

Performance Evaluation of Bootstrap Multivariate Exponentially-Weighted Moving Average (BMEWMA) Control Chart

O. Ikpotokin^{a*}, C. C. Ishiekwene^b and N. Ekhosuehi^c

^aDepartment of Mathematics and Statistics, Ambrose Alli University, Ekpoma, Edo State Nigeria; ^{b,c}Department of Mathematics, University of Benin, Benin City, Nigeria

A fundamental hypothesis in theoretical statistical quality control is that samples are independently and identically distributed; but this assumption is frequently violated in many production processes. Moreover, the presence of autocorrelated data in many process control applications gravely affects the performance of classical control charts if not appropriately accounted for. In this paper, bootstrap T^2 and bootstrap multivariate exponential weighted moving average (BMEWMA) control charts are proposed for monitoring and controlling multivariate autocorrelated processes. From numerical illustration, results obtained from the Average Run Length (ARL), standard deviation run length (SDRL), median run length (MRL) and percentiles run length (PRL) displayed in tabular and graphical forms, shows that the proposed bootstrap control methods performed better than the F -distribution T^2 control method.

Keywords: Run lengths; autoregressive model; bootstrap; mean vector; cross-covariance; autocorrelation.

1. Introduction

There are many situations in which it is necessary to monitor two or more related quality characteristics simultaneously. In such cases, multivariate statistical process monitoring procedures should be considered since there may be some relationship between the quality characteristics. The use of information from multiple variables may provide a better, more accurate, monitoring strategy compared to the used of individual variables that may inflate the overall false alarm rate (probability type I error) thereby resulting to an incorrect determination of control limits. The first solution to this problem was T^2 statistic suggested by Hotelling (1947) for monitoring the mean vector of multivariate processes. However, T^2 control chart is good in detecting large shift in process mean vector, Hwang (2008). The multivariate exponentially weighted moving average (MEWMA) was introduced by Lowry et al. (1992) as an extension of the univariate exponentially weighted moving average (EWMA). The primary goal of the MEWMA is to quickly detect small changes in a process more rapidly than other multivariate control charts based on the fact that the charting scheme takes advantage of the knowledge from previous observations in any given process. In other words, the MEWMA control chart is good at detecting small to moderate shift in the mean vector of a process.

The study of optimal design of MEWMA charts using the average run length and the median run length was carried out by Lee and Khoo (2006). Champ and Jones-Farmer (2007) studied the properties of the MEWMA control chart when parameters are estimated. Joner Jr. et al. (2008) developed a one-sided MEWMA control chart for health surveillance. Mahmoud and Zahran (2010) investigate the performance of the MEWMA chart with some different recommended values of smoothing parameters of λ when the in control parameters are estimated. A modified MEWMA control scheme for an analytical process data was introduced by Patel and Divecha (2013) with a view for detecting shifts

*Corresponding author. Email: ikpotokinosayomore@yahoo.co.uk

of all kinds in case of highly first order vector autoregressive VAR (1) process. Haq et al. (2015) developed a new maximum exponentially weighted moving average control chart for monitoring process mean and dispersion. Lu (2015) proposed an Extended Nonparametric Exponentially Weighted Moving Average Sign Control Chart. Brown and Moltchanova (2015) introduced New Exponentially Weighted Moving Average Control Charts for Monitoring Process Dispersion. Rabyk and Schmid (2016) studied the EWMA control charts for detecting changes in the mean of a long-memory process.

Generally, most parametric multivariate control charts has the advantage of being able to monitor multiple quality characteristics simultaneously for both changes in the mean vector and correlation structure while maintaining a specified probability similar to type I error (α). However, the problem involve in the use of parametric multivariate control chart is the problem of violation of the assumption of multivariate normality that is required for many charts. This study shall consider the bootstrap multivariate exponentially weighted moving average control charts in order to overcome the problem of violating the assumption of multivariate normality as well as detecting small shift in the process. The remainder of the paper is organized as follows: Section 2 introduces methods and the proposed bootstrap ARL algorithms in obtaining Hotelling's T^2 and multivariate exponentially weighted moving average considered in this study. Section 3 is devoted to the empirical study of the efficiency of the proposed methods, discussion and interpretations of results, while Section 4 is on the conclusions.

2. Multivariate Exponentially-Weighted Moving Average (MEWMA) Control Chart

Suppose $X = (X_1, X_2, \dots, X_d)$ be d -dimensional quality characteristics obtained from a process of interest. Assuming that the process is in control and is d -dimensional normal distribution with mean vector μ_0 and variancecovariance matrix Σ_0 , i.e., $X \sim N_d(\mu_0, \Sigma_0)$, where μ_0 and Σ_0 are unknown. But if μ_0 and Σ_0 can be estimated from a set of K training samples each with size n , and the process was in control when these K training samples were taken. A multivariate EWMA control chart is proposed by Lowry et al. (1992) as follows:

$$Z_i = \Lambda X_i + (1 - \Lambda)Z_{i-1} \quad (1)$$

where Λ is the drag $(\lambda_1, \lambda_2, \dots, \lambda_n = \lambda)$, $0 \leq \lambda_i \leq 1$ for $i = 1, 2, \dots, d$. If there is no a-priori reason to weight past observations differently for the d -quality characteristics being monitored, then $\lambda_1 = \lambda_2 = \dots = \lambda_n = \lambda$. The initial value Z_0 is usually obtained equal to the in-control mean vector of the process. The multivariate EWMA control chart is equivalent to the T^2 -Chart and is denoted as:

$$T_i^2 = Z_i' \Sigma_{Z_i}^{-1} Z_i > h, i = 1, 2, \dots \quad (2)$$

where Σ_{Z_i} is the variance-covariance matrix of Z_i . The value h is obtained via simulation to achieve a specified in-control ARL. The ARL performance of the MEWMA control chart depends only on the non-centrality parameter. This means that the MEWMA has the property of directional invariance. The variance-covariance matrix of Z_i is estimated as:

$$\Sigma_{Z_i} = \frac{\lambda}{2 - \lambda} [1 - (1 - \lambda)^{2i}] \Sigma \quad (3)$$

An approximation of the variance-covariance matrix Σ_{Z_i} when approaches infinity, is expressed as:

$$\Sigma_{Z_i} = \frac{\lambda}{2 - \lambda} \Sigma \quad (4)$$

From the MEWMA vector in Equation (3), Z_i is expanded recursively to obtain:

$$Z_i = \lambda X_i + \lambda(1 - \lambda)X_{i-1} + \lambda(1 - \lambda)^2 X_{i-2} + \cdots + \lambda(1 - \lambda)^{i-1} X_1 + (1 - \lambda)^i Z_0 \quad (5)$$

When μ_0 and Σ_0 are unknown, then K in-control samples of size n each are used to estimate the parameters. The in-control process mean vector is estimated by:

$$\bar{\bar{X}} = \sum_{i=1}^k \frac{\bar{X}_i}{k} \text{ and } \bar{S} = \sum_{i=1}^k \frac{S_i}{k} \quad (6)$$

where

$$\bar{X}_i = \sum_{h=1}^n \frac{X_{ih}}{n} \text{ and } S_i = \sum_{h=1}^n \frac{(X_{ih} - \bar{X}_i)(X_{ih} - \bar{X}_i)'}{n} \quad (7)$$

are the sample mean vector and sample variance-covariance matrix of the i th training sample, $i = 1, 2, \dots, k$ respectively. The charting constant (λ) may be chosen in a way that similar average run lengths are achieved under a wide range of distributions. The values acceptable for the charting constant are often very small ($0 < \lambda \leq 1$), which means putting the majority of the weight on the past observations instead of the most current, Sullivan and Stoumbos (2001), Stoumbos and Sullivan (2002), Woodall and Mahmoud (2005).

The common assumption about any given sets of observations is that they are independently and identically distributed (i.i.d.) over time. However, this assumption may not hold in many of today's applications. For example, the quality characteristics being monitored may be correlated with itself over time; this may introduce positive or negative autocorrelation that can significantly affect the performance of control chart procedure. This autocorrelation (also known as serial correlation) should be considered while monitoring the quality characteristic(s) since it has been shown both in the univariate and the multivariate cases that failure to account for such autocorrelation may lead to too many false alarms, Psarakis and Papaleonida (2007), Montgomery (2009). Positive autocorrelation (e.g. low values tend to be followed by other low values, or high values tend to follow other high values) can possibly lead to control limits that maybe too narrow, this may increase the frequency of false alarms, Lee and Jun (2012). On the other hand, negative autocorrelation (low values tend to be followed by other high values, or high values tend to follow other low values) may lead to may leads to control limits that are too wide. Hence special causes of variation that may be present in the process could be missed or not identified, Jarrett and Pan (2007), Franco et al. (2014), Leoni et al. (2015).

The usual multivariate control charts (for i.i.d. data) may not be appropriate for monitoring a multivariate autocorrelated data. One approach would be to widen the control limits to account for the autocorrelation. A multivariate model, say the vector autoregressive (VAR) model, can be fitted to the quality characteristics to obtain the residuals to be monitored. Fitting a VAR(1) model (a VAR model of lag 1) to the multivariate autocorrelated data and the residuals monitored using Hotelling's T^2 chart is an approach suggested by Jarrett and Pan (2007) and Pan and Jarrett (2007). Also the use of control chart that applies a neural network, called the Neural Network Identifier was created by Hwang (2004) and Hwang (2008). The chart not only detects the multivariate signal but

determines which variable or variables are at fault. It, however, requires a large amount of data and time to use properly.

Phaladiganon et al. (2011) introduced the percentile bootstrap method as a means of obtaining Hotelling's T^2 control limits assuming that the distribution is not multivariate normal. However, Phaladiganon et al. (2011) method bootstrapped from Hotelling's T^2 statistic obtained by collapsing the multivariate data into univariate, and this will result to control limits that is good in detecting of large shift only. Also, Adewara and Adekeye (2012) introduced the idea of bootstrapping method to tackle the problem of quality characteristics that are not correlated as well as the minimum maximum control methods. Application of the minimax control chart by way of chi square control method for multivariate manufacturing process was also introduced by Balali (2013). Kalgonda (2013) introduced the use of balance bootstrap percentile method to estimate critical value and control limits for autocorrelated processes. Gandy and Kvaloy (2013) proposed the use of definite restricted bootstrap control charts methods for performance evaluation. The block bootstrap control limits for multivariate autocorrelated process was proposed by Kalgonda (2015) having it view around the control procedure based on Z-statistics. In this study, we shall consider the bootstrap multivariate exponentially weighted moving average control charts in order to overcome the problem of violating the assumption of multivariate normality as well as detecting small shift in the process.

2.1 *Bootstrap Hotelling's T^2 and Bootstrap Multivariate Exponentially -Weighted Moving Average (BMEWMA) methods*

The multivariate T^2 (F-distribution) chart is one of the charts used by Hwang (2004) in comparing their neural-network-based identifier. Traditionally, multivariate quality control methods are proposed under the assumption that observations are normally distributed with the mean vector μ and covariance matrix Σ . The process observation vector Y_i at time t , can be denoted as

$$\mathbf{Y}_t = \boldsymbol{\mu} + \boldsymbol{\epsilon}_t, t = 1, 2, \dots \quad (8)$$

where $\boldsymbol{\epsilon}_t$ is a multivariate normal random vector with the mean vector of zeros and covariance matrix Σ . To ascertain whether the process mean vector is in control when the process covariance matrix Σ is known, the Shewhart control chart with upper control limit $UCL = \chi_{p,\alpha}^2$ is given by the statistic:

$$\chi_t^2 = (\mathbf{Y}_t - \boldsymbol{\mu}_0)' \Sigma^{-1} (\mathbf{Y}_t - \boldsymbol{\mu}_0) \quad (9)$$

where $\boldsymbol{\mu}_0$ is the interest value of the mean vector. The usual practice is that when the underlying assumption is violated, there will be an increase in the rate of false alarms. In some cases where the process observations are either positive or negative auto-correlated, the performance of the control charts maybe seriously affected, thereby leading to unnecessary correlation of the process. To overcome this problem, vector autoregressive model of lag1 denoted by VAR (1) is denoted by:

$$\mathbf{Y}_t = \boldsymbol{\mu}_t + \boldsymbol{\varphi}(\mathbf{Y}_{t-1} - \boldsymbol{\mu}_t) + \boldsymbol{\epsilon}_t \quad (10)$$

where $\boldsymbol{\mu}_t$ is the vector of mean values at time t , $\boldsymbol{\epsilon}_t$ is a vector of normal random variables with the mean vector of zeros, covariance matrix, sigma (Σ), and a $d \times d$ matrix of autocorrelation parameters ($\boldsymbol{\varphi}$). Suppose \mathbf{Y}_t is taken to be stationary, under this assumption, $\boldsymbol{\mu}_t$ is constant over time, then Equation (10) now takes the form:

$$\mathbf{Y}_t = \boldsymbol{\mu} + \boldsymbol{\varphi}(\mathbf{Y}_{t-1} - \boldsymbol{\mu}) + \boldsymbol{\epsilon}_t \quad (11)$$

Let $\gamma(t, t+h)$ represent the crosscovariance matrix between \mathbf{Y}_t and \mathbf{Y}_{t+h} , and let its (ij) th element be represented by $\gamma_{ij}(h)$, where

$$\gamma_{ij}(h) = \mathbb{E}\{(\mathbf{Y}_{it} - \boldsymbol{\mu}_{it})(\mathbf{Y}_{jt+h} - \boldsymbol{\mu}_{jt+h})\} \tag{12}$$

As a result of stationary assumption, $\boldsymbol{\mu}_t$ shall be constant of $\boldsymbol{\mu}$, while $\gamma(t, t+h)$ shall be a function of the *lagh* only and may be written as $\gamma(h)$. The crosscorrelation matrix $\boldsymbol{\rho}(h)$ at *lagh*, is denoted by:

$$\boldsymbol{\rho}(h) = V^{-1/2}\gamma(h)V^{1/2} \tag{13}$$

where

$$V = \text{diag}(\gamma_{11}(0), \gamma_{22}(0), \dots, \gamma_{dd}(0)). \tag{14}$$

Applying the Yule Walker relationship for covariance matrices of VAR (1) processes, the crosscovariance matrix $\gamma(\mathbf{0})$ at *lag0* is obtained as:

$$\gamma(\mathbf{0}) = \boldsymbol{\varphi}\gamma(\mathbf{0})\boldsymbol{\varphi}' + \boldsymbol{\Sigma} \tag{15}$$

To overcome the problem of violating multivariate normal assumption of observation vectors as well as autocorrelation of sample observations, the bootstrap methods and vector autoregressive model of lag 1 denoted by VAR (1) is proposed as shown in the algorithm.

ALGORITHM: Proposed bootstrap ARL procedures for bivariate case

Supposed a set of observations say $x_1 = x_{11}, x_{12}, \dots, x_{1n}$ and $x_2 = x_{21}, x_{22}, \dots, x_{2n}$ are given;

- (1) Combine the sample sizes of x_1 and x_2 of the sets of observation say, $x = (x_{11}, x_{12}, \dots, x_{1n}, x_{21}, x_{22}, \dots, x_{2n})$.
- (2) Draw a bootstrap sample with replacement from Step (1) to obtain $x^* = x_{11}^*, x_{12}^*, \dots, x_{1n}^*, x_{21}^*, x_{22}^*, \dots, x_{2n}^*$
- (3) Repeat Step (2) a large number of times to obtain bootstrap replications $x^* = x_{11}^{*(i)}, x_{12}^{*(i)}, \dots, x_{1n}^{*(i)}, x_{21}^{*(i)}, x_{22}^{*(i)}, \dots, x_{2n}^{*(i)}$, where $(i^* = 1, 2, \dots, B)$. In general B is a large number (e.g., $B = 3000$).
- (4) Obtain the value of sigma($\boldsymbol{\Sigma}$), autocorrelation parameters($\boldsymbol{\varphi}$), and cross-covariance matrices $\gamma(\mathbf{0})$ as described in Equations (11) and (15).
- (5) Simulate data 10,000 times from the VAR(1) models with the proposed matrices of the parameters ($\boldsymbol{\Sigma}$), ($\boldsymbol{\varphi}$), and $\gamma(\mathbf{0})$ in Step (4).
- (6) Set up a bootstrap T^2 and BMEWMA control limit for the chart that will gives the desire in-control average run length (ARL) of 200 when both the mean shift are zeros and take the values of ($\boldsymbol{\Sigma}$), ($\boldsymbol{\varphi}$), and $\gamma(\mathbf{0})$ as stated in Step (4).
- (7) Impute the control limit obtained and generate out of control ARL for the remaining mean shifts chosen.

The performance of the bootstrap T^2 and BMEWMA control charts to detect mean shift shall be compare with existing multivariate T^2 (F-distribution) charts.

2.2 Performance evaluation of a control chart-average run length

The expected number of samples taken before the chart signals is called the average run length (ARL). During the in-control period, $ARL = 1/\alpha$ and is called ARL_0 . The risk is the well known as type I error and is denoted by $\alpha = 1/ARL$, Lee and Jun (2010).

Conventionally, the average run length (ARL) serves as a very useful and standard criterion for measuring the performance of a control chart scheme. ARL is the expected number of data points collected before an out-of-control situation is signaled. When there is no shift in both variables, these kinds of processes are deemed in-control. For in-control processes, the ideal performance of control schemes should be that the control schemes cannot find any shift, Lee and Jun (2012). However, this is impossible in reality since the type I error exists. The probability of type I error is defined as the probability that a control scheme detects a shift when no shift happens in the process. A good control scheme should have small probability of type I error. In-control ARL is related to the measure of the probability of type I error. The smaller the probability of type I error is, the longer the in-control ARL; in other words, a good control scheme should have long in-control ARL.

When shift happens on any of the process variables, the process is regarded as an out-of-control situation. When a process is out-of-control, there is a probability that the control scheme deems it as in-control; this is defined as the probability of type II error. A good monitoring scheme should have small probability of type II error. The out-of-control ARL is related to the measure of the probability of type II error. The smaller the probability of type II error is, the shorter the out-of-control ARL is. And the shorter the out-of-control ARL is, the better the control scheme is. In general, a good control scheme should have long in-control ARL and short out-of-control ARL. The performance of a control chart is typically measured in terms of the ARL and SDRL. The ARL is the average number of sample points that is plotted on a chart before the first out-of-control signal is detected whereas the SDRL measures the spread of the run length distribution. When a process is out-of-control, it is desirable to have small values of ARL and SDRL.

However, the median run length (MRL) measure provides a more meaningful explanation on the in-control and out-of-control performances of the charts as in-control run length distribution based on the ARL that is highly skewed. Moreover, the MRL profile is also more readily understood by the practitioners compared to the ARL profile. The Percentiles of Run Length (PRL) are 99 points which divide an array or a distribution into 100 equal parts. They are denoted by P_1, P_2, \dots, P_{99} . For instance, the 25th and 75th percentiles of a distribution will be the values of the (25/100)th term and (75/100)th term along the distribution respectively. Thus, the percentile of run length in this work shall be based on the average of each of the percentile run length simulated 10000 times in R programming language. The percentile is an informative and robust chart performance measure. The entire run length distribution provides useful information about the performance of the chart and a number of selected percentiles should help summarize this information. In this work, bivariate auto-correlated and cross covariance matrices processes are considered. The control limits of the charts will be simulated so that each chart has the same in-control ARL of 200 approximately. In order to tune the in-control ARL to this desired value, several computer programs were written in R language to analyze the data output.

3. Empirical Study

The data used in this section is from Montgomery (2002) on a manufacturing practice amid two variables, x_1 and x_2 , as shown in Table 1.

Table 1. Two quality characteristics from a production process

| Sample | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|--------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| x_1 | 58 | 60 | 50 | 54 | 63 | 53 | 42 | 55 | 46 | 50 | 49 | 57 | 58 | 75 | 55 |
| x_2 | 32 | 33 | 27 | 31 | 38 | 30 | 20 | 31 | 25 | 29 | 27 | 30 | 33 | 45 | 27 |

The aim is to set up a bootstrap T^2 and BMEWMA control limits for monitoring out of control signals and measuring the performance of a control chart scheme adopting the Average Run Length (ARL), Standard Deviation Run Length (SDRL), Median Run Length

(MRL) and Percentiles Run Length (PRL). The results shall be compared with the F-distribution T^2 .

3.1 Presentation of results from simulation studies

Adopting Steps 1–3 from the proposed bootstrap procedures, bootstrap samples were replicated 3000 times to obtain bootstrap data equivalent to the given data in Table 1. Using Step (4) of the proposed bootstrap algorithm, both the bootstrap and original data were imported to R programming language and results for the parameters are presented as:

$$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \Sigma = \begin{bmatrix} 1 & 0.0226 \\ 0.0226 & 1 \end{bmatrix}, \varphi = \begin{bmatrix} 0.0644 & 0.1482 \\ 0.1322 & 0.0644 \end{bmatrix}, \gamma(0) = \begin{bmatrix} 3.9767 & 0.0649 \\ 0.0649 & 2.0603 \end{bmatrix}$$

for the bootstrap parameters and

$$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \Sigma = \begin{bmatrix} 1 & 0.9806 \\ 0.9806 & 1 \end{bmatrix}, \varphi = \begin{bmatrix} 0.0644 & 0.1482 \\ 0.1322 & 0.0644 \end{bmatrix}, \gamma(0) = \begin{bmatrix} 3.9767 & 0.0649 \\ 0.0649 & 2.0603 \end{bmatrix}$$

for the F-distribution parameters. Implementing the Simulation Code in the Appendix, simulation studies conducted 10,000 times from VAR(1) shows that the control limits (5.8801, 4.3135 and 4.0512) obtained adopting Step 7 from the proposed bootstrap algorithm produces an in-control ARL of 200.3049, 200.0333 and 200.3388 for F-distribution T^2 , bootstrap T^2 and BMEWMA respectively. This is shown in the ARL column of Table 2 for each of the methods when there is 0 or no shift in the process.

Table 2: Simulation results: ARL, SDRL and MRL from the three methods

| Shift | ARL | | | SDRL | | | MRL | | |
|-------|----------------|-----------------|---------|----------------|-----------------|---------|----------------|-----------------|--------|
| | F-Distrn T^2 | Bootstrap T^2 | BMEWMA | F-Distrn T^2 | Bootstrap T^2 | BMEWMA | F-Distrn T^2 | Bootstrap T^2 | BMEWMA |
| 0 | 200.3049 | 200.0333 | 200.339 | 202.459 | 194.99 | 197.286 | 137 | 141 | 140 |
| 0.5 | 174.2407 | 169.8069 | 52.5002 | 178.3 | 170.97 | 42.7994 | 118 | 117 | 41 |
| 1 | 119.11 | 101.5343 | 15.4441 | 119.544 | 100.34 | 8.4408 | 81 | 71 | 14 |
| 1.5 | 73.1259 | 49.7421 | 8.3958 | 73.3597 | 49.424 | 3.4089 | 50 | 34 | 8 |
| 2 | 42.5684 | 21.7913 | 5.8036 | 42.6046 | 21.299 | 1.8907 | 29 | 15 | 6 |
| 2.5 | 23.8237 | 9.9296 | 4.5007 | 23.0522 | 9.4776 | 1.1951 | 16 | 7 | 4 |
| 3 | 13.8727 | 5.1004 | 3.7768 | 13.5441 | 4.5655 | 0.8647 | 10 | 4 | 4 |
| 3.5 | 8.0663 | 2.8995 | 3.3162 | 7.7919 | 2.3611 | 0.6461 | 6 | 2 | 3 |
| 4 | 4.7772 | 1.8868 | 3.013 | 4.3705 | 1.3232 | 0.5335 | 3 | 1 | 3 |
| 5 | 1.8783 | 1.1715 | 2.5942 | 1.3884 | 0.4683 | 0.5069 | 1 | 1 | 3 |

Table 3: Simulation results: percentiles run length from the three methods

| Shift | 25 th Percentile | | | 75 th Percentile | | | 95 th Percentile | | |
|-------|-----------------------------|-----------------|--------|-----------------------------|-----------------|--------|-----------------------------|-----------------|--------|
| | F-Distrn T^2 | Bootstrap T^2 | BMEWMA | F-Distrn T^2 | Bootstrap T^2 | BMEWMA | F-Distrn T^2 | Bootstrap T^2 | BMEWMA |
| 0 | 58 | 59 | 60 | 278 | 278 | 275 | 598 | 590 | 598 |
| 0.5 | 50 | 49 | 22 | 238 | 234 | 69 | 531 | 514 | 137 |
| 1 | 35 | 30 | 9 | 164 | 141 | 20 | 367 | 303 | 32 |
| 1.5 | 21 | 15 | 6 | 100 | 69 | 10 | 220 | 148 | 15 |
| 2 | 13 | 7 | 4 | 59 | 30 | 7 | 126 | 64 | 9 |
| 2.5 | 7 | 3 | 4 | 33 | 14 | 5 | 70 | 28 | 7 |
| 3 | 4 | 2 | 3 | 19 | 7 | 4 | 40 | 14 | 5 |
| 3.5 | 3 | 1 | 3 | 11 | 4 | 4 | 24 | 8 | 4 |
| 4 | 2 | 1 | 3 | 6 | 2 | 3 | 13 | 5 | 4 |
| 5 | 1 | 1 | 2 | 2 | 1 | 3 | 5 | 2 | 3 |

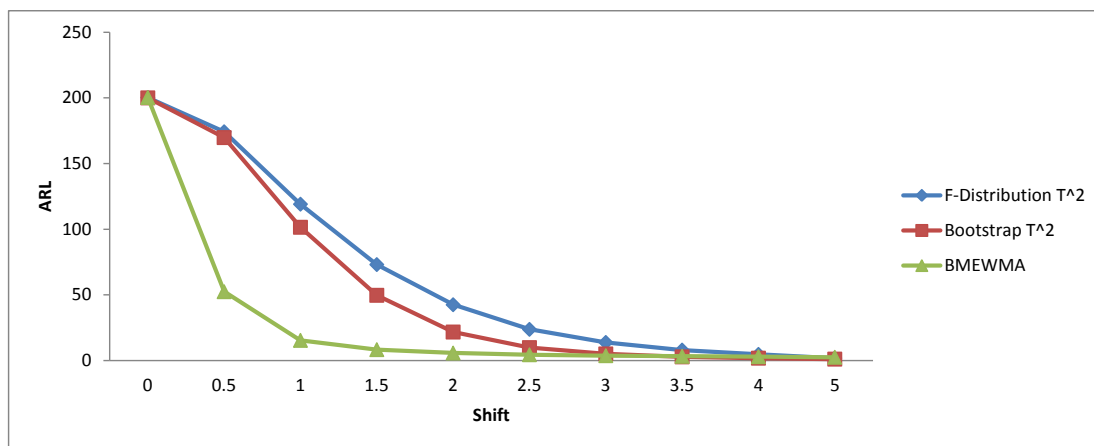


Figure 1: ARL for the three Methods Compared when CL = 5.8801, 4.3135, and 4.0512

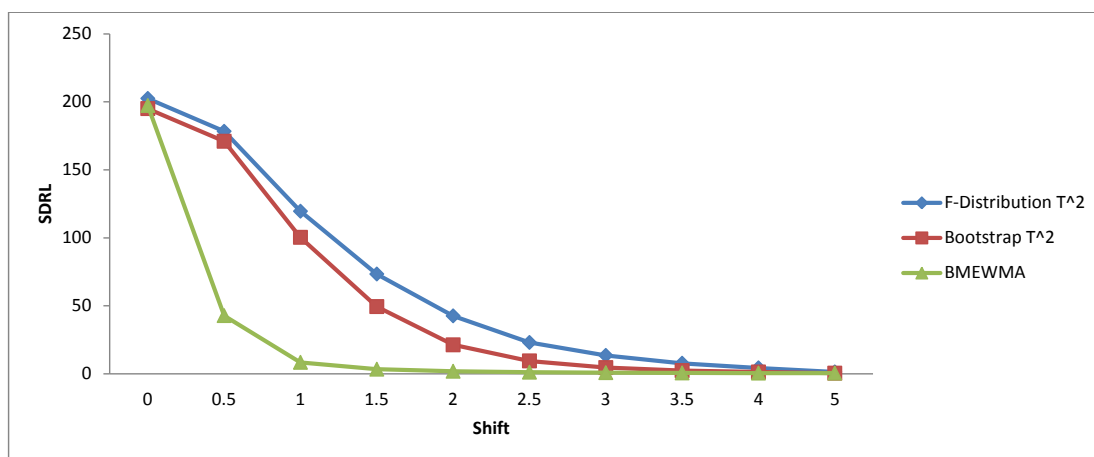


Figure 2: SDRL for the three Methods Compared when CL = 5.8801, 4.3135, and 4.0512

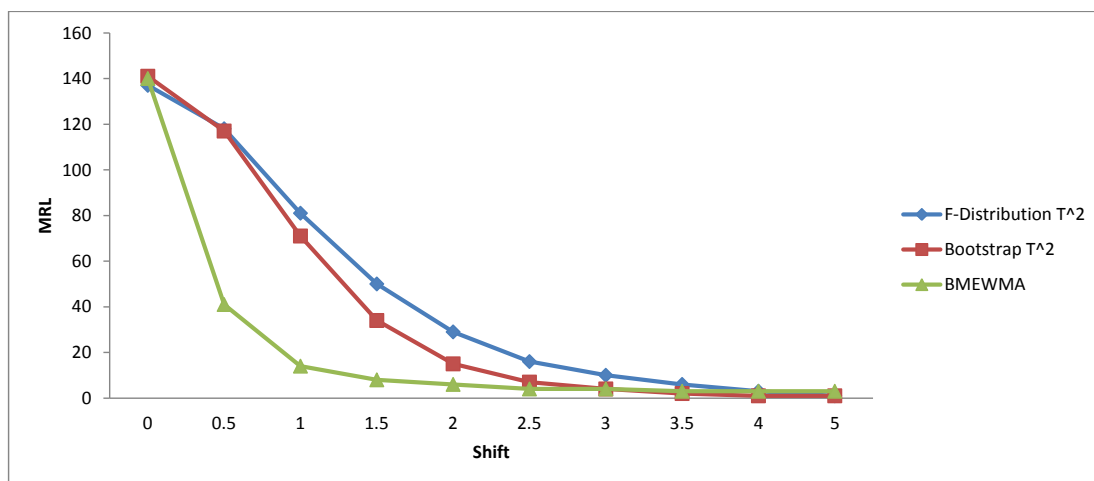


Figure 3: MRL for the three Methods Compared when CL = 5.8801, 4.3135, and 4.0512

The disparity amid the highest and smallest eigenvalue of the autocorrelation matrix is shown to be less than 1 in absolute value, meaning that the process is stable. Results of the simulation studies from the bootstrap T^2 , BMEWMA and F-distribution T^2 are summarized in Tables 2 and 3. Results in Table 2 and 3 were simulated 10000 times for each method by imputing their various parameters (see the last part of every program in

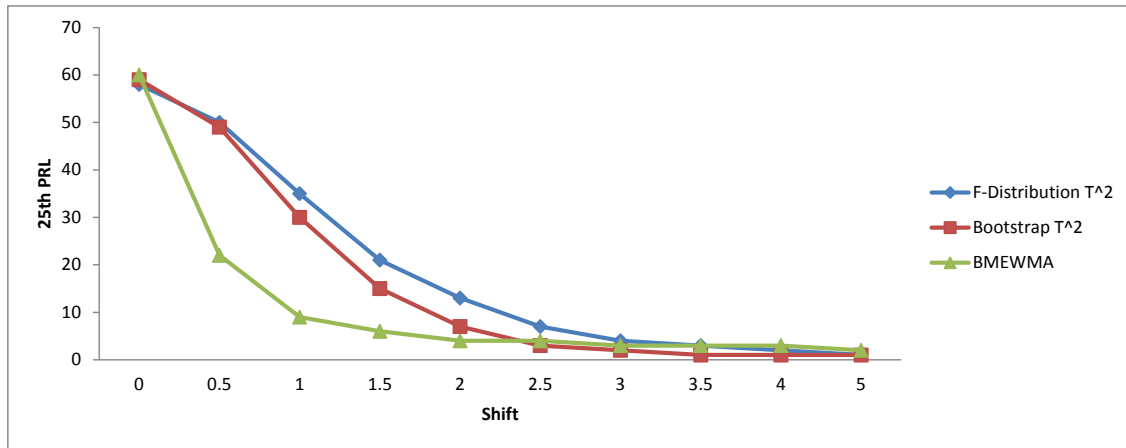


Figure 4: 25th PRL for the three Methods Compared when CL = 5.8801, 4.3135, and 4.0512

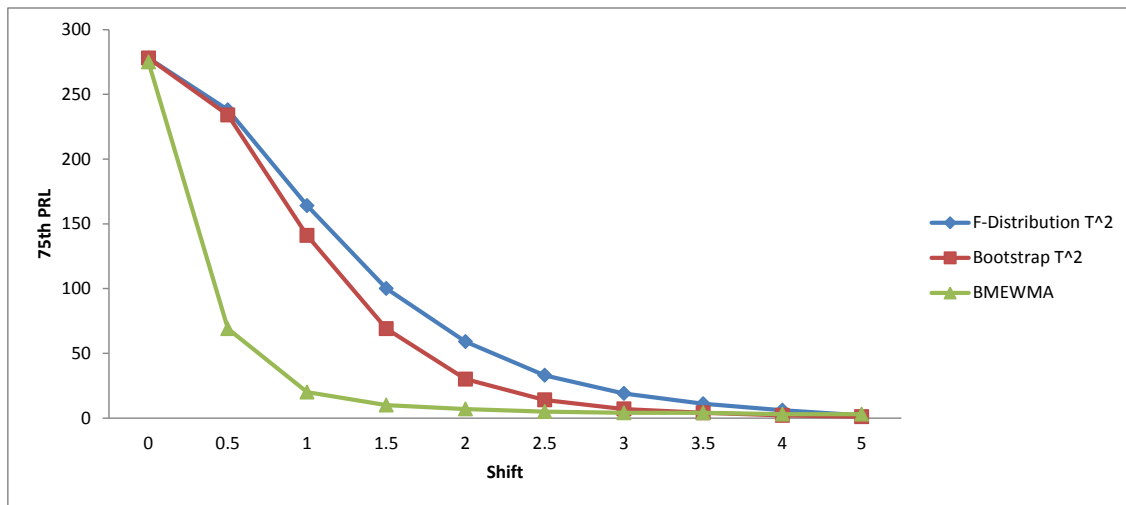


Figure 5: 75th PRL for the three Methods Compared when CL = 5.8801, 4.3135, and 4.0512

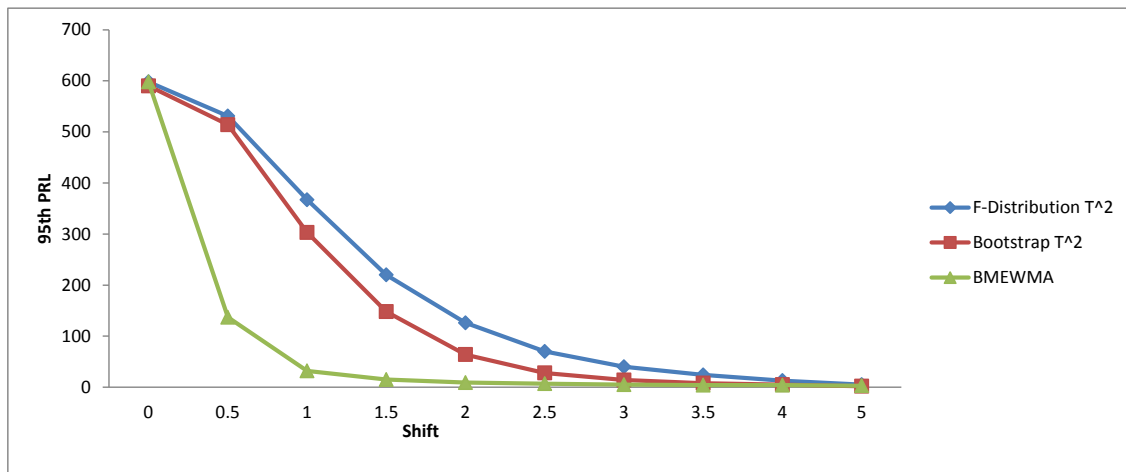


Figure 6: 95th PRL for the three Methods Compared when CL = 5.8801, 4.3135, and 4.0512

r-language from the appendix). ARL, SDRL, MRL and PRL were obtained by taking the mean/average, standard deviation, median and percentiles of the run lengths respectively as the shift in the process changes. Figures 1– 6 shows the results obtained when ARL, SDRL, MRL and PRL are compared between control charts.

3.2 Discussion and interpretation of results

Result of eigenvalues from the autocorrelation matrix is less than 1 in absolute value, meaning that the process is stationary. Tables 2 and 3 summarizes the average run length (ARL), standard deviation run length (SDRL), median run length (MRL) and percentiles run length (PRL) from the three methods. From Tables 2 and 3, it is clear that the proposed bootstrap T^2 and BMEWMA control methods are talented in identifying both large and little shifts in a given process effectively. Various shift sizes (0, 0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4, 5) where adopted for practical reasons.

The first rows in Tables 2 with zero shift in the variables is said to indicate the in-control ARL. This value is expected to be large with respect to the specified ARL 200 for any of the compared method to be adjudged high performance. Looking at the values for the respective in - control ARL, (i.e. 200.3049, 200.0333 and 200.3388) the conclusion is that no one outperforms the other because they have approximately the same value of 200. All other rows except for the first row in the table explain how fast each method detects out of control situation when there is a shift in the process. The smaller the value the faster and good such method is in detecting shift. Note that SDRL, MRL and Percentiles columns in Table 3 are only meant to substantiate the validity of the ARL column as shown in the various Figures. Figures 1-6 shows the results obtained when the three methods (F-Distribution T^2 , Bootstrap T^2 and BMEWMA) are compared using ARL, SDRL, MRL and PRL between control charts. A critical look at the tables and figures shows that BMEWMA has the ability to detect small shift, followed by Bootstrap T^2 and F-Distribution T^2 with the ability to detect large shift. Also, results from the various graphs shows that the proposed bootstrap control charts outperforms the F-distribution control chart in terms of ARL, SDRL, MRL and PRL.

4. Conclusion

This study critically looked at the performance of the bootstrap Hotelling's T^2 and BMEWMA control charts whether or not the underlying distribution is known, and the assumption of normality is satisfied. Using an empirical data set, the bootstrap results obtained in this study at different mean shift levels has been shown to be better than the existing method when compared.

Generally, out of control ARL, SDRL, MRL and PRL decrease with the increase of the shift magnitude as shown in Tables 2 & 3 and Figures 1-6. The BMEWMA detects a shift at least as quick as both the F-distribution T^2 and the Bootstrap T^2 charts. This is expected since the smoothing parameter (0.1) for the BMEWMA chart was chosen to detect small shifts. Finally, it was shown that the proposed control charts outperforms the F-distribution control chart in terms of ARL, SDRL, MRL and PRL as shown in Tables 2 & 3 and Figures 1 – 6. Therefore, the performance of bootstrap control chart obtained in this study will assist in detecting small shift.

References

- Adewara, J.A. and Adekeye, K.S. (2012). Bootstrap T technique for minimax multivariate control chart. *Open Journal of Statistics*, 2: 469–473.
- Brown, J. and Moltchanova, E. (2015). New exponentially weighted moving average control charts for monitoring process mean and process dispersion. *Quality and Reliability Engineering International*, 31(5): 877–901
- Champ, C.W. and Jones-Farmer, L.A. (2007). Properties of multivariate control

charts with estimated parameters. *Sequential Analysis*, 26: 153–169

Franco, B.C., Celano, G., Castagliola, P., Costa, A.F.B. (2014). Economic design of She-whart control charts for monitoring autocorrelated data with skip sampling strategies. *International Journal of Production Economics*, 151: 121–130. doi:10.1016/j.ijpe.2014.02.008

Gandy, A. and Kvaloy, J. T. (2013). Guaranteed conditional performance of control charts via bootstrap methods. *Scandinavian Journal of Statistics*, 40(4): 647–668.

Haq, A., Brown, J. and Moltchanova, E. (2015). A new maximum exponentially weighted moving average control chart for monitoring process mean and dispersion. *Quality and Reliability Engineering International*, 31(8): 1587–1610.

Hotelling, H. (1947). Multivariate quality control illustrated by the air testing of sample bombsights. In: Eisenhart, C., Hastay, M.W., and Wallis, W.A. (editors.), *Techniques of Statistical Analysis*, McGraw-Hill, New York, pp. 111–184.

Hwang, H. B. (2004). Detecting process mean shift in the presence of autocorrelation: a neural network based monitoring scheme. *International Journal of Production Research*, 42: 573–595.

Hwang, H. B. (2008). Toward identifying the source of mean shifts in multivariate SPC: a neural network approach. *International Journal of Production Research*, 46(20): 5531–5559.

Jarrett, J. E. and Pan, X. (2007). The quality control chart for monitoring multivariate autocorrelated processes. *Computational Statistics and Data Analysis*, 51: 3862–3870.

Joner Jr., M.D., Woodall, W.H., Reynolds Jr., M.R. and Fricker Jr., R.D. (2008). A one sided MEWMA chart for health surveillance. *Quality and Reliability Engineering International*, 24: 503–518.

Kalgonda, A. A. (2013). Bootstrap control limits for multivariate autocorrelated processes. *Journal of Academic and Industrial Research*, 1(12): 700–702.

Kalgonda, A.A. (2015). Block bootstrap control limits for multivariate autocorrelated processes. *Journal of Engineering Computers & Applied Sciences*, 4(4), 97–100.

Lee, S. H. and Jun, C. H. (2010). A new control scheme always better than X-bar chart. *Communications in Statistics-Theory and Methods*, 39(19): 3492–3503.

Lee, S. H. and Jun, C. H. (2012). A process monitoring scheme controlling false alarm discovery rate. *Communications in Statistics-Simulation and Computation*, 41(10): 1912–1920.

Lee, M.H. and Khoo, M. (2006). Optimal statistical design of a multivariate EWMA chart based on ARL and MRL. *Communication in Statistics-Simulation and Computation*, 35: 831–847

Leoni, R.C., Costa, A.F.B., and Franco, B.C. (2015). The skipping strategy to reduce the effect of the autocorrelation on the T^2 chart's performance. *The International Journal of Advanced Manufacturing Tech.*, 80(9): 1547-1559. doi:10.1007/s00170-015-7095-1

Lowry, C. A., Woodall, W. H., Champ, C. W. and Rigdon, S. E. (1992). A multivariate exponential weighted moving average control chart. *Technometrics*, 34: 46–53.

- Lu, S.L. (2015). An extended nonparametric exponentially weighted moving average sign control chart. *Quality and Reliability Engineering International*, 31(1): 3–13.
- Mahmoud, M.A. and Zahran, A. (2010). A multivariate adaptive exponentially weighted moving average control chart. *Communication in Statistics: Theory and Methods*, 39: 606–625.
- Montgomery, D.C. (2002). *Introduction to Statistical Quality Control* (4th ed.). John Wiley & Sons, New York.
- Montgomery, D.C. (2009). *Introduction to Statistical Quality Control* (6th ed.). John Wiley & Sons, New York.
- Pan, X. and Jarrett, J. (2007). Using vector autoregressive residuals to monitor multivariate processes in the presence of serial correlation. *International Journal of Production Economics*, 106(1): 204–216.
- Patel, A.K. and Divecha, J. (2013). Modified MEWMA control scheme for an analytical process data. *Global Journal of Computer Science and Technology Software & Data Engineering*, 13 (3): 23–33.
- Phaladiganon, P., Kim, S.B., Chen, V.C.P., Baek, J.G. and Park, S.K. (2011). Bootstrap based T^2 multivariate control charts. *Communication in Statistics-Simulation and Computation*, 40(5): 645–662.
- Psarakis, S. and Papaleonida, G. E. A. (2007). Statistical Process Control (SPC) procedures for monitoring autocorrelated processes. *Quality Technology and Quantitative Management*, 4(4): 501-540.
- Rabyk, L. and Schmid, W. (2016). EWMA control charts for detecting changes in the mean of a long-memory process. *Metrika*, 79: 267-301. doi:10.1007/s00184-015-0555-7
- Stoumbos, Z.G. and Sullivan, J.H. (2002). Robustness to non-normality of the multivariate EWMA control chart. *Journal of Quality Technology*, 34: 260-276.
- Sullivan, J. H. and Stoumbos, Z. G. (2001). Achieving robust performance with the MEWMA control chart. *ASA Proceedings of the Joint Statistical Meetings, American Statistical Association* (Alexandria, VA), available at: https://ww2.amstat.org/meetings/JSM/2001/index.cfm?fuseaction=abstract_details&abstractid=300471
- Woodall, W. H. and Mahmoud, M. A. (2005). The inertial properties of quality control charts. *Technometrics*, 47(4): 425–436.

Appendix

Simulation Code

```

=====
F-Distribution Hotelling's T2
=====
t2calc<-function (nsim,mu,delta,phi,sig,gam,cv) {
  #Simulates the run length of the chi-squared chart for VAR(1) data.
  #Requires the MASS package to be loaded.
  #nsim is the number of simulations to be run (number of run lengths
  # to be generated).
  #mu is the in-control mean vector.
  #delta is the vector of the mean shift.
  #phi is the autocorrelation matrix.
  #sig is the error covariance matrix.
  #gam is the cross-covariance matrix.
  #cv is the upper control limit of the chart.
  #Fix the mean vector to work in the following calculations.
  mu<-t(t(mu))
  #Initialize the run length vector and Z vector for individual Z values
  # at each time point.
  rl<-matrix(0,nsim,1)
  Z<-matrix(0,1,length(mu))
  #Create the 0 mean vector for the calculation of the error terms.
  mu3<-matrix(0,2,1)
  #Create the shifted mean.
  mu2<-mu+t(t(delta))
  #Perform the simulation to simulate each run length.
  for(i in 1:nsim) {
    k<-1 #Start the count of time points until signal.
    #Check the first time point.
    #Generate the first VAR(1) data vector.
    y<-mu2+t(t(mvrnorm(1,mu3,sig)))
    #Calculate the charting statistic.
    t2<-t(y-mu)%*%solve(gam)%*%(y-mu)
    #Continue until the statistic is above the control limit.
    while(t2<cv) {
      k<-k+1
      y<-mu2+phi%*%(y-mu2)+mvrnorm(1,mu3,sig)
      t2<-t(y-mu)%*%solve(gam)%*%(y-mu)
    }
    rl[i]<-k
  }
  #Return the vector of simulated run lengths.
  return(rl)
}
sink("ans4")
ans4=function() {
  library(MASS)
  nsim=10000
  mu=c(0,0)
  delta=c(0,0)
  phi=matrix(c(0.0644,0.1322,0.1482,0.0644),2,2)
  sig=matrix(c(1,0.9806,0.9806,1),2,2)
  gam=matrix(c(3.9767,0.0649,0.0649,2.0603),2,2)
  cv=5.8801
  t2calc(nsim,mu,delta,phi,sig,gam,cv)
}
ans4()

```

Bootstrap Hotelling's T^2

```

=====
t2calc<-function (nsim,mu,delta,phi,sig,gam,cv) {
  #Simulates the run length of the chi-squared chart for VAR(1) data.
  #Requires the MASS package to be loaded.
  #nsim is the number of simulations to be run (number of run lengths
  # to be generated).
  #mu is the in-control mean vector.
  #delta is the vector of the mean shift.
  #phi is the autocorrelation matrix.
  #sig is the error covariance matrix.
  #gam is the cross-covariance matrix.
  #cv is the upper control limit of the chart.
  #Fix the mean vector to work in the following calculations.
  mu<-t(mu)
  #Initialize the run length vector and Z vector for individual Z values
  # at each time point.
  rl<-matrix(0,nsim,1)
  Z<-matrix(0,1,length(mu))
  #Create the 0 mean vector for the calculation of the error terms.
  mu3<-matrix(0,2,1)
  #Create the shifted mean.
  mu2<-mu+t(delta)
  #Perform the simulation to simulate each run length.
  for(i in 1:nsim) {
    k<-1 #Start the count of time points until signal.
    #Check the first time point.
    #Generate the first VAR(1) data vector.
    y<-mu2+t(mvrnorm(1,mu3,sig))
    #Calculate the charting statistic.
    t2<-t(y-mu)%*%solve(gam)%*(y-mu)
    #Continue until the statistic is above the control limit.
    while(t2<cv) {
      k<-k+1
      y<-mu2+phi%*(y-mu2)+mvrnorm(1,mu3,sig)
      t2<-t(y-mu)%*%solve(gam)%*(y-mu)
    }
    rl[i]<-k
  }
  #Return the vector of simulated run lengths.
  return(rl)
}
sink("ans1")
ans1=function(){
  library(MASS)
  nsim=10000
  mu=c(0,0)
  delta=c(0,0)
  phi=matrix(c(0.0644,0.1322,0.1482,0.0644),2,2)
  sig=matrix(c(1,0.0226,0.0226,1),2,2)
  gam=matrix(c(3.9767,0.0649,0.0649,2.0603),2,2)
  cv=4.3135
  t2calc(nsim,mu,delta,phi,sig,gam,cv)
}
ans1()

```

Bootstrap MEWMA

```

=====
mewcalc<-function (nsim,mu,delta,phi,sig,gam,r,cv) {
#Simulates the run length of the MEWMA chart for VAR(1) data.
#Requires the MASS package to be loaded.
#nsim is the number of simulations to be run (number of run lengths to be
#generated).
#mu is the in-control mean vector.
#delta is the vector of the mean shift.
#phi is the autocorrelation matrix.
#sig is the error covariance matrix.
#gam is the cross-covariance matrix.
#r is the parameter of the MEWMA chart.
#cv is the upper control limit of the chart.
#Fix the mean vector to work in the following calculations.
mu<-t(t(mu))
#Initialize the run length vector and Z vector for individual Z values
# at each time point.
rl<-matrix(0,nsim,1)
Z<-matrix(0,1,length(mu))
#Create the 0 mean vector for the calculation of the error terms.
mu3<-matrix(0,2,1)
#Create the shifted mean.
mu2<-mu+t(t(delta))
#Perform the simulation to simulate each run length.
for(i in 1:nsim) {
k<-1 #Start the count of time points until signal.
#Generate the first VAR(1) data vector.
y<-mu2+t(t(mvrnorm(1,mu3,sig)))
#Calculate the sigma matrix for the MEWMA statistic.
sigz<-(r/(2-r))*(1-(1-r)^2)*gam
#Calculate the initial z value.
z<-mu3
#Calculate the charting statistic.
mew<-t(z)%*%solve(sigz)%*%z
#Continue until the statistic is above the control limit.
while(mew<cv) {
k<-k+1
y<-mu2+phi%*%(y-mu2)+t(t(mvrnorm(1,mu3,sig)))
sigz<-(r/(2-r))*(1-(1-r)^(2*k))*gam
z<-r*y+(1-r)*z
mew<-t(z)%*%solve(sigz)%*%z
}
rl[i]<-k
}
#Return the vector of simulated run lengths. 135
return(rl)
}
sink("ans3")
ans3=function(){
library(MASS)
nsim=10000
mu=c(0.0,0.0)
delta=c(0,0)
phi=matrix(c(0.0644,0.1322,0.1482,0.0644),2,2)
sig=matrix(c(1,0.0226,0.0226,1),2,2)
gam=matrix(c(3.9767,0.0649,0.0649,2.0603),2,2)
r=0.1
cv=4.0512
mewcalc(nsim,mu,delta,phi,sig,gam,r,cv)
}
ans3()

```