SYSTEMS ENGINEERING
Research Center

Security Engineering Project

A013 - Final Technical Report SERC-2012-TR-028-2

October 24, 2012

Principal Investigator: Dr. Barry Horowitz, University of Virginia

Co PI:

Dr. Peter Beling, Associate Professor, University of Virginia

Dr. Alfredo Garcia, Associate Professor, University of Virginia

Dr. Kevin Skadron, University of Virginia

Dr. Ron D. Wiliams, University of Virginia

Dr. William Melvin, Georgia Institute of Technology

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE
24 OCT 2012

2. REPORT TYPE

3. DATES COVERED
00-00-2012 to 00-00-2012

4. TITLE AND SUBTITLE
Security Engineering Project

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
University of Virginia,Department of Systems & Information Engineering,151 Engineer’s Way,Charlottesville,VA,22904

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR’S ACRONYM(S)

11. SPONSOR/MONITOR’S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**76**	

This material is based upon work supported, in whole or in part, by the U.S. Department of Defense through the Systems Engineering Research Center (SERC) under Contract H98230-08-D-0171. SERC is a federally funded University Affiliated Research Center managed by Stevens Institute of Technology

Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Department of Defense.

NO WARRANTY
THIS STEVENS INSTITUTE OF TECHNOLOGY AND SYSTEMS ENGINEERING RESEARCH CENTER MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. STEVENS INSTITUTE OF TECHNOLOGY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. STEVENS INSTITUTE OF TECHNOLOGY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

TABLE OF CONTENTS

FIGURES

1 Introduction

The Security Engineering project has focused on the development of what is referred to as System Aware Cyber Security, a novel approach for adding cyber attack defense in depth through embedding cyber security solutions to protect specified critical functions of a system within the perimeter of the system. The research efforts have focused on:

1. Concept definition for System Aware Security
2. Design and analysis of reusable system aware security designs that can serve as patterns for solutions that are repetitive at the design concept level from system to system. This includes adopting and utilizing a specification format for security design patterns.
3. Development and initial trial of a first methodology to support system engineers in the selection of groups of available design patterns for a specific system implementation.

The research has progressed to the point where work was initiated to start exploring a DoD-selected specific application: unmanned airborne vehicles (UAV's). This work involved starting the initial concept development for application of System-Aware cyber security design patterns and architecture selection methodology to a specific commercial off-the-shelf UAV system platform carrying electronic data collection apparatus (optical/IR cameras, a radar subsystem and signal collection receivers).

The remainder of this report provides the results for the three System Aware cyber security research areas identified above, as well as the initial exploratory results for the selected UAV application.

2 System Aware Cyber Security Concept

The concept development related to System Aware cyber security has been peer reviewed and published in two archival journal articles and a refereed conference paper (which won the Conference's Best Student Paper Award) . The concept offers a novel system engineering-based approach for adding cyber security to a system. The concept papers include:

R. A. Jones and B. M. Horowitz, A system-aware cyber security architecture, Systems Engineering, Volume 15, No. 2 (2012), 224-240. and J. L. Bayuk and B. M. Horowitz,

J. L. Bayuk and B.M. Horowitz, An architectural systems engineering methodology for addressing cyber security, Systems Engineering 14 (2011), 294-304.

R. A. Jones and B. M. Horowitz, System-Aware cyber security, itng, 2011 Eighth International Conference on Information Technology: New Generations, 2011, pp. 914-917.

In addition numerous presentations were provided within the DoD research community including a presentation to the Defense Science Board. Attached is a representative presentation regarding the overall System Aware cyber security concept.

In summary, the System Aware concept:

- Operates at the system *application-layer*,
    - For *security inside* of the network and perimeter protection provided for the whole system
    - Directly protects the *most critical system functions*
    - Solutions are *embedded within* the protected functions
- Addresses *supply chain* and *insider threats*
- Includes *physical systems* as well as *information systems*
- The solution-space consists of *reusable design patterns*, reducing unnecessary duplications of design and evaluation efforts
- Includes a *scoring framework* for supporting Systems Engineers in evaluating alternative architectures

3 SYSTEM-AWARE CYBER SECURITY DESIGN PATTERNS

Over the course of the RT-28 project, a number of design patterns were developed. These design patterns serve as a starting point for more customized design efforts that serve to integrate the desired patterns into the system to be protected. The following list is the initial set for which work was initiated:

1. **Diverse Redundancy** for post-attack restoration
2. **Diverse Redundancy + Verifiable Voting** for trans-attack attack deflection
3. **Physical Configuration Hopping** for moving target defense
4. **Virtual Configuration Hopping** for moving target defense
5. **Data Consistency Checking** for data integrity and operator display protection
6. **Physical Confirmations of Digital Data** for data integrity
7. **Use of Analog Components** for diversely redundant solutions

For patterns 1 and 2 above, refereed conference papers were accepted for publication and presentation. The papers served to both describe the pattern and present detailed performance evaluation results. In addition an archival journal paper was accepted for publication on design pattern number 5 above. The papers are:

B. M. Horowitz and K. M. Pierce, The integration of diversely redundant designs, dynamic system models, and state estimation technology to the cyber security of physical systems, to appear Systems Engineering, Volume 16, Number 3, 2013 (Ore-publication version attached as Appendix 1).

R.A. Jones, T.V. Nguyen, and B.M. Horowitz, System-Aware security for nuclear power systems, 2011 IEEE International Conference on Technologies for Homeland Security (HST), November, 2011, pp. 224-229.

G. L. Babineau, R. A. Jones, and B. M. Horowitz, A system-aware cyber security method for shipboard control systems with a method described to evaluate cyber security solutions, 2012 IEEE International Conference on Technologies for Homeland Security (HST), November, 2012.

The design patterns integrate design concepts from three different engineering communities: fault tolerant systems, automatic control systems and information assurance. With integrated designs, these papers show that it is possible to: 1) raise the level of difficulty of attacks significantly, 2) automatically identify attacked system components, 3) switch from attacked components to diverse back-up components prior to an attack taking effect, and 4) provide automatic rapid recovery from successful attacks.

In order to provide a starting point for the exploration and development of new secure design patterns, four patterns are presented based upon the work outlined in this report. The design pattern descriptive material is drawn from Rick Jones' PhD dissertation document, System Aware Cyber Security, University of Virginia, 2012. In order to facilitate reuse of design patterns, a standard format for use in organizing a design library would be required. The format for these patterns is based upon those used for traditional perimeter security as presented by Schumacher in his book on "Security Patterns: Integrating Security and Systems Engineering" [2006]. However, unlike the patterns presented by Schumacher, these patterns are not based upon implemented solutions, but on research cases. Research cases were chosen as, "Patterns support the understanding of problems and their solutions," [Schumacher, 2006] and, "Patterns are generic—as independent of or dependent on a particular implementation technology as need be." [Schumacher, 2006]. Thus, design patterns provide not only a means for recording implemented solutions, but a method for recording research cases so that they can be applied to problems across a wide set of domains. As System-Aware security aims to provide cyber security solutions that are applicable to many domains, design patterns provide an ideal means of recording and presenting such solutions for reuse.

The selected design pattern format includes description material divided into 12 parts: **Example, Context, Problem, Solution, Structure, Dynamics, Implementation, Example Resolved, Variants, Known Uses, Consequences** and **Related Design Patterns.**

The descriptive material required for each of these 12 categories is shown through 4 examples, presented below.

3.1 Pattern Name: Diverse Redundancy

Example: Figure 1 presents a high-level system diagram for a typical steam fed nuclear reactor powered turbine control system. As indicated in Figure 1, the turbine receives actuation commands from a controller, currently available from a variety of vendors (e.g., the GE Mark VI, and Triconex Tricon). Operators located in the main control room of the power plant are responsible for controlling the turbine. These individuals receive status information from the controller that influences their operational actions, which can include stopping the turbine and correspondingly tripping the reactor to stop steam flow into the turbine. In addition to operator actions, the controller receives sensor information (listed in Figure 1) that together influences its automatic control actions. In situations where the turbine operation is such that it is of immediate importance to stop steam flow, the reactor is automatically stopped (i.e. scrammed), with a reactor shutdown process that is supported by the sensor information related to turbine operation.

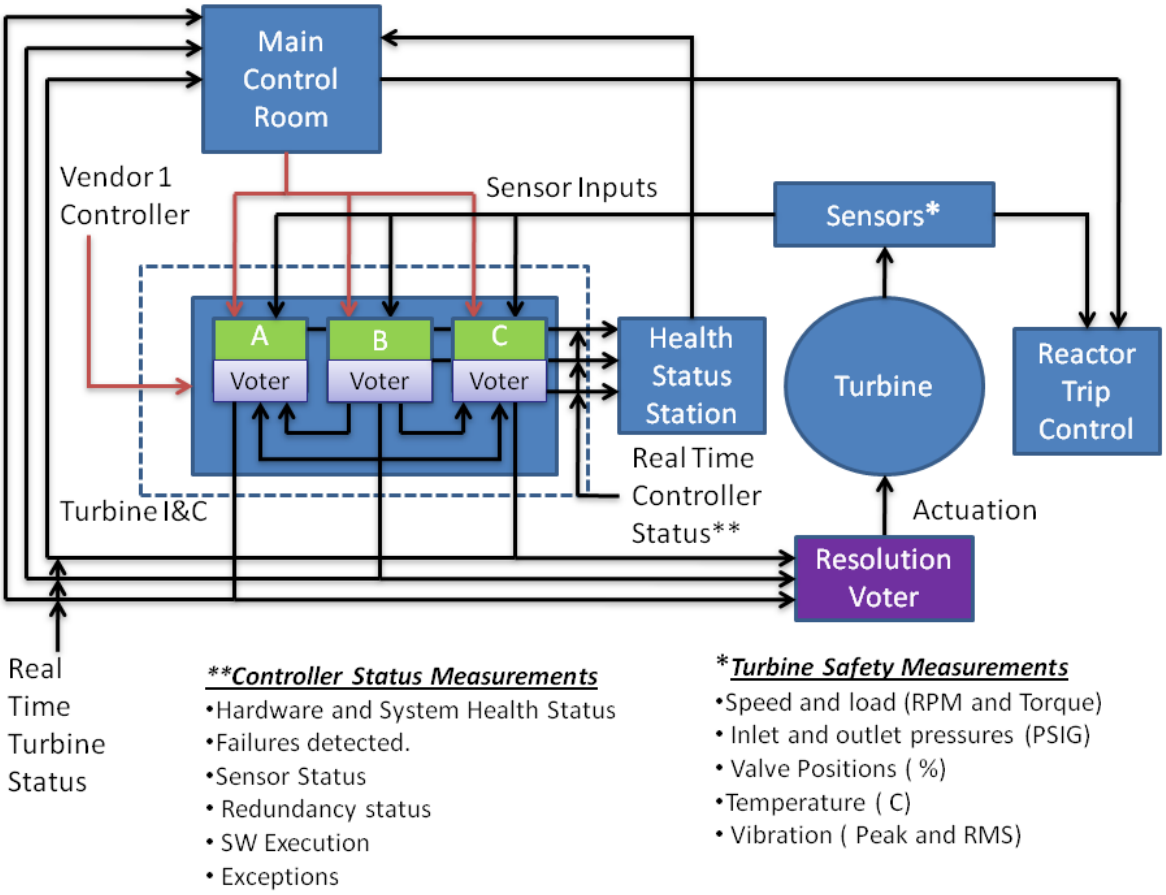


Figure 1. A high-level system diagram for a typical steam fed nuclear reactor powered turbine control system. The turbine controller is designed to meet high reliability and safety standards by employing redundancy and a resolution voter.

Figure 2 also highlights the fact that nuclear power plant turbine controllers are designed to meet high operational reliability and safety standards, and accordingly often employ various types of redundancy. However, there has recently been a rash of insider attacks embedding Trojan horses into the equipment of the supplier of the reactor's controllers. Given that the significant economic consequences of serious damage to the turbine, and the need to shut down (trip) the nuclear reactor in the event of a turbine shut-down, how can the reactor's owner continue to maintain high reliability while ensuring her system against a possible supply chain attack?

Context: Ensure that system functions critical for achieving mission objectives and high reliability requirements will be available even if one or more the components that support those functions have been compromised by a cyber attack.

Problem: While the use of redundant components in systems is a common way to assure continuity of operation, the use of components that are susceptible to a common source of failure does not provide assurance against a cyber attack that effects all of the common components.
Solving this problem requires one to resolve the following forces:

- For a cyber attack, a single exploit can be developed and used to compromise all of the identically redundant components that might otherwise provide enhanced continuity of operation
- The cyber attack can be embedded into the redundant components through the supply chain or an insider attack making it difficult to ensure that a cyber attack has not compromised all of the components
- The single exploit may be an extremely minor change (e.g. the change of a single parameter) and triggered remotely or based on a certain condition (e.g. time). As a result detecting that a component or components have been compromised can be extremely difficult.

Solution: Solutions for ensuring that the success of a cyber attack on a critical system function(s) does not result in mission failure can be based upon protection approaches developed by the fault tolerant systems community. One such technique is to utilize diversely redundant components to ensure that a system is able to carry out its mission objectives even when one of those components breaks down. This assumes that each of diversely redundant failures is independent; i.e. no common source exists to cause the same fault in all of the components. A cyber attack is one such common source that could put all redundant components at risk, and prevent a system from completing its mission objectives. This solution mitigates the capacity for a cyber attack to successfully compromise all redundant components by utilizing diverse components with a different set of attributes.

Structure:

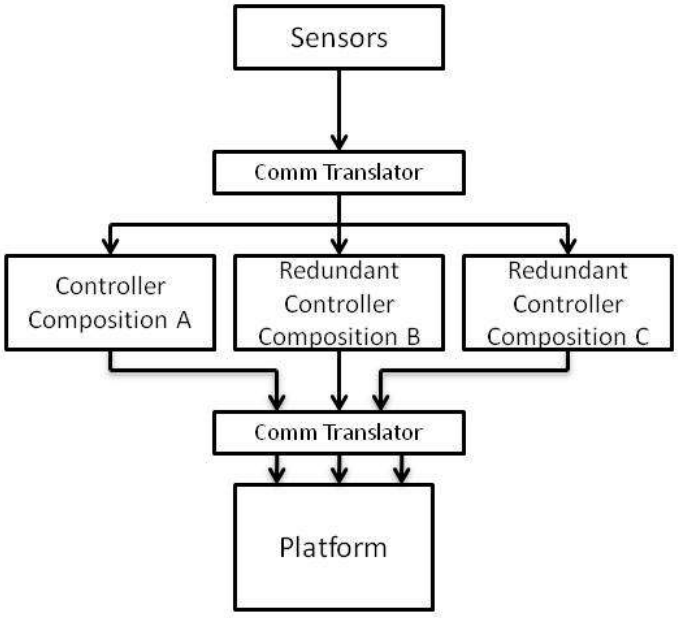


Figure 2. A simple illustration of the structure of *Diverse Redundancy*. In this instance three different controllers are used to receive inputs from a set of sensors and issue inputs to control a platform. Furthermore, each of the controllers is utilizes a diverse set of protocols. Thus, communication translators are included (i.e. the Comm Translators).

Diverse Redundancy requires the following elements:

- Two or more diversely redundant components. These components must be diverse with regards to the common source of the cyber attack. For example, if the common source is a Trojan horse injected via the supply chain, then the common source is the supplier and the components should be procured from independent suppliers.
- Special hardware may be needed to integrate the diverse components into the system. For the structure shown in Figure 2, the diverse components use special communication translators as each of the diverse controllers employs a different communication protocol.

Dynamics: As seen in Figure 2, the diverse components will possibly need to be able to receive input, generate output, and exchange information with other diverse components. Depending on whether the original system employed redundancy or not, additional infrastructure may be needed to transmit information to and from the diversely redundant components, as well as between the diversely redundant components. For example, an additional mechanism may need to be integrated into the system which is used to ensure that only one of the diversely redundant controllers is sending its information along and that the remaining are serving as backups. Alternatively, in order to avoid bumpy outputs when it is required to switch components due to a failure, a mechanism could be employed to average the outputs of the diversely redundant components. This result is then utilized as the output of the diversely integrated components.

Implementation: Diversity can encompass a large set of parameters, including hardware, software, vendor, geographical location, administrator(s), etc. Thus, it is important to consider the type(s) of diversity that will be needed to prevent an attack. For example, utilizing multiple diverse operating systems will force an adversary to develop cyber attacks for each of the operating systems, but could leave them vulnerable to an attack embedded in a common hardware component. Diverse components may require special components needed to ensure interoperability.

Example Resolved: The owner of the nuclear reactor decides to integrate two additional turbine controllers along with *Verifiable Voting* and *Physical Configuration Hopping*. As shown in Figure 3, as the reactor owner was worried about compromised components, she has decided to integrate three turbine controllers from different vendors. As each of these vendors employs its own communication protocol additional communication translators are needed to ensure interoperability. *Verifiable Voting* has been utilized to detect and isolate a controller issuing potentially damaging information, as well as to ensure that only one of the controller's command signal reaches the turbine. Finally, *Physical Configuration Hopping* is utilized to both enhance security and select which of the diversely redundant controllers data will be passed to the turbine.

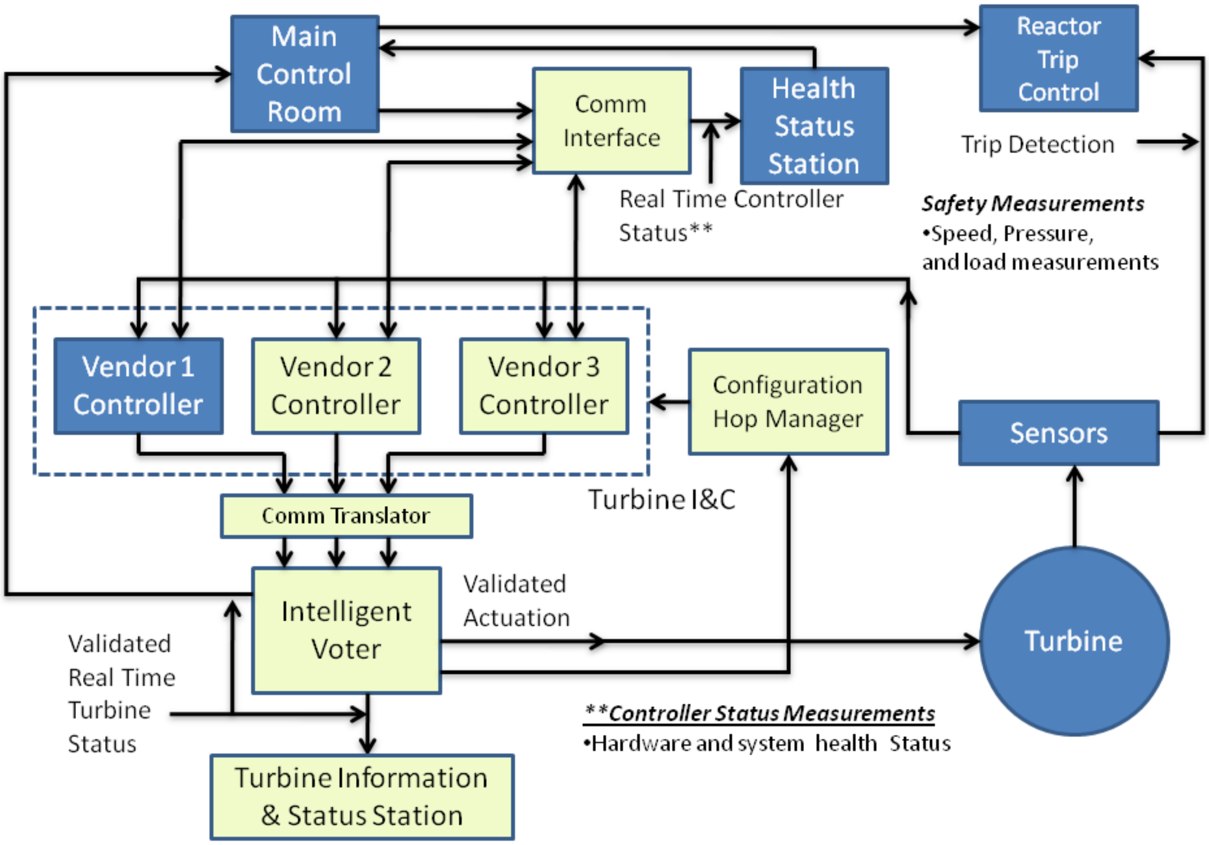


Figure 3. Resolved solution for *Diverse Redundancy*. In this instance, diver redundancy and *Verifiable Voting* have been employed to protect the turbine controller and ensure protection against a supply chain attack.

Variants: A variation includes utilizing redundant components that possess reduced or different capabilities. For example, a GPS-based navigation system can utilize an inertial navigation system as a redundant backup.

Known Uses: [Jones and Horowitz, 2011, Jones, Nguyen, and Horowitz, 2011; Jones and Horowitz, 2012; Babineau, Jones, and Horowitz, 2012]

Consequences: The following benefits may be expected from applying this pattern:

- *Diverse Redundancy* can serve to increase the complexity of an attack that would attempt to compromise all components by forcing the need for cyber attacks with specific capabilities to address each of the diversely redundant components
- In systems without redundant components, *Diverse Redundancy* can potentially increase the systems robustness to faults
- Some systems may already possess diverse components and can possibly make implementation easier

The following potential liabilities may arise from this pattern:

- *Diverse Redundancy* may require additional infrastructure to ensure interoperability with all components
- In systems without redundant components, *Diverse Redundancy* may require new infrastructure to ensure all components receive the appropriate input and that the proper output signals are sent
- As *Diverse Redundancy* requires the components to be diverse with regards to the common source of failure, the amount of commercial off the shelf (COTS) solutions for providing diversity may be limited
- Life cycle costs and training of support staff could increase due to the requirement to service *Diverse Redundant* components

Related Design Patterns: *Verifiable Voting* is a mechanism that can be combined with *Diverse Redundancy* to help detect and isolate which of the diversely redundant components have been compromised. *Diverse Redundancy* can also be combined with *Physical or Virtual Configuration Hopping* to dynamically switch which component is engaged in the operational system at any given time in order to both detect a compromised component and minimize the time available for an exploit to affect the system.

3.2 Pattern Name: Verifiable Voting

Example: A museum has recently installed a video surveillance system to protect its collection of rare and valuable artifacts. As shown in Figure 4, this system consists of a series of security cameras that transmit their data to a media server and its hot shadowed backup. Security personnel can pull the video streams from the media server to their mobile devices to observe the rooms remotely. In addition, when the museum is closed, the media servers scan all of the incoming video streams for unauthorized personnel. If the servers detect any unauthorized access an alert is sent to the security personnel. The security personnel can then decide to pull the video stream to determine the situation and take appropriate action to apprehend the intruder.

Recently the primary employee responsible for managing and maintaining the media servers was fired under the suspicion that she was planning a heist on the museum. Given the access this employee was afforded had to the media servers, the owner of the museum is concerned that employee may have tampered with the media servers as part of heist. As a result, the museum owner wishes to employ additional security to protect against a possibly malicious server.

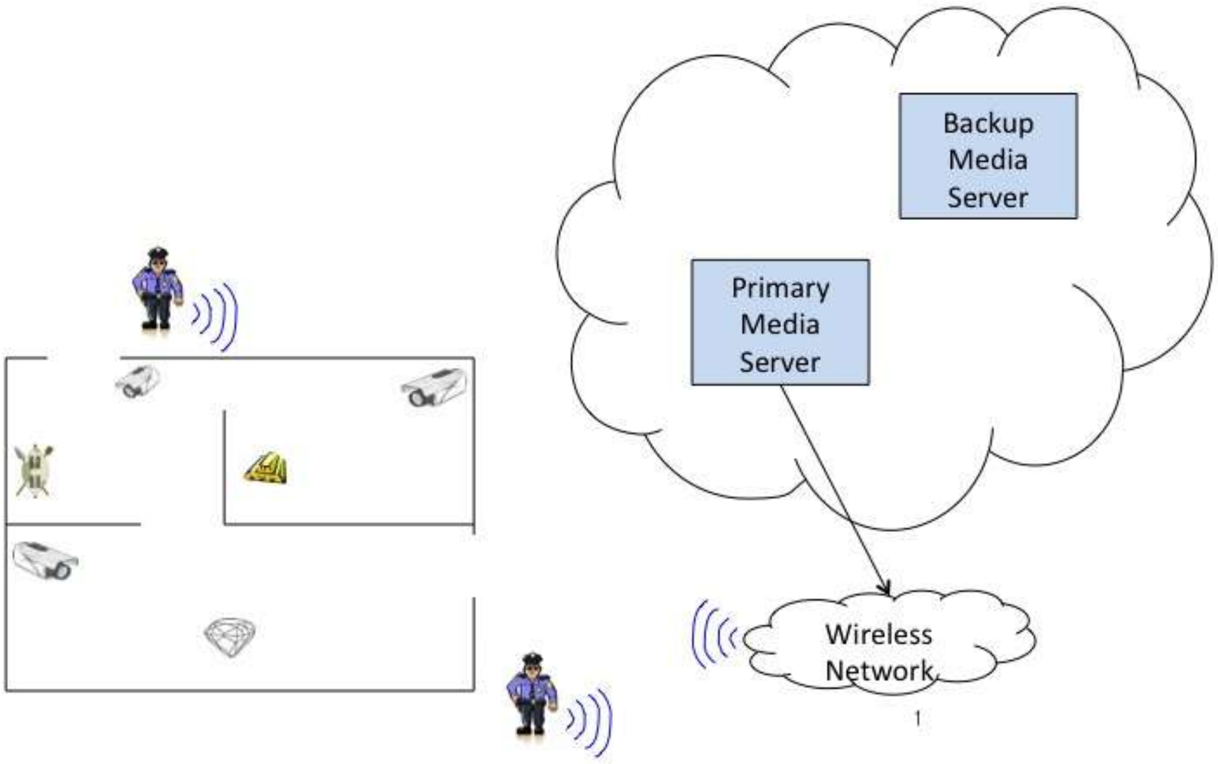


Figure 4. A high-level system diagram of a video surveillance system for a museum. The security cameras send the video surveillance to media servers that distribute the information wireless to security personnel.

Context: Systems often produce information that is critical in determining the appropriate set of actions to be taken to ensure the desired outcome. However, there is reason to suspect that the source of information may not always be producing reliable information. This can result in a significant decline in performance and can potentially result in an undesired or inferior outcome whenever this source is producing valid information, but nonetheless is not trusted, or the source is trusted, but producing bad information—such as due to a cyber attack. Thus, a method is needed to be able to detect and/or isolate those components that may be compromised and may be producing faulty information.

Problem: How can one continue to utilize (i.e. trust) the outcomes of a critical system when one suspects that the system has been compromised?
Solving this problem requires one to resolve the following forces:

- Given the support afforded by the system is critical to achieving desired outcomes, it is undesirable to simply disable it. In addition, if the system were compromised by a cyber attack it could cause considerable harm and possibly result in undesired outcomes. Thus, whenever the system is producing faulty output the information should be ignored. A method is needed to detect when the output of the system is valid and when it is misleading

- It may be possible to restore a system to working order once a compromise has been detected; however, to do so it is may be necessary to isolate the component responsible for producing the faulty output
- To protect against a cyber attack, the mechanism employed to detect and isolate systems producing faulty information must also be secured. In addition this mechanism must not impact system performance to the point of preventing the system from functioning properly

Solution: A voting scheme is typically used to detect and isolate systems that are producing faulty outputs. Voting can also be utilized to detect misleading outputs. However, in the event of misleading information that is being produced as a result of a cyber attack, it is possible that the attack may have been embedded into the component through the supply chain or from an insider. As a result, it is possible that the mechanism used to carry out the voting may be compromised. *Verifiable Voting* is utilized to provide voting in a secure manner. It is based on providing a hierarchy of voters tailored to the specific needs of the system to ensure that components acting maliciously are identified, while not significantly impacting system performance. Each of the voters in the hierarchy is designed based upon trade-off analyses regarding ease of verifiability—i.e. confidence that it has not be compromised—and ability to perform timely and complex comparisons.

Structure: *Verifiable Voting* is composed of one or more voting mechanisms implemented in hardware or software. This includes an extremely simple voting mechanism, implemented in hardware or software, which is easily verifiable; i.e. known to be secure. However, such a simple mechanism may only be capable of implementing a simple voting scheme. This may result in voting rules that do not include all available information, resulting in an unacceptable degradation of performance compared to a voting scheme that uses more information. Alternatively, using more information may make the voting logic too complex to sufficiently verify its implementation from a security standpoint. As a result, in addition to using the less sophisticated, but more verifiable voters to validate simple, but mission critical machine generated outputs (e.g. fire the gun), they can also be used periodically, as a coarse check on whether a less verifiable voter has been compromised. Finally, *Verifiable Voting* requires that there be multiple redundant systems producing output. The amount of redundancy determines how many redundant systems can be compromised before is becomes impossible to detect and isolate potentially compromised components. Figure 5 illustrates one possible hierarchy of voters that assumes only a single redundant system will be compromised at a given moment.

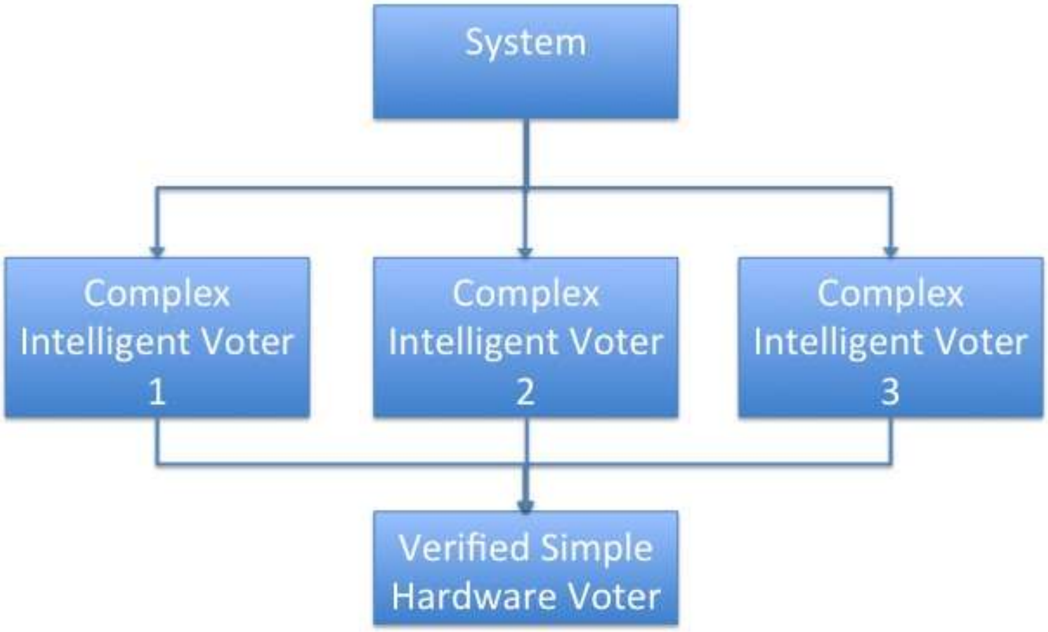

System
Complex Intelligent Voter 1
Complex Intelligent Voter 2
Complex Intelligent Voter 3
Verified Simple Hardware Voter

Figure 5. A simple example of *Verifiable Voting*. This includes three complex intelligent voters that are used to evaluate the information from the system. These results are fed to a simple hardware voter that can be easily verified.

Dynamics: All voters need to be able to receive the necessary outputs for comparison from the multiple redundant systems. It is important that the most verifiable (i.e. secure) of the hierarchy of voters be able to override the decisions of the less secure voters.

Verifiable Voting requires replication of the outputs of system in order to carry out the vote. If the system already carries the necessary redundancy or the output of the system is small (e.g. a true or false value) then the cost of this replication can be negligible. However, when the outputs being voted on are large (e.g. the output of diversely redundant video streams received over a wireless network for voting) then such voting can add significant overhead. While, this overhead can potentially be mitigated through the use of additional resources, it may also be possible to mitigate it through the use of customized system designs. For example in Figure 5 each of the three complex intelligent voters is receiving the three inputs simultaneously. However, it is possible to stagger the voting across each voter; i.e. complex intelligent voter 1 receives the three inputs and votes, than complex intelligent voter 2 receives the three inputs and votes, and finally complex intelligent voter 3 receives the three inputs and votes. Once this is done each of the complex intelligent voters can send its simplified results to the simple hardware voter for a final decision (see Figure 5). For the case of a wireless network

communicating the information, bandwidth utilization can be reduced through the application of staggered voting, with the consequence of a potential delay in detecting modification of data in one of the streams.
Implementation: When implementing *Verifiable Voting* it necessary to determine an appropriate scheme for voting as well as the input that will be voted on. Given this information, it is possible to determine the desired number of redundant system components to achieve detection and isolation. It is also possible to develop an appropriate hierarchy of voters. This hierarchy will depend on the type of information used in voting, the frequency of voting, and the desired security of the *Verifiable Voting* scheme itself. Finally, additional resources or techniques may be needed to ensure that the desired level of system performance is achieved.

Example Resolved: To defend the museums rare artifacts against a possible cyber attack embedded in the media server, the owner decides to implement *Verifiable Voting*. As there are only two media servers, *Verifiable Voting* is only able to provide detection. As the museum has security guards on patrol and possesses the capacity to rapidly lock down the artifacts, it is decided that isolation is not necessary. If the *Verifiable Voter* detects a problem (i.e. cannot reach consensus) it will alert the security personnel who can then place the museum on lockdown.

To ensure that the *Verifiable Voter* will be secured against cyber attacks, it is decided that the *Verifiable Voter* will be deployed onto mobile devices used by the security personnel for alerts. While it is possible that a single guard's device could be compromised, their would still be several additional security guards still receiving information. Thus, an attacker would have to compromise all of the mobile devices used by personnel. From the perspective of the museum owner, this is deemed an unlikely event and thus an acceptable risk.

Finally, each of the guard's devices will perform *Verifiable Voting* on the information coming from the media servers, including the video stream. Due to both the large bandwidth consumed by video and the limited bandwidth available for wireless communications, it is decide to implement a duty cycle voting scheme.

Variants: None.

Known Uses: [Jones and Horowitz, 2011, Jones, Nguyen, and Horowitz, 2011; Jones and Horowitz, 2012; Babineau, Jones, and Horowitz, 2012]

Consequences: The following benefits may be expected from applying this pattern:

- Can both detect misleading output as well as isolate the offending component
- Voting mechanism can be implemented in a more secure manner
- Offers a flexible implementation to trade off desired level of security with cost, complexity, and performance impacts

The following potential liabilities may arise from this pattern:

- Detection and isolation require the introduction of multiple redundant components with the attendant liabilities
- Depending on the information being voted upon, it can result in an increase in complexity and cost to ensure that solution meets the desired goal
- Can be defeated if enough of the redundant devices are compromised to form a majority (what constitutes a majority will depend on the voting scheme utilized)

Related Patterns: This pattern can be combined with Diverse Redundancy to potentially increase the difficulty in compromising all redundant components—e.g. through an insider or supply chain attack.

3.3 Pattern Name

Example: Modern ships are equipped with a wide set of systems and to monitor and control (e.g. engine, propulsion, fire suppression, and climate control). A company wishes to produce a lower cost ship by consolidating the network between the monitoring consoles and the physical systems into a single COTS network switch. To improve the reliability of the design, a redundant network switch is installed to resume operations in the event the primary switch fails. However, consolidating all network connections also leaves the entire ship vulnerable to any cyber attacks embedded into the primary network switch:

- Send potentially misleading information to the monitoring systems
- Could disable the ship through a denial of service attack by dropping all communications
- Modify or inject commands to the physical systems in order to damage, disable or misdirect the ship

Context: Ensure that critical system components that have been infected with a cyber attack will be unable to actively disrupt, damage, or misdirect systems operations.

Problem. Techniques exist to detect, isolate, and disable system components that are behaving in a manner to cause harm to the system. However, a system component compromised by a cyber attack has the potential to disrupt and possibly damage critical system components before such methods are successfully able to disable the offending component. In addition, such methods may be unable to prevent cyber attacks aimed at passive monitoring or more sophisticated attacks that attempt to cause disruptions and damage more subtly (e.g. Stuxnet attack).

Solving this problem requires one to resolve the following forces:

- Ensure that a cyber attack is not given enough time to cause damage or disrupt system operations; this time may be less than the time needed to detect and isolate the compromised component
- Prevent a cyber attack exploit from reading enough information to form a coherent data set for use by the attacker
- Security solution must not compromise the systems mission objectives by significantly impacting on system performance

Solution: Solutions for preventing compromised system components from taking potentially malicious action can based on the techniques developed by the cyber security community. One such technique is moving target defense that aims to dynamically switch resources. *Physical Configuration Hopping* builds on this technique by continuously shifting control between multiple redundant physical system components in order to disrupt a cyber attack before it can cause permanent damage.

Structure: As seen in Figure 6, *Physical Configuration Hopping* requires multiple redundant components to be dynamically interchanged (two in Figure 6). This dynamic reconfiguration determines which component(s) is in control at any given time. In addition, there is a mechanism utilized to the control the frequency of the dynamic readjustment as well as determine which component is in control—in Figure 6 it is the configuration hop manager. Finally, their needs to be a mechanism in place to control the switching between components; this includes the frequency of hopping as well as, the order of hopping from one component to another (pertinent to cases of higher orders of redundancy).

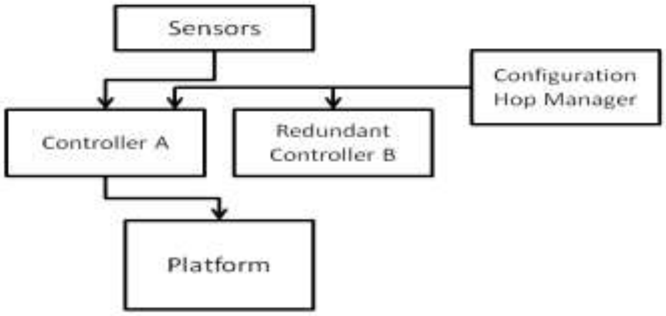


Figure 6. A simple Physical Configuration Hopping setup. This instance includes dynamic reconfiguration across two redundant controllers. Controller A is currently set to the active controller.

Dynamics: *Physical Configuration Hopping* requires that all redundant components will be able to receive and generate output to the appropriate systems, as control will need to be dynamically switched between those components. In addition, it may be necessary to ensure that the dynamic switching between components is bumpless. For example at the time of switching the multiple redundant components may be in different states; thus, the switch between components results in an unintended switching of states.

Implementation: When implementing *Physical Configuration Hopping* it is important to consider the time it will take for a compromised component to cause damage. For example, a turbine in a nuclear reactor can potentially be damaged in a matter of seconds. Alternatively, it may take several minutes or even hours to steer a ship far enough off course to be considered damaging. In addition, the sophistication involved in switching between redundant system components depends on the sophistication of the cyber attack to be prevented. For example, switching between redundant components in a round robin fashion may disrupt a cyber attack that is just trying to transmit damaging commands quickly. However, a more sophisticated attack may be able to detect the switching patterns. This information could then potentially be used to issue commands that ultimately cause damage through controlled thrashing that occurs every time a switch from the compromised component to a non-compromised component occurs. It is also important to decide how much control is given to administrators to change the frequency of hopping as well as the algorithm used to control the switching order and specific, perhaps pseudo-randomized timing.

Example Resolved: The ship building company decides to combine *Physical Configuration Hopping* with *Diverse Redundancy* in order to protect the ship from a compromised network switch. The company decides to purchase two switches from different vendors in order to help prevent a scenario where both switches are compromised via the supply chain. The company then determines that it is not worried about a Trojan horse being embedded in the new system component used for monitoring the information, control and status information between systems is not of direct value to an attacker; however, it is worried about a compromised switch causing denial of service or injecting false and/or damaging commands. It is then determined that it would take at least five minutes before a compromised network switch could cause any permanently damaging actions. Finally, the dynamic switching has the potential to cause some status information to be lost; however, the amount of information lost is small relative to the frequency of updates; i.e. no additional resources are needed for bumpless control.

Variants: None

Known Uses: [Jones and Horowitz, 2011, Jones, Nguyen, and Horowitz, 2011; Jones and Horowitz, 2012; Babineau, Jones, and Horowitz, 2012]

Consequences: The following benefits may be expected from applying this pattern:

- Prevent a compromised system component from cyber attack before it is able to compromise the mission objectives; prevention can occur independently, and faster than methods used for detection, isolation, and restoration
- Makes the development of cyber attacks more difficult by introducing time as an element

The following potential liabilities may arise from this pattern:

- Requires multiple redundant components with the attendant liabilities of the Diverse Redundancy design pattern
- Introduce the need for methods to ensure bumpless control
- Defeated if the frequency of hopping is too slow, or the algorithm for switching is predictable

Related Patterns: Can be combined with *Diverse Redundancy* to potentially mitigate the risk that multiple redundant components will be compromised.

3.4 Name: Virtual Configuration Hopping

Example: An e-commerce business stores all credit card information in a secure facility equipped with a video surveillance system. This video surveillance is maintained and routinely inspected by a private contractor to ensure that it is operating properly. Recently the company has learned that several of the companies that also use this private contractor have been the victims of theft. An investigation of each of the sites has revealed that each of the systems responsible for receiving and displaying the streams to security personnel was infected with a Trojan horse to perform a simple replay attack. Furthermore, it is suspected that an employee of the private contractor did the theft. The e-commerce site has invested significant resources in building the secure facility and video surveillance system and desires a solution to secure the video surveillance system against a possible insider attack.

Context: Ensure that critical system functions that have been infected with a cyber attack will be unable to actively disrupt, damage, or monitor systems operations.

Problem: Techniques exist to detect and isolate, disable system functions that are behaving in a manner to cause harm to the system. However, a system component compromised by a cyber attack has the potential to disrupt and possibly damage critical system functions before such methods are successfully able to disable the offending functions. In addition, such methods may be unable to prevent cyber attacks aimed at passive monitoring or more sophisticated attacks that attempt to cause disruptions and damage more subtly (e.g. Stuxnet attack). Solving this problem requires you to resolve the following forces:

- Ensure that a cyber attack is not given enough time to cause damage or disrupt system operations; the time to cause damage or disruption may be less than the time needed to detect and isolate the compromised function

- Prevent a cyber attack from reading enough information to form a coherent picture

- Security solution must not compromise the systems mission objectives by significantly impacting system performance parameters

Solution: Solutions for preventing compromised system functions from taking potentially malicious action can be based on the techniques developed by the cyber security community. One such technique is moving target defense that aims to dynamically switch resources. *Virtual Configuration Hopping* builds on this technique by continuously shifting control between multiple redundant virtualized system functions in order to disrupt a cyber attack before it can cause permanent damage.

Structure: As seen in Figure 7, *Virtual Configuration Hopping* requires multiple redundant functions to be dynamically swapped (two in Figure 7). This dynamic reconfiguration determines which function(s) is in control at any given time. In addition, there is a mechanism utilized to the control the frequency and exact timing of the dynamic readjustment as well as determine which function is in control—in Figure 7 it is it he configuration hop manager. Finally, their needs to be a mechanism in place to control the switching between function; this includes the frequency of hopping as well as the order.

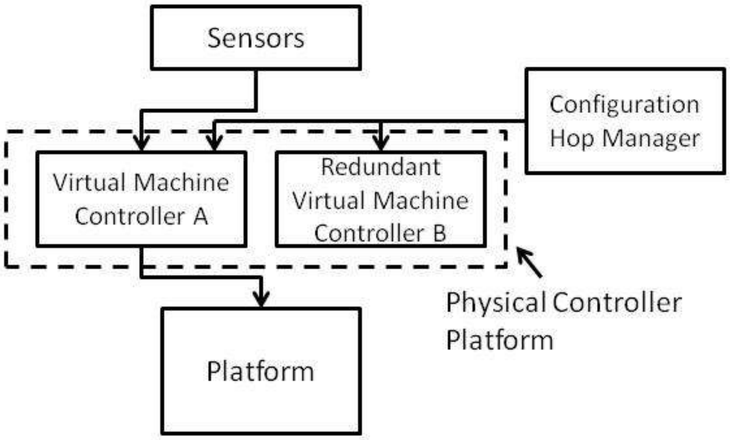

Sensors
Configuration Hop Manager
Virtual Machine Controller A
Redundant Virtual Machine Controller B
Physical Controller Platform
Platform

Figure 7. A simple *Virtual Configuration Hopping* setup. This instance includes dynamic reconfiguration across two virtually redundant controllers located on the same physical platform. Controller A is currently set to the active controller.

Dynamics: *Virtual Configuration Hopping* requires that all redundant functions will be able to receive and generate output to the appropriate systems, as control will need to be dynamically switched between those functions. In addition, it may be necessary to ensure that the dynamic switching between functions is bumpless. For example at the time of switching the multiple redundant functions may be in different states; thus, the switch between functions results in an unintended switching of states.

Implementation: When implementing *Virtual Configuration Hopping* it is important to consider the time it will take for a compromised function to cause damage. For example, a turbine in a nuclear reactor can potentially be damaged in a matter of seconds. Alternatively, it may take several minutes or hours to steer a ship far enough off course to be considered damaging. In addition, the sophistication involved in switching between redundant system functions depends on the sophistication of the cyber attack to be prevented. For example, switching between redundant functions in a round robin fashion may disrupt a cyber attack that is just trying to transmit damaging commands quickly. However, a more sophisticated attack may be able to detect the switching patterns. This information could then potentially be used to issue commands that ultimately cause damage through controlled thrashing that occurs every time a switch from the compromised function to a non-compromised component occurs. It is also important to decide how much control is given to administrators to change the frequency of hopping as well as the algorithm used to control the switching order.

Example Resolved: The concerned e-commerce business determines that the system responsible for receiving and displaying information can be virtualized quickly at minimal costs and decides to use *Virtual Configuration Hopping*. The E-commerce site sets up a virtualized environment to run multiple copies of the system. In addition, the E-commerce site obtains a video surveillance application from another vendor and adds that into its virtual environment. Once this has been set up, the e-commerce business determines that it should be concerned regarding the possibility of the credit card information stored at the protected site being stolen. It then determines that it would take an intruder at least 10 minutes to download all of the credit card information. The system is then set-up to hop between the virtualized system functions every 5 minutes. However, during switching the video feed appears to exhibit some slight distortions (i.e. it is bumby). To mitigate this effect, *Virtual Configuration Hopping* system is updated to provide a smooth (i.e. bumpless) stream.

Variants: Physical Configuration Hopping.

Known Uses: [Jones and Horowitz, 2011, Jones, Nguyen, and Horowitz, 2011; Jones and Horowitz, 2012; Babineau, Jones, and Horowitz, 2012]

Consequences: The following benefits may be expected from applying this pattern:

- Prevent a compromised system function from cyber attack before it is able to compromise the mission objectives; prevention can occur independently--as well as and faster--methods used for detection, isolation, and restoration
- Makes the development of cyber attacks more difficult by introducing time as an element

The following potential liabilities may arise from this pattern:

- Requires multiple redundant functions
- Introduce the need for methods to ensure bumpless control
- Defeated if the frequency of hopping is too slow, or algorithm for switching is predictable

  Rick: Consequences section can use the comments I've made on Physical as well.

Related Patterns: Can be combined with *Diverse Redundancy* to potentially mitigate the risk that multiple redundant functions will be compromise

4 Architecture Selection Methodology

When applying System Aware cyber security design patterns, a system engineering team must make a number of architectural decisions regarding which design patterns to use to further protect which system functions. In order to address this architectural problem, a specific objective has been developed as the means for guiding selection, namely:

Reversing cyber security asymmetry from favoring our adversaries (small investment in straightforward cyber exploits upsetting major system capabilities), to favoring the US (small investments for protecting the most critical system functions using System Aware cyber security solutions that would require very complex and high cost exploits to defeat)

Selection of an architecture requires selection of which system functions are the most important candidates for requiring more protection, which design patterns are applicable to providing additional protection for each of the selected system functions, and which combinations of functions and design patterns serve to best reverse asymmetries as described above. This selection process can be supported by a methodology engaging a multi-discipline systems engineering team, as shown below.

- Identify and prioritize critical system functions to protect - Blue Team (Consists of designers of the system being protected)
- Identify candidate highly asymmetric attack vectors – Red Team
  (Consists of people experienced with system penetrating cyber attacks)
- Select multiple design patterns for each protected function - Blue Team
  (Consists of people knowledgeable about System Aware cyber security design patterns)
- Determine architectures within specific defender budgets - Green Team
  (Consists of people with system cost analysis experience)
- Select specific architecture based on comparison of evaluations of the defenders' cost to protect versus change in attackers' costs to develop and evaluate new exploits – (Blue/Red/Green Teams)

This methodology was used to explore the application of System Aware cyber security design patterns to an unmanned airborne vehicle application, where the vehicle and its flight control system are commercial off-the-shelf products, and the aircraft was assumed to carry a variety of sensor systems to conduct data collection missions (i.e., optical/IR cameras, radar and signal collection subsystems).

A workshop was organized to support explorations related to the first prototype application of System Aware cyber security. An RT-28 task was structured to support the workshop. The Georgia Tech Research Institute was brought on board to work with UVa to develop the prototype architecture. The Workshop objective was to provide the technical interchanges required for the integration of the necessary technical and cost information required to support the development of alternative prototype development plans for a pilot application of the

System Aware cyber security design patterns resulting from project RT-28 to-date. The pilot application would serve to validate the analytical and simulation-based work accomplished to this point in time, and to highlight implementation issues that need to be addressed as part of implementing System Aware cyber security. The participants of the workshop were:

University of Virginia design team:
Overall Project Leader: Professor Barry Horowitz
Professors: Kevin Skadron, Ron Williams, and Peter Beling
Research Scientist: Carl Elks
Graduate Students: Rick Jones, Barbara Luckett

Georgia Tech integration and test team
GT Project Leader: Dr. William Melvin, Director, Georgia Tech Research Institute (GTRI), Sensors and Electromagnetic Applications Laboratory (SEAL)
Michael Brinkmann, Chief, GTRI/SEAL, Sensor Systems Engineering Division
James Perkins, Head, RF Systems Branch
Tom Owens, Research Engineer
Johanno LoTempio, Research Engineer
Joshua Hamilton, Research Engineer
Dr. S. Lawrence Marple, Chief Scientist, GTRI/SEAL

Following the architecture selection methodology for System Aware cyber security, the following results were developed by the UVa/GTRI team:

- Three categories of critical system functions to protect:
  - Platform Subsystems (platform control, navigation, mission control, air/ground comm.),
  - Sensor subsystems,
  - Human support subsystems
- Most highly asymmetric attack risks:
  - System parameter changes (e.g., waypoint changes, flight control system changes, surveillance mode changes, signal processing changes),
  - GPS navigation system corruption
  - Manipulation of sensor beam pointing functions
  - Display manipulation and aircraft control lock-out of the ground controller
- Which design patterns:
  - Data Consistency Checking - Control parameter assurance using airborne data consistency process for critical flight control parameters
  - Analog components – use of spread spectrum/low data rate air/ground radio system for security related coordination
  - Diverse redundancy - waypoint assurance for navigation system assurance using existing onboard diverse navigation sources (back-up aircraft INS, barometric altimeter, camera supporting INS)

  - Data Consistency Checking – Comparing airborne navigation and sensor Doppler information for sensor pointing assurance
- Types of architectural evaluations to be conducted in prototype program
  - Simulation for ground system portion of the architecture
  - Rapid prototyping (version 1), HW/SW in the loop emulation evaluations for airborne portions of the architecture
  - Rapid Prototyping (version 2) with live flight evaluations.
  - Requires metric development and corresponding measurement capabilities for both ground and air portions of the architecture

5 Summary and Conclusions

RT-28 has produced a novel concept for providing added cyber security solutions that directly protect mission critical system functions. The concept has been documented and peer reviewed by academia, industry and DoD. The concept involves development of reusable design patterns. Several example design patterns were developed and evaluated. Refereed conference papers and an archival journal paper document specific design patterns that were developed within the RT-28 project. In addition, an existing security design pattern format consisting of 12 categories of descriptive material for presenting a design pattern was adopted and, as examples, four(4) System Aware design patterns were documented in the recommended manner. Finally, an architectural selection methodology was established for supporting system engineers in selecting system functions requiring more protection and corresponding design patterns to afford that protection. A selection criteria related to changing asymmetries from being advantageous to attackers to being advantageous to defenders was adopted as a critical element of the architecture selection methodology. The RT-28 project was completed through an interactive architecture selection process involving a systems engineering team from UVa and GTRI utilizing the System Aware concept, design patterns and architecture selection methodology for a UAV application. The results of that effort are the basis for continuing to advance the System Aware concept via prototype applications.

APPENDIX I – PUBLICATION

THE INTEGRATION OF DIVERSELY REDUNDANT DESIGNS, DYNAMIC SYSTEM MODELS, AND STATE ESTIMATION TECHNOLOGY TO THE CYBER SECURITY OF PHYSICAL SYSTEMS

Barry M. Horowitz
Systems and Information Engineering
University of Virginia
Charlottesville, VA, 22904
Phone: 434-924-0306
Email: bh8e@virginia.edu

Katherine M. Pierce
Systems and Information Engineering
University of Virginia
Charlottesville, VA, 22904
Email: kmp7ef@virginia.edu

ABSTRACT

As exemplified in the 2010 Stuxnet attack on Iranian nuclear facilities, cyber attackers have capabilities to embed disruptive infections into equipment that is employed within physical systems. This paper presents a cyber security design approach that addresses cyber attacks that include modification of operator displays used for support in managing automatic control systems. This class of problems is especially important because our nation's critical infrastructures employ such systems. The suggested design approach builds upon fault tolerant and automatic control system techniques that, with important and necessary modifications, provide the basis for providing improved cyber security. In particular, the appropriate combination of diversely redundant security designs coupled with system dynamics models and state estimation techniques provide a potential means for detecting purposeful adjustments to operator displays. This paper provides a theoretical approach that employs diverse redundancy for designing such solutions and a corresponding set of examples with simulation-based results. In addition, the paper includes a discussion of important implementation requirements for greater assurance of such physical system security solutions.

1 INTRODUCTION

As advances in technology permit automatic control of an increasing number of the functions of physical systems, the opportunity for cyber attacks that include exploitation of such automation capabilities also increases. This class of cyber attacks is of special importance because our nation's critical infrastructures employ highly automated physical systems (e.g. electric power generation and water purification). For example, in the 2010 Stuxnet attack [Falliere, Murchu, and Chien, 2011], an embedded infection was used to successfully damage a large number of centrifuges in Iran (estimated to be 10 percent of the available capacity) [Albright, Brannan and Walrond, 2010]. While the application of perimeter security technologies—such as firewalls, encryption, and advanced user authentication—has been utilized to help manage the likelihood of cyber attackers exploiting highly automated physical systems, the rate of successful attacks continues to be problematic. Furthermore, perimeter solutions do not adequately address insider attacks and supply chain initiated attacks. As a result, it has been recognized that perimeter security needs to be augmented by other approaches for addressing potential cyber attacks [Wulf and Jones, 2009].

Frequently, as a means for added operational assurance, highly automated systems include the presentation of information that permits human operators to take controlling actions when the automated system appears to be operating in an abnormal manner. While the design of automation over-ride assurances historically has not been motivated by cyber attack threats, they nonetheless provide a mechanism for responding to certain cyber attacks. For example, the operation of a turbine may be automatically controlled, but operators can observe critical information regarding the turbine's operation, such as vibration levels, temperature, and rotation rate. If the operator observes measurements that are outside the designated region of proper operation, specific manual actions can be required of the operator in order to avoid undesirable consequences [Jones, Nguyen, and Horowitz, 2011]. However, as was the case in the Stuxnet attacks, a cyber attacker can not only manipulate a physical system's performance through infections in its control system, but can also manipulate data presented to operators; data that can, when utilized within standard operating procedures, either stimulate inappropriate control actions or serve to prevent needed control actions on an operator's part. In the case of the turbine example, a successful cyber attack can result in indications to operators that would imply that all is well when it is not, or indications that would call for disruptive operator action when, in reality, none is required (e.g., unnecessarily shutting down the turbine). Note that it is quite typical for operator displays to be designed for simplicity [Desai, 2010]. This is done to ensure that critical manual actions will not be delayed or confused by human limitations related to viewing and interpreting too much information. For example, automobiles are designed to provide a driver with observations of just a few of the many available engine state measurements that could be made available for viewing because providing additional information could cause confusion while offering little to no benefit.

A suggested approach for addressing cyber attacks that include purposeful manipulation of operator displays is to embed security features within the physical system being protected, including features that can detect inconsistent information within the protected system. For example, as shown by Jones and Horowitz [2012] in their System-Aware cyber security architecture concept, one can build on concepts developed in the fault-tolerant systems community by having diverse redundant elements perform the same system functions (e.g., three different manufacturer's turbine controllers deriving turbine control signals) and comparing their outputs to support detection of an infected element that is performing improperly as well as providing for system restoration that could include immediate elimination of the isolated element from operational use. This paper extends the System-Aware architecture concept through the introduction of a class of security solutions that are derived from concepts developed by the fault tolerant and automatic control systems communities. In particular, the use of diverse redundancy coupled with utilization of mathematical models of the dynamic behavior of physical systems and state estimation techniques is suggested as the basis for system architectures that can be employed to detect situations where information displays for system operators are being manipulated as part of a cyber attack. While the specific techniques being suggested are well known, their integration for cyber security purposes is novel, and offers the potential for an important new approach for assuring information integrity within physical systems.

Section 0 of this paper provides a general description of the suggested new class of cyber security solutions, referred to as information consistency-checking. Section 0 provides a specific theoretical example that is utilized to explore the performance of the suggested information consistency-checking solution based upon simulation results. Section 4 discusses issues related to the potential for cyber attackers to develop attacks that are responsive to the suggested information consistency-checking solution. Section 5 discusses critical design issues that must be resolved in order to transition information consistency-checking solutions into real world applications. Section 6 provides some general observations regarding the viability and potential for application of information consistency-checking as a cyber security technique for physical systems and the role of systems engineering in the development of solutions.

2 Detecting Information Inconsistencies

The premise of this paper is that, under certain assumptions, cyber attacks that create erroneous and misleading operator displays can be automatically detected and that these detections can provide the basis for responsive defensive measures. In this section it is shown how diversely redundant security designs, system dynamics models and state estimation techniques can be integrated to detect important information inconsistencies in an automatic control system while providing acceptable (near-zero) false alarm and missed detection rates. While the control system community has developed more generalized techniques regarding state estimation than are considered in this paper, in order to place the desired emphasis on the novel concept for cyber security solutions, the considered physical systems are limited to those where the system states of interest are directly measurable. Furthermore, while not necessary, the examples in Section 3 only consider systems that can be adequately modeled as linear. The purpose of the example linear system dynamics model used in Section 3 is to present specific results that serve to illuminate critical system security design issues.

The automatically controlled system to be secured is assumed to be mathematically modeled, and is represented by the following discrete-time mathematical equations:

(1) $\underline{x}(k+1) = f(\underline{x}(k), \underline{u}(k), \underline{\omega}(k), k)$

(2) $\underline{y}(k) = g(\underline{x}(k), \underline{v}(k))$

$\underline{x}$ is the n-vector state of the physical system to be protected. $\underline{u}$ is the l-vector of control inputs into the protected system: utilized for the purposes of state estimations. $\underline{y}$ is the m-vector of measurements from the protected system. $\underline{\omega}$ is a n-vector of stochastic perturbations that impact system dynamics. $\underline{v}$ is a m-vector of measurement-related perturbations. k is the interval of time used for the discrete-time model of the system being protected. f and g are discrete-time functions used to model system dynamic performance. As stated above, in order to simplify explanations regarding cyber security, for the remainder of this paper g will be assumed to provide direct measurements of a subset of the components of the full state $\underline{x}(k)$ —corrupted with corresponding measurement noise.

For the purposes of the cyber security solution to be presented, consider the component states of the overall n-vector system state, $\underline{x}(k)$, as being divided into three distinct categories of states with corresponding categories of measurements:

$$(3)\quad \underline{x}(k)=\begin{bmatrix} \underline{x}_1(k) \\ \underline{x}_2(k) \\ \underline{x}_3(k) \end{bmatrix},\ \underline{y}(k)=\begin{bmatrix} \underline{y}_1(k) \\ \underline{y}_2(k) \\ \underline{y}_3(k) \end{bmatrix}$$

The category of states $\underline{x}_3$ consists of those component states that are not measured (i.e., states for which there are no measurements $\underline{y}_3(k)$). The category $\underline{x}_1$ consists of those directly measured component states for which information is presented to operators in support of their system management responsibilities. The states comprising $\underline{x}_2$ are directly measured for other system-related purposes, such as maintenance support, and are assumed as not being presented to the operator. The operator display presents estimates of $\underline{x}_1$, that will be referred to as $\hat{\underline{x}}_{1lt}$, which can consist of either the time series of direct measurements of $\underline{y}_1$, or a time series of estimates of $\underline{x}_1$ that are expected to provide additional accuracy, derived from the integrated time series of measurements of $\underline{x}_1$. The nomenclature $\hat{\underline{x}}_{1lt}$ is used to highlight the point that, from a cyber security viewpoint, these estimates of $\underline{x}_1$ are treated as less trusted than would normally be the case, because the measurements or estimates of these states are more likely to be manipulation targets of a cyber attack than measurements and estimates of other states that are not presented to operators for decision support. Correspondingly, $\hat{\underline{x}}_{1mt}$, is defined as state estimates of $\underline{x}_1$ that are solely based upon y_2. These estimates, while likely to be less accurate than $\hat{\underline{x}}_{1lt}$, are treated as more trusted, because they are entirely based upon measurements of component states, $\underline{x}_2$, for which information is not presented to the operators, and the presumption that cyber attackers will not choose to manipulate this information in recognition of the fact that they do not play a direct role in influencing operator decisions. Furthermore, it is assumed that the system dynamics model represented by f is such that component states of $\underline{x}_2$ are sufficiently coupled to the component states in $\underline{x}_1$ so that measurements of $\underline{x}_2$ can be utilized as a sufficient basis for providing useful estimates, $\hat{\underline{x}}_{1mt}$ with regard to providing an effective cyber security capability.

Systems that satisfy these conditions allow for the utilization of two diversely redundant methods for deriving estimates of the state vector $\underline{x}_1$. The first method utilizes the direct measurements of the state vector $\underline{x}_1$ for deriving estimates $\hat{\underline{x}}_{1lt}$. The second method utilizes the direct measurements of the state vector $\underline{x}_2$ as the basis for deriving estimates $\hat{\underline{x}}_{1mt}$. This diversity serves to provide a basis for consistency-checking regarding the information presented to operators. As illustrated in Figure 1, one can configure an information consistency assessment as part of an embedded security solution for a physical system. This type of solution can be utilized to address two classes of cyber attacks. First, is where the system being protected is attacked in a manner that results in undesired behavior that would normally stimulate an operator over-ride, but the operator never initiates the over-ride command because the cyber attack includes adjustment of operator display information to provide the appearance of normal behavior. In this case, the estimates $\underline{\hat{x}}_{1mt}$ based upon measurements of $\underline{x}_2$, provide the basis for detecting that $\underline{\hat{x}}_{1lt}$, the information presented to the operator, is incorrect; i.e., the system is actually performing in an out-of-normal band of operation. Second, is where the system is operating normally, but the cyber attack involves adjustment of operator display information to provide the appearance of undesired behavior requiring operator intervention that causes undesirable consequences, such as termination of operation. Similar to the first case, the $\underline{\hat{x}}_{1mt}$ estimates provide the basis for detecting that the less trusted information, $\underline{\hat{x}}_{1lt}$, presented to the operator is incorrect; i.e., in actuality, the system is operating properly.

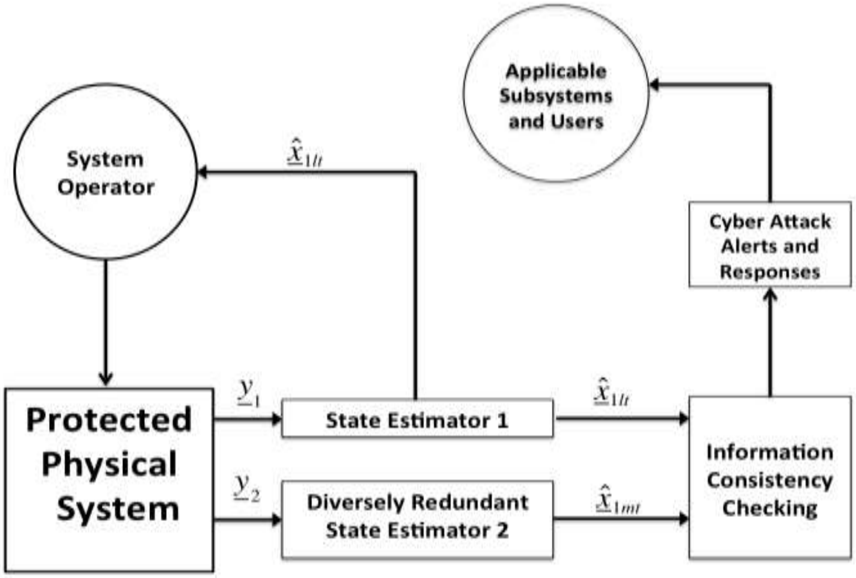

Applicable Subsystems and Users
System Operator
$\underline{\hat{x}}_{1lt}$
Cyber Attack Alerts and Responses
Protected Physical System
$\underline{y}_1$
State Estimator 1
$\underline{\hat{x}}_{1lt}$
$\underline{y}_2$
Diversely Redundant State Estimator 2
$\underline{\hat{x}}_{1mt}$
Information Consistency Checking

Figure 8. A Block Diagram Representation of a Data Consistency-Checking Embedded Security Solution.

A significant body of work exists regarding the design of state estimation techniques for systems; this includes applications to linear systems [Grewal and Andrews, 1979], non-linear systems [Slotine 1984], and accounting for a variety of assumptions regarding the qualities of the system being controlled, such as observability [Griffith and Kumar, 1971]. In addition, work has been done on application of state estimation techniques for hardware fault isolation in response to non-purposeful causes of failure [Kobayashi and Simon, 2003]. However, the authors are unaware of literature that suggests the integration of diverse redundancy, system dynamics models and state estimation techniques as part of a cyber security solution regarding the assurance of information consistency within a physical system.

While utilizing the diverse redundancy of indirect estimates as a basis for assuring information consistency provides the opportunity for a set of new cyber security solutions, there are a number of design and implementation issues that must be addressed in order to successfully provide such solutions. These issues are discussed in Sections 4 and 5. The following section provides an example application of the data consistency-checking cyber security concept.

3 Theoretical Examples

In order to expand upon the discussion in Section 0, this section presents an example of applying information consistency-checking to a linearly modeled physical system.
Consider the special case of a linear system represented by equations (4) and (5) below:

(4) $\underline{x}(k+1) = A\underline{x}(k) + B\underline{u}(k) + \underline{\omega}(k)$

(5) $\underline{y}(k) = C\underline{x}(k) + \underline{v}(k)$

A, B, and C are known n˙ n, n´ 1, and m˙ n dimensioned fixed matrices utilized to model the physical system to be secured. $\underline{\omega}(k)$ is a n-vector, zero-mean, white Gaussian, stationary stochastic system disturbance process with uncorrelated vector components represented by a diagonal covariance matrix of standard deviations, σ_{ii} with $i = 1, \mathrm{n}$. $\underline{v}(k)$ is a m-vector, zero mean, white Gaussian, stationary stochastic measurement noise process with uncorrelated vector components represented by a diagonal covariance matrix of standard deviations, σ_{jj} with $j = 1, \mathrm{m}$.
For this system of equations, a Kalman filter [Grewal and Andrews, 1979] provides the expected minimum mean squared error estimate for the state $\underline{x}$ during time intervals k. For the suggested diversely redundant cyber security solution, two different values for C would be utilized to represent the diverse approaches for gathering measurements from the system (this can be seen in Figure 1):

(6) $C = C_1$ representing the provisioning of direct measurements of $\underline{x}_1$ that are used for

developing estimates $\underline{\hat{x}}_{1lt}$ for presentation on the operator's display.

(7) $C = C_2$ representing the provisioning of measurements of $\underline{x}_2$ that are used for

developing estimates $\underline{\hat{x}}_{1mt}$ of $\underline{x}_1$, to be used for data consistency-checking.

Sequential comparison of these diverse redundant estimates provides the basis for detecting a potential cyber attack. That is, a potential cyber attack is declared, and corresponding system defensive responses are initiated, if the distance between the diversely derived estimates ($\underline{\hat{x}}_{1lt}$ and $\underline{\hat{x}}_{1mt}$) of $\underline{x}_1$ exceeds a designated attack detection threshold.

3.1 Specific Examples

This section provides numerical values for the linear control system example. Three operational cases are evaluated through simulation:

- Normal system operation.
- A cyber attack that modifies the operator's display, with the intent of causing a disruptive operator action.
- A cyber attack that changes the control objective of the system, thereby causing a disruption in performance, and also modifies the operator's display to prevent observation and intervention in response to the change in system performance (Stuxnet-like attack).

For the purposes of analysis, it is assumed that the system being protected utilizes an LQG (Linear Quadratic Gaussian) feedback control law [Athans, 1971] to regulate its states. Consider the specific example of a linear system modeled as consisting of four scalar component states, x_a, x_b, x_c, and x_d. Based on the definitions provided in Section 2, it is assumed that

(8) $\underline{x}_1 = x_a$

(9) $\underline{x}_2 = [x_b, x_c, x_d]^T$

For this example, the specific elements of equations (4) and (5) are

(8) $$A = \begin{bmatrix} 1.0 & 1.0 & -0.02 & -0.01 \\ 0.01 & 1.0 & -0.01 & 0 \\ 0.2 & 0.01 & 1.0 & 1.0 \\ -0.01 & 0.02 & -0.01 & 1.0 \end{bmatrix}$$

(9) $B = \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix}^T$, for a scalar input u

(10) $C_1 = [1 \quad 0 \quad 0 \quad 0]$

(11) $$C_2 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(12) $\sigma_{ii} = 0.2 \forall i$, $i = 1 \text{to}\, 4$

(13) $\sigma_{jj} = 0.2 \forall j$, for C_1 $j = 1$, for C_2 $j = 1 \text{to}\, 3$

For the attack detection system, a sliding window detector [Castella, 1976] of length L is used to operate on the time series of the differences between the diverse estimates ($\underline{\hat{x}}_{1lt}$ and $\underline{\hat{x}}_{1mt}$) for $\underline{x}_1$, ($\underline{x}_1$ is a scalar for this example). In particular, the example detection algorithm declares an attack when the sequence of differences between half of the last L diverse estimate pairs for $\underline{x}_1$ are consistently different in value by more than a selected threshold value, τ . The threshold value, t , would be selected based upon the desired missed detection/false alarm values for the security system and the differences that one anticipates observing between the diverse methods of estimation under normal system operation and under the duress of potential cyber attacks.

The results that follow were developed through use of a discrete time simulation model of the example control system described above. This model used an LQG controller specifically based upon a quadratic objective function directed toward regulating the operator monitored state, x_a, at a selected value of 500 and the other states at values of zero (all with equal priority). In addition, to better represent realities of bounded noise, any perturbation values in the simulation model that exceeded three standard deviations from the assumed zero mean Gaussian probability distributions were replaced by the three standard deviation value.

3.2 NORMAL SYSTEM OPERATION (NO CYBER ATTACK)

Based upon the model described above, Figures 2a and 2b present, at different viewing scales, an illustrative simulation result for the time series of Kalman filter estimated values $\underline{\hat{x}}_{1lt}$ that would be presented to an operator (using C_1 as described above). Using the diversely redundant estimates derived from measurements related to the C_2 measurement matrix (equation 11 above), Figure 3 presents the alternate time series of estimates, $\underline{\hat{x}}_{1mt}$, based upon measurements of the three state components that comprise $\underline{x}_2$. Figure 4 presents the time series of differences between the diverse estimation approaches $\underline{\hat{x}}_{1lt}$ and $\underline{\hat{x}}_{1mt}$. This time series of differences requires the application of specific detection criteria for declaring a cyber attack. The bold lines in Figure 4 serve to illustrate the point that, for the 300-point sample simulation case, 4 units bounds the sequence of differences between the diverse estimates of $\underline{x}_1$. This points to the possibility of using a difference threshold of $t = 4$ as the basis for the detection of a cyber attack. Of course, a more substantial assessment of false alarms and missed detections, including accounting for the sliding window detection process, would be required to select a suitable threshold value for t . This selection process is discussed later in the paper.

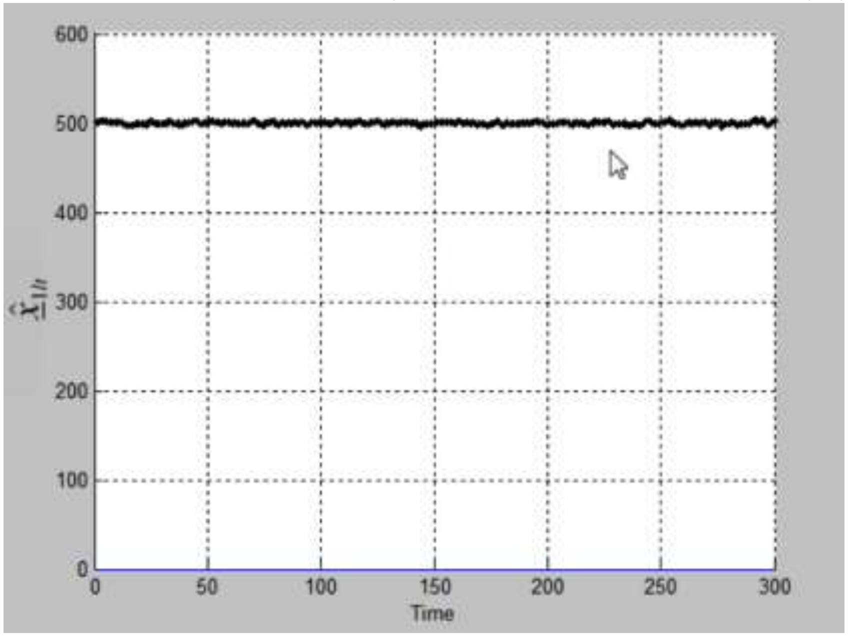


Figure 9a. Simulation-based time series of operator observations, $\hat{\underline{x}}_{1|t}$, for $\underline{x}_1$ being regulated to a value of 500 units, under normal system operations.

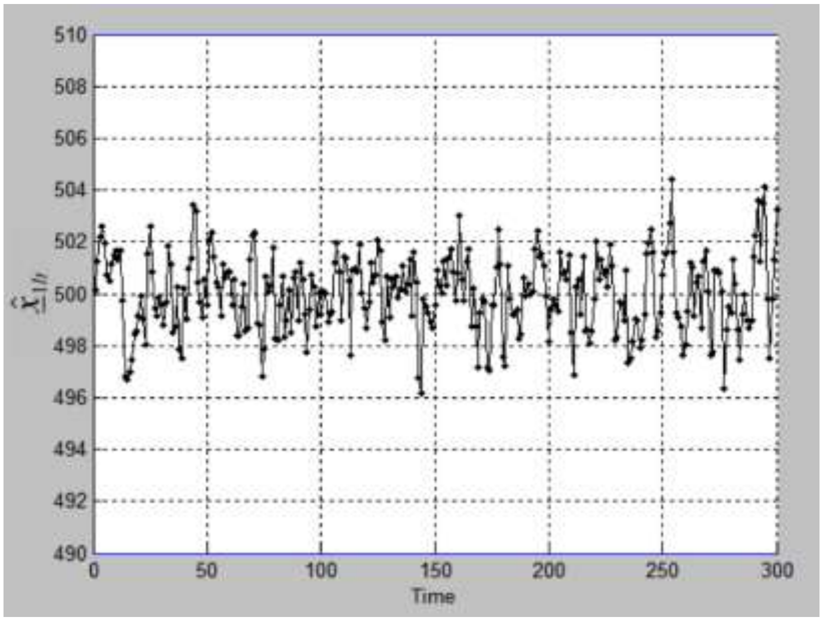


Figure 2b. A finer scaled presentation of Figure 2a.

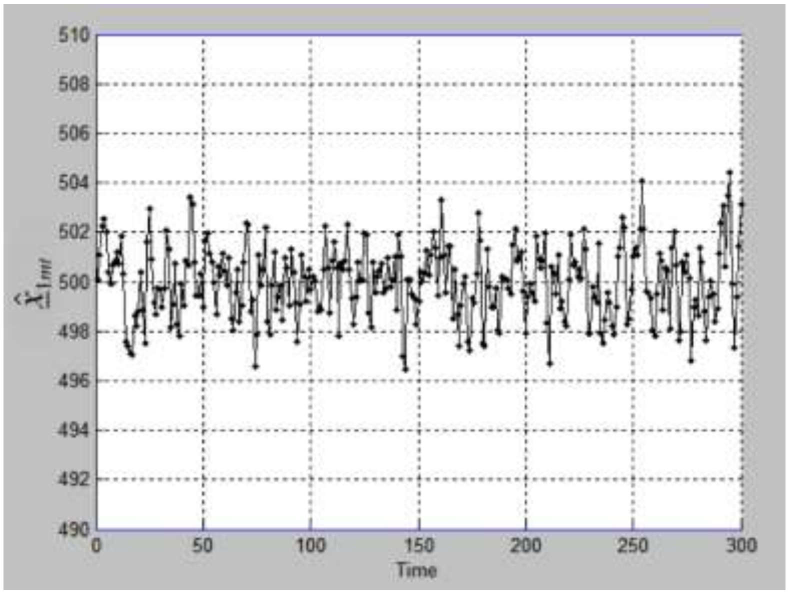


Figure 10. Simulation-based time series of diversely redundant estimates $\hat{\underline{x}}_{1mt}$, under normal system operations.

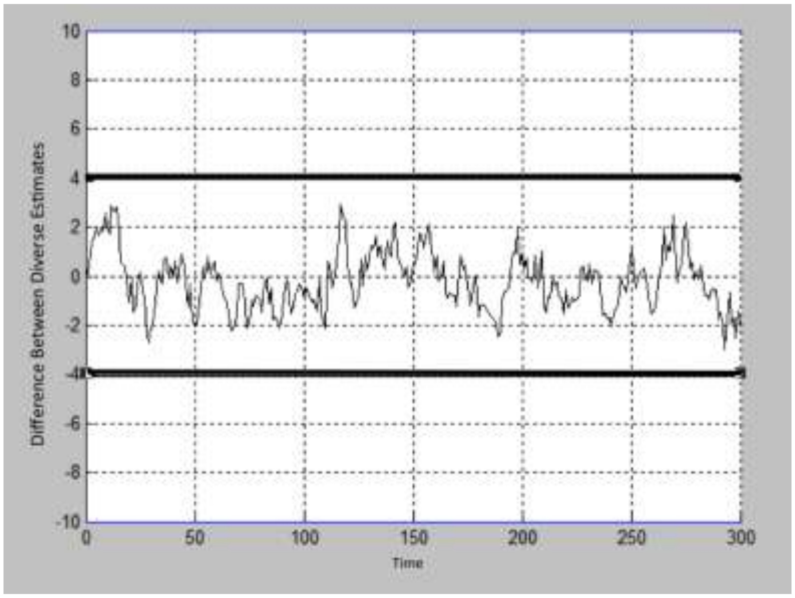


Figure 11. Simulation-based time-series of differences between the operator displayed and diversely redundant estimates of $\underline{x}_1$, under normal system operation.

3.3 Cyber Attack Scenarios

Operator Display Attack: Consider the case where a cyber attacker modifies the operator display through a replay of prior data, including manipulation of the numeric values of $\hat{\underline{x}}_{1lt}$, so as to provide the illusion of state $\underline{x}_1$ operating outside the acceptable envelope for performance. Using the simulation model discussed in Section 3.1, a purposeful addition by an attacker to the values of displayed information to the operator of 5 units is simulated, resulting in a time series of differences between the diverse estimates results as shown in Figure 5. Depending upon the actual size of the change in value selected by an attacker, and comparisons with the normal system operation results of Figure 4, these differences can potentially provide a clear opportunity for detection of the attack. The value change that an effective attacker would need to use would depend on the operational procedures for an operator over-riding the automatic system. If the necessary value change for influencing operator interaction is sufficiently large compared to the differences between diverse estimates under normal operations, attacks can be detected using detection thresholds that avoid false alarms. For the purpose of illustration, Figure 5 shows a bold line representing a τ value of 4, which based upon inspection, could potentially provide a high likelihood basis for detecting the attack while, according to the limited information of Figure 4, avoiding false alarms. If, in order for the replay attack to provoke operator response, the system being attacked required a minimum value change for $\underline{\hat{x}}_{1lt}$ of, for example, 10 units, then a larger value for τ could be usefully employed—potentially resulting in reductions in both missed detection and false alarm rates.

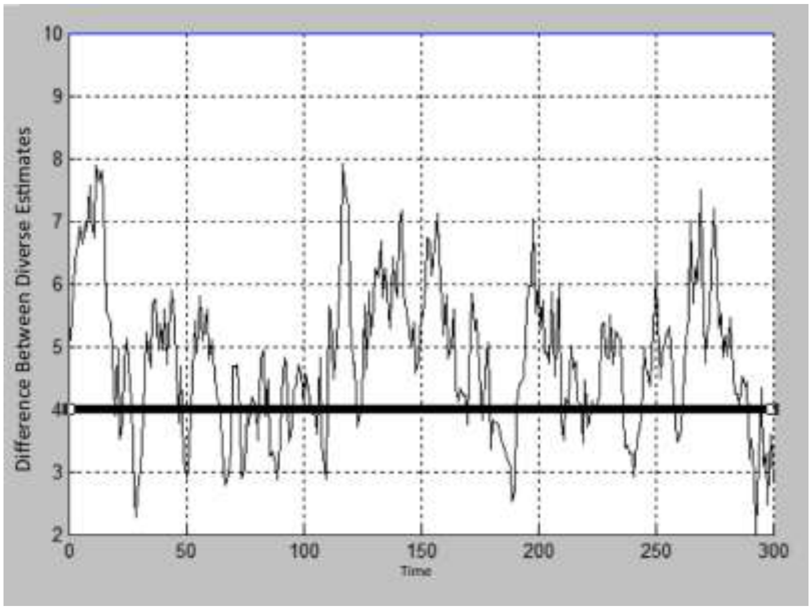


Figure 12. Simulation-based time-series of differences between the operator displayed and diversely redundant estimates of $\underline{x}_1$, under conditions of a 5 unit displacement cyber attack on the operator display information.

Control System Attack: Consider the case where an attacker is capable of changing the regulation point for the controller so that state $\underline{x}_1$ is regulated at an undesirable operating point and the operator display is correspondingly manipulated to make the situation appear as normal. Based upon the simulation model, Figure 6 presents an illustrative time series for true values of $\underline{x}_1$ for this case. Figure 7 presents the attacker adjusted time series displayed to the operator ($\underline{\hat{x}}_{1lt}$), and Figure 8 provides the corresponding simulation result for the time series of estimates $\underline{\hat{x}}_{1mt}$ of the state $\underline{x}_1$ based upon measurements of $\underline{x}_2$. Figure 9 presents the time series of differences from comparison of the diverse methods for estimating of $\underline{x}_1$. As in the first example cyber attack case, the potential opportunity to detect such attacks while avoiding false alarms is evident by inspection, and varies depending upon the forced deviation of the state, $\underline{x}_1$, that is required to create a serious problem for the system under attack.

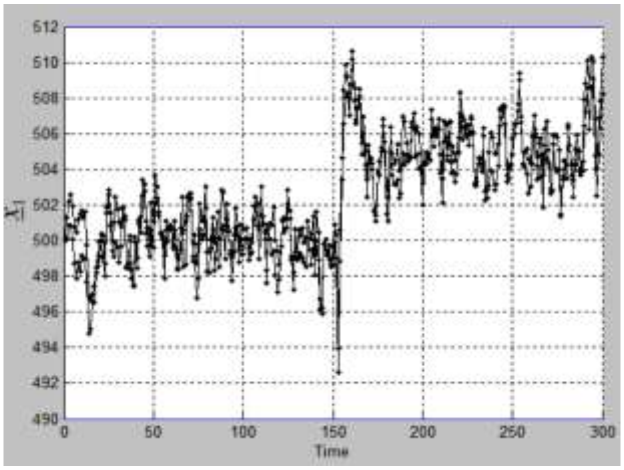


Figure 13. Simulation-based time-series of state $\underline{x}_1$ resulting from a cyber attack that adjusts the regulation objective for state $\underline{x}_1$ by 5 units.

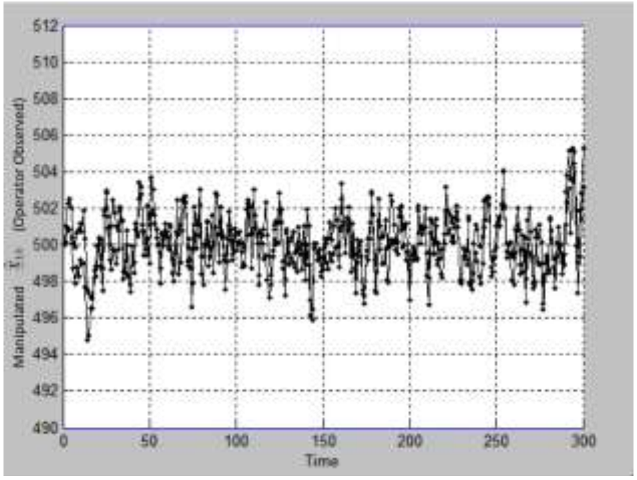


Figure 14. Simulation-based time-series of manipulated operator display information of $\hat{\underline{x}}_{1lt}$ during a cyber attack that also adjusts the regulation objective for state $\underline{x}_1$ by 5 units.

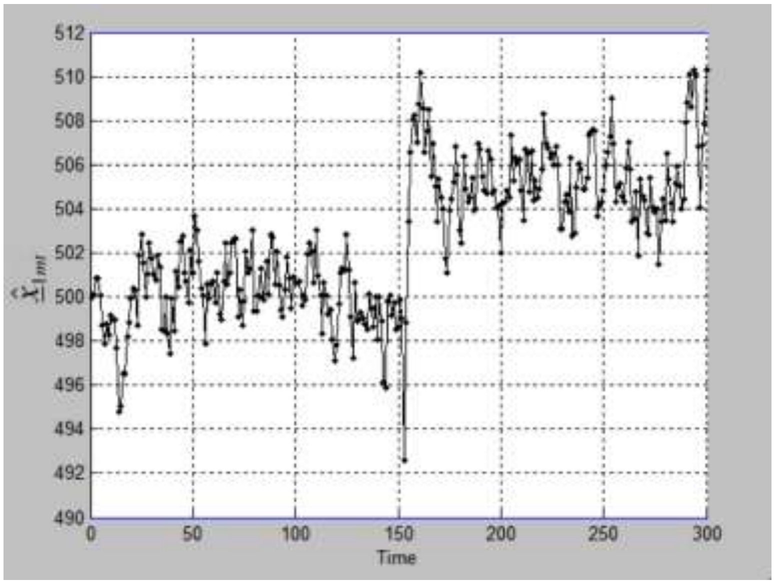


Figure 15. Simulation-based time-series of estimated values of $\hat{\underline{x}}_{1mt}$ during a cyber attack that adjusts the regulation objective for state $\underline{x}_1$ by 5 units.

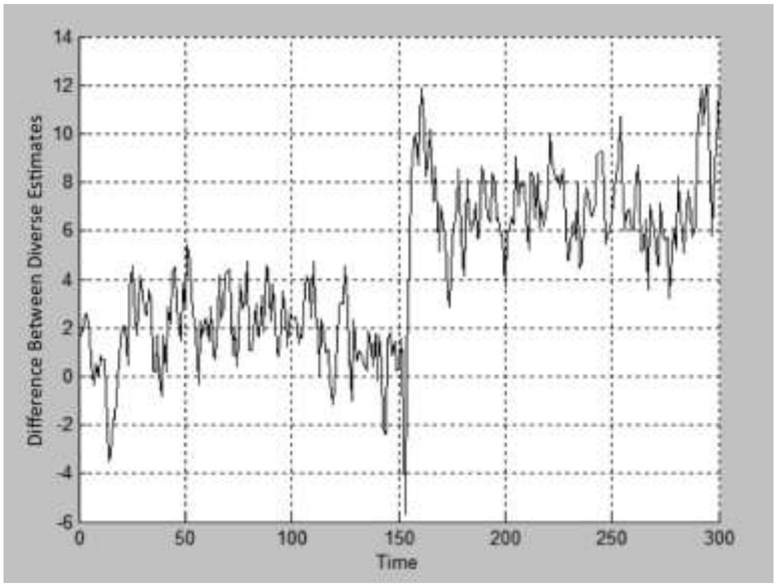


Figure 16. Simulation-based time-series of the differences between diversely redundant estimates of $\underline{x}_1$ during a cyber attack that adjusts the regulation objective for state $\underline{x}_1$ by 5 units.

Detection Threshold Considerations: Using the simulation model to aid in exploring some of the issues surrounding false alarms and their relationship to selection of values for t, for each of the three scenarios discussed above (normal, display attack and controller/display attack), an example time series consisting of 150,000 diverse state estimates was generated as the basis for deriving a corresponding sample of sequential differences between the diverse estimates of $\underline{x}_1$. The simulation results to be discussed below are based upon a cyber attack detection criteria being applied to these samples involving the use of a 30 point sliding window detector for cyber attack detection and a specific detection criteria of 15 out of 30 consistent differences in the diverse estimates of $\underline{x}_1$ exceeding a set value for t. Utilizing the simulation results for the normal operation scenario, through an iterative evaluation involving the decrementing of the value of t in integer increments, the minimum threshold value $\tau_{\min}$ that would yield no false alarms during the example simulated period was determined; i.e., a lower threshold value than $\tau_{\min}$ results in one or more false detections. For the specific example case, for which a time slice of 300 out of 150,000 points is presented in Figure 4, $\tau_{\min} = 5$; i.e., if declaration of a cyber attack requires 15 of the 30 most recent diverse estimates of $\underline{x}_1$ to be different by 5 units or more, this particular time series would yield no false attack detections, and a reduction in the threshold to 4 units or less would yield at least 1 false alarm over the 150,000 point sample time series. For the 300 points presented in Figure 4, $t = 4$ seems to be sufficient to avoid false alarms. However, analysis of the larger time series, consisting of 150,000 points, reveals that in order to avoid false alarms a detection criterion of t ³ 5 is required. While one can suggest a set of procedures for conducting simulation experiments and statistical tests to more confidently determine $\tau_{\min}$, later, in Section 5, a discussion is provided regarding the need to avoid over-dependence on the dynamic models in addressing the control of false alarm rates. Instead, in Section 4 the authors point to using model-guided field tests supported by model-based analysis for determining $\tau_{\min}$.

In order to better understand and illuminate the sensitivity of the value of $\tau_{\min}$ to the detection system's design parameters, simulation cases were conducted for a range of sensor performance σ_{jj} and input noise σ_{ii} (both ranging from 0.2 to 1.0), a range of sliding window lengths (and corresponding delays in detection associated with $L = 10,\ 20 \text{ and } 30$). Furthermore, two additional examples of C_2 measurement matrices are considered; one that corresponds to $\underline{x}_2 = x_b$, and the other that corresponds to $\underline{x}_2 =$ é ë $x_b \quad x_c$ ù û^T. Figures 10 and 11x present the values for $\tau_{\min}$ derived from an example set of 150,000-point exercises of the simulation model over the range of considered assumptions. These graphs point to the fact that there is a significant range of performance related to false alarm rates that depends upon the selection of measurements to utilize for diverse redundancy, the accuracy of these sensors, and the details of the design of the detection algorithm for declaring a cyber attack. Note the range is large because a scalar control input is being utilized to regulate four states with equal priority. As suggested in the discussions related to display system and control system attacks, these false alarm-focused results need to be related to the impact that the value of t also has on the rate of successful attack detections which, in turn, is influenced by the size of information adjustments that would be required for attackers to stimulate meaningful operator and control system performance deteriorations. Large size attacker information adjustment requirements would ease the trade-off regarding missed detections and false alarms. A desirable value for t would be both much greater than $\tau_{\min}$, and far lower than the size of the information changes required for achieving effective outcomes for an attacker.

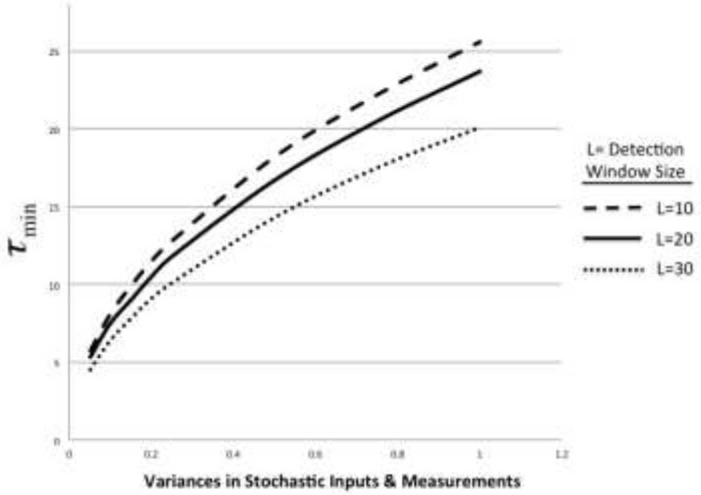


Figure 17. Sensitivity of threshold value $t_{\min}$ to the variances of system input and measurement noise, and selected window size of the sliding window detector (measured state $\underline{x}_2 = \begin{bmatrix} 0 & x_b & x_c & x_d \end{bmatrix}^T$).

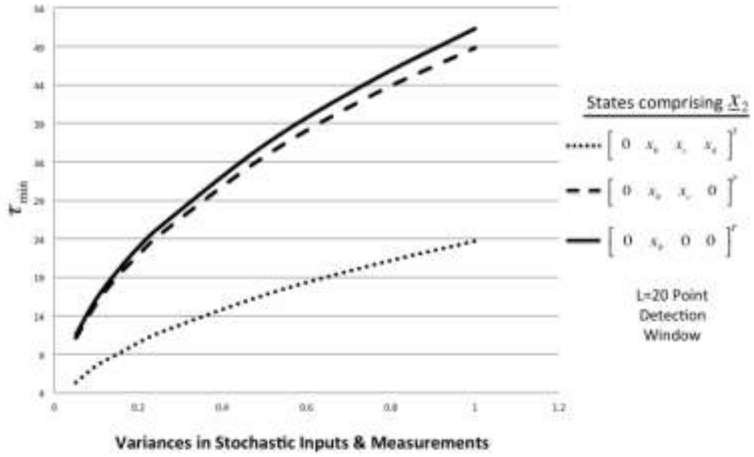


Figure 18. Sensitivity of threshold value $t_{\min}$ to the variances of system input and measurement noise and the states comprising $\underline{x}_2$.

These examples serve to indicate that for systems that satisfy the indicated conditions, the integration of diversely redundant designs, dynamic system models and state estimation technology offer a new, potentially effective approach for addressing cyber security. However, in addition to the already presented theoretical considerations, there is an important set of issues related to achieving the desired enhancement of cyber security that must also be addressed, discussed in Sections 4 and 5.

4 Responsive Attack Considerations

A critical assumption in the suggested information consistency solution is that while attackers may be successful at manipulating operator display information, it would be significantly more difficult to also effectively manipulate the more trusted, diversely redundant estimates of $\underline{\hat{x}}_{1mt}$ that are used for determining information consistency. Depending on the specific designs of solutions, there are a variety of factors that would serve to support this supposition:

1. In order to create disruption through adjustment of operator display information, an attacker would need to know operational procedures regarding operator control actions as well as how to manipulate display data. Alternatively, in order to adjust the measurements of $\underline{x}_2$ in a coherent manner (i.e., a manner that is compatible with the physical process represented by the state equations for the attacked system related to their impact on estimates $\underline{\hat{x}}_{1mt}$, or to adjust the $\underline{\hat{x}}_{1mt}$ estimation process outputs, an attacker would need to know the equations governing the physical systems behavior and the stochastic nature of disturbances and measurement errors affecting the system. Such information is far more specific than the information required for adjusting operator displays as part of an effective cyber attack. This added knowledge would likely reduce the set of attackers capable of developing such an exploit. To further assure that potential cyber attackers do not have access to such information, a part of a cyber defense solution is the need to protect model related information. While it is important to recognize that such protection cannot be relied upon as the total basis for assurance, it can be part of a set of approaches that in combination serve to complicate attackers' responsive plans.
2. In order to manipulate operator display data, an attacker could insert a cyber infection at the technical points of integration between measurement and presentation apparatus. These integration points frequently occur within the controllers of automated physical systems. These points of integration can serve to provide a clear opportunity for insiders or a technology supplier to insert the needed exploit. However, the hardware and software required for developing the diverse redundant estimates $\underline{\hat{x}}_{1mt}$ do not have system control reasons for being integrated. Thus, in order to complicate cyber attacks, $\underline{x}_2$ measurements and corresponding estimates $\underline{\hat{x}}_{1mt}$, could be distributed throughout the system being protected, avoiding, where possible, integration points that would serve to simplify attackers' exploits. Such distribution would be a new requirement for fault tolerant system designs generated by the focus on cyber attacks. Typically, to control costs, the outputs from analog sensors embedded in physical systems that would relate to $\underline{\hat{x}}_{1mt}$ are routed to a common set of electronics where analog to digital conversions occur and interconnection for digital processing occurs. Such practices would need to be modified to complicate the design of a cyber attackers exploits. In addition, measurement-related components can be purchased from a variety of suppliers, making a supply chain attack more difficult to develop. Furthermore, the in-field

maintenance of the indirect measurement subsystem can be separated from the maintenance processes for the controller and display subsystems.

3. The detection process based upon comparing operator presentation information with indirect estimates of the values presented to operators can be partitioned, distributed, and monitored at multiple locations in the physical system being protected, so that required exploitations could be further complicated. This would necessarily require a cyber attacker to synchronize the outputs of a distributed exploit with the distributed detection process that is built into the protected system. To further reduce risk, additional diverse redundancy techniques can be utilized as a method for further complicating such an attack [Jones and Horowitz, 2012]. That is, the indirect estimation process can be replicated through diverse implementations, and a simple, potentially verifiable voting process, can be utilized to isolate a successfully attacked implementation. For example, the three cases of state measurement matrices suggested by the example in Section 3 could each be separately implemented and compared to detect an exploit that impacts one of the alternatives. This solution would require an attacker to successfully address multiple designs for utilizing $\underline{x}_2$ measurements.

The premise of the authors is that the integrated set of complications for attackers described above would provide significant deterrence regarding the development of exploits, and would permit the opportunity for responsive mitigation actions in the case of successful attacks.

5 Relating Model Results to Design of Consistency-Checking Security Solutions for Actual Systems

The concern at a national level regarding cyber attacks on a critical infrastructure physical system is evidenced by US government's consideration of policies that would consider the treatment of such a cyber attack as an act of war, with corresponding responses [Claburn, 2012]. Accordingly, it is evident that false alarms must be minimized, as they could create an immediate problem regarding response. In addition, false alarms would raise collateral security issues regarding common equipment used at facilities other than the specific target of the falsely detected attack. Furthermore, recognizing the difficulties of attribution for an attack [Brenner, 2009], unwarranted issues could arise regarding the identification of the perpetrators of an attack that did not actually occur. Finally, in addition to all of the national security issues that could arise, should a false alarm result in shutting down a system (e.g., turbine for electric power generation), the revenue loss to the operator of the system can be high. An approach is suggested for treating these concerns as part of designing information consistency-checking security solutions:

1. The model-based approach suggested in this paper must be supported by sensing and estimation capabilities that result in false alarms being rare events. Specifically, the idealized understanding of the system and its security should be that false alarms are unacceptable and that false alarms that occur would be due to either component failures (including human failures) or unavoidable limitations in system modeling. For example, as suggested earlier in the paper, an idealized model cannot include

probability distributions with unrealistic ranges for random variables. This concept for false alarms being calibrated as rare events implies that the actions of an attacker that would result in operational disruption must displace system states from their values under normal operations by wide margins relative to the range of estimation errors for the values of diversely redundant estimates. The degree of influence that this condition has on developing viable solutions would depend on how different the normal and unacceptable points of operation of actual fielded systems are. For example, cyber attacks aside, based on design discussions with turbine engineers, a gas turbine typically must be rotating at a rate that is approximately 10% off norm to require an operator to apply a trip action. Given that the accuracy of measurements of the rotation rate is in the range of one part in a thousand (0.1%), and that control systems for turbines reliably contain rotation rates well within 1% of the desired rate under normal operation, then the orders of magnitude differences between measurement errors, control system performance and the necessity for tripping a turbine serves, in practice, to sufficiently assure that there will be no false alarms under normal system operation. At the same time, depending on the details of parameter values surrounding the derivation of $\hat{\underline{x}}_{1mt}$, these orders of magnitude in differences provide a significant range of detection threshold values that can potentially be sufficient to make false alarms rare events while allowing for satisfactory detection rates of cyber attacks—be it an attack that would cause an unnecessary trip of the turbine, or an attack that would confuse an operator not to trip the turbine when such an action would be desired.

2. Recognizing that false alarms must be treated as rare events, even in circumstances where a system model and its corresponding security solution satisfy the false alarm/missed detection rate criteria, an extensive effort would still be required in order to validate the quality of the model in the security context of the live system. Such validations would be based upon data collections during field use, where the collections are used to assess the possibility for alarms during normal system operations. During such tests anomalous situations (e.g., unexpected operator actions) that are not accounted for in the models may arise, causing the detection threshold for a cyber attack to be violated. These test results must be used to both refine the models and to better determine the actual false alarm rate that could result from the security solution. Note that for the case of the gas turbine discussed above, the measurement rate is in the range of 25 measurements per seconds, thereby offering about 24 billion measurements during a year of operation.

3. Should the security solution prove to be acceptable for implementation, post-implementation data should continue to be collected and evaluated as a means for continuing to refine the anomalous aspects of the system model and corresponding attack detection criteria so as to minimize the possibilities for a false alarm or missed detection. Results should impact security designs for common equipment at facilities beyond the location where an anomalous event occurs.

4. Building rapid forensic evaluation capabilities into the protected system (ideally automated and real-time) would be helpful for isolating the causes of alarms: i.e. whether a situation was caused by a cyber attack or another source. Other sources causing alarms could include human errors, software bugs or malfunctioning hardware. For example, one can build upon the use of hardware fault isolation techniques as a means for discriminating between a cyber attack and a hardware failure. A forensic capability could be employed that would provide a basis for automatic testing to determine if it is the hardware that failed or a cyber attack that has the appearance of a hardware failure [Kobayashi and Simon, 2003].

6 Conclusions and Future Work

Based on the research efforts documented in this paper, the authors consider the use of information consistency-checking based upon system dynamics models and diversely redundant state estimation techniques as providing an important new option for additional cyber security for physical systems. The availability of techniques derived from the fault tolerant and automatic control systems communities provide a starting point for development of the suggested new cyber security solutions. However, specific designs and implementations of information consistency-checking solutions need to consider the impact on the development of potential responsive cyber attacks and on the extreme importance associated with avoiding false alarms while sustaining acceptable attack detection capabilities.

Development of actual solutions will require system activities in 1) system dynamics modeling; 2) state estimation; 3) security-focused analysis regarding attack scenarios, protection needs, and identification of more trusted and less trusted components; 4) sensors and measurement characterization; 5) distributed security solution designs that serve to complicate, and hopefully deter, attacks; and 6) in-field data collections regarding selection of detection thresholds and responses to achieve acceptably low false alarm/missed detection rates. One can observe that this set of efforts requires the establishment of cyber security design teams with a much broader range of skills than the traditional information assurance community members possess. This requirement is also pertinent to the broader set of System-Aware architectures for cyber security suggested by Jones and Horowitz [2012], and provides motivation for the Systems Engineering community to assume a more significant role in development of cyber security solutions.

References

D. Albright, P. Brannan, and C. Walrond, Did stuxnet take out 1,000 centrifuges, Institute of Science and International Security, 2010.

M. Athans, The role and use of the stochastic linear-quadratic-gaussian problem in control system design, IEEE Transaction on Automatic Control, Vol. 16, no. 6 (1971), 529–552.

S.W. Brenner, Cyberthreats: the emerging fault lines of the nation state, Oxford University Press, Inc. New York, 2009.

F. R. Castella, Sliding window detection probabilities, IEEE Transactions on Aerospace and Electronic Systems, Vol. AES-12, No.6 (1976), 815-819.

T. Claburn, DOD says cyber attacks may mean war, Information Week, May, 2011.

N. Desai, The tricon-based qualified parameter display system (QSPDS) in nuclear power plants, Invensys Corporation, 2010.

N. Falliere, L. O. Murchu, and E. Chien, W32.Stuxnet Dossier, Symantec, 2011.

M. Grewal and A. Andrews, Kalman filtering theory and practice, Prentice-Hall, Englewood Cliffs, NJ, 1979.

E.W. Griffith and K.S.P. Kumar, On the observability of nonlinear systems, Journal of Mathematical Analysis and Applications, Vol. 35 (1971), 135-147.

R. A. Jones and B. M. Horowitz, A system-aware cyber security architecture, Systems Engineering, Volume 15, No. 2 (2012), 224-240.

R.A. Jones, T.V. Nguyen, and B.M. Horowitz, System-Aware security for nuclear power systems, 2011 IEEE International Conference on Technologies for Homeland Security (HST), 2011, pp. 224-229.

T. Kobayashi and D. L. Simon, Application of a bank of kalman filters for aircraft engine fault diagnostics, Turbo Expo 2003, American Society of Mechanical Engineers and the International Gas Turbine Institute, Vol. 127 (2003), 461-470.

J.E. Slotine, Sliding controller design for non-linear systems, International Journal of Control, Vol. 40, Issue 2 (1984), 421-434.

W. A. Wulf and A. K. Jones, Reflections on cyber security, Science Magazine, vol. 326, 2009, pp. 943-944.

Appendix II – Presentation

System Aware Cyber Security

Presentation by Dr. Barry Horowitz, May 8, 2012

System Aware Cyber Security

Dr. Barry Horowitz
Professor, Department of Systems & Information Engineering, University of Virginia
Member, SERC Research Council

May 8, 2012

This material is based upon work supported, in whole or in part, by the U.S. Department of Defense through the Systems Engineering Research Center (SERC) under Contract H98230-08-D-0171. The SERC is a federally funded University Affiliated Research Center (UARC) managed by Stevens Institute of Technology consisting of a collaborative network of over 20 universities. More information is available at www.SERCuarc.org

Objectives for System Aware Cyber Security Research

- Increase cyber security by developing new system engineering-based technology that provides a Point Defense option for cyber security
  - Inside the system being protected, for the most critical functions
  - Complements current defense approaches of network and perimeter cyber security
- Directly address supply chain and insider threats that perimeter security does not protect against
  - Including physical systems as well as information systems
- Provide technology design patterns that are reusable and address the assurance of data integrity and rapid forensics, as well as denial of service
- Develop a systems engineering scoring framework for evaluating cyber security architectures and what they protect, to arrive at the most cost-effective integrated solution

Publications

Published:

- Jennifer L. Bayuk and Barry M. Horowitz, An Architectural Systems Engineering Methodology for Addressing Cyber Security, Systems Engineering 14 (2011), 294-304.
- Rick A. Jones and Barry M. Horowitz, System-Aware Cyber Security, ITNG, 2011 Eighth IEEE International Conference on Information Technology: New Generations, April, 2011, pp. 914-917. (Best Student Paper Award)
- Rick A. Jones and Barry M. Horowitz, System-Aware Security for Nuclear Power Systems, 2011 IEEE International Conference on Technologies for Homeland Security, November, 2011. (Featured Conference Paper)
- Rick A. Jones and Barry M. Horowitz, A System-Aware Cyber Security Architecture, Systems Engineering, Volume 15, No. 2, 2012
- Barry M. Horowitz, Kate Pierce, Application of Dynamic System Models and State Estimation Technology to the Cyber Security of Physical Systems, Cybersecurity in Cyber-Physical Systems Workshop, NIST, April, 2012

System-Aware Cyber Security Architecture

- System-Aware Cyber Security Architectures combine design techniques from 3 communities
  - Cyber Security
  - Fault-Tolerant Systems
  - Automatic Control Systems
- The point defense solution designers need to come from the communities related to system design, providing a new orientation to complement the established approaches of the information assurance community
- New point defense solutions will have independent failure modes from traditional solutions, thereby minimizing probabilities of successful attack via greater defense in depth

A Set of Techniques Utilized in System-Aware Security

Cyber Security

*Data Provenance
*Moving Target
(Virtual Control for Hopping)
*Forensics

Fault-Tolerance

*Diverse Redundancy
(DoS, Automated Restoral)
*Redundant Component Voting
(Data Integrity, Restoral)

Automatic Control

*Physical Control for
Configuration Hopping
(Moving Target, Restoral)
*State Estimation
(Data Integrity)
*System Identification
(Tactical Forensics, Restoral)

A Set of Techniques Utilized in System-Aware Security

Cyber Security

*Data Provenance
*Moving Target
(Virtual Control for Hopping)
*Forensics

Fault-Tolerance

*Diverse Redundancy
(DoS, Automated Restoral)
*Redundant Component Voting
(Data Integrity, Restoral)

Automatic Control

*Physical Control for
Configuration Hopping
(Moving Target, Restoral)
*State Estimation
(Data Integrity)
*System Identification
(Tactical Forensics, Restoral)

- This combination of solutions requires adversaries to:
  - Understand the details of how the targeted systems actually work

A Set of Techniques Utilized in System-Aware Security

Cyber Security

*Data Provenance
*Moving Target
(Virtual Control for Hopping)
*Forensics

Fault-Tolerance

*Diverse Redundancy
(DoS, Automated Restoral)
*Redundant Component Voting
(Data Integrity, Restoral)

Automatic Control

*Physical Control for
Configuration Hopping
(Moving Target, Restoral)
*State Estimation
(Data Integrity)
*System Identification
(Tactical Forensics, Restoral)

- This combination of solutions requires adversaries to:
  - Understand the details of how the targeted systems actually work
  - Develop synchronized, distributed exploits consistent with how the attacked system actually works

A Set of Techniques Utilized in System-Aware Security

Cyber Security

*Data Provenance
*Moving Target
(Virtual Control for Hopping)
*Forensics

Fault-Tolerance

*Diverse Redundancy
(DoS, Automated Restoral)
*Redundant Component Voting
(Data Integrity, Restoral)

Automatic Control

*Physical Control for
Configuration Hopping
(Moving Target, Restoral)
*State Estimation
(Data Integrity)
*System Identification
(Tactical Forensics, Restoral)

- This combination of solutions requires adversaries to:
  - Understand the details of how the targeted systems actually work
  - Develop synchronized, distributed exploits consistent with how the attacked system actually works
  - Corrupt multiple supply chains

Example Design Patterns Under Development

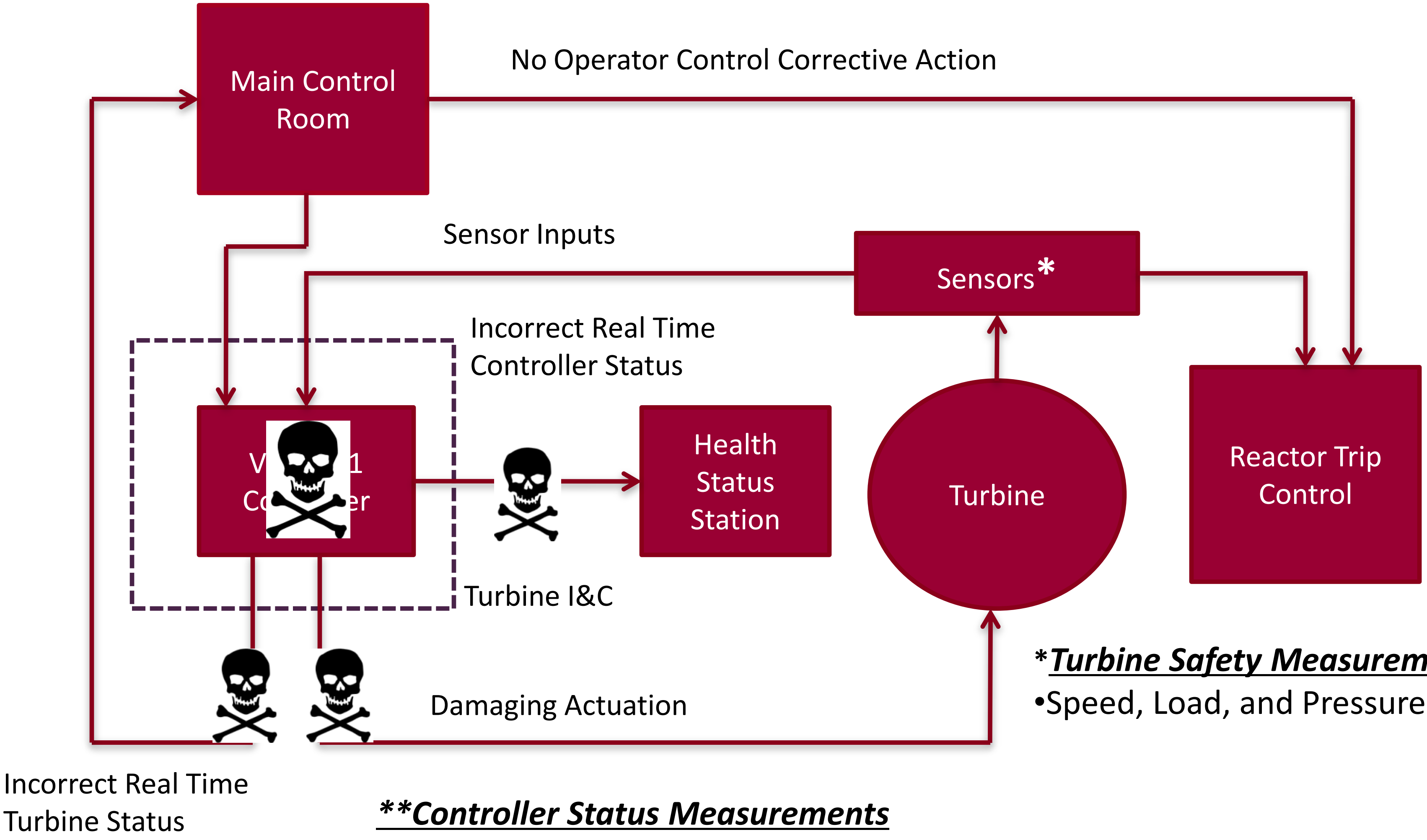
- **Diverse Redundancy** for post-attack restoration
- **Diverse Redundancy + Verifiable Voting** for trans-attack defense
- **Physical Configuration Hopping** for moving target defense
- **Virtual Configuration Hopping** for moving target defense
- **Physical Confirmations of Digital Data**
- **Data Consistency Checking**

Dynamic System Models and State Estimation Technology for Cyber Security of Physical Systems

The Problem Being Addressed

- Highly automated physical system
- Operator monitoring function, including criteria for human over-ride of the automation
- Critical system states for both operator observation and feedback control – consider as *least trusted from cyber security viewpoint*
- Other measured system states – consider as *more trusted from cyber security viewpoint*
- CYBER ATTACK: Create a problematic outcome by disrupting human display data and/or critical feedback control data.

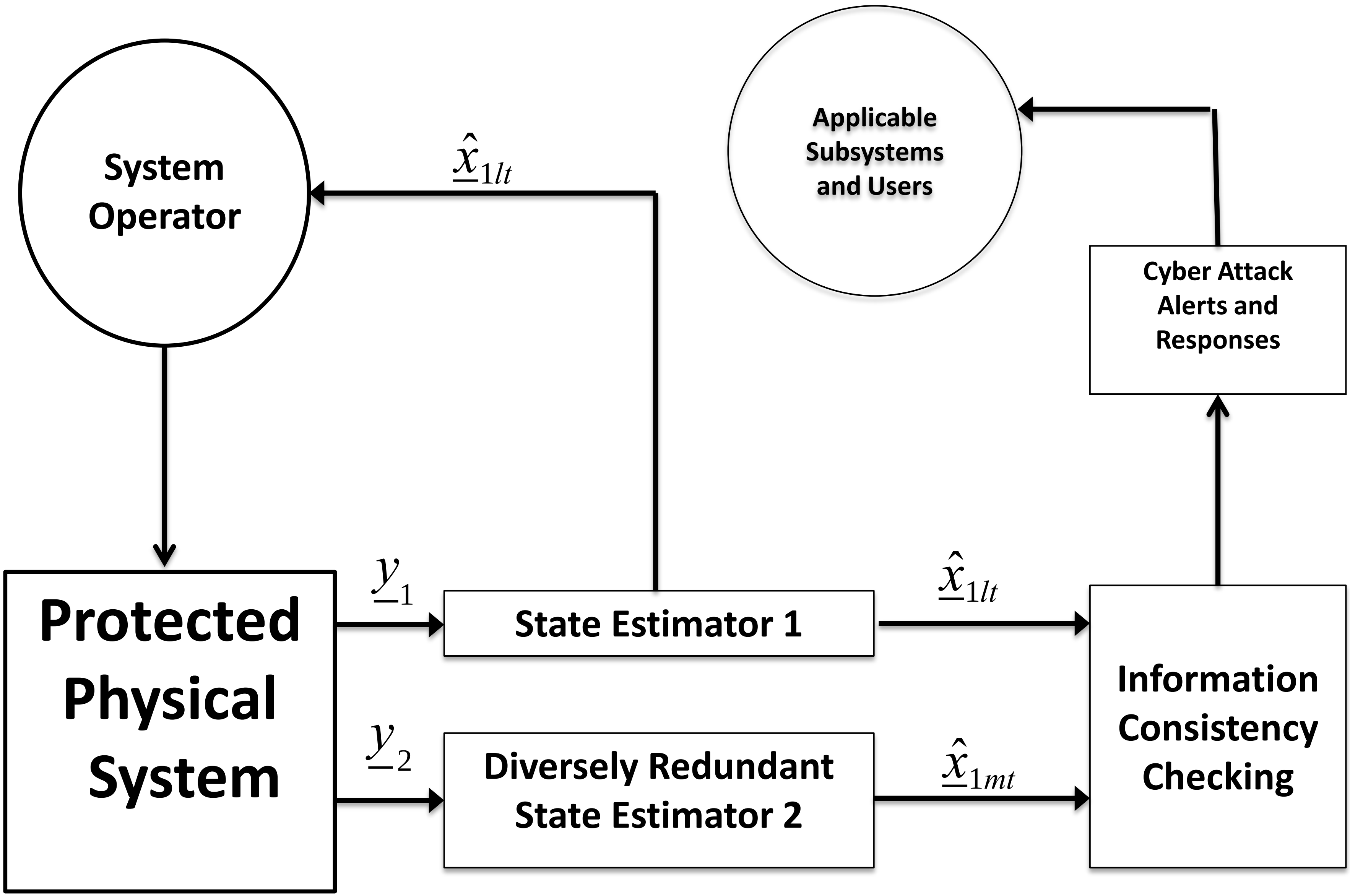
Cyber Attack: Damaging Turbine and Hiding its Effects



Turbine Safety Measurements
•Speed, Load, and Pressure

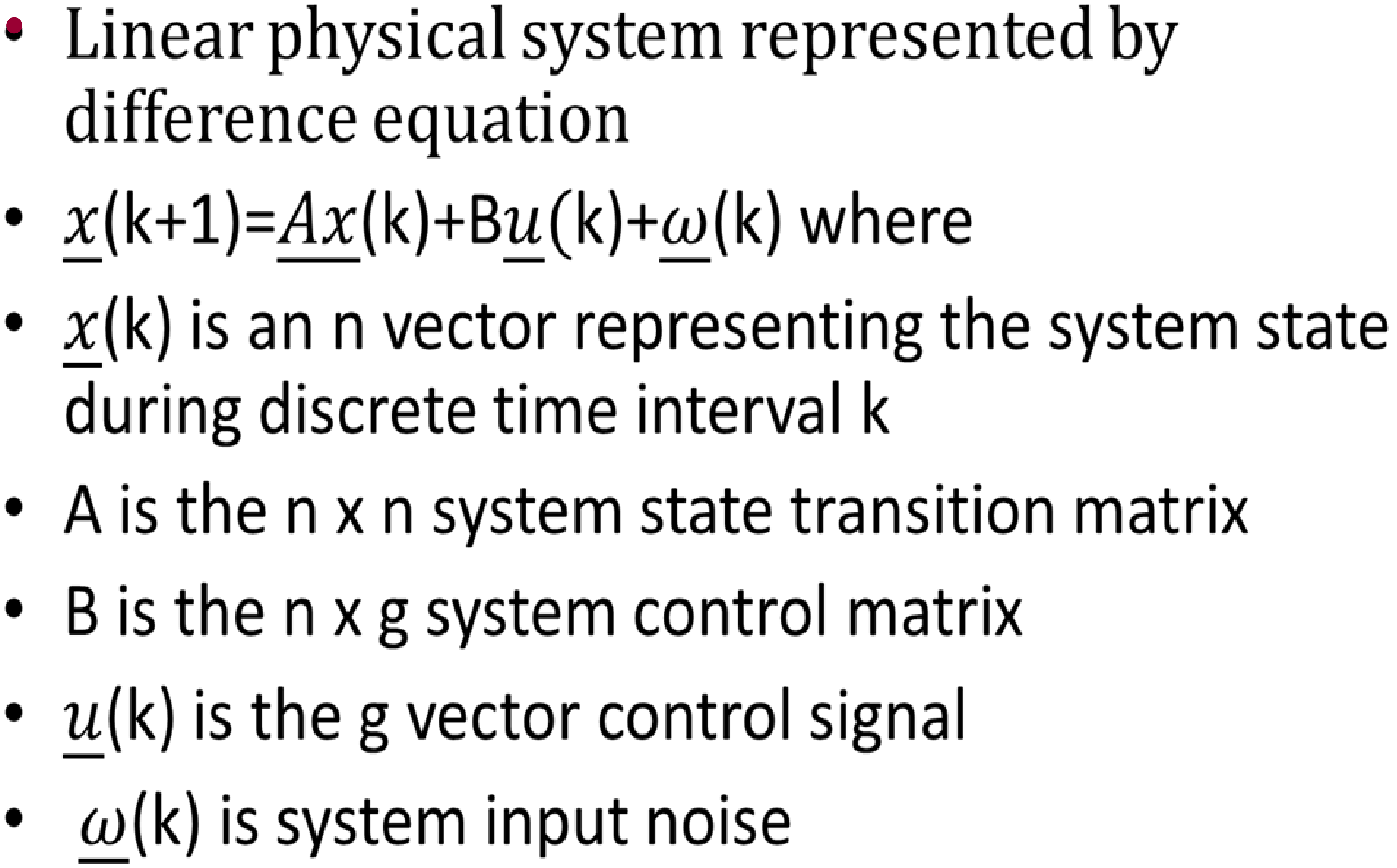
*****Controller Status Measurements***
•Hardware and System Health Status
•Software Execution Features
•I/O Status

Simplified Block Diagram for Inference-Based Data Integrity Detection System


System Operator
$\underline{\hat{x}}_{1lt}$
Applicable Subsystems and Users
Cyber Attack Alerts and Responses
Protected Physical System
$\underline{y}_1$
State Estimator 1
$\underline{\hat{x}}_{1lt}$
$\underline{y}_2$
Diversely Redundant State Estimator 2
$\underline{\hat{x}}_{1mt}$
Information Consistency Checking

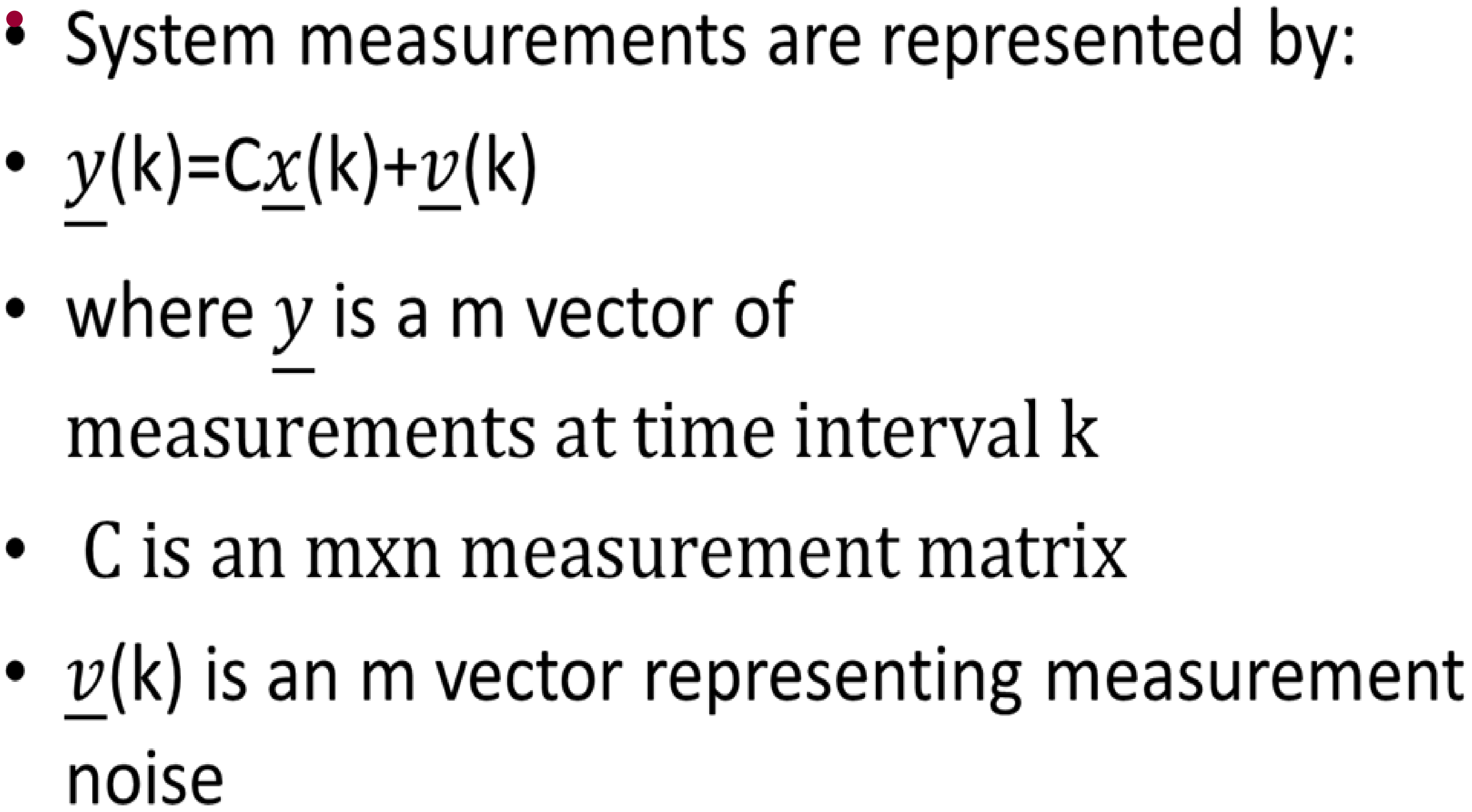
Theoretical Example

Regulating a Linear Physical System (1)



- Linear physical system represented by difference equation
- $\underline{x}(k+1)=\underline{Ax}(k)+B\underline{u}(k)+\underline{\omega}(k)$ where
- $\underline{x}(k)$ is an n vector representing the system state during discrete time interval k
- A is the n x n system state transition matrix
- B is the n x g system control matrix
- $\underline{u}(k)$ is the g vector control signal
- $\underline{\omega}(k)$ is system input noise

Regulating a Linear Physical System(2)



- System measurements are represented by:
- $\underline{y}(k)=C\underline{x}(k)+\underline{v}(k)$
- where $\underline{y}$ is a m vector of measurements at time interval k
- C is an mxn measurement matrix
- $\underline{v}$(k) is an m vector representing measurement noise

Regulating the States of the System

- Linear system controller to sustain the states of a system at designated levels

- Optimal Regulator Solution (LQG)
  - White Gaussian noise
  - Separation Theorem
  - Kalman Filter for state estimation
  - Ricatti Equation-based controller for feedback control

- Controller feed back law based upon variances of input noise, measurement noise and the A,B and C matrices of the system dynamics model

Example State Equations and Noise Assumptions

A = [1,   1. -.02, -.01

.01, 1, -.01,   0

.2, .01,   1,   1

-.01, .02, -.01, 1];

B = [0 , 1 , 0 , 0];

Operator Observed (less trusted):

C = [1, 0, 0, 0];

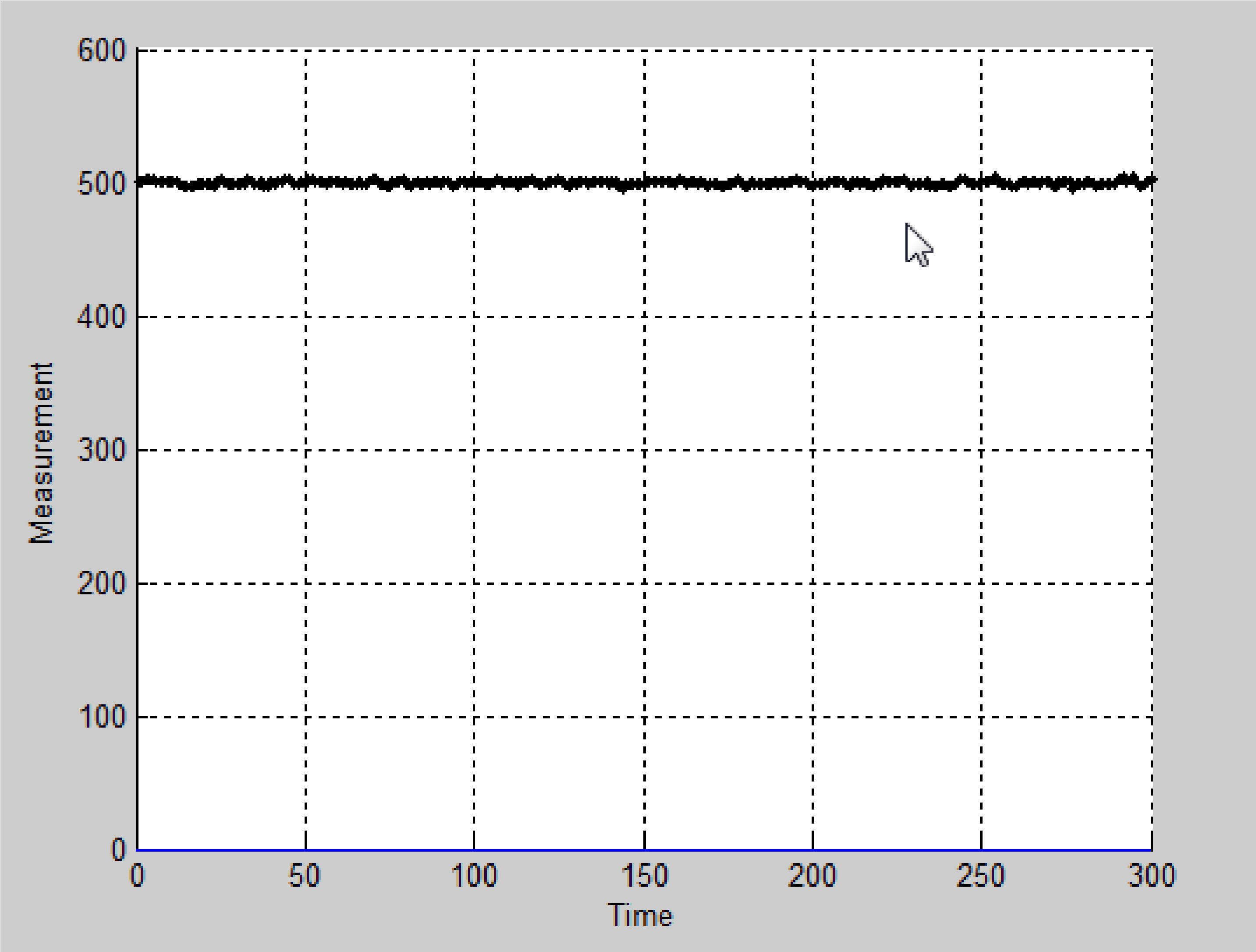
Related States (unobserved by operator, more trusted):

C2 = [0 1 0 0; 0 0 1 0; 0 0 0 1]

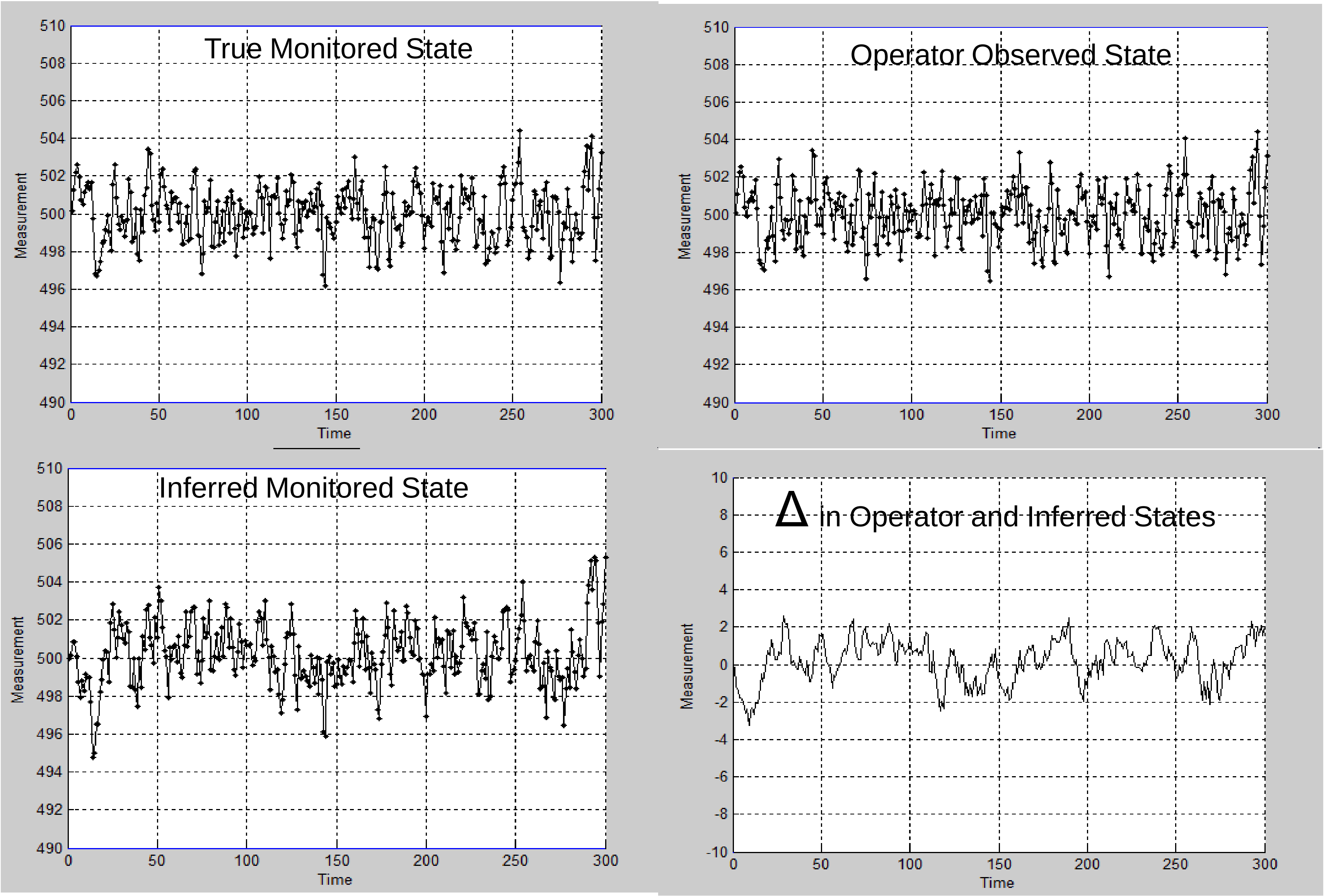
K1 = 0.25;   process noise variances for each of the states

K2 = 0.25;   sensor noise variances for each of the measurements

Simulated System Operation for Regulation of a State Component at 500

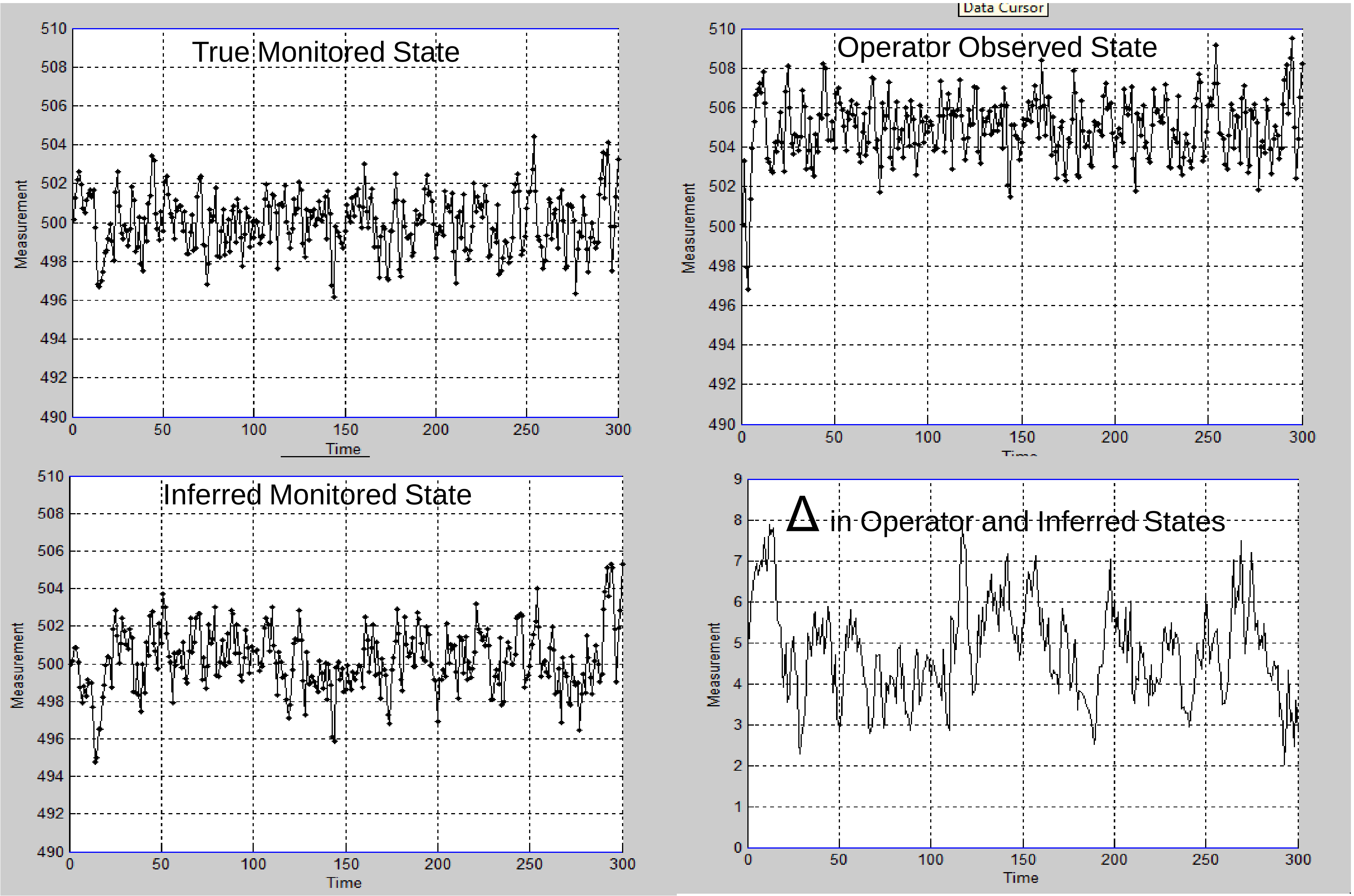


Simulated Normal Operation



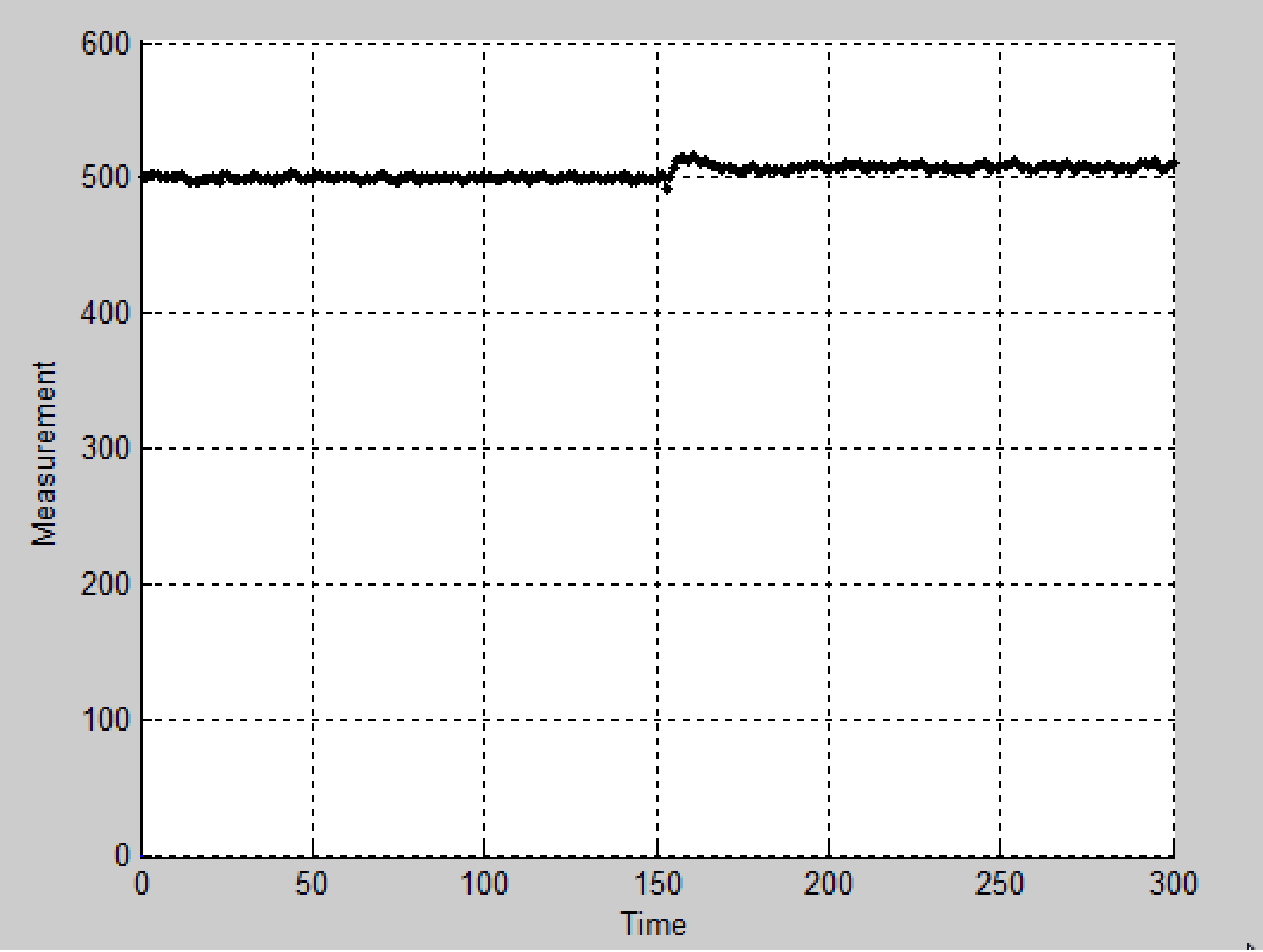
Replay Attack to Cause Erroneous Operator Action

Simulated Replay Attack

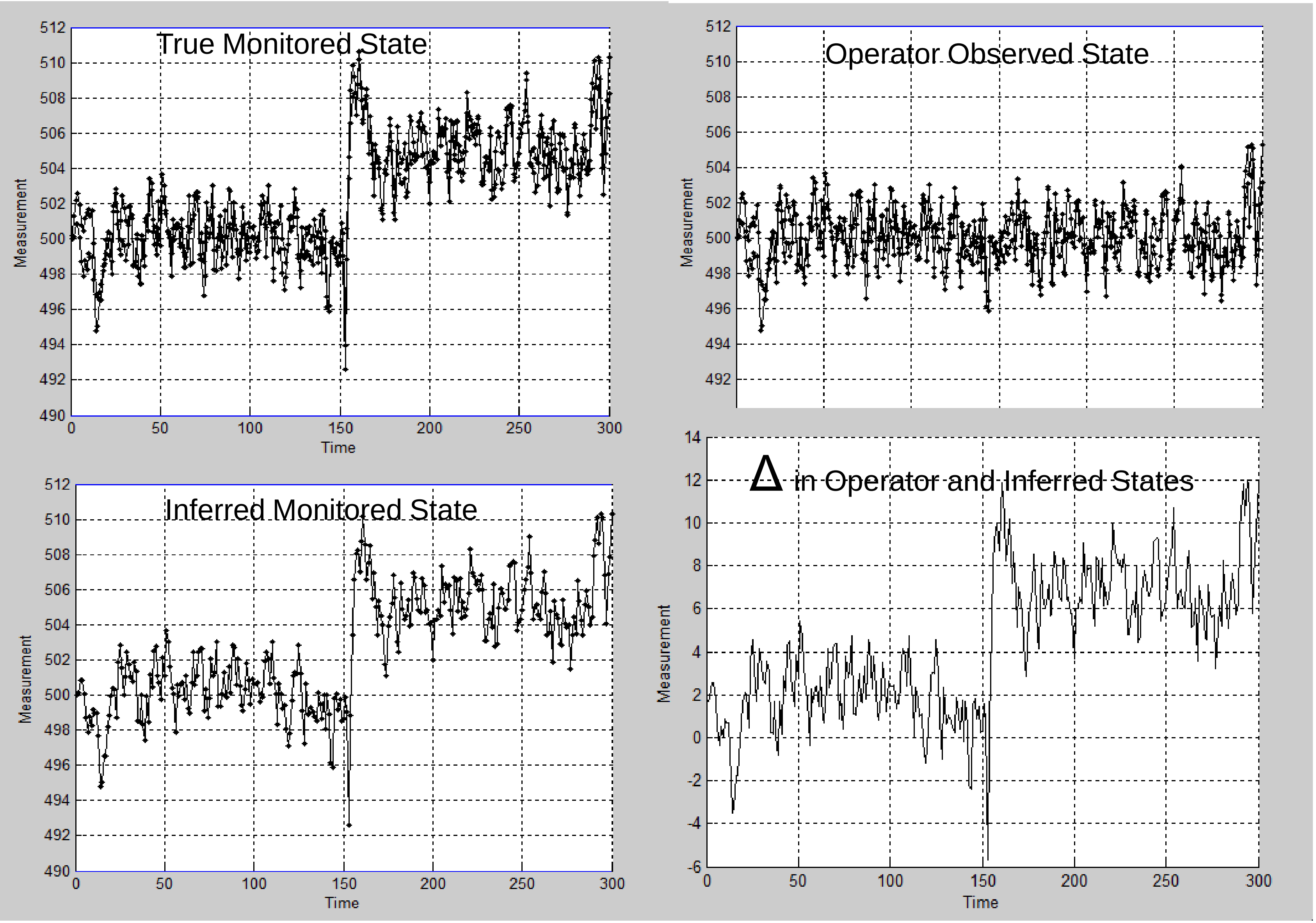


Attack to Adjust Regulator Objectives and Mask the Physical Change Through Replay Attack on Operator Displays

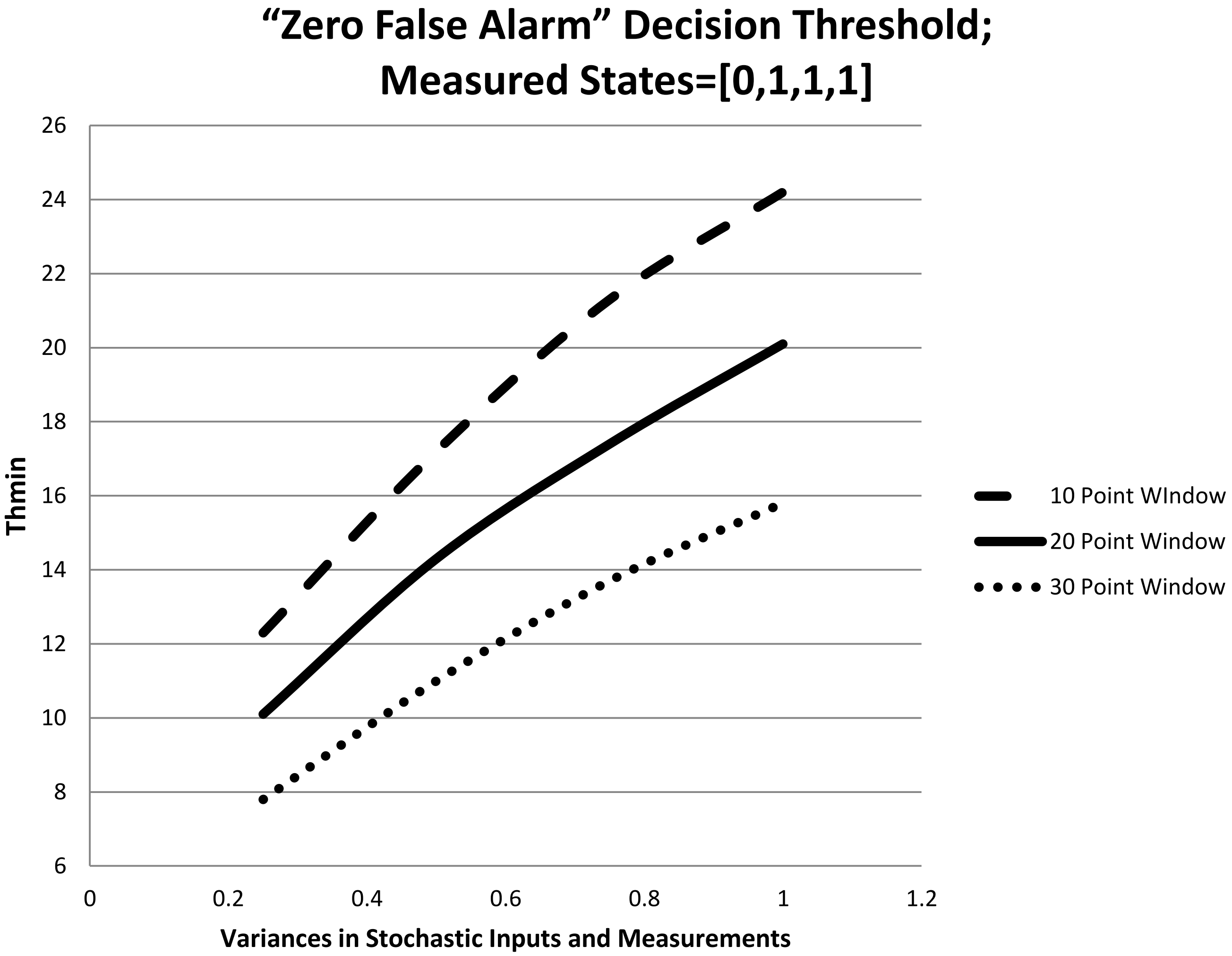
Simulated System Output Based Upon Controller Attack



Simulated Regulator Attack



Performance & Sensitivity Analysis Example



Noise clipped at 3 std deviations,
Sliding window detector: ¾ of points exceed **Thmin**

Future System Aware Research Efforts

- Formulating a UAV pilot program for design patterns (UVA)
- Formulating a corresponding pilot program for Scoring Methodology (UVA)
- Exploration of software design and assurance methodologies for point defense solutions (USC, Auburn)
- Development of broader system metrics for system requirements development (Stevens)