# DYNAMIC SIGNATURE VERIFICATION SYSTEM

**LIM WAI LOON**

**A project report submitted in partial fulfilment of the requirements for the award of the degree of Bachelor (Hons) of Mechatronics Engineering**

**Faculty of Engineering and Science**
**Universiti Tunku Abdul Rahman**

**May 2011**

## DECLARATION

I hereby declare that this project report is based on my original work except for citations and quotations which have been duly acknowledged. I also declare that it has not been previously and concurrently submitted for any other degree or award at UTAR or other institutions.

Signature : _____

Name : _____

ID No. : _____

Date : _____

## APPROVAL FOR SUBMISSION

I certify that this project report entitled **"DYNAMIC SIGNATURE VERIFICATION SYSTEM"** was prepared by **LIM WAI LOON** has met the required standard for submission in partial fulfilment of the requirements for the award of Bachelor of Engineering (Hons.) Mechatronics Engineering at Universiti Tunku Abdul Rahman.

Approved by,

Signature : _____

Supervisor : Mr. Chai Tong Yuen

Date : _____

# DYNAMIC SIGNATURE VERIFICATION SYSTEM

## ABSTRACT

This report has introduced the development of signature verification system by using the dynamic parameters such as pen pressure, velocity and the position of the signature. The input for the proposed solution comes from the SUSig Online Database. The main feature of the proposed system is to read, analyse and verify the mentioned database. Firstly, the proposed system will get the input signature from the database. The system database contains 25 genuine signatures and 25 forgery signatures. Next, the training signatures and testing signature from the designed database will proceed to Pre-processing stage. After that, the processed signatures will be saved as reference signature while the testing signature will be saved as sample signature. Then, both signatures will proceed to the verification stage. In verification stage, difference between reference and testing signatures will be calculated. Next, the standard deviation can be obtained. The calculated standard deviation value for each parameter is then compared with the system's threshold value. If the conditions of the parameter suites the threshold, the signature will be accepted as genuine signature. Else, the signature will be rejected as forgery signature. This proposed system has 14.8% of False Rejection Rate and 2.64% of False Acceptance Rate. Meanwhile, it has the error rate as 2.89% which mean it also has an approximately 97% of classification rate.

# TABLE OF CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

# LIST OF APPENDICES

# CHAPTER 1

# INTRODUCTION

## 1.1 Background

Signatures are composed of special character and flourishes and therefore most of the time they can be unreadable. Also, intrapersonal variations and interpersonal differences make it necessary to analyze them as complete images but not as letters and words put together. As signatures are the primary mechanism both for authentication and authorization in legal transactions, the need for research in efficient automated solutions for signature verification has increased in recent years.

Based on different application, signature verification system can be operated in two different modes; online and offline mode. In the online mode, the signature verification is dealing with online inputs. Meaning that, instant result will be given by the system once the signer signs a signature. In real world, online signature verification system is used in the credit card verifier. For offline mode, the opposite situation will happen. The offline signature verification system does not dealing with online inputs. It can be done as long as the signature is recorded. For example, offline signature verification is applied in the document verification in the bank.

Generally, signature verification system can be categorised into two types; dynamic and static. The dynamic signature verification system is dealing with signal processing while the offline signature verification system is dealing with image processing. Normally, dynamic signature verification system will be operated in

online mode while static signature verification system will be operated in offline mode.

Dynamics signature verification system has a common module of an online verification system. Firstly, the signer will sign the input signature by using electronic pen-tablet or other digitizing devices. Here, the pen-trajectory data will be traced by the system and form the so called velocity, acceleration, pressure, altitude, and azimuth parameters in time-domain. Next, signal processing technique will be applied on the parameters. In result, some of the features will be extracted from the data. Then, the system will perform matching techniques on the extracted features. In this process, the system will perform comparison between the input signatures with the reference signature that already stored in the database. And lastly, verification of the signature will be performed.

Compared to dynamic signature verification system, static signature verification system usually performed in the offline mode. Basically, the processing steps of static signature verification system can be divided into 4 parts; acquisition of the input (image of signature), pre-processing, feature extraction, and classification. For acquisition of the input, a 2-Dimensional image of signature can be obtained from the scanner or camera. Next, pre-processing steps will be performed on the image. The system will perform image processing techniques on the input image. Then, the system will extract some of the feature from the image and proceed to the next step. In the final step, classification of the image will be performed. Here, the processed image will be compared to the reference image. In result, the matching of the signature will verify truthiness of the signature.

The reason why the signature verification system exists is to verify whether the signature is a genuine or forgery signature. Basically, forgery signature can be classified into three types; random, simple and skilled forgeries. Random forgery signature is signed by different signer and has different shape of signature compared to the genuine signature. Compared to random forgery signature, simple forgery signature has the same shape with the genuine signature. The last type is the skilled forgery, who had been trained to sign as alike as possible as the genuine signature.

In signature verification, there are two types of error; type-I error and type-II error. Type-I error is where the sample signature is a forgery signature, but the signature verification system accepts the signature and verify it as genuine signature. Thus, type-I error is known as False Acceptance Rate (FAR). Type-II error is known as False Rejection Rate (FRR). Type-II error is totally different with the type-I error. It is where the sample signature is a genuine signature, but the signature verification system rejects and declares it as forgery signature.

## 1.2 Aims and Objectives

The main aim and objective of this project is to construct an accurate and efficient signature verification system. To achieve that, study in dynamic approach has been done. Next, an algorithm to solve the signature verification will be produced and analysed. Consideration and analysis need to be done on FAR and FRR. Thus, an accurate and efficient signature verification system can be designed.

# CHAPTER 2

# LITERATURE REVIEW

## 2.1 Literature Review

According to Oscar et al. (2007, 2008), Dynamic Time Warping and Gaussian Mixture Modelling methods have been introduced for automated signature verification system. Dynamic Time Warping is used to remove intrinsic variability from users signatures by aligning the signals acquired from the digital tablet with the user's reference signature. Gaussian Mixture Modelling has been used to model the probabilistic distribution of the set of pseudo-distances and to calculate the likelihood ratio between the sample and the reference signature.

In Quen-Zong et al. (1997) research, an on-line signature verification scheme based on Linear Prediction Coding (LPC) cestrum and neural networks is proposed. Cepstral coefficients derived from linear predictor coefficients of the writing trajectories are calculated as the features of the signatures. These coefficients are used as inputs to the neural networks. A number of single-output multilayer perceptions (MLP's), as many as the number of words in the signature, are equipped for each registered person to verify the input signature. If the summation of output values of all MLP's is larger than verification threshold, the input signature is regarded as a genuine signature; otherwise, the input signature is a forgery.

In Charles E. Pippin (2004) work, the signature verification is using separate filters with different approaches. In the first, global features of the signature, such as average velocity are considered using a Euclidian distance. In the second filter, local

features are considered. Strokes are segmented using the minima of the velocity and encoded before comparing them using dynamic time warping and signer-specific thresholds.

In Juan D. Penpgos et al. (2008) paper, the dynamic signature verification system analyzes signatures dynamically by considering their shape, time domain characteristics, such as speed and acceleration, and force domain characteristics, i.e. applied pressure. Then it compares these parameters with those of previously obtained master signatures. The results are converted into a percentage match figure to determine whether the signature is qualified as authentic or forgery. Experimental results show 92% authentic signature detection accuracy and 100% forgery signature detection accuracy. This high level of accuracy plus low computation requirements for analysis, make this system a commercially viable solution to the signature identification problem.

R. Seiler et al. (1996) explains the major difference between online and offline handwriting recognition system. While on-line recognition is based on pen trajectory data, off-line recognition has to rely on pixel data only. A comparison between an off-line and an on-line recognition system using the same databases and system design had been presented. Both systems use a sliding window technique which avoids any segmentation before recognition. The recognizer is a hybrid system containing a neural network and a hidden Markov model. New normalization and feature extraction techniques for the off-line recognition are presented, including a connectionist approach for non-linear core height estimation. Results for uppercase, cursive and mixed case word recognition are reported. Finally a system combining the on- and off-line recognition is presented.

In Musa Mailah (2008) paper, it describes the development of a handwritten signature verification system incorporating pen pressure of signature path, time duration of the signing procedure, velocity profile of signature and position of signature shape. The handwritten signals have been captured and digitized using a tablet. The main features of the proposed signature verification system are the dynamically update of handwritten signature, retries capability in verification, application of tolerance bands and threshold values, development of user friendly

Graphic User Interface, application of Common Time Axes and verification of signatures using a class of a multilayer feed-forward neural network. A novel algorithm has been applied that provides the ability to produce consistent and high accuracy verification result and maintain the speed of verification. The system has yielded 1.33% of False Reject Rate and 0% False Acceptation Rate with the verification using random forgery signatures.

In Fernando.Alonso et.al. (2009) paper, fusion of static image and dynamic information for signature verification has been introduced. This paper evaluates the combination of static image (off-line) and dynamic information (on-line) for signature verification. Two off-line and two on-line recognition approaches exploiting information at the global and local levels are used. Experimental results are given using the BiosecurID database (130 signers, 3,640 signatures). Fusion experiments are done using a trained fusion approach based on linear logistic regression. It is shown experimentally that the local systems outperform the global ones, both in the on-line and in the off-line case. We also observe a considerable improvement when combining the two on-line systems, which is not the case with the off-line systems. The best performance is obtained when fusing all the systems together, which is especially evident for skilled forgeries when enough training data is available.

# CHAPTER 3

# METHODOLOGY

## 3.1 System Overview

This dynamic signature verification system is started with the input of the User ID. The user is required to key in his/her ID to the system. Here, the system will decide whether the user ID is registered or not. If the user ID is not registered, the user has to drop down 5 sets of signature for training purpose. If the user ID is registered, the user will then be asked to sign for testing. Then, the input training signatures and the input testing signature will proceed to the Pre-processing stage. After this stage, the input training signature will be saved as reference signature in the database while the input testing signature will be saved as sample signature and proceed to Verification stage. In the Verification stage, the sample signature will be compared with the reference signature which stored in the database. If the difference between two signatures does not exceed the Threshold value, the sample signature will be accepted as genuine signature and via versa. The Figure 3.1 shows the process flow of the proposed dynamic signature verification system.

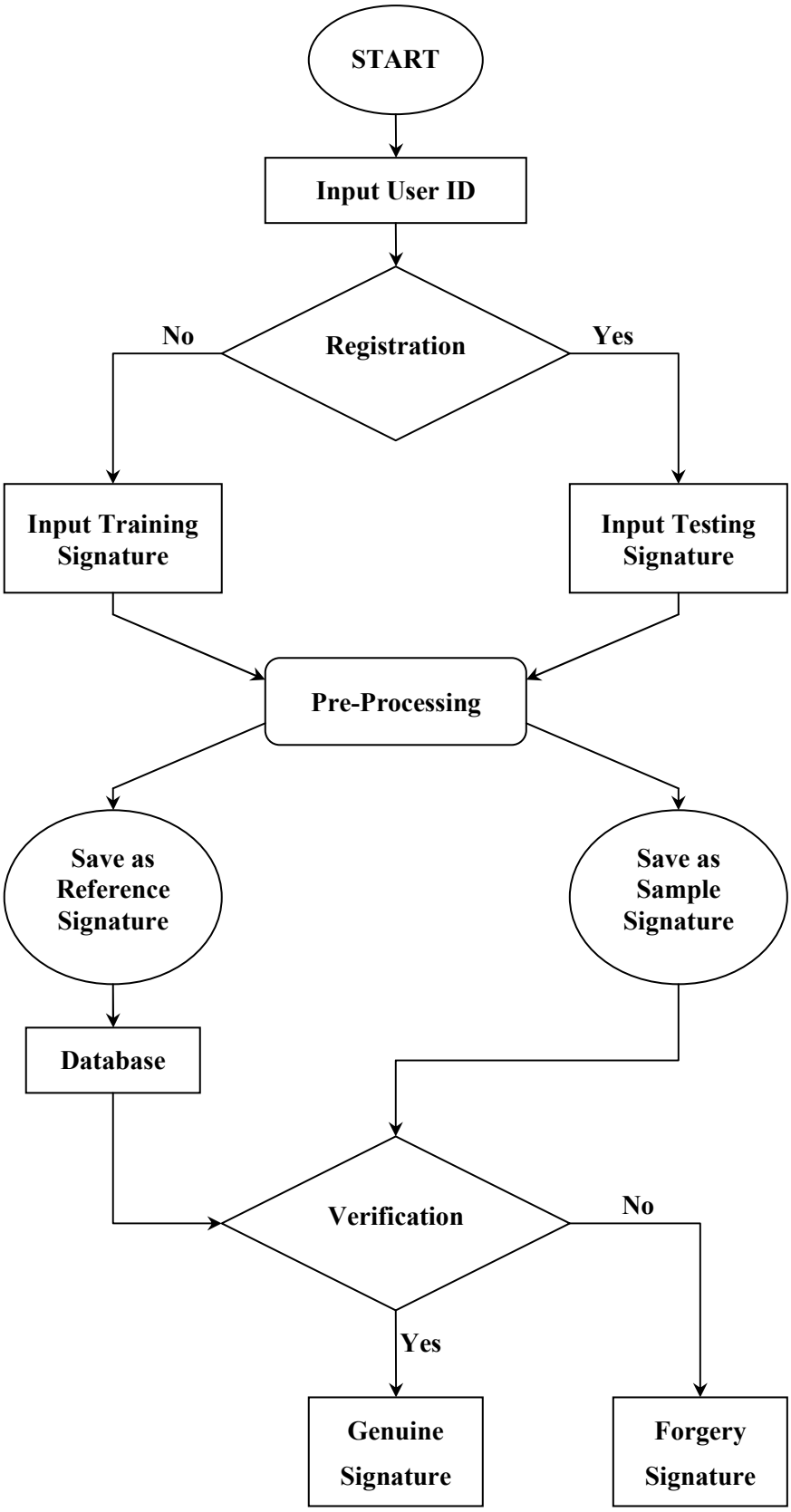


**Figure 3.1: Process Flow Chart**

## 3.2 Input Module

This is the first process of the dynamic signature verification system. The input signature is captured by using devices such as pen tablet, specialized signature pads or camera. Dynamic features in time domain such as positions, pressure and tilts will be acquired.

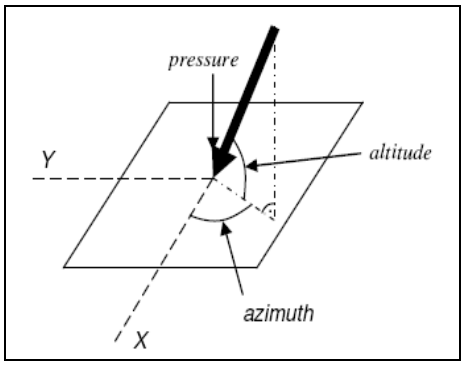


**Figure 3.2: Acquired Dynamic Features**

As shown in the Figure 3.2, the Cartesian coordinates determine the position of the traced signature. The pressure parameter describes the pen pressure inflicted on the tablet surface. The altitude is the angle between the pen and the tablet surface. The azimuth denotes the clockwise rotation of the pen around the x-axis.

Next, these parameters will be transformed into mathematic model by the ADC and DAC module as below:

(a) X-coordinate vs time

$$X = \{x_1, x_2, x_3, \ldots x_n\},\ x_i = x(t_i),$$
$$i = 1, 2, 3, \ldots n$$

(b) Y-coordinate vs time

$$Y = \{y_1, y_2, y_3, \ldots y_n\},\ y_i = y(t_i),$$
$$i = 1, 2, 3, \ldots n$$

(c) Pressure vs time

$$P = \{p_1, p_2, p_3, \ldots p_n\},\ p_i = p(t_i),$$
$$i = 1, 2, 3, \ldots n$$

(d) Altitude vs time

$L = \{l_1, l_2, l_3, \ldots l_n\}$, $l_i = l(t_i)$,

i = 1, 2, 3,…*n*

(e) Azimuth vs time

$A = \{a_1, a_2, a_3, \ldots a_n\}$, $a_i = a(t_i)$,

i = 1, 2, 3,…*n*

For simulation purpose, the SUSig Online Database will be used. In the database, there are recorded data such as x-coordinates, y-coordinates, time stamp, pressure level and a pen up or down indicator. For the proposed dynamic signature verification system, 4 dynamic parameters which are x-coordinate, y-coordinate, velocity and pressure will be tested. As shown in the Figure 3.3, the x-coordinates, y-coordinates and pressure signals in the SUSig Online Database will be used.

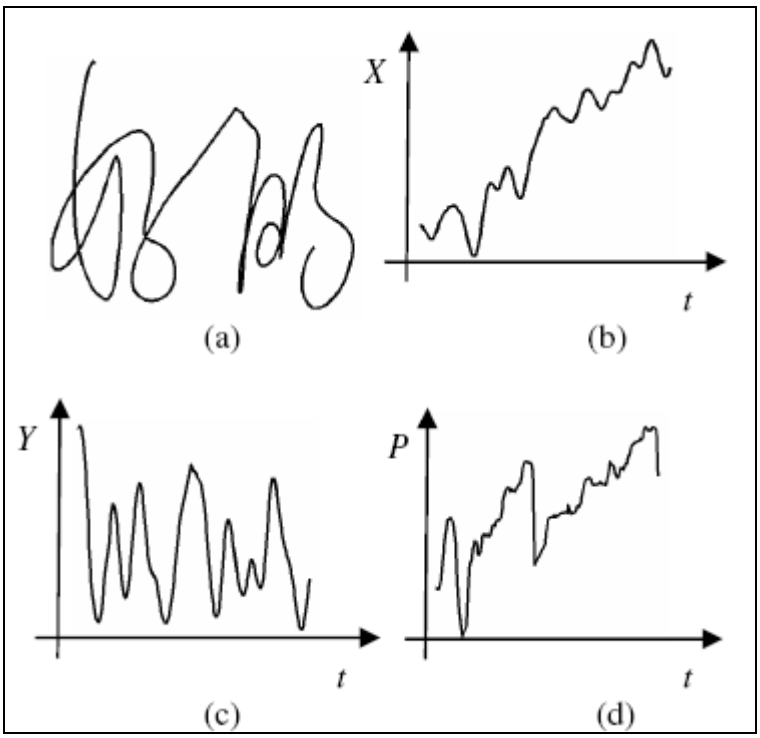


**Figure 3.3: Signal of the Acquired Dynamic Features**
**(a) Signature Trajectory, (b) X-t Graph,**
**(c) Y-t Graph, (d) P-t Graph**

### 3.2.1 The SUSig Online Database

The SUSig online signature database had been downloaded from Sabanci University Online Signature Database (http://biometrics.sabanciuniv.edu)

The SUSig database consists of two parts, namely Visual and Blind sub-corpora. Visual sub-corpus was collected using Interlink Elec. ePad Ink signature tablet with built-in LCD screen. For each subject there are 20 genuine and 10 forgery signatures. Genuine signatures were collected in two different sessions.

Blind sub-corpus was collected using Wacom Graphire2 pressure sensitive tablet. For each subject there are 10 genuine and 10 forgery signatures. Genuine signatures were collected in a single session.

The signature database file consists of x- and y-coordinates, time stamp, pressure level and a pen up or down indicator.

### 3.2.2 Input Signatures

For this project, all the input signatures are using the SUSig Online database. The SUSig Online Database's file originally is stored in .SIG format and it needs specified software to decode it. In order to develop the SUSig database into the designed database, WordPad will be used. WordPad is used to open the .SIG file and 'save as' the file in .txt format which we can read through Matlab. So, another database named system database is created for further analysis.

The system database contains 25 genuine signers and 25 forgery signers from the SUSig Online database. And, each signer will produce 10 samples of signature. Thus, there are 500 signatures totally in the system database. For each signature, 3 dynamic features such as x-coordinate, y-coordinate, and pressure have been used.

To get the reference signature, the first 5 samples will be used as training signatures. As for the testing signature, any sample from any signer can be used. In example,

*Sample 1 (S1): x1, y1, p1*
*Sample 2 (S2): x2, y2, p2*
*Sample 3 (S3): x3, y3, p3* } Samples for Reference Signature
*Sample 4 (S4): x4, y4, p4*
*Sample 5 (S5): x5, y5, p5*

*Sample T (S6): xT, yT, pT* Sample for Testing Signature

### 3.3 Pre-Processing

During the pre-processing step, the input signature will undergo normalisation, sampling and smoothing. Before this stage, 3 dynamic parameters are collected from the database which is pressure, x-coordinate and y-coordinate. However, there is one more parameter which is velocity has not be found yet. To obtain the velocity for each sample, differentiation is used.

$$Vx1 = \frac{dx1}{dt} \qquad Vy1 = \frac{dy1}{dt} \qquad \boxed{v1 = \sqrt{Vx1^2 + Vy1^2}}$$

$$Vx2 = \frac{dx2}{dt} \qquad Vy2 = \frac{dy2}{dt} \qquad \boxed{v2 = \sqrt{Vx2^2 + Vy2^2}}$$

$$Vx3 = \frac{dx3}{dt} \qquad Vy3 = \frac{dy3}{dt} \qquad \boxed{v3 = \sqrt{Vx3^2 + Vy3^2}}$$

$$Vx4 = \frac{dx4}{dt} \qquad Vy4 = \frac{dy4}{dt} \qquad \boxed{v4 = \sqrt{Vx4^2 + Vy4^2}}$$

$$Vx5 = \frac{dx5}{dt} \qquad Vy5 = \frac{dy5}{dt} \qquad \boxed{v5 = \sqrt{Vx5^2 + Vy5^2}}$$

$$VxT = \frac{dxT}{dt} \qquad VyT = \frac{dyT}{dt} \qquad \boxed{vT = \sqrt{VxT^2 + VyT^2}}$$

### 3.3.1 Normalisation

The main purpose of normalization is to scale all the values into the range of (0, 1). Linear scaling is used for the normalization of the vector (S) which represents signature parameters such as x-coordinate (x), y-coordinate (y), pressure (p) and velocity (v).

$$S = \frac{S - \min(S)}{\max(S) - \min(S)} \tag{3.1}$$

The Maximum (*max*) and minimum (*min*) values in the vector S are the global maximum and minimum points for the signal. Thus, the maximum value for the normalised signal will be 1 and the minimum value will be 0.

### 3.3.2 Re-Sampling

The main reason of this process is to resample the wavelength of the signature into the desired wavelength so that further action can be made. The wavelength of the signature is directly affected by the number of recorded data during the signing process. The more data is recorded, the longer the wavelength is. Each signature will have different signal or wavelength. In order to make them into a same wavelength, resample function in Matlab will be used.

$$y = resample(data, p, q) \tag{3.2}$$

As shown above, '*data*' is the target time-domain signal to be resampled while '*p*' and '*q*' represents a resampling factor. For the reference signature, the average wavelength of 5 training signatures is calculated.
Let $I_{ref}$ = Wavelength of the Reference Signature,

$$I_{ref} = \frac{I_1 + I_2 + I_3 + I_4 + I_5}{5} \tag{3.3}$$

As for the testing signature, the signature will undergo resample process so that the wavelength of the testing signature will be the same as reference signature at the end. The resulted signature is now called sample signature.

Let $I_T$ = Wavelength of the Testing Signature,

$$I_{T(final)} = \frac{I_{T(initial)} \times I_{ref}}{I_{T(initial)}} \tag{3.4}$$

### 3.3.3 Reference Signature

After the resample process, all 5 training signatures will have the same wavelength. Thus, the average value of these 5 signals can be calculated and resulted as the reference signature.

Let $S_{ref}$ = Reference Signature

$$S_{ref} = \frac{S_1 + S_2 + S_3 + S_4 + S_5}{5} \tag{3.5}$$

## 3.4 Verification

In the verification process, the testing signature will be compared with the reference signature. The difference between the reference signature and testing signature is calculated. Thus, standard deviation can be found and compared to the threshold value. If the comparison value is lower than the threshold value, the signature will be accepted as genuine signature. While the value is higher than the threshold value, the signature will be rejected as forgery signature.

### 3.4.1 Standard Deviation

In the first step of the Verification process, difference between the reference signature and testing signature will be calculated. Next, the Standard Deviation between these 2 signatures will be obtained.
Let $d$ = Difference between Reference and Testing Signatures

$$d = \left|(S_T - S_{ref})\right| \tag{3.6}$$

$$d^2 = \left|(S_T - S_{ref})\right|^2 \tag{3.7}$$

Let $SD$ = Standard Deviation of Reference and Testing Signatures

$$SD = \sqrt{\frac{d^2}{I_{ref}}} \tag{3.8}$$

### 3.4.2 Threshold Estimation

Generally, signature verification system has two strict requirements. Firstly, the system is able to avoid the acceptance of the forgery signature and secondly the system is able to avoid the rejection of the genuine signature. In other word, an ideal signature verification system should have as low as possible of its FRR and FAR.

With the achievement of these two requirements, then the system can start to operate in automated way. However, when the system is over sensitive to the signature, the FRR will be very low but the FAR also will become very high. This means that, a genuine signature will be rejected also because it may have a slightly differences with the model signature. Therefore a system lowest average value of FRR and FAR is recommended as Figure 3.4. Thus, threshold value can be chosen ideally.

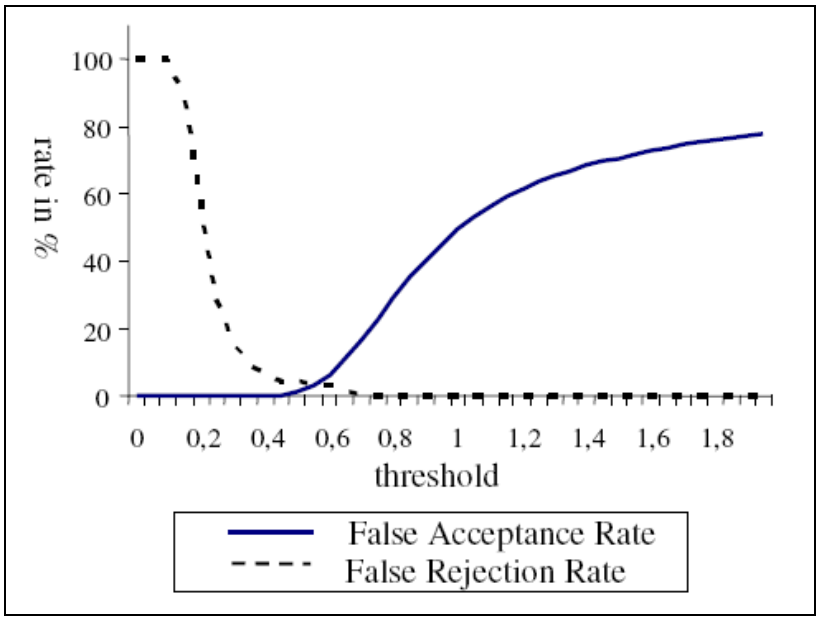


**Figure 3.4: False Acceptance Rate & False Rejection Rate**

# CHAPTER 4

# RESULTS AND DISCUSSIONS

## 4.1 Introduction

For this project, there are 2 software were used. Firstly, it's WordPad. WordPad was used to read the SUSig Online Database .SIG file. Then, the opened file was saved in .txt format. Another used program in this project was Matlab R2009a. Matlab was used to run the coding and generate the result for this project. Note that, some of the Matlab R2009a does not have the function of "*resample*". Thus, make sure the version of Matlab R2009a is updated. There is only x-coordinate's result be shown in this chapter. The same procedure is done on the other three parameters; y-coordinate, velocity, and pressure.

## 4.2 Input Signature

Firstly, the .SIG file was converted into .txt format so that Matlab could read from it. Secondly, 5 genuine signatures database files were moved into the Matlab Folder where the coding program stored. Next, by using "*textread*" function, the data in the .txt file had been successfully written in Matlab. Figure 4.1 shows the input of the training signatures.

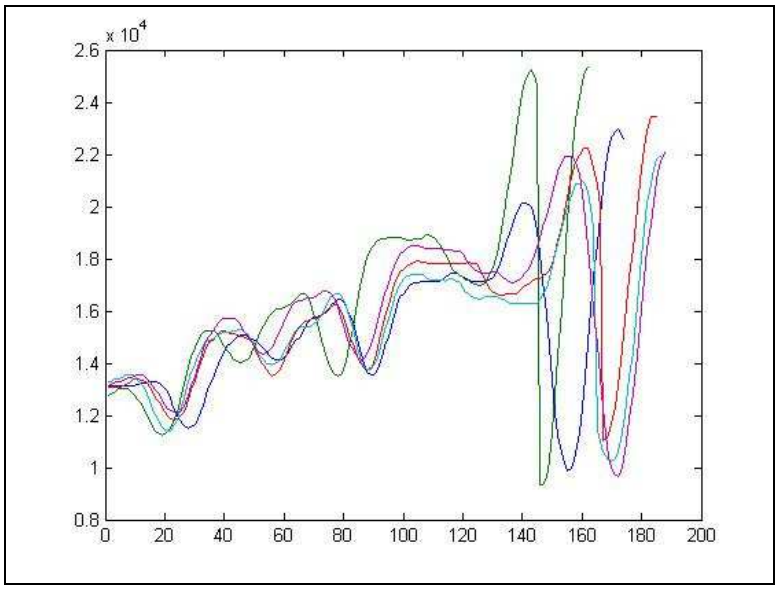


**Figure 4.1: Training Signatures**

## 4.3 Pre-processing

### 4.3.1 Normalisation

In this process, the magnitude of the signals was normalised to (0,1) as shown in the Figure 4.2.

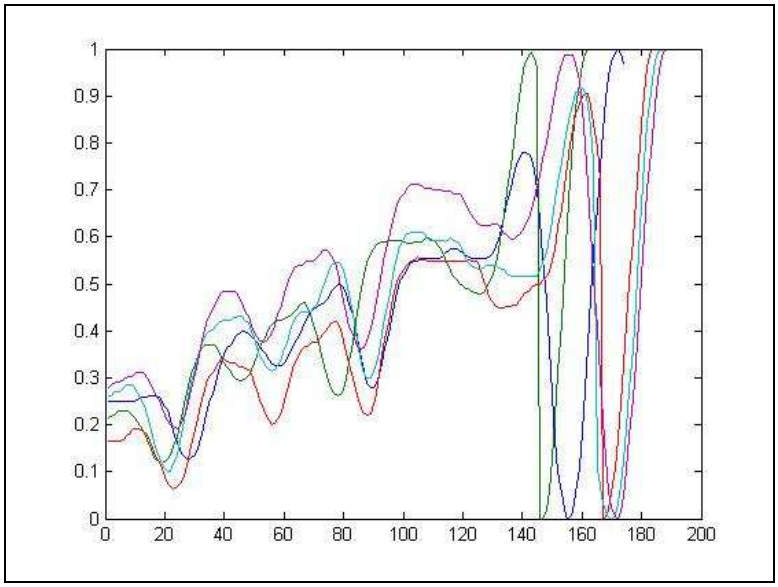


**Figure 4.2: Training Signatures after Normalisation**

### 4.3.2 Re-Sampling

In this process, each signal was resampled into the same wavelength as shown in the Figure 4.3.

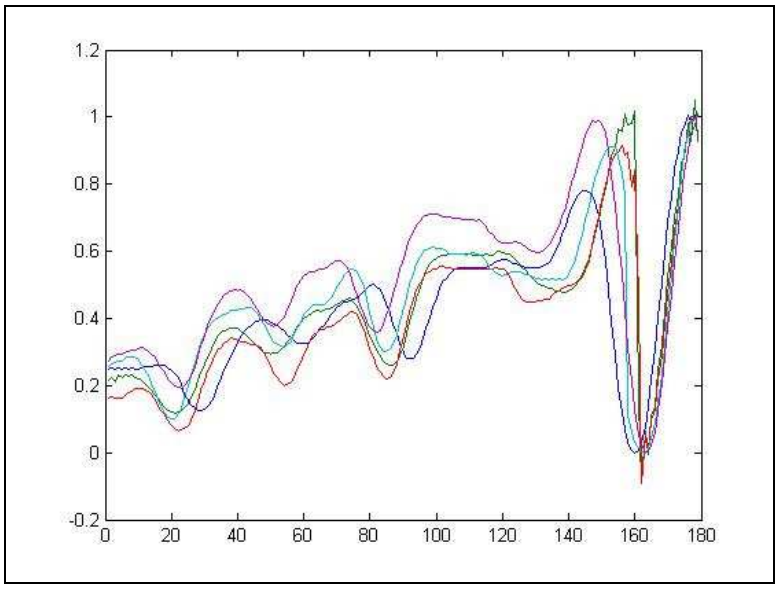


**Figure 4.3: Training Signature after Resample**

From the Figure 4.3, we found that there are some noise occur and focus on the global maximum and minimum points of the signals. Thus, smoothing function had been applied on the signals. The result is shown as Figure 4.4.

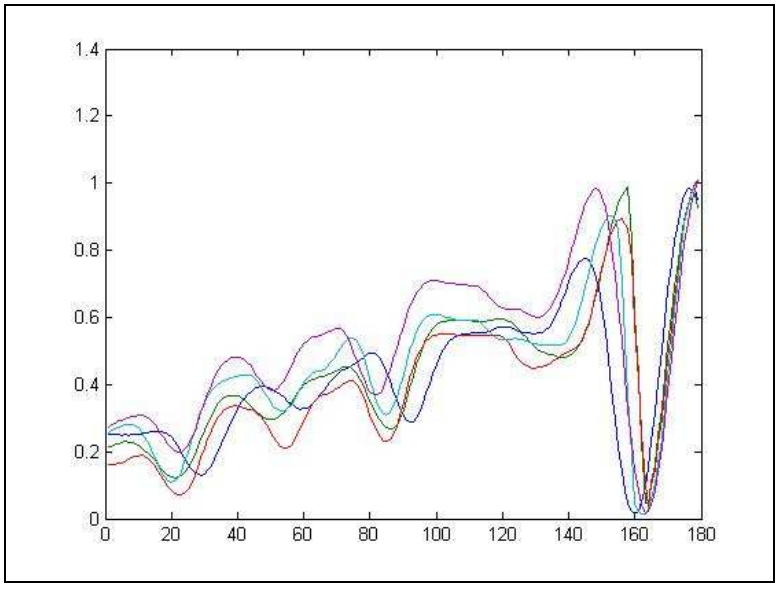


**Figure 4.4: Training Signature after Smoothing & Resample**

### 4.3.3 Reference Signature

After resample and smoothing, the processed signals had the same wavelength. Thus, the reference signature had been calculated and shown in Figure 4.5.

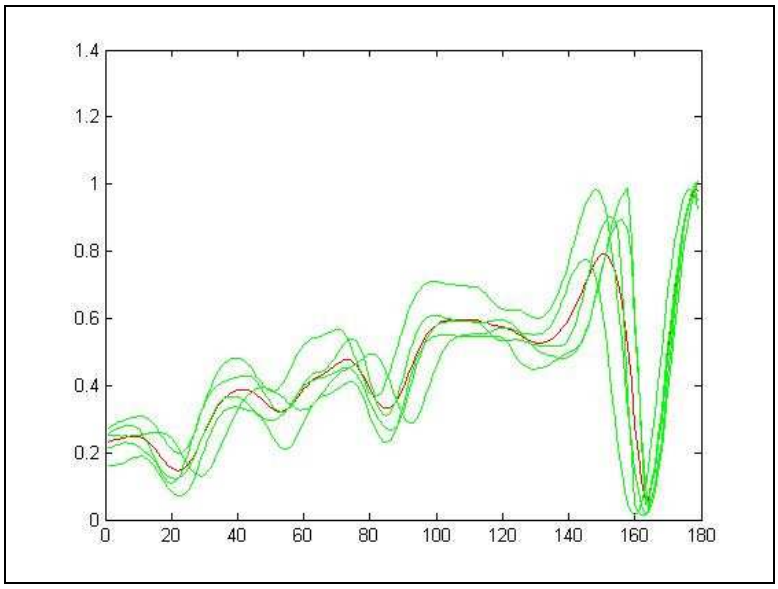


**Figure 4.5: Reference Signature & Training Signatures**

In Figure 4.5, the red colour signal represents the reference signature while the green colour signals represent the training signatures. In order to determine whether the reference signal is trustable, Figure 4.6 was plotted.

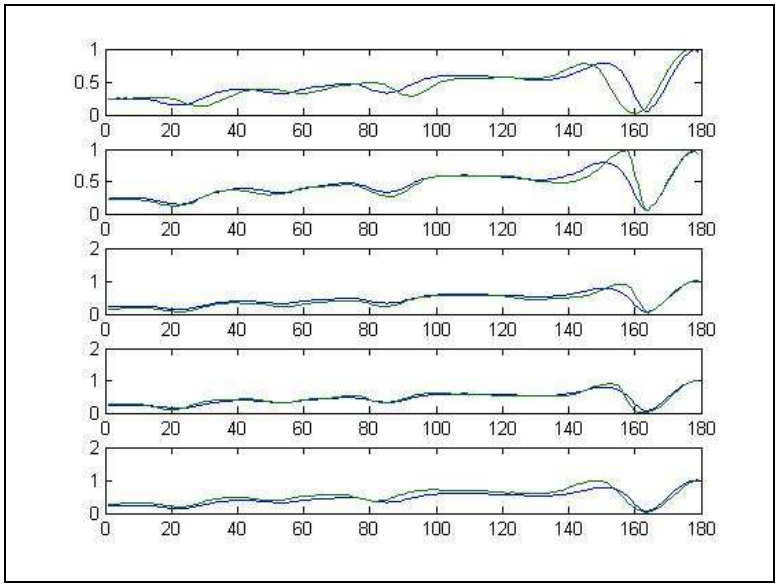


**Figure 4.6: Comparison between Reference Signal & each Training Signal**

From the Figure 4.6, we found that the reference signature was almost the same as each training signature. Thus, the average value of these 5 training signatures was accepted as reference signature.

## 4.4 Verification

After the reference signature done its pre-processing stage, the system is going into verification stage. Firstly, the input of the testing signature had been obtained. Next, it will be processed in Pre-processing stage as the reference signals done and saved as sample signature. After that, it was compared to the reference signature as shown in the Figure 4.7.

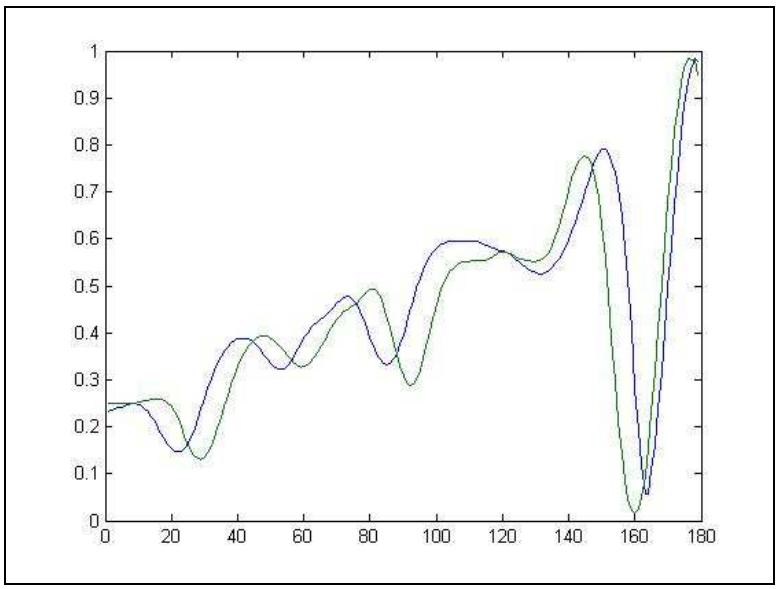


**Figure 4.7: Comparison between Reference Signature & Sample Signature**

From the Figure 4.7, the blue colour signal represents the reference signature while the green colour signal represents the sample signature. Next, the difference between these 2 signatures can be calculated. Figure 4.8 shows the magnitude difference between the reference signature and the sample signature. Since the magnitude difference was obtained, the value of standard deviation was then be calculated.

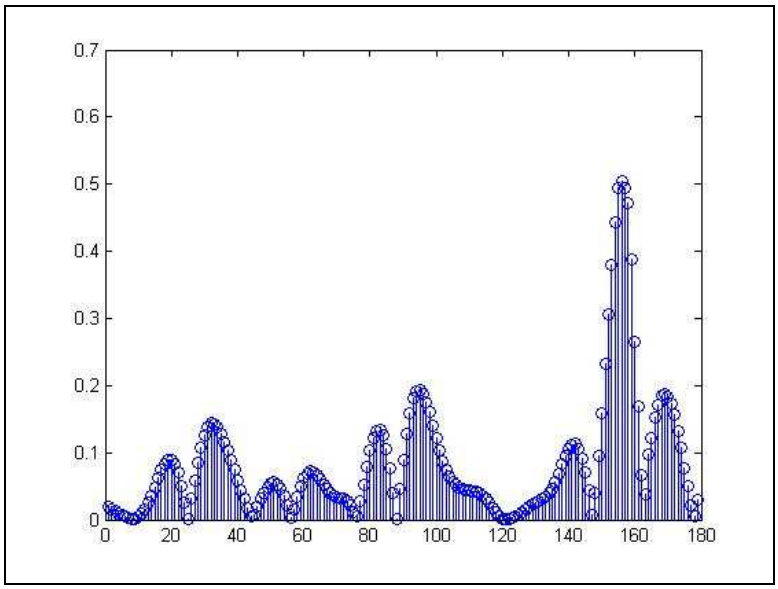


**Figure 4.8: Magnitude Difference between Reference Signature & Sample Signature**

## 4.5 Threshold Estimation

In order to estimate the threshold for each parameter; x-coordinate, y-coordinate, pressure and velocity, False Acceptance Rate (FAR) and False Rejection Rate (FRR) had been calculated by running the coding to the whole database.

Table 4.1 shows all the standard deviation values for the genuine signature. These values were obtained by comparing the genuine testing signature with the genuine reference signature. Each row represents the signature number (signer). Each column represents the sample number of the signature. For example, the $4^{th}$ column and $3^{rd}$ row number represents the $4^{th}$ sample of the $3^{rd}$ signer's signature compared with his own reference signature.

**Table 4.1: Standard Deviation (Genuine Signature)**

| | | SAMPLE | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | **1st** | **2nd** | **3rd** | **4th** | **5th** | **6th** | **7th** | **8th** | **9th** | **10th** |
| **SIGNER** | **1** | 0.126 | 0.084 | 0.089 | 0.053 | 0.106 | 0.070 | 0.076 | 0.094 | 0.192 | 0.077 |
| | **2** | 0.058 | 0.080 | 0.053 | 0.056 | 0.057 | 0.050 | 0.070 | 0.046 | 0.062 | 0.055 |
| | **3** | 0.058 | 0.059 | 0.052 | 0.030 | 0.065 | 0.075 | 0.083 | 0.059 | 0.043 | 0.043 |
| | **4** | 0.123 | 0.095 | 0.088 | 0.054 | 0.095 | 0.079 | 0.084 | 0.195 | 0.150 | 0.146 |
| | **5** | 0.131 | 0.083 | 0.060 | 0.079 | 0.077 | 0.076 | 0.080 | 0.117 | 0.068 | 0.136 |
| | **6** | 0.029 | 0.032 | 0.045 | 0.032 | 0.040 | 0.068 | 0.062 | 0.044 | 0.067 | 0.091 |
| | **7** | 0.150 | 0.101 | 0.155 | 0.110 | 0.100 | 0.104 | 0.159 | 0.146 | 0.216 | 0.159 |
| | **8** | 0.063 | 0.030 | 0.037 | 0.053 | 0.053 | 0.049 | 0.063 | 0.065 | 0.074 | 0.065 |
| | **9** | 0.090 | 0.034 | 0.034 | 0.044 | 0.050 | 0.055 | 0.067 | 0.046 | 0.046 | 0.047 |
| | **10** | 0.021 | 0.018 | 0.030 | 0.026 | 0.042 | 0.051 | 0.037 | 0.033 | **0.013** | 0.031 |
| | **11** | 0.070 | 0.065 | 0.075 | 0.106 | 0.131 | 0.146 | 0.170 | 0.109 | 0.129 | 0.078 |
| | **12** | 0.038 | 0.053 | 0.019 | 0.054 | 0.025 | 0.034 | 0.046 | 0.033 | 0.099 | 0.036 |
| | **13** | 0.096 | 0.076 | 0.082 | 0.076 | 0.076 | 0.097 | 0.092 | 0.119 | 0.105 | 0.085 |
| | **14** | 0.179 | 0.064 | 0.066 | 0.065 | 0.109 | 0.111 | 0.055 | 0.088 | 0.072 | 0.076 |
| | **15** | 0.041 | 0.024 | 0.017 | 0.031 | 0.035 | 0.039 | 0.052 | 0.042 | 0.041 | 0.051 |
| | **16** | 0.127 | 0.069 | 0.072 | 0.146 | 0.094 | 0.084 | 0.088 | 0.088 | 0.104 | 0.098 |
| | **17** | 0.034 | 0.034 | 0.028 | 0.032 | 0.046 | 0.033 | 0.042 | 0.061 | 0.055 | 0.036 |
| | **18** | 0.031 | 0.038 | 0.025 | 0.019 | 0.020 | 0.038 | 0.044 | 0.039 | 0.039 | 0.030 |
| | **19** | 0.055 | 0.025 | 0.072 | 0.076 | 0.047 | 0.098 | 0.078 | 0.072 | 0.073 | 0.078 |
| | **20** | 0.061 | 0.046 | 0.057 | 0.074 | 0.029 | 0.032 | 0.036 | 0.055 | 0.066 | 0.037 |
| | **21** | 0.080 | 0.086 | 0.060 | 0.075 | 0.110 | 0.095 | 0.091 | 0.088 | 0.095 | 0.105 |
| | **22** | 0.094 | 0.064 | 0.052 | 0.097 | 0.053 | 0.139 | 0.079 | 0.081 | 0.064 | 0.133 |
| | **23** | 0.117 | 0.111 | 0.084 | 0.121 | 0.097 | 0.088 | 0.055 | 0.120 | **0.221** | 0.097 |
| | **24** | 0.071 | 0.089 | 0.049 | 0.065 | 0.121 | 0.117 | 0.109 | 0.135 | 0.101 | 0.120 |
| | **25** | 0.065 | 0.089 | 0.077 | 0.062 | 0.065 | 0.108 | 0.092 | 0.182 | 0.099 | 0.079 |

From this table, we found the minimum and maximum standard deviation value. Thus, Threshold value (*TH_x*) will be adjusted within these two values by 10 sectors.

$$TH\_x = \max - (\max - \min)TH \tag{4.1}$$

*where*

*Max value* = 0.221

*Min* value = 0.013

Next, each genuine sample will be compared with the threshold value. If the value does not exceed the Threshold value (*TH_x*), the sample signature will be

accepted and via versa. If $TH = 0.2$, the false rejection sample is highlighted in Table 4.2.

$$TH\_x = 0.221 - (0.221 - 0.013)0.2$$

$$TH\_x = 0.1794$$

**Table 4.2: False Rejection Sample ($TH = 0.2$)**

| | | SAMPLE | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | **1st** | **2nd** | **3rd** | **4th** | **5th** | **6th** | **7th** | **8th** | **9th** | **10th** |
| **SIGNER** | **1** | 0.126 | 0.084 | 0.089 | 0.053 | 0.106 | 0.070 | 0.076 | 0.094 | **0.192** | 0.077 |
| | **2** | 0.058 | 0.080 | 0.053 | 0.056 | 0.057 | 0.050 | 0.070 | 0.046 | 0.062 | 0.055 |
| | **3** | 0.058 | 0.059 | 0.052 | 0.030 | 0.065 | 0.075 | 0.083 | 0.059 | 0.043 | 0.043 |
| | **4** | 0.123 | 0.095 | 0.088 | 0.054 | 0.095 | 0.079 | 0.084 | **0.195** | 0.150 | 0.146 |
| | **5** | 0.131 | 0.083 | 0.060 | 0.079 | 0.077 | 0.076 | 0.080 | 0.117 | 0.068 | 0.136 |
| | **6** | 0.029 | 0.032 | 0.045 | 0.032 | 0.040 | 0.068 | 0.062 | 0.044 | 0.067 | 0.091 |
| | **7** | 0.150 | 0.101 | 0.155 | 0.110 | 0.100 | 0.104 | 0.159 | 0.146 | **0.216** | 0.159 |
| | **8** | 0.063 | 0.030 | 0.037 | 0.053 | 0.053 | 0.049 | 0.063 | 0.065 | 0.074 | 0.065 |
| | **9** | 0.090 | 0.034 | 0.034 | 0.044 | 0.050 | 0.055 | 0.067 | 0.046 | 0.046 | 0.047 |
| | **10** | 0.021 | 0.018 | 0.030 | 0.026 | 0.042 | 0.051 | 0.037 | 0.033 | 0.013 | 0.031 |
| | **11** | 0.070 | 0.065 | 0.075 | 0.106 | 0.131 | 0.146 | 0.170 | 0.109 | 0.129 | 0.078 |
| | **12** | 0.038 | 0.053 | 0.019 | 0.054 | 0.025 | 0.034 | 0.046 | 0.033 | 0.099 | 0.036 |
| | **13** | 0.096 | 0.076 | 0.082 | 0.076 | 0.076 | 0.097 | 0.092 | 0.119 | 0.105 | 0.085 |
| | **14** | **0.179** | 0.064 | 0.066 | 0.065 | 0.109 | 0.111 | 0.055 | 0.088 | 0.072 | 0.076 |
| | **15** | 0.041 | 0.024 | 0.017 | 0.031 | 0.035 | 0.039 | 0.052 | 0.042 | 0.041 | 0.051 |
| | **16** | 0.127 | 0.069 | 0.072 | 0.146 | 0.094 | 0.084 | 0.088 | 0.088 | 0.104 | 0.098 |
| | **17** | 0.034 | 0.034 | 0.028 | 0.032 | 0.046 | 0.033 | 0.042 | 0.061 | 0.055 | 0.036 |
| | **18** | 0.031 | 0.038 | 0.025 | 0.019 | 0.020 | 0.038 | 0.044 | 0.039 | 0.039 | 0.030 |
| | **19** | 0.055 | 0.025 | 0.072 | 0.076 | 0.047 | 0.098 | 0.078 | 0.072 | 0.073 | 0.078 |
| | **20** | 0.061 | 0.046 | 0.057 | 0.074 | 0.029 | 0.032 | 0.036 | 0.055 | 0.066 | 0.037 |
| | **21** | 0.080 | 0.086 | 0.060 | 0.075 | 0.110 | 0.095 | 0.091 | 0.088 | 0.095 | 0.105 |
| | **22** | 0.094 | 0.064 | 0.052 | 0.097 | 0.053 | 0.139 | 0.079 | 0.081 | 0.064 | 0.133 |
| | **23** | 0.117 | 0.111 | 0.084 | 0.121 | 0.097 | 0.088 | 0.055 | 0.120 | **0.221** | 0.097 |
| | **24** | 0.071 | 0.089 | 0.049 | 0.065 | 0.121 | 0.117 | 0.109 | 0.135 | 0.101 | 0.120 |
| | **25** | 0.065 | 0.089 | 0.077 | 0.062 | 0.065 | 0.108 | 0.092 | 0.182 | 0.099 | 0.079 |

Thus, the number of false rejection signature is found as shown as Table 4.3:

**Table 4.3: Number of False Rejection Signature**

| Threshold (TH) | X-Coordinate | Y-Coordinate | Pressure | Velocity |
|---|---|---|---|---|
| **0.0** | 0 | 0 | 0 | 0 |
| **0.1** | 2 | 1 | 2 | 5 |
| **0.2** | 5 | 1 | 2 | 8 |
| **0.3** | 8 | 2 | 4 | 10 |
| **0.4** | 17 | 7 | 6 | 17 |
| **0.5** | 32 | 20 | 21 | 27 |
| **0.6** | 59 | 44 | 47 | 41 |
| **0.7** | 111 | 82 | 93 | 59 |
| **0.8** | 166 | 141 | 163 | 71 |
| **0.9** | 221 | 212 | 237 | 146 |
| **1.0** | 250 | 250 | 250 | 250 |

Next, the FRR can be calculated and the result is shown as Table 4.3.

$$FRR = \frac{No.\_of\_false\_rejection\_sample}{No.\_of\_geniune\_signature} \times 100\% \quad (4.2)$$

**Table 4.4: Percentage of False Rejection Signature**

| Threshold | X-Coordinate (%) | Y-Coordinate (%) | Pressure (%) | Velocity (%) |
|---|---|---|---|---|
| **0.0** | 0.00 | 0.00 | 0.00 | 0.00 |
| **0.1** | 0.80 | 0.40 | 0.80 | 2.00 |
| **0.2** | 2.00 | 0.40 | 0.80 | 3.20 |
| **0.3** | 3.20 | 0.80 | 1.60 | 4.00 |
| **0.4** | 6.80 | 2.80 | 2.40 | 6.80 |
| **0.5** | 12.80 | 8.00 | 8.40 | 10.80 |
| **0.6** | 23.60 | 17.60 | 18.80 | 16.40 |
| **0.7** | 44.40 | 32.80 | 37.20 | 23.60 |
| **0.8** | 66.40 | 56.40 | 65.20 | 28.40 |
| **0.9** | 88.40 | 84.80 | 94.80 | 58.40 |
| **1.0** | 100.00 | 100.00 | 100.00 | 100.00 |

For FAR, the threshold value is needed to compared between each signer's reference signature with all the forgery samples. In example, when $TH$ = 0.2, the false acceptance sample for first signer will be highlighted as shown as Table 4.4.

**Table 4.5: False Acceptance Sample (*TH* = 0.2)**

| SIGNER | SAMPLE | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | 1st | 2nd | 3rd | 4th | 5th | 6th | 7th | 8th | 9th | 10th |
| 2 | 0.189 | 0.199 | 0.228 | 0.216 | 0.216 | 0.226 | 0.240 | 0.234 | 0.231 | 0.221 |
| 3 | 0.284 | 0.271 | 0.312 | 0.285 | 0.295 | 0.295 | 0.288 | 0.284 | 0.288 | 0.292 |
| 4 | 0.265 | 0.241 | 0.272 | 0.248 | 0.237 | 0.237 | 0.241 | 0.207 | 0.226 | 0.245 |
| 5 | 0.203 | 0.204 | 0.200 | 0.215 | 0.214 | 0.208 | 0.194 | 0.231 | 0.206 | 0.267 |
| 6 | 0.248 | 0.245 | 0.235 | 0.256 | 0.244 | 0.246 | 0.248 | 0.232 | 0.238 | 0.246 |
| 7 | 0.241 | 0.287 | 0.305 | 0.226 | 0.250 | 0.256 | 0.263 | 0.252 | 0.246 | 0.239 |
| 8 | 0.202 | 0.219 | 0.216 | 0.229 | 0.226 | 0.209 | 0.224 | 0.211 | 0.232 | 0.227 |
| 9 | 0.214 | 0.189 | 0.203 | 0.210 | 0.209 | 0.215 | 0.185 | 0.198 | 0.213 | 0.212 |
| 10 | 0.204 | 0.208 | 0.219 | 0.225 | 0.211 | 0.215 | 0.216 | 0.217 | 0.212 | 0.211 |
| 11 | **0.131** | **0.161** | **0.175** | 0.213 | 0.183 | 0.226 | **0.148** | 0.181 | 0.187 | **0.155** |
| 12 | 0.196 | 0.248 | 0.230 | 0.218 | 0.212 | 0.227 | 0.207 | 0.202 | **0.178** | 0.207 |
| 13 | 0.213 | **0.160** | 0.205 | **0.172** | **0.175** | **0.162** | 0.196 | 0.193 | **0.174** | **0.173** |
| 14 | 0.176 | 0.135 | 0.144 | 0.157 | 0.192 | 0.171 | 0.146 | 0.169 | 0.148 | 0.161 |
| 15 | 0.240 | 0.248 | 0.251 | 0.250 | 0.257 | 0.254 | 0.244 | 0.247 | 0.250 | 0.255 |
| 16 | 0.312 | 0.311 | 0.305 | 0.308 | 0.304 | 0.281 | 0.305 | 0.309 | 0.289 | 0.300 |
| 17 | 0.218 | 0.220 | 0.221 | 0.228 | 0.222 | 0.214 | 0.222 | 0.220 | 0.224 | 0.215 |
| 18 | 0.234 | 0.220 | 0.223 | 0.227 | 0.226 | 0.217 | 0.211 | 0.221 | 0.222 | 0.222 |
| 19 | 0.287 | 0.269 | 0.265 | 0.279 | 0.237 | 0.275 | 0.250 | 0.248 | 0.262 | 0.240 |
| 20 | 0.192 | 0.189 | 0.192 | 0.203 | 0.196 | 0.194 | 0.195 | 0.202 | 0.167 | 0.191 |
| 21 | 0.272 | 0.249 | 0.243 | 0.251 | 0.223 | 0.246 | 0.251 | 0.266 | 0.256 | 0.229 |
| 22 | 0.249 | 0.195 | 0.220 | 0.199 | 0.216 | 0.174 | 0.182 | 0.212 | 0.216 | 0.260 |
| 23 | 0.212 | 0.258 | 0.265 | 0.319 | 0.272 | 0.264 | 0.265 | 0.296 | 0.274 | 0.305 |
| 24 | 0.196 | **0.171** | **0.177** | 0.187 | 0.209 | **0.173** | 0.226 | 0.237 | **0.177** | 0.201 |
| 25 | 0.215 | 0.223 | 0.222 | 0.212 | 0.218 | 0.233 | 0.208 | 0.183 | 0.185 | 0.207 |
| 26 | 0.267 | 0.263 | 0.268 | 0.286 | 0.250 | 0.251 | **0.166** | 0.194 | **0.177** | **0.170** |
| 27 | 0.230 | 0.227 | 0.224 | 0.215 | 0.220 | 0.226 | 0.255 | 0.219 | 0.209 | 0.210 |
| 28 | 0.305 | 0.296 | 0.303 | 0.319 | 0.307 | 0.277 | 0.264 | 0.298 | 0.265 | 0.297 |
| 29 | 0.281 | 0.271 | 0.266 | 0.277 | 0.259 | 0.259 | 0.233 | 0.237 | 0.220 | 0.218 |
| 30 | 0.204 | 0.224 | 0.210 | 0.218 | 0.205 | 0.211 | 0.210 | 0.207 | 0.198 | 0.210 |
| 31 | 0.220 | 0.217 | 0.246 | 0.239 | 0.249 | 0.189 | 0.216 | 0.224 | 0.221 | 0.228 |
| 32 | 0.266 | 0.248 | 0.258 | 0.238 | 0.238 | 0.294 | 0.263 | 0.240 | 0.259 | 0.244 |
| 33 | 0.184 | 0.225 | 0.194 | 0.215 | 0.216 | 0.189 | 0.161 | 0.211 | 0.203 | 0.204 |
| 34 | 0.182 | 0.197 | 0.192 | **0.176** | 0.193 | 0.240 | 0.224 | 0.187 | 0.206 | **0.153** |
| 35 | 0.216 | 0.216 | 0.228 | 0.217 | 0.219 | 0.188 | 0.202 | 0.203 | 0.197 | 0.192 |
| 36 | 0.190 | 0.239 | 0.232 | 0.221 | 0.232 | 0.242 | 0.219 | 0.202 | 0.228 | 0.233 |
| 37 | 0.228 | 0.228 | 0.258 | 0.257 | 0.266 | 0.254 | 0.244 | 0.280 | 0.268 | 0.269 |
| 38 | 0.162 | 0.156 | 0.144 | 0.136 | 0.149 | 0.204 | 0.218 | 0.225 | 0.214 | 0.227 |
| 39 | **0.118** | **0.140** | **0.131** | **0.148** | **0.143** | 0.218 | 0.233 | **0.160** | **0.169** | 0.216 |
| 40 | 0.227 | 0.235 | 0.242 | 0.242 | 0.234 | 0.214 | 0.232 | 0.242 | 0.241 | 0.246 |
| 41 | 0.328 | 0.320 | 0.312 | 0.313 | 0.312 | 0.293 | 0.260 | 0.285 | 0.298 | 0.317 |

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | **42** | 0.219 | 0.217 | 0.239 | 0.226 | 0.223 | 0.202 | 0.211 | 0.214 | 0.211 | 0.212 |
| | **43** | 0.193 | 0.202 | 0.210 | 0.207 | 0.189 | 0.215 | 0.221 | 0.188 | 0.195 | 0.195 |
| | **44** | 0.248 | 0.215 | 0.222 | 0.222 | 0.225 | 0.266 | 0.236 | 0.254 | 0.257 | 0.240 |
| | **45** | 0.200 | 0.188 | 0.192 | **0.175** | 0.195 | 0.188 | 0.245 | 0.214 | 0.223 | 0.225 |
| | **46** | 0.276 | 0.280 | 0.290 | 0.258 | 0.271 | 0.269 | 0.262 | 0.242 | 0.225 | 0.269 |
| | **47** | 0.206 | **0.168** | 0.203 | 0.199 | 0.196 | 0.222 | 0.200 | 0.253 | 0.209 | 0.257 |
| | **48** | 0.284 | 0.275 | 0.294 | 0.286 | 0.279 | 0.236 | 0.228 | 0.249 | 0.257 | 0.284 |
| | **49** | 0.231 | 0.223 | 0.221 | 0.216 | 0.202 | **0.155** | **0.147** | **0.136** | **0.148** | **0.132** |
| | **50** | 0.221 | 0.208 | 0.236 | 0.241 | 0.231 | 0.187 | 0.230 | 0.214 | 0.222 | 0.216 |

Table 4.5 only show the false acceptance sample between the first signer's reference signature with the remaining 490 should-be-rejected signatures. Same procedure will be repeated for the remaining 24 genuine signers. Thus, the number of false acceptance signature is found as shown as Table 4.6.

**Table 4.6: Number of False Acceptance Signature**

| Threshold | X-Coordinate | Y-Coordinate | Pressure | Velocity |
|---|---|---|---|---|
| **0.0** | 6363 | 10901 | 11757 | 9630 |
| **0.1** | 4861 | 9314 | 11433 | 8857 |
| **0.2** | 3456 | 6763 | 10912 | 8120 |
| **0.3** | 2325 | 3957 | 9977 | 7517 |
| **0.4** | 1400 | 1523 | 8337 | 6980 |
| **0.5** | 697 | 374 | 5922 | 6572 |
| **0.6** | 242 | 103 | 3180 | 5975 |
| **0.7** | 61 | 33 | 982 | 5118 |
| **0.8** | 21 | 8 | 134 | 2934 |
| **0.9** | 3 | 1 | 5 | 245 |
| **1.0** | 0 | 0 | 1 | 0 |

Next, the FAR can be calculated and the result is shown as Table 4.7.

$$FAR = \frac{No.\_of\_false\_accep\tan ce\_sample}{No.\_of\_forgery\_signature} \times 100\% \qquad (4.3)$$

**Table 4.7: Percentage of False Acceptance Signature**

| Threshold | X-Coordinate (%) | Y-Coordinate (%) | Pressure (%) | Velocity (%) |
|---|---|---|---|---|
| **0.0** | 51.94 | 88.99 | 95.98 | 78.61 |
| **0.1** | 39.68 | 76.03 | 93.33 | 72.30 |
| **0.2** | 28.21 | 55.21 | 89.08 | 66.29 |
| **0.3** | 18.98 | 32.30 | 81.44 | 61.36 |
| **0.4** | 11.43 | 12.43 | 68.06 | 56.98 |
| **0.5** | 5.69 | 3.05 | 48.34 | 53.65 |
| **0.6** | 1.98 | 0.84 | 25.96 | 48.78 |
| **0.7** | 0.50 | 0.27 | 8.02 | 41.78 |
| **0.8** | 0.17 | 0.07 | 1.09 | 23.95 |
| **0.9** | 0.02 | 0.01 | 0.04 | 2.00 |
| **1.0** | 0.00 | 0.00 | 0.01 | 0.00 |

From these tables, FAR and FRR graph had been plotted for each parameter as shown as figures below:

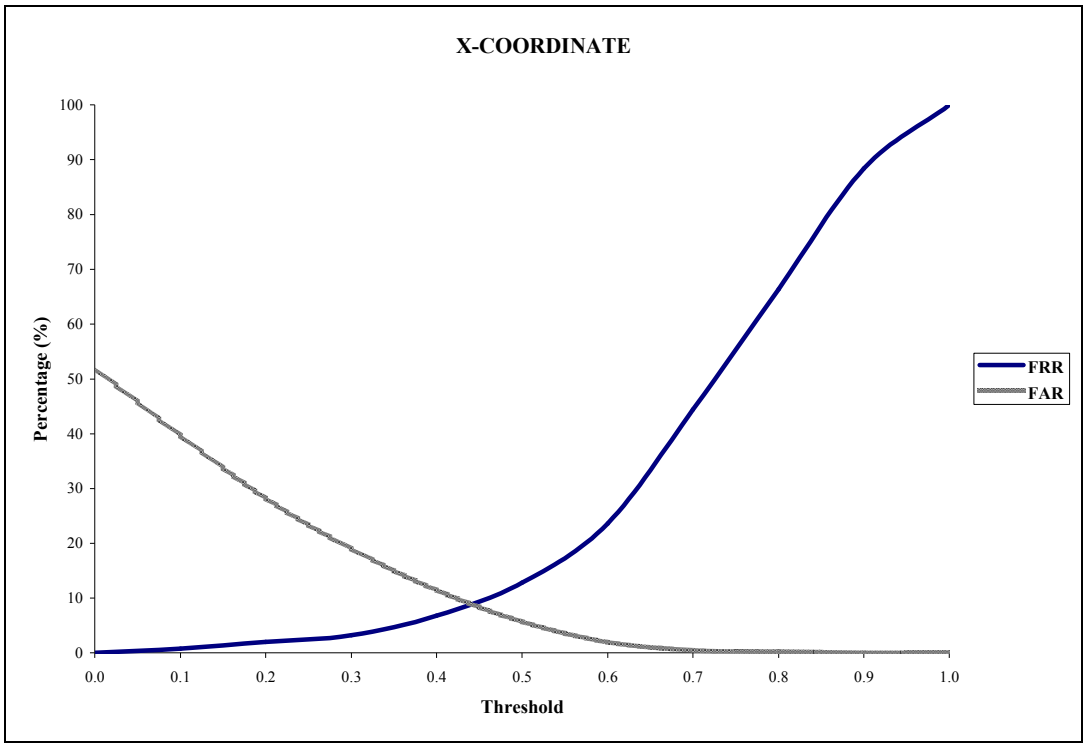


**Figure 4.9: FRR & FAR for X-Coordinate**

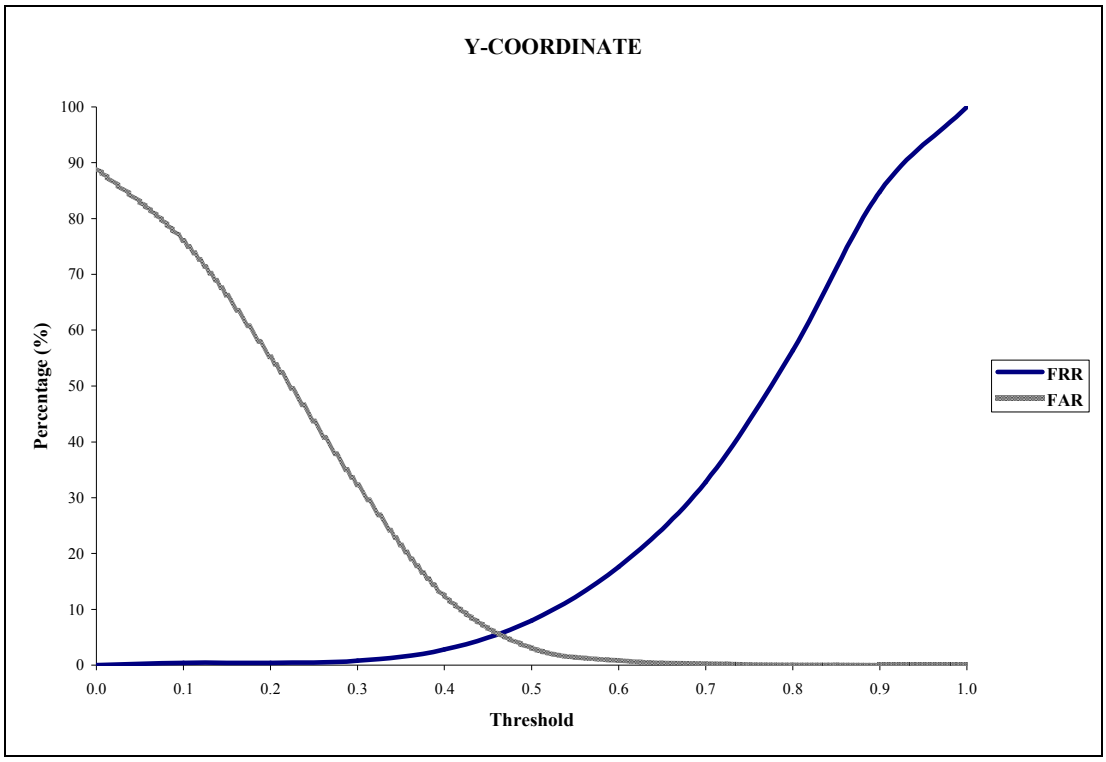


**Figure 4.10: FRR & FAR for Y-Coordinate**

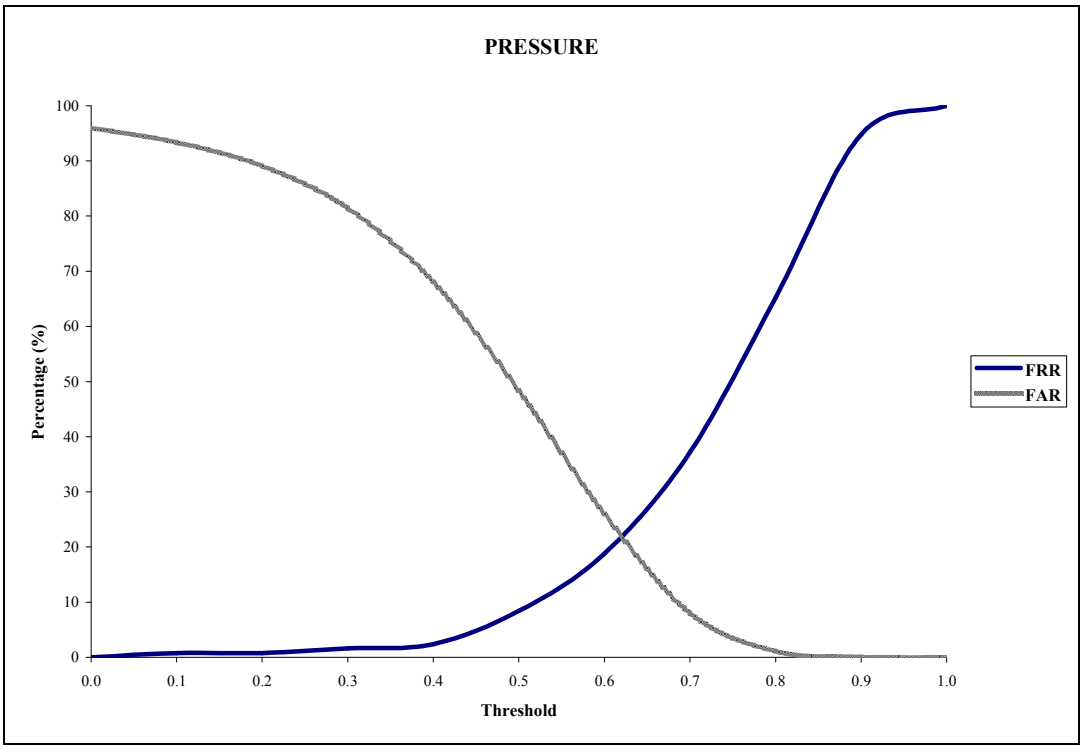


**Figure 4.11: FRR & FAR for Pressure**

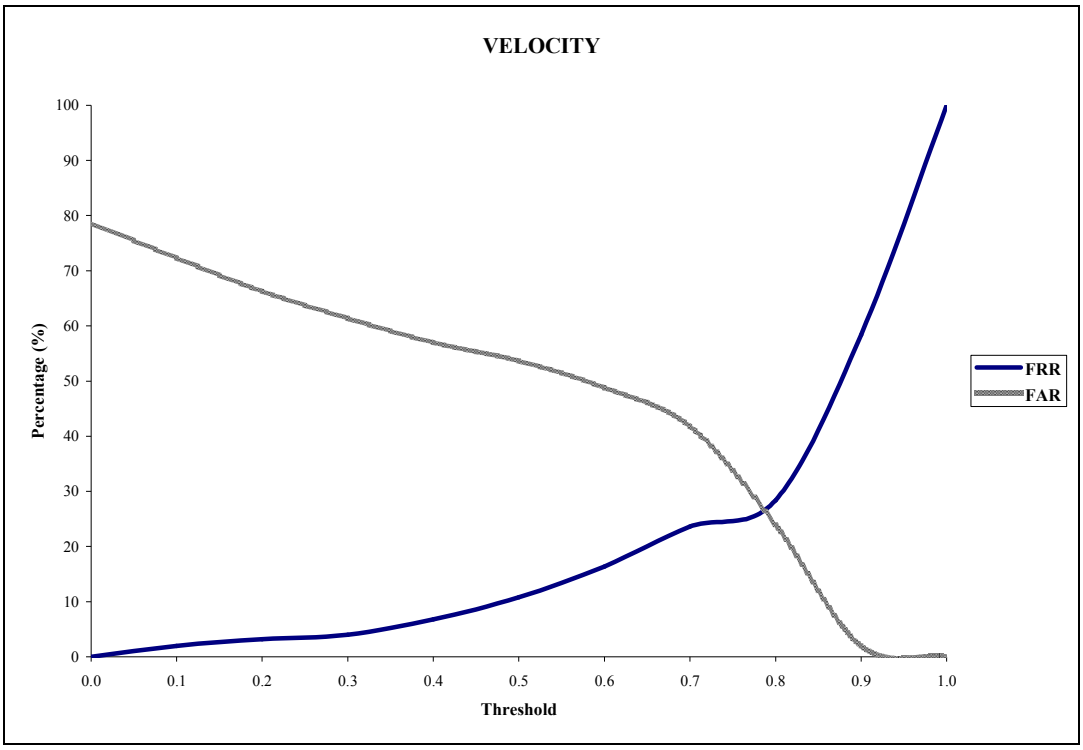


**Figure 4.12: FRR & FAR for Velocity**

From each graph, the Equal Error Rate (EER) which is the interception point between the FRR and FAR had been estimated. The estimated threshold for each feature is shown in Table 4.8.

**Table 4.8: Estimated Threshold for *X*, *Y*, *P* & *V***

| Parameter | Threshold |
|---|---|
| **X** | 0.4394 |
| **Y** | 0.466 |
| **P** | 0.6197 |
| **V** | 0.7795 |

These thresholds were been tested once again to find out the FAR and FRR of the system. Table 4.9 shows the result of it.

**Table 4.9: FAR & FRR by using the Estimated Threshold**

| Parameter | Threshold | EER (%) |
|---|---|---|
| **X** | 0.4394 | 9.166 |
| **Y** | 0.4660 | 6.235 |
| **P** | 0.6197 | 22.43 |
| **V** | 0.7795 | 27.42 |

According to the Table 4.9, we found that the information of x-coordinate and y-coordinate are most dependable followed by the pressure and lasted by the velocity. Thus, condition to accept a signature had been designed as Table 4.10.

**Table 4.10: Condition of Acceptance for the System**

| X | Y | P | V | Decision |
|---|---|---|---|---|
| TRUE | TRUE | TRUE | TRUE | **TRUE** |
| TRUE | TRUE | TRUE | FALSE | **TRUE** |
| TRUE | TRUE | FALSE | TRUE | **TRUE** |

To accept a signature, X and Y and (P or V) must give a TRUE output. This means the system will accept a signature when X is accepted, Y is accepted, and either P or V or both is accepted. To conclude, the decision of the system will be as the logic equation as below:

$$X \bigcap Y \bigcap (P \bigcup V) = ACCEPT \tag{4.4}$$

## 4.6 Classification

Classification has been done for the experimental purpose. The result is shown in Table 4.11.

**Table 4.11: FAR & FRR by using Different Combination of the Selected Parameters**

| Selected Parameter | FRR (%) | FAR (%) | ERROR (%) |
|---|---|---|---|
| **X,P,V** | 48.80 | 1.45 | 2.40 |
| **X,P** | 26.00 | 3.87 | 4.31 |
| **X,V** | 35.20 | 4.97 | 5.58 |
| **Y,P,V** | 45.60 | 0.93 | 1.82 |
| **Y,V** | 32.00 | 3.20 | 3.78 |
| **Y,P** | 22.80 | 2.57 | 2.98 |
| **X,Y** | 11.20 | 3.13 | 3.29 |
| **X,Y,V,P** | **14.80** | **2.64** | **2.89** |

From Table 4.11, we found that the combination of x-coordinate, y-coordinate, pressure and velocity is the best combination among the others. Although the error rate is not the lowest, this combination provided a balance FRR and FAR. Besides that, it also gives enough information to the system for verification purpose. While x-coordinate and y-coordinate represent the position information of the signature, velocity and pressure represent the unique characteristic of the each signer. Thus, the combination of x,y,v and p was selected.

## 4.7 Data Analysis

After troubleshooting the proposed system, we found that this system has 14.8% of FRR and 2.64% of FAR. For an ideal dynamic signature verification system, it should have equal FAR and FRR so that the system will provide a better result. In this proposed system, it has non equal FRR and FAR. Low FAR and high FRR show that this system is slightly over sensitive to the signature.

However, it is acceptable because the testing database is much larger than the reference database. The reference database contains of 25 signers' signatures. However, the testing database contains of 50 (25 genuine + 25 forgery) signers' signatures. Meaning that, each reference signature is required to compare with 500 testing signatures. In these 500 testing signatures, there are only 10 samples of signature are genuine to the reference signature. Here, we can see that the rejection rate is about 490 signatures compared to 10 signatures which should be accepted. Thus, a more sensitive system is required for this database.

The error of the proposed system is 2.89% which is approximate 3% error. In other word, if 100 signatures are being tested by this system, there will be 3 wrong decisions made by the system. In order word, this system achieve approximate 97% classification rate. The same reason can be used to explain why this system has such high classification rate. The reason is this system is over sensitive, and the testing database is much larger than the reference database.

# CHAPTER 5

# CONCLUSION AND RECOMMENDATIONS

## 5.1 Conclusion and Recommendations

In conclusion, the proposed dynamic signature verification system had presented an approach to dynamic signature verification system and some verification problems. An accurate and efficient signature verification system had been constructed. This proposed system has 14.8% of FRR and 2.64% of FAR. Meanwhile, it has the error rate as 2.89% which mean it also has an approximate 97% of classification rate. Next, study in dynamic approach of signature verification system had been done.

On the other hand, an algorithm to solve the signature verification had been produced. To accept a signature, x-coordinate (X) and y-coordinate (Y) and pressure (P) or velocity (V) must give a TRUE output. This means the system will accept a signature when X is accepted, Y is accepted, and either P or V or both is accepted. The decision of the proposed system is shown as below:

$$\boxed{X \cap Y \cap (P \cup V) = ACCEPT}$$

This input of the proposed system comes from the SUSig Online Database. Thus, the result is fully depends on the database as well. This proposed system has not been tested practically yet. In future, this proposed system is recommended to be tested practically in real world.

According to Table 4.9, we found that the error rate of velocity and pressure is quite high compared to the x- and y-coordinate. This is a common issue for allk the signature verification system because of the inconsistency of signing the signature. Thus, improvement in this aspect is hoped to be done in future. Thus, a better result will be produced.

# REFERENCES


Charles E. Pippin. (July 2004). Dynamic Signature Verification using Local and Global Features. *IEEE.*

David Fenton, Martin Bouchard and Tet H.Yeap. (2006). Evaluation of Features and Normalization Techniques for Signature Verification using Dynamic Time Warping. *IEEE Trans. ICASSP*.

Fernando.Alonso, Julian.Fierrez, Marcos.Martinez, and Javier.Ortega. (2009). Fusion of Static Image and Dynamic Information for Signature Verification. *IEEE Trans.*

Juan D. Penpgos, Nagarajan Prabhakaran and Subbarao V. Wunnava. (1996). An Efficient Scheme for Dynamic Signature Verification. *IEEE Trans.*

Marcin Adamski, and Khalid Saeed. (2008). Online Signature Classification and its Verification System. *IEEE Trans.* Computer Society

Musa Mailah and Lim Boon Han. (2008). Biometric Signature Verification using Pen Position, Time, Velocity, and Pressure Parameters. *Technology Journal from UTM*.

Oscar Miguel-Hurtado, Luis Mengibar-Pozo and Andrzej Pacut. (2008). A New Algorithm for Signature Verification System Based on Dynamic Time Warping and Gaussian Mixture Models. *IEEE Trans. ICCST*, pp. 206-213.

Oscar Miguel-Hurtado, Luis Mengibar-Pozo, Michael G. Lorenz and Judith Liu-Kimenez. (2007). Online Signature Verification System by Dynamic Time Warping and Gaussian Mixture Models. *IEEE Trans. ICCST*, pp. 23-29.

Quen-Zong Wu, I-Chang Jou and Suh-Yin Lee. (February 1997). On-Line Signature Verification Using LPC Cepstrum & Neural Networks. *IEEE Trans.* SMC, Vol. 27, no. 1, pp. 148-153.

R. Plamondon and S.N. Srihari. (January 2000). On-line and Off-line Handwriting Recognition: A Comprehensive Survey. *IEEE Trans. PAMI*, vol. 22, no. 1, pp. 63–84.

R. Seiler M. Schenkel and E Eggimann. (1996). Off-Line Cursive Handwriting Recognition Compared with On-Line Recognition. *IEEE Trans. ACPR*.

Ronny Martens, and Luc Claesen. (1997). Dynamic Programming Optimisation for On-line Signature Verification. *IEEE Tans*.

Won-Du Chang, and Jungpil Shin. (2005) Modified Dynamic Time Warping for Stroke-based On-line Signature Verification. *IEEE Trans.* Pattern Recognition.

# APPENDICES

## APPENDIX A: Matlab Coding for the Main Program

```
%\\Main Program
clear all;

%User Entry
user_1 = input('User(Pattern) Number? (1-25) ');
User_database;
parameter = 0;

%Pre-processing for Pattern(X,Y,P)
%X-Coordinate
j1 = x1;
j2 = x2;
j3 = x3;
j4 = x4;
j5 = x5;

Normalisation;
ref_x = ref;

%Y-Coordinate
j1 = y1;
j2 = y2;
j3 = y3;
j4 = y4;
j5 = y5;

Normalisation;
ref_y = ref;

%Pressure
j1 = Pressure1;
j2 = Pressure2;
j3 = Pressure3;
j4 = Pressure4;
j5 = Pressure5;

parameter = 'p';
Normalisation;
ref_p = ref;
```

```
user_2 = input('User(Testing Sample) number? (1-50) ');
sample = input('Sample number? (1-10) ');
Sample_database;

%Pre-processing for Sample (X,Y,P)
%X-Coordinate
xT = (xT-min(xT))/(max(xT)-min(xT));
iT = max(size(xT));
xT = resample(xT,i_ARG,iT);
xT = smooth(xT,'moving');

%Y-Coordinate
yT = (yT-min(yT))/(max(yT)-min(yT));
iT = max(size(yT));
yT = resample(yT,i_ARG,iT);
yT = smooth(yT,'moving');

%Pressure
pT = PressureT;
pT = (pT-min(pT))/(max(pT)-min(pT));
iT = max(size(pT));
pT = resample(pT,i_ARG,iT);
pT = smooth(pT,'moving');
pT = smooth(pT,'moving');

%\\Calculation of Standard Deviation (X,Y,P)
dx_total = 0;
dx_square_total = 0;
dy_total = 0;
dy_square_total = 0;
dp_total = 0;
dp_square_total = 0;

for o=1:i_ARG
    dx = ref_x(o)-xT(o);
    dx = abs(dx);
    dx_square = dx*dx;
    dx_total = dx_total + dx;
    dx_square_total = dx_square_total + dx_square;

    dy = ref_y(o)-yT(o);
    dy = abs(dy);
    dy_square = dy*dy;
    dy_total = dy_total + dy;
    dy_square_total = dy_square_total + dy_square;

    dp = ref_p(o)-pT(o);
    dp = abs(dp);
    dp_square = dp*dp;
    dp_total = dp_total + dp;
    dp_square_total = dp_square_total + dp_square;
end

x_deviation = dx_total/i_ARG;
x_variance = dx_square_total/i_ARG;
x_standard_deviation = sqrt(x_variance);

y_deviation = dy_total/i_ARG;
y_variance = dy_square_total/i_ARG;
```

```
y_standard_deviation = sqrt(y_variance);

p_deviation = dp_total/i_ARG;
p_variance = dp_square_total/i_ARG;
p_standard_deviation = sqrt(p_variance);

%Calculation for Velocity
User_database;
parameter = 0;

%Pre-processing for Pattern (V)
v_x1 = diff(x1);
v_x2 = diff(x2);
v_x3 = diff(x3);
v_x4 = diff(x4);
v_x5 = diff(x5);

v_y1 = diff(y1);
v_y2 = diff(y2);
v_y3 = diff(y3);
v_y4 = diff(y4);
v_y5 = diff(y5);

v1 = sqrt((v_x1.*v_x1)+(v_y1.*v_y1));
v2 = sqrt((v_x2.*v_x2)+(v_y2.*v_y2));
v3 = sqrt((v_x3.*v_x3)+(v_y3.*v_y3));
v4 = sqrt((v_x4.*v_x4)+(v_y4.*v_y4));
v5 = sqrt((v_x5.*v_x5)+(v_y5.*v_y5));

j1 = v1;
j2 = v2;
j3 = v3;
j4 = v4;
j5 = v5;

Normalisation;
ref_v = ref;
Sample_database;

%Pre-processing for Sample (V)
v_xT = diff(xT);
v_yT = diff(yT);
vT = sqrt((v_xT.*v_xT)+(v_yT.*v_yT));

vT = (vT-min(vT))/(max(vT)-min(vT));
iT = max(size(vT));
vT = resample(vT,i_ARG,iT);
vT = smooth(vT,'moving');

%\\Calculation of Standard Deviation (V)
dv = ref_v-vT;
dv = abs(dv);
dv_square = dv.*dv;

dv_total = sum(dv);
dv_square_total = sum(dv_square);

v_deviation = dv_total/i_ARG;
v_variance = dv_square_total/length(dv);
```

```
v_standard_deviation = sqrt(v_variance);

%Verification (X,Y,V,P)
TH_x = 0.2211 - (0.2211 - 0.0129)*0.4394;
TH_y = 0.3608 - (0.3608 - 0.0314)*0.466;
TH_p = 0.2809 - (0.2809 - 0.0515)*0.6197;
TH_v = 0.2374 - (0.2374 - 0.0123)*0.7795;

if x_standard_deviation <= TH_x
    x = 1;
else
    x = 0;
end

if y_standard_deviation <= TH_y
    y = 1;
else
    y = 0;
end

if p_standard_deviation <= TH_p
    p = 1;
else
    p = 0;
end

if v_standard_deviation <= TH_v
    v = 1;
else
    v = 0;
end

%Verification (Final)
x = x*0.3;
y = y*0.4;
p = p*0.2;
v = v*0.1;
TH_all = x + y + p + v;
TH = 0.75;

if TH_all > TH
    disp('ACCEPT!');
else
    disp('REJECT!')
end
```

## APPENDIX B: Matlab Coding for Pre-processing

```
%\\Magnitude Normalisation
j1 = (j1-min(j1))/(max(j1)-min(j1));
j2 = (j2-min(j2))/(max(j2)-min(j2));
j3 = (j3-min(j3))/(max(j3)-min(j3));
j4 = (j4-min(j4))/(max(j4)-min(j4));
j5 = (j5-min(j5))/(max(j5)-min(j5));

%\\Time Normalisation
i1 = max(size(j1));
i2 = max(size(j2));
i3 = max(size(j3));
i4 = max(size(j4));
i5 = max(size(j5));

i_ARG = (i1+i2+i3+i4+i5)/5;
i_ARG = round(i_ARG);
i = 1:i_ARG;

j1 = resample(j1,i_ARG,i1);
j2 = resample(j2,i_ARG,i2);
j3 = resample(j3,i_ARG,i3);
j4 = resample(j4,i_ARG,i4);
j5 = resample(j5,i_ARG,i5);

j1 = smooth(j1,'moving');
j2 = smooth(j2,'moving');
j3 = smooth(j3,'moving');
j4 = smooth(j4,'moving');
j5 = smooth(j5,'moving');

if parameter == 'p';
    j1 = smooth(j1,'moving');
    j2 = smooth(j2,'moving');
    j3 = smooth(j3,'moving');
    j4 = smooth(j4,'moving');
    j5 = smooth(j5,'moving');
end

ref =  (j1+j2+j3+j4+j5)/5;
```

## APPENDIX C: Matlab Coding for Verification

```
%\\Normalisation of Sample
if parameter == 'x';
    jT = xT;

elseif parameter == 'y';
    jT = yT;

elseif parameter == 'p';
    jT = PressureT;

elseif parameter == 'v';
    v_xT = diff(xT);
    v_yT = diff(yT);
    vT = sqrt((v_xT.*v_xT)+(v_yT.*v_yT));
    jT = vT;
end;

jT = (jT-min(jT))/(max(jT)-min(jT));
iT = length(jT);
jT = resample(jT,i_ARG,iT);
jT = smooth(jT,'moving');

if parameter == 'p';
    jT = smooth(jT,'moving');
end

d_total = 0;
d_square_total = 0;

for o=1:i_ARG
    d = ref(o)-jT(o);
    d = abs(d);
    d_square = d*d;
    d_total = d_total + d;
    d_square_total = d_square_total + d_square;
end

deviation = d_total/i_ARG;
variance = d_square_total/i_ARG;
standard_deviation = sqrt(variance);

disp('Standard Deviation = ');
disp(standard_deviation);

%Thresholding
if parameter == 'x';
```

```
    TH_x = 0.2211 - (0.2211 - 0.0129)*0.4394;
    if standard_deviation <= TH_x;
        disp('Accept')
    else
        disp('Reject')
    end

elseif parameter == 'y';
    TH_y = 0.3608 - (0.3608 - 0.0314)*0.466;
    if standard_deviation <= TH_y;
        disp('Accept')
    else
        disp('Reject')
    end

elseif parameter == 'p';
    TH_p = 0.2809 - (0.2809 - 0.0515)*0.6197;
    if standard_deviation <= TH_p;
        disp('Accept')
    else
        disp('Reject')
    end

elseif parameter == 'v';
    TH_v = 0.2374 - (0.2374 - 0.0123)*0.7795;
    if standard_deviation <= TH_v;
        disp('Accept')
    else
        disp('Reject')
    end
end
```