



**Calhoun: The NPS Institutional Archive**  
**DSpace Repository**

---

Theses and Dissertations

1. Thesis and Dissertation Collection, all items

---

2007-06

A comparative analysis of multivariate  
statistical detection methods applied to  
syndromic surveillance

Matthew C. Knitt.

Monterey, California. Naval Postgraduate School

---

<http://hdl.handle.net/10945/3417>

*Downloaded from NPS Archive: Calhoun*



Calhoun is a project of the Dudley Knox Library at NPS, furthering the precepts and goals of open government and government transparency. All information contained herein has been approved for release by the NPS Public Affairs Officer.

**Dudley Knox Library / Naval Postgraduate School**  
**411 Dyer Road / 1 University Circle**  
**Monterey, California USA 93943**

<http://www.nps.edu/library>



# NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

## THESIS

**A COMPARATIVE ANALYSIS OF MULTIVARIATE  
STATISTICAL DETECTION METHODS APPLIED TO  
SYNDROMIC SURVEILLANCE**

by

Cecilia X. Hu  
Matthew C. Knitt

June 2007

Thesis Advisor:  
Second Reader:

Ronald D. Fricker, Jr.  
David H. Olwell

**Approved for public release; distribution is unlimited**

THIS PAGE INTENTIONALLY LEFT BLANK

<b>REPORT DOCUMENTATION PAGE</b>			<i>Form Approved OMB No. 0704-0188</i>
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.			
<b>1. AGENCY USE ONLY (Leave blank)</b>	<b>2. REPORT DATE</b> June 2007	<b>3. REPORT TYPE AND DATES COVERED</b> Master's Thesis	
<b>4. TITLE AND SUBTITLE</b> A Comparative Analysis of Multivariate Statistical Detection Methods Applied to Syndromic Surveillance		<b>5. FUNDING NUMBERS</b>	
<b>6. AUTHOR(S)</b> Hu, Cecilia X. and Knitt, Matthew C.		<b>8. PERFORMING ORGANIZATION REPORT NUMBER</b>	
<b>7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)</b> Naval Postgraduate School Monterey, CA 93943-5000		<b>10. SPONSORING/MONITORING AGENCY REPORT NUMBER</b>	
<b>9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES)</b> N/A		<b>11. SUPPLEMENTARY NOTES</b> The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.	
<b>12a. DISTRIBUTION / AVAILABILITY STATEMENT</b> Approved for public release; distribution is unlimited.		<b>12b. DISTRIBUTION CODE</b> A	
<b>13. ABSTRACT (maximum 200 words)</b> Biological terrorism is a threat to the security and well-being of the United States. It is critical to detect the presence of these attacks in a timely manner, in order to provide sufficient and effective responses to minimize or contain the damage inflicted. Syndromic surveillance is the process of monitoring public health-related data and applying statistical tests to determine the potential presence of a disease outbreak in the observed system. Our research involved a comparative analysis of two multivariate statistical methods, the multivariate CUSUM (MCUSUM) and the multivariate exponentially weighted moving average (MEWMA), both modified to look only for increases in disease incidence. While neither of these methods is currently in use in a biosurveillance system, they are among the most promising multivariate methods for this application. Our analysis was based on a series of simulations using synthetic syndromic surveillance data that mimics various types of background disease incidence and outbreaks. We found that, similar to results for the univariate CUSUM and EWMA, the directionally-sensitive MCUSUM and MEWMA perform very similarly.			
<b>14. SUBJECT TERMS</b> Biosurveillance, Multivariate CUSUM, Multivariate EWMA, Statistical Process Control, Syndromic Surveillance		<b>15. NUMBER OF PAGES</b> 95	
		<b>16. PRICE CODE</b>	
<b>17. SECURITY CLASSIFICATION OF REPORT</b> Unclassified	<b>18. SECURITY CLASSIFICATION OF THIS PAGE</b> Unclassified	<b>19. SECURITY CLASSIFICATION OF ABSTRACT</b> Unclassified	<b>20. LIMITATION OF ABSTRACT</b> UL

THIS PAGE INTENTIONALLY LEFT BLANK

**A COMPARATIVE ANALYSIS OF MULTIVARIATE STATISTICAL  
DETECTION METHODS APPLIED TO SYNDROMIC SURVEILLANCE**

Cecilia X. Hu  
Ensign, United States Navy  
B.S., United States Naval Academy, 2006

Matthew C. Knitt  
Ensign, United States Navy  
B.S., United States Naval Academy, 2006

Submitted in partial fulfillment of the  
requirements for the degree of

**MASTER OF SCIENCE IN APPLIED SCIENCE (OPERATIONS RESEARCH)**

from the

**NAVAL POSTGRADUATE SCHOOL  
June 2007**

Authors: Cecilia X. Hu

Matthew C. Knitt

Approved by: Ronald D. Fricker, Jr.  
Advisor

David H. Olwell  
Second Reader

James N. Eagle  
Chairman, Operations Research Department

THIS PAGE INTENTIONALLY LEFT BLANK

## **ABSTRACT**

Biological terrorism is a threat to the security and well-being of the United States. It is critical to detect the presence of these attacks in a timely manner, in order to provide sufficient and effective responses to minimize or contain the damage inflicted. Syndromic surveillance is the process of monitoring public health-related data and applying statistical tests to determine the presence of a disease outbreak in the observed system.

Our research involved a comparative analysis of two multivariate statistical methods, the multivariate CUSUM (MCUSUM) and the multivariate exponentially weighted moving average (MEWMA), both modified to only look for increases in disease incidence. While neither of these methods is currently in use in a biosurveillance system, they are among the most promising multivariate methods for this application.

Our analysis was based on a series of simulations using synthetic syndromic surveillance data that mimics various types of background disease incidence and outbreaks. We found that, similar to results for the univariate CUSUM and EWMA, the directionally-sensitive MCUSUM and MEWMA perform very similarly.



THIS PAGE INTENTIONALLY LEFT BLANK

# TABLE OF CONTENTS

<b>I.</b>	<b>INTRODUCTION .....</b>	<b>1</b>
	A. BACKGROUND INFORMATION.....	1
	B. PROBLEM STATEMENT .....	2
	C. THESIS OUTLINE .....	2
<b>II.</b>	<b>DESCRIPTION OF THE MULTIVARIATE METHODS.....</b>	<b>5</b>
	A. MCUSUM.....	5
	B. MEWMA .....	6
	C. ADAPTIVE REGRESSION .....	8
	D. MODIFIED MULTIVARIATE METHODS .....	10
<b>III.</b>	<b>METHOD PARAMETERS AND COMPARISON METRICS.....</b>	<b>13</b>
	A. CHOOSING PARAMETERS $\lambda$ AND $k$ .....	13
	B. SETTING THRESHOLDS .....	16
	C. PERFORMANCE METRICS .....	16
<b>IV.</b>	<b>MODELING PUBLIC HEALTH DATA.....</b>	<b>19</b>
	A. BACKGROUND DISEASE INCIDENCE .....	19
	B. MODELING DISEASE OUTBREAK BEHAVIOR.....	21
<b>V.</b>	<b>THE SIMULATIONS.....</b>	<b>25</b>
	A. CASES .....	25
	B. SIMULATION SEQUENCE OF EVENTS.....	27
	C. PERFORMANCE RESULTS.....	27
<b>VI.</b>	<b>DISCUSSION AND CONCLUSIONS .....</b>	<b>31</b>
	A. DISCUSSION .....	31
	B. FUTURE RESEARCH .....	32
<b>APPENDIX A. ....</b>		<b>35</b>
	A. MEWMA (INITIAL) .....	35
	B. MCUSUM (INITIAL) .....	36
	C. DATA GENERATION AND ADAPTIVE REGRESSION TOOL .....	37
	D. MODIFIED MEWMA FOR SYNDROMIC SURVEILLANCE DATA..	39
	E. MODIFIED MCUSUM FOR SYNDROMIC SURVEILLANCE DATA.....	43
	F. MAIN SIMULATION ENGINE.....	47
<b>APPENDIX B. ....</b>		<b>51</b>
<b>APPENDIX C. ....</b>		<b>61</b>
	A. AVERAGE TIME TO FIRST SIGNAL.....	62
	B. ATFS GIVEN TRUE SIGNAL .....	65
	C. PERCENT MISS.....	68
<b>LIST OF REFERENCES .....</b>		<b>71</b>
<b>INITIAL DISTRIBUTION LIST .....</b>		<b>73</b>

THIS PAGE INTENTIONALLY LEFT BLANK

## LIST OF FIGURES

Figure 1.	Transforming one year of syndromic surveillance data using adaptive regression – Parameters: $\beta = 90$ , $A = 20$ , $\sigma = 10$ , $n = 35$ ..... 11
Figure 2.	Plot of the relative ARL performance for multiple MCUSUM instances against the MEWMA with $\lambda = 0.2$ illustrating evidence that $k = 0.74$ makes the MCUSUM perform the closest to the MEWMA. .... 14
Figure 3.	A second plot of the relative ARL performance for multiple MCUSUM instances against the MEWMA with $\lambda = 0.2$ illustrating evidence that $k = 0.74$ makes the MCUSUM perform the closest to the MEWMA. This plot illustrates the behavior of the MCUSUM for more extreme $k$ values. ... 15
Figure 4.	An example of generated background disease incidence data with parameters: $\beta = 90$ , $A = 20$ , $\sigma = 10$ ..... 21
Figure 5.	An example outbreak with duration = 9 days, peak = 45, startday = 0..... 22
Figure 6.	Plots for Case 1 simulation: (a) ATFS Given True Signal, (b) Percent Miss.. 28
Figure 7.	Plots for cases 1, 7, and 13 (solid lines - MEWMA, dashed lines - MCUSUM). All three cases have $\beta = 90$ , $\sigma = 30$ , and $p = 45$ , illustrating the invariance of method performance with respect to changes in the amplitude component. .... 29

THIS PAGE INTENTIONALLY LEFT BLANK

**LIST OF TABLES**

Table 1. Parameter Values for  $\beta = 90$  .....25  
Table 2. Simulation Parameters.....26

THIS PAGE INTENTIONALLY LEFT BLANK

## LIST OF SYMBOLS, ACRONYMS AND/OR ABBREVIATIONS

$h$	control limit or threshold
$k$	reference value parameter for MCUSUM
$\lambda$	smoothing parameter for MEWMA
ARL	average run length
$ARL_0$	in-control average run length
ATFS	average time to first signal
$ATFS_0$	in-control average time to first signal
CDC	Centers for Disease Control and Prevention
<i>i.i.d.</i>	independent and identically distributed
MCUSUM	multivariate cumulative sum
MEWMA	multivariate exponentially weighted moving average
OOB	out-of-control
SPC	statistical process control



THIS PAGE INTENTIONALLY LEFT BLANK

## **ACKNOWLEDGMENTS**

We would like to extend our deepest gratitude to Dr. Ronald D. Fricker, Jr. at the Naval Postgraduate School, our dedicated thesis advisor, for his invaluable advice, suggestions, and guidance during the course of this research. With his constant mentorship and vast insight in the field of syndromic surveillance, we have gained a significantly broader and more comprehensive understanding of this field. We appreciate and laud his dedication toward exploring this sector of public health.

THIS PAGE INTENTIONALLY LEFT BLANK

## EXECUTIVE SUMMARY

Syndromic surveillance is the process of prospectively monitoring health data in order to alert public health officials to a suspected increase in disease incidence in the general population as soon as possible. Prior to the advent of syndromic surveillance, public health surveillance had generally been limited to retrospectively analyzing medical data. However, the retrospective analyses often proved to be too time-consuming, failing to signal an alert within a reasonable time period. Timely alarms may help prevent spread of the disease and has the potential to significantly improve effective response to an outbreak or bioterrorism attack, particularly if the disease is contagious. The main goal when using these methods is to detect outbreaks as quickly as possible, in order to optimize the response time for the public health officials.

In an effort to reduce the delay time, public health organizations have begun to implement syndromic surveillance systems that are intended to provide for earlier detection, often applying existing statistical process control (SPC) methods such as the cumulative sum (CUSUM) and the exponentially weighted moving average (EWMA) control charts. These control charts monitor the distribution of a test statistic and signal when an out-of-control threshold has been exceeded. While univariate versions of the CUSUM and EWMA methods are also viable analysis techniques in syndromic surveillance, this thesis concentrates on evaluating a multivariate CUSUM (MCUSUM) and a multivariate EWMA (MEWMA), both modified to only look for increases in disease incidence. While neither of these methods is currently in use in a biosurveillance system, they are among the most promising multivariate methods for this application.

This thesis compares and contrasts the relative performance of the directionally sensitive MCUSUM and MEWMA on simulated data designed to mimic various types of background disease incidence and outbreaks. Syndromic surveillance data is inherently non-stationary and positively correlated – characteristics which do not coincide with traditional *i.i.d.* SPC assumptions. To account for these characteristics, the MCUSUM and MEWMA are applied to the residuals of an adaptive regression with a sliding baseline.

In order to appropriately compare the two, each method's various parameters ( $\lambda$  for MEWMA and  $k$  for MCUSUM) were first determined such that both methods performed as similarly as possible under the standard *i.i.d.* assumptions. Following recommendations in the literature, we set  $\lambda = 0.2$  throughout all the simulations and through an iterative process found the  $k$  for which the MCUSUM performed as similarly to the MEWMA with  $\lambda = 0.2$  as possible ( $k = 0.74$ ).

Once the methods' parameters were determined, control limits were chosen to produce a certain "in-control" average run length (ARL) or average time to first signal (ATFS), denoted by  $ARL_0$  or  $ATFS_0$ . That is, the thresholds were set so that the two methods performed equally well in the absence of an outbreak. This guaranteed a fair comparison of the two methods when evaluating their performance at detecting various types of outbreaks. For the simulations in this study, we found control limits that set the in-control ARL and ATFS values are within one day of a 100 day target.

The accepted metric in SPC practice is the ARL, which measures the average number of time units it takes before a method signals an alarm. However, in the context of syndromic surveillance, the transient nature of disease outbreaks often precludes the use of the ARL metric. That is, in the industrial SPC world, a common and reasonable assumption is that once an "out-of-control" condition occurs, it persists until a detection method signals. Hence, using the "out-of-control" ARL is an effective metric for comparing the relative performance of methods to detect various out-of-control conditions. However, in syndromic surveillance, disease outbreaks start, peak, and then ultimately subside and go away. Thus, it is important to also assess performance in terms of a procedure's ability to detect an outbreak within the period of the outbreak.

In this study, two metrics are used to assess the methods' performance on syndromic surveillance data. The first is the percentage of the time that a method does not detect during an outbreak period, which provides insight regarding the utility of the signals. If the detections occur after the outbreak has ended, then the method is not providing a useful signal. The second metric is the conditional average time to first signal (ATFS) given that the method did detect during the outbreak period, which illustrates how fast the method is signaling when it detects an outbreak.

The main simulations in this thesis were conducted on 18 cases with varying amplitude, standard deviation, and outbreak peak. These cases generated ATFS, ATFS given a true signal, and Percent Miss statistics along with their respective standard errors. We found that, similar to results for the univariate CUSUM and EWMA, the directionally-sensitive MCUSUM and MEWMA perform almost identically under all of the 18 cases evaluated. While there seems to be no performance advantage of one method over the other, the MEWMA is preferred for procedural simplicity as compared to the MCUSUM.

THIS PAGE INTENTIONALLY LEFT BLANK

# I. INTRODUCTION

## A. BACKGROUND INFORMATION

Syndromic surveillance is the process of monitoring health data in order to alert public health officials to a suspected increase in disease incidence in the general population as soon as possible. Traditional biosurveillance methods have been limited to retrospectively analyzing medical and public health data to verify the existence of a disease outbreak (Shmueli 2006). Via these traditional methods, the process of collecting and analyzing can take days, even weeks, before definitively concluding that an outbreak has occurred and alerting officials. The timeliness of the alarm may help prevent spread of the disease and has the potential to significantly improve effective response to an outbreak or bioterrorism attack, particularly if the disease is contagious. It may also improve medical response to the way a disease is treated, especially diseases with an exponentially decreasing survival probability as time elapses. For example, the probability of surviving anthrax if caught in the first three days compared to a diagnosis beyond the three day threshold drops significantly (Goldenberg 2002).

In recent times, public health organizations have begun augmenting their retrospective biosurveillance processes with prospective methods that are intended to provide for earlier detection, minimizing the lag between disease outbreak and an appropriate containment response. These new biosurveillance efforts apply control chart methods borrowed from statistical process control (SPC), such as the cumulative sum (CUSUM) and the exponentially weighted moving average control chart (EWMA). These control charts are used to detect changes in the distribution of a test statistic, and signal when a pre-determined out-of-control threshold has been exceeded, which provides some evidence that an outbreak has occurred (Joner 2006). See Montgomery (2001) for a review of the fundamentals of statistical process control, including descriptions of the standard CUSUM and EWMA procedures. For a review of the use of control charts in the broader context of health care and public health surveillance, see Woodall (2006).



When monitoring for outbreaks in a single region or a single time series, the univariate EWMA and the one-sided univariate CUSUM can be effective methods for achieving timely detections. Woodall and Ncube (1985) first proposed the application of simultaneous univariate CUSUMs in a multivariate application. Multiple simultaneous univariate procedures have the advantages of ease of implementation and interpretation, but then can be less sensitive to some types of changes when compared to multivariate methods. Thus, when the data are available from multiple regions or multiple time series, a multivariate method has the potential to be more effective because it uses information from multiple data streams or regions.

## **B. PROBLEM STATEMENT**

This thesis compares and contrasts the relative performance of the multivariate CUSUM (MCUSUM) and multivariate EWMA (MEWMA) methods proposed by Fricker (2007) and Joner et al. (2006), respectively. While neither of these methods is currently in use in a syndromic surveillance system, they are among the most promising multivariate methods for such applications. The comparisons were conducted using simulated syndromic surveillance data in a variety of scenarios designed to mimic the main features of real syndromic surveillance data. The goal of this research was to gain some insight into the relative performance of the two methods, particularly whether there are some scenarios under which one method is to be preferred over another.

## **C. THESIS OUTLINE**

The thesis is organized as follows: In Chapter II, the standard MCUSUM and MEWMA methods are described, including the modifications necessary to make them directionally sensitive. Chapter III describes how the procedures' parameters were determined so that both methods perform similarly under the standard *i.i.d.* assumptions. Chapter IV describes how the simulated background disease incidence counts and outbreaks were generated. Chapter V defines the set of simulation scenarios and presents the results from the comparative analysis. Chapter VI concludes with a summary of the results and offers some recommendations on the use of performance metrics as applied to

syndromic surveillance. Appendix A includes all of the MATLAB code used for the simulations in this thesis. Appendix B is a collection of all results plots from the main simulations. Appendix C shows the Microsoft Excel spreadsheet that we used to assess the statistical significance of the observed differences between the two methods.

THIS PAGE INTENTIONALLY LEFT BLANK

## II. DESCRIPTION OF THE MULTIVARIATE METHODS

The one-sided univariate CUSUM control chart can be effective for the timely detection of an outbreak in a single region. However, when data streams are available from multiple regions, a multivariate method has the potential to be more effective. Such a multivariate monitoring method appropriately combines information from several regions into a single control chart. Multiple univariate procedures have the advantages of ease of implementation and interpretation, but may be less sensitive to certain changes as compared to multivariate forms (Fricker 2006). The multivariate approach also avoids multiple-testing, which can inflate the false-alarm rate, as well as increase the power to detect triggers that the univariate charts were not able to pick up (Shmueli 2006).

Many multivariate methods are directionally invariant, meaning that they are designed to detect changes in a mean vector equally well in all directions (Fricker 2006). While directional invariance is important when the objective is to detect a departure in any direction from a specific vector, the primary goal in syndromic surveillance is to detect increases in one or more disease incidence counts. Simply put, in syndromic surveillance applications, it is only relevant to flag increases; there is no need to signal if the disease count is lower than expected. Therefore, in this research we compare and contrast two directional multivariate methods, the directionally sensitive multivariate CUSUM from Fricker (2007) and a directional MEWMA control chart from Joner et al. (2006).

The basic methods are described below, how they were applied to the syndromic surveillance data is described in Chapter III, Section C, and the MATLAB code implementing the two methods in the simulations is contained in Appendix A.

### A. MCUSUM

Suppose that we observe  $\mathbf{X}_i = \{X_1, X_2, \dots, X_p\}$ , a  $p$ -dimensional set of observations at time  $i$ . In the syndromic surveillance case, this may be, for example, a

vector of chief complaint counts at the  $p$  different hospitals in the area. Crosier (1988) introduced a multivariate CUSUM that at each time  $i$  calculates the statistic  $\mathbf{S}_i$  as

$$\mathbf{S}_i = (\mathbf{S}_{i-1} + \mathbf{X}_i - \boldsymbol{\mu})(1 - k/C_i), \text{ if } C_i > k, \quad (2.1)$$

where  $\boldsymbol{\mu}$  is the target vector representing the mean of the process when it is “in-control,”  $k$  is a predetermined statistical distance,

$$C_i = \sqrt{(\mathbf{S}_{i-1} + \mathbf{X}_i - \boldsymbol{\mu})' \boldsymbol{\Sigma}^{-1} (\mathbf{S}_{i-1} + \mathbf{X}_i - \boldsymbol{\mu})}, \quad (2.2)$$

and  $\boldsymbol{\Sigma}$  is the covariance matrix of the “in-control” observation streams. If  $C_i \leq k$ , the process resets  $\mathbf{S}_i = \mathbf{0}$ . The MCUSUM starts with  $\mathbf{S}_0 = \mathbf{0}$  and sequentially calculates

$$Y_i = \sqrt{\mathbf{S}_i' \boldsymbol{\Sigma}^{-1} \mathbf{S}_i}. \quad (2.3)$$

The MCUSUM triggers an alarm when  $Y_i$  exceeds a pre-determined threshold  $h$  that is chosen to achieve a desired in-control average time to first signal (ATFS). Crosier states that selecting an appropriate parameter  $k$  “[i]n the univariate [CUSUM] case, the quantity  $\mathbf{S}_{i-1} + \mathbf{X}_i - \boldsymbol{\mu}$  is shrunk towards  $\mathbf{0}$  by  $k$  standard deviations. If this is to hold for the multivariate case,  $\mathbf{k}$  must satisfy  $\mathbf{k}' \boldsymbol{\Sigma}^{-1} \mathbf{k} = k^2$  – that is,  $\mathbf{k}$  must be of length  $k$ , where length is defined by using the covariance matrix  $\boldsymbol{\Sigma}$ ” (Crosier 1988).

Fricker (2007) modified Crosier’s MCUSUM to make it directionally sensitive to increases in one or more dimensions. Directionality is achieved by bounding each component of the of the cumulative sum vector in Equation (2.1) by zero, much like the univariate CUSUM is reflected at zero. That is, when  $C_i > k$ , replace Equation (2.1) with

$$S_{i,j} = \max\left[0, (S_{i-1,j} + X_{i,j} - \mu_j)(1 - k/C_i)\right], \quad (2.4)$$

for each component of the  $\mathbf{S}_i$  vector,  $j = 1, \dots, p$ .

## B. MEWMA

Lowry et al. (1992) introduced an alternative to the MCUSUM called the multivariate exponentially weighted moving average (MEWMA). It is a simple

expansion of the univariate EWMA through the use of vectors. Joner et al. (2006) subsequently modified the MEWMA to make it directionally sensitive.

As with the MCUSUM, assume we observe  $\mathbf{X}_i = \{X_1, X_2, \dots, X_p\}$ . Denote the in-control mean for  $\mathbf{X}_i$  by the vector  $\boldsymbol{\mu}$  and let  $\boldsymbol{\Sigma}$  be the covariance matrix. Assume that  $\boldsymbol{\mu}$  and  $\boldsymbol{\Sigma}$  are known. These quantities can be estimated from historical data. Calculate

$$\mathbf{Z}_i = \begin{cases} \mathbf{max}[\lambda(\mathbf{X}_i - \boldsymbol{\mu}) + (1-\lambda)\mathbf{Z}_{i-1}, \mathbf{0}] & \text{for } i > 0 \\ \mathbf{0} & \text{for } i = 0 \end{cases}, \quad (2.5)$$

where the **max** operator is applied to the argument vectors component by component, effectively bounding each element of  $\mathbf{Z}_i$  by 0.  $\mathbf{Z}_i$  is essentially a weighted average of the current observation standardized around  $\mathbf{0}$  and the previous  $\mathbf{Z}$  statistic. Since all elements of  $\mathbf{Z}_i$  are bounded by zero, the resulting chart will tend to signal only as a result of increases in the observed incidence rate (Joner 2006). The parameter  $\lambda > 0$  is known as the smoothing parameter and  $\lambda$  controls the weight assigned to the new observation vector.

In Joner et al. (2006), the covariance matrix for  $\mathbf{Z}_i$  is shown to be

$$\boldsymbol{\Sigma}_{\mathbf{Z}_i} = \frac{\lambda [1 - (1-\lambda)^{2i}]}{2-\lambda} \boldsymbol{\Sigma}. \quad (2.6)$$

Taking the limit as  $i \rightarrow \infty$ ,

$$\boldsymbol{\Sigma}_{\mathbf{Z}_\infty} = \frac{\lambda}{2-\lambda} \boldsymbol{\Sigma}. \quad (2.7)$$

$\boldsymbol{\Sigma}_{\mathbf{Z}_\infty}$  is then used to calculate the test statistic  $\text{MEW}_i$  where

$$\text{MEW}_i = \mathbf{Z}_i' \boldsymbol{\Sigma}_{\mathbf{Z}_\infty}^{-1} \mathbf{Z}_i. \quad (2.8)$$

The MEWMA process signals an alarm whenever  $\text{MEW}_i$  exceeds a predetermined threshold value  $h$  which is set to achieve a desired ATFS. If  $\text{MEW}_i$  does

not exceed  $h$ , then the MEWMA process iterates through the next time step with a new observation vector, recalculating the test statistic, and continuing until such time as the  $MEW_i > h$ .

The MEWMA (and MCUSUM) assumes that the observations are generated by a  $p$ -dimensional multivariate normal distribution,  $N_p(\boldsymbol{\mu}, \boldsymbol{\Sigma})$  over the entire period of observation. In particular, it assumes that the mean vector  $\boldsymbol{\mu}$  does not change over time (i.e., it is “stationary”), an assumption that generally does not apply to syndromic surveillance.

To account for the non-stationary nature of syndromic surveillance data, the MCUSUM and MEWMA are not applied to the raw data. Rather, they are applied to the standardized residuals from an “adaptive regression with sliding baseline” (Burkom et al. 2006) designed to first remove any systematic trends from the data (such as seasonal fluctuations in disease incidence).

### C. ADAPTIVE REGRESSION

Systematic trends, such as seasonal cycles and other patterns, are a natural characteristic of syndromic surveillance data. Not only are the data non-stationary, but the trends also tend to occur over all of the dimensions in the scenario, showing that positive correlation exists. The non-stationary, correlated nature of syndromic surveillance does not coincide with the traditional *i.i.d.* SPC assumptions, which is to say that these methods, as they stand, assume that the data *do not* contain such trends (Fricker 2006).

One solution to this is to perform an adaptive regression on the data to forecast the next day’s observation and then apply the MEWMA and MCUSUM methods to the forecast errors. Examples of this and similar approaches in the literature include Brillman et al. (2005), who apply the CUSUM to prediction errors; the CDC’s cyclical regression models discussed in Hutwagner et al. (2003); log-linear regression models in Farrington et al. (1996); and time series models in Reis et al. (2003). See Shmueli (2006) for additional discussion of the use of regression and time series methods for syndromic

surveillance and Burkom et al. (2006) for a comparison of two regression-based methods and an exponential smoothing method using adaptive regression in the context of syndromic surveillance. Also see Lotze et al. (2006) for a detailed discussion of preconditioning applied to syndromic surveillance data.

One option for handling the systematic component in syndromic surveillance data is to use the “adaptive regression model with sliding baseline” of Burkom et al. (2006). It is parameterized as follows: Let  $X_i$  be the observation (say chief complaint count on day  $i$ ). These observations are then regressed against time, for some fixed number of time periods  $n$  (Burkom et al. used an 8-week period,  $n = 56$ ). Then the regression would look like

$$X_i = \beta_0 + \beta_1 i + \varepsilon \quad (2.9)$$

where  $\beta_0$  is an intercept term,  $\beta_1$  is the slope, and  $\varepsilon$  is the error term. The model is fit using the usual ordinary least squares approach. The estimates for each time period  $i, \dots, i - (n - 1)$ , where time is always relative to the current observation, are

$$\hat{X}_i = \hat{\beta}_0 + \hat{\beta}_1 i, \quad (2.10)$$

where  $\hat{\beta}_0$  is the estimated intercept and  $\hat{\beta}_1$  is the estimated slope. The forecast error for time  $i + 1$  is

$$r_{i+1} = X_{i+1} - [\hat{\beta}_0 + \hat{\beta}_1 (i+1)]. \quad (2.11)$$

Within the framework of syndromic surveillance, at each time  $i$  we refit the models and, if the models fit well, then the  $r_i$ s should be “small” according to mean square deviation.

We extend this approach for multivariate data by applying the regression model in Equation (2.9) to each dimension  $j$  independently and then calculating the respective forecast errors as shown in Equations (2.12), (2.13), and (2.14).

$$X_{i,j} = \beta_{0,j} + \beta_{1,j} i + \varepsilon_j \quad (2.12)$$

$$\hat{X}_{i,j} = \hat{\beta}_{0,j} + \hat{\beta}_{1,j} i \quad (2.13)$$



$$r_{i+1,j} = X_{i+1,j} - [\hat{\beta}_{0,j} + \hat{\beta}_{1,j}(i+1)] \quad (2.14)$$

Dunfee and Hegler (2007) studied the performance of adaptive regression in one dimension to determine “optimal” sliding baselines ( $n$ ) for different background scenarios using a tool similar to the data generation and adaptive regression tool described in Appendix A. The “optimal” sliding baselines were chosen by inspecting a plot of the mean squared deviation from the adaptive regressions over various  $n$  values and choosing the smallest  $n$  that achieved close to the minimum mean squared deviation. Since our multivariate extension is a sequence of independent adaptive regressions in each dimension, we apply these “optimal” sliding baseline values to the simulation cases described in Chapter IV.

#### **D. MODIFIED MULTIVARIATE METHODS**

As was previously described, the MCUSUM and MEWMA are applied to the residuals from the adaptive regression models in order to remove the systematic trends from the data. Figure 1 illustrates this transformation from syndromic surveillance counts to residuals, where on the left side in Figure 1 the systematic trends are clearly evident in the data but on the right side the residuals do not show any visual evidence of the trends. The procedure for simulating syndromic surveillance data is described in Chapter IV, Section A.

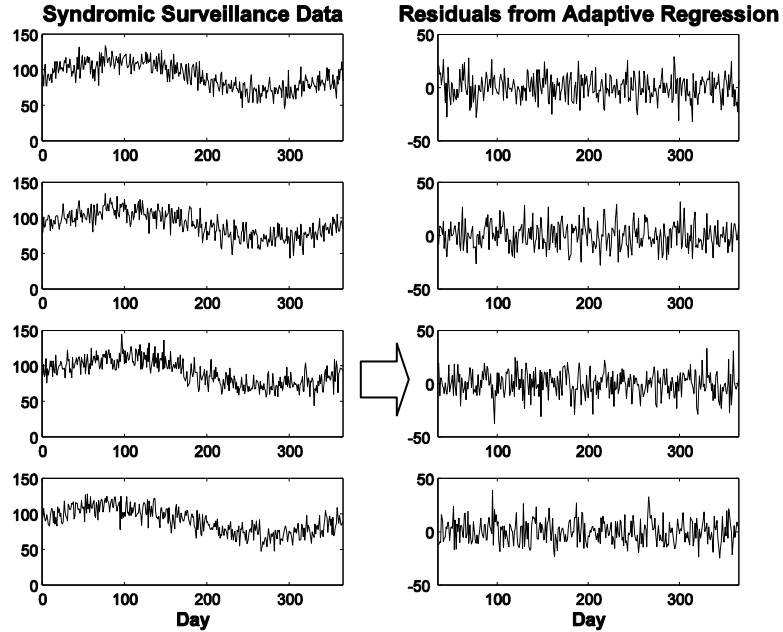


Figure 1. Transforming one year of syndromic surveillance data using adaptive regression – Parameters:  $\beta = 90$ ,  $A = 20$ ,  $\sigma = 10$ ,  $n = 35$

The MCUSUM and MEWMA can now be applied to these residuals. Assuming that there is enough historical data in place to regress over the “optimal” sliding baseline, the modified versions of the methods include the following sequence of events:

- Observe the next day’s count.
- Apply the multivariate adaptive regression technique to the data and calculate the residual value,  $r_{i,j}$ , for each dimension  $j$ , where  $i$  is the current day.
- Update the MCUSUM or MEWMA statistic using the current day’s residuals.
- If the control limit threshold is exceeded, signal an alarm. If the threshold is not exceeded, repeat the process.

This modified version introduces a complication in implementing the methods. The methods are no longer run on the original data, so different input parameters are needed. Both the MCUSUM and MEWMA require the in-control covariance matrix of the residuals as inputs. Although the correlation is removed from the data after adaptive regression, the standard deviation of the residuals is slightly greater than the original standard deviation from the noise in the data. In order to overcome this problem, we

estimate the standard deviation of the residuals by generating background disease incidence data (no outbreaks) and run the adaptive regression technique for many days. We used the sample standard deviation from these residuals,  $\hat{\sigma}_R$ , as an estimate of the true standard deviation. The in-control covariance matrix input is now  $\Sigma = \hat{\sigma}_R^2 \cdot \mathbf{I}$ . To accommodate for this change, we standardize the residuals by dividing each  $r_{i,j}$  by  $\hat{\sigma}_R$ . Instead of running the methods on  $r_i = \{r_{i,1}, r_{i,2}, r_{i,3}, r_{i,4}\}$ , we will run them on  $\bar{r}_i = \{r_{i,1}/\hat{\sigma}_R, r_{i,2}/\hat{\sigma}_R, r_{i,3}/\hat{\sigma}_R, r_{i,4}/\hat{\sigma}_R\}$  on day  $i$ .

The MATLAB code for these modified procedures is in Appendix A.

### III. METHOD PARAMETERS AND COMPARISON METRICS

Before it is possible to compare the performance of the MCUSUM and the MEWMA, this chapter describes how the methods' various parameters were determined. It also describes the metrics used to compare and contrast the methods' performance in the syndromic surveillance problem.

#### A. CHOOSING PARAMETERS $\lambda$ AND $k$

In order to appropriately compare the MEWMA to the MCUSUM, we determined the parameter settings such that both methods performed as similarly as possible under the standard *i.i.d.* assumptions. Given these parameter settings, we then compared and contrasted their performance under various syndromic surveillance scenarios (Chapter V). Since we will subsequently use these methods to monitor the forecast residuals from adaptive regression models, we expect that the residuals of good regression models should be approximately independently normally distributed. Thus, this initial work helped us to understand how to apply these methods in this application.

For these simulations, the data was modeled as independent identically distributed multivariate  $N(\mathbf{0}, \Sigma)$  observations (in four dimensions). The out-of-control conditions were modeled as a uniform increase in the mean vector across all four dimensions. Because these assumptions do not include the characteristics of public health data, we used the ARL metric to compare performance.

Following the recommendation of Joner and Montgomery, we set  $\lambda = 0.2$  throughout all of the simulations (2006; 2001) and then control limits were determined so that the MEWMA achieved an in-control average run length (ARL) of 100 days. The ARL measures the average number of time units it takes before a method signals an alarm. We chose this in-control ARL value for both methods standardizing them for our controlled comparisons. We assume the methods will have similar relative performance results regardless of the specific value of the in-control ARL. Depending upon the context of the observed system in the real world, the in-control ARL can be fixed to

achieve any desired false alarm rate. For a discussion on setting control limit thresholds, see Chapter III, Section B. Setting  $\lambda = 0.2$  is consistent with the recommended range for  $\lambda$  for the univariate EWMA (see Montgomery 2001, for example). Similarly, given a value of  $k$ , control limits can be chosen for the MCUSUM to achieve an in-control ARL of 100 days. The issue then was to find the value of  $k$  such that the MCUSUM performed as similarly to the MEWMA as possible.

After initially trying  $k$  values ranging from  $k = 0.2$  to  $k = 1.8$ , as shown in Figure 2 we determined that  $k = 0.74$  gives the closest MCUSUM ARL performance to the MEWMA with  $\lambda = 0.2$ , where we set all of the components of the  $\mathbf{k}$  vector equally (e.g., for  $k = 0.74$  we set  $\mathbf{k} = \{0.37, .037, 0.37, 0.37\}$ ).

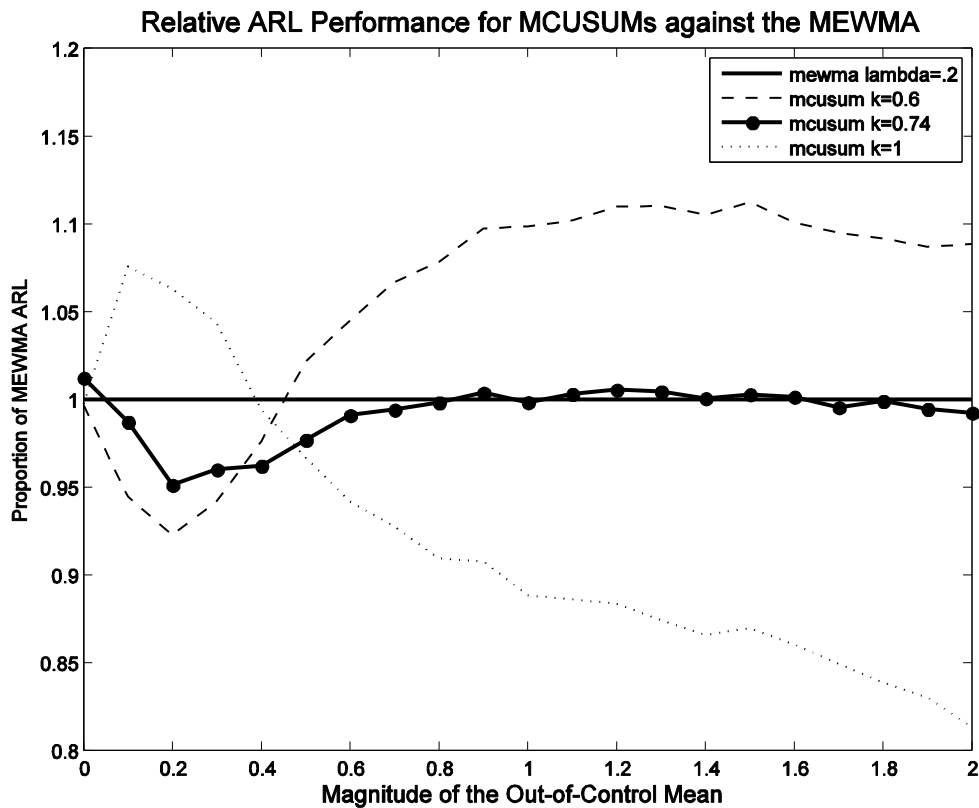


Figure 2. Plot of the relative ARL performance for multiple MCUSUM instances against the MEWMA with  $\lambda = 0.2$  illustrating evidence that  $k = 0.74$  makes the MCUSUM perform the closest to the MEWMA.

As Figures 2 and 3 show, for  $k < 0.74$ , the MCUSUM signals faster than the MEWMA for small values of the out-of-control mean but signals much slower as the magnitude of the out-of-control mean increases. Conversely, as  $k$  increases from 0.74, the MCUSUM begins to signal slower than the MEWMA for small values of the out-of-control mean but signals much faster as the magnitude of the out-of-control mean increases.

Thus, for the purposes of our simulation comparisons in Chapter V, we set  $\lambda = 0.2$  in the MEWMA and  $k = 0.74$  in the MCUSUM.

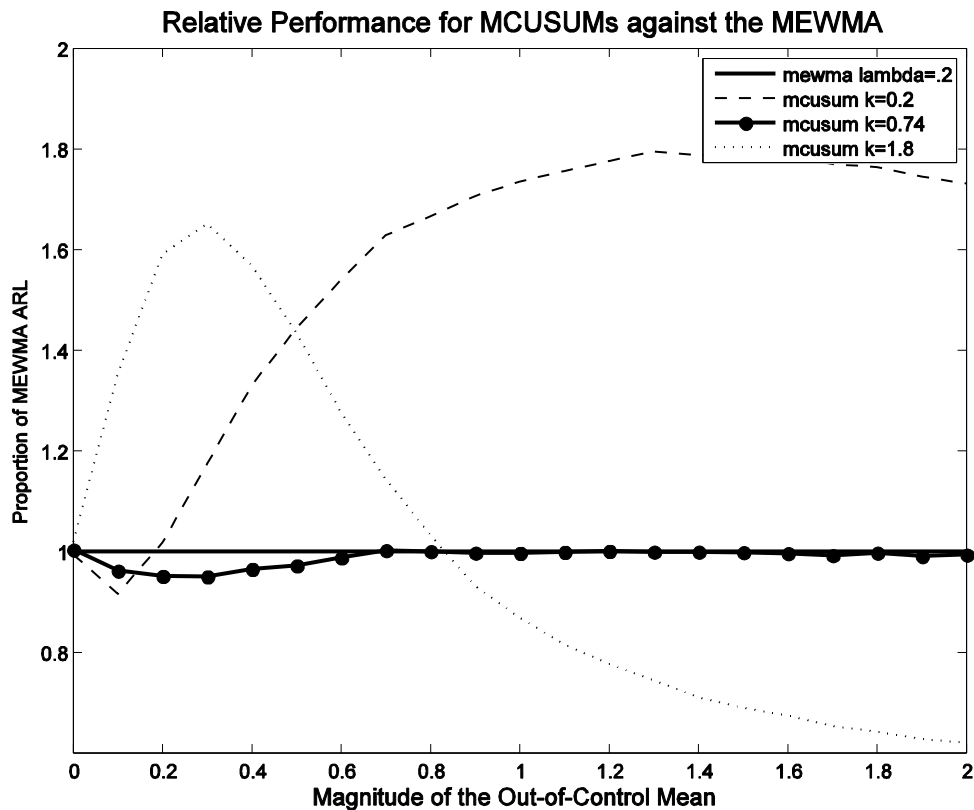


Figure 3. A second plot of the relative ARL performance for multiple MCUSUM instances against the MEWMA with  $\lambda = 0.2$  illustrating evidence that  $k = 0.74$  makes the MCUSUM perform the closest to the MEWMA. This plot illustrates the behavior of the MCUSUM for more extreme  $k$  values.

## B. SETTING THRESHOLDS

Application of the MEWMA and MCUSUM requires setting specific threshold values or control limits,  $h$ . The thresholds are chosen to produce a certain “in-control” ARL or ATFS, denoted by  $ARL_0$  or  $ATFS_0$ , similar to how “a researcher chooses a critical value corresponding to a certain  $\alpha$  level in a hypothesis test” (Joner 2006). A syndromic surveillance process is considered “in-control” in SPC terms when no outbreak is present. The concept of being “in-control” is really a misnomer in the syndromic surveillance setting since no control is actually exhibited over the background disease incidence (Fricker and Rolka 2006), but the terminology is used to draw a parallel to the SPC literature.

For a given scenario, the thresholds are set so that the two methods perform equally well in the absence of an outbreak – that is, they are set to have equal  $ARL_0$ s or  $ATFS_0$ s. This guarantees a fair comparison of the two methods when evaluating their performance at detecting various types of outbreaks. Once the thresholds are determined that result in equal  $ARL_0$ s or  $ATFS_0$ s for a given scenario, the methods’ ability to detect various outbreaks are compared in terms of ATFS given a true signal and percent miss.

In this work, the thresholds were determined empirically as follows. For a particular method and data scenario, choose an initial guess for  $h$  and run the method’s algorithm  $m$  times. Using the output  $ARL_0/ATFS_0$  estimate from this initial trial, iteratively adjust the threshold  $h$ , re-running the method until a desired  $ARL_0/ATFS_0$  is achieved. Larger values of  $m$  will return more precise estimates of the  $ARL_0/ATFS_0$ . For the simulations in this study, we found control limits that set the in-control ARL and ATFS values are within one day of a 100 day target, using  $m$  large enough such that the standard error of these estimates were less than one day.

## C. PERFORMANCE METRICS

The syndromic surveillance literature measures the performance of detection measures in terms of sensitivity, specificity, and timeliness (Shmueli 2006). In the

context of syndromic surveillance, holding all else constant, timely detection of a disease outbreak is the most important dimension. Specifically, the main goal when using these methods is to detect outbreaks as quickly as possible, in order to optimize the response time for the public health officials.

A standard measure of performance in the SPC literature is the ARL. While the ARL metric was used in the initial comparisons in order to determine  $\lambda$  and  $k$ , in the context of syndromic surveillance, the problem with using the ARL as a measure of performance is the transient nature of disease outbreaks. That is, in the industrial SPC world, a common and reasonable assumption is that once an “out-of-control” condition occurs, it persists until a detection method signals. Hence, using the “out-of-control” ARL is an effective metric for comparing the relative performance of methods to detect various out-of-control conditions. However, in syndromic surveillance, disease outbreaks start, peak, and then ultimately subside and go away. Thus, it is important to also assess performance in terms of a procedure’s ability to detect an outbreak *within the period of the outbreak*.

Because of this, two metrics are used to assess the methods’ performance on syndromic surveillance data. The first is the percentage of the time that a method does not detect during an outbreak period, which is referred to herein as “percent miss.” This measure provides insight regarding the utility of the signals. If the detections occur after the outbreak has ended, then the method is not providing a useful signal. The second metric is the conditional average time to first signal (ATFS) given that the method did detect during the outbreak period (also referred to as ATFS given a true signal). This measure illustrates how fast the method is signaling when it detects an outbreak. It is similar to the “average time to signal (ATS)” in the SPC literature, but because of the autocorrelation that is often present in syndromic surveillance data (which can and often does result in repeated alarms), we will only measure the time until first signals.



THIS PAGE INTENTIONALLY LEFT BLANK

## IV. MODELING PUBLIC HEALTH DATA

### A. BACKGROUND DISEASE INCIDENCE

The background disease incidence behavior in real public health data exhibits some natural trends. In the context of syndromic surveillance, we define this background behavior to be the “in-control” state of a system. In attempting to describe this non-stationary nature of real public health data, there are several candidate effects to consider. These include a long-term increase effect, a seasonal cycle effect, day-of-the-week effects, and even holiday effects (Shmueli 2006).

Our model of this background behavior simulates daily counts through a simplified additive model. In this model, the background disease incidence is characterized by a seasonal cycle around a baseline mean. This cycle peaks during the winter season, between the months of December and February. This winter peak is related to the seasonal recurrence of influenza, which we consider background behavior in the context of bioterrorism (Shmueli 2006). In the model, this seasonal cycle is a sinusoid with a one year period. Daily observations are based on a systematic component determined by the day in the cycle and the overall mean (baseline) and a stochastic component representing random noise in the data. Because public health data are in the form of discrete counts, the generated observations are forced to be nonnegative integers. Specifically, an observation  $X_{i,j}$  for day  $i$  in dimension  $j$  is modeled by

$$X_{i,j} = \max\left(0, \lceil \beta + \alpha_i + Y_{i,j}(\beta) \rceil\right) \quad (4.1)$$

where:

- $\beta$  is the baseline value for the disease incidence (grand mean)
- $\alpha_i$  is the seasonal deviation from the baseline mean, calculated as  $\alpha_i = A \cdot \sin(2\pi i / 365)$ .  $A$  is the amplitude parameter (maximum systematic deviation from  $\beta$ ) with  $i = 1$  corresponding to October 1. This start date allows the peak of the cycle to occur during the winter.
- $Y_{i,j}(\beta)$  is the random noise around the systematic component with

$$Y_{i,j}(\beta) \sim N(0, \sigma^2)$$

- $\lceil x \rceil$  is the ceiling function, which rounds  $x$  up to the next largest integer. This is needed to incorporate the discrete nature of the counts in public health data.

Some additional assumptions from the above description are:

- *Constant year length.* Years are always modeled as 365 days long. Extending this to account for leap years is an unnecessary complication that will not affect the results or conclusions.
- *No linear trends.* Growing or shrinking populations, or changes in health conditions, could result in linear (or other) trends in the disease incidence. A trend term is included in Equation (4.1) since, if the procedures can appropriately adjust for the seasonal component, it can also adjust for a linear trend.
- *No holiday or other such effects.* Holiday and day-of-the-week trends may be present in real syndromic surveillance data. However, while these effects are not included in Equation (4.1) and not accounted for in this thesis, it would be relatively straightforward to extend these methods to account for these effects. Furthermore, Dunfee and Hegler (2007) concluded that such effects had minimal to no consequence on the statistical methods and subsequently viewed as an unnecessary complication of the analysis.

For a given day  $i$ , the observations for each data stream  $j$  are generated independently of the others. The correlation still exists in the data because all of the data streams have a common systematic component. Throughout the simulations in this study, we always generate four streams of data. Figure 4 illustrates a year of generated background disease incidence data using this model.

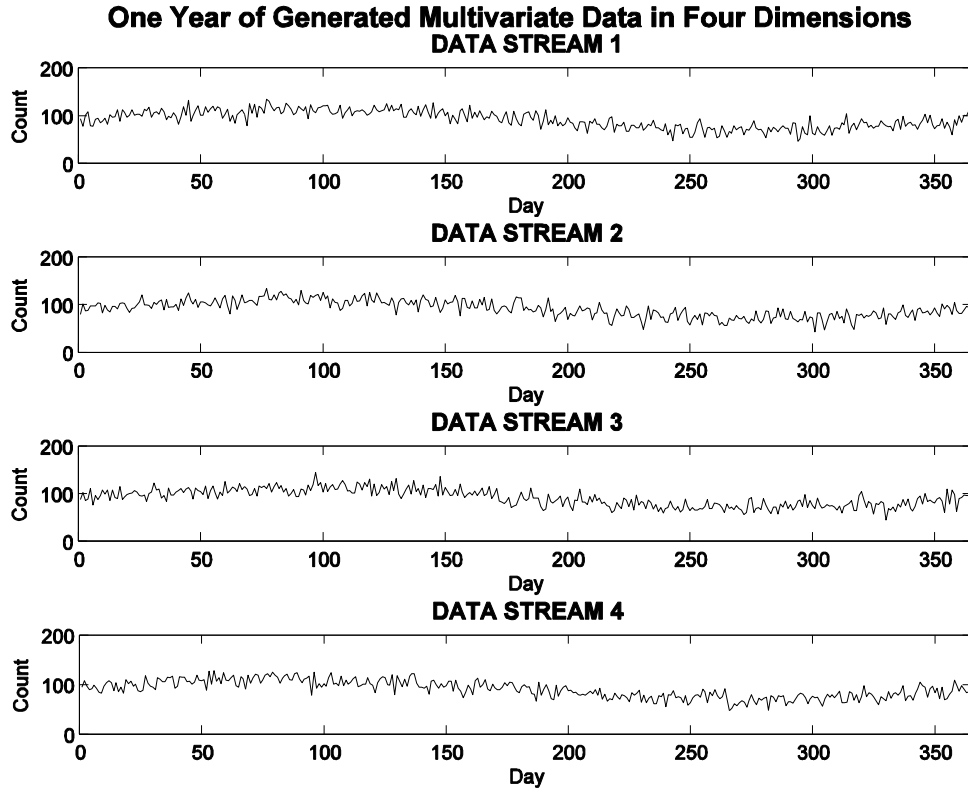


Figure 4. An example of generated background disease incidence data with parameters:  $\beta = 90$ ,  $A = 20$ ,  $\sigma = 10$

## B. MODELING DISEASE OUTBREAK BEHAVIOR

In order to assess the relative performance of the methods in the context of syndromic surveillance, we define the out-of-control condition by the disease outbreak behavior. We consider two candidates for outbreak behavior, namely, constant (e.g., Reis et al. 2003; Brillman et al. 2005) and linear (e.g., Goldenberg et al. 2002; Stoto et al. 2006) increases to the incidence counts.

Our outbreak model is parameterized by the starting day, the peak, and the duration of the outbreak. Instead of modeling the out-of-control condition as a constant increase in the data (as in the initial simulations), our disease outbreaks are modeled by a linear increase in the data until the peak occurs, followed by a linear decrease throughout the rest of the outbreak period. These outbreaks are assumed to be symmetric around the center of the outbreak period.

A systematic disease outbreak component,  $\omega_i$ , is included as another term in the additive model from Equation (4.1) so that the observation  $X_{i,j}$  for day  $i$  in dimension  $j$  is

$$X_{i,j} = \max\left(0, \left\lceil \beta + \alpha_i + \omega_i + Y_{i,j}(\beta) \right\rceil\right) \quad (4.2)$$

where

- $\omega_i = \begin{cases} p - \frac{p}{1 + \left(\frac{d-1}{2}\right)} \left| i - \left( s + \frac{d-1}{2} \right) \right|, & i \text{ is in the outbreak period} \\ 0, & i \text{ is outside the outbreak period} \end{cases}$
- $p$  is the peak magnitude of the outbreak component.
- $d$  is the duration of the outbreak in days. We will only consider outbreak durations of an odd number of days to eliminate the case of two consecutive peak days.
- $s$  is the starting day of the outbreak.

During the outbreak period,  $\omega_i$  represents the systematic departure from the background disease incidence behavior. Figure 5 illustrates the change in the  $\omega_i$  component for a particular instance of a disease outbreak.

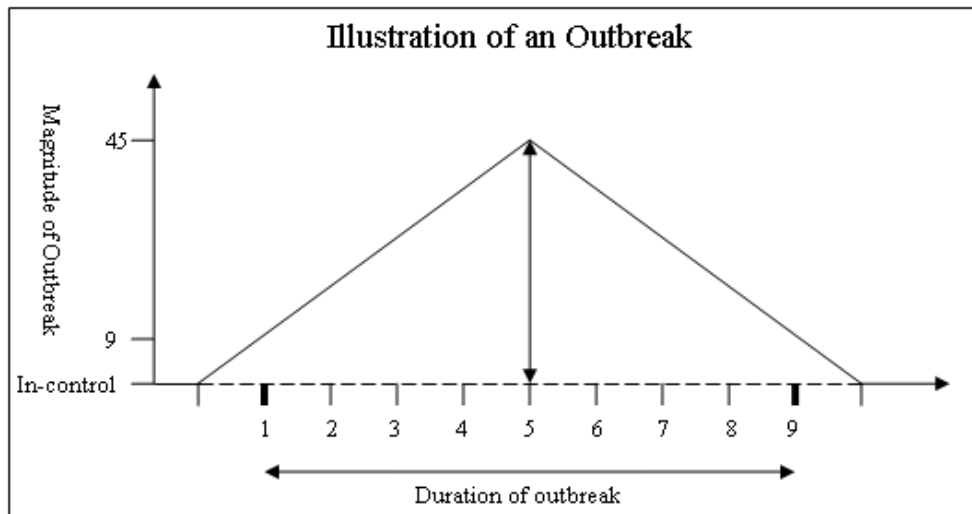


Figure 5. An example outbreak with duration = 9 days, peak = 45, startday = 0

In the simulations, we will define the peak magnitude  $p$  in terms of a percentage  $\pi$  of the baseline, such that  $p = \pi \cdot \beta$ .

THIS PAGE INTENTIONALLY LEFT BLANK

## V. THE SIMULATIONS

The main simulations in this study compare the relative performance of the modified MCUSUM and MEWMA methods over various background disease incidence behaviors and various outbreak magnitudes, as summarized in Table 1.

	<i>None</i>	<i>Small</i>	<i>Large</i>
<i>A</i>	0	20	80
$\sigma$	n/a	10	30
$\pi$	10%	25%	50%

Table 1. Parameter Values for  $\beta = 90$

All simulations are conducted in MATLAB 7.3.0 (R2006b).

### A. CASES

With a single background disease incidence baseline set to  $\beta = 90$ , we developed eighteen simulation cases based on all combinations of the input parameters from Table 1. These cases are presented in Table 2.



Case	Baseline $\beta$	Amplitude $A$	Data Sigma $\sigma$	Outbreak Peak $p$	Sliding Baseline $n$	Estimate of Residual SD $\hat{\sigma}_R$	MEWMA ContLim $h_{\text{MEWMA}}$	MCUSUM ContLim $h_{\text{MCUSUM}}$
1	90	80	30	45	30	30.11	3.28	4.64
2	90	80	30	22.5	30	30.11	3.28	4.64
3	90	80	30	9	30	30.11	3.28	4.64
4	90	80	10	45	30	10.62	3.31	4.78
5	90	80	10	22.5	30	10.62	3.31	4.78
6	90	80	10	9	30	10.62	3.31	4.78
7	90	20	30	45	40	31.46	3.26	4.6
8	90	20	30	22.5	40	31.46	3.26	4.6
9	90	20	30	9	40	31.46	3.26	4.6
10	90	20	10	45	35	10.59	3.26	4.6
11	90	20	10	22.5	35	10.59	3.26	4.6
12	90	20	10	9	35	10.59	3.26	4.6
13	90	0	30	45	45	31.29	3.27	4.62
14	90	0	30	22.5	45	31.29	3.27	4.62
15	90	0	30	9	45	31.29	3.27	4.62
16	90	0	10	45	35	10.58	3.25	4.57
17	90	0	10	22.5	35	10.58	3.25	4.57
18	90	0	10	9	35	10.58	3.25	4.57

Table 2. Simulation Parameters

The scenario parameters  $n$ ,  $\hat{\sigma}_R$ ,  $h_{\text{MEWMA}}$ , and  $h_{\text{MCUSUM}}$  are functions of the defined background behavior  $(\beta, A, \sigma)$ . We determined the values of these parameters empirically, as follows:

- $n$  - from the univariate work of Dunfee and Hegler (2007),
- $\hat{\sigma}_R$  - using adaptive regression on 800,000 days of background disease incidence data for each background scenario, as suggested in Chapter IV, Section C,
- $h_{\text{MEWMA}}$  and  $h_{\text{MCUSUM}}$  - using the method calibration technique described in Chapter II, Section D.

In addition to the varying outbreak peaks ( $p$ ), each case is evaluated over multiple outbreak durations. Specifically, we ran each case over odd day outbreak durations from 3 – 15 days and generated plots to examine the relative performance of the methods as the outbreak duration changed.

## B. SIMULATION SEQUENCE OF EVENTS

In order to run the methods on simulated data we developed the following startup procedures:

- Pick a uniformly distributed random start day in the seasonal cycle (1 to 365). Each time the simulation resets, a new start day is assigned. This allows for the methods to run equally on all regions of the seasonal cycle.
- Generate startup data for the number of days given by the “optimal” sliding baseline  $n$ . This data is required before the adaptive regression technique can begin.
- Begin applying the method as described in Chapter IV, Section C. The outbreak begins 100 days after the method begins running. This startup period effectively loads background disease behavior into the method’s memory.

In addition, the following rules applied in our simulations:

- If an alarm signals before the outbreak begins (i.e., during the startup period), reset all method statistics. This alarm is not counted, because we are only concerned with comparing the methods’ performance during outbreak periods. In other words, all statistics are collected conditional on the event that an outbreak has begun after an initial in-control startup period.
- Begin counting days for output statistics on the first day of the outbreak.
- Run the main sequence until the first signal occurs, and record the run length.
- Each scenario is run for 2,500 replications.

Using the recorded run lengths, we derived estimates of the ATFS, ATFS given a true signal, and Percent Miss statistics along with their respective standard errors.

## C. PERFORMANCE RESULTS

The simulations show that the MEWMA and MCUSUM perform almost identically under all of the 18 cases we evaluated. Figure 6 illustrates the results from simulation case 1.

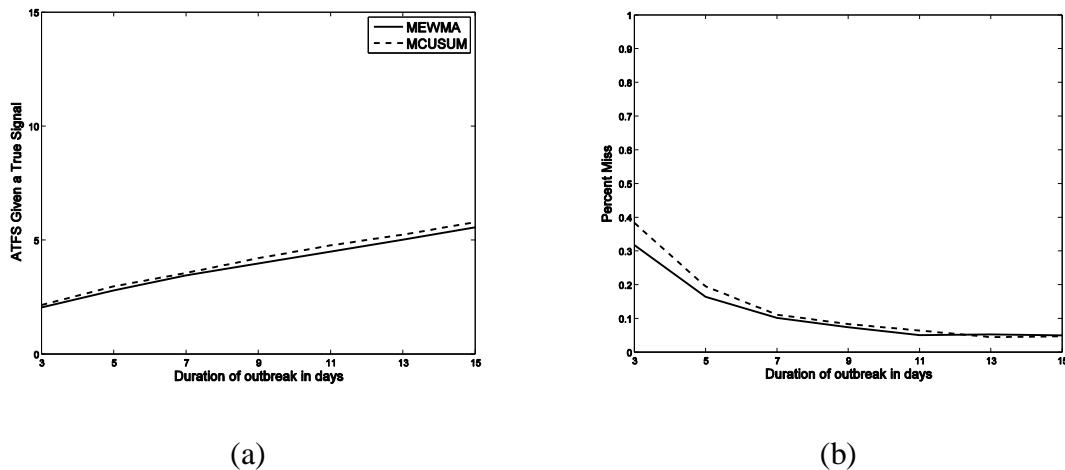
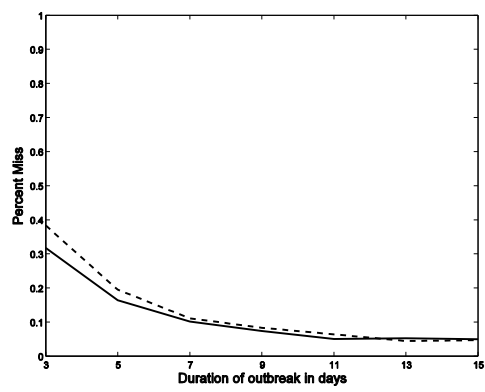
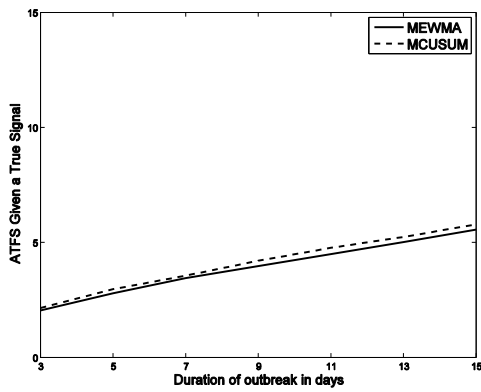


Figure 6. Plots for Case 1 simulation: (a) ATFS Given True Signal, (b) Percent Miss

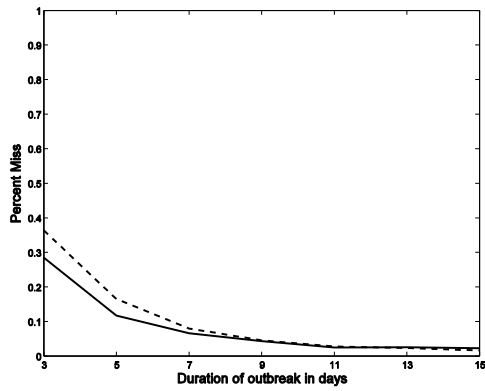
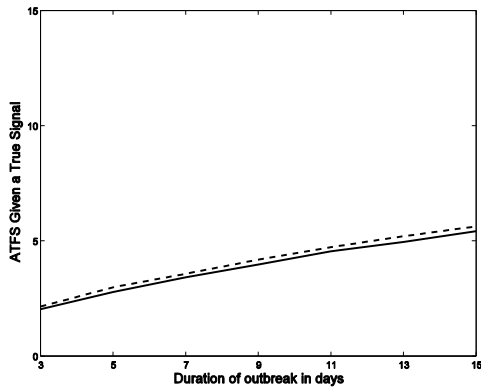
Using the hypothesis test described in Appendix C, we conclude that there is a statistical difference in the ATFS Given True Signal statistics. On average, the MEWMA signals faster during the outbreak period, however this statistical significance is not practically significant. In the case 1 simulation, the maximum difference in the ATFS Given True Signal statistic is 0.2693 days. This difference is so small that the methods can be considered equivalent in practice. This small statistical difference is present in all of the cases except cases 3, 9, and 15 (where  $\sigma = 30$  and  $p = 9$ ). In these cases, the hypothesis test shows no difference at all between the two methods.

In addition, it is important to note that the observed differences could simply be the result of some imprecision in how we set  $\lambda$  and  $k$ . Specifically, it is quite possible that a slight additional adjustment in  $k$  would result in statistical insignificance between the MEWMA and MCUSUM performance under this measure.

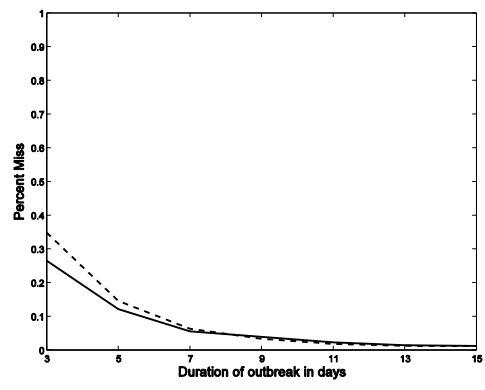
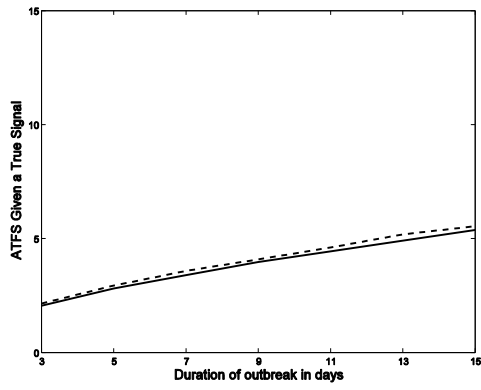
Both methods performed even more similarly under the Percent Miss metric, with statistically significant differences occurring only under short outbreak durations. Again, these are very small differences, but in all cases where a difference exists, it is the MEWMA that has a smaller Percent Miss estimate. For a comprehensive listing of results plots and statistical difference calculations, see Appendix B and Appendix C, respectively.



Case 1 –  $A = 80$



Case 7 –  $A = 20$



Case 13 –  $A = 0$

Figure 7. Plots for cases 1, 7, and 13 (solid lines - MEWMA, dashed lines - MCUSUM). All three cases have  $\beta = 90, \sigma = 30,$  and  $p = 45$ , illustrating the invariance of method performance with respect to changes in the amplitude component.

Figure 7 illustrates that the adaptive regression procedure in the modified methods effectively removes the systematic component of the synthetic syndromic surveillance data. For cases 1, 7, and 13 all of the parameters are fixed ( $\beta = 90$ ,  $\sigma = 30$ ,  $p = 45$ ) except for the amplitude component of the seasonal sinusoid. The seasonal sinusoid is not included in case 13 ( $A = 0$ ), so the case 13 plots demonstrate performance results on stationary data similar to traditional SPC data. Inspection of the resulting plots for cases 1 and 7 show a general invariance to changes in the amplitude component. Although there is a slight decrease in Percent Miss as the amplitude increases, we conclude that the adaptive regression technique is quite successful at accounting for the systematic, non-stationary nature of syndromic surveillance data.

## VI. DISCUSSION AND CONCLUSIONS

### A. DISCUSSION

After performing the comparative analysis, we determined that the ATFS statistic is not a useful measure in the context of syndromic surveillance. We have shown that the MEWMA and the MCUSUM perform similarly under our controlled scenarios using the ATFS measure. This result does not give an observer any information about the timeliness of detecting the outbreak. Many of the run lengths collected for the ATFS statistic were longer than the outbreak period. Even though the method does signal later in time, the signal is too late to be considered helpful to any efforts by a public health response team. For longer run lengths, we also cannot tell if the alarm was caused by a delayed response to the outbreak itself, or if in fact, the alarm is a false signal caused by noise in the system.

Instead, the other two measures, ATFS Given True Signal and Percent Miss, both provide constructive information to a medical response worker. The ATFS Given True Signal demonstrates how fast a method signals, when it actually does catch the outbreak present. The Percent Miss metric illustrates how good the method is able to detect outbreaks. These measures are applicable in a controlled setting where the analyst knows that there is an outbreak present. In the real world, the presence of an outbreak is not known, at least not until after the outbreak has occurred, so these measures are not applicable in practice.

Since there seems to be no performance advantage to using the MEWMA or the MCUSUM, this result leads us to prefer the MEWMA method for procedural reasons. Past literature has provided sufficient analysis and intuitive recommendation for the choice of the  $\lambda$  parameter; however, choosing an appropriate  $k$  is much more difficult to understand. Currently, there is no research to guide one on the trade-offs in various  $k$  values, thereby preventing well-stepped grounds for any choice in  $k$ . While the methods perform with no statistical difference, we thus prefer the MEWMA over the MCUSUM simply for the procedural simplicity of the former.

## **B. FUTURE RESEARCH**

The work presented in this thesis conducts uses idealized, simulated data streams which do not account for holiday or day-of-the-week effects. While such an inclusion would provide a more realistic data series, Dunfee and Hegler (2007) concluded that these effects were of minimal to no consequence. However, future work might look to confirm this finding for multivariate methods applied to multivariate data. In addition, the systematic effects in this work were characterized by a smooth sinusoidal function. Comparison of the methods' performance when the systematic effects are more random and/or less smooth would be useful.

The disease outbreak behavior modeled in this thesis assumes that disease outbreaks are temporary in nature. Future research might consider the effects of long-term outbreaks on performance. The form of the outbreaks was a simple linear increase and linear decrease. Future research should evaluate the performance of the methods for other forms of outbreaks. In addition, this work incorporated an outbreak component uniformly across all dimensions of the data. Further study may evaluate the sensitivity of these methods when an outbreak is only present in one or some dimensions.

The work presented in this thesis analyzed only two specific multivariate detection methods. Further exploration into other methods may identify even better techniques for the syndromic surveillance problem. Fricker (2007) found the MCUSUM outperformed some other methods (such as Hotelling's  $T^2$  and Healy's MCUSUM), but additional comparisons between the methods listed herein against others are always warranted. (For a description of other multivariate methods see Shmueli and Fienberg, 2006.)

Little work has been done to compare the performance of multivariate methods against multiple univariate techniques in the syndromic surveillance setting. While some introductory research has explored this topic (see, for example, Fricker 2007), further research is warranted.

The methods presented here were tested on simulated data that exhibit the major characteristics of syndromic surveillance data but do not completely mimic realistic data. The multivariate CUSUM and EWMA methods were applied to the same set of data in an attempt to rate the relative performance against each other in a more controlled fashion than could be achieved from analysis on real data. Future research in this topic may include assessing the procedures' performance on actual data in order to confirm that these results do generalize to the real public health problem.



THIS PAGE INTENTIONALLY LEFT BLANK

## APPENDIX A.

This appendix includes the main MATLAB code developed in this thesis.

### A. MEWMA (INITIAL)

```
% The mewma function generates data based on the traditional SPC
% assumptions and input parameters and runs the MEWMA method on these
% data recording the length of each individual run (number of days
% until the method signals). This function is used in the initial
% simulations.
% Input - a structure containing input parameters including the
% elements:
%     contLim    -- the control limit
%     lambda     -- the constant lambda parameter
%     oocMean    -- the magnitude of the out-of-control mean
%     icMean     -- the value of the in-control mean
%     icSigma    -- the in-control covariance matrix
%     numStreams -- the number of data streams (dimensions) to
%                   generate
%     numLoops   -- the number of times to run the MCUSUM
% Output - a structure containing the run lengths from the simulation
%     runLengths -- the vector of generated run lengths

function [mewmaResults]= mewma(inputParameters)

% Initialize the Parameter Values-----
contLim = inputParameters.mewmaContLim;
lambda =inputParameters.lambda;
oocMean = inputParameters.oocMean;
icMean = inputParameters.icMean;
icSigma = inputParameters.icSigma;
numStreams = inputParameters.numStreams;
numLoops = inputParameters.numLoops;

% Calculate the Covariance Matrix and its Inverse for 'z infinity'
% for the MEWMA method
sigmaZInfinity = (lambda / (2-lambda)) * icSigma;
sigmaZInfinityInverse = inv(sigmaZInfinity);

% Run the MEWMA method-----
indivRuns = zeros(numLoops,1);
for j=1:numLoops
    z = zeros(numStreams,1);
    mew = 0.0;
    runLength = 0;
    while (mew <= contLim)
        newObs = mvnrnd(oocMean,icSigma)';
        temp = (lambda * (newObs - icMean)) + ((1-lambda)*z);
        for q=1:numStreams
            if temp(q,1) < 0
```

```

        z(q,1) = 0;
    else
        z(q,1) = temp(q,1);
    end
end
end
mew = sqrt(z'*sigmaZInfinityInverse*z);
runLength = runLength + 1;
end
indivRuns(j,1) = runLength;
end

% Save the vector of run lengths in the output structure-----
mewmaResults.runLengths = indivRuns;

```

## B. MCUSUM (INITIAL)

```

% The mcusum function generates data based on the traditional SPC
% assumptions and input parameters and runs the MCUSUM method on these
% data recording the length of each individual run (number of days
% until the method signals). This function is used in the initial
% simulations.
% Input - a structure containing input parameters including the
% elements:
%     contLim    -- the control limit
%     k          -- the constant k parameter
%     oocMean    -- the magnitude of the out-of-control mean
%     icMean     -- the value of the in-control mean
%     numStreams -- the number of data streams (dimensions) to
%                 generate
%     numLoops   -- the number of times to run the MCUSUM
% Output - a structure containing the run lengths from the simulation
%     runLengths -- the vector of generated run lengths

function [mcusumResults] = mcusum(inputParameters)

% Initialize the Parameter Values-----
contLim = inputParameters.mcusumContLim;
k = inputParameters.k;
oocMean = inputParameters.oocMean;
icMean = inputParameters.icMean;
icSigma = inputParameters.icSigma;
numStreams = inputParameters.numStreams;
numLoops = inputParameters.numLoops;

% Define the in-control mean vector and covariance matrix-----
icSigmaInv = inv(icSigma);
kVector = ones(numStreams,1)*k;
MCUSUMk = sqrt(kVector'*icSigmaInv*kVector);

% Run the MCUSUM method-----
indivRuns = zeros(numLoops,1);
for i=1:numLoops

```

```

s = zeros(numStreams,1);
y = 0.0;
runLength = 0;
while y < contLim
    temp = s;
    newObs = mvnrnd(oocMean,icSigma)';
    c = sqrt((temp+newObs-icMean)'*icSigmaInv*(temp+newObs-
        icMean));
    if c > MCUSUMk
        temp2 = (temp + newObs - icMean)*(1-(MCUSUMk/c));
    else
        temp2 = zeros(numStreams,1);
    end
    for stream = 1:numStreams
        if temp2(stream,1) < 0
            s(stream,1) = 0;
        else
            s(stream,1) = temp2(stream,1);
        end
    end
    y = sqrt(s'*icSigmaInv*s);
    runLength = runLength + 1;
end
indivRuns(i,1) = runLength;
end

% Save the vector of run lengths in the output structure-----
mcusumResults.runLengths = indivRuns;

```

### C. DATA GENERATION AND ADAPTIVE REGRESSION TOOL

```

% The dataGeneratorRegress function is a tool that generates
% multivariate syndromic surveillance data and runs the adaptive
% regression technique on this data. It displays the sample
% correlation matrices of the generated data and the adaptive
% regression residuals. It also constructs a plot including the
% generated data streams and the adaptive regression residuals.
% Input - numStreams -- the number of streams/dimensions to model
%         numDays    -- the number of days of data to simulate
%         baseline   -- the baseline of the data, beta
%         amplitude  -- the amplitude of the seasonal cycle, A
%         sigma      -- the standard deviation of the generated noise
%                   in the data
%         lookBack   -- the sliding baseline used in the adaptive
%                   regression technique
% Output - dataCorr  -- the sample correlation matrix of the data
%         residCorr -- the sample correlation matrix of the
%                   residuals
%         dataAndResidualsPlot
%                   -- a MATLAB figure containing plots of the
%                   generated data streams and the residual
%                   "streams"

% INPUT PARAMETERS-----

```

```

% The input parameters shown here are only an example.
numStreams = 4;
numDays = 365;
% baseline is either 90 (normal noise) or 0 (lognormal noise)
baseline = 90;
amplitude = 20;
sigma = 10;
lookBack = 35;

% INTIIALIZE ARRAYS-----
dataStreams = zeros(numDays,numStreams);
predictions = zeros(numDays-lookBack,numStreams);
residuals = zeros(numDays-lookBack,numStreams);
means = zeros(numDays,numStreams);
noise = zeros(numDays,numStreams);

% designMatrix is used in the regressions
designMatrix = ones(lookBack,2);
designMatrix(:,2) = (1:lookBack)';

% DATA GENERATION-----
% Generate the underlying sinusoid
for day = 1:numDays
    means(day,:)...
        = ones(1,numStreams)*(amplitude*(sin(2*pi*day/365))+ baseline);
end

% Generate the noise for all streams and all days
if(baseline == 90)
    noise = random('norm',0.0,sigma,numDays,numStreams);
elseif(baseline == 0)
    noise = random('logn',1.0,sigma,numDays,numStreams);
end

% Store the day's observations, force minimum count to zero
dataStreams = max(dataStreams,ceil(means+noise));

% Report the correlation structure of the generated data
dataCorr = corr(dataStreams);
disp('The sample correlation matrix is:');
disp(dataCorr);

% REGRESSION SECTION-----
for stream = 1:numStreams
    for day = (lookBack+1):numDays

        % Grab the counts from the previous "lookBack" # of days
        regressData = dataStreams((day-lookBack):(day-1),stream);

        % Daily regression calculation, b holds the model coefficients
        b = regress(regressData, designMatrix);
    end
end

```

```

    % Predict tomorrow's count
    tomorrowData = [1 lookBack+1];
    predictions = tomorrowData*b;

    % Calculate residual values
    residuals(day-lookBack,stream)...
        = dataStreams(day,stream) - predictions;
end
end

residCorr = corr(residuals);
disp('The correlation matrix of the residuals is:');
disp(residCorr);

% GENERATE PLOTS-----
% The remaining code plots the generated data streams and the residual
% "streams" from the regression in a single figure
dataRegress = figure('Name','Data and Residuals','NumberTitle','Off');
for index = 1:numStreams
    subplot(numStreams,1,index);
    plot(1:numDays,dataStreams(:,index));
    xlabel('Day Number');
    ylabel('Count');
    subplot(numStreams,2,2*index-1);
    plot(1:numDays,dataStreams(:,index));
    xlabel('Day number');
    ylabel('Count');
    subplot(numStreams,2,2*index);
    plot((lookBack+1:numDays),residuals(:,index));
    xlabel('Day number');
    ylabel('Residuals');
end
saveas(dataRegress,'dataAndResidualsPlot.fig');

```

#### D. MODIFIED MEWMA FOR SYNDROMIC SURVEILLANCE DATA

```

% The mewmaData function generates syndromic surveillance data based on
% the input parameters and runs the modified MEWMA method on
% these data (using adaptive regression) recording multiple useful
% statistics. This function is used in the main simulation analysis.
% Input - a structure containing input parameters including the
% elements:
%     contLim    -- the control limit
%     lambda     -- the constant lambda parameter
%     icMean     -- the value of the in-control mean
%     icSigma    -- the in-control covariance matrix
%     numStreams -- the number of data streams (dimensions) to
%                   generate
%     numStreams -- the number of streams/dimensions to model
%     numLoops   -- the number of times to run the MCUSUM
%     baseline   -- the baseline of the data, beta
%     amplitude  -- the amplitude of the seasonal cycle, A

```

```

%         dataSigma  -- the standard deviation of the noise in the
%                   data
%         lookBack   -- the "optimal" sliding baseline, n
%         outbreak   -- boolean, true if an outbreak is modeled
%         outbreakLength -- the duration of the outbreak in days
%         outbreakMax    -- the peak magnitude of the outbreak
%                   component
% Output - a structure containing the run lengths from the simulation
%         runLengths -- the vector of generated run lengths
%         ATFS       -- estimate of the ATFS
%         ATFSsse    -- standard error of the ATFS estimate
%         detections -- logical vector, 1 if detected during the
%                   outbreak, 0 otherwise
%         percentDetections -- percentage of runs detected during the
%                   outbreak
%         percentMiss    -- percentage of runs not detected during
%                   the outbreak
%         percentMissSE  -- standard error of percentMiss estimate
%         ATFSGivenDetect -- ATFS given that the signal was detected
%                   during the outbreak
%         ATFSSeGivenDetect -- standard error of the ATFSGivenDetect
%                   estimate

```

```
function [mewmaResults]= mewmaData(inputParameters)
```

```

% INITIALIZE INPUT PARAMETERS-----
contLim = inputParameters.mewmaContLim;
lambda =inputParameters.lambda;
icMean = inputParameters.icMean;
icSigma = inputParameters.icSigma;
numStreams = inputParameters.numStreams;
numLoops = inputParameters.numLoops;
baseline = inputParameters.baseline;
amplitude = inputParameters.amplitude;
dataSigma = inputParameters.dataSigma;
lookBack = inputParameters.lookBack;
outbreak = inputParameters.outbreak;
if(outbreak)
    outbreakLength = inputParameters.outbreakLength;
    outbreakMax = inputParameters.outbreakMax;
    outbreakPeakDay = (outbreakLength + 1) / 2;
    outbreakSlope = outbreakMax/outbreakPeakDay;
end

% Define the designMatrix used in the regressions
designMatrix = ones(lookBack,2);
designMatrix(:,2) = (1:lookBack)';
residual = zeros(numStreams,1);

% Calculate the Covariance Matrix and its Inverse for 'z infinity'
% for the MEWMA method
sigmaZInfinity = (lambda / (2-lambda)) * icSigma;
sigmaZInfinityInverse = inv(sigmaZInfinity);

```

```

% GENERATE DATA AND RUN THE MODIFIED MEWMA-----
runLengths = zeros(numLoops,1);
detections = zeros(numLoops,1);
for j=1:numLoops
    z = zeros(numStreams,1);
    mew = 0.0;
    runLength = 0;
    randomStartDay = ceil(365*rand);
    currentDay = randomStartDay;
    outbreakStartDay = randomStartDay + 100 + lookBack;
    dataStreams = zeros(500,numStreams);
    % Generate Startup Data
    for i=1:lookBack
        dataMean = ones(1,numStreams)*(amplitude* ...
            (sin(2*pi*currentDay/365))+ baseline);
        if(baseline == 90)
            dataNoise = random('norm',0.0,dataSigma,1,numStreams);
        elseif(baseline == 0)
            dataNoise = random('logn',1.0,dataSigma,1,numStreams);
        end
        dataStreams(currentDay,:) = ...
            max(zeros(1,numStreams),ceil(dataMean + dataNoise));
        currentDay = currentDay + 1;
    end
    % Run the method
    while (mew <= contLim)
        % Calculate the level of the outbreak for the current day
        if(outbreak)
            if(currentDay < outbreakStartDay || currentDay >= ...
                outbreakStartDay + outbreakLength)
                outbreakLevel = 0.0;
            elseif(currentDay < outbreakStartDay + outbreakPeakDay)
                outbreakLevel = outbreakSlope*(currentDay + 1 - ...
                    outbreakStartDay);
            else
                outbreakLevel = outbreakMax - outbreakSlope* ...
                    (currentDay - (outbreakPeakDay+outbreakStartDay - 1));
            end
        end
        % Generate new data point
        if(outbreak)
            dataMean = ones(1,numStreams)* ...
                (amplitude*(sin(2*pi*currentDay/365)) ...
                    + baseline + outbreakLevel);
        else
            dataMean = ones(1,numStreams)* ...
                (amplitude*(sin(2*pi*currentDay/365))+ baseline);
        end
        if(baseline == 90)
            dataNoise = random('norm',0.0,dataSigma,1,numStreams);
        elseif(baseline == 0)
            dataNoise = random('logn',1.0,dataSigma,1,numStreams);
        end
        dataStreams(currentDay,:) =
            max(zeros(1,numStreams),ceil(dataMean + dataNoise));
    end
end

```



```

% Regress on the data to get residual
for stream = 1:numStreams

    % Grab the counts from the previous "lookBack" # of days
    regressData = ...
        dataStreams((currentDay-lookBack):(currentDay-1),stream);

    % Daily regression calculation, b holds the model
    % coefficients
    b = regress(regressData, designMatrix);

    % Predict tomorrow's count
    tomorrowData = [1 lookBack+1];
    predictions = tomorrowData*b;

    % Calculate residual values
    residual(stream,1) = ...
        dataStreams(currentDay,stream) - predictions;
end
% Run the MEWMA method on the residual
newObs = residual;
temp = (lambda * (newObs - icMean)) + ((1-lambda)*z);
for q=1:numStreams
    if temp(q,1) < 0
        z(q,1) = 0;
    else
        z(q,1) = temp(q,1);
    end
end
mew = sqrt(z'*sigmaZInfinityInverse*z);
% If the method signals in the init period (first 100 days of
% running) the MEW statistic resets.
if((currentDay < randomStartDay + 100 + lookBack) && ...
    mew > contLim)
    mew = 0.0;
end
% Start the runLength counter after the init period
if(currentDay >= randomStartDay + 100 + lookBack)
    runLength = runLength + 1;
end
currentDay = currentDay + 1;
end
runLengths(j,1) = runLength;
if(outbreak && runLengths(j,1) <= outbreakLength)
    detections(j,1) = 1;
end
end
end

% SAVE THE RESULTS-----
mewmaResults.runLengths = runLengths;
mewmaResults.ATFS = mean(runLengths);
mewmaResults.ATFSse = std(runLengths)/sqrt(numLoops);
if(outbreak)
    percentDetections = mean(detections);
end

```

```

percentMiss = 1 - percentDetections;
percentMissSE = sqrt(percentDetections*percentMiss/numLoops);
if(detections == 0)
    ATFSGivenDetect = 0;
    ATFSsGivenDetect = 0;
else
    detectionsLogical = logical(detections);
    runLengthsGivenDetect = runLengths(detectionsLogical);
    ATFSGivenDetect = mean(runLengthsGivenDetect);
    ATFSsGivenDetect = ...
        std(runLengthsGivenDetect)/sqrt(length(runLengthsGivenDetect));
end
mewmaResults.detections = detections;
mewmaResults.percentDetections = percentDetections;
mewmaResults.percentMiss = percentMiss;
mewmaResults.percentMissSE = percentMissSE;
mewmaResults.ATFSGivenDetect = ATFSGivenDetect;
mewmaResults.ATFSsGivenDetect = ATFSsGivenDetect;
end

```

## E. MODIFIED MCUSUM FOR SYNDROMIC SURVEILLANCE DATA

```

% The mcusumData function generates syndromic surveillance data based
% on the input parameters and runs the modified MCUSUM method on
% these data (using adaptive regression) recording multiple useful
% statistics. This function is used in the main simulation analysis.
% Input - a structure containing input parameters including the
% elements:
%     contLim    -- the control limit
%     k          -- the constant k parameter
%     icSigma    -- the in-control covariance matrix
%     numStreams -- the number of data streams (dimensions) to
%                   generate
%     numLoops  -- the number of times to run the MCUSUM
%     baseline  -- the baseline of the data, beta
%     amplitude -- the amplitude of the seasonal cycle, A
%     dataSigma -- the standard deviation of the noise in the
%                   data
%     lookBack  -- the "optimal" sliding baseline, n
%     outbreak  -- boolean, true if an outbreak is modeled
%     outbreakLength -- the duration of the outbreak in days
%     outbreakMax -- the peak magnitude of the outbreak
%                   component
% Output - a structure containing the run lengths from the simulation
%     runLengths -- the vector of generated run lengths
%     ATFS       -- estimate of the ATFS
%     ATFSs      -- standard error of the ATFS estimate
%     detections -- logical vector, 1 if detected during the
%                   outbreak, 0 otherwise
%     percentDetections -- percentage of runs detected during the
%                   outbreak
%     percentMiss  -- percentage of runs not detected during
%                   the outbreak
%     percentMissSE -- standard error of percentMiss estimate

```

```

%           ATFSGivenDetect    -- ATFS given that the signal was detected
%                               during the outbreak
%           ATFSsseGivenDetect -- standard error of the ATFSGivenDetect
%                               estimate

function [mcusumResults] = mcusumData(inputParameters)

% INITIALIZE INPUT PARAMETERS-----
contLim = inputParameters.mcusumContLim;
k = inputParameters.k;
icSigma = inputParameters.icSigma;
numStreams = inputParameters.numStreams;
numLoops = inputParameters.numLoops;
baseline = inputParameters.baseline;
amplitude = inputParameters.amplitude;
dataSigma = inputParameters.dataSigma;
lookBack = inputParameters.lookBack;
outbreak = inputParameters.outbreak;

if(outbreak)
    outbreakLength = inputParameters.outbreakLength;
    outbreakMax = inputParameters.outbreakMax;
    outbreakPeakDay = (outbreakLength + 1) / 2;
    outbreakSlope = outbreakMax/outbreakPeakDay;
end

% designMatrix is used in the regressions
designMatrix = ones(lookBack,2);
designMatrix(:,2) = (1:lookBack)';
residual = zeros(numStreams,1);

% Define the in control mean vector and covariance matrix
icSigmaInv = inv(icSigma);
kVector = ones(numStreams,1)*k;
MCUSUMk = sqrt(kVector'*icSigmaInv*kVector);

% GENERATE DATA AND RUN MODIFIED MCUSUM-----
runLengths = zeros(numLoops,1);
detections = zeros(numLoops,1);
for j=1:numLoops
    s = zeros(numStreams,1);
    y = 0.0;
    runLength = 0;
    randomStartDay = ceil(365*rand);
    currentDay = randomStartDay;
    outbreakStartDay = randomStartDay + 100 + lookBack;
    dataStreams = zeros(500,numStreams);
    % Generate Startup Data
    for i=1:lookBack
        dataMean = ones(1,numStreams)* ...
            (amplitude*(sin(2*pi*currentDay/365))+ baseline);
        if(baseline == 90)
            dataNoise = random('norm',0.0,dataSigma,1,numStreams);
        elseif(baseline == 0)

```

```

        dataNoise = random('logn',1.0,dataSigma,1,numStreams);
    end
    dataStreams(currentDay,:) = max(zeros(1,numStreams), ...
        ceil(dataMean + dataNoise));
    currentDay = currentDay + 1;
end
% Run the method
while y < contLim
    % Calculate the level of the outbreak for the current day
    if(outbreak)
        if(currentDay < outbreakStartDay || currentDay >= ...
            outbreakStartDay + outbreakLength)
            outbreakLevel = 0.0;
        elseif(currentDay < outbreakStartDay + outbreakPeakDay)
            outbreakLevel = outbreakSlope* ...
                (currentDay + 1 - outbreakStartDay);
        else
            outbreakLevel = outbreakMax - outbreakSlope* ...
                (currentDay - (outbreakPeakDay+outbreakStartDay - 1));
        end
    end
    % Generate new data point
    if(outbreak)
        dataMean = ones(1,numStreams)* ...
            (amplitude*(sin(2*pi*currentDay/365))+ ...
            baseline + outbreakLevel);
    else
        dataMean = ones(1,numStreams)* ...
            (amplitude*(sin(2*pi*currentDay/365))+ baseline);
    end
    if(baseline == 90)
        dataNoise = random('norm',0.0,dataSigma,1,numStreams);
    elseif(baseline == 0)
        dataNoise = random('logn',1.0,dataSigma,1,numStreams);
    end
    dataStreams(currentDay,:) = ...
        max(zeros(1,numStreams),ceil(dataMean + dataNoise));
    % Regress on the data to get residual
    for stream = 1:numStreams

        % Grab the counts from the previous "lookBack" # of days
        regressData = ...
            dataStreams((currentDay-lookBack):(currentDay-1),stream);

        % Daily regression calculation, b holds the model
        % coefficients
        b = regress(regressData, designMatrix);

        % Predict tomorrow's count
        tomorrowData = [1 lookBack+1];
        predictions = tomorrowData*b;

        % Calculate residual values
        residual(stream,1) = ...
            dataStreams(currentDay,stream) - predictions;
    end
end

```

```

end
% Run the MCUSUM method on the residual
temp = s;
newObs = residual;
c = sqrt((temp+newObs)'*icSigmaInv*(temp+newObs));
if c > MCUSUMk
    temp2 = (temp + newObs)*(1-(MCUSUMk/c));
else
    temp2 = zeros(numStreams,1);
end
for stream = 1:numStreams
    if temp2(stream,1) < 0
        s(stream,1) = 0;
    else
        s(stream,1) = temp2(stream,1);
    end
end
y = sqrt(s'*icSigmaInv*s);
% If the method signals in the init period (first 100 days of
% running) the MCUSUM statistic resets.
if((currentDay < randomStartDay + 100 + lookBack) && y >
    contLim)
    s = zeros(numStreams,1);
    y = 0.0;
end
% Start the runLength counter after the init period
if(currentDay >= randomStartDay + 100 + lookBack)
    runLength = runLength + 1;
end
currentDay = currentDay + 1;
end
runLengths(j,1) = runLength;
if(outbreak && runLengths(j,1) <= outbreakLength)
    detections(j,1) = 1;
end
end

% SAVE THE RESULTS-----
mcusumResults.runLengths = runLengths;
mcusumResults.ATFS = mean(runLengths);
mcusumResults.ATFSse = std(runLengths)/sqrt(numLoops);
if(outbreak)
    percentDetections = mean(detections);
    percentMiss = 1 - percentDetections;
    percentMissSE = sqrt(percentDetections*percentMiss/numLoops);
    if(detections == 0)
        ATFSGivenDetect = 0;
        ATFSseGivenDetect = 0;
    else
        detectionsLogical = logical(detections);
        runLengthsGivenDetect = runLengths(detectionsLogical);
        ATFSGivenDetect = mean(runLengthsGivenDetect);
        ATFSseGivenDetect = std(runLengthsGivenDetect) ...
            /sqrt(length(runLengthsGivenDetect));
    end
end

```

```

mcusumResults.detections = detections;
mcusumResults.percentDetections = percentDetections;
mcusumResults.percentMiss = percentMiss;
mcusumResults.percentMissSE = percentMissSE;
mcusumResults.ATFSGivenDetect = ATFSGivenDetect;
mcusumResults.ATFSseGivenDetect = ATFSseGivenDetect;
end

```

## F. MAIN SIMULATION ENGINE

```

% The mainSimulation file is the main engine used to run the
% simulations in this comparative study. It needs access to mewmData.m
% and mcusumData.m in order to perform the adaptive regression
% technique and modified MEWMA/MCUSUM methods. Given a specific
% background disease incidence scenario and maximum outbreak magnitude,
% mainSimulation runs both methods for a given number of times over a
% varying outbreak duration (3-15 days) and records the ATFS, ATFS
% Given True Signal, and PercentMiss statistics along with the
% respective plots.
% Input - The inputs are not read into the program from another file.
% They are entered in the appropriate sections below. The
% input variables are:
%     caseNumber      -- the ID number of the simulated case
%     mewmaContLim    -- the MEWMA control limit
%     mcusumContLim   -- the MCUSUM control limit
%     baseline        -- the baseline for the data, beta
%     amplitude       -- the amplitude of the seasonal cycle,
%                     A
%     dataSigma       -- the standard deviation of the noise
%                     in the data
%     outbreakPercent -- the max magnitude of the outbreak as
%                     a percentage of the baseline
%     numLoops        -- the number of times to run the
%                     methods
% The rest of the input parameters are STATIC throughout all
% simulations and do not need to be changed.
% Output - This simulation saves records all ATFS, ATFS Given True
% Signal, and Percent Miss estimates (with standard errors),
% and it also generates plots of these three measures.

% INPUT THE CASE PARAMETERS-----
caseNumber = 1;
input.mewmaContLim = 3.3;
input.mcusumContLim = 4.76;
input.baseline = 90;
input.amplitude = 80;
input.dataSigma = 30;
input.lookBack = 30;
residSigma = 30.11;
outbreakPercent = 50;
input.outbreakMax = outbreakPercent*90/100;
caseParameters = ['Baseline = ' num2str(input.baseline) ...
                 ' Amplitude = ' num2str(input.amplitude) ...

```

```

        ' Sigma = ' num2str(input.dataSigma) ...
        ' Outbreak Peak = ' num2str(input.outbreakMax)];

% INPUT THE SIMULATION PARAMETERS-----
input.numLoops = 2500;

% DEFINE THE STATIC METHOD PARAMETERS-----
input.lambda = 0.2;
input.unscaledK = 0.37;
input.numStreams = 4;

% DEFINE THE STATIC OUTBREAK PARAMTERS-----
input.outbreak = true;
outbreakLengthMin = 3;
outbreakLengthMax = 15;
outbreakLengthStep = 2;
outbreakLengthArray = outbreakLengthMin : outbreakLengthStep :
outbreakLengthMax;

% ASSIGN IN-CONTROL CONDITIONS-----
% Assume the in-control mean is all zeros.
% The in-control sigma matrix depends on the case.
input.k = residSigma*input.unscaledK;
input.icMean = zeros(input.numStreams,1);
input.icSigma = residSigma^2*eye(input.numStreams);

% DECLARE STATISTIC HOLDING ARRAYS-----
% (The first column is MEWMA. The second column is MCUSUM)
ATFSGivenTrueSignalArray = zeros(7,2);
ATFSsGivenTrueSignalArray = zeros(7,2);
percentMissArray = zeros(7,2);
percentMissSEArray = zeros(7,2);
ATFSArray = zeros(7,2);
ATFSsArray = zeros(7,2);
arrayIndex = 1;

% MAIN SIMULATION EXECUTION LOOP-----
for index = outbreakLengthMin : outbreakLengthStep : outbreakLengthMax
    input.outbreakLength = index;
    mewmaResults = mewmaData(input);
    mcusumResults = mcusumData(input);
    ATFSGivenTrueSignalArray(arrayIndex,1) = ...
        mewmaResults.ATFSGivenDetect;
    ATFSsGivenTrueSignalArray(arrayIndex,1) = ...
        mewmaResults.ATFSsGivenDetect;
    percentMissArray(arrayIndex,1) = mewmaResults.percentMiss;
    percentMissSEArray(arrayIndex,1) = mewmaResults.percentMissSE;
    ATFSArray(arrayIndex,1) = mewmaResults.ATFS;
    ATFSsArray(arrayIndex,1) = mewmaResults.ATFSs;
    ATFSGivenTrueSignalArray(arrayIndex,2) = ...
        mcusumResults.ATFSGivenDetect;
    ATFSsGivenTrueSignalArray(arrayIndex,2) = ...
        mcusumResults.ATFSsGivenDetect;
    percentMissArray(arrayIndex,2) = mcusumResults.percentMiss;

```

```

    percentMissSEArray(arrayIndex,2) = mcusumResults.percentMissSE;
    ATFSArray(arrayIndex,2) = mcusumResults.ATFS;
    ATFSsseArray(arrayIndex,2) = mcusumResults.ATFSsse;
    arrayIndex = arrayIndex + 1;
end
save(['case' num2str(caseNumber) 'Stats'],'caseParameters',...
    'ATFSGivenTrueSignalArray','ATFSsseGivenTrueSignalArray',...
    'percentMissArray','percentMissSEArray',...
    'ATFSArray','ATFSsseArray');

% CONSTRUCT PLOTS FOR THE 3 MOES-----
% ATFS GIVEN TRUE SIGNAL PLOT
ATFSGivenTruePlot = figure('Name','ATFS Given True Signal Plot', ...
    'NumberTitle','off','Color','w');
set(0,'DefaultAxesColorOrder',[0 0 0], ...
    'DefaultAxesLineStyleOrder','-|--');
plot(outbreakLengthArray,ATFSGivenTrueSignalArray,'LineWidth',2);
set(gca,'XTick',outbreakLengthMin : outbreakLengthStep : ...
    outbreakLengthMax);
axis([3, 15, 0, 15]);
xlabel('Duration of outbreak in days','FontSize',14);
ylabel('ATFS Given a True Signal','FontSize',14);
saveas(ATFSGivenTruePlot,['case' num2str(caseNumber) ...
    'ATFSGivenTrueSignalPlot.fig']);
print('-tiff','-deps',['case' num2str(caseNumber) ...
    'ATFSGivenTrueSignalPic.eps']);

% PERCENT MISS PLOT
percentMissPlot = figure('Name','Percent Miss Plot', ...
    'NumberTitle','off','Color','w');
set(0,'DefaultAxesColorOrder',[0 0 0], ...
    'DefaultAxesLineStyleOrder','-|--');
plot(outbreakLengthArray,percentMissArray,'LineWidth',2);
set(gca,'XTick',outbreakLengthMin : outbreakLengthStep :
    outbreakLengthMax);
axis([3, 15, 0, 1]);
xlabel('Duration of outbreak in days','FontSize',14);
ylabel('Percent Miss','FontSize',14);
saveas(percentMissPlot,['case' num2str(caseNumber) ...
    'percentMissPlot.fig']);
print('-tiff','-deps',['case' num2str(caseNumber) ...
    'percentMissPic.eps']);

% ATFS PLOT
ATFSPlot = figure('Name','ATFS Plot','NumberTitle','off','Color','w');
set(0,'DefaultAxesColorOrder',[0 0 0], ...
    'DefaultAxesLineStyleOrder','-|--');
plot(outbreakLengthArray,ATFSArray,'LineWidth',2);
set(gca,'XTick',outbreakLengthMin : outbreakLengthStep : ...
    outbreakLengthMax);
axis('tight');
xlabel('Duration of outbreak in days','FontSize',14);
ylabel('ATFS','FontSize',14);
saveas(ATFSPlot,['case' num2str(caseNumber) 'ATFS.fig']);
print('-tiff','-deps',['case' num2str(caseNumber) 'ATFSPic.eps']);

```



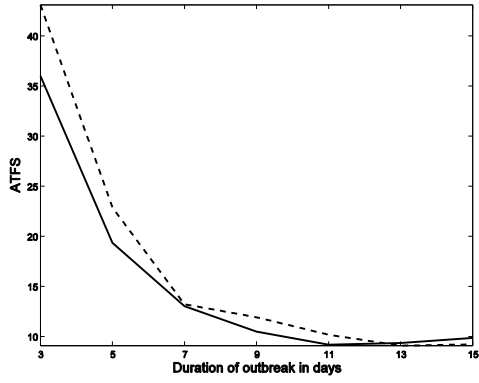
THIS PAGE INTENTIONALLY LEFT BLANK

## APPENDIX B.

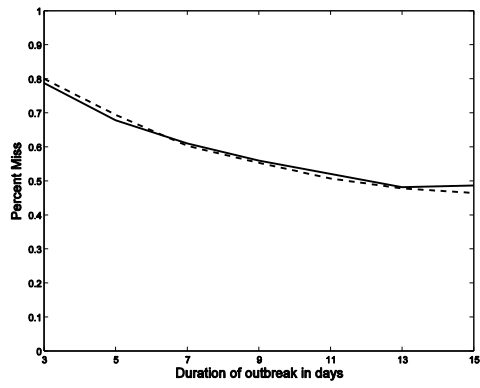
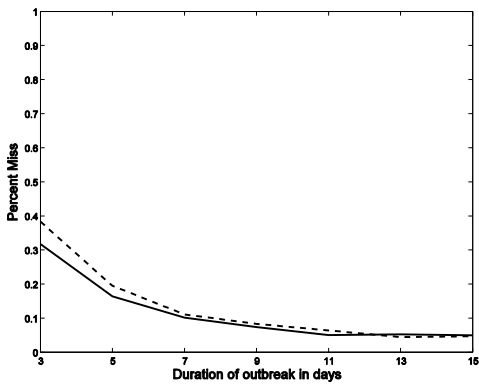
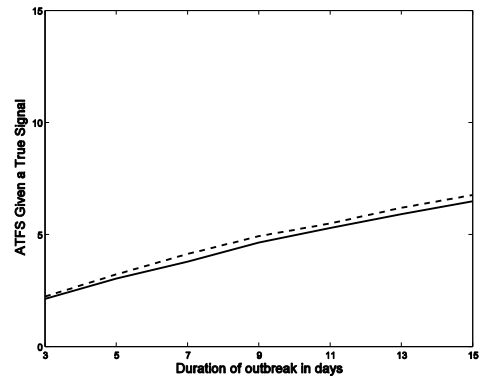
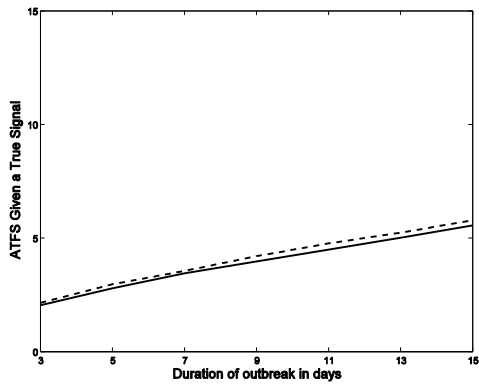
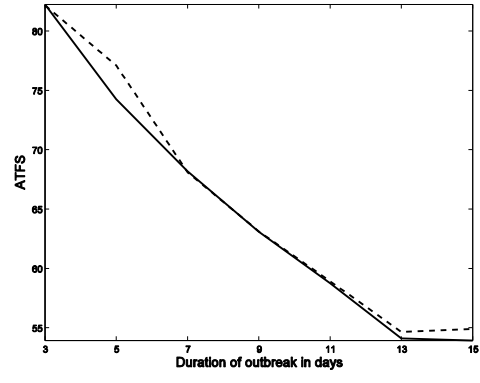
This appendix includes the results plots from all 18 simulation cases. In order of presentation, the plots show (1) ATFS, (2) ATFS given a signal during the outbreak, and (3) Percent Miss over varying outbreak durations. Note that the solid lines represent the modified MEWMA and the dashed lines represent the modified MCUSUM.

Case	Baseline $\beta$	Amplitude $A$	Data Sigma $\sigma$	Outbreak Peak $p$	Sliding Baseline $n$	Estimate of Residual SD $\hat{\sigma}_R$	MEWMA ContLim $h_{\text{MEWMA}}$	MCUSUM ContLim $h_{\text{MCUSUM}}$
1	90	80	30	45	30	30.11	3.28	4.64
2	90	80	30	22.5	30	30.11	3.28	4.64
3	90	80	30	9	30	30.11	3.28	4.64
4	90	80	10	45	30	10.62	3.31	4.78
5	90	80	10	22.5	30	10.62	3.31	4.78
6	90	80	10	9	30	10.62	3.31	4.78
7	90	20	30	45	40	31.46	3.26	4.6
8	90	20	30	22.5	40	31.46	3.26	4.6
9	90	20	30	9	40	31.46	3.26	4.6
10	90	20	10	45	35	10.59	3.26	4.6
11	90	20	10	22.5	35	10.59	3.26	4.6
12	90	20	10	9	35	10.59	3.26	4.6
13	90	0	30	45	45	31.29	3.27	4.62
14	90	0	30	22.5	45	31.29	3.27	4.62
15	90	0	30	9	45	31.29	3.27	4.62
16	90	0	10	45	35	10.58	3.25	4.57
17	90	0	10	22.5	35	10.58	3.25	4.57
18	90	0	10	9	35	10.58	3.25	4.57

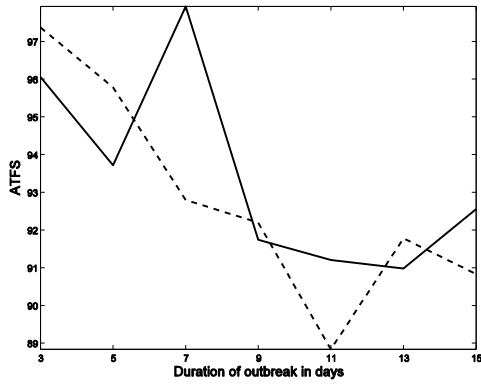
Case 1



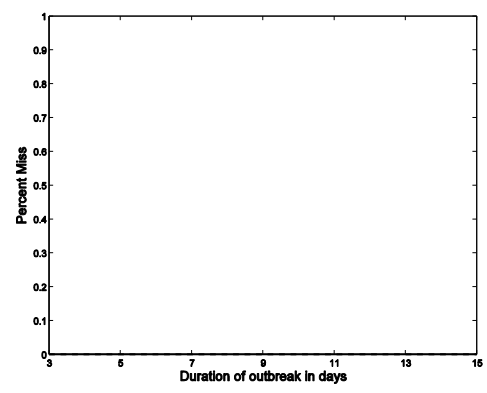
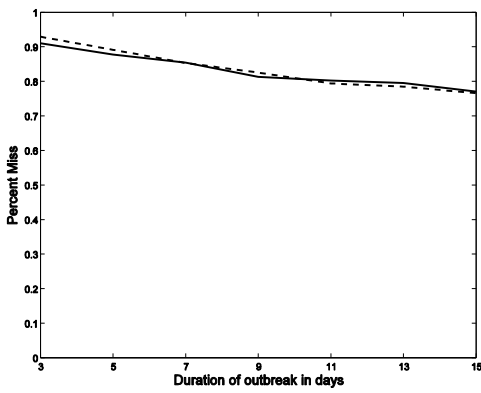
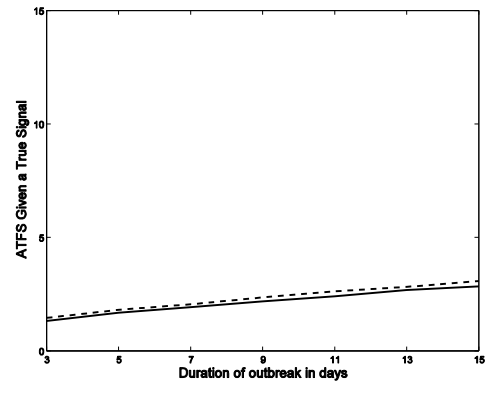
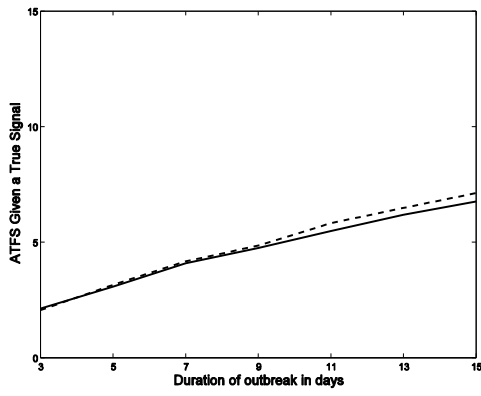
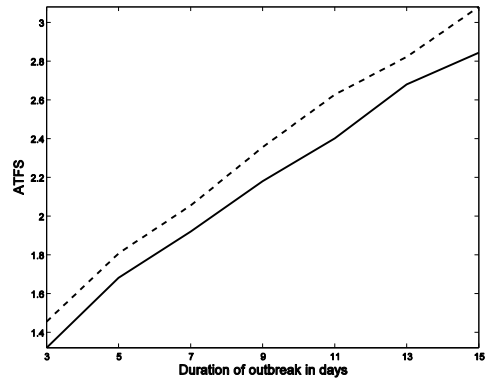
Case 2



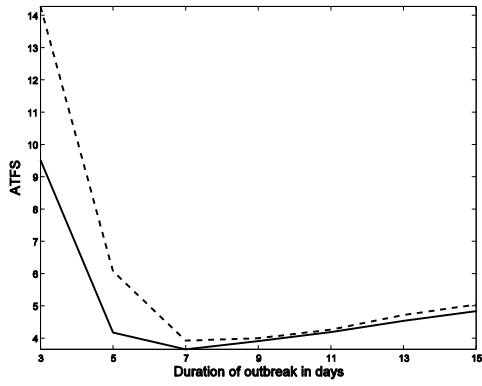
Case 3



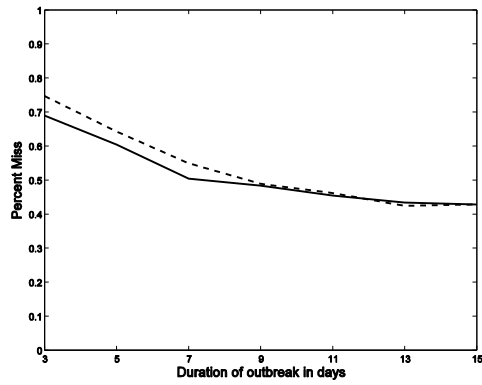
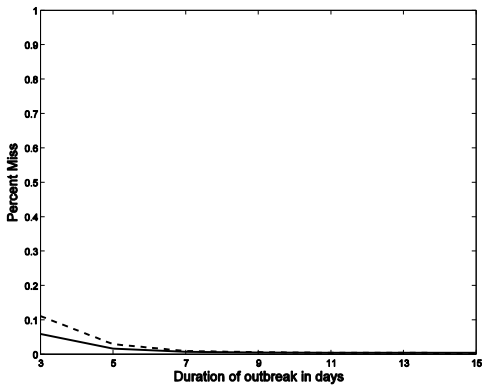
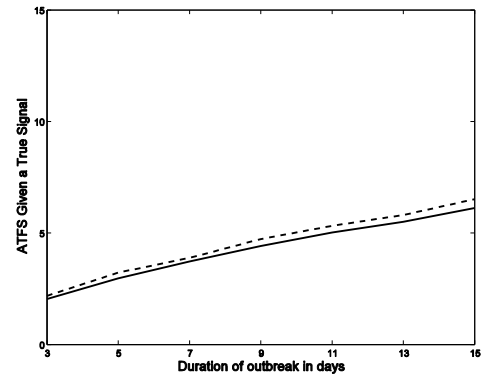
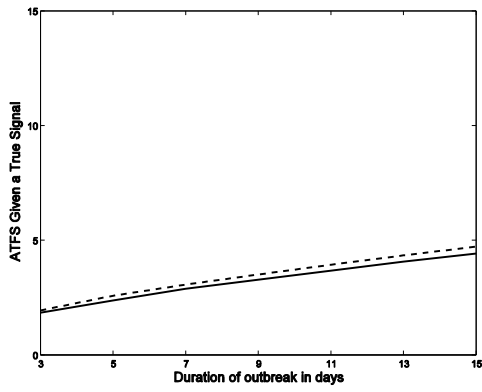
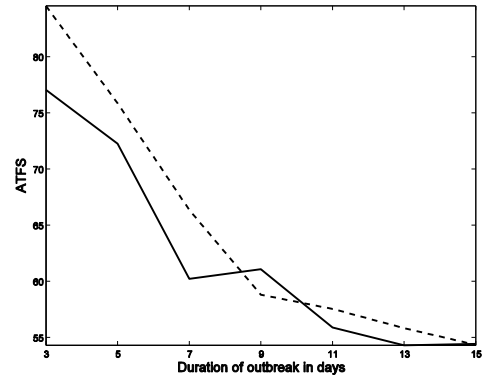
Case 4



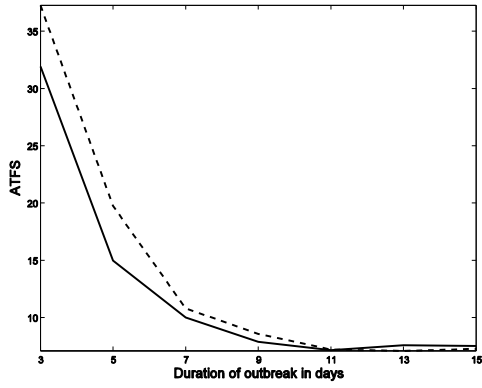
Case 5



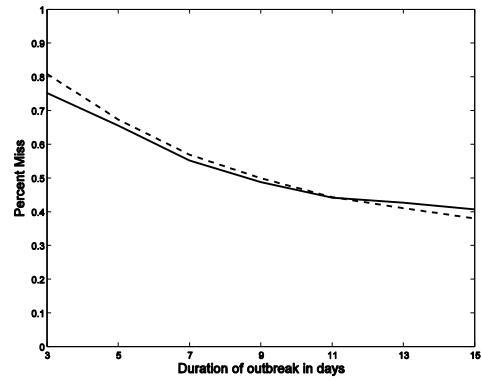
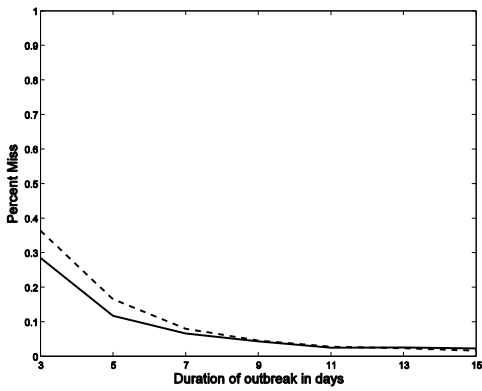
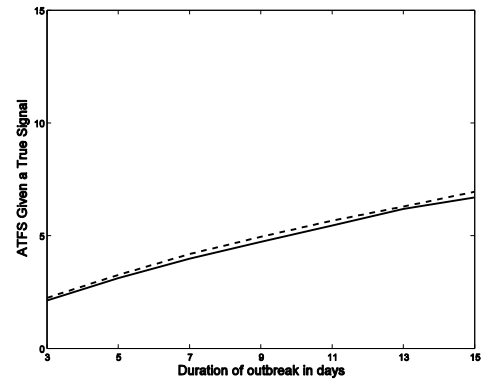
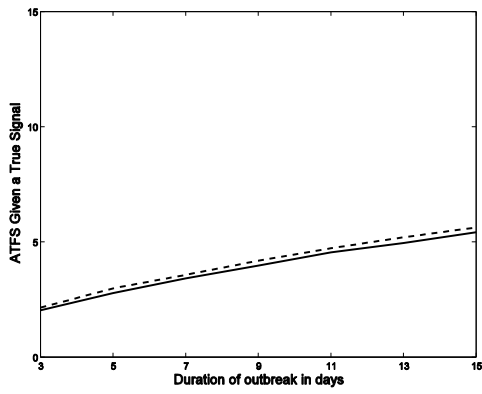
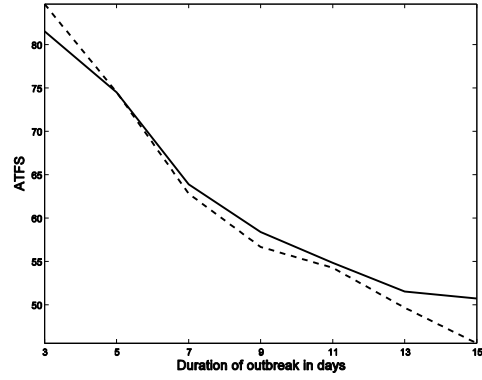
Case 6



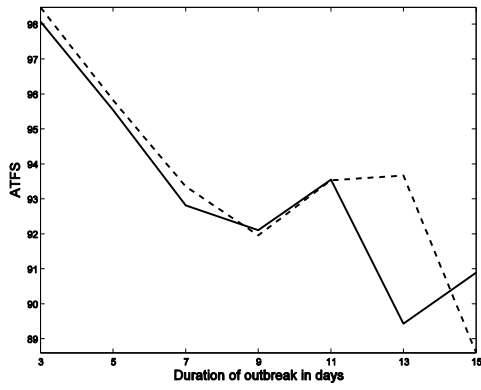
Case 7



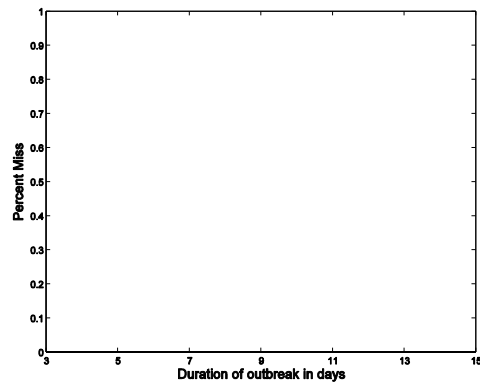
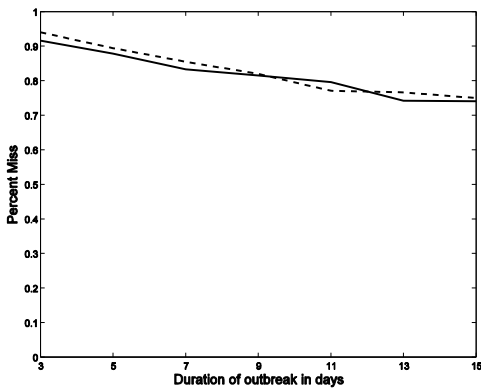
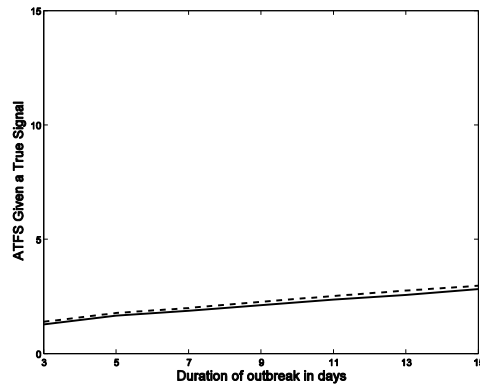
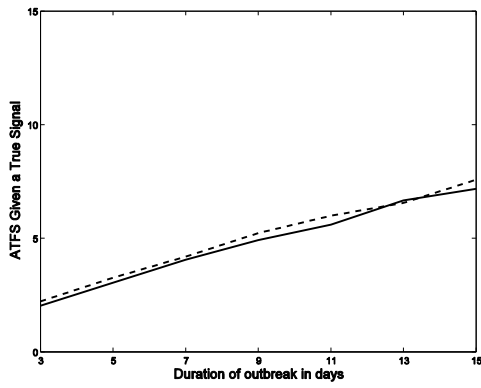
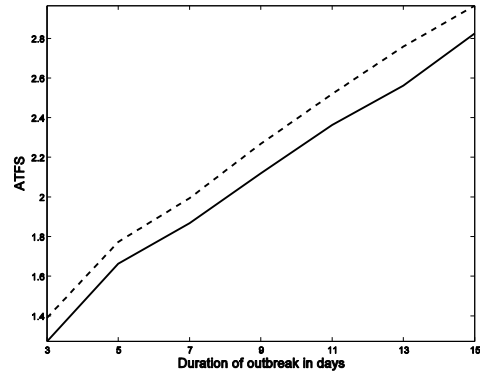
Case 8



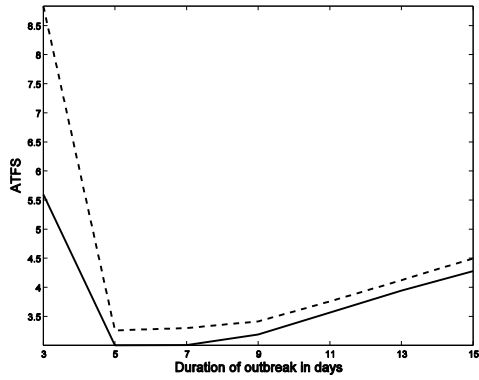
Case 9



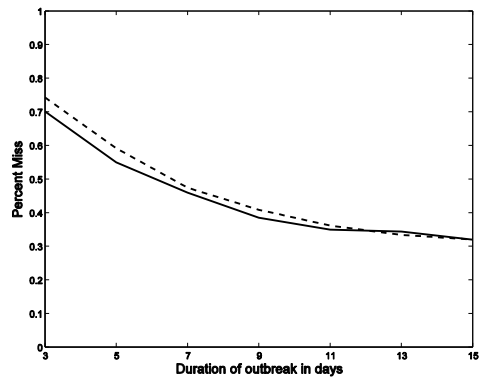
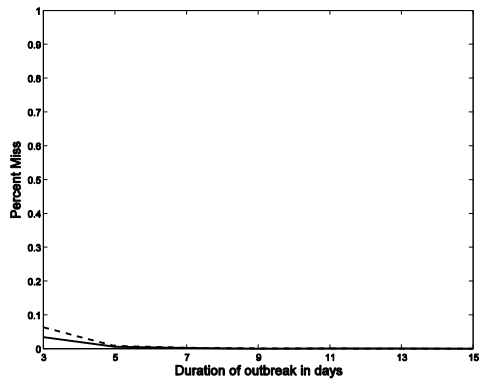
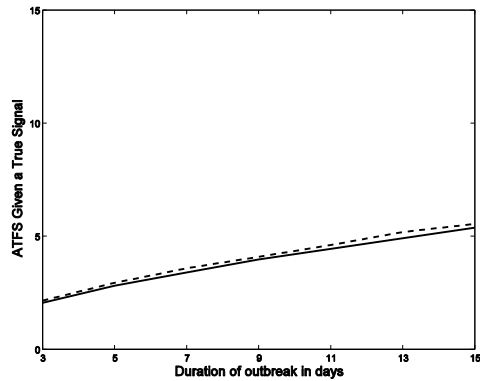
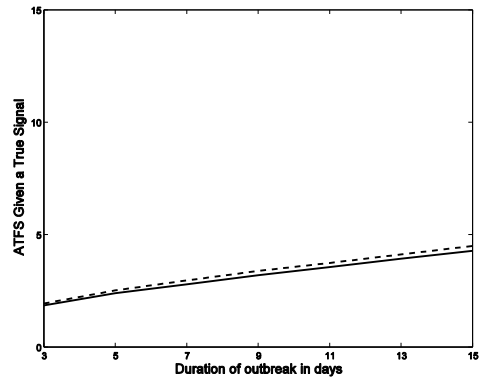
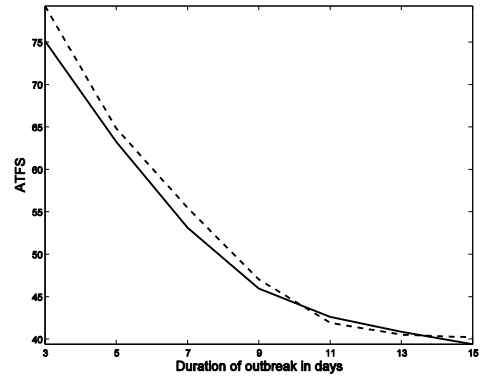
Case 10



Case 11

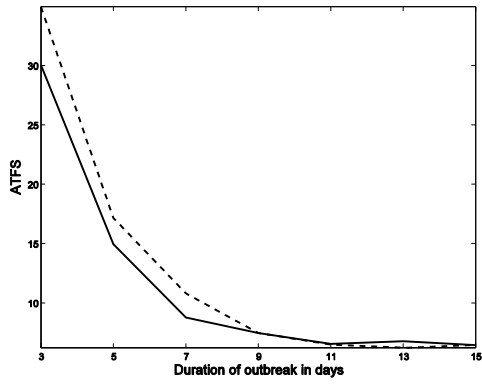


Case 12

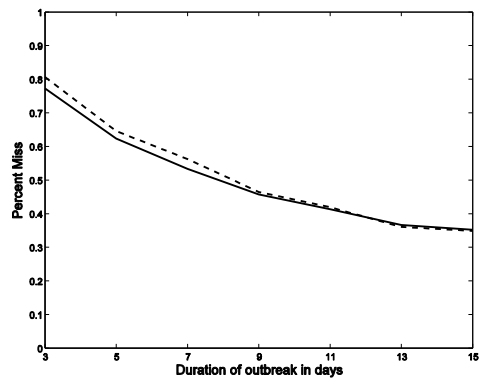
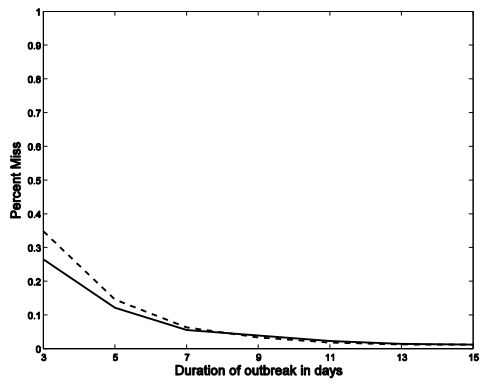
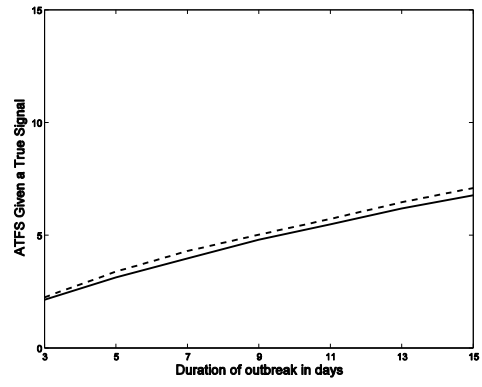
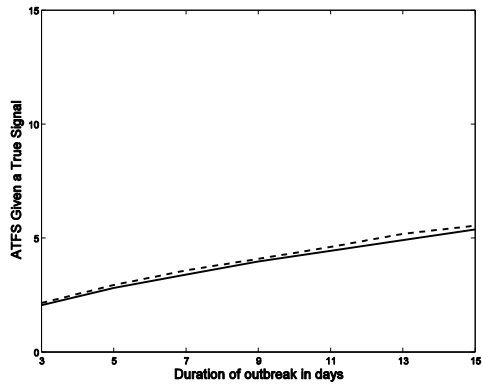
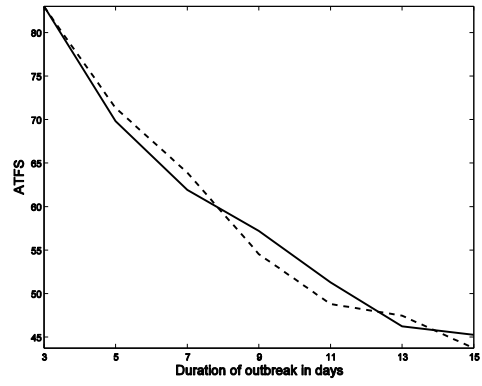




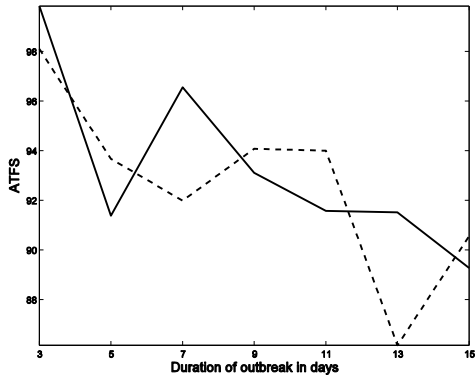
Case 13



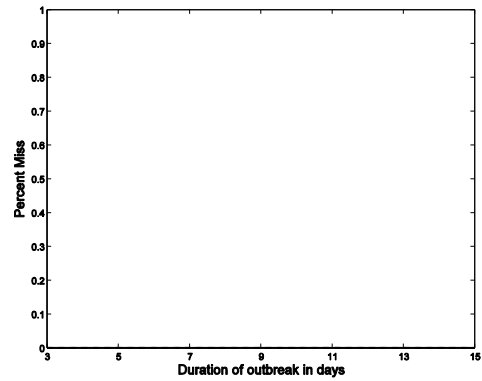
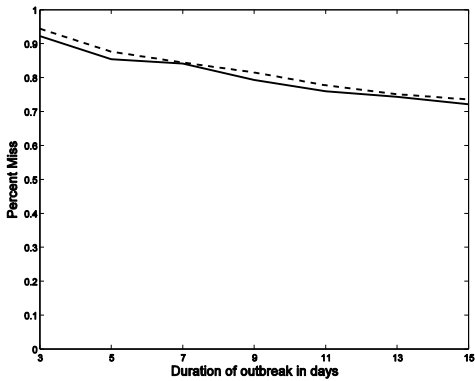
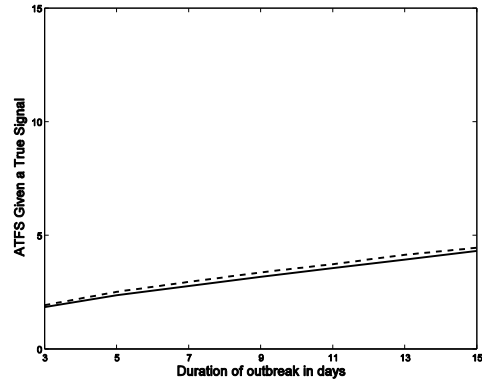
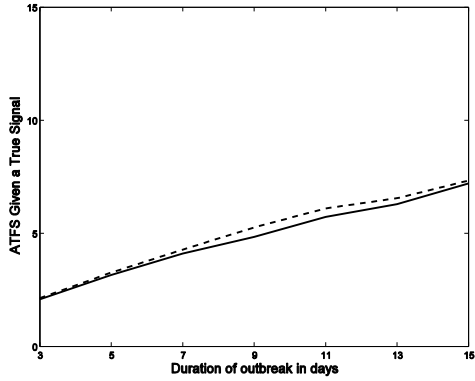
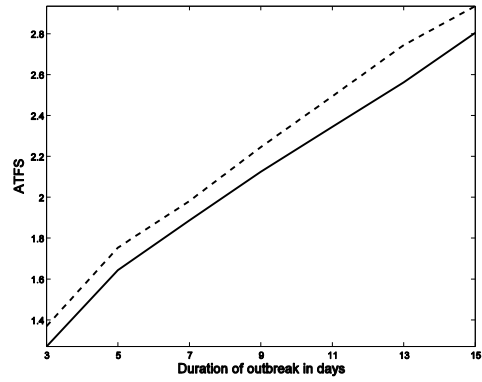
Case 14



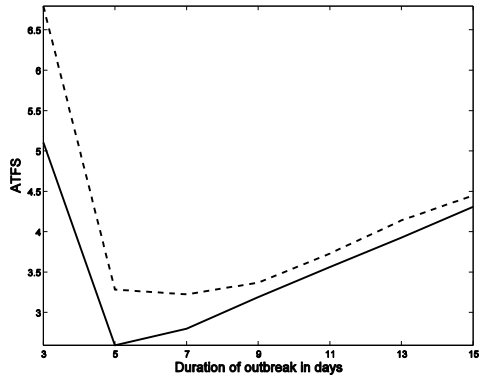
Case 15



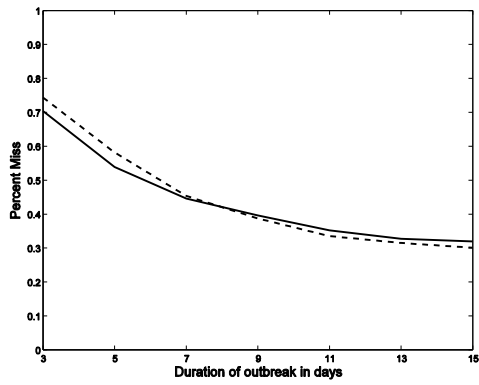
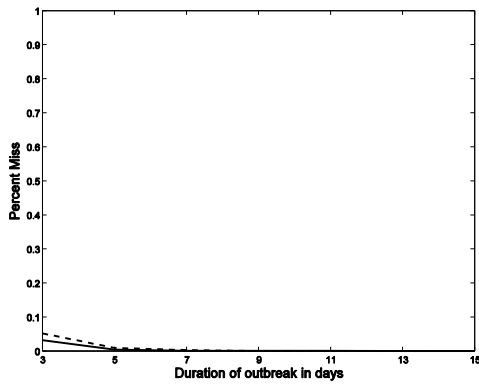
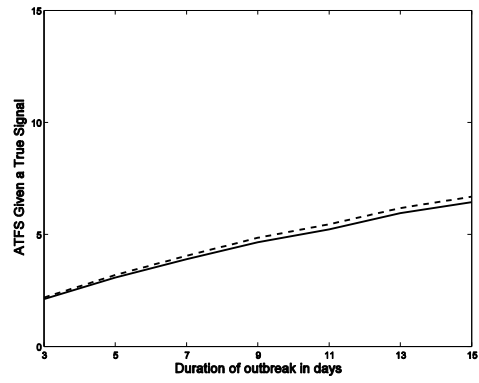
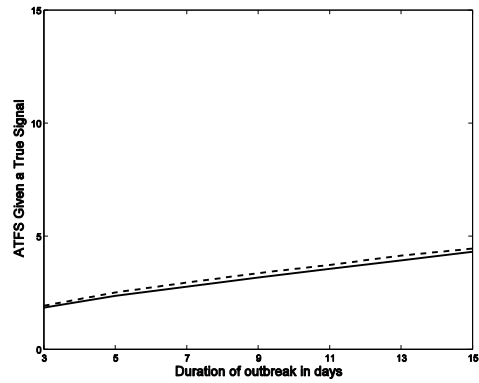
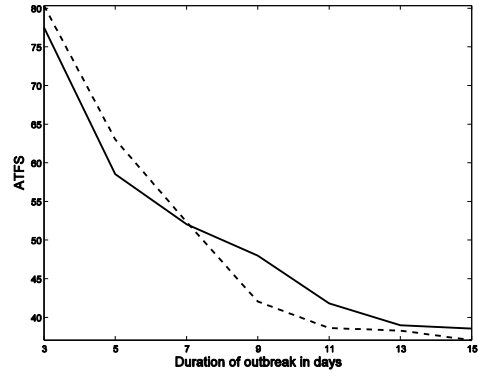
Case 16



Case 17



Case 18



## APPENDIX C.

This appendix includes the Microsoft Excel spreadsheet we used to assess whether the methods' performance was statistically significantly different in terms of Average Time to First Signal (ATFS), ATFS Given True Signal, and Percent Miss. The "widths" are computed by taking the sum of the standard errors of the two methods and multiplying by 2. Comparing the differences between the MEWMA and MCUSUM methods with the "widths" of the standard errors of the methods, the gray highlighted cell entries denote a "width" that is smaller than the difference of the given performance metric. This is an approximate two-sample hypothesis test with a 5 percent significance level.

**A. AVERAGE TIME TO FIRST SIGNAL**

Case	ATFS			se		widths
	MEWMA	MCUSUM	Difference	MEWMA	MCUSUM	
1	35.996	43.1048	7.1088	1.5204	1.6494	6.3396
	19.3484	22.888	3.5396	1.0609	1.1569	4.4356
	13.0332	13.2184	0.1852	0.789	0.7574	3.0928
	10.4888	11.9264	1.4376	0.6474	0.6913	2.6774
	9.1864	10.1744	0.988	0.5598	0.5843	2.2882
	9.3476	9.0936	0.254	0.5069	0.4882	1.9902
	9.8628	9.2204	0.6424	0.5587	0.4494	2.0162
2	82.2448	82.1708	0.074	1.8829	1.9152	7.5962
	74.2488	77.0992	2.8504	1.89	1.9123	7.6046
	68.1792	68.0976	0.0816	1.8275	1.86	7.375
	63.1012	63.1204	0.0192	1.7644	1.8261	7.181
	58.7412	58.9208	0.1796	1.706	1.7287	6.8694
	54.1096	54.6408	0.5312	1.6532	1.6273	6.561
	53.9268	54.8968	0.97	1.5347	1.6419	6.3532
3	96.0572	97.3696	1.3124	1.9743	1.9713	7.8912
	93.7144	95.7844	2.07	1.9397	1.9243	7.728
	97.9308	92.7952	5.1356	1.9983	1.9134	7.8234
	91.7412	92.1948	0.4536	1.9892	1.8836	7.7456
	91.2048	88.8364	2.3684	1.9329	1.8465	7.5588
	90.9804	91.7796	0.7992	1.8919	1.9157	7.6152
	92.556	90.832	1.724	1.9561	1.8925	7.6972
4	1.3208	1.4552	0.1344	0.0093	0.01	0.0386
	1.6812	1.8076	0.1264	0.0098	0.0089	0.0374
	1.9208	2.0544	0.1336	0.0112	0.0108	0.044
	2.1808	2.356	0.1752	0.0133	0.0128	0.0522
	2.4008	2.6268	0.226	0.0149	0.0143	0.0584
	2.6804	2.822	0.1416	0.0167	0.0157	0.0648
	2.8436	3.0804	0.2368	0.0187	0.0178	0.073
5	9.5108	14.2784	4.7676	0.7232	0.8797	3.2058
	4.1692	6.0712	1.902	0.3126	0.4615	1.5482
	3.6516	3.9224	0.2708	0.2102	0.2128	0.846
	3.9032	3.994	0.0908	0.1893	0.1652	0.709
	4.1868	4.2608	0.074	0.1856	0.1274	0.626
	4.534	4.7148	0.1808	0.1715	0.1368	0.6166
	4.8384	5.0396	0.2012	0.1686	0.1361	0.6094
6	77.0572	84.5268	7.4696	1.7675	1.8489	7.2328
	72.258	75.862	3.604	1.7241	1.7177	6.8836
	60.2312	66.3656	6.1344	1.6191	1.6577	6.5536
	61.0808	58.806	2.2748	1.6679	1.5615	6.4588
	55.8964	57.5468	1.6504	1.5517	1.568	6.2394
	54.3048	55.8352	1.5304	1.5282	1.5403	6.137
	54.43	54.3408	0.0892	1.5688	1.5134	6.1644

7	31.908	37.2636	5.3556	1.4164	1.4426	5.718
	14.9604	19.7476	4.7872	0.9157	1.0466	3.9246
	10.0004	10.8008	0.8004	0.6803	0.6655	2.6916
	7.8792	8.5736	0.6944	0.48	0.5735	2.107
	7.1432	7.1984	0.0552	0.4126	0.4448	1.7148
	7.574	7.0892	0.4848	0.43	0.3393	1.5386
	7.5196	7.2612	0.2584	0.4013	0.3727	1.548
8	81.5456	84.688	3.1424	1.9988	1.952	7.9016
	74.5012	74.5252	0.024	1.987	1.9175	7.809
	63.8872	62.8192	1.068	1.7861	1.7143	7.0008
	58.3876	56.6692	1.7184	1.7051	1.6056	6.6214
	54.8284	54.2676	0.5608	1.7245	1.6507	6.7504
	51.5272	49.658	1.8692	1.6201	1.5526	6.3454
	50.7256	45.5596	5.166	1.6303	1.493	6.2466
9	98.0696	98.4796	0.41	2.0141	1.956	7.9402
	95.5452	95.8252	0.28	2.0455	2.0027	8.0964
	92.8092	93.3532	0.544	1.9396	1.8941	7.6674
	92.104	91.9516	0.1524	1.9258	1.8703	7.5922
	93.5464	93.5296	0.0168	1.9773	1.9951	7.9448
	89.4308	93.6644	4.2336	1.9529	2.0181	7.942
	90.8908	88.594	2.2968	1.899	1.8456	7.4892
10	1.2712	1.3896	0.1184	0.0089	0.0098	0.0374
	1.6632	1.7728	0.1096	0.0097	0.0088	0.037
	1.8676	1.9936	0.126	0.0103	0.0099	0.0404
	2.1184	2.268	0.1496	0.0125	0.0122	0.0494
	2.3628	2.5192	0.1564	0.0137	0.0131	0.0536
	2.5608	2.758	0.1972	0.0149	0.015	0.0598
	2.8248	2.9648	0.14	0.0164	0.0165	0.0658
11	5.5968	8.8368	3.24	0.5068	0.7597	2.533
	3.0024	3.258	0.2556	0.2137	0.2279	0.8832
	3.0056	3.2992	0.2936	0.1161	0.1571	0.5464
	3.1884	3.4128	0.2244	0.0193	0.0276	0.0938
	3.5656	3.7568	0.1912	0.0266	0.0267	0.1066
	3.9404	4.1216	0.1812	0.0289	0.0249	0.1076
	4.2796	4.4968	0.2172	0.0281	0.0287	0.1136
12	75.118	79.26	4.142	1.8068	1.8652	7.344
	63.2436	64.83	1.5864	1.7956	1.7606	7.1124
	53.1076	55.4744	2.3668	1.6316	1.7198	6.7028
	45.9408	47.0348	1.094	1.5957	1.5546	6.3006
	42.624	41.9264	0.6976	1.5203	1.3922	5.825
	40.86	40.5076	0.3524	1.3831	1.4269	5.62
	39.392	40.1988	0.8068	1.4138	1.4708	5.7692
13	30.0236	34.946	4.9224	1.3689	1.3975	5.5328
	14.9448	17.1596	2.2148	0.8758	0.9325	3.6166
	8.7704	10.8004	2.03	0.5607	0.7825	2.6864
	7.474	7.4664	0.0076	0.4383	0.4671	1.8108
	6.556	6.4892	0.0668	0.355	0.443	1.596
	6.7608	6.2232	0.5376	0.4721	0.2289	1.402
	6.44	6.456	0.016	0.242	0.2334	0.9508

14	83.0212	82.9684	0.0528	1.9529	1.8772	7.6602
	69.8128	71.3376	1.5248	1.9061	1.8628	7.5378
	61.9108	63.9184	2.0076	1.8548	1.7905	7.2906
	57.1908	54.5304	2.6604	1.7526	1.669	6.8432
	51.298	48.8024	2.4956	1.6747	1.5898	6.529
	46.2408	47.4668	1.226	1.5866	1.6002	6.3736
	45.2508	43.73	1.5208	1.5164	1.5384	6.1096
15	99.8272	98.094	1.7332	1.982	1.9166	7.7972
	91.3728	93.668	2.2952	1.9527	1.9542	7.8138
	96.5468	91.9844	4.5624	2.0929	1.9187	8.0232
	93.1012	94.0712	0.97	1.9574	1.9689	7.8526
	91.5664	93.9976	2.4312	2.0194	2.037	8.1128
	91.5092	86.1528	5.3564	1.9662	1.8612	7.6548
	89.2668	90.5608	1.294	1.9327	1.9796	7.8246
16	1.27	1.3692	0.0992	0.0089	0.0097	0.0372
	1.6432	1.7532	0.11	0.0097	0.0089	0.0372
	1.8872	1.9816	0.0944	0.0101	0.0101	0.0404
	2.1248	2.2456	0.1208	0.012	0.0122	0.0484
	2.344	2.4932	0.1492	0.0139	0.0131	0.054
	2.5616	2.7444	0.1828	0.0147	0.0145	0.0584
	2.804	2.9348	0.1308	0.0166	0.0161	0.0654
17	5.1052	6.7952	1.69	0.4986	0.6275	2.2522
	2.5928	3.2828	0.69	0.0843	0.2107	0.59
	2.7972	3.2236	0.4264	0.0261	0.1867	0.4256
	3.1904	3.3688	0.1784	0.0263	0.0193	0.0912
	3.5632	3.7288	0.1656	0.0249	0.0215	0.0928
	3.9256	4.14	0.2144	0.0246	0.0243	0.0978
	4.308	4.4488	0.1408	0.0278	0.0271	0.1098
18	77.5112	80.3436	2.8324	1.9155	1.9427	7.7164
	58.5388	63.0164	4.4776	1.7251	1.7697	6.9896
	52.0572	52.3544	0.2972	1.713	1.6894	6.8048
	47.9748	42.0432	5.9316	1.644	1.4417	6.1714
	41.8172	38.6192	3.198	1.4844	1.4458	5.8604
	38.9652	38.2904	0.6748	1.4425	1.4501	5.7852
	38.5532	37.0436	1.5096	1.4433	1.3955	5.6776

**B. ATFS GIVEN TRUE SIGNAL**

Case	ATFSGivenTrue			se		widths
	MEWMA	MCUSUM	Difference	MEWMA	MCUSUM	
1	2.0428	2.1473	0.1045	0.013	0.0142	0.0544
	2.7919	2.9647	0.1728	0.0178	0.018	0.0716
	3.4441	3.5574	0.1133	0.0224	0.0225	0.0898
	3.9659	4.2029	0.237	0.0267	0.0268	0.107
	4.4952	4.7645	0.2693	0.0311	0.0311	0.1244
	5.016	5.2394	0.2234	0.0353	0.0348	0.1402
	5.5556	5.7801	0.2245	0.0402	0.0385	0.1574
2	2.1316	2.23	0.0984	0.0282	0.0299	0.1162
	3.0286	3.2235	0.1949	0.0381	0.0371	0.1504
	3.7889	4.1319	0.343	0.0445	0.0435	0.176
	4.6388	4.9258	0.287	0.0492	0.049	0.1964
	5.2886	5.4955	0.2069	0.0586	0.0565	0.2302
	5.9098	6.1953	0.2855	0.061	0.0654	0.2528
	6.4868	6.7647	0.2779	0.0715	0.0677	0.2784
3	2.1289	2.0621	0.0668	0.051	0.059	0.22
	3.0814	3.1544	0.073	0.0757	0.0756	0.3026
	4.0849	4.1744	0.0895	0.0875	0.0879	0.3508
	4.7457	4.8513	0.1056	0.0975	0.1005	0.396
	5.4828	5.8252	0.3424	0.1195	0.1083	0.4556
	6.1855	6.4814	0.2959	0.1241	0.1269	0.502
	6.7635	7.128	0.3645	0.1464	0.1394	0.5716
4	1.3208	1.4552	0.1344	0.0093	0.01	0.0386
	1.6812	1.8076	0.1264	0.0098	0.0089	0.0374
	1.9208	2.0544	0.1336	0.0112	0.0108	0.044
	2.1808	2.356	0.1752	0.0133	0.0128	0.0522
	2.4008	2.6268	0.226	0.0149	0.0143	0.0584
	2.6804	2.822	0.1416	0.0167	0.0157	0.0648
	2.8436	3.0804	0.2368	0.0187	0.0178	0.073
5	1.8415	1.9393	0.0978	0.0099	0.0102	0.0402
	2.3717	2.5781	0.2064	0.0148	0.0145	0.0586
	2.8832	3.0646	0.1814	0.0181	0.0177	0.0716
	3.2758	3.5042	0.2284	0.0217	0.0211	0.0856
	3.6771	3.9305	0.2534	0.0254	0.0239	0.0986
	4.0627	4.3407	0.278	0.0289	0.0284	0.1146
	4.4158	4.72	0.3042	0.0328	0.0322	0.13
6	2.0463	2.1801	0.1338	0.023	0.0257	0.0974
	2.9717	3.2237	0.252	0.0323	0.0337	0.132
	3.725	3.8873	0.1623	0.0387	0.0407	0.1588
	4.4136	4.7293	0.3157	0.0451	0.0448	0.1798
	5.0293	5.3254	0.2961	0.0527	0.0507	0.2068
	5.5099	5.8118	0.3019	0.0591	0.0554	0.229
	6.126	6.5213	0.3953	0.065	0.0644	0.2588



7	2.0375	2.1477	0.1102	0.0136	0.014	0.0552
	2.7822	2.9823	0.2001	0.0172	0.0178	0.07
	3.4135	3.5706	0.1571	0.0215	0.0217	0.0864
	3.9737	4.1832	0.2095	0.0249	0.0264	0.1026
	4.5414	4.7269	0.1855	0.0288	0.0294	0.1164
	4.9561	5.2002	0.2441	0.034	0.0324	0.1328
	5.4182	5.6285	0.2103	0.0367	0.0373	0.148
8	2.1323	2.2469	0.1146	0.0274	0.0297	0.1142
	3.1185	3.2619	0.1434	0.0359	0.0376	0.147
	3.9804	4.1874	0.207	0.0416	0.0421	0.1674
	4.7237	4.9505	0.2268	0.0479	0.0461	0.188
	5.4513	5.6695	0.2182	0.0509	0.0521	0.206
	6.1877	6.2931	0.1054	0.059	0.0608	0.2396
	6.6959	6.9484	0.2525	0.0654	0.064	0.2588
9	2.0286	2.22	0.1914	0.0518	0.0597	0.223
	3.0428	3.2604	0.2176	0.075	0.077	0.304
	4.0574	4.1901	0.1327	0.0844	0.0907	0.3502
	4.9158	5.2262	0.3104	0.1005	0.0973	0.3956
	5.6027	5.9913	0.3886	0.117	0.1054	0.4448
	6.6641	6.5573	0.1068	0.1157	0.1186	0.4686
	7.1806	7.5783	0.3977	0.1335	0.1329	0.5328
10	1.2712	1.3896	0.1184	0.0089	0.0098	0.0374
	1.6632	1.7728	0.1096	0.0097	0.0088	0.037
	1.8676	1.9936	0.126	0.0103	0.0099	0.0404
	2.1184	2.268	0.1496	0.0125	0.0122	0.0494
	2.3628	2.5192	0.1564	0.0137	0.0131	0.0536
	2.5608	2.758	0.1972	0.0149	0.015	0.0598
	2.8248	2.9648	0.14	0.0164	0.0165	0.0658
11	1.8542	1.9381	0.0839	0.0092	0.0093	0.037
	2.3926	2.5206	0.128	0.0133	0.0133	0.0532
	2.7924	2.9619	0.1695	0.0163	0.0163	0.0652
	3.1884	3.3847	0.1963	0.0193	0.0191	0.0768
	3.5514	3.7419	0.1905	0.0225	0.0221	0.0892
	3.9268	4.1216	0.1948	0.0255	0.0249	0.1008
	4.2796	4.4968	0.2172	0.0281	0.0287	0.1136
12	2.0908	2.2081	0.1173	0.024	0.0265	0.101
	3.1075	3.2184	0.1109	0.0305	0.0312	0.1234
	3.8395	4.0601	0.2206	0.0349	0.0366	0.143
	4.5533	4.8208	0.2675	0.0391	0.041	0.1602
	5.2176	5.5366	0.319	0.0465	0.0469	0.1868
	5.8867	6.0912	0.2045	0.0523	0.053	0.2106
	6.3839	6.6965	0.3126	0.0568	0.0561	0.2258
13	2.0533	2.1485	0.0952	0.0129	0.0139	0.0536
	2.8074	2.9363	0.1289	0.0174	0.0183	0.0714
	3.3994	3.5796	0.1802	0.0218	0.0211	0.0858
	3.9742	4.0873	0.1131	0.0254	0.0253	0.1014
	4.4409	4.6069	0.166	0.029	0.0278	0.1136
	4.9128	5.1843	0.2715	0.0327	0.0318	0.129
	5.3764	5.5399	0.1635	0.0371	0.036	0.1462

14	2.1371	2.25	0.1129	0.0279	0.0309	0.1176
	3.1265	3.386	0.2595	0.0331	0.0363	0.1388
	3.9709	4.2962	0.3253	0.0409	0.0416	0.165
	4.8069	5.0246	0.2177	0.0468	0.0465	0.1866
	5.4837	5.7204	0.2367	0.0506	0.0532	0.2076
	6.1893	6.4647	0.2754	0.0552	0.0572	0.2248
	6.7752	7.0946	0.3194	0.0641	0.0641	0.2564
15	2.0821	2.1357	0.0536	0.0516	0.0651	0.2334
	3.1589	3.2686	0.1097	0.0666	0.0734	0.28
	4.1111	4.2828	0.1717	0.0898	0.0871	0.3538
	4.8398	5.2662	0.4264	0.0998	0.0966	0.3928
	5.7233	6.1059	0.3826	0.1095	0.1122	0.4434
	6.2975	6.5563	0.2588	0.1172	0.1216	0.4776
	7.2069	7.3369	0.13	0.1253	0.1265	0.5036
16	1.27	1.3692	0.0992	0.0089	0.0097	0.0372
	1.6432	1.7532	0.11	0.0097	0.0089	0.0372
	1.8872	1.9816	0.0944	0.0101	0.0101	0.0404
	2.1248	2.2456	0.1208	0.012	0.0122	0.0484
	2.344	2.4932	0.1492	0.0139	0.0131	0.054
	2.5616	2.7444	0.1828	0.0147	0.0145	0.0584
	2.804	2.9348	0.1308	0.0166	0.0161	0.0654
17	1.8381	1.9258	0.0877	0.0092	0.0095	0.0374
	2.3647	2.5127	0.148	0.0132	0.0131	0.0526
	2.7682	2.9471	0.1789	0.0161	0.0158	0.0638
	3.1725	3.3688	0.1963	0.0192	0.0193	0.077
	3.5514	3.7288	0.1774	0.0219	0.0215	0.0868
	3.9256	4.14	0.2144	0.0246	0.0243	0.0978
	4.308	4.4488	0.1408	0.0278	0.0271	0.1098
18	2.1132	2.1841	0.0709	0.0242	0.0256	0.0996
	3.0807	3.1885	0.1078	0.0294	0.0309	0.1206
	3.8975	4.0505	0.153	0.0352	0.035	0.1404
	4.6541	4.8525	0.1984	0.0408	0.0394	0.1604
	5.2329	5.4561	0.2232	0.0465	0.0445	0.182
	5.9501	6.1798	0.2297	0.0526	0.0505	0.2062
	6.4401	6.6842	0.2441	0.0574	0.0574	0.2296

**C. PERCENT MISS**

Case	PercentMiss			se		widths
	MEWMA	MCUSUM	Difference	MEWMA	MCUSUM	
1	0.3172	0.3836	0.0664	0.0093	0.0097	0.038
	0.164	0.1948	0.0308	0.0074	0.0079	0.0306
	0.1012	0.1108	0.0096	0.006	0.0063	0.0246
	0.074	0.0832	0.0092	0.0052	0.0055	0.0214
	0.05	0.064	0.014	0.0044	0.0049	0.0186
	0.0524	0.0444	0.008	0.0045	0.0041	0.0172
	0.0496	0.0468	0.0028	0.0043	0.0042	0.017
2	0.7872	0.8	0.0128	0.0082	0.008	0.0324
	0.678	0.694	0.016	0.0093	0.0092	0.037
	0.6096	0.6028	0.0068	0.0098	0.0098	0.0392
	0.5592	0.5528	0.0064	0.0099	0.0099	0.0396
	0.5204	0.5068	0.0136	0.01	0.01	0.04
	0.4812	0.4776	0.0036	0.01	0.01	0.04
	0.4864	0.4644	0.022	0.01	0.01	0.04
3	0.91	0.9292	0.0192	0.0057	0.0051	0.0216
	0.8772	0.8912	0.014	0.0066	0.0062	0.0256
	0.854	0.8532	0.0008	0.0071	0.0071	0.0284
	0.8128	0.8252	0.0124	0.0078	0.0076	0.0308
	0.802	0.794	0.008	0.008	0.0081	0.0322
	0.7952	0.7848	0.0104	0.0081	0.0082	0.0326
	0.77	0.7656	0.0044	0.0084	0.0085	0.0338
4	0	0	0	0	0	0
	0	0	0	0	0	0
	0	0	0	0	0	0
	0	0	0	0	0	0
	0	0	0	0	0	0
	0	0	0	0	0	0
	0	0	0	0	0	0
5	0.0588	0.1108	0.052	0.0047	0.0063	0.022
	0.0164	0.0292	0.0128	0.0025	0.0034	0.0118
	0.0068	0.0092	0.0024	0.0016	0.0019	0.007
	0.0052	0.006	0.0008	0.0014	0.0015	0.0058
	0.004	0.004	0	0.0013	0.0013	0.0052
	0.004	0.0044	0.0004	0.0013	0.0013	0.0052
	0.0044	0.0028	0.0016	0.0013	0.0011	0.0048
6	0.6892	0.7468	0.0576	0.0093	0.0087	0.036
	0.6044	0.6424	0.038	0.0098	0.0096	0.0388
	0.504	0.5492	0.0452	0.01	0.01	0.04
	0.4836	0.4888	0.0052	0.01	0.01	0.04
	0.454	0.4616	0.0076	0.01	0.01	0.04
	0.4336	0.424	0.0096	0.0099	0.0099	0.0396
	0.4284	0.4284	0	0.0099	0.0099	0.0396

7	0.2848	0.3636	0.0788	0.009	0.0096	0.0372
	0.1168	0.1648	0.048	0.0064	0.0074	0.0276
	0.0656	0.0796	0.014	0.005	0.0054	0.0208
	0.0432	0.0456	0.0024	0.0041	0.0042	0.0166
	0.0248	0.0276	0.0028	0.0031	0.0033	0.0128
	0.0256	0.0232	0.0024	0.0032	0.003	0.0124
	0.0224	0.016	0.0064	0.003	0.0025	0.011
8	0.752	0.8088	0.0568	0.0086	0.0079	0.033
	0.6556	0.6732	0.0176	0.0095	0.0094	0.0378
	0.5516	0.5688	0.0172	0.0099	0.0099	0.0396
	0.4876	0.4992	0.0116	0.01	0.01	0.04
	0.4416	0.4432	0.0016	0.0099	0.0099	0.0396
	0.4268	0.4104	0.0164	0.0099	0.0098	0.0394
	0.4068	0.38	0.0268	0.0098	0.0097	0.039
9	0.916	0.94	0.024	0.0055	0.0047	0.0204
	0.8784	0.894	0.0156	0.0065	0.0062	0.0254
	0.8328	0.8548	0.022	0.0075	0.007	0.029
	0.8148	0.8196	0.0048	0.0078	0.0077	0.031
	0.7956	0.7708	0.0248	0.0081	0.0084	0.033
	0.7416	0.766	0.0244	0.0088	0.0085	0.0346
	0.7408	0.7496	0.0088	0.0088	0.0087	0.035
10	0	0	0	0	0	0
	0	0	0	0	0	0
	0	0	0	0	0	0
	0	0	0	0	0	0
	0	0	0	0	0	0
	0	0	0	0	0	0
	0	0	0	0	0	0
11	0.034	0.0628	0.0288	0.0036	0.0049	0.017
	0.0056	0.008	0.0024	0.0015	0.0018	0.0066
	0.002	0.0024	0.0004	0.0009	0.001	0.0038
	0	0.0008	0.0008	0	0.0006	0.0012
	0.0004	0.0004	0	0.0004	0.0004	0.0016
	0.0004	0	0.0004	0.0004	0	0.0008
	0	0	0	0	0	0
12	0.7004	0.7424	0.042	0.0092	0.0087	0.0358
	0.5496	0.5916	0.042	0.01	0.0098	0.0396
	0.4592	0.474	0.0148	0.01	0.01	0.04
	0.3848	0.4084	0.0236	0.0097	0.0098	0.039
	0.3492	0.3612	0.012	0.0095	0.0096	0.0382
	0.3436	0.3336	0.01	0.0095	0.0094	0.0378
	0.3196	0.32	0.0004	0.0093	0.0093	0.0372
13	0.2644	0.348	0.0836	0.0088	0.0095	0.0366
	0.1216	0.1456	0.024	0.0065	0.0071	0.0272
	0.0556	0.0628	0.0072	0.0046	0.0049	0.019
	0.0392	0.0332	0.006	0.0039	0.0036	0.015
	0.0228	0.018	0.0048	0.003	0.0027	0.0114
	0.014	0.0124	0.0016	0.0023	0.0022	0.009
	0.0116	0.0116	0	0.0021	0.0021	0.0084

14	0.7724	0.8064	0.034	0.0084	0.0079	0.0326
	0.6236	0.6456	0.022	0.0097	0.0096	0.0386
	0.5332	0.5624	0.0292	0.01	0.0099	0.0398
	0.4572	0.4644	0.0072	0.01	0.01	0.04
	0.4128	0.4192	0.0064	0.0098	0.0099	0.0394
	0.366	0.3604	0.0056	0.0096	0.0096	0.0384
	0.3524	0.3488	0.0036	0.0096	0.0095	0.0382
15	0.922	0.944	0.022	0.0054	0.0046	0.02
	0.854	0.8764	0.0224	0.0071	0.0066	0.0274
	0.8416	0.8444	0.0028	0.0073	0.0072	0.029
	0.7928	0.8152	0.0224	0.0081	0.0078	0.0318
	0.76	0.7772	0.0172	0.0085	0.0083	0.0336
	0.7432	0.7512	0.008	0.0087	0.0086	0.0346
	0.7216	0.7352	0.0136	0.009	0.0088	0.0356
16	0	0	0	0	0	0
	0	0	0	0	0	0
	0	0	0	0	0	0
	0	0	0	0	0	0
	0	0	0	0	0	0
	0	0	0	0	0	0
	0	0	0	0	0	0
17	0.0316	0.0516	0.02	0.0035	0.0044	0.0158
	0.004	0.0092	0.0052	0.0013	0.0019	0.0064
	0.0008	0.0028	0.002	0.0006	0.0011	0.0034
	0.0004	0	0.0004	0.0004	0	0.0008
	0.0004	0	0.0004	0.0004	0	0.0008
	0	0	0	0	0	0
	0	0	0	0	0	0
18	0.7032	0.7436	0.0404	0.0091	0.0087	0.0356
	0.5388	0.582	0.0432	0.01	0.0099	0.0398
	0.446	0.4536	0.0076	0.0099	0.01	0.0398
	0.3964	0.3872	0.0092	0.0098	0.0097	0.039
	0.3524	0.3352	0.0172	0.0096	0.0094	0.038
	0.3272	0.3148	0.0124	0.0094	0.0093	0.0374
	0.3192	0.3008	0.0184	0.0093	0.0092	0.037

## LIST OF REFERENCES

- Brillman, J.C., T. Burr, D. Forslund, E. Joyce, R. Picard, and E. Umland, "Modeling Emergency Department Visit Patterns for Infectious Disease Complaints: Results and Application to Disease Surveillance," *BMC Medical Informatics and Decision Making*, 5 (2005).
- Burkom, H.S., S.P. Murphy, and G. Shmueli, "Automated Time Series Forecasting for Biosurveillance" (draft), in submission to *Statistics in Medicine* (2006). Accessible on-line at <http://www.isdsjournal.org/article/viewFile/806/693> on June 5, 2007.
- Crosier, R.B., "Multivariate Generalizations of Cumulative Sum Quality Control Schemes", *Technometrics*, 30 (1988): 291-303.
- Farrington, C.P., N.J. Andrews, A.D. Beale, and M.A. Catchpole, "A Statistical Algorithm for the Early Detection of Outbreaks of Infectious Disease," *Journal of the Royal Statistical Society, Series A (Statistics in Society)*, **159** (1996): 547-563.
- Fricker, Jr. R.D., "Directionally Sensitive Multivariate Statistical Process Control Methods with Application to Syndromic Surveillance," *Advances in Disease Surveillance*, 3:1 (2007). Accessible on-line at [www.isdsjournal.org](http://www.isdsjournal.org) on June 5, 2007.
- Fricker, Jr. R.D., "Syndromic Surveillance," *Encyclopedia for Quantitative Risk Assessment* (to appear), John Wiley & Sons, (2006).
- Fricker, Jr. R.D., and H. Rolka, "Protecting Against Biological Terrorism: Statistical Issues in Electronic Biosurveillance," *Chance*, 91 (2006): 4-13.
- Goldenberg, A., G. Shmueli, R.A. Caruana, and S.E. Fienberg, "Early Statistical detection of Anthrax Outbreaks by Tracking Over-the-counter Medication Sales," *Proceeding of the National Academy of Sciences*, 99, (2002): 5237-5240.
- Dunfee, D.A. and B.L. Hegler, "Biological Terrorism Preparedness: Evaluating the Performance of the Early Aberration Reporting System (EARS) Syndromic Surveillance Algorithms," Thesis, Naval Postgraduate School, Monterey, CA (2007).
- Healy, J.D., "A Note on Multivariate CUSUM Procedures," *Technometrics*, **29**, (1987): 409--412.
- Hotelling, H., "Multivariate Quality Control--Illustrated by the Air Testing of Sample Bombsights," in *Techniques of Statistical Analysis*, eds. C. Eisenhart, M.W. Hastay, and W.A. Wallis, McGraw-Hill, New York, (1947): 409-412.

- Hutwagner, L., W. Thompson, G.M. Seeman, and T. Treadwell, "The Bioterrorism Preparedness and Response Early Aberration Reporting System (EARS)," *Journal of Urban Health: Bulletin of the New York Academy of Medicine*, **80** (2003): 89i-96i.
- Joner, Jr., M.D., W.H. Woodall, M.R. Reynolds, Jr., and R.D. Fricker, Jr., "The Use of Multivariate Control Charts in Public Health Surveillance," draft dated 2006.
- Lotze, T., S.P. Murphy, and G. Shmueli, "Preparing Biosurveillance Data for Classic Monitoring" draft in submission to *Advances in Disease Surveillance* (2006).
- Lowry, C.A., W.H. Woodall, C.W. Champ, and S.E. Rigdon, "A Multivariate Exponentially Weighted Moving Average Control Chart," *Technometrics*, **34** (1992): 46-53.
- Montgomery, D.C., *Introduction to Statistical Quality Control*, New York: John Wiley & Sons, 2001, 4th edition.
- Reis, B. Y., M. Pagano, and K.D. Mandl, "Using Temporal Context to Improve Biosurveillance," *Proceedings of the National Academy of Sciences*, 100 (2003): 1961-1965.
- Shmueli, G., "Statistical Challenges in Modern Biosurveillance," in submission to *Technometrics*, draft dated September 18, 2006.
- Shmueli, G., and S. Fienberg, "Current and Potential Statistical Methods for Monitoring Multiple Data Streams for Bio-Surveillance," in *Statistical Methods in Counterterrorism: Game Theory, Modeling, Syndromic Surveillance, and Biometric Authentication*, A. Wilson, G. Wilson, and D. Olwell, eds., New York: Springer, 2006.
- Stoto, M.A., R.D. Fricker, Jr., A. Jain, A. Diamond, J.O. Davies-Cole, C. Glymph, G. Kidane, G. Lum, L. Jones, K. Dehan, and C. Yuan. "Evaluating Statistical Methods for Syndromic Surveillance," *Statistical Methods in Counterterrorism: Game Theory, Modeling, Syndromic Surveillance, and Biometric Authentication*, New York: Springer, 2006.
- Woodall, W.H., and M.M. Ncube, "Multivariate CUSUM Quality Control Procedures," *Technometrics*, 27 (1985): 46-53.
- Woodall, W.H., "The Use of Control Charts in Health-Care and Public-Health Surveillance," *Journal of Quality Technology*, 38 (2006): 1-16.

## INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center  
Ft. Belvoir, VA
2. Dudley Knox Library  
Naval Postgraduate School  
Monterey, CA
3. Associate Professor R.D. Fricker, Jr.  
Naval Postgraduate School  
Monterey, CA
4. Chairman David Olwell  
Naval Postgraduate School  
Monterey, CA
5. Professor Annette Neu  
Naval Postgraduate School  
Monterey, CA
6. Professor Ellen Gordon  
Naval Postgraduate School  
Monterey, CA
7. Dr. Jerome Tokars  
Centers for Disease Control and Prevention  
Atlanta, GA
8. Dr. Henry Rolka  
Centers for Disease Control and Prevention  
Atlanta, GA
9. Ms. Lori Hutwagner  
Centers for Disease Control and Prevention  
Atlanta, GA
10. Ms. Heather Issvoran  
Naval Postgraduate School  
Monterey, CA
11. Professor William Woodall  
Virginia Polytechnic Institute and State University  
Blacksburg, VA