TEXT and rectangles in blue will NOT show on printed copy

Type final title of thesis or dissertation (M.S. and Ph.D.) below . If your title has changed since your submitted an Application for Graduate Degree, notify Graduate Office.

Initial Application of Fault Detection and Parameter Estimation Techniques to Cybersecurity Intrusion Detection

---

A Thesis

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

by

Name

Louis William DiValentin

Month degree is awarded December

Year 2013

APPROVAL SHEET

The thesis

is submitted in partial fulfillment of the requirements

for the degree of

Master of Science

AUTHOR *signature*

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

Please insert committee member names below:

Barry Horowitz

Advisor

Randy Cogill

Greg Lewin

Accepted for the School of Engineering and Applied Science:

Dean, School of Engineering and Applied Science

Month degree is awarded December

Year 2013

Print Form

INITIAL APPLICATION OF FAULT DETECTION AND PARAMETER ESTIMATION TECHNIQUES TO CYBERSECURITY INTRUSION DETECTION

A Thesis

Presented to

The Faculty of the School of Engineering and Applied Science

University of Virginia

Examination Committee:

Dr. Barry Horowitz (Advisor)

Dr. Randy Cogill (Committee Chair)

Dr. Greg Lewin (Committee Member)

In Partial Fulfillment

Of the Requirements for the Degree

Master of Science

in System Engineering

by

Louis DiValentin

June 2013

`

ABSTRACT

Industrial control systems recently have become the targets of cyber-attacks that manipulate the parameters of their normal operating procedures to produce unstable behavior. Previous research has shown that security solutions embedded within the system being protected can provide a method for cyber-attack detection. Fault detection, specifically system identification, can offer multiple methods of detection of deviations from set system parameters in a dynamic model representation of the industrial control system using the measurements obtained and the inputs specified during operation. In particular, this research effort uses different system identification techniques to determine if a system is operating as designed and configured. During the investigation for this Thesis a detection algorithm was created that monitors a system by comparing real time estimates of the dynamic model of the system with the known designed system dynamic model. When sufficient deviations between the estimated dynamic model and the known dynamic model are judged by the similarity algorithms, the detection algorithm informs system operators of the possible existence of an attack. The operators of the systems then use a series of guidelines created in this Thesis that examines the conditions and the situational disutility surrounding the event to help determine the likelihood of a cyber-attack versus a hardware or software failure. This Thesis will compare multiple existing systems identification techniques to determine how effective the selected techniques are at detecting cyber-attacks, with the criteria of success being the true positive rates, the false alarm rates, and the detection time.

ABSTRACT 1
1. PROBLEM STATEMENT 4
1.1 Motivation 4
1.2 Problem Definition 5
1.3 Proposed Solution 6
2. LITERATURE REVIEW 8
2.1 Previous Attempts at Problem Solutions 8
2.2 Fault Detection using Systems Identification 9
2.3 Background 10
2.3.1 Mathematical Representation of the Dynamic System 10
2.3.2 Kalman Filtering for Estimate Smoothing 12
2.3.3 Parameter Estimation 12
3. APPROACH 16
3.1 RLC Filter system 16
3.2 Description of the Fuel Injection System 18
3.3 Attack Detection Methodology 22
3.3.1 Simulation Setup 23
3.3.2 Similarity Algorithms 24
3.3.2.1 Scaled Sum of Squares Method. 25
3.3.2.2 Bayesian Update Heuristic 25
3.3.2.3 Binomial Method 26
3.3.3 Event detection 26
3.4 Independent Trials 27
5. RESULTS 31
5.1 RLC Filter 31
5.1.1 Effect of Percentage of Deviations (POD) allowed 31
5.1.2 Effect of Process Noise 36
5.1.3 Effect of Measurement Noise 39
5.1.4 Worst Case Scenario 42
5.1.5 Overall Statistics 46
5.2 Fuel Injection System 50
5.2.1 Effect of Percentage of Deviations (POD) allowed 50
5.2.2 Effect of Process Noise 54

5.2.3 Effect of Measurement Noise 56
5.2.4 Worst Case Scenario 59
5.2.5 Overall Statistics 62
6. DISCUSSION 66
6.1 Results Discussion 66
6.2 Implementation Considerations 67
6.2.1 Assumptions 68
6.2.2 Cyber-Attack Isolation 69
6.3 Potential Improvements to the Detection Algorithm 69
7. EVENT CLASSIFICATION 71
7.1 Disutility of Events 71
7.2 Situational Context 72
7.3 Human Error 72
7.4 Previous Events 73
7.5 Cyber-attack Checklist 73
8. CONCLUSION 76
9. REFERENCES 77
10. APPENDICES 80
10.1 Simulation Parameters 80
10.1.1 RLC filter 80
10.1.2 Fuel Injection system 81
10.2 Performance Charts 82
10.2.1 RLC ROC charts. 82
10.2.2 RLC False Alarm rate vs Detection Time Charts. 92
10.2.3 Fuel Injection ROC charts 102
10.2.3 Fuel Injection False Alarm rate vs. Detection Time charts. 115

1. PROBLEM STATEMENT

1.1 Motivation

In the past, typical cyber-attacks have focused on stealing information and data, denying access to services, or defacing and damaging online applications. Recently some cyber-attacks have attempted to manipulate the control systems of infrastructure with the intent of damaging or hampering the operation of the physical systems, a previously unseen motive. The Stuxnet and Maroochy attacks are openly documented examples [1][2][3]. In the Stuxnet attack, a computer worm compromised the Supervisory Control and Data Acquisition (SCADA) of multiple Iranian centrifuges and caused them to spin at damaging speeds. The worm accomplished this by manipulating computer controlled parameters that maintained the rotation rate of the motors in the centrifuges for short periods of time. These manipulations sabotaged the normal operating conditions of the centrifuges and caused them to fail more quickly [1]. In the Maroochy Shire attack, an insider manipulated the controller of multiple sewage pumping stations for a waste management system and caused approximately a million liters of sewage to be released into the waterways. The pumps were centrally monitored and controlled, allowing the insider to manipulate the pumps at will and to disable any alarms that were triggered; his combined access to both systems allowed him to remain undetected for a long period of time [3][22]. Both of these cyber-attacks show that perimeter security on critical control systems no longer offers enough protection from damage from motivated attackers. The fallibilities of perimeter security have necessitated the addition of layers of security that are embedded within the systems being protected [5] [20]. Furthermore, insider and supply chain attacks are initiated within an attacked system, calling for additional security inside the protected system.

1.2 Problem Definition

The advent of cyber-attacks on control systems requires new methods of perimeter security and cyber-attack detection to protect critical systems. Because of the impossibility of providing completely comprehensive perimeter security, cyber-attack detection becomes essential to maintaining the normal operating conditions of industrial control systems. Many industrial control systems use bounds in the form of control charts on states of operation to provide a form of cyber-attack or fault detection [22]. Sensors monitor specific states of the physical system to determine if the values of the states exceed an upper and lower threshold. If an value exceeds the threshold (from here on referred to as an event), then the operator is notified and appropriate actions are taken. However, if a cyber-attack can change the parameters of the dynamic system then it is possible that the attack can disrupt the system without causing the states of the system to exceed their bounded values. Stuxnet is an example of an attack that changed the parameters of the dynamic system and avoided event detection for a long period of time. In the case of Stuxnet, the worm caused the system to operate at non optimal levels and to fail at a quicker rate then designed. This thesis attempts to detect cyber-attacks that follow the same attack pattern (i.e. for long periods of time the cyber-attack remains undetected, but the cyber-attack continually impacts the performance of the physical system). For attacks that avoid detection in this way, the underlying structure of the controller and the physical dynamic system can be used to provide additional information regarding the likelihood of a cyber-attack. Additionally, once an event has been flagged the event must be categorized as a human error, a fault, a cyber-attack, or some other degradation of the system. This will allow the operator to take the appropriate actions to rectify the disruption to the system.

1.3 Proposed Solution

This research project uses Fault Detection techniques, specifically Systems Identification and Parameter Estimation, to detect when cyber-attacks that modify a dynamic system's parameter values occur. Parameter Estimation algorithms offer multiple methods for the detection of deviations from designed and configured system parameters in the mathematical representation of the industrial control system. To detect cyber-attacks that modify system components, a monitoring system will compare a real time estimate of the dynamic model of a system created using recent measurement and input values with the known designed system dynamic model. When there are sufficient discrepancies between the estimated dynamic model and the known dynamic model, the monitoring system will inform system operators of the possible existence of an attack. The operators of the systems will then use the conditions and the situational disutility surrounding the event to determine the likelihood of a cyber-attack versus a hardware or software failure and, if possible, will detect the source of deviations in the parameters and restore the system to its designed operation mode. Specifically, if a cyber-attack changes one of the components in the physical system and that change is reflected in the real time estimate of the parameters of linear dynamical system, then the physical system will begin to differ from the mathematical representation of the system used to control the physical system. This research project uses existing Systems Identification techniques that create estimates of linear dynamic systems using the measurements of the inputs and outputs to extract the current parameters of the physical system. It then compares the real time estimate of the parameters of the mathematical model (specifically the A matrix) to the parameters used by the current mathematical representation of the system. The different estimation methods and similarity algorithms will be evaluated by their:

- False Alarm Rate
- Missed Detection Rate
- Detection time
- Inherent Limitations

An estimation method is considered successful if it correctly detects changes in the A matrix of the linear dynamic system within a predetermined sample time period with set success rates and set false alarm rates determined by the application. Additionally, the deviation detection algorithms are tested against the same criteria. This research project also develops criteria using the conditions and the situational disutility surrounding the event to determine the likelihood of a cyber-attack as opposed to alternative causes for faults like hardware or software failures.

2. LITERATURE REVIEW

2.1 Previous Attempts at Problem Solutions

The resilient systems and cyber security disciplines each have attempted to detect attacks similar to Stuxnet in different ways. The resilient systems community has adapted fault detection techniques to detect unplanned changes in the operation of critical systems. [4] defined and simulated three types of cyberattacks; surge, bias and geometric. A surge attack on a control system attempt to inflict the most possible damage the instant the system is compromised. Bias attacks modify the control system to produce less then optimal output while remaining undetected for the longest amount of time possible. Geometric attacks combine surge and bias attacks; geometric attacks try to remain undetected until the system becomes vulnerable and then attempt to cause the maximum amount of damage. Bias attacks are the hardest of the attacks to detect, but geometric attacks typically cause the most damage; Stuxnet was classified as a bias attack [4]. It is assumed that if a bias attack on a control system is detectable, then a geometric or surge attack on the same system will also be detectable. For this reason the detection of bias attacks will be the focus of this Thesis. A method for detecting 'man in the middle' attacks on sensors reporting measurements is discussed in [5]. The authors used a set of states not directly shown to the operator to predict the value of the displayed states which could then be used as a form of validation and manipulation detection. Similar to this Thesis, the authors used the structure of a mathematical model representation of the physical system to gather information about the possibility of a cyber-attack on one of the displayed states.

The cyber security community has taken a different track attempting to detect the presence of attacks similar to Stuxnet. They have decided, instead, to monitor the computers that provide the control for the dynamic systems to detect aberrations. Power Fingerprinting characterizes the voltage draws for the processor in clean systems for various system processes and detected anomalies in characterized and actual power draws [6]. HyperCheck uses external cloud computing processors to independently validate

the integrity of core operation system files [7]. Behavior modeling monitors the series of low-level function calls of various vulnerable programs and compares them to known good series of function calls. A weakness of these cyber security approaches is the assumption that a computer that has been compromised will still be in a state that will allow it to monitor itself. The systems identification approach can be run independently of the control system and requires only the measurements and inputs from the physical system. A design issue for this approach will be obtaining and trusting the measurements and inputs of the physical system through sensors; however, using the method described in [5] can provide a method for validating the measurements.

2.2 Fault Detection using Systems Identification

Fault detection is the procedure for identifying failing or failed components in a physical system in a timely manner, so the system can either be restored or be taken offline to prevent further damage to the system in question. Cyber-attacks on the software controllers of physical systems can cause performance degradations in a physical system in a manner very similar to a hardware failure event, usually by adjusting the configuration of hardware devices to create faults. This Thesis hypothesized that fault detection techniques will translate to the detection of cyber-attacks on control systems because of the similar disruptions in service that occur. However, much of fault detection is centered on hard failures; whereas cyber security is focused on software induced failures. Cyber-attack patterns are limited to parameters that can be computer controlled.

Systems Identification has been used as a method of fault detection to determine when performance of the physical system has degraded to unacceptable levels. Systems Identification is used in [9] to classify the states of the physical system during the failure of three modules on a jet engine and determine from those states the failed module or modules. System Identification also is used in [10] to detect perturbations introduced into in the spin rate of a DC motor by estimating the parameters of the mathematical model at each time step. Similarly [11] uses Systems Identification to determine the

parameters of a DC motor in six different fault states in continuous time using a block pulse function. [30] proposes using the balanced realization algorithm, a Systems Identification technique, to detect faults in systems. This Thesis uses Systems Identification to identify cyber-attacks by comparing the real time estimate of the dynamic system to the mathematical model used to control and simulate the system.

2.3 Background

2.3.1 Mathematical Representation of the Dynamic System

Industrial Control Systems are typically represented by multiple mathematical models in discrete time. They are constructed to be piecewise functions of the current operating state that change over time as the physical system changes. Assuming errors are zero mean, Gaussian, and stationary, a linear dynamic system can be represented by the following discrete-time mathematical equations:

$$x_{k+1} = A * x_k + B * u_k + w_k$$

$$y_k = C * x_k + D * u_k + v_k$$

where x is a $nx1$ vector of the states of the system at time step k, A is nxn matrix of parameter values, B is nxm matrix of parameter values, u is a $mx1$ vector of input values, y is a $px1$ vector of measurement values at time step k, C is a pxn matrix of parameter values, D is a px1 matrix of parameter values, w_k is the $nx1$ process error of the system at time step k, and v_k is the $px1$ measurement error of the system at time step k. n refers to the number of states of the system, m refers to the number of inputs in the system, and p refers to the number of measurements taken from the system. The parameter values in A, B, and C are combinations of the component values in the system. In some cases it is possible to create a set of linear equations that can extract the individual component values. When C is a full rank square matrix the values of all the states can be approximated at each time step by calculating $C^{-1}y_k$ under the assumption that v_k is zero mean, Gaussian, and stationary. When C is less than full rank, alternative methods must be used to obtain the values of the states at each time step.

Another important construct used from the field of control theory is the minimum state companion form representation of a state space. Given an *A*, *B*, *C* and *D* matrix, the transfer function of a Single Input and Single Output (SISO) state space can be calculated with the formula:

$$H(s) = C * (s * I - A)^{-1} * B + D = \frac{b_0 * s^n + b_1 * s^{n-1} + \cdots b_n}{s^n + a_1 * s^{n-1} + \cdots a_n}$$

Assuming the transfer function is in its minimal state (any potential reductions in the order have been factored out) then a specific realization of the state space can be constructed shown below:

$$x_{k+1} = \begin{bmatrix} -a_1 & -a_2 & \dots & -a_{n-1} & -a_n \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \dots & \dots & \cdots & 0 & 0 \\ 0 & 0 & \dots & 1 & 0 \end{bmatrix} * x_k + \begin{bmatrix} 1 \\ 0 \\ \dots \\ 0 \\ 0 \end{bmatrix} * u_k + w_k$$

$$y_k = [b_1 \quad b_2 \quad \dots \quad b_n] * x_k + b_0 * u_k + v_k$$

This transformation can be extended to Multiple Input and Multiple Output (MIMO) systems. The main advantage of this form is the ability to compare realizations of different transfer functions easily. Because there are infinite numbers of state space realization for the same transfer function, real time estimates created may have the same underlying transfer function, but look completely different in their realizations. Putting all the estimates in a similar form allows the similarity algorithms to compare the real time estimates much quicker and efficiently. Additionally, realizing the transfer function in companion form reduces the number of variable parameters in the state matrix A. A downside to realizing the matrix in companion form is the states created in this realization will be intangible states and not linked directly to physical processes which will make interpretation more difficult. The measurements of the system however will remain the same no matter what transformation is made. Comparing the estimates in companion form also makes fault isolation more difficult as it is very difficult to retrieve the previous realization of the system which contains useful information about the combinations of components in its parameters.

2.3.2 Kalman Filtering for Estimate Smoothing

Kalman filtering is an approach that is often used to mitigate the effects of process and measurement error on the stability of the mathematical model that is being used to predict the control inputs for the physical system. The process of Kalman filtering uses the mathematical representation of the system defined in 2.3.1, and is shown in [13] and [17]. Many of the parameter estimation techniques that are applied perform better in systems of low noise, so mathematical models of systems are often smoothed using a Kalman filter to reduce the amount of noise inherent in the physical system. Additionally, Kalman filtering provides a method of real time estimation of parameters discussed in Section 2.3.3.2.

2.3.3 Parameter Estimation

Parameter Estimation is a branch of Systems Identification used for determining the optimum parameter values for a mathematical model that represents the physical system in question. Typically, this is accomplished by minimizing either the variance or the mean squared errors between the predicted measurements and inputs and the actual measurements and inputs for the selected model for a specific time period. Multiple potential methods of Parameter Estimation are laid out in this section. Linear Least Squares Estimation and Kalman Filter Parameter Estimation require all states of the mathematical model to be directly extractable from the measurements and inputs, while Subspace System Identification and Expectation-Maximization only require the measurements, inputs, and number of states to fit mathematical models. Extractable is defined as either the values of all the states are all measured directly using the C matrix (The C matrix would be represented as an identity matrix) or the C matrix is full rank and a system of linear equations can be solved to determine the values of all the states. Each of the methods will produce an estimate of the state space matrix for the mathematical representation of the dynamic system.

2.3.3.1 Linear Least Squares Estimation

A common method for determining parameter values of a mathematical model of a dynamic system that contains many measurements is the linear least squares method. The process for computing the linear least squares estimate is shown in [12]. Linear Least Squares Estimation will fit a mathematical model that minimizes the sum of squares residuals between a set of predictions produced by the model and observed values. Linear Least Squares Estimation assumes that the uncertainties in the sources of system perturbations and measurements are zero mean, Gaussian, and stationary. Linear Least Squares Estimation is a regression method and could potentially be replaced with different regression methods like Ridge Regression or Principal Components Regression that could produce better performing results in some cases. However, for the purposes of this Thesis, Linear Least Squares regression was deemed to capture a sufficient amount of the predictive quality offered by regression methods.

2.3.3.2 Kalman Filtering as a Parameter Estimation Technique

Kalman filtering can be used as a form of parameter estimation when reformulated into a model where the parameters of the A matrix become the states of the mathematical model and are driven towards their true values by the Kalman filter. A reformulated model and algorithm can be seen in [18]. At each time step, the Kalman filter computes the innovation error between the predicted measurements and the observed measurements and moves the parameter values in a direction to decrease the error between the two. Each parameter value converges to some value (assuming the parameter values are stationary) that attempts to minimize the innovation error. One potential downfall of this method is that the error covariance matrix P will continue to decrease in size and after long time periods the filter will become less responsive to changes in the parameters of the physical system. The decrease in sensitivity can be avoided by resetting the covariance matrix P at specific time intervals causing the filter to become responsive to its prediction errors again and to converge to a new set of parameter estimates. This rapid convergence allows changes in parameter values to be recognized quicker than would occur with a

smaller covariance matrix P but can cause instability in the estimates of the parameters in situations where the parameters have not been changed recently. This is an iterative method that only has memory based on the prior value of the estimates of the states and the current value of the covariance matrix P.

2.3.3.3 Subspace Systems Identification Algorithms

The Subspace System Identification algorithms are methods for estimating the parameters of A, B, C, D, Q, and R matrices of a mathematical model using only the measurements of the physical system, the number of states, and the system inputs [19]. Unlike the previous two parameter estimation methods described above, Subspace Systems Identification algorithms do not require the values of all the states of the physical system to be directly extractable from the measurements. Subspace System Identification algorithms overcome the lack of information about the values of the states by using a sequence of orthogonal and oblique projections between Hankel matrices formed from the measurements and inputs yielding Kalman smoothing estimates of the extended observability matrix T_i and the values of the states of the system at two consecutive time indices estimate x_i and estimate x_{i+1}. From the estimates of x_i and T_i, the parameter estimates are calculated using Linear Least Squares Estimation [16]. A more in depth description of overarching model with the most basic Subspace System Identification algorithm are described in [19]. All Subspace System Identification algorithms make the assumption that the noises w and v are uncorrelated with the input u, an important point later. Two common methods of Subspace Systems Identification algorithms are Multivariate Output-Error State Space(MOESP) model and Canonical Variate Analysis(CVA) model [24][25]. Each algorithm is created by using a different weighting matrix during the orthogonal projection process. [26] shows the differences in the weighting matrices between the most common Subspace System Identification algorithms. MOESP and CVA are used in this Thesis to create real time estimates of the physical system's mathematical model both in situations where the values of the states are known and situations where the values of the states are not known.

2.3.3.4 Expectation-Maximization Algorithm

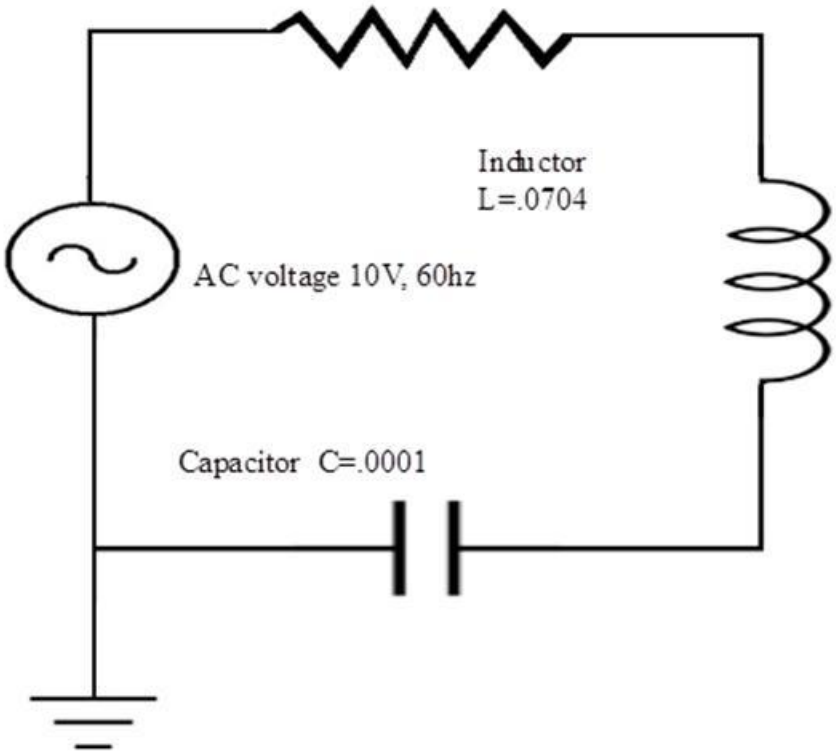
The Expectation-Maximization algorithm proposed in [15] calculates the maximum likelihood estimate of unknown parameters iteratively until they converge to some value. The process consists of two steps; first, an expectation of the log-likelihood evaluated using the current estimate for the parameters, second, a search for parameters that maximize the expected log-likelihood. This algorithm runs iteratively through these two steps until a minimum change threshold is reached. This differs from the Subspace Systems Identification algorithms, which are deterministic. Like the Subspace Systems Identification algorithms, the values of all states of the physical system do not have to be directly extractable from the measurements. However, it is necessary to know the total number of states and hidden states that there are in the physical system.

3. APPROACH

Two Systems will be simulated to determine the effectiveness of Systems Identification methods for cyber-attack detection: a software controlled RLC circuit structured as a band pass filter and the fuel injection system for a turbine. The values of states of the RLC circuit are all directly extractable so all four Systems Identification methods can be applied to the system. The fuel injection system has a single measurement that is comprised of a combination of either five or ten states, therefore, since all the values of the states are not directly extractable, only the Subspace Identification methods and the Expectation Maximization method were used on it.

3.1 RLC Filter system

Figure 1: RLC circuit with varying resistance



The RLC circuit shown in Figure 1 is the initial system upon which the parameter estimation methods will be tested. The resistance is computer controlled but has Gaussian noise around its value

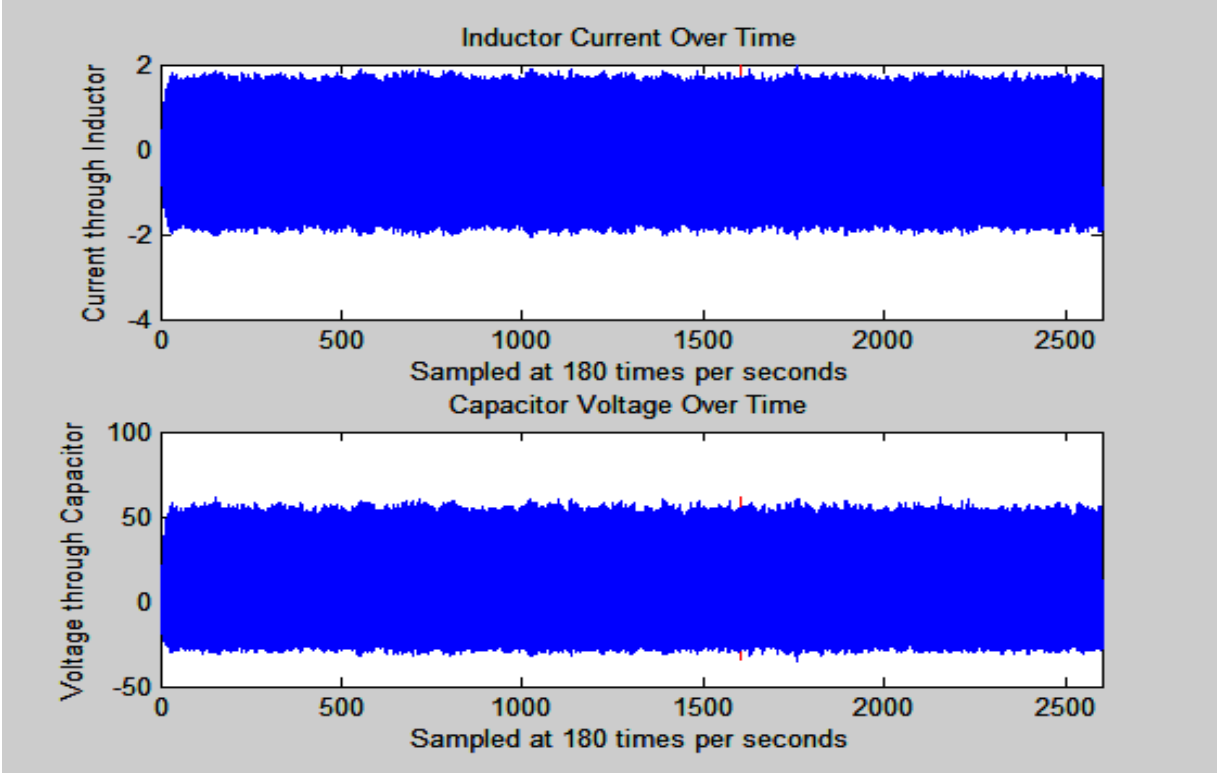
while the inductor and capacitor are physically set. There are two states in this dynamic system, the current across the inductor x_1 and the voltage across the capacitor x_2. The equations of the state space are obtained through Kerchoffs laws and shown below:

$$\begin{bmatrix} \dot{x_1} \\ \dot{x_2} \end{bmatrix} = \begin{bmatrix} -R/L & -1/L \\ 1/C & 0 \end{bmatrix} * \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1/L \\ 0 \end{bmatrix} * u + w$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} * \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + v$$

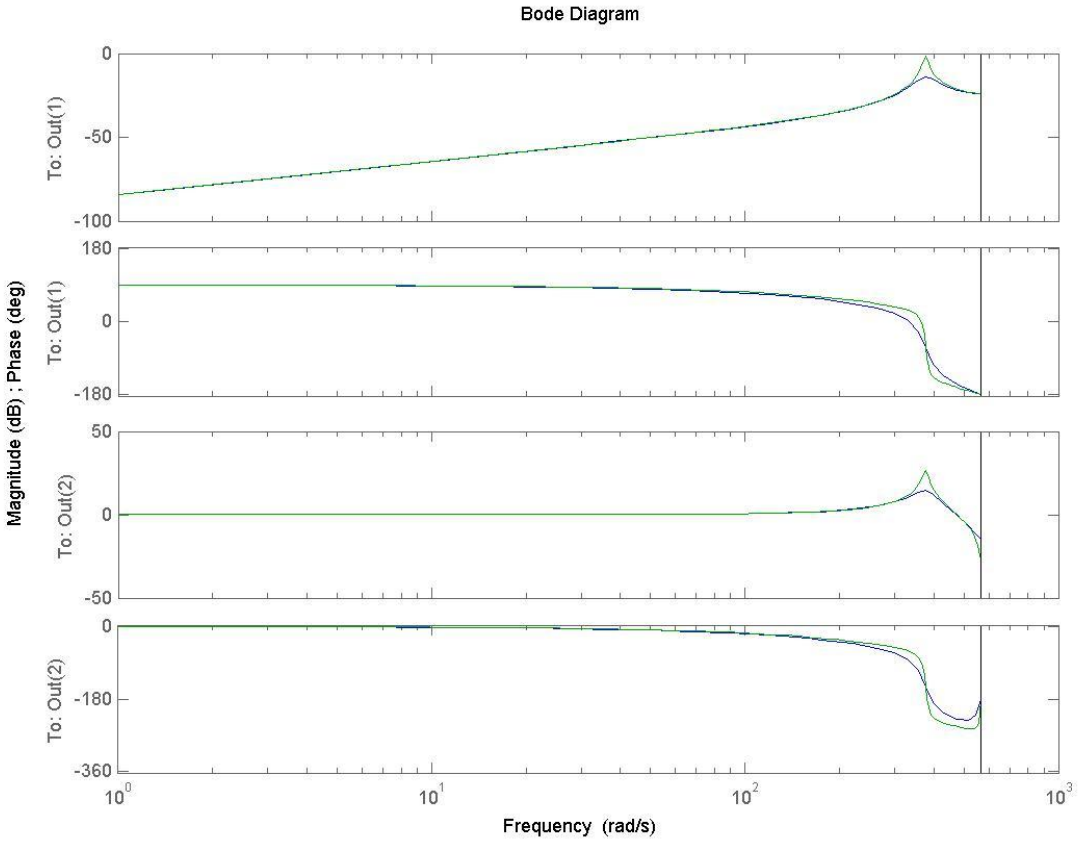
where R is the resistance of the resistor, L is the inductance of the inductor, C is the capacitance of the capacitor, w is the process error, and v is the measurement error. The system running during normal operation is shown in Figure 2.

Figure 2: RLC filter system during normal operation



The simulated cyber-attack on the RLC filter changes to the value of the computer controlled resistor through software controls and causes the RLC filter to more sensitive to specific frequencies. An example attack is shown in Figure 3, where blue is the original response and green is the attack response.

Figure 3: Bode Diagram showing modifications to the magnitude and phase of the system

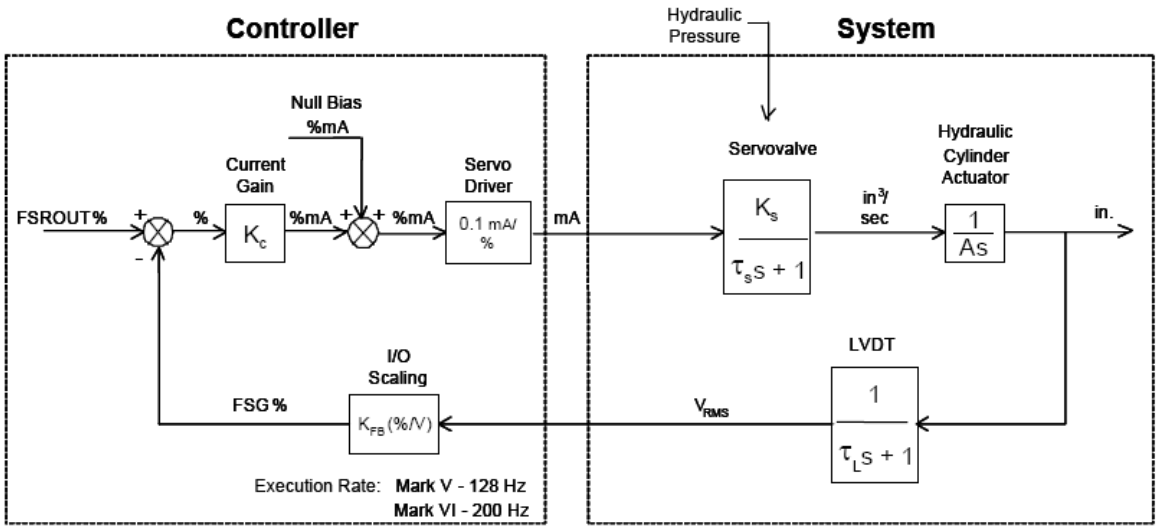


3.2 Description of the Fuel Injection System

The second dynamic system modeled is a gas turbine fuel regulator system provided by General Electric(GE) whose physical model is composed of a large number of internal states, a single input state, and a single measurement output; the measured output is the intercavity pressure in the P2 turbine compartment. The fuel injection system's goal is to maintain a constant volume outflow rate by regulating the intercavity pressure in the compartment via a hydraulic servovalve that controls an inflow rate varied

by Gaussian noise to the compartment. Figure 4 shows a model of the Position Regulator for the servovalve/actuator system, a subsystem of the fuel injector system.

Figure 4: Position Regulator Block Diagram [27]

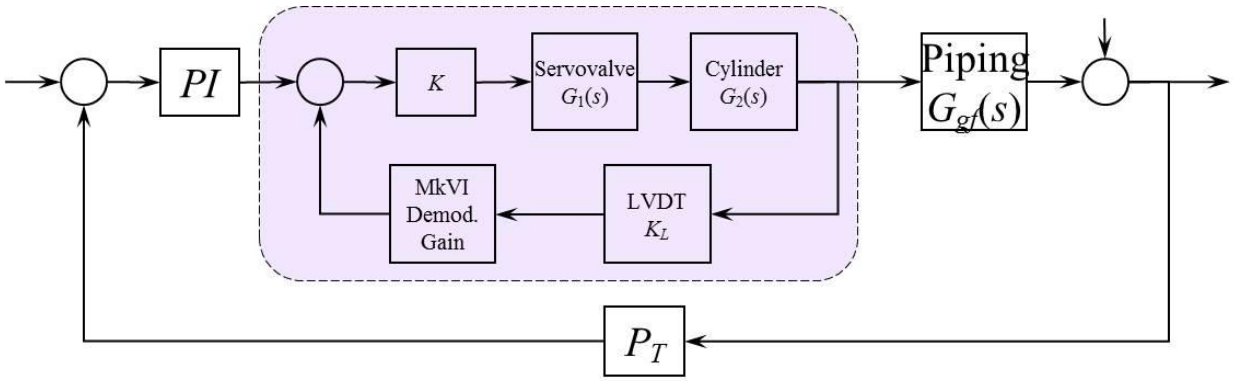


The open loop form of the transfer function for this system (including a 5ms computation delay approximated by a second order Padé approximant) is shown below:

$$H(s) = \frac{K_c * D * K_s * K_L * K_f * (s^2 + p_1 * s + p_2)}{A * s * (1 + \tau * s) * (s^2 + p_1 * s + p_2)}$$

The Position Regulator is a subsystem of the full block model, which consists of a Proportion Integral controller (PI), the Position Regulator subsystem(P_R), the response of the gas fuel piping system (G_{gf}), and the P2 pressure transducer (P_T). The full fuel injection block model is shown in Figure 5.

Figure 5: Fuel Injector System Block diagram [27]



The transfer function for the above block (including an additional computation delay of 40 ms approximated as a second order Padé approximant (PA)) is:

$$(PI)(P_R)(G_{gf})(P_T)(PA)$$

GE provided two approximations of the response of the gas fuel piping system, one at light-off or start up conditions, and one at full speed-no load or normal operating conditions [27]. This Thesis only covers cyber-attacks occurring during normal operations, so the full speed-no load approximation was used for all simulations. GE also provided an approximation of the Pressure Regulator subsystem approximated to a first order equation shown below.

$$H(s) = \frac{K_c * D * K_s * K_L * K_f * (s^2 + p_1 * s + p_2)}{A * s * (1 + \tau * s) * (s^2 + p_1 * s + p_2)} \approx \left(\frac{1}{.1 * s + 1}\right)$$

The pressure transducer was assumed to have a transfer function equal to 1. With these assumptions, the equation for a Proportion Integral controller, and the approximate computation delay, the resulting transfer function is shown below:

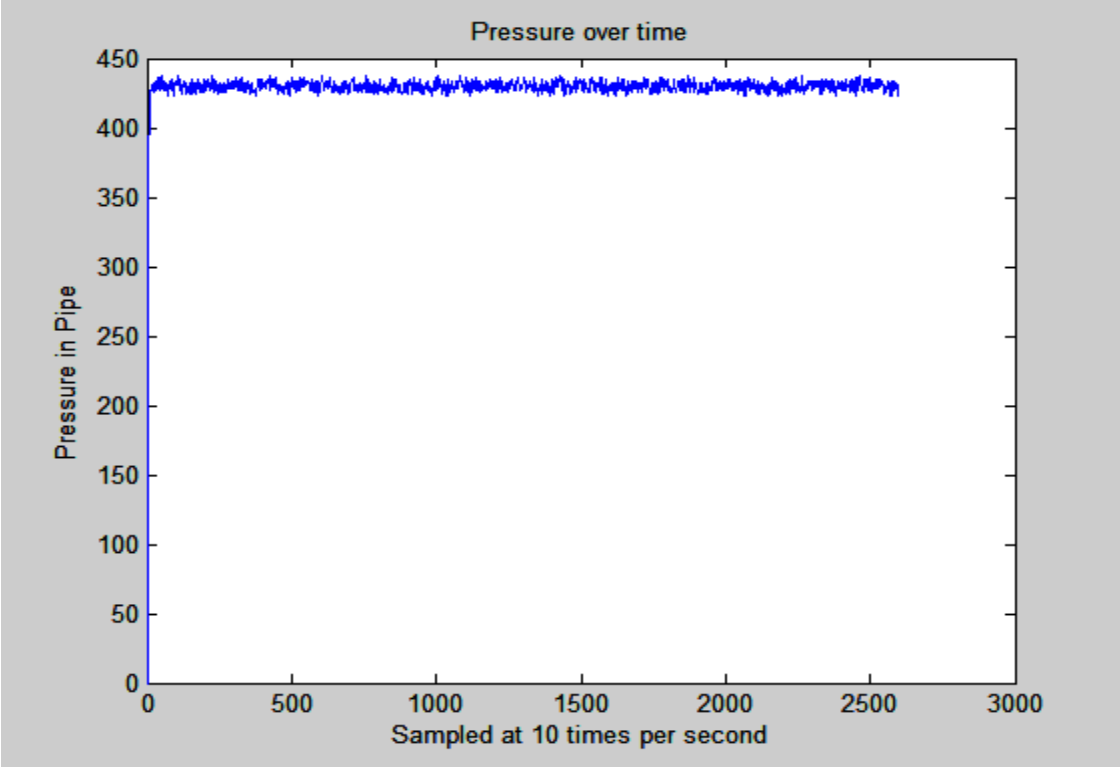
$$H(s) = \left(\frac{K_p * s + K_i}{s}\right)\left(\frac{1}{.1 * s + 1}\right)\left(\frac{5}{.015 * s + 1}\right)\frac{(s^2 + p_3 * s + p_4)}{(s^2 - p_3 * s + p_4)}$$

GE also provided the tuned parameters of Proportion Integral controller, along with the physical parameters for a real life system. The transfer function in open loop form with all the physical parameters plugged in is shown below:

$$H(s) = \frac{0.65 * s^3 - 93.06 * s^2 + 4209 * s + 33300}{0.0015 * s^5 + .34 * s^4 + 29.5 * s^3 + 1013 * s^2 + 7500 * s}$$

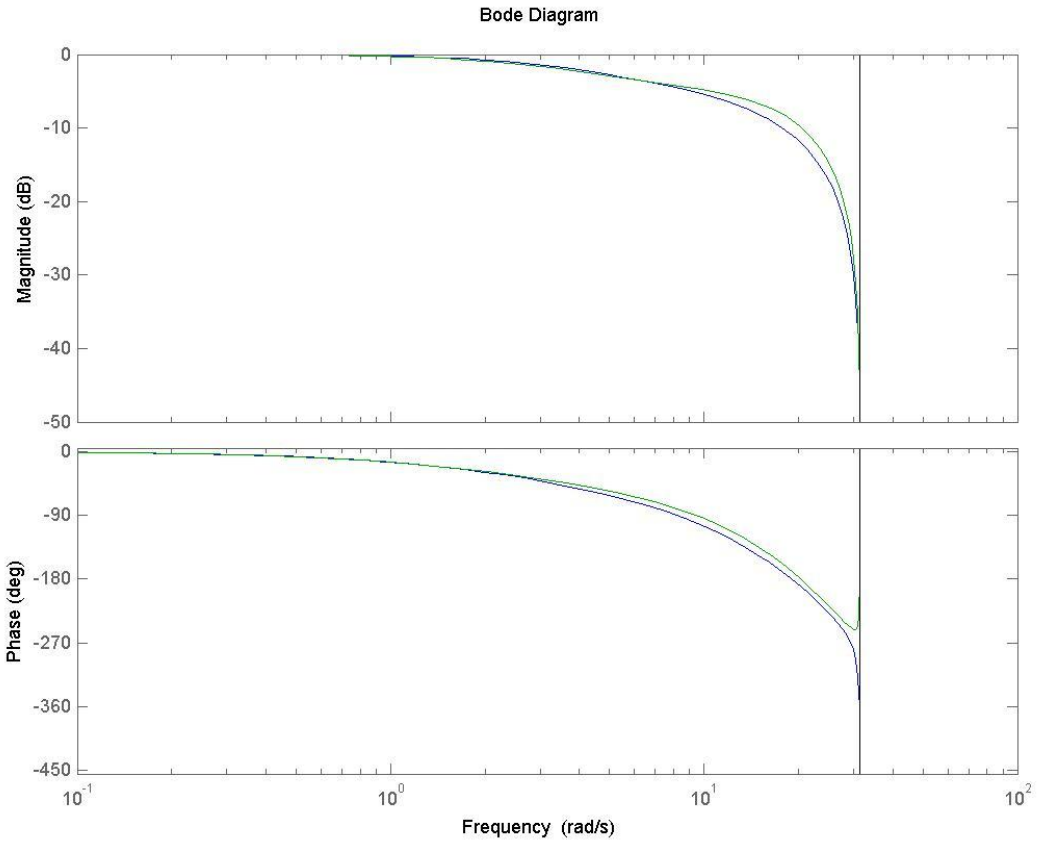
Figure 6 shows a simulation of the approximate mathematical model for a fuel injection system operating at normal conditions with a variable input.

Figure 6: Fuel injection system during normal operation



In the fuel injection system the vulnerable computer controlled variables were identified as both of the terms in the PI regulator (K_p , K_i) and the demodulation gain (D). This Thesis focuses on attacks on the PI regulator; The PI regulator is a feedback mechanism widely used in the field of control systems. The simulated cyber-attacks on the system will change either the K_p or K_i gain in the PI regulator, causing the filter to either overact to errors in the system or overact to the bias in the system, respectively. An example attack is shown in Figure 7, where blue is the designed frequency response, and green is a possible attack on the system.

Figure 7: Bode Diagram showing modifications to the magnitude and phase of the system



3.3 Attack Detection Methodology

Both the RLC filter and the Fuel injection system were approached with the same general methodology during this Thesis. The following methodology is a top level description of the detection algorithm. First, both dynamics systems were simulated with measurement and process noise. After a set warm-up time, the detection algorithms began to collect the observations necessary to create the real time estimates. Once the detection algorithms finished collecting enough observations to fill a window, a real time estimate of the parameters of the mathematical model is generated. A similarity algorithm then compares the A matrices of the real time estimate of the mathematical model behind the dynamic system with the designed and configured mathematical model. If the similarity algorithm judges the difference between the models to be above a threshold, then a deviation is flagged. If more than a set number of

deviations occur in a sliding window, then an event is flagged. After an event has been flagged, the problem then moves into the domain of event categorization.

3.3.1 Simulation Setup

Both simulations are set up similarly. The initial conditions are set as normal operating condition and at each time step a new value of the states is computed based on the designed A matrix and the input to the system. It is assumed for this Thesis that the B, C, and D matrices are known and stay constant throughout the attack. The states are then corrupted with process noise and a measurement is made. The measurement is then corrupted with measurement noise and the final measurement and current input are recorded as observations. The measurement noise and process noise are modeled as independent Gaussian white noise with an expected value of zero. Later in this Thesis, the measurement noise and process noise are selected as independent variables upon which the effectiveness of the detection algorithms is tested. For the RLC filter, the values of the states undergo Kalman filtering to attempt to reduce the noise of the measurements. The simulations iterate forward until it is judged that they are in fact at normal operating conditions at which point the detection algorithms start collecting observations to use to build their estimates. The detections algorithms all use the same measurements to create their estimates with the exception of the Kalman Filter Parameter Estimation which uses only the current values of the parameters, the current Covariance matrix P and the current Kalman filter gain K. For the first time step Kalman Filter Parameter Estimation needs to be initialized with a set of parameter values, for this Thesis the designed values of the parameters of the mathematical model are used. The detection algorithms do not create an estimate at each time step due to some algorithms being unable to perform the calculations necessary within the small time intervals in the models (The RLC filter is sampled at 180 times per second, 1.5 times the Nyquist rate) Instead the detection algorithms create estimates after set intervals of time have passed. The real time estimation period is split into two equally sized sections, the attack-free phase and the attack phase. The True Positive rates and False Alarm rates are set based on the number of

correct detections and the number of false alarms at specific thresholds that are made in each of these sections.

3.3.2 Similarity Algorithms

Three different algorithms are used to determine if the A matrices of the designed system and the real time estimate are sufficiently different to warrant a deviation flag. In order to reduce the number of parameters in the A matrix (which grows exponentially as the number of states increase) and to facilitate comparison of the real time estimates even if the realizations of each estimate's transfer functions are not similar to each other, both the designed matrix and the real time estimate were converted into the companion form representation of the matrices before comparison. It is as this point the Expectation Maximization algorithm became unfeasible. The Expectation-Maximization algorithm returned real time estimates of the system that were rank deficient during some circumstances. The rank deficient estimates could not be converted into companion form so they could not be compared to the designed and configured system. The Expectation-Maximization algorithm was not able to provide meaningful estimates and therefore was excluded as a candidate for cyber-attack detection.

Before each trial in the simulation, the selected model is run for a long period of time at the set independent variables. During this time each real time estimation method creates an estimate of the mathematical model at each time step. The values of each parameter for each estimate are recorded to establish the variability of each parameter for each estimation method. These historical parameter values are used to create distributions of the deviations from the real parameter values for each parameter and for each method. The similarity algorithms use these distributions during the independent trials to determine whether the real time estimates of the parameters move within an acceptable range or exceed the acceptable movement during normal operating conditions. Each of the similarity algorithms combines the deviations from the true designed values in different ways that affect the sensitivity of the threshold value.

3.3.2.1 Scaled Sum of Squares Method.

The scaled sum of squares method compares the parameters of the A matrices using the deviation between each parameter and the historical calculated standard deviation of each parameter value. The historically calculated standard deviations were computed in the previously described long run attack-free state. The current deviations from the parameters are scaled by the historical standard deviations so they can be combined into a single metric. The scaled deviations are then squared and summed resulting in a single value that can be compared to a threshold. The only assumption this method makes is that the historical values of each parameter follow a Gaussian distribution.

3.3.2.2 Bayesian Update Heuristic

This method compares each of the estimated parameter values obtained in the long run to the designed values of each of the parameters in the mathematical model to find Gamma distributions of the residuals during normal operating conditions for each parameter value and for each estimation method. During the simulation, the cumulative distribution function of each Gamma distribution gives the percentage of residuals from the actual value that fall below the current residual. When interpreted as a probability, the percentages of each parameter can be combined using a Bayesian Update function:

$$P(A|X_1, X_2, \dots, X_n) = \frac{P(A)\prod_{i=1}^{n} P(X_i|A)}{P(A)\prod_{i=1}^{n} P(X_i|A) + (1 - P(A))\prod_{i=1}^{n}(1 - P(X_i|A))}$$

where $P(A)$ is the probability of an attack and $P(X_i|A)$ are the percentage of residuals from the actual value that fall below the current residual for each of the parameters. The $P(A)$ can be tuned to the desired false alarm rate for an attack, in this case the desired false alarm rate is one event per month, given 8 hours of operation a day and 30 days of operation. Depending on the sampling rate (240hz for the RLC circuit, and 10hz for the fuel injection system) the probability of attack will be

$$P(A) = \frac{1}{30\ days * 8\ hours * 60\ mins * 60\ secs * sampling\ rate}$$

It should be noted that the Bayesian Update process requires that the updates be independent of each other. The parameter values in each A matrix are unlikely to be independent of one another, forcing this method to be considered a heuristic. The value of $P(A|X_1, X_2, \dots, X_n)$ can be compared to a probability threshold to flag deviations from the norm. Because of the method that the Bayesian Update Heuristic combines the deviations, it is the most effective similarity algorithm for detecting small changes to multiple parameters in contrast to a large change in a single parameter.

3.3.2.3 Binomial Method

The Binomial Method also uses the Gamma distributions of the residuals created from the long run simulation albeit in a different method. A similar percentage threshold for each of the parameters is set; residuals in the cumulative distribution that fall below this threshold are considered zeros and residuals that fall above this threshold are considered ones. When interpreted as a binomial distribution where the zeros are failures, the ones are successes, and below the percentage threshold is the probability of a zero occurring, the binomial distribution can combine the residuals of the parameters and compute the likelihood that multiple residuals exceed the percentage threshold. This probability of success can be compared to a probability threshold to flag incongruities. In this Thesis, the probability of failure was set at 95%, only deviations larger than 95% of the historical values of the residuals would result in a success. The probability thresholds for the Binomial Method operate differently than the previous two methods because of the reduced number of outcomes for the Binomial distribution. The number of outcomes the Binomial method can take is limited to the number of states plus one and any further thresholds will overlap on previous thresholds

3.3.3 Event detection

Event detection uses a window of a set number of past real time estimates to determine if an event has occurred. If the percentage of deviations in the window exceeds a set percentage, then the detection algorithm reports that an event has occurred. The simulation makes the event judgment at each new real

time estimate of the dynamic system and makes an equal number of judgments in the attack-free time period and the attack time period. The Percentage of Deviations (POD) allowed is an independent variable used in the simulation, a larger percentage allowed reduces the number of false alarms, while a shorter percentage allowed reduces the detection time and missed detection rate. The events detected during the attack free phase are marked as false positives, while events detected during the attack phrase are marked as true positives. Likewise, non-events in the attack free phase are considered true negatives, and non-events in the attack phase are considered missed detections. Each value in a specific contingency table has the total count for each outcome, and each combination of estimation method, similarity algorithm, and detection threshold level has its own contingency table.

3.4 Independent Trials

Three variables were varied to test the robustness and performance of the each of the estimation methods and similarity algorithms under different conditions. The process noise and measurement noise determine the overall variability of the system, albeit, the measurement noise is usually smaller than the process noise due to the quality of modern sensors. The percentage of deviations allowed determines how responsive the algorithms are to attacks and the importance of false alarms versus missed detections. Six combinations of the variables were selected as trials shown in Figure 8. The experiment was set up orthogonally so that the effect of each of the independent variable could be measured; a few selected interesting cases were also included. Each independent variable had three levels of values shown in Figure 9. The trails were set so that process noise of each state is set to 10%, 5%, and 1% of the standard deviation of the normal values of the states and the measurement noise is set to 5%, 1%, and .5% of the standard deviation of the normal values of the states.

Figure 8: Independent Variable combinations

	Process Noise	Measurement Noise	Incongruity percentage allowed
1. Good System, Slow detection time	Low	Low	High
2.Good System, Quick detection time	Low	Low	Low
3.Bad System, Slow detection time	High	High	High
4.Average System	Middle	Middle	Middle
5.Bad Process Noise, Good Measurement Noise	High	Low	High
6.Good Process Noise, Bad Measurement Noise	Low	High	Low

Figure 9: Levels for each Independent Variable for each system.

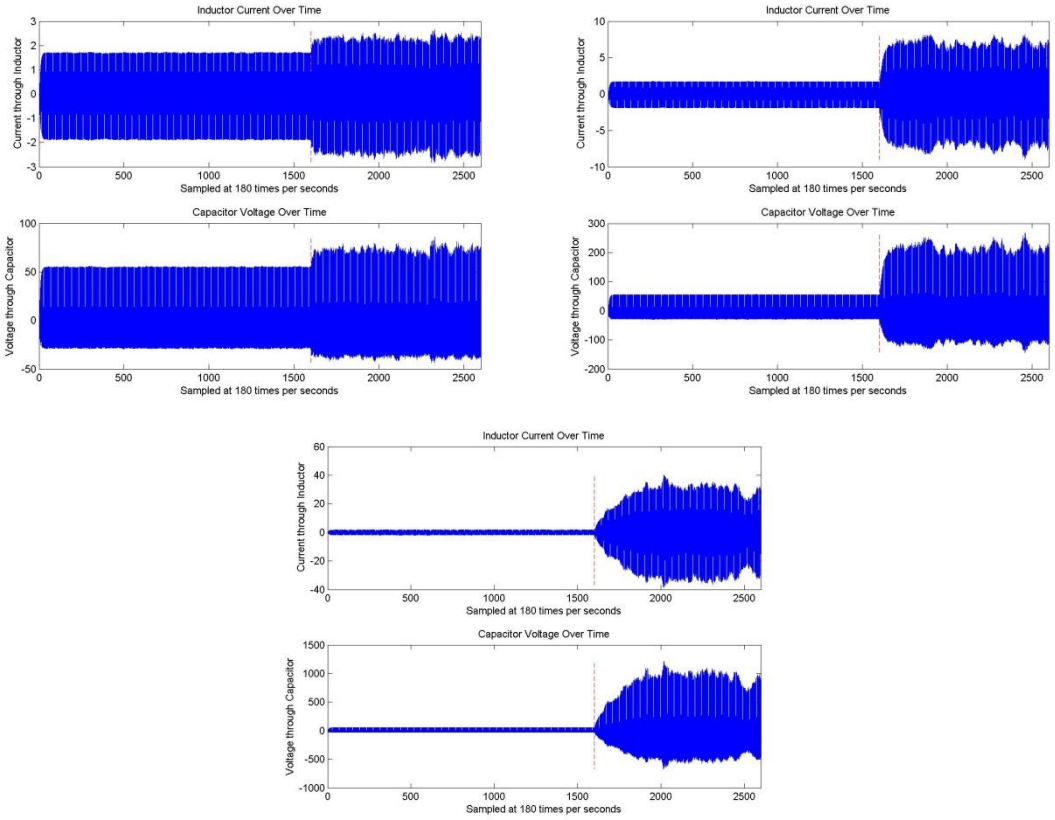
	RLC filter			Fuel injection system		
	Measurement noise(MN)	Process Noise(PN)	Incongruity percentage allowed(POI)	Measurement noise(MN)	Process Noise(PN)	Incongruity percentage allowed(POI)
Low	.005	.01	25	.01	.1	25
Middle	.01	.05	50	.1	1	50
High	.05	.1	75	1	5	75

A full list of the variable values used in each simulation can be found in Appendix 10.1. Additionally three levels of attacks were simulated for each model shown in Figure 10. The following Figures 11 and 12 show the impact of each attack on the behavior of the system.

Figure 10: Levels for each cyber-attack on each system.

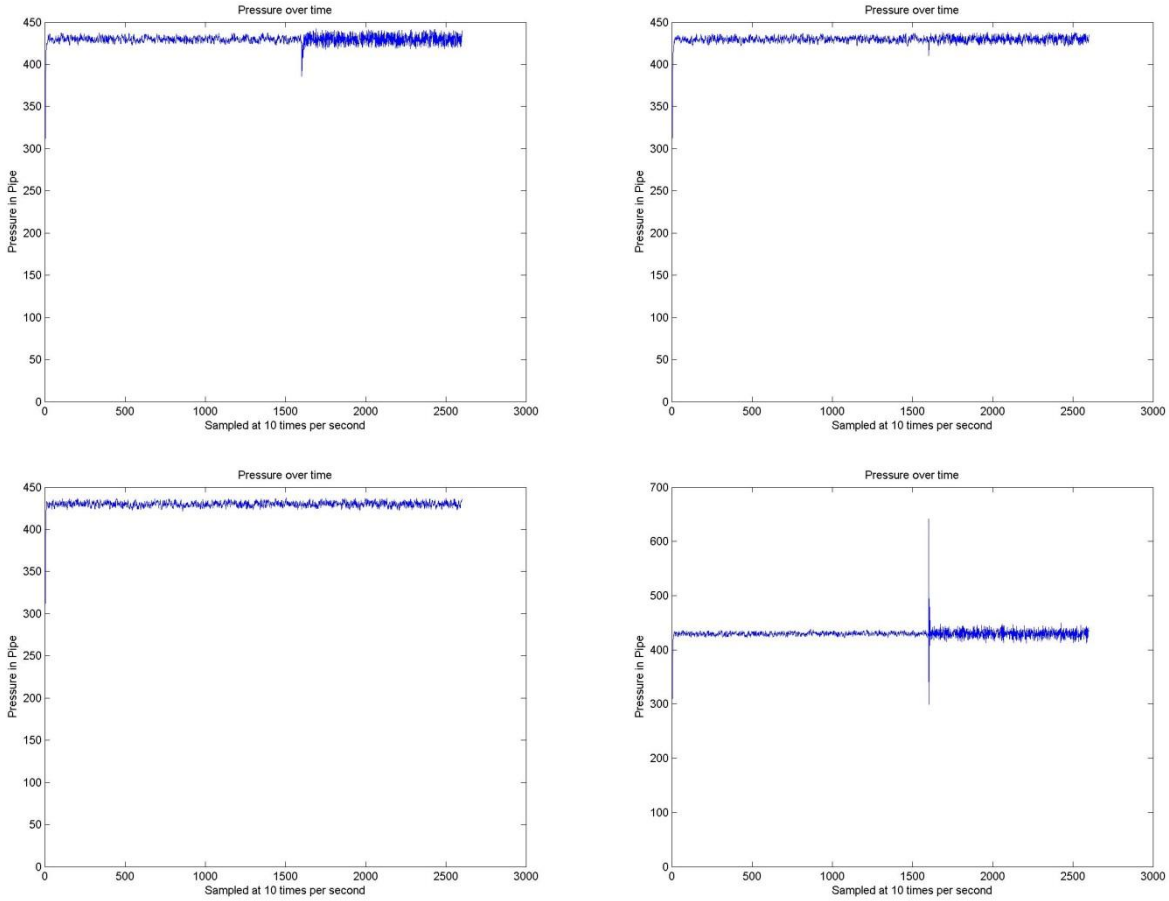
	RLC filter	Fuel Injection System	
Attack variable	Resistance R(originally 4 ohms)	Proportion gain K_p (originally .13)	Integral K_i gain (originally .888)
High	.25ohms	.52	3.552
Medium	1ohms	.26	Not simulated
Low	3ohms	.163	Not simulated

Figure 11: RLC filter cyber-attacks listed in order: The Resistance R is changed from 4 ohms to 3, 1, and .25 respectively at the dashed red line.



The cyber-attacks shown in Figure 11 show the effects of changing the resistance in the circuit to specific levels. At .25ohms and 1ohm the change is readily apparent, however at 3ohms, it is harder to differentiate the attack.

Figure 12: Fuel Injection system listed in order: K_p is changed from .13 to .52, .26, and .163 or K_i is changed from .888 to 3.552



The cyber-attacks shown in Figure 12 show the effects of changing gains in the PID controller to specific levels. Unlike the RLC filter system, it is not readily apparent that the system has changed in the attacks on the PID controller. After the initial response to the attack, all the systems settle into values that are difficult to distinguish from the attack free state. Methods that can detect subtle cyber-attacks like the ones shown would be valuable to operators of critical systems.

5. RESULTS

Each trial was compared using two tradeoff criteria, first, the False Alarm versus True Positive rate, and, second, the False Alarm versus Detection Time. The False Alarm versus True Positive was the primary method used to compared estimation methods and similarity algorithms. The experiments are also evaluated secondarily from a False Alarm versus Detection Time viewpoint to determine if there are large changes in the detection time. Users of the cyber-attack detection algorithms fall into two groups, those who place the most value in minimizing the number of missed detections and the detection time, and those who are interested in minimizing the number of false alarms to reduce operator fatigue. Each viewpoint is explained and the results are analyzed from both viewpoints. For both of the simulated systems, the false alarm rate needs to be sufficiently low so that the detection algorithm is not causing operator fatigue. The large time scales suggest that an upper bound on the percentage of false alarms be less than 10%. All the ROC curves are scaled so that the maximum false alarm rate shown is 10%. It is likely that this rate is still too high for normal operation, but on high risk and high value systems, the possibility of a missed detection may outweigh the possibility of operator fatigue.

5.1 RLC Filter

5.1.1 Effect of Percentage of Deviations (POD) allowed

The POD allowed was hypothesized to affect the false alarm and true detection rates of the algorithms. Figures 13 and 14 shows the effect the POD has on the ROC curves. The results are as expected; decreasing the POD increases the true detection rate, but does not cause the false alarm rate to decrease. The Bayesian Heuristic and Binomial method both perform better than the SSS method in this experiment and show the predicted decrease in false alarm rate and true positive rate. The similarity algorithm that performs better in the high POD experiment is the Binomial method, while the method that performs better in the low POD experiment is the Bayesian method. Of the estimation methods the KFPE

performs better than the alternative options in nearly all of the experiments. In Figures 15 and 16, for the Bayesian method, the false alarm rate decreases a large amount without sacraficing much in terms of detection time. The Binomial method shows a slightly lower false alarm rate for the same detection time in the higher POD experiment. The Binomial method is much faster then the Bayesian Heuristic method with the maximum time being much slower at a zero false alarm rate. The slowest estimation method in all cases is the KFPE.

Figure 13: ROC of POD=.75 at Attack level 2, MN=.005 and PN=.01

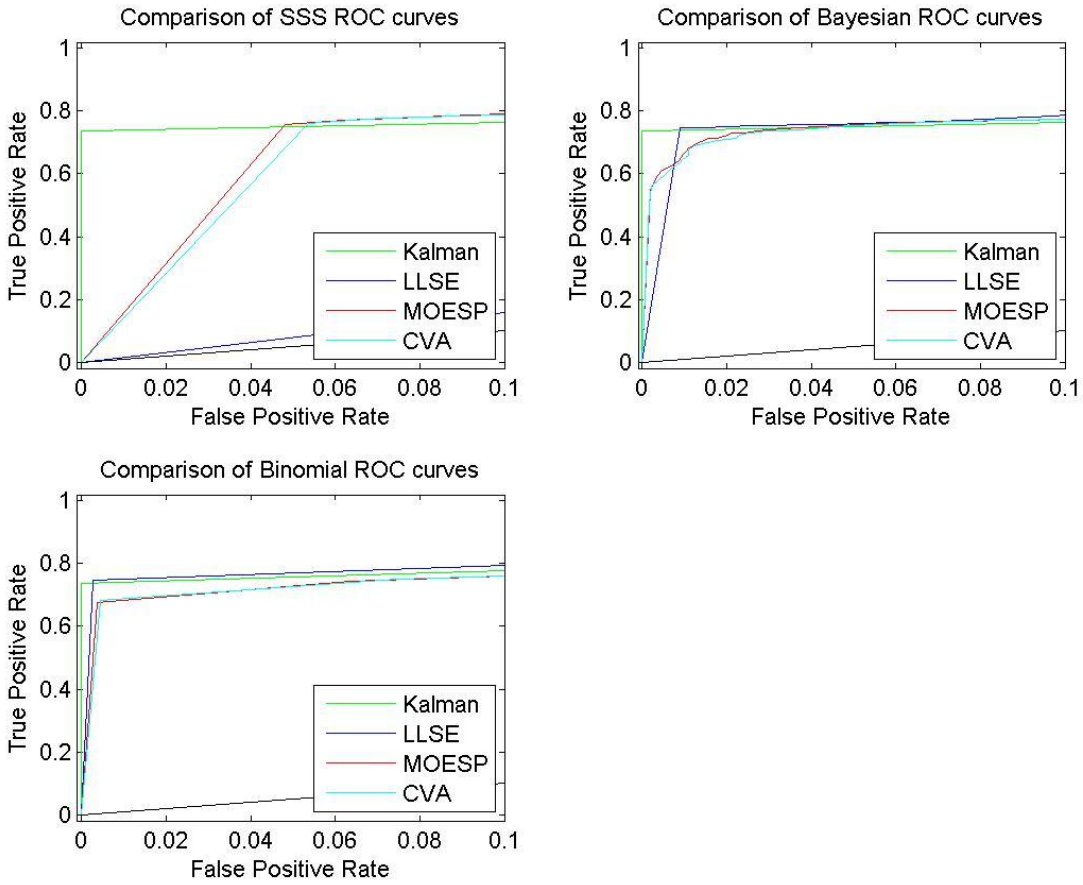


Figure 14: ROC of POD=.25 at Attack level 2, MN=.005 and PN=.01

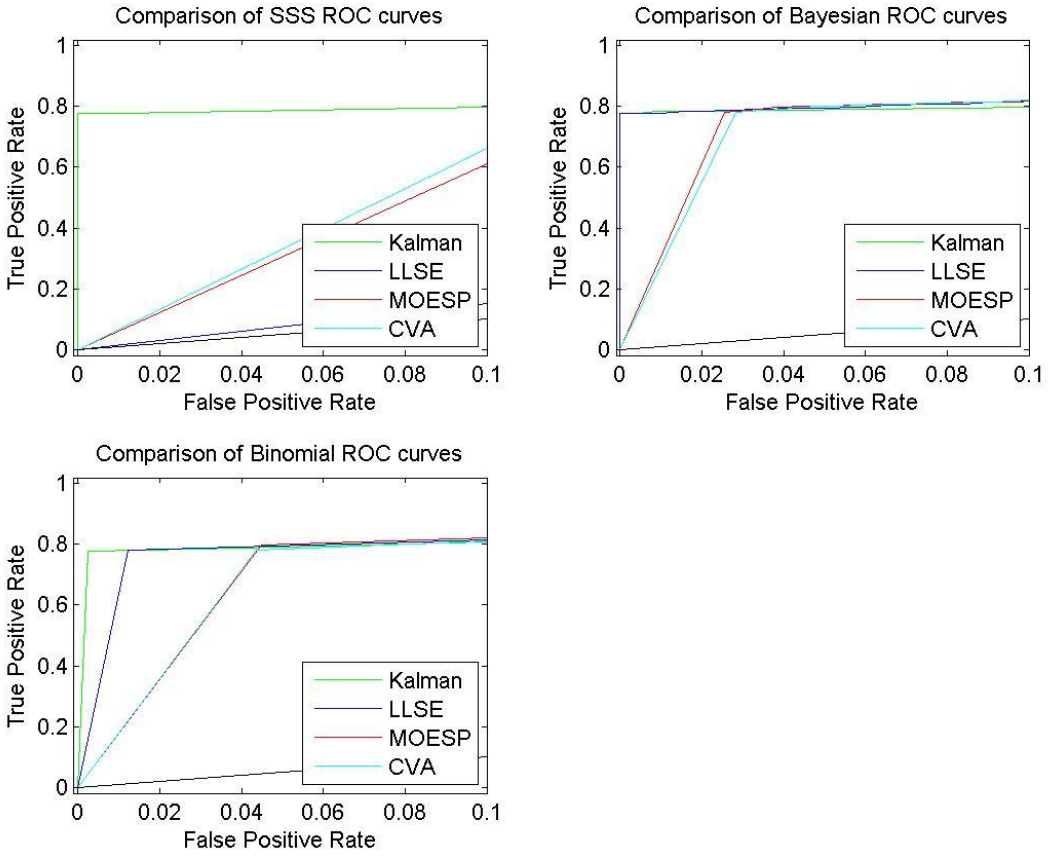


Figure 15: False Alarm rate vs. Detection Time of POD=.75 at Attack level 2, MN=.005 and PN=.01

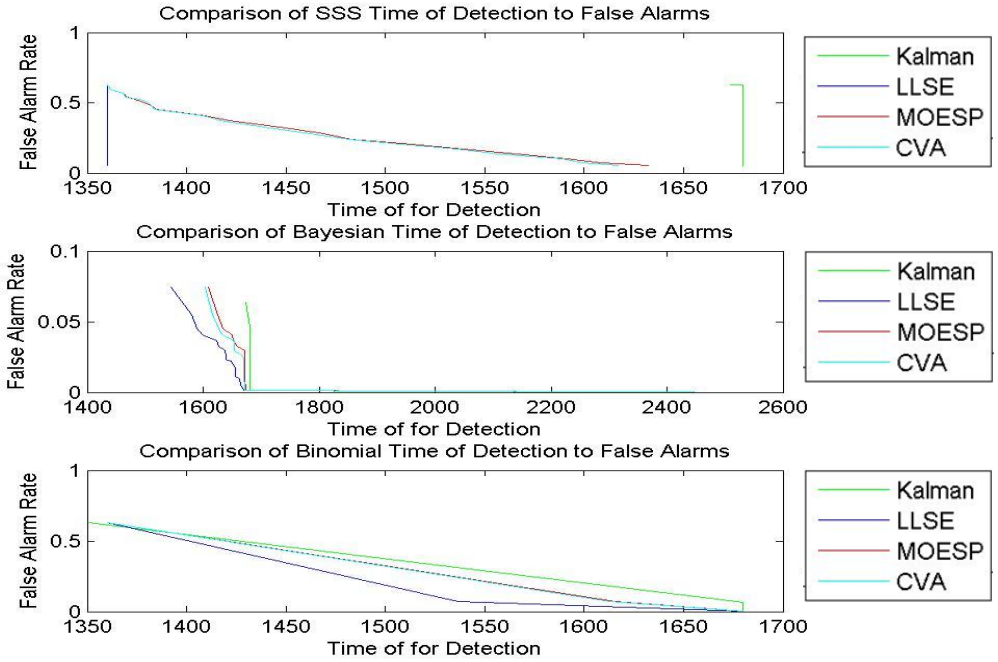
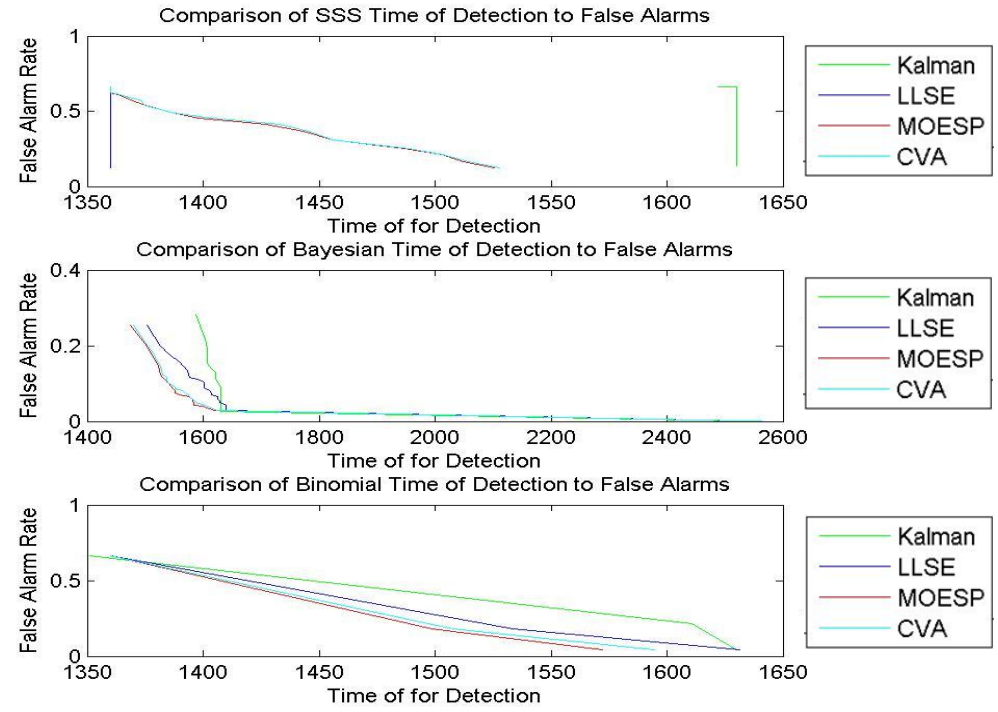


Figure 16: False Alarm rate vs. Detection Time of POD=.25 at Attack level 2, MN=.005 and PN=.01



5.1.2 Effect of Process Noise

The second experiment shows the effect of increased process noise on the predictability of the system. The only differentiator between the experiments conducted to generate Figure 13 and Figure 17 is the size of the process noise in the system. Adding process noise to the system makes the SSS method and Bayesian Heuristic method more predictive, and makes the Binomial method less predictive. A possible explanation for the increase in the false alarm rate is that the distributions created from the long run historical data are too narrow and not assigning enough weight to deviations in the tails. Increasing the noise would cause the fitted distributions of the deviations to widen and increase the size of the tails. This would lead to lower probabilities for large deviations and would reduce the false alarm rate. The increase in predictive power could also be due to the estimation methods being over determined at the lower noise levels and producing better estimates from relaxed bounds on the noise. By constraining the noise in the system, the estimation methods could have been forced to operate on over determined data. Another alternative reason for the increase in the false alarm rate is that the SSS and Bayesian Heuristic methods are more sensitive at detecting a multitude of small changes to the parameters in the A matrix than the Binomial method. If the additional process noise causes the estimation methods to register smaller changes in multiple variables as opposed to large changes in single variables, then the SSS and Bayesian Heuristic methods would perform better on the system, while the Binomial method experiences a reduction in predictive power. One more hypothesis for the increase in the false alarm rate is that increasing the process noise may force the PI controller to contribute more to the stability of the system. The contributions made by the PI controller may create a more noticeable presence in the real time estimates mathematical model and make detection of attacks on the PI controller easier to detect. Of the four estimation methods the KFPE method once again performs better than the three alternatives. Figure 18 shows that the KFPE method still takes the longest time detecting and attack, and that increasing the process noise increases the detection time across the board. The Binomial method still performs much better than the Bayesian method in detection time at a zero false alarm rate.

Figure 17: ROC of POD=.75 at Attack level 2, MN=.005 and PN=.1

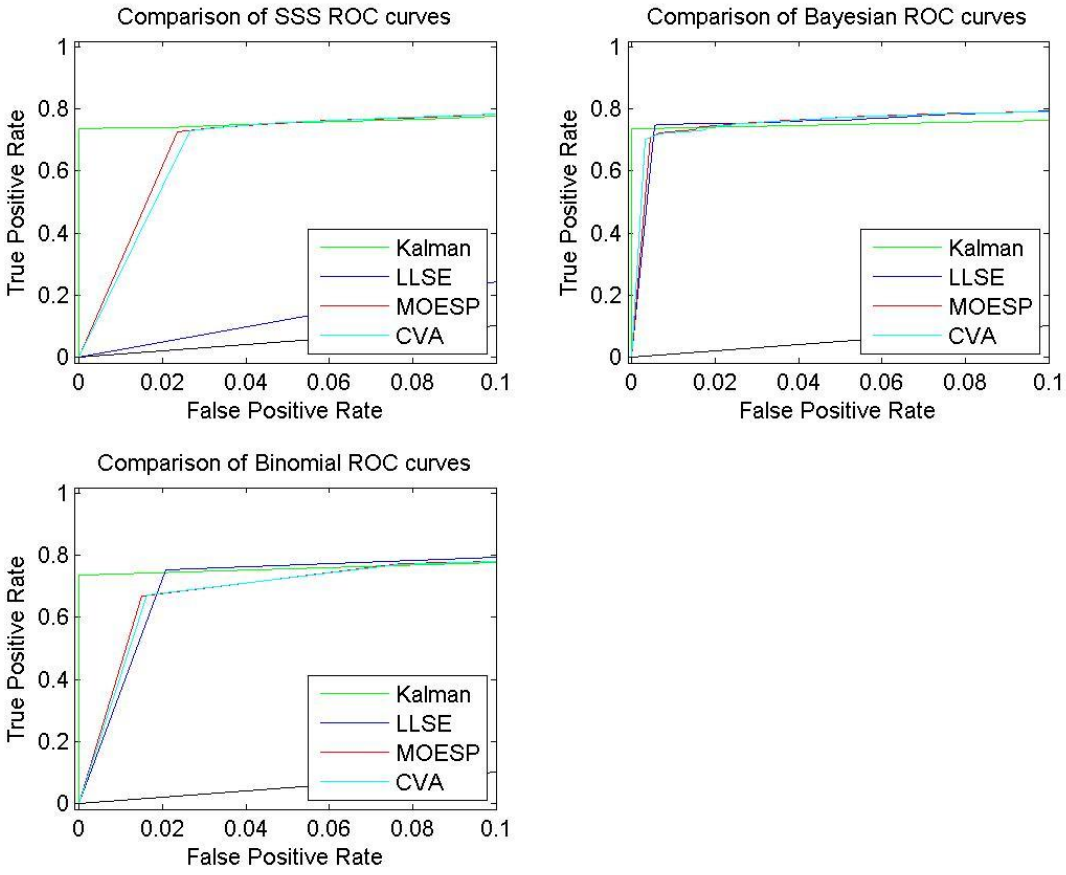
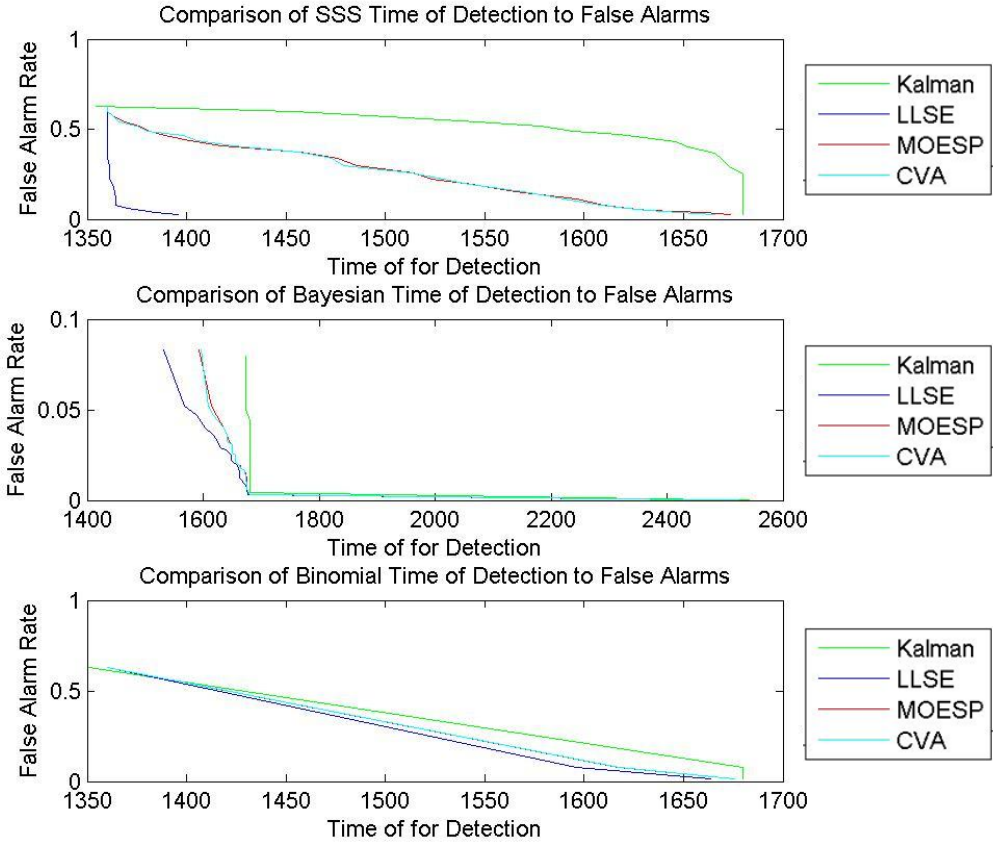


Figure 18: False Alarm rate vs. Detection Time of POD=.75 at Attack level 2, MN=.005 and PN=.1



5.1.3 Effect of Measurement Noise

The effect of measurement noise can be measure by comparing Figure 13 and Figure 19 All other independent variables during the experiment are held constant. Increasing the measurement noise causes a decrease in the predictive power of the SSS method and the Binomial method, and an increase in the predictive power of the Bayesian method. Like Figure 34, the predictive power of the Bayesian Heuristic method increases. The Bayesian Heuristic method is the most sensitive of the similarity algorithms to the values of multiple parameters, reinforcing the hypothesis that increasing the noise places more value on collective change in the parameters of the A matrix as opposed to single individual changes. The LLSE method outperforms the KFPE method in the Bayesian ROC Curves however performs badly in the SSS ROC curves. Figure 20 shows that for the Bayesian Heuristic, the effect of the process noise kept the time for detection to about the same level for all experiments, but increased the false alarm rate by a large amount. For the Binomial the detection time was about the same for all cases except for the LLSE method which increase to the levels of the Bayesian Heuristic method.

Figure 19: ROC of POD=.75 at Attack level 2, MN=.05 and PN=.05

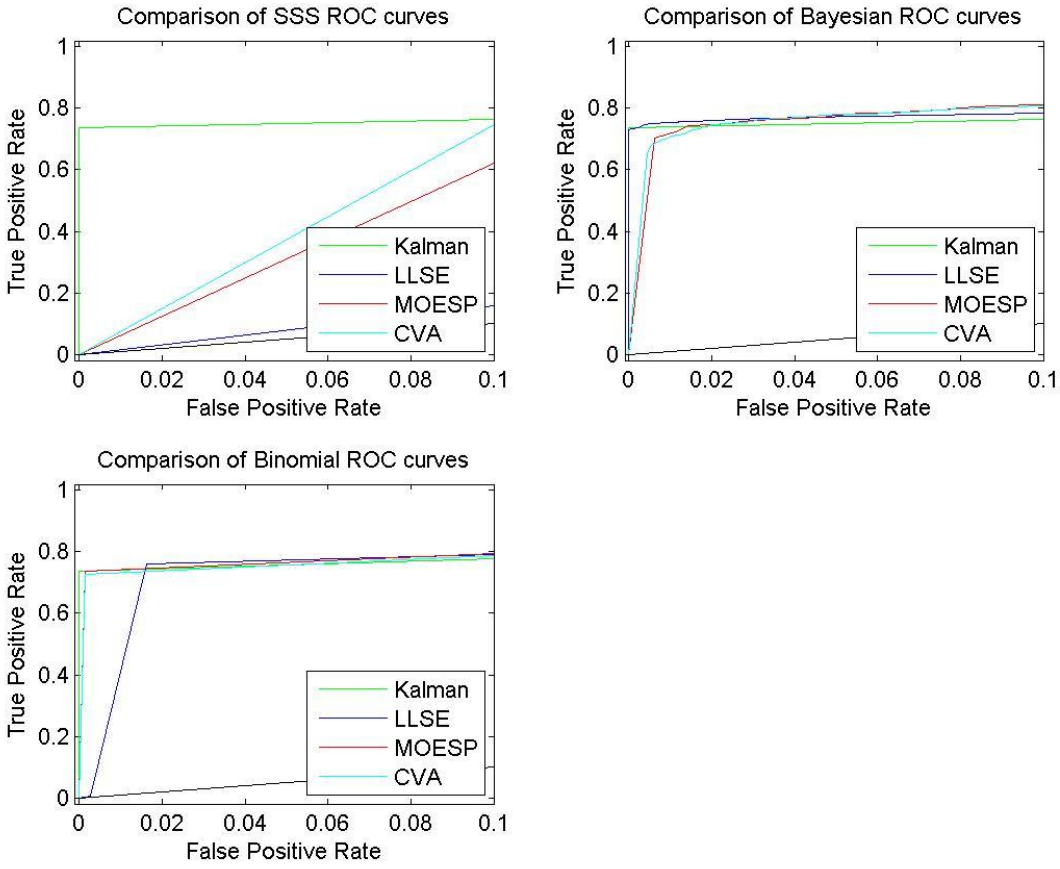
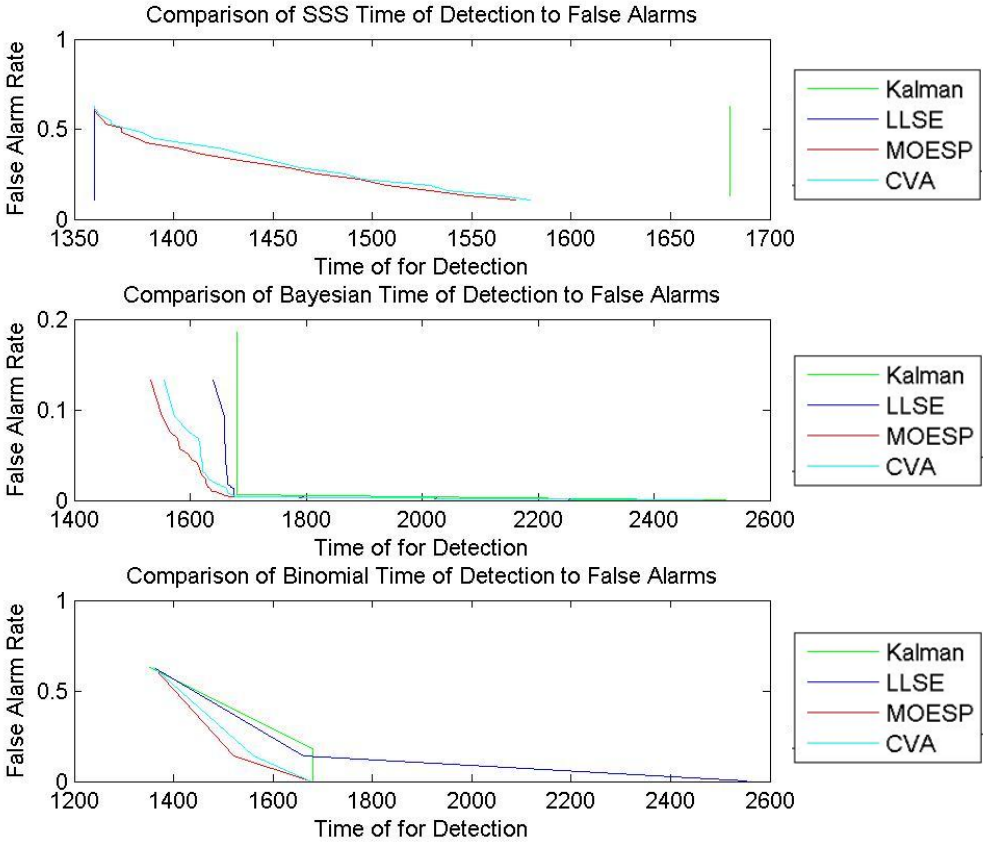


Figure 20: False Alarm rate vs. Detection Time of POD=.75 at Attack level 2, MN=.05 and PN=.05



5.1.4 Worst Case Scenario

The ROC curve showed in Figure 21 shows the toughest experiment given to the detection algorithm. The process noise and measurement noise are both set to their highest values, the POD is set to 75% and the attack is set to the smallest change in parameters simulated. The KFPE still is able to detect the change in parameters with a near zero false alarm rate and a 77% true positive rate. The actual parameter values for the system in a single simulation are shown in Figure 39. The blue values are a_1 and the green values are a_2 in Figure 22 but it is clear that a change is detected when the attack starts. The parameters attempt to return to their previous values after a certain period of time in all cases but the KFPE case. This is possibly due to the Kalman smoother on the measurements pulling the values back to normal operation. Figure 23 shows the false alarm rate vs the detection time for the worst case scenario. The detection time is similar to what has been seen in the other experiments and acceptable for the range of detection necessary. The Bayesian Heuristic method still performs much worse the Binomial method.

Figure 21: False Alarm rate vs. Detection Time of POD=.75 at Attack level 3, MN=.05 and PN=.1

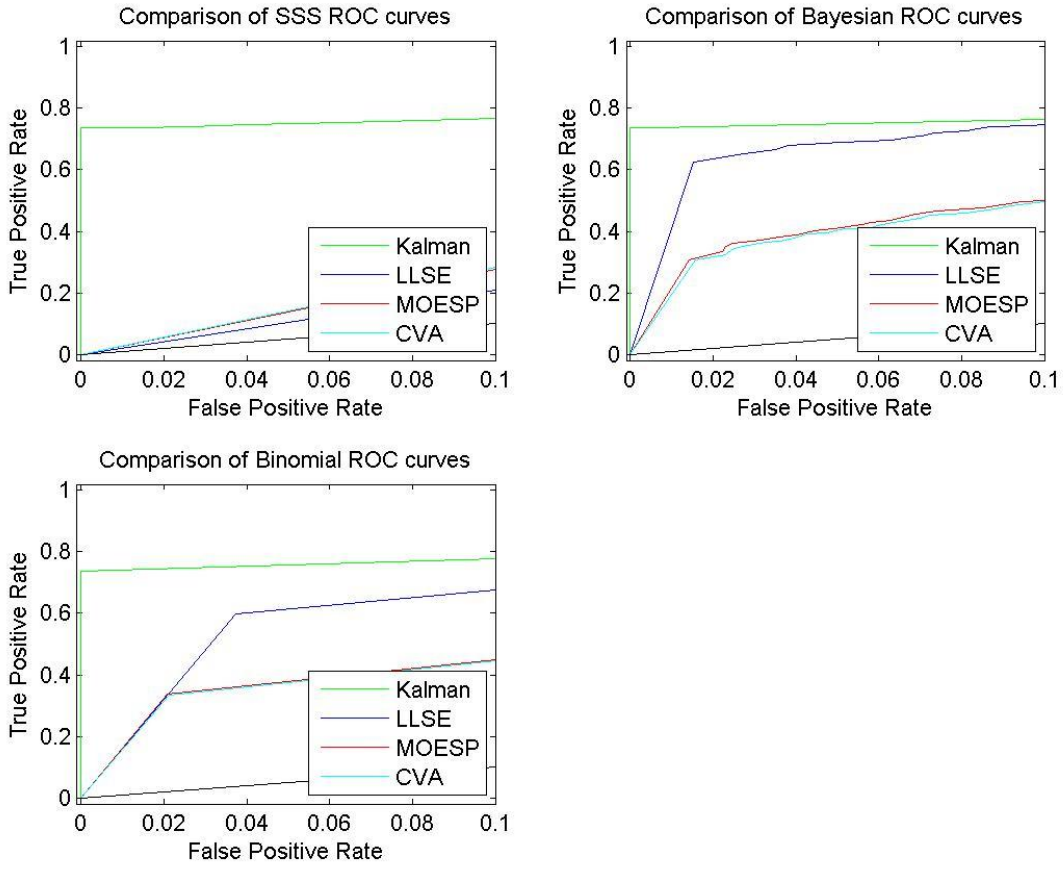


Figure 22: Real time estimates of parameter values a_1 and a_2 in the state matrix A at POD=.75, Attack level 3, MN=.05 and PN=.1

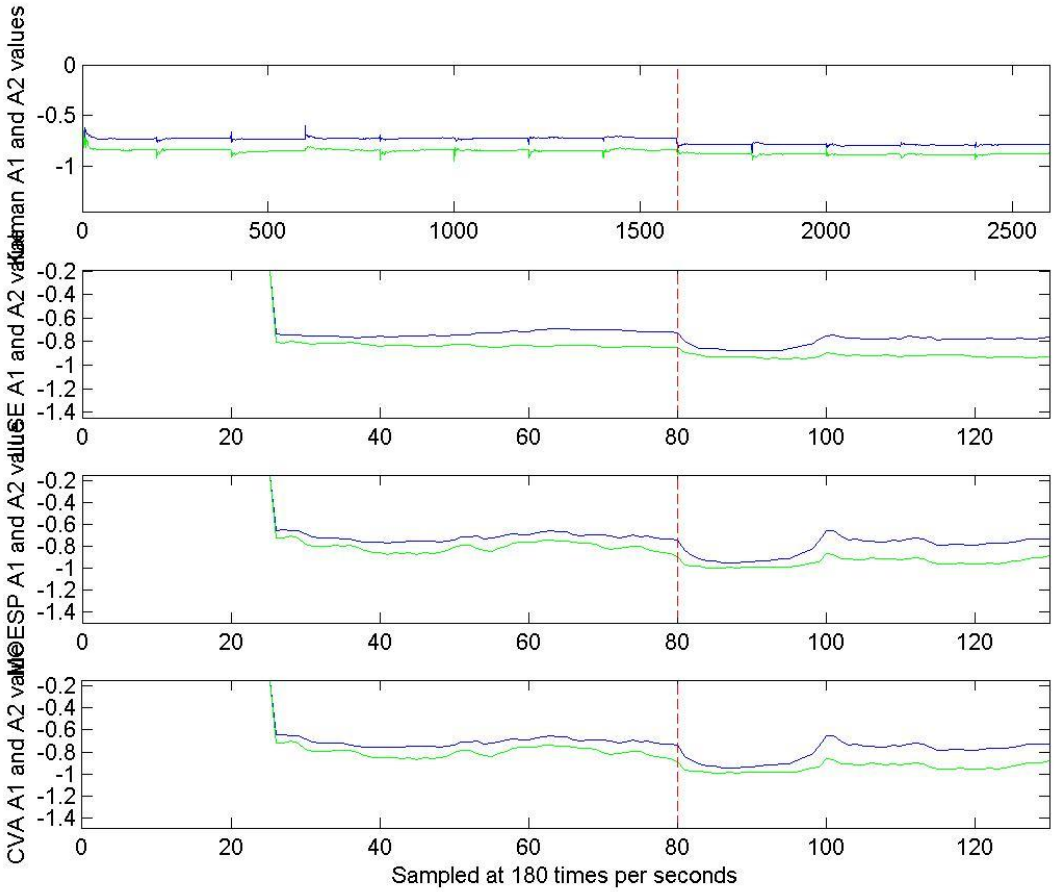
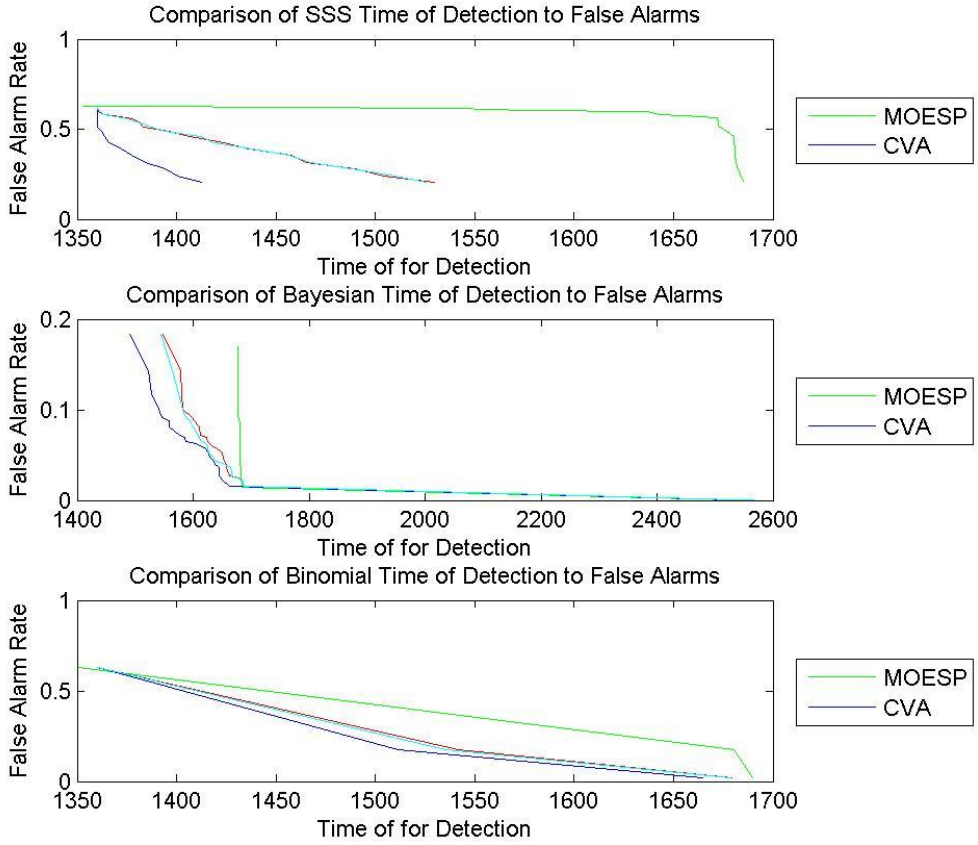


Figure 23: False alarm vs. Missed detection rate at POD=.75, Attack level 3, MN=.05 and PN=.1



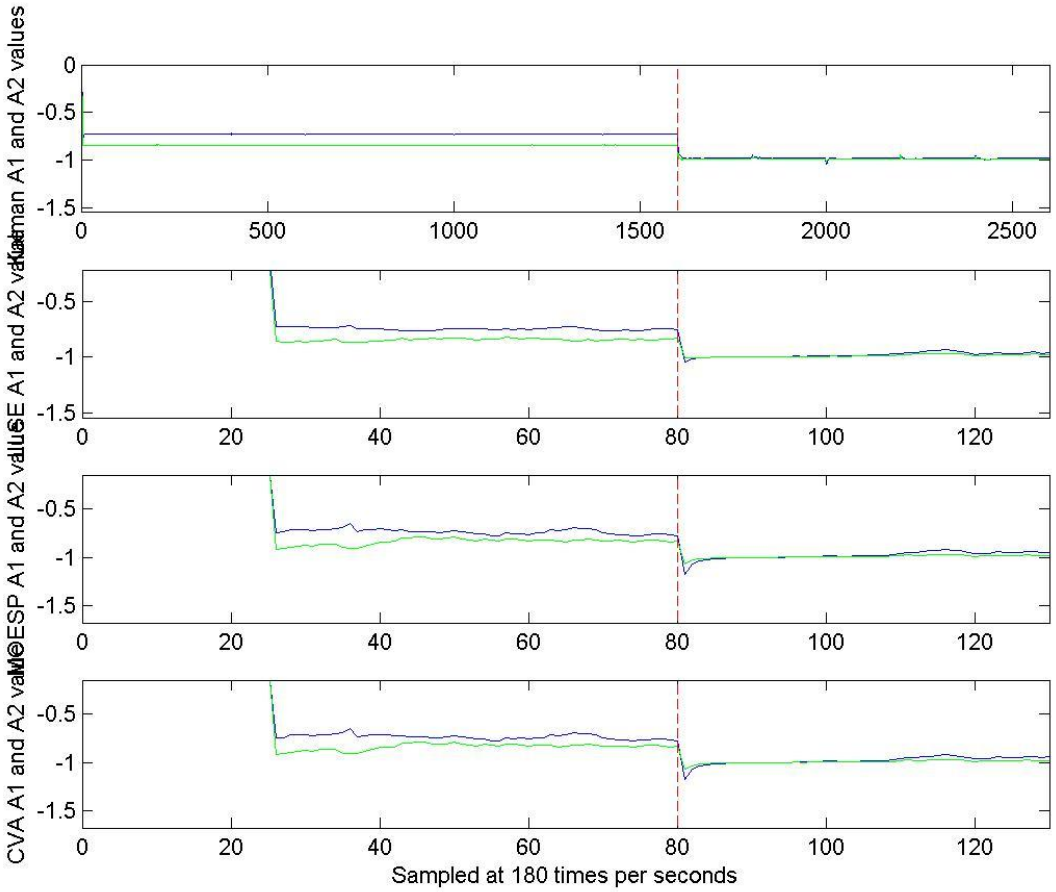
5.1.5 Overall Statistics

Overall the best similarity algorithm was the Bayesian Heuristic method in 87.5% of the experiments, the Binomial method in 6.25% of the experiments and indistinguishable in 6.25% of the experiments. The best estimation method was KFPE in 75% experiments, LLSE in 14% of the experiments, MOESP in 5% of the experiments, and indistinguishable in 5% of the experiments. For this system KFPE combined with the Bayesian Heuristic similarity algorithm produced the best predictor of an attack. The KFPE produced estimates that were very consistent in all trials of the experiment, the difference in the parameter estimates between the best case scenario and the worst case scenario is imperceptible. Figures 22 and 24 show the estimates of the parameters at each time step for each estimation method in the best case scenario and in the worst case scenario. In both cases the attack on the systems can be detected by estimating the values of the parameters at each time step. The remaining performance ROC charts and False Alarm vs Detection Time charts an can be found in Appendixes 10.2.1 and Appendix 10.2.2

Percentage of Success	LLSE	KFPE	MOESP	CVA	No difference
False Alarm Reductionist	15%	83%	2%	0	0
Missed Detection Reductionist	28%	59%	13%	0	0

Percentage of Success	SSS	Bayesian	Binomial	No difference
False Alarm Reductionist	0	89%	5.5%	5.5%
Missed Detection Reductionist	0	89%	5.5%	5.5%

Figure 24: Real time estimates of parameter values a_1 and a_2 in the state matrix A at POD=.75, Attack level 1, MN=.005 and PN=.01



5.1.5.1 Evaluation from a False Alarm Reductionist Viewpoint

In this scenario the operator is primarily focused with reducing false alarms to a minimum, and secondarily focused on maximizing the true detection rate and minimizing detection time. For the RLC filter system the best estimation method in most cases is the KFPE method. In practically all experimental cases at all attack levels KFPE registered a false alarm rate of zero while still maintaining at least a 73% true detection rate. When the POD was set to 75%, the true detection rate could be maximized to 73% without triggering a false alarm. When the POD was set to 25%, the Binomial Method triggered false alarms in two experimental cases below the 73% true detection rate. The best similarity algorithm for reducing the false alarm rate was the Bayesian Heuristic method; with the POD set to 25% it increased the true detection rate to 77% without triggering a false alarm. The KFPE consistently offered a reduced false alarm rate at the cost of an increase in detection time, which may be acceptable for some operators.

Detailed Breakdown:

False Alarm Reductionist	LLSE	KFPE	MOESP	CVA	No difference
SSS	0	0	0	0	0
Bayesian	4	11	1	0	0
Binomial	1	0	0	0	0
No difference	0	1	0	0	0

The Detailed Breakdown shows that for the False Alarm Reductionist, the results are as expected, with the KFPE combined with the Bayesian Heuristic performing the best in most of the cases.

5.1.5.1 Evaluation from a Missed Detection Reductionist Viewpoint

In this scenario the operator is primarily focused with reducing missed detections to a minimum, and secondarily focused on minimizing detection time and minimizing the false alarm rate. The variable that had the largest effect on the true detection rate was the POD, with a lower POD increasing the true detection rate. For the RLC filter system the best estimation method in most cases is the KFPE method. In some cases the LLSE method performs better than KFPE method, but the KFPE method consistently outperforms the LLSE method in experiments where the cyber-attack changes the component value a minimal amount. The Bayesian Heuristic performed better than the other similarity algorithms at maximizing the true detection rate. In both operator scenarios for this experiment, the KFPE method and Bayesian Heuristic method were the optimal selections, with the differentiating factor being the POD.

Detailed Breakdown:

Missed Detection Reductionist	LLSE	KFPE	MOESP	CVA	No difference
SSS	0	0	0	0	0
Bayesian	4	6	6	0	0
Binomial	1	0	0	0	0
No difference	1	0	0	0	0

The detailed breakdown shows that the results were actually much closer then suggested across estimation methods. The reason for the decrease in disparity is that often when evaluating estimation methods the Bayesian Heuristic similarity algorithm continued the best estimation method across all screening algorithms. In the SSS and Binomial methods, however, the KFPE estimation method performed better than the other estimation methods, inflating its score.

5.2 Fuel Injection System

5.2.1 Effect of Percentage of Deviations (POD) allowed

The first case examined shows the difference the percentage of deviations has on the false alarm and true positive rate. Figure 25 shows the POD rate at 75% and Figure 26 shows the POD rate at 25% for the second attack. Increasing the percentage has nearly no effect on the SSS Method, a clear positive effect on the false alarm rate and negative effect on the true positive rate for the Bayesian method, and a positive effect on the false alarm rate and negative effect on the true positive rate for the Binomial method. The Binomial performs better in the first experiment, but the Bayesian Heuristic appears to perform better in the second experiment. Both operator cases would select it over the other similarity algorithms. MOESP outperforms CVA at a three to one ratio in these two cases (The SSS similarity algorithm shows both estimation methods too close to determine the most effective method). The false alarm vs. detection rate shown in Figures 27 and 28 showed that the Binomial method performed better than the alternatives and that the CVA usually outperformed the MOESP algorithm. The differences between the maximum amounts of time to detect are much smaller in this simulation.

Figure 25: ROC of POD=.75 at Attack level 1, MN=.01 and PN=.1

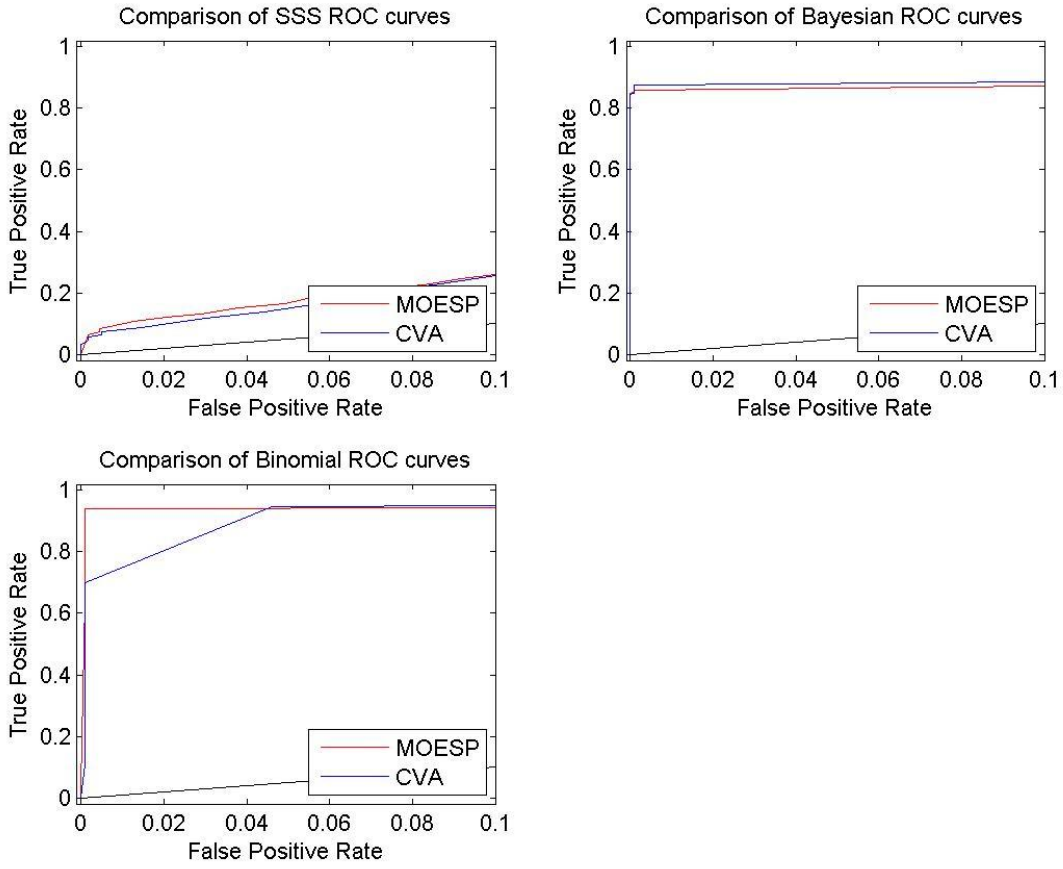


Figure 26: ROC of POD=.25 at Attack level 1, MN=.01 and PN=.1

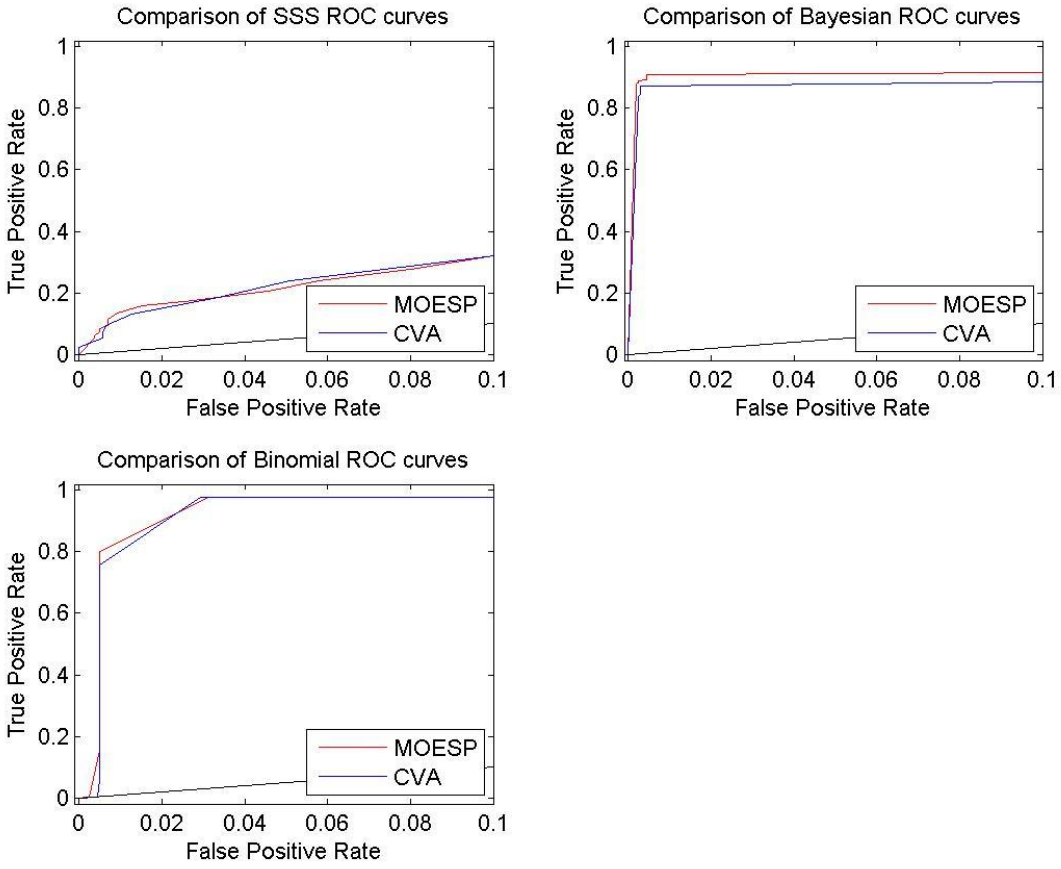


Figure 27: False Alarm rate vs. Detection Time at POD=.75, Attack level 1, MN=.01 and PN=.1

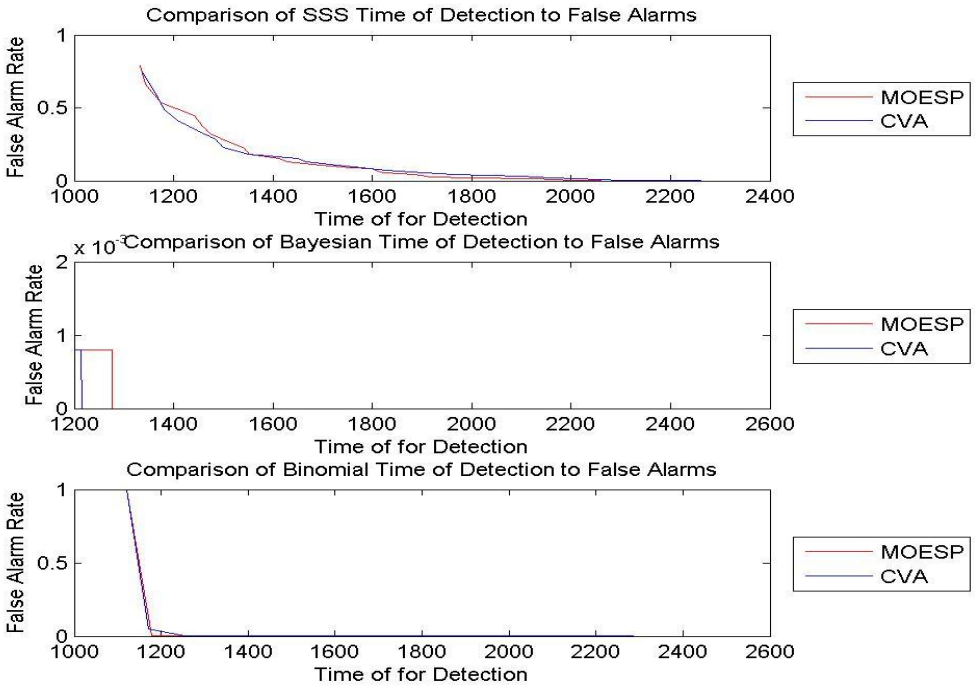
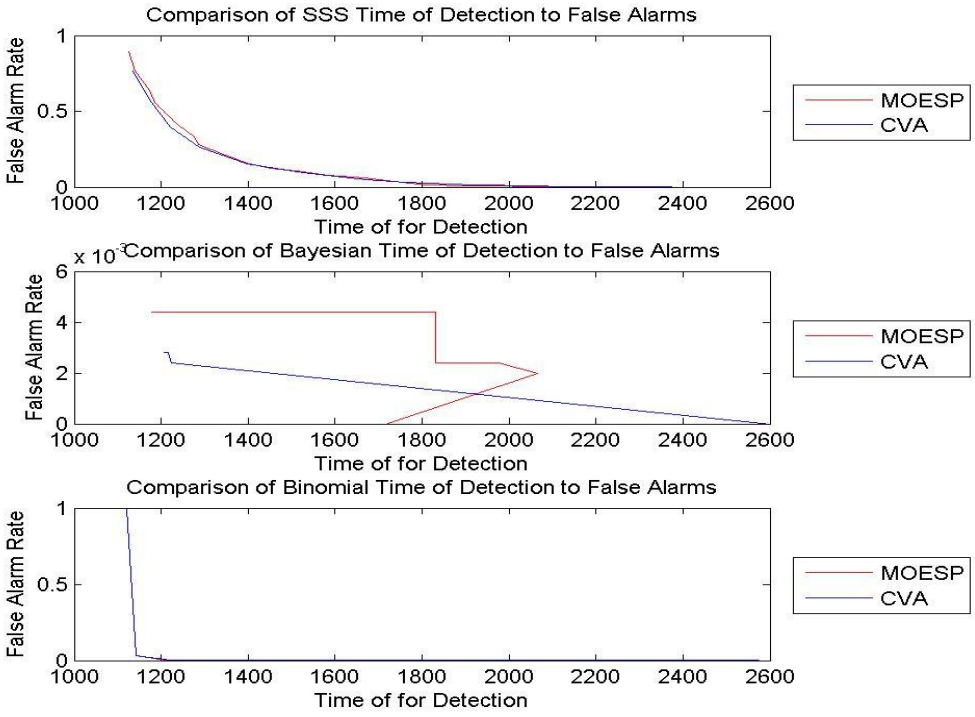


Figure 28: False Alarm rate vs. Detection Time at POD=.25, Attack level 1, MN=.01 and PN=.1



5.2.2 Effect of Process Noise

The second case shows the effect of increased process noise on the predictability of the system. The only differentiator between the experiments that generated Figure 25 with Figure 29 is the size of the process noise in the system. There does not appear to be a change in the Sum of Squares ROC curves, while it is clear that the Bayesian Heuristic method and Binomial method suffer from the increase in process noise. The Bayesian Heuristic Method maximizes the false alarm rate but decreases in the true positive rate whereas the Binomial method has the highest true positive rate but increases in the false alarm rate. The final decision will depend on the operator preference. Figure 30 shows the False Alarm vs Detection Time chart with the detection time and rate being comparable to the previous two experiments. The reason the MOESP line is missing from the Bayesian Heuristic method is the false alarm rate is zero for the every trial (There were 25 threshold levels for the Bayesian Heuristic starting at 0 and continuing to 1 with a threshold level at each .04). The zero false alarm rates mean that the threshold levels for the MOESP could potentially be lowered even more and to continue to increase the true detection rate while maintaining the zero false alarm rate.

Figure 29: ROC of POD=.75 at Attack level 1, MN=.01 and PN=5

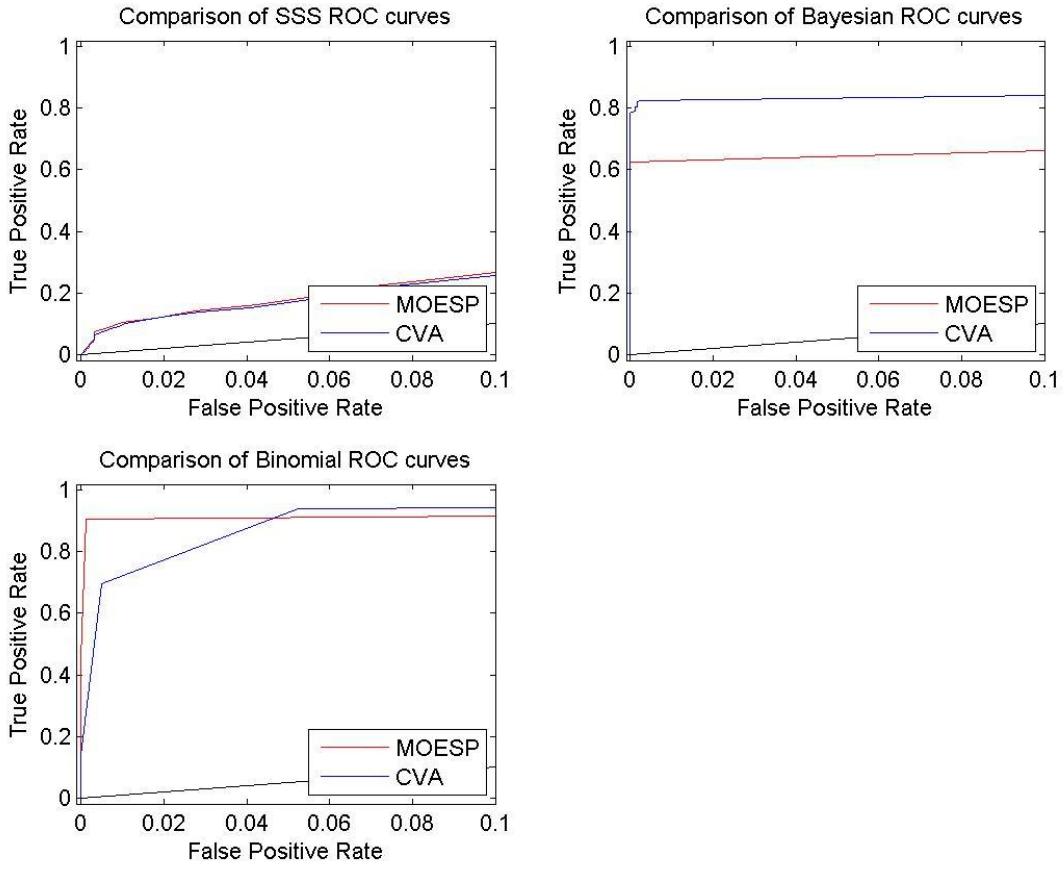
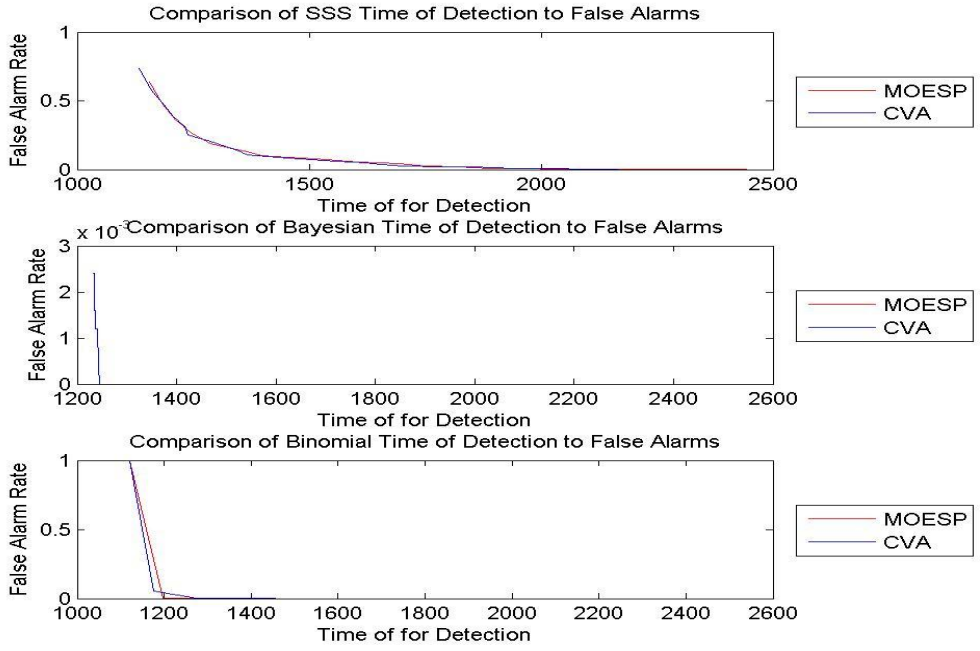


Figure 30: False Alarm rate vs. Detection Time at POD=.75, Attack level 1, MN=.01 and PN=5



5.2.3 Effect of Measurement Noise

The effect of measurement noise can be measured by comparing Figure 25 and Figure 31. All other variables during the experiment are held constant. Increasing the measurement noise has no impact on the SSS method, a small negative impact in the true positive rate for the Bayesian method, and a large negative impact on the Binomial Method. The best predictor for this case is the Bayesian Heuristic; however the best estimation method changes to MOESP. The False Alarm rate vs. Detection Time chart shown in Figure 32 show that the detection time does not change significantly between most of the previous graphs, however the scale for the false alarms does increase. The Binomial method has been similar for each of these charts shown, suggesting that it is consistent in its time for detection.

Figure 31: ROC at POD=.75, Attack level 1, MN=1 and PN=.1

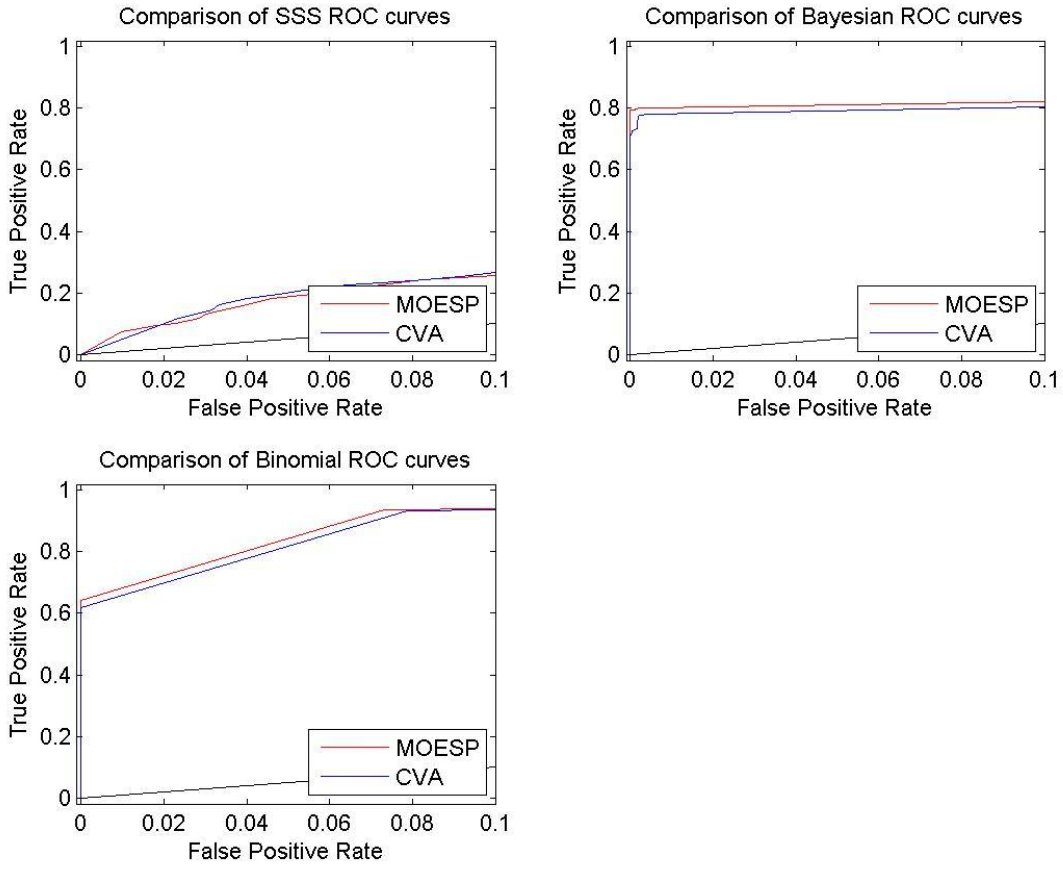
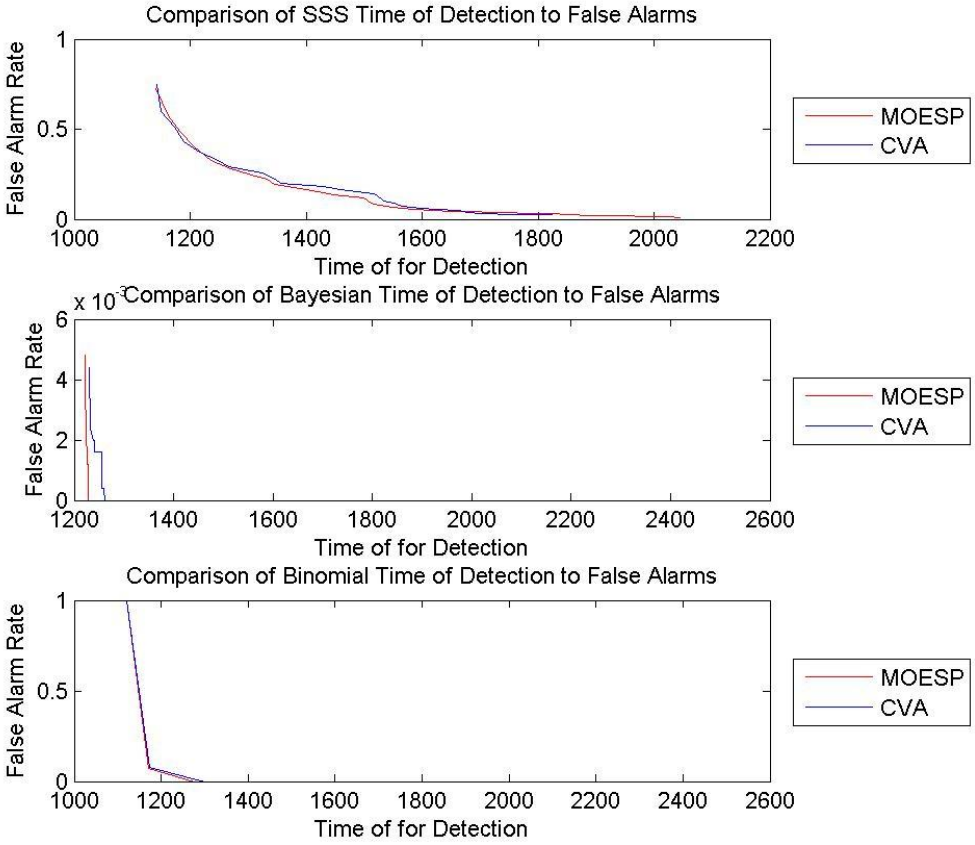


Figure 32: False Alarm rate vs. Detection Time at POD=.75, Attack level 1, MN=1 and PN=.1



5.2.4 Worst Case Scenario

The ROC curves shown in Figure 33 show the worst case scenario simulated, with the independent variables set so that the process noise and measurement noise were at their highest values, the POD was set to 75% and the attack was to the smallest change in parameters (K_p= .13 originally, attack sets K_p= .163). At these independent variable levels the detection algorithm runs into major problems differentiating between the attack-free state and the attack state. The detection algorithm would not be applied to a system that operates at these conditions because there would not be any value added. In Figure 34 the Bayesian Heuristic method experiences a large increase in the detection time for this experiment, most likely due to the poor detection rates. Figure 35 shows the real time estimate of a single parameter for both cases. It is very difficult to determine from the real time estimates the presence of an attack.

Figure 33: ROC at POD=.75, Attack level 3, MN=1 and PN=5

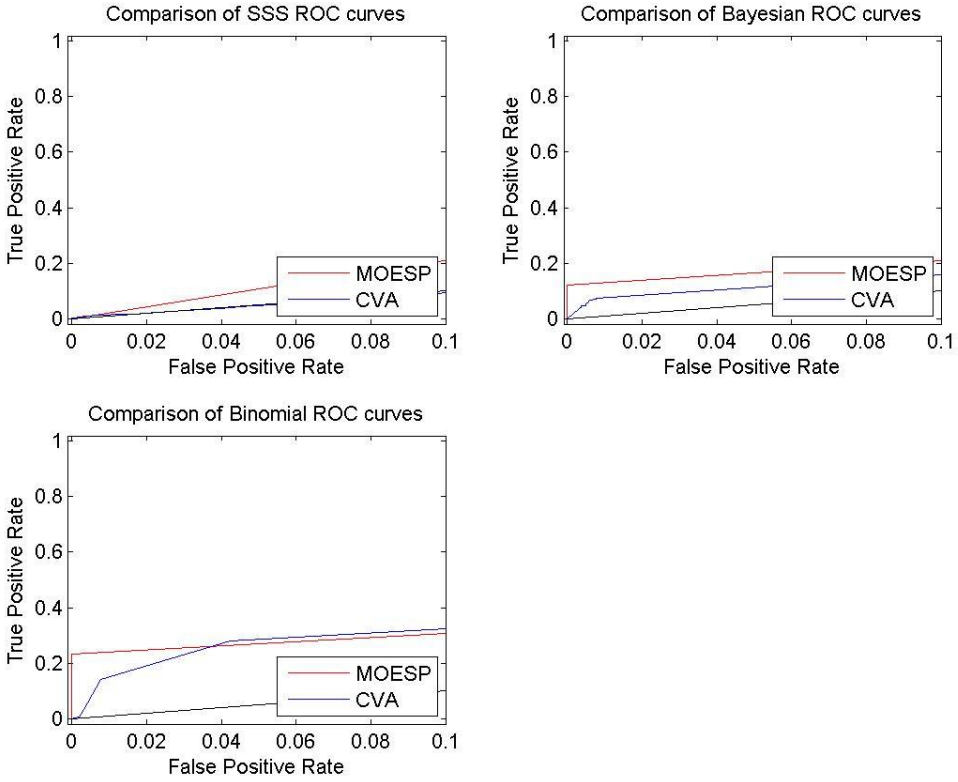


Figure 34: False Alarm rate vs. Detection Time at POD=.75, Attack level 1, MN=1 and PN=5

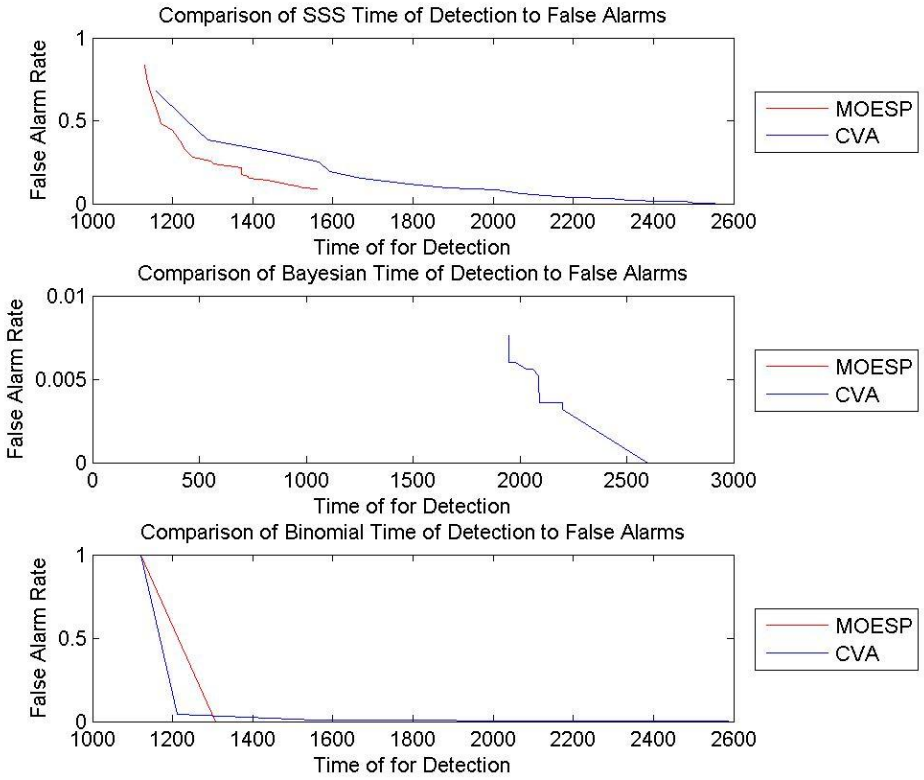
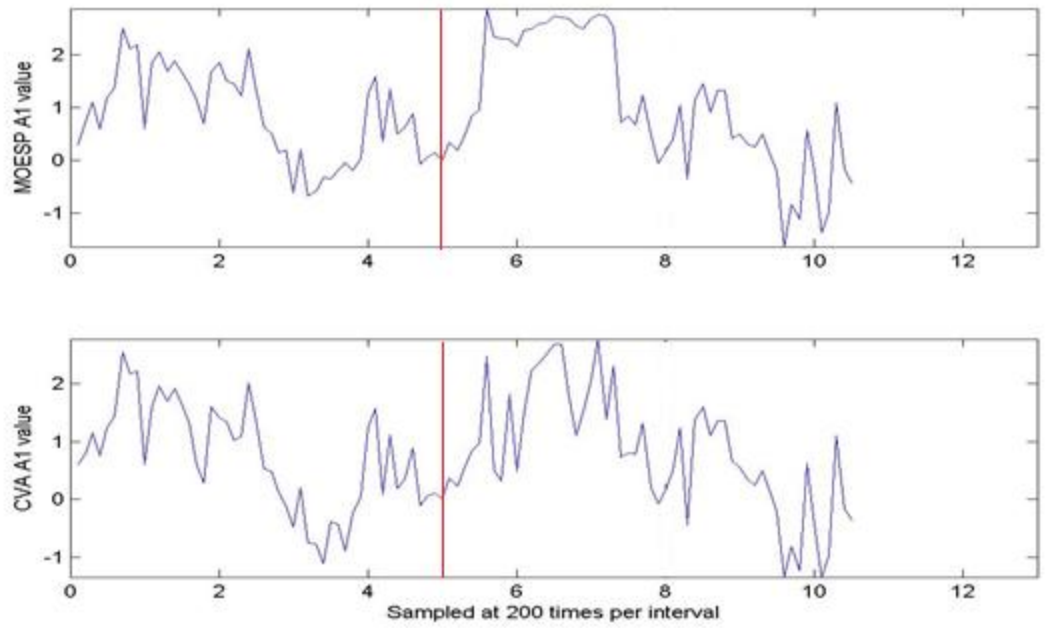


Figure 35: Real time estimates of parameter values a_1 and a_2 in the state matrix A at POD=.75, Attack level 3, MN=1 and PN=5



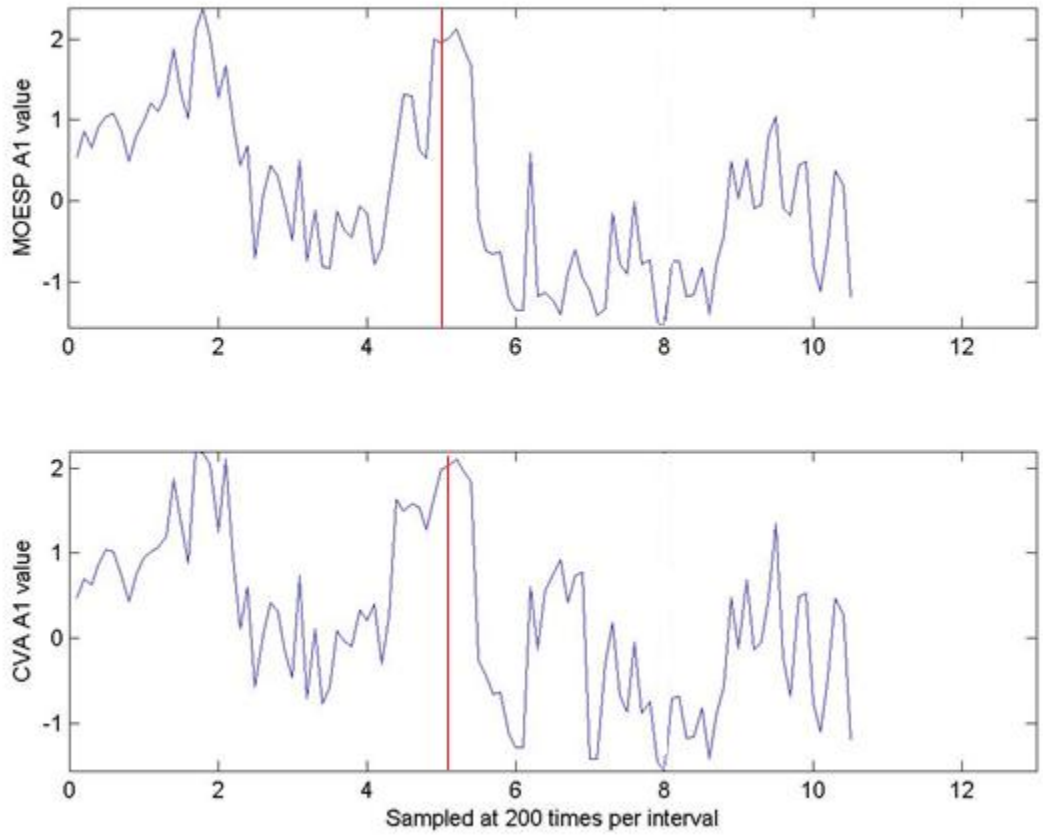
5.2.5 Overall Statistics

Overall, excluding the SSS method which was inefficient across all trials, the Bayesian Heuristic method performed best 13% of the time, the Binomial Model performed best 58% of the time, and 29% of the time the curves were inconclusive and subject to operator judgment based on their values. The Bayesian Heuristic method typically performed better in the low variance cases while the Binomial method performed better in high variance cases. This split suggests that the selection of either method is application specific, and the tuning and configuration of the models could have a significant effect. The MOESP estimation method performed best in 65% of the cases, the CVA performed best in 19% of the cases, and 16% of the time the curves were inconclusive and subject to operator judgment based on their values. MOESP was the best estimation method for this system; however the CVA curve was only marginally worse than the MOESP curve in most cases. In cases of low noise there was a noticeable difference in the values of the parameters before and after the attack; this is shown in Figure 36. The detection time was fairly constant among the combinations similarity algorithms and estimation methods, reducing the metrics of evaluation to the false alarm rate and the true detection rate. The remaining performance ROC charts and False Alarm rate vs Detection Time charts an can be found in Appendixes 10.2.3 and Appendix 10.2.4

Percentage of success	MOESP	CVA	Undetermined
False Alarm Reductionist	77%	19%	4%
Missed Detection Reductionist	63%	27%	10%

Percentage of success	SSS	Bayesian	Binomial	Undetermined
False Alarm Reductionist	0	58%	42%	5.50%
Missed Detection Reductionist	0	8%	92%	5.50%

Figure 36: Real time estimates of parameter values a_1 and a_2 in the state matrix A at POD=.75, Attack level 1, MN=.01 and PN=.1



5.1.4.1 Evaluation from a False Alarm Reductionist Viewpoint

Evaluating the ROC curves from a fault reductionist viewpoint led to conflicting selections. The SS method was immediately ruled out for poor performance. The choice of one similarity algorithm over another was very dependent on the independent trials, the attack size and the estimation method. The Bayesian Heuristic method maintained the lowest false alarm rate 58% of the time but often sacrificed large portions of the true positive rate in the process. Selecting the best similarity method will be application specific and will require testing and tuning. The decision between estimation methods was much more definitive then the decision between similarity algorithms. MOESP obtained values that minimized the false alarm rate 77% of the experiments, where CVA only minimized values 19% of the experiments. Between the four combinations of methods, the difference in detection was not large, so the major tradeoffs should be limited to false alarm rate and true positive rate.

Detailed Breakdown:

False Alarm Reductionist	MOESP	CVA	No difference
SSS	0	0	0
Bayesian	11	4	0
Binomial	7	1	1
No difference	1	0	0

The detailed breakdown of the experiments shows the same behavior of the summarized cases, with the Bayesian Heuristic similarity method and MOESP estimation algorithm performing better than the alternatives.

5.1.4.2 Evaluation from a Missed Detection Reductionist Viewpoint

From the viewpoint of an operator who wants to reduce missed detections, the Binomial method was the clear choice from the available similarity algorithms in 92% of the experiments. The MOESP

estimation method performed better than the CVA estimation method in 63% of the experiments. The Binomial method was the clear choice for maximizing the true positive rate and performed much better than the Bayesian Heuristic method. The percentage of times the MOESP estimation method was better decreased in the missed detection viewpoint, but it was still a better predictor than the CVA estimation method.

Detailed Breakdown:

Missed Detection Reductionist	MOESP	CVA	No difference
SSS	0	0	0
Bayesian	1	2	0
Binomial	15	4	1
No difference	0	1	0

The detailed breakdown shows that the behavior shown in the summarized cases holds for the detailed case. The Binomial Heuristic similarity method and the MOESP estimation algorithm perform better then the alternatives.

6. DISCUSSION

6.1 Results Discussion

The experiments conducted on the two simulated systems showed that in certain cases, Systems Identification Parameter Estimation can provide a method for cyber-attack detection. In the RLC filter system, a combination of KFPE estimation method and the Bayesian Heuristic similarity algorithm were able to detect cyber-attacks on the system that changed the value of a single component reflected in the parameters of the A matrix. In the Fuel Injection system, it is show that under certain conditions, it is possible to use advanced Systems Identification techniques to estimate the real time values of the parameters of an A matrix of a complex system given only a single state of measurements, the inputs and the number of states in the system. These parameters can then be used to determine if an attack is occurring and notify an operator. The best combination of estimation methods and similarity algorithms varied for the Fuel Injection system with the Bayesian Heuristic method performing best from a false alarm reductionist viewpoint and the Binomial method performing best from a missed detection reductionist viewpoint. The MOESP estimation method performed better than the CVA estimation method for both similarity algorithms justifying its use. The independent variable that had the largest impact on the false alarm rate was the POD allowed, however in some experiments where the false alarm rate was already zero it only had a negative impact on the performance by reducing the true positive rate. The process and measurement noise were shown to be large factors in the performance of the algorithms and that different estimation methods reacted differently to increases in them. Increases in the process and measurement noise did not always negatively impact the performance of the estimations methods, a surprising result. It is hypothesized that this affect is either due to the distributions of the historical deviations being too narrow, the estimation methods being over determined, or similarity algorithms better recognizing small changes in the system parameters. Changes in the measurement noise resulted in larger changes in the simulated systems than expected, despite the smaller size. The SSS similarity

algorithm did not perform better than the competing algorithms and should not be considered as a similarity method for either of the systems simulated. The detection time was similar in most combinations of the algorithms and was not a large differentiator between algorithms and methods. It should be used as a secondary metric when deciding between estimation methods and similarity algorithms with the false alarm rate and true positive rate being the primary metrics. Because of the variable performance across the experiments, any implementation of this detection algorithm in the field will require large amounts of operational data to determine the applicability and fine tune the performance of the algorithm. Of the estimation methods, the MOESP and CVA algorithms will apply to the most different types of systems because of reduced number of assumptions made during the estimation process. However they can perform at levels comparable to the LLSE and KFPE in certain experiments.

6.2 Implementation Considerations

The detection algorithms presented carry with them multiple factors that guide their implementation. The first such factor is the false alarm rate of the detection algorithm. The false alarm rate must be minimized to prevent operator fatigue and to reduce system downtime. Otherwise, the operators of the system will begin to devalue the events that are detected and the revenue loss resulting from system downtime will grow infeasible. For this reason, it is suggested that the detection algorithm be tuned on actual operation data to detect on average one event per month. Additionally, when examining the data, the false alarm detections tended to disappear shortly after being detected, while the true positive detections tended to persist for longer durations. The temporality of the false alarm detections suggests that the size of the deviation windows could continue to be increased to reduce the number of false alarms detected by the algorithm.

It should be implicitly obvious from the variability of the results that any detection algorithms implement will require tuning in order to be beneficial. The optimal algorithms for each system will vary along with the optimal estimation methods. Systems that do not have directly extractable measurements

will preclude themselves from the LLSE and KFPE algorithms, but will still have the MOESP and CVA algorithms available to them. Before implementation, the system in question should be tested to determine the optimal estimation method and similarity algorithm to maximize the true detections, minimize the number of false alarms, and minimize the detection time. After implementation, data should be continually collected during field use to help fine tune the set of algorithms chosen.

6.2.1 Assumptions

Many of the estimation methods and similarity algorithms used in this Thesis carry assumptions along with them that are violated. The following section describes the assumptions that are violated during this Thesis.

1. The Bayesian update algorithm used as a similarity method assumes that the variables $X_1, X_2, \dots, X_n$ that P(A) is conditioned on be mutually independent of each other. When the systems are converted into companion form the parameters of the original matrix become intertwined in the parameters of the output matrix and the individual effect of a component can affect all the non-static values in the companion form matrix. Therefore an attack on a single component can affect all the non-static values of the companion form matrix. The independence assumption is violated because the estimated values of the parameters in the companion form matrix, which are variables $X_1, X_2, \dots, X_n$ during the Bayesian update algorithm, are not independent of each other.

2. The Subspace Systems Identification algorithms make the assumption that the noise in the system(w and v) is uncorrelated with the input u. In closed loop systems, the input to the system is usually to correct for noise in the previous time steps. Implementing Subspace Systems Identification algorithms on closed-loop systems introduces bias into the estimation of the dynamic system [23].

3. The Gaussian model used to fit the deviations of the long run attack-free model may be biased in a direction, producing not optimal standard deviations

All of the violations of the assumptions that occur in the detection algorithm potentially decrease the predictive power of their algorithms. If violating the assumption had a negative impact upon the predictive power, then the negative impact would be present in the ROC curve generated for that algorithm and the detection time for that algorithm.

6.2.2 Cyber-Attack Isolation

Once a cyber-attack has been detected, the source of the attack will need to be identified. Unfortunately, when the detection algorithm transforms the system into companion form it is very difficult to return to the designed realization of the state space system. Being able to return to the designed realization would allow an operator to examine the parameters of that realization and the components that make up each parameter and potentially solve a system of equations to determine the values of each component that makes up the parameters. Without the ability to return to the designed realization the options for cyber-attack isolation are limited. Fault isolation techniques are potentially applicable and a survey of current fault detection techniques is shown in [29]. The operator can also characterize the behavior of each parameter in the companion form based on changes of a single component, but this process quickly becomes tedious if the parameters are interrelated. The number of components that require characterizing can be reduced by eliminating components that are not computer controlled, however even after this the number of components could still be very large. More research is need into this subject to identify the source cyber-attacks quicker and restore the systems under attack to normal operation faster.

6.3 Potential Improvements to the Detection Algorithm

In many of the experiments simulated, multiple combinations of estimation methods and similarity algorithms provided approximately the same predictive power for the false alarm and true detection rates. To potentially improve the total predictive power of the detection algorithm, an ensemble forecast could be created from the different combinations of estimation methods and similarity

algorithms. The ensemble forecast would reduce the modeling error created during the algorithm formulations and possibly produce a better classifier then any single model could achieve. The predictive methods selected to be part of the ensemble forecast would be application specific, with the most robust methods being selected.

A Cumulative sum function (CUSUM) of each sequential observation would eliminate the need for a sliding window to filter the results. The models would be able to have a sequential value at each time step that would determine whether the system is experiencing an event or operating under normal conditions. The percentage of deviations variable would become unnecessary with this function, and the threshold for the CUSUM function would become the new independent variable. The benefit of the CUSUM method is the ability to detect cyber-attacks that manipulate the system with a small bias in one direction more effectively than a sliding window.

Another potential improvement to the algorithm only applies to the Subspace Identification algorithms. For practical purposes, the detection algorithm made the assumption that the B, C, and D matrices remained constant during operation of the system. Relaxing this constraint, specifically on the C matrix, precludes the use of the LLSE method and KFPE method, but can allow us a larger number of parameters to check for deviations. The additional parameters can be seen in the description of the Companion form matrix in 2.3.1. The LLSE and KFPE method no longer apply to systems that relax this assumption because they require a constant C matrix to extract the states of the system. However, for systems that are already rank deficient in the C matrix, using the values of the C matrix will offer more predictive power to the detection algorithm.

7. EVENT CLASSIFICATION

The detection algorithm created in this Thesis does not differentiate between faults, cyber-attacks or other causes of disturbances in the physical system. Once an event has been flagged the event must be categorized to determine if it is the result of a fault, human error, cyber-attack, or other causes. Cyber-attacks on control systems have previously been discovered through the process of elimination, the operators of the system ruled out hardware faults, human errors, software bugs, among other causes before finally considering cyber-attacks. Fault Modeling classifies changes in the parameter of a dynamic system as multiplicative faults and attempts to use fault isolation algorithms to locate the source of the fault [29]. However the fault isolation algorithms are rarely comprehensive of all the components in the system and often focused solely on hardware components. Human errors can be checked via logs; however, in systems with large numbers of computer controlled components, it can be difficult to locate the source of the error. Software bugs take extensive testing to find and often will only present themselves under certain conditions. Overall, by the time all the previous methods have been ruled out, the system may have been taken offline for a long period of time or will have been operating under the influence of a cyber-attack causing damage or sub-optimal performance. This Thesis proposes that under some conditions cyber-attacks are more likely and should be examined with greater urgency. A streamlined process to help operators label events as potential cyber-attacks was created to help operators recover from events quickly.

7.1 Disutility of Events

Assuming a cyber-attacker has compromised the system, they will attempt to maximize the damage performed while still remaining undetected. The Disutility of an event is defined as the importance of the current functions of the physical system. It is posited that cyber-attacks are more likely during critical system functions, maximizing the damage done to either the system itself, or the output of the system. The Disutility can be also be defined as the difficultly of shutting the affected system down,

maximizing the amount of time cyber-attack is allowed to affect the machine. Periods of time where the physical system will be forced to continue operating are considered prime targets for a cyber-attack. Lastly, the Disutility can be defined as the length of the downtime for the system and the amount of time the system requires to restart. Attacks that can disable multiple systems for long periods of time can effectively reduce the performance to suboptimal levels. If an event occurs during any of these critical periods of time, the probability increases that the event can be classified as a cyber-attack.

7.2 Situational Context

The Situational Context of an event is defined as the set of circumstances and conditions surrounding the event. Any recent modifications or changes made to the system are potentially responsible for the change in system dynamics rather than a cyber-attack. These modifications could include changes in configuration, operator changes, recent maintenance, among other reasons. The presence of one of these modifications may be the underlying reason for the change in system behavior, making the likelihood the change of behavior is a result of a cyber-attack less likely. The Situational Context can also mean the absence of changes to the system including large times since the last fault (which can be modeled as a Gamma distribution) or large times between maintenance. The absence of a change could increase of decrease the probability depending on the case.

7.3 Human Error

Human error accounts for a large portion of the failure events in the operation of systems. Human error is defined as unintentional changes to the parameters controlling the designed and configured operation of a system. Cyber-attacks and insider attacks on systems mimic human error failures because of the similar way they manipulate parameters. Logs of parameter changes by operators should be kept to rule out cyber-attacks originating from outside the perimeter and internal checks should prevent insiders from manipulating the parameters to extreme values.

7.4 Previous Events

The operators of the system likely keep logs of previous events and faults that can provide information about the probability of an attack. If the current event mimics a previous event, then it could possibly be a repetition of the same event. If the cause of this previous event was determined to be unrelated to a cyber-attack, then it suggests that the cause of the current event is less likely to be a cyber-attack. Additionally, systems that experience events at increased rates compared to other systems have a reduced a priori probability of being the victims of cyber-attacks. Likewise the opposite is true and additionally if no previous event is similar to the current attack, then it can be considered unique and more likely to be a cyber-attack.

7.5 Cyber-attack Checklist

To account for the previous event classification factors, a checklist has been created to organize the decision making process. The first step in preparing this checklist is to chart the components vulnerable to cyber-attack. Components are considered vulnerable if their operation can be manipulated by commands issued from a computer. This limits the number of systems that the checklist will apply to and increase response time. Prior to an event, the operators should classify all the periods of system operating time into High Risk, Medium High Risk, Medium Low Risk, and Low Risk. Operations receive the High Risk designation if they are both difficult to stop when in progress or take a long time to restart and likely to cause damage to the machine or output if administered inappropriately. Operations receive the Medium High Risk designation if they are likely to cause damage to the machine or output if administered inappropriately, but able to stopped quickly if events are detected and can be restarted quickly. Operations receive the Medium Low Risk designation if they difficult to stop when in progress or take a long time to restart but have minimal lasting effects. All other operations are deemed Low Risk.

When an event is discovered using the detection algorithm, the first step on the checklist is to classify its risk level. On High Risk systems it may be necessary to take the system offline in order to prevent damage. The next step on the checklist is to compare the logs of the system to previous failures and determine if any of the previous failures resemble the current failure. This comparison can help separate the common hardware failure cases from the uncommon failure cases and, if there have been previous cyber-attacks, can potentially identify the current event case faster than normal. The next step on the checklist involves looking at the Situational Context of the system. First the system should be examined for recent changes. If no recent changes can be identified on the checklist, then the absence of changes should be examined. If the Situational Context provides no additional information about the system event, the case is unique, and the system functions are High Risk, then the event should be strongly considered as a potential cyber-attack and the computers controlling the system should be tested for malware and monitored. The checklists likely will have to be tailored for each system to add more detail, or reduce extraneous fields. An example checklist is shown in Figure 37.

Figure 37. Potential Checklist format.

Risk Classification of affected system:

1. High Risk 2. Medium-High Risk 3. Medium-Low Risk 4. Low Risk

Previous Event similarity:

1. Known Cyber-Attack 2. Unknown Event 3. Known not a Cyber-Attack

Possible Human error:

□ Were the system parameters recently modified by an operator?

Situational Context Recent changes:

□ Maintenance

□ Operator

□ Configuration

□ Policy

□ Temperature

□ Computer Software

□ Shift

□ Inputs

□ Power Source

Previous failures:

□ Days since last failure exceeds average?

□ Days since last maintenance exceeds average?

□ Does this system fail more often than other systems?

If the total sum of the Risk Classification rating, Human Error checkmarks, Situational Context checkmarks, and Previous Events checkmarks less than four, strongly consider the possibility of a cyber-attack.

---

8. CONCLUSION

In conclusion, Fault Detection techniques, specifically Systems Identification Parameter Estimation techniques, have shown that they can be valuable methods for detecting cyber-attacks that change the operational parameter of systems. By creating real-time estimates of the parameters of the system and comparing those estimates to the designed and configured parameters, deviations from the norm can be recognized and reported to the operators. The operators can then use the guidelines created in this Thesis to determine a qualitative likelihood that the reported event fits the criteria of a cyber-attack and the correct response to the event. In the two systems simulated, the detection algorithm was able to detect cyber-attacks in non-trivial cases within a reasonable time after the attack. The estimation methods and similarity algorithms were evaluated on each of the simulated systems to determine the most robust methods for implementation. The effects of the percentage of deviations allowed, the process noise, and the measurement noise were explored to determine the applicability of the detection algorithm to different environments. For the RLC filter, the most effective combination of estimation method and similarity algorithm was the Kalman Filter Parameter Estimation method and the Binomial method. For the Fuel Injection system the most effective combination of estimation method and similarity algorithm varied, with the false alarm reductionist selecting the Bayesian Heuristic method and the MOESP method, and the missed detection reductionist selecting the Binomial method and the MOESP method. Differences in the effectiveness of each algorithm on each system have shown the application specific nature of the estimation methods and similarity algorithms. Any implementations of this detection algorithm will have to be tested and tuned to determine the best methods for implementation and to optimize the detection rates, false alarms and detection time. For categorizing events a check list was created that looks at the Situational Disutility, the Situational context, the Human Error component, and the Previous Failures to give a qualitative approach to determining the likelihood the event can be classified as a cyber-attack.

9. REFERENCES

[1] Farwell, James P., and Rafal Rohozinski. "Stuxnet and the future of cyber war."*Survival* 53.1 (2011): 23-40.

[2] Kerr, Paul K., John Rollins, and Catherine A. Theohary. "The Stuxnet Computer Worm: Harbinger of an Emerging Warfare Capability." Congressional Research Service, Library of Congress, 2010.

[3] Slay, Jill, and Michael Miller. "Lessons learned from the maroochy water breach." *Critical Infrastructure Protection*. Springer US, 2007. 73-82.

[4] Cárdenas, Alvaro A., et al. "Attacks against process control systems: risk assessment, detection, and response." *Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security*. ACM, 2011.

[5] Horowitz, Barry, and Kate Pierce. "Application of Dynamic System Models and State Estimation Technology to the Cyber Security of Physical Systems."

[6] Gonzalez, Carlos R. Aguayo, and Jeffrey H. Reed. "Power fingerprinting in SDR & CR integrity assessment." *Military Communications Conference, 2009. MILCOM 2009. IEEE*. IEEE, 2009.

[7] Wang, Jiang, Angelos Stavrou, and Anup Ghosh. "HyperCheck: A hardware-assisted integrity monitor." *Recent Advances in Intrusion Detection*. Springer Berlin Heidelberg, 2010. Welch, Greg, and Gary Bishop. "An introduction to the Kalman filter." (1995).

[8] Skormin, Victor, et al. "Customized normalcy profiles for the detection of targeted attacks." *Applications of Evolutionary Computation*. Springer Berlin Heidelberg, 2012. 487-496.

[9] Baskiotis, C., J. Raymond, and A. Rault. "Parameter identification and discriminant analysis for jet engine machanical state diagnosis." *Decision and Control including the Symposium on Adaptive Processes, 1979 18th IEEE Conference on*. Vol. 18. IEEE, 1979.

[10] Moseler, Olaf, and Rolf Isermann. "Application of model-based fault detection to a brushless DC motor." *Industrial Electronics, IEEE Transactions on* 47.5 (2000): 1015-1020.

[11] Liu, Xiang-Qun, et al. "Fault detection and diagnosis of permanent-magnet DC motor based on parameter estimation and neural network." *Industrial Electronics, IEEE Transactions on* 47.5 (2000): 1021-1030.

[12] Johnson, Michael L., and Lindsay M. Faunt. "Parameter estimation by least-squares methods." *Methods Enzymol* 210.1 (1992): 37.

[13] Welch, Greg, and Gary Bishop. "An introduction to the Kalman filter." (1995).

[14]De Callafon, Raymond A., et al. "General realization algorithm for modal identification of linear dynamic systems." *Journal of engineering mechanics* 134.9 (2008): 712-722.

[15]Dempster, Arthur P., Nan M. Laird, and Donald B. Rubin. "Maximum likelihood from incomplete data via the EM algorithm." *Journal of the Royal Statistical Society. Series B (Methodological)* (1977): 1-38.

[16] Favoreel, Wouter, Bart De Moor, and Peter Van Overschee. "Subspace state space system identification for industrial processes." *Journal of Process Control* 10.2 (2000): 149-155.

[17] Jacobs, Oliver Louis Robert, and O. L. R. Jacobs. *Introduction to control theory*. Vol. 2. Oxford: Oxford University Press, 1993.

[18] Walker, David M. "Parameter estimation using Kalman filters with constraints."*International Journal of Bifurcation and Chaos* 16.04 (2006): 1067-1078.

[19] Van Overschee, Peter, and Bart De Moor. "N4SID: Subspace algorithms for the identification of combined deterministic-stochastic systems." *Automatica* 30.1 (1994): 75-93.

[20] Jones, Rick A., and Barry Horowitz. "System-Aware Cyber Security." *Information Technology: New Generations (ITNG), 2011 Eighth International Conference on*. IEEE, 2011.

[21]Abrams, Marshall and Weiss, Joe. "Malicious Control System Cyber Security Attack Case Study-Maroochy Water Services, Australia" (2007)

[22] Bersimis, Sotiris, Stelios Psarakis, and John Panaretos. "Multivariate statistical process control charts: an overview." *Quality and Reliability Engineering International* 23.5 (2007): 517-543.

[23] Qin, S. Joe. "An overview of subspace identification." *Computers & chemical engineering* 30.10 (2006): 1502-1513.

[24] Verhaegen, Michel. "Application of a subspace model identification technique to identify LTI systems operating in closed-loop." *Automatica* 29.4 (1993): 1027-1040.

[25] Larimore, Wallace E. "Canonical variate analysis in identification, filtering, and adaptive control." *Decision and Control, 1990., Proceedings of the 29th IEEE Conference on*. IEEE, 1990.

[26] Van Overschee, Peter, and Bart De Moor. "A unifying theorem for three subspace system identification algorithms." *Automatica* 31.12 (1995): 1853-1864.

[27] Seely, Bill "Edison Program Servo Loop Homework Problem: P2 Regulator Design Problem" 13 September 2011

[28] Holzhauer, D. F., Seely, Bill "Hydraulic Servo Regulator Model Bode Response." 11 October 2006. Matlab file.

[29] Hwang, Inseok, et al. "A survey of fault detection, isolation, and reconfiguration methods." *Control Systems Technology, IEEE Transactions on* 18.3 (2010): 636-653.

[30] Basseville, Michele, Maher Abdelghani, and Albert Benveniste. "Subspace-based fault detection algorithms for vibration monitoring." *Automatica* 36.1 (2000): 101-109.

10. APPENDICES

10.1 Simulation Parameters

10.1.1 RLC filter

Trials=25

Simulations=100

Sampling rate =180hz

Covariance matrix P reset time = 200 iterations

Settling time=100 iterations

Sliding window size =400

Prediction period = 400 iterations

Long run simulation size =10000 iterations

Non attack period = 1000 iterations

Attack period =1000 iterations

Time between estimations = 20 iterations

Probability of attack = 1/144000

Capacitor = .0001 F

Set resistance = 4 ohms

Inductor= .0704 H

Q= v^2*[.1 0 ; 0 2.5] where w=process noise

M=w^2*[.1 0; 0 .1] where v=measurement noise

Voltage source= 10V AC 60hz

10.1.2 Fuel Injection system

Trials=25

Simulations=100

Sampling rate =10hz

Settling time=100 iterations

Sliding window size =200

Prediction period = 400 iterations

Long run simulation size =10000 iterations

Non attack period = 1000 iterations

Attack period =1000 iterations

Time between estimations = 20 iterations

Probability of attack = 1/1000000

Q= v^2*5x5 identity matrix where w=process noise

M=w^2 where v=measurement noise

Probability of attack = 1/144000

Input is noisy constant with a mean = 420liters/sec

10.2 Performance Charts

10.2.1 RLC ROC charts.

List of ROC charts in order. The types can be found on pages 27 and 28

Type 1 Attack 1

Type 1 Attack 2

Type 1 Attack 3

Type 2 Attack 1

Type 2 Attack 2

Type 2 Attack 3

Type 3 Attack 1

Type 3 Attack 2

Type 3 Attack 3

Type 4 Attack 1

Type 4 Attack 2

Type 4 Attack 3

Type 5 Attack 1

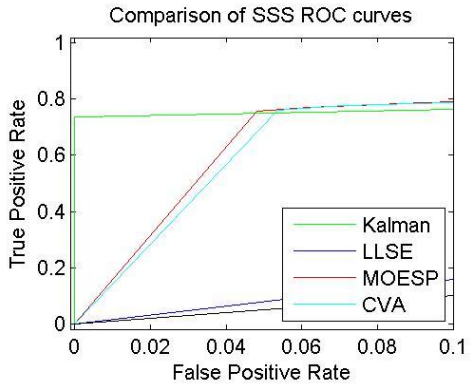
Type 5 Attack 2

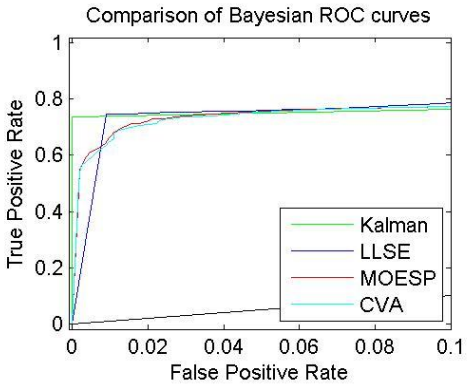
Type 5 Attack 3

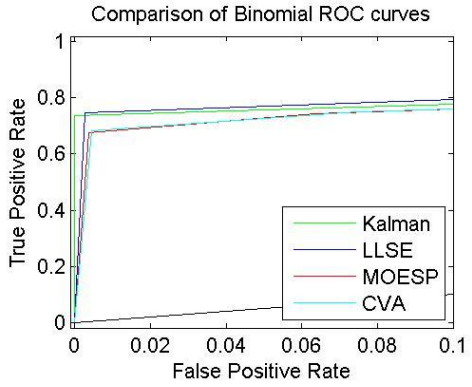
Type 6 Attack 1

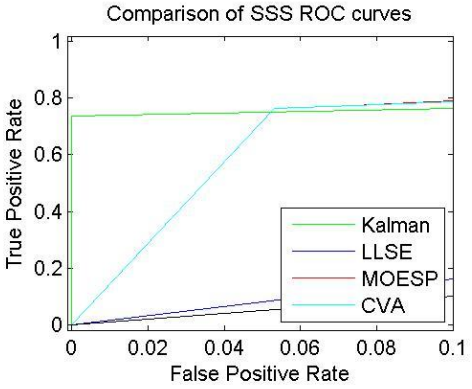
Type 6 Attack 2

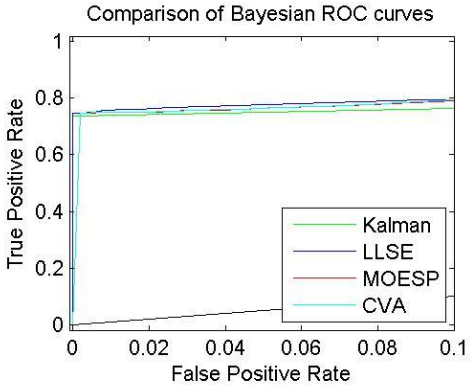
Type 6 Attack 3

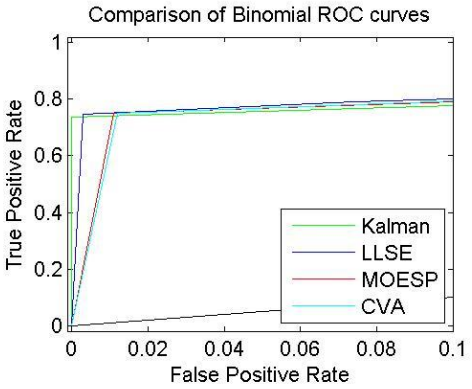

Comparison of SSS ROC curves
True Positive Rate
False Positive Rate
Kalman
LLSE
MOESP
CVA

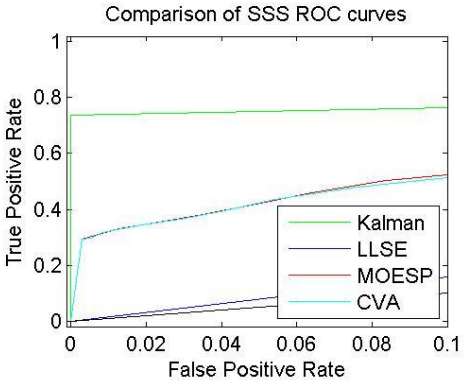

Comparison of Bayesian ROC curves
True Positive Rate
False Positive Rate
Kalman
LLSE
MOESP
CVA

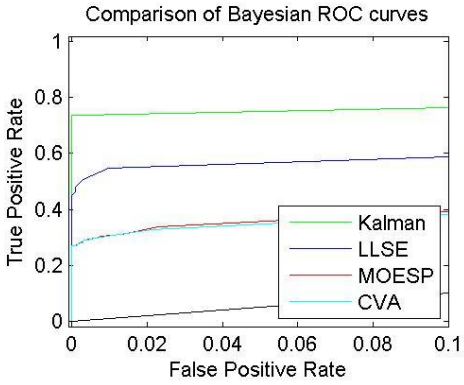

Comparison of Binomial ROC curves
True Positive Rate
False Positive Rate
Kalman
LLSE
MOESP
CVA

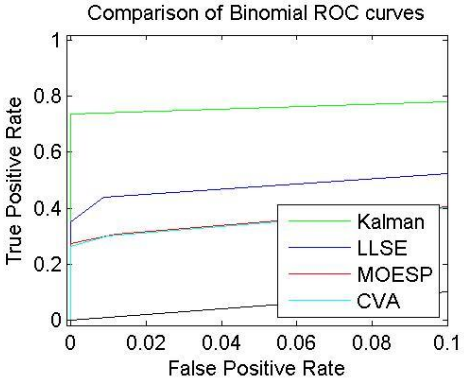

Comparison of SSS ROC curves
True Positive Rate
False Positive Rate
Kalman
LLSE
MOESP
CVA

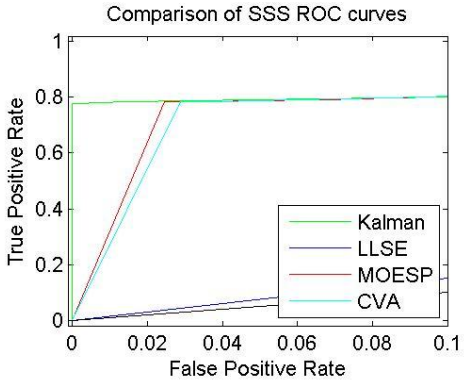

Comparison of Bayesian ROC curves
True Positive Rate
False Positive Rate
Kalman
LLSE
MOESP
CVA

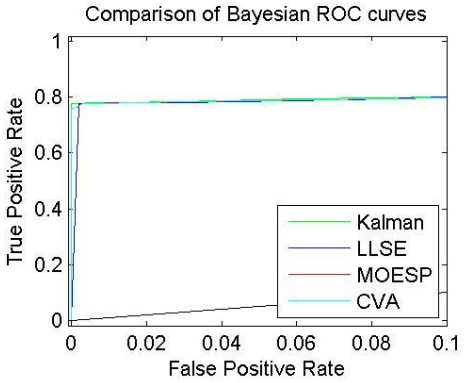

Comparison of Binomial ROC curves
True Positive Rate
False Positive Rate
Kalman
LLSE
MOESP
CVA

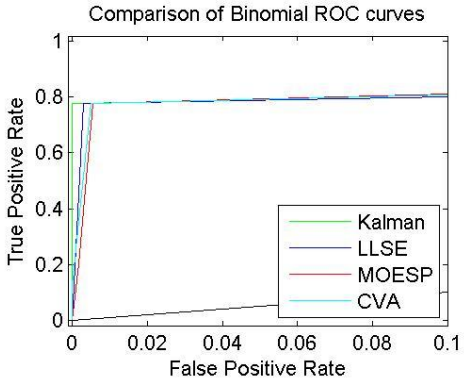

Comparison of SSS ROC curves
True Positive Rate
False Positive Rate
Kalman
LLSE
MOESP
CVA

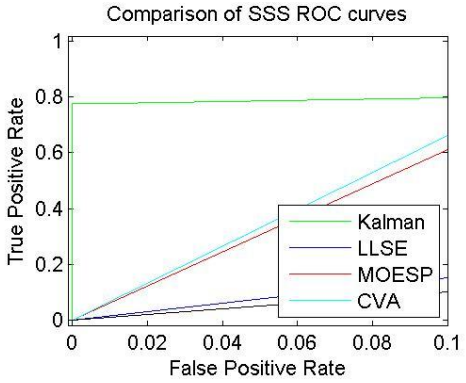

Comparison of Bayesian ROC curves
True Positive Rate
False Positive Rate
Kalman
LLSE
MOESP
CVA

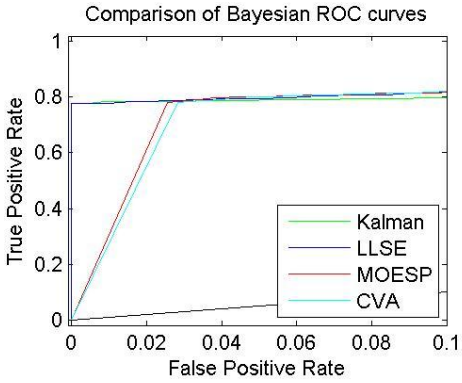

Comparison of Binomial ROC curves
True Positive Rate
False Positive Rate
Kalman
LLSE
MOESP
CVA

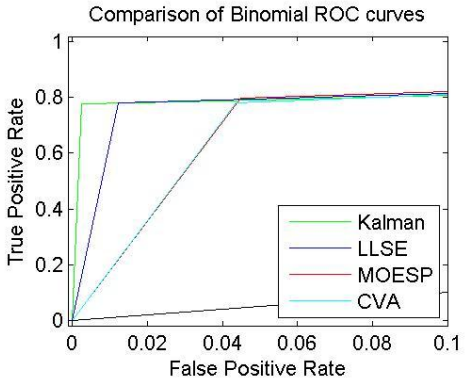

Comparison of SSS ROC curves
True Positive Rate
False Positive Rate
Kalman
LLSE
MOESP
CVA

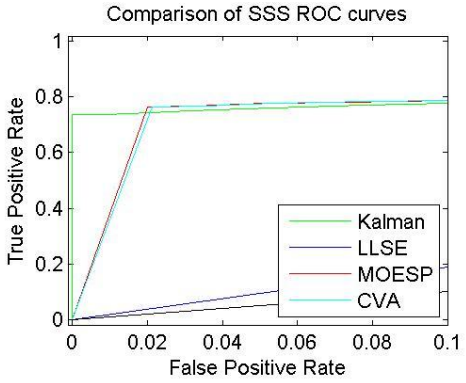

Comparison of Bayesian ROC curves
True Positive Rate
False Positive Rate
Kalman
LLSE
MOESP
CVA

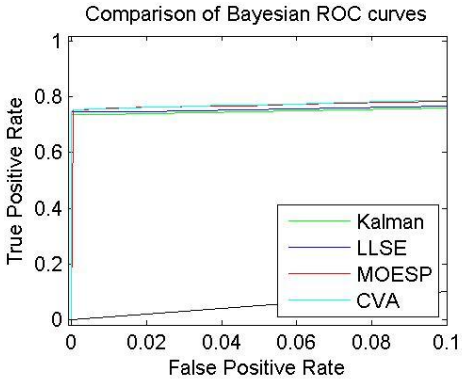

Comparison of Binomial ROC curves
True Positive Rate
False Positive Rate
Kalman
LLSE
MOESP
CVA

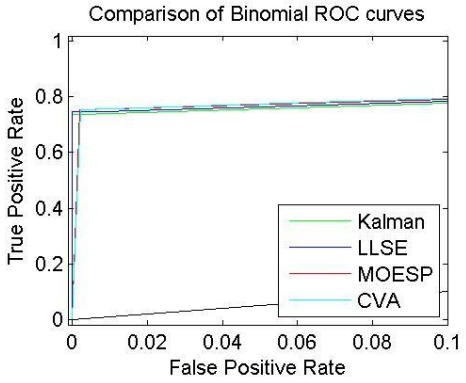

Comparison of SSS ROC curves
True Positive Rate
False Positive Rate
Kalman
LLSE
MOESP
CVA

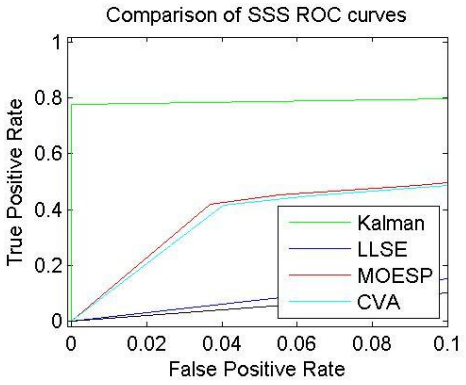

Comparison of Bayesian ROC curves
True Positive Rate
False Positive Rate
Kalman
LLSE
MOESP
CVA

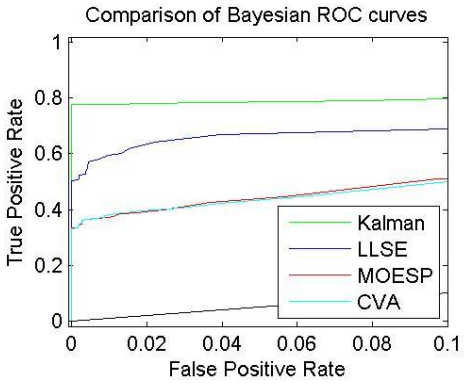

Comparison of Binomial ROC curves
True Positive Rate
False Positive Rate
Kalman
LLSE
MOESP
CVA

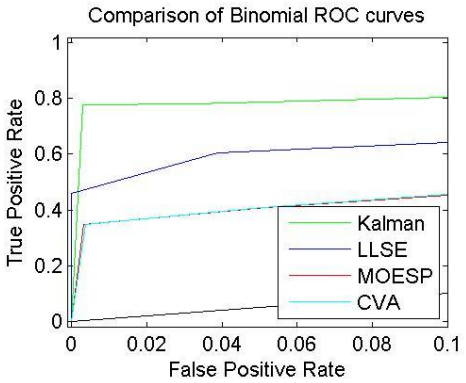

Comparison of SSS ROC curves
True Positive Rate
False Positive Rate
Kalman
LLSE
MOESP
CVA

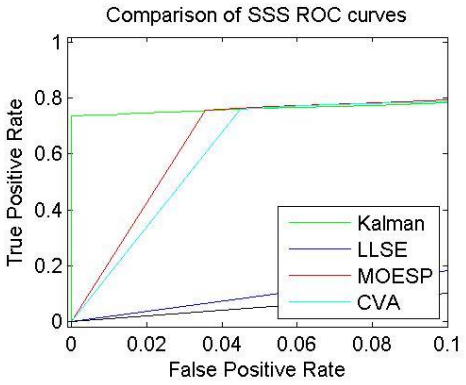

Comparison of Bayesian ROC curves
True Positive Rate
False Positive Rate
Kalman
LLSE
MOESP
CVA

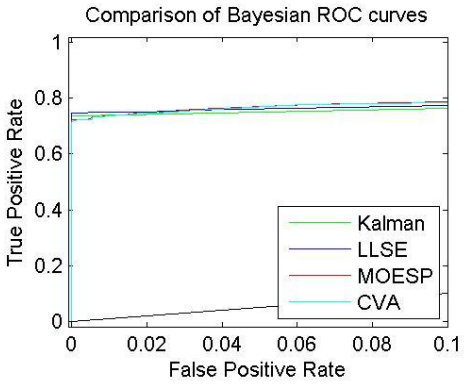

Comparison of Binomial ROC curves
True Positive Rate
False Positive Rate
Kalman
LLSE
MOESP
CVA

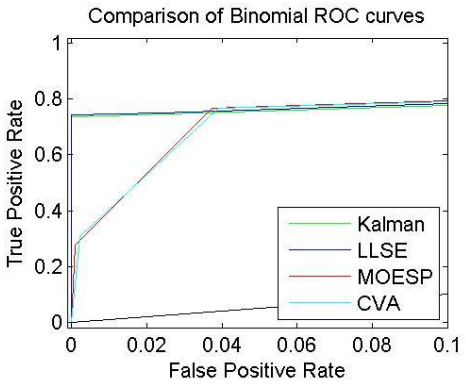

Comparison of SSS ROC curves
True Positive Rate
False Positive Rate
Kalman
LLSE
MOESP
CVA

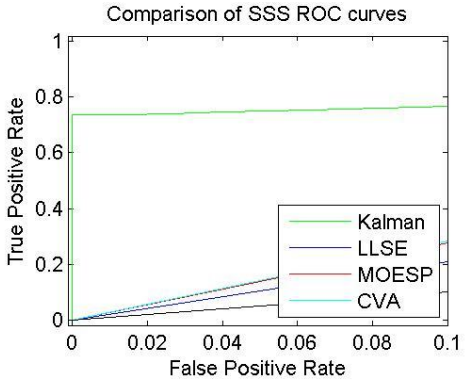

Comparison of Bayesian ROC curves
True Positive Rate
False Positive Rate
Kalman
LLSE
MOESP
CVA

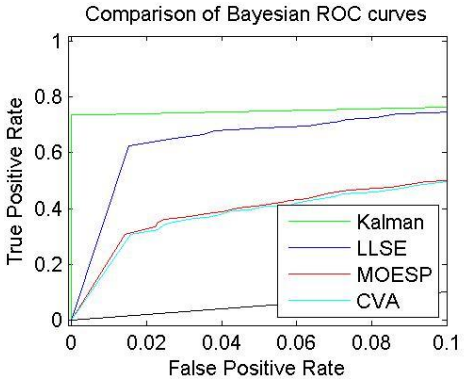

Comparison of Binomial ROC curves
True Positive Rate
False Positive Rate
Kalman
LLSE
MOESP
CVA

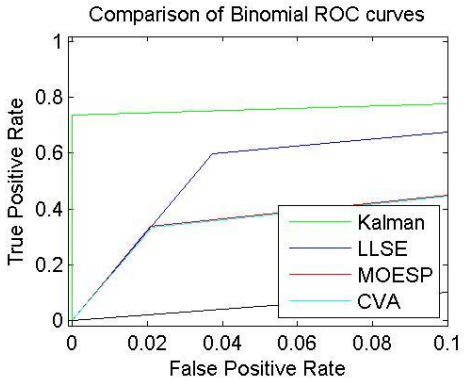

Comparison of SSS ROC curves
True Positive Rate
False Positive Rate
Kalman
LLSE
MOESP
CVA

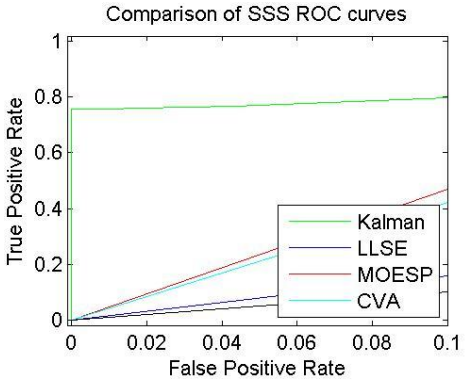

Comparison of Bayesian ROC curves
True Positive Rate
False Positive Rate
Kalman
LLSE
MOESP
CVA

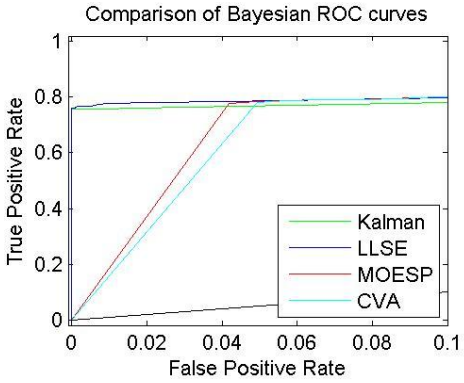

Comparison of Binomial ROC curves
True Positive Rate
False Positive Rate
Kalman
LLSE
MOESP
CVA

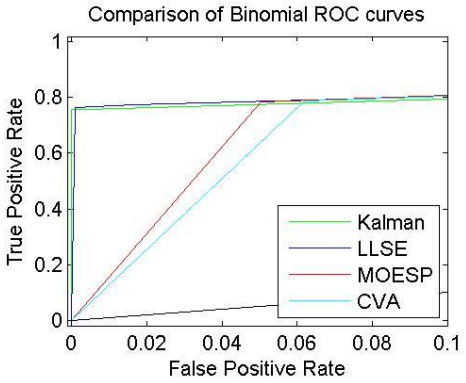

Comparison of SSS ROC curves
True Positive Rate
False Positive Rate
Kalman
LLSE
MOESP
CVA

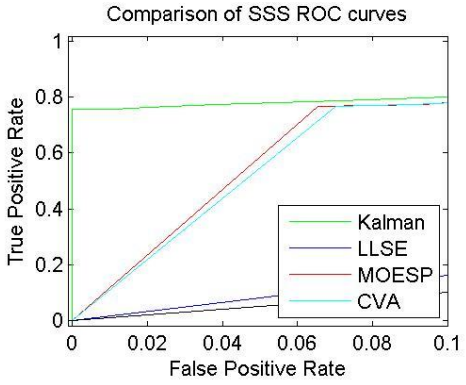

Comparison of Bayesian ROC curves
True Positive Rate
False Positive Rate
Kalman
LLSE
MOESP
CVA

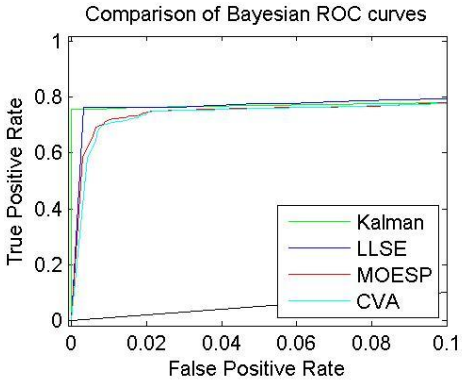

Comparison of Binomial ROC curves
True Positive Rate
False Positive Rate
Kalman
LLSE
MOESP
CVA

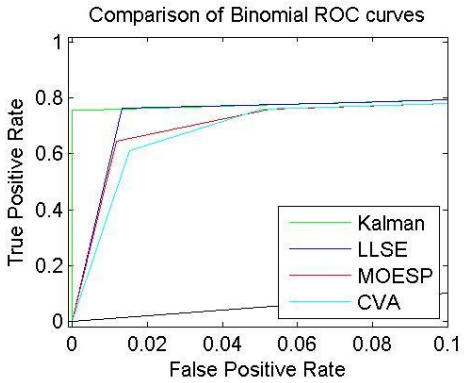

Comparison of SSS ROC curves
True Positive Rate
False Positive Rate
Kalman
LLSE
MOESP
CVA

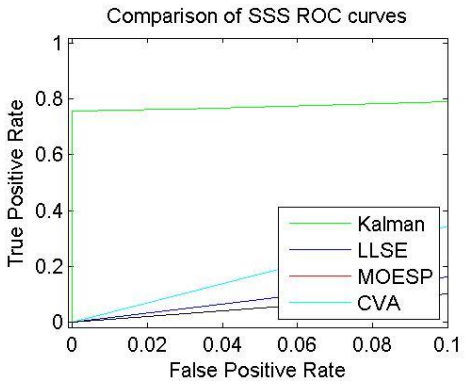

Comparison of Bayesian ROC curves
True Positive Rate
False Positive Rate
Kalman
LLSE
MOESP
CVA

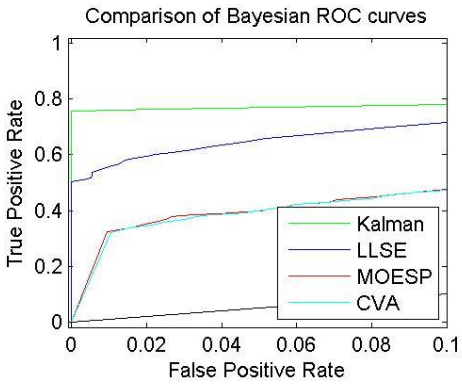

Comparison of Binomial ROC curves
True Positive Rate
False Positive Rate
Kalman
LLSE
MOESP
CVA

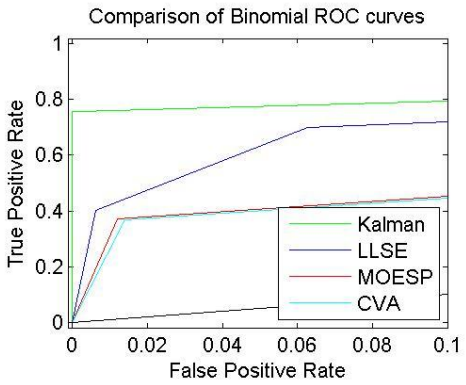

Comparison of SSS ROC curves
True Positive Rate
False Positive Rate
Kalman
LLSE
MOESP
CVA

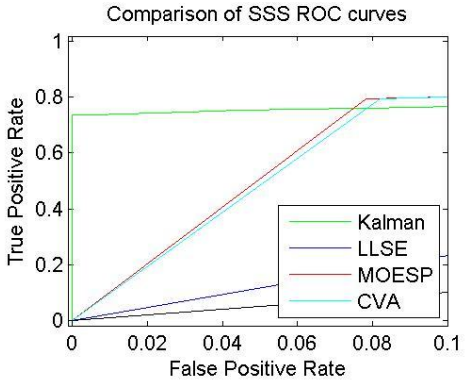

Comparison of Bayesian ROC curves
True Positive Rate
False Positive Rate
Kalman
LLSE
MOESP
CVA

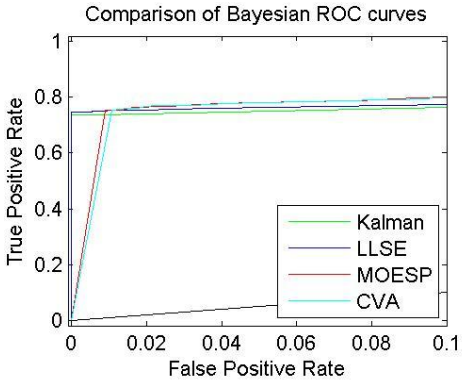

Comparison of Binomial ROC curves
True Positive Rate
False Positive Rate
Kalman
LLSE
MOESP
CVA

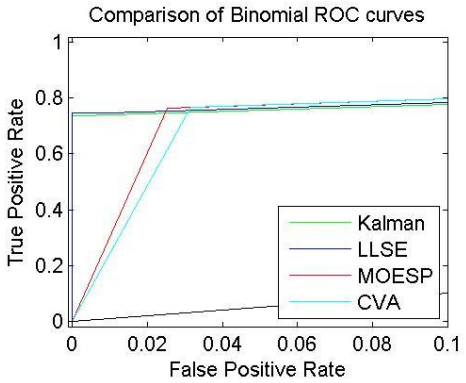

Comparison of SSS ROC curves
True Positive Rate
False Positive Rate
Kalman
LLSE
MOESP
CVA

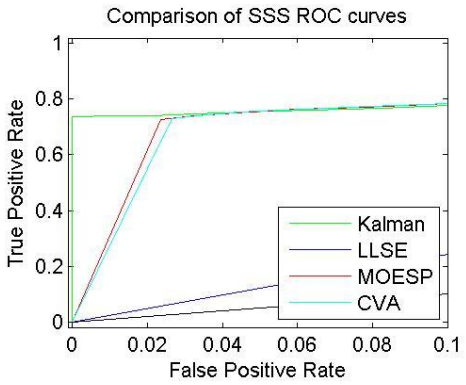

Comparison of Bayesian ROC curves
True Positive Rate
False Positive Rate
Kalman
LLSE
MOESP
CVA

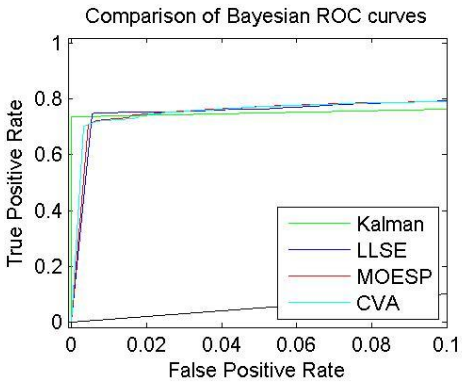

Comparison of Binomial ROC curves
True Positive Rate
False Positive Rate
Kalman
LLSE
MOESP
CVA

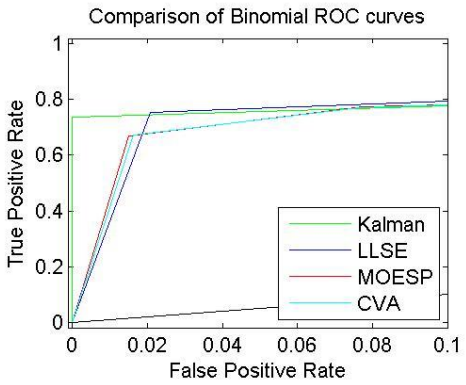

Comparison of SSS ROC curves
True Positive Rate
False Positive Rate
Kalman
LLSE
MOESP
CVA

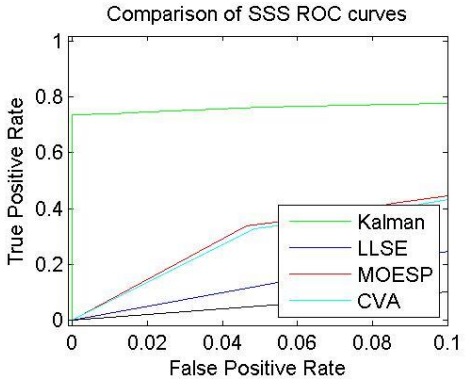

Comparison of Bayesian ROC curves
True Positive Rate
False Positive Rate
Kalman
LLSE
MOESP
CVA

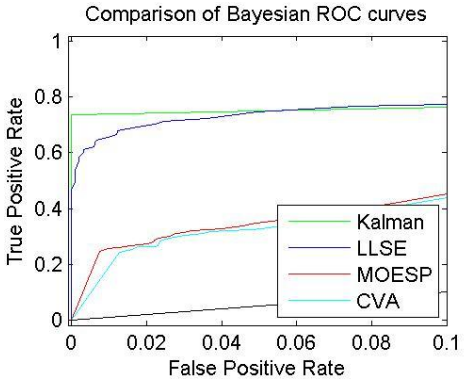

Comparison of Binomial ROC curves
True Positive Rate
False Positive Rate
Kalman
LLSE
MOESP
CVA

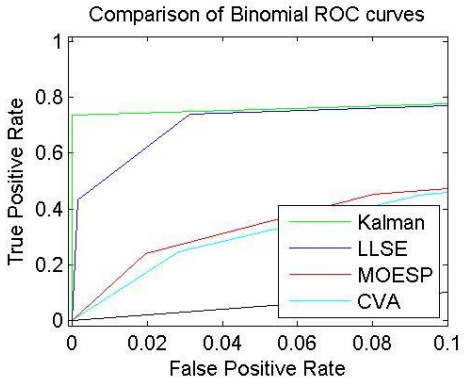

Comparison of SSS ROC curves
True Positive Rate
False Positive Rate
Kalman
LLSE
MOESP
CVA

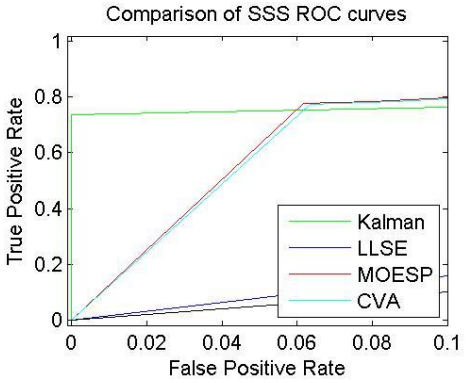

Comparison of Bayesian ROC curves
True Positive Rate
False Positive Rate
Kalman
LLSE
MOESP
CVA

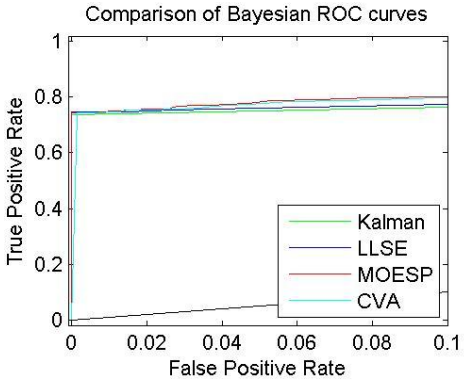

Comparison of Binomial ROC curves
True Positive Rate
False Positive Rate
Kalman
LLSE
MOESP
CVA

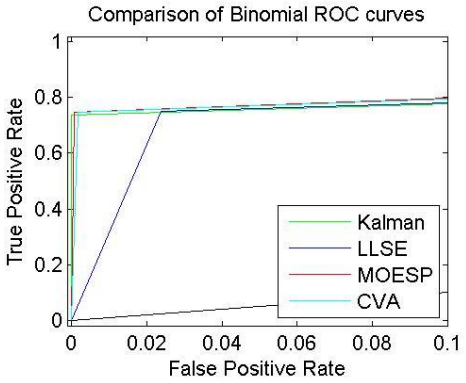

Comparison of SSS ROC curves
True Positive Rate
False Positive Rate
Kalman
LLSE
MOESP
CVA

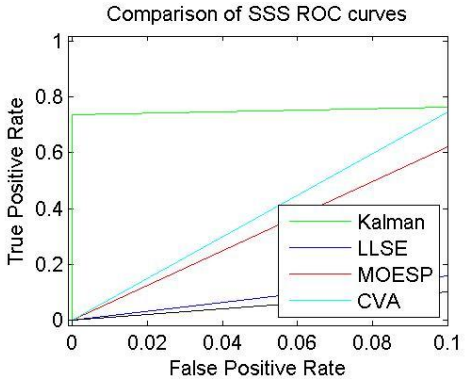

Comparison of Bayesian ROC curves
True Positive Rate
False Positive Rate
Kalman
LLSE
MOESP
CVA

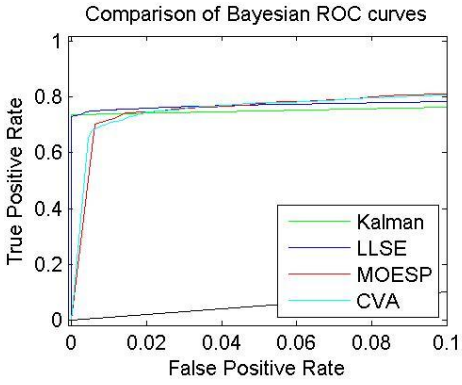

Comparison of Binomial ROC curves
True Positive Rate
False Positive Rate
Kalman
LLSE
MOESP
CVA

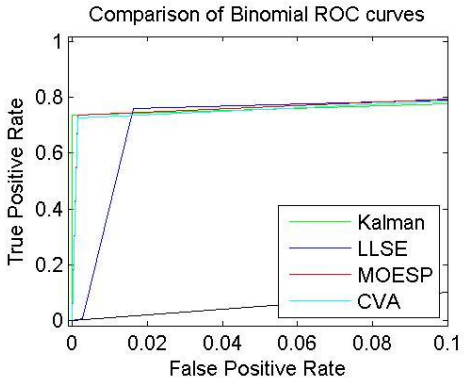

Comparison of SSS ROC curves
True Positive Rate
False Positive Rate
Kalman
LLSE
MOESP
CVA

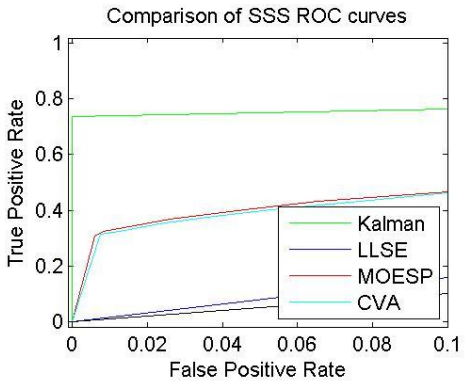

Comparison of Bayesian ROC curves
True Positive Rate
False Positive Rate
Kalman
LLSE
MOESP
CVA

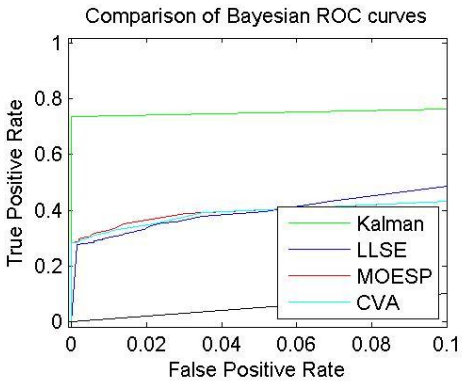

Comparison of Binomial ROC curves
True Positive Rate
False Positive Rate
Kalman
LLSE
MOESP
CVA

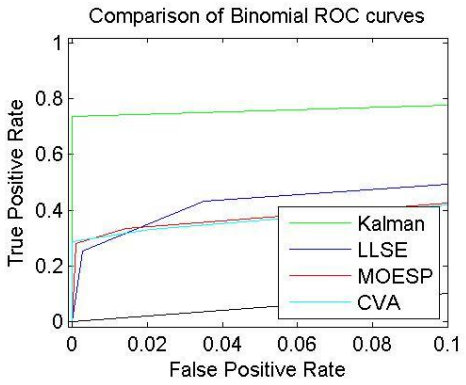

Comparison of SSS ROC curves
True Positive Rate
False Positive Rate
Kalman
LLSE
MOESP
CVA


Comparison of Bayesian ROC curves
True Positive Rate
False Positive Rate
Kalman
LLSE
MOESP
CVA


Comparison of Binomial ROC curves
True Positive Rate
False Positive Rate
Kalman
LLSE
MOESP
CVA


Comparison of SSS ROC curves
True Positive Rate
False Positive Rate
Kalman
LLSE
MOESP
CVA


Comparison of Bayesian ROC curves
True Positive Rate
False Positive Rate
Kalman
LLSE
MOESP
CVA


Comparison of Binomial ROC curves
True Positive Rate
False Positive Rate
Kalman
LLSE
MOESP
CVA

10.2.2 RLC False Alarm rate vs Detection Time Charts.

List of False Alarm rate vs. Detection Time charts in order. The types can be found on pages 27 and 28

Type 1 Attack 1

Type 1 Attack 2

Type 1 Attack 3

Type 2 Attack 1

Type 2 Attack 2

Type 2 Attack 3

Type 3 Attack 1

Type 3 Attack 2

Type 3 Attack 3

Type 4 Attack 1

Type 4 Attack 2

Type 4 Attack 3

Type 5 Attack 1

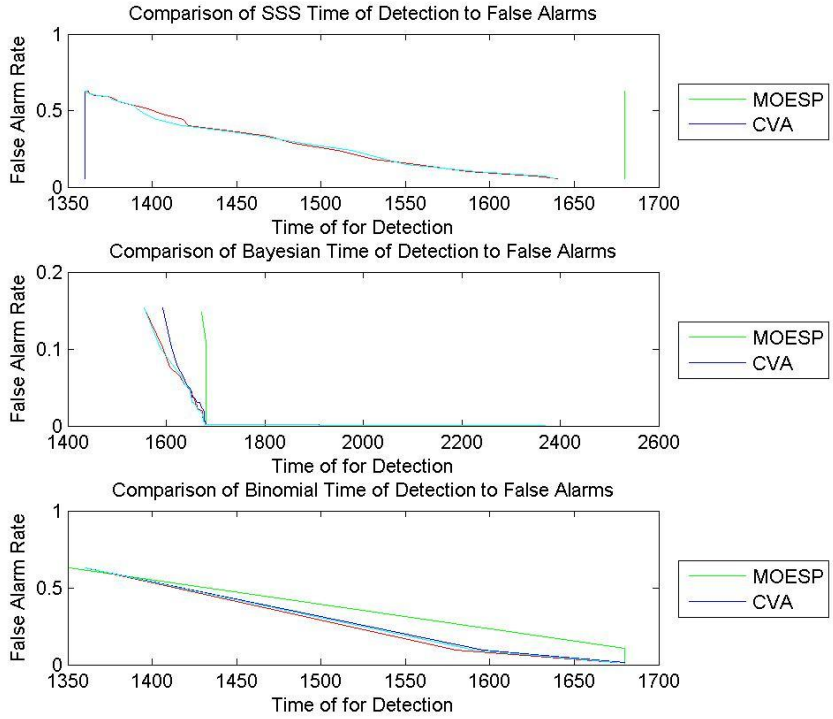
Type 5 Attack 2

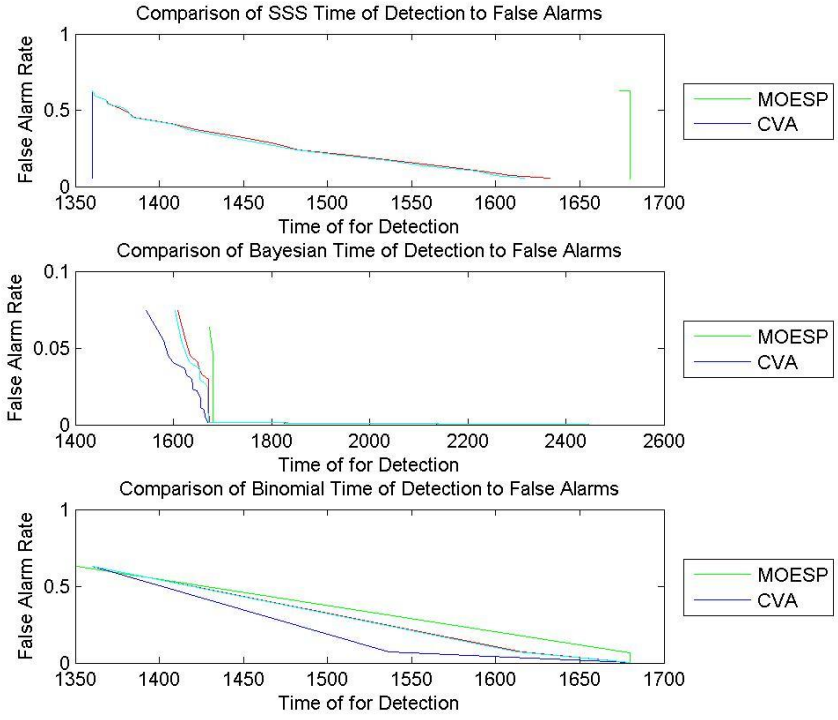
Type 5 Attack 3

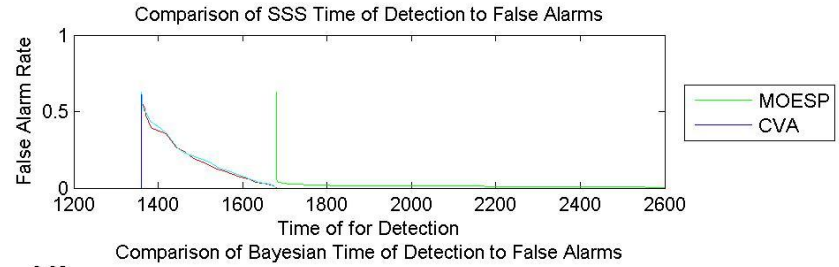
Type 6 Attack 1

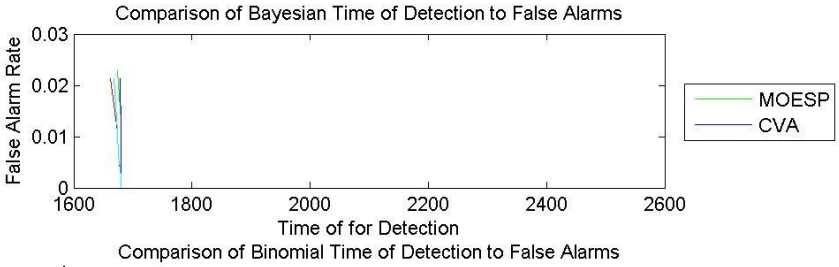
Type 6 Attack 2

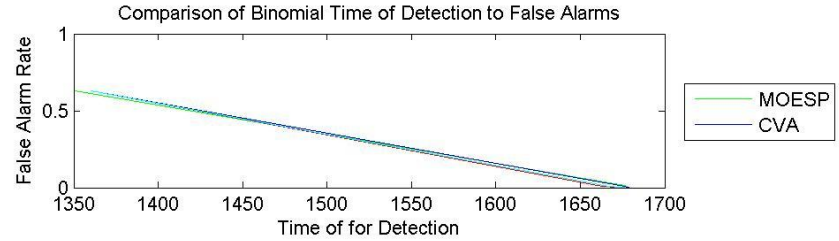
Type 6 Attack 3

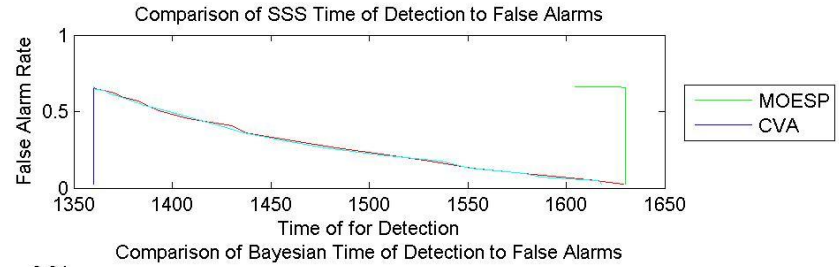

Comparison of SSS Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA
Comparison of Bayesian Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA
Comparison of Binomial Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA

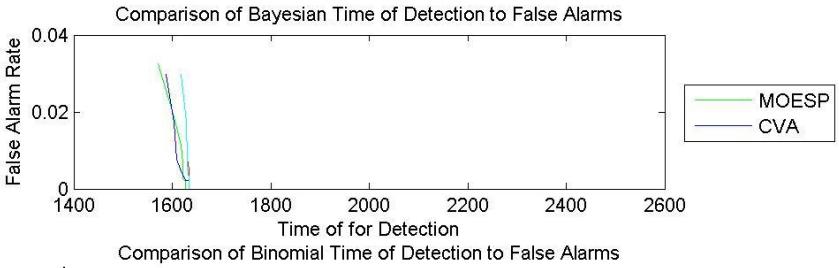

Comparison of SSS Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA
Comparison of Bayesian Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA
Comparison of Binomial Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA

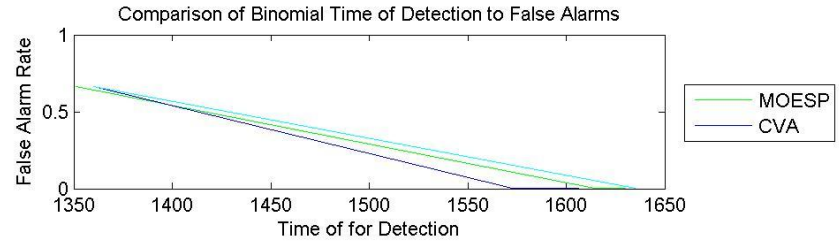

Comparison of SSS Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA

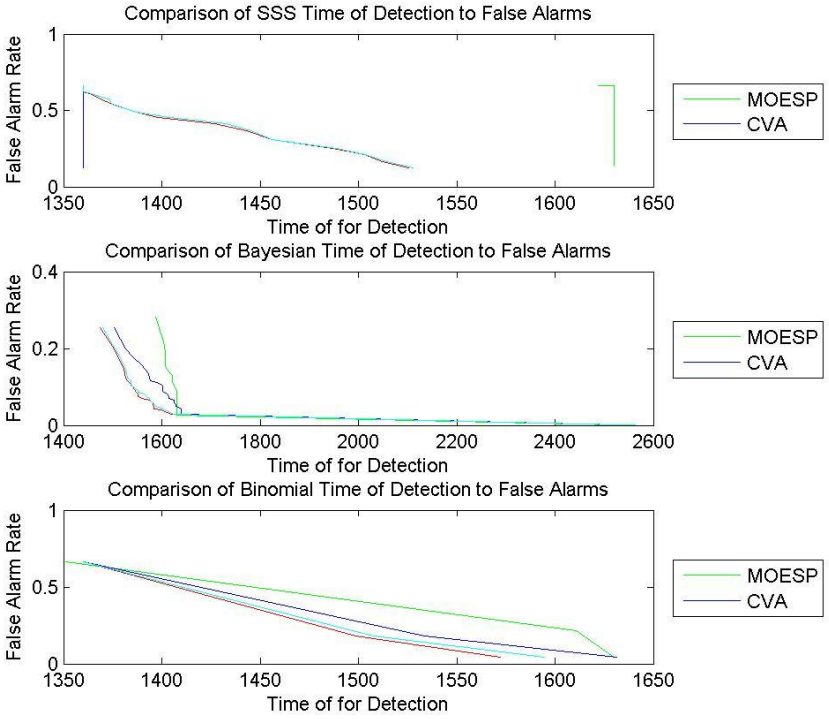

Comparison of Bayesian Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA

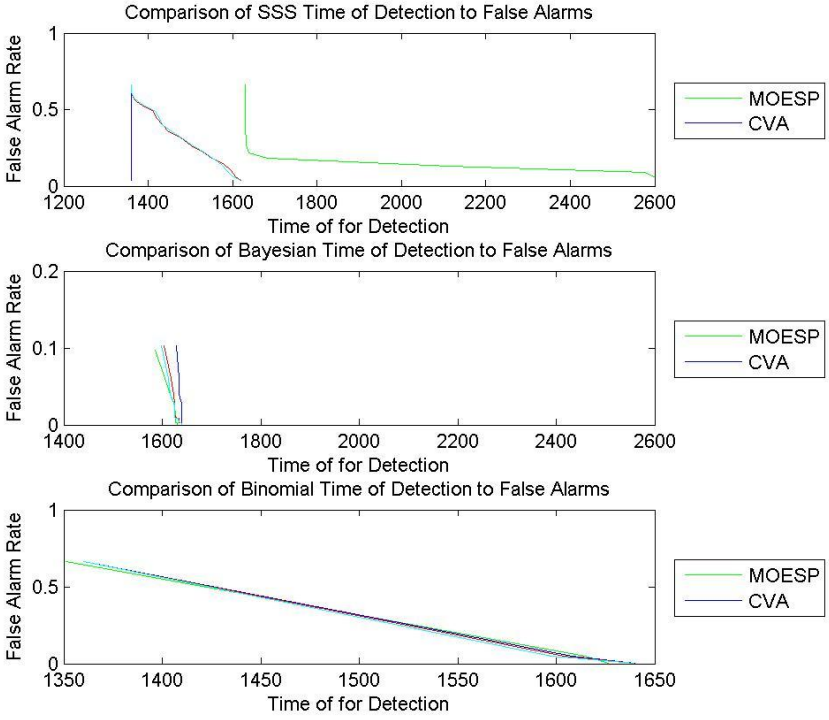

Comparison of Binomial Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA

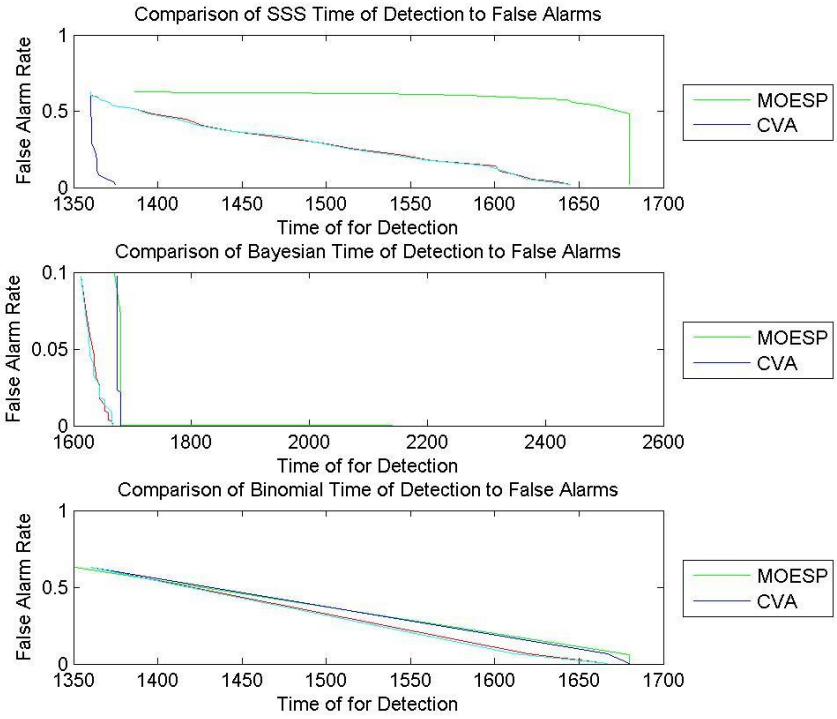

Comparison of SSS Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA

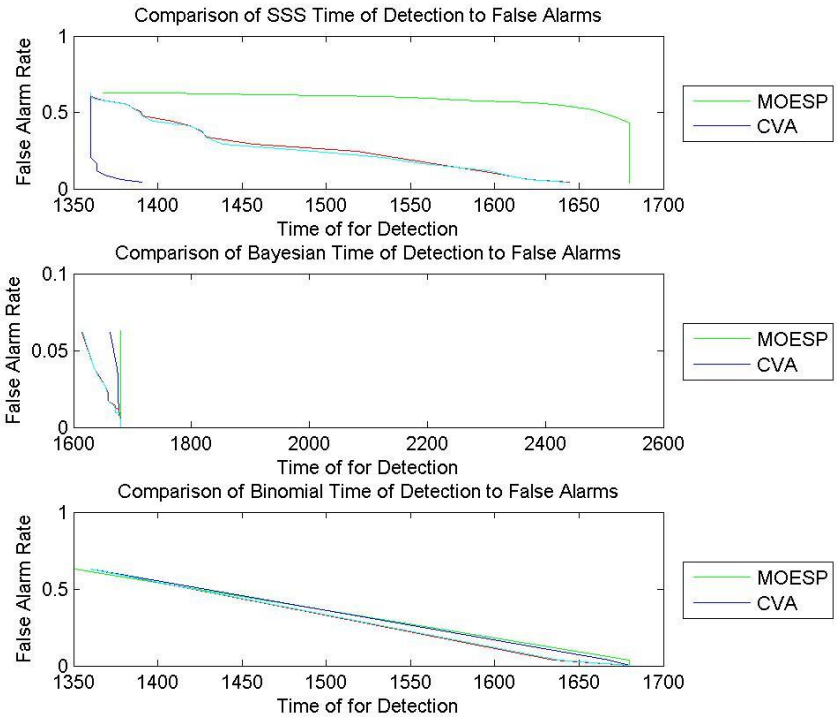

Comparison of Bayesian Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA

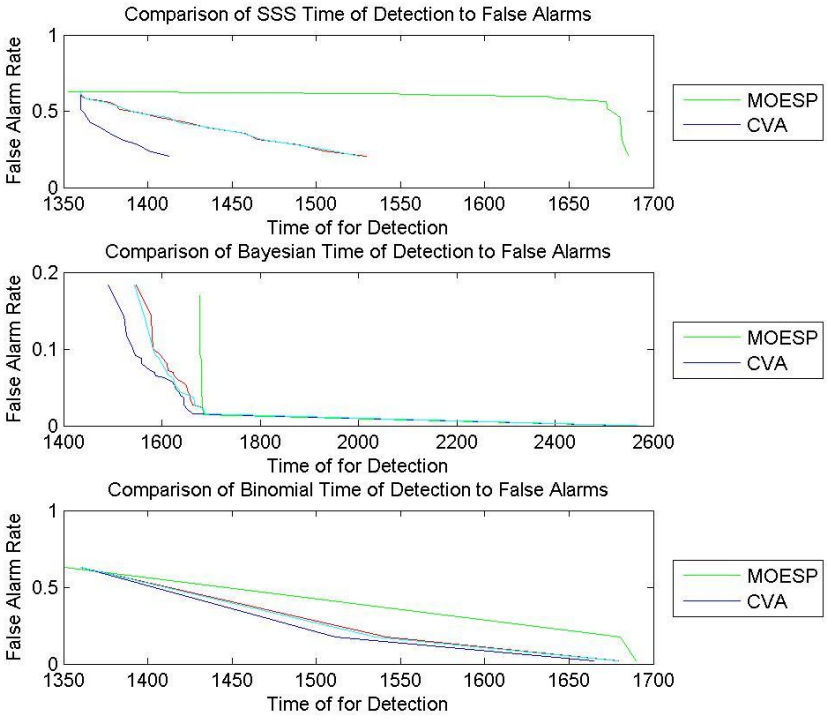

Comparison of Binomial Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA

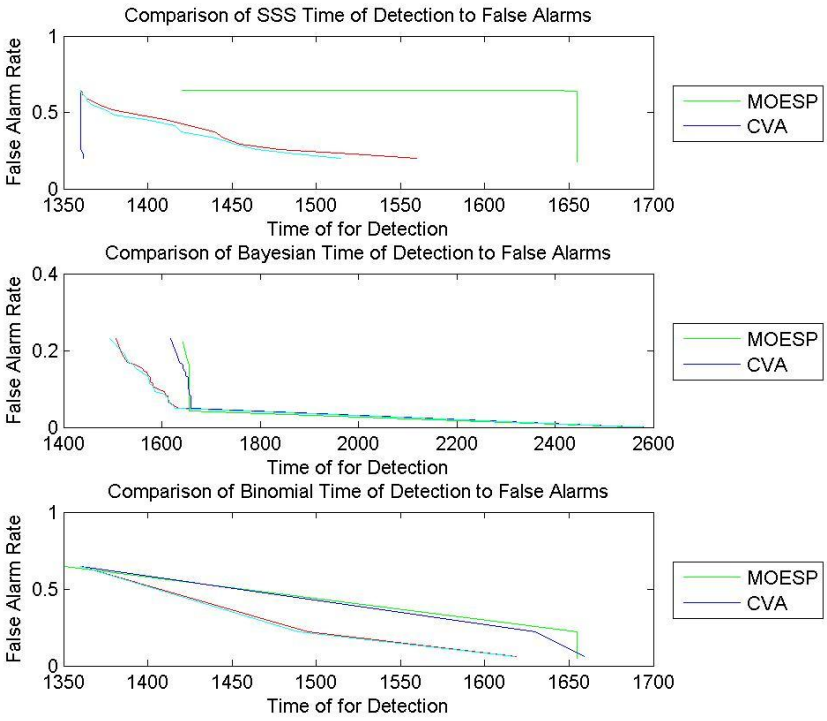

Comparison of SSS Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA
Comparison of Bayesian Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA
Comparison of Binomial Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA

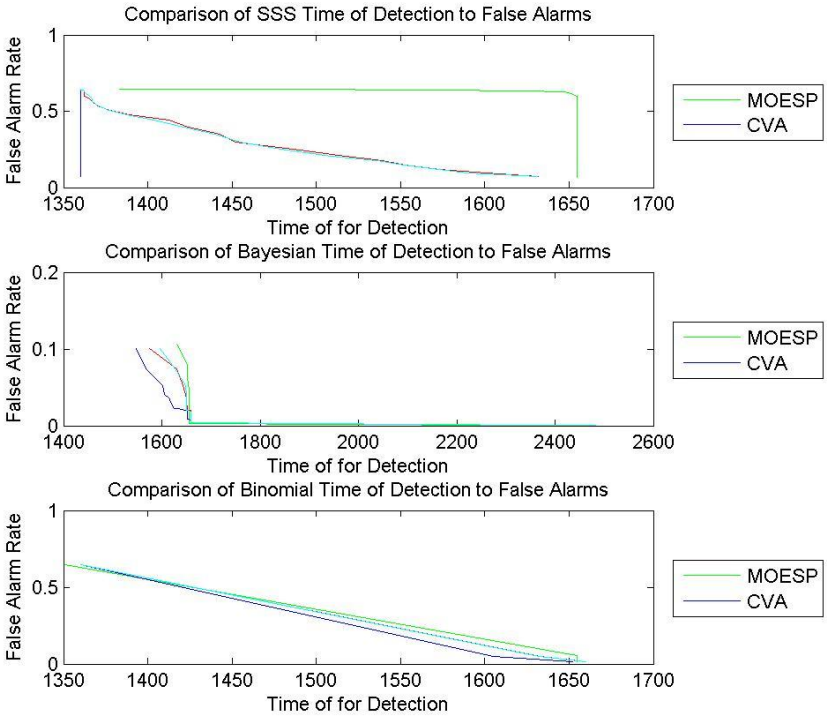

Comparison of SSS Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA
Comparison of Bayesian Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA
Comparison of Binomial Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA

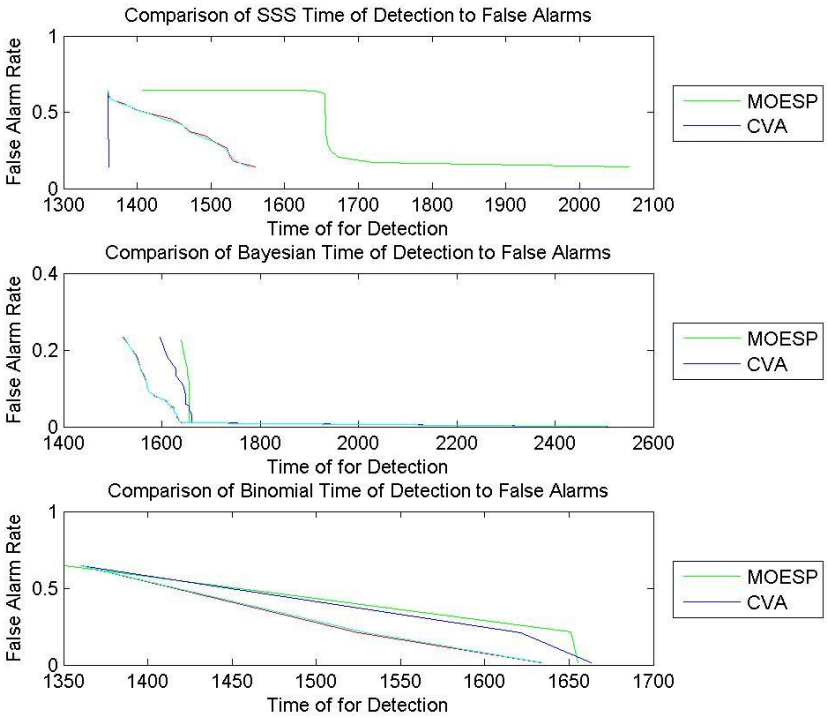

Comparison of SSS Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA
Comparison of Bayesian Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA
Comparison of Binomial Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA

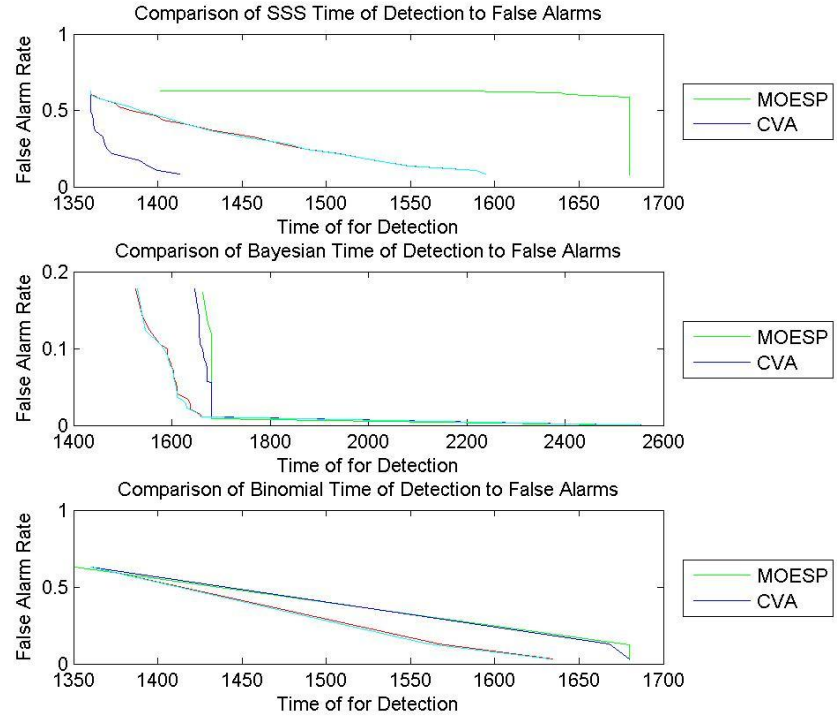

Comparison of SSS Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA
Comparison of Bayesian Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA
Comparison of Binomial Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA

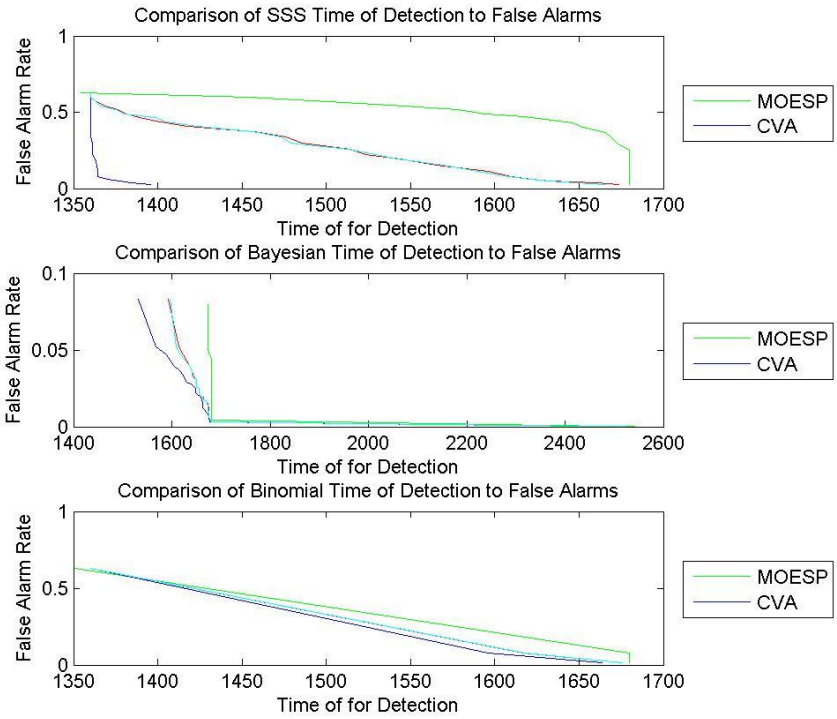

Comparison of SSS Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA
Comparison of Bayesian Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA
Comparison of Binomial Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA

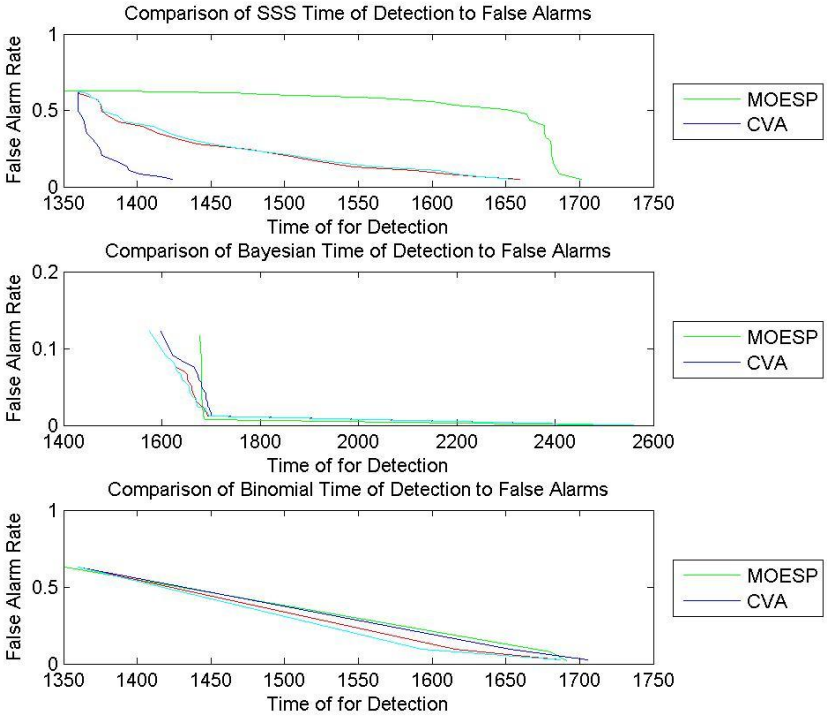

Comparison of SSS Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA
Comparison of Bayesian Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA
Comparison of Binomial Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA

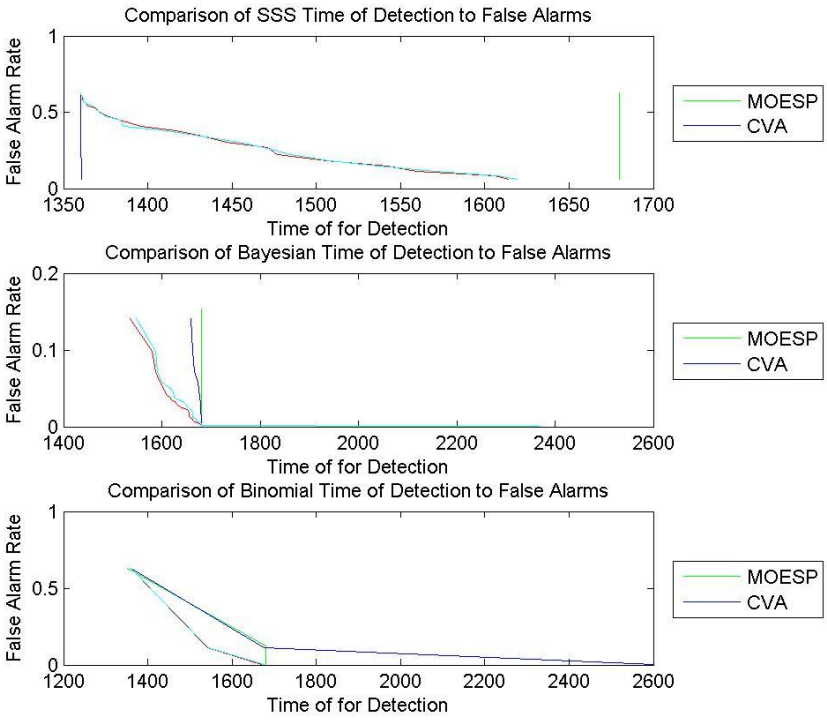

Comparison of SSS Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA
Comparison of Bayesian Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA
Comparison of Binomial Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA

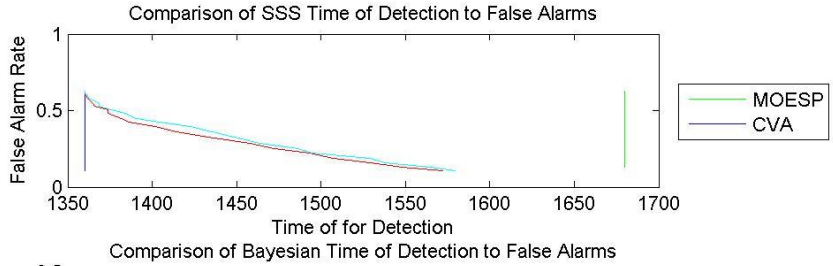

Comparison of SSS Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA
Comparison of Bayesian Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA
Comparison of Binomial Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA

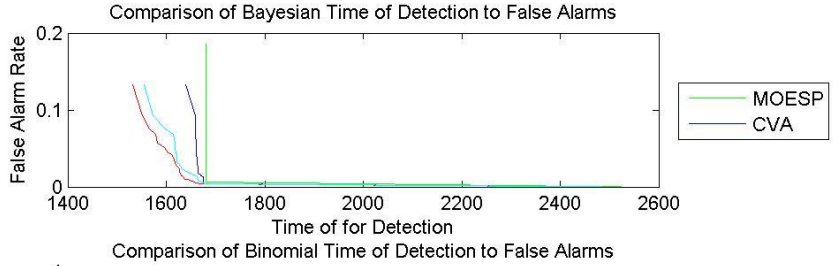

Comparison of SSS Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA
Comparison of Bayesian Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA
Comparison of Binomial Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA

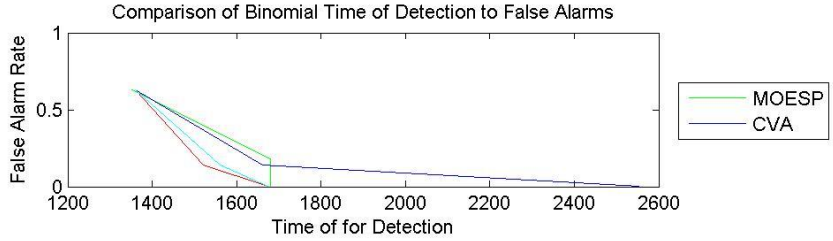

Comparison of SSS Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA
Comparison of Bayesian Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA
Comparison of Binomial Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA

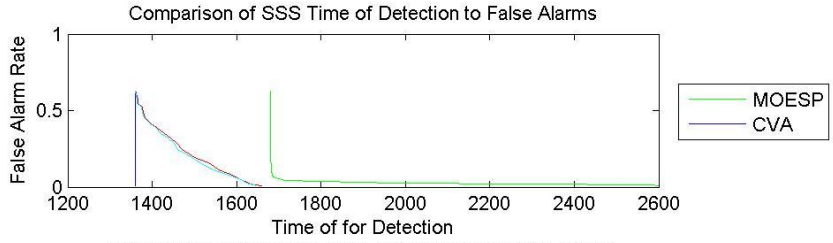

Comparison of SSS Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA
Comparison of Bayesian Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA
Comparison of Binomial Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA

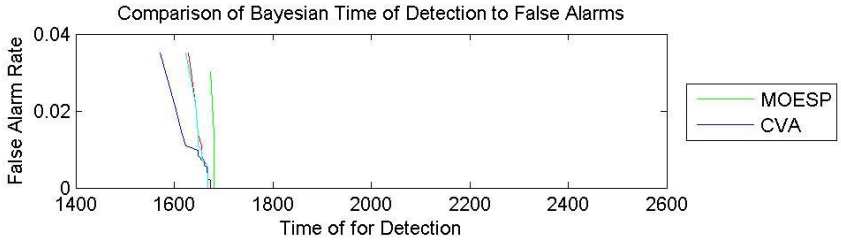

Comparison of SSS Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA
Comparison of Bayesian Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA
Comparison of Binomial Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA

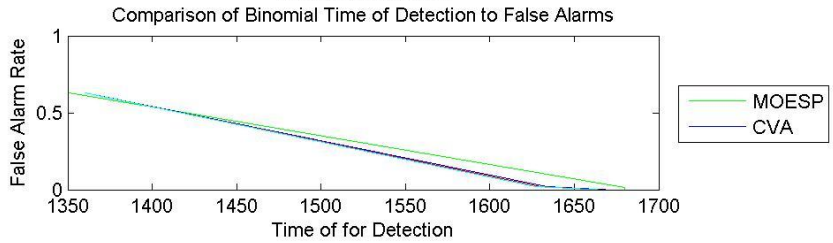

Comparison of SSS Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA


Comparison of Bayesian Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA


Comparison of Binomial Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA


Comparison of SSS Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA


Comparison of Bayesian Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA


Comparison of Binomial Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA

10.2.3 Fuel Injection ROC charts.

List of ROC charts in order. The types can be found on pages 27 and 28

Type 1 Attack 1

Type 1 Attack 2

Type 1 Attack 3

Type 1 Attack 4

Type 2 Attack 1

Type 2 Attack 2

Type 2 Attack 3

Type 2 Attack 4

Type 3 Attack 1

Type 3 Attack 2

Type 3 Attack 3

Type 3 Attack 4

Type 4 Attack 1

Type 4 Attack 2

Type 4 Attack 3

Type 4 Attack 4

Type 5 Attack 1

Type 5 Attack 2

Type 5 Attack 3

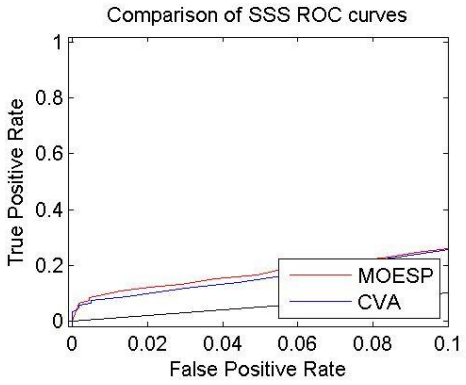
Type 5 Attack 4

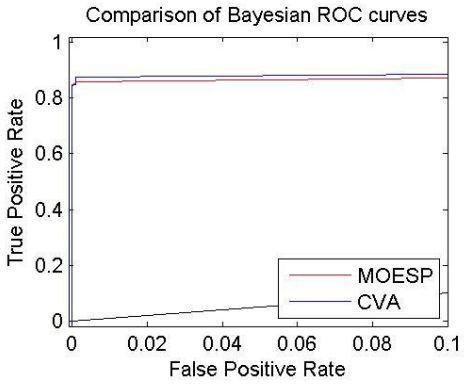
Type 6 Attack 1

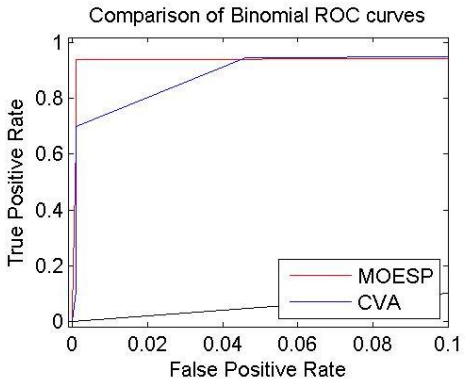
Type 6 Attack 2

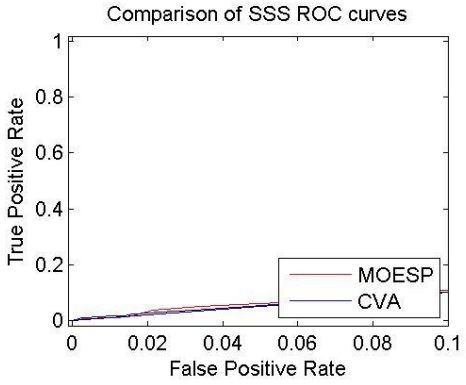
Type 6 Attack 3

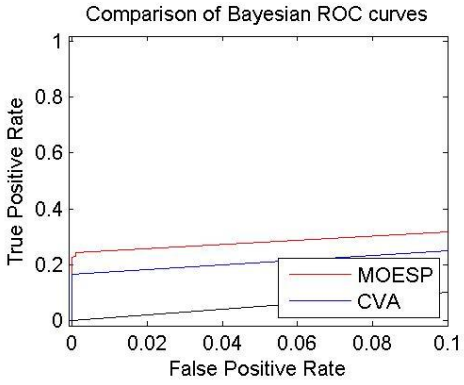
Type 6 Attack 4

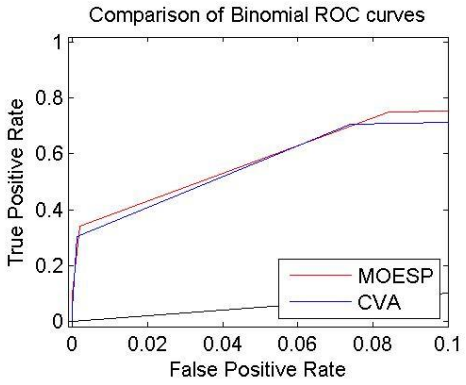

Comparison of SSS ROC curves
True Positive Rate
False Positive Rate
MOESP
CVA

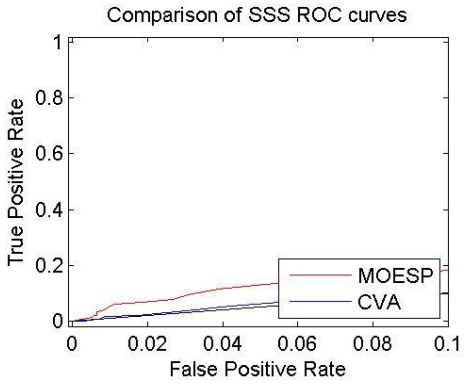

Comparison of Bayesian ROC curves
True Positive Rate
False Positive Rate
MOESP
CVA

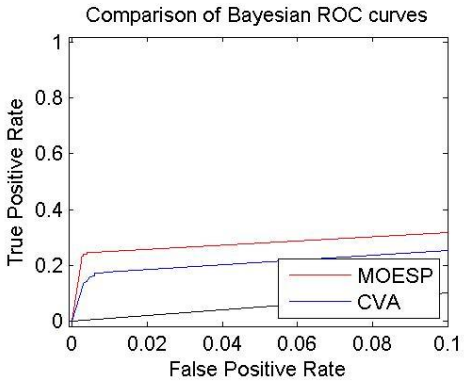

Comparison of Binomial ROC curves
True Positive Rate
False Positive Rate
MOESP
CVA

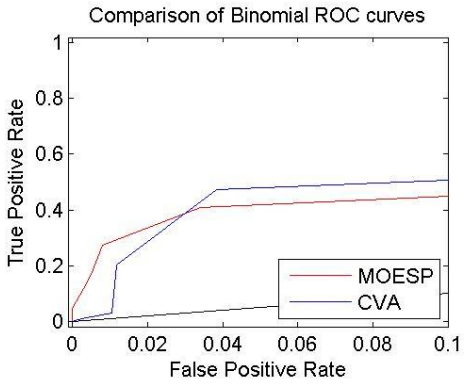

Comparison of SSS ROC curves
True Positive Rate
False Positive Rate
MOESP
CVA

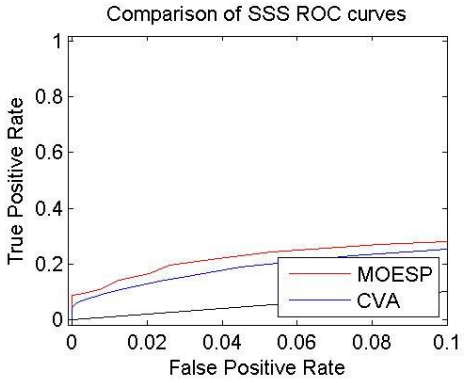

Comparison of Bayesian ROC curves
True Positive Rate
False Positive Rate
MOESP
CVA

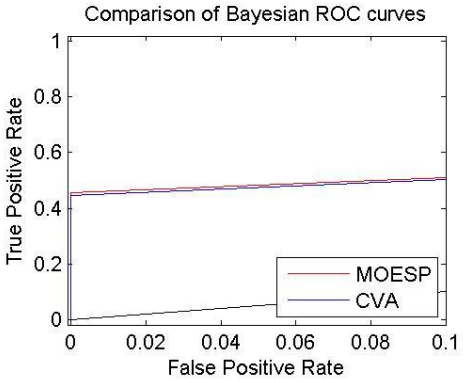

Comparison of Binomial ROC curves
True Positive Rate
False Positive Rate
MOESP
CVA

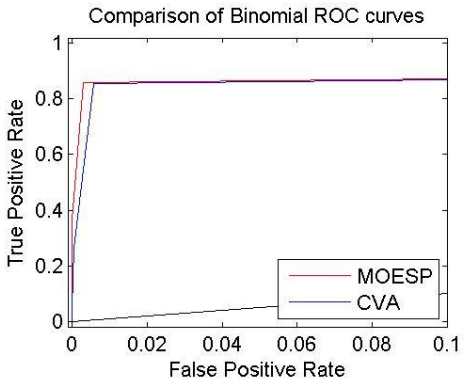

Comparison of SSS ROC curves
True Positive Rate
False Positive Rate
MOESP
CVA

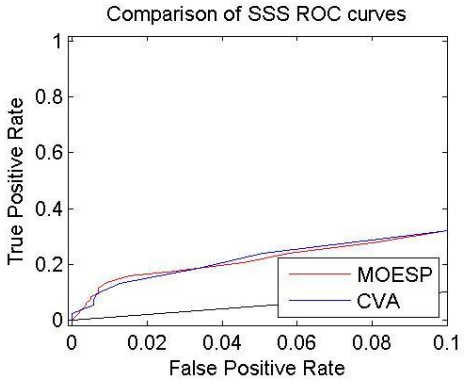

Comparison of Bayesian ROC curves
True Positive Rate
False Positive Rate
MOESP
CVA

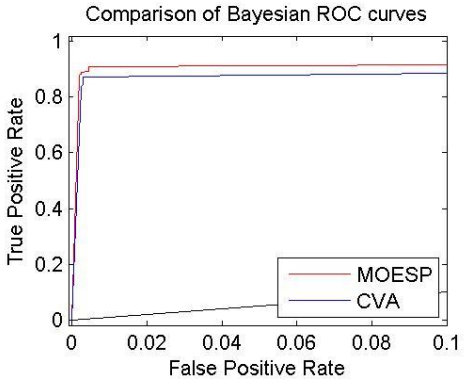

Comparison of Binomial ROC curves
True Positive Rate
False Positive Rate
MOESP
CVA

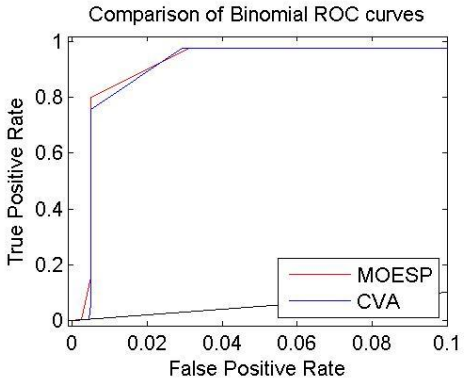

Comparison of SSS ROC curves
True Positive Rate
False Positive Rate
MOESP
CVA

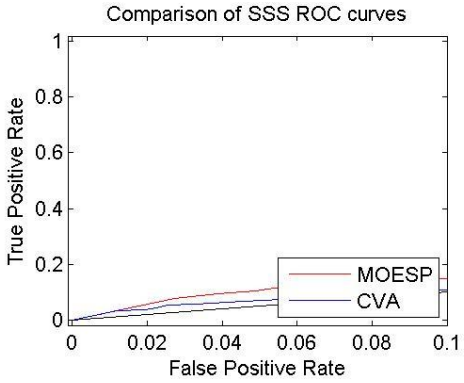

Comparison of Bayesian ROC curves
True Positive Rate
False Positive Rate
MOESP
CVA

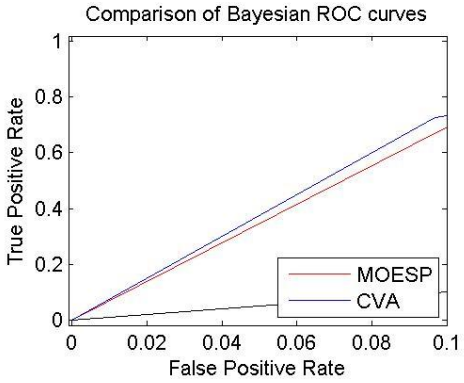

Comparison of Binomial ROC curves
True Positive Rate
False Positive Rate
MOESP
CVA

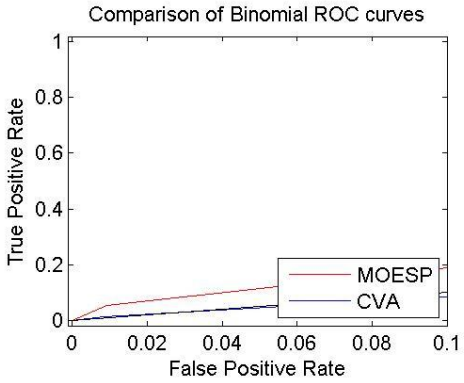

Comparison of SSS ROC curves
True Positive Rate
False Positive Rate
MOESP
CVA

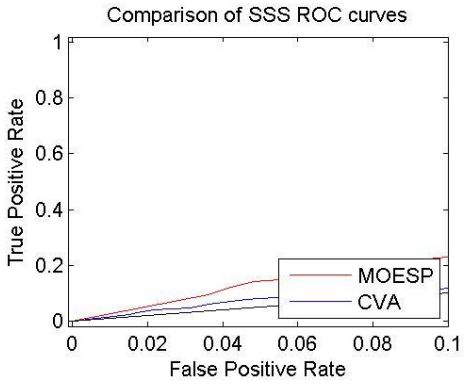

Comparison of Bayesian ROC curves
True Positive Rate
False Positive Rate
MOESP
CVA

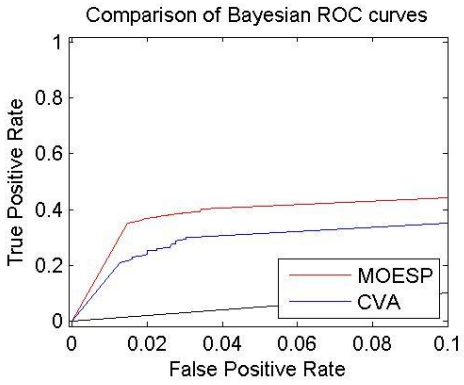

Comparison of Binomial ROC curves
True Positive Rate
False Positive Rate
MOESP
CVA

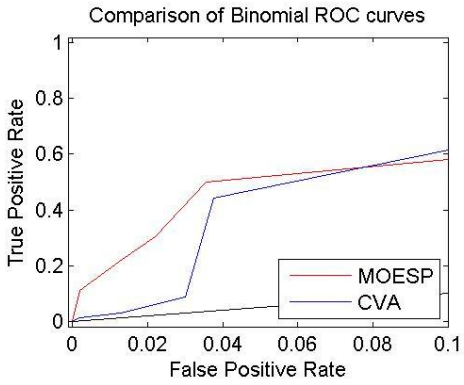

Comparison of SSS ROC curves
True Positive Rate
False Positive Rate
MOESP
CVA

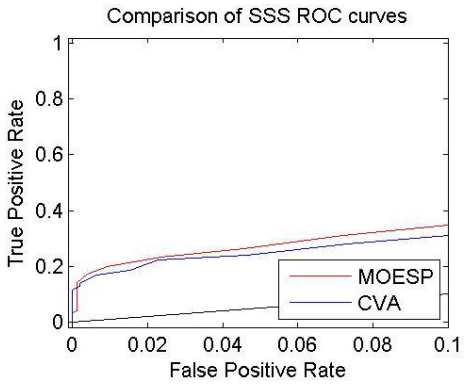

Comparison of Bayesian ROC curves
True Positive Rate
False Positive Rate
MOESP
CVA

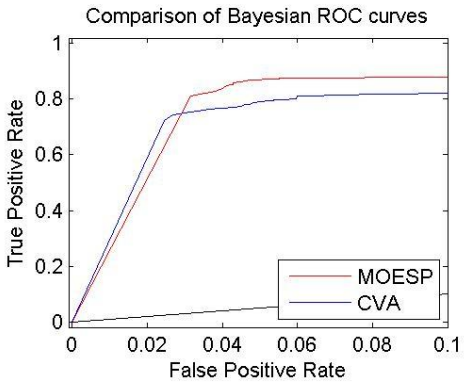

Comparison of Binomial ROC curves
True Positive Rate
False Positive Rate
MOESP
CVA

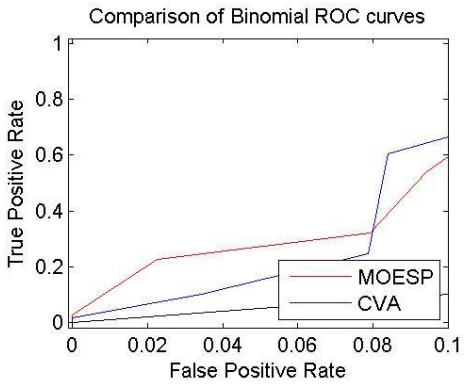

Comparison of SSS ROC curves
True Positive Rate
False Positive Rate
MOESP
CVA
0
0.2
0.4
0.6
0.8
1
0.02
0.04
0.06
0.08
0.1

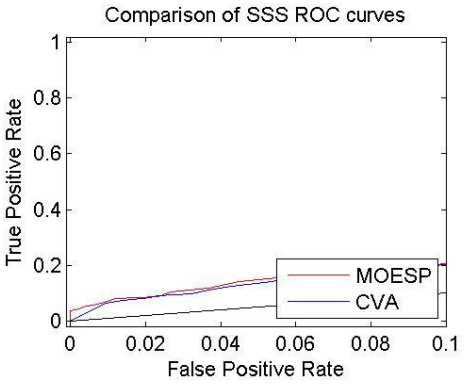

Comparison of Bayesian ROC curves
True Positive Rate
False Positive Rate
MOESP
CVA
0
0.2
0.4
0.6
0.8
1
0.02
0.04
0.06
0.08
0.1

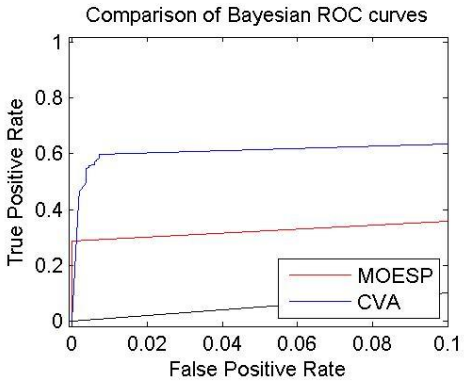

Comparison of Binomial ROC curves
True Positive Rate
False Positive Rate
MOESP
CVA
0
0.2
0.4
0.6
0.8
1
0.02
0.04
0.06
0.08
0.1

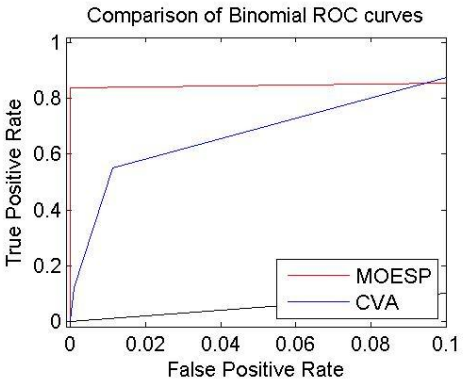

Comparison of SSS ROC curves
True Positive Rate
False Positive Rate
MOESP
CVA
0
0.2
0.4
0.6
0.8
1
0.02
0.04
0.06
0.08
0.1

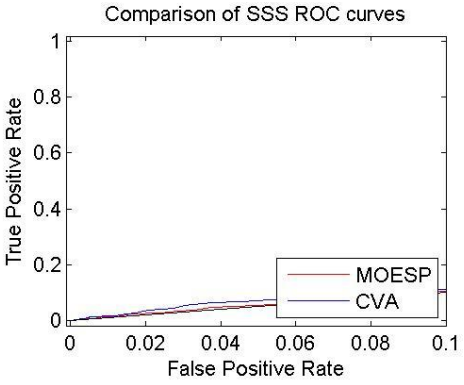

Comparison of Bayesian ROC curves
True Positive Rate
False Positive Rate
MOESP
CVA
0
0.2
0.4
0.6
0.8
1
0.02
0.04
0.06
0.08
0.1

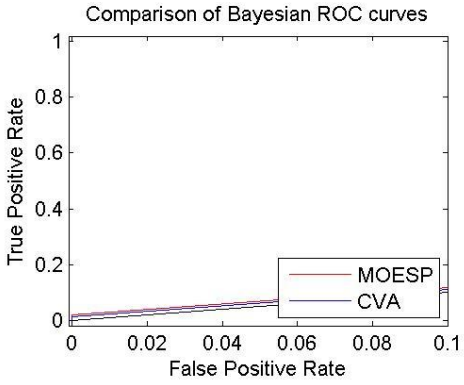

Comparison of Binomial ROC curves
True Positive Rate
False Positive Rate
MOESP
CVA
0
0.2
0.4
0.6
0.8
1
0.02
0.04
0.06
0.08
0.1

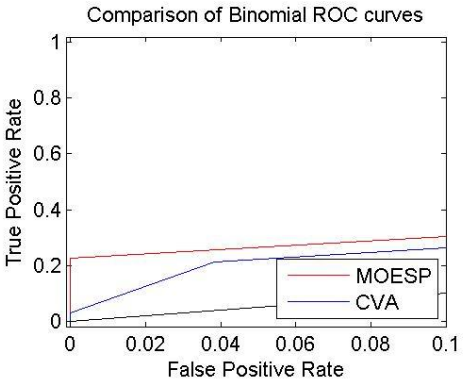

Comparison of SSS ROC curves
True Positive Rate
False Positive Rate
MOESP
CVA

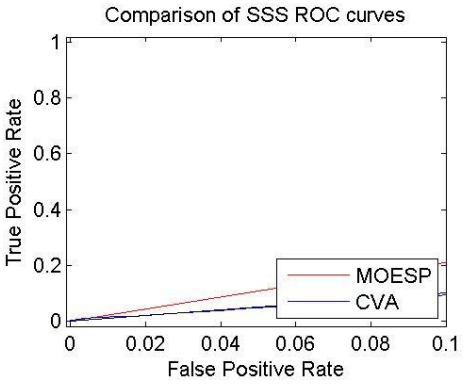

Comparison of Bayesian ROC curves
True Positive Rate
False Positive Rate
MOESP
CVA

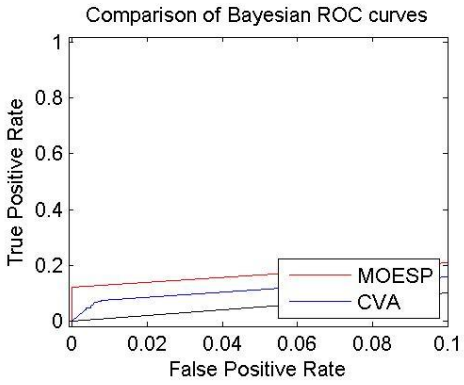

Comparison of Binomial ROC curves
True Positive Rate
False Positive Rate
MOESP
CVA

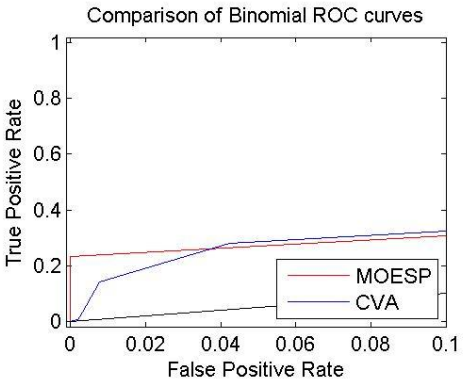

Comparison of SSS ROC curves
True Positive Rate
False Positive Rate
MOESP
CVA

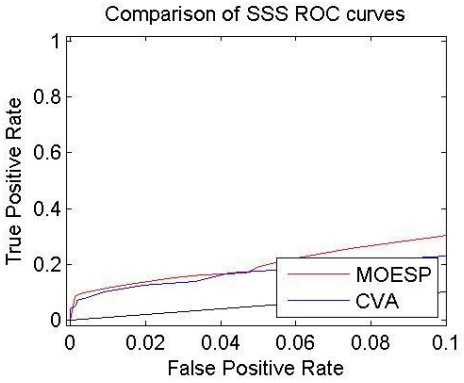

Comparison of Bayesian ROC curves
True Positive Rate
False Positive Rate
MOESP
CVA

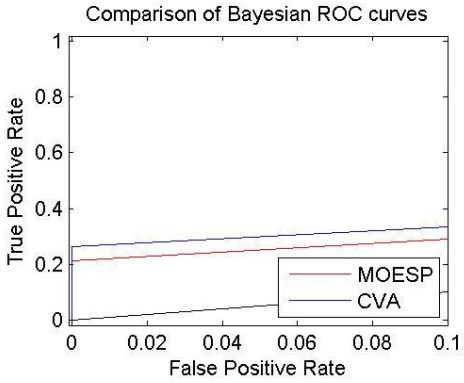

Comparison of Binomial ROC curves
True Positive Rate
False Positive Rate
MOESP
CVA

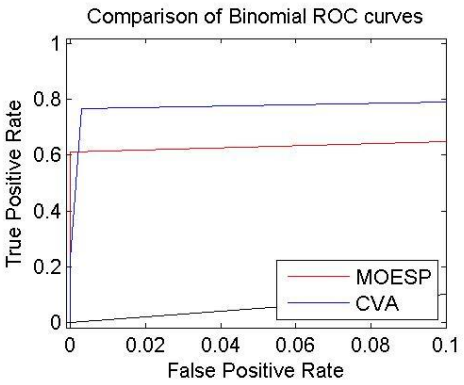

Comparison of SSS ROC curves
True Positive Rate
False Positive Rate
MOESP
CVA

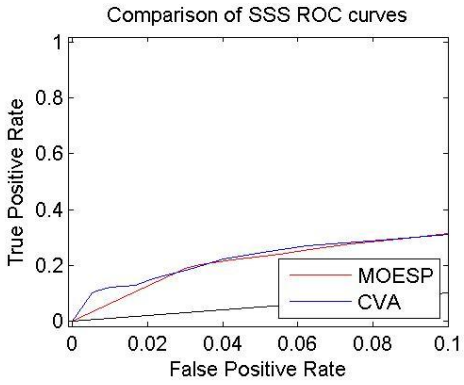

Comparison of Bayesian ROC curves
True Positive Rate
False Positive Rate
MOESP
CVA

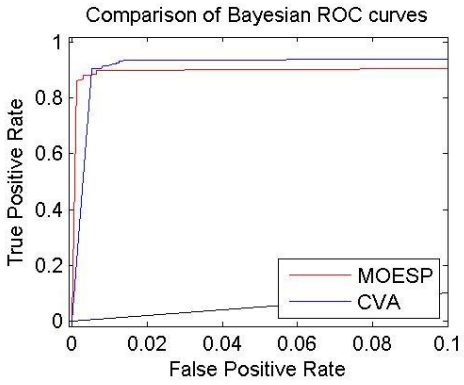

Comparison of Binomial ROC curves
True Positive Rate
False Positive Rate
MOESP
CVA

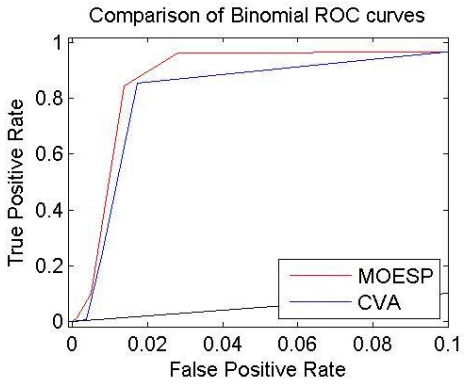

Comparison of SSS ROC curves
True Positive Rate
False Positive Rate
MOESP
CVA

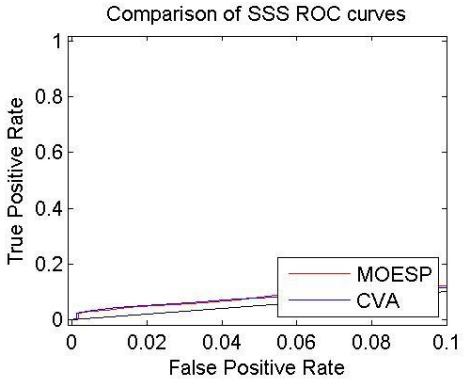

Comparison of Bayesian ROC curves
True Positive Rate
False Positive Rate
MOESP
CVA

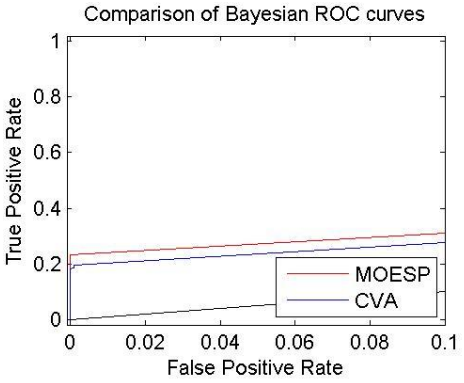

Comparison of Binomial ROC curves
True Positive Rate
False Positive Rate
MOESP
CVA

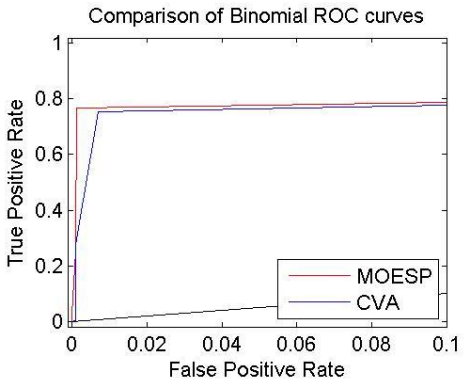

Comparison of SSS ROC curves
True Positive Rate
False Positive Rate
MOESP
CVA

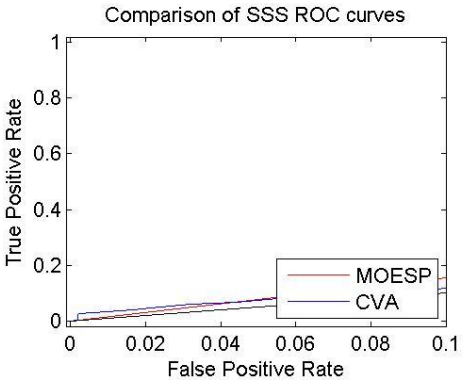

Comparison of Bayesian ROC curves
True Positive Rate
False Positive Rate
MOESP
CVA

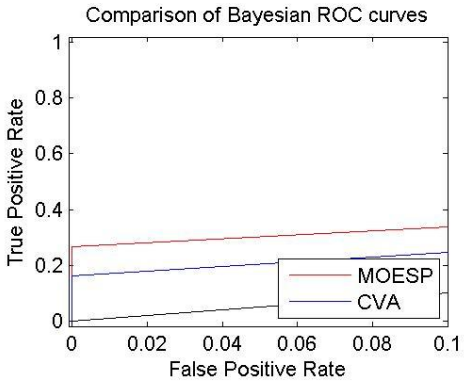

Comparison of Binomial ROC curves
True Positive Rate
False Positive Rate
MOESP
CVA

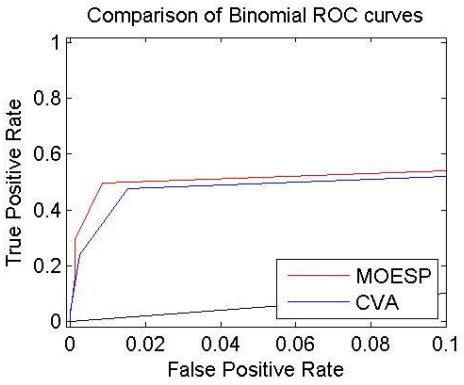

Comparison of SSS ROC curves
True Positive Rate
False Positive Rate
MOESP
CVA

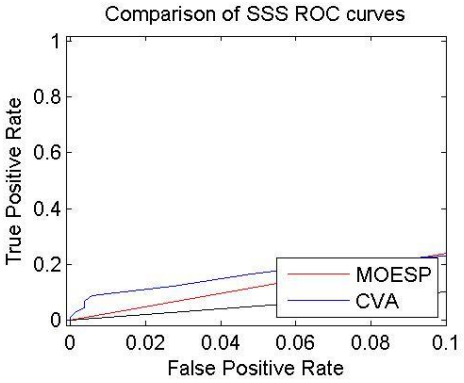

Comparison of Bayesian ROC curves
True Positive Rate
False Positive Rate
MOESP
CVA

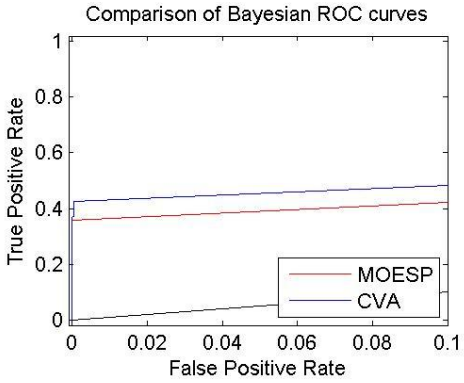

Comparison of Binomial ROC curves
True Positive Rate
False Positive Rate
MOESP
CVA

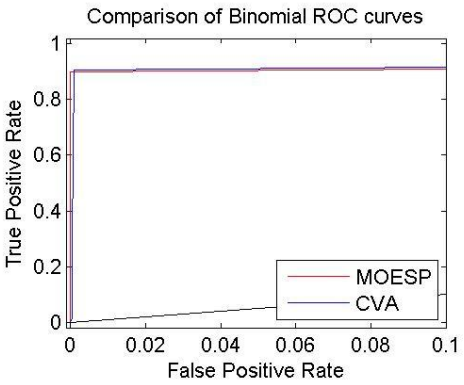

Comparison of SSS ROC curves
True Positive Rate
False Positive Rate
MOESP
CVA
0
0.2
0.4
0.6
0.8
1
0.02
0.04
0.06
0.08
0.1

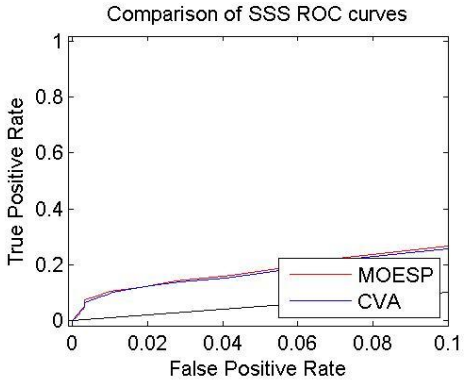

Comparison of Bayesian ROC curves
True Positive Rate
False Positive Rate
MOESP
CVA
0
0.2
0.4
0.6
0.8
1
0.02
0.04
0.06
0.08
0.1

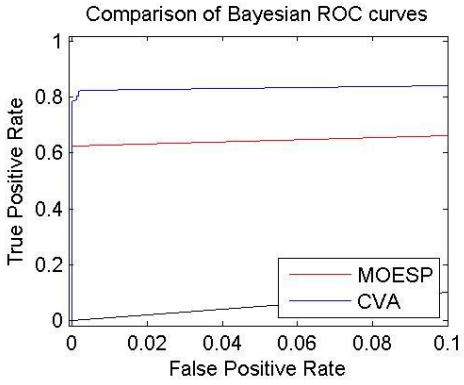

Comparison of Binomial ROC curves
True Positive Rate
False Positive Rate
MOESP
CVA
0
0.2
0.4
0.6
0.8
1
0.02
0.04
0.06
0.08
0.1

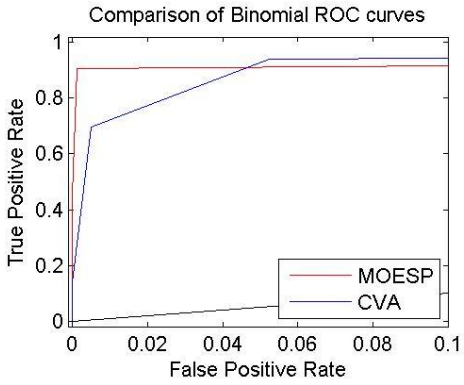

Comparison of SSS ROC curves
True Positive Rate
False Positive Rate
MOESP
CVA
0
0.2
0.4
0.6
0.8
1
0.02
0.04
0.06
0.08
0.1

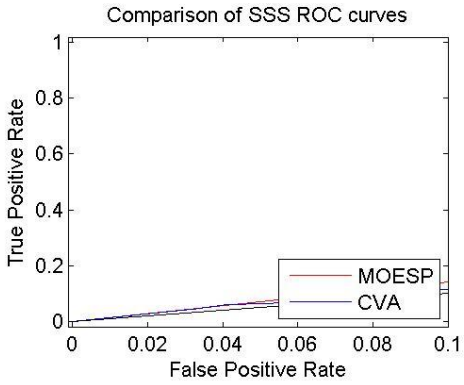

Comparison of Bayesian ROC curves
True Positive Rate
False Positive Rate
MOESP
CVA
0
0.2
0.4
0.6
0.8
1
0.02
0.04
0.06
0.08
0.1

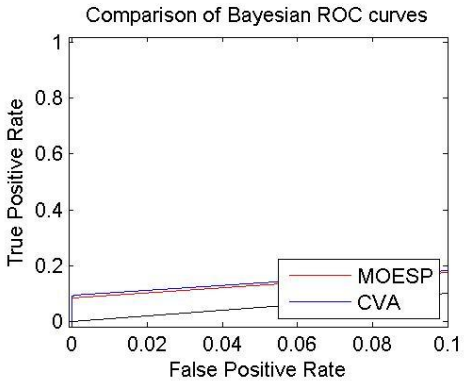

Comparison of Binomial ROC curves
True Positive Rate
False Positive Rate
MOESP
CVA
0
0.2
0.4
0.6
0.8
1
0.02
0.04
0.06
0.08
0.1

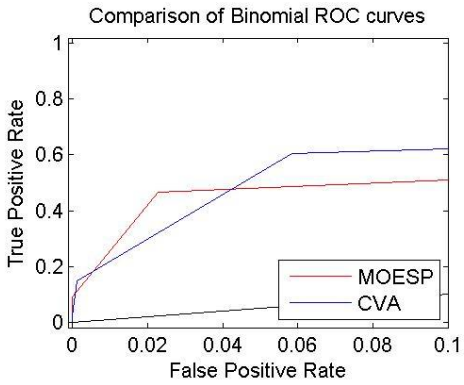

Comparison of SSS ROC curves
True Positive Rate
False Positive Rate
MOESP
CVA
0
0.2
0.4
0.6
0.8
1
0.02
0.04
0.06
0.08
0.1

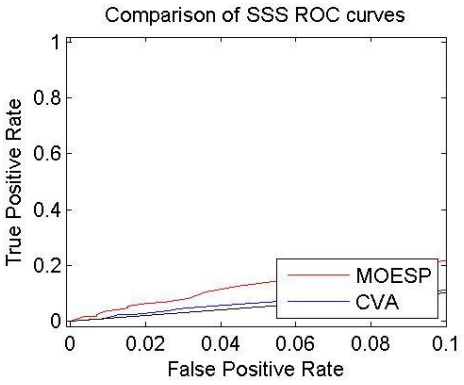

Comparison of Bayesian ROC curves
True Positive Rate
False Positive Rate
MOESP
CVA
0
0.2
0.4
0.6
0.8
1
0.02
0.04
0.06
0.08
0.1

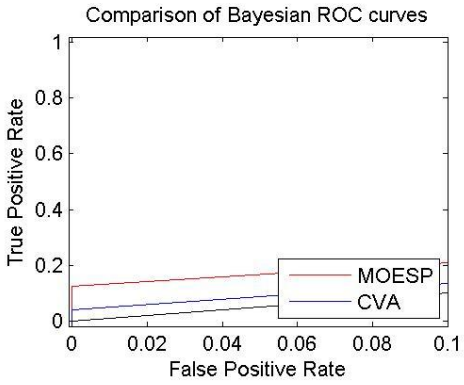

Comparison of Binomial ROC curves
True Positive Rate
False Positive Rate
MOESP
CVA
0
0.2
0.4
0.6
0.8
1
0.02
0.04
0.06
0.08
0.1

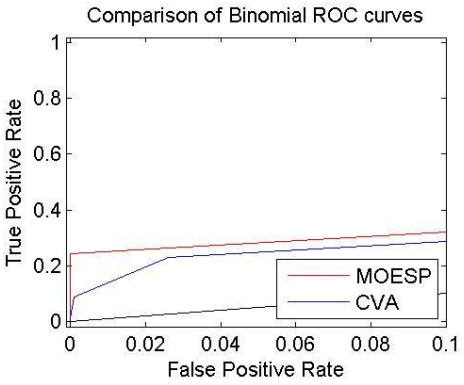

Comparison of SSS ROC curves
True Positive Rate
False Positive Rate
MOESP
CVA
0
0.2
0.4
0.6
0.8
1
0.02
0.04
0.06
0.08
0.1

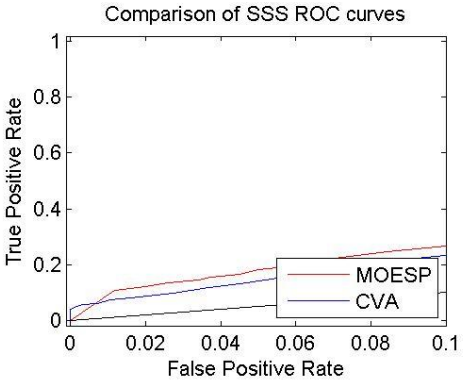

Comparison of Bayesian ROC curves
True Positive Rate
False Positive Rate
MOESP
CVA
0
0.2
0.4
0.6
0.8
1
0.02
0.04
0.06
0.08
0.1

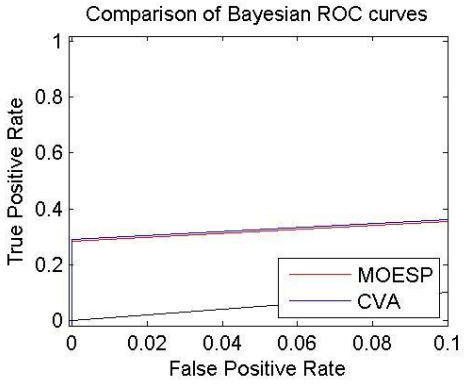

Comparison of Binomial ROC curves
True Positive Rate
False Positive Rate
MOESP
CVA
0
0.2
0.4
0.6
0.8
1
0.02
0.04
0.06
0.08
0.1

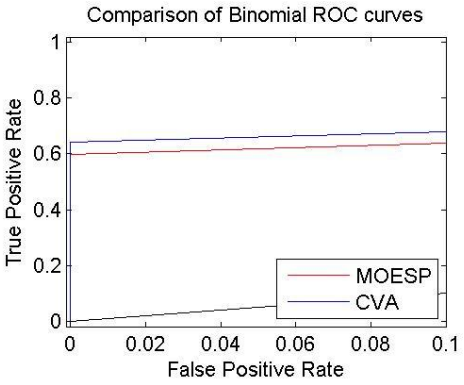

Comparison of SSS ROC curves
True Positive Rate
False Positive Rate
MOESP
CVA

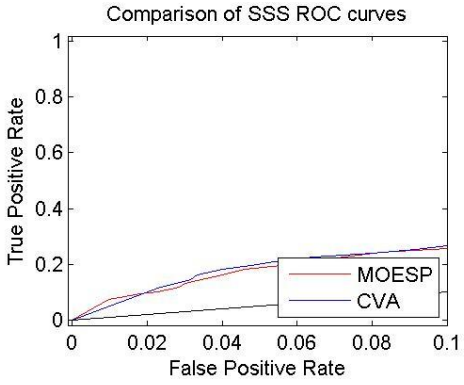

Comparison of Bayesian ROC curves
True Positive Rate
False Positive Rate
MOESP
CVA

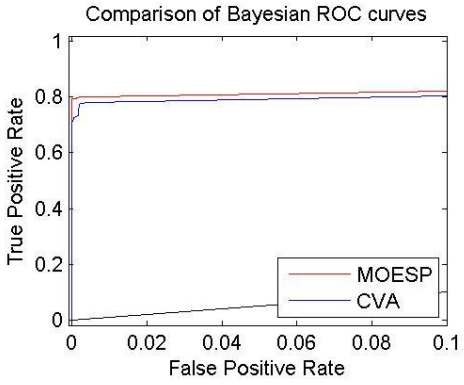

Comparison of Binomial ROC curves
True Positive Rate
False Positive Rate
MOESP
CVA

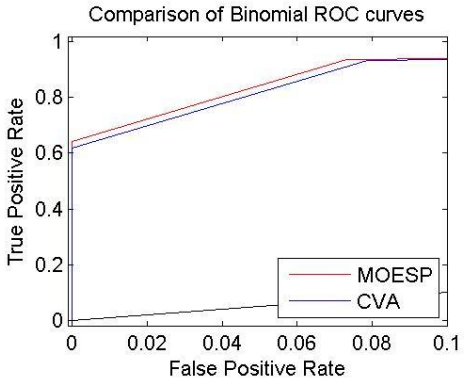

Comparison of SSS ROC curves
True Positive Rate
False Positive Rate
MOESP
CVA

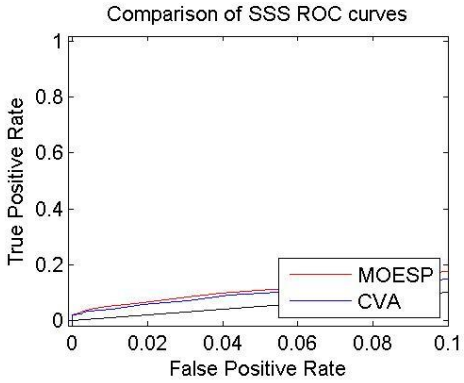

Comparison of Bayesian ROC curves
True Positive Rate
False Positive Rate
MOESP
CVA

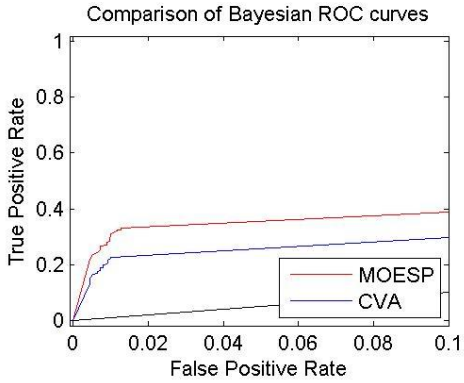

Comparison of Binomial ROC curves
True Positive Rate
False Positive Rate
MOESP
CVA

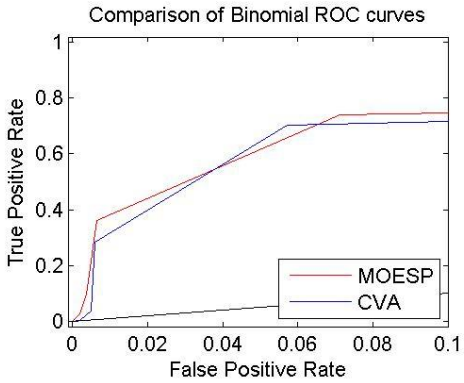

Comparison of SSS ROC curves
True Positive Rate
False Positive Rate
MOESP
CVA

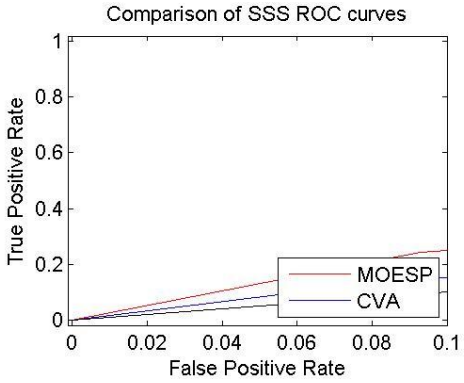

Comparison of Bayesian ROC curves
True Positive Rate
False Positive Rate
MOESP
CVA

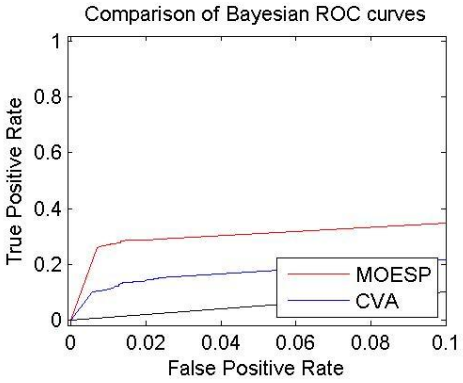

Comparison of Binomial ROC curves
True Positive Rate
False Positive Rate
MOESP
CVA

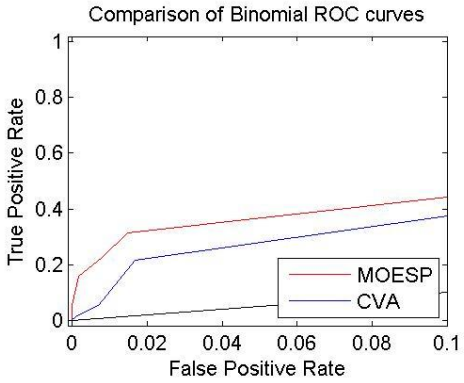

Comparison of SSS ROC curves
True Positive Rate
False Positive Rate
MOESP
CVA

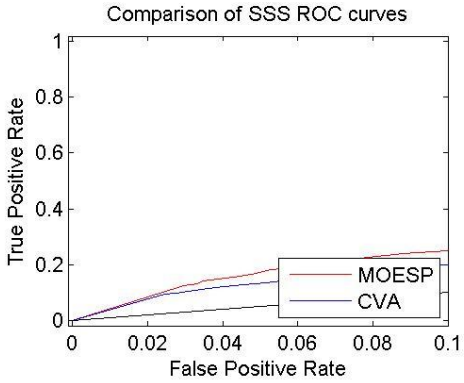

Comparison of Bayesian ROC curves
True Positive Rate
False Positive Rate
MOESP
CVA

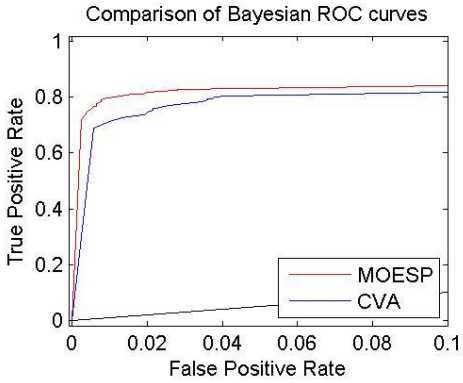

Comparison of Binomial ROC curves
True Positive Rate
False Positive Rate
MOESP
CVA

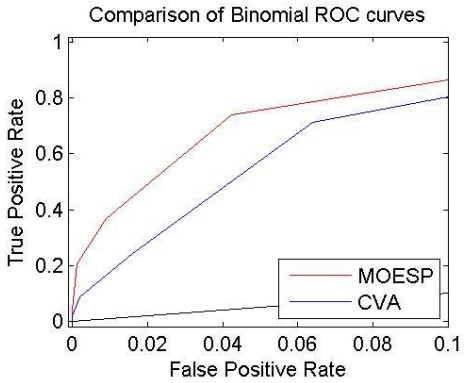

Comparison of SSS ROC curves
True Positive Rate
False Positive Rate
MOESP
CVA
0
0.2
0.4
0.6
0.8
1
0.02
0.04
0.06
0.08
0.1


Comparison of Bayesian ROC curves
True Positive Rate
False Positive Rate
MOESP
CVA
0
0.2
0.4
0.6
0.8
1
0.02
0.04
0.06
0.08
0.1


Comparison of Binomial ROC curves
True Positive Rate
False Positive Rate
MOESP
CVA
0
0.2
0.4
0.6
0.8
1
0.02
0.04
0.06
0.08
0.1


Comparison of SSS ROC curves
True Positive Rate
False Positive Rate
MOESP
CVA
0
0.2
0.4
0.6
0.8
1
0.02
0.04
0.06
0.08
0.1


Comparison of Bayesian ROC curves
True Positive Rate
False Positive Rate
MOESP
CVA
0
0.2
0.4
0.6
0.8
1
0.02
0.04
0.06
0.08
0.1


Comparison of Binomial ROC curves
True Positive Rate
False Positive Rate
MOESP
CVA
0
0.2
0.4
0.6
0.8
1
0.02
0.04
0.06
0.08
0.1

10.2.3 Fuel Injection False Alarm rate vs. Detection Time charts.

List of False Alarm rate vs. Detection Time charts in order. The types can be found on pages 27 and 28

Type 1 Attack 1

Type 1 Attack 2

Type 1 Attack 3

Type 1 Attack 4

Type 2 Attack 1

Type 2 Attack 2

Type 2 Attack 3

Type 2 Attack 4

Type 3 Attack 1

Type 3 Attack 2

Type 3 Attack 3

Type 3 Attack 4

Type 4 Attack 1

Type 4 Attack 2

Type 4 Attack 3

Type 4 Attack 4

Type 5 Attack 1

Type 5 Attack 2

Type 5 Attack 3

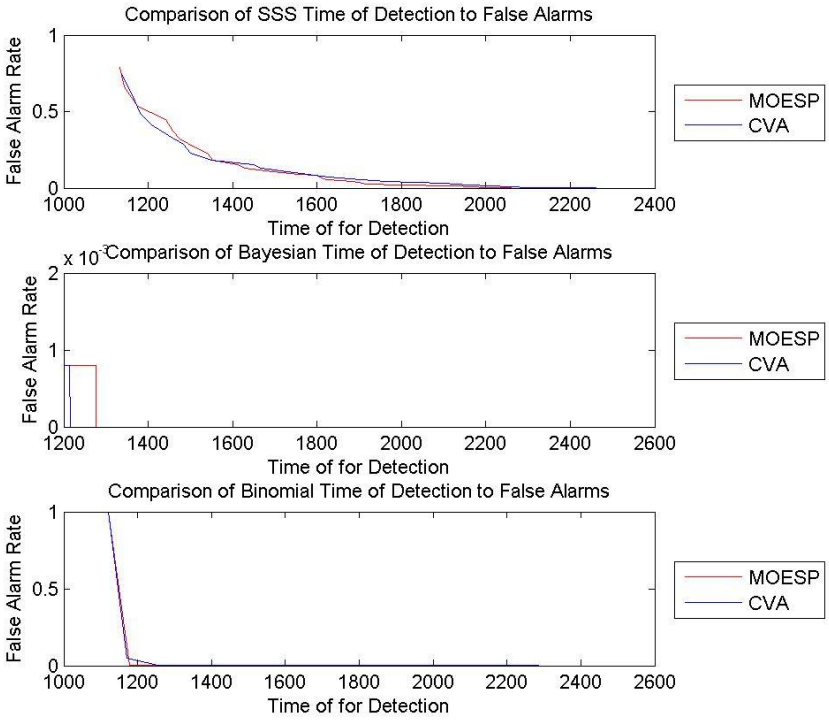
Type 5 Attack 4

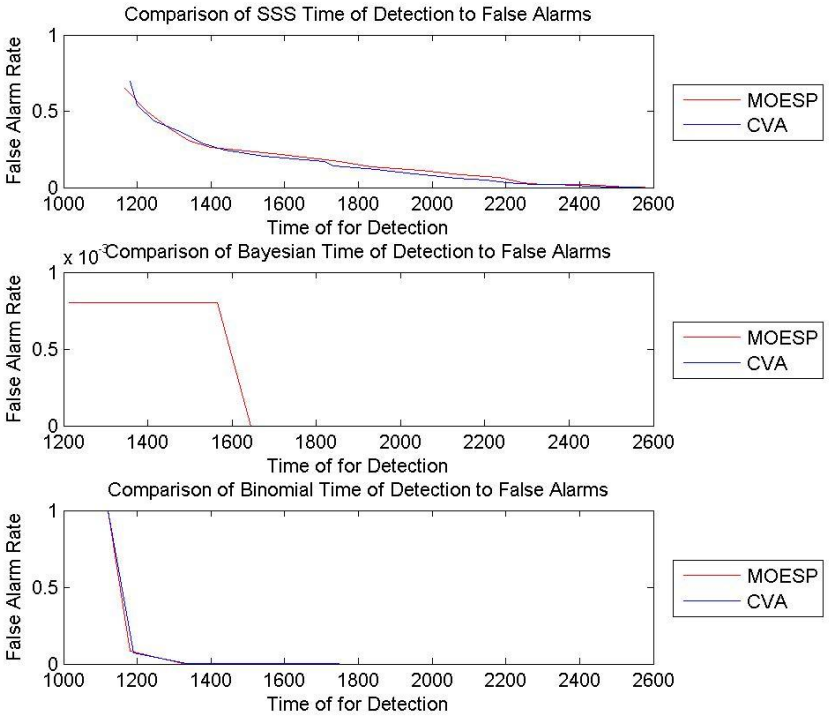
Type 6 Attack 1

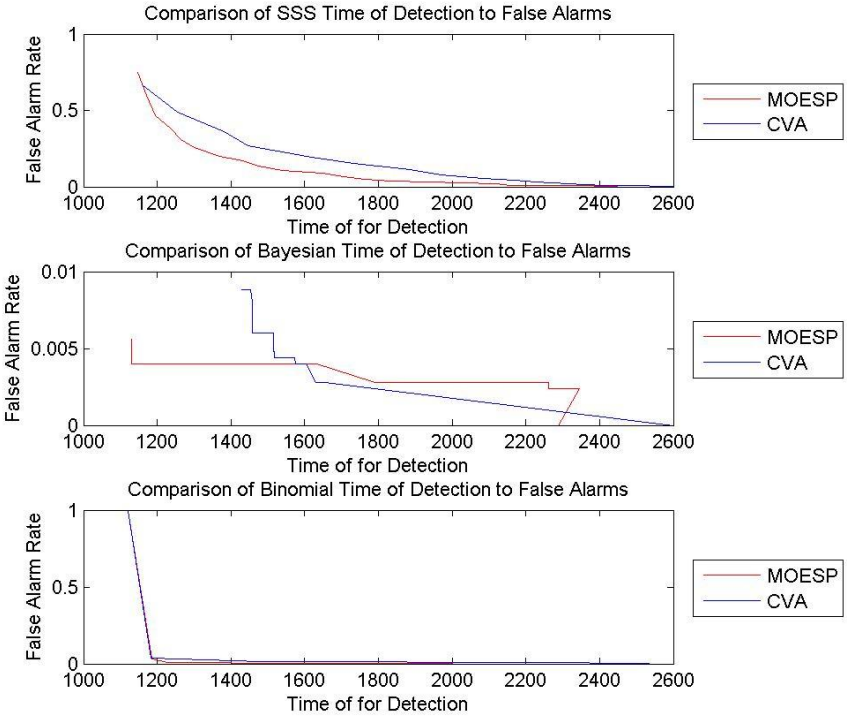
Type 6 Attack 2

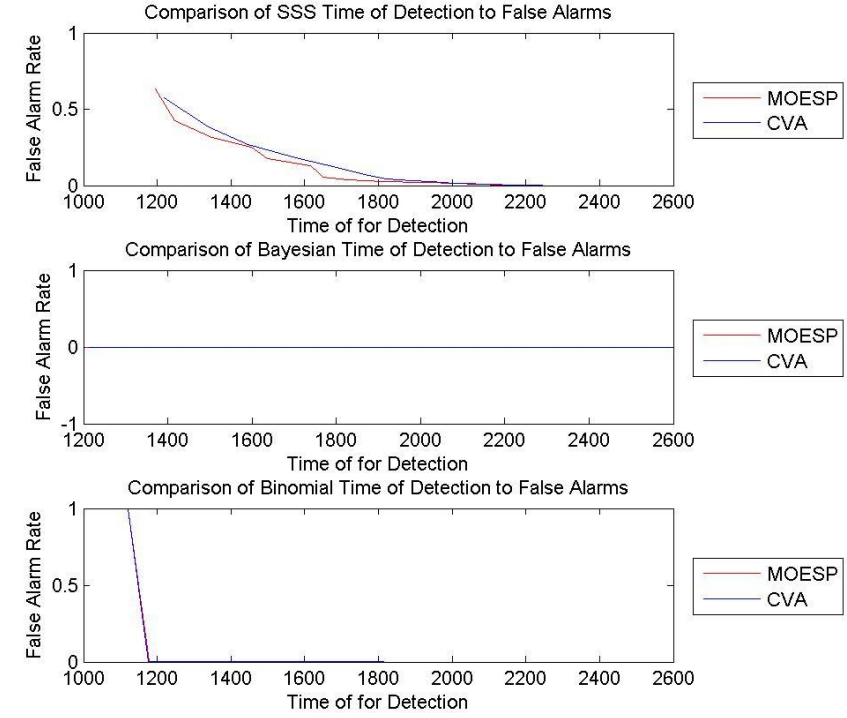
Type 6 Attack 3

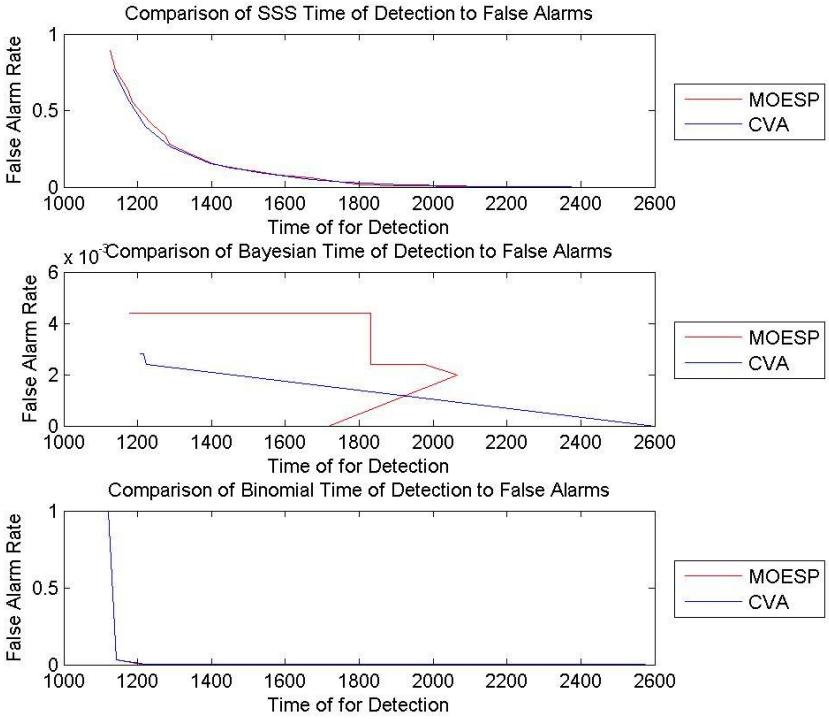
Type 6 Attack 4

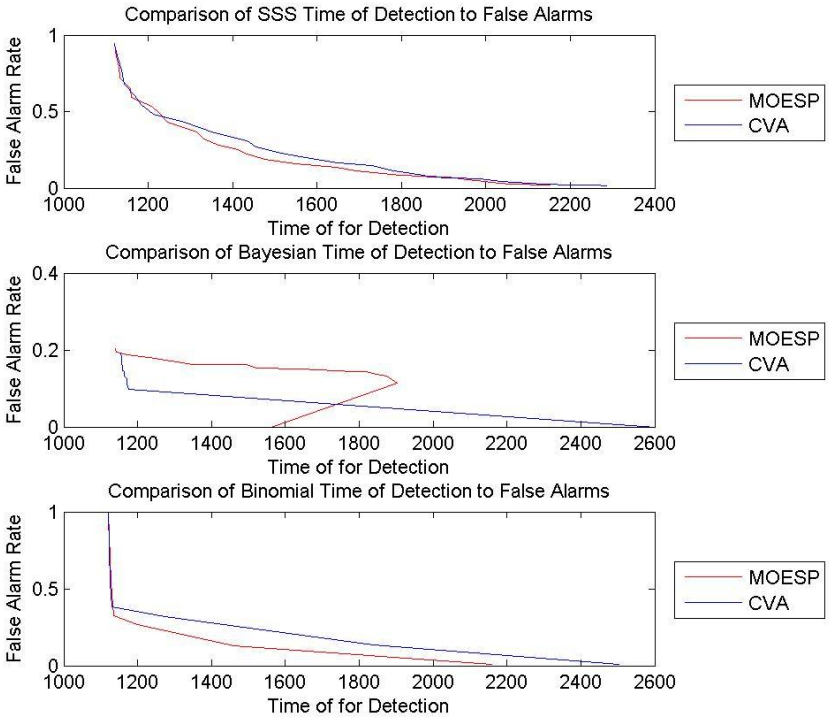

Comparison of SSS Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA
x 10^-3
Comparison of Bayesian Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA
Comparison of Binomial Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA

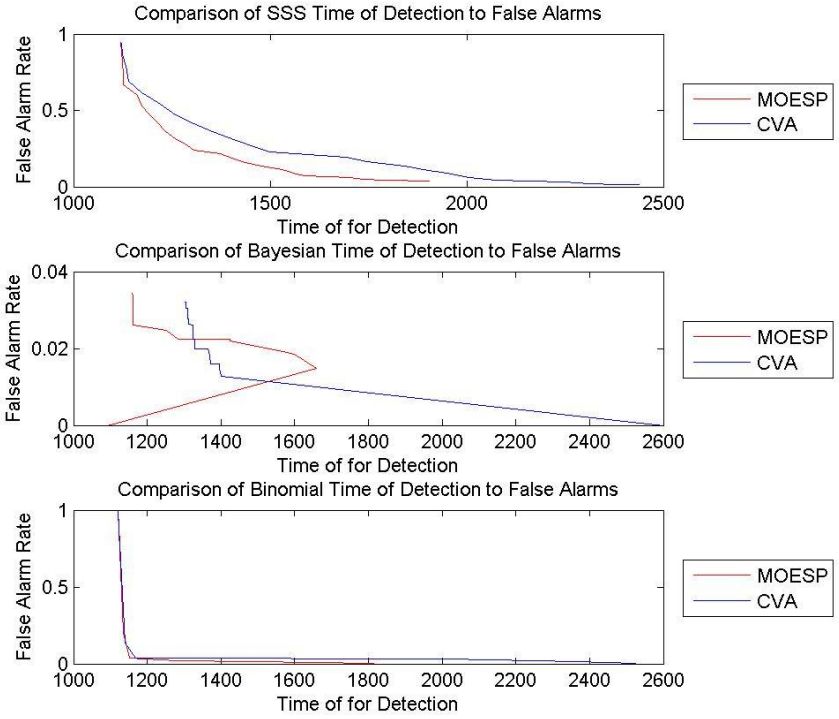

Comparison of SSS Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA
x 10^-3
Comparison of Bayesian Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA
Comparison of Binomial Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA

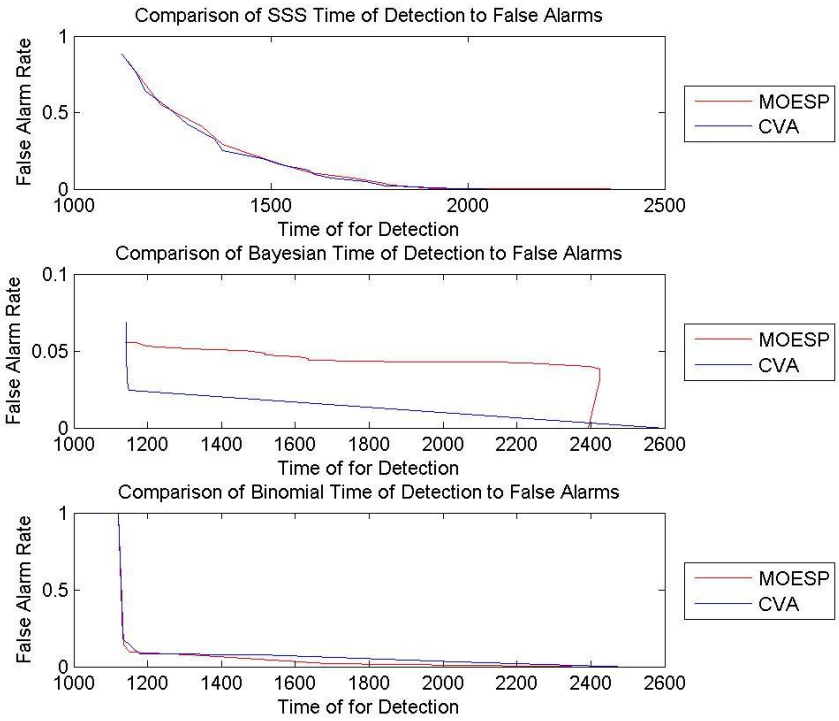

Comparison of SSS Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA
Comparison of Bayesian Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA
Comparison of Binomial Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA

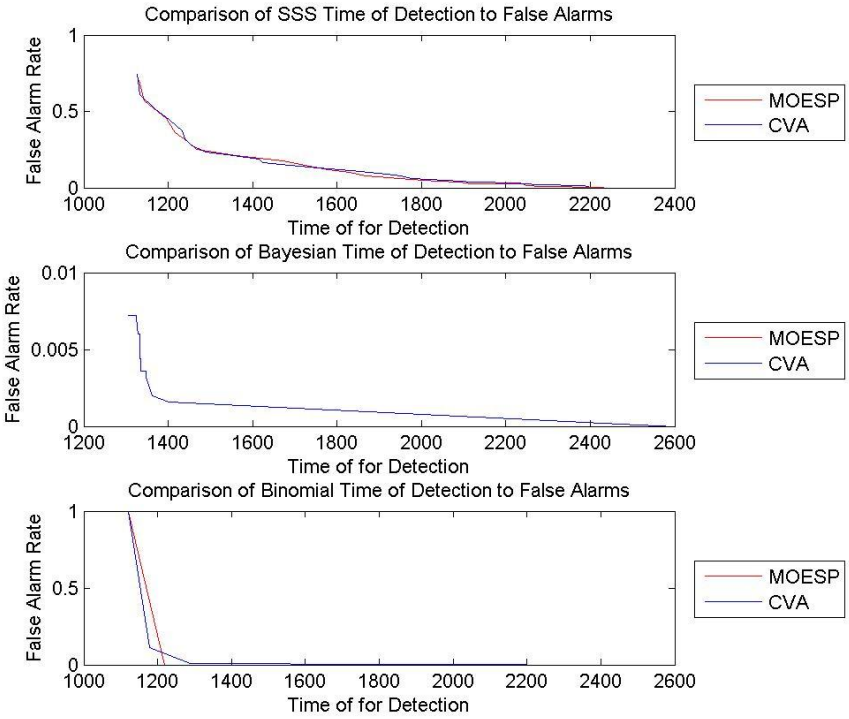

Comparison of SSS Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA
Comparison of Bayesian Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA
Comparison of Binomial Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA

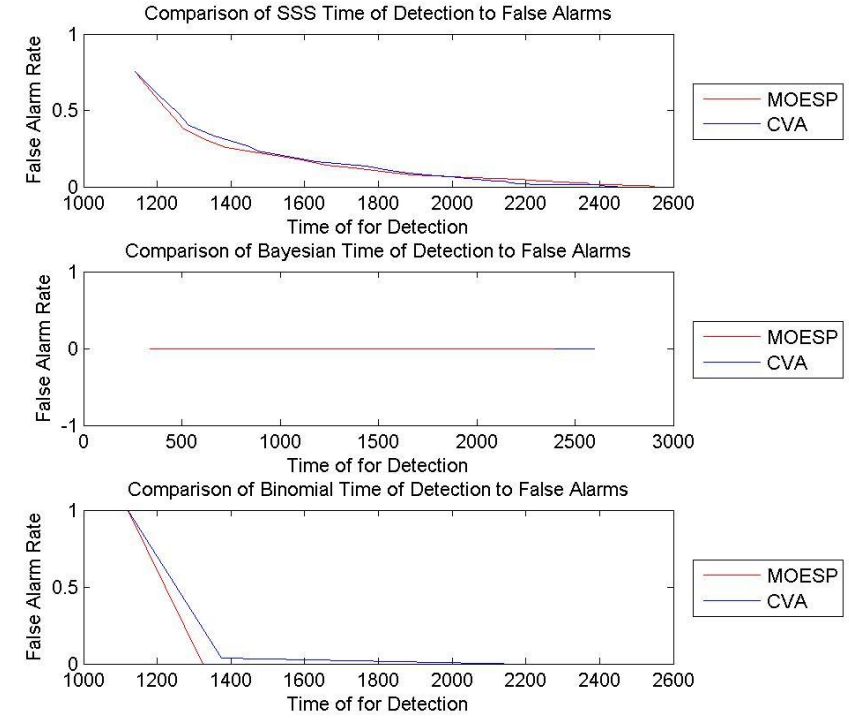

Comparison of SSS Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA
Comparison of Bayesian Time of Detection to False Alarms
x 10^-3
False Alarm Rate
Time of for Detection
MOESP
CVA
Comparison of Binomial Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA

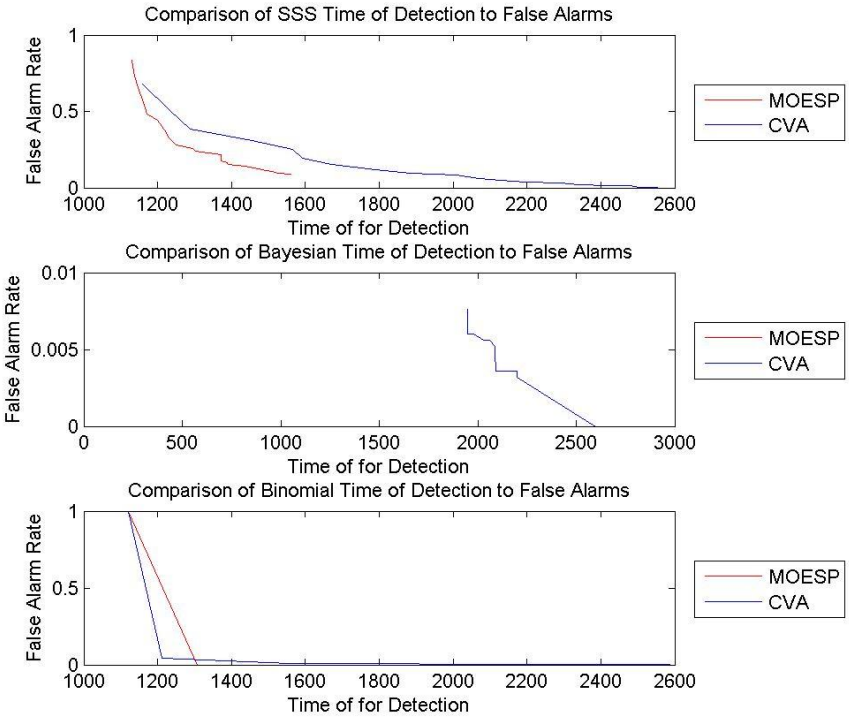

Comparison of SSS Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA
Comparison of Bayesian Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA
Comparison of Binomial Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA

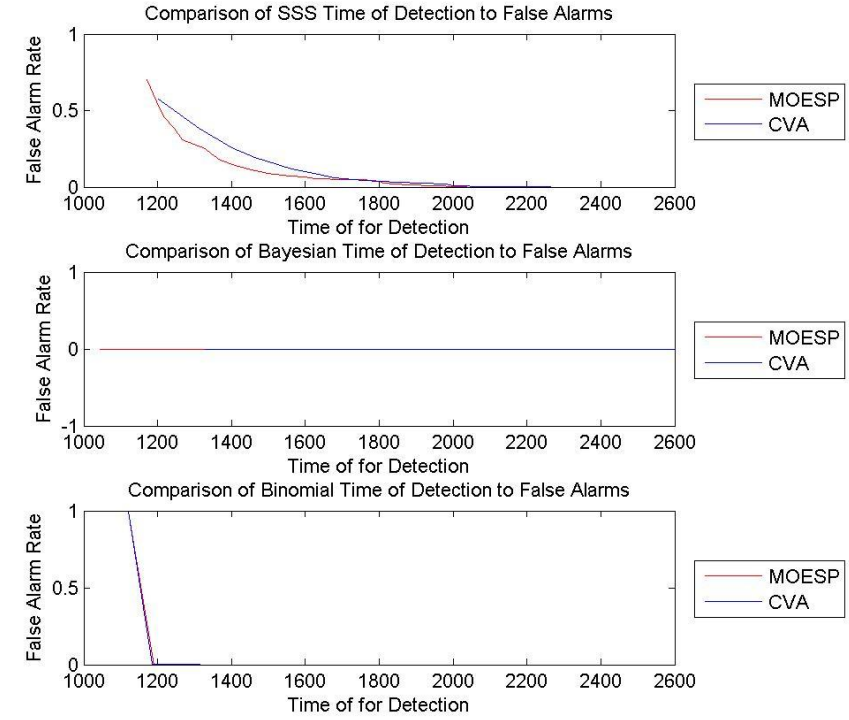

Comparison of SSS Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA
Comparison of Bayesian Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA
Comparison of Binomial Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA

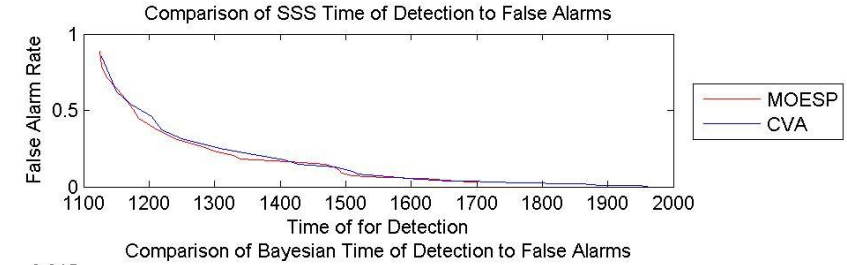

Comparison of SSS Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA
Comparison of Bayesian Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA
Comparison of Binomial Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA

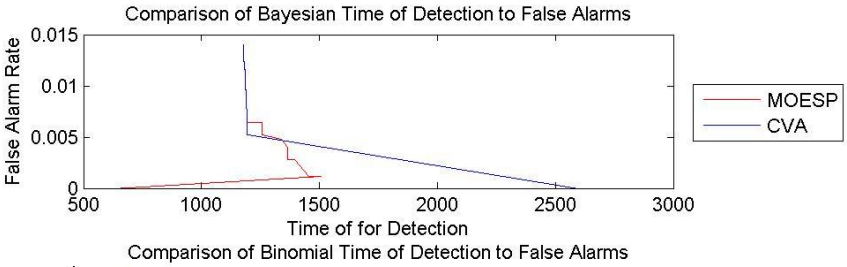

Comparison of SSS Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA
Comparison of Bayesian Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA
Comparison of Binomial Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA

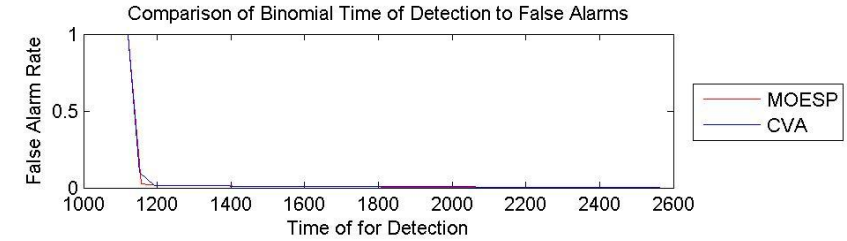

Comparison of SSS Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA
Comparison of Bayesian Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA
Comparison of Binomial Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA

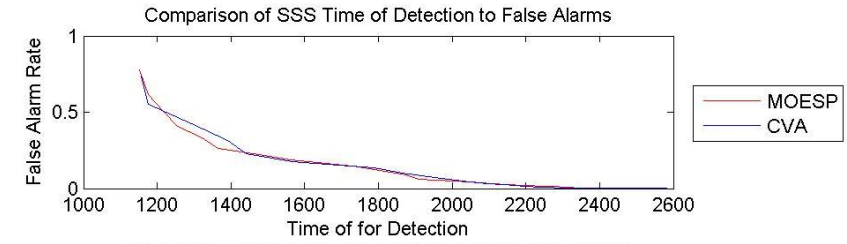

Comparison of SSS Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA
Comparison of Bayesian Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA
Comparison of Binomial Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA

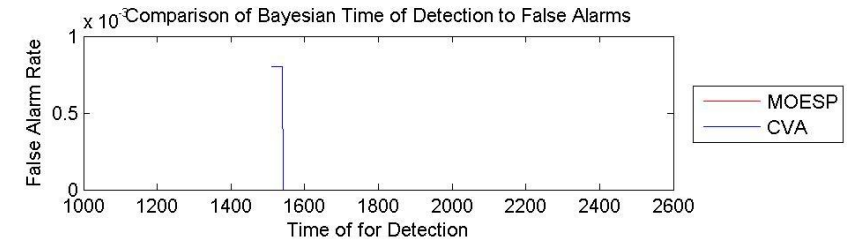

Comparison of SSS Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA
Comparison of Bayesian Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA
Comparison of Binomial Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA

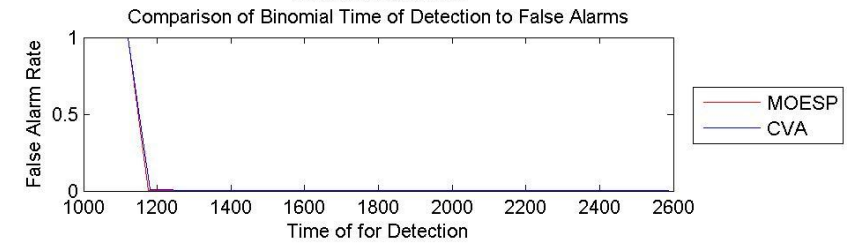

Comparison of SSS Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA

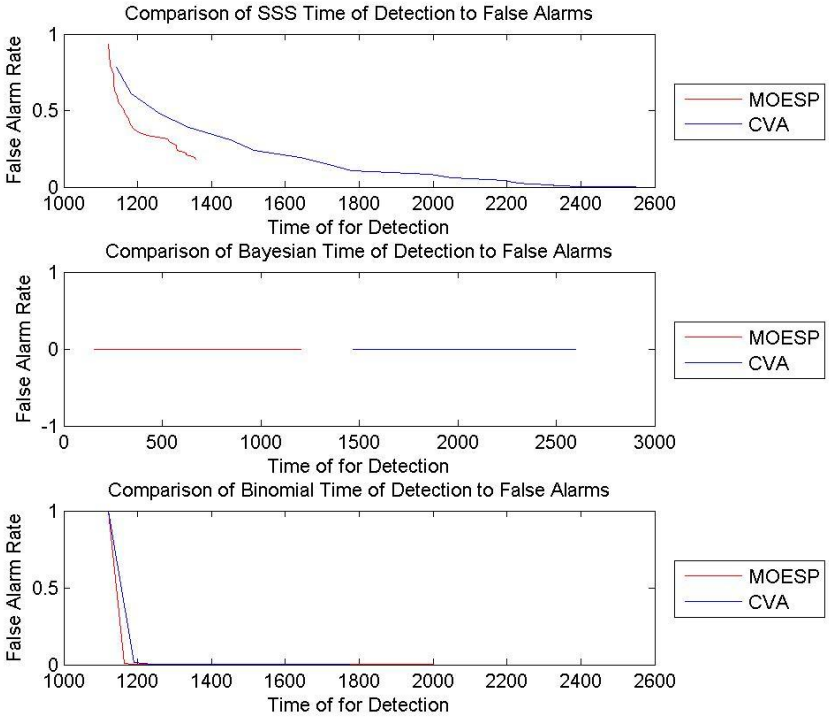

Comparison of Bayesian Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA

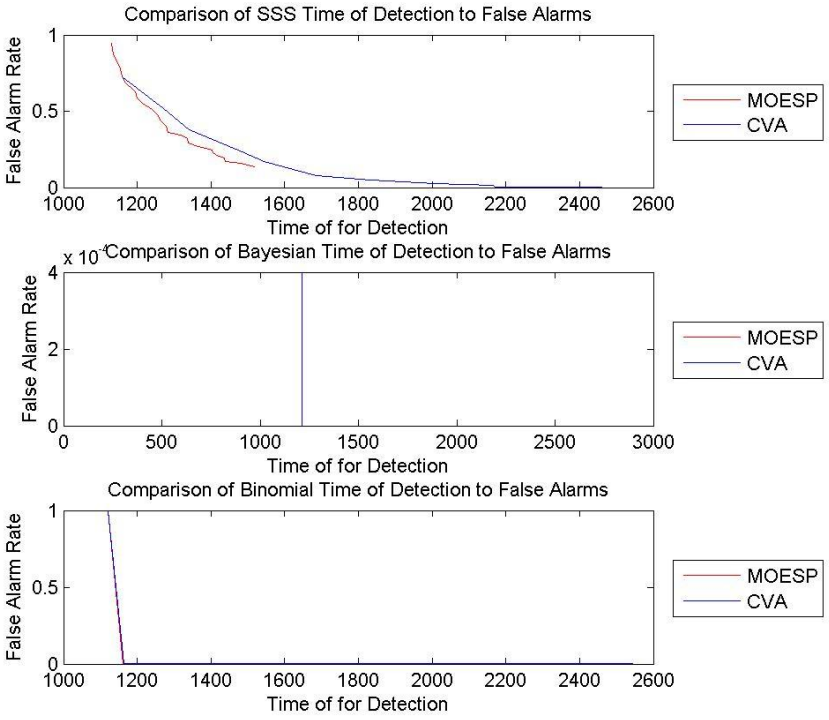

Comparison of Binomial Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA

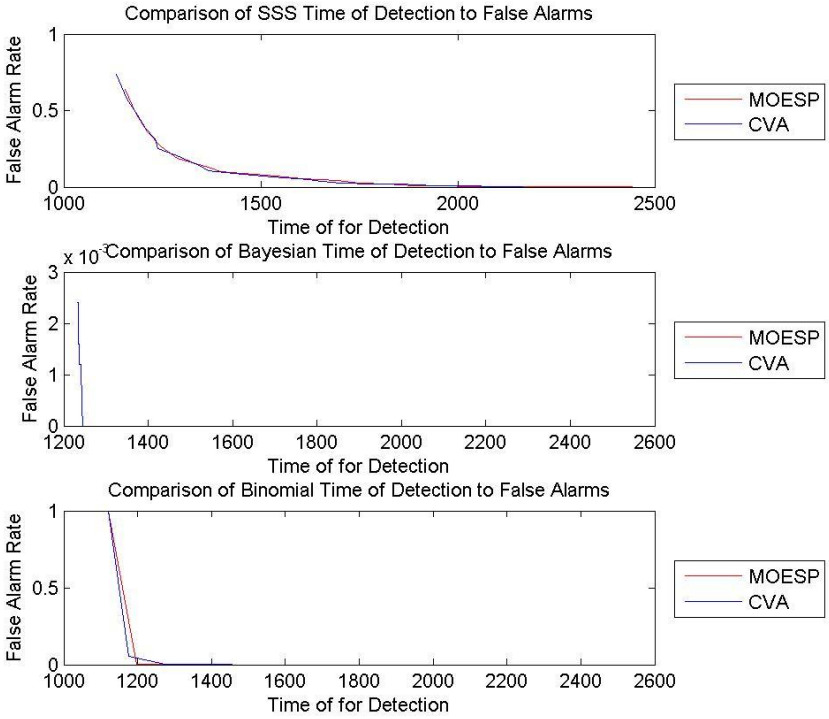

Comparison of SSS Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA

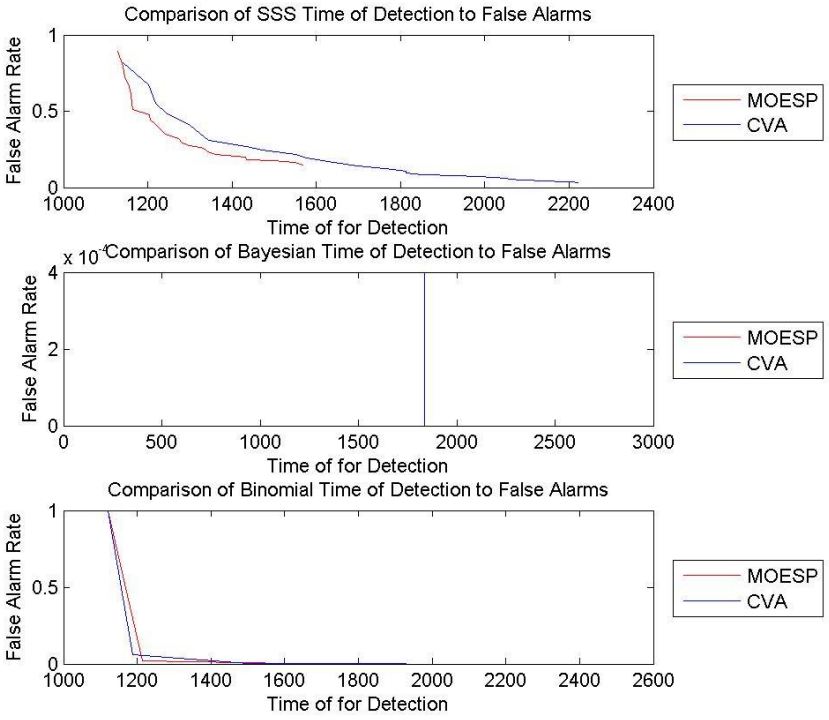

x 10^-3
Comparison of Bayesian Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA

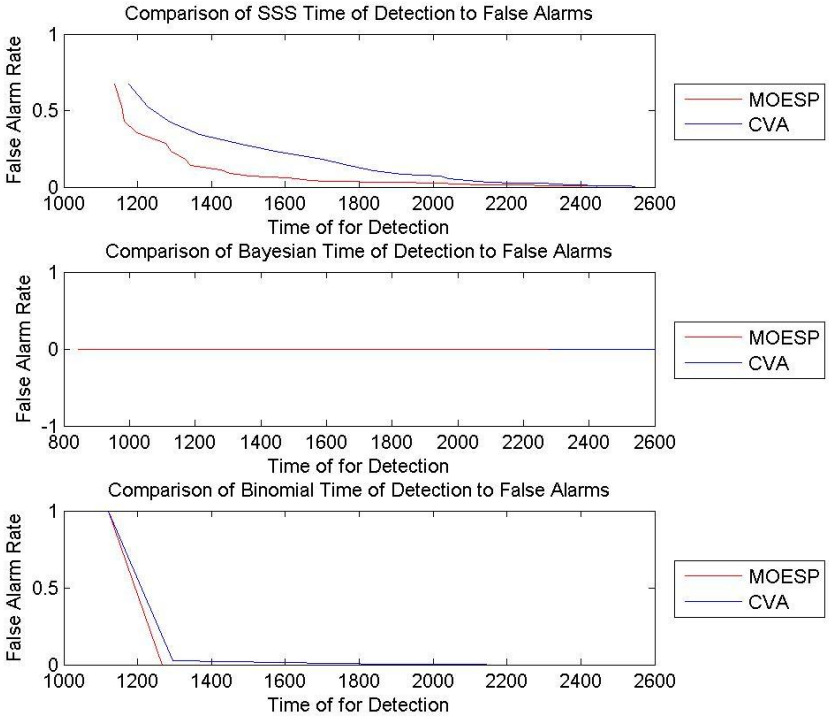

Comparison of Binomial Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA

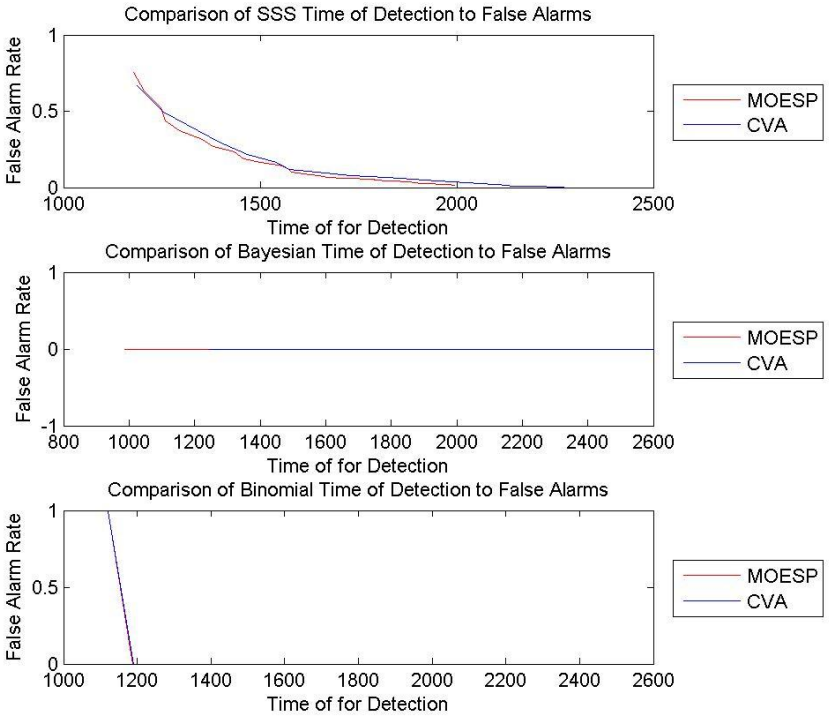

Comparison of SSS Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA
Comparison of Bayesian Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA
Comparison of Binomial Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA

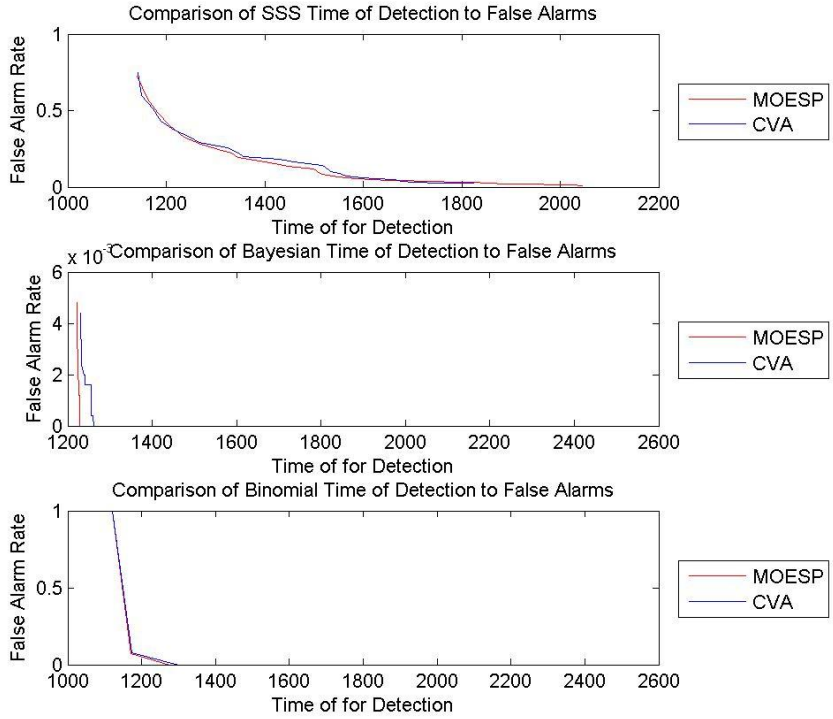

Comparison of SSS Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA
x 10^-4
Comparison of Bayesian Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA
Comparison of Binomial Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA

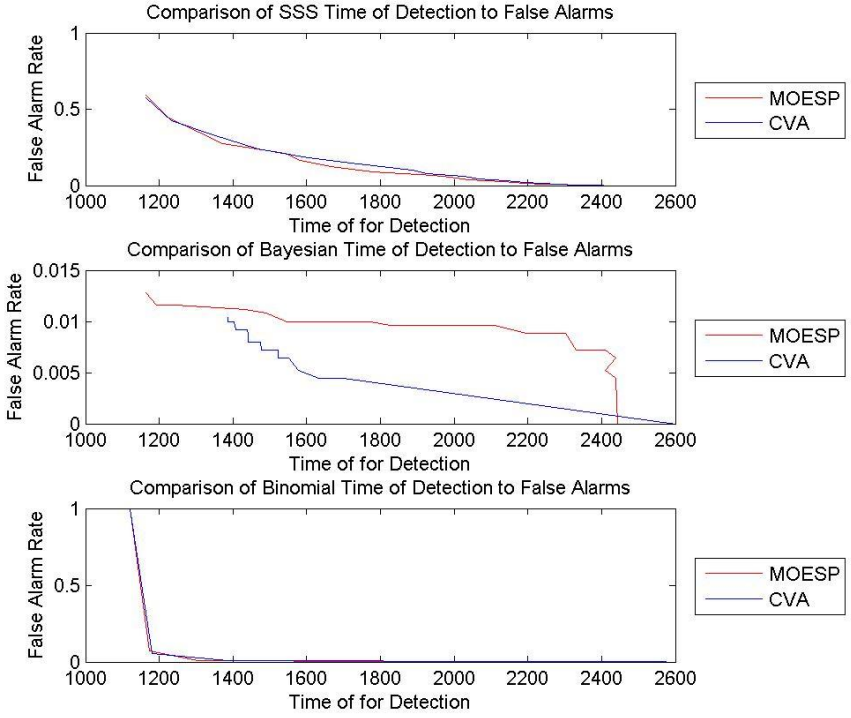

Comparison of SSS Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA
x 10^-3 Comparison of Bayesian Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA
Comparison of Binomial Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA

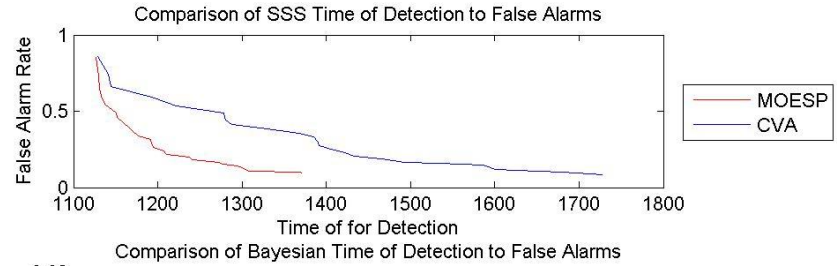

Comparison of SSS Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA
x 10^-4 Comparison of Bayesian Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA
Comparison of Binomial Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA

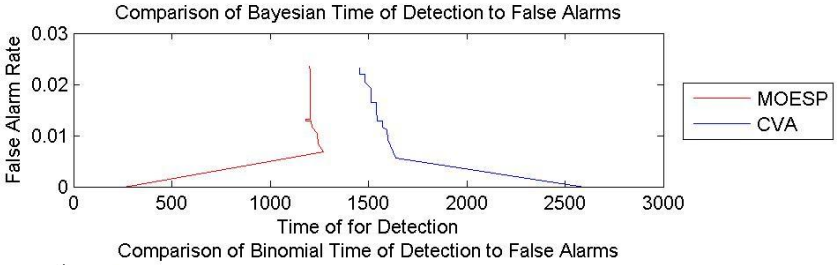

Comparison of SSS Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA
Comparison of Bayesian Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA
Comparison of Binomial Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA

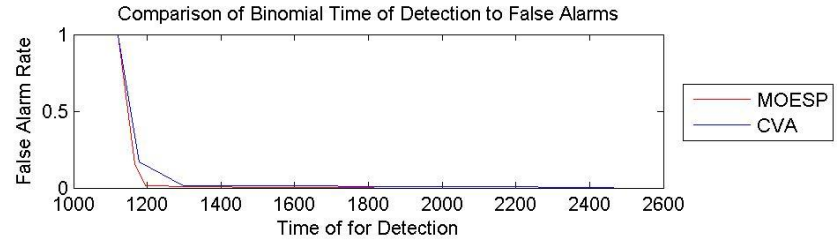

Comparison of SSS Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA
Comparison of Bayesian Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA
Comparison of Binomial Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA

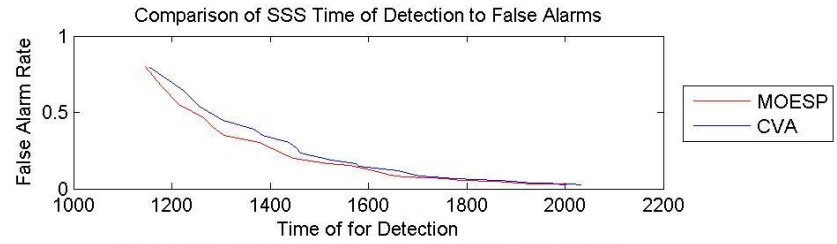

Comparison of SSS Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA
x 10^-3
Comparison of Bayesian Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA
Comparison of Binomial Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA

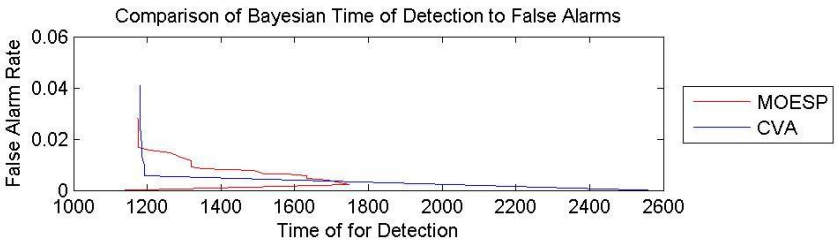

Comparison of SSS Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA
Comparison of Bayesian Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA
Comparison of Binomial Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA

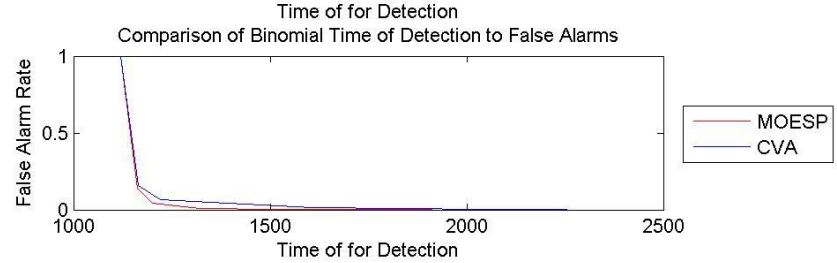

Comparison of SSS Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA


Comparison of Bayesian Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA


Comparison of Binomial Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA


Comparison of SSS Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA


Comparison of Bayesian Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA


Comparison of Binomial Time of Detection to False Alarms
False Alarm Rate
Time of for Detection
MOESP
CVA