A Method for the Detection and Diagnosis of Stealthy False Data Injection Attacks in Cyber-Physical Systems

A Thesis Document

Presented to

The Faculty of the School of Engineering and Applied Science

University of Virginia

In partial fulfillment of the requirements for the degree:

Master of Science in Systems and Information Engineering

By

J. Vince Pulido

May 2014

APPROVAL DOCUMENT

The thesis is submitted in partial fulfillment of the requirements for the degree of Master of Science in Systems Engineering

_____

J. Vince Pulido

The thesis has been read and approved by the examining Committee:

_____

Ronald Williams – Thesis Co-Advisor

_____

Barry Horowitz – Thesis Co-Advisor

_____

Amy LaViers – Committee Chair

_____

Carl Elks – Committee Member

_____

Stephen Patek – Committee Member

Accepted for the School of Engineering and Applied Science:

_____

James H. Aylor – Dean, School of Engineering and Applied Science

ACKNOWLEDGEMENTS

Foremost, I would like to express my sincere gratitude to my advisor Professor Ron Williams and Barry Horowitz for the continuous support of my study and research, for their patience, motivation, enthusiasm, and immense knowledge. Their indispensable guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisors and mentors for my Masters study.

My sincere thanks also goes to Dr. Carl Elks who essentially taught me how to write. I also thank him for his insight and encouragement.

Last, I would like to thank my parents, Joseph F. Pulido and Evelyn Pulido, for their continual support and love. Thank you!

Abstract

Cyber-Physical Systems (CPS) combine computational, communication, sensory and control capabilities to monitor and regulate physical domain processes. CPSs are becoming increasingly networked with the cyber world, opening access to communication with control rooms, command and control stations, other computer based systems, or even the Internet. Examples of cyber-physical systems include transportation networks, Unmanned Aerial Vehicle Systems (UAV's), nuclear power generation, electric power distribution networks, water and gas distribution networks, and advanced communication systems. In all cases, current technology has introduced the capability of integrating information from numerous instrumentation and control systems and transmitting needed information to operations personnel in a timely manner.

While the application of perimeter security technologies has been utilized to help manage the possibility of cyber attackers exploiting highly automated cyber physical systems, the rate of successful attacks against critical infrastructures continues to be problematic and increasing [1]. Furthermore, the trend in adversarial attacks is moving toward well-formed coordinated multi-vector attacks that compromise the system in such a way that detection and identification is challenging for perimeter security solutions and human monitoring.

This research effort constructed a methodology to defend against stealthy, low probability of detection, and high impact cyber-attacks on CPS. The goal is to increase the level of difficulty to perform a stealthy attack by improving the probability of detection, isolation and limiting the impact of an attack. The study uses the example of a UAV navigation system comprising of a redundant set of INS and GPS units solving the problem posed by Kwon et al [2] that there exist false injection attacks that evade fault detection techniques, allowing the adversary to deviate an aircraft.

The examined architecture is comprised of a diverse sensory architecture within the CPS, avoiding supply chain vulnerabilities, and provides several possible trustworthy references. Expanding from a system with multiple components, a similarity measurement between INSs and GPSs is developed leveraging their unique characteristics and relationship. Assuming that an adversary is restricted to attacking a singular navigation component, the method is able to detect and isolate persistent cyber-attack for a large enough deviation.

An analytic attack model of a UAV navigation system comprising of multiple INS/GPS is validated with a complementary simulation, using a combination of a logical decision tree and similarity measurement analyses, the method correctly detects an infected component with a low false alarm rate(0.01) The latency of the attack decreases as the rate of deviation increases. The maximum deviation an adversary can deviate an INS without being detected is about 30m of a 30min flight, on INSs with 0.05 and 0.07 m/s^2 acceleration measurement error. The maximum deviation an adversary can deviate an GPS without being detected is about 16m of a 30min flight, on GPSs with 3 and 4 m/s^2 position measurement error.

TABLE OF CONTENTS

LIST OF FIGURES

SECTION 1: INTRODUCTION

1.1 CYBER PHYSICAL SYSTEMS

The term Cyber-Physical Systems (CPS) refers to systems with integrated computational and physical capabilities that can interact with humans through various modalities. We call these interactions with other systems (and humans) the environment or the context of the given cyber physical system. CPS is commonly defined as a system with the following three capabilities: (i) sensing physical world (e.g., the position of a valve controlling a tank filling process), (ii) making decisions (e.g., whether it is necessary to open or close the valve), and (iii) performing actions in physical world (e.g. open or close valve to maintain tank fluid level). CPS's broadly focuses on the control and monitoring of physical processes.

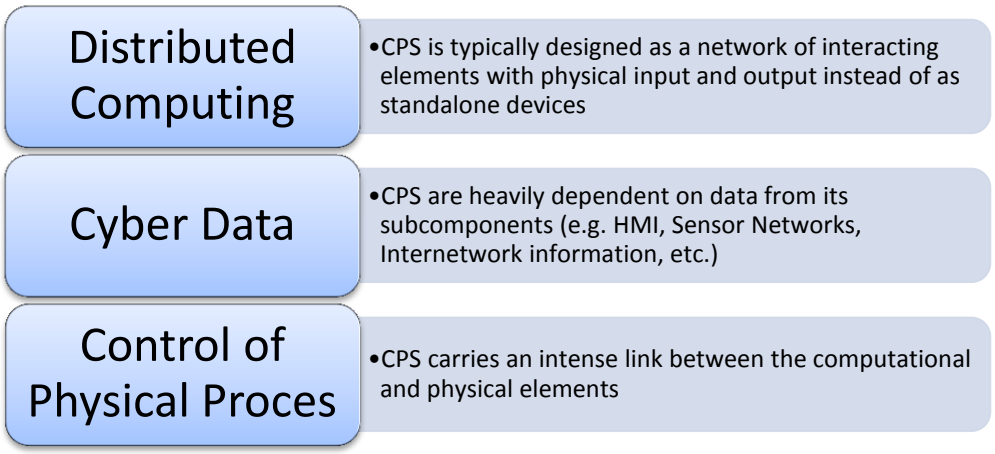

Figure 1: Lists the major characteristics and capabilities of modern CPS. CPS involves a network computation of heterogeneous data to control physical processes.

CPSs are prevalent in the civilian (i.e. power grid, public utility services, financial infrastructure, etc.) and defense space (i.e. search and rescue missions and command, control, and conquer (C3) systems). One significant rising technology within the CPS envelop is the Unmanned Vehicle. There is a recent surge in the application and

development of these autonomous vehicles, revolutionizing the commercial and military spheres, improving the safety of the individuals and decreasing the cost of human labor necessary to operate these systems.

Commercial examples of autonomous vehicles include:

1. Amazon.com, Inc.: developing a network of Autonomous Aerial Vehicles to deliver small packages to their customers' doorstep; a developing program named Amazon Prime Air.
2. Google, Inc.: developing the next generation of Driverless Cars in a program called Google Chauffer.

Even with the increasing potential of Autonomous Vehicles in the private space, the armed forces continues to dominate the deployment and application of Autonomous Vehicles with Unmanned Aerial Vehicles (UAV) taking significant roles in aerial reconnaissance and offensive missions.

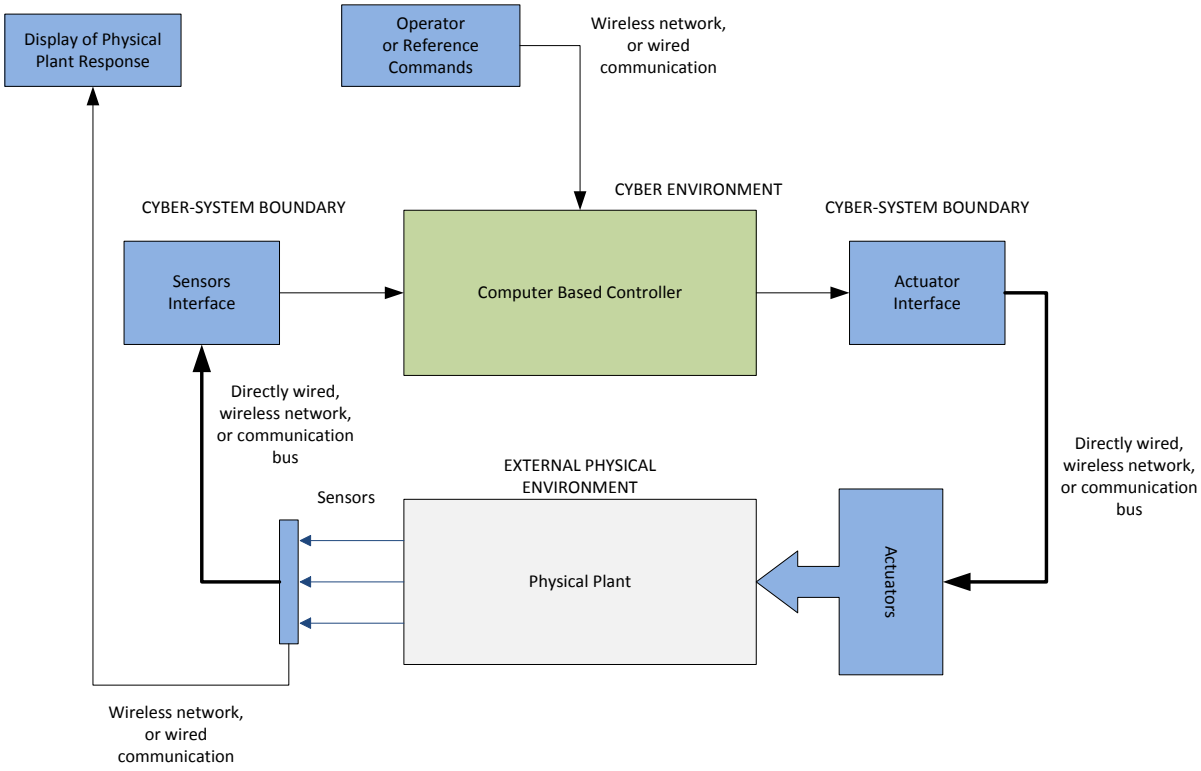

Figure 2: Block diagram of a generalized CPS.

Figure 2 depicts the border between the CPS and its environment is the called the system boundary or cyber system boundary where sensors continuously provide sensor data to the controller reflecting the physical state of the plant, and actuators receive control commands from the computing machine to effect and control the physical plant. The External Physical world consists of all the physical devices external to the computing machine. This includes the plant being controlled, as well as the environment in which the plant operates. The environment also includes disturbances to the plant from the natural world, such as wind, radiated energy, vibrations, etc. The computer based controller consists of all of the hardware, control logic and algorithms used generate control actions applied to the plant. The computer actions are represented by discrete time state space, reflecting the digital nature of the computer. The service delivered by a cyber-physical system is its behavior as it is perceived by its user(s) receiving services from the CPS. The

delivered service can be viewed from two perspectives: 1) a sequence of commands, status, and requested information between the physical plant and the controller, or 2) the commodity flow of the plant (e.g. electric power distribution, oil in a pipeline, etc.). The first service type (type 1) is at the boundary of the cyber-physical system, the second (type 2) is in the physical world altogether. Examples of type 2 services may be flying an aircraft to a destination, supplying power over an electric grid, or regulating the speed of a motor. Note that a controller may sequentially or simultaneously be giving and receiving with respect to another system, i.e., deliver service to and receive service from that other system.

Attackers are increasingly aware of the importance and vulnerability of critical infrastructures and are targeting physical systems to damage them and harm civilians [3]. We are encountering a new wave of cyber-attacks, a new type of infiltration; well-honed, massively coordinated, sophisticated attack that encompasses not only stored information in computers and information traveling through networks, but also controllers, sensor networks and SCADA systems. It is widely speculated that hacking's latest surge may include terrorist cyber strikes against these critical systems [1].

The prevalence of these critical Cyber Physical Systems in our society coupled with a lack of robust security defenses has led to increased concern that cyber-attacks on CPS's that may result in catastrophic events which could cause significant disruption in energy, financial, and infrastructure services.

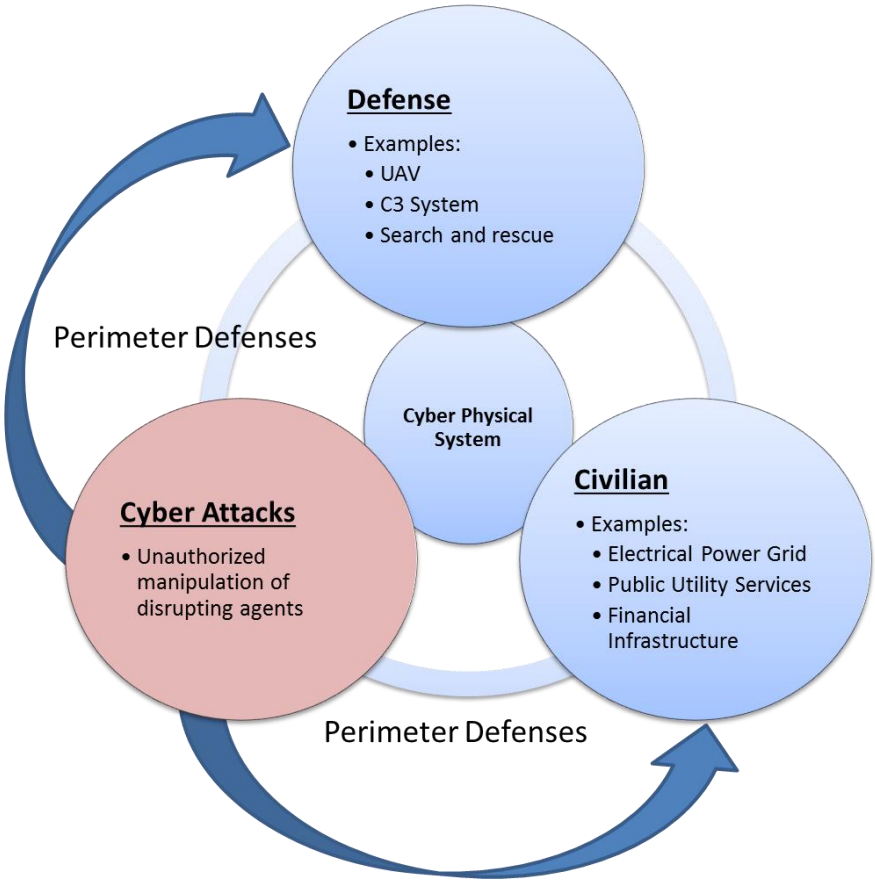

Figure 3: The ubiquity of CPS spans civilian and defense application. Cyber Attacks are increasingly involved in the issues of CPS. Disrupting agents are increasingly capable of unauthorized manipulating these CPS upon which civilians and military personnel heavily rely. Because of this reliance, malicious cyber adversaries can damage and cause severe harm to the broad range of stakeholders of these systems.

Cyber based threats or cyber faults to cyber physical systems manifest in two domains; the information universe, and the boundary between the information universe and the physical universe – the cyber system boundary. More precisely, information in a computer is characterized by symbols, and the interpretation and manipulation of those symbols. Symbols are represented by instructions and the data that instructions operate on. Cyber faults can corrupt symbols rendering them into different symbols, non-symbols or reconstitute the interpretation of symbols. Cyber faults in the information universe are usually manifested as modifications to data and instruction symbols. Finally, cyber failures

are associated with the cyber system boundary, which is where the user of the system eventually sees the effects of faults and errors. With cyber failures, a distinction can be made between cyber primary failures, cyber secondary failures and cyber performance failures.

- Cyber Primary failures: A primary failure is caused by a cyber-fault (flaw) in the software or hardware so that the system output does not meet its intended action or specification. A system output could be a command to the physical plant or a display of the status of the plant at the HMI.
- Cyber Secondary failures: A secondary cyber fault occurs when the input to a computer does not comply with the specification. This can happen when the computer and its software are used in a way it is not designed for, or inputs applied to the system are not anticipated or of the wrong type. Inputs can be from sensors, other systems, or from operators. Cyber Secondary failures may induce primary failures.
- Cyber Performance failures: Cyber Performance failures occur at the cyber-system boundary. Performance failures can occur in two ways. One way is Timeliness. This is a measure of time from input of data to output of data. When there is a sustained increase in this measure, a timeliness performance effect has occurred. Timeliness also includes so called real-time failures. This occurs when the computer delivers the correct output but at a time that is beyond the real-time deadline of the system. This is also called a real-time violation or failure. Another performance failure can occur when the accuracy or precision of the output is affected. When the output is not 100% of the expected output, a precision performance effect has occurred.

For example, when a process crashes before completing execution, output could be less than 100% of the expected.

Because cyber-based threats and intrusions are an artifact of design (a human crafted the exploit), all combinations of the above types of cyber-failures can occur, and in any order.

Concerns about the security of CPS's are not new, as the numerous published material on systems fault detection, isolation, and recovery testify [3, 4, 5]. At present, the state of the practice for augmenting Cyber Physical Systems security is perimeter based security (such as firewalls, intrusion detection mechanisms, anti-viral signature software, encryption, and advanced user authentication). While the application of perimeter security technologies has been utilized to help manage the possibility of cyber attackers exploiting highly automated cyber physical systems, the rate of successful attacks against critical infrastructures continues to be increasingly problematic [6]. Furthermore, the trend in adversarial attacks is moving toward well-formed coordinated multi-vector attacks that compromise the system in such a way that detection and identification is challenging for perimeter security solutions and human monitoring.

From a cyber-war prospective, cyber-attacks provides a non-kinetic means to deny, degrade, disrupt or even destroy an adversary's ability to fight and function. Bytes instead of bombs can potentially render an adversary's command and control, critical infrastructure or logistics useless. In the present states of affairs, adversaries have shown they can plan their attacks carefully over time and launch the attacks at times of their choosing all with modest technical resources and skills.

Furthermore, an asymmetrical conflict arises where defending is expensive while attacking is cheap: an attacker choses the attack vector which gives him the highest

probability of success i.e. attackers can chose to target the weakest component of a perimeter defense. Moreover, adversaries may take as much time as they seem necessary to deceive, obfuscate, and deploy attacks. Any sophisticated security protocol that a defender implements can be breached given ample time, resources, and knowledge of the system [7].

On the other hand, a defender has a minute amount of time to react to an attack originating from all sides of the network perimeter. Defenders are forced to invest tremendous expenses to secure physical systems from cyber-attacks on the defender's side. Recent events [7, 8] maintain that current perimeter defenses are *necessary* for defending from cyber-attacks to prevent the bulk of attacks from entering a physical system, but are not *sufficient*. The Stuxnet virus provides evidence to this assertion [9]. Once the malicious virus went beyond the facility's firewalls and security, it was virtually undetectable until an operator noticed irreparable damages to the target physical system.

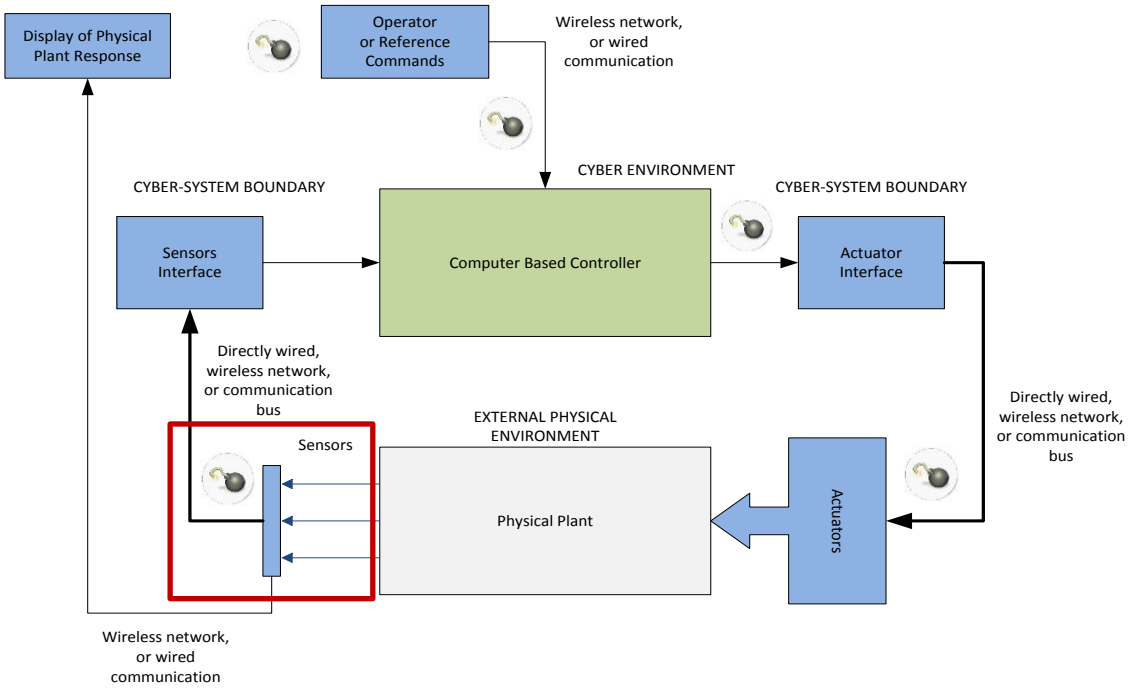

Figure 4: Block diagram depicting the different classes of vulnerabilities. This body of work focuses on attacks against the sensor components of the Cyber Physical System.

Figure 4 depicts the various classes of vulnerabilities and exploitation that an adversary may target [3]. We focus the scope of this study to the defense and detection of cyber-attacks targeting sensor components of CPS.

1.2 Autonomous Aerial Vehicles and Sensor Vulnerabilities

In the beginning of manned flight, aircrafts were predominantly controlled by trained pilots manually keeping the stability of the aircraft and driving the actuation of the aircraft to complete a specific mission. UAVs heralded a new generation of aerospace hardware replacing conventional manual flight controls with electronic interfaces. Movements of flight controls are converted to electronic signals transmitted to a controller to determine the movement of the actuators at each control surface to provide the ordered response allowing the actuators to perform functions without a pilot's input to stabilize or direct the aircraft. The UAV's require minimal input from a remote pilot to carry out airspace missions that are too mundane or dangerous for manned aircrafts.

Without the eyes, ears, and instincts of the pilot, a UAV must profoundly depend on their on-board sensors to maintain its stability and mission integrity during flight. One of a UAV's dominant sensors is its navigation units. Before GPSs became popular, UAVs carried a singular strap-on INS. These types of stand-alone navigation system, however, caused imprecise geo-positioning as errors compound in time [10].

With the low costs of GPS units, embedded GPS/INS units are becoming the standard navigation system for UAVs. Figure 5 is a block diagram depicting the navigation resolution between INS and GPS measurements. Unlike INS, which calculates position based on additive acceloremetric and gyroscopic readings; GPS components receive time signals from geo-stationary satellites to estimate the position. Sohne et al [10] presented several techniques for fusing INS and GPS data including *Update with precise GPS position* and

Cascade integration. Another way to correct INS and GPS data is to continually fuse INS and GPS data using a Kalman filter [11].

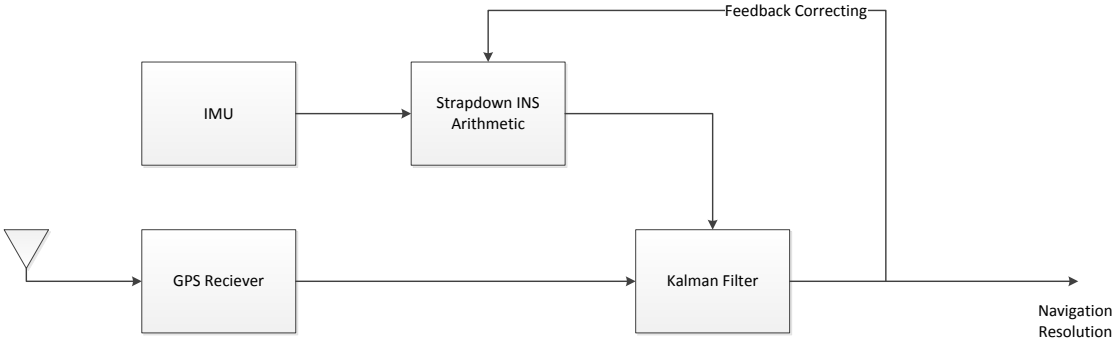

Figure 5: Navigation Resolution between a strap-down INS/GPS system.

The benefits of using GPS with an INS are that the INS may be calibrated by the GPS signals and that the INS can provide position and angle updates at a quicker rate than GPS. For high dynamic vehicles, such as missiles and aircraft, INS fills in the gaps between GPS positions. Additionally, GPS may lose its signal and the INS can continue to compute the position and angle during the period of lost GPS signal. The two systems are complementary and are often employed together.

Seeing the significance and prolific use of UAVs in today's airspace, adversaries are looking for ways to tamper with the components essential to the operation and integrity of the UAV. A major vulnerability is the adversarial manipulation of navigation sensors [12]. Since UAVs are heavily reliant on their on-board navigation equipment (Global Positioning System (GPS), Inertial Navigation System (INS), Gyroscopes, etc.) as the principal input to regulate the actuators and govern the aircrafts' flight dynamics, cyber adversaries may take control of the aircraft, possibly compromising airspace missions, by injecting false information to these navigation components [4, 2] making these types of attack vectors attractive. By manipulating one component of the UAV the adversary has total control of the flight path of the aircraft.

False injection attacks against CPS can originate from two methods:

1. *Spoofing attacks* which are disruptive attacks originating from outside the development, assembly, and operation of the CPS.
2. *Supply chain attacks* which are attacks that infect systems' components via compromising organizations, people, activities, information, and resources involved in moving the component from supplier to customer.

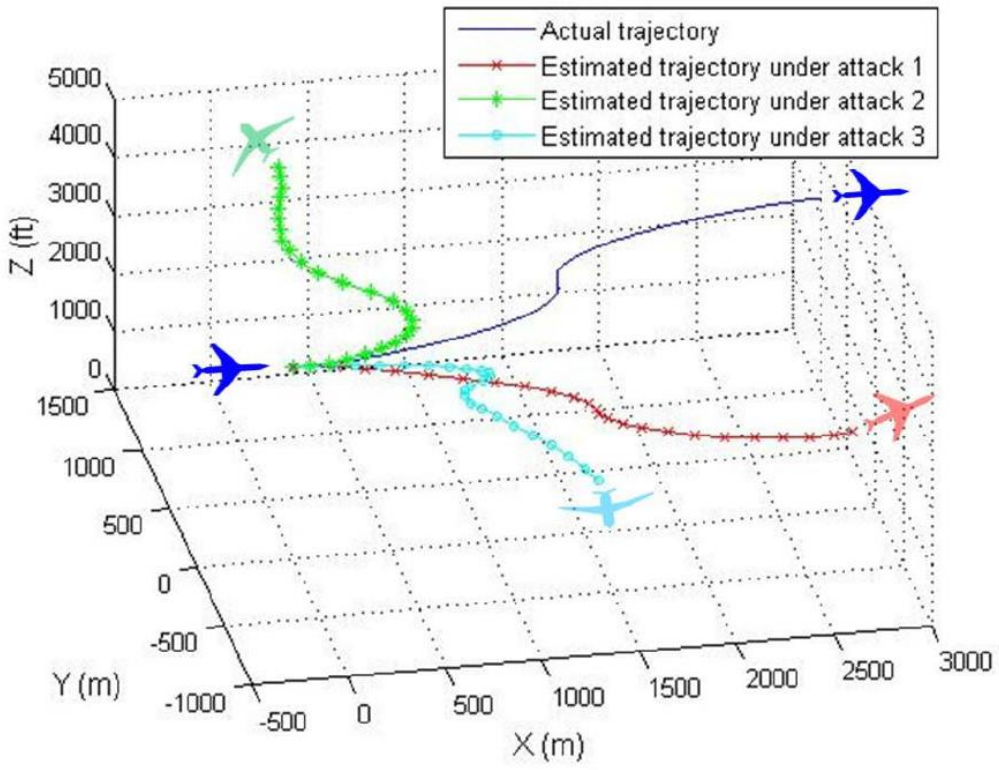

Figure 6: Graphic on the navigation impact of stealthy false injection attack against UAV navigation systems [2].

Figure 6 illustrates the impact of a stealthy false injection attack targeting the navigation components of a UAV. The severity of cyber-attacks against key military apparatus threatens the capabilities of these rising technologies, weakens its applicability and diminishes the value they offer to the military and civilian space. The cyber defense of these UAV sensors is imperative, then, in the promotion and application of these technologies.

1.3 Problem Statement

The security of sensors against adversarial false injection attacks is critical to keep the UAV under the control of its proper managers. As demonstrated by the December 2011 confiscation of an RQ-170 Sentinel by Iranian military forces, cyber adversaries are increasingly capable of manipulating UAVs via sensor manipulation through false injection attacks. Consider a case where cyber-attacks targeting navigation components originate from supply chain attacks and/or GPS sensor carried out by way of stealthy means. Adversaries could apply gradual, and persistent false injections, allowing them to furtively waver a UAV away from its designated flight path, thus compromising its mission by keeping geographic locations inaccessible, and ultimately, concealing intelligence from the operators who seek it [13]. The impact of such a cyber-attack, where an aircraft is deviated away from the intended flight path, is shown in Figure 7.

Furthermore, Kwon et al [2] argues that there exists sequences of false injections that would go unnoticed using fault detection techniques, meaning that an adversary could manipulate a UAV away from its intended flight path without alerting the operators. This is a vulnerability that adversaries may pursue. The adversary could have total control of the aircraft just by manipulating the navigation sensor of the UAV, making this type of attack an attractive option. The work herein presents a solution to increase the difficulty to successfully manipulate sensory components of UAVs through the use of System-Aware design patterns of diverse redundancy and data consistency checking.

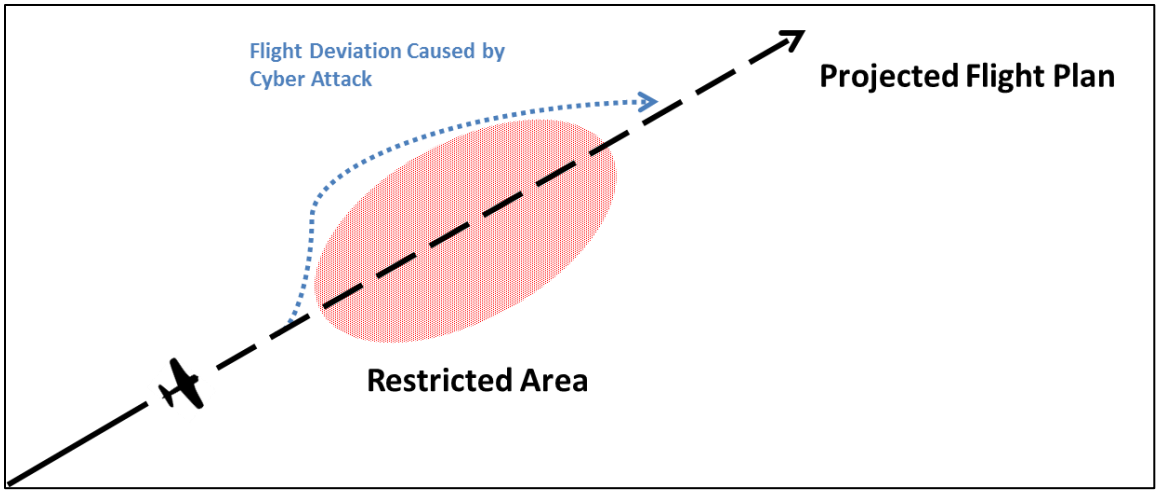

Figure 7: Depicts a scenario where the adversary use false injection attacks to deviate a UAV to detract its way from its intended flight path, avoiding the area the adversary is concealing.

The study aims to decrease the desirability for an adversary to choose false injection attacks against sensor components by accomplishing 3 key objectives:

1. Objective 1: Detect stealthy false-injection attacks against sensor components
2. Objective 2: Isolate infected sensor components
3. Objective 3: Limit impact of such cyber-attacks

These methods can be generalized to other cyber physical systems with concerns over false injection attacks against their sensor components.

SECTION 2: REVIEW OF RELEVANT LITERATURE

Section 2 provides the background and expected contribution of this body of work. Section 2.1 explains the background behind false injections and the methods developed applied to Water SCADA Systems. This section also introduces the work of Kwon et al [2] stating that there exist false injection attacks that evades fault detection techniques. Section 2.2 introduces a new System-Aware paradigm whose design patterns manage a system's operations before, during, and after a cyber-attack.

2.1 FALSE INJECTION VULNERABILITIES

Secure control theory studies the impact, detection, and of cyber-attacks against the control components of a physical system. Secure control theory addresses two major attack models. The first model, *denial of service* (DoS) attacks, refers to types of attacks that obstruct the communication between network agents [14]. Adversaries carry out these types of attacks by jamming communication channels, attacking routing protocols, etc. Examples of defense against these types of attacks are addressed in [15, 16, 17].

The second type of attack is *deception attacks* which attackers relay false information injections into the automated system allowing adversaries to stealthily manipulate the system to attain various objectives. Attackers who employ these attacks usually have a great amount of knowledge of the unique system architecture and fault-detection methods, making the detection of these types of attacks difficult via perimeter defenses and fault-tolerant techniques.

Due to the recent popularity of the topic of cyber defense, there has been a surge in methods developed to detect stealthy deception attacks against cyber physical systems.

Previous work on this area is the study of false data injection attacks in control systems [18] and the intrusion detection models of [3] and [19].

Amin et al. [8] applied model-based detection systems on Water SCADA Systems applied to the supervisory control layer. Cardenas et al [3] introduces a model-based method viable for intrusion detection. Their method is applied to intrusion detection of Water Treatment SCADA Systems [8] where they develop an attack detection scheme using hydrodynamic models [20].

Kwon et al [2] examines the effects of false injection deception attacks against navigation components of a UAV. Known el al. has analytically proven that adversaries can furtively manipulate systems, undetected by fault-detection residual tests. Attackers can generate attack sequences that gradually deteriorate navigation data of an embedded INS/GPS using a Kalman filter to resolve the navigation position between the two components. Kwon et al [2] showed that false injection attacks can cause large errors in position and all the while maintaining their clandestine operations. The study tested three sequences of false injections on the INS, GPS and both INS/GPS that created various error rates that do not trigger any alarms using a compound scalar test. Figure 8 compares the values of position estimation error due to false injection attacks while Figure 9 shows the residual power of the nominal system and the system under three different attacks. This demonstrates that the residuals under attack have no significant change in their statistical properties which implies the attacks are undetectable via residual tests.

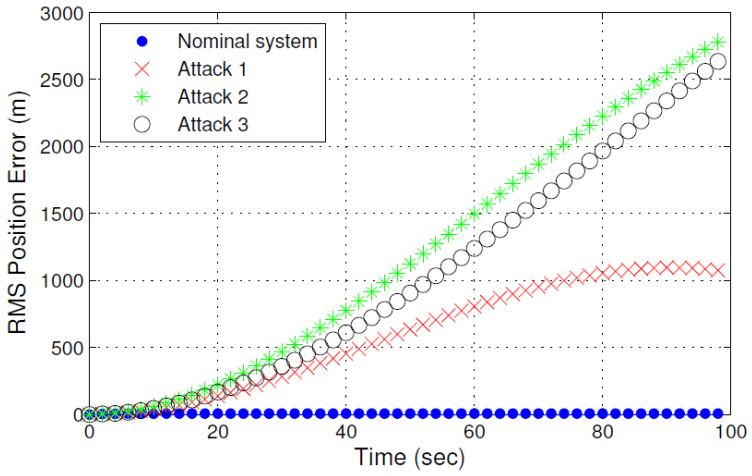

Figure 8: Position error between nominal case (no-attack) and three different attack scenarios [2]. The figure shows that there is an increase in position error for each attack.

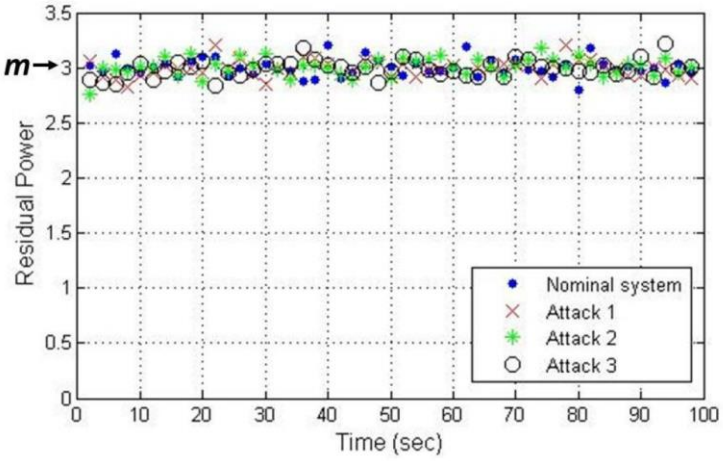

Figure 9: Compound scalar test that demonstrate that residuals caused by attacks do not have significant changes in the statistical properties [2]. Shows the undetectability of a false injection attack against sensor components based on statistical changes in the residual of the INS and GPS.

Kwon [2] also concluded that a deviation solely on the INS is bounded if the adversary is avoiding detection; while GPS deviation is unbounded. Thus, GPS deviation is more dangerous and impactful than that concerning a deviation of the INS.

2.2 System-Aware Cyber Security

Jones et al [7] introduces new a security paradigm to impede attackers from successfully infiltrating components of physical systems. Coined as *System-Aware Cyber Security Architecture*, it combines design techniques from three distinct communities: Cyber Security, Fault-Tolerant Systems, and Automatic Control Systems. The central purpose of System-Aware Security isn't merely focused on intrusion detection; instead, its purpose is to manage a system's "pre-attack, trans-attack and restorative methods" in hopes to increase the adversarial cost of attack. System-Aware design patterns include 1) diverse redundancy, 2) voting, 3) configuration hopping, and 4) data consistency checking. This architecture has been conceptually applied to controllers for nuclear power systems [21] and shipboard control systems [22] using design patterns under the System-Aware umbrella. Unlike the traditional perimeter defense where users are interested in authentication, signature defense, information anomalies, etc., System-Aware simplified defense to fact-checking and searching for inconsistencies among its critical components.

SECTION 3: METHODS

In order to support the process of defense, it is important to capably describe and judge the CPSs current security status [23]. To this extent, this study develops, and validates detection methods based on the behavior of residuals between disparate navigation units commonly used in UAV navigation. Section 3.1 argues for the use of a set of diverse, redundant navigation components to diminish the chances of adversarial supply chain attacks [7, 21]. Building on the benefits of System-Aware defense paradigm, a detection method leveraging the use of multiple diverse components and data consistency checking is developed.

Section 3.2 introduces an analytic attack model of a UAV navigation system comprising of multiple independent INS and GPS. Section 3.3 A logical decision tree is created by asking three unique questions on the agreements between components in order to isolate infected components. Section 3.4 outlines the unique similarity analysis between components used by [2]. Each of these analyses uses unique analysis respecting the difference in behavior of each component. Section 3.5 develops the detection rule based on the arrival times of the disagreement signal. Section 3.6 describes the simulation test analyzing the behavior of the method during an INS attack and a GPS. Using the simulation model, the study concludes that the method successfully detects and isolates an inconsistent component. Aside from a successful detection and isolation, false alarms and latency of alarms are presented.

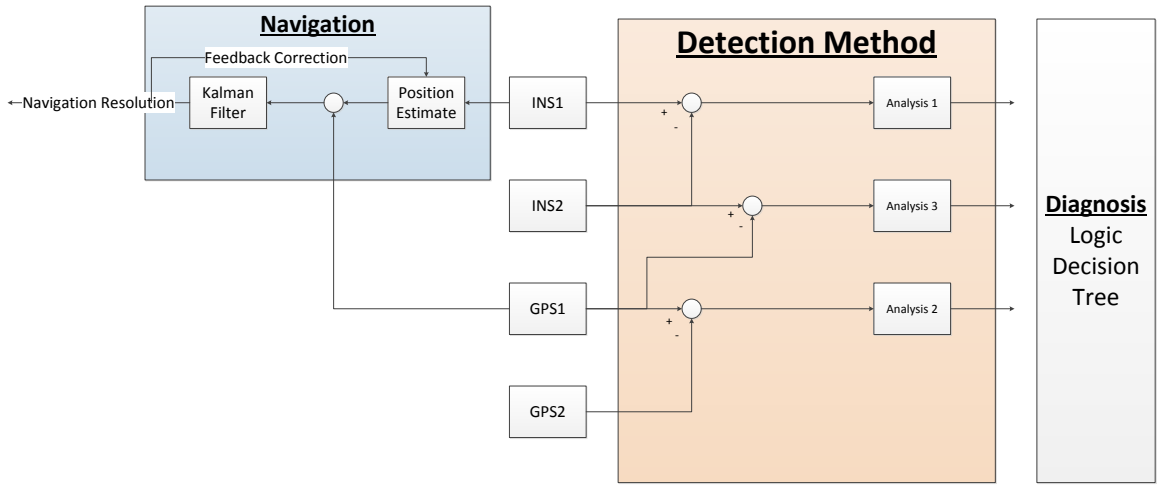

Figure 10: Summary of the detection method. This method is distinct from the navigation resolution between and embedded INS/GPS. In this case, INS1 is filtered with GPS1 to resolve the estimated position of the aircraft. The method tracks the measurement of INS1and is not corrected by the Kalman filter. The measurements from these components are then fed through their corresponding Similarity analysis which determines if a component is in agreement with another. Disagreement signals are then processed to determine the state of the UAV's security using a logical decision tree.

3.1 Diverse Redundant Sensory Components

This work argues that false injections against sensory components can originate from two fronts: 1) spoofing attacks, 2) supply chain attacks. Borrowing from ideas of [7] and [24], diversity of components solves supply chain infiltrations. Figure 11 summarizes the possible effect of carrying sensors from a single vendor. The adversary can infiltrate a single supply line and possibly have total control of the CPS. Figure 12 shows how carrying a redundant set of sensors from diverse vendors alleviates the problem of a supply chain attack. Having multiple sources for vendors allows the CPS to have alternative trusted references. Figure 13 shows the increased difficulty required for an adversary to control a CPS. Since diverse components are drawn from multiple vendors, adversaries are required to infiltrate multiple supply lines, which can be a cumbersome undertaking, possibly motivating the adversary to look for other avenues. Additionally, diverse, redundant

systems promote redundant information on the navigation to assist with checking data consistency.

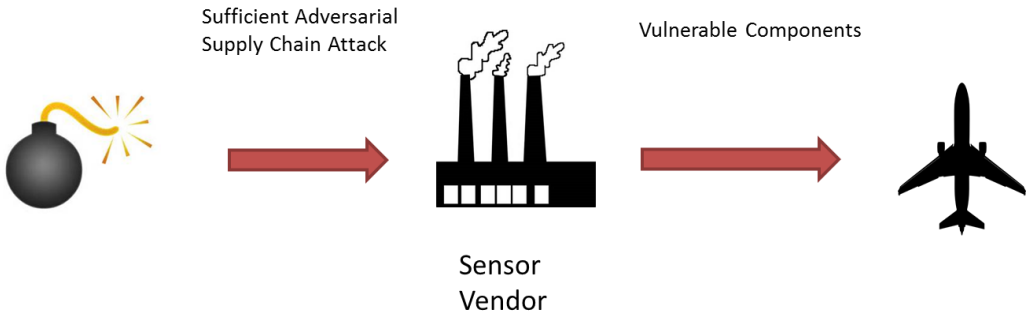

Figure 11: Illustration of a supply chain attack against a component with a single sensor vendor. In this case, the adversary infiltrates the operation of a single vendor. If the UAV sensor components come from one supply chain, then the UAV is susceptible to supply chain cyber-attacks.

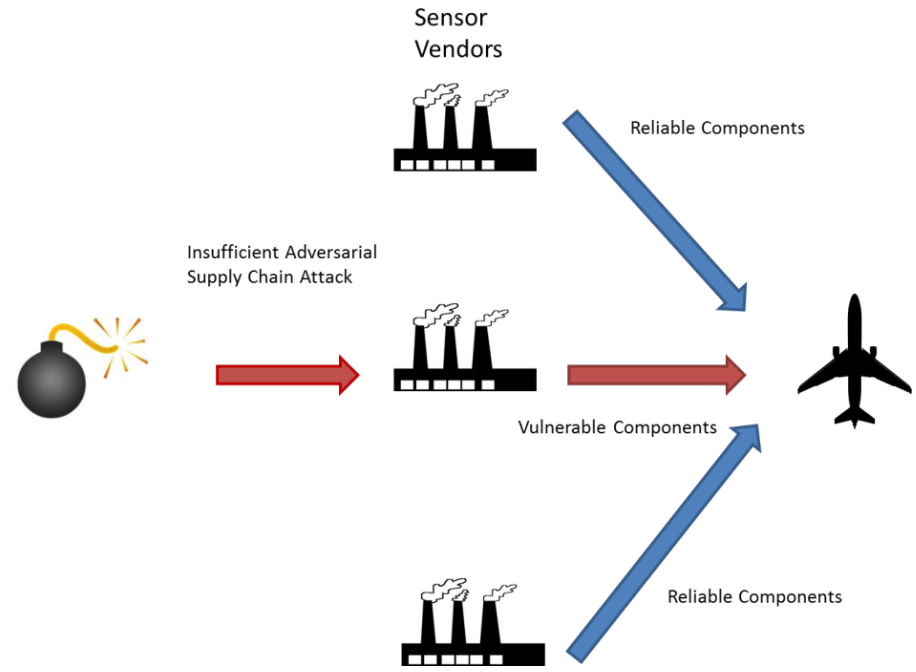

Figure 12: Illustration of a multiple vendor supply chain. In this case, an adversary attacks only a single sensor vendor. Because the UAV developer applies diverse, redundant components, he has potentially multiple trustworthy references.

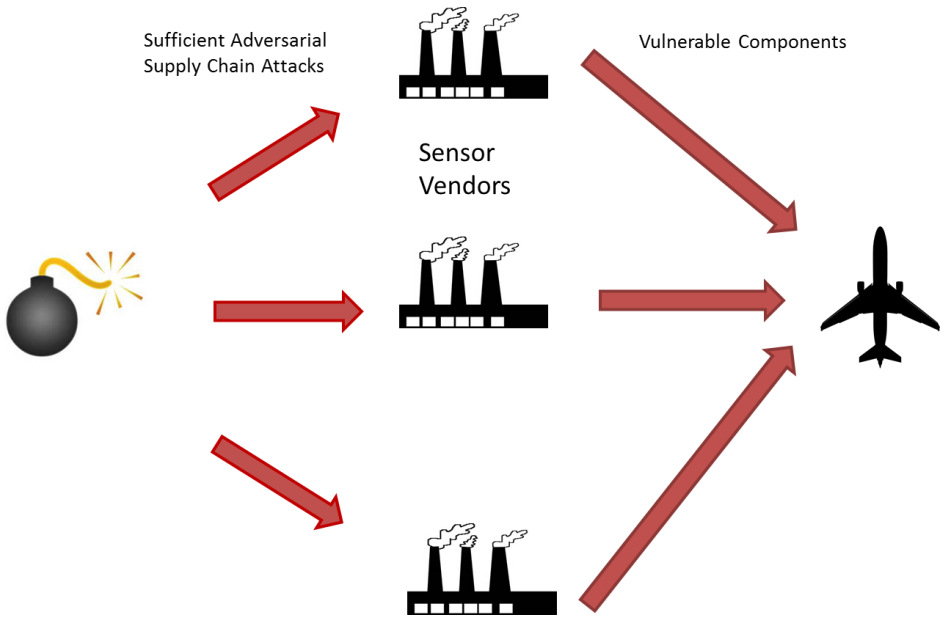

Figure 13: Illustration of an adversary infecting multiple vendor supply chain. This is a successful infiltration of the UAV components. Although this is possible, the cost and effort on the adversary side is much greater than that of a single vendor supply chain. The adversary may be motivated to look for other avenues to attack a system.

Diverse-redundancy affords the assumption that, excluding a GPS signal spoof where all GPSs are deviating together simultaneously, an adversary may only inject false information into one on-board navigation sensor. Using diversity, the problem becomes that of detection using dissimilarities between the available components.

3.2 Analytic System Description and Attack Model

3.2.1 Singular Embedded INS/GPS Model

The behavior of physical systems can generally be described by a mathematical dynamical system. This study assumes a CPS modelled by a linear time-invariant stochastic system with Gaussian noise (which accounts for modeling errors, uncertainties, or external perturbations in the system). The navigation system architecture involves an Inertial Navigation System (INS), and Global Positioning Systems (GPS) components. INS units are navigation components that use accelerometers and gyroscopes to continuously calculate

the position, orientation, velocity and speed of a moving object. All inertial navigation systems suffer from integration drift which are small cumulative errors in measurement of acceleration and angular velocity. INS units while precise in the short-term can cause inconsistencies as accumulated errors compound. It is for this reason designers fuse INS information with GPS measurements, which are noisy but accurate in the long-run [11]. A commonly adopted technique to fuse navigation information is the use of an Extended Kalman Filter. This study considers a simplified UAV navigation system neglecting the altitude, attitude, and rotational motions of a UAV. The input u of the model denotes the directional component of acceleration applied to the UAV in the X- and Y-axis.

Let x be the state of the aircraft whose states are:

- $x_1 \coloneqq$ x-axis coordinate position
- $x_2 \coloneqq$ x-axis component of the velocity
- $x_3 \coloneqq$ y-axis coordinate position
- $x_4 \coloneqq$ y-axis component of the velocity

Then the modelled linear discrete-time system is:

$$x_a(k+1) = Ax_a(k) + Bu(k) + B_c a_c(k) + \partial T w(k)$$

$$z_a(k) = Cx_a(k) + B_o a_o(k) + v(k)$$

Where $x_a(k) \in \Re^{\varrho}$, $u(k) \in \Re^{p}$, $z_a(k) \in \Re^{l}$ are the system state, inputs of the INS unit, and the measurement of the GPS component, and $w^j(k) \in \Re^{q}$, $v^i(k) \in \Re^{\,l}$ are process and measurement noise. It is assumed that $w(k)$ and $v(k)$ are Gaussian white noise of the INS and GPS measurements, respectively, with constant covariance matrices Q and R. Let

$$A = \begin{bmatrix} 1 & T_s & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T_s \\ 0 & 0 & 0 & 1 \end{bmatrix}, \qquad B = \begin{bmatrix} \frac{T_s^2}{2} & 0 \\ T_s & 0 \\ 0 & \frac{T_s^2}{2} \\ 0 & T_s \end{bmatrix}, \qquad C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \qquad \partial T = \begin{bmatrix} \frac{T_s^2}{2} \\ T_s \\ \frac{T_s^2}{2} \\ T_s \end{bmatrix}$$

B_o, B_c are the attack matrices and $a_o(k)$, $a_c(k)$ are persistent, linear deception attacks against the GPS and INS at time k, respectively. Note that the system matrix pairs (A, B) and (A, C) are controllable and observable, respectively.

The aircraft's GPS and auxiliary GPSs takes measurements at every time period k, denoted by $z_a(k)$.

Suppose that K is the steady state Kalman Gain, then the estimate of the attackable system is given by the following steady-state Kalman Filter:

$$\hat{x}_a(k+1) = A\hat{x}_a(k) + Bu_a(k) + K(\mathrm{z_a}(k+1) - CA\hat{x}_a(k) - CBu_a(k))$$

The system inputs is then a function of the estimated state feedback, $u_a(k) = (A - FB)\hat{x}_a$, i.e. the actuators response is a function of the Kalman state estimate. Since the attackers inject attack inputs $a_o(k)$, he or she can directly affect the state of the aircraft.

The true state of the aircraft in any security state is noted as

$$x_a(k+1) = Ax_a(k) + Bu_a(k) + \partial Tw(k)$$

3.2.2 Redundant Components Model

Now suppose a system with redundant sensory components is applied to the model. The system model of M INS units and N GPS units sensory components becomes

$$x_a^{(1)}(k+1) = Ax_a(k) + Bu_a(k) + B_c a_c(k) + \partial Tw^{(1)}(k)$$

$$x_a^{(2)}(k+1) = Ax_a(k) + Bu_a(k) + B_c a_c(k) + \partial Tw^{(2)}(k)$$

$$\vdots$$

$$x_a^{(M)}(k+1) = Ax_a(k) + Bu_a(k) + B_c a_c(k) + \partial Tw^{(M)}(k)$$

$$z_a^{(1)}(k) = C^{(1)}x_a(k) + B_o a_o(k) + v^{(1)}(k)$$

$$z_a^{(2)}(k) = C^{(2)}x_a(k) + B_o a_o(k) + v^{(2)}(k)$$

$$\vdots$$

$$z_a^{(N)}(k) = C^{(N)}x_a(k) + B_o a_o(k) + v^{(N)}(k)$$

$z_a^{(n)}(k)$ denotes the measurements of the n^{th} GPS unit at time k. $x_a^{(m)}(k+1)$ denote the m^{th} INS estimate at time $k+1$.

To use a specific case, this study uses 2 INS units and 2 GPS units:

INS1:

$$x_a^{(1)}(k+1) = Ax_a^{(1)}(k) + Bu_a(k) + B_c a_c^{(1)}(k) + \partial T w^{(1)}(k)$$

INS2:

$$x_a^{(2)}(k+1) = Ax_a^{(2)}(k) + Bu_a(k) + B_c a_c^{(2)}(k) + \partial T w^{(2)}(k)$$

GPS1:

$$z_a^{(1)}(k) = Cx_a(k) + B_o a_o^{(1)}(k) + v^{(1)}(k)$$

GPS2:

$$z_a^{(2)}(k) = Cx_a(k) + B_o a_o^{(2)}(k) + v^{(2)}(k)$$

Where

- $x_1^{(i)}$ ≔ x-axis coordinate position of the i^{th} INS
- $x_2^{(i)}$ ≔ x-axis component of the velocity of the i^{th} INS
- $x_3^{(i)}$ ≔ y-axis coordinate position of the i^{th} INS
- $x_4^{(i)}$ ≔ y-axis component of the velocity of the i^{th} INS
- $z_1^{(j)}$ ≔ x-axis coordinate position of the j^{th} GPS
- $z_2^{(j)}$ ≔ y-axis coordinate position of the j^{th} GPS

3.3 Fault Isolation

The isolation method begins with a Logical Decision Tree which diagnoses which component is infected. There are three analyses questioning the agreement between two components (See Figure 14).

1. Analysis 1: Are the INSs in agreement/similar?

2. Analysis 2: Are the GPS in agreement/similar?
3. Analysis 3: Are GPS1 and INS1 in agreement/similar?

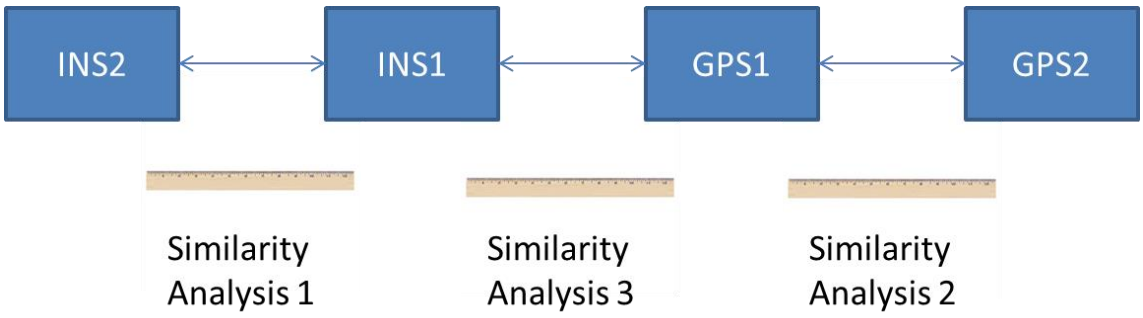

Figure 14: Diagram depicting the relationship and their corresponding analysis between two components.

These three questions are sufficient in the isolation process of an infected component. Figure 15 diagrams the logical decision tree that detects and isolates an infected component. For example, a normative scenario with no cyber-attacks necessitates the condition that all the questions are in the affirmative. However, an attack on GPS1 necessitates the condition that Question 1 are in the affirmative and Question 2 and 3 are in the negative. Table 1 shows the mutually exclusive and exhaustive scenarios, under the assumption that an adversary can only infect one supply chain, and their resulting detection and isolation diagnosis.

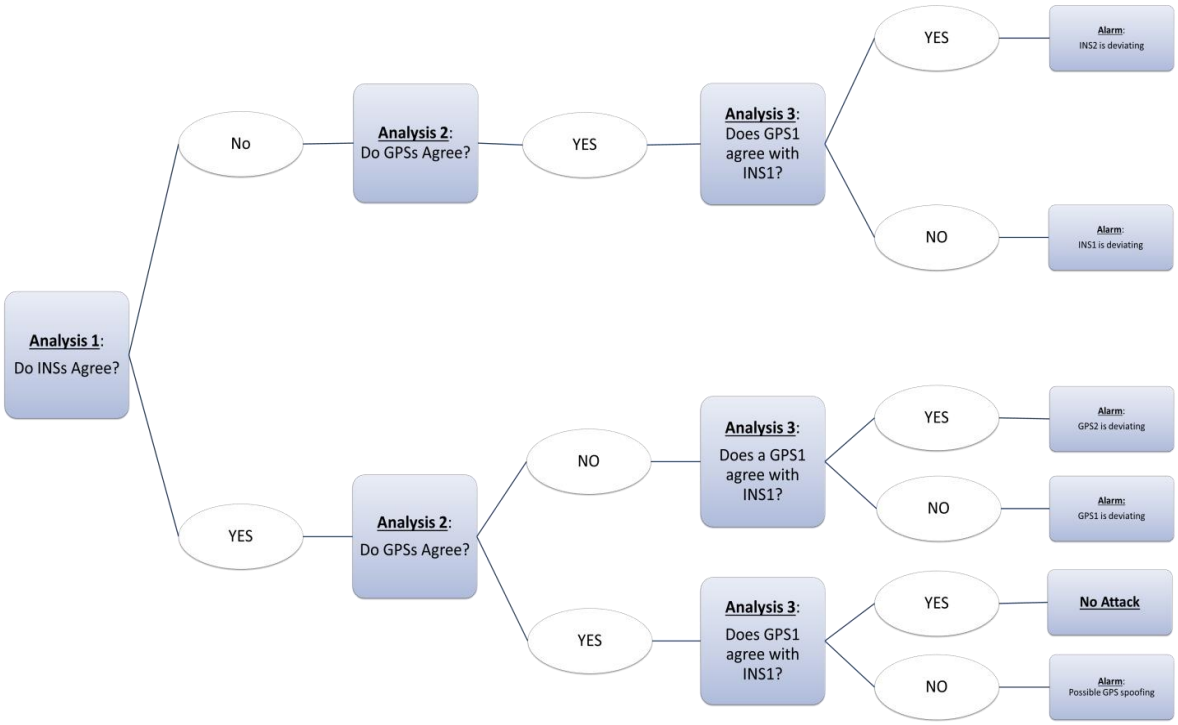

Figure 15: Logical Decision Tree that asses the fault given certain conditions. Analysis 1 checks the INS agreement. Analysis 2 checks the agreement between the GPSs. Analysis 3 checks the similarity between GPS1 and INS1. Analysis 4 checks the similarity between GPS2 and INS1.

Table 1: Summary of attack diagnosis

Agreement Analysis	1	2	3
	RESULTS		
No Attack	0	0	0
INS2 is deviating	1	0	0
INS1 is deviating	1	1	0
GPS1 is deviating	0	1	1
GPS2 is deviating	0	0	1
GPS signal is spoofed	0	1	0

3.4 SIMILARITY ANALYSIS

To begin the process of checking the consistency of the navigation measurement output, the method involves looking at the agreement between 2 distinct sensors. To compare the results between Kwon et al [2], the method In Section 3.2, a list of agreement analysis is developed to determine a cyber-attack. Since these sensors are disparate in varying levels, each similarity analysis requires a unique method of measuring similarity. Section 3.4.1-3 describes the similarity analyses and their derivation.

3.4.1 Similarity Analysis 1: INS1-INS2

This section describes how the method assesses the agreement between two INSs. Since INSs suffer from compounding drift, the method looks instead at the velocity and acceleration residuals between the two components. Note that the INS discrete time-invariant linear model introduced in Section 3.1.2 for the ith INS unit:

$$x_a^{(i)}(k+1) = Ax_a^{(i)}(k) + Bu_a(k) + B_c a_c^{(i)}(k) + \partial T w^{(i)}(k)$$

For 2 INSs, this similarity method generates the residual of the acceleration measurement of the 2 INSs.

$$r_1(k) = \left(Bu_a^{(1)}(k) + B_c a_c^{(1)}(k) + \partial T w^{(1)}(k)\right) - \left(Bu_a^{(2)}(k) + B_c a_c^{(2)}(k) + \partial T w^{(2)}(k)\right)$$

$$= B\left(u_a^{(1)}(k) - u_a^{(2)}(k)\right) + \partial T\left(w^{(1)}(k) - w^{(2)}(k)\right) + \alpha_c(k)$$

where $\alpha_c(k) = B\left(a_c^{(1)}(k) - a_c^{(2)}(k)\right)$. Since $w^{(i)}(k)$ is a zero-mean Gaussian, and since $u_a^{(1)}(k) = u_a^{(2)}(k)$:

$$N\left(0, \partial T'\left(Q^{(1)} + Q^{(2)}\right)\partial T\right) \sim \partial T\left(w^{(1)}(k) - w^{(2)}(k)\right)$$

If $\alpha_c(k) \neq 0$, then $r(k)$ loses its non-zero Gaussian characteristics. Thus, a valid test for intrusion is to test the non-zero mean normality of the residuals. The compound scalar test allows one to do so. Since $r(k)$ is a bivariate standard normally distributed random variable. Let

$$\mathbb{R}_1(k) = r_1(k)^T \left((C^T \partial T^T C) \Sigma_{r_1}^{-1} (C^T \partial T C) \right) r_1(k)$$

be the sum of squares of the residual with 2 degrees of freedom between the two INS acceleration measurements with covariance matrix

$$\Sigma_{r_1} = Q^{(1)} + Q^{(2)}$$

The method performs a compound scalar test to test the normality of the residuals [5]. The hypothesis test becomes

$$\mathcal{H}_0 : \chi\big(\mathbb{R}_1(k)\big) < threshold$$

$$\mathcal{H}_1 : \chi\big(\mathbb{R}_1(k)\big) > threshold$$

where $\mathcal{H}_1$ signifies a non-agreement at time k. Suppose $threshold = .99$.

3.4.2 Similarity Analysis 2:GPS1-GPS2

This section describes a method to measure the similarity of the two GPS units. The model for the GPS measurement is

$$z_a^{(i)}(k) = C x_a(k) + B_o a_o^{(i)}(k) + v^{(i)}(k)$$

for the ith GPS sensor. The residual for 2 GPS (GPS1 and GPS2) is

$$\begin{aligned} r_2(k) &= z_a^{(1)}(k) - z_a^{(2)}(k) \\ &= \left(C x_a(k) + B_o a_o^{(1)}(k) + v^{(1)}(k) \right) - \left(C x_a(k) - B_o a_o^{(2)}(k) - v^{(2)}(k) \right) \\ &= \left(v^{(1)}(k) - v^{(2)}(k) \right) + \alpha_o(k) \end{aligned}$$

where $\alpha_o(k) = B_o \left(a_o^{(1)}(k) - a_o^{(2)}(k) \right)$. Since $v^{(i)}$ is a zero-mean Gaussian random variable:

$$N\big(\mathbf{0}, R^{(1)} + R^{(2)}\big) \sim \left(v^{(1)}(k) - v^{(2)}(k) \right)$$

If $\alpha_o(k) \neq 0$, then r_1 loses its non-zero Gaussian characteristics. Thus, a valid test for intrusion is the compound scalar test. Let X be the sum of squares of the residual with 2 degrees of freedom between the two INS acceleration measurements

$$\mathbb{R}_2(k) = r_2(k)^T\left(\Sigma_{r_2}^{-1}\right)r_2(k)$$

$$\Sigma_{r_2} = R^{(1)} + R^{(2)}$$

The method performs a compound scalar test to test the normality of the residuals [5]. The compound scalar hypothesis test becomes

$$\mathcal{H}_0: \chi\left(\mathbb{R}_2(k)\right) < threshold$$

$$\mathcal{H}_1: \chi\left(\mathbb{R}_2(k)\right) > threshold$$

where $\mathcal{H}_1$ signifies a non-agreement between the GPSs at time k. Suppose $threshold = .99$.

3.4.2 Similarity Analysis 3 and 4: GPS1-INS1

GPS and INS units are distinct components carrying unique characteristics. Although INSs measures acceleration directly, GPS does not measure the direct acceleration of an aircraft. Thus, we are forced to perform analysis on the GPS and INS position measurements domain. Due to INS drift, the residuals between INS position estimates and GPS positions are non-zero mean Gaussian. However, modern INSs, calibrated sufficiently, are capable to keep an precise position of an aircraft. Assuming the case where the INS are capable enough to track the aircraft position, we simplify the similarity measurements assuming that the residuals between and INS and GPS have non-zero mean Gaussian characteristics. The model for the simultaneous INS1 position and GPS1 position is

$$x_a^{(1)}(k+1) = Ax_a(k) + Bu(k) + B_c a_c^{(1)}(k) + \partial T w^{(1)}(k)$$

$$z_a^{(1)}(k) = Cx_a(k) + B_o a_o^{(1)}(k) + v^{(1)}(k)$$

Then the residual of the INS and GPS position, or

$$r_3(k) = x_a^{(1)}(k) - z_a^{(1)}(k)$$
$$= CAx_a^{(1)}(k-1) + CBu(k-1) - Cx_a(k) + \left[B_o a_o^{(1)}(k) - B_c a_c^{(1)}(k)\right]$$
$$+ \left[C\partial T w^{(1)}(k-1) - v^{(1)}(k)\right]$$

has non-zero mean Gaussian characteristics. Assume that

$$CAx_a^{(1)}(k-1) + CBu(k-1) = Cx_a(k)$$

and if there are no cyber-attacks:

$$B_o a_o^{(1)}(k) + B_c a_c^{(1)}(k) = 0$$

then $r_3(k)$ is a Gaussian distributed random variable, with covariance

$$\Sigma_{r_3} = C\partial T^T C^T Q^{(1)} C^T \partial T C + R^{(1)}$$

And the sum of square is

$$\mathbb{R}_3(k) = r_3(k)^T \Sigma_{r_3}^{-1} r_3(k)$$

Then the compound scalar hypothesis test becomes

$$\mathcal{H}_0 : \chi\big(\mathbb{R}_3(k)\big) < threshold$$

$$\mathcal{H}_1 : \chi\big(\mathbb{R}_3(k)\big) > threshold$$

Let $threshold = .99$.

3.5 Detection Rule

Finally, this section constructs a rule for detection. Suppose the disagreement signal is an exponential distribution with a fixed arrival rate (false disagreement rate). If X_i is the time of the ith false disagreement, then if > 0 ,$X_{i-N} - X_i$ is the time between ith false disagreement and $i + Nth$ false disagreement. If there is a deviation, then disagreement signals cluster (i.e. $X_{i-N} - X_i$) will be short (See Figure 16 and Figure 17).

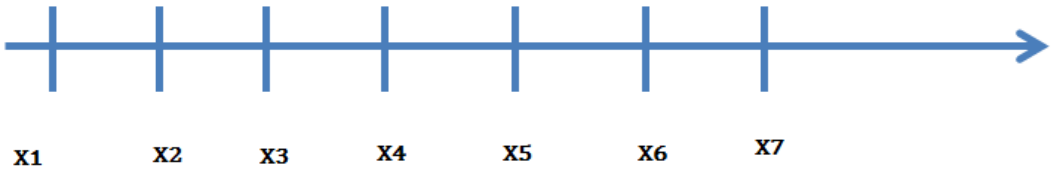

Figure 16: Arrival times for a no-attack normative scenario. We see that the arrival time of the disagreement signals are distributed evenly with rate ***R***.

Figure 17: Disagreement times for disagreeing components. We see that the times between arrivals are decreasing.

For a given probability P, then if $X_{i-N} - X_i < T$ then we declare a disagreement where T is the inverse Gamma function:

$$T = InvGamma(P, N, 1/R)$$

where

$$N = number\ of\ disagreement\ signals$$

$$R = time\ of\ arrival\ for\ each\ disagreement\ signal$$

Consider the values

$$P = .05$$

$$N = 10$$

$$R = rate\ of\ false\ alarms\ during\ a\ normative\ scenario$$

then $T = 542$.

In sum, if 10 disagreement signals occur within $T = 542$ then we raise an alarm.

3.6 Validation Methods

The validation of this detection method begins with a simulation model of an attack against a UAV's navigation system of four components (INS1, INS2, GPS1, and GPS2). StatisticalbasedIntrusionDetection.m Matlab script found in APPENDIX A simulates the flight path and the measurement readings for each sensor component. The Matlab script simulates outputs from the GPS and INS, given an initial position (de facto $= [0\ 0]^T$) and target (de facto $= [1500\ 1500]^T$). In the simulation, the aircraft carries two INS units, labeled INS1 and INS2; and two GPS units, GPS1, and GPS2.

The study involves a simulation for the component interactions, estimation of locations, residual generation. The simulation is simplified to exclude the dynamics and actuating control of the aircraft. Also, assume that the linear model is an acceptable approximation of the non-linear dynamics of flight behavior (as assumed by [2]). The simulation also assumes several constraints:

1. The speed of the aircraft cannot exceed 15 meter distance per second, and has a constant acceleration of 20 m/s^2.
2. $T_s = .1$sec, is the sample time (i.e. 10 Hz)
3. The attack matrix B_o has such property that $B_o = [0\ 1]^T$ and $B_c = [0\ 0\ 1\ 0]^T$, i.e. the adversary applies additive false injection in the y-direction.
4. Uninfected GPSs will measure the true position of the aircraft whose actuators are affected by the Kalman estimate between INS1 and GPS1.
5. Attacks on INS1 and GPS1 occur at 250^{th} second of the simulation.
6. Table 2 summarizes the noise magnitude of each navigation sensor. GPSs have a between 3-4m standard deviation. And INSs typically vary in their accuracy (simulation uses .05 and .07m/s^2).

7. Because of the presence of false disagreements, the method requires 50% disagreement signal cluster rate per 10 seconds for each of the 4 analysis to conclude that a sensor is deviating from one another.

Table 2: List of the various component standard deviations tested in the simulation

Standard Deviation of each navigation component	
INS1	0.05 m/s^2
INS2	0.07 m/s^2
GPS1	3 m
GPS2	4 m

The study involves the analysis of false alarms in the system and an analysis of the latency of alarm for various deviation rates. The Section 4 only considers attack on INS1 (Section 4.2), GPS1 (Section 4.3), and a GPS spoof (Section 4.4).

SECTION 4: RESULTS

Section 4 presents the results and capabilities of the detection method. Section 4.1 analyzes the normative case of having no attacks on the system. In this case, false alarms generated by the method are measured using multiple trials of the navigation simulation. Section 4.1 simulates a normative flight scenario with no adversarial manipulation. This section, also records 100 flight Section 4.2 simulates the behavior of the detection method during an attack of 0.1 m/s^2 rate of deviation injected to INS1. Section 4.3 simulates the behavior of the detection method during an attack of 0.1 m/s rate of deviation injected to GPS1. Finally, an analysis of the detection latency against various deviation rates is presented in Section 4.4.

4.1 DEVIATION RATES AGAINST TRADITIONAL GPS AND INS

Suppose a UAV has only one INS and one GPS. This section establishes the effects of 2 scenarios: 1) $0.1m/s^2$ deviation on INS and 2) 0.1m/s deviation on GPS much like that of Kwon et al [2] so as to compare the effectiveness of the detection method. Figure 18 shows that statistical nature of the residuals between the Kalman Corrected INS and the GPS when an adversary applies a $0.1m/s^2$ deviation on the INS. Notice that residual analysis does not strongly detect the false injection.

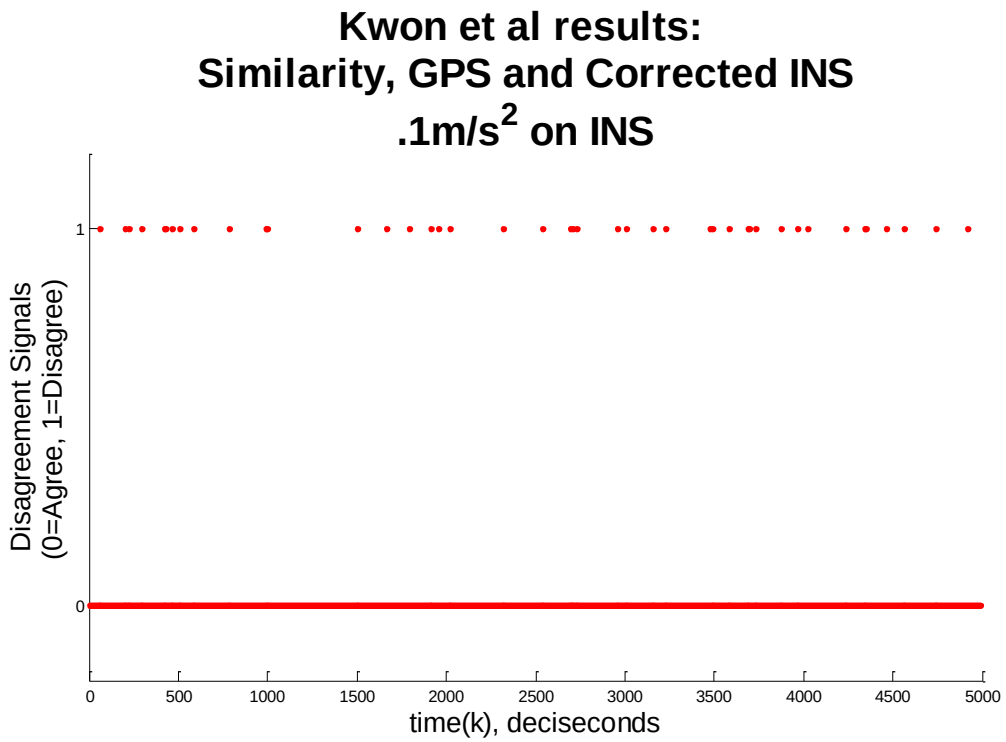

Figure 18: Residual Analysis of a .01m/s^2 deviation applied to the INS. Notice that the residuals are detecting the deviation between INS and GPS.

In contrast to a 0.1m/s false injection attack on GPS, Figure 19 shows a few alarms; however, they do not strongly detect a false injection; again, aligning with the results of Kwon et al [2].

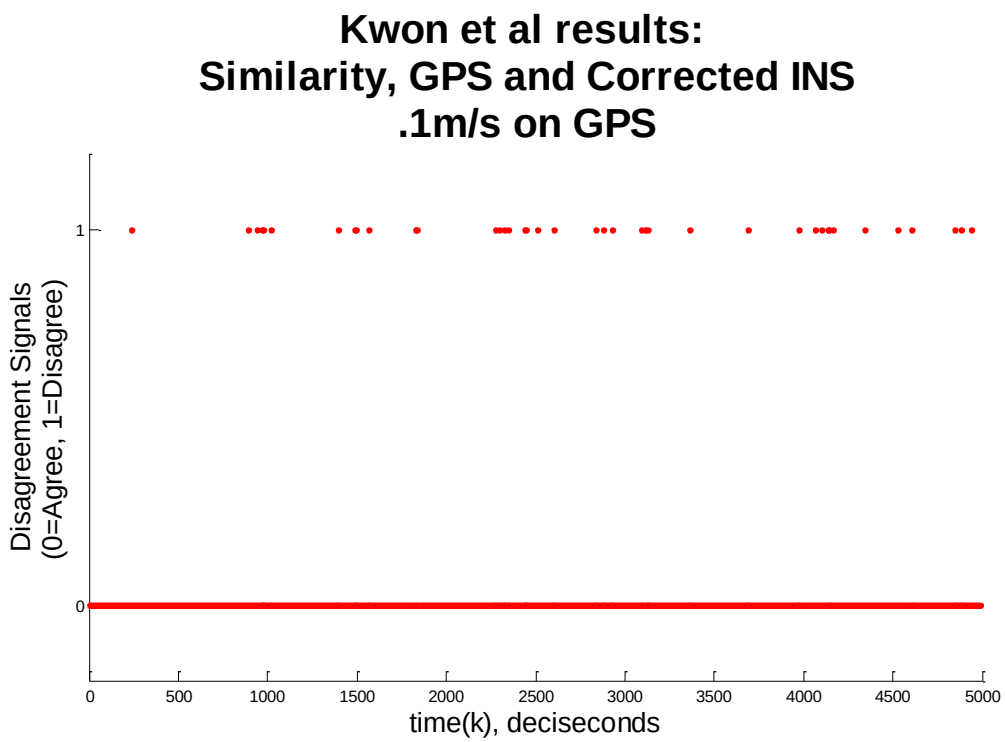

Figure 19: Residual Analysis for INS/GPS, attack rate of 1m/s applied to GPS. We see that there are a few alarms however the signals do not cluster, concluding that residual analysis does not detect attack.

The study chooses deviation rates less than $0.1 m/s^2$, for the INS, and 0.1m/s deviation rate, on the GPS to test the responsiveness of the detection method.

4.2 Normative No-Attack Scenario

Figure 20 illustrates the simulated position measurements for GPS1 and INS1 for a simulated flight duration of 500sec (about 8 min). Notice that the INS (blue line) does not escape the error bounds of the GPS (red). The normative scenario examines the behavior of the similarity analyses between the components. Figure 21-Figure 23 show that all components are in agreement during a no-attack flight simulation trial.

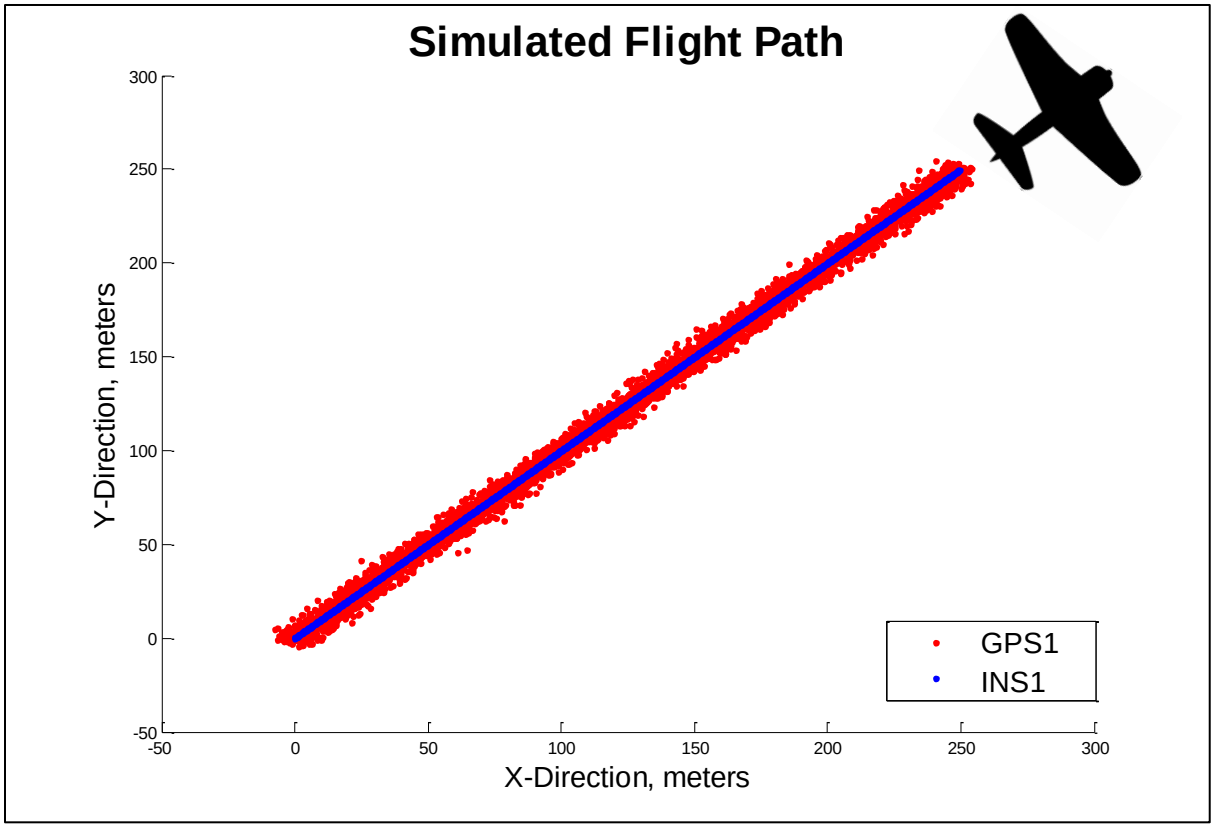

Figure 20: Simulated GPS and INS measurements of a no-attack scenario of an aircraft flight path.

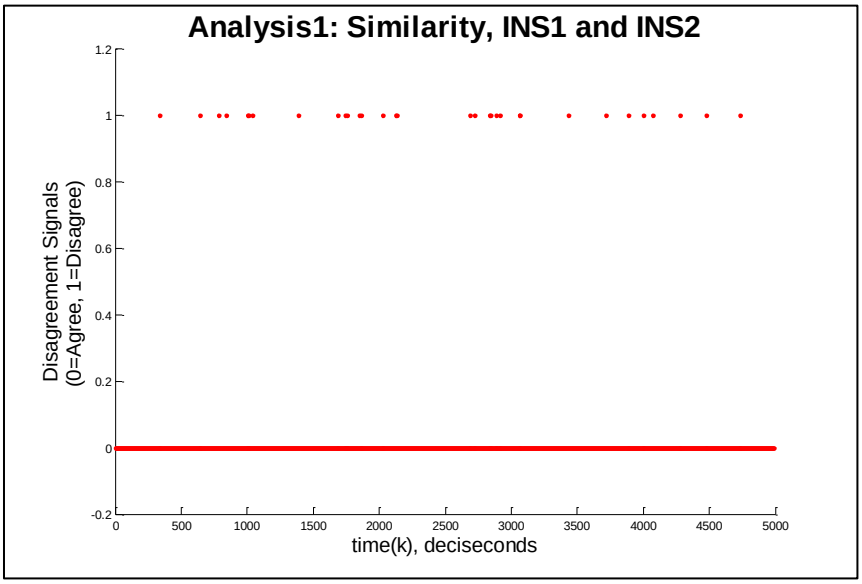

Figure 21: Analysis 1 shows that INSs are in agreement.

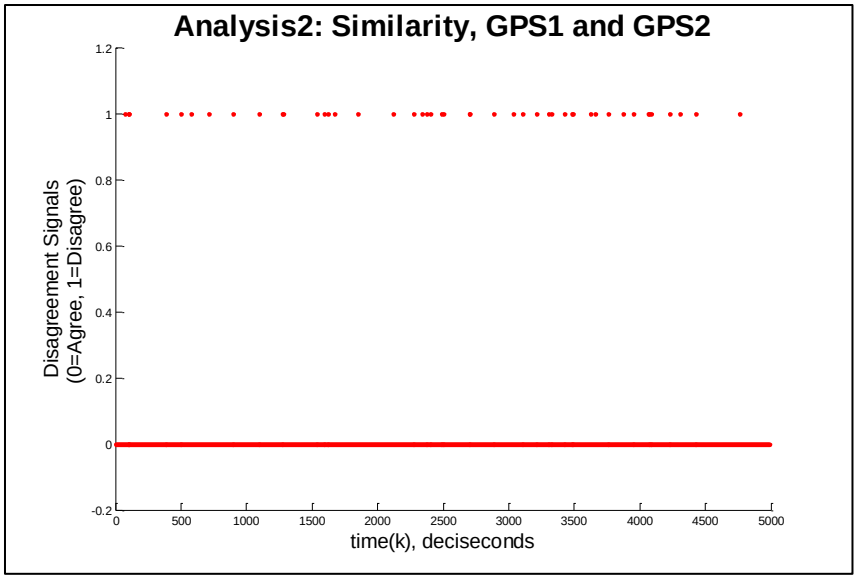

Figure 22: Analysis 2 shows that GPSs are in agreement.

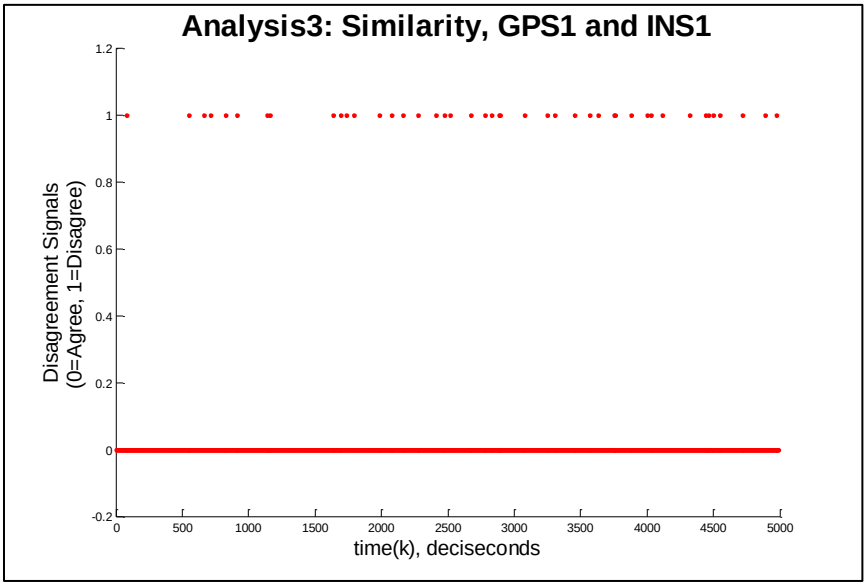

Figure 23: Analysis 3 shows that INS1 and GPS1 are in agreement.

The results show that there are some disagreements between the sensors; however, notice that these disagreement signals do not cluster significantly. Table 1: Summary of attack diagnosis lists the average number of disagreements for each analysis for 500sec of 100 simulated flight trials (or $5x10^3$ sample points). The false disagreement rates are about 1%.

Table 3: Lists the false alarm rate of each of the Similarity Analysis (1-3)

Analysis	Average Number of Disagreement Signals	Standard Deviation	False Disagreement Rate
1	50	4.8	0.01
2	49	6.8	0.0098
3	51	7.3	0.0102

4.3 FALSE INJECTION ATTACK ON INS1

This section simulates a 0.1 m/s^2 false injection attack on INS1. Figure 24 depicts the impact of such an attack for a 500sec simulated flight. The blue line deviates away from the true location of the aircraft (i.e. GPS1 in red). Figure 25 show that there are clustering disagreements between INS1 and INS2. Figure 26 show that there are no disagreements between GPS1 and GPS2. Figure 27 shows that there are clustering disagreements between GPS1 and INS1. Thus, the method correctly detects and isolates the infected INS1 component.

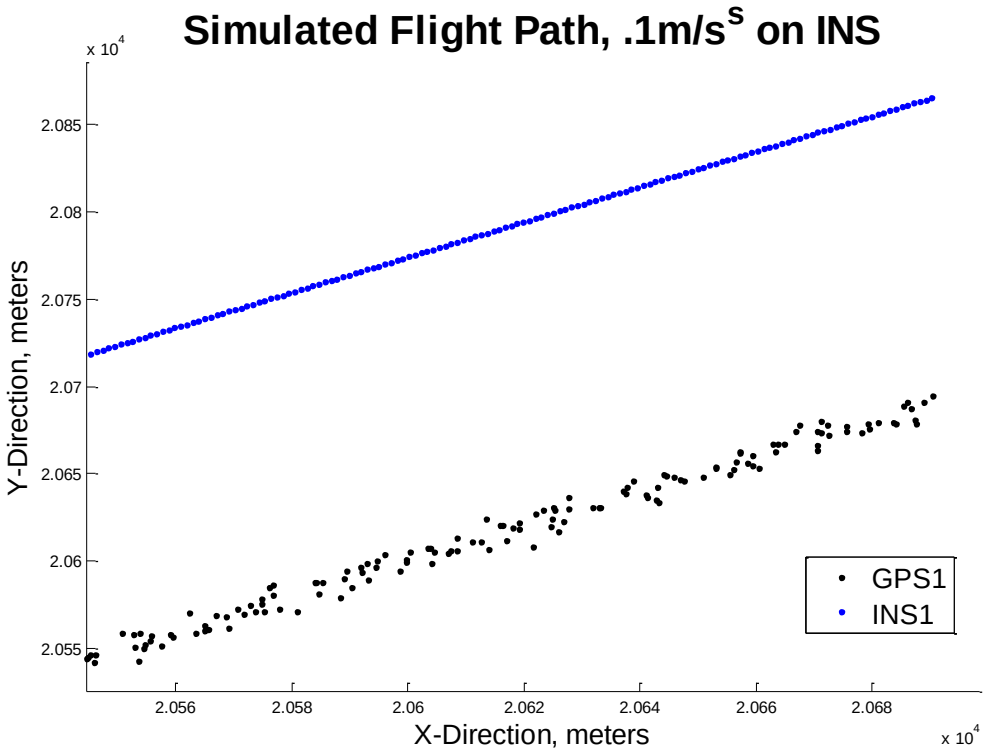

Figure 24: Flight path of a 500sec duration with a .1 m/s^2 deviation applied to INS1. The effect of the deviation is about 2km in a 30min flight

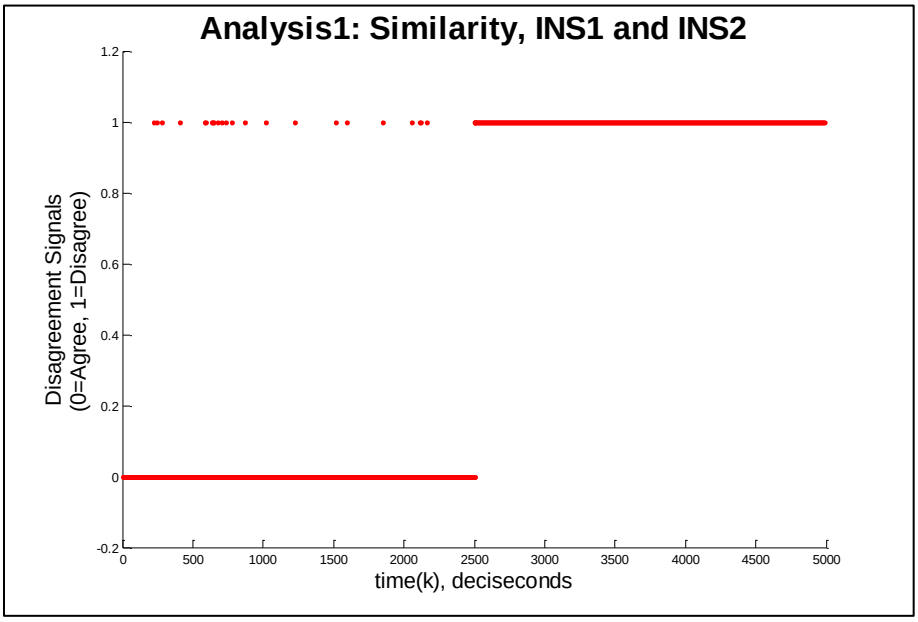

Figure 25: Analysis 1 shows that INSs are in disagreement.

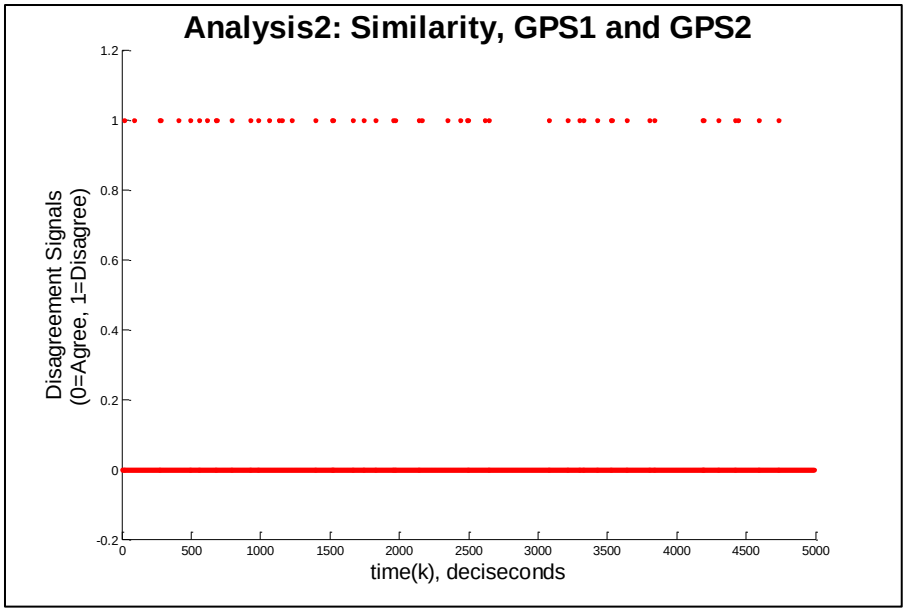

Figure 26: Analysis 2 shows that GPSs are in agreement.

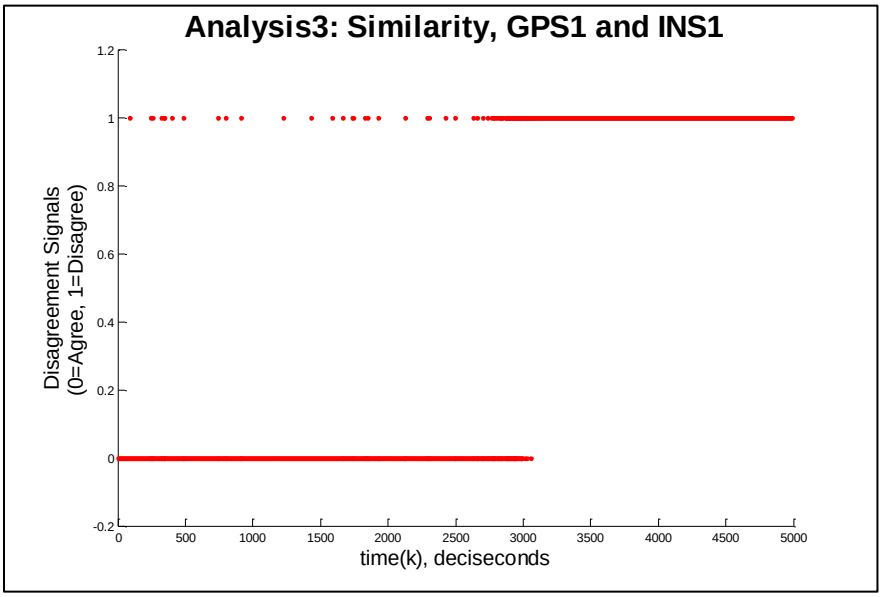

Figure 27: Analysis 3 shows that GPS1 and INS1 are in disagreement.

4.4 False injection attack on GPS1

This section simulates a .1m/s^2 rate of deviation false injection attack on INS1. Figure 24 depicts the impact of such an attack for a 500sec simulated flight. The GPS1 (red clusters) deviates away from the true location of the aircraft (i.e. INS1 in blue). Figure 25 show that there are clustering disagreements between INS1 and INS2. Figure 26 show that there are disagreements between GPS1 and GPS2. Figure 27 shows that there are clustering disagreements between GPS1 and INS1. Thus the method correctly detects and isolates the infected GPS1 component.

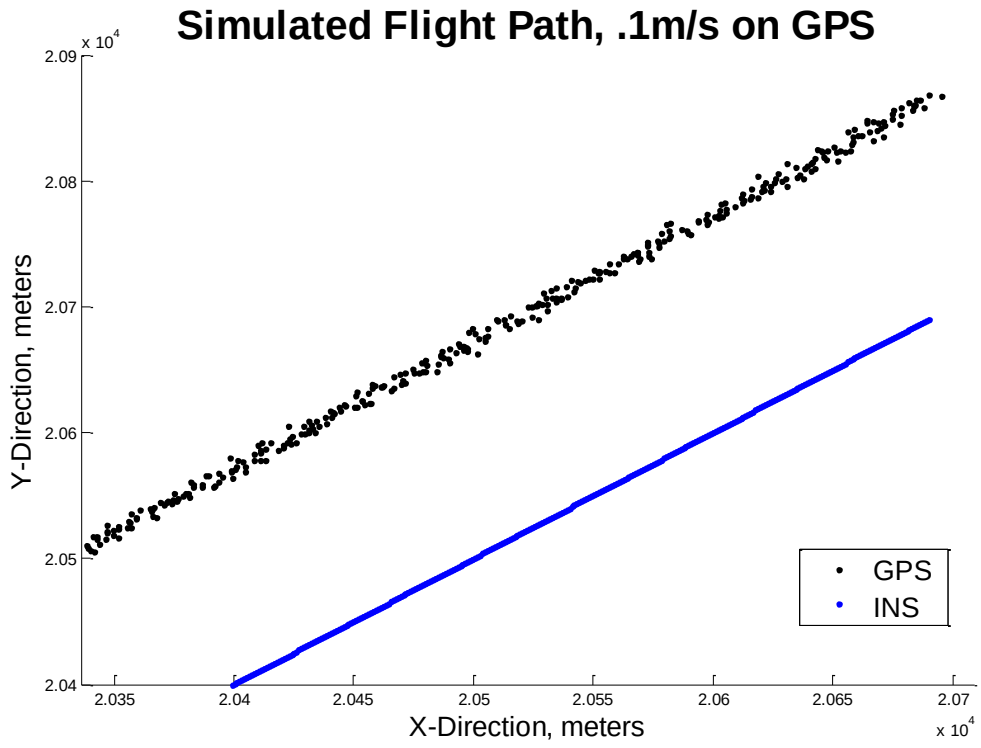

Figure 28:Simulated flight path and false injection attacks against GPS1. The effect of the deviation is about 3km in a 30min flight

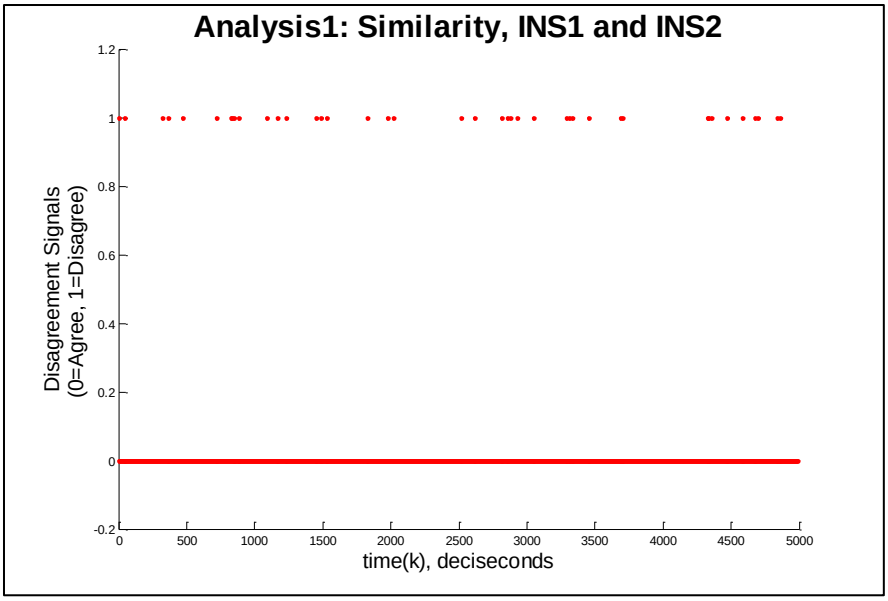

Figure 29: Analysis 1 shows that INSs are in agreement.

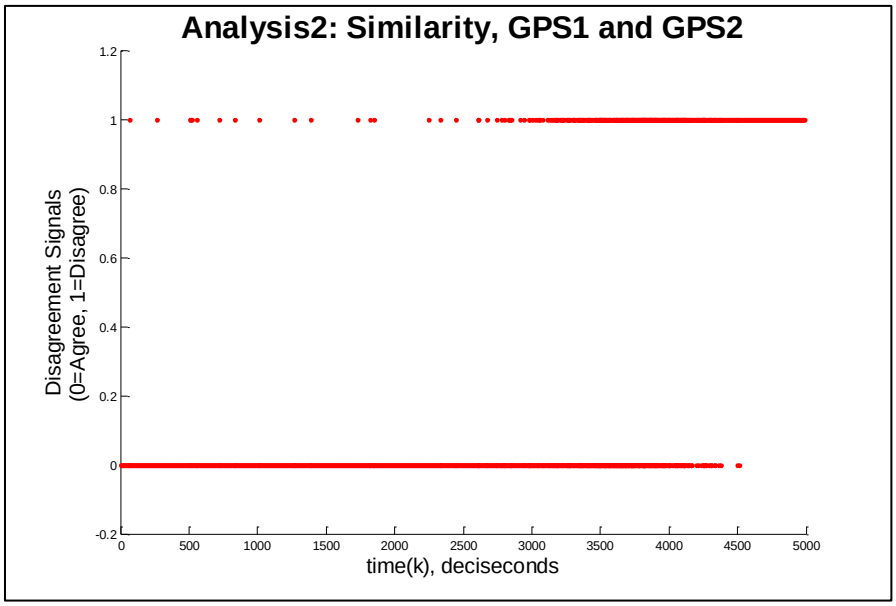

Figure 30: Analysis 2 shows that GPS1and GPS2 are in disagreement.

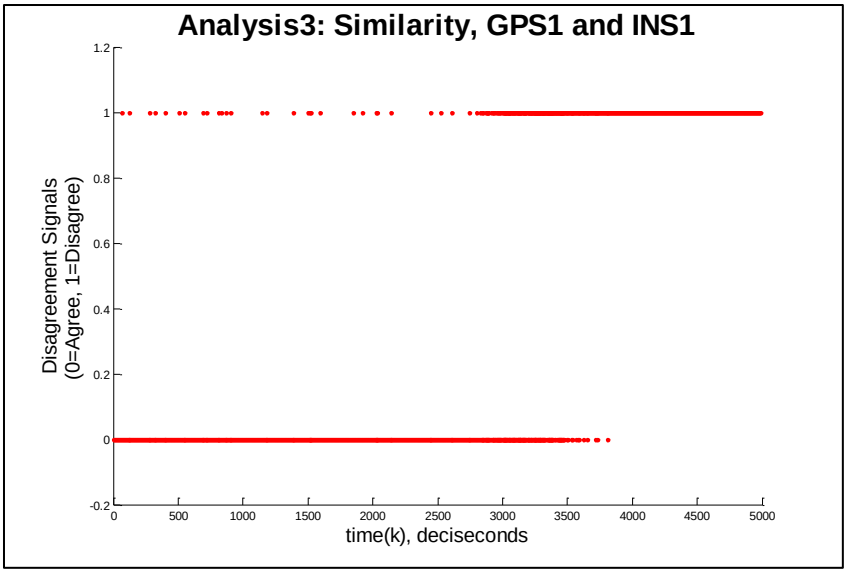

Figure 31: Analysis 3 shows that GPS1 and INS1 are in disagreement.

4.5 GPS SPOOFING ATTACK

This section examines the detection method behavior on a spoofing attack. A spoofing attack involves the hi-jacking of a GPS satellite signal. The target GPS receivers receive false signals from the adversarial transmitters, allowing the adversary to fool the position of the receiver. Consider a GPS spoof that hi-jack the signal of all GPS receivers present in the UAV (i.e. GPS1 and GPS2). The attack biases the GPS positions in the y-direction at a rate of $0.1 m/s^2$. Flight path of a GPS spoofing attack which hi-jacks GPS1 and GPS2. depicts the effect of the position measurements of the GPS spoofing attack. Figure 33 and Figure 34 conclude that INSs are in agreement and GPSs are in agreement; however, Figure 35 show that INS1 and GPS1 are in disagreement. This is enough to conclude that a GPS spoofing attack is occuring.

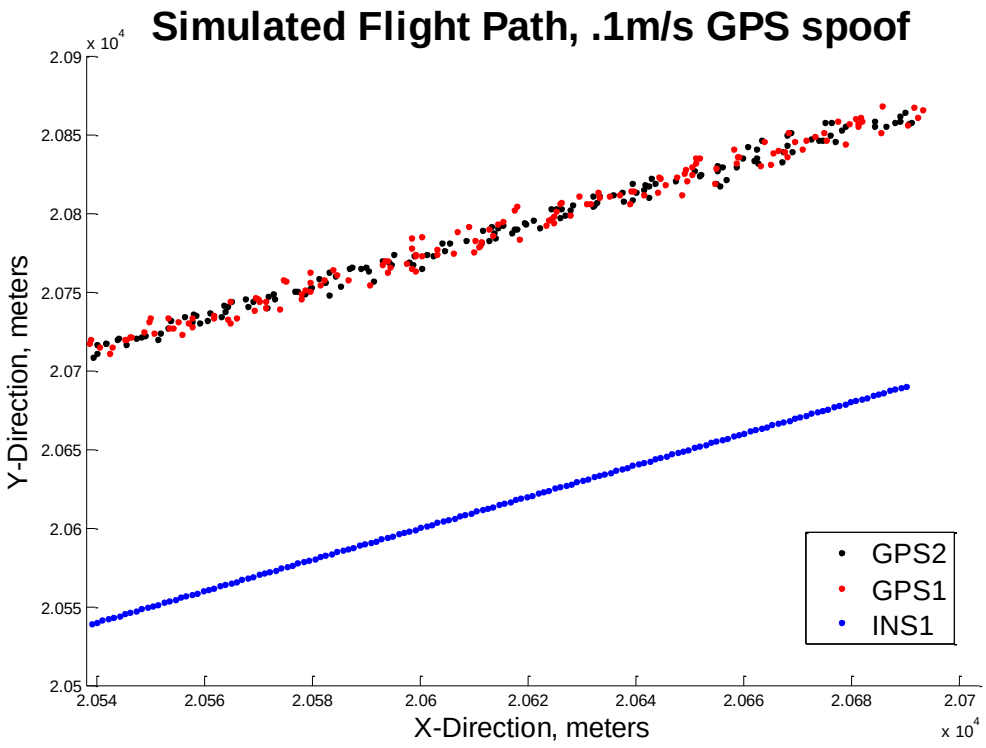

Figure 32: Flight path of a GPS spoofing attack which hi-jacks GPS1 and GPS2. The effect of the deviation is about 3km in a 30min flight

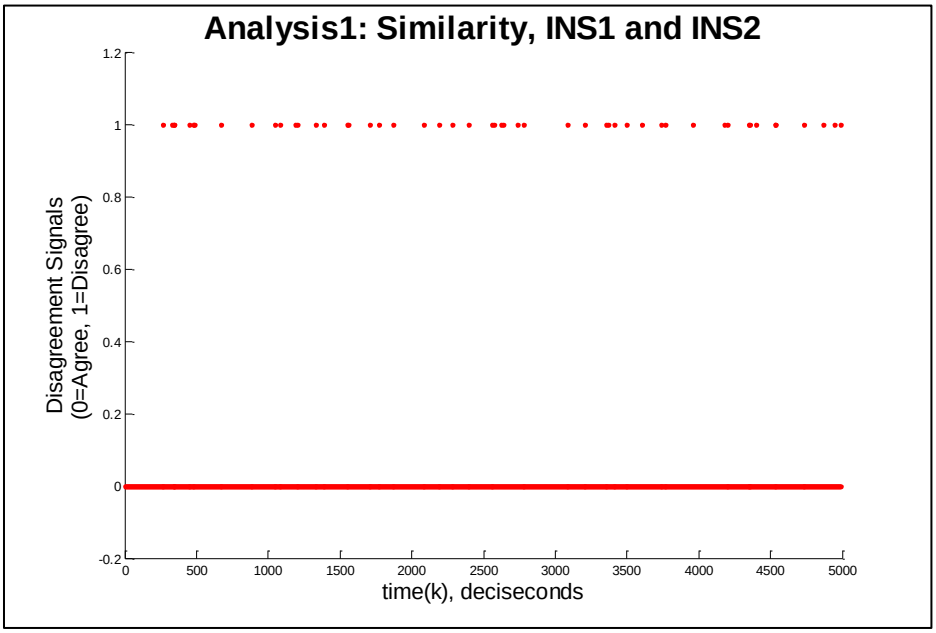

Figure 33: Analysis 1 concludes that INS1 and INS2 are in agreement.

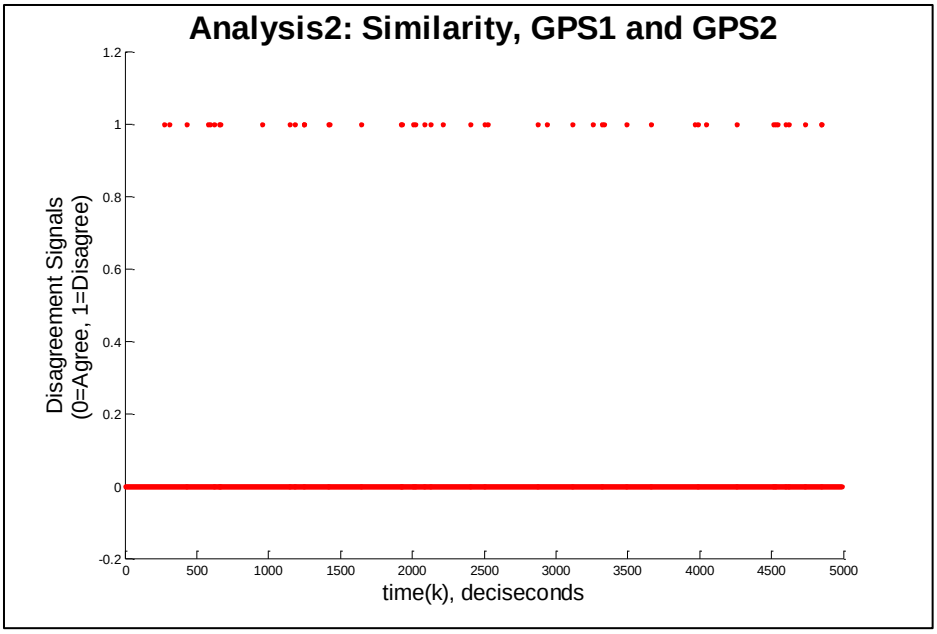

Figure 34: Analysis 2 concludes that GPS1 and GPS2 are in agreement.

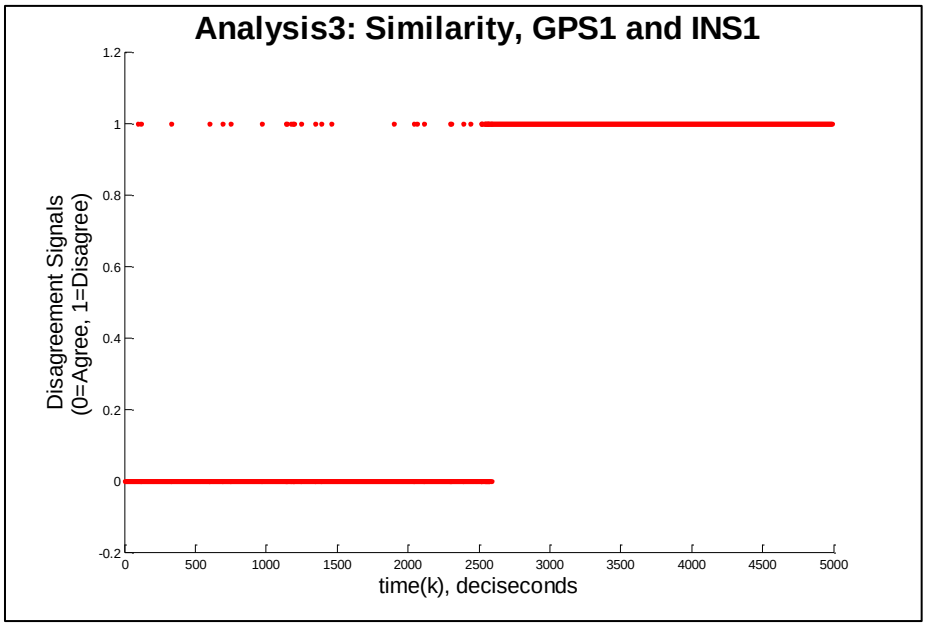

Figure 35: Analysis 3 shows that GPS1 and INS1 are in disagreement. Thus, the method concludes that there exists a GPS spoofing attack.

4.6 LATENCY ANALYSIS

This section analyzes the latency, or the detection delay, of the method for various rates of deviations. Note that the attacks occur at 250^{th} second of the simulation. We use detection rule described in Section 3.4 using an inverse Gamma function with parameters:

$$P = 0.05$$

$$N = 10$$

$$R = 0.01\ (False\ alarms)$$

Table 4 lists the average detection time of 100 simulation trials of an attack on INS1. As the rate of deviation increases linearly the average latency decreases marginally.

Table 4: Describes the latency of various devation applied to GPS1. Notice that as the rate of deviation increases the average latency decreases

Rate of Deviation(m/s)	Time until Detection (min)	Distance after 250sec
0.9	0.20	10.96
0.8	0.25	12.05
0.7	0.31	12.88
0.6	0.34	12.17
0.5	0.41	12.39
0.4	0.50	12.00
0.3	0.72	12.98
0.2	1.13	13.56
0.1	2.35	14.10

Table 5 lists the average detection time of 100 simulation trials of an attack on GPS1. As the rate of deviation increases linearly the average latency decreases marginally.

Table 5: Describes the latency of various devation applied to INS1. Notice that as the rate of deviation increases the average latency decreases.

Rate of Deviation(m/s^2)	Time until Detection (min)	Deviation Impact
0.1	1.34	8.04
0.09	1.55	8.37
0.08	1.75	8.40
0.07	2.06	8.65
0.06	2.34	8.43
0.05	2.77	8.31
0.04	3.24	7.77
0.03	3.72	6.69
0.02	3.94	4.73

4.7 Parameter Design

In order to decrease the amount of false alarms, the designer should set the parameter α as high as possible and β as low as possible. This section gives a guideline on how to design these parameters.

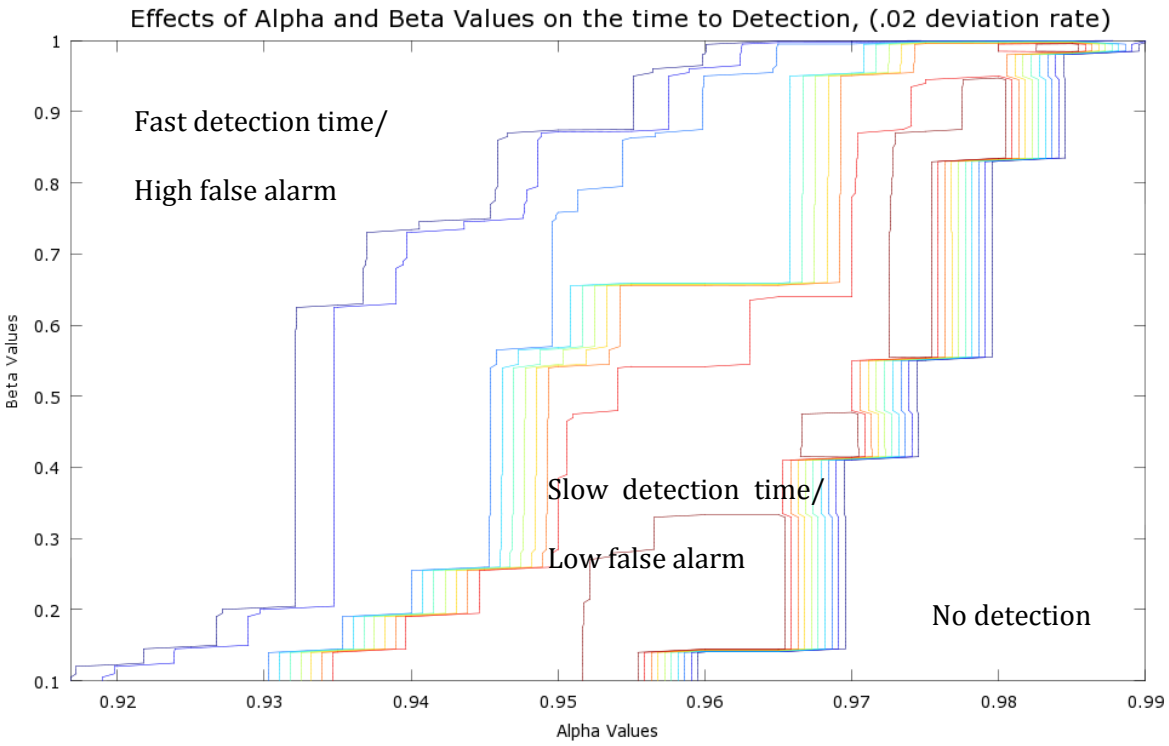

Figure above depicts the contour plot of the detection time of a .02 m/s deviation rate with respect to varying values of the β and α values. The red contour shows the highest detection time while the dark blue contours depict relatively lower detection times. While it may be favorable to attain lower detection times, setting a lower α and higher β increases the false alarm rates.

SECTION 5: DISCUSSION

Section 5 discusses the potential impact of applying such a method described in this research effort. Section 5.1 addresses the reversal of the conflict asymmetry between adversaries and defenders. Section 5.2 argues how the increase in components from the 2 INS and 2 GPS architecture only increases the benefits of diverse, redundant components; it does not improve the latency of the detection methods. Section 5.3 discusses some of the parameters in the method that is left for the operators to tweak. Section 5.4 discusses the vulnerability within Analysis 1, similarity between INS1 and INS2; however, it also discusses the bounded impact of the vulnerability.

5.1 REVERSING ASYMMETRICAL CONFLICT

The method proposed in this section reverses the asymmetrical conflict innate between the defender and the adversary. In order for an adversary to successfully inject false information, she must be able to infect multiple supply chains. An adversary must also be able to successfully initiate an attack on the majority of sensors simultaneously which creates a Byzantine General problem for the adversary.

This body of work addresses the problem posed by Kwon et al [2], that is, an attacker may apply a sequence of false injections to deviate an aircraft away from its planned flight path without alarming fault detection techniques by extending System-Aware portfolio of design patterns to defend sensors. Using multiple reference signals via diverse redundant components, the method checks the consistency of component outputs and isolates a faulty component if the detection method detects disagreements between the measurement output between two components.

The method developed increases the cost of false injection by meeting the three objectives listed in Section 1.3:

1. Objective 1: Detect stealthy false-injection attacks against sensor components
2. Objective 2: Isolate infected sensor components
3. Objective 3: Limit impact of such cyber-attacks

Section 4.3 and 4.4 shows that the detection method is successful in the detection of persistent attacks against navigation system. A logical decision tree concludes that a component is deviating allowing us to isolate an infected component. Section 4.6 shows detection latency for various deviation rates. Thus, the method increase the difficulty of an adversary to successfully attack sensor components of UAVs.

5.3 Increasing Number of Components

Increasing the number of components would improve the defense of the UAV navigation system in exchange for a more complex decision tree. If, perhaps, an adversary attacks 2 components in a 2-INS/2-GPS navigation system (excluding GPS spoofing), then the adversary may obfuscate the results of the logical decision tree. Using an additional navigation component, the operator must include an additional similarity measurement analysis. An additional navigation component forces the adversary to exploit an additional component. Although an addition component takes advantage of diverse redundancy, an additional component does not shorten the latency of attack.

5.4 Vulnerability

Choosing a high threshold as the compound scalar testing threshold can be a vulnerability for Analysis 1 (INS similarity). Since the method only measures similarity of the acceleration component of each INS, the adversary could hide injections under the noise of

the INSs applying a minute bias away from the true position of the aircraft. Using the INSs noise magnitude of .05 and .07m/s^2 and a confidence interval of .99 for the compound scalar test for Analysis 1, The maximum deviation an adversary can apply to the INS is

$$norminv(.99,0,\Sigma_{r_1}) = .0174m/s^2$$

the effect of which is depicted in Figure 36. The impact of the adversary is thus limited to an absolute deviation of about 30m for a flight duration of 2000sec (about 33min).

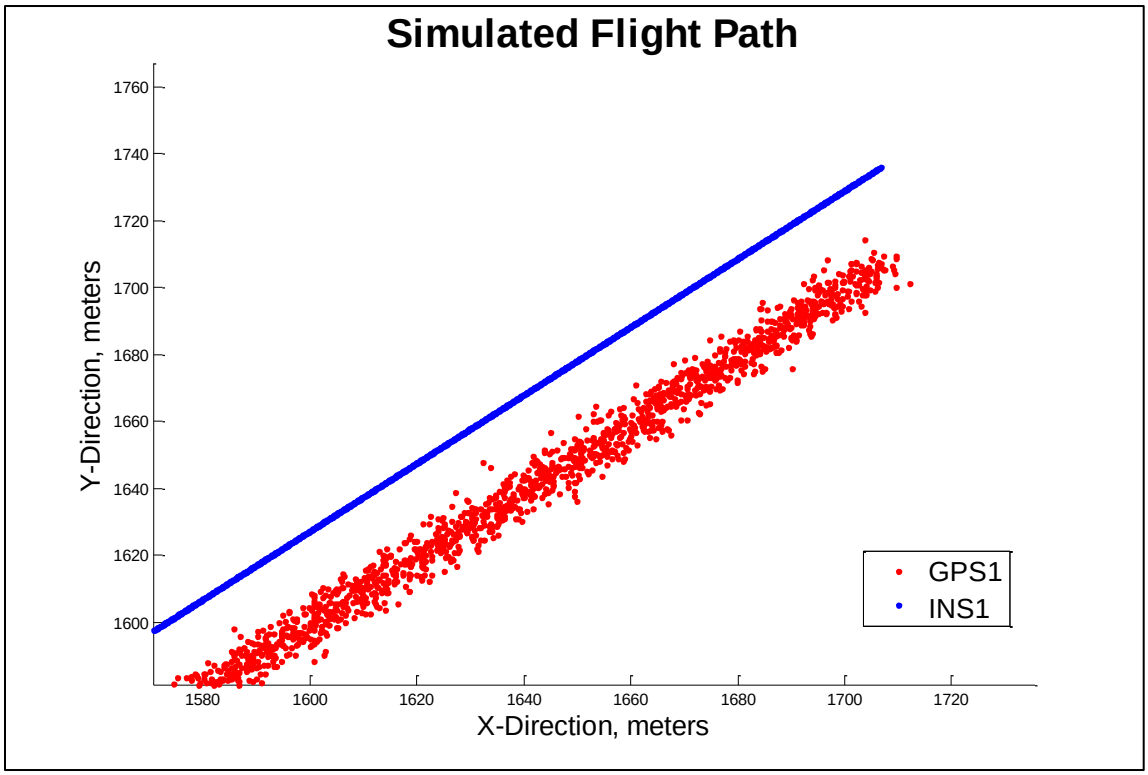

Figure 36: The impact of a .017 m/s^2 deviation applied to INS1. The maximum deviation in 2000sec (about 33min) is 30m.

5.5 Future Work

The next step in the development of the detection method is to test the method in an aircraft environment.

Other similarity measurements can also be developed to improve upon the capabilities of the current method. Some that could be considered is developing belief functions and detection based on Dempster-Shafer Theory [25]. Detection methods based on a priori models depicting different types of attacks can also be considered.

APPENDIX A: SIMULATION SCRIPT

StatisticalbasedIntrusionDetection.m

```
clear; clc; close all;


flightduration = 500;
dt = .1;                % sampling time
attack_rate = [0 0 0 0]*dt;

falsealarm1 = [];
falsealarm2 = [];
falsealarm3 = [];
falsealarm4 = [];

for sim = 1:1
RES1 = [];
RES2 = [];
RES3 = [];
RES4 = [];

%% Define update equations (Coefficent matrices): A physics
based model for where we expect the Aircraft to be [state
transition (state + velocity)] + [input control
(acceleration)]
A = [1 dt 0 0; 0 1 0 0; 0 0 1 dt; 0 0 0 1] ;% state
transition matrix:  expected flight of the Aircraft (state
prediction)
B = [dt^2/2 0; dt 0; 0 dt^2/2; 0 dt];       % input control
matrix:  expected effect of the input accceleration on the
state.
C = [1 0 0 0; 0 0 1 0];                     % measurement
matrix: the expected measurement given the predicted state
(likelihood)
dT = [(dt^2/2)*randn; dt;  (dt^2/2)*randn; dt];


%% define main variables
INS1= [0; 0 ; 0; 0];                        %initized state-
-it has two components: [position; velocity] of the
Aircraft
INS2 = [0; 0 ; 0; 0];
INS1input = [];
INS2input = [];
GPS1 = [0; 0];
```

```
GPS2 = [0; 0];
x_real = [0; 0; 0; 0];
INS1GPS1 = [0; 0; 0; 0];


UAVAccel_noise = [.05 .07];                %INS process
noise(stdv of acceleration: meters/sec^2)
GPS_noise_mag = [3 4];                       %GPS measurement
noise

%% Kalman Parameters
PINS1GPS1 = [0.1628    0.0149         0         0,
    0.0149    0.0027         0         0,
         0         0    0.1628    0.0149,
         0         0    0.0149    0.0027];
PINS1GPS2 = [0.2510    0.0198         0         0
    0.0198    0.0031         0         0
         0         0    0.2510    0.0198
         0         0    0.0198    0.0031];

KINS1GPS1 = PINS1GPS1*C'*inv(C*PINS1GPS1*C'+ 3*eye(2,2));

Sig_RES1 =
(UAVAccel_noise(1)^2+UAVAccel_noise(2)^2)*eye(2,2);
Sig_RES2 =
(GPS_noise_mag(1)^2+GPS_noise_mag(2)^2)*eye(2,2);
Sig_RES3 = C*PINS1GPS1*C'+GPS_noise_mag(1)^2*eye(2,2);
Sig_RES4 = C*PINS1GPS2*C'+GPS_noise_mag(2)^2*eye(2,2);
%% Calculate Steady-State Parameters



target = [10000 10000]';
time_to_attack = 250;
i = 0;
j=0;
for t = 1  : dt: flightduration

    if (t>=time_to_attack)
        i=1+i;
    end

    if (t>=time_to_attack)
        j=1;
    end
    u = 10*[target-INS1GPS1(1:2:3,end)]/...
        (norm([target-INS1GPS1(1:2:3,end)]));
```

```
    % Generate true Aircraft data
    x_real = [x_real A * x_real(:,end) + B * u];
    if (x_real(2,end) > 10)
        x_real(2,end) = 10;
    end
    if (x_real(4,end) > 10)
        x_real(4,end) = 10;
    end
    INS1input = [INS1input (B * u + [0 0 attack_rate(1)*j
0]' +...
        UAVAccel_noise(1) * [(dt^2/2)*randn; 0;
(dt^2/2)*randn; 0])];
    INS2input = [INS2input (B * u + [0 0 attack_rate(2)*j
0]' + UAVAccel_noise(2) * ...
        [(dt^2/2)*randn; 0;  (dt^2/2)*randn; 0])];
    INS1 = [INS1 (A * INS1(:,end) + INS1input(:,end))];
    INS2 = [INS2 (A * INS2(:,end) + INS2input(:,end))];

    if (INS1(2,end) > 10)
        INS1(2,end) = 10;
    end

    if (INS1(4,end) > 10)
        INS1(4,end) = 10;
    end

    if (INS2(2,end) > 10)
        INS2(2,end) = 10;
    end
    if (INS2(4,end) > 10)
        INS2(4,end) = 10;
    end


    % Filter GPS1 and INS1
    GPS1 = [GPS1 (C * x_real(:,end) + [0 attack_rate(3)*i]'
+ [GPS_noise_mag(1)*randn GPS_noise_mag(1)*randn]')];
    GPS2 = [GPS2 (C * x_real(:,end) + [0 attack_rate(4)*i]'
+ [GPS_noise_mag(2)*randn GPS_noise_mag(2)*randn]')];
    INS1GPS1 = [INS1GPS1 (INS1(:,end) + KINS1GPS1 *
(GPS1(:,end) - C * INS1(:,end)))];

    if (INS1GPS1(2,end) > 10)
        INS1GPS1(2,end) = 10;
    end
    if (INS1GPS1(4,end) > 10)
```

```
        INS1GPS1(4,end) = 10;
    end
%% Generate Array of Standardized Faults

    RES1 = [RES1 (INS1input(1:2:4,end)-
INS2input(1:2:4,end))'*inv(((dt^2/2)*eye(2,2))'*Sig_RES1*(d
t^2/2)*eye(2,2))*(INS1input(1:2:4,end)-
INS2input(1:2:4,end))];
    RES2 = [RES2 (GPS1(:,end)-
GPS2(:,end))'*inv(Sig_RES2)*(GPS1(:,end)-GPS2(:,end))];
    RES3 = [RES3 (INS1(1:2:4,end)-
GPS1(:,end))'*inv(Sig_RES3)*(INS1(1:2:4,end)-GPS1(:,end))];
    RES4 = [RES4 (INS1(1:2:4,end)-
GPS2(:,end))'*inv(Sig_RES4)*(INS1(1:2:4,end)-GPS2(:,end))];


end

figure(1); clf; hold on;
plot(GPS2(1,:),GPS2(2,:),'k.');
plot(GPS1(1,:),GPS1(2,:),'r.');
plot(INS1(1,:),INS1(3,:),'b.');
% plot(INS1GPS1(1,:),INS1GPS1(3,:),'y.');
% plot(INS2(1,:),INS2(3,:),'g.');
xlabel('X-Direction, meters','FontSize',16);
ylabel('Y-Direction, meters','FontSize',16);
title('Simulated Flight
Path','FontWeight','bold','FontSize',24);
legend1 = legend('GPS2','GPS1','INS1');
set(legend1,'Position',[0.754221732745963 0.187739463601532
0.128854625550661 0.0985221674876847],...
    'FontSize',16);


figure(2); clf; hold on;
plot(RES1,'r-');hold on;
plot(RES2,'k-');hold on;
plot(RES3,'b-');hold on;
plot(RES4,'g-');

%% Analysis 1: Similarity Measurement between 2 INS
% Analysis 1 calls for residual analysis of the 2 INS
inputs
    A1 = chi2cdf(RES1,2);
    figure(3); clf; plot(A1);

    countA1 = 0;
```

```
    attacktimeA1 = zeros(1,length(A1));
    allgood1=0;
    for i=1:length(A1)
        if A1(i) >= .99
            countA1 = countA1 + 1;
            attacktimeA1(i) = 1;
        end
        if A1(i) <= .99
            allgood1=i;
        end
    end

    falsealarm1 = [falsealarm1 countA1];
%% Analysis 2: Similarity Measurement between 2 GPS
%Analysis 1 calls for residual analysis of the 2 INS inputs
    A2 = chi2cdf(RES2,2);
    figure(4); clf; plot(A2);

    countA2 = 0;
    attacktimeA2 = zeros(1,length(A2));
    allgood2=0;
    for i=1:length(A2)
        if A2(i) >= .99
            countA2 = countA2 + 1;
            attacktimeA2(i) = 1;
        end
        if A2(i) <= .99
            allgood2=i;
        end
    end

    falsealarm2 = [falsealarm2 countA2];

%% Analysis 3: Similarity Measurement between INS1 and GPS1
%Analysis 3 calls for spectral analysis between INS1 and
GPS1

    A3 = chi2cdf(RES3,2);
    figure(5); clf; plot(A3);

    countA3 = 0;
    attacktimeA3 = zeros(1,length(A2));
    allgood3=0;
    for i=1:length(A3)
        if A3(i) >= .99
            countA3 = countA3 + 1;
            attacktimeA3(i) = 1;
```

```
        end
        if A3(i) <= .99
            allgood3=i;
        end
    end

    falsealarm3 = [falsealarm3 countA3];


end

%mean(falsealarm1)
mean(falsealarm2)
% mean(falsealarm3)
% mean(falsealarm4)


figure(7); clf; hold on;
plot(attacktimeA1,'r.');
title('Analysis1: Similarity, INS1 and
INS2','FontWeight','bold',...
    'FontSize',24);
ylabel(['Disagreement Signals',sprintf('\n'),'(0=Agree,
1=Disagree)'],...
    'FontSize',16);
xlabel('time(k), deciseconds','FontSize',16);

figure(8); clf; hold on;
plot(attacktimeA2,'r.');
title('Analysis2: Similarity, GPS1 and
GPS2','FontWeight','bold',...
    'FontSize',24);
ylabel(['Disagreement Signals',sprintf('\n'),'(0=Agree,
1=Disagree)'],...
    'FontSize',16);
xlabel('time(k), deciseconds','FontSize',16);

figure(9); clf; hold on;
plot(attacktimeA3,'r.');
title('Analysis3: Similarity, GPS1 and
INS1','FontWeight','bold',...
    'FontSize',24);
ylabel(['Disagreement Signals',sprintf('\n'),'(0=Agree,
1=Disagree)'],...
    'FontSize',16);
xlabel('time(k), deciseconds','FontSize',16);
```

REFERENCES

[1] "2012 Threats Predictions," McAfee Labs, 2012.

[2] C. Kwon, W. Liu and I. Hwang, "Security Analysis for Cyber-Physical Systems against Stealthy Detection Attacks," in *American Control Conference, 2013*, Washington, DC, 2013.

[3] A. A. Cardenas, S. Amin and Z. Lin, "Attacks Against Process Control Systems: Risk Assessment, Detection, And Response," *ASIACCS,* 2011.

[4] M. Rahman and H. Mohsenian-Rad, "False data injection attacks with incomplete information against smart power grids," in *Global Communications Conference (GLOBECOM), IEEE* , 2012.

[5] J. Gertler, "Survey of model-based failure detection and isolation in complex plants," *IFAC Proceedings Series,* no. 7, 1987.

[6] A. Cardenas, S. Amin and S. Sastry, "Research Challenges for the security of Control Systems," *3rd USENIZ workshop on Hot Topics in Security,* 2008.

[7] R. Jones and B. Horowitz, "System-Aware Cybersecurity," *2011 Eighth International Conference on Information Technology: New Generations,* pp. 914 - 917, 2011.

[8] S. Amin, X. Litrico, S. Sastry and A. M. Mayen, "Cyber Security of Water SCADA Systems: (I) Analysis and Experimentation of Stealthy Deception Attacks," *Control Systems Technology, IEEE Transactions on,* vol. PP, no. 99, p. 1, 2012.

[9] R. Langner, "Stuxnet: Disecting a Cyber Weapon," *Security & Privacy, IEEE,* vol. 9, no. 3, pp. 49-51, 2011.

[10] W. Sohne, O. Heinze and E. Groten, "Integrated INS/GPS System for High Precision

Navigation Applications," in *Position Location and Navigation Symposium, IEEE* , 1994.

[11] D. W. J. Titterton, "Strapdown Inertial Navigation Technology," in *AIAA*, 2004.

[12] A. Kim, B. Wampler, J. Goppert, I. Hwang and H. Aldridge, "Cyber Attack Vulnerabilities Analysis for Unmanned Aerial Vehicles," American Institute or Aeronautics and Astronautics.

[13] S. M. Giray, "Antomy of Umanned Aerial Vehicle Hijacking with Signal Spoofing," in *Recent Advances in Space Technologies (RAST), 2013 6th International Conference on* , Istanbul, Turkey, 2013.

[14] D. Gregg, W. Blackert, D. Heinbuch and D. Furnanage, "Assessing and quantifying denial of service attacks," in *Military Communications Conference, 2001. MILCOM 2001. Communications for Network-Centric Operations: Creating the Information Force, IEEE* , 2001.

[15] X. Yang, T. Bin, L. Qi, Z. Jian-yi and H. Zheng-Ming, "Networks, A Novel Framework of Defense System Against DoS Attacks in Wireless Sensor," in *Wireless Communications, Networking and Mobile Computing (WiCOM), 2011 7th International Conference on*, 2011.

[16] R. Rejimol Robinson and C. Thomas, "Evaluation of mitigation methods for distributed denial of service attacks," in *Industrial Electronics and Applications (ICIEA), 2012 7th IEEE Conference on* , 2012.

[17] D. Gregg, W. Blackert, D. Heinbuch and D. Furnanage, "Assessing and quantifying denial of service attacks," in *Military Communications Conference, 2001. MILCOM 2001. Communications for Network-Centric Operations: Creating the Information Force. IEEE* , 2001.

[18] Y. Mo and B. Sinopoli, "False data injection attacks in control systems," *Prepreints of*

the First Workshop on Secure Control Systems, 2010.

[19] J. Rrushi, "Composite intrusion Detection in Process Control Networks," Universita Degli Studi Di Milano, 2009.

[20] S. Amin, X. Litrico, S. S. Sastry and A. M. Bayen, "Cyber Security of Water SCADA Systems-Part II: Attack Detection Using Enhanced Hydrodunamic Models," *IEEE Transactions on Control Systems Technology,* vol. PP, no. 99, pp. 1-15, 2012.

[21] R. A. Jones, T. V. Nguyen and B. M. Horowitz, "System-Aware Security for Nuclear Power Systems," *Technologies for Homeland Security (HST), 2011 IEEE International Conference on,* pp. 224 - 229, 2011.

[22] G. L. Babineau, R. A. Jones and B. Horowitz, "A system-aware cyber security method for shipboard control systems with a method described to evaluate cyber security solutions," *Homeland Security (HST), 2012 IEEE Conference on Technologies for,* pp. 99 - 104, 2012.

[23] K. Hartmann and C. Steup, "The vulnerability of UAVs to cyber attacks - An approach to the risk assessment," in *Cyber Conflict (CyCon), 2013 5th International Conference on* , 2013.

[24] B. Littlewood and L. Strigini, "Redundancy and Diversity in Security," *Lecture Notes in Computer Science,* vol. 3193, pp. 423-438, 2004.

[25] G. Shafer, A Mathematical Theory of Evidence, Princeton: Princeton University Press, 1976.

[26] R. Baheti and H. Gill, "Cyber-physical Systems," *The Impact of Control Technology,* 2011.

[27] B. Kerbs, "Cyber Incedent Blamed for Nuclear Power Plant Shutdown," *Washington*

Post, June 2008.

[28] O. Kosut, L. Jia, R. J. Thomas and a. L. Tong, "On malicious data attacks on power system estate estimation," *In UPEC,* 2010.

[29] M. Blanke, M. Kinnaert, J. Lunze and M. Staroswiecki, Diagnosis and Fault-Tolerant Control, Springer-Verlag, 2006.

[30] S. Cha, "Comprehensive Survey on Distance/Similarity Measures between Probability Desnity Functions," *International Journal of Mathematical Models and Methods in Applied Sciences,* vol. 1, no. 4, pp. 300-307, 2007.

[31] A. Sharma, L. Golubchik and R. Covindan, "On the Prevalence of Sensor Faults in Real-Wold Depolyments," in *SECON*, 2007.

[32] G. Tolle, J. Polastre, R. Szewczyk, D. Culler, N. Turner, K. Tu, S. Burgess, T. Dawson, P. Duonadonna, D. Gay and W. Hong, "A macroscope in the Redwoods," in *Proceedings of the 2nd international conference on Embedded netowrked sensor systems*, 2005.

[33] N. Ramanathan, L. Balzano, M. Burt, D. Estrin, E. Kohler, T. Harmon, C. Harvey, J. Jay, S. Rothenberg and M. Srivastava, "Rapid Deployment with Confidence: Calibration and Fault Detection in Environmental Sensor Networks," in *CENS*, 2006.

[34] H. Hu, L. Xu and Y. Zhan, "The Study on Position Error Correction of C/A Code GPS Receiver," in *Knowledge Acquisition and Modeling Workshop, 2008. KAM Workshop 2008. IEEE International Symposium on* , 2008.