

COMPARISON OF THE CUSUM, FIRST-ORDER,
SELF-TUNING AND KALMAN FILTERS

By

DANIEL FAKUNLE

Bachelor of Science in Chemical Engineering

Obafemi Awolowo University

Ile-Ife, Nigeria

2005

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, 2010

COMPARISON OF THE CUSUM, FIRST-ORDER,
SELF-TUNING AND KALMAN FILTERS

Thesis Approved:

Dr. R.R. Rhinehart

Thesis Adviser

Dr. A.J. Johannes

Dr. Rob Whiteley

Dr. A. Gordon Emslie

Dean of the Graduate College

ACKNOWLEDGMENTS

I will like to express my appreciation and gratitude to Dr. Rhinehart, Dr. Gasem, Dr. Johannes, Dr. Whiteley, and Dr. Fahlenkamp.

TABLE OF CONTENTS.....	5
LIST OF TABLES.....	6
LIST OF FIGURES	7
CHAPTER I.....	8
INTRODUCTION	8
CHAPTER II.....	11
REVIEW OF LITERATURE	11
2.1 Noise.....	11
2.2 Why filter	11
2.3 Use of filters and Statistical Process Control (SPC)	12
2.4 Filters.....	12
2.4.1 First-Order filter:	13
2.4.2 Self-Tuning Filter	13
2.4.3 CUSUM Filter	14
2.4.4 Kalman filter.....	15
CHAPTER III	17
METHODOLOGY	17
3.1 Criteria for comparing filters-	17
3.2 Integral sum of errors	17

3.2.1 Base value with a step change introduced	19
3.2.2 Base value with a ramp change introduced.	21
3.2.3 Base value with an oscillatory change introduced.....	22
3.3 Cost.....	24
3.4 Ease of understanding	24
3.5 Filter Equations and code:.....	24
3.5.1 CUSUM filter -	24
3.5.2 First-order filter	25
3.5.3 Self-tuning filter	25
3.5.4 Kalman Filter.....	26
CHAPTER IV	28
RESULTS AND DISCUSSION OF RESULTS.....	28
4.1 Part 1	28
4.2 CUSUM Filter Results	29
4.3 First-order filter (FOF).....	34
Self-Tuning Filter (STF).....	39
4.4. The Kalman Filter	43
4.5 Comparison of Results between the four filters:.....	48
4.2 EASE OF UNDERSTANDING THE FILTERS	60
4.3 Cost.....	61

4.3.1 Ranking of filters based on cost analysis derived from the total operations	62
3.2. Ranking of filters based on cost analysis derived from the Number of	65
CHAPTER V	67
RECOMMENDATION	67
REFERENCES	68
APPENDIX.....	69

LIST OF TABLES

Table	Page
1. CUSUM ISE values to determine optimal trigger value	30
2. First-order ISE values to determine optimal Filtering factor (λ).....	35
3. STF ISE values to determine optimal Filter tuning value (E_f)	40
4. Kalman ISE values to determine optimal filter tuning value (Q).....	44
5. Comparison of ISE results of CUSUM, First-order, Self-tuning and Kalman filter for introduction of a step change to steady state base value.....	48
6. Comparison of ISE results of CUSUM, First-order, Self-tuning and Kalman... filter for introduction of a ramp change to steady state base value	53
7. Comparison of ISE results of CUSUM, First-order, Self-tuning and Kalman filter for introduction of a step change to steady state base value.....	55
8. Ease of understanding the filter.....	61
9 Evaluation of operations required in CUSUM filter.....	62
10 Evaluation of operations required in first-order filter.....	62
11 Evaluation of operations required in self-tuning filter.....	63
12 Evaluation of operations required in Kalman filter.....	64
13. Ranking of filter in increasing order of relative cost.....	65
14 Ranking of filter based on the required number of memory..... location / required variables	66

LIST OF FIGURES

Figure	Page
4.1: Plot Of CUSUM ISE values with the introduction of a step change to the base value vs CUSUM ISE values with no change in the base steady state value.....	32
4.2: Plot Of First-order ISE values with the introduction of a step change to the base value vs First-order ISE values with no change in the base steady state value.....	37
4.3. Plot of Self-tuning ISE values with the introduction of a step change to the base value vs Self-tuning ISE values with no change in the base steady state value.....	42
4.4. Plot of Kalman ISE values with the introduction of a step change to the base value vs Kalman ISE values with no change in the steady state base value.....	46
4.5. Plot of ISE values with the introduction of a step change to the steady state base value vs ISE values with no change introduction for the CUSUM, First-order , STF, and Kalman filter.....	50
4.6. Plot of ISE values with the introduction of a ramp change to the steady state base value vs ISE values with no change introduction for the CUSUM, First-order , STF, and Kalman filter.....	54
4.7. Plot of ISE values with the introduction of an oscillatory change to the steady state base value vs ISE values with no change introduction for the CUSUM, First-order , STF, and Kalman filter.....	58

CHAPTER I

INTRODUCTION

Filtering is the process of removing the noise associated with process measurements in order to observe the noiseless “or true” value. Filters seek to identify the true process signal in the presence of noise [6].

It is desired to have a filter that gives an accurate process signal for a variety of patterns in process. This is because filters are used to “filter out” the noise in a process and the more accurate the resulting signal, the better the filter. In this work, four filters are evaluated and compared for performance; and based on the comparison, conclusions are made. The comparison aspects include speed of tracking a change in the process (measured by ISE (Integral Sum of errors)), closeness of the filtered value to the true signal when there is no change in the process (also measured by ISE), computational burden, complexity for human understanding, and cost issues.

A series of simulations was done to generate data for the comparison using Visual Basic for Applications Programming. Noise was added to the simulation in order to corrupt the produced signal. The noisy signal was then filtered to obtain the true value which is known. Based on the filtered value, measures of goodness were calculated for each filter

in each simulated run. The simulation involved a steady base value and the introduction of a change to the base value. This steady base value is used to obtain the accuracy of the filter in finding the true signal value for a steady value, while the change is introduced in order to obtain the speed of tracking a change and the accuracy of the filter in tracking the change. The performance of the filters was then evaluated using the calculated ISE values. Step change, ramp change and oscillatory changes were introduced to the base value and the effect on the four filter outputs was then evaluated for results.

The experiments also included range of tuning factors to determine the best value to give a non-dominated individual on a multi-objective (ISE1 vs. ISE2) plane. Where ISE1 involved no change in a steady base value and ISE2 involved the introduction of a change to the steady state base value. The experiments included three changes; step, ramp and oscillatory.

The filters considered were the first-order filter (FOF) [6], the self-tuning filter (STF) [1], the CUSUM filter [6] and the Kalman filter [4].

The first-order filter was chosen because it is used in the industry as a common industrial filtering practice [1]. The self-tuning filter was chosen because it can achieve a high ratio of noise reduction without giving up its fast tracking of a process [1]. The Kalman filter was chosen because it is a well-known and often-used tool for mathematical stochastic estimation from noisy sensor measurements [3]. The CUSUM filter was chosen as an innovative filter to compare its performance to other industrially used filter such as the first-order and the Kalman filter.

The first-order filter averages data to reduce the effects of noise, but it also causes undesirable lag in the filtered variable. The greater the noise amplitude, the higher the required filtering. It is an Infinite Impulse Response (IIR) filter.

The objective of the self-tuning filter is to determine the filtering factor (λ) such that there is a 95% probability that the true value lies within the user chosen confidence interval about the average [1].

The CUSUM filter is a novel filter based on a cumulative sum of deviations. It is derived from Statistical Process Control Concepts (SPC) [6]. The CUSUM filter action holds the filtered value constant until there is sufficient statistical level of confidence that a change has occurred.

The Kalman filter is a recursive data processing algorithm [5]. A Kalman filter combines all available measurement data, plus prior knowledge about the system and measuring devices, to produce an estimate of the desired variables. It estimates a process by using a form of feedback control by utilizing time update equations and measurement update equations.

CHAPTER II

REVIEW OF LITERATURE

2.1 Noise

In this work, noise is defined as a fluctuating influence on process measurement that cannot be controlled because it is either: 1) a fluctuating misrepresentation of the true Process Variable (PV) (for example, some random electrical noise is added to the signal through the sensor and the electrical circuits) or 2) a transient influence on the PV.

Noise is often described as a zero-mean, independent, Gaussian distributed addition to the true PV (process variable) value [6]. This time varying ratio of “pure” signal to the electrical noise affects the quality of the information. [7].

2.2 Why filter

Filtering is done in order to remove noise and obtain the true process value. In process control, process noise and instrument noise are undesirable. This is because the controller will respond to the noise rather than the true process change. The noise masks the true

process variable. A result of this would be an increase in the process variability and increased wear and tear in final control element.

2.3 Use of filters and Statistical Process Control (SPC)

Usually the process variable (PV) is filtered and the filtered value is fed to the controller. Statistical Process Control (SPC) is a philosophy that overrides the desire to implement a change until there is sufficient statistical evidence to implement the change [6]. Statistical Process Control involves a broad scope of activities including the design of sampling and inspection schemes, experimental design and analysis to detect shifts in the mean or variability of a process [3].

2.4 Filters

There are two models for a digital filter. These are the Infinite Impulse Response filter (IIR) and the Finite Impulse Response filter (FIR) [8]. FIR has a finite memory. It is non recursive because it does not require a feedback loop in its implementation. IIR has infinite memory. It is called a recursive filter because the output is fed back into the input.

One of the most important criteria for filters is the squared error minimization [1]. This is because a filter with a low value of the squared error gives an output that is very close to the true value of the signal it is filtering, while one with a high value of the squared error gives an output that is far away from the true value of the filtered signal.

2.4.1 First-Order filter:

The first-order filter averages data to reduce the effects of noise. It is a recursive filter. It has an undesirable characteristic of introducing a lag in the control loop. This lag in the loop affects stability, and undesirably limits the aggressiveness of the controller. When there is a change in process variable (PV) value, the first-order filter output lags behind. During a noisy steady state period, the first-order filter output keeps changing at every sample. The FOF equation is given as [1]:

$$x_{fi} = \lambda * (x_{fi-1}) + (1 - \lambda) * x_i \quad (1)$$

Where x_f is the filtered value

λ is the filtering factor

i is the sample counter

And x_i is the input signal

2.4.2 Self-Tuning Filter

The self-tuning filter automatically adjusts λ in the FOF. It is based on the 95% confidence interval for engineering economic decisions. Its objective is to determine λ such that there is a 95% probability that the true steady state process value is within the user chosen confidence interval about the average. The three equations for the self-tuning filter are [1]:

$$\delta_{fi} = 0.1 * (x - x_{old})^2 + 0.9 * \delta_{fi-1} \quad (2)$$

$$\lambda = \min \left\{ \left(\frac{1}{2} + \frac{1.1668 * \delta_f^2}{E^2} \right)^{-1}, 1 \right\} \quad (3)$$

$$x_{fi} = (1 - \lambda)x_{fi-1} + \lambda x_i \quad (4)$$

Where x is the current measured value of the signal.

Where x_{old} is the old signal measured value.

Where x_{fi} is the resulting filtered value.

Where E is the error between the true (unknown) value and the 2.5% extreme value of the filtered value. It is chosen by the user.

δ_{fi} is a measure of data variance.

2.4.3 CUSUM Filter

In statistics, 3-sigma is the two specific points on a normal distribution centered about the mean. Out of a population, 99.73% of the population will fall between these values. It represents the entire population and +3sigma and -3sigma represents the probable range of variation [2]. It is used in the industry as an organized and systematic method for strategic process improvement, new product and service development relying on statistical methods and the scientific method to make dramatic reductions in customer defined defect rates [9]. The CUSUM filter is based on this principle. The filter attempts to find the mean within noisy data. The concept holds the mean is unchanged, until there is sufficient statistical confidence that a real change has occurred. If the Process Variable (PV) “really” changes, then the average will shift. The CUSUM gives the criteria for making a change as [6]:

$$CUSUM = \sum \frac{(x - \bar{x}_{old})}{\sigma_x} \quad (5)$$

$$\text{If } |CUSUM| > \text{TRIGGER} * \sqrt{N} \quad (6)$$

$$\sigma_{fnew}^2 = \left(\frac{M-2}{M-1}\right) \sigma_{fold}^2 + \left(\frac{1}{M-1}\right) (x_{new} - x_{old})^2 \quad (7)$$

$$\sigma_x = \sqrt{\frac{\sigma_{fnew}^2}{2}} \quad (8)$$

Where N is the number of samples for which CUSUM had been calculated.

x is the measured value of the variable.

\bar{x}_{old} is the old value of the calculated mean of the measured variable.

σ_x is defined by equations 7 and 8

$M \cong 11$ (This gives the “best” balance of removing variability from the estimate)

Trigger value represents the SPC confidence level in a decision. The trigger value of 1 represents 68.3% confidence level. A value of 2 represents traditional economic decision trigger values of $\pm 2\sigma$ representing a 95% confidence in a decision. A value of 3 is the 3-sigma 99.73% confidence level. Values from 2 to 4 are generally chosen to balance responsiveness and false alarms for particular SPC applications [6].

2.4.4 Kalman filter

The Kalman filter is the application of the combination of the method of combining a model of the process with adjusted process measurements to produce an estimate of the state [4]. It named after Rudolph E. Kalman, who published a paper describing a recursive solution to the discrete-data linear filtering problem in 1960 [8]. The equations are given below [3].

The process model is given as

$$\mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1} + \mathbf{B}u_{k-1} + \mathbf{w}_{k-1} \quad (9)$$

Measurement model is given as

$$\mathbf{Z}_k = \mathbf{H}\mathbf{x}_k + \mathbf{v}_k \quad (10)$$

\mathbf{w}_k and \mathbf{v}_k represent the measurement and process noise. They are assumed to be independent, white, and with normal probability distributions.

$$P(\mathbf{w}) \sim N(0, \mathbf{Q}) \quad (11)$$

$$P(v) \sim N(0, R) \tag{12}$$

Q is the process noise covariance and R is the measurement noise covariance matrix.

They usually change with each time step, but are assumed to be constant for simplicity in this work. Matrix A relates the state at previous time step $k-1$ to the current time step k .

Matrix B relates the optional control input u to the state x . Matrix H in the measurement equation relates the state to the measurement Z_k .

CHAPTER III

METHODOLOGY

3.1 Criteria for comparing filters

In this work, the following were the criteria used for the filter comparisons:

1. Integral Sum of errors (ISE)
2. Cost
3. Ease of understanding

3.2 Integral sum of errors

The integral sum of the errors is used as a basis for the filters comparison. This shows how close to the true value that the filtered value of the noisy signal goes. The higher the value of the ISE the less desirable the filter.

The ISE is calculated using the equation:

$$ISE = \sum(x_f - x_t)^2 \quad (13)$$

Where x_f is the filtered value while x_t is the true value of the signal. Each simulation run gives the value of the ISE for each of the four filters.

All simulation was performed with the following variables

$$\sigma_n=2$$

$$dt=0.1$$

$T=100s$

$\tau_n=2$

σ_n is the standard deviation of the noise

τ_n is noise periodic time

dt is the time step

T is the total duration of the simulation

Gaussian distributed noise was added to the signal to corrupt it. This is to enable the evaluation of the filter ability to filter out the noise. Noise is generated using equation

(14)

$$x_{noise} = \sigma_{naw} * \text{Sqr}(-2 * \text{Log}(\text{Rnd})) * \text{Sin}(2 * 3.14159 * \text{Rnd}) \quad (14)$$

Where σ_{naw} is the noise standard deviation.

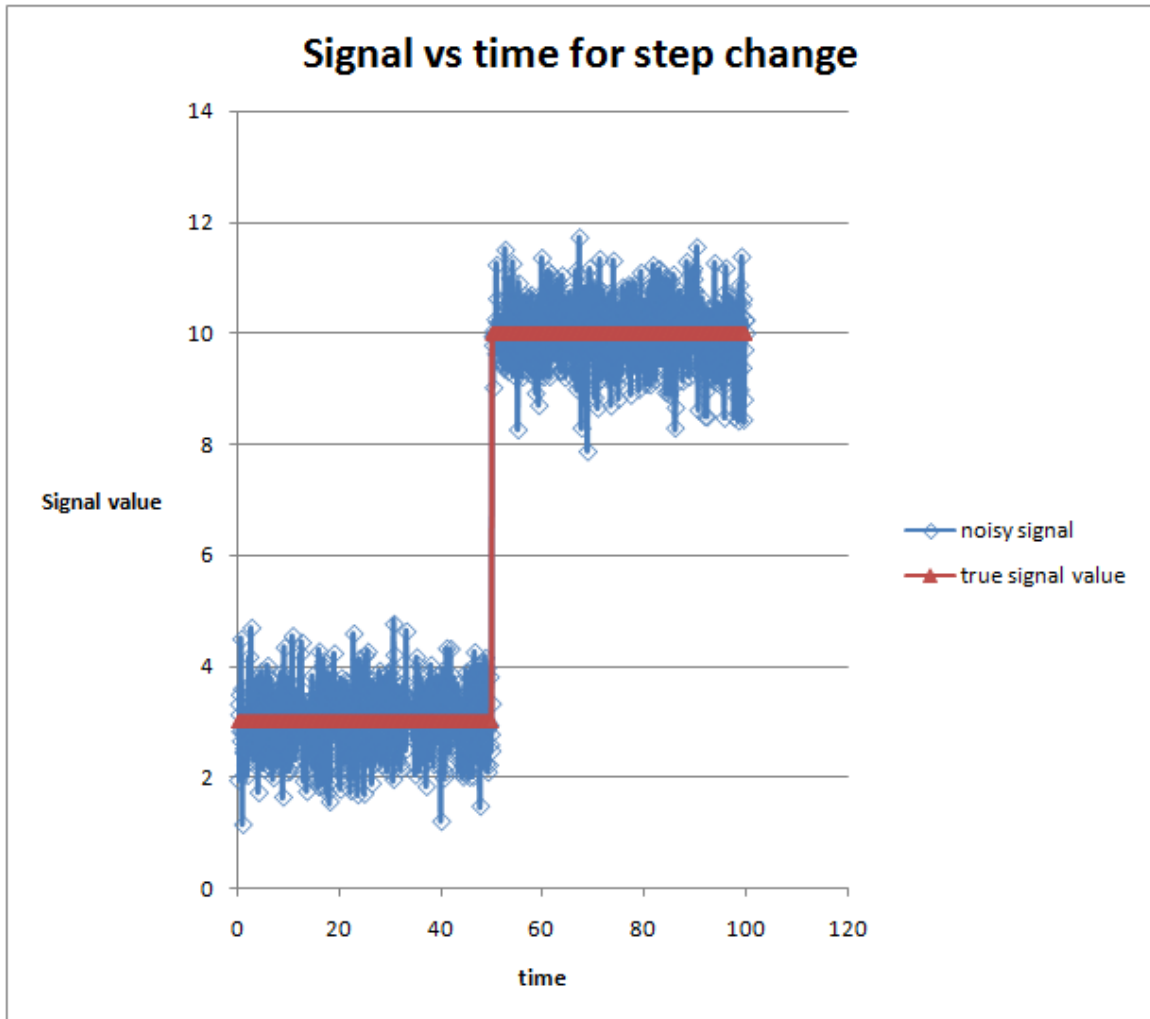
Rnd is the random number generator to generate a random number for generating the value of the noise.

x_{noise} is the generated noise.

An optimal filtering value was required for each of the four filters. The optimal filtering value is required in order to use this value to tune the filter for the purpose of comparison of filter performance relative to the other filters evaluated in this study. An optimal filtering value is the value of the filtering factor that produces the lowest minimization of squared error values (i.e. lowest ISE values). To obtain the optimal filtering value, series of simulation was run for various tuning values.

3.2.1 Base value with a step change introduced

Figure 3.1 Sample of step change

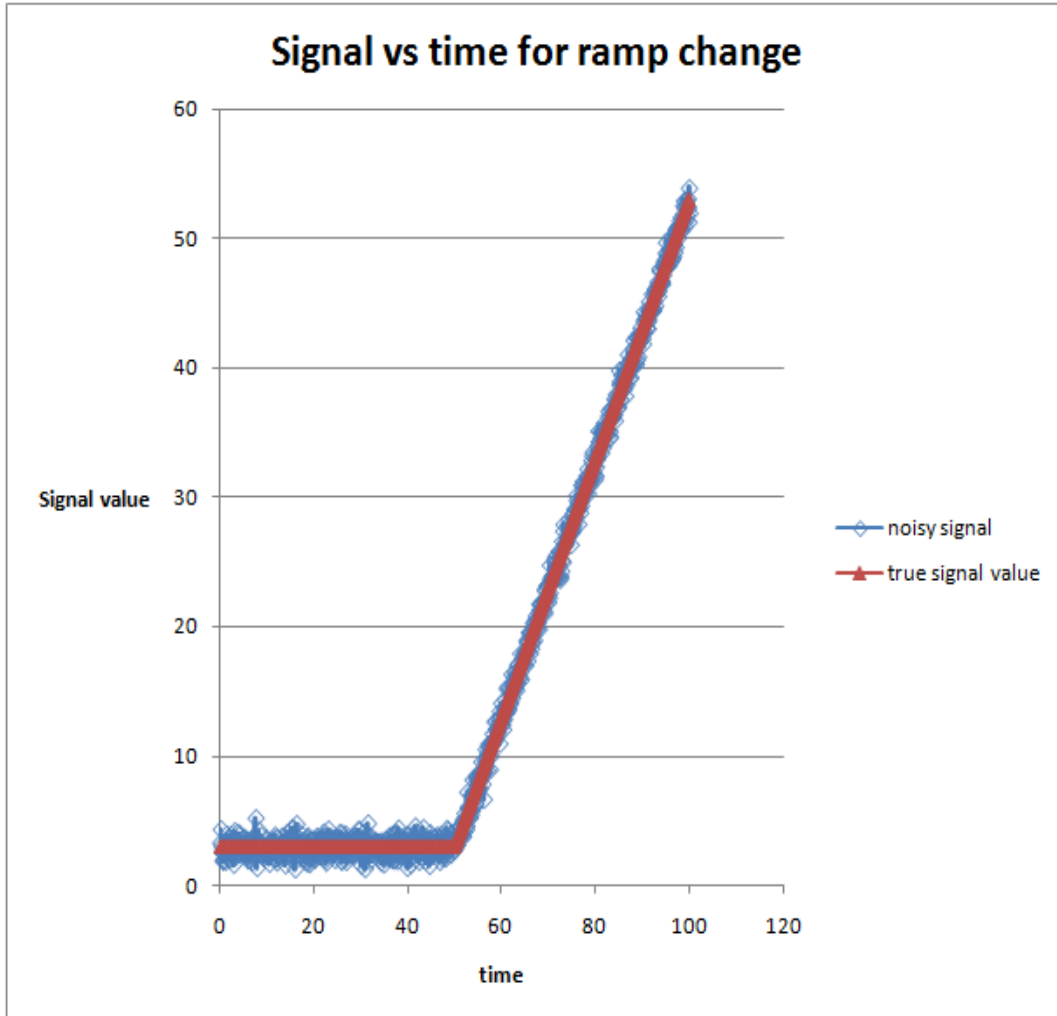


In order to obtain the ISE values during the process with the introduction of a step change, the ISE values calculated using the formula above during the steady state is subtracted from that obtained during the step change. This gives the true ISE value for the introduction of the change.

The process is allowed to run for 50 seconds with a time step of 0.1seconds with a steady state base value. After the initial 50 seconds, a step change was introduced to by changing the steady state base value to a value greater than the steady state base plus the standard deviation of the noise (0.1). The process is made to run for another 50 seconds. The time duration of the steady state and the step window is the same in order to be able to obtain the ISE due to the introduction of the change to the base value i.e. all variables are kept constant except the change. The ISE due to the introduction of the change in the signal base value is obtained by subtracting the ISE obtained during the steady state window from the step window. This obtained value after the subtraction represents the ISE due to the change, since the signal was maintained at constant values before and after the change respectively.

3.2.2 Base value with a ramp change introduced.

Figure 3.2 Sample of Ramp Change



In order to obtain the ISE values during the process with the introduction of a ramp change, the ISE values were calculated using the formula above during the steady state and then with the ramp change introduced.

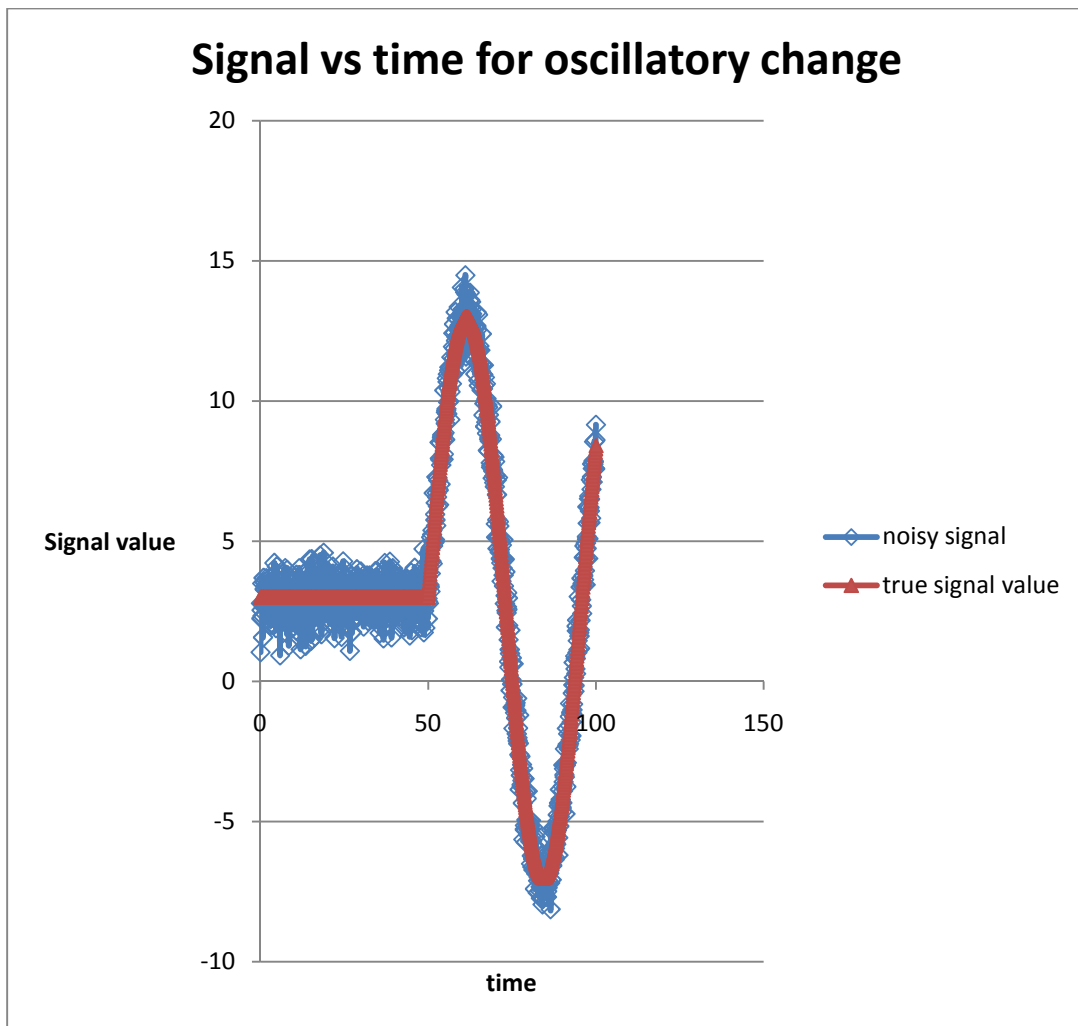
The simulation was performed with the steady state base value was held constant for 50 seconds with a time step of “0.1seconds”. After, the initial 50 seconds, a ramp change was introduced using the formula “ $x = 3 + (t - 50)$ ” for a subsequent 50 seconds. This gives

a ramp signal with a slope of “1”. The signal has an intercept of 3 at time $t=50$ in order to begin the ramp at the previous steady state value.

“ x ” is the true value of the signal and “ t ” is the time.

3.2.3 Base value with an oscillatory change introduced.

Figure 3.3 Sample of oscillatory change



In order to obtain the ISE values for a process with the introduction of an oscillatory change, the ISE values were calculated using the ISE formula above during the steady state and then also with the oscillatory change introduced.

The base value was fixed for 50 seconds with a time step of “0.1seconds”. After, the initial 50 seconds an oscillatory change was introduced using equation (15)

$$x = 3 + (-0.729535369032546 * SIN(radians\left(\left(\frac{5*\pi}{2}\right)(t - 50)\right))) \quad (15)$$

Where:

x is the true value of the signal

t is the time/duration of the signal

radians converts the angle $\left(\frac{5*\pi}{2}\right)(t - 50)$ from radian angles into degrees. (The simulation program was written in VBA which defaults to radians).

-0.729535369032546 is the amplitude of the oscillatory function.

3 is the base value about which the signal oscillates.

These values (i.e. angle " $\left(\frac{5*\pi}{2}\right)(t - 50)$ ", amplitude " $- 0.729535369032546$ " were chosen by using the “goal seek” trial and error function of Excel in order to obtain an oscillatory signal that initiates at the steady state base value and oscillates about this value.

The base value is maintained at the same value as the earlier simulation runs of the signals with the introduction of a step change and the ramp change.

A total of 1000 time steps were simulated with 500 time steps for steady base value and 500 time steps for the base value with an oscillatory change introduced.

3.3 Cost

The cost of the filter was analyzed by

- a. The number of the required arithmetic operations
- b. The number of the required variables.

The higher the number of the required arithmetic operations the higher the cost since a faster processor would be required for the process. While a filter that requires a higher number of variables would have a greater cost since more memory space would be required to carry out the process.

3.4 Ease of understanding

This is a relative comparison based on the education required to understand how the filter works. Based on opinion numbers are assigned to each of the filters. The filter with the lower numbers require a relatively lower level of education for understanding while those with the higher numbers require higher level of education for the understanding.

3.5 Filter Equations and code:

3.5.1 CUSUM filter -

The code for the used for the CUSUM filter is given below [6].

```
IF (first call) THEN
  N = 0
  XOLD = 0.0
  XSPC = 0.0
  V = 0.0
  CUSUM = 0.0
  M=11
  FF1 = REAL((M-2)/(M-1))
  FF2 = (1.0 - FF1)/2.0
END IF
Obtain X
N = N + 1
```

```

V = FF1* V + FF2*(X - XOLD)**2
XOLD = X
CUSUM = CUSUM + X - XSPC
IF (ABS (CUSUM).GT.TRIGGER*SQR(V*N)) THEN
    XSPC = XSPC + CUSUM/N
    N = 0
    CUSUM = 0.0
END IF

```

If the absolute value of CUSUM is greater than the critical value of “Trigger*(σ)*N”, the filter assumes that a true change in process has occurred, and it updates the filtered value as XSPC= XSPC + CUSUM/N. And then the variables “N” and “CUSUM” are reset. If the CUSUM is less than the critical value, then the filter assumes that the change in input is due to normal variability of the process, not a true change in level. Hence, it does not change the filtered output.

3.5.2 First-order filter

The code for the first-order filter is given below [1].

$$XF = FF * X + (1 - FF) * XF$$

Where XF is the filtered value

FF is the filtering factor

And X is the input signal to be filtered for noise

This filter has the disadvantage that it lags behind the true value.

3.5.3 Self-tuning filter

The code equations used for the self-tuning filter are given below[1].

```

DELTA F = 0.9 * DELTA F + 0.1 * (X-XOLD) ^ 2
XOLD=X
LAMBDA=1/(0.5+1.1668*DELTA F/EF^2)
IF LAMBDA>1 THEN LAMBDA=1
XF=LAMBDA*X+(1-LAMBDA)*XF
where:

```

XF is the filtered value

LAMBDA is defined above as $1 / (0.5 + 1.1668 * DELTAF / EF ^ 2)$

EF is the filtering value

X is the measured value of the signal

DELTAF is a measure of data variance

3.5.4 Kalman Filter

The Kalman filter has two sets of equations. They are the time update equations and the measurement update equations. The equations are given below [7].

Time update:

$$\text{XHATMINUS}(i) = \text{XHAT}(i-1)$$

$$\text{PMINUS}(i) = \text{P}(i - 1) + Q$$

Measurement update:

$$K(i) = \text{PMINUS}(i) / (\text{PMINUS}(i) + R)$$

$$\text{XHAT}(i) = \text{XHATMINUS}(i) + K(i) * (\text{XMEAS} - \text{XHATMINUS}(i))$$

$$P(i) = (1 - K(i)) * \text{PMINUS}(i)$$

$$\text{XF} = \text{XHAT}(i)$$

where

XHATMINUS is the projected state

XHAT is actual state measurement vector

PMINUS is projected error covariance

XMEAS is the measured value of the signal to be filtered

P is the updated error covariance

Q is the noise covariance

R is the variance associated with the measurement

K is the Kalman gain

X_F is the Kalman filtered value

CHAPTER IV

RESULTS AND DISCUSSION OF RESULTS

The following results were obtained from the simulated experiment to compare the performance of four filters (CUSUM, FOF, STF and Kalman) using the ISE (integral sum of errors). In order to carry out the comparison, the experiment was divided into three parts. Part one evaluated the filters performance for a simulated process involving an initial fixed base steady state value with the introduction of a subsequent step change to the base value. The second part had a fixed base value with the introduction of a subsequent ramp change. The final part had a fixed base value with the introduction of a subsequent oscillatory change.

4.1 Part 1

The details of the obtained results are given below:

Column 1 of Table 1 shows the number of the simulation run, the other columns show the name of the filter, the filter tuning value and whether a change was introduced or not. The first of each of the pairs of columns shows the ISE values for a steady state signal with no change introduced. The second columns show the ISE values with the introduction of a step change.

4.2 CUSUM Filter Results

Using the filtering values of 1.5, 2, 2.5, 3, 3.5 and 4, a series of simulations was run in order to obtain the filtering value which is non-dominated by the other values. A non-dominated value is a value which has no other ISE value lower than itself. The obtained results are as shown in Table 1.

Table 1. CUSUM ISE values to determine optimal trigger value

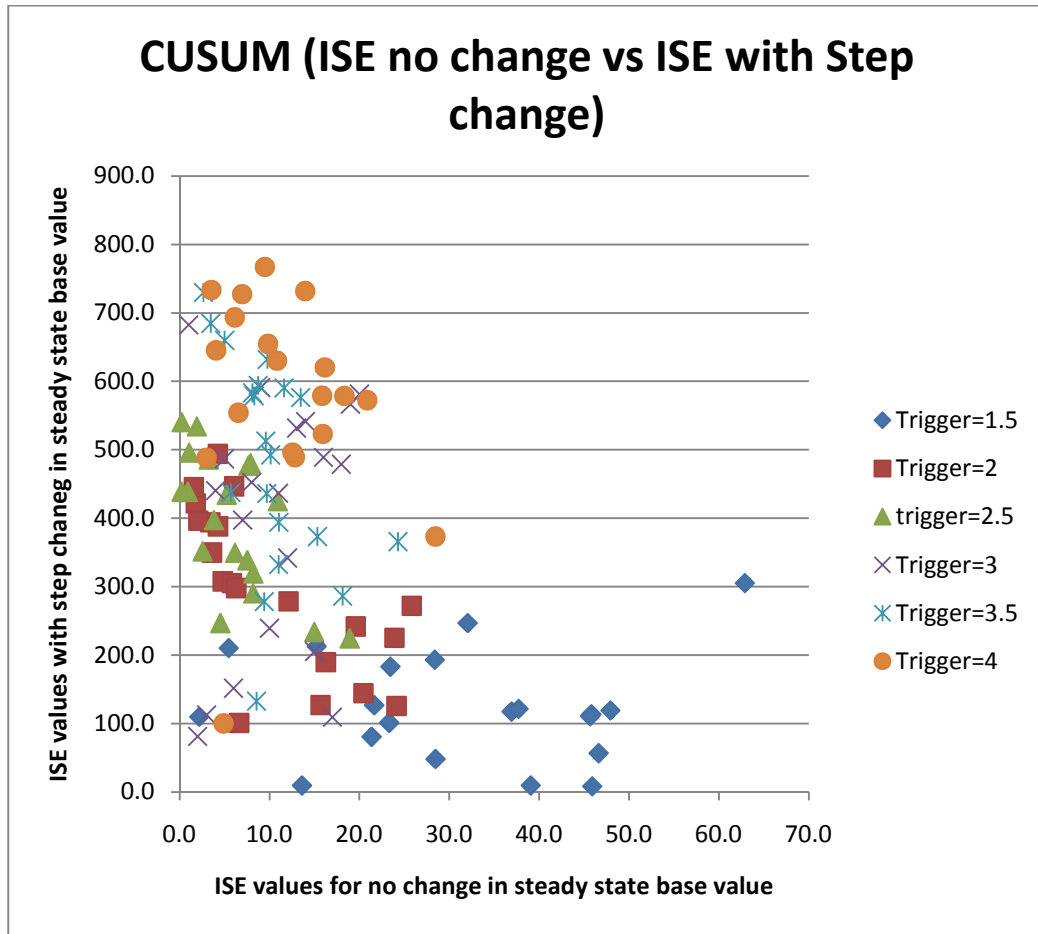
Number of Simulation	STEP CHANGE					
	CUSUM 1.5		CUSUM 2		CUSUM 2.5	
	Trigger= No change	with step change	Trigger= No change	with step change	Trigger= No change	with step change
1	28.4	193.1	3.6	349.9	7.9	480.1
2	47.9	119.1	1.8	421.5	1.1	495.6
3	15.2	212.8	15.7	127.1	18.9	224.4
4	23.4	182.9	4.2	494.1	0.2	540.1
5	2.2	109.6	6.1	446.7	1.9	534.1
6	39.1	9.4	3.4	394.1	4.5	246.8
7	5.4	210.2	25.8	272.0	8.2	319.2
8	62.9	305.1	1.6	445.6	0.9	438.6
9	28.5	48.0	23.9	225.3	10.9	424.9
10	13.6	9.3	16.3	189.9	6.2	349.4
11	45.9	8.2	2.1	396.4	7.5	338.4
12	36.9	117.4	4.8	307.8	3.2	485.7
13	21.6	126.7	6.6	100.8	3.8	397.2
14	23.3	101.0	24.1	125.6	5.2	434.1
15	32.0	246.7	20.4	144.4	8.2	290.1
16	45.8	113.4	12.1	278.5	2.6	351.6
17	45.7	110.8	5.8	305.3	0.2	438.5
18	37.7	121.3	4.3	388.1	7.6	476.9
19	46.6	56.6	19.6	241.8	15.0	233.8
20	21.3	80.5	6.3	297.9	10.5	338.9
Average=	31.2	124.1	10.4	297.6	6.2	391.9
Variance=	247.2	6546.5	71.2	14216.0	24.7	9621.5
Std dev=	15.7	80.9	8.4	119.2	5.0	98.1
2*σ(95%conf.)	7.0	36.2	3.8	53.3	2.2	43.9

Table 1 continued.

		STEP CHANGE					
Number of Simulation	CUSUM		CUSUM		CUSUM		
	Trigger= No change	3 with step change	Trigger= No change	3.5 with step change	Trigger= No change	4 with step change	
1	1.5	682.6	24.3	365.7	12.8	489.2	
2	11.2	81.4	9.4	278.1	15.9	579.0	
3	22.1	112.5	5.7	437.6	6.5	554.2	
4	5.0	440.3	2.6	729.6	3.0	488.3	
5	5.7	486.9	8.7	594.2	16.2	620.3	
6	1.4	151.7	8.3	578.9	14.0	731.8	
7	5.9	397.4	11.0	394.0	4.9	100.3	
8	9.1	452.5	9.7	632.1	9.5	767.0	
9	3.5	590.9	8.1	583.1	10.8	630.0	
10	14.2	239.2	13.5	576.3	7.0	727.5	
11	7.3	436.4	9.7	436.1	9.8	654.6	
12	8.5	342.0	3.5	684.6	28.4	373.3	
13	3.2	531.8	15.3	373.2	3.5	733.4	
14	3.2	541.3	8.5	132.7	15.9	523.0	
15	14.6	205.5	11.0	332.2	6.1	693.5	
16	3.7	489.1	11.6	590.3	4.1	645.3	
17	23.2	109.5	18.1	286.3	20.9	572.4	
18	4.3	479.0	9.6	512.8	12.6	495.9	
19	16.6	566.6	10.1	492.5	18.3	578.6	
20	7.4	581.3	5.0	659.9	9.5	438.7	
Average=	8.6	395.9	10.2	483.5	11.5	569.8	
Variance=	41.6	33614.5	24.5	24835.9	42.7	23608.9	
Std dev=	6.5	183.3	5.0	157.6	6.5	153.7	
2*σ(95%conf.)	2.9	82.0	2.2	70.5	2.9	68.7	

Column 1 shows the number of the simulation run, the other columns show the name of the filter, the filter tuning value and whether a change was introduced or not. The first of each pair of columns shows the ISE values for a steady state signal with no change introduced. Each of the second columns shows the ISE values with the introduction of a step change.

Figure 4.1 Plot of CUSUM ISE values with the introduction of a step change to the base value vs CUSUM ISE values with no change in the base steady state value.



In Figure 4.1, the x-axis represents the ISE value during the steady state window, while the y-axis represents the ISE values during the window involving the introduction of a step change to the initial steady state value.

The diamonds represent a trigger value of 1.5

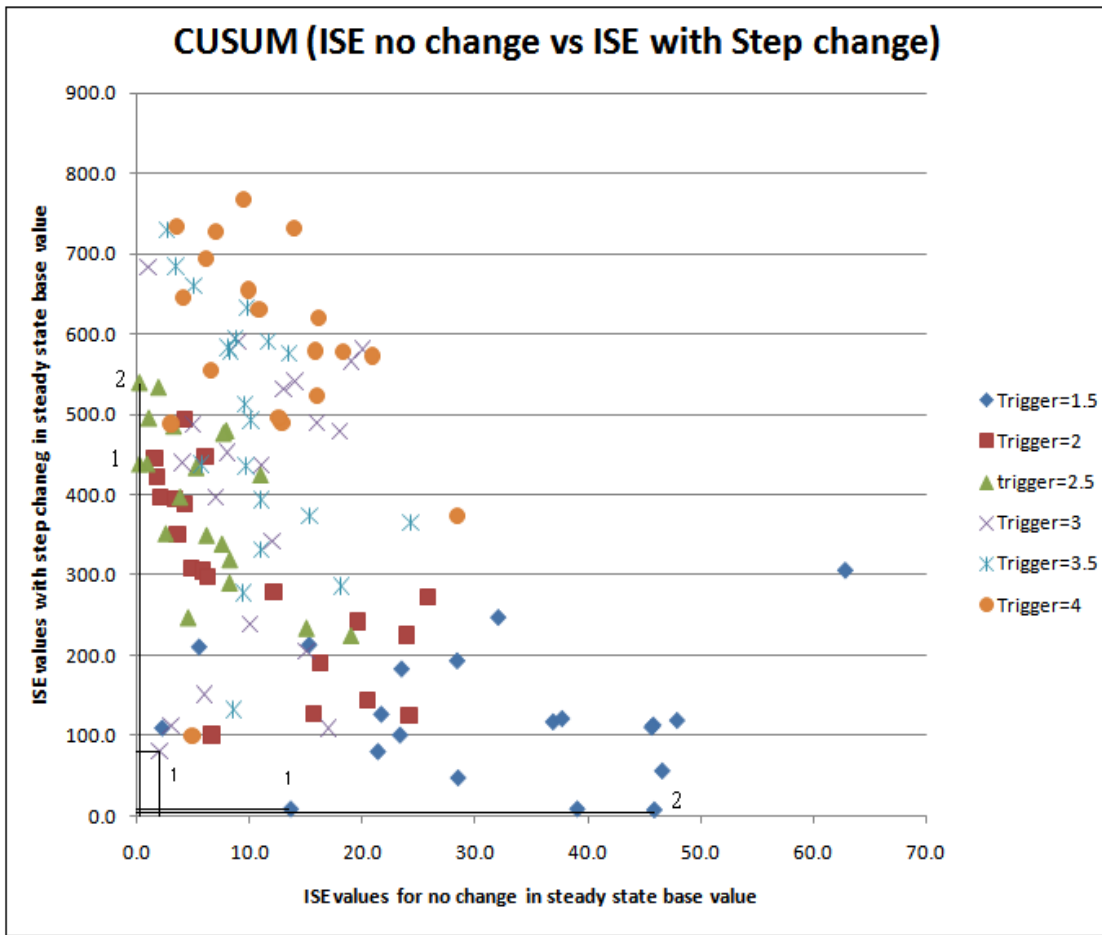
The squares represent a trigger value of 2

The “X” represents a trigger value of 3

The “✕” represents a trigger value of 3.5

The circles represent a trigger value of 4

Figure 4.1.1 selection of optimum CUSUM filter tuning value



The non-dominated values are

1. The diamonds representing a filter tuning value of 1.5 (There are 2 non dominated diamonds as shown by the markers in Figure 4.1.1)
2. The triangles representing a filter tuning value of 2.5 (There are 2 non-dominated triangles as shown by the markers in Figure 4.1.1)

3 The “X” representing a filter tuning value of 3 (There is 1 non-dominated “X” as shown by the marker in Figure 4.1.1).

The chosen optimum trigger tuning value is 3. This is because there are two non dominated triangles. And the triangles dominate the “X” on the x-axis.

4.3 First-order filter (FOF)

A series of simulation was run for FOF with filtering values of 0.1, 0.2, 0.3, 0.4, 0.5 and 0.6. From Figure 4.2, the filtering factor of 0.2 is non-dominated; hence this is selected as the optimum filtering value for the FOF. The results are shown in Table 2.

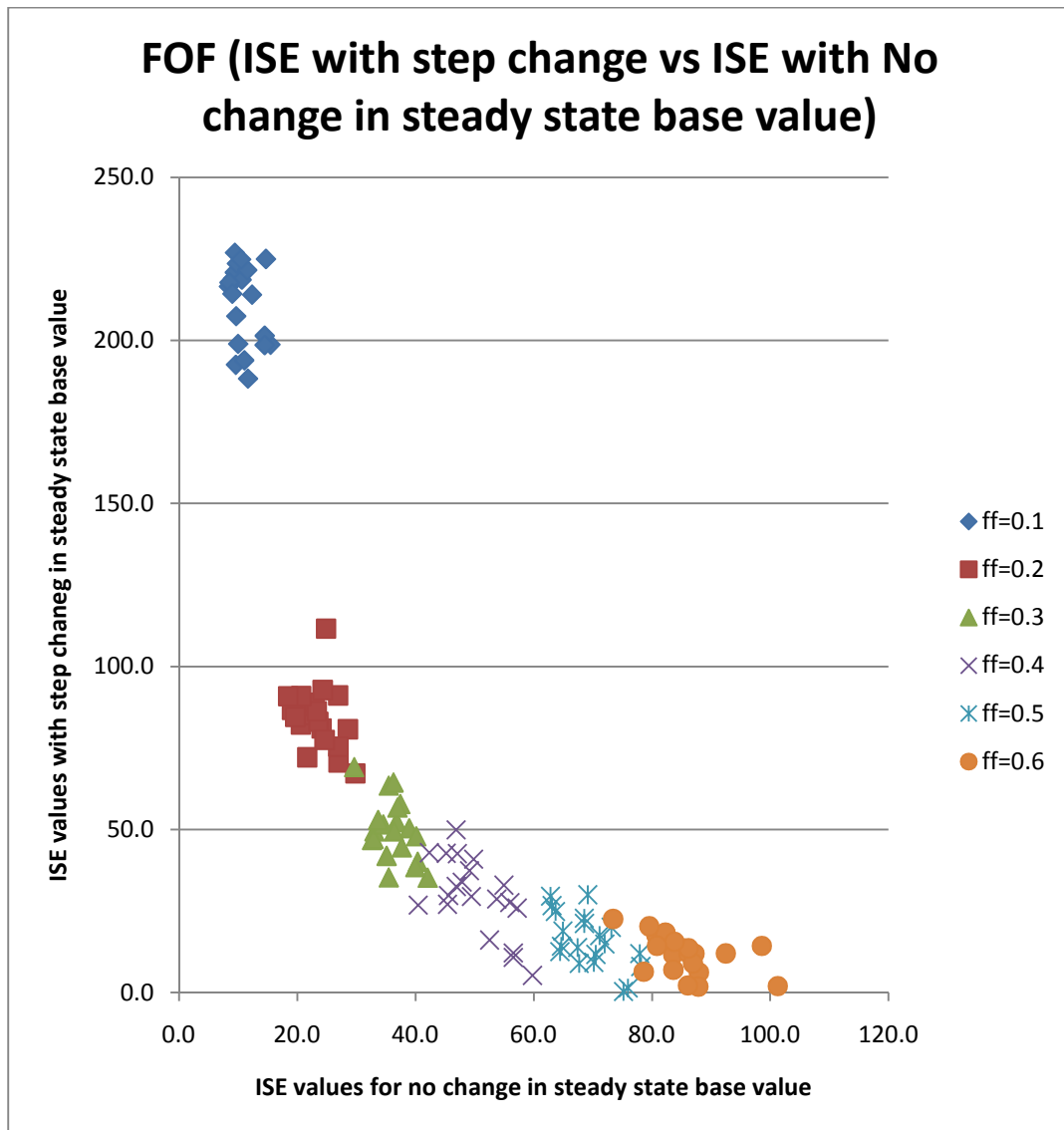
Table 2. First-order ISE values to determine optimal Filtering factor (λ)

Number of simulation	STEP CHANGE					
	FOF		FOF		FOF	
	FF=	0.1	FF=	0.2	FF=	0.3
	No change	with step change	No change	with step change	No change	with step change
1	9.1	219.0	24.1	81.0	29.6	69.1
2	9.6	192.6	26.9	91.1	32.7	46.7
3	8.4	216.6	20.6	82.2	42.0	35.2
4	11.7	188.3	20.7	85.1	37.4	57.8
5	9.4	220.9	20.7	87.9	35.1	41.9
6	12.4	214.0	19.1	86.7	35.5	63.4
7	10.0	199.0	29.8	67.2	36.8	52.2
8	14.7	225.0	22.9	88.9	36.3	64.4
9	10.6	218.6	24.9	111.6	37.7	44.5
10	15.5	198.7	27.0	70.5	37.0	56.7
11	8.6	217.8	23.6	83.1	33.1	49.4
12	14.5	201.4	23.3	86.2	38.9	50.3
13	9.8	223.6	20.7	90.9	32.9	47.1
14	9.0	214.3	28.6	80.7	40.1	47.9
15	11.0	193.9	27.0	75.4	34.6	51.5
16	9.4	226.9	19.7	84.5	40.4	40.1
17	9.7	207.5	18.5	90.8	33.7	52.9
18	11.5	221.6	21.7	72.1	36.4	49.3
19	14.5	198.6	24.3	92.8	40.0	38.4
20	10.5	224.9	24.6	77.4	35.5	35.3
Average=	11.0	211.1	23.4	84.3	36.3	49.7
Variance=	4.9	154.0	10.6	93.7	9.5	87.4
Std Dev.=	2.2	12.4	3.2	9.7	3.1	9.3
2* σ (95%conf.	1.0	5.5	1.5	4.3	1.4	4.2

Table 2 Continued

Number of simulation	STEP CHANGE					
	FOF		FOF		FOF	
	FF=	0.4	FF=	0.5	FF=	0.6
	No change	with step change	No change	with step change	No change	with step change
1	42.4	42.9	69.1	30.0	92.5	12.0
2	49.5	29.4	70.2	9.3	85.5	13.5
3	52.5	16.1	73.1	20.0	83.6	11.6
4	45.4	27.0	67.4	13.8	87.9	6.2
5	57.2	25.9	63.7	24.9	86.2	11.9
6	46.9	32.5	68.6	21.2	87.2	12.0
7	47.1	42.6	78.1	8.0	86.9	9.0
8	45.6	29.8	64.9	18.8	78.6	6.4
9	53.8	28.7	76.0	1.4	101.3	2.0
10	45.2	42.7	67.7	8.9	86.2	13.6
11	47.9	33.8	71.1	17.5	80.7	17.6
12	56.5	10.8	64.4	12.5	87.9	1.8
13	40.4	26.8	70.5	11.7	86.1	2.2
14	54.9	32.9	64.7	14.2	73.4	22.6
15	49.8	40.9	72.0	14.9	82.3	18.4
16	59.8	5.2	77.9	11.9	83.8	15.5
17	56.0	27.6	63.1	26.7	80.8	14.4
18	49.1	37.4	62.8	29.5	79.5	20.3
19	46.9	49.9	75.2	0.3	98.6	14.3
20	56.6	12.2	68.6	22.8	83.6	6.9
Average=	50.2	29.8	69.5	15.9	85.6	11.6
Variance=	29.5	138.0	23.2	71.0	41.1	35.6
Std Dev.=	5.4	11.7	4.8	8.4	6.4	6.0
2*σ(95%conf.	2.4	5.3	2.2	3.8	2.9	2.7

Figure 4.2 Plot Of First-order ISE values with the introduction of a step change to the base value vs First-order ISE values with no change in the base steady state value.



In Figure 4.2

The diamonds represent a filtering factor value of 0.1

The squares represent a filtering factor value of 0.2

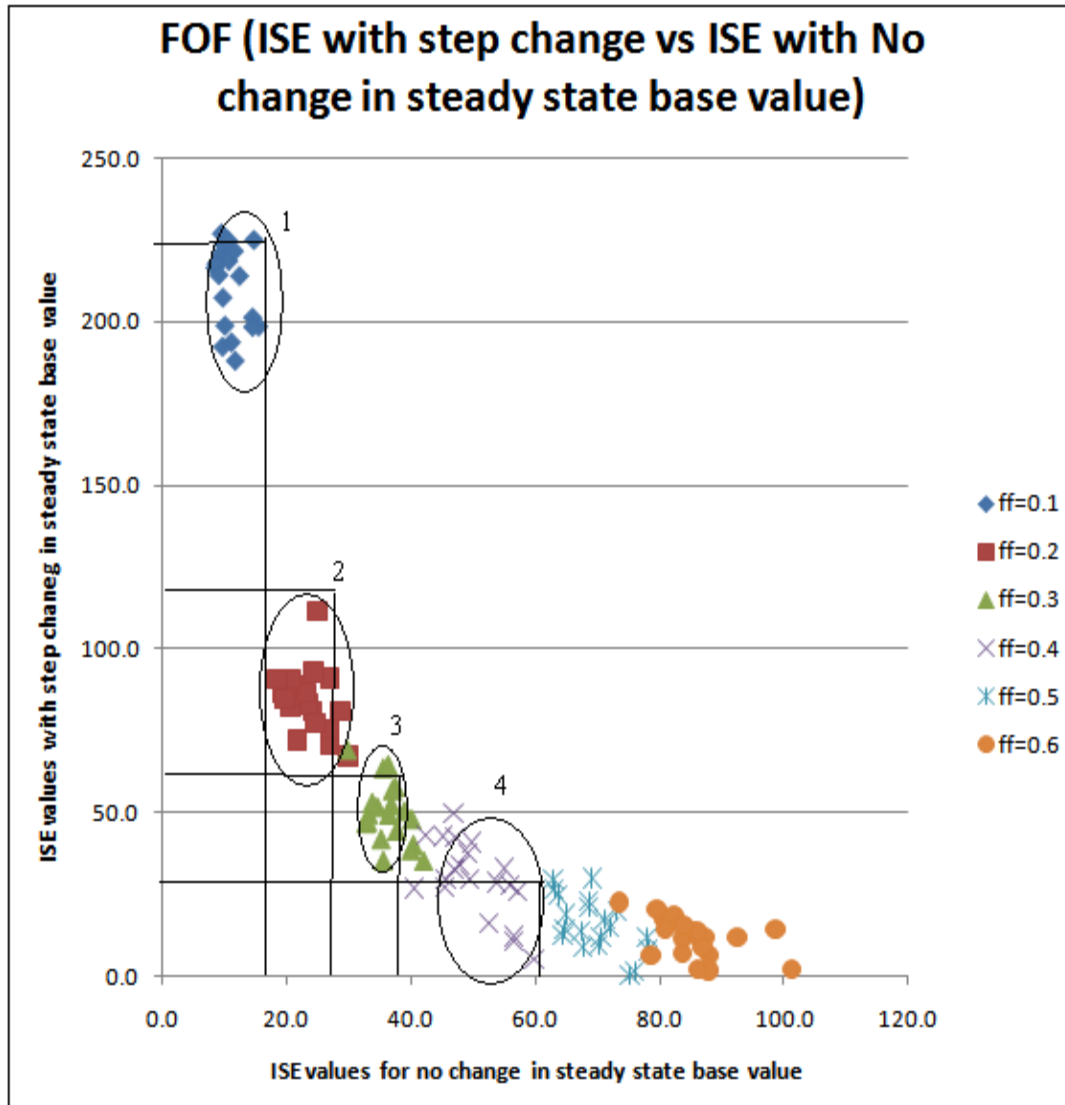
The triangles represent a filtering factor value of 0.3

The “X” represent a filtering factor value of 0.35

The “✱” represent a filtering factor value of 0.4

The circles represent a filtering factor value of 0.45

Figure 4.2.1 Selection of optimum first-order filter tuning value



The non-dominated values are:

1. The diamond representing tuning value of 0.1 (Enclosed by circle 1 in Figure 4.2.1)
2. The squares representing tuning value of 0.2 (Enclosed by circle 2 in Figure 4.2.1)
3. The triangles representing tuning value 0.3 (Enclosed by circle 3 in Figure 4.2.1)

The chosen optimum filter tuning value is 0.2. This is the choice because the values of the triangle enclosed by circle 2 dominate the other tuning values enclosed by circle 3 and circle 4 on the x-axis and dominate circle 1 on the y-axis.

Self-Tuning Filter (STF)

The self-tuning filter simulation results are given in Table 3.

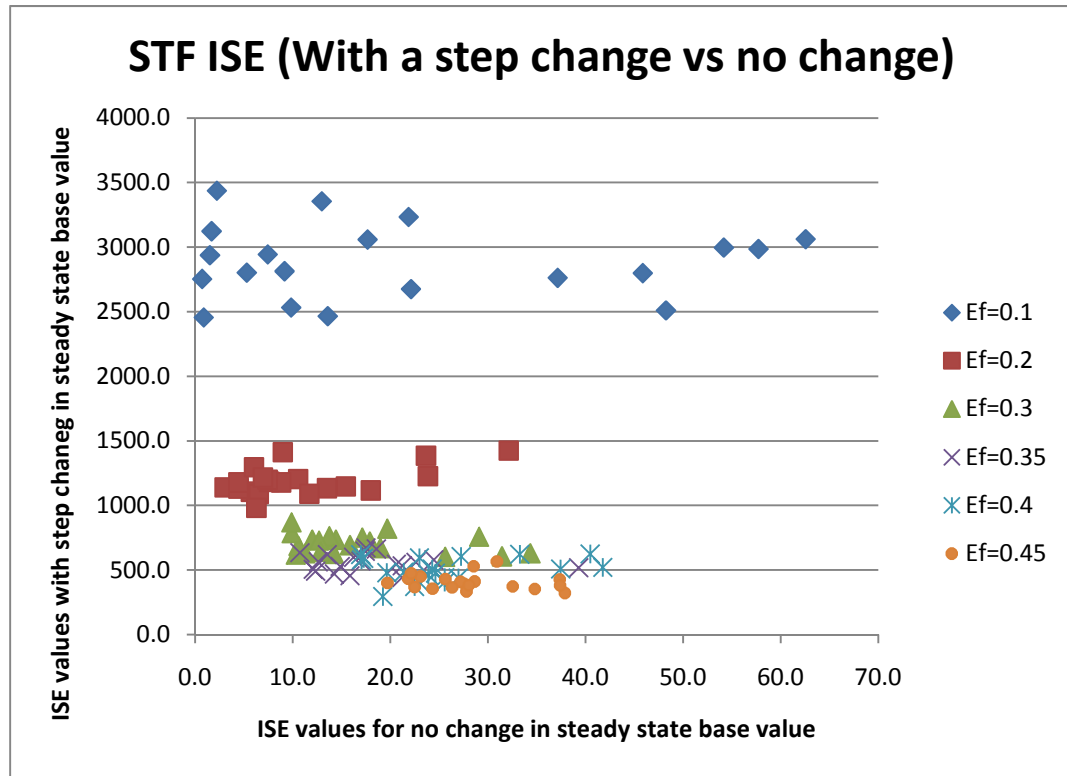
Table 3.STF ISE values to determine optimal filter tuning value (Ef)

Number of Simulation	STEP CHANGE					
	STF		STF		STF	
	EF=	0.1	EF=	0.2	EF=	0.3
	No Change	with step change	No Change	with step change	No Change	with step change
1	2.2	3435.8	3.0	1139.2	34.4	630.6
2	13.6	2464.7	23.9	1226.3	10.3	619.7
3	0.7	2753.1	5.7	1106.8	14.4	735.2
4	37.2	2761.4	4.4	1131.2	19.0	670.7
5	48.3	2509.4	4.4	1178.5	25.6	603.2
6	5.3	2801.7	9.0	1412.2	13.8	762.7
7	45.9	2798.4	10.6	1205.1	31.4	606.2
8	9.2	2813.2	6.0	1295.4	19.7	821.0
9	17.7	3058.1	23.7	1384.5	10.6	686.0
10	21.9	3232.7	15.4	1147.4	11.3	629.3
11	0.9	2455.3	18.0	1116.2	15.9	693.2
12	1.5	2936.8	6.5	1087.9	17.1	752.1
13	13.0	3354.2	7.6	1183.4	14.2	625.6
14	62.6	3061.4	8.8	1178.1	12.0	736.0
15	1.7	3122.9	32.1	1425.0	12.7	728.8
16	54.2	2995.7	6.3	980.7	9.9	785.8
17	7.5	2942.9	7.4	1200.2	17.9	721.4
18	57.7	2985.7	13.5	1134.0	9.9	869.9
19	22.1	2674.8	7.0	1216.5	12.9	642.2
20	9.9	2531.3	11.7	1090.9	29.1	758.2
Average=	21.6	2884.5	11.3	1192.0	17.1	703.9
Variance=	449.5	79709.7	60.1	12817.0	55.1	5728.1
Std. Dev.=	21.2	282.3	7.8	113.2	7.4	75.7
2*σ(95%conf.)	9.5	126.3	3.5	50.6	3.3	33.8

Table 3 Continued

		STEP CHANGE					
Number of Simulation	STF		STF		STF		
	EF= No Change	0.35 with step change	EF= No Change	0.4 with step change	EF= No Change	0.45 with step change	
1	12.1	505.4	24.3	507.2	23.1	449.9	
2	39.3	517.4	22.9	410.2	28.7	411.3	
3	14.9	527.8	23.0	594.9	27.6	393.6	
4	16.2	599.0	41.8	521.0	27.8	334.4	
5	20.6	545.6	25.6	406.7	25.7	429.8	
6	14.3	470.0	27.3	604.2	22.5	368.3	
7	18.6	663.2	21.4	483.9	30.9	567.0	
8	12.6	563.5	37.5	509.0	27.2	409.1	
9	21.2	533.1	24.6	476.8	19.7	400.9	
10	17.5	671.2	25.6	442.0	37.9	322.7	
11	24.5	578.6	19.2	295.2	21.8	433.2	
12	22.6	560.9	27.0	438.6	34.8	353.4	
13	17.4	645.6	17.3	591.3	28.1	383.1	
14	12.4	491.0	17.1	580.5	37.4	427.0	
15	15.9	455.3	40.5	625.4	32.5	373.2	
16	21.2	442.1	23.9	461.5	37.4	381.2	
17	10.7	634.9	19.6	479.5	24.3	356.2	
18	13.5	619.3	22.5	371.6	26.3	365.3	
19	17.0	570.5	17.0	619.3	22.1	474.1	
20	18.2	521.0	33.3	622.5	28.6	529.3	
Average=	18.0	555.8	25.6	502.1	28.2	408.1	
Variance=	39.3	4584.6	54.1	8692.2	29.9	3798.1	
Std. Dev.=	6.3	67.7	7.4	93.2	5.5	61.6	
2*σ(95%conf.)	2.8	30.3	3.3	41.7	2.4	27.6	

Figure 4.3. Plot of self-tuning ISE values with the introduction of a step change to the base value vs self-tuning ISE values with no change in the base steady state value.



In Figure 4.3

The diamonds represent a filtering factor value of 0.1

The squares represent a filtering factor value of 0.2

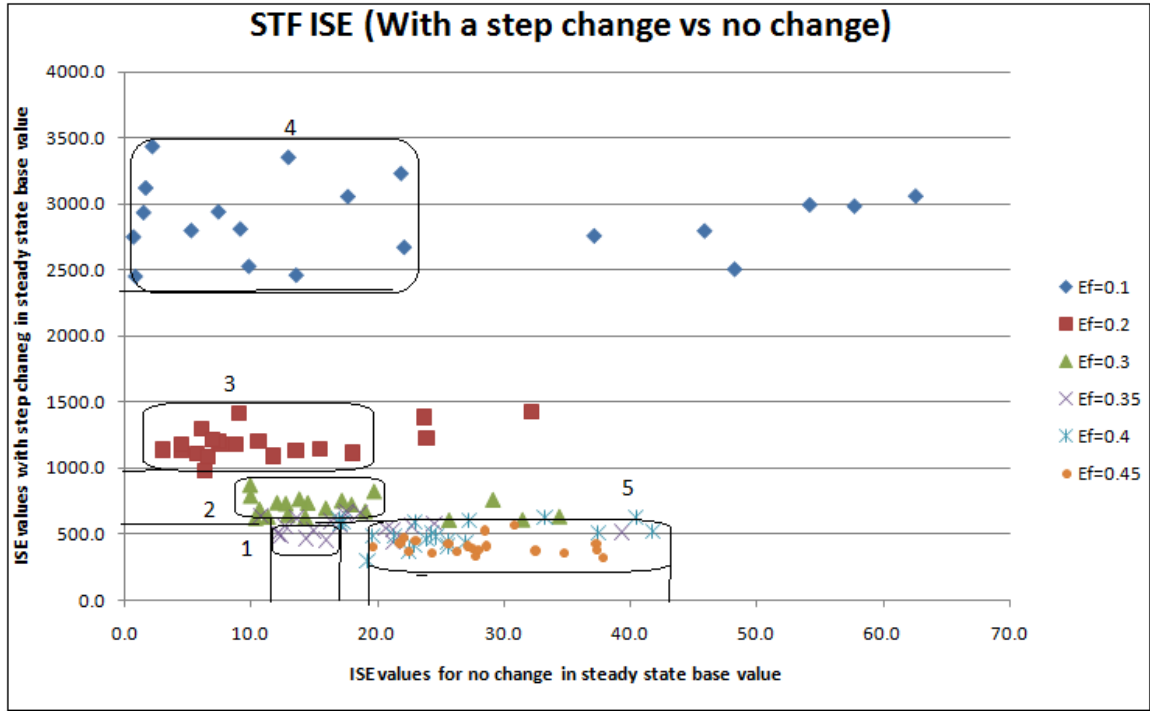
The triangles represent a filtering factor value of 0.3

The “X” represent a a filtering factor value of 0.35

The “✕” represent a filtering factor value of 0.4

The circles represent a filtering factor value of 0.45

Figure 4.3.1 Selection of optimum self-tuning filter tuning value



The chosen optimum filter tuning value for the self-tuning filter is 0.35 represented by the “X”. This is because the “X” enclosed by rounded rectangle 1 dominate the tuning value 0.3 (enclosed by rounded rectangle 2), 0.2 (enclosed by rounded rectangle 3) and 0.1 (enclosed by rounded rectangle 4) on the y-axis. Tuning value 0.35 (in rounded rectangle 1) also dominates the other tuning values 0.4 and 0.45 (enclosed by rounded rectangle 5) on the x-axis. Thus its choice as optimum tuning value.

4.4. The Kalman Filter

The Kalman filter results are as given in table 4.

Column 1 with numbers 1 to 20 shows the number of the simulation run. There was a total of 20 simulation runs done to generate each of the 20 results in Table 4 below.

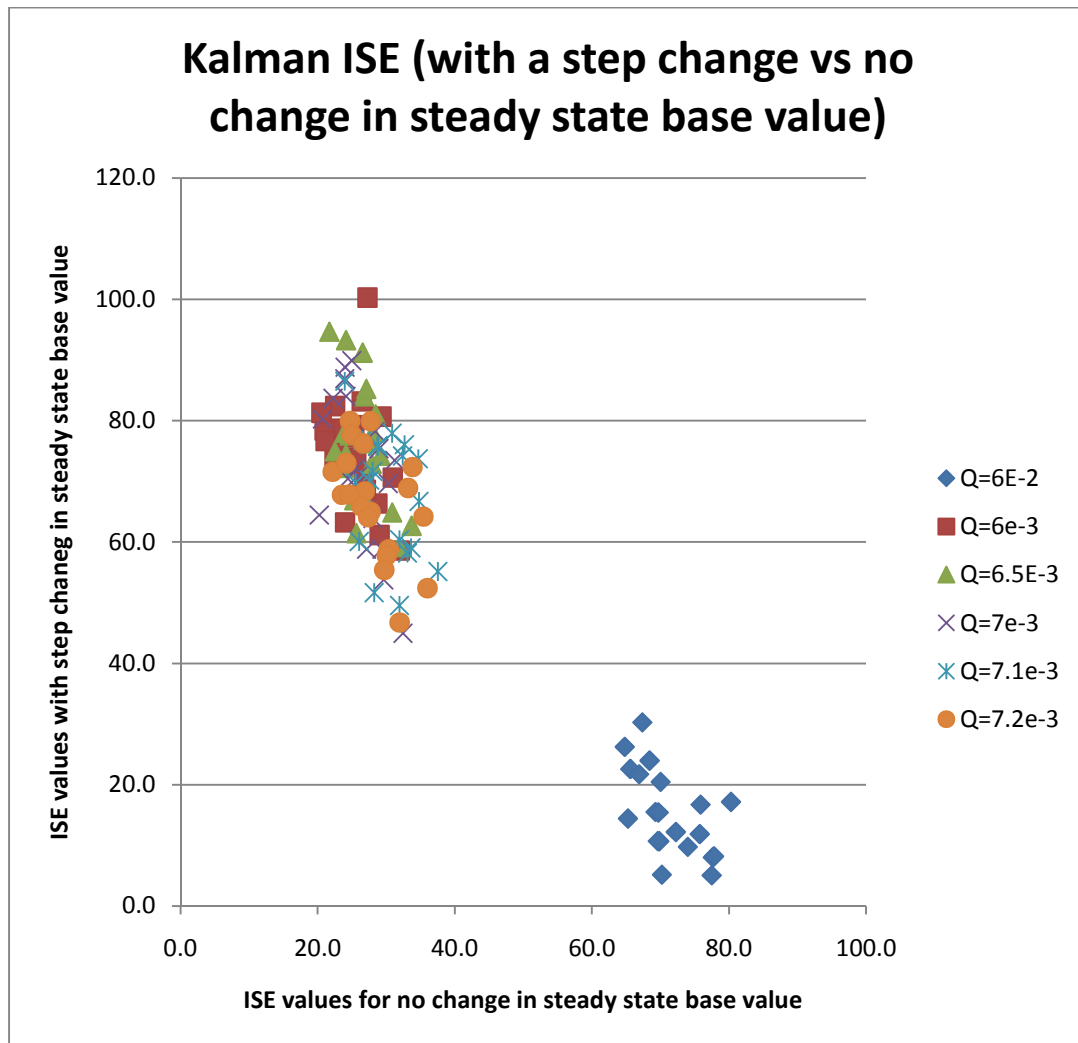
Table 4. Kalman ISE values to determine optimal filter tuning value (Q)

Number of Simulation	STEP CHANGE					
	Kalman 6.00E-02		Kalman 6.00E-03		Kalman 6.50E-03	
	Q= No Change	with step change	Q= No Change	with step change	Q= No Change	with step change
1	70.0	20.4	26.8	71.5	21.7	94.6
2	70.2	5.2	29.3	80.7	22.5	74.9
3	64.8	26.2	22.5	73.7	31.9	59.2
4	65.3	14.4	22.7	75.3	26.8	84.0
5	69.8	10.7	22.7	78.7	25.3	66.9
6	75.9	16.7	21.0	78.4	24.2	93.2
7	68.4	24.0	32.0	58.6	28.1	78.0
8	77.5	5.1	25.1	79.3	26.6	91.2
9	74.0	9.7	27.3	100.3	27.9	72.9
10	77.6	8.0	29.0	61.2	27.1	85.2
11	65.6	22.6	25.7	73.8	24.1	74.4
12	75.7	11.9	25.4	77.2	28.5	81.0
13	66.9	21.7	22.6	82.4	23.9	72.2
14	69.3	15.5	31.0	70.6	29.2	74.2
15	69.6	10.7	28.7	66.3	24.7	77.6
16	67.4	30.3	21.2	76.7	30.9	64.9
17	69.7	15.4	20.6	81.3	23.9	78.0
18	80.3	17.2	24.0	63.2	27.5	76.2
19	72.3	12.2	26.4	83.2	33.7	62.7
20	77.8	8.2	27.1	68.6	25.7	61.5
Average=	71.4	15.3	25.5	75.0	26.7	76.1
variance=	22.1	50.1	11.4	86.0	9.8	103.8
std. dev.=	4.7	7.1	3.4	9.3	3.1	10.2
2*σ(95%conf.)	2.1	3.2	1.5	4.1	1.4	4.6

Table 4 continued

Number of Simulation	STEP CHANGE					
	Kalman		Kalman		Kalman	
	Q= No Change	7.00E-03 with step change	Q= No Change	7.10E-03 with step change	Q= No Change	7.20E-03 with step change
1	20.7	80.3	32.4	74.2	33.9	72.4
2	28.1	63.9	27.0	67.6	32.0	46.7
3	27.1	58.8	28.8	75.9	22.2	71.6
4	24.4	70.4	28.7	75.8	24.7	80.0
5	28.9	75.4	25.5	69.8	30.4	58.8
6	22.3	83.7	28.0	71.6	27.7	65.0
7	25.0	89.9	37.6	55.1	24.2	73.0
8	25.8	70.8	27.6	70.2	23.5	67.8
9	30.3	69.6	33.1	58.2	33.2	68.9
10	24.0	88.8	28.3	51.7	27.7	79.9
11	25.6	72.2	34.8	66.7	25.0	77.7
12	29.6	53.8	26.1	60.1	30.1	57.8
13	20.2	64.4	33.6	59.0	26.9	68.4
14	28.4	78.2	32.0	60.4	26.4	65.9
15	24.1	84.0	30.9	77.9	29.7	55.4
16	32.4	45.0	34.7	73.7	26.6	76.3
17	31.2	73.5	24.0	86.5	36.0	52.4
18	25.9	77.6	26.6	76.8	24.6	67.8
19	24.0	86.9	31.9	49.5	35.4	64.2
20	29.4	58.8	32.7	76.1	27.4	64.1
Average=	26.4	72.3	30.2	67.8	28.4	66.7
variance=	11.6	147.3	13.3	98.3	16.4	82.0
std. dev.=	3.4	12.1	3.6	9.9	4.1	9.1
2*σ(95%conf.)	1.5	5.4	1.6	4.4	1.8	4.0

Figure 4.4 Plot of Kalman ISE values with the introduction of a step change to the base value vs Kalman ISE values with no change in the steady state base value.



In Figure 4.4

The diamonds represent a filtering factor value of 6E-2

The squares represent a filtering factor value of 6E-3

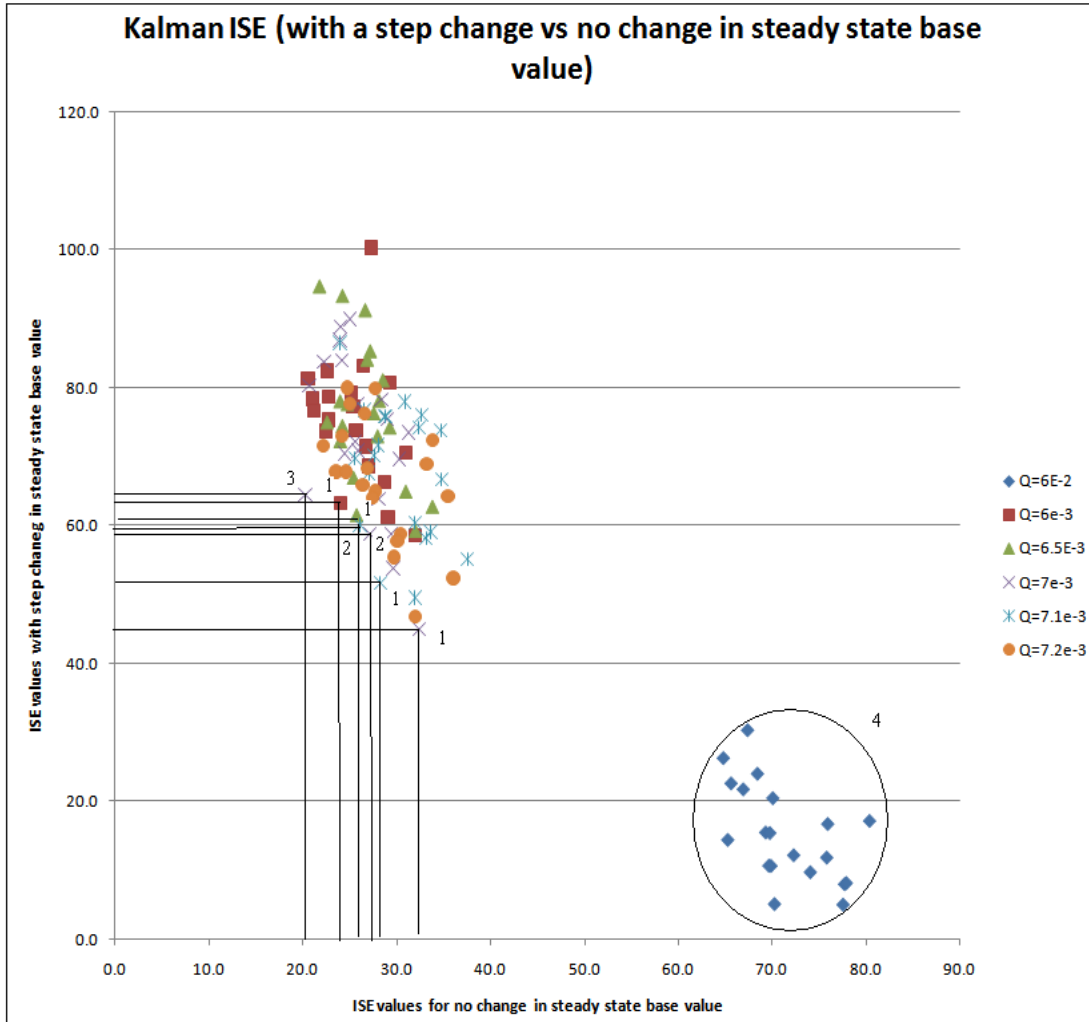
The triangles represent a filtering factor value of 6.5E-3

The “X” represent a a filtering factor value of 7E-3

The “ ✱ ” represent a filtering factor value of 7.1E-3

The circles represent a filtering factor value of $7.2E-3$

Figure 4.4.1 Selection of optimum Kalman filter tuning value



The non-dominated values are

1. The square representing a filtering factor value of $6E-3$ (There is 1 non dominated value as shown in Figure 4.4.1)
2. The triangle representing a filtering factor value of $6.5E-3$ (There is 1 non dominated value as shown in Figure 4.4.1)
3. The "X" representing a filtering value of $7E-3$ (There are 3 non-dominated value)

The chosen optimum tuning value is $7E-3$. This is because there are 3 non dominated “X” which is more than the other non dominated shapes. The diamonds enclosed by circle 4 are non chosen because they are all dominated on the x-axis by the other tuning values

4.5 Comparison of Results between the four filters

Part 1

Step Change

Having obtained the optimum filter tuning values discussed above, these values were then used for the comparison to obtain the filter with the lowest ISE.

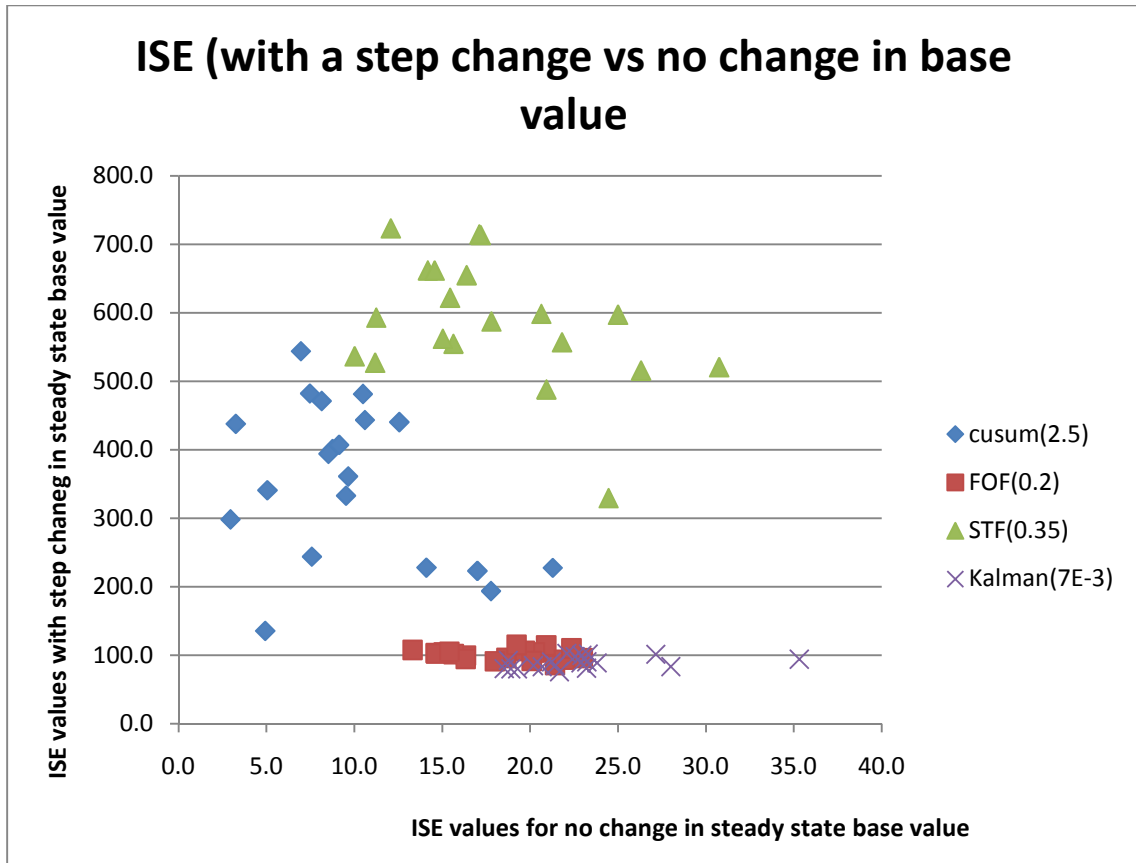
The result is shown below:

Column 1 with numbers 1 to 20 shows the number of the simulation run. Twenty simulation runs was done to generate each of the 20 results below:

Table 5. Comparison of ISE results of CUSUM, First-order, Self-tuning and Kalman filter for introduction of a step change to steady state base value.

Number of simulation	STEP CHANGE							
	CUSUM		FOF		STF		KALMAN	
	Trigger=	2.5	ff= 0.2		Ef= 0.35		Q= 7.00E-03	
	No change	with step change	No change	with step change	No change	with step change	No change change	with step change
1	6.9	494.4	18.1	93.7	16.0	616.2	25.5	88.3
2	8.8	401.0	20.9	114.3	15.6	554.7	28.0	83.6
3	9.5	332.9	18.7	96.2	11.2	593.0	35.3	94.2
4	7.0	543.6	20.3	104.1	24.5	329.7	23.8	88.8
5	10.6	443.3	21.4	85.7	21.8	556.9	20.2	85.0
6	8.5	394.0	19.7	106.4	10.0	536.5	18.8	92.7
7	4.9	135.7	20.1	91.7	26.3	515.7	27.2	101.7
8	3.2	437.8	23.0	95.8	14.2	661.8	22.6	98.4
9	21.3	227.6	18.0	91.2	16.4	654.7	23.2	90.5
10	10.5	481.3	21.9	93.6	17.2	713.2	22.9	99.8
11	8.1	471.1	16.4	99.5	15.0	562.1	20.7	83.4
12	17.8	193.8	15.7	101.7	12.1	723.0	18.5	80.4
13	14.1	228.0	19.2	115.4	17.1	714.3	21.4	85.2
14	12.6	440.3	15.1	104.0	30.7	520.5	21.7	76.0
15	9.1	407.0	14.6	103.5	11.2	527.1	18.9	80.3
16	5.1	340.9	13.3	107.8	25.0	597.1	23.3	101.8
17	3.0	298.3	22.4	110.2	14.6	661.6	23.2	81.2
18	17.0	223.2	16.3	94.5	20.6	598.4	22.1	102.7
19	7.6	243.9	14.6	102.4	20.9	488.1	19.3	80.2
20	7.5	482.1	15.4	105.2	15.4	621.9	21.2	90.4
Average=	9.6	361.0	18.3	100.9	17.8	587.3	22.9	89.2
Variance=	23.3	14116.0	8.6	63.6	31.5	8574.8	15.3	68.8
Std. Dev.	4.8	118.8	2.9	8.0	5.6	92.6	3.9	8.3
2*σ(95%conf.)=	2.2	53.1	1.3	3.6	2.5	41.4	1.7	3.7

Figure 4.5 Plot of ISE values with the introduction of a step change to the steady state base value vs ISE values with no change introduction for the CUSUM, First-order , STF, and Kalman filter.



In Figure 4.5

The diamonds represent the CUSUM filter

The squares represent the first-order filter

The triangles represent the self-tuning filter

The “X” represent the Kalman filter

Figure 4.5.1 Comparison of filter performance for step change

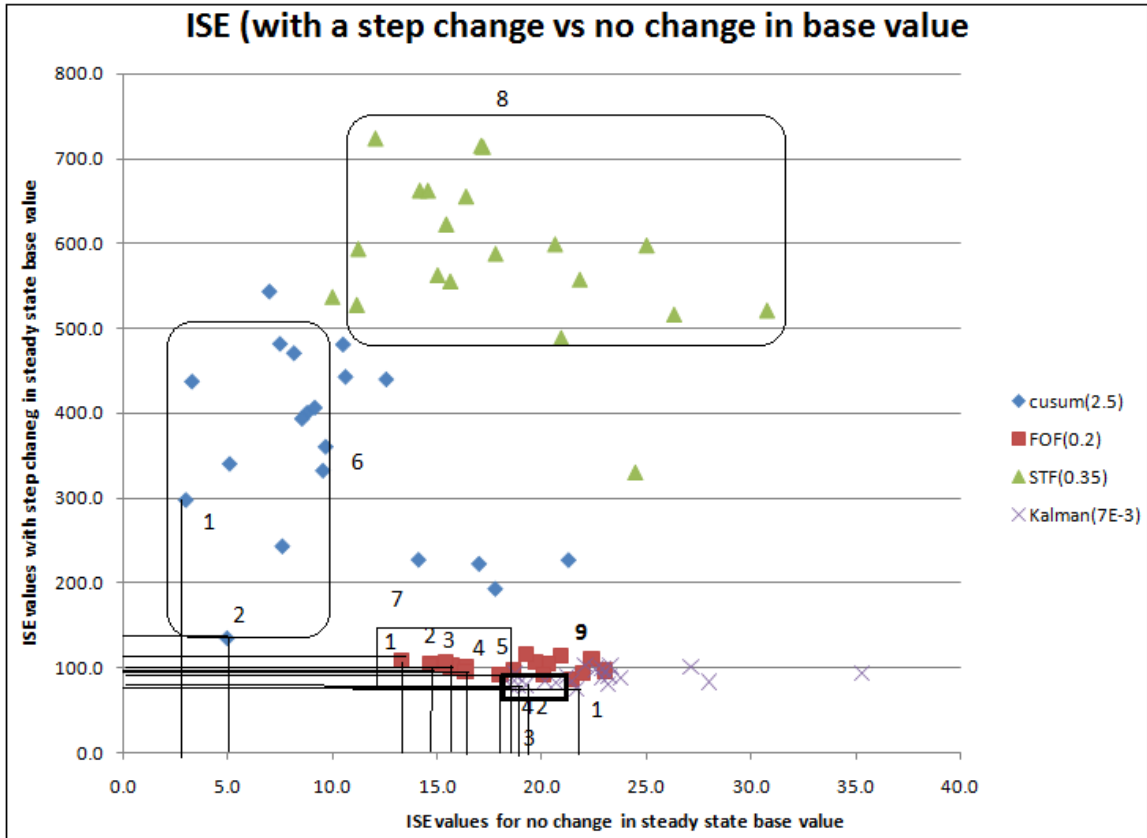


Figure 4.5.1 shows that the self-tuning filter (rounded rectangle 8) is dominated by the first-order filter (in rectangle 7) the CUSUM filter (in rounded rectangle 6) and by the Kalman filter (thick rectangle 9) on the y-axis, While the CUSUM filter is dominated by the first-order filter and Kalman filter on the y-axis. The Kalman filter (enclosed by thick rectangle 7) does well on the y-axis but is dominated by the first order filter (in rectangle 7) on the x-axis. Hence, from Figure 4.5.1 considering rectangles 6, 7 and 8, it can be concluded that the CUSUM filter performs best for a steady state process with no change in base value (i.e. is not dominated on the x-axis during the steady state window), while the Kalman filter performs better for a process that involves a step change to a steady state process.

Part 2 Ramp

Based on the optimum filter filtering values from the ramp tests, a series of simulation runs was done involving the introduction of a ramp change to the steady state base value and the ISE values for the steady state portion as well as the portion with the change was calculated. The results are given in Table 6.

Comparison of the four filter simulation results:

Table 6 Comparison of ISE results of CUSUM, First-order, Self-tuning and Kalman filter for introduction of a ramp change to steady state base value.

Number of simulation	RAMP CHANGE							
	CUSUM		FOF		STF		KALMAN	
	Trigger= 2.5		ff= 0.2		Ef= 0.2		Q= 6.50E-03	
	No Change	with ramp change	No change	with ramp change	No change	with ramp change	No change	with ramp change
1	7.2	263.6	18.0	100.5	16.0	318.0	20.3	84.7
2	17.5	185.4	22.5	95.3	22.2	206.1	17.9	74.1
3	10.4	228.0	24.3	101.0	10.5	203.2	26.6	74.3
4	29.5	190.8	18.3	105.5	40.5	244.0	21.7	66.9
5	6.2	204.4	26.3	93.1	44.4	280.7	21.9	73.3
6	6.5	164.3	18.8	106.7	11.8	231.9	23.3	83.4
7	5.4	169.2	28.2	110.0	46.8	322.6	20.6	65.7
8	28.5	208.1	22.3	106.3	18.3	186.5	21.5	78.2
9	7.9	184.1	27.1	101.6	24.0	271.5	19.1	70.2
10	9.3	195.3	18.6	108.1	28.8	293.8	30.6	76.1
11	12.4	210.2	23.9	106.7	13.6	233.1	38.9	77.0
12	11.5	179.1	20.9	90.2	11.0	229.3	32.9	64.3
13	8.2	192.6	25.1	105.8	21.9	239.4	21.2	75.2
14	15.7	200.2	19.3	90.7	62.9	262.0	20.3	66.4
15	11.1	226.7	19.5	97.5	10.8	274.1	21.4	86.4
16	4.9	264.1	18.0	109.9	48.9	327.7	18.6	67.9
17	8.9	185.3	21.0	109.7	16.0	327.9	17.7	73.4
18	11.4	252.9	21.2	116.6	46.8	277.7	17.3	79.5
19	4.0	203.5	24.3	100.5	30.9	225.3	22.1	74.3
20	9.7	174.8	26.4	101.9	17.7	233.4	19.4	69.3
Average=	11.3	204.1	22.2	102.9	27.2	259.4	22.7	74.0
Variance=	48.3	866.1	11.0	48.8	248.3	1857.5	30.7	40.2
Std. Dev.=	7.0	29.4	3.3	7.0	15.8	43.1	5.5	6.3
2*σ=	13.9	58.9	6.6	14.0	31.5	86.2	11.1	12.7

Column 1 shows the number of the simulation run, the other columns show the name of the filter, the filter tuning value and whether a change was introduced or not. The first of each pair of the two columns shows the ISE values for a steady state signal with no change introduced. Each of the second columns shows the ISE values with the introduction of a ramp change.

Figure 4.6 Plot of ISE values with the introduction of a ramp change to the steady state base value vs ISE values with no change introduction for the CUSUM, First-order, STF, and Kalman filter.

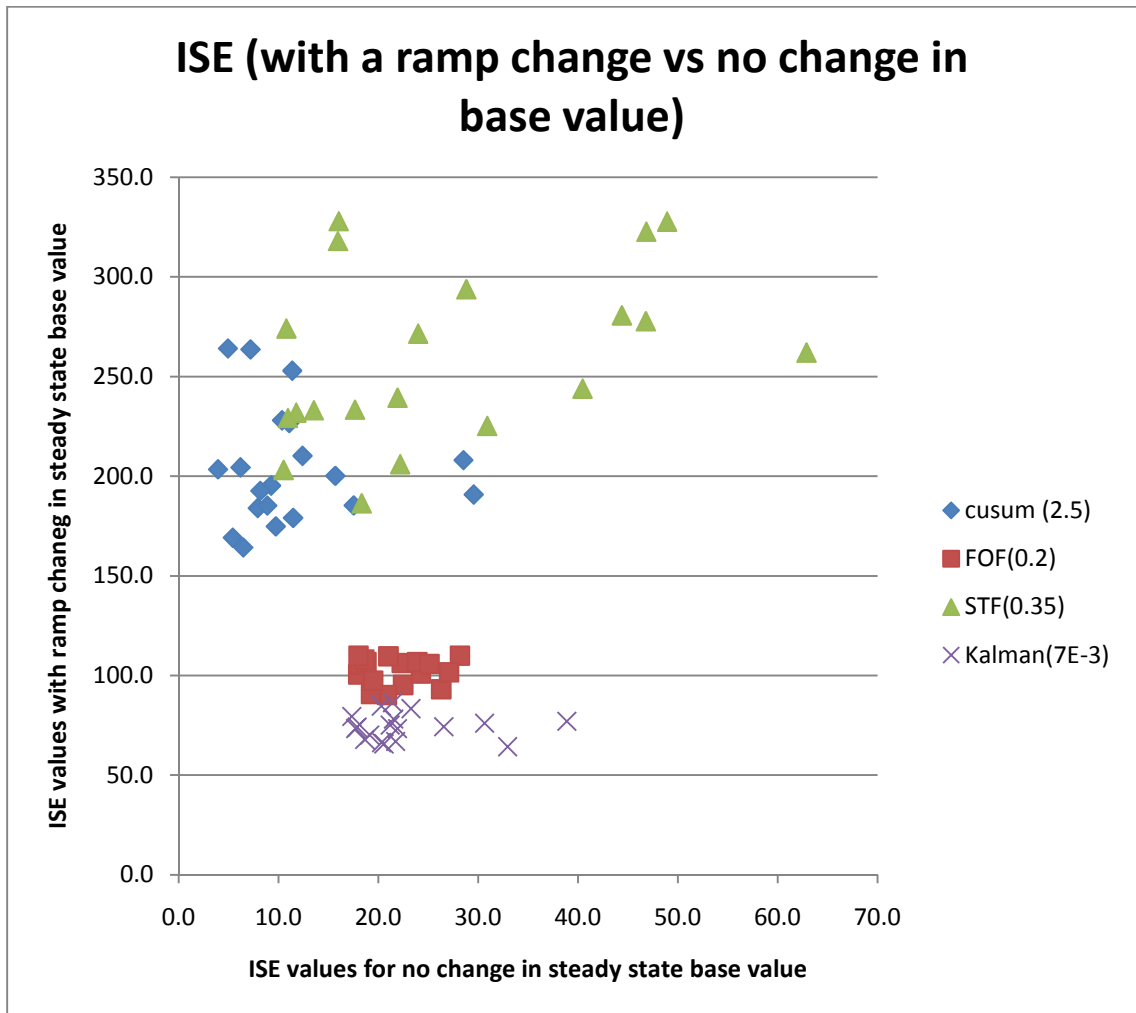


Figure 4.6 is a plot of the ISE values for simulation runs with a ramp change introduced to the base point on the vertical axis, while the horizontal axis shows values for the steady state base point with no change introduced.

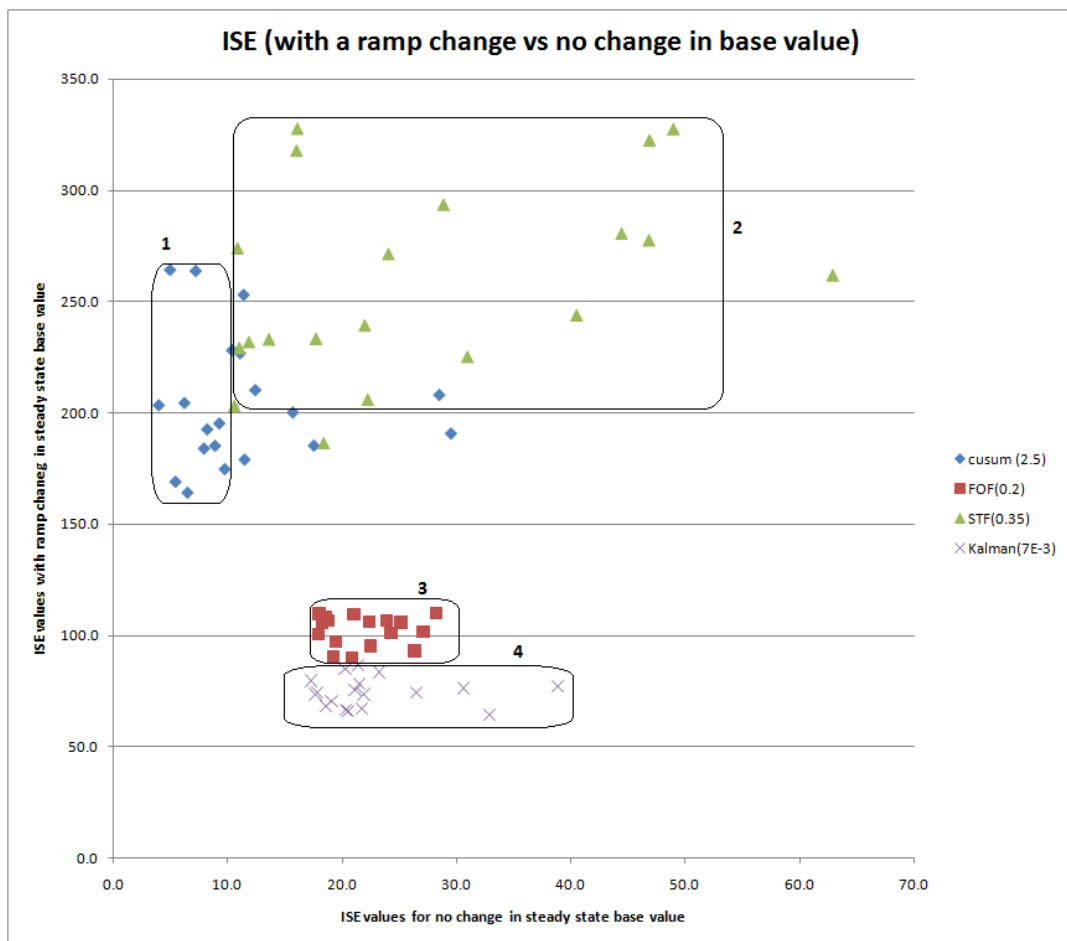
The diamonds represent the CUSUM ISE values

The squares represent the FOF ISE values

The triangles represent the STF ISE values

The "X" represent the Kalman ISE values

Figure 4.6.1 Comparison of filter performance for ramp change



It is observed in Figure 4.6.1 that the Kalman filter (in rounded rectangle 4) dominates the first-order filter (in rounded rectangle 3) and that the CUSUM filter (in rounded rectangle 1) dominates the self-tuning filter (in rounded rectangle 2) on the x-axis. From this observation, it can be concluded that the CUSUM filter performs best for a steady state process with no change in base value (i.e. is not dominated on the x-axis), while the Kalman filter performs best for a process that involves a ramp change to the steady state process (i.e. is not dominated during the ramp change window on the y-axis).

Part 3 (Oscillatory signal)

The result of simulations with the introduction of an oscillatory change is shown below. Column 1 shows the number of the simulation run, the other columns show the name of the filter, the filter tuning value and whether a change was introduced or not. The first of each pair of columns shows the ISE values for a steady state signal with no change introduced. Each of the second columns shows the ISE values with the introduction of an oscillatory change.

Table 7. Comparison of ISE results of CUSUM, First-order, Self-tuning and Kalman filter for introduction of a step change to steady state base value.

		OSCILLATORY CHANGE							
Number of Simulation	CUSUM		FOF		STF		KALMAN		
	Trigger= 2.5		ff= 0.2		Ef= 0.35		Q= 7.00E-03		
	No change	with Oscillatory Change	No Change	with Oscillatory Change	No Change	with Oscillatory Change	No Change	with Oscillatory Change	
1	7.9	45.8	20.2	18.3	16.0	13.8	22.6	20.7	
2	15.0	54.3	17.1	17.7	14.4	20.4	20.1	15.5	
3	3.7	48.9	17.0	16.8	11.5	20.6	23.7	19.4	
4	10.3	55.6	15.3	17.3	17.3	23.4	23.2	19.6	
5	9.2	54.0	18.2	23.9	13.9	16.0	23.5	21.1	
6	19.2	38.4	18.1	17.7	10.3	12.3	20.2	18.9	
7	6.5	47.6	15.2	13.9	22.1	15.7	22.6	20.7	
8	5.8	38.3	16.7	16.3	13.6	15.6	16.3	22.0	
9	3.4	43.2	15.9	13.5	14.9	17.5	19.8	19.1	
10	15.5	63.0	18.3	14.6	15.3	16.1	24.3	16.0	
11	7.1	51.0	20.5	21.1	16.0	15.6	24.8	20.4	
12	9.2	42.7	21.3	11.9	12.9	27.6	20.5	21.2	
13	4.3	60.0	20.1	15.9	15.8	16.7	22.6	19.1	
14	22.3	41.3	17.4	18.5	22.9	16.4	25.6	18.8	
15	0.2	54.2	22.8	15.4	11.8	17.2	18.7	18.5	
16	6.5	38.2	18.6	21.2	19.6	14.5	20.3	16.4	
17	1.8	45.1	20.4	14.2	14.4	19.4	27.3	26.6	
18	3.5	53.0	20.4	18.1	13.7	14.0	22.9	19.1	
19	14.7	46.3	15.6	16.2	18.8	17.8	23.3	19.9	
20	15.6	54.6	15.2	17.1	15.8	21.9	19.9	23.5	
Ave=	9.1	48.8	18.2	17.0	15.5	17.6	22.1	19.8	
Variance=	37.1	52.6	5.1	8.2	10.8	13.5	6.7	6.3	
Std. Dev.=	6.1	7.3	2.3	2.9	3.3	3.7	2.6	2.5	
2*σ=	12.2	14.5	4.5	5.7	6.6	7.3	5.2	5.0	

Figure 4.7 Plot of ISE values with the introduction of an oscillatory change to the steady state base value vs ISE values with no change introduction for the CUSUM, First-order , STF, and Kalman filter.

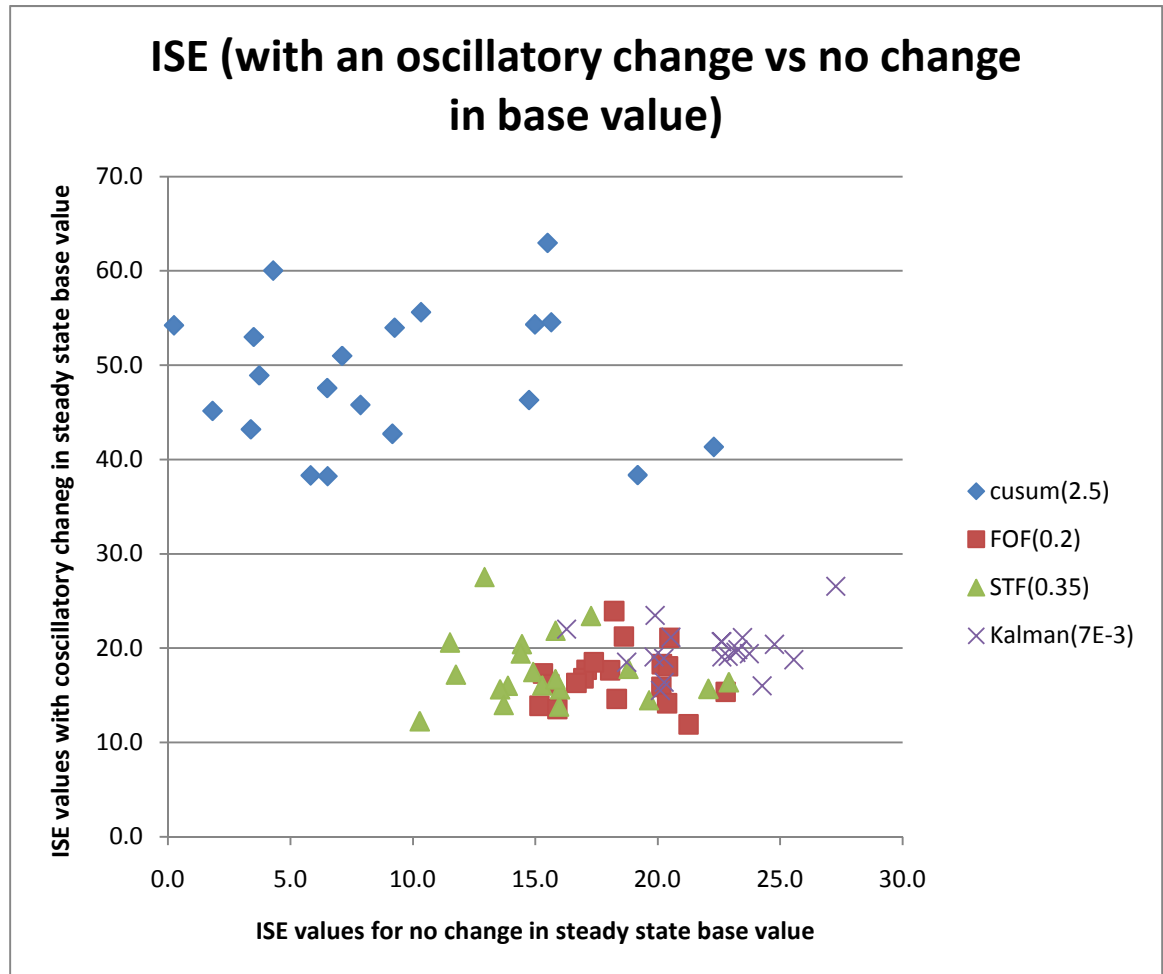


Figure 4.7 is a plot of the ISE values for simulation runs with an oscillatory change introduced to the base value on the vertical axis, while the horizontal axis shows values for the steady state base value with no change introduced.

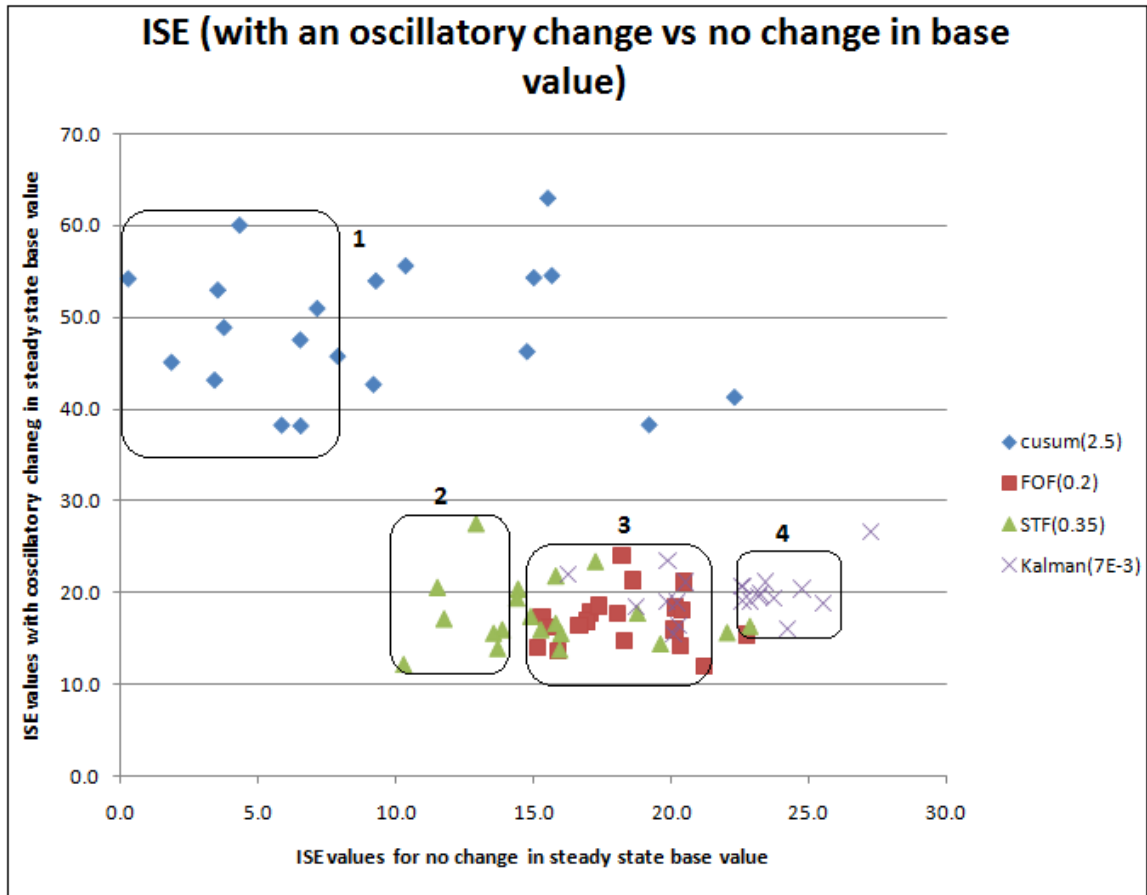
The diamonds represent the CUSUM ISE values

The squares represent the FOF ISE values

The triangles represent the STF ISE values

The “X” represent the Kalman ISE values

Figure 4.7.1 Comparison of filter performance for oscillatory change



From Figure 4.7.1, it is observed that the CUSUM filter enclosed in rounded rectangle 1 performs well on the x-axis. However, the CUSUM filter is dominated on the y-axis by the self-tuning filter, the Kalman filter, and the first-order filter. The self-tuning filter enclosed in rounded rectangle 2 dominates first-order filter (in rounded rectangle 3) and the Kalman filter (in rounded rectangle 4) on the x-axis. Hence, it can be concluded that the CUSUM filter performs best for a steady state process with no change to the steady state base value, while the self-tuning filter performs best for a process that involves an

oscillatory change to the steady state process. This is because the self-tuning filter is not dominated on the y-axis by any of the other filters, but it dominates the first-order and Kalman on the x-axis. Thus the choice of the self-tuning filter.

4.2 EASE OF UNDERSTANDING THE FILTERS

Based on the human ease of understanding, the four filters are grouped into four levels.

With level 1 having the relatively easiest understanding and level 100 being the relatively most complex.

Table 8. Ease of understanding the filters

Filter	Relative rank based on ease of understanding
First order Filter (FOF)	1.
Self-tuning filter (STF)	4.
CUSUM	5.
Kalman filter	100.

FOF is assigned a rank of 1 for ease of understanding because it involved relatively simple mathematics of multiplication, addition and subtraction.

STF is assigned a rank of 4 because it involves relatively higher computations of exponential, conditional statement and more variables with assigned values.

CUSUM filter is assigned a rank of 5 because it involves relatively higher computations than the FOF and STF with even more variables, conditional statements and calculations.

Kalman filter is assigned a rank of 100 because it involves the relatively most complex calculations of the four filters. It involves the use of matrices and matrix operations for the time update and measurement update operations.

4.3 Cost

The results for analysis based on arithmetic operations are as given below. The first column shows the number of arithmetic operations (such as “*”, “+”, “-”, “^” etc) involved in the filter equation. The second column shows the actual equation. The third column shows the number of “Get” operations required to read a certain input. And, the

final column shows the number of “put” operations required to assign a value to a certain variable.

4.3.1 Ranking of filters based on cost analysis derived from the total operations required in the filter equation.

Table 9 Evaluation of operations required in CUSUM filter

Equation Number	Equation Evaluated	1	2	3
	Equation	Arithmetic Operation	Get(Read)	Put
1	$n=n+1$	1	1	1
2	$V=ff1*v+ff2*(xx-xxold)^2$	5	5	1
3	$CUSUM=CUSUM+xx-xxspc$	2	3	1
4	$xmeas=xxspc$		1	1
5	$(xx=xmeas)$		1	1
6	$(xxold=xx)$		1	1
	Sum	8	12	6
	Total	26		

Table 10 Evaluation of operations required in first-order filter

Equation Number	Equation	Evaluation 1 (Arithmetic operation)	Evaluation 2 Get (Read)	Evaluation3 Put
1	$xmeasff=ff*(xmeas)+(1-ff)*(xmeasff)$	4	4	1
	Total	9		

$xmeas$ is the signal measured value

$xmeasff$ is the filtered value

Table 11 Evaluation of operations required in self-tuning filter

Equation Number	Equation Evaluated	1	2	3
		(Arithmetic operation)	Get (Read)	Put
1	$Ef=2*\text{sigmad}$	1	1	1
2	$\text{deltaf}=0.9*\text{deltaf}+0.1*(\text{xmeas}-\text{xmeasold})^2$	5	3	1
3	$\text{lambda}=(0.5+1.1668*\text{deltaf}/E_f^2)^{-1}$	4	2	1
4	$(\text{xmeasold}=\text{xmeas})$		1	1
5	$(\text{xmeasf}=\text{lambda}*\text{xmeas}+(1-\text{lambda})*\text{xmeasf})$	4	4	1
	Sum	14	11	5
	Total	30		

sigmad is the standard deviation of the measurement

xmeasf is the filtered value

Table 12 Evaluation of operations required in Kalman filter

Equation Number	Equation Evaluated	Arithmetic operation	Get(Read)	Put
1	$\hat{x}_{i-1} = \hat{x}_{i-1}$	1	2	1
2	$P_i = P_{i-1} + Q$	2	3	1
3	$\hat{x}_i = \hat{x}_{i-1} + k_i(x_{meas} - \hat{x}_{i-1})$	3	4	1
4*	$k_i = P_i / (P_i + R)$	3	3	1
5	$P_i = (1 - k_i) * P_i$	2	3	1
6	$x_{measkf} = \hat{x}_i$		2	1
7	$R = (\text{value})$		1	1
8	$Q = (\text{Value})$		1	1
	Sum	11	15	8
	Total	34		

In equation 4* the inversion operation would require several steps of arithmetic operation when it involves matrices. (This shows that the Kalman filter will involve much more operations than indicated above)

x_{measkf} is the kalman filtered value

x_{meas} is the signal measurement value

Given the results above the filters are ranked on a scale of one to four, with one being the cheapest to operate and four being the most expensive.

Table 13. Ranking of filter in increasing order of relative cost.

Filter	Relative Rank (based on total number of operations)	Total number of operations
FOF	1	9
CUSUM	3	26
STF	3	30
Kalman	4	34

From the above scale, the first-order filter is the cheapest to operate in terms of arithmetic operations, while the Kalman filter is the most expensive in terms of arithmetical operations.

3.2. Ranking of filters based on cost analysis derived from the Number of Variables used in Filter Equation.

Based on the number of variables required in the filter equation, the four filters are ranked as shown below, with a rank of 1 requiring the least number of variables and 4 requiring the most number of variables:

Table 14 Ranking of filter based on the required number of memory location/required variables

Filter	Rank based on required number of variables	Number of memory location (or variables required)
FOF	1	3
STF	2	7
Kalman	3	10
CUSUM	4	12

From the result above, The First-order filter requires the least number of variables and thus is the least expensive in terms of the required number of variables while the CUSUM filter requires the most number of variables and is thus the most expensive in terms of the required number of variables.

CHAPTER V

CONCLUSION AND RECOMMENDATION

The CUSUM filter is the best filter to use in a steady state process involving no change. In a process that involves a step change, the Kalman filter is the best filter to use. A process that involves a ramp change is best filtered using the Kalman filter. And, an oscillatory process is best filtered using the self-tuning filter.

First-order filter is the easiest to understand, while the Kalman filter is the most complex to understand.

The first-order filter is the cheapest to operate in terms of arithmetic operations, while the Kalman filter is the most expensive in terms of arithmetical operations.

The first-order filter is the least expensive considering required number of variables, while the CUSUM filter requires the most number of variables and is thus the most expensive in terms of the required number of variables.

It is recommended that further analysis involving different signal to noise ratio should be carried out. Also, the filter speed of tracking the signal be analyzed in subsequent studies.

REFERENCES

1. Cao, S. and Rhinehart, R. R. (1997). "A Self-Tuning Filter." Journal of Process Control 7(Number 2): 139-148.
2. Douglas, C. M. (1993). Process/Industrial Instruments and Controls Handbook. N Y, McGraw-Hill.
3. Harris, T. J. and Ross, W. J. (1991). "Statistical Process Control Procedures for Correlated Observations." Canadian Journal of Chemical Engineering 69(1): 48-57.
4. Koenig, D. M. (2009). Practical Control Engineering: A Guide for Engineers, Managers, and Practitioners. New York, McGraw-Hill.
5. Maybeck, P. M. (1979). Stochastic Models, Estimation and Control. New York, Academic Press.
6. Muthiah, N. and Rhinehart, R. R. (2009). "Evaluation of a Statistically Based Controller Override on a Pilot-Scale Flow Loop." ISA Transactions: 31.
7. Welch, G and Bishop, G. (2001). An Introduction to the Kalman Filter. NC, ACM Inc.
8. Vu, Ky M. (2007). The ARIMA and VARIMA Time series: Their Modelings, Analyses and Applications. Ottawa ON, Library and Archives Canada Cataloguing in Publication.
9. Zu, X., Fredendall, L. D. and Douglas, T. J. (2008). "The Evolving Theory of Quality Management: The role of Six Sigma." Journal of Operations Management 26: 630–650.

APPENDIX

A

VBA codes used in the simulation of steady state signal with the addition of a step change

'VBA codes written with help from Dr. R.R. Rhinehart
'Department of Chemical Engineering, Oklahoma State University
'Daniel Fakunle
'MSC Research Project
'10-13-09
'A sub program to generate noisy signal and filter the noise using a
'cusum filter,first order filter and Kalman filter

Dim t As Single
Dim sigmad As Single
Dim dt As Single
Dim i As Single
Dim sp As Single
Dim xmeas As Single
Dim ff As Single
Dim kc As Single
Dim taud As Single
Dim n As Single
Dim xx As Single
Dim xxold As Single
Dim xxspc As Single
Dim v As Single
Dim cusum As Single
Dim m As Single
Dim ff1 As Single
Dim ff2 As Single
Dim Trigger As Single
Dim xhatminus(1000)
Dim Ef As Single
Dim xhat(1000)
Dim Pminus(1000)
Dim P(1000)
Dim k(1000)

Dim xkp As Single

```

Dim pkp As Single
Dim xold As Single
Dim pkold As Single
Dim Q As Single
Dim R As Single
Dim kk As Single
Dim xk As Single
Dim pk As Single
Dim xmeaskf As Variant 'Kalman filtered value
Dim deltaf As Single
Dim lambda As Single
Dim xmeasold As Single
Dim xmeassf As Single 'Self tuning filtered value
Dim xmeasff As Single 'First order filtered value
Dim xmeasc As Single 'cusum filtered value

```

```

Sub signal()
Application.ScreenUpdating = False

```

```

'Initial values of constants
ff = Sheet3.Cells(7, 24)

```

```

dt = Sheet3.Cells(3, 24)
Trigger = Sheet3.Cells(8, 24)
kp = Sheet3.Cells(11, 24)
Ef = Sheet3.Cells(10, 24)
'kalman filter constants
xhat(0) = 1
Pminus(0) = 1
Randomize

```

```

'Main Program
For i = 1 To 1000 'For loop
t = dt * i
x = 3
If i > 500 Then x = 10
Call Process(x, xmeas)

```

```

'Call filter
Call ffilter(xmeas, xmeasff)
Call cusumf(xmeas, xmeasc)
Call kalman(xmeas, xmeaskf)
Call selftuningfilter(xmeas, xmeassf)
'Print out observable results

```

```

'Application.ScreenUpdating = False
Sheet3.Cells(2 + i, 1).Value = t
Sheet3.Cells(2 + i, 3).Value = xmeas
Sheet3.Cells(i + 2, 4).Value = xmeasff 'STF
Sheet3.Cells(i + 2, 5).Value = xmeasc 'Cusum
Sheet3.Cells(i + 2, 16).Value = xmeaskf 'Kalmnan
Sheet3.Cells(i + 2, 17).Value = xhat(i)
Sheet3.Cells(i + 2, 18).Value = Pminus(i)
Sheet3.Cells(i + 2, 19) = k
Sheet3.Cells(i + 2, 20) = xhatminus(i)
Sheet3.Cells(i + 2, 21) = P(i)
Sheet3.Cells(i + 2, 22) = xmeasff 'FOF filter
'Sheet3.Cells(i + 2, 44) = Ef 'FOFilter

```

```

Next i
'Call signal2
Application.ScreenUpdating = True
End Sub

```

'FIRST ORDER FILTER

```

Sub ffilter(xmeas, xmeasff)
  xmeasff = ff * (xmeas) + (1 - ff) * (xmeasff)
End Sub

```

'CUSUM FILTER

```

Sub cusumf(xmeas, xmeasc)
  If i = 1 Then
    n = 0
    xx = 0
    xxold = 0
    xxspc = 0
    v = 0
    cusum = 0
    m = 11
    ff1 = ((m - 2) / (m - 1))
    ff2 = (1 - ff1) / 2
  End If

  xx = xmeas
  n = n + 1
  v = ff1 * v + ff2 * (xx - xxold) ^ 2
  xxold = xx
  cusum = cusum + xx - xxspc
  If (Abs(cusum) > Trigger * Sqr(v * n)) Then
    xxspc = xxspc + cusum / n
  End If
End Sub

```



```

    cusum = 0
    n = 0
End If
xmeasc = xxspc
End Sub

```

```

Sub kalman(xmeas, xmeaskf)

```

```

'KALMAN FILTER

```

```

R = Sheet3.Cells(9, 24).Value

```

```

Q = Sheet3.Cells(4, 24).Value

```

```

'Time Update

```

```

xhatminus(i) = xhat(i - 1)

```

```

Pminus(i) = P(i - 1) + Q

```

```

'Measurement Update

```

```

k(i) = Pminus(i) / (Pminus(i) + R)

```

```

xhat(i) = xhatminus(i) + k(i) * (xmeas - xhatminus(i))

```

```

P(i) = (1 - k(i)) * Pminus(i)

```

```

xmeaskf = xhat(i)

```

```

'xold = xk

```

```

End Sub

```

```

Sub Process(x, xmeas)

```

```

taun = Sheet3.Cells(5, 24)

```

```

    If taun = 0 Then taun = 0.0001

```

```

sigmad = Sheet3.Cells(6, 24)

```

```

alpha1 = Exp(-dt / taun)

```

```

beta0 = 1 - alpha1

```

```

sigmaw = sigmad * Sqr((1 + alpha1) / beta0)

```

```

xnoise = sigmaw * Sqr(-2 * Log(Rnd)) * Sin(2 * 3.14159 * Rnd) 'Gaussian distributed
noise

```

```

    sp = x

```

```

    xmeas = x + xnoise

```

```

End Sub

```

```

Sub selftuningfilter(xmeas, xmeassf)

```

```

deltaf = 0.9 * deltax + 0.1 * (xmeas - xmeasold) ^ 2

```

```

lambda = 1 / (0.5 + 1.1668 * deltax / Ef ^ 2)

```

```

If lambda > 1 Then lambda = 1

```

```

xmeasold = xmeas

```

```

xmeassf = lambda * xmeas + (1 - lambda) * xmeassf

```

```

End Sub

```

APPENDIX

B

VBA codes used in the simulation of steady state signal with the addition of a ramp change

'VBA codes written with help from Dr. R.R. Rhinehart
'Department of Chemical Engineering, Oklahoma state university
'Daniel Fakunle
'MSC Research Project
'10-13-09
'A sub program to generate noisy signal and filter the noise using a
'cusum filter,first order filter and Kalman filter

Sub signalramp()
Application.ScreenUpdating = False

'Initial values of constants
ff = Sheet4.Cells(7, 24)

dt = Sheet4.Cells(3, 24)
Trigger = Sheet4.Cells(8, 24)
Ef = Sheet4.Cells(10, 24)

'kalman filter constants
xhat(0) = xmeas
Pminus(0) = 1
Randomize
dt = 0.1

'Main Program
For i = 1 To 1000 'For loop
t = dt * i
x = 3 + (t - 50)
If i > 500 Then x = 3 + (t - 50)
Call Process(x, xmeas)

'Call filter
Call ffilter(xmeas, xmeasff)
Call cusumf(xmeas, xmeasc)
Call kalman(xmeas, xmeaskf)
Call selftuningfilter(xmeas, xmeassf)
'Print out observable results

```

Application.ScreenUpdating = False
Sheet4.Cells(2 + i, 1).Value = t
Sheet4.Cells(2 + i, 2).Value = x
Sheet4.Cells(2 + i, 3).Value = xmeas
Sheet4.Cells(i + 2, 4).Value = xmeassf 'STF
Sheet4.Cells(i + 2, 5).Value = xmeasc 'Cusum
Sheet4.Cells(i + 2, 16).Value = xmeaskf 'Kalmnan
Sheet4.Cells(i + 2, 22) = xmeasff 'FOFilter

```

```

Next i
Application.ScreenUpdating = True
End Sub

```

'FIRST ORDER FILTER

```

Sub ffilter(xmeas, xmeasff)
  xmeasff = ff * (xmeas) + (1 - ff) * (xmeasff)
End Sub

```

'CUSUM FILTER

```

Sub cusumf(xmeas, xmeasc)
  If i = 1 Then
    n = 0
    xx = 0
    xxold = 0
    xxspc = 0
    v = 0
    cusum = 0
    m = 11
    ff1 = ((m - 2) / (m - 1))
    ff2 = (1 - ff1) / 2
  End If

  xx = xmeas
  n = n + 1
  v = ff1 * v + ff2 * (xx - xxold) ^ 2
  xxold = xx
  cusum = cusum + xx - xxspc
  If (Abs(cusum) > Trigger * Sqr(v * n)) Then
    xxspc = xxspc + cusum / n
    cusum = 0
    n = 0
  End If
  xmeasc = xxspc
End Sub

```

```
Sub kalman(xmeas, xmeaskf)
```

```
'KALMAN FILTER
```

```
R = Sheet4.Cells(9, 24).Value
```

```
Q = Sheet4.Cells(4, 24).Value
```

```
'Time Update
```

```
xhatminus(i) = xhat(i - 1)
```

```
Pminus(i) = P(i - 1) + Q
```

```
'Measurement Update
```

```
k(i) = Pminus(i) / (Pminus(i) + R)
```

```
xhat(i) = xhatminus(i) + k(i) * (xmeas - xhatminus(i))
```

```
P(i) = (1 - k(i)) * Pminus(i)
```

```
xmeaskf = xhat(i)
```

```
End Sub
```

```
Sub Process(x, xmeas)
```

```
taun = Sheet4.Cells(5, 24)
```

```
  If taun = 0 Then taun = 0.0001
```

```
sigmad = Sheet4.Cells(6, 24)
```

```
alpha1 = Exp(-dt / taun)
```

```
beta0 = 1 - alpha1
```

```
sigmaw = sigmad * Sqr((1 + alpha1) / beta0)
```

```
xnoise = sigmaw * Sqr(-2 * Log(Rnd)) * Sin(2 * 3.14159 * Rnd) 'Gaussian distributed  
  noise
```

```
  xmeas = x + xnoise
```

```
End Sub
```

```
Sub selftuningfilter(xmeas, xmeassf)
```

```
deltaf = 0.9 * deltax + 0.1 * (xmeas - xmeasold) ^ 2
```

```
lambda = 1 / (0.5 + 1.1668 * deltax / Ef ^ 2)
```

```
If lambda > 1 Then lambda = 1
```

```
xmeasold = xmeas
```

```
xmeassf = lambda * xmeas + (1 - lambda) * xmeassf
```

```
End Sub
```

APPENDIX

C

VBA codes used in the simulation of steady state signal with the addition of an oscillatory change

'VBA codes written with help from Dr. R.R. Rhinehart
'Department of Chemical Engineering, Oklahoma state university
'Daniel Fakunle
'MSC Research Project
'10-13-09
'A sub program to generate noisy signal and filter the noise using a
'cusum filter,first order filter and Kalman filter

Sub signaloscillatory()
Application.ScreenUpdating = False

'Initial values of constants
ff = Sheet5.Cells(7, 24)
dt = Sheet5.Cells(3, 24)
Trigger = Sheet5.Cells(8, 24)
Ef = Sheet5.Cells(10, 24)
'kalman filter constants
xhat(0) = xmeas
Pminus(0) = 1
Randomize
dt = 0.1
'Main Program
For i = 1 To 1000 'For loop
t = dt * i
x = 3
If i > 500 Then x = 3 + -0.729535369032456 *
 Sin(Application.WorksheetFunction.Radians(50 *
 Application.WorksheetFunction.Pi() * (t - 50) / 20))
Call Process4(x, xmeas)

'Call filter
Call ffilter4(xmeas, xmeasff)
Call cusumf4(xmeas, xmeasc)
Call kalman4(xmeas, xmeaskf)

```

Call selftuningfilter4(xmeas, xmeassf)
'Print out observable results
Application.ScreenUpdating = False
Sheet5.Cells(2 + i, 1).Value = t
Sheet5.Cells(2 + i, 2).Value = x
Sheet5.Cells(2 + i, 3).Value = xmeas
Sheet5.Cells(i + 2, 4).Value = xmeassf 'STF
Sheet5.Cells(i + 2, 5).Value = xmeasc 'Cusum
Sheet5.Cells(i + 2, 16).Value = xmeaskf 'Kalmnan
Sheet5.Cells(i + 2, 22) = xmeasff 'FOFilter

```

```

Next i
Application.ScreenUpdating = True
End Sub

```

'FIRST ORDER FILTER

```

Sub ffilter4(xmeas, xmeasff)
  xmeasff = ff * (xmeas) + (1 - ff) * (xmeasff)
End Sub

```

'CUSUM FILTER

```

Sub cusumf4(xmeas, xmeasc)
  If i = 1 Then
    n = 0
    xx = 0
    xxold = 0
    xxspc = 0
    v = 0
    cusum = 0
    m = 11
    ff1 = ((m - 2) / (m - 1))
    ff2 = (1 - ff1) / 2
  End If

  xx = xmeas
  n = n + 1
  v = ff1 * v + ff2 * (xx - xxold) ^ 2
  xxold = xx
  cusum = cusum + xx - xxspc
  If (Abs(cusum) > Trigger * Sqr(v * n)) Then
    xxspc = xxspc + cusum / n
    cusum = 0
    n = 0
  End If
  xmeasc = xxspc

```

End Sub

Sub kalman4(xmeas, xmeaskf)

'KALMAN FILTER

R = Sheet5.Cells(9, 24).Value

Q = Sheet5.Cells(4, 24).Value

'Time Update

xhatminus(i) = xhat(i - 1)

Pminus(i) = P(i - 1) + Q

'Measurement Update

k(i) = Pminus(i) / (Pminus(i) + R)

xhat(i) = xhatminus(i) + k(i) * (xmeas - xhatminus(i))

P(i) = (1 - k(i)) * Pminus(i)

xmeaskf = xhat(i)

End Sub

Sub Process4(x, xmeas)

taun = Sheet5.Cells(5, 24)

If taun = 0 Then taun = 0.0001

sigmad = Sheet5.Cells(6, 24)

alpha1 = Exp(-dt / taun)

beta0 = 1 - alpha1

sigmaw = sigmad * Sqr((1 + alpha1) / beta0)

xnoise = sigmaw * Sqr(-2 * Log(Rnd)) * Sin(2 * 3.14159 * Rnd) 'Gaussian distributed
noise

sp = x

xmeas = x + xnoise

End Sub

Sub selftuningfilter4(xmeas, xmeassf)

deltaf = 0.9 * deltax + 0.1 * (xmeas - xmeasold) ^ 2

lambda = 1 / (0.5 + 1.1668 * deltax / Ef ^ 2)

If lambda > 1 Then lambda = 1

xmeasold = xmeas

xmeassf = lambda * xmeas + (1 - lambda) * xmeassf

End Sub

VITA

DANIEL FAKUNLE

Candidate for the Degree of

Master of Science

Thesis: COMPARISON OF THE CUSUM, FIRST ORDER, SELF-TUNING AND
KALMAN FILTERS

Major Field: Chemical engineering

Biographical:

Personal Data:

Education:

Completed the requirements for the Master of Science in Chemical Engineering at
Oklahoma State University, Stillwater, Oklahoma in May, 2010.

Completed the requirements for the Master of Science in Petroleum Engineering at
University of Houston, Houston, Texas in December, 2008.

Completed the requirements for the Bachelor of Science/Arts in your major at
Obafemi Awolowo University, Ile-Ife in 2005.

Name: Daniel Fakunle

Date of Degree: May, 2010

Institution: Oklahoma State University

Location: Stillwater, Oklahoma

Title of Study: COMPARISON OF THE CUSUM, FIRST-ORDER, SELF-TUNING
AND KALMAN FILTERS

Pages in Study: 79

Candidate for the Degree of Master of Science

Major Field: Chemical Engineering

Scope and Method of Study: This work involved the comparison of the CUSUM filter, the first-order filter, the self-tuning filter and the Kalman filter. The filters were compared on the basis of the minimization of the integral sum of errors (ISE), the relative cost and the relative ease of understanding.

The ISE was compared for processes involving a step change, a ramp change and an oscillatory process change. These three processes were simulated using Visual Basic for Applications (VBA).

The four filters were also compared based on the relative ease of human understanding based on the complexity of equations.

The relative cost of the filters was based on the number of operations and the required number of storage variables.

Findings and Conclusions: The CUSUM filter is the best filter to use in a steady state process involving no change. In a process that involves a step change, the Kalman filter is the best filter to use. A process that involves a ramp change is best filtered using the Kalman filter. And, an oscillatory process is best filtered using the self-tuning filter.

First-order filter is the easiest to understand, while the Kalman filter is the most complex to understand.

The first-order filter is the cheapest to operate in terms of arithmetic operations, while the Kalman filter is the most expensive in terms of arithmetical operations.

The first-order filter is the least expensive considering required number of variables, while the CUSUM filter requires the most number of variables and is thus the most expensive in terms of the required number of variables.

ADVISER'S APPROVAL: Dr. R. R. Rhinehart
