ISSN 0280-5316
ISRN LUTFD2/TFRT--5765--SE

Error Detection in the Active Front Steering system

Camran Razavi

Department of Automatic Control
Lund University
February 2006

Department of Automatic Control
Lund Institute of Technology
Box 118
SE-221 00 Lund Sweden

Document name
MASTER THESIS

Date of issue
February 2006

Document Number
ISRNLUTFD2/FRT--5765--SE

Author(s)
Camran Razavi

Supervisor
Wolfgang Reinelt and Samuel Malinen at ZFLenksysteme in Schwäbisch Gmûnd, Germany
Brad Schofield and Tore Hägglund at Automatic Control in Lund, Sweden.

Sponsoring organization

Title and subtitle
Error Detection in the Active Front Steering system (Feldetektering i ett aktivt framhjulsstyrningssystem)

Abstract
Parameters to the CUSUM algorithm are calculated with the aid of the statistical properties of the input data and change detection theory. It works on Gaussian sequences but not on the data from the measurements. Also an approach called Local CUSUM, which lets detectors act over different intervals
of the input data is presented and turned out to give good results. These approaches are compared to other detectors such as GMA.
The detectors have been used to detect errors in the actuator dynamics of the Active Front Steering system and to detect erroneous wheel velocities in a car.

Keywords
Change detection, error detection, CUSUM, LCUSUM, GMA, Active Front Steering, actuator dynamics, wheel velocity.

Classification system and/or index terms (if any)

Supplementary bibliographical information

ISSN and key title
0280-5316

ISBN

Language
English

Number of pages
70

Recipient's notes

Security classification

The report may be ordered from the Department of Automatic Control or borrowed through:University Library, Box 3, SE-221 00 Lund, Sweden Fax +46 46 222 42 43

Notation

Acronyms

AC	Adyn Compare
ACF	Adyn Compare Fast
ACS	Adyn Compare Slow
ADYN	Actuator DYNamics monitoring
AFS	Active Front Steering
ARL	Average Run Length function
CDA	Change Detection Algorithm
CUSUM	CUmulative SUM
ELU	Electromagnetic Locking Unit
FAR	False Alarm Rate
GMA	Geometric Moving Average
MTD	Mean Time to Detection
MTFA	Mean Time between False Alarms
LCUSUM	Local CUSUM
LGCUSUM	Local and Global CUSUM
VSR	Variable Steering Ratio
ZF	Zahnrad Fabrik (Cogwheel Factory)

Basic Notation

AFS system

δ_S	steering wheel angle
δ_M	motor angle
δ_G	pinion angle
i_S	steering wheel angle to pinion angle ratio
i_M	motor angle to pinion angle ratio
δ_F	average front wheel angle
F_{SG}	pinion angle to rack displacement static nonlinearity
δ_{Md}	desired motor angle
ϵ_{pos}	position residual in ADYN
ϵ_{vel}	velocity residual in ADYN
ϵ_{acc}	acceleration residual in ADYN

Change Detection

y_i	independent random variables, assumed to be normal distributed (input)
θ	"changing quantity"
p_θ	probability density
s	log-likelihood ratio
$\mathbf{E}_{\theta_i}$	expectation of the random variables under the distribution p_{θ_i}
σ^2	standard deviance and $\mathbf{E}(y_k^2)$
m	mean value
m_0	mean value before change
m_1	mean value after change
ϵ_i	residuals
Θ_i	expected value of the residuals
v_i	residual deviation modeled as white noise
S_k	integrated log-likelihood ratio
g_t	drift in change detection algorithms
h	threshold (GMA and CUSUM)
ν	drift parameter (CUSUM)
μ	$\theta - \nu$
σ_0^2	variance before change
σ_1^2	variance after change
n_k	$\frac{y_k}{\sigma}$
σ^*	$\frac{\ln \sigma_0^2 - \ln \sigma_1^2}{\sigma_1^{-2} - \sigma_0^{-2}}$
z	value of the cumulative sum
L_z	general ARL function
L_0	ARL function when the drift starts in 0
$\mathbf{E}(T_{0,h}\|z)$	the average number of samples from the current sample that it takes to reach the lower threshold 0 for the cumulative sum (the drift) when starting in z
$\mathbf{P}(0\|z)$	the probability that the cumulative sum (drift) reaches the threshold h when starting in z
$N(z)$	$\mathbf{E}(T_{0,h}\|z)$
$P(z)$	$\mathbf{P}(0\|z)$
f_θ	density function for the squared input sequence
γ_i	GMA weights
α	GMA forgetting factor

Plausibility

v_1 or v_{FL}	velocity of the front left wheel
v_2 or v_{FR}	velocity of the front right wheel
v_3 or v_{RL}	velocity of the rear left wheel
v_4 or v_{RR}	velocity of the rear right wheel
ϵ_{ij}	different residuals
δ_F	average front wheel angle
δ_l	left front wheel angle (Ackermann)
δ_r	right front wheel angle (Ackermann)
l	distance between rear and front axis
b	distance between left and right wheels

Contents

Chapter 1

Introduction

This thesis consists of two different parts. They both concern error/change detection. The first part is about error detection in the actuator dynamics of the Active Front Steering system (AFS) and the second part is about plausibility of the wheel velocities. Here an introduction to them is given.

1.1 Error Detection in the Actuator Dynamics of the AFS System

1.1.1 The Active Front Steering System

Functionality

The Active Front Steering system is developed and patented by ZF Lenksysteme. Without loosing the mechanical connection between the steering wheel and the road wheels i.e. it is not a steer by wire system, it provides an electronically controlled superposition of an angle to the steering wheel angle. Figure 1.1 shows the essentials of the system and Figure 1.2 shows the effect of the system.

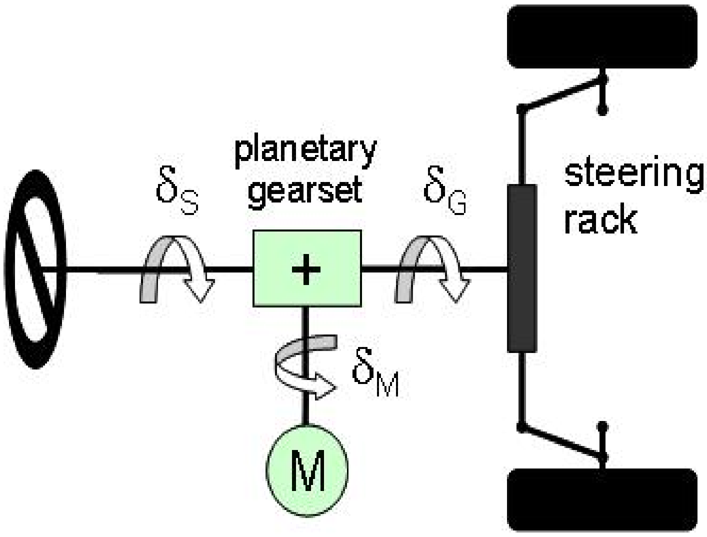

Figure 1.1: AFS principle: the planetary gearbox enables the superposition of δ_M to the steering wheel angle δ_S without loss of mechanical connection between the steering wheel and the road wheels.

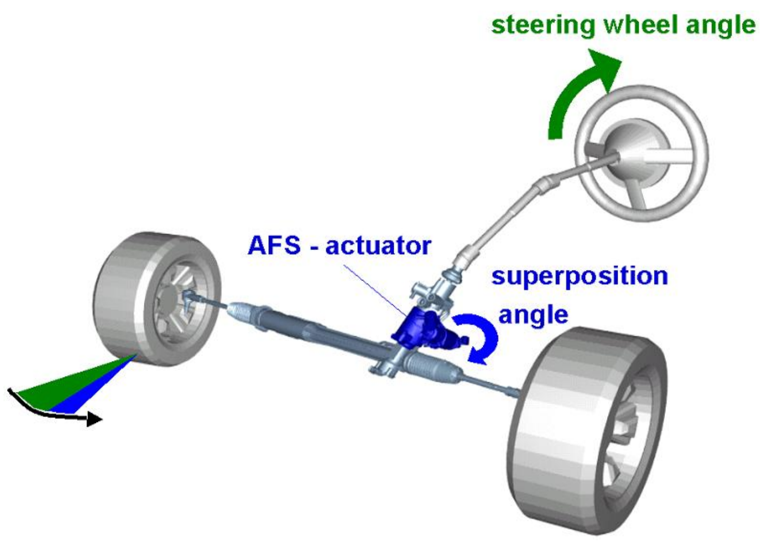

Figure 1.2: AFS effect: An "extra" angle is added/subtracted to the wheels with the AFS system.

The angles are related as:

$$\frac{1}{i_S}\delta_S + \frac{1}{i_M}\delta_M = \delta_G \quad (1.1)$$

where δ_S is the steering wheel angle, δ_M is the angle the AFS system superimposes to the system, i_M and i_S are their respective ratios and δ_G is the pinion angle.

The relation between the road wheels, δ_F and the pinion angle is:

$$\delta_F = F_{SG}(\delta_G) \quad (1.2)$$

F_{SG} is a static nonlinearity that accounts for the relation between pinion angle and the rack displacement. Note that δ_F is an average angle of the two front wheels, the wheels have different angles according to the Ackermann steering geometry [1, page 336].

In Figure 1.3 it is seen how δ_M is generated; the AFS motor and the planetary gearbox are shown.

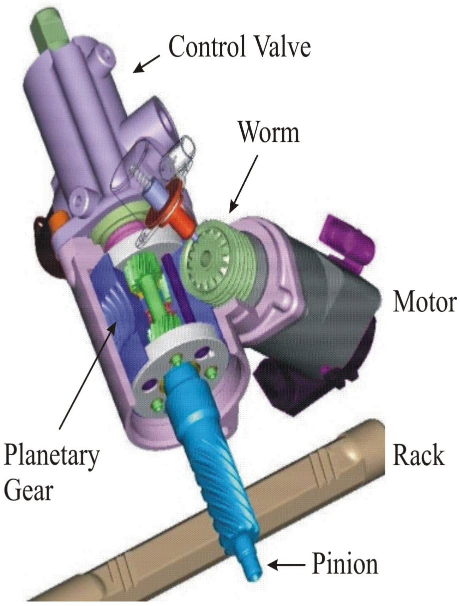

Figure 1.3: The AFS motor mechanism.

By controlling the AFS motor the steering behavior of the car can be changed. The *Variable Steering Ratio* (VSR) adds an angle depending on the velocity of the car. At low velocities a large angle is superimposed, increasing the comfort when e.g. parking or turning. At high velocities a negative angle is added, this makes the car less sensitive to steering wheel movement and therefore safer. The AFS system can also be used for vehicle stabilization (active safety) and for giving a faster response to the driver's input [2].

Of course it is very important that the AFS system and the AFS motor gives the correct angle δ_M. To ensure this many safety features monitor the system. ADYN is one of these safety features.

Actuator Dynamics Monitoring - ADYN

ADYN monitors the actuator dynamics in the AFS system. ADYN's inputs are the desired motor angle δ_{Md} and the real angle captured from a sensor δ_M. Two filters form the velocity and the acceleration of these angles.

$$\begin{array}{ll} \text{motor angle} & \delta_M \\ \text{motor angular velocity} & \dot{\delta}_M \\ \text{motor angular acceleration} & \ddot{\delta}_M \\ \text{desired motor angle} & \delta_{Md} \\ \text{desired motor angular velocity} & \dot{\delta}_{Md} \\ \text{desired motor angular acceleration} & \ddot{\delta}_{Md} \end{array} \tag{1.3}$$

The following residuals are formed:

$$\epsilon_{pos} = \delta_M - \delta_{Md} \tag{1.4}$$

$$\epsilon_{vel} = \dot{\delta}_M - \dot{\delta}_{Md} \tag{1.5}$$

$$\epsilon_{acc} = \ddot{\delta}_M - \ddot{\delta}_{Md} \tag{1.6}$$

Different change detection algorithms (CDA) monitor that these residuals do not become too large. When a change detection algorithm gives an alarm, the AFS motor is stopped with an electromagnetic locking unit (ELU), see Figure 1.4. The change detection algorithms should be tuned not to give false alarms when there is no error and to give alarm as fast as possible when there is an error.

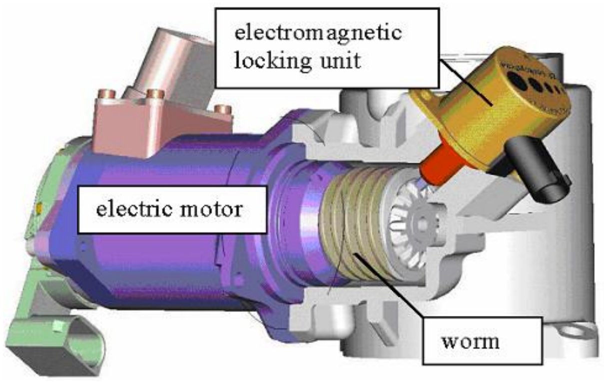

Figure 1.4: The electromagnetic locking unit locks the motor if an error is detected.

1.1.2 Objectives

The objectives of this part of the thesis are:

- With statistical properties of the residuals and change detection theory find parameters to the CUSUM change detection algorithm.

- Find parameters for existing CUSUM and GMA algorithms with trial and error.
- Optimize the error detection in ADYN.
- Compare results for the different approaches with each other and existing detectors.

1.1.3 Measurements

Two different sets of measurements were used to tune and test the different change detection algorithms:

Set A

The measurements used come from ten driving scenarios. These ten driving scenarios generate the biggest residuals possible.

Set B

For one of the approaches it was not enough to use the largest residuals possible. This approach operates on different magnitude intervals of the residuals, so residuals covering all possible magnitudes were needed. Set B covers 50 driving scenarios, 10 of them are set A.

1.1.4 Outline

In Chapters 2 and 3 basic theory that is used is described. With this theory different approaches for change detection are tested and the results can be seen in Chapters 4. Conclusions are made in Chapter 5.

1.2 Plausibility of the Wheel Velocities

1.2.1 Calculation of the Car Velocity

The car velocity is used as input to many functions in the AFS system. It is calculated with the aid of the velocities of the four wheels of the car which sensors provide. As it is implemented now, the function that calculates the car velocity takes into account if any of the sensor's selfdiagnostics says it is not working correctly (voltage-check etc) or if the CAN-bus communication is not working properly. Also the ESP system tells if one wheel velocity is false, in particular so called *sensor errors* and *outliers* (see Section 1.2.2) which are not detected by the AFS system.

1.2.2 Erroneous Wheel Velocities

Outliers

In some driving scenarios the velocity of the wheels differ a lot. It can be one wheel that for some reason is skidding, or two wheels skidding. This is possible if the driver of the car e.g. is driving actively or if two wheels are on a more slippery surface in the so called “μ-split scenario”.

1.2.3 Objectives

The objectives of this part of the thesis are:

- With change detectors detect outliers and sensor errors fast enough for AFS purposes.
- Examine if two erroneous wheel velocities can be detected and how this should be handled.

1.2.4 Measurements

Data from 58 driving scenarios was used. All data found was used, to cover as many situations as possible.

1.2.5 Outline

In Section 6.1 different approaches are presented, the results can be seen in Section 6.2 and conclusions are made in Section 6.3.

Chapter 2

Change Detection Algorithms

Here the CUSUM and GMA change detection algorithms are described. The description covers what needs to be known to understand this thesis. If the reader wants to learn more, [3] and [4] are recommended. These books has been used as references when writing this.

2.1 Change Detection

2.1.1 Change Detection Fundamentals

The variables where changes should be detected $\{y_i\}$, are a sequence of independent random variables with a probability density $p_\theta(y)$. The probability density depends on one parameter θ, typically mean or variance, which changes from θ_0 to θ_1. This change is what should be detected.

The CUSUM and GMA algorithms use a special property in the log-likelihood ratio defined as

$$s(y) = \ln \frac{p_{\theta_1}(y)}{p_{\theta_0}(y)} \tag{2.1}$$

The property is

$$\mathbf{E}_{\theta_0}(s) < 0 \qquad \text{and} \qquad \mathbf{E}_{\theta_1}(s) > 0 \tag{2.2}$$

where $\mathbf{E}_{\theta_i}$ is the expectation of the random variables under the distribution p_{θ_i}.

This means that when the parameter θ changes from θ_0 to θ_1, the sign of the mean value of the log-likelihood ratio changes. This sign change is what is used in the detection. See [3, page 25] for more details.

Assume Gaussian distribution $y \in N(m, \sigma)$, the probability density function is:

$$p_\theta(y) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(y-m)^2/2\sigma^2} \qquad (-\infty < y < \infty) \tag{2.3}$$

2.1.2 Change in Mean Value

Consider a change in the mean value from m_0 to m_1 i.e θ is m . The variance σ^2 is assumed to be constant. (2.1) will be:

$$s_i = \ln \left(\frac{\frac{1}{\sigma\sqrt{2\pi}} e^{-(y_i - m_1)^2/2\sigma^2}}{\frac{1}{\sigma\sqrt{2\pi}} e^{-(y_i - m_0)^2/2\sigma^2}} \right) = \frac{m_1 - m_0}{\sigma^2} \left(y_i - \frac{m_0 + m_1}{2} \right) \tag{2.4}$$

Assume σ and m_1 to be known and $m_0 = 0$ then

$$s_i = y_i \tag{2.5}$$

is appropriate to be used to detect the change in the mean. See [3, page 27] for more details.

2.1.3 Change in Variance

In this case θ is σ^2 and changes from σ_0^2 to σ_1^2. The mean m is assumed to be constant and (2.1) becomes:

$$s_i = \ln \left(\frac{\frac{1}{\sigma_1\sqrt{2\pi}} e^{-(y_i - m)^2/2\sigma_1^2}}{\frac{1}{\sigma_0\sqrt{2\pi}} e^{-(y_i - m)^2/2\sigma_0^2}} \right) = \ln \frac{\sigma_0}{\sigma_1} + \left(\frac{1}{\sigma_0^2} - \frac{1}{\sigma_1^2} \right) \frac{(y_i - m)^2}{2} \tag{2.6}$$

If σ_0 and σ_1 are assumed to be known and $m = 0$ then

$$s_i = y_i^2 \tag{2.7}$$

should be used to detect the change. See [3, page 31] for more details.

2.1.4 Residual Generation

In this thesis the variables $\{y_i\}$ are residuals. Figure 2.1 shows how they are generated in general. In the ideal case the residuals are zero before a change

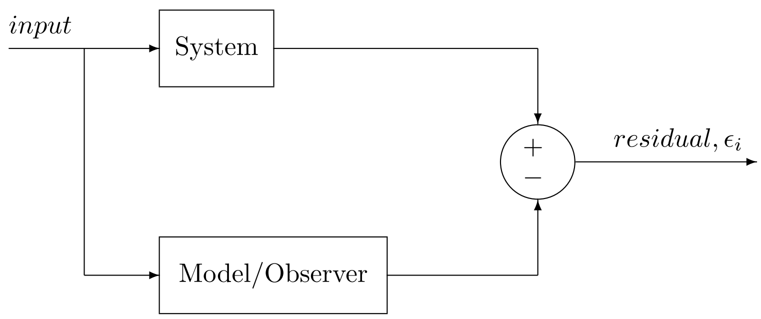

Figure 2.1: Generation of residuals.

and nonzero after a change. Of course this is not the case in general due to e.g.

not perfect model, sensor disturbances and measurement noise. The residuals are modeled as:

$$\epsilon_i = \Theta_i + v_i \tag{2.8}$$

where $\Theta_i = 0$ and v_i is white noise with Gaussian distribution, $N \in (0, \sigma^2)$.

From Equation 2.8 it can be seen that when the model is correct and no change has occurred, the residuals are white noise. When a change occurs in either the mean or the variance, the residuals change and as written above it is ϵ_i and ϵ_i^2 that should be used to detect these changes. Because of the white noise and the disturbances, some kind of function that averages intuitively seems like a good change detector [4, page 18]. The CUSUM and GMA algorithms are two well known detectors which do exactly this.

The residuals that are used in this thesis differ from the residuals in Figure 2.1. The residuals are written in Equations 1.4, 1.5 and 1.6. They are the difference between desired and actual values in a regulator loop. The values of the residuals highly depend on how good the controller of the AFS motor manages to follow the desired values. To the above written behavior of the residuals the performance of the AFS motor controller has to be added as a disturbance.

2.1.5 Performance Measures

Change detection algorithms should preferably be fast and only alarm when a change has occurred. Different change detectors do this differently well. To be able to compare change detectors the following performance measures exist [4, page 28-29]:

- *Mean Time between False Alarms (MTFA).* Tells how long time there is between two alarms (in average) when no change has occurred. This should of course be as large as possible. $\frac{1}{MTFA}$ is called the *False Alarm Rate (FAR).*

- *Mean Time to Detection (MTD).* Tells how long time it takes to detect a change. This should be as small as possible.

- *Average Run Length function, ARL(θ).* Tells how long time it takes until an alarm is given after a change of size θ has occurred. ARL(θ) generalizes MTFA and MTD:

$$\begin{aligned} \text{ARL}(0) &= \text{MTFA} = \tfrac{1}{FAR} \\ \text{ARL}(\theta) &= \text{MTD}(\theta) \end{aligned} \tag{2.9}$$

2.2 The CUSUM Algorithm

2.2.1 Intuitive Derivation

The cumulative sum (CUSUM) algorithm uses the sign change in Equation 2.1 when a change occurs. The log-likelihood ratios are sum up and form:

$$S_k = \sum_{i=0}^{k} s_i \tag{2.10}$$

Because of the sign change, $\{S_k\}$ will show a negative drift before a change and a positive drift after. Figure 2.2 shows an example when a change at time 15 occurs.

To detect the change, the difference between the minimum value of $\{S_k\}$ and the current value is in focus. When the difference is larger than a threshold h a change is said to have occurred.

Instead of letting $\{S_k\}$ drift and get a negative value, it is convenient to let it be zero when it is drifting negatively. Then when a change occurs the drift will rise from zero and be positive. We get the following equation which is based on Equation 2.10:

$$\begin{aligned} & g_t = \max\ (g_{t-1} + s_t, 0) \\ & g_0 = 0 \end{aligned} \tag{2.11}$$

There is one problem still; in Figure 2.2 it can be seen that the drift *in average* has a negative drift before the change. This causes that when S_k is positive for many time instants following each other before a change, equation (2.11) will get a positive drift. To avoid this a drift parameter ν is introduced:

$$\begin{aligned} & g_t = \max\ (g_{t-1} + s_t - \nu, 0) \\ & g_0 = 0 \end{aligned} \tag{2.12}$$

And with the threshold, the CUSUM algorithm looks as follows [3, chapter 2.2.1]:

$$\begin{aligned} & g_t = \max(g_{t-1} + s_t - \nu, 0) \\ & g_0 = 0 \\ & \text{alarm if } g_t > h \end{aligned} \tag{2.13}$$

In Figure 2.3 the typical behavior of the CUSUM algorithm can be seen.

2.2.2 Choosing the Parameters

Trial and Error

There are two parameters that has to be chosen: ν and h. How they are chosen depends on how the input to the CUSUM algorithm looks like i.e. the log-likelihood ratio which in Equation 2.13 is $\{s_t\}$. If it has a big variance, ν has to be bigger than if the variance is small. One rule of thumb is that ν should be chosen so that the drift before a change is zero more than 50 percent of the time [4, page 70].

Now assume a mean change should be detected. From Equation 2.5 it is seen that in this case s_t is ϵ_k. A very important aspect worth considering when choosing ν is that *a change smaller than ν can not*[1] *be detected.* So when choosing ν, also how small changes that can be detected is chosen.

The threshold h depends on both the input to the CUSUM algorithm and on how ν has been chosen. If ν is chosen big, the random positive drift will be small making it possible to choose h small to detect changes faster. If ν is chosen small, the random positive drift will be larger which forces the threshold to be bigger. h affects the FAR and the MTD and should be chosen so that both of them are as small as possible, this is always a compromise. In Figure 2.4 it can be seen how the CUSUM detector acts with different parameter settings.

[1]In some cases when the random positive drift is quite large and the threshold is small it can be detected.

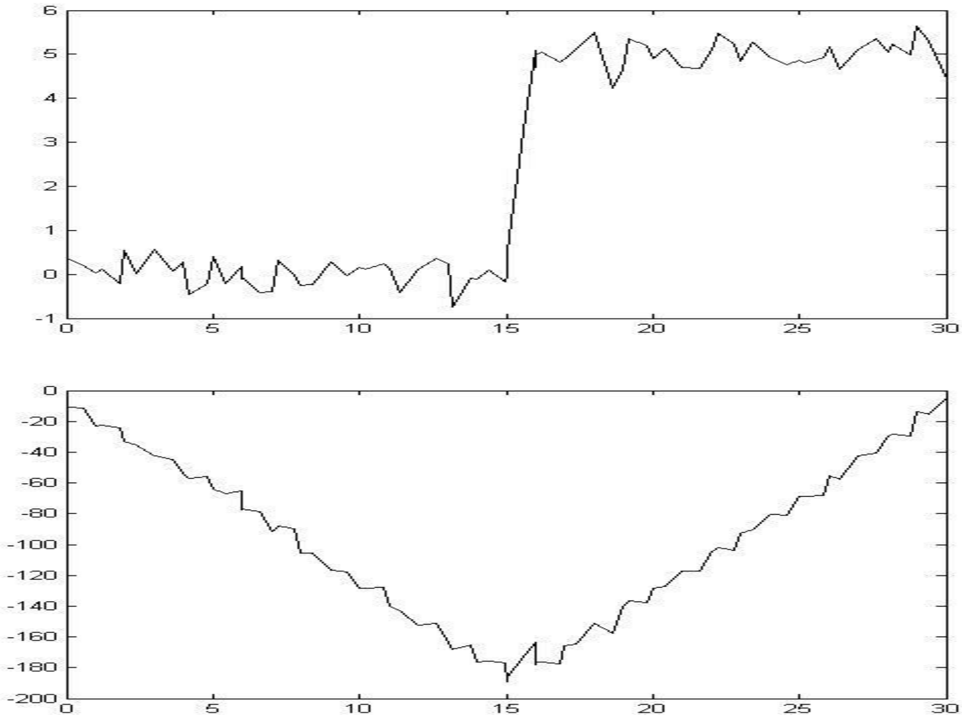

Figure 2.2: A Gaussian sequence with mean change at time 15 (top). The negative and positive drift can clearly be seen (bottom).

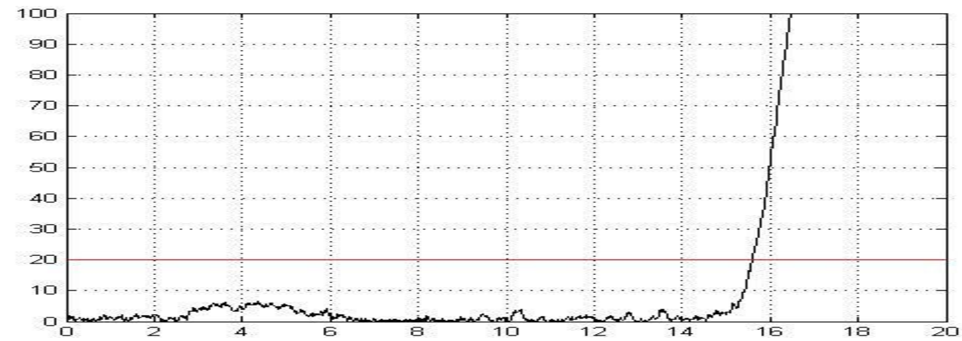

Figure 2.3: Typical behavior of the CUSUM algorithm. The threshold h is 20.

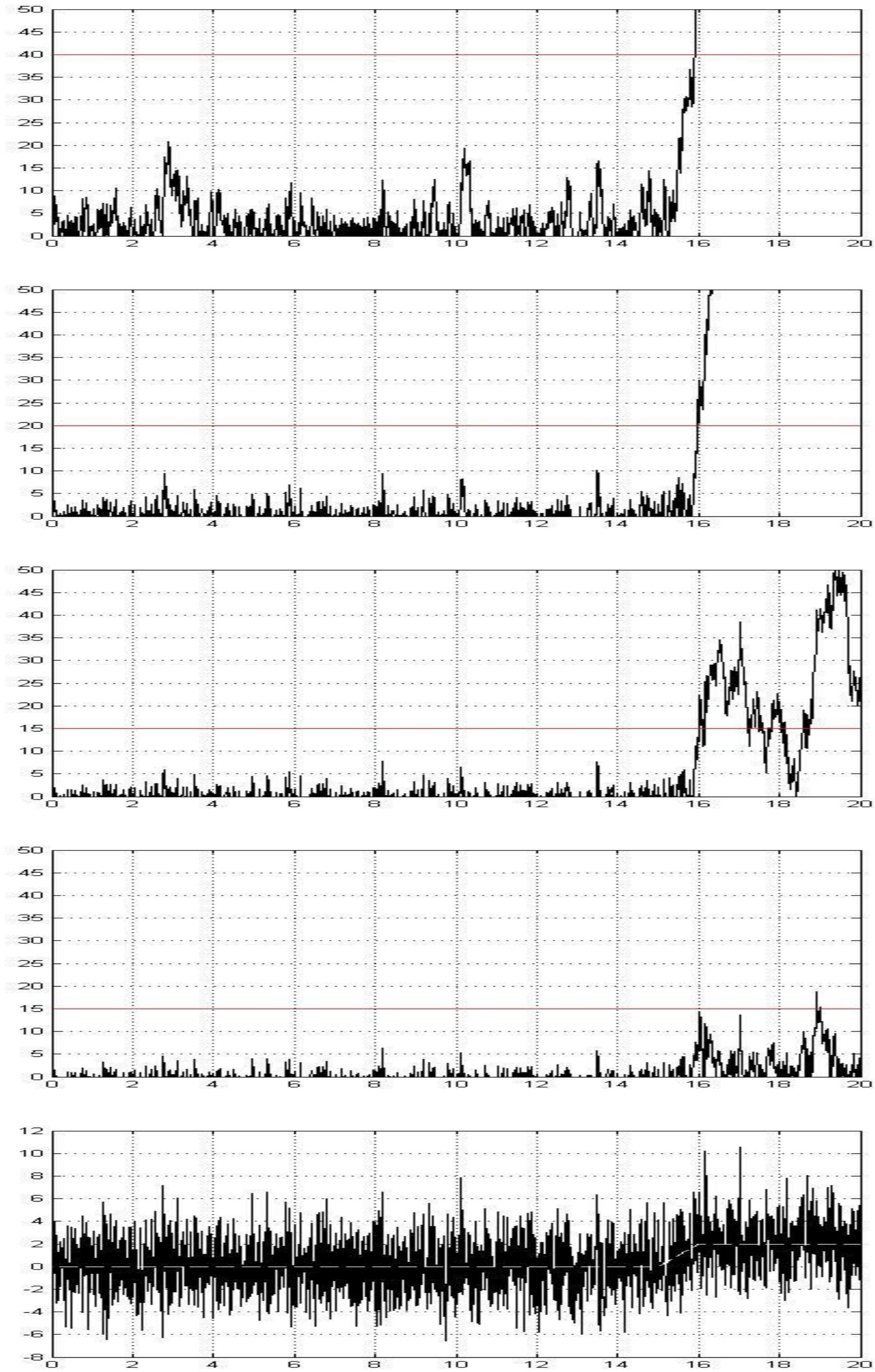

Figure 2.4: Parameters for the CUSUM algorithm. Different ν are chosen and affects how h can be chosen. From the top ν is 0.5, 1.5, 2.0 and 2.5. Note how the magnitude of the drift changes as ν is changed, as ν gets closer to the mean change which is of the magnitude 2 the change gets more difficult to detect. Also note that this input sequence (bottom) is perfectly Gaussian and therefore very easy to handle.

CUSUM Parameters with Statistics, Mean Change

The parameters can be chosen in a more systematic way. This is done with the aid of the ARL function for the CUSUM algorithm. The "exact" ARL function is complex [3, page 168], and given by an equation which consists of two Fredholm integral equations. The equation can be seen in (2.20). But in the case of Gaussian distribution for the residuals and mean change two approximations exist: Wald's approximation and Siegmund's approximation [3, page 171-175] [4, page 442]. Wald's approximation:

$$\mathrm{ARL} = \frac{e^{-2\frac{h\mu}{\sigma^2}} - 1 + 2\frac{h\mu}{\sigma^2}}{2\frac{\mu^2}{\sigma^2}} \tag{2.14}$$

Siegmund's approximation:

$$\mathrm{ARL} = \begin{cases} \frac{e^{-2(h/\sigma+1.166)\mu/\sigma} - 1 + 2(h/\sigma+1.166)\mu/\sigma}{2\mu^2/\sigma^2} & \text{if } \mu \neq 0 \\ (h/\sigma + 1.166)^2 & \text{if } \mu = 0 \end{cases} \tag{2.15}$$

In both equations $\mu = \theta - \nu$. The Siegmund approximation is more accurate than Wald's [4, page 443] and therefore the Siegmund approximation is used.

The idea is to first conform the residuals to a normal distribution, this will give σ in (2.15). Then $h(\nu)$, the threshold as a function of drift can be formed from the following equation:

$$\mathrm{ARL}(0, h, \nu) = \frac{1}{\mathrm{FAR}} \tag{2.16}$$

where an appropriate FAR is chosen. Assuming the change θ to be known, this gives:

$$\mathrm{ARL}(\theta, h(\nu), \nu) = \mathrm{MTD}(\theta) \tag{2.17}$$

This is an equation that only depends on ν. The value of ν that minimizes (2.17) is found, and with this ν, h is found from (2.16). Both the parameters in the CUSUM algorithm have now been determined in a systematic way.

CUSUM Parameters with Statistics, Variance Change

When detecting variance changes, the CUSUM algorithm can be rewritten. Assume a variance change from σ_0^2 to σ_1^2 in the Gaussian sequence $\{y_k\}$ with mean $\mathbf{E}(y_k)$=0. The CUSUM algorithm can be written as:

$$\begin{aligned} & g_k = \max(g_{k-1} + n_{k-1}^2 - \theta, 0) \\ & g_0 = 0 \\ & \text{alarm if } g_k > h \end{aligned} \tag{2.18}$$

where

$$\begin{aligned} & n_k = \tfrac{y_k}{\sigma} \\ & \theta = (\tfrac{\sigma^*}{\sigma})^2 \\ & \sigma^* = \tfrac{\ln \sigma_0^2 - \ln \sigma_1^2}{\sigma_1^{-2} - \sigma_0^{-2}} \\ & \sigma^2 = \mathbf{E}(y_k^2) \end{aligned} \tag{2.19}$$

In this CUSUM algorithm [3, page 170] there is only one parameter to be determined: h. The other parameters are determined from $\{y_k\}$ and are assumed to be known. h is determined with the aid of the ARL function. However in this case an approximation of the ARL function does not exist so the work was done with the "exact" function [3, page 168]:

$$L_z = \mathbf{E}(T_{0,h}|z) + \mathbf{P}(0|z)L_0 = N(z) + \mathbf{P}(z)L_0 \tag{2.20}$$

$\mathbf{E}(T_{0,h}|z)$ is the average number of samples from the current sample that it takes to reach the lower threshold 0 for the cumulative sum i.e. the drift, when starting in z. $\mathbf{P}(0|z)$ is the probability that the cumulative sum reaches the threshold h when starting in z. So the ARL function is weighted with measures for reaching either h or zero when starting in z in the cumulative sum. Now assume $z = 0$, (2.20) turns into:

$$L_0 = \frac{N(0)}{1 - P(0)} \tag{2.21}$$

$N(0)$ and $P(0)$ are found from the following equations:

$$\mathbf{P}(z) = \int_{-\infty}^{-z} f_\theta(x)dx + \int_0^h \mathbf{P}(x) f_\theta(x - z)dx \tag{2.22}$$

$$N(z) = 1 + \int_0^h N(x) f_\theta(x - z)dx \tag{2.23}$$

f_θ is the density function for the squared sequence $\{y_k^2\}$ which when $\{y_k\}$ is Gaussian is a $\chi^2(1)$ distribution.

$$f_\theta(x) = \begin{cases} \frac{e^{-\frac{x+\theta}{2}}}{\sqrt{2}\Gamma(\frac{1}{2})\sqrt{x+\theta}} & \text{if } x + \theta > 0 \\ 0 & \text{if } x + \theta \leq 0 \end{cases} \tag{2.24}$$

Equations 2.22 and 2.23 are Fredholm integral equations of the second kind which must be solved numerically. Several different methods exist, e.g. the Nystrom method [6] or with the integral equation Neumann series [7]. The method in [3, page 170-171] with some simplifications was used. Also the Nystrom method was implemented but did not work since the kernel was separable [6, page 4].

In Equations 2.21, 2.22 and 2.23 it is seen that the ARL is a function of h and θ. Finding h for different values of θ and ARL is what is of interest and an algorithm that does this has been implemented.

2.2.3 Double Sided CUSUM

In Equation 2.13 only positive changes are detected. To detect negative changes as well, the absolute value of the residuals can be used in case of mean change. In the case of variance change this does not matter since the squared residuals always are positive, making it impossible to detect a decrease in variance with this algorithm:

$$g_t = \text{abs}(\epsilon_t) \tag{2.25}$$

There is a better way though; form two CUSUM detectors, one as (2.13) which detects positive changes, and one where the drift parameter is positive and the threshold is negative detecting negative changes:

$$\begin{aligned} &g_t = \min(g_{t-1} + s_t + \nu, 0) \\ &g_0 = 0 \\ &\text{alarm if } g_t < h \end{aligned} \tag{2.26}$$

This will give better results since the random drift in both the detectors individually will be less than if (2.13) was used with (2.25). This makes it possible to choose smaller drift parameters and smaller thresholds making the detectors more sensitive and faster. When using two CUSUM detectors as described, the algorithm is called "double-sided CUSUM" [4, page 67].

2.3 GMA

2.3.1 Derivation

GMA stands for Geometric Moving Average. The idea behind this change detector is to form a weighted sum:

$$g_t = \sum_{i=0}^{\infty} \gamma_i s_{t-i} \tag{2.27}$$

where $\{s_i\}$ is the log-likelihood ratio in Equation 2.1 and the weights are:

$$\gamma_i = \alpha(1-\alpha)^i, \qquad 0 \le \alpha < 1 \tag{2.28}$$

The weights are exponential and act as a forgetting factor. In Figure 2.5 two different set of weights can be seen.

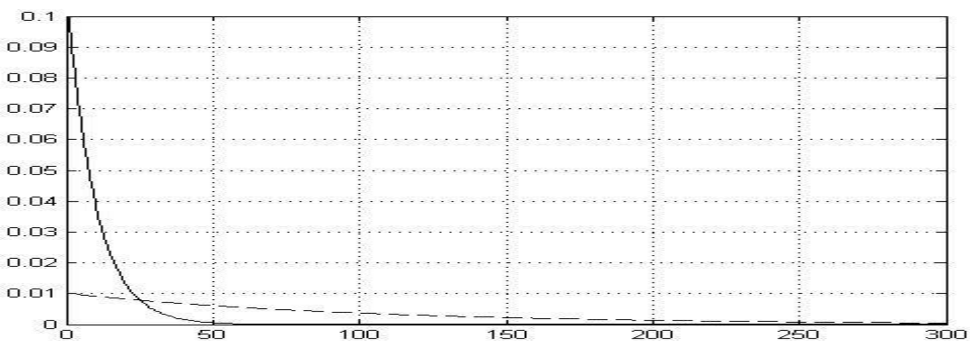

Figure 2.5: Two examples of the exponential weights, $\alpha = 0.90$ in the solid curve and $\alpha = 0.99$ in the dashed.

When the weighted sum exceeds a threshold h a change is detected. The GMA detector looks as follows, where Equation 2.27 has been rewritten to be recursive:

$$\begin{aligned} &g_k = (1-\alpha)g_{k-1} + \alpha s_k \\ &g_0 = 0 \\ &\text{alarm if } g_k > h \end{aligned} \tag{2.29}$$

Recent data are more important and therefore get weighted higher. Depending on how α is chosen the "memory" of the GMA will have different sizes. This "memory" is also called the *sliding window*. See [3, page 28-31] for more details.

2.3.2 Choosing the Parameters

The parameters to be chosen in the GMA detector are the forgetting factor α and the threshold h. α is in the range $0 \leq \alpha < 1$. In Figure 2.5 it shows that if α is chosen small the exponential function will become small fast and the sliding window will be smaller, i.e. the old inputs are relatively fast forgotten. When α is chosen large (close to one), then it is the opposite: The exponential function becomes small slower and the sliding window is larger, i.e. the inputs will be remembered longer. This gives the following behavior:

The smaller α is, the faster the GMA will be. But the detector will also be more affected by disturbances causing smaller margins and possibly larger FAR. The detector will also not be so sensitive and not able to detect small changes.

If α is chosen bigger, the GMA will be slower, more resistent to disturbances and able to detect smaller changes. So dependent on how α is chosen the GMA detector will either be slow with low FAR or fast with larger FAR. The importance of these properties chooses how α is chosen, a compromise between FAR and fastness has to be made.

The threshold depends on how α is chosen. Generally a larger α gives a smaller threshold and vice versa. See Figure 2.6 for an example of what is written above.

2.3.3 Double sided GMA

The same idea as in the case with the CUSUM detector; a negative GMA is implemented for the negative changes [4, page 67]. The only parameter that changes is the threshold h that will be negative instead if positive. The negative GMA looks as follows:

$$
\begin{aligned}
& g_k = (1-\alpha)g_{k-1} + \alpha s_k \\
& g_0 = 0 \\
& \text{alarm if } g_k < h
\end{aligned}
\tag{2.30}
$$

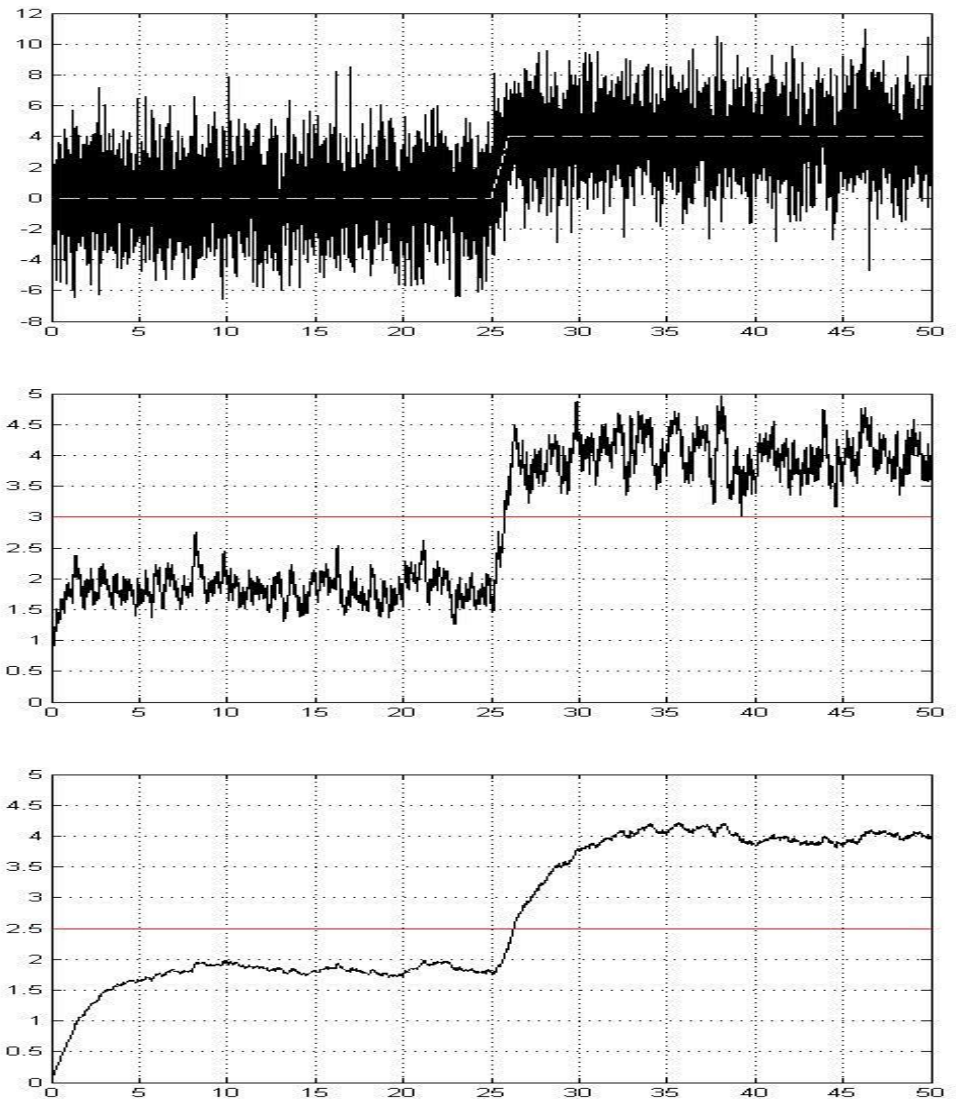

Figure 2.6: The input sequence is Gaussian with variance 5, at time 25 a mean change from 0 to 4 occurs. In the graph in the middle $\alpha = 0.95$ and in the lower graph $\alpha = 0.995$, these graphs show the drift for the GMA (2.29). Thresholds has also been chosen. Note that the threshold in the lower case is smaller than in the middle but the change is despite this detected slower in the lower case.

Chapter 3

Local CUSUM

The residuals in 1.4, 1.5 and 1.6 turned out to most often be relatively small but in some driving scenarios very large. In Figure 3.1 the difference in magnitude for ϵ_{pos} is seen, the difference for ϵ_{vel} and ϵ_{acc} is larger. Since these "few" large residuals worsen the performance of the change detectors significantly an approach that circumvents this has been implemented and is presented here.

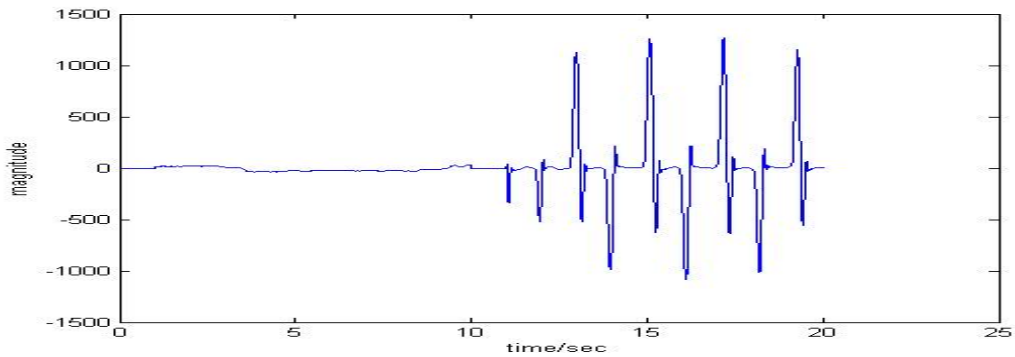

Figure 3.1: Difference between residuals (ϵ_{pos}) in two scenarios. The small residuals (left) are from a normal driving scenario and the large (right) from a actively driven driving scenario (rapid steering wheel movements).

The distribution of the position residuals (set A) can be seen in Figure 3.2, note that most of the residuals are very small despite that this is data with many actively driven scenarios (rapid steering wheel movements) which generates large residuals. A filter that simply discarded the residuals over a certain value was implemented and the CUSUM algorithm was used on these residuals.

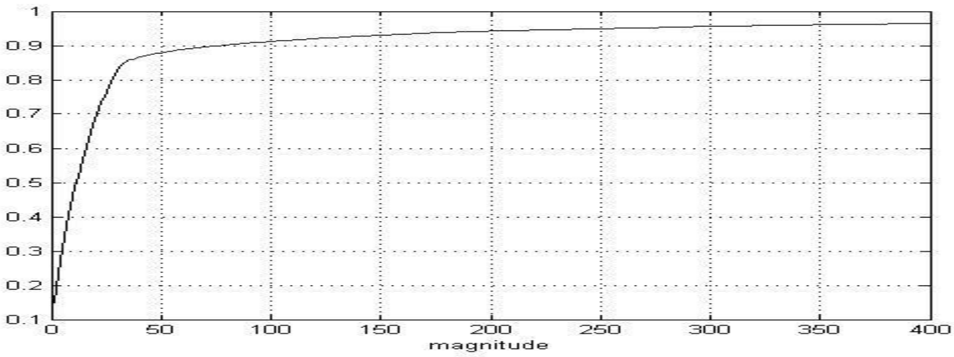

Figure 3.2: The graph shows how many percent of the pos residuals that are in the region $-x < pos < x$.

The filters looked like this:

```
if(residual>value)
out=0
else
out=residual
end
```

One problem with this filter is that when the residuals are larger than *value*, no change can be detected. Because of this many filters with different sizes on the *value* parameter were implemented and attached to different CUSUM detectors working in parallel. This way the range where the CUSUM algorithms can detect changes increases to what is needed.

Chapter 4

Results

4.1 Results - Statistical Approach

The results obtained with the statistical approach are from set A. The figures show residuals and results from ϵ_{pos} (1.4).

4.1.1 CUSUM Parameters with Statistics - Mean Change

Finding ν

In Figures 4.1, 4.2 and 4.3 Equation 2.17 has been plotted for different θ, σ and FAR. From the figures it can be seen that the suggested ν is always $\theta/2$ regardless of how big the variance is, the value of θ and the FAR.

Now compare figures with same θ and FAR and different σ e.g. top diagram in Figures 4.1 and 4.2. On the y-axis the MTD is shown and it is seen that when the variance increases the MTD increases. This agrees with what one intuitively would think; when the variance increases, the detection of a change gets more difficult, which makes the mean time to detection larger.

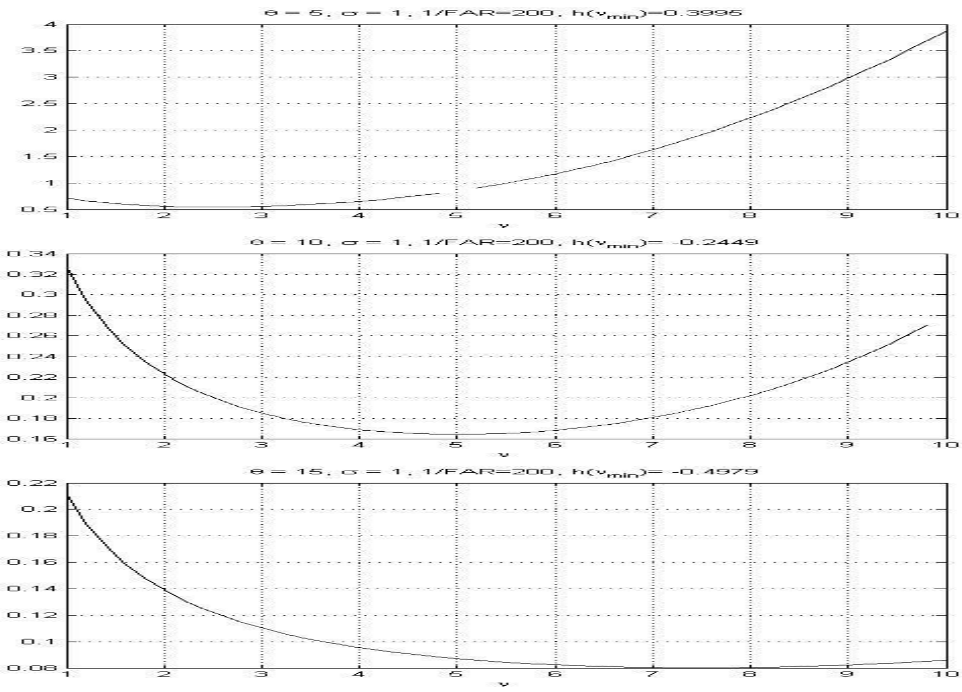

Figure 4.1: MTD(ν). $\sigma = 1,\ \theta = 5,\ 10$ and $15,\ \frac{1}{FAR} = 200$

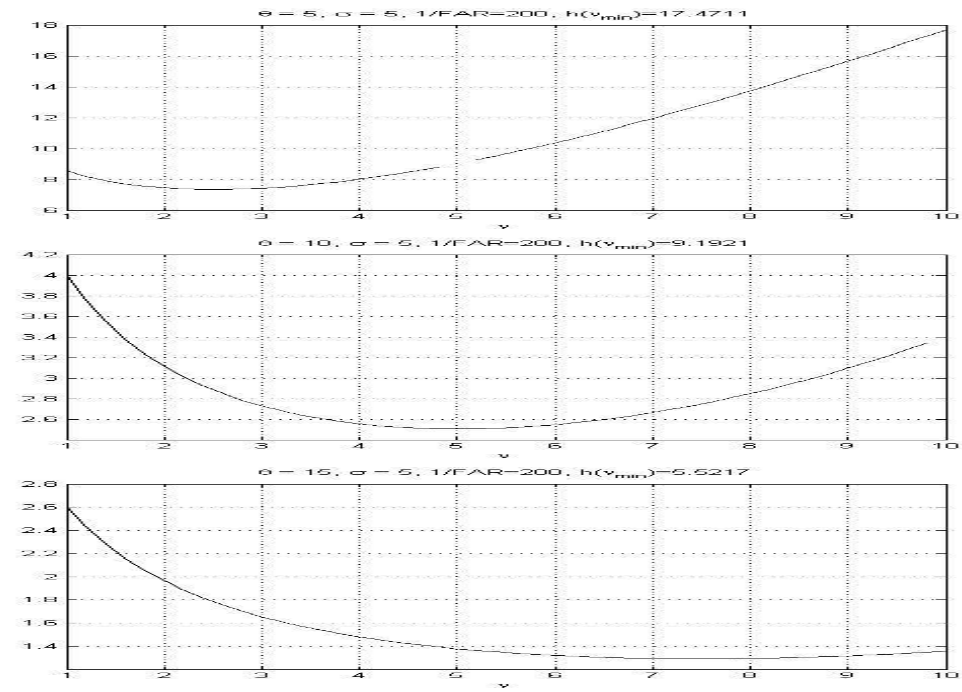

Figure 4.2: MTD(ν). $\sigma = 5,\ \theta = 5,\ 10$ and $15,\ \frac{1}{FAR} = 200$

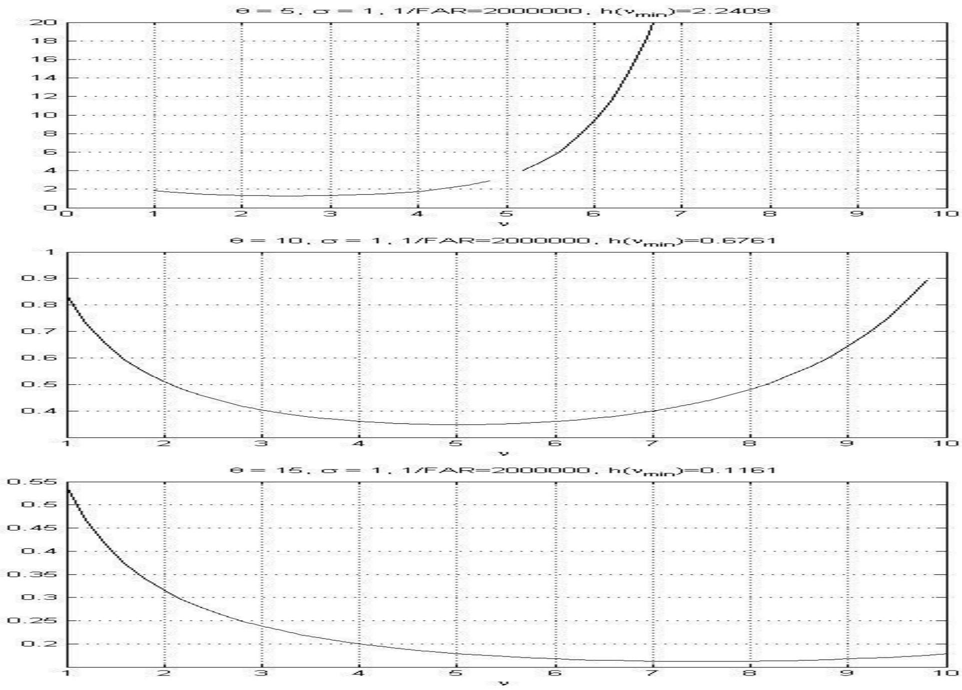

Figure 4.3: MTD(ν). $\sigma = 1,\ \theta = 5,\ 10$ and $15,\ \frac{1}{FAR} = 2000000$

Finding h

Since ν always is $\theta/2$, h is always $h(\theta/2)$. But h depends on more than ν. In Equations 2.16 and 2.15 it is seen that the threshold also depends on σ and the FAR. In Figure 4.4 the σ dependence can be seen and in Figure 4.5 the dependence of the FAR.

The curves look reasonable: In (4.4) it is seen that h is an increasing function of σ, intuitively this is correct since a larger variance forces the threshold to be larger to avoid false alarms. In (4.5) it is seen that h is an increasing function of 1/FAR. Also this intuitively seems to be correct; when less false alarms are desired, the threshold is increased.

Another property of h that must be fulfilled is $h > 0$. In Figure 4.4 this is not fulfilled for small σ and in Figure 4.5 this is not fulfilled for large FAR. It is concluded that this algorithm not always gives a threshold that can be used, particularly when σ is small and the FAR is large.

In Figure 4.6 it can be seen that h decreases as ν is increased, also this intuitively seems correct because when the drift variable is increased the random drift is decreased making it possible to choose the threshold smaller. This does not really matter since ν is always chosen as $\theta/2$, but it is seen that the implemented algorithm is working properly.

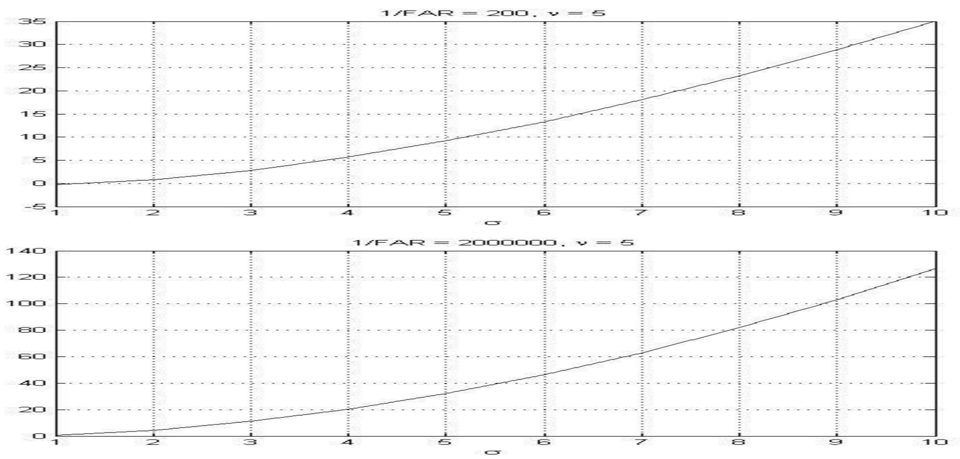

Figure 4.4: h is an increasing function of σ. In the top diagram $\frac{1}{FAR} = 200$ and in the bottom $\frac{1}{FAR} = 2000000$. In both diagrams $\nu = 5$.

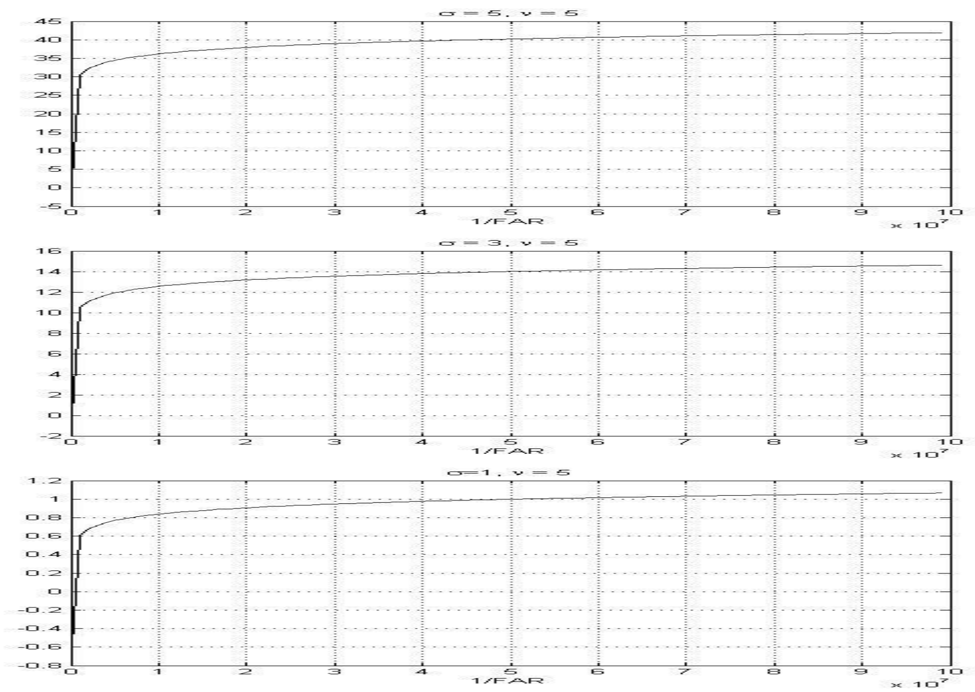

Figure 4.5: h;s dependance of the FAR. In the top diagram $\sigma = 5$, in the middle diagram $\sigma = 3$ and in the bottom diagram $\sigma = 1$. In all diagrams $\nu = 5$.

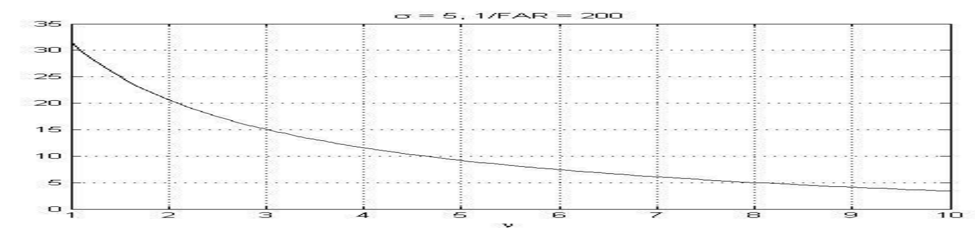

Figure 4.6: h is a decreasing function of ν.

Test on Gaussian Sequences

The algorithm above intuitively seems to work, here it will be tested on Gaussian sequences to see if the calculated parameters are working. In the figures the parameters that are called "suggested" have been calculated. Sample time is always 0.01.

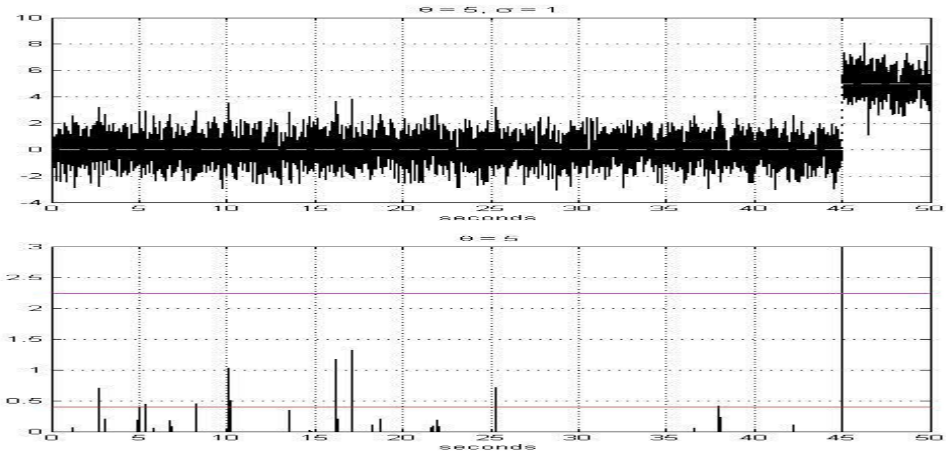

Figure 4.7: Input with $\theta = 5$ at time 45 sec and $\sigma^2 = 1$. $\nu = 2.5$ is suggested and for $\frac{1}{FAR} = 200$ $h = 0.3995$ is suggested, for $\frac{1}{FAR} = 2000000$ $h = 2.2409$ is suggested. When $h = 0.3995$ there are 8 false alarms. Sample time is 0.01.

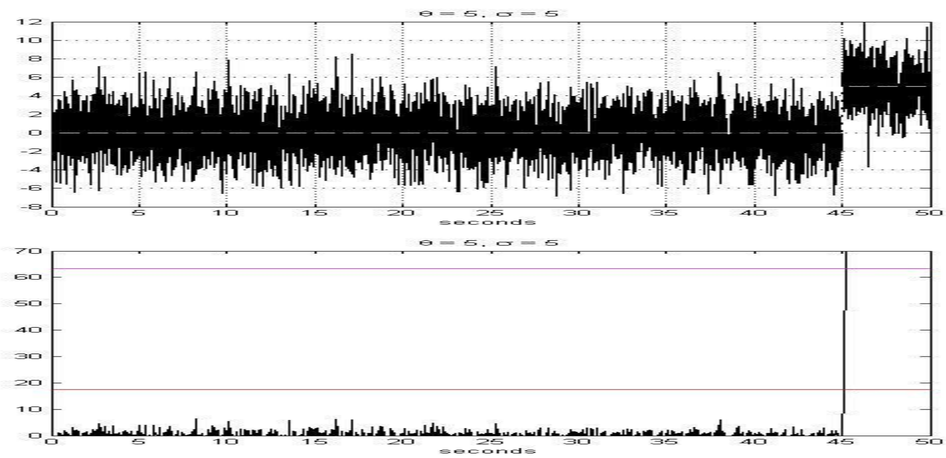

Figure 4.8: The input is a step at time 45 sec where $\theta = 5$ and $\sigma^2 = 5$. $\nu = 2.5$ is suggested and for $\frac{1}{FAR} = 200$ $h = 17.4711$ is suggested, for $\frac{1}{FAR} = 2000000$ $h = 63.2476$ is suggested.

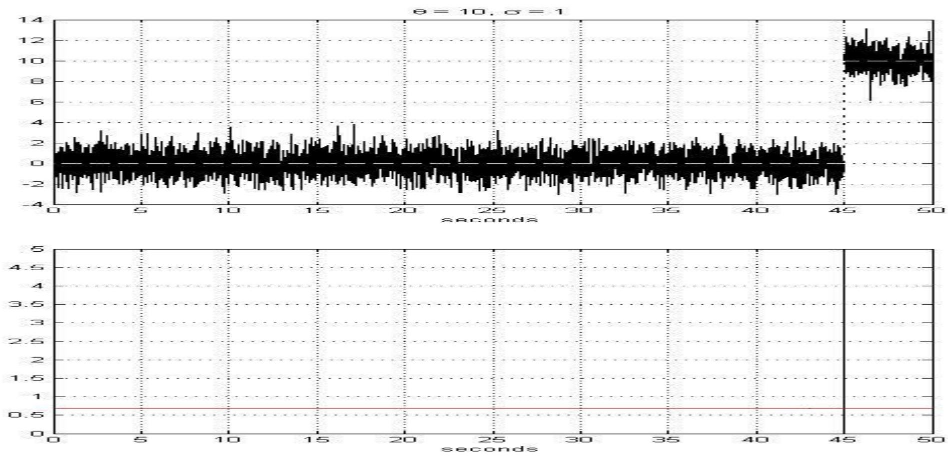

Figure 4.9: The input is a step at time 45 sec where $\theta = 10$ and $\sigma^2 = 1$. $\nu = 5$ is suggested and for $\frac{1}{FAR} = 200$ $h = -0.2449$ is suggested, for $\frac{1}{FAR} = 2000000$ $h = 0.6761$ is suggested.

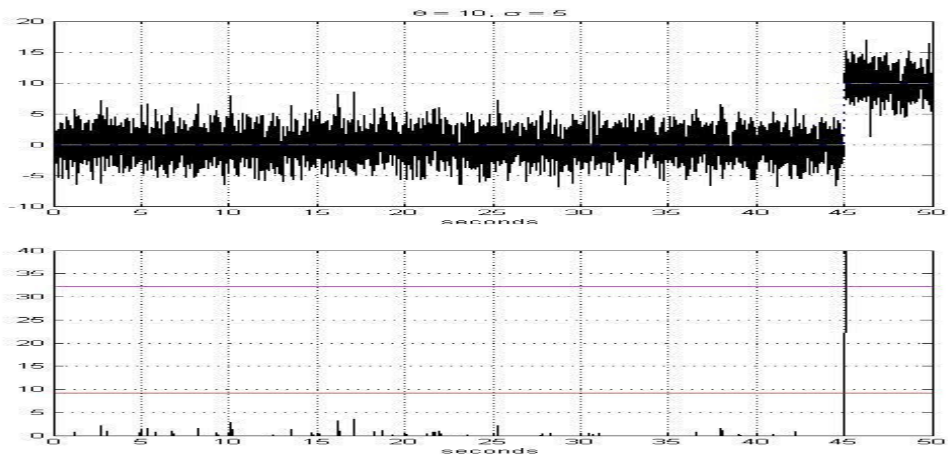

Figure 4.10: The input is a step at time 45 sec where $\theta = 10$ and $\sigma^2 = 5$. $\nu = 5$ is suggested and for $\frac{1}{FAR} = 200$ $h = 9.1921$ is suggested, for $\frac{1}{FAR} = 2000000$ $h = 32.1745$ is suggested.

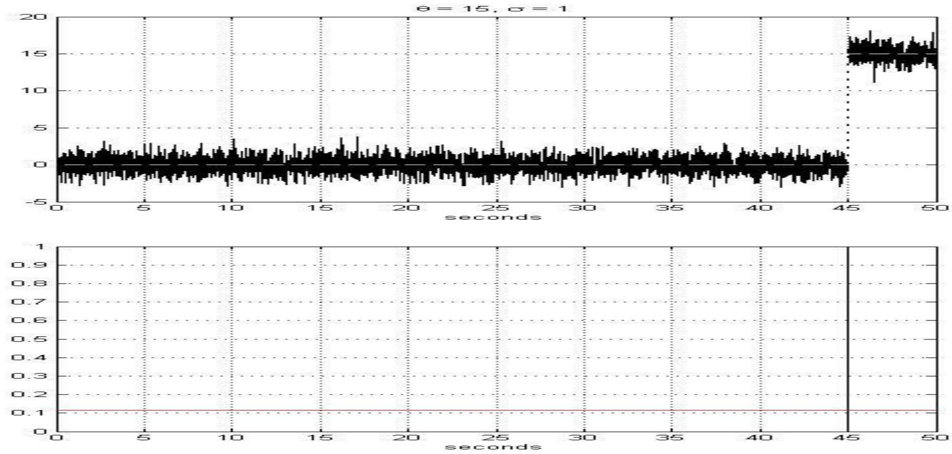

Figure 4.11: The input is a step at time 45 sec where $\theta = 15$ and $\sigma^2 = 1$. $\nu = 7.5$ is suggested and for $\frac{1}{FAR} = 200$ $h = -0.4979$ is suggested, for $\frac{1}{FAR} = 2000000$ $h = 0.1161$ is suggested.

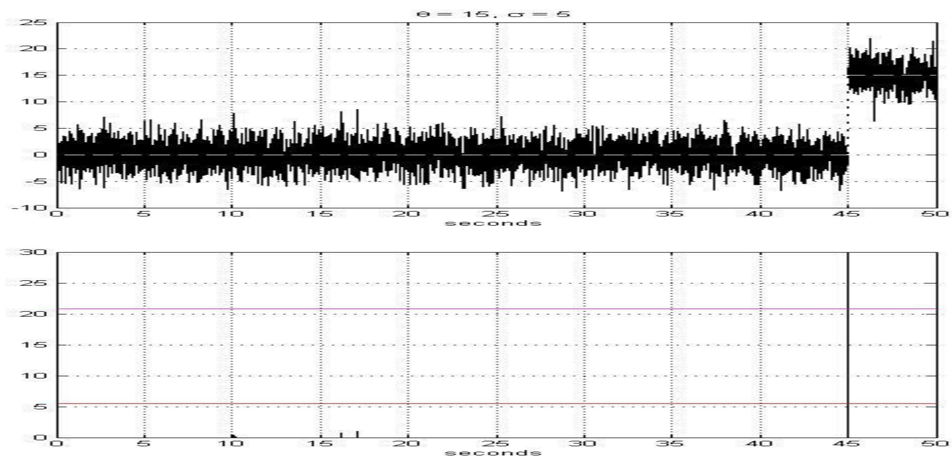

Figure 4.12: The input is a step at time 45 sec where $\theta = 15$ and $\sigma^2 = 5$. $\nu = 7.5$ is suggested and for $\frac{1}{FAR} = 200$ $h = 5.5217$ is suggested, for $\frac{1}{FAR} = 2000000$ $h = 20.8579$ is suggested.

Test of Calculated Parameters on Real Residuals

Conform the Residuals to Gaussian Distribution

The distribution of the residuals can be seen in Figure 4.13. In Figures 4.14, 4.15, 4.16 and 4.17 the residuals in different ways have been modified to fit to a Gaussian distribution curve. Sigma = σ = standard deviation.

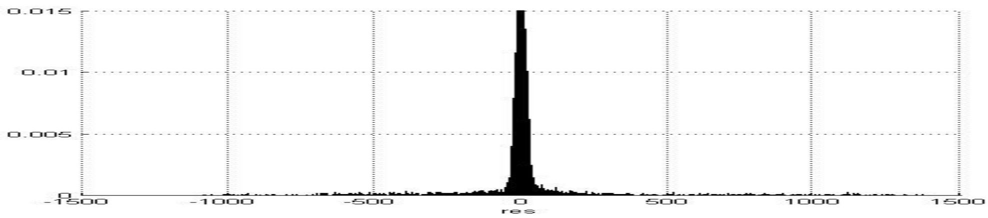

Figure 4.13: All the residuals have been normalized so that the integral is one.

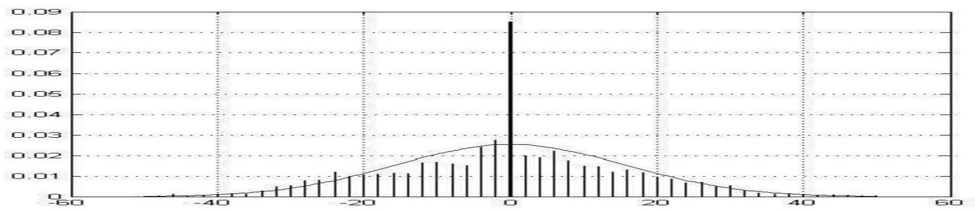

Figure 4.14: All the residuals have been normalized so that the integral is one. All residuals abs(res)>50 have been removed. The solid curve is a Gaussian curve with $\sigma = 15.7$ and mean=0. The calculated σ is 15.7.

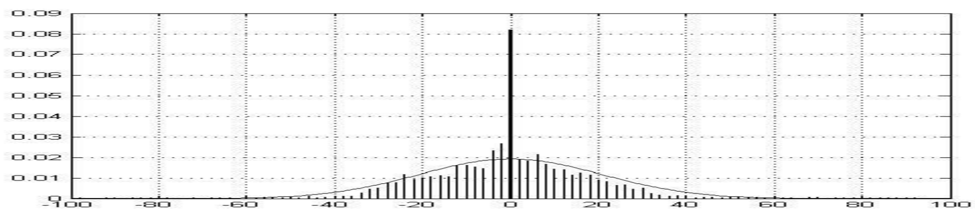

Figure 4.15: All the residuals have been normalized so that the integral is one. All residuals abs(res)>100 have been removed. The solid curve is a Gaussian curve with $\sigma = 20.7$ and mean=0. The calculated σ is 20.7.

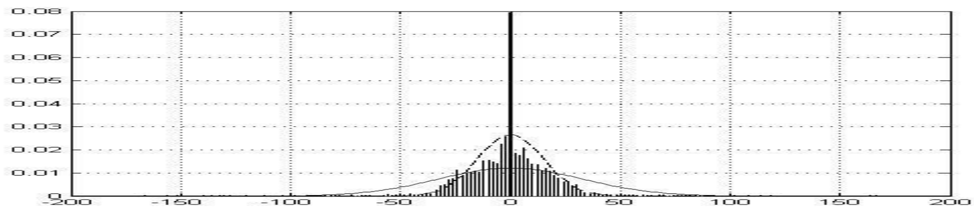

Figure 4.16: All the residuals have been normalized so that the integral is one. All residuals abs(res)>200 have been removed. The solid curve is a Gaussian curve with $\sigma = 33.1$ and mean=0. The calculated σ is 33.1. The dashed curve is a Gaussian curve with $\sigma = 15$ and mean=0.

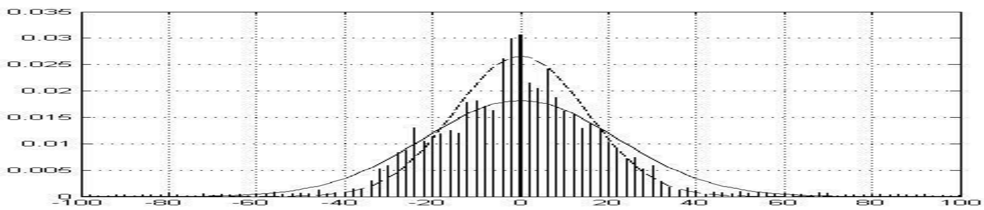

Figure 4.17: All the residuals have been normalized so that the integral is one. All residuals abs(res)>100 or =0 have been removed. The solid curve is a Gaussian curve with $\sigma = 21.9$ and mean=0. The calculated σ is 21.9. The dashed curve is a Gaussian curve with $\sigma = 15$ and mean=0.

Results with Conformed Residuals

First parameters that worked was searched for, but since no such parameters existed parameters that minimized θ were found, i.e. a drift parameter that is as “sensitive” as possible. In the figures there has not been any changes added i.e. no change should be detected.

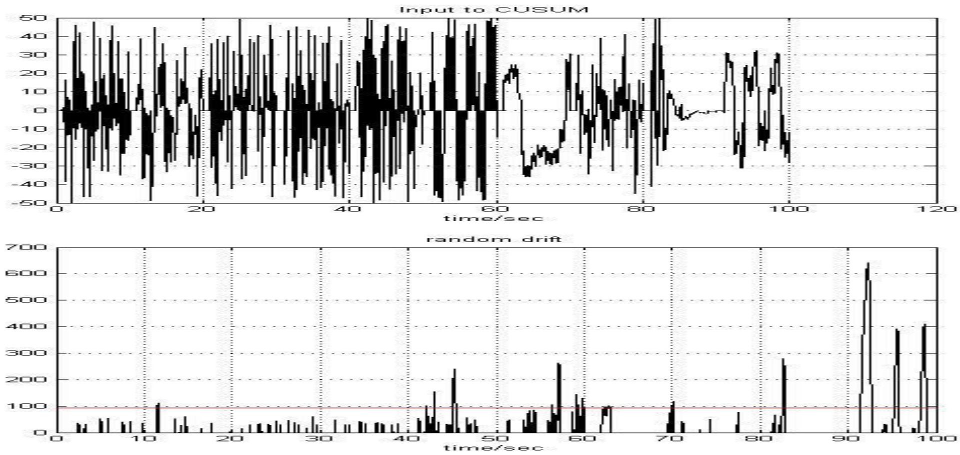

Figure 4.18: abs(res)$\leq$50 and $\geq$0. $\theta = 40$, $\sigma = 15.7$ and $\frac{1}{FAR} = 20000000$. The suggested parameters are: $\nu = 20$ and $h = 92.5$.

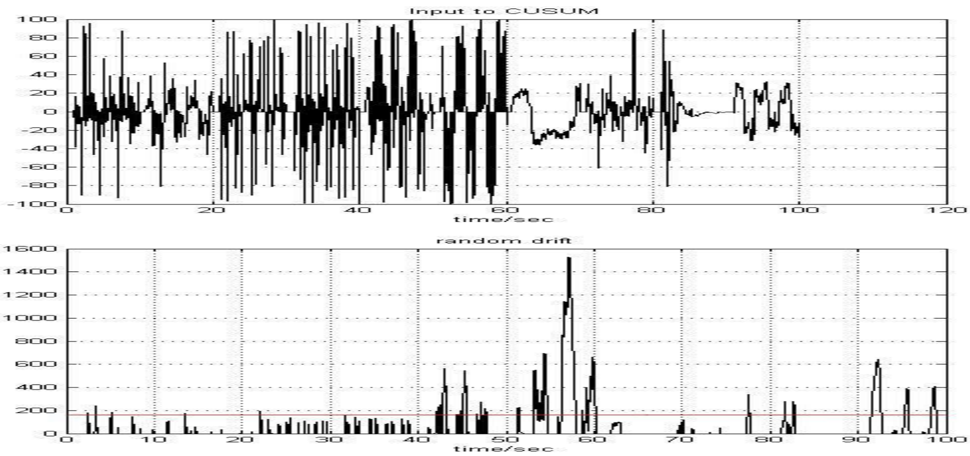

Figure 4.19: abs(res)$\leq$100 and $\geq$0. $\theta = 40$, $\sigma = 20.7$ and $\frac{1}{FAR} = 20000000$. The suggested parameters are: $\nu = 20$ and $h = 162.6$.

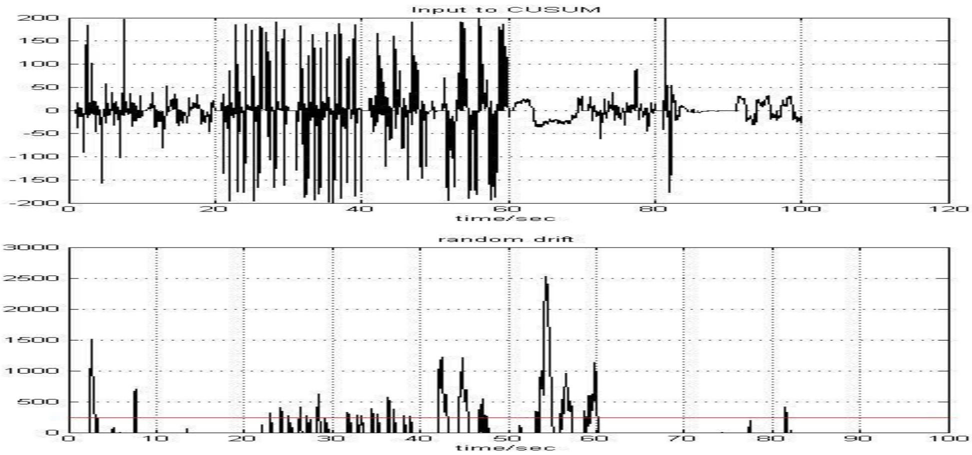

Figure 4.20: abs(res)≤200 and ≥0. $\theta = 70$, $\sigma = 33.1$ and $\frac{1}{FAR} = 20000000$. The suggested parameters are: $\nu = 35$ and $h = 237.1$.

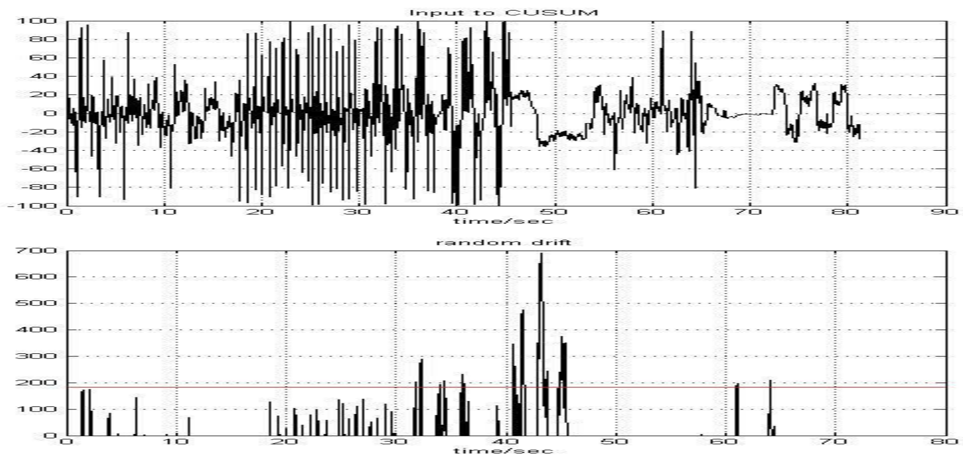

Figure 4.21: abs(res)≤100 and >0. $\theta = 40$, $\sigma = 21.9$ and $\frac{1}{FAR} = 20000000$. The suggested parameters are: $\nu = 20$ and $h = 182.2$ ($\sigma = 15$ gives $h = 84.2$).

4.1.2 CUSUM Parameters with Statistics - Variance Change

Finding θ

As seen in (2.19), (2.21), (2.22), (2.23) and (2.24), θ depends on the variance change, the mean value of the squared input sequence and the ARL value. In Figure 4.22 it is seen that θ is an increasing function of the variance change. This agrees with Equation 2.19.

In Figure 4.23 it is seen that θ is a decreasing function of the mean value of the squared input sequence, σ. This agrees with Equation 2.19 and intuitively seems correct since a small starting variance should lead to a smaller θ.

In Figure 4.24 θ is plotted as a function of the ARL. As the ARL is chosen larger, θ increases, this agrees with what one would expect. A larger θ decreases the random drift and causes longer detection times for changes = larger ARL.

Actually, none of the above written dependencies are used to determine θ. In Equation 2.19 it is seen that θ is defined by the input sequence's statistical properties. But it is seen that the functions seem to be correct and that the implemented algorithm works.

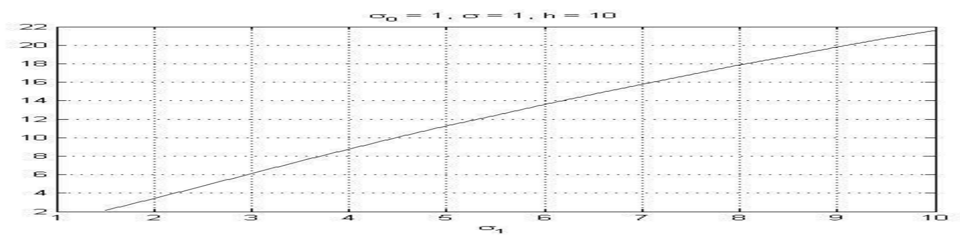

Figure 4.22: θ is an increasing function of the variance change.

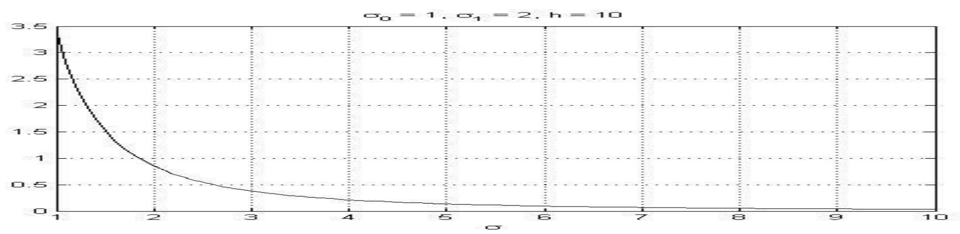

Figure 4.23: θ is a decreasing function of σ.

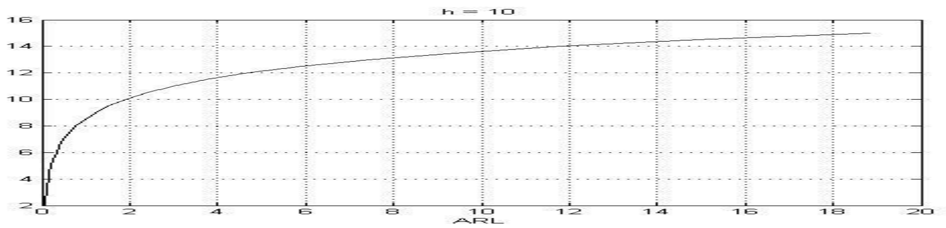

Figure 4.24: θ is an increasing function of the ARL.

Finding h

How h depends on the ARL and θ can be seen in Figures 4.25 and 4.26. That h increases as the ARL increases was expected since a larger ARL value means that it should take longer time to detect a change and this is exactly what happens when the threshold is chosen bigger.

That h decreases as θ increases also seems correct; when θ increases the random positive drift in the cumulative sum decreases. This makes it possible to choose the threshold smaller which is what Figure 4.26 shows.

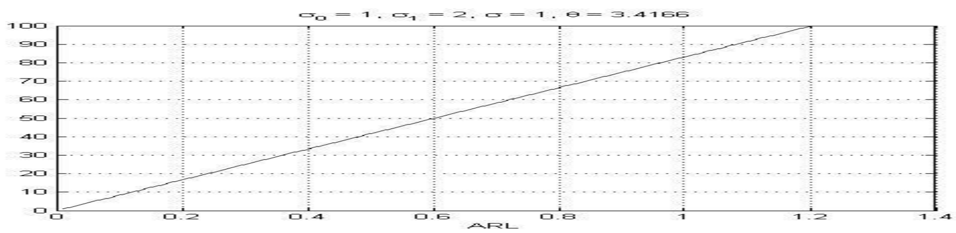

Figure 4.25: h is an increasing function of the ARL.

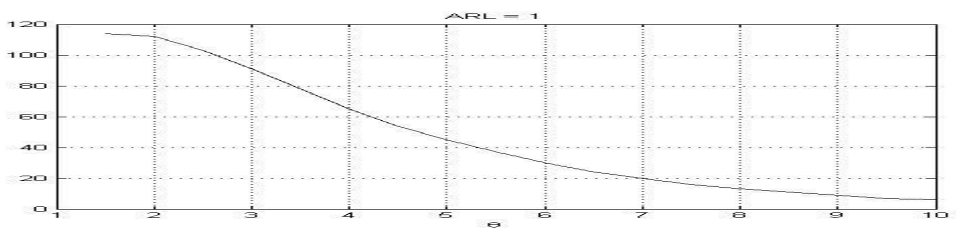

Figure 4.26: h is a decreasing function of θ.

4.1.3 Test of Parameters on Gaussian Sequences

The implemented algorithm that finds h in Equation 2.18 are here tested on Gaussian sequences.

As can be seen from the figures the variance changes tested are detected most of the time. The parameter θ is obviously not optimal but the suggested value is in most cases good and could be used as a good starting value.

The parameter h depends on how the ARL is chosen. In Figure 4.32 it can be seen that the threshold is calculated as negative. Increasing the ARL will make the suggested h positive and give it a more reasonable value.

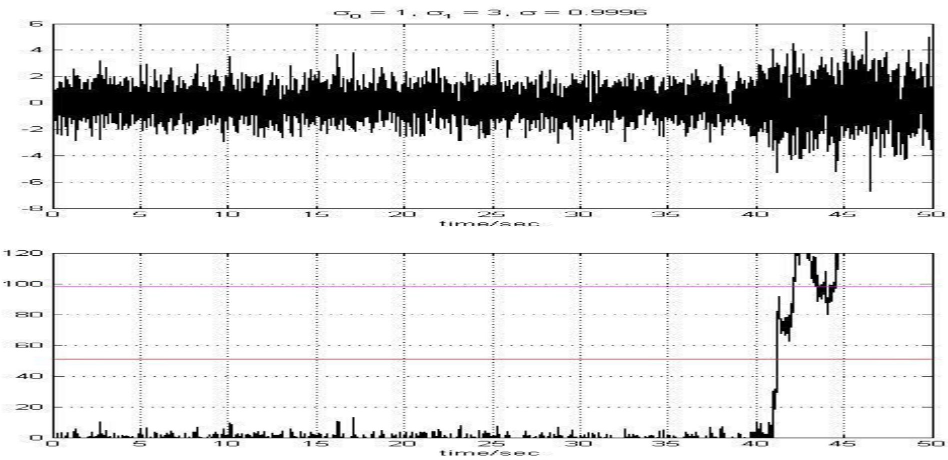

Figure 4.27: At time 40 sec the Gaussian input changes its variance from 1 to 3. ARL = 0.5 gives h = 51 and ARL = 1 gives h = 98. θ is 2.7177. Sample time is 0.01.

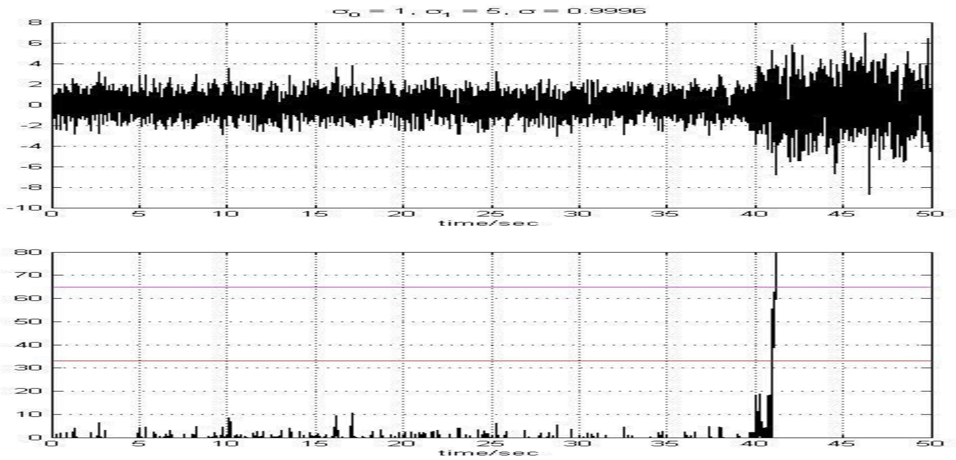

Figure 4.28: At time 40 sec the Gaussian input changes its variance from 1 to 5. ARL = 0.5 gives h = 33 and ARL = 1 gives h = 65. θ is 4.0503. Sample time is 0.01.

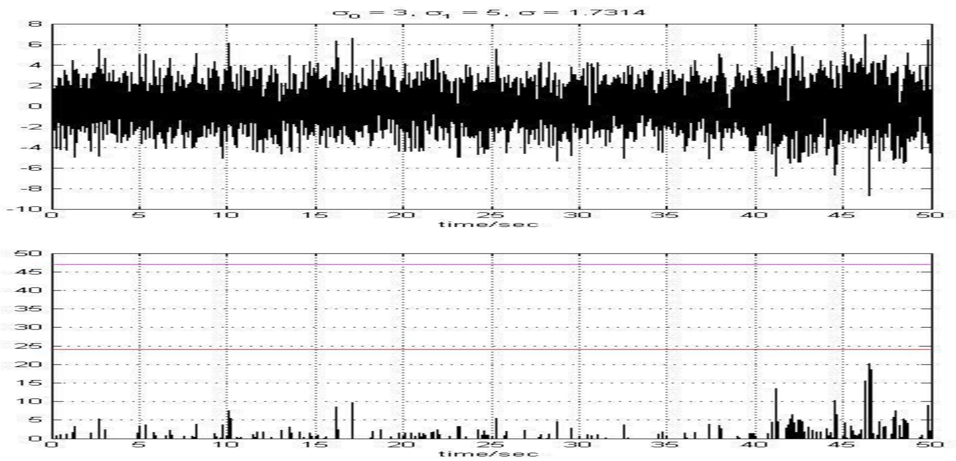

Figure 4.29: At time 40 sec the Gaussian input changes its variance from 3 to 5. ARL = 0.5 gives h = 24 and ARL = 1 gives h = 47. θ is 4.8963. Sample time is 0.01.

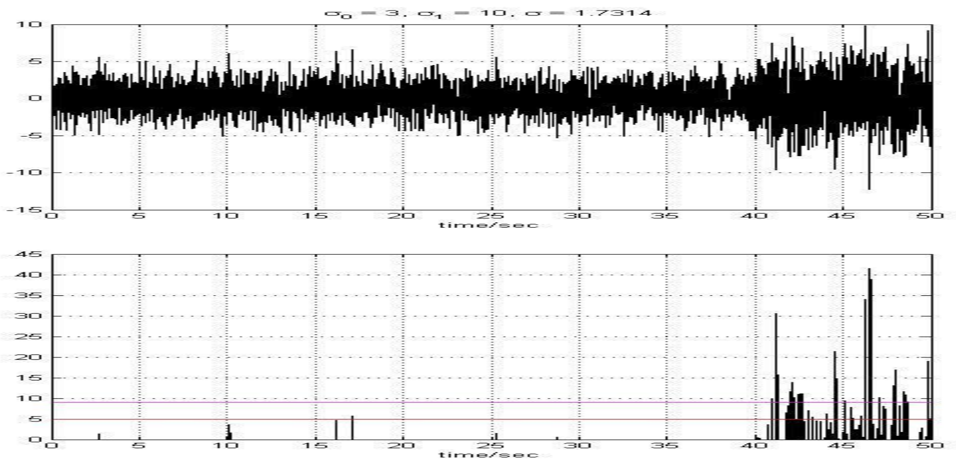

Figure 4.30: At time 40 sec the Gaussian input changes its variance from 3 to 10. ARL = 0.5 gives $h = 5$ and ARL = 1 gives $h = 9$. θ is 8.8814. Sample time is 0.01.

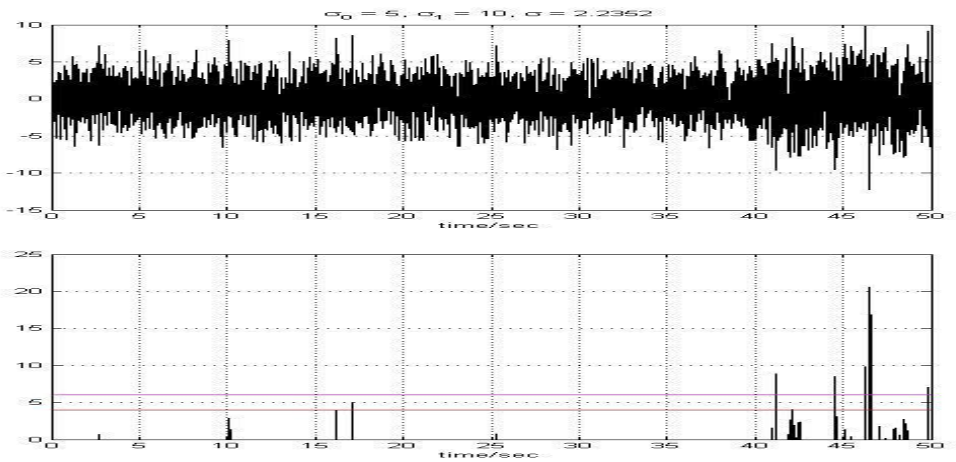

Figure 4.31: At time 40 sec the Gaussian input changes its variance from 5 to 10. ARL = 0.5 gives $h = 4$ and ARL = 1 gives $h = 6$. θ is 9.6261. Sample time is 0.01.

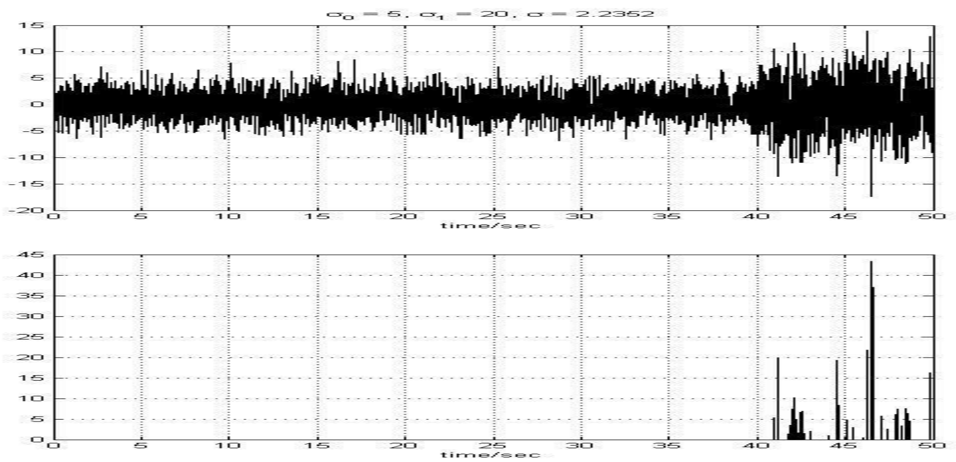

Figure 4.32: At time 40 sec the Gaussian input changes its variance from 5 to 20. ARL = 0.5 gives $h < 0$ and ARL = 1 gives $h < 0$. θ is 17.0955. Sample time is 0.01.

Test on Residuals - Variance Change

The conformed residuals in Figures 4.14, 4.15, 4.16 and 4.17 was tested. The parameters (σ_1 and ARL) have been chosen to get as good CUSUM parameters as possible. No working parameters were found.

residuals	σ	σ_0	σ_1	θ	ARL	h
(4.14)	15.7	15.7	15.71	247.75	6.44e50 4.83e50 3.22e50 1.61e50	4 3 2 1
(4.15)	20.7	20.7	20.71	748.45	-	-
(4.16)	33.1	33.1	33.11	4.89e3	-	-
(4.17)	21.9	21.9	21.91	937.64	-	-

In the three lower cases the the ARL values were to big for matlab to handle. No scaling was necessary since in Figure 4.24 it is seen that the ARL will increase and this will give worse results than in the first case.

4.2 Results - Trial and Error

Results from simulated errors and from a development vehicle are documented here.

4.2.1 Explanation of the Tests

The tests have been done in two different environments:

1. Made in an old ADYN where the residuals are set to zero the first second.
2. Made in the current ADYN environment, the only difference of importance is that the residuals are not set to zero the first second.

Explanation of the tests:

- CUSUM, normal CUSUM test, environment 1, tuned after measurement set A.
- double CUSUM, normal double sided CUSUM, environment 1, tuned after measurement set A.
- ACS, Adyn Compare Slow, integrates the residuals and sets the sum to zero periodically, if the sum exceeds a threshold an alarm is given, environment 1, tuned after measurement set A.
- ACF, Adyn Compare Fast, same as ACS with other parameter settings, environment 1, tuned after measurement set A.
- AC, Adyn Compare, if(abs(residual)>threshold) an alarm is given, environment 1, tuned after measurement set A.
- GMA, normal GMA test, environment 1, tuned after measurement set A.

- LCUSUM, local CUSUM tests acting in different intervals, environment 1, tuned after measurement set A.
- double LCUSUM, double sided LCUSUM, each CUSUM in every interval in LCUSUM is double sided, environment 1, tuned after measurement set A.
- double LGCUSUM, double sided local and global CUSUM, double LCUSUM combined with a normal double sided CUSUM, environment 2, tuned after measurement set B.
- ADYN, current ADYN, consists of GMA detectors with low α values.

The numbers tell when the change was detected, in time units. No change detected is noted with x. * means that there was a very short alarm in the beginning of the test, * is always combined with a number or a x.

4.2.2 Simulated Errors

No error

Method	Sc1	Sc2	Sc3	Sc4	Sc5	Sc6	Sc7	Sc8	Sc9	Sc10
CUSUM	x	x	x	x	x	x	x	x	x	x
ACS	x	x	2.9	1.8	x	1.8	x	x	2.0	x
ACF	x	x	3.0	1.9	x	x	x	x	x	x
AC	x	x	x	x	x	x	x	x	x	x
GMA	x	x	x	x	x	x	x	x	x	x
LCUSUM	x	x	x	x	x	x	x	x	x	x
double LCUSUM	x	x	x	x	x	x	x	x	x	x
double CUSUM	x	x	x	x	x	x	x	x	x	x
double LGCUSUM	x	x	x	x	x	x	x	x	x	x
ADYN	x	x	x*	x	x*	x	x*	x	x	x

Constant 20

Method	Sc1	Sc2	Sc3	Sc4	Sc5	Sc6	Sc7	Sc8	Sc9	Sc10
CUSUM	x	x	7.3	x	x	x	x	x	x	x
ACS	x	x	2.9	1.8	x	1.8	x	x	2.0	x
ACF	x	x	3.1	1.9	x	x	x	x	x	x
AC	x	x	x	x	x	x	x	x	x	x
GMA	x	x	x	x	x	x	x	x	x	x
LCUSUM	x	9.3p	x	x	8.9p	5.5p	4.8p	x	x	x
double LCUSUM	x	8.8	x	x	x	5.3p	5.2p	x	x	x
double CUSUM	x	x	6.2	x	x	x	x	x	x	x
double LGCUSUM	x	x	x	x	x	x	8.0	x	x	x
ADYN	x	x*	x*	x	x*	x*	x	x	x	x

Constant 30

Method	Sc1	Sc2	Sc3	Sc4	Sc5	Sc6	Sc7	Sc8	Sc9	Sc10
LCUSUM	3.7p	3.2p	x	7.8p	4.0p	4.9p	4.3p	3.4p	5.8p	3.6p

Constant 50

Method	Sc1	Sc2	Sc3	Sc4	Sc5	Sc6	Sc7	Sc8	Sc9	Sc10
CUSUM	x	x	6.1	x	x	x	x	x	x	x
ACS	x	x	2.9	1.8	x	1.8	x	x	2.0	x
ACF	x	x	3.1	1.9	x	x	x	x	x	x
AC	x	x	x	x	x	x	x	x	x	x
GMA	x	x	7.4	x	x	x	x	x	x	x
LCUSUM	2.0p	2.3p	2.6p	3.8p	2.5p	3.0p	3.5p	2.2p	3.2p	2.8p
double LCUSUM	2.8p	2.7p	3.5p	4.1p	3.6p	2.6p	3.7p	2.7p	3.7p	3.2p
double CUSUM	x	x	6.1	x	x	x	x	x	x	x
double LGCUSUM	7.3	6.3	6.1	x	x	x	5.6	6.7	8.4	7.1
ADYN	x	x*	x*	x	x*	x*	x	x	x	x

Constant 100

Method	Sc1	Sc2	Sc3	Sc4	Sc5	Sc6	Sc7	Sc8	Sc9	Sc10
CUSUM	x	x	5.1	2.7	x	x	x	x	x	x
ACS	x	x	2.0	1.8	x	1.8	x	x	1.9	x
ACF	x	x	3.1	1.9	x	x	x	x	x	x
AC	x	x	x	x	x	x	x	x	x	x
GMA	x	x	6.1	x	x	x	x	x	x	x
LCUSUM	2.0p	1.6p	2.3p	2.5p	2.5p	1.6p	1.5p	1.6p	2.6p	1.8p
double LCUSUM	1.7p	1.9p	1.8p	2.4p	1.8p	2.1p	1.9p	1.8p	2.3p	2.0p
double CUSUM	x	x	4.0	4.3	x	4.9	8.4	x	8.7	x
double LGCUSUM	3.5	3.1	4.0	4.2	4.0	5.0	2.9	3.1	4.6	3.2
ADYN	x	x*	x*	x	x*	x*	x	x	x	x

Constant 500

Method	Sc1	Sc2	Sc3	Sc4	Sc5	Sc6	Sc7	Sc8	Sc9	Sc10
double LGCUSUM	0.5	0.6	0.6	0.8	0.4	0.7	0.5	0.6	0.7	0.6
ADYN	x	x*	4.0*	x	x*	x*	x	x	x	x

Constant 1000

Method	Sc1	Sc2	Sc3	Sc4	Sc5	Sc6	Sc7	Sc8	Sc9	Sc10
double LGCUSUM	0.2	0.3	0.3	0.3	0.2	0.3	0.2	0.3	0.3	0.2
ADYN	x	x*	1.9*	1.0	x*	1.6*	x	x	0.9	x

Constant 1500

Method	Sc1	Sc2	Sc3	Sc4	Sc5	Sc6	Sc7	Sc8	Sc9	Sc10
double LGCUSUM	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2
ADYN	0.2	0.2	0.2	0.2	0.0	0.2	0.2	0.2	0.2	0.2

Band limited white noise, noise power 50

Method	Sc1	Sc2	Sc3	Sc4	Sc5	Sc6	Sc7	Sc8	Sc9	Sc10
CUSUM	x	x	7.3	x	x	x	x	x	x	x
ACS	x	x	2.9	1.8	x	1.8	x	x	2.0	x
ACF	x	x	3.0	1.9	x	x	x	x	x	x
AC	x	x	6.3	x	x	x	x	x	x	x
GMA	x	x	9.5	x	x	x	x	x	x	x
LCUSUM	x	x	x	x	2.3v	5.3p	6.2p	x	x	5.2p
double LCUSUM	x	5.7v	6.6a	x	2.2v	4.0p	2.0a	x	4.1v	5.2p
double CUSUM	x	x	8.2	x	x	x	x	x	x	x
double LGCUSUM	4.7	2.7	2.6	3.8	3.7	3.8	4.8	5.3	5.2	5.4
ADYN	x	x*	x*	x	x*	x*	x	x	x	x

Band limited white noise, noise power 100

Method	Sc1	Sc2	Sc3	Sc4	Sc5	Sc6	Sc7	Sc8	Sc9	Sc10
CUSUM	x	x	7.3	x	x	x	x	x	x	x
ACS	x	x	2.9	1.8	x	1.8	x	x	x	x
ACF	x	x	3.0	1.9	x	x	x	x	x	x
AC	x	x	6.3	x	x	x	x	x	x	x
GMA	x	x	7.4	x	x	x	x	x	x	x
LCUSUM	7.4v	6.2p	7.0v	x	2.3v	2.5v	4.6p	7.7p	7.5p	2.9p
double LCUSUM	x	5.8v	3.7a	5.7v	2.3v	2.5v	2.0a	2.1a	x	5.0p
double CUSUM	x	x	7.3	x	x	x	x	x	x	x
double LGCUSUM	2.7	1.6	1.6	2.4	2.1	2.2	2.7	3.0	3.0	3.0
ADYN	x	x*	x*	x	x*	x*	x	x	x	x

Sine wave, amp: 40. freq 20 Hz

Method	Sc1	Sc2	Sc3	Sc4	Sc5	Sc6	Sc7	Sc8	Sc9	Sc10
CUSUM	7.1a	9.6a	2.9a	2.2a	6.3a	5.5a	x	x	8.3a	x
ACS	x	x	2.9	1.8	x	1.8	x	x	2.0	x
ACF	x	x	3.0	1.9	x	x	x	x	x	x
AC	x	x	1.0	2.1	x	7.7	x	x	2.1	x
GMA	7.1a	7.8a	4.1a	4.3a	6.4a	6.2a	7.8a	8.7a	8.3a	8.4a
LCUSUM	1.4v	1.2v	1.4v	1.5v	1.2v	1.3v	1.3v	1.2v	1.7v	1.2v
double LCUSUM	1.6v	1.6v	1.6v	4.1v	1.5v	1.6v	1.5v	1.7v	1.4v	2.8v
double CUSUM	x	x	3.1	2.7	x	x	x	x	x	x
double LGCUSUM	1.9	1.8	5.9	7.8	2.7	3.2	1.6	1.8	3.3	1.5
ADYN	x	x*	x*	x*	x*	x*	x	x	x	x

Slope: 20 start 5

Method	Sc1	Sc2	Sc3	Sc4	Sc5	Sc6	Sc7	Sc8	Sc9	Sc10
CUSUM	x	x	7.4	x	x	x	x	x	x	x
ACS	x	x	2.9	1.8	x	1.8	x	x	2.0	x
ACF	x	x	3.0	1.9	x	x	x	x	x	x
AC	x	x	x	x	x	x	x	x	x	x
GMA	x	x	x	x	x	x	x	x	x	x
LCUSUM	8.1	8.0	8.7	8.5	7.9	8.0	6.4	8.3	7.8	7.6
double LCUSUM	8.5p	8.2p	9.1p	8.7p	6.1v	6.3v	6.7p	8.3p	8.1p	8.2p
double CUSUM	x	x	8.2	x	x	x	x	x	x	x
double LGCUSUM	x	x	8.2	x	x	x	8.9	x	x	x
ADYN	x	x*	x*	x	x*	x*	x	x	x	x

Slope: 50 start 5

Method	Sc1	Sc2	Sc3	Sc4	Sc5	Sc6	Sc7	Sc8	Sc9	Sc10
CUSUM	x	x	7.3	x	x	x	x	x	x	x
ACS	9.7	9.2	2.9	1.8	9.2	1.8	9.2	9.2	2.0	9.2
ACF	x	x	3.0	1.9	x	x	x	x	x	x
AC	x	x	x	x	x	x	x	x	x	x
GMA	x	x	8.2	x	x	x	x	x	x	x
LCUSUM	6.7	6.4	6.8	6.7	6.1	6.0	5.8	6.7	6.6	6.4
double LCUSUM	7.0p	6.7p	7.7p	7.3p	5.9v	7.1p	6.0p	6.9p	6.8p	6.8p
double CUSUM	9.6	9.5	7.3	8.0	9.5	8.2	9.1	9.6	9.5	9.6
double LGCUSUM	8.7	8.5	8.0	8.6	8.8	9.4	8.0	8.6	8.4	8.6
ADYN	x	x*	x*	x	x*	x*	x	x	x	x

Slope: 500 start 5

Method	Sc1	Sc2	Sc3	Sc4	Sc5	Sc6	Sc7	Sc8	Sc9	Sc10
CUSUM	6.3	6.3	5.7	6.1	6.3	6.4	6.3	6.3	6.3	6.3
ACS	5.9	5.8	2.9	1.8	5.9	1.8	5.8	5.8	2.0	5.8
ACF	7.0	7.0	3.0	1.8	7.1	7.1	7.0	7.0	7.0	7.0
AC	8.0	8.0	6.1	7.8	7.9	7.5	8.0	8.1	8.0	8.1
GMA	6.7	6.7	6.0	6.2	6.6	6.7	6.7	6.7	6.7	6.7
LCUSUM	5.4	5.4	5.5	5.8	5.2	5.4	5.2	5.2	5.2	5.2
double LCUSUM	5.5v	5.4v	5.7v	5.7v	5.3v	5.2v	5.2v	5.2v	5.2v	5.0a
double CUSUM	6.0	6.0	5.9	6.0	6.0	6.0	6.0	6.0	6.0	6.0
double LGCUSUM	5.8	5.6	6.0	5.9	5.9	6.2	5.4	5.3	5.3	5.3
ADYN	7.2	7.4*	6.1*	7.4	7.3*	7.3*	7.2	7.3	7.3	7.3

Chapter 5

Conclusions

5.1 Obtaining CUSUM Parameters with Statistics - Mean Change

5.1.1 Gaussian Sequences

It is clear that in most cases the algorithm works and parameters that work are suggested. In some cases when σ is small and $\frac{1}{FAR}$ is small a negative value of the threshold is suggested e.g. in Figure 4.11. It is concluded that when σ is small, $\frac{1}{FAR}$ is small or θ is big (making ν big), h can be suggested as to small. This agrees with how h depends on these variables. A solution is to choose $\frac{1}{FAR}$ larger.

But the threshold should never be chosen negative unless FAR is chosen as ∞ which was not the case, and if it was, $h = 0$ would be more appropriate. The main problem with the suggested parameters is exactly this; they do not coincide with the FAR that has been chosen. If one chooses FAR to be one, then there should in average be one false alarm per sample and nothing else. This is not the case with the implemented algorithm. The reason for this might be the approximations made in the equations used and that the change detection theory is not accurate enough.

Looking at the results it is seen that the FAR in some cases agrees better with the chosen one than in other cases. In general it seems to agree better the smaller θ is.

That ν always is chosen as $\theta/2$ is not good. The first problem is that in most cases θ is not known and the other problem is that ν can be chosen better. But if instead of calling θ the change to be detected it is called the *minimum* change to be detected there will be a big improvement. This would lead to ν not being suggested as very big making the CUSUM detector unsensitive to small changes when a big change should be detected.

Looking at the performance of the whole CUSUM algorithm with the chosen parameters there is much to be wished for; someone who has found parameters for numerous CUSUM detectors with trial and error will see that the parameters that has been suggested in the results are not very good. They work, but they are not optimal. With trial and error anyone who knows the CUSUM algorithm can find parameters that are more sensitive to disturbances and that are faster and it does not take long time to find them, in particular since the sequences are Gaussian.

Now the question is if this theoretically systematic way of finding the parameters can be used. It can: as can be seen in section 4.1.1 changing the FAR and θ will change the threshold, h and the drift parameter, ν in the right direction. It is known that the FAR value is not correct in most cases but θ agrees pretty good with the correct value. Having this in mind the FAR and θ can be changed to come closer to the desired CUSUM settings. Imagine two sliding bars where the FAR and θ can be chosen, the markings for the values on the sliding bar of the FAR are not very precise and more precise for θ.

One might wonder why it is better to "play" with the FAR and θ instead of "playing" with h and ν directly which is exactly what trial and error is. For someone who knows how the CUSUM algorithm works there is no reason, all the theory and all the equations is useless, finding h and ν using trial and error on the FAR and θ is just a detour through theory, approximations and calculations. But for someone who do not know how the CUSUM algorithm works it can be a very helpful tool. Simply drag the FAR and θ in the desired direction and the CUSUM parameters will adjust after the specifications.

5.1.2 Test on Residuals

Even though $\frac{1}{FAR}$ and θ were chosen in a way to get good parameters, no working parameters could be found. The suggested ν's gave a good looking random drift. But since ν always is $\theta/2$ and this was known there was no problem to find a good θ. It is also clear that the value of FAR that was inserted to the algorithm and the observed FAR differ a lot.

This algorithm can not be used to find good parameters, it can be used to find starting values for the parameters that then has to be changed and optimized.

The reason for why it does not work on the real residuals are: the conformed residuals are not Gaussian enough and the algorithm does not work so good for large θ and σ.

5.2 CUSUM Parameters with Statistics - Variance Change

5.2.1 Gaussian Sequences

In the results it is seen that the suggested parameters in most of the tested cases work. When σ_0 is small the suggested θ works very good. But when σ_0 increases, θ increases to fast. This is also the case when σ_1 increases, look at the graphs in Figures 4.22 and 5.1. The result is that finding parameters for the following cases do not work well:

1. the difference between σ_1 and σ_0 is large
2. σ_0 or σ_1 is large

The fact that θ grows so fast is the reason for why h is calculated to be to small or even negative in the same cases as θ is to big. The graph in Figure 4.26 drops below zero at $\theta \approx 15$.

Can this implemented algorithm that uses the ARL function to find the parameters be used? When σ_0 and σ_1 are small it works well and the calculated parameters are good. But when σ_1 grows the calculated parameters are not good and the detector works poorly, if at all. Since θ in a way is static and only depends on the statistical properties of the input sequence not much can be done to change θ. h on the other hand can be changed by changing the ARL value. But just as in the case with mean change discussed in (5.1.1), the ARL value is not correct in magnitude. If ARL is chosen as three for a certain θ then the change should be detected after three samples. This is not the case, but if the ARL value is increased h will increase and this will lead to what is desired: that it takes longer time to detect a change. So the ARL value can be used to change h in the correct direction. It would be easier to change h directly though, and it would work just as good.

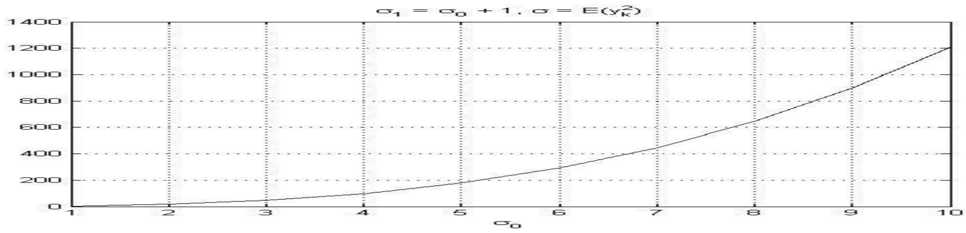

Figure 5.1: θ strongly increases as σ_0 and σ_1 increase.

5.2.2 Test on Residuals

In all cases θ is chosen too big. This because σ_0 and σ_1 are big. This phenomenon was discussed in the case with Gaussian sequences. σ_2 was chosen to minimize θ. A θ that is bigger than the limit where the residuals have been cut off does not work.

To get a positive threshold the ARL value has to be chosen absurdly big. The reason for this is that θ is big, see Figures 4.25 and 4.26.

The results show that it does not work even though the ARL value is chosen absurdly big. The parameters are not Gaussian "enough" and the standard deviation is too large. And this algorithm does not work with too large standard deviations.

5.3 Change Detectors Tuned with Trial and Error

5.3.1 CUSUM and GMA

The differences between the CUSUM and GMA detectors are very small. The CUSUM detector is slightly better overall. What limit them are the residuals with the largest magnitudes. The CDAs are tuned never to give false alarms, of course errors smaller than the largest residuals can not be detected if the CDAs are tuned with big margins. To get around this problem it has been

Scenario	ν	h
scenario_01	200	20000
scenario_02	25	1000

Table 5.1: CUSUM parameters for two different driving scenarios.

suggested having different parameter settings for different driving scenarios. Different scenarios give different residuals. The problem is to decide which driving scenario the driver currently is in, how should this decision be taken? This is a complex problem which is difficult to solve. One idea is to study how the variance of the residuals depend on the time derivative of the desired motor angle δ_{Md}.

5.3.2 Local CUSUM

To get around this the residuals can be split up in magnitude, i.e. different CDAs act in different intervals, see the "filter" in chapter 3. Consider the following example: in one of the driving scenarios used, called "scenario_01" the residuals are in the interval -1080 < res < 1266. In another driving scenario called "scenario_02" the residuals are in the interval -36 < res < 31, see Figure 3.1. Using a CUSUM detector, the parameters in Table 5.1 would work.

To not give false alarms when covering both these scenarios the parameters in scenario_01 has to be used. These parameters are far from optimal for the scenario_02 scenario and prevent small errors to be detected in this scenario. But if the residuals are split up and different CDAs are used in different magnitude intervals this limitation will be circumvented.

It is not the same thing as having different CDAs for different driving scenarios but the difference is not big if the intervals are well chosen. The results also show that this approach can detect much smaller errors and there is a big improvement in detection speed.

One big problem is that there is a chance for false alarms. If the maximum residual is of the magnitude 1266, residuals can have *any* value between 0 and 1266 (absolute value for ease). This means that if a local CUSUM acts in the interval 0 < res < 100 and one particular driving scenario which generates residuals of magnitude 99 is driven, it would generate a false alarm regardless of how big the margin is. The thing is that the drift parameter must be smaller than 100 to make any sense. Because of this the intervals the local CUSUM's act in must be very carefully chosen.

Other drawbacks with this approach is that is uses more computational power and that it needs more data to tune the parameters correctly.

5.3.3 Double Sided

The double sided change detectors do not show a big improvement compared to the one sided. When the double sided CDAs were tuned, the parameters in the CUSUM detectors often were decreased to half which possibly could lead to that errors that were half the size could be detected. That this is not the case in the simulations with constant errors we can not explain, one explanation could be how the parameters have been chosen but this should not affect the results as much as observed. That no improvement at all can be expected when testing on real data can be explained with the fact that if the double sided detectors have parameters half the size of the one sided, then there are equally many residuals that are positive and negative. And since the double sided detectors only get half the residuals, the performance will be the same as for the one sided.

5.3.4 AC, ACS, ACF

Presumably ACF/ACS can give results similar to CUSUM and GMA if they are tuned not to give false alarms.

AC is very fast when it detects an error, but often it does not. The changes detected by AC are so big that any of the other detectors also will detect it fast, it is redundant.

5.3.5 LGCUSUM and ADYN

LGCUSUM is slower than LCUSUM+CUSUM and can not detect as small errors. This is expected since LGCUSUM is tuned after more data where LCUSUM gives false alarms.

LGCUSUM can detect smaller errors and is almost always faster than the existing ADYN change detectors. ADYN is very fast when it detects errors, the parameters in ADYN are chosen not to be sensitive to small changes but to be very fast for larger errors. For large errors ADYN is the fastest of all of the detectors, the difference is small though. Sometimes ADYN gives false alarms and short alarms in the beginning of the simulations.

5.3.6 Final Conclusion

The change detectors tuned here are tuned after specific data and most of the simulations are also made on these data. One could argue and say that of course they will work the best on these data, but not on other. Measurements from a development vehicle show that this is not the case, but yes, as written above it is possible that LCUSUM give false alarms in other driving scenarios. The measurements have been chosen so that this will not happen.

It is clear that the approach with split up residuals is the best. It is almost always the fastest and it can detect much smaller errors than the others. It would be interesting to test this approach with GMA detectors. If the residuals are not split up the change detectors has to be tuned in one of the following ways:

- To be able to detect small changes with low MTD in normal driving scenarios. Always give false alarms in extreme maneuvers (large residuals).
- Larger MTD and not able to detect small changes in normal driving scenarios. Never give false alarms in extreme maneuvers.

But with the technique LGCUSUM uses, a good compromise is made because it is fast, able to detect small changes and the possibility for false alarms in extreme maneuvers is small.

From what is written above the following change detector seems to be the best: LGCUSUM without the double sided detectors, these clearly do not help and only use computational power. Also choose the thresholds and drift parameters a little bit larger making the margins larger to prevent possible false alarms from other driving scenarios, and finally test on a large number of scenarios.

Compared to the existing ADYN which do not detect constant actual-desired angle differences that are as big as 1000 degrees LGCUSUM is a big improvement. 1000 degrees in the motor correspond to the car turning with a radius of 80 meters (compare to a curve on autobahn). Maybe constant errors of size 1000 is not common, but it should preferably be detected.

Chapter 6

Plausibility of the Wheel Velocities

As written in the introduction it is important that the calculated car velocity is correct. The car velocity is computed with the aid of the four weighted wheel velocities. How should erroneous wheel velocities be detected? The idea was to let CUSUM detectors act on different residuals generated from the four different wheel velocities.

6.1 Approaches

The velocities of the four wheels were first low-pass filtered, then residuals were formed using two different strategies. In Table 6.1 the notation used can be seen.

wheel	notation
Front Left	v_{FL} or v_1
Front Right	v_{FR} or v_2
Rear Left	v_{RL} or v_3
Rear Right	v_{RR} or v_4

Table 6.1: Notation used.

6.1.1 Compare Wheel to Wheel

Wheel to Wheel A

Every wheel is compared to all other wheels and the following residuals are formed:

$$\begin{aligned}\epsilon_{12} &= |v_1 - v_2| \\ \epsilon_{13} &= |v_1 - v_3| \\ \epsilon_{14} &= |v_1 - v_4| \\ \epsilon_{23} &= |v_2 - v_3| \\ \epsilon_{24} &= |v_2 - v_4| \\ \epsilon_{34} &= |v_3 - v_4|\end{aligned} \tag{6.1}$$

CUSUM detectors act on these residuals and generate the corresponding alarm$_{ij}$. These can be used in different ways to detect different errors. Here they have been used to as safe as possible detect as many different kinds of errors as possible i.e. different combinations of wheel velocity errors among the wheels. The implemented function recognizes 13 combinations of errors among the wheels:

0. No error.
1. v_{FL} is erroneous.
2. v_{FR} is erroneous.
3. v_{RL} is erroneous.
4. v_{RR} is erroneous.
5. v_{FL} **and** v_{FR} differ from v_{RL} **and** v_{RR}. Either v_{FL} **and** v_{FR} are erroneous or v_{RL} **and** v_{RR} are erroneous.
6. v_{FL} **and** v_{RL} differ from v_{FR} **and** v_{RR}.
7. v_{FL} **and** v_{RR} differ from v_{FR} **and** v_{RL}.
8. v_{RL} and v_{RR} correct, v_{FL} differ from all other wheels, v_{FR} differ from all other wheels: v_{FL} erroneous, v_{FR} erroneous.
9. v_{FL} erroneous, v_{RL} erroneous.
10. v_{FL} erroneous, v_{RR} erroneous.
11. v_{FR} erroneous, v_{RL} erroneous.
12. v_{FR} erroneous, v_{RR} erroneous.
13. v_{RL} erroneous, v_{RR} erroneous.

Note: In 5, 6 and 7 the velocities within the pairs *are not different*. In 8, 9, 10, 11, 12 and 13 two wheels has erroneous values, these erroneous values *differ from each other*, why it is likely that they are not correct.

Wheel to Wheel B

The residuals in Equations 6.1 are also used here. They are combined to only detect errors on single wheels and hopefully faster.

6.1.2 Compare Wheel to Average

Average values are formed and compared in the following way:

$$\begin{aligned}
\epsilon_1 &= |v_1 - (v_2 + v_3 + v_4)/3| \\
\epsilon_2 &= |v_2 - (v_1 + v_3 + v_4)/3| \\
\epsilon_3 &= |v_3 - (v_1 + v_2 + v_4)/3| \\
\epsilon_4 &= |v_4 - (v_1 + v_2 + v_3)/3| \\
\epsilon_{front_back} &= |(v_1 + v_2)/2 - (v_3 + v_4)/2| \\
\epsilon_{sides} &= |(v_1 + v_3)/2 - (v_2 + v_4)/2| \\
\epsilon_{diag} &= |(v_1 + v_4)/2 - (v_2 + v_3)/2|
\end{aligned} \tag{6.2}$$

CUSUM detectors act on these residuals and form the corresponding alarms. When the residuals are averaged information is lost. This method turned out to not be able to detect as many different errors as when using (6.1). Assume $v_{FL} = v_1$ is erroneous and look at the residuals in (6.2). It is seen that *all* CUSUM detectors will give an alarm. If this method should be used the alarm time has to be checked i.e. which detector gives an alarm first.

6.1.3 Taking Wheel Placement into Consideration

The fact that the wheels on a car have different positions cause that the wheels have different velocities when turning. It was tested if the residuals would be smaller if the wheel placement was taken into consideration.

Simplified Wheel Geometry

Consider the simplified wheel geometry to the left in Figure 6.1. When turning, the front wheel will have a larger velocity than the wheel in the back e.g. if $\delta_F = \pi/2$ the back wheel will not move at all. Assume a left-turn and consider the simplified wheel geometry to the right in Figure 6.1. The outer wheel (right) will move faster than the inner wheel. The relations between the wheels using this simple approach are derived below.

Back-front relation, see Figure 6.1:

$$\begin{array}{ll}
v_f = \text{front wheel velocity}, & v_r = \text{rear wheel velocity} \\
\omega = \text{angular velocity of the car}, & \delta_F = \text{average front wheel angle} \\
\omega * r_1 = v_f & \\
\omega * r_2 = v_r & \\
r_2 = r_1 * \cos\delta_F & \\
\Rightarrow & \\
v_f * \cos\delta_F = v_r &
\end{array} \tag{6.3}$$

Inside-outside relation, left turn, see Figure 6.1:

$$\begin{array}{ll}
b = \text{distance between left-right wheels}, & r = \text{turning radius} \\
l = \text{distance between front-back wheels}, & v_i = \text{inner wheel velocity} \\
v_o = \text{outer wheel velocity} & \\
\omega * r = v_i & \\
\omega * (r + b) = v_o & \\
r = \frac{l}{\delta_F} \text{ [1, page 340]} & \\
\Rightarrow & \\
v_o * \frac{l}{l + b * \delta_F} = v_i &
\end{array} \tag{6.4}$$

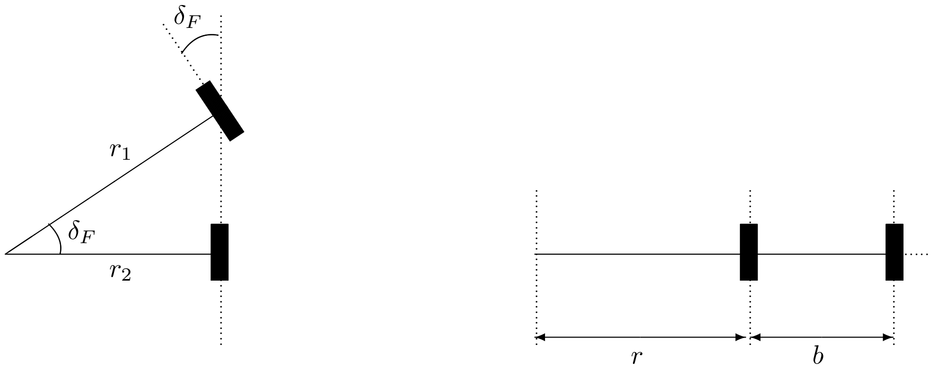

Figure 6.1: Simple wheel geometry: The left part is used to find the back-front relation and the right part is used to find the inside-outside relation. Picture from [8] modified.

Exact Geometry

Here the exact relations between the wheels are derived, the car has Ackermann steering geometry, see Figure 6.2.

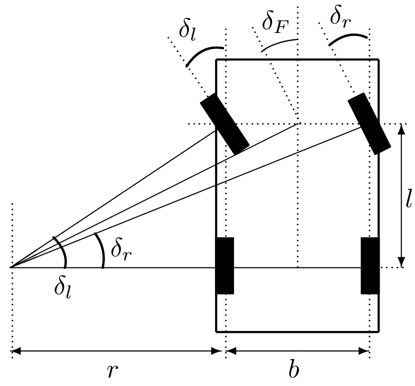

Figure 6.2: Car with Ackermann steering geometry. Picture from [8] slightly modified.

$$
\begin{aligned}
&\tfrac{l}{r} = \tan\delta_l \\
&\tfrac{l}{b+r} = \tan\delta_r \\
&v_{RL} = \omega * r \\
&v_{RR} = \omega * (r+b) \\
&v_{FL} = \omega * \tfrac{r}{\cos\delta_l} = \omega * \tfrac{l}{\sin\delta_l} \\
&v_{FR} = \omega * \tfrac{r+b}{\cos\delta_r} = \omega * \tfrac{l}{\sin\delta_r} \\
&\Rightarrow \\
&\tfrac{v_{RL}}{v_{RR}} = \tfrac{l}{l+b\tan\delta_l} \\
&\tfrac{v_{RL}}{v_{FL}} = \cos\delta_l \\
&\tfrac{v_{RL}}{v_{FR}} = \tfrac{\sin\delta_r}{\tan\delta_l}
\end{aligned}
\tag{6.5}
$$

Note that these relations are only valid in a left turn, when turning right these variables has to be swapped:

$$
\begin{aligned}
&v_{RR} \leftrightarrow v_{RL} \\
&v_{FL} \leftrightarrow v_{FR} \\
&\delta_l \leftrightarrow \delta_r
\end{aligned}
\tag{6.6}
$$

6.2 Results

6.2.1 Simulated Errors

Wheel to Wheel A

Error Type & wheel	detection time/sec
Constant 4, RR	56.24(alarm_status=4)
Constant 6, RL	15.42(alarm_status=3)
Constant 13, RL	6.38(alarm_status=3)
Constant 20, RL	3.51(alarm_status=3)
Constant 50, RL	1.11(alarm_status=3)
Constant 13, RL. Constant 20, RR	3.01(alarm_status=4) and 6.38(alarm_status=13)
Constant 15, RL. Constant 13, RR	6.63(alarm_status=5)

Table 6.2: Method: Wheel to Wheel A, simulated errors.

Error Type & wheel	detection time/sec
Constant 4, RR	RR:30.61
Constant 6, RL	RL:15.42
Constant 13, RL	RL:5.80
Constant 20, RL	RL:2.51
Constant 50, RL	RL:0.84
Constant 13, RL. Constant 20, RR	RR:2.18, FR:5.8, FL:6.38
Constant 15, RL. Constant 13, RR	FL:5.74

Table 6.3: Method: Wheel to Wheel B, simulated errors.

Average

Error Type & wheel	ϵ_1/sec	ϵ_2	ϵ_3	ϵ_4	ϵ_{front_back}	ϵ_{sides}	ϵ_{diag}
Const 1, RR	-	-	-	965.28	-	-	25.39
Const 4, RR	363.57	249.22	209.98	20.52	63.94	628.81	3.05
Const 6, RL	113.57	82.85	14.35	77.96	36.82	69.77	1.95
Const 13, RL	17.63	27.37	4.85	17.63	15.32	5.39	0.98
Const 20, RL	8.98	16.23	2.87	8.98	8.98	2.45	0.71
Const 50, RL	3.07	4.33	1.19	3.07	3.07	0.91	0.38
Const 13, RL. Const 20, RR	4.80	7.58	13.48	3.27	4.80	58.76	1.68
Const 15, RL. Const 13, RR	5.83	9.84	6.44	7.05	5.83	-	8.51

Table 6.4: Method: Average, simulated errors. The grey marked times are the alarms that are used.

6.2.2 Data from Development Vehicle

Scenario	error type	error time/sec and wheel(s)
fr_3_2_muesplit	outlier	FR:4.33-7.44, 7.92-9.01, 11.93-22.29
fr_3_2_muehigh	outlier	FL:2.05-2.77, 3.43-5.55, 6.27-6.45, FR:4.21-7.01
fr_3_3_muehigh	outlier	FL:3.35-5.05, FR:2.45-6.87
fr_3_5_muesplit	outlier	FL:2.64-9.17
fr_3_6	outlier	FL:4.91-6.86, 17.68-19.48, 22.39-28.57
		FR:1.65-6.86, 17.21-28.45
fr_3_6_b	outlier	FL: 15.08-23.48, FR:15.08-23.24
f020_m129_r021_s001_001	sensor error	RR:7.06-22.06
f020_m129_r020_s001_002	sensor error	RL:5.78-20.78

Table 6.5: Exact times for outliers and sensor errors, see Figures 6.3 and 6.4.

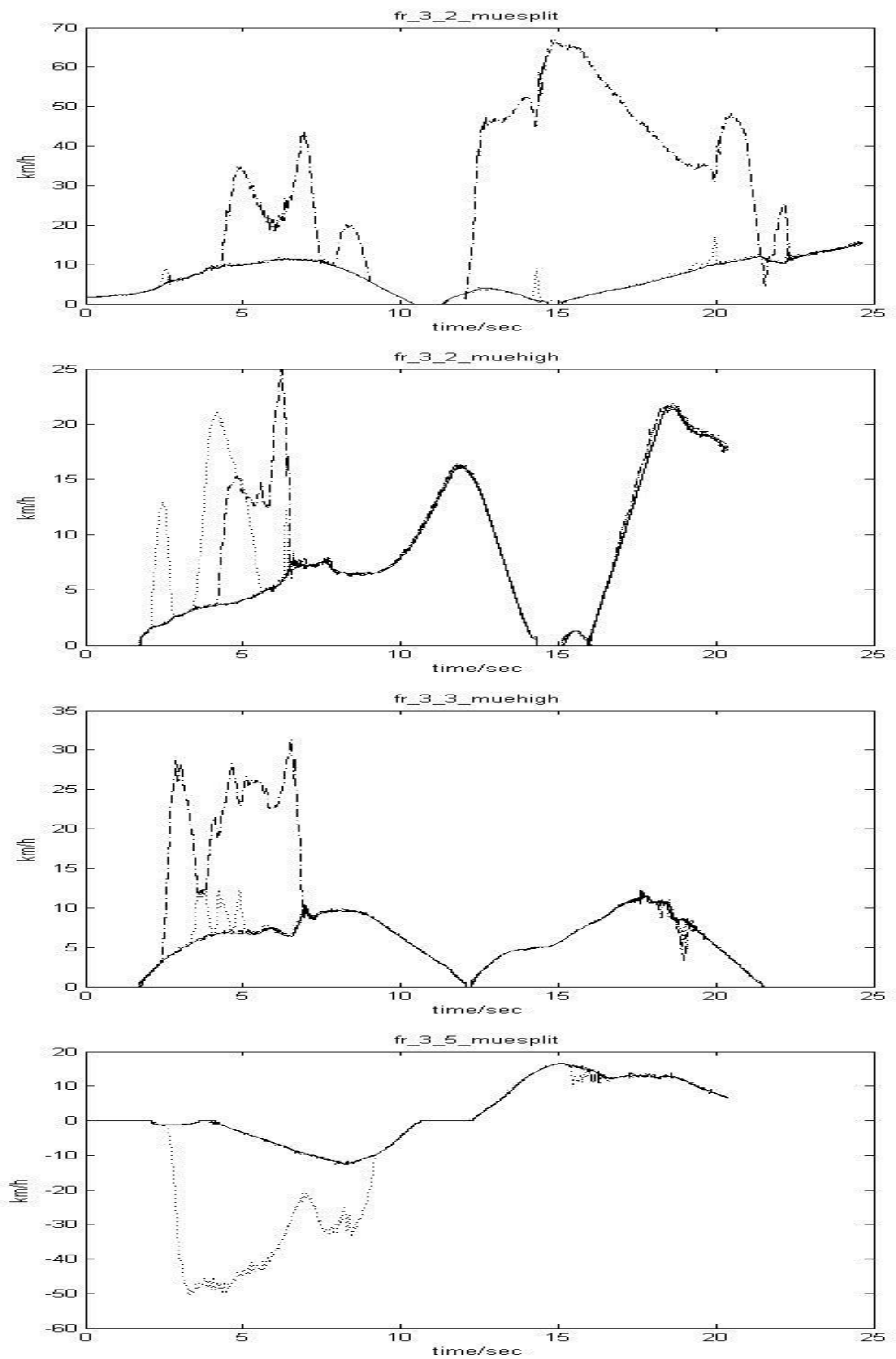

Figure 6.3: Dotted = v_1, dashed dotted = v_2, dashed = v_3, solid = v_4. Top: fr_3_2_muesplit, second: fr_3_2_muehigh, third: fr_3_3_muehigh, bottom: fr_3_5_muesplit. See Table 6.5 for details.

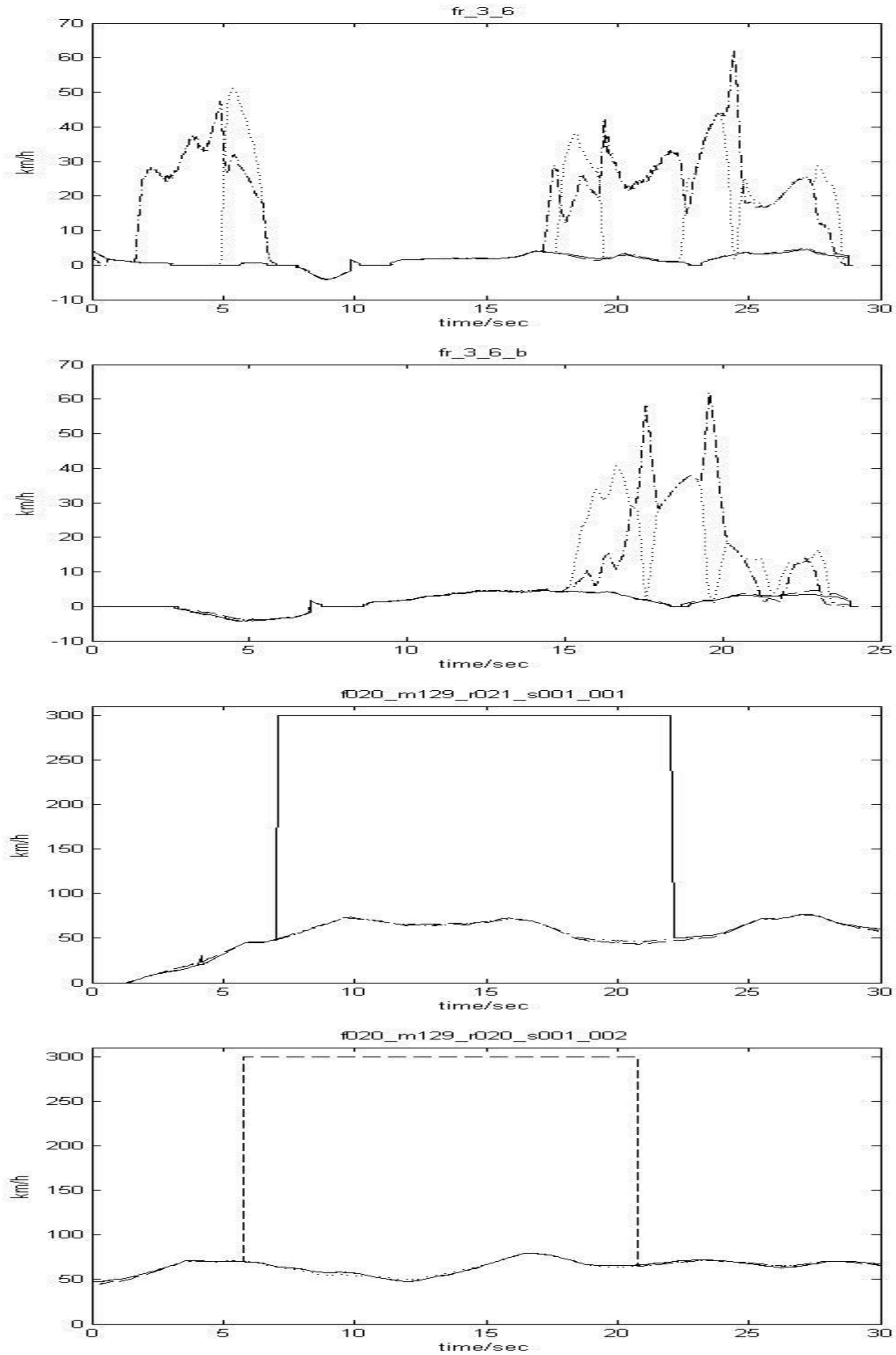

Figure 6.4: Dotted = v_1, dashed dotted = v_2, dashed = v_3, solid = v_4. Top: fr_3_6, second: fr_3_6_b, third: f020_m129_r021_s001_001, bottom: f020_m129_r020_s001_002. See Table 6.5 for details.

Wheel to Wheel

Scenario	detection time/sec, alarm_status (a_s)
fr_3_2_muesplit	13.66 (alarm_status=2)
fr_3_2_muehigh	-
fr_3_3_muehigh	-
fr_3_5_muesplit	3.78 (alarm_status=1)
fr_3_6	3.97:a_s=2, 5.91:a_s=8, 9.29:a_s=2, 16.71:a_s=0
	17.54:a_s=2, 18.94:a_s=8
fr_3_6_b	16.97:alarm_status=1, 18.50:alarm_status=8
f020_m129_r021_s001_001	7.36:alarm_status=4
f020_m129_r020_s001_002	6.13:alarm_status=3

Table 6.6: Method: Wheel to Wheel A. Detection times for errors from development vehicle.

Scenario	detection time/sec
fr_3_2_muesplit	FR:13.22
fr_3_2_muehigh	-
fr_3_3_muehigh	FR:6.93-7.09
fr_3_5_muesplit	FL:3.64
fr_3_6	FR:3.75 FL:5.75 FR:9.78 FL:18.88
fr_3_6_b	FL:16.76
f020_m129_r021_s001_001	RR:7.29
f020_m129_r020_s001_002	RL:6.06

Table 6.7: Method: Wheel to Wheel B. Detection times for errors from development vehicle.

Average

Scenario	detection time/sec	ϵ_{diag}/sec
fr_3_2_muesplit	FR:13.19	5.14
fr_3_2_muehigh	-	4.30
fr_3_3_muehigh	-	3.27
fr_3_5_muesplit	FL:4.07	3.27
fr_3_6	FL:4.21	2.39
fr_3_6_b	FL:17.50	16.13
f020_m129_r021_s001_001	RR:7.44	7.21
f020_m129_r020_s001_002	RL:6.18	5.94

Table 6.8: Method Average. Detection times for errors from development vehicle.

6.2.3 Decreasing the Residuals

The top row show the average size of the residuals when the geometry of the wheel placement is not taken into consideration, the middle when the “simplified geometry” is taken into consideration and the lower row show when the “exact geometry” is applied.

All Scenarios

Residual	ϵ_{12}	ϵ_{13}	ϵ_{14}	ϵ_{23}	ϵ_{24}	ϵ_{34}
No geometry	0.6422	0.1733	0.6974	0.6816	0.1777	0.6215
Simple geometry	0.5924	0.3389	0.7319	0.7422	0.3677	0.5902
Exact geometry	0.7852	0.4342	0.6675	0.7052	0.4586	0.5937

Table 6.9: Average values of the residuals for all scenarios.

Residual	ϵ_1	ϵ_2	ϵ_3	ϵ_4	ϵ_{front_back}	ϵ_{sides}	ϵ_{diag}
No geometry	0.4700	0.4627	0.4476	0.4602	0.1622	0.6301	0.0347
Simple geometry	0.4800	0.4998	0.4916	0.4974	0.3501	0.5887	0.0460
Exact geometry	0.5479	0.5624	0.4939	0.4906	0.3412	0.6447	0.3888

Table 6.10: Average values of the residuals for all scenarios.

Driving in a Circle

Residual	ϵ_{12}	ϵ_{13}	ϵ_{14}	ϵ_{23}	ϵ_{24}	ϵ_{34}
No geometry	2.7121	0.2842	2.9927	2.4280	0.2805	2.7085
Simple geometry	0.2058	0.8225	0.6443	1.0284	0.8502	0.1782
Exact geometry	1.1607	1.2354	0.9323	0.0747	0.2283	0.3030

Table 6.11: Average values of the residuals for one scenario: driving in a circle on asphalt, 30 km/h, radius 10 m, (f020_m051_r001_s001_001).

Residual	ϵ_1	ϵ_2	ϵ_3	ϵ_4	ϵ_{front_back}	ϵ_{sides}	ϵ_{diag}
No geometry	1.9963	1.6199	1.6174	1.9939	0.2824	2.7103	0.0081
Simple geometry	0.4203	0.6948	0.6764	0.4388	0.8364	0.1920	0.0146
Exact geometry	1.1095	0.4381	0.5377	0.1337	0.5035	0.4288	0.7318

Table 6.12: Average values of the residuals for one scenario: driving in a circle on asphalt, 30 km/h, radius 10 m, (f020_m051_r001_s001_001).

6.3 Conclusions

6.3.1 Detecting Erroneous Wheel Velocities

The Average approach (AV) can detect the smallest errors, for errors smaller than 13 km/h it is also the fastest. For larger errors Wheel to Wheel B (WB) is the fastest. Both these approaches can only detect errors on single wheels, when more than one wheel is erroneous they both will report the wheel with the largest error as erroneous. Wheel to Wheel A (WA) can also detect when two wheels are erroneous but is slower than the other two almost always. For small errors the difference is large but decreases the larger the errors get and for large errors WA is faster than AV.

Detecting sensor errors is not a problem for any of the approaches, it is done very fast. In the measurements the sensor errors lead to very large residuals, but if the velocity of the car was very high (close to 300 km/h) the residuals would not be that large leading to longer detection time. Most drivers do not drive at that speed.

How good outliers are detected depends on how large the outliers are and for how long they exist. It has been wished for outliers larger than 13 km/h to be detected in maximum one second. This can not be done with any of the approaches presented here, WB can detect constant errors of size 20 in 2.51 seconds and of size 50 in 0.84 seconds.

When an erroneous wheel velocity is detected the wheel is set to erroneous for a predefined time t, then the detectors will be checked again. One problem with AV is that when a wheel is erroneous and this is detected all residuals will give an alarm after sufficiently long time. Looking at Equations 6.2 shows that this will happen. In Table 6.4 this is also seen. The implemented algorithm checks all the residuals and looks which alarm is given first. This could lead to that when after an erroneous wheel velocity is detected and the detectors are checked again after time t all alarms will set to high. To avoid this the alarms can be reset, this would lead to detection delay. It would be better to combine AV with WB, where WB can be checked after time t, for small errors WB would help in the case written above.

Another interesting property of the AV is the ϵ_{diag} residual/alarm. In all the results it can be seen that this alarm is given very fast and it even detects outliers larger than 13 km/h in less then one second. Unfortunately this alarm does not give any information regarding which wheel is erroneous, just that something is wrong.

To sum up:

- AV: Can detect very small errors, is the fastest for small errors.
- WA: Never the fastest, can detect combinations of errors.
- WB: Detects large errors fastest.

6.3.2 Decreasing the Residuals

In Tables 6.9 and 6.10 it is seen that the residuals for all the scenarios do not decrease when the wheel placement is taken into consideration. After having looked at the data it was discovered that in most of the scenarios the steering wheel movement is very rapid. This gives a very “nonlinear” behavior of the car and to reduce the residuals in this case a very complex model of the car has to be made.

In Tables 6.11 and 6.12 a simple scenario with “easy” steering wheel movement is tested and the residuals are at most reduced to as little 6.6 percent of the original value. In normal driving scenarios this could be used to significantly decrease the detection times and the error size that can be detected. To get the same result in extreme maneuvers such as many in the measurement data used, as written above a complex car model can be used.

Bibliography

[1] J. Y. Wong. *Theory of Ground Vehicles.* ISBN 0-471-35461-9, John Wiley & Sons Ltd, 2001.

[2] W. Klier, G. Reimann and W.Reinelt. *Concept and Functionality of the Active Front Steering System.* SAE Paper 2004-21-0073, 2004.

[3] M. Basseville, and I. V. Nikiforov. *Detection of Abrupt Changes: Theory and Applications.* Prentice Hall, Englewood Cliffs, NJ, 1993.

[4] F. Gustafsson. *Adaptive Filtering and Change Detection.* ISBN 0-471-49287-6, John Wiley & Sons Ltd, 2000.

[5] http://www.phil.gu.se/ann/annintr.html, last visited 2005.10.01. Helge Malmgren *Artificiella Neurala Nätverk - en kort introduktion.*

[6] http://www.library.cornell.edu/nr/bookcpdf/c18-1.pdf, last visited 2005.08.10. Sample page from *Numerical recipes in c: the art of scientific computing* (ISBN 0-521-43108-5) Copyright (C) 1988-1992 by Cambridge University Press.

[7] http://photoinjectors.net/article_3.htm, last visited 2005.08.10. Neumann series solution to Fredholm integral equations of the second kind.

[8] O. Enqvist. Master's thesis, Automatic Control and Communication Systems, Dept. of Electrical Engineering at Linköpings University, LITH-ISY-EX-06/3752-SE, *AFS-Assisted Trailer Reversing*, 2006.