both these programs provide mechanisms for setting up a specific network topology and simulating electrical traffic within that network.

The Virtual Power System Testbed (VPST) was designed to permit realistic simulations of power flow, network communications, and SCADA systems all working in conjunction. Within his Master's thesis, Bergman describes the testbed and his improvements on several different aspects [11]. The testbed is composed of three primary components VPST-E, VPST-C, and VPST-R-local. VPST-E handles the electrical simulation and utilizes PowerWorld. VPST-C handles the network simulation using RINSE, an existing system that has been used for Internet-related research which provides a simulation of communications via the DNP3 protocol. VPST-R-local consists of real (i.e., physical) devices that are attached the network and not simulated, enabling testing of real components (e.g., a smart meter) without resorting to potentially non-realistic simulations.

A general-purpose Internet testbed is the DETER testbed, which is jointly hosted by USC and UC Berkeley [10]. While the testbed was originally constructed to enable Internet-related experiments, it could potentially be expanded to permit testing of the Smart Grid. However, such simulations would be software-only and would likely require a significant installation procedure onto DETER.

Beside software testbeds that primarily rely on simulation, there are also hardware testbeds designed to provide high-fidelity data for researchers. A small-scale example is the SmartGridLab testbed constructed by Lu, De, and Song [14]. This testbed permits researchers to easily adjust the generator, load, and topology of a network. However, their validation utilizes only household equipment (e.g., a light bulb as the load) and no suggestion is made as to how the testbed might be scaled up.

A much larger physical testbed is located at Sandia National Laboratories and is designed for testing SCADA systems [4]. However, this testbed is not available for public use.

An upcoming project at Stanford University is GridSpice [2]. This project aims to construct a virtual platform for simulating all aspects of a Smart Grid, ranging from the electrical network (e.g., generation, transmission, load, etc.) to wholesale and retail electrical markets to customer response to electricity pricing. This project appears to be brand-new, so details are currently unavailable.

3 Objectives

Our goals in this work were threefold. We wished to:

1. Provide sensible and concise definitions for testing-related terminology.
2. Propose an experimental framework for rigorous testing of the Smart Grid, with an emphasis on the testing of detection, diagnosis, and remediation systems.
3. Identify potential impediments to accurate testing on the Smart Grid, suggesting mitigation strategies where available.

Where possible, we wanted to provide specific examples for these concepts within the Smart Grid Regional Demonstration Program (SGRDP), a Department of Energy (DoE) funded collaboration between the Jet Propulsion Lab (JPL), the Los Angeles Department of Water and Power (LADWP), the University of Southern California (USC), and the University of California at Los Angeles (UCLA). The aim of the SGRDP is to demonstrate the efficacy of Smart Grid technologies by deploying them in the Los Angeles area. Such a demonstration will require the ability to perform rigorous testing.

4 Approach

Our first task was to examine the terminology used by participants in the SGRDP project and within the research community and identify testing-related terms. In many cases, we found that definitions for this terminology differed from person to person and group to group, oftentimes quite substantially. This resulted in considerable confusion

and miscommunication. For example, one of the goals of the SGRDP is to demonstrate "how the Smart Grid can operate resiliently against physical and cyber attack." When different groups have different ideas of what "resilience" entails, significant problems can result. Clearing up such confusion was a critical first task.

We then proceeded to generate an experimental framework for testing of the Smart Grid. Our focus on DDR systems stemmed from a lack of previous investigation by the research community. It is expected that these systems will play a vital role in the creation of a robust Smart Grid since they will be capable of learning over time, permitting adaption to ever-changing conditions. Generating the experimental framework allowed us to identify issues that need to be examined in further depth, such as the selection of specific faults that DDR systems should detect, the development of appropriate testing strategies, and the creation of suitable metrics and data collection regimes.

Finally, we examined potential impediments to accurate testing by identifying threats to reproducibility and pinpointing potential error sources. Where possible, we suggested mitigation strategies or design requirements that would facilitate rigorous testing.

5 Results

5.1 System Layout

Figure 1 depicts a high-level outline of the SGRDP. Services provided by the program are ranked from the most critical (blue) at the bottom to the least critical (orange) at the top. Unsurprisingly, the most critical services are those involving the generation, transmission, and distribution of electrical power. These are the services that must be preserved at all costs.

In the middle (green), are the services that involve billing or permit the utility to launch a Demand Response (DR) event. DR events are launched when the utility is reaching their maximum online power generation capacity. Customers that are enrolled in the DR program are asked to conserve electricity, say by turning off major appliances. Customers have the choice of allowing the utility to automatically turn off certain appliances if they cannot do so manually. The green tier is separated into the customer-side services (e.g., those involving the meter) and utility-side services (e.g., those involving the storage of billing data).

In the top tier (orange), are services that involve the execution of DR events, electric vehicle (EV) charging, and autonomous distribution, outage, and DR management. As with the middle tier, we divide the top tier into the customer-side services (e.g., home DR execution) and utility-side services (e.g., advanced outage management). Note that autonomous DDR systems can be deployed in areas other than the aforementioned autonomous functions. In fact, virtually every service in Figure 1 can support its own autonomous DDR systems.

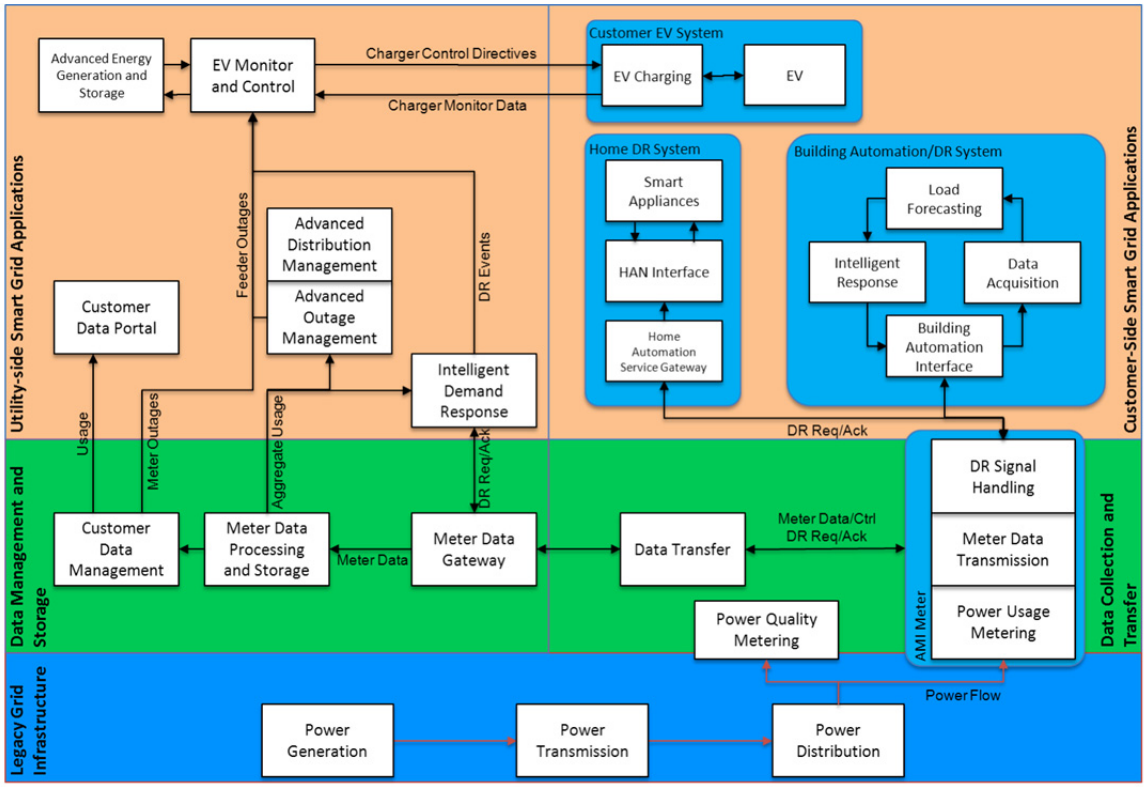

Figure 1: The different services within the SGRDP. The three colors indicate services of different importances, from least important at the bottom to most important at the top.

5.2 Definitions

As previously mentioned, one of the main goals of the project was to arrive at definitions for a variety of critical terms. For the sake of brevity, we will only touch on the most important definitions. We will begin with the terms robustness, prevention, and resilience, which are all interconnected. Note that the ultimate purpose of automated DDR systems is to help create a robust Smart Grid.

Robustness – The ability of the Smart Grid to implement preventative measures to prevent the occurrence of faults and to maintain or recover normal service after a fault occurs.
Prevention – Prevention refers to the measures taken to prevent the occurrence of faults on the Smart Grid. Any techniques that presume a fault has already occurred do not fall into this category.
Resilience – Resilience is the ability of the Smart Grid to maintain or recover normal service after a fault occurs. Resilience measures include masking the problem by switching to redundant systems or repairing the problem, as might be done by automated DDR systems.

The second set of definitions threats to the nominal operations of the SGRDP. Figure 2 shows the relationship between the next group of terms.
Fault – A fault within the Smart Grid results in off-nominal operations of the Smart Grid. In this work, we have classified faults into five different categories, which will be discussed below.
Failure – Faults cause components of the Smart Grid to fail. Note that faults in one context can be failures in another; the two concepts share a recursive relationship. For example, a tree branch might fall on a section of a distribution line (fault), resulting in power going out along the whole line (failure). This might cause a severe load imbalance in the local electrical grid, so the power going out

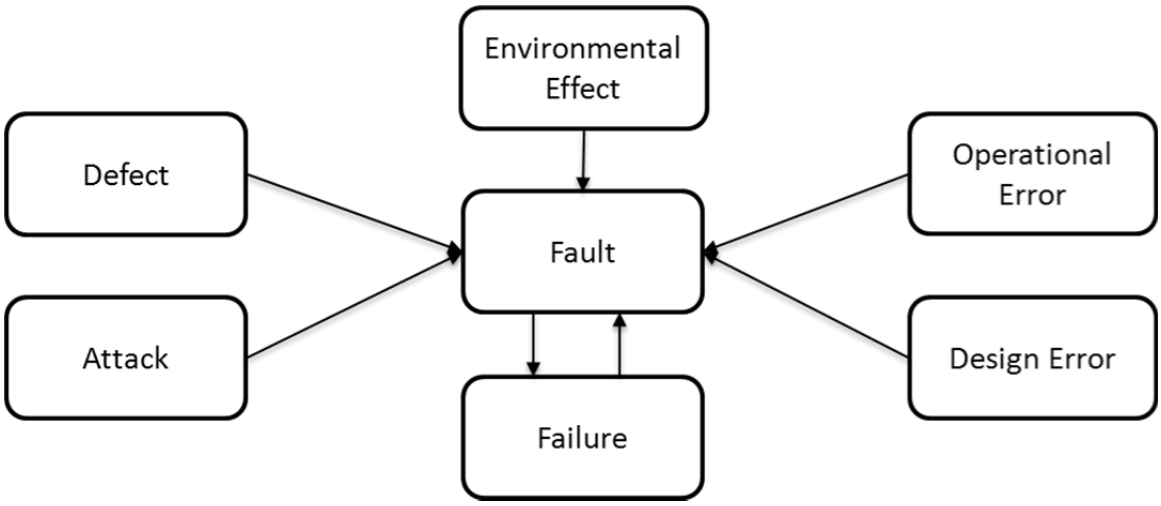

Figure 2: The relationship between the second set of terminology introduced. Note that the fault/failure relationship is recursive; a failure in one context can be a fault in another.

along this distribution line (fault) might cause power in the entire neighborhood to go out (failure).
Attack – Attacks are faults caused deliberately by a malicious entity. One example is a customer that tampers with an electric meter in order to steal power.
Defect – Defects are non-malicious faults that arise in hardware or software components, usually due to wear-and-tear (hardware) or insufficient testing of code (software).
Abnormal Environmental Effect – Abnormal environmental effects are those that lie outside the expected operation conditions of a particular piece of hardware and cause it to operate off-nominally. An extreme example would be a tornado that sucks up numerous distribution power lines.
Design Errors – If off-nominal behavior in the Smart Grid occurs because capacities are exceeded or components interact in an improper manner, then a design error has occurred. One example would be designing a communications system to have too little bandwidth to support all of the billing, DR, and other applications using it.
Operational Errors – Off-nominal behavior of the Smart Grid that results when an automated or human operator makes a mistake is an operational error. One example is a human operator that reroutes power incorrectly, causing certain distribution lines to overload and fail.

5.3 Experimental Framework

When designing an experimental framework, there are four major questions to keep in mind:

1. What is being tested?
2. How is it being tested?
3. How are we measuring the test?
4. What data will be gathered?

We will address each of these questions in turn. Where possible, we will try to use examples of systems that are depicted in Figure 1.

5.3.1 What is being tested?

This question is relatively easy to answer for us. We are interested in testing autonomous DDR systems that are designed to detect faults and failures in components, sub-systems, and systems within the SGRDP. Note that some DDR systems focus on individual pieces of hardware, such as a meter, while others will focus on the entire demonstration project.

5.3.2 How is it being tested?

At the most simplistic level, the experimental methodology for testing an automated DDR system follows six basic steps:

1. Connect the component/system and its corresponding DDR system to the external environment and ensure both the component/system and the DDR system are running properly.
2. Start up the system under nominal operating conditions, simulating external inputs where necessary.
3. Measure the performance of the DDR system under nominal operating conditions (the system should indicate that everything is operating properly).
4. Inject a single fault into the system (say, by misconfiguring a piece of software).
5. Note whether the DDR system took the correct actions, and the time required to take those actions.
6. Repair the fault; inject a new fault; repeat the measurement.

However, this simplistic methodology does not address the specifics of the external environment. Currently, we envision four different environments that might be used: laboratory, simulation, hybrid, and field. We now briefly summarize each of these environments.

Laboratory Environment – Generally laboratory testing involves the use of a testbed, such as the DETER testbed [1]. Experimenters will be able to very closely monitor the behavior of both the component and its associated DDR system. Additionally, experimenters will be permitted to inject a wide range of the faults into the system, including those that would be impermissible to inject into an operating system. Ground truth is also readily available in laboratory testing, since experimenters will be certain which faults they are injected into the system. Finally, laboratory environments are high-fidelity, since actual physical components are utilized.

Simulation Environment – Simulation testing is completely computerized with all components and DDR systems being simulated in a computer. The primary advantage of simulations is that it permits testing on a much larger scale than is possible through laboratory environments. Unfortunately, simulation testing can introduce many sources of errors that are not present in laboratory testing. These sources of error result in simulation testing being lower-fidelity than laboratory testing.

Hybrid Environment – Hybrid testing is a combination of laboratory and simulation testing. Some components are physically represented (i.e., by a piece of hardware) while others are merely simulated. This environment will likely see a good deal of use since it permits some components to be represented in high-fidelity while using a lower-fidelity representation of less important components.

Field Environment – Once components and their associated DDR systems are deployed, experimentation will be relatively limited, especially if testing might cause damage to the operational system. However, some field testing can still be performed. Aside from injecting faults into the system, as suggested in our 6-step methodology, researchers can also wait for faults to naturally occur during the operation of the system and observe the performance of DDR systems. Ascertaining ground truth will be considerably more difficult in this case, since researchers must work to discover what fault(s) were present on the system.

5.3.3 How are we measuring the test?

When evaluating an automated DDR system, there are two types of metrics that we will consider:
primary and inherited.

Primary metrics are those that directly measure the performance of a DDR system. These might deal with the resources consumed (e.g., bandwidth, CPU, RAM), the latency introduced, and the actual correctness of the DDR system (i.e., correct detection, diagnosis, and remediation).

Inherited metrics are the metrics used to measure the performance of the component or system to which the automated DDR system is attached. For the entire SGRDP, these metrics might include the availability or quality of electricity. Observing these metrics for the component or system with and without the automated DDR system will permit us to evaluate the value of adding such a system.

5.3.4 What data will be gathered?

There are two primary types of data that will be collected, one set corresponding to the primary metrics employed and one set corresponding to the inherited metrics. The former set of data consists of data that will be used to train the automated DDR system and additional data used to test the performance of the system. Primary metrics

measuring the performance of the DDR system also falls into this set. The latter set of data consists of measurements of the inherited metrics to determine the performance of the component or system that the autonomous DDR system is attached to.

5.4 Impediments to Accurate Testing

While we are interested in designing an accurate test for autonomous DDR system, it is informative to explore the ways a test can fail to be accurate. Below, we have identified eight issues that might lead to inaccurate tests.

5.4.1 Reproducibility Errors

Reproducibility errors refers to attempts to replicate a prior experiment that obtain significantly different results. These generally result when one experiment is conducted without any lingering effects from a prior experiment while the other is conducted with such effects. It can also occur when different lingering effects are present for each experiment. In either case, an autonomous DDR system may be confused by the lingering effects and operate differently in the two situations. If such reproducibility errors are abundant, it may become impossible to determine how well a particular autonomous DDR system is performing; comparing two different systems will be equally impossible.

5.4.2 Measurement errors

Measurement errors are the most familiar form of error and exist within any of the four types of testing – laboratory, simulation, hybrid, or field. These errors are uncertainties inherent in the measurement tools being used; the only way to decrease these errors are to use better tools. Measurement errors can affect the performance of a DDR system since it can result in (slightly) corrupted training and test data. However, it is not expected that this will be a major source of error.

5.4.3 Modelling errors

Modelling errors result whenever the behaviors of a physical system are abstracted by replacing it with a model. This most common case of modelling errors occur when one wishes to simulate a physical system. Modelling errors tend to be particularly large towards the edge of or outside of the nominal operating range. Errors can be reduced by subjecting a physical system to rigorous testing to ensure that the behavior of the model replicates the behavior of the actual system. Even subtle errors in a model can cause emergent behaviors in a system to go undetected in simulation.

5.4.4 Scaling issues

Scaling issues arise when performing testing on a small scale and deploying on a larger scale. In such a situation, behaviors may emerge on the larger scale that were not present in the smaller scale. The safest way to reduce this type of error is to test on as large a scale as is possible.

5.4.5 Out-of-sync errors

Out-of-sync errors arise when different components in a simulation have slightly different clocks. This can cause components to behave differently than they would in the physical world. For example, components can react to events that occur in the future.

5.4.6 Numerical accuracy issues

Numerical accuracy issues are most commonly found in simulations. They arise because computers cannot perfectly represent real numbers, which introduces small rounding errors. Over time, such errors can accumulate until the simulation deviates significantly from reality.

5.4.7 Ground truth errors

Establishing ground truth is a tricky subject in many disciplines, and the Smart Grid is no different. In the case of testing a DDR system, ground truth refers to several different items.

First, we may wish to test the performance of the DDR system when the system is operating nominally. This requires the system to be provided with data reflecting nominal operating conditions. If we accidentally provide the system with data where a fault is present, then the system may not behave correctly when exposed to truly nominal data.

Second, when we inject artificial faults into a system, there is some baseline behavior present in the system. If this baseline behavior already contains faults, then the DDR system may behave in a surprising or altogether inappropriate manner. Experimenters may conclude that the DDR system erred in its judgement, when it did not.

Finally, when conducting field testing, ground truth is determined based on the conclusions of experts. If these experts reach an incorrect conclusion, while the DDR system reaches the correct conclusion, then experimenters may believe that the DDR system failed when it did not.

5.4.8 Failure injection artifacts

When performing failure injections, extreme care must be taken to avoid introducing artifacts into the data. Otherwise, the DDR system may be making decisions on the basis of the artifacts instead of the injected failures. This can lead to poor performance with the system is deployed.

For example, consider a DDR system within a cyber-security system. During testing, the system is asked to identify two different types of attacks – denial of service and worm – or state that there is no attack present. However, suppose that denial of service attacks always come from one particular IP address while worm attacks come from another IP address. Experimenters observe that the system works perfectly, and decide to deploy it in the field. However, the system was actually identifying the IP addresses instead of the failures caused by the injected attacks. Thus, when it is deployed in the field, it always states that there is no attack present. Operators of the cyber-security system are given the illusion of security since they believe the DDR system will alert them of errors, when it is actually completely ineffectual.

6 Design Requirements to Effect Accurate Testing

The desire to perform testing of automated DDR systems will necessitate numerous design requirements for the Smart Grid. We have identified a eight particularly key requirements:

1. The creation of a cyber-physical testbed that produces nearly identical results to field deployments. DETER [1] may be used as a launching point for this goal.
2. The ability to inject faults into components or systems being tested. While easy in cyber systems, it will be difficult in physical systems, especially in the field.
3. The ability to swiftly repair these faults so that other faults can be injected.
4. The deployment of the sensors required to evaluate all inherited metrics.
5. The deployment of the sensors required by the DDR systems to perform their tasks.
6. The capability to increase the number of sensors present to accommodate future DDR systems that may require additional data. In practice, this may be difficult to do, so it is better to err on the side of too many sensors than too few when performing the initial deployment.
7. The ability to capture and store all of the data being collected and retain it for the duration required by the DDR systems.
8. The ability to exert sufficient control to permit reproducible experiments.

7 Future Work

Currently, we are attempting to influence the design of the SGRDP so that it incorporates as many of the design requirements as possible. Once the design of the SGRDP solidifies, we will work closely with the groups responsible for different components to design specific experiments that can test the performance of DDR systems. In the long term, we anticipate revising this testing framework as new issues and concerns arise.

References

[1] DETER website, accessed on August 14th 2011.

[2] Gridspice website, accessed on July 20th 2011.

[3] PowerWorld website, accessed July 20th 2011.

[4] Sandia National Laboratory testbeds website, accessed on July 20th 2011.

[5] SimPowerSystems website, accessed July 20th 2011.

[6] Simulink website, accessed July 20th 2011.

[7] Mahnoosh Alizadeh, Anna Scaglione, and Zhifang Wang. On the Impact of SmartGrid Metering Infrastructure on Load Forecasting. In *Proceedings of the 48th Annual Conference on Communication, Control, and Computing (Allerton)*, pages 1628–1636, 2011.

[8] Zahid Anwar. *Automatic Security Assessment of Control Systems for Critical Cyber-Infrastructures*. PhD thesis, University of Illinois at Urbana-Champaign, 2008.

[9] Angel A. Aquino-Lugo. *Distributed and Decentralized Control of the Power Grid*. PhD thesis, University of Illinois at Urbana-Champaign, 2010.

[10] T. Benzel, R. Braden, E. Fraser, A. Joseph, D. Kim, J. Mehringer, C. Neuman, R. Ostrenga, S. Schwab, and Dan Sterne. Design of the DETER Security Testbed. Technical report, USC Information Sciences Institute and University of California at Berkeley and McAfee Research, May 2004.

[11] David Carl Bergman. Power Grid Simulation, Evaluation, and Test Framework. Master's thesis, University of Illinois at Urbana-Champaign, 2010.

[12] Robin Berthier, William H. Sanders, and Himanshu Khurana. Intrusion Detection for Advanced Metering Infrastructures: Requirements and Architectural Directions. In *Proceedings of the IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pages 350–355, 2010.

[13] Anupama Kowli and George Cross. Quantifying the Variable Effects of Systems with Demand Response Resources. In *Proceedings of the IEEE IREP Symposium on Bulk Power System Dynamics and Control - VIII (IREP)*, pages 1–10. IEEE, 2010.

[14] Gang Lu, Debraj De, and Wen-Zhan Song. SmartGridLab: A Laboratory-Based Smart Grid Testbed. In *Proceedings of the First IEEE International Conference on Smart Grid Communications (SmartGrid- Comm)*, pages 143–148. IEEE, 2010.

[15] Min Ouyang and Leonardo Duenas-Osorio. Resilience Modeling and Simulation of Smart Grids. In *Proceedings from the ASCE 2011 Structures Congress*, 2011.

[16] Arun Viswanathan, Alefiya Hussain, Jelena Mirkovic, Stephen Schwab, and JohnWroclawski. A Semantic Framework for Data Analysis in Networked Systems. In *Proceedings of the USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2011.