Jurnal Teknologi, 48(A) Jun 2008: 35 - 54

BIOMETRIC SIGNATURE VERIFICATION USING PEN POSITION, TIME, VELOCITY AND PRESSURE PARAMETERS

MUSA MAILAH[1] & LIM BOON HAN[2]

Abstract. The paper 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.

Keywords: Biometrics, signature verification, data acquisition, tolerance bands, neural network

Abstrak. Kertas kerja ini menerangkan tentang pembangunan satu sistem pengesahan tandatangan bertulis tangan yang melibatkan tekanan pen terhadap laluan tandatangan, masa ketika menandatangan, profil kelajuan dan kedudukan rupa bentuk tandatangan. Isyarat bertulis tangan telah diperoleh dan diolah secara berdigit menggunakan *tablet*. Ciri utama sistem pengesahan tandatangan yang dicadangkan ialah tandatangan bertulis tangan yang dikemaskini secara dinamik, keupayaan cuba semula semasa pengesahan, kegunaan jalur terima beserta nilai ambang, pembangunan mesra pengguna berdasarkan antara muka grafik pengguna, penggunaan kaedah paksi masa sepunya dan pengesahan tandatangan menggunakan satu kelas rangkaian neural laluan hadapan berlapis. Satu algoritma khusus telah diguna pakai yang dapat memberikan keputusan pengesahan dengan ketepatan yang baik serta lebih cepat. Sistem telah menghasilkan kadar penolakan palsu sebesar 1.3% dan kadar penerimaan palsu 0% dengan pengesahan dilakukan menggunakan tandatangan palsu yang telah diciplak.

Kata kunci: Biometrik, penentusahan tandatangan, perolehan data, jalur terima, rangkaian neural

1.0 INTRODUCTION

Biometrics refers to a field of study that is concerned with any characteristic or personal trait that can be used to identify or verify a person. This characteristic is

[1&2] Department of Applied Mechanics, Faculty of Mechanical Engineering, Universiti Teknologi Malaysia, 81310 Skudai, Johor, Malaysia. Email: musa@fkm.utm.my, bhlim80@yahoo.com

essentially distinctive, will not (or hardly) change with time and more often than not, unique to each and every individual person. Examples can be found in human fingerprints, signatures, voices, dental records, DNA signatures and shapes of iris [1]. The subject of interest in this research is signature verification that deals with the process of verifying the written signature patterns of human individuals. On-line data acquisition, signatures registration, preprocessing, signature storage, verification algorithms and off-line neural network verification are investigated in this research. There are basically two types of signature verification, on-line and off-line signature verifications. Off-line method identifies signatures using an image processing procedure whereby the user is supposed to have written down completely the signature onto a template that is later captured by a CCD camera or scanner to be processed. On-line signature verification involved the capturing of dynamic signature signals such as pressure of pen tips, time duration of whole signature and velocity along signature path. Types of signature verification, methods and performance evaluation are well reviewed by Plamondon and Lorette [2], LecLerc and Plamondon [3] and Dimauro *et al.* [4]. Zimmer and Lee [5] have incorporated on-line and off-line methods in signature verification system.

Various devices are used for on-line data acquisitions. Nelson and Kishon [6] and Hamilton *et al.* [7] use pressure sensitive tablet to retrieve online data while Sakamoto *et al.* [8] and Tham *et al.* [9] utilise pen equipped with pressure sensor and signal conditioning element that can extract the pressure distribution characteristics of the written signature. There are several algorithms applied to preprocessing of the data retrieved on-line. Among them are normalization [10], linear prediction coding [11], dynamic time warping [12-14], tree matching [15], smoothing of data [16], noise reduction [15] and segmentation [5, 16-18]. It is necessary to combine more than one of the algorithms listed above to preprocessing a signature signal. For instance, Hastie *et al.* [16] have applied normalization, dynamic time warping, smoothing of data and segmentation to preprocess the data retrieved. Artificial intelligence (AI) techniques such as the use of neural network [11, 14, 19], fuzzy logic [10] and fuzzy neural [20] are common tools that are used in verifying handwritten signatures. Other researchers use combination of neural network and autoregressive [21], statistical model [16], Hidden Markov Model [18, 22] and string matching [23].

The paper is structured as follows: the first part describes the architecture of the proposed system followed by a description on the data acquisition process and development of GUI. Data pre-and post-processing of the captured signals are next elaborated with emphasis given on the normalization and smoothing of the signals. The application of a class neural network with the classic error back-propagation algorithm is then highlighted that includes the off-line training of data acquired and processed earlier. Consequently, verification and evaluation of the proposed biometric system is carried out towards the end of paper with conclusion and direction for future study is finally outlined.

2.0 ARCHITECTURE OF PROPOSED SYSTEM

A biometric system comprising hardware and software elements is proposed. Tablet with pressure sensitive surface was used in the study to capture absolute pen pressure signal, time and *XY* coordinate cooperating with a special pen. This is the main input device for data acquisition.

The software architecture of the system involves the use of Microsoft Visual Basic, MATLAB v 6.0, Microsoft Jet Engine, Microsoft Access and their associated components. Wintab32 API and its library were employed to enable communication between the tablet and Microsoft Visual Basic program. MATLAB was used to perform the preprocessing and verification processes while Microsoft Jet engine was chosen as a mean to handle the database (mdb) files in order to store and retrieve signature information from database. The interaction between all the components described above is shown in Figure 1. The implementation of the scheme involves a number of stages as shown in Figure 2.

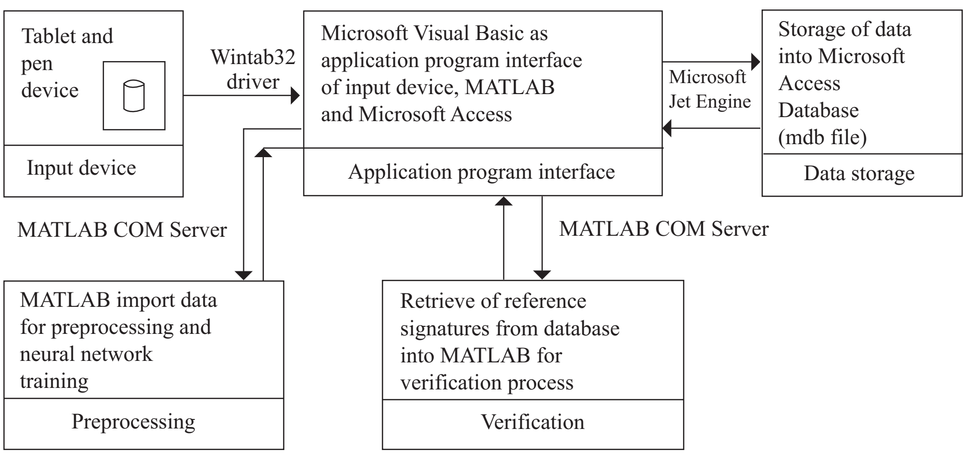


Figure 1 Process flow showing the interactive of the main hardware and software components

The first part is the registration procedure in which the user is required to provide the handwritten input signature. A digitised tablet with a writing device (pen) attached is used to accomplish this task. The information obtained shall provide the essential on-line raw data to the computer database. User is prompted to sign his/her signature five times for the initial data acquisition phase and training using neural network. The second phase is the preprocessing of the signatures involving the use of suitable algorithms that will be discussed in Section 4. Next part involves the actual neural network training and identifying process of the input signature patterns that will yield suitable weights and biases. This is immediately followed by storing of the processed information in a database. Finally, test signature pattern from the user shall be verified (also after being preprocessed) using the proposed verification algorithms with the application of neural network method taking into account some tolerance bands and threshold values. If the test pattern matches those in database with respect to a number of parameters (pressure distribution, time, velocity profile and coordinate positions), it shall be accepted otherwise it will be outright rejected.

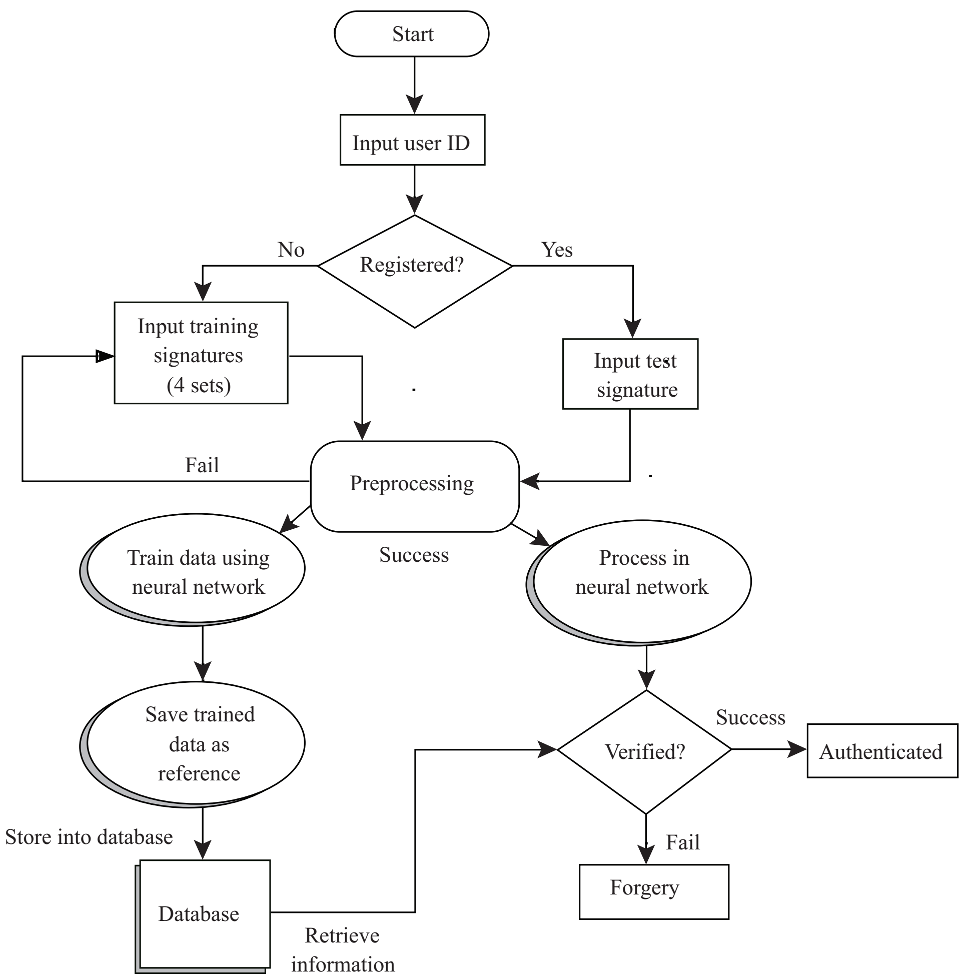


Figure 2 Signature verification process flow

3.0 DATA ACQUISITION AND GRAPHIC USER INTERFACE

Genius 4x3 PenWizard tablet with a writing device (pen) was selected as the main input hardware for the data acquisition procedure. This device sends data in packet through the Universal Serial Bus (USB) port [24] to a PC. Pressure, *X* and *Y* coordinates and time information can be extracted from this on-line operation. The term 'on-line' is used because the data acquisition procedure is performed during the actual signing process. The on-line data are then saved into a database. Figure 3 shows a photograph of the whole system used in the study.

The software program incorporates GUI elements to enable the user to input his/her information and signature in a user's friendly environment with the aid of 'buttons' and 'field boxes'. Figure 4 shows GUI window for 'new user' while Figure 5 is created for the 'existing user'. Figure 6 shows the layout of the GUI window for signature registration which prompted the new user to place his written signatures (repeated for five times) in the box provided to enable preprocessing and training of the data to take place. Existing user is required to key-in his/her identification number (IC) in which his/her information could be directly accessed and retrieved from the database prior to the signing process for verification. Later, the user is required to sign his/her signature for authentication (three times of retries are allowed). The GUI window for this procedure can be seen in Figure 7. All signature information will be stored in a database and can be easily retrieved and reproduced whenever required. The design of the storage database is shown in Figure 8.

Figure 3 Hardware involved for data acquisition of handwritten signature

Figure 4 GUI window for new user

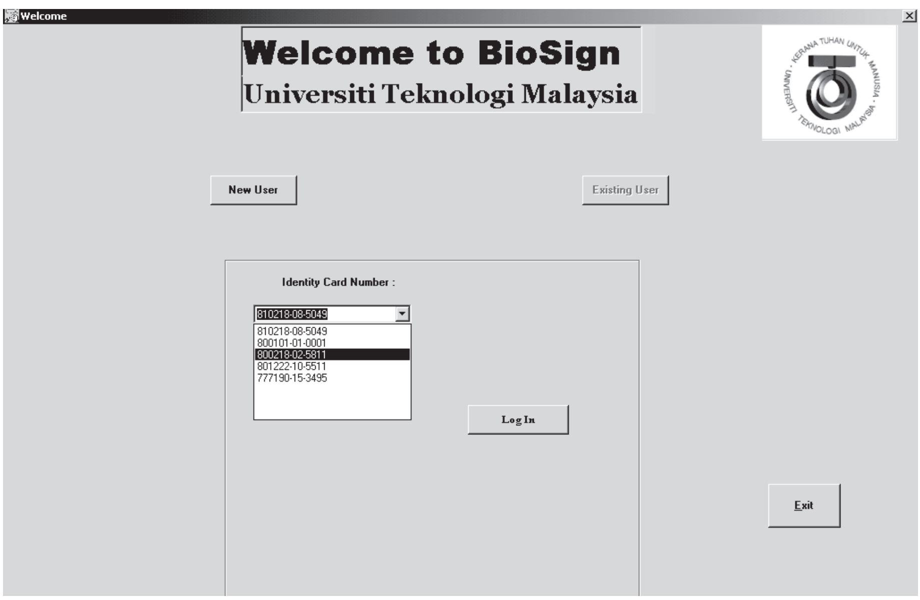


Figure 5 GUI window for existing user

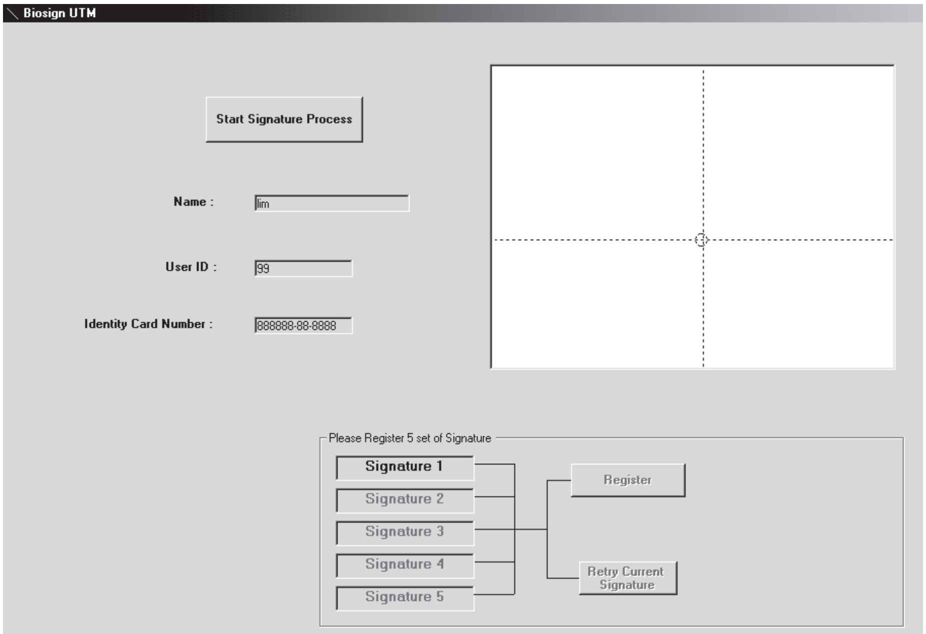


Figure 6 GUI window prompting new user to sign five times for training purpose

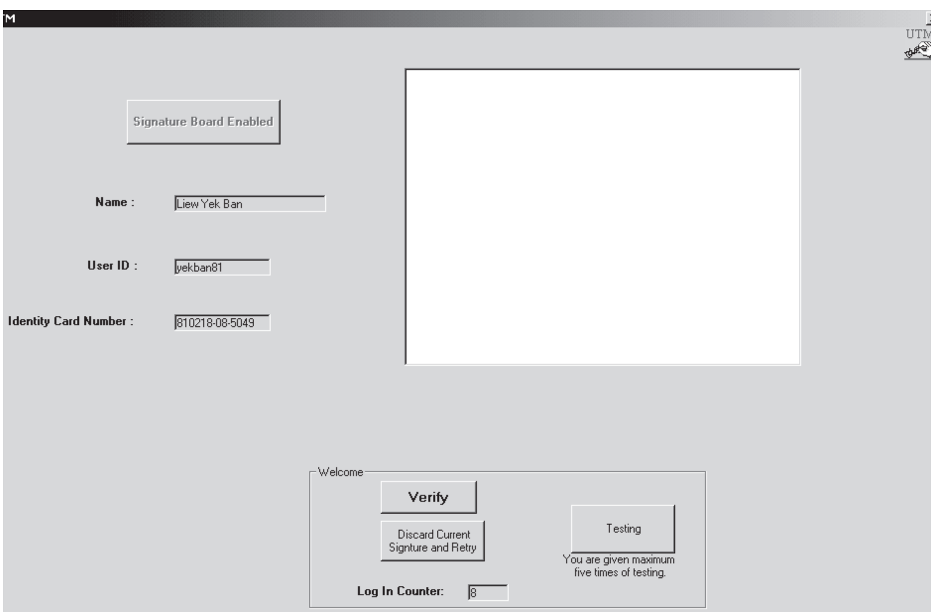
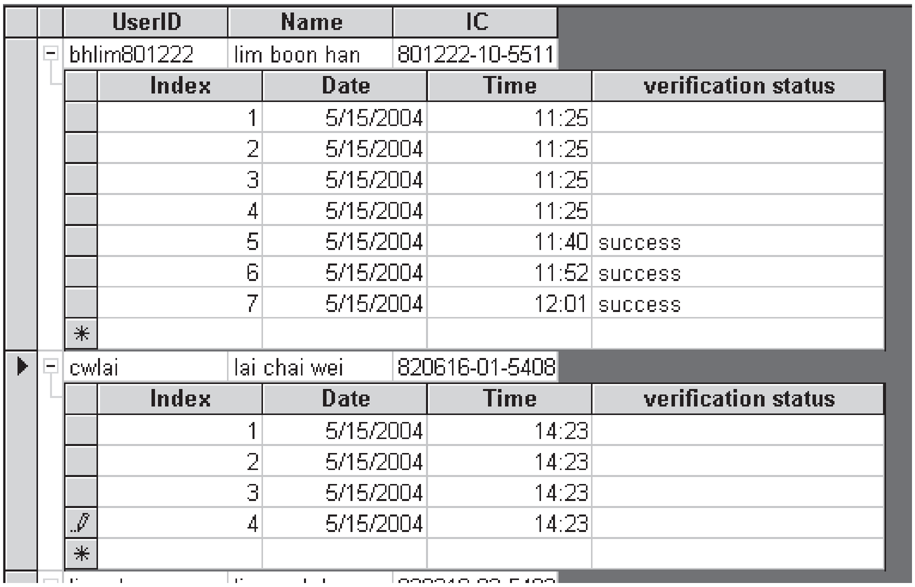


Figure 7 GUI window prompting existing user to undergo verification process (three retries are allowed)



UserID	Name	IC
bhlim801222	lim boon han	801222-10-5511

Index	Date	Time	verification status
1	5/15/2004	11:25	
2	5/15/2004	11:25	
3	5/15/2004	11:25	
4	5/15/2004	11:25	
5	5/15/2004	11:40	success
6	5/15/2004	11:52	success
7	5/15/2004	12:01	success

UserID	Name	IC
cwlai	lai chai wei	820616-01-5408

Index	Date	Time	verification status
1	5/15/2004	14:23	
2	5/15/2004	14:23	
3	5/15/2004	14:23	
4	5/15/2004	14:23	

Figure 8 Database design for signature storage

4.0 DATA PROCESSING

All the input data that were acquired on-line will be imported into MATLAB environment to be preprocessed via a number of techniques. The first step of preprocessing is to normalize the signatures. Four signatures (second, third, fourth and fifth) will be rotated, scaled and transformed against the first signature. The initial transformation procedure is given as:

$$\begin{bmatrix} x_t \\ y_t \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} T_x \\ T_y \end{bmatrix} \tag{1}$$

where (x, y) is the original coordinate
(x_t, y_t) is the location to translate
T_x, T_y is the magnitude of translation

For the rotation procedure, the X and Y coordinates of the signature sets (with respect to time) were initially changed to polar form and then rotated about a reference point at an angle before being converted back to Cartesian coordinate expressed as follows:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{2}$$

where (x', y') is the new coordinate after rotation
θ is the angle to rotate

The next step in normalization is to scale the signatures to fit the first signature in X and Y-axis. This is done by finding the maximum and minimum value of X and Y coordinates of the first signature set and fit other signatures to its size as follows:

$$\Theta X^i = ((\Theta X^1_{max} - \Theta X^1_{min}) / (\Theta X^i_{max} - \Theta X^i_{min}))\, X^i \tag{3}$$

$$\Theta Y^i = ((\Theta Y^1_{max} - \Theta Y^1_{min}) / (\Theta Y^i_{max} - \Theta Y^i_{min}))\, Y^i \tag{4}$$

where ΘX is the scaled x coordinate
ΘY is the scaled y coordinate

max, min are the maximum and minimum values of each x and y coordinate.

Generally, all signatures from the same individual are almost similar but they are not identical from each other and hence different signature has correspondingly different time duration. This different timing pattern is normally not linear and many researchers use Dynamic Time Warping (DTW) to obtain a point to point correspondent between two similar signatures. DTW is typically employed to accommodate the timing but not the X and Y coordinates of the signature. By applying DTW into X and Y coordinates, it will distort the shape of signature [16]. Due to time consumption of DTW algorithm, part of DTW algorithm has been extracted for preprocessing the pressure signal from the data acquired known as Common Time Axes (CTA). This will save memory usage and enhanced processing speed. The algorithm [25] is given as:

$$S1 = R(\mathrm{a}) \quad a = 1,2,3\ldots.A \tag{5}$$

$$S2 = T(\mathrm{b}) \quad b = 1,2,3\ldots B \tag{6}$$

where $S1$ is the first signature signal
$S2$ is the second signature signal
$R(a)$ is the function representing the first signature with "a" number of points
$T(b)$ is the function representing the second signature with "b" number of points

$$a = i(n)\ n = 1,2,3\ldots N \tag{7}$$

$$b = j(n)\ n = 1,2,3\ldots N \tag{8}$$

where n is the number of points for common time axis
$i(n)$ is the function representing a with common time axis n
$j(n)$ is the function representing b with common time axis n

All the input data were then smoothed out using cubic smoothing spline algorithm that will remove undesirable spikes and noises when recording input from tablet [26]. The noise data points are normally less than 1% of the total signal [11]. Finally, the velocities of both X and Y coordinates are calculated as [27]:

$$v = (v_x, v_y) = (\dot{x}, \dot{y}) \tag{9}$$

where v is the velocity magnitude
v_x is the X component velocity or $\dot{x}$ being the derivative of X signal
vy is the Y component velocity or $\dot{y}$ being the derivative of Y signal
After the input data pass through all the preprocessing stage, the results are plotted in MATLAB. Figures 9 to 13 show the raw data acquired from the tablet (input hardware).

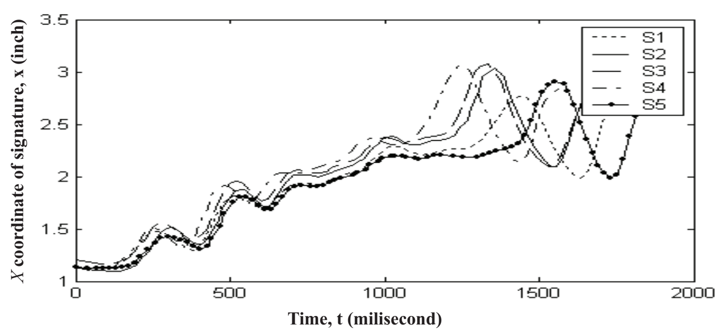


Figure 9 Original X coordinate of signature

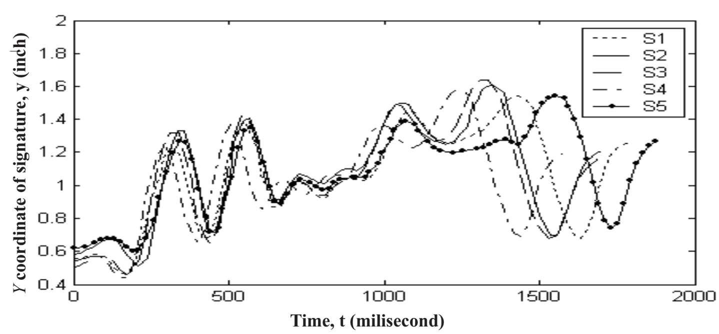


Figure 10 Original Y coordinate of signature

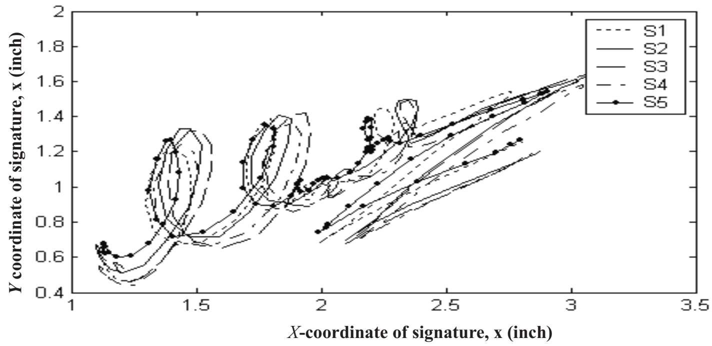


Figure 11 Original shape of signature

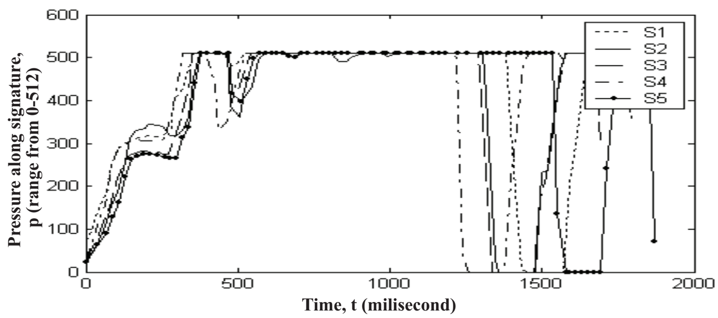


Figure 12 Original pressure signal of signature

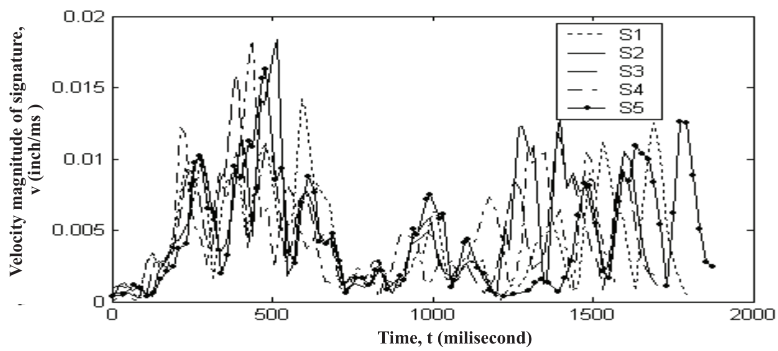


Figure 13 Original velocity magnitude of signature

Figures 14 to 18 show the plots of the signals after preprocessing for each of the *X*, *Y*, signature shape, pressure information and velocity magnitude of signature. It is obvious that the processed data plots are smoother, more presentable and suitable for further analysis or operation compared to those of the previous set.

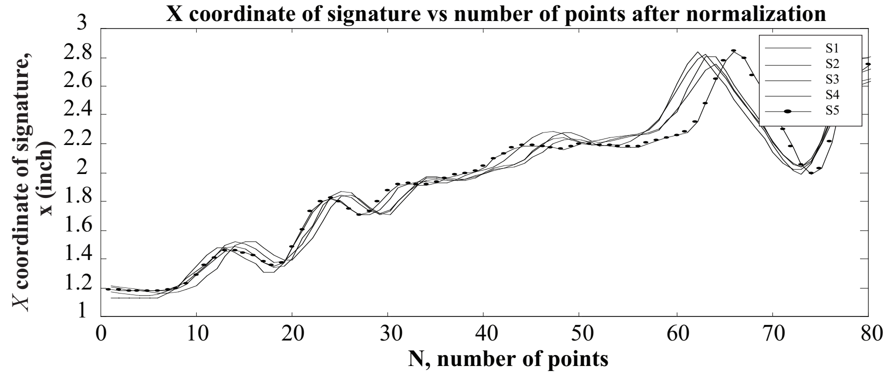


Figure 14 *X* coordinate of signature after preprocessing

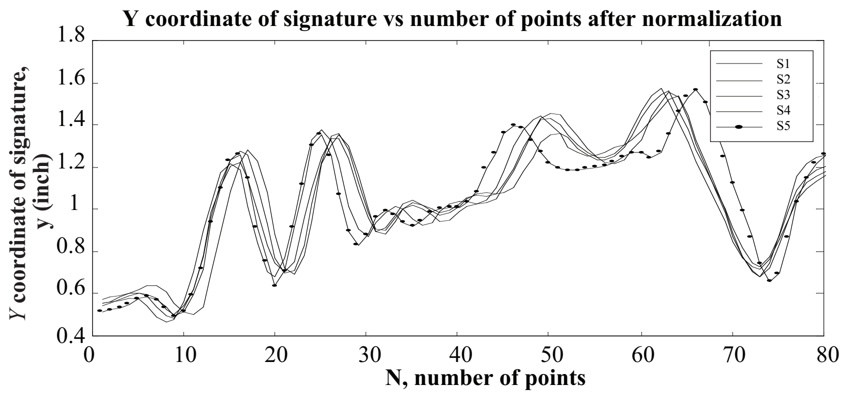


Figure 15 *Y* coordinate of signature after preprocessing

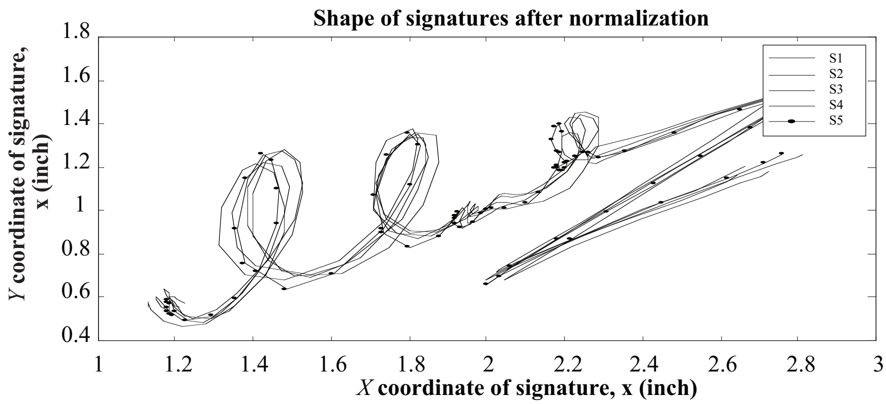


Figure 16 Shape of signature after preprocessing

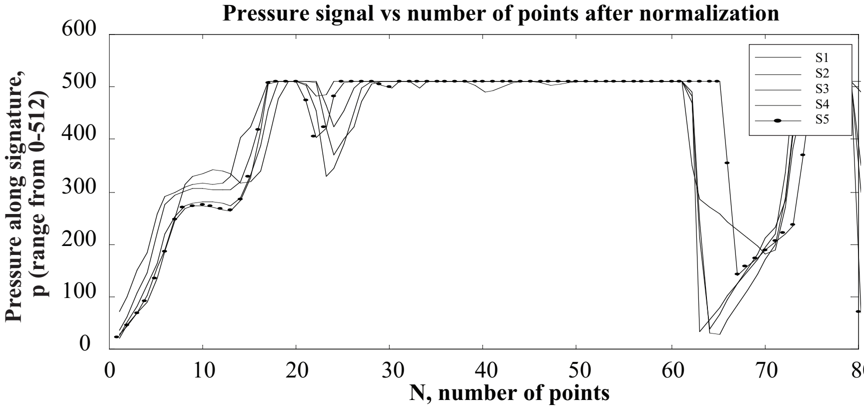


Figure 17 Pressure signal of signature after preprocessing

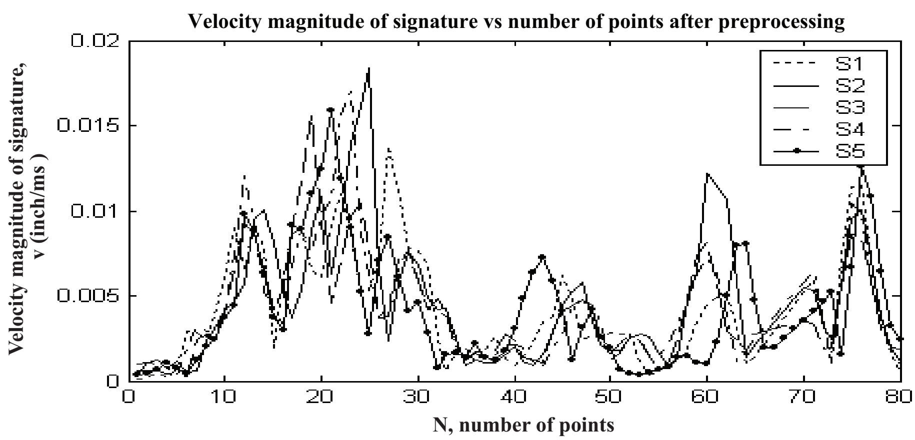


Figure 18 Velocity magnitude of signature after preprocessing

5.0 APPLICATION OF NEURAL NETWORK

In this study, the capability of the multilayer feed-forward (MLF) neural network to arbitrarily approximate any functions are exploited and applied [28]. Two neural networks of similar MLF structure were employed in the study to train and subsequently, test or verify the data. The first network deals with the computation of the pressure distribution data whilst the second estimates the velocity magnitude. For both networks, a set of 300 data pairs (input-output) was used to train the network off-line before it can be implemented in the verification process. The data from the fully trained network was stored in the database as reference signatures for future matching/verification operation.

Levenberg-Marquardt update or training algorithm has been applied to both networks and is expressed as follows [29]:

$$\Delta \boldsymbol{W} = (\boldsymbol{J}^T \boldsymbol{J} + \mu \, \boldsymbol{I})^{-1} \boldsymbol{J}^T e \tag{10}$$

where $\Delta \boldsymbol{W}$ is the weight change matrix
$\boldsymbol{J}$ is the Jacobian matrix of derivatives of each error to each weight
μ is a scalar quantity
e is an error vector
$\boldsymbol{I}$ is an identity matrix

A typical three layer network was used with *XY* coordinate assigned to the input layer, hyperbolic tangent sigmoid function used in hidden layer while linear function applied in the output layer before the target output value was computed. The relevant parameters of both the MLF networks are tabulated in Table 1.

Table 1 Neural network training parameters

Network parameter \ **Network**		**First network**	**Second network**
Input signals		*XY*-coordinates	*XY*-coordinates
Output signals		Pressure distribution	Velocity magnitude
Number of neurons	Input layer	$\frac{N}{20}$	$\frac{N}{20}$
	Hidden layer	20	20
	Output layer	$\frac{N}{20}$	$\frac{N}{20}$
mu (initial value for μ)		0.65	0.70
Goal		0.001	0.0075
Performance function		MSE (mean squared error)	MSE

Figure 19 shows the topology of the networks. It should be noted that the final network topology was achieved through a number of procedures as outlined in [30]. The networks were made to converge to the desired error goals based on some performance criteria as defined in Table 1.

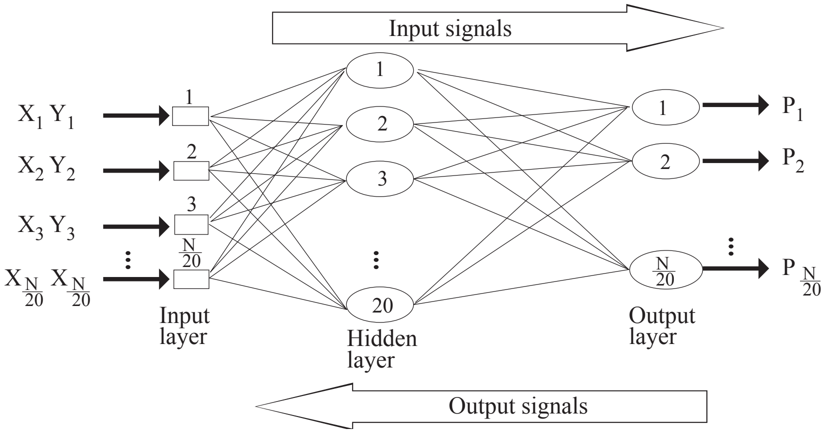


Figure 19 Neural network structure

6.0 VERIFICATION ALGORITHM AND PERFORMANCE EVALUATION

For the verification process, existing user is required to tell the system his/her registered identification number. MATLAB will then retrieve additional information like neural network weights and biases earlier trained off-line from the database. Upon complete loading of the user's information into the memory workspace, the user is required to manually sign his/her signature using the tablet and pen before verification can start. The input signal is immediately captured and sent to MATLAB for verification process. The verification involved three steps. The first step in verification is checking the signature time duration. This procedure is expressed as:

$$\tau_{mean} \times 1.15 > \tau_{current} > \tau_{mean} \times 0.85 \tag{11}$$

where τ_{mean} is the current signature time duration
$\tau_{current}$ is the mean time duration for all signatures

Once the signature time duration check has been performed, the *X* and *Y* coordinates and pressure signals will be preprocessed. The second stage of verification involves tolerance bands checking for *Y* coordinate, pressure distribution and velocity magnitude. The verification result is illustrated in Figure 20. The verification signature signals must be located within the bands to ensure only successful signature is accepted. The final step in signature verification is checking of the neural network simulated results on pressure distribution and velocity magnitude of the signature profile as shown in Figure 21.

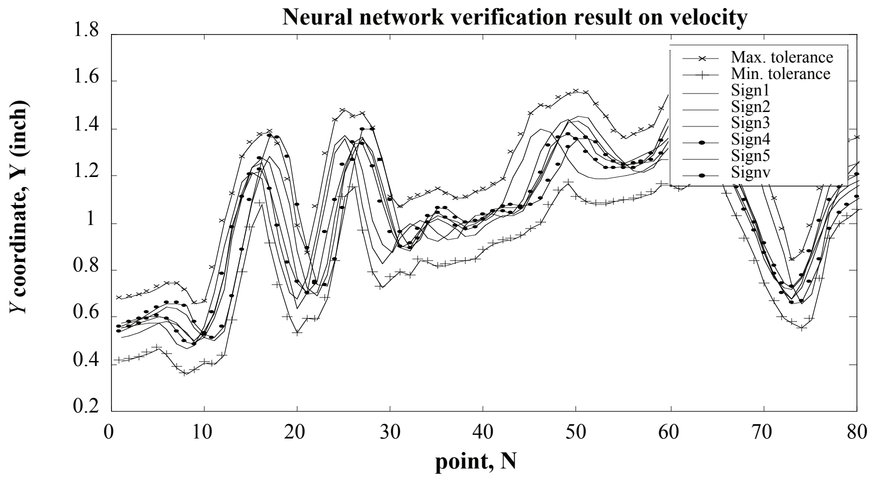


Figure 20 *Y* coordinate tolerance bands

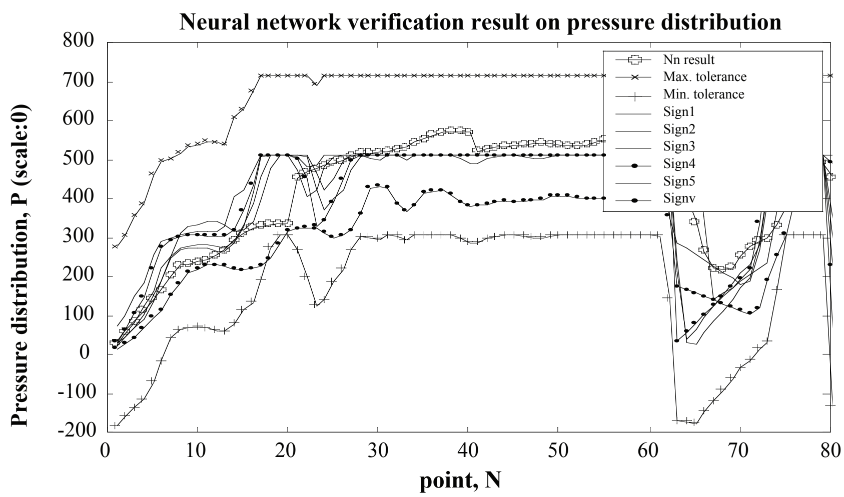


Figure 21 Neural network simulated result

Only the signature that manages to successfully pass all the steps will be verified as genuine signature. The successfully verified signature will be subsequently used as data for further training using neural network and ultimately saved into database as reference signatures. The oldest signature will be discarded to ensure only five sets of signatures to be taken as reference signatures. This feature is known as dynamically update of signature.

The performance evaluation of the system developed involved 150 genuine signatures and 150 simple and random forgery signatures. The system has achieved

1.33% of False Rejection Rate (FRR) and 0% of False Acceptation Rate (FAR). The results were made possible from 300 signatures from five users. However, only 150 simple and random forged signatures were used in the FAR evaluation. No public database and expert forged signatures are involved in the verification process and thus the error rates are artificially decreased.

7.0 CONCLUSION

A practical signature verification system incorporating feed-forward neural network has been successfully developed and evaluated. The development of the system includes data acquisition, database construction, application program interfaces creation, preprocessing of the signals, neural network training of the signatures and writing of verification algorithm. In spite of signatures variation, the registration and verification algorithms play a significant role to ensure the success of the verification. Various algorithms are involved in the preprocessing stage. Among them are normalization, common time axes creation, velocity magnitude calculation and signature validity checking on signals. Meanwhile, distinctive verification algorithms have been applied to increase the accuracy of the system. Tolerance band which is different from other ordinary verification algorithm has proved its effectiveness in verification of a user. Several features like retry on failure of submitting a genuine signature, dynamic update of signatures, sound effect, user-friendly GUI are introduced in the system.

REFERENCES

[1] Woodward, J., N. M. Orlans, and P. T. Higgins. 2003. *Biometrics*. CA: McGraw-Hill.

[2] Plamondon, R. and G. Lorette. 1989. Automatic Signature Verification and Writer Identification - The State of the Art. *Pattern Recognition*. 22(2): 107-131.

[3] LecLerc, F. and R. Plamondon. 1994. Automatic Signature Verification: The State of the Art - 1989 – 1993. *International Journal of Pattern Recognition and Artificial Intelligence*. 8(3): 643-660.

[4] Dimauro, G., S. Impedovo, M. G. Lucchese, R. Modugno, and G. Pirlo. 2004. Recent Advancements in Automatic Signature Verification. Proceedings of the 9th Int'l Workshop on Frontiers in Handwriting Recognition. 179-184.

[5] Zimmer, A. and L. L. Lee. 2003. A Hybrid on/off Line Handwritten Signature Verification System. Proceedings of the 7th International Conference on Document Analysis and Recognition. Brazil. 424-428.

[6] Nelson, W. and E. Kishon. 1991. Use of Dynamic Features for Signature Verification. *IEEE Transactions on Systems, Man and Cybernetics*. 201-205.

[7] Hamilton, D. J., J. Whelan, A. McLaren, I. MacIntyre, and A. Tizzard. 1995. Low Cost Dynamic Signature Verification System. Proceedings of European Convention on Security and Detection. Brighton Publisher. 202-206.

[8] Sakamoto, D., H. Morita, T. Ohishi, Y. Komiya, and T. Matsumoto. 2001. On-line Signature Verification Algorithm Incorporating Pen Position, Pen Pressure and Pen Inclination Trajectories. Proceedings of

IEEE International Conference on Acoustics, Speech, and Signal Processing. Salt Lake City (UT). 993-996.

[9] Tham, H. K., R. Palaniappan, P. Raveendran, and F. Takeda. 2001. Signature Verification System Using Pen Pressure for Internet and E-commerce Application. International Symposium on Software Reliability Engineering. Copyright ISSRE and Chillarege Corp.

[10] Ma, M. M., W. S. Wijesoma, and E. Sung. 2000. An Automatic On-line Signature Verification System based on Three Models. Proceedings of Conference on Electrical and Computer Engineering. Halifax (NS). 890-894.

[11] Wu, Q. Z., I. C. Jou, and S. Y. Lee. 1997. On-line Signature Verification Using LPC Ceptrum and Neural Networks. *IEEE Transaction on System, Man, and Cybernetics*. 27(1): 148-153.

[12] Martens, R. and L. Claesen. 1996. On-line Signature Verification by Dynamic Time-Warping. Proceedings of the 13th International Conference on Pattern Recognition. Vienna. 38-42.

[13] Martens, R. and L. Claesen. 1997. Dynamic Programming Optimization for On-line Signature Verification. Proceedings of the 4th International Conference on Document Analysis and Recognition. Ulm. 653-656.

[14] Lee, W. S. 1998. *A Neural Network Based On-Line Signature Verification Systems*. Ph.D. Dissertation. University of Detroit Mercy.

[15] Dullink, H., B. V. Daalen, J. Nijhuis, L. Spaanenburg, and H. Zuidhof. 1995. *Implementing a DSP Kernel for Online Dynamic Handwritten Signature Verification Using the TMS320 DSP Family*. France: Texas Instruments Incorporated.

[16] Hastie, T., E. Kishon, M. Clark, and J. Fan. 1992. A Model for Signature Verification. Technical Report. AT&T Bell Laboratories.

[17] Clark, R. M., T. Hastie, and E. Kishon. 1990. A Model for Comparing Signature. Proceedings of IEEE International Conference on Systems, Man and Cybernetics. Los Angeles (CA). 326-330.

[18] Shafiei, M. M. and H. R. Rabiee. 2003. A New On-line Signature Verification Algorithm Using Variable Length Segmentation and Hidden Markov Models. Proceedings of the 7th International Conference on Document Analysis and Recognition. 443-446.

[19] Oldham, D. R. 2001. *Biometric Identification for Dynamic Signature Verification Using Time Delay Neural Networks*. Ph.D. Dissertation Case. Western Reserve University.

[20] Zhou, R. W. and C. Quek. 1996. An Automatic Fuzzy Neural Network Driven Signature Verification System. IEEE International Conference on Neural Networks. Washington (DC). 1034-1039.

[21] Mohankrishnan, N., M. J. Paulik, and M. Khalil. 1996. Multi-Layer Neural Network Classification of On-line Signatures. Proceedings of the IEEE 39th Midwest Symposium on Circuits and Systems. Ames (IA). 831-834.

[22] Wessels, T. and C. W. Omlin. 2000. A Hybrid System for Signature Verification. Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. Como. 509-514.

[23] Jain, A. K., F. D. Griess, and S. D. Connell. 2002. On-Line Signature Verification. *Pattern Recognition*. 35(12): 2963-2972.

[24] Poyner, R. 1996. *Wintab™ Interface Specification 1.1:16- and 32- bit API Reference*. LCS/Telegraphics.

[25] Myers, C., L. R. Rabiner, and A. E. Rosenberg. 1980. Performance Tradeoffs in Dynamic Time Warping Algorithms for Isolated Word Recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*. 623 – 633.

[26] Lam, C. F. and D. Kamins. 1989. Signature Recognition Through Spectral Analysis. *Pattern Recognition*. 39 – 44.

[27] Nelson, W., W. Turin, and T. Hastie, T. 1994. Statistical Methods for On-line Signature Verification. *International Journal of Pattern Recognition and Artificial Intelligence*. 749-770.

[28] Hornik, K., M. Stinchcombe, and H. White. 1989. Multilayer Feedforward Networks are Universal Approximators. *Neural Networks*. 359-366.

[29] Demuth, B. D. 1994. *The Neural Network Toolbox for Use with MATLAB*. The Math Works Inc.

[30] Mason, J. C. and P. C. Parks. 1992. Selection of Neural Network Structures: Some Approximation Theory Guidelines. In K.Warwick, G. W. Irwin, and K. J. Hunt (Eds.). *Neural Networks for Control and Systems*. Peter Peregrinus Ltd. 151-180.