

EVALUATION OF AUTOMATED STEADY STATE
AND TRANSIENT STATE DETECTION ALGORITHMS

By

SARVESH S GORE

Bachelor of Technology in Chemical Engineering

Visvesvaraya National Institute of Technology

Nagpur, India

2011

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
May, 2014

EVALUATION OF AUTOMATED STEADY STATE
AND TRANSIENT STATE DETECTION ALGORITHMS

Thesis Approved:

Dr. R. Russell Rhinehart

Thesis Adviser

Dr. Rob Whiteley

Dr. Martin Hagan

Dr. Balabhaskar Balasundaram

ACKNOWLEDGEMENTS

I would like to thank all the people who made this thesis study an unforgettable and enjoyable experience for me.

First and foremost, I would like to express my sincere gratitude towards Dr. R. Russell Rhinehart for providing me with continuous support and guidance throughout my master's study. Without his motivation and enthusiasm, this experience would not have been as memorable. He showed immense patience and provided support throughout numerous research meetings. His words have always inspired and brought the best out of me.

Besides my advisor, I would also like to thank my committee members, Dr. Rob Whiteley, Dr. Hagan, and Dr. Balabhaskar Balasundaram for their insightful comments and suggestions.

In addition, I would also like to show my appreciation to Dr. Anand Vennavalli for helping me see how the research was applicable in industry.

I would also like to thank Sushrut Pimple, for his encouragement and his insightful discussions throughout the research. Most importantly, I wish to thank my parents and my family for providing me with a carefree environment, so that I could concentrate on my study. Without their encouragement and continuous support it would have been a very daunting aspect to complete my education so far from my homeland.

Name: SARVESH S GORE

Date of Degree: MAY, 2014

Title of Study: EVALUATION OF AUTOMATED STEADY STATE AND
TRANSIENT STATE DETECTION ALGORITHMS

Major Field: CHEMICAL ENGINEERING

Abstract:

A comprehensive comparison of two statistical methods for automated identification of probable steady state and probable transient state in a noisy process signal is performed. Both approaches use the R-statistic method, which is the ratio of estimated variances, for steady state identification and are independent of noise variance. The performance of both approaches is determined based on probability of occurrence of Type-I, Type-II errors and the Average Run Length (ARL) at points of change in the process signal. The effectiveness of both approaches with respect to computational burden, computational time, ease of understanding, storage, etc. is analyzed for step change as well as ramp change in the noisy process signal.

TABLE OF CONTENTS

Chapter	Page
I. INTRODUCTION.....	1
1.1 OVERVIEW.....	1
1.2 LITERATURE SURVEY.....	4
1.2.1 LINEAR REGRESSION TECHNIQUE.....	4
1.2.2 T-TEST TYPE STATISTIC APPROACH.....	4
1.2.3 SPLIT WINDOW APPROACH.....	5
1.2.4 POLYNOMIAL INTERPOLATION APPROACH.....	5
1.2.5 WAVELET APPROACH.....	5
1.2.6 R-STATISTIC APPROACH.....	6
1.2.7 NULL HYPOTHESIS AND CRITICAL VALUES.....	6
1.3 SUMMARY.....	7
II. METHOD.....	8
2.1 R-STATISTIC MODEL – FILTER APPROACH.....	8
2.2 R-STATISTIC MODEL – ARRAY APPROACH.....	11
2.3 EXPLANATION OF CRITICAL VALUES.....	14
2.4 UNDESIRABLES IN THE APPROACH.....	17
2.5 INITIALIZATION OF THE ALGORITHMS.....	18
III. SIMULATION PROCEDURE.....	19
3.1 PROCEDURE.....	19
3.2 SELECTION OF CRITICAL VALUES.....	21
IV. RESULTS AND DISCUSSION.....	22
4.1 PERFORMANCE WITH RESPECT TO UNDESIRABLES IN THE PROCESS.....	22
4.1.1 STEP CHANGE.....	23
4.1.2 RAMP CHANGE.....	32
4.2 SUMMARY OF COMPARISON BASED ON PERFORMANCE.....	41
4.3 COMPARISON BASED ON FUNCTIONING OF THE APPROACHES.....	42
4.3.1 EASE OF UNDERSTANDING.....	43
4.3.2 STORAGE.....	43
4.3.3 COMPUTATIONAL BURDEN.....	44

4.3.4 EASE OF TUNING.....	44
V. CONCLUSIONS AND RECOMMENDATIONS	45
5.1 CONCLUSIONS	46
5.2 RECOMMENDATIONS FOR THE FUTURE	46
REFERENCES	47
APPENDICES	49

LIST OF TABLES

Table	Page
Table 4.1: Summary of comparison of the filter approach and the array approach for different holding windows for ramp as well as step change.....	41

LIST OF FIGURES

Figure	Page
Figure 2.1: The Filter concept illustration	9
Figure 2.2: The array concept illustration.....	13
Figure 2.3: R-Statistic distribution at steady state	15
Figure 2.4: R-Statistic distribution showing high probability of not being at steady state (Transient state condition)	16
Figure 2.5: R-Statistic distribution showing steady state identification	17
Figure 4.1: ARL vs step to noise ratio for N=75 and $\lambda_1=0.2, \lambda_2=0.1, \lambda_3=0.1$ for a step test.....	24
Figure 4.2: ARL vs step to noise ratio for N=55 and $\lambda_1=0.2, \lambda_2=0.1, \lambda_3=0.1$ for a step test.....	24
Figure 4.3: ARL vs step to noise ratio for N=35 and $\lambda_1=0.2, \lambda_2=0.1, \lambda_3=0.1$ for a step test.....	24
Figure 4.4: ARL vs step to noise ratio for N=20 and $\lambda_1=0.2, \lambda_2=0.1, \lambda_3=0.1$ for a step test.....	25
Figure 4.5: T-II error vs signal to noise ratio for N=35 and $\lambda_1=0.2, \lambda_2=0.1, \lambda_3=0.1$ for a step test	26
Figure 4.6: ARL vs step to noise ratio for N=75 and $\lambda_1=0.1, \lambda_2=0.1, \lambda_3=0.1$ for a step test.....	26
Figure 4.7: ARL vs step to noise ratio for N=55 and $\lambda_1=0.1, \lambda_2=0.1, \lambda_3=0.1$ for a step test.....	26
Figure 4.8: ARL vs step to noise ratio for N=40 and $\lambda_1=0.1, \lambda_2=0.1, \lambda_3=0.1$ for a step test.....	27
Figure 4.9: ARL vs step to noise ratio for N=20 and $\lambda_1=0.1, \lambda_2=0.1, \lambda_3=0.1$ for a step test.....	27
Figure 4.10: T-II error vs signal to noise ratio for N=40 and $\lambda_1=0.1, \lambda_2=0.1, \lambda_3=0.1$ for a step test.....	28
Figure 4.11: ARL vs step to noise ratio for N=75 and $\lambda_1=0.1, \lambda_2=0.05, \lambda_3=0.05$ for a step test.....	28

Figure	Page
Figure 4.12: ARL vs step to noise ratio for $N=70$ and $\lambda_1=0.1, \lambda_2=0.05, \lambda_3=0.05$ for a step test.....	28
Figure 4.13: ARL vs step to noise ratio for $N=50$ and $\lambda_1=0.1, \lambda_2=0.05, \lambda_3=0.05$ for a step test.....	29
Figure 4.14: ARL vs step to noise ratio for $N=25$ and $\lambda_1=0.1, \lambda_2=0.05, \lambda_3=0.05$ for a step test.....	29
Figure 4.15: T-II error vs signal to noise ratio for $N=70$ and $\lambda_1=0.1, \lambda_2=0.05, \lambda_3=0.05$ for a step test	30
Figure 4.16: ARL vs step to noise ratio for $N=75$ and $\lambda_1=0.05, \lambda_2=0.05, \lambda_3=0.05$ for a step test.....	30
Figure 4.17: ARL vs step to noise ratio for $N=70$ and $\lambda_1=0.05, \lambda_2=0.05, \lambda_3=0.05$ for a step test	30
Figure 4.18: ARL vs step to noise ratio for $N=55$ and $\lambda_1=0.05, \lambda_2=0.05, \lambda_3=0.05$ for a step test	31
Figure 4.19: ARL vs step to noise ratio for $N=40$ and $\lambda_1=0.05, \lambda_2=0.05, \lambda_3=0.05$ for a step test	31
Figure 4.20: T-II error vs signal to noise ratio for $N=75$ and $\lambda_1=0.05, \lambda_2=0.05, \lambda_3=0.05$ for a step test	31
Figure 4.21: ARL vs step to noise ratio for $N=75$ and $\lambda_1=0.2, \lambda_2=0.1, \lambda_3=0.1$ for a ramp test.....	32
Figure 4.22: ARL vs step to noise ratio for $N=55$ and $\lambda_1=0.2, \lambda_2=0.1, \lambda_3=0.1$ for a ramp test.....	33
Figure 4.23: ARL vs step to noise ratio for $N=40$ and $\lambda_1=0.2, \lambda_2=0.1, \lambda_3=0.1$ for a ramp test.....	33
Figure 4.24: ARL vs step to noise ratio for $N=25$ and $\lambda_1=0.2, \lambda_2=0.1, \lambda_3=0.1$ for a ramp test.....	33
Figure 4.25: T-II error vs signal to noise ratio for $N=25$ and $\lambda_1=0.2, \lambda_2=0.1, \lambda_3=0.1$ for a ramp test.....	34
Figure 4.26: ARL vs step to noise ratio for $N=65$ and $\lambda_1=0.1, \lambda_2=0.1, \lambda_3=0.1$ for a ramp test.....	35
Figure 4.27: ARL vs step to noise ratio for $N=55$ and $\lambda_1=0.1, \lambda_2=0.1, \lambda_3=0.1$ for a ramp test.....	35
Figure 4.28: ARL vs step to noise ratio for $N=40$ and $\lambda_1=0.1, \lambda_2=0.1, \lambda_3=0.1$ for a ramp test.....	35

Figure 4.29: ARL vs step to noise ratio for $N=30$ and $\lambda_1=0.1, \lambda_2=0.1, \lambda_3=0.1$ for a ramp test	36
Figure 4.30: T-II error vs signal to noise ratio for $N=30$ and $\lambda_1=0.1, \lambda_2=0.1, \lambda_3=0.1$ for a ramp test.....	36
Figure 4.31: ARL vs step to noise ratio for $N=75$ and $\lambda_1=0.1, \lambda_2=0.05, \lambda_3=0.05$ for a ramp test.....	37
Figure 4.32: ARL vs step to noise ratio for $N=60$ and $\lambda_1=0.1, \lambda_2=0.05, \lambda_3=0.05$ for a ramp test.....	37
Figure 4.33: ARL vs step to noise ratio for $N=40$ and $\lambda_1=0.1, \lambda_2=0.05, \lambda_3=0.05$ for a ramp test.....	37
Figure 4.34: ARL vs step to noise ratio for $N=20$ and $\lambda_1=0.1, \lambda_2=0.05, \lambda_3=0.05$ for a ramp test.....	38
Figure 4.35: T-II error vs signal to noise ratio for $N=40$ and $\lambda_1=0.1, \lambda_2=0.05, \lambda_3=0.05$ for a ramp test	38
Figure 4.36: ARL vs step to noise ratio for $N=75$ and $\lambda_1=0.05, \lambda_2=0.05, \lambda_3=0.05$ for a ramp test.....	39
Figure 4.37: ARL vs step to noise ratio for $N=65$ and $\lambda_1=0.05, \lambda_2=0.05, \lambda_3=0.05$ for a ramp test.....	39
Figure 4.38: ARL vs step to noise ratio for $N=35$ and $\lambda_1=0.05, \lambda_2=0.05, \lambda_3=0.05$ for a ramp test.....	39
Figure 4.39: ARL vs step to noise ratio for $N=20$ and $\lambda_1=0.05, \lambda_2=0.05, \lambda_3=0.05$ for a ramp test.....	40
Figure 4.40: T-II error vs signal to noise ratio for $N=65$ and $\lambda_1=0.05, \lambda_2=0.05, \lambda_3=0.05$ for a ramp test	40

CHAPTER I

INTRODUCTION

1.1 Overview

Manufacturing and chemical processes are usually noisy and non-stationary in nature. The process when stable, might be confounded by noise due to variety of reasons like environmental effects, measurement errors in input as well as output of the process, maintenance issues etc. Identification of steady state is therefore important for process control, analysis and development.

In a noisy process signal, steady state can be defined as the state at which, in a given window under consideration, the only change in the value of process variable will be due to the noise in the process. Steady state identification sets up the collection of data for fault detection in the process, and can be useful for process optimization, sensor analysis, data reconciliation, online process analysis, scheduling of applications, etc. Similarly, detection of a transient state, which can occur due to undesired changes in process or set point changes, can be useful in determining points of change in the process, collection of data for dynamic modelling, determining process response to an event, etc.

Statistical based steady state identification techniques are used in this study since they can be more robust for analyzing noisy process signals.

Often, in a manufacturing setup, engineers run sequence of experiments throughout the range of operating conditions for collection of data and process analysis. It is up to the process operators to observe the run and determine steady state in the process in order to trigger next stage of trial with new set of operating conditions. However, this visual method of identification requires continual human attention and is subject to human error in steady state identification due to noisy process data. Also, slow process changes, change of shift timing, or complicated process dynamics might affect the visual interpretation.

Alternately, time scheduling can be done to trigger new set of operating conditions for the experiment. However if the scheduled time interval is insufficient for the process to reach steady state, then the data collected might not be useful. Also, the method can be inefficient if the scheduling is for unnecessarily long time. It is nearly impossible to predict the holding window as with varying operating conditions amount of time required for the process to reach steady state might vary for each trial.

Consequently, an automated online real-time steady state identifier would be very useful to plan and schedule any experimental run or trial. The two steady state/transient state identification approaches explored in this study are based on the R-statistic method [1]. The first of the two approaches uses three first-order digital filters in order to calculate variance in a noisy process signal whereas the second approach evaluates variance based on the mean values and uses arrays for storage. A noisy process signal

was generated and the performance of the two approaches was evaluated based on the undesirables of the two approaches.

The simulated experiments were based on two types of changes in the noisy process signal – 1. Step changes made by varying step to noise ratio for different simulations. 2. Ramp changes made by varying slope of the ramp for different simulations. The holding window for both the changes was varied and the data of undesirables for both the approaches was collected from each simulation. The idea was to compare and evaluate the two approaches based on number of parameters: 1. Performance with respect to the undesirables in the methodologies by keeping other variables in the process same for both the approaches. 2. Ease of Understanding. 3. Computational burden. 4. Storage required for the approaches. 5. Ease of tuning the approaches.

The findings for the simulations revealed, 1. For step changes, the array approach is more sensitive to transient state identification compared to the filter approach. 2. For ramp changes, most often the performance of both the approaches is similar for larger holding window or lower slope of ramp (step to noise ratio). 3. For higher slope of the ramp, array approach shows better performance with respect to sensitivity to detection of transient state in the process. 4. Array approach is easier to tune and understand compared to filter approach. 5. Filter approach requires less storage and is computationally less burdensome.

1.2 Literature Survey

1.2.1 Linear Regression Technique

The linear regression technique is a direct approach to steady state identification. The technique involves performing linear regression of a sequence of data and evaluating the slope of the linear regression line. If this slope is close to zero, then there is high probability of the process being at steady state. On the other hand, if the slope is significantly different from zero, the process is probably at transient state [2]. If this approach was to be applied online in the process, it would require large storage and user expertise to determine the length of the holding window. For example, the slope of the linear regression is zero for an oscillating response, which might lead to a wrong steady state identification. Since the selection of length of data window requires human judgment, this approach is not automated. Also, this approach is computationally burdensome as the whole data window must be updated at each time interval.

1.2.2 T-Test type Statistic Approach

The T-Test approach evaluates average and standard deviation value in successive data windows of N samples and compares the two successively values using T-statistic which is the difference in averages divided by standard error of the average. If the process is at steady state, the averages will ideally be equal. If the variation is the process is high, the T-statistic value will exceed critical value to confidently claim that the process is at transient state. This approach however is computational expensive as the averages and standard deviation are to be computed and compared at each instant.

1.2.3 Split-window Approach

Split window approach was originally used to monitor control loops [13]. The idea was to split the data window under consideration in half and calculate the mean and variance in each half, then compute the statistic based on the ratio of difference of the averages in each half scaled by their standard deviations. The scaling was used to normalizing the statistic by having signal scaled by noise. The process claimed steady state when the ratio was equal to unity. This approach computes average and variance in two halves of the window and can be computationally expensive.

1.2.4 Polynomial Interpolation Approach

In the polynomial interpolation approach [14], a window length of N is selected and the data is fitted with a polynomial of order less than N . The statistic is the derivative of the polynomial at the center of the data window. In this approach, steady state is claimed if the statistic is nearly zero. However, this method does not scale the signal by the noise level. The detection of transient state also does not have defined critical values.

1.2.5 Wavelet Approach

Wavelet based Approach [15] was initially used for multi-time-scale process. The process was to model the process trends as wavelets and a wavelet transform modulus index ranging between 0 and 1 was used to trigger steady state. Steady state was claimed the index was nearly zero. This approach did not scale the process signal by the noise level.

1.2.6 R-statistic Approach

The R-statistic approach evaluates the ratio of two types of variances measured on the same set of data by using two different methods. R-statistic approach is explained in detail in Chapter 2. This approach was originally presented in the tutorial authored by Cao and Rhinehart (1995). Subsequently, the approach was demonstrated on a number of lab-scale and pilot-scale applications including its application to automate transitions between experimental runs on a two-phase flow unit and pilot-scale distillation process [3-5]. This approach was also demonstrated on an application in monitoring flow rate in a pilot-scale gas absorption unit [6] and subsequently on a commercial scale multi-variable distillation unit. It was also demonstrated as a convergence criterion in optimization of stochastic models and also its application stopping criterion in neural network training [7,8]. The R-statistic approach basically uses defined critical values for implementing the statistical method. This study demonstrates use of R-statistic approach in evaluation of performance of steady state and transient state detection algorithms.

1.2.7 Null Hypothesis and Critical values

A null hypothesis is commonly used in statistical analysis. It is basically the statement of the assumed state. Here, the null hypothesis is that the process is at steady state. So, in a statistical test, the choice of statistic will have an expected value when the null hypothesis is true and a different value if it were not true. However, if the process is noisy, the statistic value will not be uniform and will fluctuate from sample to sample.

If the statistic value has a small deviation from the expected value, then there is no evidence to claim with confidence that the null hypothesis is not true. On the other

hand, if the statistic value largely deviates from the expected value, then the likely explanation is that the null hypothesis is not true and larger the deviation, larger is the confidence that the null hypothesis can be rejected. The critical values are based on the extreme value of the statistic that would have a small chance of occurring if the null hypothesis were true. So, based on comparison of the statistic value with the critical value, the null hypothesis can be accepted or rejected.

1.3 Summary

An automated online steady state and transient state detection approach is preferable to visual identification techniques. This work demonstrates the R-statistic approach for identification of steady state and transient state detection using two different algorithms. The algorithms are evaluated based on various factors like their performance in recognizing changes in the process, ease of understanding, storage capacity required, computational burden, ease of tuning etc.

CHAPTER II

Method

2.1 R-statistic model – Filter Approach

Both the array approach and the filter approach use the R-statistic model, a ratio of two types of variances measured on the same set of data [1]. The filter approach, in order to reduce computational burden, uses three digital first-order filters to compute unbiased estimate of exponentially weighted moving average and variances. The array approach, on the other hand, uses conventional methods to estimate average and variance.

The R-statistic model for the filter concept can be illustrated using Fig (2.1). The dots in Fig (2.1) represents the noisy measured process data. Initially, the trend is at steady state until the time of 30 is reached. Then, it ramps up from the value of 15 to around 23. This ramp represents the transient state in the process, where the set point is changed. After this change the process again remains at steady state. The filtered value of the process measurement, represented by the solid line, and those representing variances are estimated using three first-order digital filters. Two types of variances are estimated by the filter approach. First, the deviation d_1 , indicated in the lower part of Fig (2.1),

is the difference between sequential data. Second, the deviation d2, indicated in upper right part of Fig (2.1) gives the difference between measured value and the filtered value.

If the process is at steady state, the filtered value is almost the same as average of the measured value. This can be observed in Fig (2.1) in the time period 0 to 30 and 70 to 100. In this case, the variance estimated by deviation d1, σ_{d1}^2 , will be ideally equal to variance estimated by deviation d2, σ_{d2}^2 , and the ratio of the two types of variances, denoted by the R-value, will be approximately unity, $R = \frac{\sigma_{d2}^2}{\sigma_{d1}^2} \approx 1$.

Alternatively, if the process is at transient state, as is shown in the ramp part of Fig (2.1), illustrated in the time period 30 to 70, then the filtered value deviates from the measured value and the variance estimated by deviation d2, σ_{d2}^2 , is much larger than the variance estimated by deviation d1 from sequential data, σ_{d1}^2 . So, in this case, the ratio of the variances given by R- statistic value will be much greater than unity as numerator is much greater than the denominator, $R = \frac{\sigma_{d2}^2}{\sigma_{d1}^2} \gg 1$

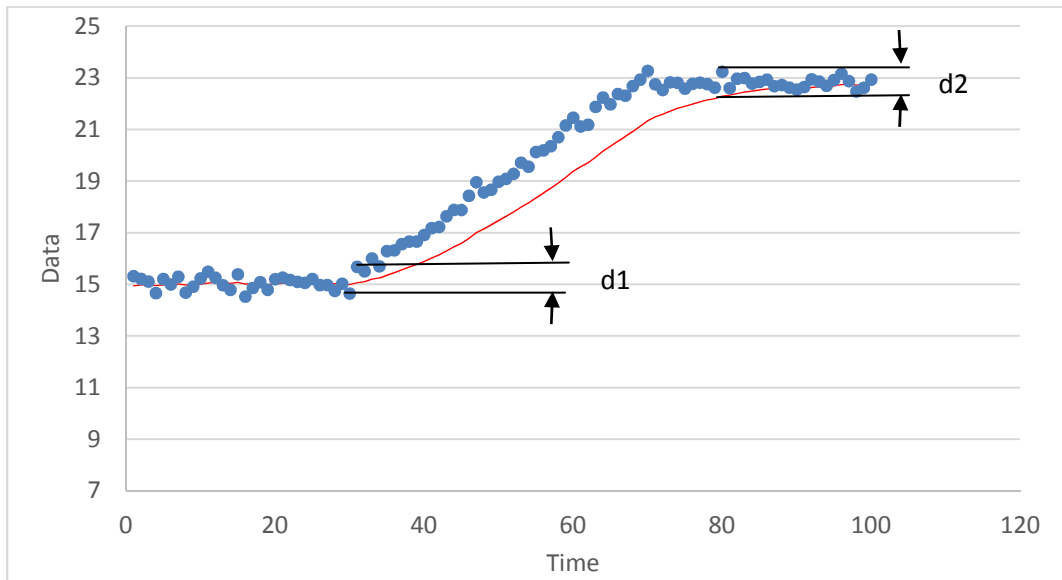


Figure 2.1: The Filter concept illustration

The filter approach uses three first-order filter in order to reduce computational burden for estimating variance or deviation from the measured trend. It estimates filtered trend based on following set of equations.

The first filter provides an approximation of the mean value of the sample data.

$$X_{f,i} = \lambda_1 X_i + (1 - \lambda_1) X_{f,i-1} \quad (1)$$

Where,

X = the process variable

X_f = Filtered value of X

λ_1 = Filter factor

i = Time sampling index

The second filter estimates the exponentially weighted moving variance based on the square of the difference between measured value and the filtered value which was estimated using Equation (1).

$$v^2_{f,i} = \lambda_2 (X_i - X_{f,i-1})^2 + (1 - \lambda_2) v^2_{f,i-1} \quad (2)$$

Where,

$v^2_{f,i}$ = Filtered value of a measure of variance based on variation from filtered trend

$v^2_{f,i-1}$ = Previous filtered value

Another approach for estimating the variance is to evaluate exponentially weighted moving variance based on the difference between sequential data. This approach is similar to the one used to evaluate Equation (2).

$$\delta_{f,i}^2 = \lambda_3(X_i - X_{i-1})^2 + (1 - \lambda_3)\delta_{f,i-1}^2 \quad (3)$$

Where,

$\delta_{f,i}^2$ = Filtered value of a measure of variance based on variation between sequential data

$\delta_{f,i-1}^2$ = Previous filtered value

Equation (2) gives the numerator of the R-statistic calculation whereas Equation (3) gives the denominator estimation. In order to avoid auto-correlation from biasing the variance estimation, the previous filtered value is used in both the cases to estimate numerator and denominator of the ratio instead of the most recently updated value of the variance.

The ratio of the variances can now be estimating using Equation (2) and Equation (3). The R-statistic value will be given as:

$$R = \frac{(2-\lambda_1)v_{f,i}^2}{\delta_{f,i}^2} \quad (4)$$

The lambda values in the filter approach can be related to the length of the holding window. Lower values of lambda represent higher number of data points used for analysis. In order to ensure higher confidence or faster identification of steady state, lambda values are recommended [7,9,10].

2.2 R-statistic Model – Array Approach

The major advantage of using three first-order filters is to reduce computational burden. However, the filter structure is not as comfortable as the concept of conventional sums in calculating variances for many people. Also, the concept of a holding window of

N samples is easier to grasp compared to the interpretation of three lambda values and the exponentially weighted infinite window.

The concept of the array approach is illustrated using the same example as the filter approach. The measured data is represented by the dots whereas the solid line represents the mean of the data. The deviation, d_1 , indicated in the lower part of Fig (2.2), give the difference between the sequential data. This deviation is the same as deviation, d_1 used for the filter approach. The deviation, d_2 , shown in the central part of Fig (2.2), evaluates the difference between measured data and the mean of the data in a particular sampling interval.

If the process is at steady state, the variance estimated by deviation d_1 will be ideally equal to the variance estimated by deviation d_2 and the ratio of the two types of variances will be approximately unity, $R = \frac{\sigma_{d_2}^2}{\sigma_{d_1}^2} \approx 1$. However, if the process is at transient state, then the variance estimated by deviation d_2 , is much larger compared to the variance estimated by the deviation d_1 . This can be seen in the time period 30 to 70, where the deviation of measured data is much larger from the mean of the particular sampling interval in comparison to the deviation in the sequential data. So, in this case, the ratio of the R-statistic value will be very greater than unity, $R = \frac{\sigma_{d_2}^2}{\sigma_{d_1}^2} \gg 1$

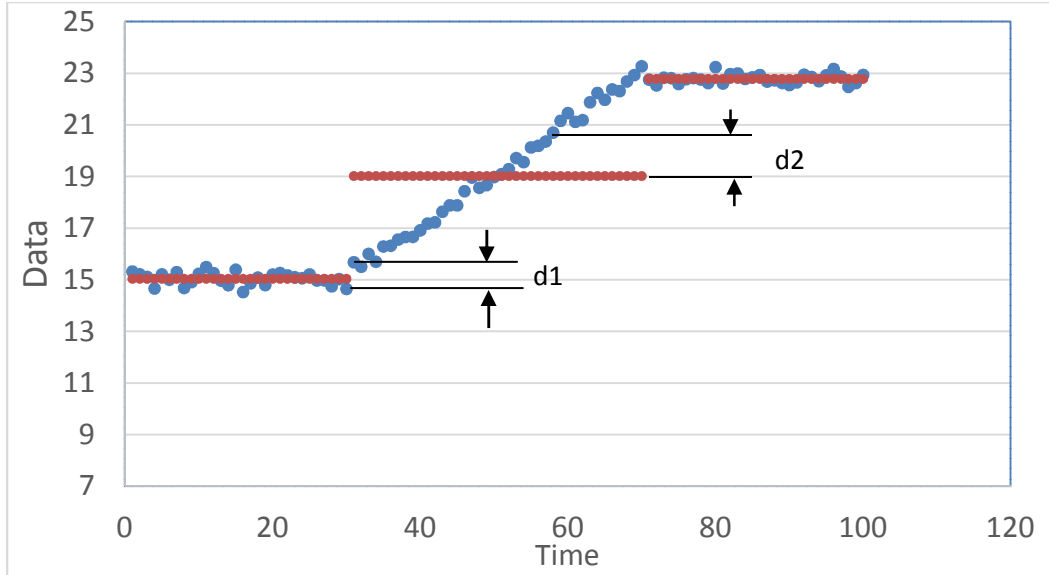


Figure 2.2: The array concept illustration

First, the average of the data measurements in the holding window of N is evaluated,

$$\bar{X} = \frac{1}{N} \sum_{i=1}^N X_i \quad (5)$$

N = Sampling interval

\bar{X} = average

The conventional variance is evaluated in an expanded form in equation (6)

$$\sigma_1^2 = \frac{1}{N-1} \left\{ \left(\sum_{i=1}^N X_i^2 \right) - \frac{1}{N} \left(\sum_{i=1}^N X_i \right)^2 \right\} \quad (6)$$

Where,

N = sampling interval

i = sample counter

X_i = measured data

Assuming no auto-correlation, the variance estimated from the difference of the sequential data is given in Equation (7)

$$\sigma_2^2 = \frac{1}{2(N-1)} (\sum_{i=1}^{N-1} (X_i - X_{i-1})^2) \quad (7)$$

Then the ratio of the variances is estimated as:

$$R = \frac{\sigma_1^2}{\sigma_2^2} \cong \frac{1}{2} \left(\frac{\frac{1}{N-1} \left\{ (\sum_{i=1}^N X_i^2) - \frac{1}{N} (\sum_{i=1}^N X_i)^2 \right\}}{\frac{1}{2(N-1)} \left\{ (\sum_{i=1}^{N-1} (x_i - x_{i+1})^2) \right\}} \right) = \left(\frac{sum1 + \frac{1}{N}(sum2)^2}{sum3} \right) \quad (8)$$

This R-statistic value evaluated for both the approaches needs to be compared to critical R-values in order to determine the Steady State (SS) and the Transient State (TS). The three sums are stored in an array and incremented or decremented at each sampling. A pointer is used in the array where the new value is added. This pointer is incremented at each sampling so that each time oldest data in the window is replaced with new values.

2.3 Explanation of Critical Values

Fig (2.3) shows the distribution of the R-Statistic at steady state. The average or the mean of the R-Statistic distribution is 1. However, due to statistical vagaries created due to noisy process data, the R-value will have variability from the mean.

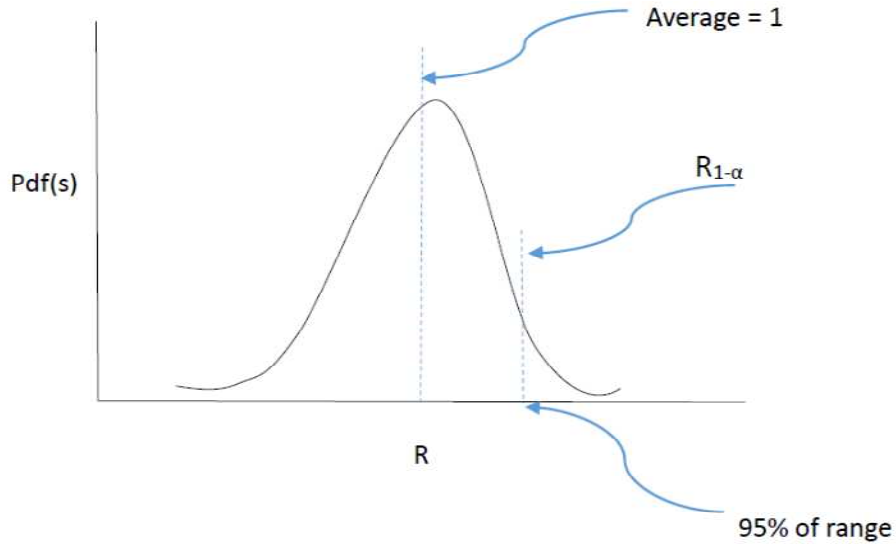


Figure 2.3: R-Statistic distribution at steady state

$R_{1-\alpha}$ represents the upper critical value. If the R value is less than the 95% confidence value ($R_{1-\alpha}$ value), then the process may be at steady state. On the other hand, if the R value is larger than $R_{1-\alpha}$ value, then there is a very low chance (5% chance) of process being at steady state and it is highly probable that the process is at transient state.

Fig (2.4) includes the R -statistic distribution of the process which is at transient state along with the steady state distribution. For a process which is not at steady state, there is a high probability of transient state condition, almost 70% as shown in this illustration. So, for R value greater than the critical $R_{1-\alpha}$ value, the odds of the R value coming from steady state distribution to the transient state distribution are around 1:15. In this case, transient state can be claimed with sufficient confidence.

However, if the R value is less than the critical $R_{1-\alpha}$ value, then the odds of the R value coming from steady state distribution are 95% as against the value coming from

transient state distribution which is around 30%. The 3:1 odds are not very definitive for claiming steady state confidently.

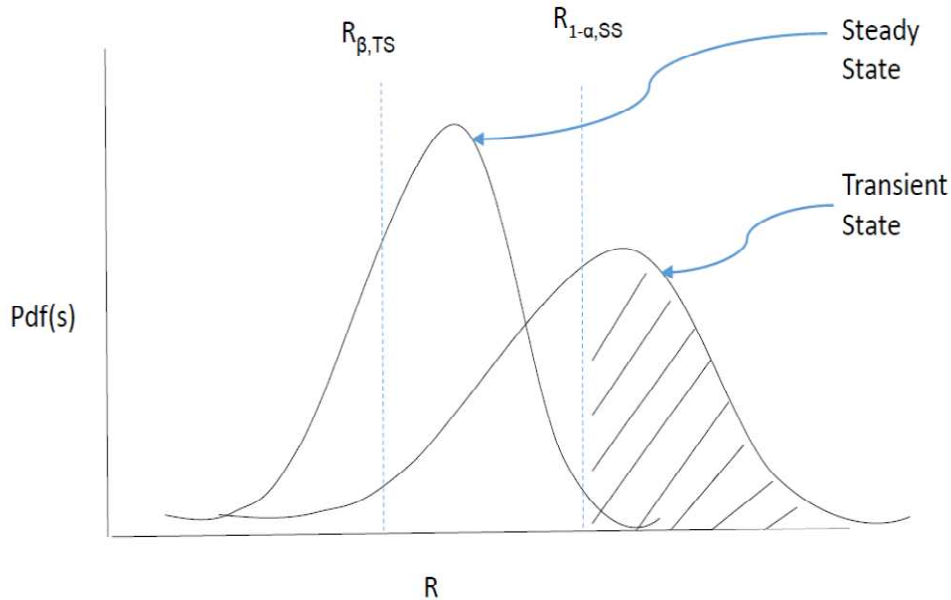


Figure 2.4: R-Statistic distribution showing high probability of not being at steady state (Transient state condition)

For claiming confidently that the process is at steady state, another critical value can be considered. Fig (2.5) shows a critical value, R_{β} , on the left side of the distribution. If the process is in steady state and $R < R_{\beta}$ then the probability of the R value coming from steady state distribution is about 40% as illustrated in Fig (2.5). However, if the process is at transient state, then the odds of the R value being less than the R_{β} value is very less, about 1% in this case. So, if $R < R_{\beta}$ then the odds of the process being at steady state are 40:1.

If $R_{\beta} < R < R_{1-\alpha}$ then one cannot confidently claim either steady state or transient state. But the chances of the process being at previous condition are high, and so the last

decision is held in this case. The critical values can be decided based on the requirements of the process [9,11]

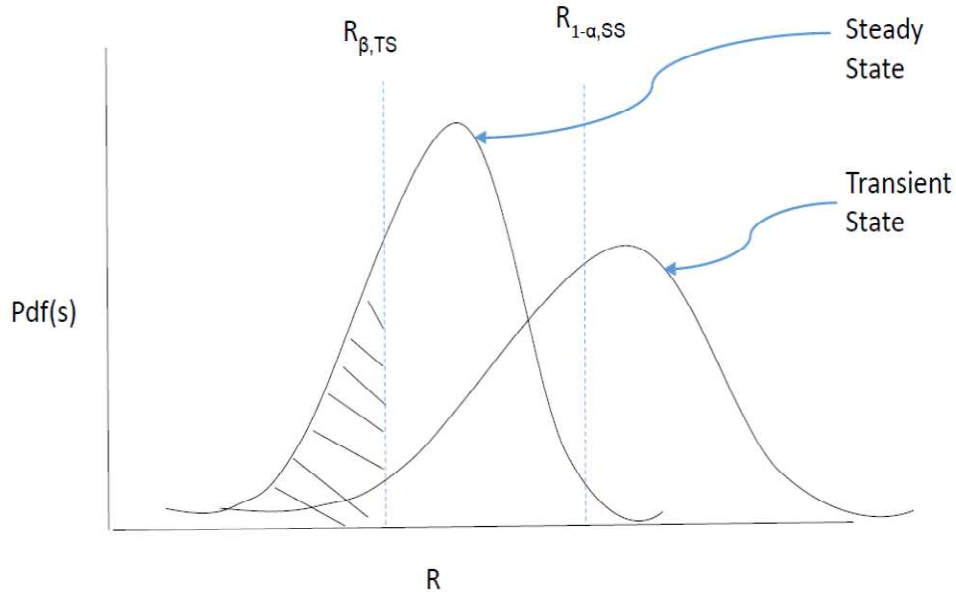


Figure 2.5: R-Statistic distribution showing steady state identification

2.4 Undesirables in the approach

Due to random variations and vagaries of the statistical methods, the steady state/transient state identifier might show wrong outputs. When $R < R_{\beta}$, there is a high probability of the process being at steady state, however, there is still a small probability of the process being at transient state. A similar case might occur when $R > R_{1-\alpha}$, when there is a slight chance of the process being at steady state. These wrong outputs of the steady state identifier can be summed up as errors or undesirables of the approaches.

Type-I error: It is the trigger of “not at steady state” claim when the process is actually at steady state. This trigger might happen if due to noisy measured data, the R-statistic value crosses the upper critical value. This error is dependent on the choice of α

value for determining the $R_{1-\alpha}$ critical value. The obvious way to reduce the Type-I error is to choose a smaller value of α [11].

Alternate Type-II error: It is the trigger for “at steady state” claim when the process is actually at transient state. If a very small value of α is selected then, the R_β value will increase and even if the process is at TS the condition might not be triggered since the R-value didn’t cross the critical value. Normally, this would be called a Type-II error and not Alternate Type-II error if the critical value was $R_{1-\alpha}$ instead of R_β .

Average Run Length (ARL): This gives the number of data sample points the method takes to recognize the point of change. ARL depends on the length of the sample window. For the array method, the average ARL is usually equal to twice the size of the sample window used.

2.5 Initialization of the algorithms

While initializing the algorithms, the steady state indicator is set at 0.5, in order to represent that initially the algorithms have no basis at all to claim either transient state (SS indicator = 0) or steady state (SS indicator = 1). Subsequently, as the R value is estimated from the ratio of the two types of variances, steady state can be claimed if R value is less than lower critical value. Alternately, if it is greater than upper critical value, transient state can be claimed confidently.

CHAPTER III

SIMULATION PROCEDURE

3.1 Procedure

The approach for comparison of performance of the filter method and the array was to compare the undesirables of the two approaches. A noisy process signal was generated using VBA-Excel and the two methods were tested on that signal. The Box-Muller Gaussian noise algorithm [12] was used to add noise to the process signal. Step and ramp changes of varying magnitude were made in the process signal for the same variance in noise in order to generate the performance results of the two approaches. The input variables for the methods used in the process were signal to noise ratio (s/n ratio), critical values, N for the array method, λ_1 (L1), λ_2 (L2), λ_3 (L3) for the filter approach, Type-I (T-I), Type-II (T-II) errors and Average Run Length (ARL) value. Following set of steps were used in order to carry out the simulations and generate data for comparing the two approaches:

1. Select signal to noise ratio.
2. Select N and $\lambda_1, \lambda_2, \lambda_3$ values.
3. Use a true Steady state data to estimate the R-critical values, $R_1 - \alpha$ and $R\beta$.

4. Select α and β values to be 0.01% and 25% respectively.
5. Perform step tests and ramp tests to replicate runs in order to reduce the effect of variability of the process on findings.
6. Obtain data for T-II and ARL for each method.
7. Change N (or $\lambda_1, \lambda_2, \lambda_3$) values and repeat Steps 3, 4 and 5.
8. Repeat the whole procedure for a different signal to noise ratio.
9. Choose lambda and N values, so that the ARL of both the methods is almost same and analyze the probability of T-II error vs s/n data.

For the sake of this study, T-II and ARL were compared keeping the third negative, probability of T-I error equivalent and negligible. For ramp test the signal to noise ratio represents the ratio of slope of the ramp to the variance of the noise. For the step test, the signal to noise ratio represents the ratio of step change to the variance in the noise (which was kept constant at 0.5). N for the array method was varied from 20 to 75 for each set of lambda for step test as well as ramp test. Lambda values were varied from 0.2 to 0.05.

3.2 Selection of critical values

The $1-\alpha$ and β values selected for determining the critical values were 99.9% ($\alpha = 0.01\%$) and 25%. A very high value of $1-\alpha$ was selected for reducing the probability of Type I error to a negligible value. In order to determine critical values, firstly steady state noisy process signal was generated and a cumulative distribution function of the R-values was plotted. Based on this distribution, the 25% value was selected as lower critical value and the 99.9% value was selected as the upper critical value. If the upper critical value is 99.9%, it means there is only 0.001 probability of Type I error which is negligible.

CHAPTER IV

RESULTS AND DISCUSSION

4.1 Performance with respect to undesirables in the approach

The simulation data is represented using different plots over a range of N values for each set of $\lambda_1, \lambda_2, \lambda_3$. ARL and T-II are plotted against s/n ratio for comparison of both approaches. The N values are varied from 75 to 20 window length and the lambda values are varied from 0.2 to 0.05.

ARL depends on the length of the holding window which depends on the N and lambda values. The larger the holding window, the larger will the maximum ARL value be and vice versa. T-II error on the other hand depends on number of parameters such as N, lambda, critical values, and s/n ratio. For comparison of the two approaches used in this study, it is best to select set of values in such a way that one of the undesirable, ARL or T-II error, is comparable for both approaches and then compare based on the other undesirable.

In order to evaluate performance of the array approach and filter approach, trend of ARL is observed for each set of lambda values. The N value, for which the ARL of

both the array approach and the filter approach is comparable, is selected. The T-II performance of both the algorithms for this set of N and lambda values is then evaluated. The simulation is carried out for step as well as ramp changes and the two approaches are evaluated based on their performance with respect to T-II error.

4.1.1 Step Change

Figs (4.1) to (4.20) show the trend of how ARL values change with change in N and lambda values for a step change. The diamond markers represent data for the filter method and the triangular markers represent data for the array method. Replicate trials are visible as vertically stacked markers for a particular s/n value. It can be observed that as the N value decreases from 75 to 20 for each set of lambda values, the ARL value of the array approach also decreases. Also, as the lambda values decrease from 0.2 to 0.05, the ARL of the filter approach increases. The following charts show different sequences of lambda values. For each sequence, the figures below show the trend of the response of the ARL values to changing values of N. The chart titles for all the figures below represent λ_1 , λ_2 , λ_3 as L1, L2 and L3 respectively.

1. Figs (4.1) to (4.4) show the trend of ARL values for changing values of N. This sequence has lambda values as $\lambda_1 = 0.2$, $\lambda_2 = 0.1$, $\lambda_3 = 0.1$.

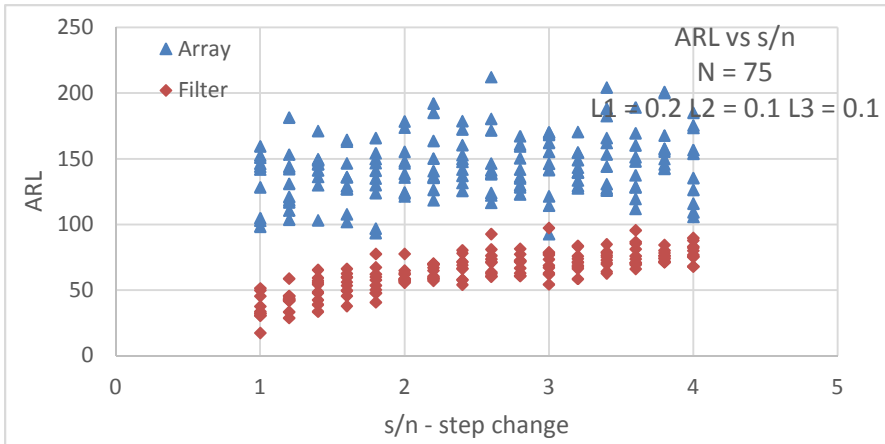


Figure 4.1: ARL vs step to noise ratio for $N=75$ and $\lambda_1=0.2$, $\lambda_2=0.1$, $\lambda_3=0.1$ for a step test

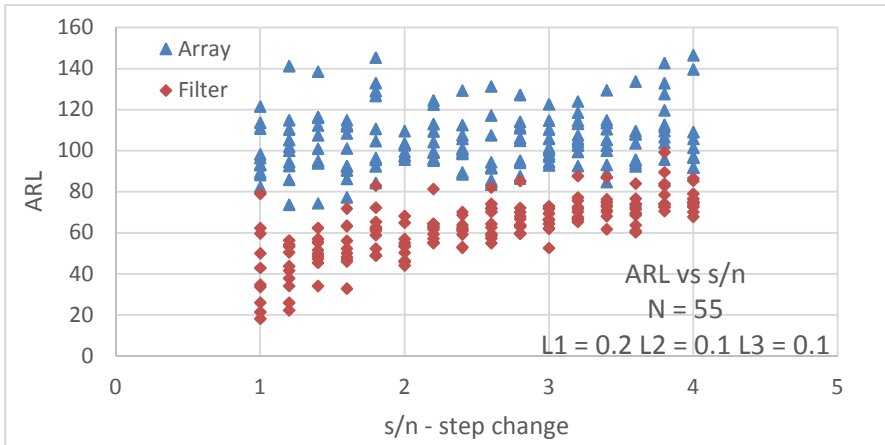


Figure 4.2: ARL vs step to noise ratio for $N=55$ and $\lambda_1=0.2$, $\lambda_2=0.1$, $\lambda_3=0.1$ for a step test

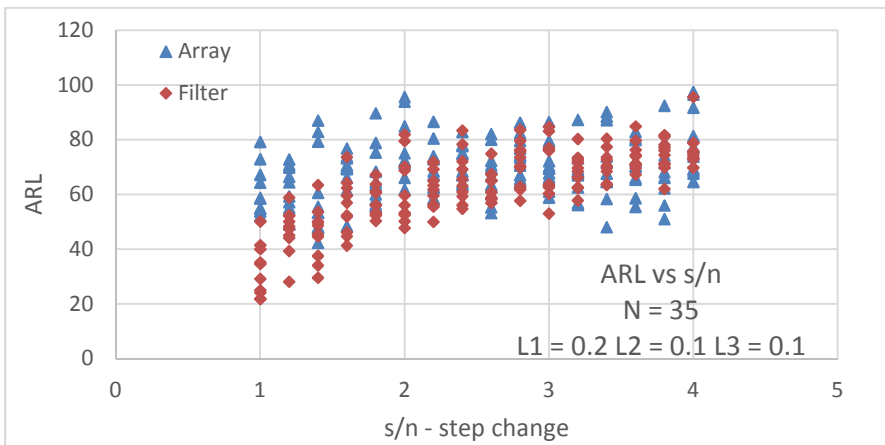


Figure 4.3: ARL vs step to noise ratio for $N=35$ and $\lambda_1=0.2$, $\lambda_2=0.1$, $\lambda_3=0.1$ for a step test

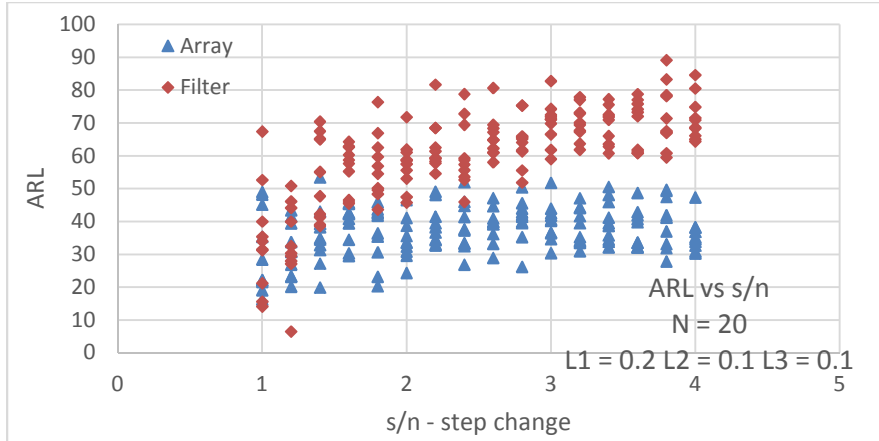


Figure 4.4: ARL vs step to noise ratio for $N=20$ and $\lambda_1=0.2, \lambda_2=0.1, \lambda_3=0.1$ for a step test

Here, Fig (4.3) depicts comparable ARL values for both the approaches indicating that $N=35$ has equivalent ARL (and T-I error) to lambda value sequence of 0.2, 0.1, 0.1. Fig (4.5) shows the T-II error vs s/n ratio plot for the corresponding values of N and lambda as used in Fig (4.3). For equivalent T-I error and ARL, the array approach, in this case, shows better sensitivity towards transient state detection compared to the filter approach.

However, comparing the performance of the two approaches based on only one set of lambda vales is not conclusive. In order to verify the performance of the two methods over a range of N and lambda values, the same step test was performed for lower lambda values, thus increasing the number of data points considered for the filter approach.

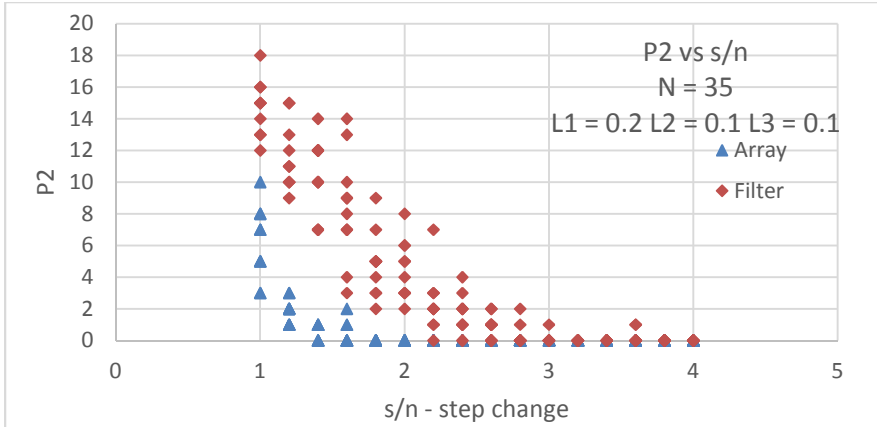


Figure 4.5: T-II error vs signal to noise ratio for $N=35$ and $\lambda_1=0.2, \lambda_2=0.1, \lambda_3=0.1$ for a step test

2. Figs (4.6) to (4.9) repeat the comparison for lambda values, $\lambda_1 = 0.1, \lambda_2 = 0.1, \lambda_3 = 0.1$.

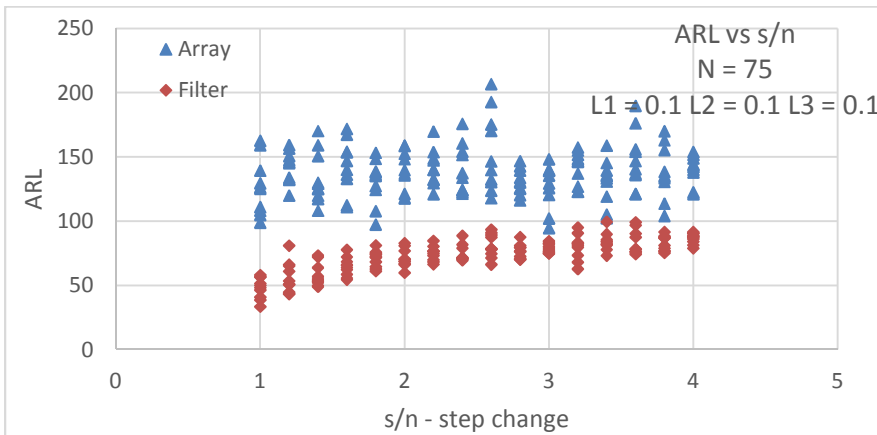


Figure 4.6: ARL vs step to noise ratio for $N=75$ and $\lambda_1=0.1, \lambda_2=0.1, \lambda_3=0.1$ for a step test

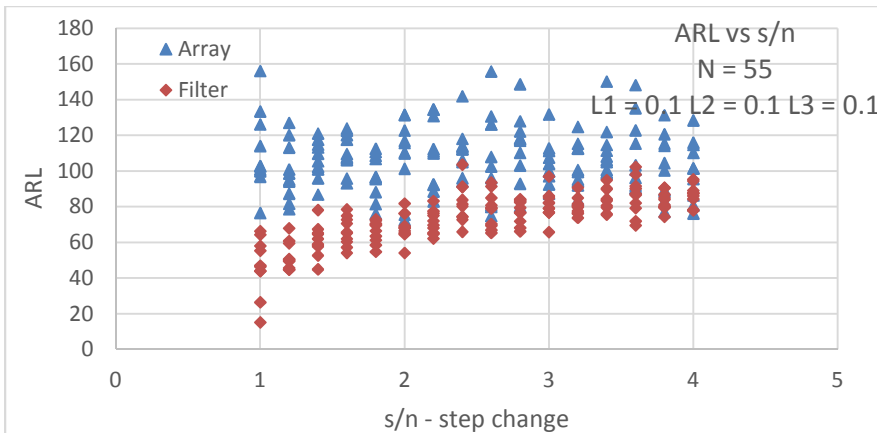


Figure 4.7: ARL vs step to noise ratio for $N=55$ and $\lambda_1=0.1, \lambda_2=0.1, \lambda_3=0.1$ for a step test

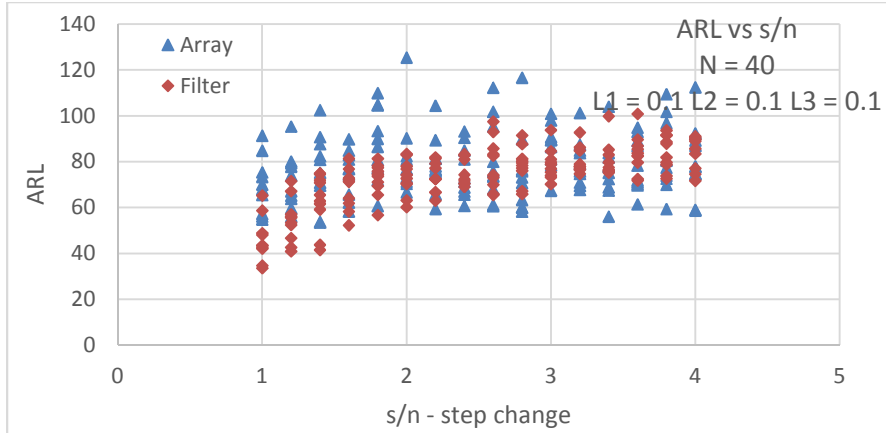


Figure 4.8: ARL vs step to noise ratio for $N=40$ and $\lambda_1=0.1, \lambda_2=0.1, \lambda_3=0.1$ for a step test

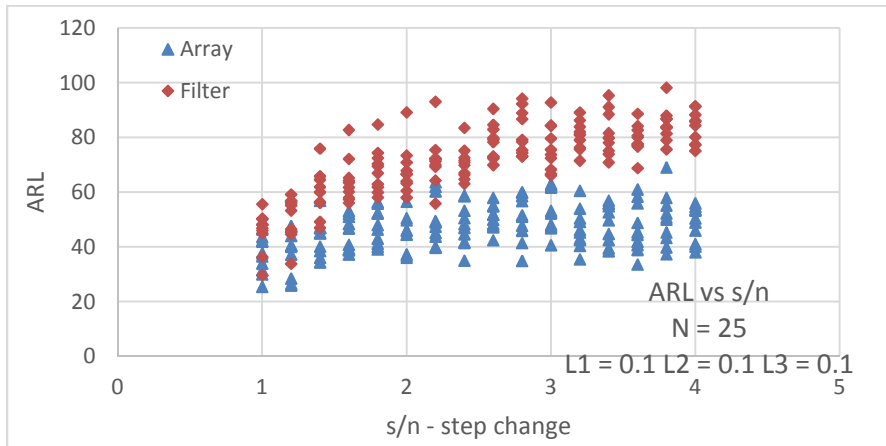


Figure 4.9: ARL vs step to noise ratio for $N=25$ and $\lambda_1=0.1, \lambda_2=0.1, \lambda_3=0.1$ for a step test

In this case, the array approach and the filter show similar ARL value for $N = 40$.

Fig (4.10) depicts the corresponding T-II plot for both approaches and array approach shows better performance for detecting transient state than the filter approach.

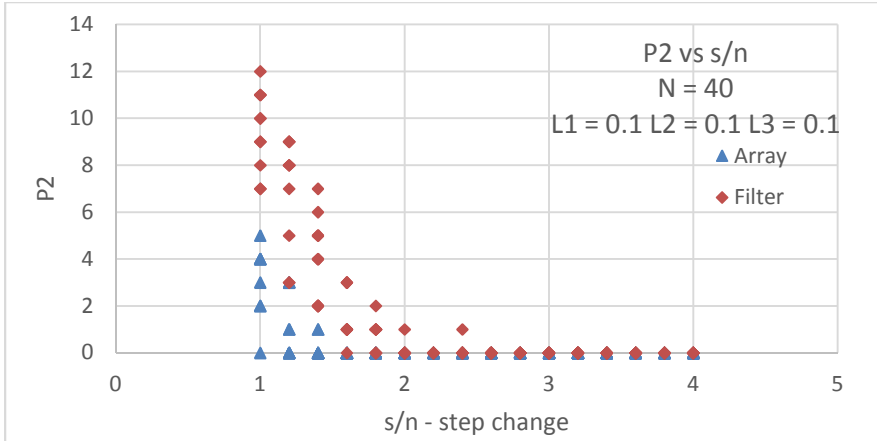


Figure 4.10: T-II error vs signal to noise ratio for $N=40$ and $\lambda_1=0.1, \lambda_2=0.1, \lambda_3=0.1$ for a step test

- The comparison of two approaches is repeated in Figs (4.11) to (4.14) for a different sequence of lambda values, $\lambda_1 = 0.1, \lambda_2 = 0.05, \lambda_3 = 0.05$.

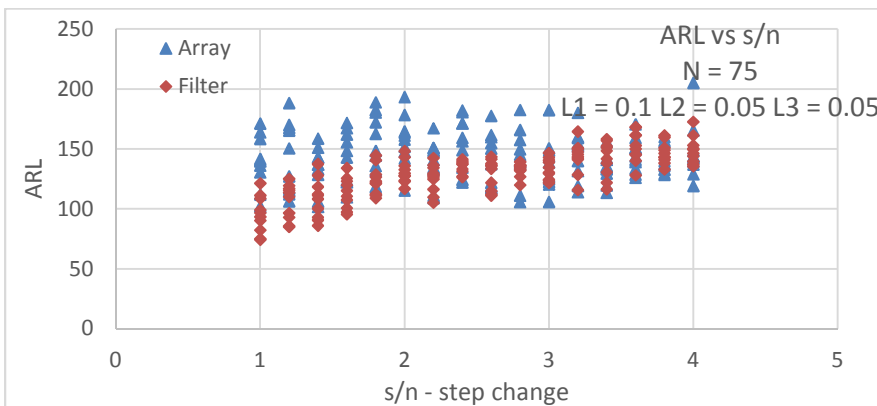


Figure 4.11: ARL vs step to noise ratio for $N=75$ and $\lambda_1=0.1, \lambda_2=0.05, \lambda_3=0.05$ for a step test

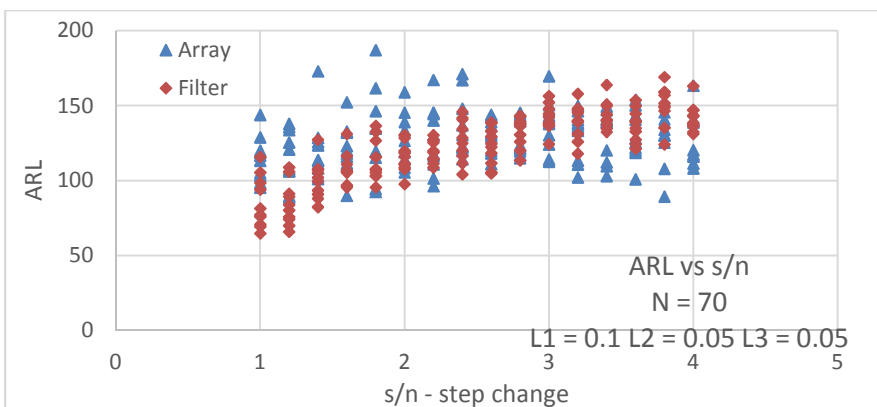


Figure 4.12: ARL vs step to noise ratio for $N=70$ and $\lambda_1=0.1, \lambda_2=0.05, \lambda_3=0.05$ for a step test

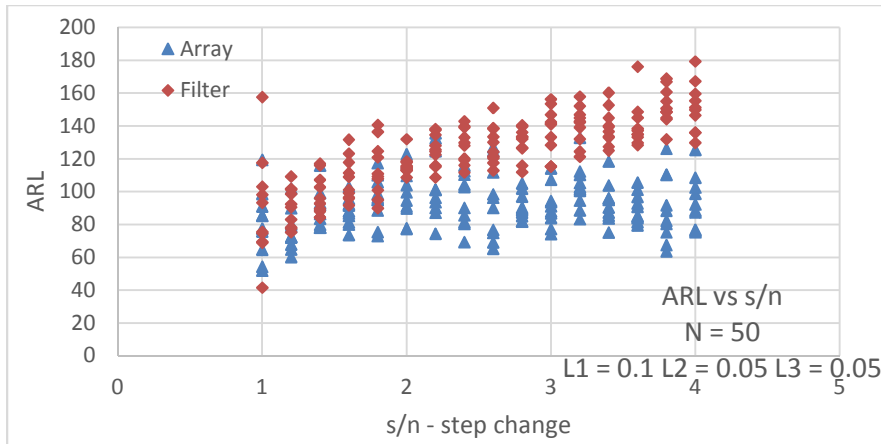


Figure 4.13: ARL vs step to noise ratio for $N=50$ and $\lambda_1=0.1$, $\lambda_2=0.05$, $\lambda_3=0.05$ for a step test

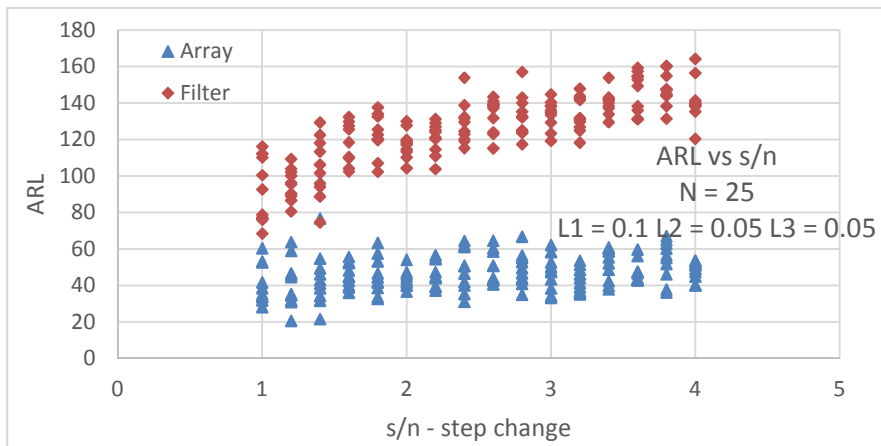


Figure 4.14: ARL vs step to noise ratio for $N=25$ and $\lambda_1=0.1$, $\lambda_2=0.05$, $\lambda_3=0.05$ for a step test

As the lambda values decrease, the ARL for the filter approach increases. So, a larger value of N shows comparable ARL values. Fig (4.15) indicates, at lower values of s/n ratio, the array approach gives better performance at detecting the transient state. The probability of T-II error depends on sample window used. A larger window length leads to a lower probability of T-II error. For higher values of s/n ratio, the step size is large enough for both the approaches to detect transient state for the larger holding window size.

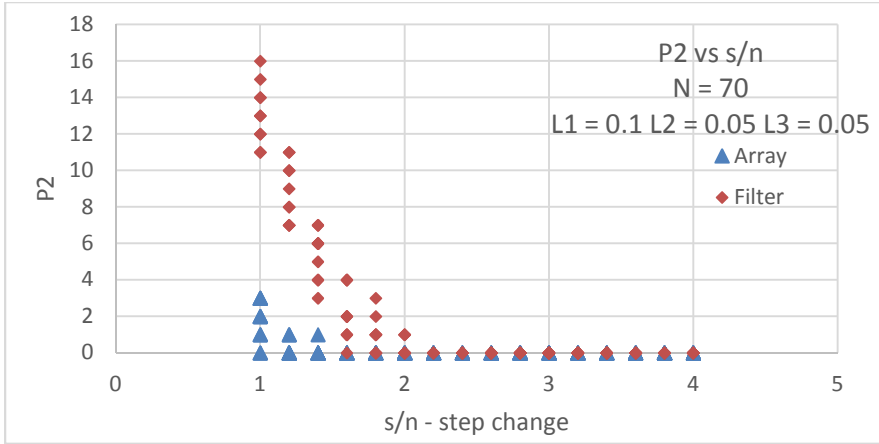


Figure 4.15: T-II error vs signal to noise ratio for $N=70$ and $\lambda_1=0.1$, $\lambda_2=0.05$, $\lambda_3=0.05$ for a step test

4. In order to be confident of the results derived from Fig (4.5), (4.10), (4.15), the comparison was repeated on another sequence, $\lambda_1 = 0.05$, $\lambda_2 = 0.05$, $\lambda_3 = 0.05$

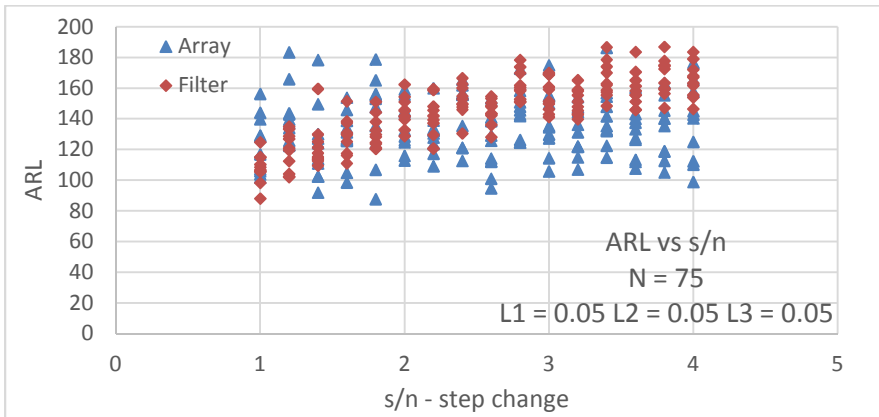


Figure 4.16: ARL vs step to noise ratio for $N=75$ and $\lambda_1=0.05$, $\lambda_2=0.05$, $\lambda_3=0.05$ for a step test

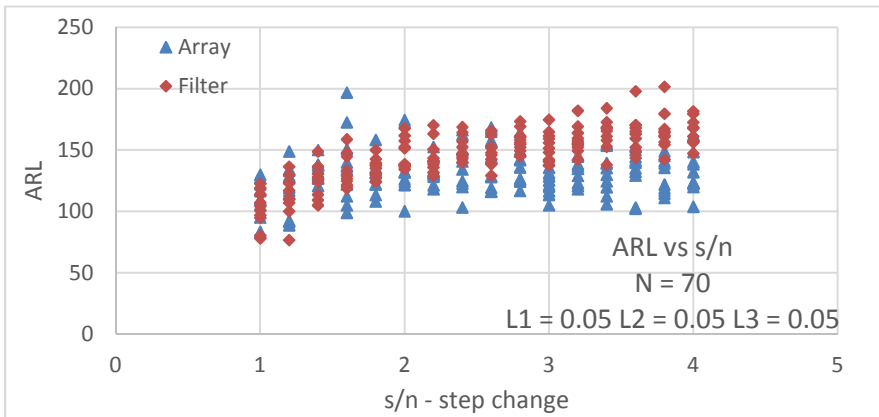


Figure 4.17: ARL vs step to noise ratio for $N=70$ and $\lambda_1=0.05$, $\lambda_2=0.05$, $\lambda_3=0.05$ for a step test

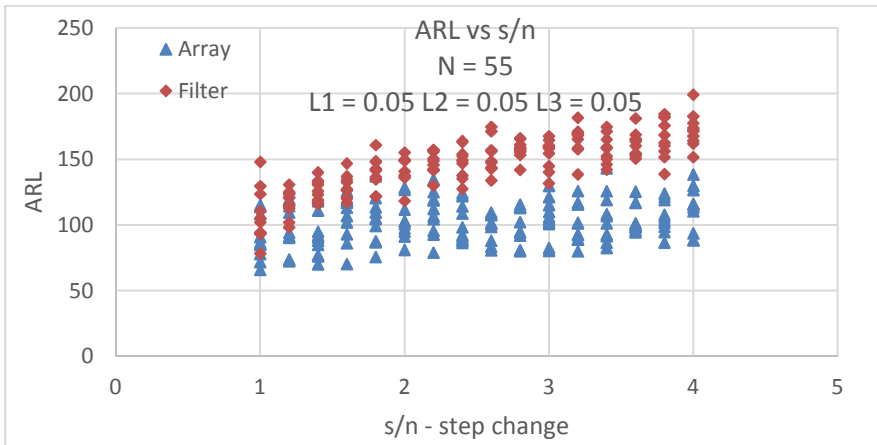


Figure 4.18: ARL vs step to noise ratio for $N=55$ and $\lambda_1=0.05$, $\lambda_2=0.05$, $\lambda_3=0.05$ for a step test

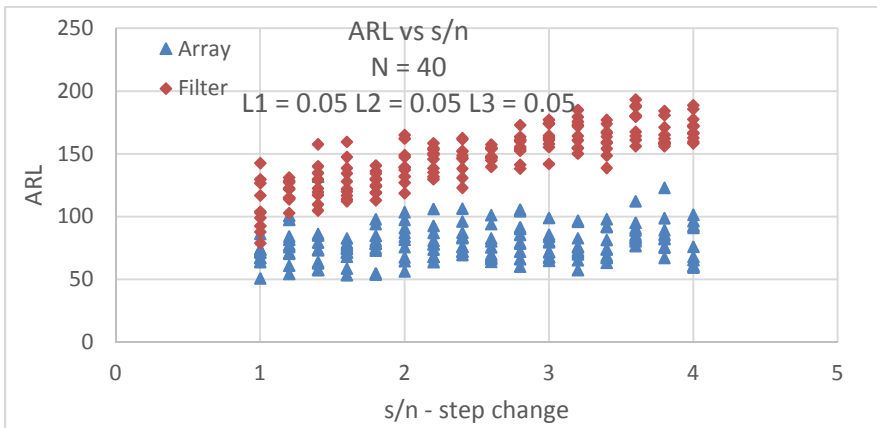


Figure 4.19: ARL vs step to noise ratio for $N=40$ and $\lambda_1=0.05$, $\lambda_2=0.05$, $\lambda_3=0.05$ for a step test

As the size of lambda is further decreased, the ARL values of both the approaches are comparable for $N=75$. Fig (4.20) represents T-II error plot for both approaches.

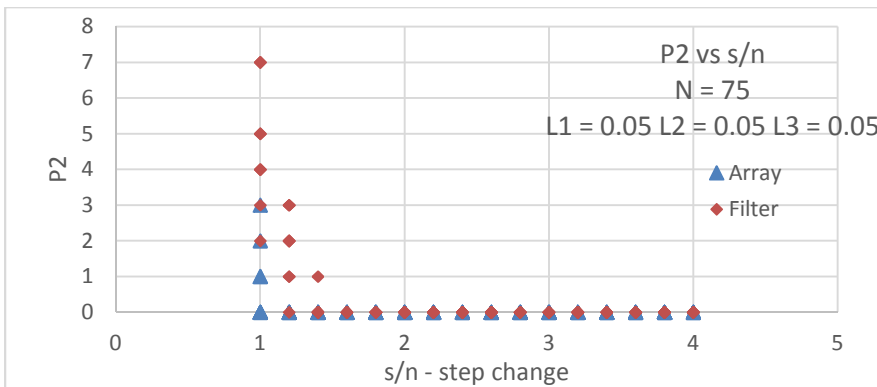


Figure 4.20: T-II error vs signal to noise ratio for $N=75$ and $\lambda_1=0.05$, $\lambda_2=0.05$, $\lambda_3=0.05$ for a step test

Figs (4.5), (4.10), (4.15), (4.20) reveal that for almost same ARL values the array approach gives a better performance than the filter approach in terms of T-II error, meaning the array method shows lower probability of triggering the “at steady state” condition when the process is at transient state. So, for a step test, the array method is more sensitive to change in the process compared to the filter approach.

4.1.2 Ramp Change

Simulations similar to step change were carried out for ramp changes. The data obtained was plotted to represent ARL vs s/n ratio and T-II vs s/n ratio. The variation of noise is fixed at 0.5 for this case. The s/n ratio represents the slope of the ramp. The approach used for evaluating the performance of array and filter algorithms for ramp change in process signals was similar to one used for step change. Same set of lambda and N were used for the ramp change simulations.

1. The first set of sequence of lambda values used was $\lambda_1 = 0.2$, $\lambda_2 = 0.1$, $\lambda_3 = 0.1$.
Again, the T-II error was compared for values of lambda and N which would give equivalent ARL and T-I error.

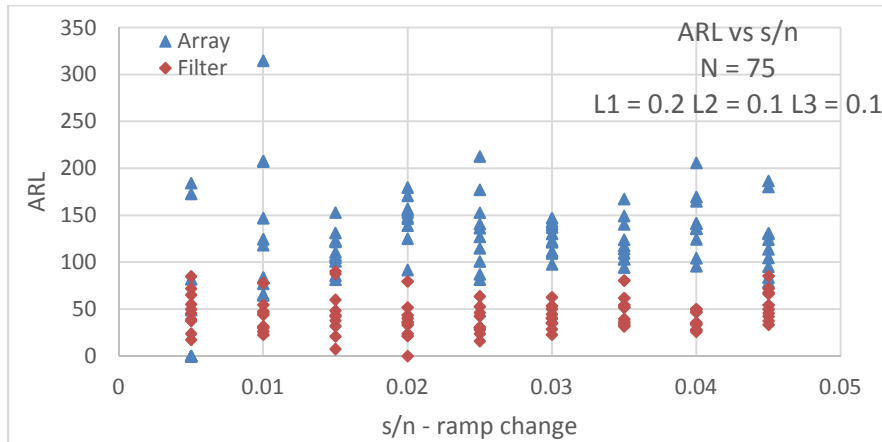


Figure 4.21: ARL vs step to noise ratio for N=75 and $\lambda_1=0.2$, $\lambda_2=0.1$, $\lambda_3=0.1$ for a ramp test

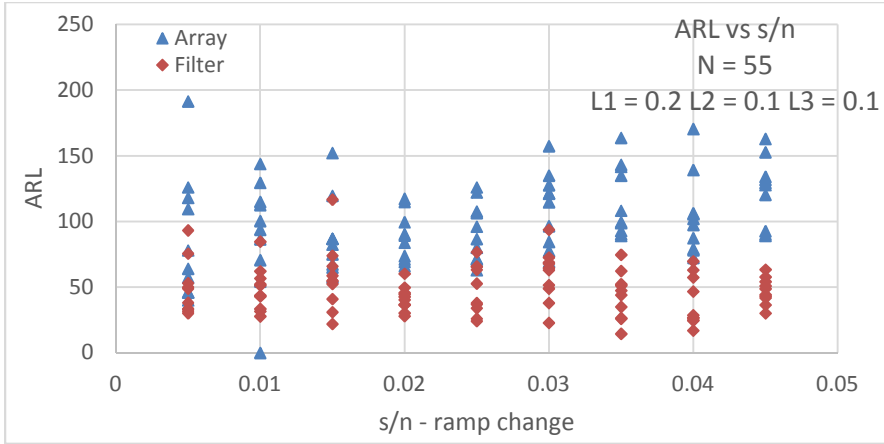


Figure 4.22: ARL vs step to noise ratio for N=55 and $\lambda_1=0.2$, $\lambda_2=0.1$, $\lambda_3=0.1$ for a ramp test

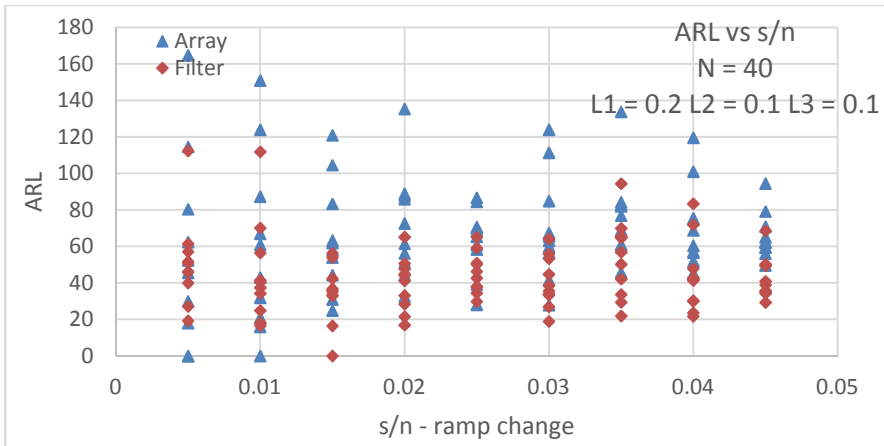


Figure 4.23: ARL vs step to noise ratio for N=40 and $\lambda_1=0.2$, $\lambda_2=0.1$, $\lambda_3=0.1$ for a ramp test

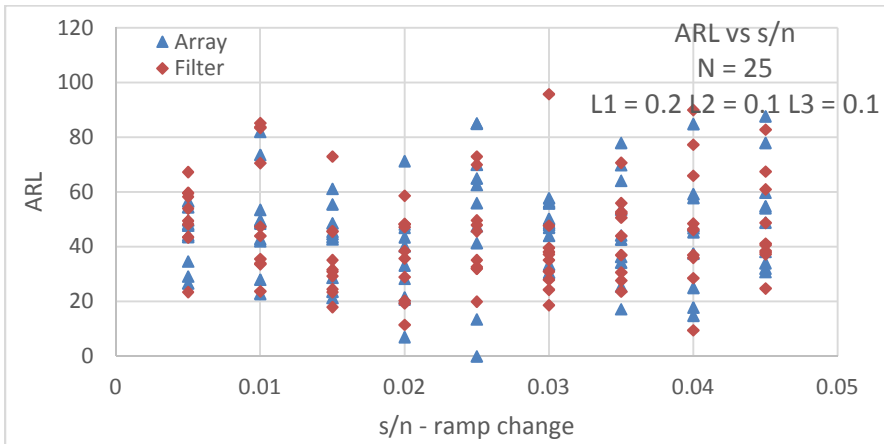


Figure 4.24: ARL vs step to noise ratio for N=25 and $\lambda_1=0.2$, $\lambda_2=0.1$, $\lambda_3=0.1$ for a ramp test

For $\lambda_1 = 0.2, \lambda_2 = 0.1, \lambda_3 = 0.1$, the filter ARL values due to change from steady state to transient state are comparable to array ARL values when $N = 25$. Fig (4.25) represents the corresponding T-II error plot. The array approach shows slightly better performance compared to the filter approach. Fig (4.25) highlights that, the performance based on probability of T-II error of both the methods for a ramp test is similar for lower signal to noise ratio and a smaller holding window. However, for higher s/n ratio, the array approach shows improved performance and more sensitivity to TS detection.

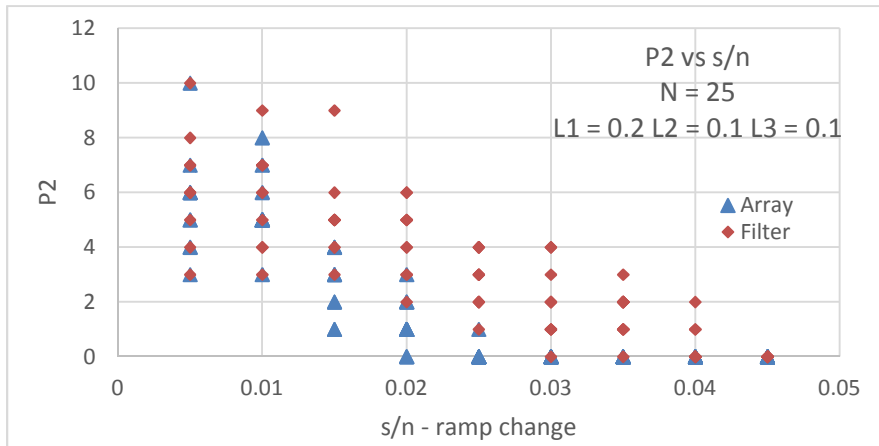


Figure 4.25: T-II error vs signal to noise ratio for $N=25$ and $\lambda_1=0.2, \lambda_2=0.1, \lambda_3=0.1$ for a ramp test

2. Figs (4.26) to (4.29) show the comparison for ramp changes for the sequence of lambda values, $\lambda_1 = 0.1, \lambda_2 = 0.1, \lambda_3 = 0.1$.

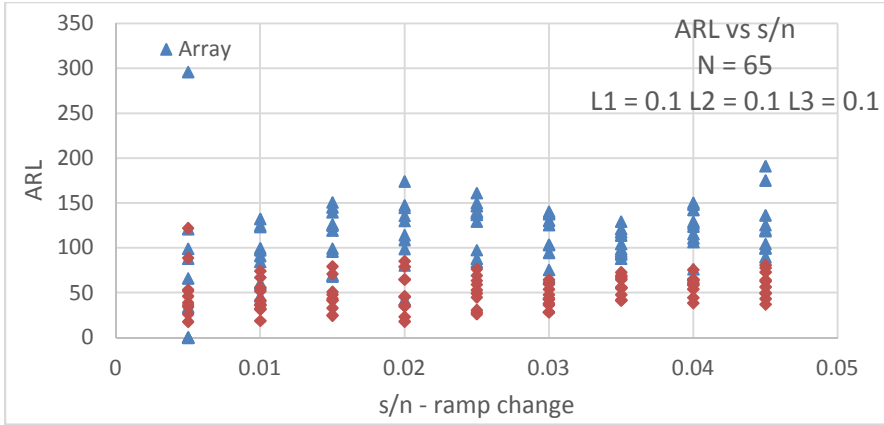


Figure 4.26: ARL vs step to noise ratio for $N=65$ and $\lambda_1=0.1$, $\lambda_2=0.1$, $\lambda_3=0.1$ for a ramp test

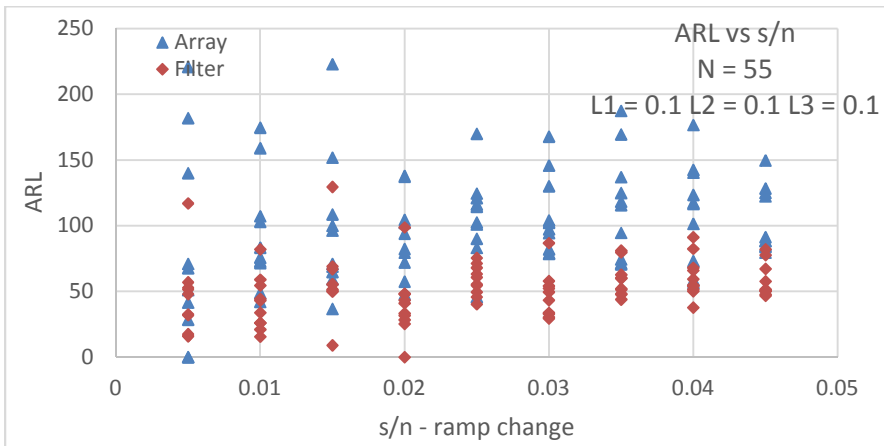


Figure 4.27: ARL vs step to noise ratio for $N=55$ and $\lambda_1=0.1$, $\lambda_2=0.1$, $\lambda_3=0.1$ for a ramp test

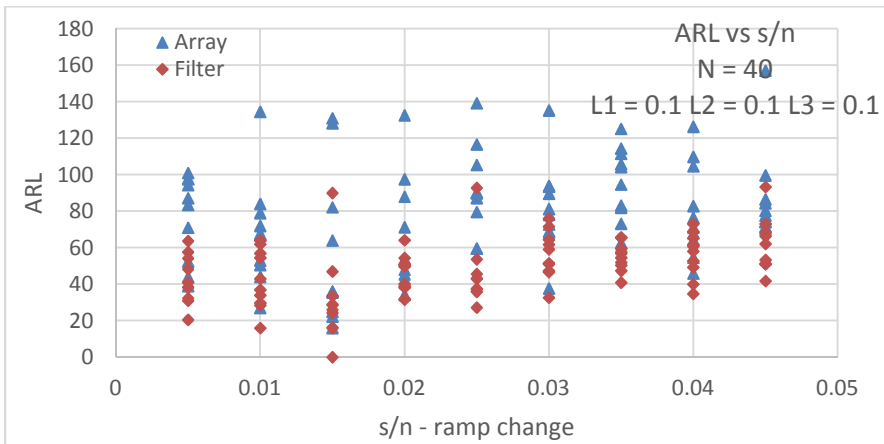


Figure 4.28: ARL vs step to noise ratio for $N=40$ and $\lambda_1=0.1$, $\lambda_2=0.1$, $\lambda_3=0.1$ for a ramp test

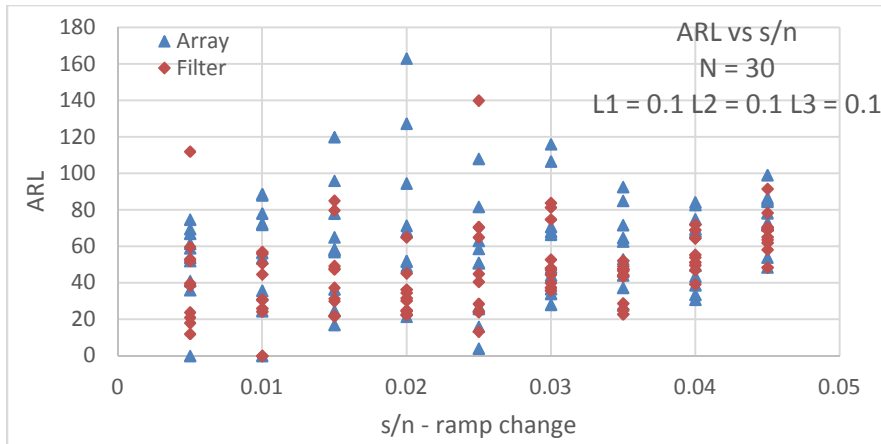


Figure 4.29: ARL vs step to noise ratio for $N=30$ and $\lambda_1=0.1$, $\lambda_2=0.1$, $\lambda_3=0.1$ for a ramp test

For $\lambda_1 = 0.1$, $\lambda_2 = 0.1$, $\lambda_3 = 0.1$, the ARL for the array approach and the filter approach are equivalent for $N = 30$. Fig (4.30) highlights that conclusions that both the approaches show comparable performances with respect to T-II error for this sequence of N and lambda values.

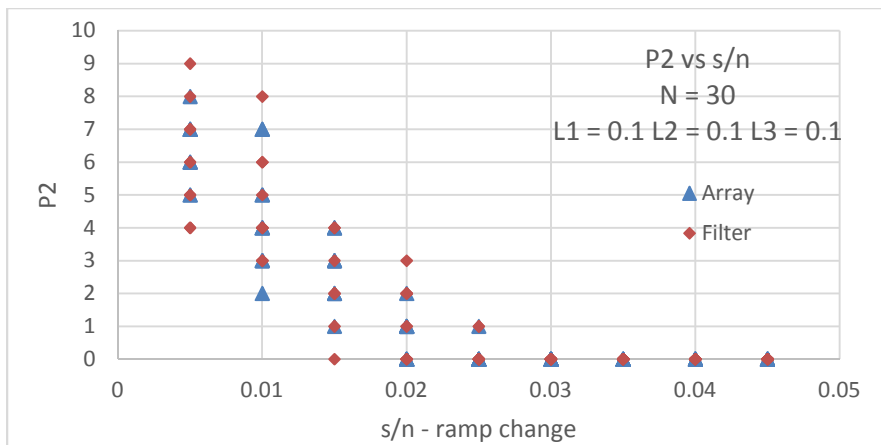


Figure 4.30: T-II error vs signal to noise ratio for $N=30$ and $\lambda_1=0.1$, $\lambda_2=0.1$, $\lambda_3=0.1$ for a ramp test

3. Figs (4.31) to (4.34) show the comparison of the two approaches for mid-size holding window with lambda values, $\lambda_1 = 0.1$, $\lambda_2 = 0.05$, $\lambda_3 = 0.05$.

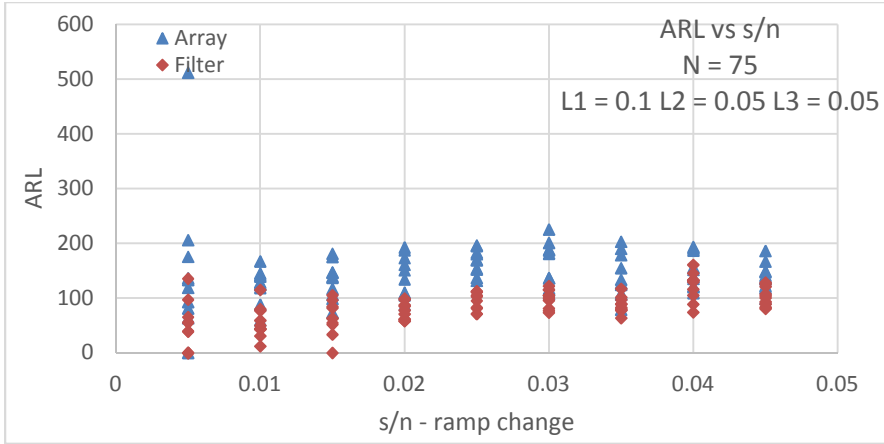


Figure 4.31: ARL vs step to noise ratio for $N=75$ and $\lambda_1=0.1$, $\lambda_2=0.05$, $\lambda_3=0.05$ for a ramp test

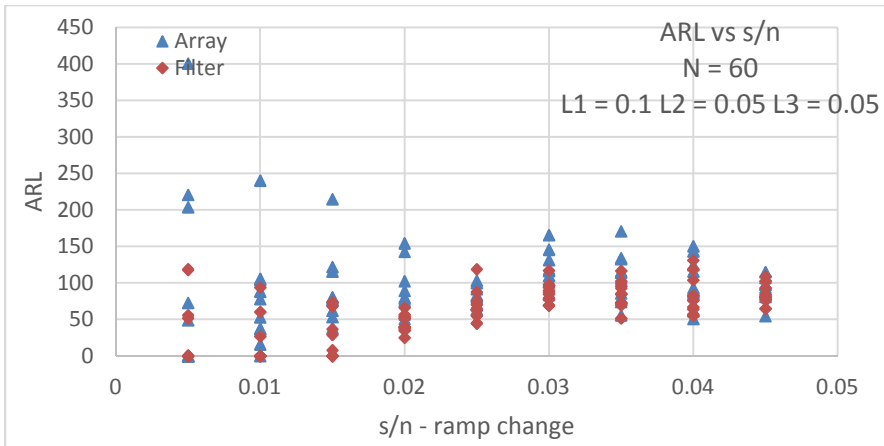


Figure 4.32: ARL vs step to noise ratio for $N=60$ and $\lambda_1=0.1$, $\lambda_2=0.05$, $\lambda_3=0.05$ for a ramp test

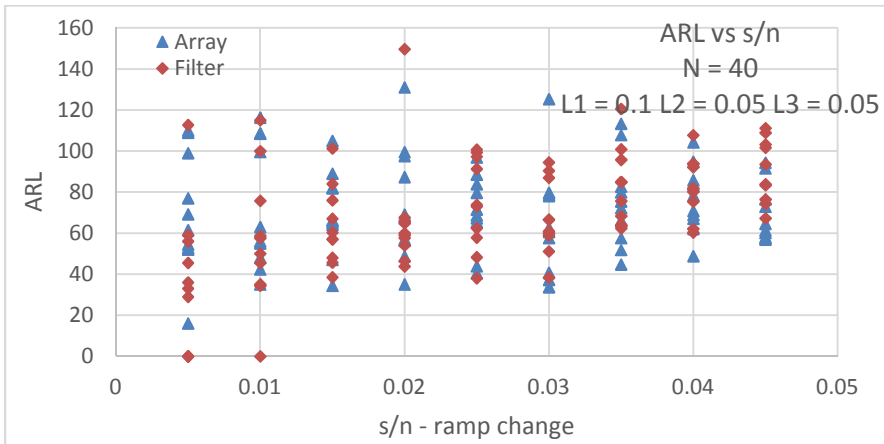


Figure 4.33: ARL vs step to noise ratio for $N=40$ and $\lambda_1=0.1$, $\lambda_2=0.05$, $\lambda_3=0.05$ for a ramp test

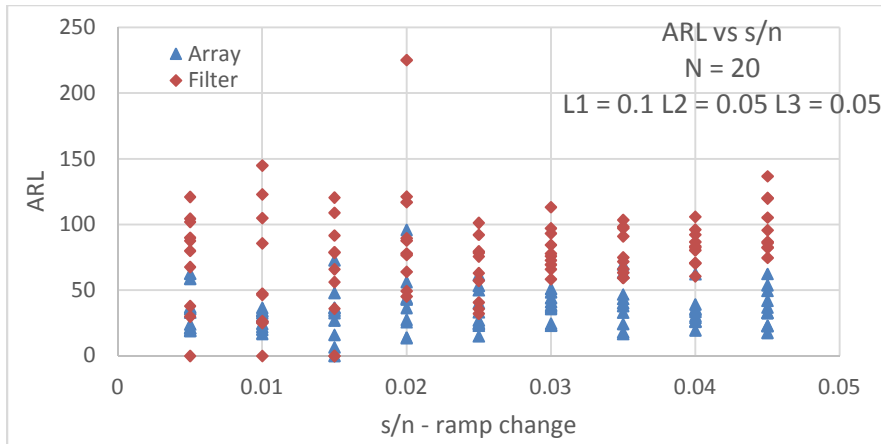


Figure 4.34: ARL vs step to noise ratio for $N=20$ and $\lambda_1=0.1$, $\lambda_2=0.05$, $\lambda_3=0.05$ for a ramp test

Fig (4.35) illustrates similar results as observed in previous set of lambda values. This sequence concludes that for smaller as well as mid-size holding window, array approach shows better sensitivity with respect to transient state detection for lower slope of the ramp changes in process signal.

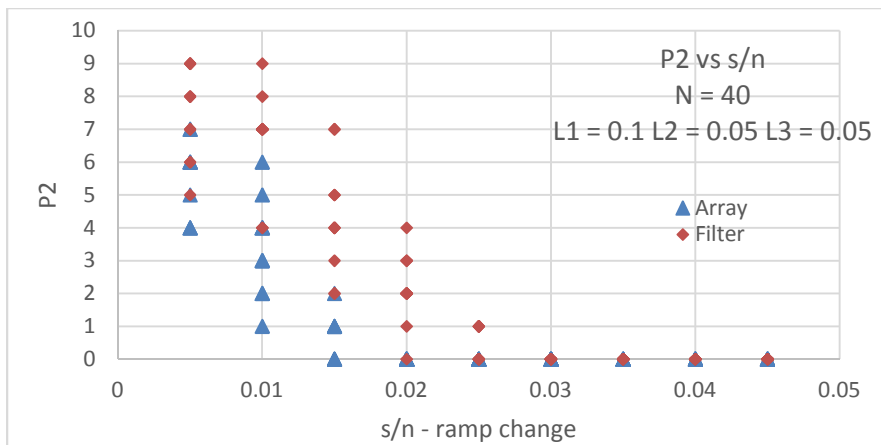


Figure 4.35: T-II error vs signal to noise ratio for $N=40$ and $\lambda_1=0.1$, $\lambda_2=0.05$, $\lambda_3=0.05$ for a ramp test

4. Another sequence of lambda values, $\lambda_1 = 0.05$, $\lambda_2 = 0.05$, $\lambda_3 = 0.05$ is used to be confident about the findings of the previous sequences of lambdas used for ramp changes. This sequence represents a larger holding window compares to the previous sequences.

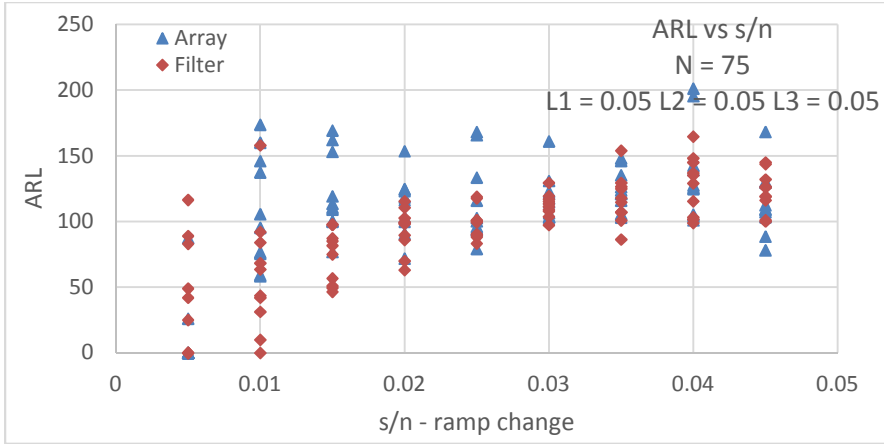


Figure 4.36: ARL vs step to noise ratio for $N=75$ and $\lambda_1=0.05$, $\lambda_2=0.05$, $\lambda_3=0.05$ for a ramp test

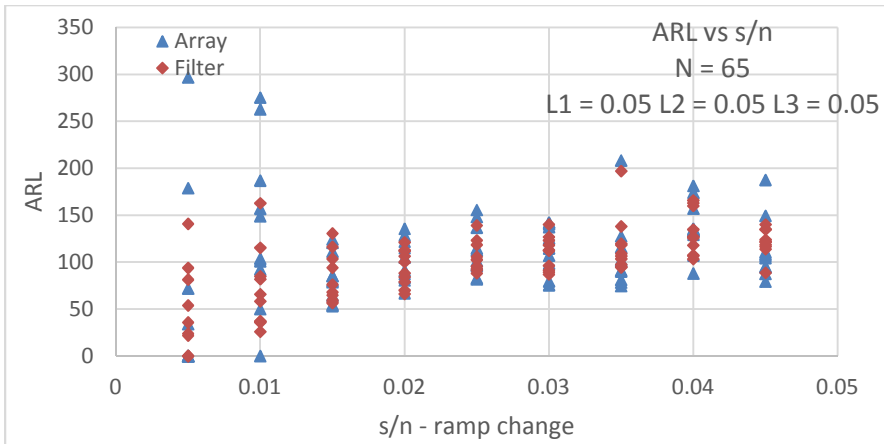


Figure 4.37: ARL vs step to noise ratio for $N=65$ and $\lambda_1=0.05$, $\lambda_2=0.05$, $\lambda_3=0.05$ for a ramp test

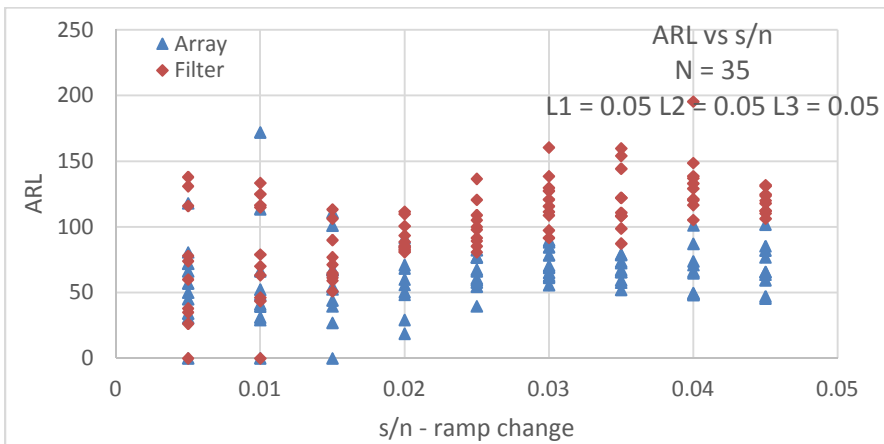


Figure 4.38: ARL vs step to noise ratio for $N=35$ and $\lambda_1=0.05$, $\lambda_2=0.05$, $\lambda_3=0.05$ for a ramp test

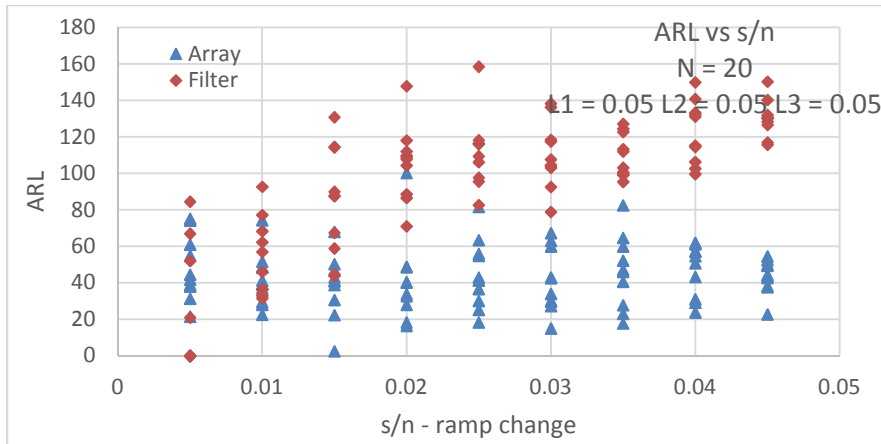


Figure 4.39: ARL vs step to noise ratio for $N=20$ and $\lambda_1=0.05$, $\lambda_2=0.05$, $\lambda_3=0.05$ for a ramp test

Fig (4.40) represents the data for a larger sample window for a ramp test. For the same ARL, the two methods show almost similar performance with respect to T-II error.

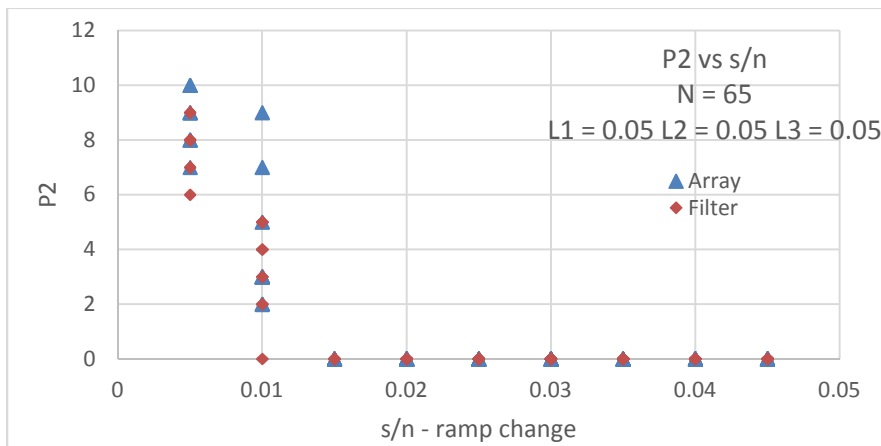


Figure 4.40: T-II error vs signal to noise ratio for $N=65$ and $\lambda_1=0.05$, $\lambda_2=0.05$, $\lambda_3=0.05$ for a ramp test

Figs (4.25), (4.30), (4.35), (4.40) highlight that, for a smaller holding window, both the approaches are unable to determine change in steady state if the s/n is comparable to noise variation. However, for larger slope of the ramp the array approach is better for detecting transient state. For a mid-size holding window, the array approach is better than the filter approach in detecting transient state at lower slope of the ramp changes. On the other hand, if the holding window is large enough or the slope of the

ramp is large enough, then both approaches successfully detect transient state and their performance is comparable.

4.2 Summary of comparison based on performance with respect to T-II error

Lambda sequence	N value for equivalent ARL and T-I error	Conclusions on the performances of both approaches
Step Change		
$\lambda_1 = 0.2, \lambda_2 = 0.1, \lambda_3 = 0.1$	35	The array approach shows better sensitivity towards detection of transient state compared to the filter approach.
$\lambda_1 = 0.1, \lambda_2 = 0.1, \lambda_3 = 0.1$	40	The array approach has less probability of showing T-II error compared to the filter approach.
$\lambda_1 = 0.1, \lambda_2 = 0.05, \lambda_3 = 0.05$	70	The array approach shows better performance with respect to T-II error for lower values of s/n ratio for a larger holding window.
$\lambda_1 = 0.05, \lambda_2 = 0.05, \lambda_3 = 0.05$	75	For larger holding window, both the approaches show comparable performances. The array approach is better when the s/n ratio is small.
Ramp Change		
$\lambda_1 = 0.2, \lambda_2 = 0.1, \lambda_3 = 0.1$	25	For lower s/n ratio, both approaches show similar performances. The array approach shows improved

		performance with respect to transient state detection for higher values of s/n ratio
$\lambda_1 = 0.1, \lambda_2 = 0.1, \lambda_3 = 0.1$	30	For this set of lambda and N values, both the approaches show comparable performances with respect to T-II error for equivalent ARL and T-I error.
$\lambda_1 = 0.1, \lambda_2 = 0.05, \lambda_3 = 0.05$	40	For a mid-size holding window, the array approach shows more sensitivity to transient state detection for lower slope of the ramp change. For a higher change in slope, both approach detect the transient state effectively.
$\lambda_1 = 0.05, \lambda_2 = 0.05, \lambda_3 = 0.05$	65	For a larger holding window, both approaches mostly determine change in steady state effectively and show comparable results.

Table 4.1: Summary of comparison of the filter approach and the array approach for different holding windows for ramp as well as step change

4.3 Comparison based on functioning of two approaches

Performance of the two approaches based on the ability to effectively detect change in steady state is very important parameter for comparison of the array and filter approach as it is the primary function of the two approaches. However, there are also some in-use issues which must also be considered for smooth execution of the approaches in industry. Following are some of the in-use parameters which can be considered while selecting between the two approaches:

4.3.1 Ease of Understanding

The steady state identifiers are to be used by process operators in order to sequence a stage in an experimental trial. The concept of the array method uses a conventional approach to evaluate standard deviation and variance. Basic statistical knowledge is enough to understand the array method algorithm. However, the concept of three first-order filters used to evaluate variance is complicated for many to understand. Also, the holding window of fixed length N is easier to grasp than the exponentially weighted infinite window or the interpretation of three lambda values influencing window size. The array approach is therefore comparatively easier to understand for many than the filter approach.

4.3.2 Storage

The filter approach has less than 10 storage variables – the process variable, X_i , and its previous value, X_{i-1} , the filtered process variable, X_f , the filtered variance, v_f^2 , the filtered value of sequential variance, δ_f^2 , the filter tuning factors, λ_1 , λ_2 , λ_3 , and the R-statistic value, R.

The array approach, on the other hand, stores all the N values in the holding window in an array in addition to the other variables – sum1, sum2, sum3 used to calculate the numerator and denominator of the R-statistic ratio, and the R-statistic value, R. Lower values of N will increase the probability of the errors in the process and N values of about 50 were required to have low T-II error. So, there is a high chance that the array approach will use greater storage compared to filter approach.

4.3.3 Computational Burden

Computationally, the filter approach has an advantage as it uses three simple first-order equations to evaluate two types of variances. On the other hand, the array approach has to evaluate square of number of terms, sum of squares, as well as sum of squared deviation. On an average, for equivalent T-II and ARL values, filter approach took half the amount of time to execute compared to the array approach.

4.3.4 Ease of Tuning

The array approach has only one tuning factor, N , and is less complicated to tune. Whereas the filter approach has three tuning factors, λ_1 , λ_2 , λ_3 , and can be difficult to tune if the influence of the each filter on the algorithm as well as the holding window is unknown.

CHAPTER V

CONCLUSIONS AND RECOMMENDATIONS

In order to evaluate the array approach and the filter approach for detecting steady state, a noisy process signal was simulated. Step and ramp changes have been simulated in the process signal and multiple runs had been carried in order to collect data to evaluate performance of the two approaches based on the undesirables in the statistical methods. The strength of following conclusions is based on experimental tests using simulated step and hold, as well as ramp change, with a Gaussian independent noise. The data was analyzed and following conclusions have been reached.

5.1 Conclusions

1. For the step change in the process signal, the array approach is more sensitive to transient state detection compared to the filter approach.
2. For ramp changes in the process signal, the array approach shows improved performance with respect to detection of transient state for smaller slope of the ramp if the holding window is mid-size.
3. For smaller window size, both approaches show comparable and not very effective

performance for smaller slope of the ramp. However, the array approach shows improved performance at higher slope of the ramp changes.

4. For larger slope of ramp change and bigger sample size, both the array approach and the filter approach show comparable performances.
5. The array approach is easier to understand compared to the filter approach
6. The array approach is more convenient in terms of tuning as it uses only one tuning factor, compared to filter approach which uses three tuning factors.
7. The filter approach uses less storage memory compared to array approach
8. The filter approach is computationally less burdensome than the array approach.

5.2 Recommendations for Future Work

1. As the sample size used increases, both approaches show better performance. However, more amount of computation and storage memory is required in that case. Also, as the sample size increases, the ARL might also increase. If a smaller sample size is used, ARL might decrease, however, the probability of T-II error will then increase. Optimization technique can be used to find optimum set of parameters in order to get best performance from the steady state identifier.
2. The above study can be verified by evaluating the two approaches on real time data instead of simulated data.
3. The two approaches can be compared on process signals other than step change or ramp change like sinusoidal function, or other non-conventional functions.

REFERENCES

1. Cao, S. and R.R. Rhinehart, *An efficient method for on-line identification of steady state*. Journal of Process Control, 1995. **5**(6): p. 363-374.
2. Bethea, R.M. and R.R. Rhinehart, *Applied Engineering Statistics*, 1991, Marcel Dekker, Inc.: New York.
3. Brown, P.R. and R.R. Rhinehart, *Demonstration of a method for automated steady-state identification in multivariable systems*. Hydrocarbon Processing, 2000. **79**(9): p. 79-83.
4. Katterhenry, P. and R.R. Rhinehart, *Use a virtual employee to trigger a sequence of conditions*. Control for the process industries, 2001. **XIV**(10): p. 53-55.
5. Szela, J.T. and R.R. Rhinehart, *A virtual employee to trigger a sequence of conditions*. Journal of Process Analytical Chemistry, 2003. **8**(1).
6. Huang, T and R.R Rhinehart, *Steady State and Transient State Identification for flow rate on a Pilot Scale Absorption Column*, Proceedings of 2013 American Control Conference, 2013, Washington D.C.
7. Iyer, M.S. and R.R. Rhinehart. *A novel method to stop neural network training*. in *American Control Conference, 2000. Proceedings of the 2000*. 2000.
8. Padmanabhan, V. and R.R. Rhinehart. *A novel termination criterion for optimization*. in *American Control Conference, 2005. Proceedings of the 2005*. 2005.
9. Shrowti, N.A., K.P. Vilankar, and R.R. Rhinehart, *Type-II critical values for a steady-state identifier*. Journal of Process Control, 2010. **20**(7): p. 885-890.
10. Bhat, S.A. and D.N. Saraf, *Steady-State Identification, Gross Error Detection, and Data Reconciliation for Industrial Process Units*. Industrial & Engineering Chemistry Research, 2004. **43**(15): p. 4323-4336.

11. Cao, S. and R.R. Rhinehart, *Critical values for a steady-state identifier*. Journal of Process Control, 1997. 7(2): p. 149-152.
12. Box, G.E.P and M.E. Muller, *A Note on the Generation of Random Normal Deviates*, The Annals of Mathematical Statistics, 1958. 29(2): p. 610-611.
13. Vasyutynskyy, V, *Passive monitoring of control loops in building automation*, Technical Report, Technische Universitat Dresden, Department of Computer Science, 2005.
14. Le Roux, G. A. C, B. F. Santoro, F. F. Sotelo, M. Teissier, X. Joulia, *Improving steady state identification*, Proceedings of the 18th European Symposium on Computer Aided Process Engineering – ESCAPE 18, Elsevier B. V./Ltd., London, 2008.
15. Jiang, T. B, Chen, X. He, Paul Stuart, *Application of steady state detection method based on wavelet transform*, Computers and Chemical Engineering, 2003, 27: p. 569-578.
16. Rhinehart, R. R, *Tutorial: Automated Steady and Transient State Identification in Noisy Processes*, Proceedings of the 2013 American Control Conference, Washington DC, June, 2013.

APPENDICES

This section lists the excel vba code for the simulation system. The comments in the code mention array approach as clock approach. Initially the process was visualized as a storage involving clock structure where the pointer points to different location each time a new value is used. This can be easily carried out using arrays and so the method was called array approach.

1. Step Change

' SSID Filter and Clock Approaches

' Sarvesh Gore

' Edited from code of Dr. R. Russell Rhinehart 12 Sept 2012

' Simulator to generate noisy data and apply the r-statistic method to

' determine probable Steady State and Transient State.

'

' average divided by that estimated by sequential data deviations. But the clock approach to

' data processing greatly reduces the computational burden.

'Global counter

Public globalcount As Single

' Declare variables in Main Subroutine

Dim i 'sampling counter

Dim true_value 'true process value

Dim measurement 'measurement - true value plus noise

' Declare variables in Clock approach

Dim x(100) 'store 100 data for a maximum window length of N=100

Dim dx2(100) 'store the $(x - x_{old})^2$ values

Dim N >window length, number of samples

Dim counter_prior >window index

Dim counter >window index

Dim counter_next >window index

Dim sum1 'sum of x^2

Dim sum2 'sum of x

Dim sum3 'sum of $(x-xold)^2$

Dim R_Clock 'r-statistic for the clock approach

Dim ID_Clock '=1 if confidently at SS, = 0 if confidently in a TS

' Declare variables in Filter approach

Dim l1 'lambda 1

Dim c11 '(1-l1) the complement to lambda 1

Dim l2 'lambda 2

Dim c12 '(1-l2) the complement to lambda 2

Dim l3 'lambda 3

Dim c13 '(1-l3) the complement to lambda 3

Dim measurement_old 'old measurment value

Dim xf 'filtered measurement value

Dim nu2f 'filtered numerator measure of the variance

Dim delta2f 'filtered denominator measure of the variance

Dim R_Filter 'r-statistic for the filter approach

Dim ID_Filter '=1 if confidently at SS, = 0 if confidently in a TS

' Declare variables for step change

Dim a(1 To 10150) As Single 'array for doing alternate step changes

Dim b(1 To 10150) As Single 'array for doing alternate step changes

' Declare variables for arl

'for clock approach

Dim carl As Single 'counter variable

Dim clkarl(10000) As Single 'array for final arl value

'for filter approach

Dim farl As Single 'counter variable

Dim filarl(10000) As Single 'array for final arl value

' Declare variables for p2 error

'for clock approach

```

Dim clkcountp2 As Single      'counter variable

Dim clkp2 As Single

Dim nosamclk As Single      'no of non zero sample points for arl of clock
'for filter approach

Dim fcountp2 As Single      'counter variable

Dim fp2 As Single

Dim nosamfil As Single      'no of non zero sample points for arl of filter

Dim ms(20) As Single        'array for storing step/ramp changes to be made

Dim r As Single             'counter variable

Sub mainloop()

For r = 1 To 16 Step 1      'runs the main loop for different values of step/ramp
changes

ms(r) = Sheet1.Cells(r + 3, 30).Value

For s = 1 To 10 Step 1      'runs each simulation 10 times

Call main

globalcount = globalcount + 1

```

Next s

Next r

End Sub

,

,

Sub main()

For j = 500 To 9500 Step 1000

a(j) = j

Next j

For k = 1000 To 9000 Step 1000

b(k) = k

Next k

true_value = 5

Randomize 'initialize the random number generator with the computer clock
time

For i = 1 To 10000

 'If i = a(i) Then true_value = true_value + 0

 'If i = b(i) Then true_value = true_value - 0

If i = a(i) Then true_value = true_value + ms(r)

If i = b(i) Then true_value = true_value - ms(r)

measurement = true_value + 0.5 * Sqr(-2 * Log(Rnd())) * Sin(2 * 3.14159 * Rnd()) 'add

noise - Box-Mueller method

Call SSID_Clock

Call SSID_Filter

Cells(i + 5, 1) = i 'output results

Cells(i + 5, 2) = true_value

Cells(i + 5, 3) = measurement

Cells(i + 5, 4) = R_Clock

Cells(i + 5, 5) = ID_Clock

Cells(i + 5, 6) = R_Filter

Cells(i + 5, 7) = ID_Filter

Cells(i + 5, 9) = sum2 / N

Cells(i + 5, 11) = Sqr((sum1 - (sum2 ^ 2) / N) / (N - 1))

Cells(i + 5, 14) = Sqr(sum3 / (N - 1) / 2)

Next i

Call ar1

Call p2

DoEvents 'unnecessary statement to update the display.

'globalcount = globalcount + 1

End Sub

,

Sub SSID_Clock()

If i = 1 Then 'initialize on first call

N = 75 'for equivalence in window observation N should be about
3.5/smaller_lambda.

For j = 1 To N

x(j) = 0

dx2(j) = 0

Next j

counter = 1

x(counter) = measurement

sum1 = x(counter) ^ 2

sum2 = x(counter)

$$\text{sum3} = 0$$

Else

$$\text{counter_prior} = \text{counter}$$

$$\text{counter} = \text{counter} + 1$$

$$\text{If counter} > N \text{ Then counter} = 1$$

$$\text{counter_next} = \text{counter} + 1$$

$$\text{If counter_next} > N \text{ Then counter_next} = 1$$

$$\text{sum1} = \text{sum1} - x(\text{counter})^2$$

$$\text{sum2} = \text{sum2} - x(\text{counter})$$

$$\text{sum3} = \text{sum3} - dx2(\text{counter_next})$$

$$x(\text{counter}) = \text{measurement}$$

$$dx2(\text{counter}) = (x(\text{counter}) - x(\text{counter_prior}))^2$$

$$\text{sum1} = \text{sum1} + x(\text{counter})^2$$

$$\text{sum2} = \text{sum2} + x(\text{counter})$$

$$\text{sum3} = \text{sum3} + dx2(\text{counter})$$

$$R_Clock = 2 * (N - 1) * (\text{sum1} - (\text{sum2}^2) / N) / \text{sum3} / N$$

$$\text{If } R_Clock > 1.6172 \text{ Then ID_Clock} = 0 \quad \text{'using 2.5 as upper critical value to}$$

reject SS

```
        If R_Clock < 0.9179 Then ID_Clock = 1      'using 1.0 as lower critical value to  
accept SS
```

```
    End If
```

```
End Sub
```

```
,
```

```
,
```

```
Sub SSID_Filter()
```

```
    If i = 1 Then      'initialize values on first call
```

```
        l1 = 0.05
```

```
        l2 = 0.05
```

```
        l3 = 0.05
```

```
        cl1 = 1 - l1
```

```
        cl2 = 1 - l2
```

```
        cl3 = 1 - l3
```

```
        xf = 0
```

```
        nu2f = 0
```

```
        delta2f = 0
```

```
        measurement_old = 0
```

ID_Filter = 0.5

End If

$nu2f = 12 * (measurement - xf) ^ 2 + c12 * nu2f$

$xf = 11 * measurement + c11 * xf$

$delta2f = 13 * (measurement - measurement_old) ^ 2 + c13 * delta2f$

measurement_old = measurement

$R_Filter = (2 - 11) * nu2f / delta2f$

If R_Filter > 2.0662 Then ID_Filter = 0 'using 2.5 as upper critical value to
reject SS

If R_Filter < 0.9093 Then ID_Filter = 1 'using 1.0 as lower critical value to
accept SS

End Sub

,

,

Sub arl()

'sub to calculate arl for clock method

For i = 500 To 9500 Step 500

carl = 0

For j = i To (i + 400) Step 1

 If Sheet1.Cells(j, 5).Value = 0 Then carl = carl + 1

Next j

 clkarl(i) = carl

 ' Sheet1.Cells(((i / 500) + 4), 24).Value = clkarl(i)

Next i

'determining no of non-zero sample points out of 19 step changes for clock method

nosamclk = 19

For i = 1 To 19 Step 1

 If clkarl(i * 500) = 0 Then nosamclk = nosamclk - 1

Next i

'determining if there was a p2 error

For i = 500 To 9500 Step 500

 clkcountp2 = 0

 For j = (i - 20) To (i + 60) Step 1

 If Sheet1.Cells(j + 5, 5).Value = 1 Then clkcountp2 = clkcountp2 + 1

 Next j

If clkcountp2 >= 78 Then

clkp2 = clkp2 + 1

clkarl(i) = 0

End If

Next i

'outputting the ARL values at each step change

For k = 1 To 19 Step 1

Sheet1.Cells(k + 5, 24).Value = clkarl(k * 500)

Next k

'calculating average of all arl value

countclkav = 0

For i = 1 To 19 Step 1

countclkav = countclkav + clkarl(i * 500)

Next i

If nosamclk <> 0 Then

clkarlav = countclkav / nosamclk

Else

clkarlav = 0

End If

'sub to calculate arl for filter method

For i = 500 To 9500 Step 500

farl = 0

For j = i To (i + 400) Step 1

 If Sheet1.Cells(j, 7).Value = 0 Then farl = farl + 1

Next j

 filarl(i) = farl

Next i

'determining no of non-zero sample points out of 19 step changes for filter method

nosamfil = 19

For i = 1 To 19 Step 1

 If filarl(i * 500) = 0 Then nosamfil = nosamfil - 1

Next i

'determining if there was a p2 error

For i = 500 To 9500 Step 500

fcountp2 = 0

For j = (i - 20) To (i + 60) Step 1

 If Sheet1.Cells(j + 5, 7).Value = 1 Then fcountp2 = fcountp2 + 1

Next j

 If fcountp2 >= 78 Then

 fp2 = fp2 + 1

 filarl(i) = 0

 End If

Next i

 Sheet1.Cells(6, 27).Value = fp2

'outputting the ARL values at each step change

For k = 1 To 19 Step 1

 Sheet1.Cells(k + 5, 26).Value = filarl(k * 500)

Next k

'calculating average of all arl value

countfilav = 0

For i = 1 To 19 Step 1

countfilav = countfilav + filarl(i * 500)

Next i

If nosamfil <> 0 Then

filarlav = countfilav / nosamfil

Else

filarlav = 0

End If

Sheet1.Cells(8 + globalcount, 22).Value = filarlav

End Sub

Sub p2()

'sub to calculate p2 error for clock

clkp2 = 0

For i = 500 To 9500 Step 500

clkcountp2 = 0

For j = (i - 20) To (i + 60) Step 1

 If Sheet1.Cells(j + 5, 5).Value = 1 Then clkcountp2 = clkcountp2 + 1

Next j

 If clkcountp2 >= 78 Then clkp2 = clkp2 + 1

Next i

 Sheet1.Cells(6, 25).Value = clkp2

Sheet1.Cells(8 + globalcount, 21).Value = clkp2

'sub to calculate p2 error for filter

fp2 = 0

For i = 500 To 9500 Step 500

 fcountp2 = 0

 For j = (i - 20) To (i + 60) Step 1

 If Sheet1.Cells(j + 5, 7).Value = 1 Then fcountp2 = fcountp2 + 1

 Next j

 If fcountp2 >= 78 Then fp2 = fp2 + 1

Next i

```
Sheet1.Cells(6, 27).Value = fp2
```

```
Sheet1.Cells(8 + globalcount, 23).Value = fp2
```

End Sub

2. Ramp Change

' SSID Filter and Clock Approaches

' Sarvesh Gore

' Edited from code of Dr. R. Russell Rhinehart 12 Sept 2012

'

' Simulator to generate noisy data and apply the r-statistic method to

' determine probable Steady State and Transient State.

'

' Compares both the filter (Cao and Rhinehart 1995) and clock approaches to the r-statistic

' The clock approach is true to the Crowe et al. 1956 variance ratio estimate as the deviations from

' average divided by that estimated by sequential data deviations. But the clock approach to

' data processing greatly reduces the computational burden.

'Global counter

Public globalcount As Single

Public vari1 As Single

Public vari2 As Single

Public vari3 As Single

Public vari4 As Single

' Declare variables in Main Subroutine

Dim i 'sampling counter

Dim true_value 'true process value

Dim measurement 'measurement - true value plus noise

' Declare variables in Clock approach

Dim x(100) 'store 100 data for a maximum window length of N=100

Dim dx2(100) 'store the $(x - xold)^2$ values

Dim N >window length, number of samples

Dim counter_prior >window index

Dim counter >window index

Dim counter_next >window index

Dim sum1 'sum of x^2

Dim sum2 'sum of x

Dim sum3 'sum of $(x-xold)^2$

Dim R_Clock 'r-statistic for the clock approach

Dim ID_Clock '=1 if confidently at SS, = 0 if confidently in a TS

' Declare variables in Filter approach

Dim l1 'lambda 1

Dim cl1 '(1-11) the complement to lambda 1
Dim l2 'lambda 2
Dim cl2 '(1-12) the complement to lambda 2
Dim l3 'lambda 3
Dim cl3 '(1-13) the complement to lambda 3
Dim measurement_old 'old measuerment value
Dim xf 'filtered measurement value
Dim nu2f 'filtered numerator measure of the variance
Dim delta2f 'filtered denominator measure of the variance
Dim R_Filter 'r-statistic for the filter approach
Dim ID_Filter '=1 if confidently at SS, = 0 if confidently in a TS

' Declare variables for step change

Dim a(1 To 20150) As Single 'array for doing alternate step changes

Dim b(1 To 20150) As Single 'array for doing alternate step changes

' Declare variables for arl

'for clock approach

Dim carl As Single 'counter variable

Dim clkarl(20000) As Single 'array for final arl value

'for filter approach

Dim farl As Single 'counter variable

Dim filarl(20000) As Single 'array for final arl value

' Declare variables for p2 error

```

'for clock approach
Dim clkcountp2 As Single      'counter variable
Dim clkp2 As Single
Dim nosamclk As Single        'no of non zero sample points for arl of clock
'for filter approach
Dim fcountp2 As Single        'counter variable
Dim fp2 As Single
Dim nosamfil As Single        'no of non zero sample points for arl of filter

Dim ms(20) As Single          'array for storing step/ramp changes to be made
Dim r As Single                'counter variable

Sub mainloop()

For r = 1 To 9 Step 1          'runs simulation for different values of step/ramp
changes
ms(r) = Sheet1.Cells(r + 3, 32).Value

For s = 1 To 10 Step 1        'runs simulation multiple times in order to consider
statistical vagaries

Call main

globalcount = globalcount + 1

Next s

Next r

End Sub

```

Sub main()

For j = 1000 To 17000 Step 4000

a(j) = j

Next j

For k = 3000 To 19000 Step 4000

b(k) = k

Next k

true_value = 5

Randomize 'initialize the random number generator with the computer clock
time

For i = 1 To 20000

'Used to ramp up using desired slope

 If i = a(i) Then

 vari1 = a(i)

 vari2 = a(i) + 1000

 End If

 If i > vari1 And i < vari2 Then true_value = true_value + ms(r)

 If true_value = (5 + (ms(r) * 1000)) Then true_value = true_value + 0

'Used to ramp down using desired slope

If i = b(i) Then

vari3 = b(i)

vari4 = b(i) + 1000

End If

If i > vari3 And i < vari4 Then true_value = true_value - ms(r)

If true_value = 5 Then true_value = true_value + 0

measurement = true_value + 0.5 * Sqr(-2 * Log(Rnd())) * Sin(2 * 3.14159 * Rnd())
'add noise - Box-Mueller method

Call SSID_Clock

Call SSID_Filter

Cells(i + 5, 1) = i 'output results

Cells(i + 5, 2) = true_value

Cells(i + 5, 3) = measurement

Cells(i + 5, 4) = R_Clock

Cells(i + 5, 5) = ID_Clock

Cells(i + 5, 6) = R_Filter

Cells(i + 5, 7) = ID_Filter

Cells(i + 5, 9) = sum2 / N

Cells(i + 5, 11) = Sqr((sum1 - (sum2 ^ 2) / N) / (N - 1))

Cells(i + 5, 14) = Sqr(sum3 / (N - 1) / 2)

Next i

Call arlts ' Arl when process goes from steady state to transient
state(not used anywhere)

Call arlss ' Arl when process goes from transient state to steady state

Call p21 ' p2 error for ramp change

DoEvents 'unnecessary statement to update the display.

End Sub

Sub SSID_Clock()

 If i = 1 Then 'initialize on first call

 N = 75 'for equivalence in window observation N should be about
3.5/smaller_lambda.

 For j = 1 To N

 x(j) = 0

 dx2(j) = 0

 Next j

 counter = 1

 x(counter) = measurement

 sum1 = x(counter) ^ 2

 sum2 = x(counter)

 sum3 = 0

 Else

 counter_prior = counter

 counter = counter + 1

 If counter > N Then counter = 1

```

counter_next = counter + 1

If counter_next > N Then counter_next = 1

sum1 = sum1 - x(counter) ^ 2

sum2 = sum2 - x(counter)

sum3 = sum3 - dx2(counter_next)

x(counter) = measurement

dx2(counter) = (x(counter) - x(counter_prior)) ^ 2

sum1 = sum1 + x(counter) ^ 2

sum2 = sum2 + x(counter)

sum3 = sum3 + dx2(counter)

R_Clock = 2 * (N - 1) * (sum1 - (sum2 ^ 2) / N) / sum3 / N

If R_Clock > 1.6172 Then ID_Clock = 0      'using 2.5 as upper critical value to
reject SS

If R_Clock < 0.9179 Then ID_Clock = 1      'using 1.0 as lower critical value to
accept SS

End If

End Sub

,

,

Sub SSID_Filter()

If i = 1 Then      'initialize values on first call

l1 = 0.05

l2 = 0.05

l3 = 0.05

c11 = 1 - l1

```

$c12 = 1 - l2$

$c13 = 1 - l3$

$xf = 0$

$nu2f = 0$

$delta2f = 0$

$measurement_old = 0$

$ID_Filter = 0.5$

End If

$nu2f = l2 * (measurement - xf) ^ 2 + c12 * nu2f$

$xf = 11 * measurement + c11 * xf$

$delta2f = l3 * (measurement - measurement_old) ^ 2 + c13 * delta2f$

$measurement_old = measurement$

$R_Filter = (2 - 11) * nu2f / delta2f$

If $R_Filter > 2.0662$ Then $ID_Filter = 0$ 'using 2.5 as upper critical value to
reject SS

If $R_Filter < 0.9093$ Then $ID_Filter = 1$ 'using 1.0 as lower critical value to
accept SS

End Sub

Sub arlts()

'sub to calculate arl for clock method

```
For i = (2000) To (18000) Step 2000
```

```
    carl = 0
```

```
    For j = i To (i + 995) Step 1
```

```
        If Sheet1.Cells(j + 5, 5).Value = 0 Then carl = carl + 1
```

```
    Next j
```

```
        clkarl(i) = carl
```

```
Next i
```

```
'determining if there was a p2 error
```

```
For i = (1000) To (19000) Step 2000
```

```
    clkcountp2 = 0
```

```
    For j = (i - 20) To (i + 980) Step 1
```

```
        If Sheet1.Cells(j + 5, 5).Value = 1 Then clkcountp2 = clkcountp2 + 1
```

```
    Next j
```

```
        If clkcountp2 >= 998 Then
```

```
            clkp2 = clkp2 + 1
```

```
            clkarl(i) = 0
```

```
        End If
```

```
Next i
```

```
'determining no of non-zero sample points out of 19 step changes for clock method
```

```
nosamclk = 9
```

```
For i = 1 To 9 Step 1
```

```
    If clkarl(i * 2000) = 0 Then nosamclk = nosamclk - 1
```

```
Next i
```

'Calculating the average of the ARL

countclkav = 0

For i = 1 To 9 Step 1

countclkav = countclkav + clkarl(i * 2000)

Next i

If nosamclk <> 0 Then

clkarlav = countclkav / nosamclk

Else

clkarlav = 0

End If

Sheet1.Cells(8 + globalcount, 20).Value = clkarlav

'sub to calculate arl for filter method

For i = (2000) To (18000) Step 2000

farl = 0

For j = i To (i + 995) Step 1

 If Sheet1.Cells(j + 5, 7).Value = 0 Then farl = farl + 1

Next j

 filarl(i) = farl

Next i

'Determining if there was a p2 error

For i = (1000) To (19000) Step 2000

fcountp2 = 0

For j = (i - 20) To (i + 980) Step 1

 If Sheet1.Cells(j + 5, 7).Value = 1 Then fcountp2 = fcountp2 + 1

Next j

 If fcountp2 >= 998 Then

 fp2 = fp2 + 1

 filarl(i) = 0

 End If

Next i

 Sheet1.Cells(6, 27).Value = fp2

'determining no of non-zero sample points out of 19 step changes for filter method

nosamfil = 9

For i = 1 To 9 Step 1

 If filarl(i * 2000) = 0 Then nosamfil = nosamfil - 1

Next i

'calculating average arl

countfilav = 0

For i = 1 To 9 Step 1

 countfilav = countfilav + filarl(i * 2000)

Next i

If nosamfil <> 0 Then

filarlav = countfilav / nosamfil

Else

filarlav = 0

End If

Sheet1.Cells(8 + globalcount, 22).Value = filarlav

End Sub

Sub arlss()

'sub to calculate arl for clock method when process goes

For i = (1000) To (19000) Step 2000

carl = 0

For j = i To (i + 995) Step 1

 If Sheet1.Cells(j + 5, 5).Value = 1 Then carl = carl + 1

Next j

 ckarl(i) = carl

Next i

'determining no of non-zero sample points out of 19 step changes for clock method

nosamclk = 10


```
For i = 1 To 10 Step 1
    If clkarl(((i * 2000) - 1000)) = 0 Then nosamclk = nosamclk - 1
Next i
```

```
'calculating average of arl
```

```
countclkav = 0
```

```
For i = 1 To 10 Step 1
```

```
countclkav = countclkav + clkarl(((i * 2000) - 1000))
```

```
Next i
```

```
If nosamclk <> 0 Then
```

```
clkarlav = countclkav / nosamclk
```

```
Else
```

```
clkarlav = 0
```

```
End If
```

```
Sheet1.Cells(8 + globalcount, 21).Value = clkarlav
```

```
'sub to calculate arl for filter method
```

```
For i = (1000) To (19000) Step 2000
```

```
farl = 0
```

```
For j = i To (i + 995) Step 1
```

```
    If Sheet1.Cells(j + 5, 7).Value = 1 Then farl = farl + 1
```

```
Next j
```

```

    filarl(i) = farl
Next i

'determining no of non-zero sample points out of 19 step changes for filter method
nosamfil = 10
For i = 1 To 10 Step 1
    If filarl(((i * 2000) - 1000)) = 0 Then nosamfil = nosamfil - 1
Next i

' calculating average arl
countfilav = 0
For i = 1 To 10 Step 1
    countfilav = countfilav + filarl(((i * 2000) - 1000))
Next i

If nosamfil <> 0 Then
    filarlav = countfilav / nosamfil
Else
    filarlav = 0
End If
Sheet1.Cells(8 + globalcount, 23).Value = filarlav
End Sub
,
,

```

```

Sub p21()
'sub to calculate p2 error for clock
'for clock method
clkp2 = 0
For i = (1000) To (19000) Step 2000
clkcountp2 = 0
For j = (i - 20) To (i + 980) Step 1
    If Sheet1.Cells(j + 5, 5).Value = 1 Then clkcountp2 = clkcountp2 + 1
Next j
    If clkcountp2 >= 998 Then clkp2 = clkp2 + 1
Next i
Sheet1.Cells(8 + globalcount, 24).Value = clkp2

'for filter method
fp2 = 0
For i = (1000) To (19000) Step 2000
fcountp2 = 0
For j = (i - 20) To (i + 980) Step 1
    If Sheet1.Cells(j + 5, 7).Value = 1 Then fcountp2 = fcountp2 + 1
Next j
    If fcountp2 >= 998 Then fp2 = fp2 + 1
Next i
Sheet1.Cells(8 + globalcount, 25).Value = fp2

End Sub

```

VITA

Sarvesh Gore

Candidate for the Degree of

Master of Science

Thesis: EVALUATION OF AUTOMATED STEADY STATE AND TRANSIENT
STATE DETECTION ALGORITHMS

Major Field: Chemical Engineering

Biographical:

Education:

Completed the requirements for the Master of Science in Chemical Engineering at Oklahoma State University, Stillwater, Oklahoma in December, May, 2014.

Completed the requirements for the Bachelor of Science in Chemical Engineering at Visvesvaraya National Institute of Technology, Nagpur, India in May, 2011.

Experience:

- Research Assistant for Dr. R Russell Rhinehart, OSU, August 2012 – May 2014
- Teaching Assistant for Dr. AJ Johannes, OSU, August 2012 – December 2012
- Teaching Assistant for Dr. Karen High, OSU, December 2012 – December 2013
- Assistant Manager for Hyderabad Industries Limited, Hyderabad, India June 2011 – July 2012

Professional Memberships:

- International Society of Automation (ISA), OSU Section