UNIVERSITY OF THE WITWATERSRAND, JOHANNESBURG

APPROACHING REAL TIME DYNAMIC SIGNATURE VERIFICATION FROM A SYSTEMS AND CONTROL PERSPECTIVE

Yi Gu

A research report submitted to the Faculty of Engineering, University of the Witwatersrand, Johannesburg, in fulfilment of the requirements for the degree of Master of Science in Engineering

Johannesburg, September 2002

Declaration

I declare that this project report is my own, unaided work, except where otherwise acknowledged. It is being submitted for the degree of Masters of Science in Engineering in the University of the Witwatersrand, Johannesburg. It has not been submitted before for any degree or examination in any other university.

Signed this _______ day of _________________________

____
Yi Gu

Abstract

This thesis outlines the development of a unique systems based dynamic signature verification algorithm. The origins of handwriting idiosyncrasies and habituation are explained using systems theory, and it is shown that the 2/3 power law governing biomechanics motion also applies to handwriting. This leads to the conclusion that it is possible to derive handwriting velocity profiles from a static image, and that a successful forgery of a signature is only possible in the event of the forger being able to generate a signature using natural ballistic motion. It is also shown that significant portion of the underlying dynamic system governing the generation of handwritten signatures can be inferred by deriving time segmented transfer function models of the x and y co-ordinate velocity profiles of a signature. The prototype algorithm consequently developed uses x and y components of pen-tip velocity profiles ($v_x[n]$ and $v_y[n]$) to create signature representations based on autoregression-with-exogenous-input (ARX) models. Verification is accomplished using a similarity measure based on the results of a k-step ahead predictor and 5 complementary metrics. Using 350 signatures collected from 21 signers, the *system's* false acceptance (FAR) and false rejection (FRR) rates were 2.19% and 27.05% respectively. This high FRR is a result of measurement inadequacies, and it is believed that the *algorithm's* FRR is approximately 18%.

Acknowledgement

The financial assistance of the National Research Foundation (NRF) towards this research is hereby acknowledged. Opinions expressed and conclusion arrived at, are those of the author and are not necessarily to be attributed to the NRF.

The author also wishes to acknowledge the invaluable inputs from my supervisors Dr. Charles Pritchard and Dr. David Rubin, as well as those from Prof. Brian Wigdorowitz, Mr. Michael Barker and Mr. Tumisho Matlala during the course of the research – your assistance is much appreciated!

Table of Contents

List of Figures

List of Tables

List of Symbols

$x(t)$, $y(t)$	:	Time domain functions of x and y Cartesian co-ordinates.
$v(t)$, $v_x(t)$, $v_y(t)$	:	Time domain velocity function in 2D, in the x direction and in the y direction respectively.
$a(t)$, $a_x(t)$, $a_y(t)$	:	Time domain acceleration function in 2D, in the x direction and in the y direction respectively.
$p(t)$, $p_x(t)$, $p_y(t)$	:	Time domain pressure function in 2D, in the x direction and in the y direction respectively.
$u[n]$, $x[n]$, $y[n]$ etc.	:	Sampled digital representation of $u(t)$, $x(t)$, $y(t)$ etc.
$\hat{v}_y[n]$	:	Predicted version of the y velocity component $v_y[n]$.
`arx`, `ident` etc.	:	MATLAB commands and function names.

Nomenclature

Allograph : A letter of an alphabet in a particular shape (as A or a).

Ballistic handwriting : Handwriting characterized by a spurt of activity without incorporating feedback.

FAR : False Acceptance Rate is the error rate resulting from the treatment of a forged signature as a genuine one.

FRR : False Rejection Rate is the error rate resulting from the treatment of a genuine signature as a forgery.

Discriminancy : Capability to distinguish between genuine and forged signatures.

Skilled forgery : Signature produced as a result of a forger's deliberate attempt in reproducing a genuine signature, in terms of both form and signature dynamics

Random forgery : A randomly chosen signature compared against a specific template. It needs to have no resemblance to the signature represented by the template whatsoever in terms of both shape and signature dynamics.

Inter-personal : That which is associated with different persons.

Intra-personal : That which is associated with the same person.

Test signature : Signature submitted for authenticity verification.

1. Introduction

Since the advent of the personal computer, computer-based automation has been an effective means of realising increased work efficiency and reduced human errors within various work and business processes. The process models have in turn undergone continuous optimisation over the years in a constant effort to achieve higher degrees of automation in order to remain competitive in the global market place. Yet, the drive towards full computer-based automation is accompanied by a need to reduce the number of paper-driven processes in the work and business environments. Consequently, there is currently a pressing need to increase the level of seamless integration between the electronic and paper-driven parts of work and business models, particularly with regards to the handling of handwritten signatures.

The established social acceptance of the handwritten signature as the primary form of legal attestation in particular has been a major hindrance in the aforementioned integration effort. It is often the case that handwritten signatures constitute a critical component of an otherwise fully automated process, yet the general lack of competent tools in the electronic handling of handwritten signatures has meant an inability to totally eliminate papers in semi-automated processes. In spite of this, research in the area of automated signature verification conducted over the past few decades, as well as the current availability of relatively inexpensive high-performance hardware and software components have made real-time Dynamic Signature Verification (DSV) particularly attractive as a widely deployable biometric technology capable of bridging the longstanding computer-paper gap.

1.1 Social Context of the Research

From the author's involvement in the market of DSV technology in South Africa, there is clear evidence of significant demands for DSV-based man-machine interface (MMI) products in the country's offices, particularly those within South Africa's prominent financial institutions. However, due to the total lack of local products, demand in this market is being capitalized upon by foreign firms based in developed nations. In view of this, this research, apart from its technical value, also stands as an initiative to develop a local product in support of the South African economy.

1.2 Introduction to the Systems and Control Paradigm Towards DSV

As is evident from its name, DSV is concerned with signature verification based on dynamic aspects of a person's signature, emphasising not only on a signature's appearance but also on the process of producing a handwritten signature in the time domain. Numerous DSV algorithms with varying quality of performance can be found in the literature. However there is a common underlay to almost all existing DSV algorithms examined in the current research. Specifically they deal with the problem of dynamic signature verification by manipulating and examining strictly the time domain signals of a signature, typically the sampled x(t), y(t) and perhaps z(t) (pressure) signals captured using an electronic tablet and/or a pressure-sensitive pen.

The focus of the research conducted was however the investigation of feasible ways to approach the problem of dynamic signature verification from a systems and control perspective, being based on concepts developed for Linear Time Invariant (LTI) systems. This notion implies that the signing process could be viewed as a plant in action, whose parameter values are to be inferred from the observation of the relationship between the plant inputs and outputs. The essence of the research hereby described was thus in exploring suitable ways in which the signing process could be modelled, the plant format identified, the plant parameters extracted and finally how verification could be implemented based on the plant attributes.

This approach is, at least in theory, superior to methods that are based predominantly on statistics and mathematical models, whose classification output relies strictly on the appearance of the plotted time domain signals. To appreciate the basis to this statement, consider the treatment of the signing process as the action of a plant modelled as a pure integrator 1/s, when an input in the form of an impulse δ(t) is applied to the plant, the output would be a unit step u(t), however if a unit step u(t) is applied to the plant, the output would be a unit ramp r(t) i.e. the following mapping is obtained for the plant:

$$\delta(t) \xleftrightarrow{1/s} u(t)$$
$$u(t) \xleftrightarrow{1/s} r(t)$$

Equation 1

If it is supposed that the plant $\frac{1}{s}$ represents a person, the output u(t) denotes a genuine reference signature, and r(t) denotes a genuine signature to be tested for authenticity, then it should be clear that examination of the shape of the time domain signals alone would indicate that r(t) is a forged signature, since u(t) and r(t) are completely different in appearance in the time domain. This type of verification mistake is far less likely to occur if a systems and control approach was adopted, since it is clearly evident that the underlying transfer function is identical in both cases i.e. 1/s. Consequently, although the form of the outputs maybe different, the signature r(t) would still be correctly classified to be genuine.

In this case, it is the LTI nature of the signature model that provides this independence to signal's shape for correct classification. The advantage hereof in practical terms is that the verification function would inherently handle intra-personal variations in time and space, thereby reducing the need to perform any signal alignment or warping for signal-to-signal comparison – processes often required in many existing DSV algorithms.

Based on the above philosophy, a number of research objectives were identified. Firstly, the research aimed to understand the origin of handwriting idiosyncrasies from a systems perspective, on the basis of which investigations can be then carried out regarding the application and development of systems concepts and tools to the problem of DSV. Secondly, the research should realise a DSV algorithm implementation based on the findings and results of the first objective. Finally, the performance and the practical viability of the systems-approach-based DSV algorithm developed are to be critically assessed.

From the comprehensive literature survey conducted, the areas of knowledge in signature verification and systems-and-control have seldom, if ever, been put into synergy. As a result, the research constitutes a valuable initial effort into the investigation of the advantages, disadvantages and the feasibility of a systems approach to DSV algorithms.

1.3 Organisation of the Dissertation

Having presented the general philosophy of research, Section 2 continues the thought by analysing and thereby identifying the origin of the uniqueness of writing habits in a systems framework. This is done by an analysis of human motion, and specifically by the presentation of the 2/3 power law, as well as its major consequences and implications. Section 3 precedes onto describing both the generic architecture of DSV algorithms, as well as the challenges facing algorithm design, whilst also introducing a number of essential evaluation tools and concepts pertinent to DSV algorithms.

Sections 4 and 5 discusses the selection of algorithm inputs and signature representations respectively, where critical analysis on each topic is presented based on the findings of the literature survey and independent experiments conducted by the author. The core design considerations and concepts of a systems approach to DSV is encompassed in Section 6, where the topics include specific application of plant models, the choice and evaluation of model structures and influence of model parameters on performance. This section constitutes the bulk of the expended time and effort in research.

Section 7 provides a discussion of the various types of classifiers commonly encountered, and provides an account of the development and operation of the classifier used by the DSV algorithm developed. Finally, the performance of the prototype DSV algorithm is assessed, analysed and benchmarked in Section 8.

2. Handwriting and Human Motion Analysis

By describing the context of DSV in the area of handwriting research, as well as the provision of requisite knowledge in human motion analysis, this section aims to develop a reader's appreciation of the problem at hand. This is necessary in forming a framework within which a precise problem definition for the research can be constructed.

2.1. Dynamic Signature Verification in Handwriting Research

The first step towards understanding the problem is to appreciate the context of signature verification in the broader realm of handwriting related researches, since a misnomer of the various functional objectives in this area can cause considerable confusion. In two comprehensive surveys conducted by Plamondon, Srihari [1] and Lorette [2], handwriting research can be classified into four main categories, namely:

- Handwriting recognition
- Handwriting interpretation
- Handwriting identification
- Signature verification

With handwriting recognition, the aim is to recognise and transform handwritten allographs into its symbolic representation e.g. from to a. Handwriting interpretation on the other hand goes one step further, in that the task involves the extraction and determination of meanings from handwritten texts e.g. interpreting a handwritten address, bank cheque reading.

In handwriting identification, the attention departs from the actual content of a written message, but is instead placed on the idiosyncratic characteristics of handwritten texts in identifying the author of a sample text – with the assumption that each person's handwriting is unique in style. Finally, signature verification is the task of determining whether a given handwritten signature was produced by the authentic owner of the signature.

The field of signature verification can in turn be further divided into two sub-disciplines. On the one hand is Static Signature Verification based solely on the image of a signature that may have been scanned or photographically captured. Research in this area relates predominantly to image processing techniques [3, 4]. On the other hand is Dynamic Signature Verification, which is verification based on the time domain characteristics of a signature and is considered to be a biometric technology. The latter is the area in which the research under discussion was conducted.

It should be evident from the above descriptions that of the four sub-categories within handwriting research, the first two are largely concerned with the output of the writing process, where the objective is to converge writings with individual variations towards a common, a-priori-known character set. Interest in handwriting recognition theory, algorithms, software and systems has grown rapidly in the recent years with the proliferation of hand-held Personal Digital Assistants (PDA) that use on-line recognition algorithms, and it is evident that such human-machine interfaces are becoming increasing popular and important due to society's natural familiarity with styluses.

The latter two sub-categories are however concerned with the idiosyncratic variations within writing samples. In the case of writer identification the aim is to diverge each sample text towards a specific element in a pool of elements each distinguished by a set of unique characteristics, while in the case of signature verification, the aim is in quantifying the correlation between signature specimens and thereby classifying a signature as either a genuine or a forgery. Although the application of writer identification tools has been largely limited to the field of forensics, a number of commercial imaged based (off-line) and dynamics based (on-line) signature verification software packages are available.

Signature verification software products have been employed predominantly in the banking and office environments, where they facilitate the attachment of non-repudiation to the authorship, review, approval and changes pertinent to hand-signed electronic documents, exactly as the case with signed paper documents. Although the current application of signature verification software are limited to document transactions – the traditional placement of signatures in society, the biometric nature of the technology could extend its application to also encapsulate security functions, such as building access control, computer logins etc.

2.2. Human Motion Analysis

Since the emphasis of the research was not strictly on the development of correlation techniques to compare time domain outputs of the hand writing process, but rather on the modelling of the signing process and characteristics as a plant, it was crucial to first investigate the nature and behaviour of the human neuro-mechanical system. The findings of the investigation would be instrumental in assessing what is and is not viable in subsequent attempts to derive a plant.

Considering that it is a well known fact that writing styles differ from one person to the next, a major motive behind the approach of the DSV problem from a systems perspective has been the hope to discover and capture the very entity that differentiates handwriting styles. Under this framework, the relationships between the spatial co-ordinate, velocity and pressure profiles of handwriting were investigated in turn and presented in this section.

2.2.1. The 2/3 Power Law

The first and most obvious question contemplated was 'Given that individual writers had different writing styles, what elements in the dynamics[1] of the writing system are then unique from writer to writer?'. A finding of particular importance encountered in numerous human motion researches was that of the empirical Two-Thirds Power Law [5, 6, 7, 8], which relates the tangential velocity of motion to the curvature of movement in all human motion – from the swing of arms to the production of handwriting.

$$v(t) = kR(t)^{\frac{1}{3}} \text{ or } v(t) = kC(t)^{\frac{2}{3}}$$

Equation 2 2/3 power law A

$$v(t) = k(t)\left(\frac{R(t)}{1+\alpha R(t)}\right)^{1-\beta}$$

Equation 3 2/3 power law B

Equation 2 and Equation 3 are two versions of the 2/3 power law encountered in literature. In both cases k or k(t) is a constant or function of proportionality and R(t) is the running radius of curvature of movement. C(t) in Equation 2 denotes the curvature function defined as 1/R(t), and in Equation 3, $\alpha \leq 0$, $k(t) \leq 0$ and $\beta \approx 1/3$ (so that $1 - \beta = 2/3$). From the above expressions, it is

[1] Dynamics as in not in the static features of a writing system e.g. the attributes of the bones, tendons and muscles, since even two persons with identical hand, wrist and arm attributes could have different writing styles.

evident that Equation 2 and Equation 3 are referred to as 2/3 power laws because of the value of the exponent.

Early studies on handwriting and drawing movements observed that there is a systematic relationship between the end-point velocity and the trajectory that it describes. This observation was quantified as the 2/3 power law, which has become widely accepted as an important invariant in biological movement trajectories [7]. Although studies on the 2/3 power law have been confined largely to planar drawing patterns of relatively small size, where the law is well obeyed (as verified by experiments in this research), a study conducted by Schaal and Sternad [7] as well as several others cited in their work have shown that the power law is not always accurate in movements of larger magnitude and with higher degrees of freedom. Schaal and Sternad hypothesized that instead of the law being a primary property of the central nervous system, it is likely that is it rather a by-product of more fundamental properties within the neurophysiological and biomechanic systems. Irrespective of the origin of the 2/3 power law, the velocity to trajectory relationship remains an important tool in the understanding of the signature process.

Two pieces of information can be gathered by simple inspection of Equation 2 and Equation 3. Firstly, the 2/3 power law indicates that the path and velocity of human movements are not only tightly but mathematically coupled, where if the path curvature is high, the tangential velocity of movement would be low and vice versa. Secondly, the fact that v(t) is directly proportional to $R(t)^{1/3}$ and not R(t) suggests that the amplitude of variation in v(t) is less amplified than in R(t), and thus an exponent term with value less than 1 is needed to attenuate the more bi-modal values of R(t).

In the research conducted, the definition of Equation 2 was adopted, firstly because of its simplicity, and secondly it correlated extremely well with experimental data. The particular experiment involved 10 individuals[2], each of whom were asked to produce natural cursive handwriting of varying size and at varying speeds on a pressure sensitive surface, which sampled x and y co-ordinate pairs at 600 Hz (further details of the hardware device are contained in Appendix 1). By calculating the Euclidean distance between successive sample points, the two-dimensional velocity profiles were obtained. For example, for the signature shown in Figure 1 the velocity profile obtained is as shown in Figure 2.

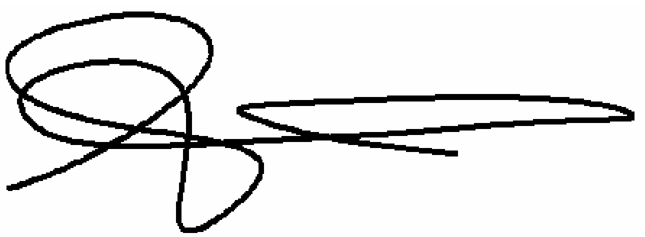


Figure 1 Sample signature

[2] Consisting of people of different sex, cultural background and handedness.

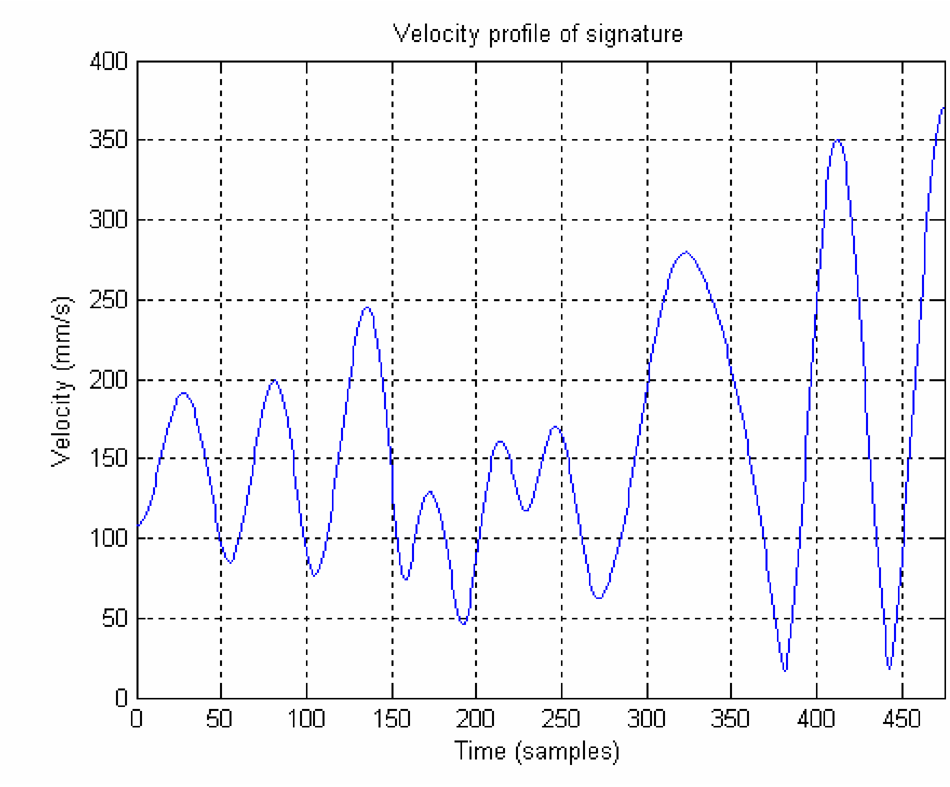


Figure 2 Velocity profile of signature

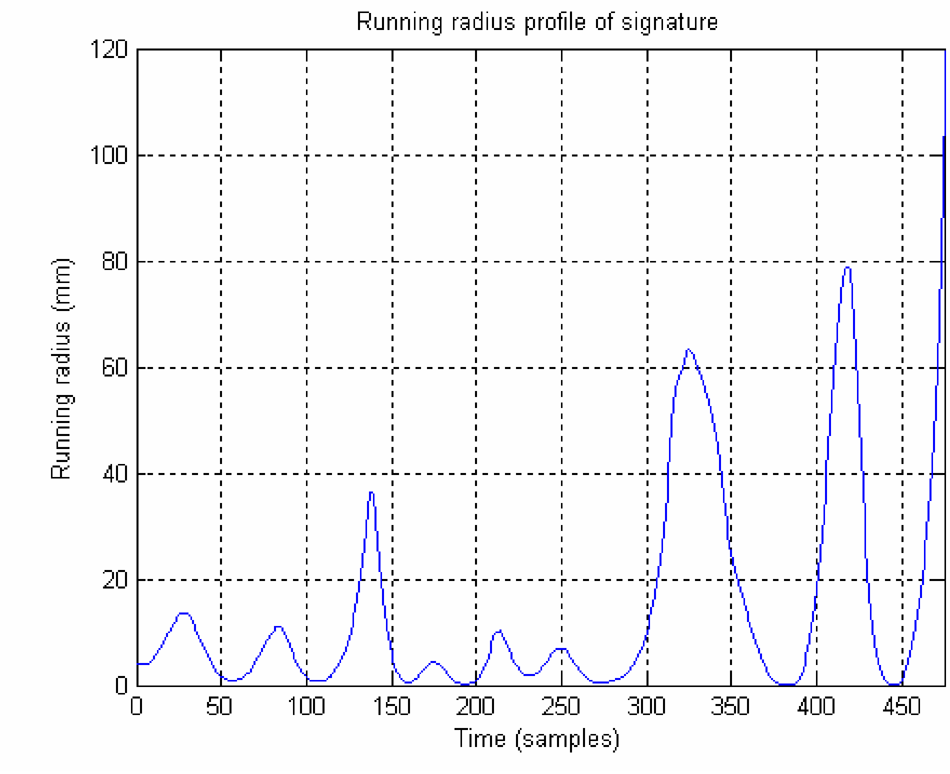


Figure 3 Running radius profile of signature

From the sampled x and y co-ordinate series, it is also possible to determine the running radius function of the signature, which was accomplished using a procedure of iterative arc fitting (for more details refer to Appendix 2). The resultant running radius of curvature as a function of time is shown in Figure 3, while Figure 4 shows the function after having been raised to the power of 1/3.

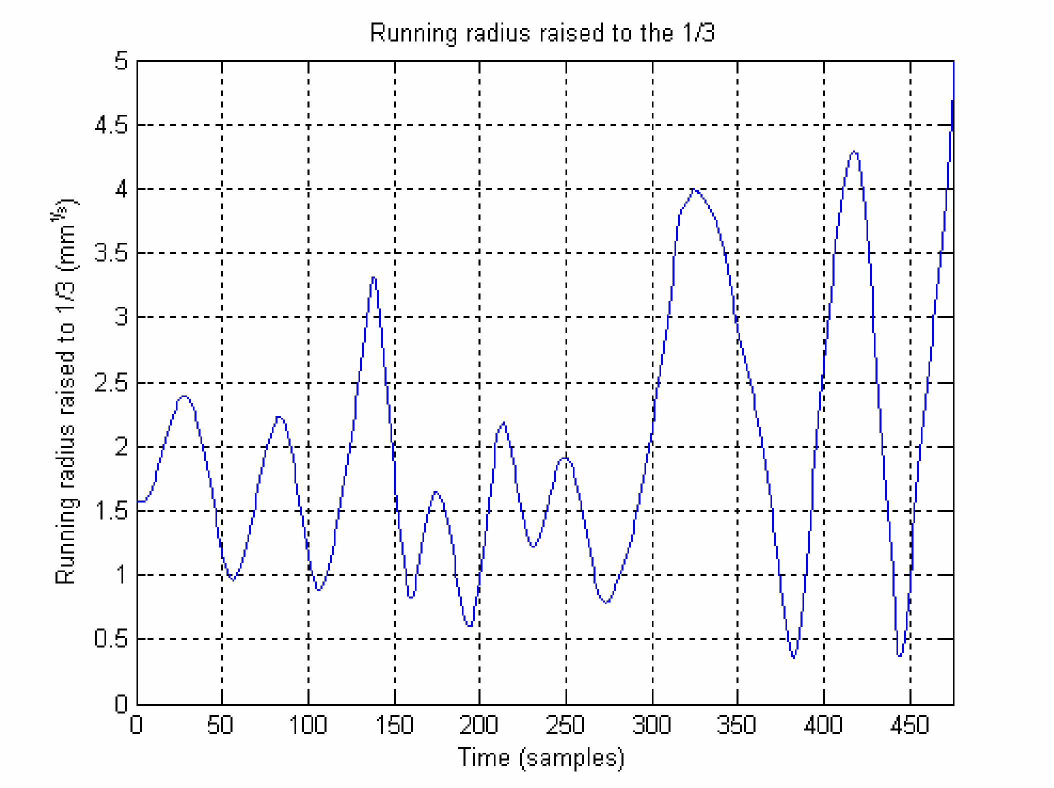


Figure 4 Running radius raised to the 1/3

It is clearly evident that the resemblance between the graphs of Figure 2 and Figure 4 are remarkable, differing only in orders of magnitude. This serves to verify the applicability of the 2/3 power law in the context of handwriting, while the fact that data obtained for all subjects yielded similar results suggests that the empirical relationship is universally applicable as expected. The only remaining term not yet discussed in the expression of Equation 2 is the constants k.

Since for any given piece of handwriting both v(t) and R(t) are the directly accessible and thus non-variable terms of the 2/3 power law, the nature of k was then studied. By direct division of v(t) by $R(t)^{1/3}$, it was observed that the value of k was not consistent across all values of v(t), and this raised questions into the nature of k. Specifically, is k a time invariant function of v(t), and furthermore could k(v) be the differentiating entity between individual writers?

In answering the first question, a mapping of k values to velocity values were performed for each writer, where each element within a row vector of k were bin-sorted according to the value of the corresponding element in a row vector v [3]. The statistical behaviour of values in each bin of k

[3] The row vector v is the data series representing the velocity profile of a particular writing specimen produced by one person.

was then observed. It was found that all k values corresponding to various ranges of v values exhibited a pattern of narrow scatter around a mean value, with one standard deviation[4] typically not exceeding 6% of the mean value. This implied that the mapping between k and v is fixed and thus independent of both time and context i.e. k is a time invariant function of v. A typical behaviour of a k(v) curve is shown in Figure 5 below[5].

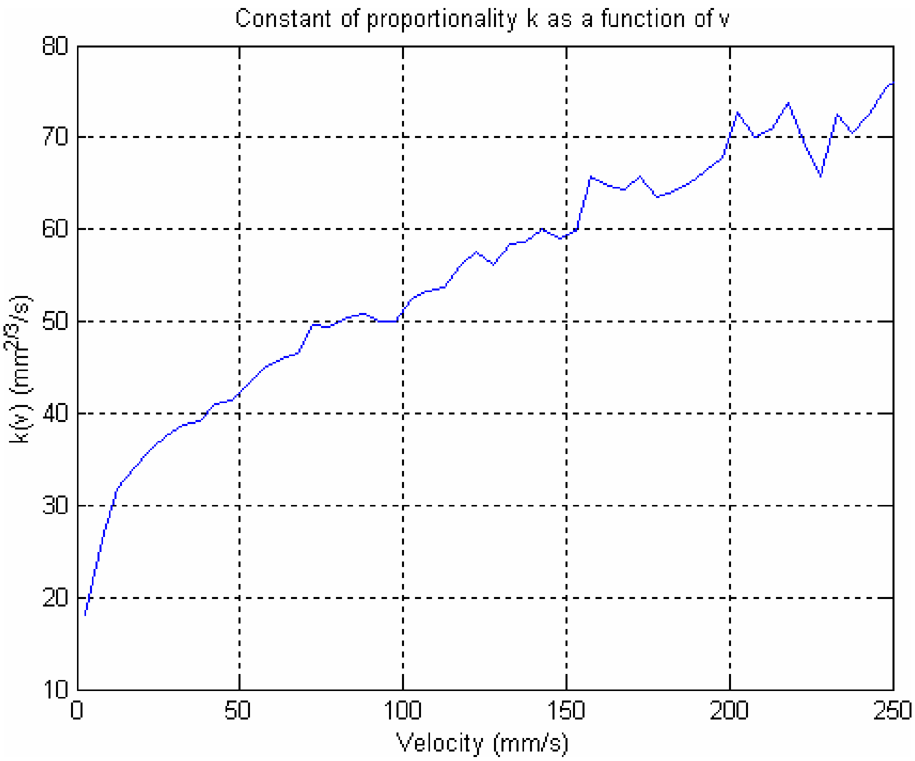


Figure 5 Typical k(v) curve

Figure 5 shows that k is a steadily increasing function of pen-tip speed, and the relationship is also almost linear beyond 20 mm/s. The initial steep rise in k below 20 mm/s is a result of the accuracy limit of the radius approximation method used, and it is expected that the true k(v) would be highly linear for up to at least 250 mm/s. The statement is based on findings that are presented in more detail shortly.

It should also be noted that firstly, the curve never intersects the origin in theory, since an arc of zero radius has no length and thus cannot have an associated velocity. Secondly, the graph does not extend to infinity for two reasons:

1. Velocity values much larger than those shown in Figure 5 are not commonly found in cursive writing and handwritten signatures, thus they fall beyond the scope of interest.
2. The 2/3 power law only holds for ranges of R and v natural to human motion, in particular the human hand, wrist and forearm complexes are unable produce near-perfect straight lines of ink associated with very large R and thus velocity values.

[4] Encompassing approximately 68% of the sample population.

[5] If k is a time-invariant function of v, then it is also a time-invariant function of R, since the 2/3 power law is a simple linear relationship between v(t) and $R(t)^{1/3}$.

Clearly, the investigation and extrapolation of k(v) into regions of higher v values would be of little value and meaning in the context of the current research, where the focus is on handwriting production.

To answer the question of whether k(v) could be the differentiating entity was a simple matter of plotting the k(v) graphs to all test subjects on the same set of axes and determining if they were sufficiently different from one another – Figure 6 shows the result of such an attempt. Clearly, by taking into consideration the effects of statistical scatter the k(v) profile to all 5 writers are essentially identical in each case, although the behaviour of k(v) beyond 140 mm/s is slightly more erratic – it is expected that should more and more writing specimens be taken, a more systematic scatter would result around the linear mean of the region.

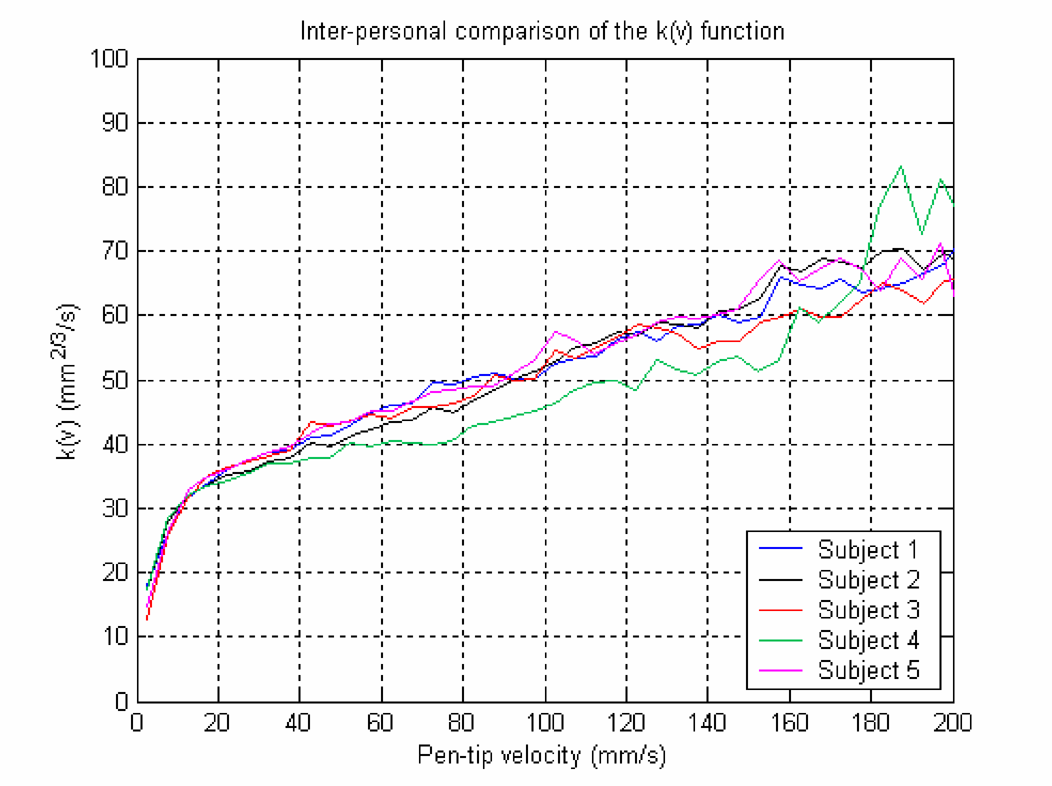
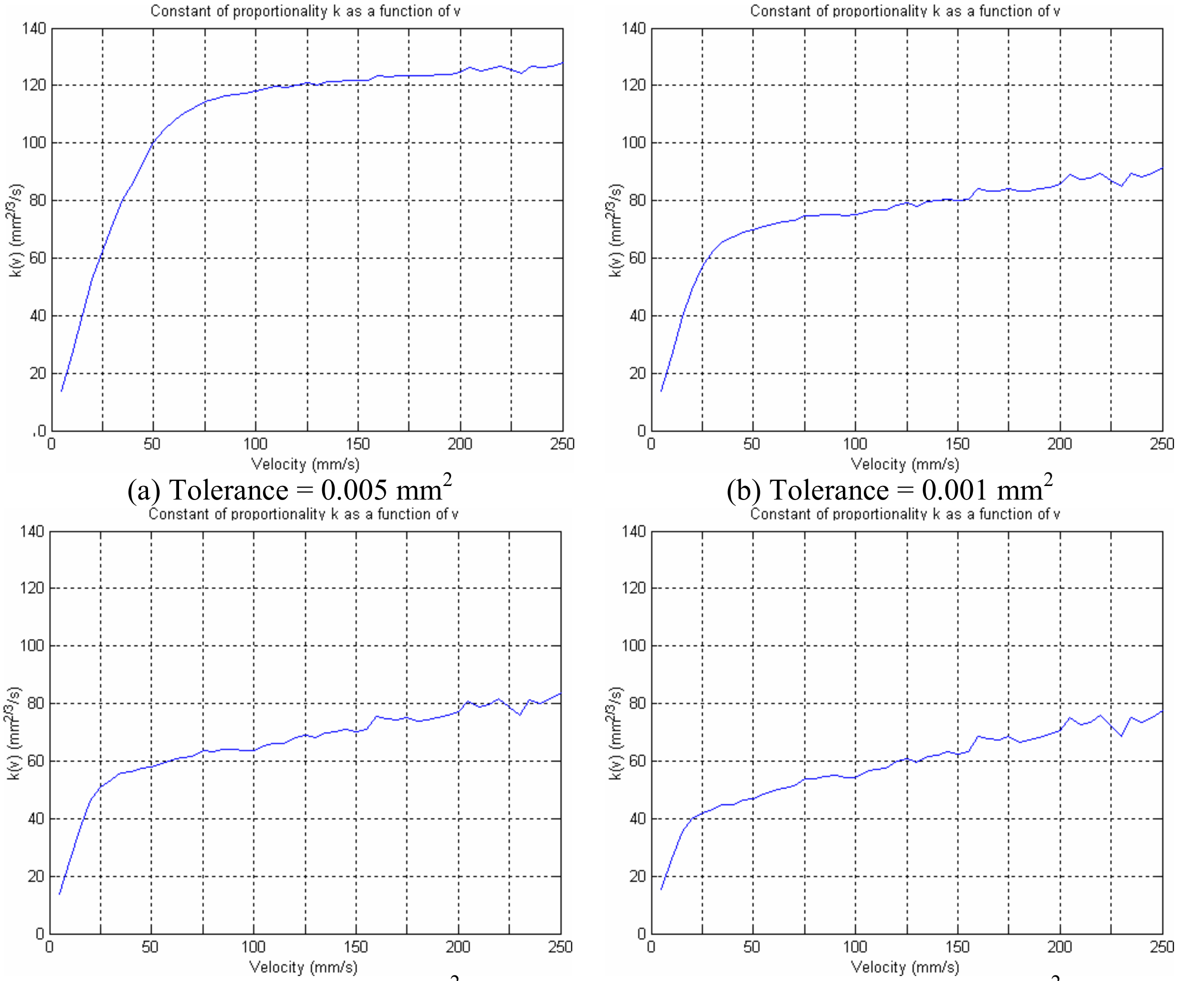


Figure 6 Comparison of k(v) of various writers

From Figure 6, it is evident that k(v) for all writers is in fact, at least approximately, identical for all writers, thereby suggesting that the 2/3 power law is not where the differentiating entity resides, since the same set of k values apply to the mapping of R to v for all writers. Note that the smoothness of the function reduces with increasing value of velocity. This is not the characteristic of k(v), but rather a consequence of the fact that the samples are captured in the time domain, where there are a lot more low velocity samples than there are high velocity samples, thus the size of the sample set reduces while the standard deviation increases as the velocity increases. In theory, k(v) should be an exact function with zero standard deviation.

During evaluation of k(v), it was also noticed that the tolerance allowed in the calculation of the running radius function R(t) had a notable influence on the appearance of the k(v) profile obtained. As explained in Appendix 2, the nth sample along a smooth path is closely approximated by finding a radius R, such that the area of the segment and the area of the triangle formed by joining the three consecutive x-y co-ordinate pairs are equal to within given tolerance levels. In particular, Figure 5 and Figure 6 were obtained when the tolerance was a mere 0.0001 mm^2.

At higher tolerance levels however, the initial steep rise of k with v becomes more and more pronounced. This is shown in parts (a) to (d) Figure 7, where the k(v) profile of Figure 5 was evaluated at tolerances of 0.005 mm^2, 0.001 mm^2, 0.0005 mm^2 and 0.0002 mm^2 respectively. Such behaviour seems to suggest that a higher accuracy would result in a lesser initial steep rise, and in the limit that the R calculation is exactly accurate, the steep rise would disappear altogether. This is the reason why it was stated earlier that the true k(v) is expected to be highly linear for up to at least 250 mm/s.



(a) Tolerance = 0.005 mm^2

(b) Tolerance = 0.001 mm^2

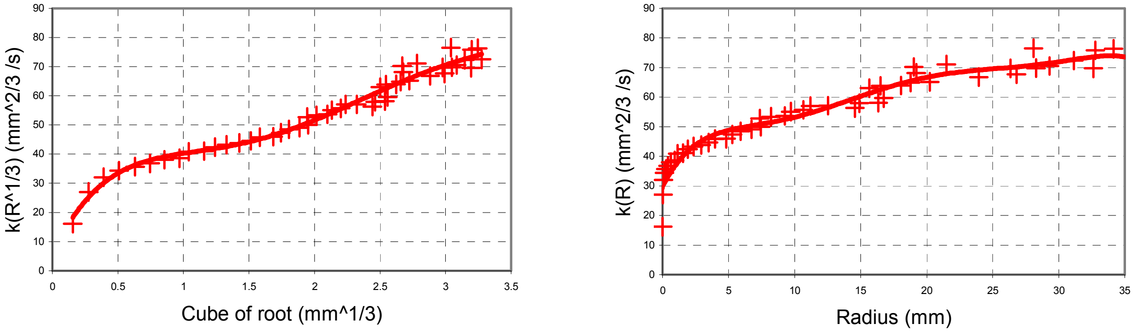
(c) Tolerance = 0.0005 mm^2

(d) Tolerance = 0.0002 mm^2

Figure 7 The influence of tolerance allowed in evaluation of R on k(v) profile

Bear in mind however, that regardless of the tolerance used and the specific k(v) profile obtained, the issue of true importance is that the 2/3 power law is obeyed in an identical manner by all writers as indicated by Figure 6.

Since the 2/3 power law is a linear function in $R^{1/3}$, the relationship derived for k(v) can be used to used to plot the graph pf k as a function of R i.e. first obtain a mapping between k and $R^{1/3}$ and then R in turn – the result of such action is shown in Figure 8, plotted using data gathered from the experiment mentioned previously. Note in particular the steep initial increase in k with small increases in R.



(a) Constant of proportionality k as a function of $R^{1/3}$

(b) Constant of proportionality k as a function of R

Figure 8 Relationship between k and R

Now, since both the applicability of the 2/3 power law in handwriting analysis, as well as the consistency of k(v) from writer to writer has been established, it can be stated, with a reasonable degree of confidence, based upon the evidence and discussions presented in this section that t*he relationship between the shape and velocity profiles of handwriting specimens is completely and exactly defined by Equation 2 for all writers.*

This is a remarkable finding, since it indicates that the spatial and time characteristics of handwriting do not exists independently, instead they are universally governed by an underlying function that links the variables in the space and time domains – a function that is obeyed by every person without discrimination. This is analogous to the discovery of the laws of gravity, which provided a complete model of the relationship between the behaviour of moving objects in the space and time domains. As a consequence of this discovery, a further number of crucial points of the research followed.

2.2.2. The Signature Process as a Controlled System

The findings associated with the 2/3 power law suggest that the differences in handwriting of people do not stem from differences in the governance of handwriting dynamics, but rather from more complex and subtle differences in a hidden biological motor control system. In this thesis, the analysis of the motor system necessitates the presentation of the proposed systems-and-control model of the signature process.

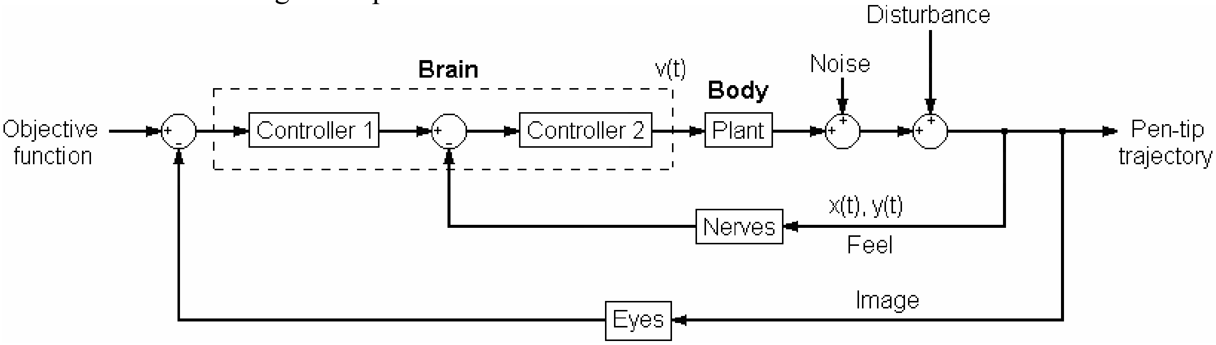


Figure 9 A systems abstraction of the writing process

At the *motor coordination level*, the writing process begins with the writer envisaging, either explicitly or implicitly, an objective function i.e. an allograph or an image of a word. Since the processes of progression of expression, spelling, grammar, punctuation, word selection etc. associated with normal writing would occur further 'upstream' in the writing process, both handwriting and signature production can also be incorporated into the above model.

In realising the objective function, it is fed into the controller[6] that corresponds to the motor cortex of the human brain, which in turn actuates the plant i.e. the combination of the limb, wrist and hand complexes. In this case, the actuating signal is the firing of neurons at specific intensities and durations in stimulating the various muscles so that the envisaged handwriting or signature pattern is produced.

Now consider the life cycle of a practical controller, the first stage in the design of which is the gathering of plant information e.g. response times, bandwidth, gain, saturation, so that the set of 'best' control strategies can be formulated to actuate the plant to achieve the desired output – in this case the desired appearance of handwriting or signature. By the same token, the plant parameters in this case include the size, flexibility, mass and limitations of the bones, tendons and muscles in the writer's fingers, hand, wrist and forearm, which are naturally unique from person to person i.e. each person embodies a unique plant.

Once more, just as with real-world controllers, the task of the conceptual controller in the motor cortex is also that of producing the desired output with its knowledge of the plant characteristics i.e. given an objective function, the motor cortex begins firing its neurons at specific intensities and durations in actuating various muscles so that the envisaged pattern is produced. In an evaluation compiled by Plamondon and Maarse [9], a number of motor models of handwriting were presented that correspond to the plant component of the proposed model, and several points of note and findings thereof are of particular interest.

Firstly, the results of simulation experiments conducted as a part of the evaluation indicated that it is highly likely that the actuating signals from the motor cortex to the muscles exist in the velocity domain. In other words, the muscles used for handwriting are co-ordinated by neural signals whose intensities correspond to desired velocity values. This is why the path between the controller and the plant in Figure 9 is labelled v(t) – not to be confused with $v_x(t)$ and $v_y(t)$ of pen-tip motion.

Secondly, it was also acknowledged that neuron firing intensities and durations were generally highly difficult to quantify. As a result, the choice of the actuating function in various studies has instead been guided by the established practice of modeling the neural inputs as step inputs of varying amplitude and duration [9]. The significance hereof is that, although we may know the form and meaning of actuating signals, they remain largely inaccessible both physically and numerically; as a result of this, the model proposed in Figure 9 cannot be taken at its face value, since after all, much thereof remains abstract in nature.

Thirdly, it was noted that although no physical evidence was cited that analytically supported such a view, all of the models analysed in [9] were LTI systems representable by a Laplace transform. In the opinion of the author, this stemmed purely as a result of convenience and simplicity, since the theories and principles of LTI systems has long been established and understood. Nonetheless, Plamondon and Maarse [9] showed that if the LTI model approach

[6] Controllers C1 and C2 cascaded.

were adopted, and that the inputs represented intensities and durations of neural signals (the so-called neural firing domain inputs), then an LTI model of at least third order would be necessary.

Returning to the model of Figure 9, the handwriting motor system, like many real-world systems, also suffers from uncertainty, so that its behaviour given identical actuating inputs may differ from time to time. For instance, if the signer were in an awkward posture, or shivers due to cold or has an hand injury, then the quality of the outputs x(t) and y(t) would be deteriorated compared to the plant's standard level of performance, even though the controller may try its best to realise the objective function.

With reference to the 2/3 power law, the plant is also the part of the model at the output of which the law is observed, and it is remarkable that irrespective of the objective function and irrespective of the variations in the plant parameters, the controller invariably actuates the plant in such a way that the dynamics of the plant outputs x(t), y(t), x'(t) and y'(t) obeys the 2/3 power law, without failure.

One reason for such behaviour may be that the motor cortex (the controller) has inherent knowledge of the 2/3 power law as well as the plant, and purposefully generate neural signals that would result in outputs that are in conformance to the 2/3 power law. In this case, it may simply be a matter of coincidence that the 2/3 power law is observed in human motion, since the motor cortex could then in fact have chosen any arbitrary law, or no law at all with regards to motion dynamics.

Alternatively, it could also be that the controller has no knowledge of the 2/3 power law, and it is instead a result of the inherent physical limitations of the plant irrespective of the actuation signal [7]. Whatever the case, the 2/3 power law is after all an empirical relationship whose validity is irrefutable in handwriting, and empirically was exactly how it was used in this research.

The writing system also incorporates feedback via two main sensory loops. Most obviously, the visual image of the writing produced is captured via the visual channel of the eyes back into the brain. This is applicable when the writer has to pay special attention to the size, proportion, location or orientation of a signature being signed. A further function of the visual feedback loop is in the assessment of inking patterns, since a pen that is not inking properly may require a deliberate increase of pressure by the writer for example.

The second feedback loop consists of the sensory nerves within the fingers, hand, wrist and forearm complexes, via which the writer remains aware of his own posture even with his eyes closed. Such postural information affords the brain the competency to produce legible writing without visual feedback. An implication hereof is that handwriting can in fact never be totally ballistic, since there would always be postural feedback to the brain whether the writer acknowledges it explicitly or not. In addition, there remain other types of information that can only be sensed via the nerves, including the texture of the writing surface, the 'feel' of the writing instrument, writing friction, inconspicuous unevenness in the writing surface etc. all of which are important external factors that influence the control action.

The final elements in the proposed model are the output noise and disturbance inputs. Noise models the variability inherent of biological systems, and in particular Harris and Wolpert [6] proposed that the neural control signals are corrupted by noise whose variance increases with the size of the control signal, and the shape of the final trajectory is selected to minimize such variance [6]. Disturbance on the other hand encapsulates any environmental factor that leads to the deterioration of signature quality relative to the objective function. For instance, the paper on which the signature is placed may slip, or the writing surface could be uneven or vibrating etc. In any case, such disturbances would be registered via the feedback loops at the brain, which would in turn attempt to compensate for the disturbances.

2.2.3. The Differentiating Entity

The prior paragraphs have outlined the various components of the proposed signature producing system. Although the introduction of the systems model of the writing mechanism is in itself significant, its real value is however in its purpose as the stepping-stone towards a more insightful understanding of the real focus of this section, which is the entity that differentiates the writing styles of writers. A theory is hereby proposed based on the systems model presented above that would explain the variations in handwriting styles, the difficulties in the forgery of signatures and ultimately locate the differentiating entity itself.

In studying the cause and development of individual writing styles, Smith [10] identified a number of factors in determining a person's eventual style of handwriting, under the assumption that most people nowadays are taught to write in a formalised and group setting, such as in a classroom with a relatively large number of students. Smith noted that even during the learning process, the writing of the student already starts to exhibit individual deviations from the writing of the teacher, with the main influencing factors being [10]:

1. The geography of the classroom
2. Seat assignment of individual students
3. The physical dimensions of the writer's fingers, hand, and arm
4. The dexterity thereof
5. The posture of the learner
6. The way the student held the writing instrument
7. The feel of the writing surface
8. The feel of the writing instrument
9. The learner's mental state during the learning process.

The above of course is not taking into account that all the students would be copying the handwriting of the teacher, which may in itself carry individual trait marks of the teacher.

It is the combination of the above factors and variables along with the resultant written product, that constitutes a student's first and lasting impression of a writing task. Learning theory dictates that as one begins a new task, such as writing, that task begins to form a matrix of ordered steps in the mind of the learner, which then becomes *fixed* in the mental processes of the person practicing this task repeatedly. In time, when the learner has completed numerous successful writing tasks, the matrix of patterns then becomes somewhat automatic of the person, since the more times the task is performed, the deeper the pattern becomes imprinted on the learner's subconscious. At this point, the mental processes and the physical co-ordination of muscles and nerves would also have assimilated into a single idiosyncrasy [10].

The result of all of the above influences and processes is the subconscious *habitual* writing patterns of the author. It should thus be clear that the skill of writing is a consequence and a function of the mental processes of perception and the physical ability of the writer. These, in conjunction with the variables listed above, imply that the writing style of an individual is unique, permanent and thus a biometric attribute of a person, which is highly unlikely to be identical to that of another person.

Viewing the above findings of Smith in the context and terminology of a systems view of the writing process, the learning process is then essentially a process of conditional synthesis of the controller. Refer to Figure 10.

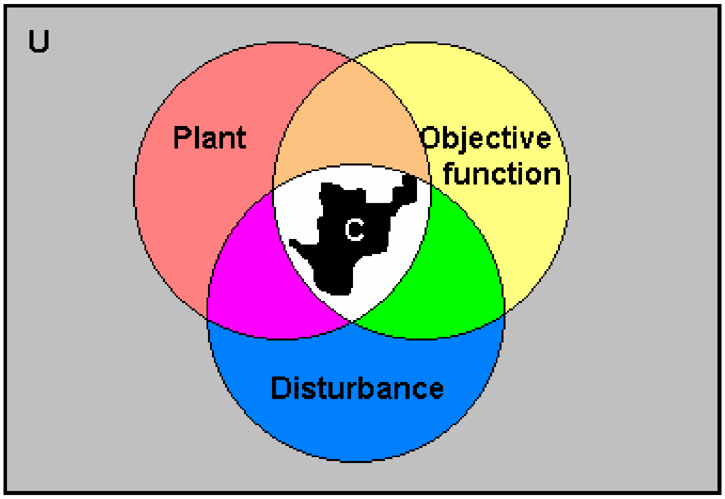


Figure 10 Conditional controller synthesis

Consider a universe U of all possible handwriting motor control strategies in the control strategy space. Since each writer embodies only a subset of the three universal sets of plants, objective functions and disturbances, the learning experience of a given writer forms the intersection of the sets that circumscribes the region of applicable control strategies[7]. Clearly, the eventual nature and characteristics of the controller are immediately functions of the subsets of plant characteristics, objective functions, and external disturbances, which correspond to points 1 to 2, 3 to 6 and 7 to 8 of Smith's list respectively.

Depending on the true origin of the 2/3 power law, two scenarios exist in the definition of the region of feasible control strategy. In scenario 1, it is assumed that the 2/3 power law is an explicit property of the central neural system, and in scenario 2, the 2/3 power is treated as an inherent property of the combination of the controller and plant modules – a view supported by Schaal and Sternad [7]. Since the origin of the 2/3 power law has so far not been exactly identified, and the aim of the current research is not in identifying the origin thereof, explanations regarding Figure 10 for both scenarios are provided below.

If scenario 1 is true, then only a subset of the control strategies within the region of intersection is suitable, since the combined behaviour of the controller and plant has to obey the 2/3 power law – this is represented by the black region labelled C in Figure 10. This subset of control strategies then embodies the characteristics of the customised controller of a writer.

If scenario 2 is true however, then the region of feasible control strategy (region C) would occupy the entire region of intersection, with the implication that variations in the shape and location of the region of intersection alone is sufficient to account for the various individual writing styles. The fact that region C occupies the entire region of intersection in this case suggests that all control strategies within this region are feasible i.e. the 2/3 power law is an inherent property of the human biomechanical system, and it will be obeyed regardless of the specific control strategy employed.

[7] The universal sets are drawn as circles for simplicity sake.

The existence of confining intersections and the possible selection of subsets enclosing feasible control strategies makes the synthesis of the motor controller conditional as opposed to general.

Under this framework, the synthesis of the biological controller corresponds to the learning and training process undergone by the learner. In this phase, by systematically incorporating the characteristics of the plant that is being gradually discovered, as well as the enforcement of the 2/3 power law, the biological controller iteratively and simultaneously finds the boundaries of both the region of intersection and the region C. On the same token, the increase in a learner's proficiency in writing and the eventual habituation thereof correspond to the consolidation of the boundary of region C within the previously inept controller.

The above analogy indicates that the motor cortex of the human brain is in fact a highly customised controller towards handwriting production, with the boundary of region C ensuring the adherence to the 2/3 power law – either implicitly or explicitly. Due to this fact, it would be significantly difficult for a writer established in the skill of writing to mimic the writing style of another writer flawlessly, since the region C of the forger would not have been optimised for a different set of plant characteristics and objective functions. This is precisely what makes handwritten signatures the traditionally accepted way of verifying the authenticity of legal documents.

From the preceding analysis, it is clear that all evidence point to shape of region C in the strategy space of Figure 10 as being the entity that differentiates writers, since the biological synthesis of the motor controller has been seen to ensure that the shape of region C for each writer would be unique. In addition, the origin of variations in handwriting styles or the source of the uniqueness of a controller, can be traced back to the differences in the shape of the region of intersection between the subsets of plant characteristics, objective functions and disturbances pertinent to an individual writer. Now since the traits of handwriting dynamics are inevitably carried over into the domain of signatures, the unique dynamics associated with a signature constitutes a biometric attribute of a person, and the difficulty in forging a signature can also be easily understood.

The above reasoning concludes the identification of the differentiating entity in handwriting. In hindsight, an understanding of the nature and implications of the 2/3 power law, as well as a systems approach to the modelling of the signing process have both been critical components in the search of the differentiating entity.

2.2.4. The Challenge Facing Forgers

A second major finding in this thesis, following the discovery of the profound implications of the 2/3 power law, was that: given that a forgery of close resemblance in shape to the genuine signature was written with natural[8] rhythm, the velocity profile of the forgery would inevitably be similar to that of the genuine signature. This is true because of the 2/3 power law, and in such cases the time domain characteristics would offer little security if submitted to a DSV algorithm.

In view of the above, the one and only question all forgers should ask is 'Can I produce a good imitation of the genuine signature in *shape* with a *natural rhythm* at all?'. The reason for this is because if the answer is 'Yes', then the security of a DSV algorithm would already have been successfully circumvented, and any fear of the rejection of the forgery would be unfounded.

[8] Naturally in the sense of not tracing, and not requiring tremendous deliberation during the task, but casually performed with natural fluency.

From a systems perspective, the problem of forgery is in the application of a foreign objective function (corresponding to the signature being forged) to a controller-plant combination that is not customised for the given input, and this is likely to result in a signature of inferior quality. For example, where certain strokes in a signature require the forger's fingers or hands to move in a way that he or she is not accustomed to, the forger's region C may not be adequate in actuating the plant to produce that stroke in a habitual way, and this would in turn degrade the quality of the final product i.e. the forger's region C is incapable of realising the objective function.

In summary, the crux of the finding is that natural production of the target signature is the only challenge facing a forger, which if surmounted, no grounds would exist for the dismissal of a forgery as not being a genuine signature (because of the 2/3 power law). This highlights the fact that the only security associated with DSV algorithms lie within the differences in the shapes of individual C regions, and in particular the region C of the owner of the genuine signature, and the region C of the forger – which of course is a variable.

2.2.5. Jitter in Traced Signatures

The third significant finding was a logical deduction following the 2/3 power law. It was realised, as well as experimentally confirmed, that any pen-tip trajectory associated with tracing action is bound to suffer from jitters, which are small yet frequent deviations from the target path in the traced image that exceeds the quantisation errors of the hardware used to capture the on-line signatures

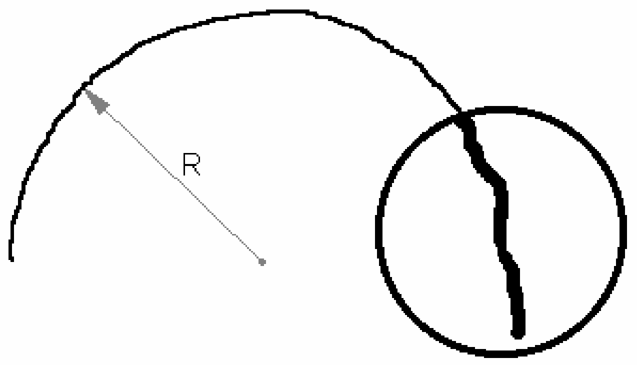


Figure 11 Jitter of slow drawing action

In an experiment into the examination of violations of the 2/3 power law, arcs as the one shown in Figure 11 were produced at disproportionately low pen-tip speeds, and it was anticipated that the R(t) would assume values close to R, whilst v(t) would be consistently low. To the contrary, while the v(t) values were indeed low, the running radius function R(t) contained only very small values corresponding to the tight curvatures of individual microscopic jitters instead of values close to R.

Such experimental results seems to suggest that *the 2/3 power cannot in fact be violated* i.e. a low velocity must correspond to small radii, and so it is physically impossible to produce a microscopically smooth curve from tracing or slow drawing actions.

A physical interpretation of jitter provided by Nalwa [11] is that when a forger attempts to trace a pre-existing curve closely, jitter is produced as a result of a constant effort to correct the pen trajectory to conform to the curve being traced. This explanation however does not account for the jitter found in slow drawing motion not associated with tracing, and again the 2/3 power law forms the only plausible explanation.

While the velocity v(t) is usually a function of R(t) in the 2/3 power law and not the converse i.e. people are more concerned with what they write than how they write, the above experiment however adopted the inverse relationship of R(t) being a function of v(t), and the results show that the 2/3 power law is in fact a reciprocal relationship i.e. a tight curvature dictates low velocity, and a low velocity dictates tight curvature. As a result of this reciprocity, a measure of jitter along with the duration of signature time form a powerful combination of metrics in the detection of tracing and thus forgery in the forensic examination of signatures.

2.2.6. Reciprocity of the 2/3 Power Law

A further and final significant discovery based on the 2/3 power law was that the velocity profile of writing is, at least in theory, completely deducible from the spatial image of a handwriting specimen, which contains all curvature information. In other words, by visual inspection of an ink trace, the velocity profile associated with the trace can be completely defined.

However, a subtlety associated with the deduction of v(t) from an image is that because the image exists in space, while the information to be extracted exists in the time domain, a transform would be necessary in the correct representation of space information in the time domain. In particular, the extraction of v(t) from R would begin with parameterisation of the R function over the path length *l* of the image i.e. R(*l*), and from R(*l*), v(*l*) is deducible using the 2/3 power law. The aforementioned transformation involves the re-parameterisation of v(*l*) into v(t), which requires the sampling of v(*l*) at equal time intervals rather than equal length intervals, and this is possible since v(*l*) inherently contains time information.

A second point of note on the topic is that certain sampling criterion has to be satisfied in the correct extraction of v(t) from R(*l*). Given that the least number of points needed in a plane to approximate a two-dimensional non-rectilinear trajectory is three, there must be a lower limit to the tightest curve the points can accurately represent. Accordingly there must also be a theoretical limit on the resolution of v(*l*) given a particular equi-spaced sampling resolution over *l*.

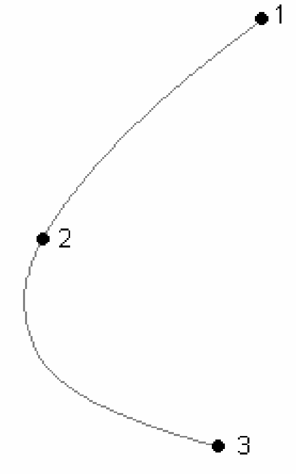


Figure 12 Lossy sampling

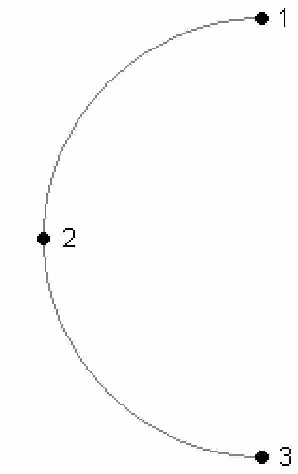


Figure 13 Lossless sampling

Consider three sampling points equi-distant over *l* as shown in Figure 12 and Figure 13. In the former case the sampling over *l* is lossy, since between samples 2 and 3 there is a tighter curve that is not captured and thus cannot be accurately reconstructed. Note also that in Figure 12 the trajectory contains multiple radius components. In Figure 13 however, the curve being sampled is a pure curve with only one radius component, thus the samples in this case do not approximate the image, but exactly represent the image, since one would be able to reconstruct the original trajectory if given the three samples.

What the above points to is that the smallest curvature that can be represented by three consecutive samples in lossless sampling is if the samples lay on an arc of constant radius. Consequently, from simple geometry it is easy to show that:

In order to ensure lossless sampling of an entire image trace, the samples can be separated by at most rπ/2 units from each other, where r is the smallest radius component in the sampled image.

2.2.7. Ballistic Motion

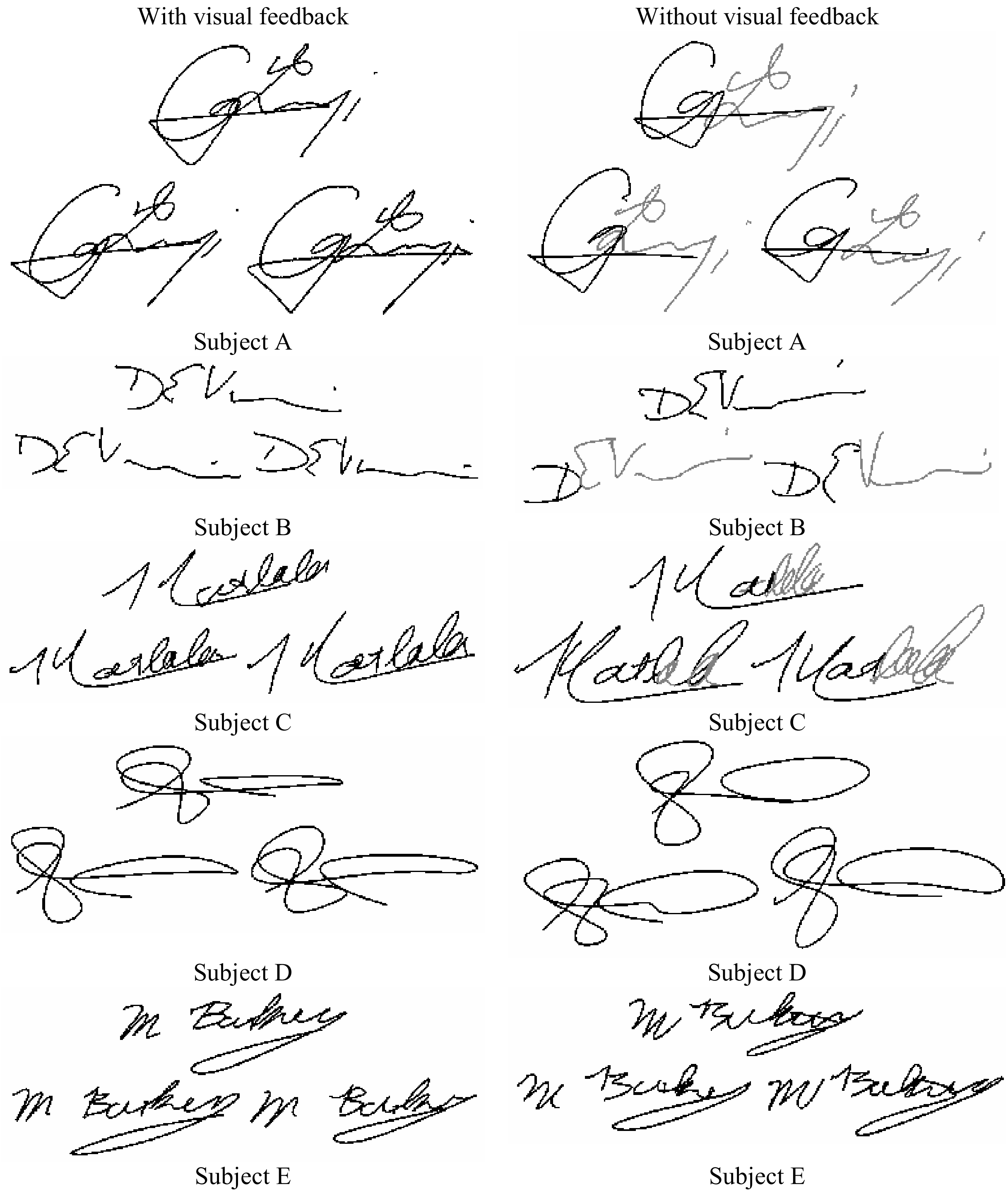
Although the proposed handwriting motor system has been modelled as a closed loop controlled system, an underlying belief held by many DSV algorithms designers is that the process of signing is necessarily a *reflex action* or a *ballistic motion* rather than a *deliberate action*. From a neurophysiological perspective ballistic motion results from the firing of neurons within the brain's motor cortex at a *predetermined* intensity and duration in activating the correct muscles in a predetermined order, which gives rise to consistency in signature dynamics [2]. In the terminology of systems and control, this corresponds to open-loop control, which voids the value of the sensory feedback loops.

If the assumption that signatures are written ballistically were true, then all DSV algorithm designers would be afforded the luxury of infallible signal consistency in all aspects of a signature. However, it is evident from a system performance analysis of Ostrem and Crane [12] and a study performed on pen point kinematics by Schomaker and Plamondon [13] that both the biomechanical consistency and biomechanical coupling between the various dynamic features are not necessarily emphatic. This suggests that a signer always has, to a larger or lesser of an extent, deliberate control over a signature being signed[9], thereby reducing the ballistic component of the signing process while increasing the level of intra-personal variation. The lack of evidence in literature to inarguably and solidly support the ballisticity of signatures, necessitates an investigation into the true mixture of ballisticity and deliberation in a handwritten signature.

An experiment was set up to examine the degree of ballisticity in a signature, where 10 subjects were asked to sign 5 times first with their eyes open, and then with their eyes closed. During each trial, the subjects were allowed to signal if they felt that the signature they signed was of poor quality, since an admittedly bad signature should not be classified to be within the range of normal intra-personal variations. During 'blind' sessions, the subjects were prohibited from seeing what they signed after each trial, since that would constitute delayed feedback that could be used in the next signature. The aim of the experiment was to see if there were significant differences between the signatures that are signed with and without visual feedback.

[9] Recall that there would always be postural feedback from the nerves in the fingers, hand wrist and forearm of a signer.

If signatures are in fact signed ballistically, then the variations observed amongst ‘blind’ signatures should fall within the normal range of variations for the ‘seeing’ signatures. If this is not the case however, then the inferior quality of ‘blind’ signatures would suggest that visual feedback is an essential facility in the signing process i.e. a closed loop system. Some of the specimens of ‘seeing’ and ‘blind’ signatures collected are illustrated below.



With visual feedback

Without visual feedback

Subject A

Subject A

Subject B

Subject B

Subject C

Subject C

Subject D

Subject D

Subject E

Subject E

Figure 14 A test of ballisticity in signatures

It is evident that signatures in the right column are systematically of lower quality than those in the left. Specifically, with regards to subjects A to C, where pen lifts are present in the signatures, the relative position of parts of signatures became inconsistent when signed without visual feedback – these are shown in grey. Furthermore, the grey parts of the signatures of subjects A and B had relative positional deviations in the orthogonal directions, whilst parts of subject C's signature were overlapping due to the fact that the writing hand was not moving sufficiently fast to the right. The above observations can be explained by the fact that a typical seeing personal is not well trained in knowing the exact position of his or her hands and arms based on neural feedback alone.

The signature of subject D was written without pen lifts, thus no offset problems were encountered, although the overall shape of the signatures written without visual feedback can be seen to deviate from the norm, especially with reference to the more pronounced ending loop. In the signature of subject E, apart from the ending stroke of the 'M' and the fact that the ending loop is consistently not being closed in the 'blind' signatures, no significant differences are otherwise visible.

From the above observations, it is quite apparent that signatures are not written entirely ballistically, and visual feedback is crucial for signatures containing pen lifts (subjects A and B), and remain so even for some uninterrupted one-stroke signatures (Subject C). Furthermore, the dependency on visual feedback may also be a function of the signer, since it has been seen that subject E, whose signature contained pen lifts, did not suffer from distortions or positional deviations as much as the other four subjects did.

Nevertheless, within the context of individual segments, consistency in the shape of the written product is equally unmistakable. This serves confirms that ballistic motion can indeed yield high levels of dynamic consistency in this context. Evidence for this is that if the positional deviations in the 'blind' signatures are neglected, then it is evident that the individual segments in the signatures of subjects A, B, D and E are produced approximately consistently each time, which is consistent with the characteristics of ballistic motion.

In summary, the major findings of this subsection are:

1. The signature process is a mixture of ballistic and deliberate motion.
2. There would always be at least some feedback in the signing process, particularly via the nervous system.
3. The criticality of visual feedback is dependent on the form of the specific signature.
4. Individuals are capable of varying levels of ballistic motion *ceteris paribus*.

In summary, the core finding of the experiment was that between pen lifts, ballisticity dominates writing motion, and can indeed offer a high level of consistency in handwriting dynamics. However, visual feedback remains essential for the accurate placement of individual handwriting or signature segments, and in fact improves the overall level of consistency when pen lifts are involved. The conclusion is therefore that ballisticity and deliberation dominates separate aspects of handwriting and signature, and play mutually complementary roles in ensuring a high level of overall consistency.

To summarise: this section has first explained the context of DSV in the field of handwriting related researches, followed by an analysis of various aspects of human movements starting with a discussion of the 2/3 power law, which formed the basis of numerous ideas and findings subsequently presented. The signing process as a feedback control system was examined, which led to the identification of the shape and location of the region of feasible control strategy as being the entity that differentiates the writing style of writers. Other major findings included the identification of the real challenge facing signature forgers, the inviolability and reciprocity of the 2/3 power law, the extraction of velocity information from static images, and the description of the signature process as a function of both ballistic and deliberate motions.

3. DSV Algorithm Design Considerations

With knowledge of the findings presented in the previous section, the generic structure of DSV systems and challenges relevant to DSV algorithms remain to be discussed prior to the actual presentation of the details pertaining to the design effort. The aim of this section is to identify the generic components of DSV algorithms that are necessary in carrying out structured and systematic design, and also to discuss the nature of problem to which DSV algorithms are applied. At the end of the section, the reader should gain an appreciation of the design decisions relevant to DSV algorithms, which would be instrumental in understanding the organisation and the logical progression of the remaining sections of this document.

3.1. Generic Architecture of DSV Algorithms

A simple categorisation can be made along the three principal axes for any on-line signature verification system, namely:

1. The primary inputs used
2. The signature representation
3. The type of classifier

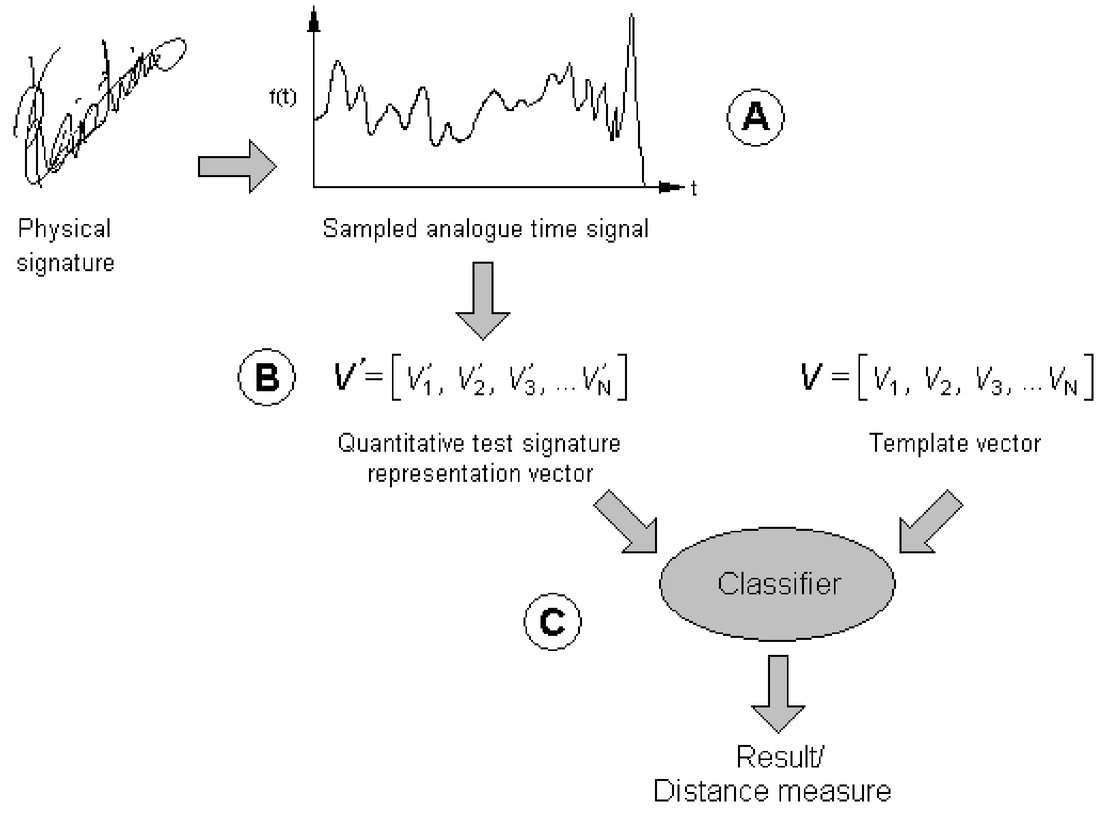

f(t)
t
A
Physical signature
Sampled analogue time signal
B
$V'=[V'_1, V'_2, V'_3, \dots V'_N]$
Quantitative test signature representation vector
$V=[V_1, V_2, V_3, \dots V_N]$
Template vector
C
Classifier
Result/ Distance measure

Figure 15 The model of DSV implementations

Figure 15 shows the generic architecture of most if not all DSV algorithms. Module A is the sampled analogue time domain signal representing some aspect of a handwritten signature – this is the digital representation of the signature itself, and is the primary source of information on which verification is based.

Module B, although not explicitly shown, is the process of generating the 'Quantitative signature representation vector'. To understand the nature and purpose of module B, it is useful to consider the distinction between data and information. Data in the crudest sense is any piece of fact, which in itself is meaningless, and valueless, while information is knowledge deducible from data (or simple facts) that can be used to construct an event and thereby enable decision-making. Analogously, samples of an analogue signal are the meaningless raw data in the context of signature verification, but constitute the entity from which information can be deduced – the representation vector being the useful information extracted.

The vector is called a representation vector because the combination of the vector's elements epitomises or is a representation of the physical signature, and it is also quantitative because its elements are invariably objective, explicit numeric values. In view of these, it is clear that the second module is essentially the information extraction function, the quality of whose outcome is dependent both on the particular extraction method used, as well as the quality (in terms of the discriminant power) of the input data set chosen. This step is necessary in any DSV algorithm since machine-based decision-making using classifiers can only be implemented using quantitative and explicit values.

Module C is the final stage of a verification algorithm. Given a signature representation as yielded by module B, the function of module C is to offer an indication of how far[10] away the test signature is relative to the template, and as such it provides the facility for the designer to adjust the FRR and FAR by tuning an explicit threshold value.

It should be evident from the above that the generic architectural form adopted by DSV algorithms does indeed seem to be the most logical construct for a verification system. In addition, Figure 15 has also illustrated that the design of a DSV algorithm is principally a three degree of freedom task i.e. the choice of input set A, the choice of representation vector B and the choice of the classifier C, which are discussed in detail in Sections 4, 5 and 6 respectively. Indeed, it is along these three principal axes of freedom that the diversity of DSV algorithms stems.

3.2. The Challenge Facing DSV Algorithm Design

As noted in the evaluation of Plamondon and Lorette [2], signature verification can ideally be presented as a two-class partitioning problem, in which a binary decision would suffice in describing the qualification of a given signature i.e. either definitely genuine or definitely forged. However, in spite of the underlying consistencies in handwritten signatures, people are not systems capable of reproducing precise and identical signature characteristics from one instance to the next.

Consequently, two classes of variabilities need to be defined to better describe the problem:

1. Intra-personal variation: normal range of variations observed in a set of genuine signatures signed by the same person.
2. Inter-personal variation: variations observed between a genuine and a forged signature produced by two separate signers.

What the above facts and definitions point to is that there are inevitable variations in handwritten signatures, making the signature process a fuzzy system.

[10] In the sense of the extent of similarity between the template and the signature.

Consider the association of a given signature with a finite number of characteristics, the set of characteristics is arbitrarily divided into two sub-sets A and B, and a singular global measure is then calculated for each set e.g. 'a' from set A and 'b' from set B. When each signature is plotted in the a-b plane according to its (a, b) co-ordinate, the resultant distribution pattern would appear as shown in Figure 16 and Figure 17.

In particular, Figure 16 shows the *idealised* problem of signature verification, where signatures are either distinctly genuine or distinctly counterfeit. Figure 17 however shows the *practical* problem of verification, where variations in signatures cause the scatter of genuine and forged signatures to overlap in the a-b plane. This is because the intra-personal and inter-personal variations do not often form exclusive sets in reality. It should also be noted that if only one feature set is considered, then the plots reduce to a conventional statistical distribution plot e.g. a normal distribution.

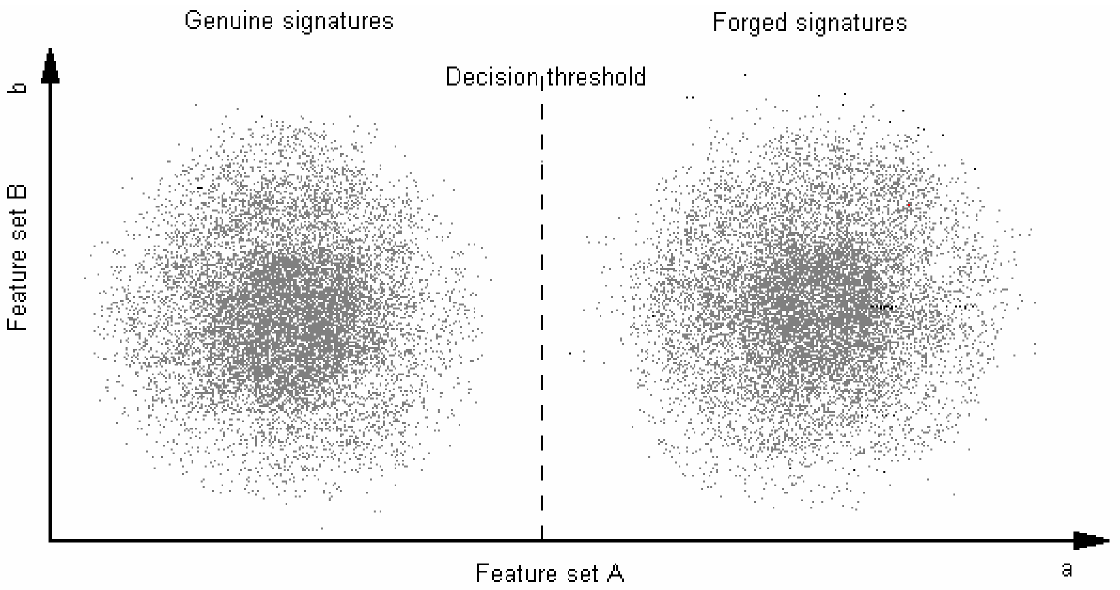


Figure 16 Idealised problem of signature verification

Given the stochastic nature of the problem, it is clear that it is inappropriate to have a binary value as a classifier output; rather, the classifying function should provide a measure of the degree of confidence in grading a signature as a genuine or a forgery, thereby incorporating a degree of fuzziness into the decision making process.

Furthermore, the reality of the problem makes it possible for a classifier to grade a genuine signature as a forgery – False Rejection Rate (FRR), or classify a forged signature as a genuine – False Acceptance Rate (FAR). Naturally, the ideal value for both FRR and FAR in practical applications would be 0% i.e. zero overlap, although the nature of the problem excludes this possibility.

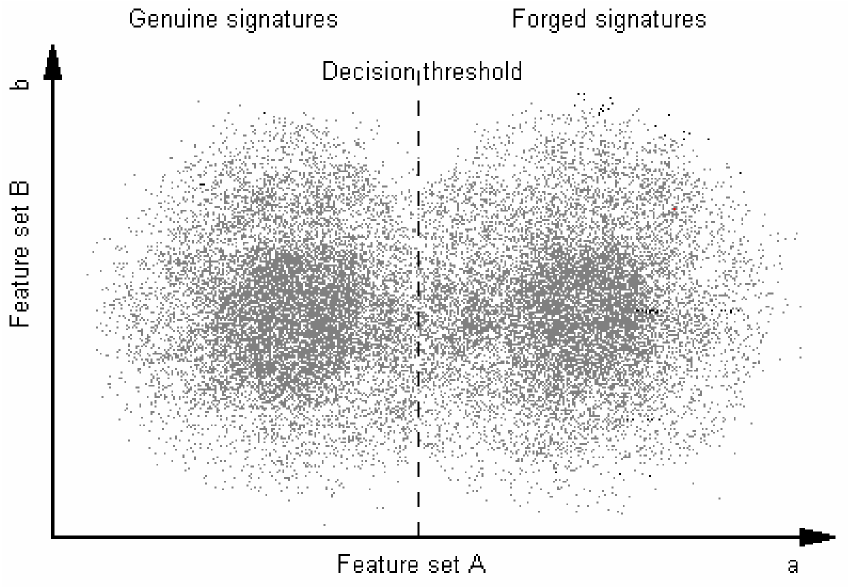


Figure 17 Practical problem of signature verification

Three degrees of freedom exist in the design of DSV algorithms that can influence the magnitude of FRR and FAR, which are used as indicators of algorithm quality. Firstly, the choice of feature set A and B for the same pool of signatures directly affects the global measures a and b, consequently different choices of A and B would result in different placements of the data scatters in the a-b plane. Specifically, a good choice of A and B would increase the separation between the genuine and forged signature scatters, thereby reducing the distribution overlap and the FRR and FAR in turn. As mentioned earlier, the problem remains unchanged when reduced to incorporate only a single feature set A say, in which case the set A becomes synonymous with the representation matrix discussed in the previous section.

Secondly, for the same choice of A and B, different definitions of a and b can be developed to obtain different a and b values from common feature sets, and this would also in turn affect the overlap pattern between the data scatters. It is precisely this design consideration that distinguishes one classifier from the other.

Lastly, the FRR and FAR can also be varied by the placement of the decision threshold. This behaviour is in fact crucial in its provision of a mechanism to allow performance tuning of DSV algorithms, since it allows the designer to vary the system characteristic without major re-engineering efforts; furthermore this behaviour also facilitates quantitative evaluation of system performances by means of error trade-off curves, an example of which is shown in Figure 18.

In Figure 18, it can be seen that as the decision criterion becomes increasingly stringent, decreasing amounts of intra-personal variations would be tolerated, thereby causing an increase in FRR. Conversely, as the decision threshold is loosened as it shifts right, more of the inter-personal difference would be tolerated, thereby increasing the FAR. In the context of practical application of DSV, a higher FRR would usually be more acceptable than a high FAR, since the simple implication thereof is that the authentic signer would have to sign again to be verified should verification have failed the first time. On the contrary, a high FAR would be rather perturbing to the user, since the implication thereof includes the compromise of personal security, privacy and assets. In real-world applications, both high FRR and FAR would be unacceptable, and the user tolerance for FRR and FAR of a DSV scheme is ultimately dependent on the specific application.

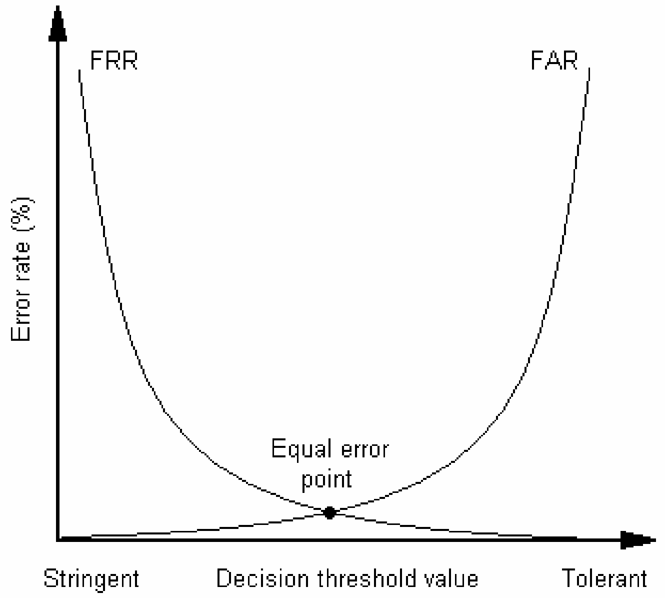


Figure 18 The error trade-off graph

One final point of discussion remains before proceeding onto the actual design of the DSV algorithm. The equal error rate is the point in the graph of Figure 18 where both the FRR and FAR assumes the same value. This point is often quoted as a unique measure for characterizing the security level of a biometric system, although the decision threshold is usually tuned to reduce the FAR for reasons described above. It should now be clear that FRR and FAR are not independently fixable parameters, but are strictly co-varying values that are linked by the underlying choice of representation vector, distance measure definition and the value of decision threshold chosen.

In summary, Section 3 has described the generic architecture of DSV algorithms, qualified the nature of problem, highlighted the challenges surrounding DSV classifiers, listed the critical design considerations, and described the widely adopted means of system characterization. With the aforementioned knowledge in mind, the document proceeds onto the main body of this document.

4. System Inputs

This section forms the first of three sections discussing and analysing each of the DSV algorithm's main architectural components. In order to provide a feel and appreciation of the commonly adopted approaches for each component, representative examples and brief descriptions of topics of interest drawn from various algorithms encountered in literature are also provided. This in the opinion of the author would present more meaningful comparisons and holistic notions with regards to the design of on-line signature verification algorithms, and is similarly applicable in Sections 5 and 6.

All digitally automated DSV schemes require inputs in the form of the digitised time-domain signals representing specific aspects of a signature, and the acquisition thereof invariably involves the capturing of pen or pen-tip movements in 2 or 3 dimensions whilst a signature is being produced. The commonly used inputs include:

- Spatial trajectory
- 2D pen-tip velocity, pen-tip velocity in orthogonal directions
- 2D pen-tip acceleration, pen-tip acceleration in orthogonal directions
- Pen inclination
- Normal pen-tip pressure, axial pen-tip pressure, planar-orthogonal pen-tip pressures

The above list of quantities is commonly chosen because they are the most rudimentary and accessible features of a handwritten signature, and can all be measured and captured relatively easily using semiconductor-based components and instruments.

From the literature survey conducted, it emerged that the inputs to all DSV algorithms are indeed invariably those listed above. In particular, the DSV implementations proposed by Nalwa [11], Wu *et al* [14], McCabe [15], Dullink *et al* [16] and Wu *et al* [17] were all based on only x(t) and y(t) signals captured using electronic digitising tablets. Martínez-R *et al* [18] however derived their x(t) and y(t) signals as the indefinite double integral of orthogonal accelerations measured using accelerometers – the main advantage being that of low susceptance to signal noise.

The equipment used for the current research was based on a prior project also relating to dynamic signature verification, and as mentioned previously the hardware device used was a combination of a pressure sensitive surface[11] and control circuitry designed by the author (Appendix 1).

Briefly, x and y co-ordinate pairs are sampled at 600 Hz at 8-bit resolution, and are sent to the PC via the 9-pin RS232 serial interface. In software, MATLAB code has been developed to retrieve the co-ordinate pairs into the MATLAB workspace, where three pre-processing tasks are performed. Firstly, wild samples are removed i.e. single samples whose values are significantly different from its neighbours are repaired. Secondly, 0-value samples resulting from pen lifts during signatures are removed, since they provide no information towards plant modelling[12]. Thirdly the data series is low-pass filtered using a 60th order zero-phase-distortion digital filter

[11] The pressure sensitive surface is a binary one, where touches are sensed only as binary states with no continuous analogue pressure values yielded.

[12] The duration and location of pen lifts in time are stored so that the unabridged profile can be reconstructed.

with a cut-off frequency of 20 Hz. The numerator and denominator to the filter were obtained using MATLAB's Signal Processing Toolbox's filter design toolkit[13].

The final format of data is an N by 3 matrix, where N is the number of samples captured. The first column of the matrix contains all x co-ordinate values, with all elements of the second column being the corresponding y co-ordinate value. Column 3 contains pressure data, which is the next topic of discussion.

A number of algorithms exploiting the pressure characteristics of a signature were also found[14], with three distinct types of pen-tip pressures being distinguished, namely: axial pen pressure, normal pen pressure and planar-orthogonal pen pressure.

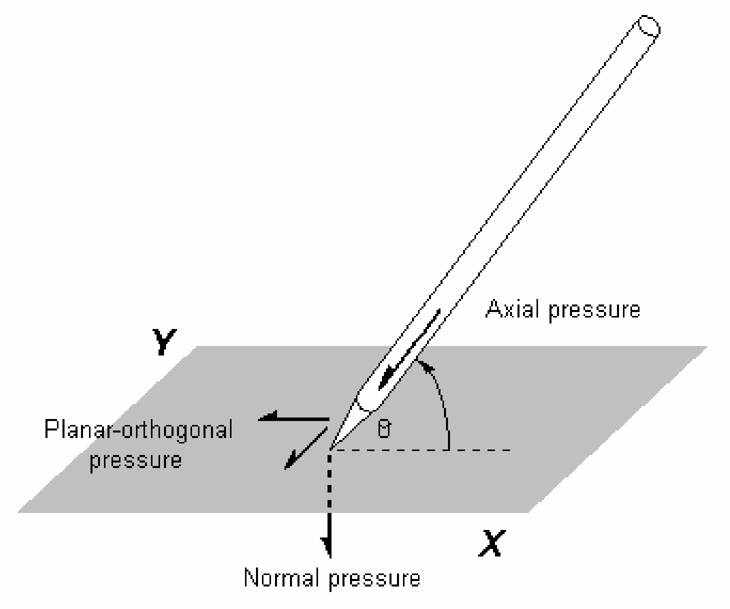


Figure 19 An analysis of pen-tip pressure vectors

The axial pen pressure refers to the pressure exerted along the longitudinal axis of the pen, this pressure is directly related to the normal pen pressure, which is directed perpendicularly into the x-y plane. From the geometry shown in Figure 19, it is trivial to deduce that:

$$\text{Normal pressure} = \text{Axial pressure} \times \sin\theta$$

Equation 4 Relationship between normal and axial pen pressure

where θ is the angle of elevation with respect to the x-y plane. The presence of the axial and normal pressure vectors stems from the need to apply downward force in causing the pen to ink sufficiently on the writing surface in order to leave behind a visible trace.

Amongst those that utilises the normal pen pressure in addition to the x(t) and y(t) signals are algorithms of Martínez-R and Alcántara-S [19], Nakanishi *et al* [20] and Hastie *et al* [21]. On the

[13] The filtering was only necessary to minimise the discretisation errors of an 8-bit ADC, and the occasional noise interference, since the output signal from the hardware was of such high quality that little filtering was needed indeed.

[14] The word 'pressure' can also be understood to mean with 'force' in this context, since the two are directly related via an area metric, and force is in fact often the quantity that is really being measured.

other hand, relatively few researchers have adopted the axial pressure as an input, since the measurement of axial pressure necessitates the use of a specialised digital pen capable of measuring compressive forces applied along its longitudinal axis. Nonetheless, the verification method proposed by Liu *et al* [22] is one example of such an algorithm. The implementation suggested by Keit *et al* [23] seems to indicate likewise as evident from the content of their paper.

The last type of pressure is the planar-orthogonal pressure, which refers specifically to the forces experienced by the pen-tip as a result of horizontal dragging action across the writing surface, which is necessary to overcome the frictional forces of the writing surface. Work done by Crane and Ostrem [12] is of particular interest in terms of the choice of input set, since their verification effort is based solely on the measurements of a three-axis force-sensitive pen without the utilisation of any x(t) or y(t) data. In this instance, the shape of a signature played only a marginal and indirect role in the verification process. As was the case with axial pressures, the measurement of planar-orthogonal forces required a specially instrumented ballpoint pen fitted with strain gauges.

The research under discussion employed the axial pen pressure of writing, using an existing pressure sensitive pen that belonged to the same hardware kit originally developed for a prior DSV project. The pen is fitted with a stiff spring, across the ends of which a pair of infrared (IR) photo-emitter and receiver is mounted. Pressure is registered as variations in the intensity of IR radiation sensed by the IR receiver as the spring is compressed during writing. The range of pressure sensed is between 0 kPa to 1.33 kPa, which is also sampled at 600 Hz and sent to the PC via the 9-pin RS232 serial interface (refer to Appendix1). The series of pressure data forms the third column of the signature data matrix.

Thus far, the DSV system inputs have been identified and chosen without due regards to the basis of such decisions. The motive of the various developers and researchers for choosing the particular input sets needs now be placed under critical evaluation, so that the grounds for choosing particular input sets and the discriminant quality thereof may be established.

It was noted that all researchers basing their algorithm solely on x(t) and y(t) have expressed their belief in the critical importance of stroke timing in one's signature – a notion in fact well supported both from a forensics and neurophysiological points of view. D Ellen in his book *The Scientific Examination of Documents: Methods and Techniques* [24] described a number of commonly used signature forgery techniques well known to forensic document analysts, these include:

- Freehand simulations
- Slowly made simulations
- Rapidly made simulations
- Direct tracing

What transpired from the description of the range of techniques is that the production of fraudulent signatures are inevitably synthetic in that they do not exhibit the natural rhythms and inking patterns of spontaneous writing. The implication hereof is that the task of producing an unfamiliar yet complex pattern to a high degree of precision would inevitably result in deviations in the construct of the forged writing from its natural form, and this establishes timing information as a critical factor in the detection of forged signatures. Indeed, in the absence of time information, questioned document examiners resort to the scrutiny of inking patterns and line quality to establish hints of forgery.

From the neurophysiological perspective, the process of signing is viewed as a series of co-ordinated neuro-muscle interactions within the forearm, wrist and hand complexes. Due to the fact that the natural movement of the hand in producing handwriting obeys the 2/3-power law [5], the velocity profile represents a powerful weapon against all except well-executed freehand-simulated forgeries, since tracing would inevitably result in jitters as has been explained in Section 2.2.5. From the preceding discussions, it is thus clear that timing information is indeed a mostly impregnable feature of a handwritten signature subject to the findings of Section 2.2.4, and this suggests that the argument of basing a DSV algorithm on x(t), y(t) or derivatives thereof as did in the research is in fact well founded.

In accordance with the above, the importance of time-domain signature profiles has been used as a major selling point of on-line signature verification as biometric technology, the claim being that it is practically impossible for a forger to simultaneously imitate both the static (2D) and dynamic characteristics of a signature. In spite of this, Nalwa [11] made a rather well founded point to the contrary that highlighted the conflicting nature of DSV as a technology.

Nalwa's argument was that the other well-known selling point of on-line signature verification apart from its security aspect is the long-established social acceptance of the legal value of a handwritten signature, making DSV the most natural choice for society in integrating paper and electronic transactions. However, the velocity profile that brings about security requires the signer to not only be consistent in the physical appearance alone, but also in the additional dimensions of time and force – elements that are absent in the traditional notion of a genuine signature. Alteration of this traditional expectation would in turn weaken probably the strongest argument in favour of the continued use of handwritten signatures for verification [11].

This contradiction is however not a fundamental problem to the systems approach to DSV, since within this paradigm strict consistency in time is not a requisite. This is because the algorithm looks beyond the appearance of the time domain signals, focusing rather on the dynamic characteristics of a plant in action.

The final point of discussion in this section is the use of signature pressure profiles as a verification input. A number of algorithms were encountered in the literature that utilises the pressure profile in one way or another [22, 23, 19, 20, 21, 12], the particulars of which has been briefly described previously. It was evident from literature that many algorithm developers simply assumed that the pressure profile would also be a consistent and discriminant feature of a signature – an assumption that is most probably a result of the more fundamental supposition that signatures are written ballistically.

Plamondon and Schomaker [13] conducted a thorough investigation into the nature and correlation of writing pressure with respect to other hand kinematic variables, which in turn assessed the validity of two models of writing pressure control. The one model is the 'biomechanical' model, in which it is proposed that writing pressure is a co-variant function of other kinematic variables in a manner governed by certain underlying biological motor laws i.e. given sufficient knowledge of hand kinematics the writing pressure can be inferred. The other model is the 'central' model, where writing pressure is viewed as an actively and independently controlled variable essentially decoupled from the state and behaviour of other hand kinematic variables.

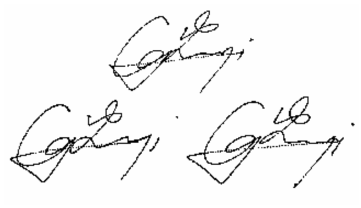
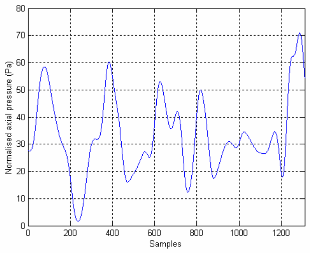
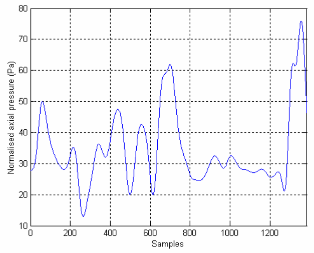
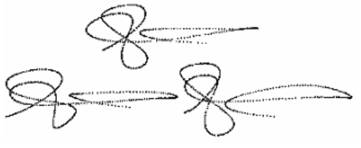
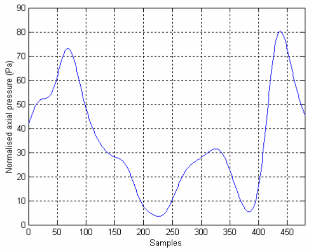
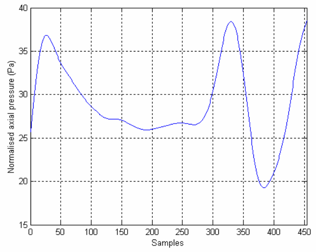
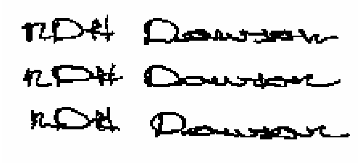
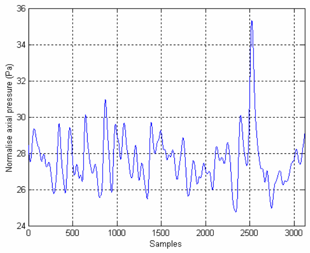
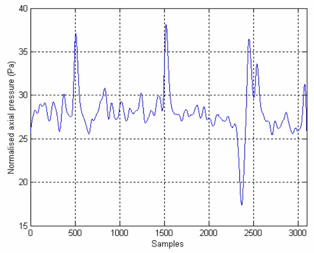
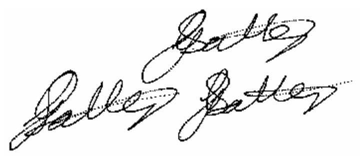
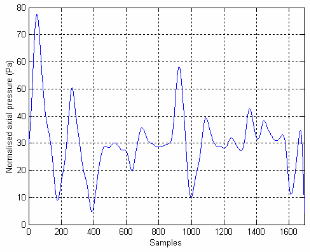
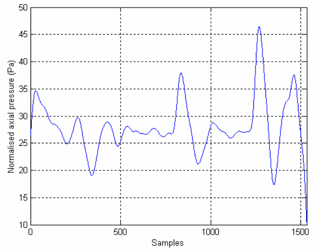
Through a comprehensive experiment, results analysed by Plamondon and Schomaker [13] indicated that three dominant classes of writers in terms of the writing pressure for cursive writing (a form which signatures often adopt) could be identified. Supposing the three groups are known as types A, B and C subjects, the result of their work can be summarised as follows:

- Type A subjects displayed complex but replicable relationships between the axial pen pressure and displacement. The replicability of the writing pressure pattern and the instantaneous force-displacement correlation pattern both support the notion of independent, feedforward control of the force by the central nervous system. Most experimental subjects fall into this category.
- Type B subjects exhibited a strong correlation between axial pen force and movement kinematics. In particular, it was noted that the axial pen pressure consistently increased in down strokes.
- Type C subjects were characterized by low replicability of both kinematics as well as axial pen pressure profile.

The conclusion was that due to the low levels of correlation between pen-tip kinematics and axial pen pressure observed, a simple biomechanical coupling between axial pen pressure and kinematics is unlikely [13]. This finding is in fact in agreement with [22] as well as experiments carried out as a part of the research. However, this fact is not always mentioned in literature since it would bring into question the true security of DSV algorithms that rely heavily on either normal or axial pen pressure.

Table 1 illustrates the typical level of consistency observed for a set of genuine signatures. In the first column the actual signature specimens are shown, and in columns 2 and 3 two instances of the corresponding pressure profiles are given. Evidently, although gross consistency can at times be observed, it is nevertheless considered to be insufficient for the purposes of DSV algorithms. The author is of the opinion that given that there is a current lack of a firm understanding and insight to the laws governing the control of writing pressure, the superficial treatment of pressure signals for signature verification is not justified. Nonetheless it cannot be denied that signature pressure profiles can be instrumental in complementing the performance of algorithms based predominantly on x(t), y(t) and derivatives thereof. Consequently, the axial pressure can be employed without harm as a readily accessible dynamic feature via the instrumented pen available to research.

Table 1 Typical levels of consistency achievable with axial pen pressure

Signature specimen	One instance	Another instance
		
		
		
		

5. Signature Representation

5.1. Signature Representation Outline

Once analogue signals have been captured into the digital domain, appropriate quantitative representations of signatures are needed to facilitate subjective comparison of two signatures. Since it is often the characteristics of the sampled analogue signals rather than the signals themselves that are important, a choice has to be made in deciding on a good abstraction of a signature.

In a comprehensive study performed by Plamondon and Lorette [2], approaches to signature representation were classified into two major categories: functions and parameters.

In the case of function-based representation, operations and comparisons are carried out on entire functions of x[n], v_x[n], p[n] etc., where a signature would be treated in its entirety during verification. In this case, the quantitative signature representation vector (or matrix) would for example take the form:

$$V' = [v_1', v_2', v_3']$$
$$v_1' = x[n], \; v_2' = y[n], v_3' = p[n] \qquad \text{for } 1 \le n \le N$$

where elements of the representation vector *V'* are the time series samples of the analogue x(t), y(t) and p(t) signals with N being the total number of samples acquired.

Under the parameter representation scheme however, geometric, timing and specific dynamic parameters associated with time signals of signatures are typically extracted.. The parametric representation vector would for example have the form:

$$V' = [v_1', v_2', v_3' \ldots v_N']$$

$$v_1' = X_{Max}, \; v_2' = Vel_{Max}, v_3' = Vel_{Ave \ldots} v_N' = P_{Ave} \qquad \text{for } 1 \le n \le N$$

where entries of the representation vector are single numbers indicating the value of the various parameters epitomising the signature, with N being the total number of parameters chosen.

As with most design decisions, there are advantages and disadvantages associated with each of the modes of representation, specifically with regards to the discriminancy power, memory requirement and processing times in the context of automated DSV, which are briefly discussed below.

Since most of the digitising equipment available for capturing 2D co-ordinates, velocity, acceleration and pressure sample at a few hundred hertz, and a typical signature lasts no longer than 10 seconds, function-based representation of signatures typically occupy a few kilobytes – the approximate size of templates used under this scheme. Consequences of this relatively large set of data (compared to parameter representation) are that of larger template sizes, higher memory requirement and probably longer processing time. Nevertheless, the argument in favour of a function-based representation is that since no compression of information occurs, it is ensured that the verification result is derived from every available data point of the test signature, thereby making use of all available information.

On the other hand, verification algorithms based on parameter representation typically have templates ranging from tens to hundreds of bytes, which in turn requires less storage space and typically less processing time. But unlike algorithms based on function representation, results of parameter-based verification systems are more critically affected by the choice of the parameter set used. In other words, a parameter-based algorithm is only as good as the discriminant quality of the set of parameters chosen, this is because the majority of the data from which the parameters are calculated are eventually discarded in a process of information compression, and it is possible for critical verification information to be lost during the process.

The challenging problem of choosing the 'best' set of parameters for verification is in fact a well acknowledged one, since it is no trivial task in determining the set of most stable parameters, the set of most discriminant parameters, the most effective and discriminant combinations of features and the optimal number of parameters to use [21, 2].

5.2. Formats of Representation Encountered

The objective of this section is to describe the representation vector generation process adopted by some of the algorithms encountered in literature. The approaches researchers have adopted in generating a signature representation have been especially diverse in nature, often requiring extensive knowledge and experience in specialised areas of mathematics, pattern recognition, engineering and computer science. Consequently, only high-level descriptions of each system will be given, with the objective of presenting the general philosophy rather than the low level details, which are less conducive to an overall understanding and comparison of the various methods.

The algorithms developed Liu *et al* [22], Hastie *et al* [21], Nalwa [11], Martínez-R and Alcántara-S [19], Wu *et al* [14], McCabe [15] and Nakanishi *et al* [20] are among those that rely on function representations of signatures.

5.2.1. Liu *et al*

In the algorithm of Liu, Herbst and Anthony, a test signature is represented by a combination of segments of the acceleration functions $a_x[n]$ and $a_y[n]$ acquired between two pen-lifts. In their paper, it was noted that the pressure profile was dependent on external elements such as how the pen is inking etc., and thus exhibited only gross correlation. Consequently, the pressure representation of the signature was defined as the offset-removed minutiae of $p_{Axial}[n]$, which too was segmented. The representation matrix in this case is:

$$V' = \begin{bmatrix} a_{xseg1}[n] & a_{yseg1}[n] & p'_{Axialseg1}[n] \\ a_{xseg2}[n] & a_{yseg2}[n] & p'_{Axialseg2}[n] \\ \vdots & \vdots & \vdots \\ a_{xsegm}[n] & a_{ysegm}[n] & p'_{Axialsegm}[n] \end{bmatrix}$$

Each column of the matrix contains one data series, and each row corresponds to one of m segments of the overall signals $a_x[n]$, $a_y[n]$ and $p'_{Axial}[n]$[15].

[15] The offset removed pressure signal.

5.2.2. Hastie *et al*

With Hastie *et al*'s algorithm, each signature is modelled as a piece-wise rotation, size, offset and time-warp compensated version of the template signature with an added stochastic component according to the following two matrix equations:

$$X(t) = Y[h(t)]$$

Equation 5 Expression for a general signature

$$Y(u) = A(u)F(u) + \mu(u) + e(u)$$

Equation 6 Hastie's model of a signature

where:

$X(t)$ = vector of raw data samples of a test signature
$Y[h(t)]$ = time-warp compensated version of $X(t)$
$h(t)$ = the time-warp function
$F(u)$ = idealised template of a signature parameterised over a universal parameter
$A(u)$ = size and rotational compensator
$\mu(u)$ = offset compensator
$e(t)$ = measurement error

The compensated segments of x[n] and y[n] then form the basis of a distance calculation algorithm in the last step of verification i.e.

$$V' = \begin{bmatrix} F'_{sxeg1}[n] & F'_{yseg1}[n] \\ F'_{xseg2}[n] & F'_{yseg2}[n] \\ \vdots & \vdots \\ F'_{xsegm}[n] & F'_{ysegm}[n] \end{bmatrix}$$

where $F'_x[n]$ is the compensated form of X[n]. Although it is claimed that a forgery can usually be detected based on the time-warp function h(t) alone, the effects of the attempts to compensate X(t) towards F(u) are however not clear, since such actions could in fact be compensating for variances in a forgery that would in turn work in favour of the forger.

5.2.3. Nalwa

In Nalwa's DSV algorithm [11], which uses x[n] and y[n] only, the trajectory of a signature is first parameterised over a normalised arc-length[16] *l*. Based on this parameterisation, five characteristic functions of a signature are then computed, each of which represents the trend in one aspect of the signature trajectory. The five features chosen to characterise a trajectory are: the coordinates $\overline{x}(l)$ and $\overline{y}(l)$ of the centre of mass of the trajectory along *l*, the torque T(*l*) exerted about the origin, and the two curvature-ellipse measures, denoted $s_1(l)$ and $s_2(l)$, which describes the curvature and orientation of the strokes along *l*. The normalised characteristic functions combined then constitute the representation matrix i.e.:

[16] The parameterisation of signatures over a spatial variable stems from Nalwa's belief that verification should be based on spatial characteristics rather than on temporal characteristics.

$$V' = \left[\bar{x}(l) \quad \bar{y}(l) \quad T(l) \quad s_1(l) \quad s_2(l) \right]$$

where each element of the matrix is a column vector representing the normalised characteristic functions as a function of *l*.

5.2.4. Martínez-R and Alcántara-S

Nalwa's algorithm was also the basis for a DSV scheme developed by Martínez-R and Alcántara-S [19], which in addition to the five features used by Nalwa, further incorporated five additional dynamic features consisting of v(t), p(t), p'(t), base of writing as a function of time Bw(t) and lastly top of writing as a function of time Tw(t) [17].

A major part of the work done by Martínez-R and Alcántara-S also centred on the problem of using genetic algorithm to create an optimised feature model, whose elements included an optimal averaged prototype function P, a representative function of the feature ω, a consistency function C, and a weighting factor W. From this, the representation matrix then becomes:

$$V' = \left[\bar{\mathbf{X}} \quad \bar{\mathbf{Y}} \quad \mathbf{T} \quad \mathbf{S_1} \quad \mathbf{S_2} \quad \mathbf{v} \quad \mathbf{P} \quad \mathbf{P'} \quad \mathbf{Bw} \quad \mathbf{Tw} \right]$$

$$\text{where } \bar{\mathbf{X}} = \left[P_{\bar{x}(l)} \quad \omega_{\bar{x}(l)} \quad C_{\bar{x}(l)} \quad W_{\bar{x}(l)} \right]^T \text{ etc.}$$

where each of the elements shown in bold constitutes a column vector containing the elements of each feature's model as described by the second equation.

5.2.5. Wu *et al*

In the algorithm proposed by Wu, Jou and Lee [14] the signature representation is based on four functions viz. x[n], y[n], $v_x[n]$ and $v_y[n]$. The entire algorithm is exceptionally transparent in that one-step transformed versions of the spatial and velocity signals are used directly for the evaluation of similarity distance to a reference template. Details associated with the transformation process are outlined below.

It is a widely acknowledged fact that even signatures produced by the same authentic owner would always vary in terms of their instantaneous tempo and overall duration from one signature to the next, the implications of which are:

- The features (peaks and troughs) in the spatial and velocity signals will not occur at precisely the same point on the time axis from one data set to the next
- The number of sample points will almost inevitably differ from the one data set to the next

The consequences of the above are that no point-to-point comparison can be made between two sample series representing two signatures. In order to overcome this problem, at least one of the data series would need to be normalised to achieve equal length, and then transformed in order to align the corresponding points and regions on each signal to facilitate meaningful comparison. This is typically realised by employing Dynamic Time Warping (DTW)[18], where optimum

[17] These are straight lines that touch the highest or lowest points of each written letter in a signature.

[18] A dynamic programming method that is heavily used in speech recognition [37]

alignment between two signals is achieved by creating non-linear local dilation or compression of the time axis of one of the signals relative to the time axis of the other.

In the algorithm of Wu *et al* however, a less elegant method called the 'split-and-merge' algorithm was used to find the 'best' match between the test signature and the template, a process where samples are artificially inserted so that proportionally, two signals would look similar. Suppose the transformed x[n], y[n], $v_x[n]$ and $v_y[n]$ are denoted x'[n], y'[n], $v'_x[n]$ and $v'_y[n]$ respectively, the representation matrix based on the split-and-merge algorithm is then:

$$V' = [x'[n] \quad y'[n] \quad v'_x[n] \quad v'_y[n]]$$

5.2.6. McCabe

A further algorithm encountered based on function representation was that of McCabe [15]. Being based on a DSV algorithm proposed by Gupta and Joyce, the scheme uses x[n] and y[n] data only. The particular abstraction of signature used is the encoding of all 2D features of the pen-tip trajectory in sequence of occurrence. For example if an x maximum is denoted A, an x minimum is denoted B, a y maximum C, and a y minimum D, then the circular trajectory of Figure 20 would be encoded as in {C, B, D, A}.

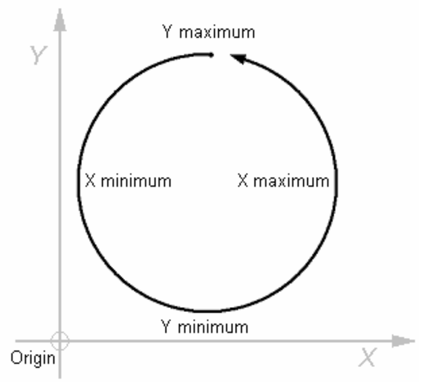


Figure 20 Encoding of a trajectory

Under this framework, a set of encoding definitions were drawn up, resulting in representation vector of the form:

$$V' = [v_1 \quad v_2 \quad v_3 \ldots v_m] \qquad v_i\Big|_1^m \in S$$

where each of the vector's elements from the 1st to the mth belongs to the encoding definition set *S*. It should be noted that as with parameter-based representation, the definition set *S* employed plays a critical role in determining the strength of a DSV algorithm, since a set of bad definitions may contain little discriminatory information, or redundant information.

5.2.7. Nakanishi *et al*

The algorithm of Nakanishi *et al* employed multiple resolution analysis based on Discrete Time Wavelets (DTW) for on-line signature verification [20]. Based on the Daubechies' wavelet as the mother function and with ten resolution levels, ten wavelet components are obtained for each of x[n] and y[n], where components from level 5 to 8 are chosen as the inputs to the classifying stage [20].

In view of the above, the representation matrix is:

$$V' = \left[x_5[n] \quad x_6[n] \quad x_7[n] \quad x_8[n] \quad y_5[n] \quad y_6[n] \quad y_7[n] \quad y_8[n] \right]$$

The subscript indices indicate the level of wavelet component.

Thus far, a number of relatively wide-ranging function representation methods have been described, including orthogonal acceleration profiles, compensated orthogonal trajectory model, parameterised characteristic functions, feature models, time-warped trajectory and velocity profiles, encoded feature strings and lastly discrete time wavelet components. It is clearly evident that the representations described above are indeed diverse in their approaches.

Before moving on, note that whereas most of the other algorithms used representation based on direct derivatives of the input data sets, Nalwa's approach stood out in that it is somewhat mid-way between purely function-based and purely parameter-based approaches. His approach has utilised the *hidden features extracted* from the input data set, not unlike the methods used to generate parameter representations, which is the focus of the next part of this section.

Algorithms discussed under parameter-based representations include those proposed by Dullink *et al* [16], Martínez-R *et al* [18], Wu *et al* [17], Keit *et al* [23] and Crane and Ostrem [12]

5.2.8. Dullink *et al*

This algorithm utilized nine signature parameters in constructing the representation vector, only one of which is related to the shape of a signature. In particular, the parameter set chosen contained only those that were easy to determine and compute from the x(t) and y(t) data, including the total signing time t_{total}, number of sign changes in the x and y velocities and x and y accelerations (n_{ux}, n_{uy}, n_{vx}, and n_{vy}), the number of zero values in the x and y accelerations (z_{ax}, z_{ay}), the pen-up time t_{penup} and finally the total path length of the signature l_{total}.

Note that the inclusion of the total path length in DSV algorithms as a comparison parameter has the effect of reducing the algorithm's tolerance for signatures of different sizes, although the advantage thereof is the institution of resistance against random forgeries. Nonetheless, the representation vector in this case is:

$$V' = \left[t_{total} \quad n_{ux} \quad n_{uy} \quad n_{vx} \quad n_{vy} \quad z_{ax} \quad z_{ay} \quad t_{penup} \quad l_{total} \right]$$

5.2.9. Martínez-R

This is a separate algorithm by J. Martínez-R *et al* apart from the one previously described. In this case the algorithm relies on a space frequency representation of a signature, where segments of a sample signature would be placed into rows of a matrix, for which the power density spectrum is then evaluated for each row. Because the matrix formed is two dimensional, the collective of all the rows' power density spectrums forms a spectrogram over the matrix from which characteristic features can be extracted to represent a signature. Overall, the scheme uses seven features in the representation vector – three static and four dynamic, which in equation form is:

$$V' = \lfloor N \quad X/Y \quad M \quad PL \quad C_x \quad C_y \quad V_y \rfloor$$

where:

N = mean value of normalizing factor
X/Y = width to height ratio
M = number of macrostructures (segments)
PL = number of pen lifts
C_x = mean value of the centroid in X
C_y = mean value of the centroid in Y
V_y = root mean square value of the vertical velocity

The values of centroids are calculated from the spectrogram matrix. Clearly, both the form and the time domain variation patterns of a signature are considered important in the design of this algorithm.

5.2.10. Wu *et al*

For this algorithm, Wu had adapted speech recognition techniques for the problem of signature verification, which is indeed rather similar in nature in terms of the inputs and objectives. In this approach, the input data series are first segmented into a number of separate subdivisions, after which a technique commonly employed in speech processing called Linear Predictive Coding (LPC) is applied[19], so that a set of linear prediction coefficients are obtained for each segment. Instead of using the LPC coefficients directly as the representative features, a performance comparison carried out by Wu *et al* indicated that using the cepstral coefficients[20] of the LPC model would yield a lower error rate and thus constituted a more suitable parameter set [20]. The combination of all cepstral coefficients for all frames of both the x[n] and y[n] series then form the representation matrix i.e.

$$V' = \begin{bmatrix} \boldsymbol{x}_1 & \boldsymbol{y}_1 \\ \boldsymbol{x}_2 & \boldsymbol{y}_2 \\ \vdots & \vdots \\ \boldsymbol{x}_m & \boldsymbol{y}_m \end{bmatrix}$$

Each element of the first and second columns is a vector containing the cepstral coefficients of each of m frames of x[n] and y[n] respectively.

5.2.11. Keit *et al*

This algorithm is the only one that is reliant on only one input data series – and a pressure series at that, which had been shown in earlier sections to be insufficiently consistent for signature verification. Nevertheless, the core technique adopted for feature generation was time series modelling with autoregression (AR)[21], whose output is a set of AR coefficients from all the

[19] For a given time data series, an N^{th} order linear predictor of the series is a linear combination of the N previous samples i.e. a difference equation where the current sample is a function of the N previous samples.

[20] These are coefficients of the LPC cepstrum, which is defined as the Fourier representation of the logarithmic amplitude spectrum based on LPC modelling.

[21] Autoregression expresses one or more stochastic variable(s) as a linear function of its/their own past values and/or the past values of other variables being considered, as well as a serially uncorrelated error term.

segments. It is stated in their work that autoregressive models have proven superiority to Fourier methods due to the ability of autoregressive models to handle short segments of data while giving better frequency resolution and smoother power spectrum [23].

The AR coefficients obtained above are then used to obtain the Power Spectral Density (PSD) values to represent each segment, and the combination of all PSD values then represent the signature. The representation matrix for this algorithm is in fact identical to the representation used by Wu *et al*, except there would only be one column whose elements are vectors containing the PSD coefficients of each segment of the p[n] series instead of cepstral coefficients.

5.2.12. Crane and Ostrem

The final DSV scheme discussed is one developed by Crane and Ostrem [12] that uses planar-orthogonal and axial pen pressures. It was recognized in their work that the success of a feature or parameter-based approach to DSV is pivoted on the choice of the parameter set. On the one hand the parameters must contain as much discriminant information as possible, while on the other hand too high a number of features could result in information redundancy. Consequently, from a pool of 18 conceptually different pressure and timing related variables initially proposed (44 in total), a final list of 25 features were chosen to construct the ‘best’ set of features e.g. the scaled mean, the standard deviation, minimum values, average positive values, total signing time, total time with pen down etc.

Since the list of chosen parameters is easily computable, it is evident that Crane and Ostrem’s approach is perhaps the one of the simplest and the most direct of parameter-based algorithms. Accordingly, the representation vector in this case would simply be a vector containing 25 elements each corresponding to one parameter of the ‘best’ set.

This concludes the discussion on parameter-based algorithms. In summary five algorithms were investigated, where the parameters chosen ranged from discrete features associated with data series e.g. duration, number of turning points, maximum and minimum values, to centroids of spectrograms and finally to cepstral or PSD coefficients of data segments. As with function-based representation, the approaches adopted are relatively wide ranging having originated from different technical spheres e.g. speech-processing, statistics, image processing etc. Whatever the parametric abstraction employed, the objective of the parametric paradigm is always to reduce redundancy by compression, and the selection of a parameter set that contains the maximum amount of discriminant information.

As a comment to this sub-section, one of the problems noted in the literature survey is that in most cases the segmentation step, whenever mentioned, is stated briefly without much attention to its specifics. The consequence of this is that the quality of the segmentation technique used is often overlooked by the unwary reader – this being applicable to both function and parameter-based algorithms. The emphasis here is that whatever the representation format and classifier used, the correct and effective working of an algorithm involving signal segmentation pivots on the assumption that genuine signatures would always be segmented in an identical or similar manner. From the experience of the author, this is not always the case for most segmentation techniques encountered – and this in turn has great implications on the practical deployability of a proposed DSV system.

5.3. A Systems Approach to Signature Representation

Based on the systems model of signature proposed in Section 2.2.2, the following section assesses the feasibility of performing system identification on the signature process, and formulates strategies to estimate the plant models and their feasibility.

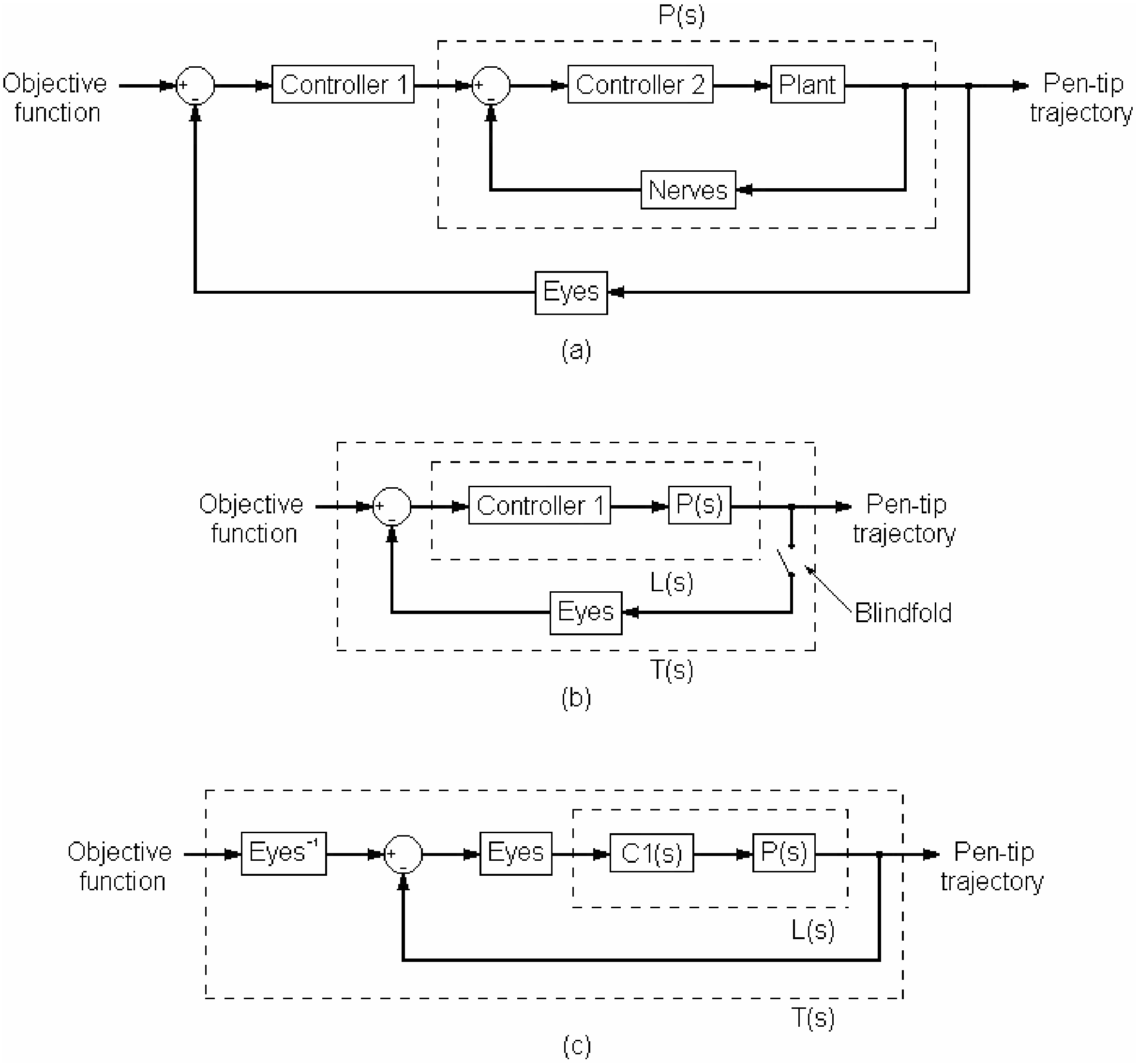


Figure 21 Signature model analysis

In the model of Figure 21 (a), the problem has been simplified by the omission of the noise and disturbance inputs to allow for a more straightforward analysis. One of the first things to be taken note of is the fact that the model is a multiple-feedback loop system, where the neural sensory facility constitutes the inner feedback loop. This is so because there is no clear-cut way to effectively open this inner loop, since the nerves are essentially an integral part of the plant itself.

Specifically, the opening of the inner loops would involve the identification and separation of *sensory* nerves from the *motor* nerves in the finger, hand, wrist and forearm complexes, after which the sensory nerves would be temporarily disabled so that a test subject would be able to sign without neural feedback. Clearly this was impossible to accomplish given the resources available to the research. Moreover, this would be even more unacceptable should it be widely deployed, where people would have to be injected with anaesthetics so that their controller and plant combination can be modelled for a DSV algorithm.

Consequently, the most appropriate abstraction of the collection of the inner loop components would be to model them as a single plant P(s). Note that the Laplace transform has been used here not to imply in any way that the plants are LTI, but simply as a convenient syntactical representation of the model components. With the lumping of controller 2, the plant and the neural feedback loop into a single plant P(s), the systems model of the signing process then becomes the one shown in Figure 21 (b).

At this point, it is already apparent that the finest granularity practically obtainable in terms of the low-level plant models is the separate modelling of P(s), controller 1 and the transfer function of the eyes. Again, since the modelling procedure can only be done non-invasively, plus the fact that the point of signal output from the brain's motor cortex as well as the signal path between controller 1 and P(s) is not clearly defined, it is not possible to clearly isolate P(s) and controller 1 as individual components. Consequently, a further lumping is needed to form the L(s) of part b of the figure. At this stage, the abstraction contains only L(s) and the transfer function of the eye as sub-components.

In order to characterise L(s), the system must be driven in open loop mode i.e. the subject must sign with their eyes closed, which is quite feasible. It is thus established that L(s) as defined in Figure 21 is an easily accessible component of the overall system. Furthermore, part (c) of Figure 21 represents the usual case of signing i.e. with visual and neural feedback. This configuration can readily provide the end-to-end behaviour of the overall feedback system T(s), where the objective function can be defined as the *mean* of a large sample of a test subject's signature, and the output defined as one *particular* signature. Under the framework of a linear approximation, the eyes' transfer function I(s) can be obtained by the following sequence of operations involving T(s) and L(s):

$$\begin{aligned} \mathrm{T(s)} &= \frac{\mathrm{L(s)}}{1+\mathrm{I(s)L(s)}} \\ \mathrm{T(s)} + \mathrm{T(s)I(s)L(s)} &= \mathrm{L(s)} \\ \mathrm{T(s)I(s)L(s)} &= \mathrm{L(s)} - \mathrm{T(s)} \\ \mathrm{I(s)} &= \frac{\mathrm{L(s)} - \mathrm{T(s)}}{\mathrm{T(s)L(s)}} \\ \mathrm{I(s)} &= \frac{1}{\mathrm{T(s)}} - \frac{1}{\mathrm{L(s)}} \end{aligned}$$

From the section on ballistic motion, it is evident that the contribution due to visual feedback is only noticeable at points of pen lifts associated with sudden changes in the x and y values of the pen tip, whilst the parts of a signature written between a pen-down and a pen-up (associated with midrange frequencies) are produced predominantly ballistically. What these suggest is that I(s) is likely to have lesser influence at the midrange frequencies, and greater influence at high frequencies, a characteristic that is likely to be similar for most signatures.

From the above, it is both clear and unfortunate, that only L(s) and I(s) are readily accessible components of the model, where L(s) is inclusive of the two sub-controllers, the plant and the neural feedback loop. Now, from the 'blind' signatures presented in Section 2.2.7, it is evident that the degradation in signature quality is not consistent in the open loop scenario, specifically in the signatures of subjects 1 and 2 for example, the positional deviations of parts of the signatures are different in each of the 'blind' signatures. What this implies is that the L(s) evaluated based on a few 'blind' signatures alone is not likely to be accurate, and in order to obtain a truly representative L(s) for a specific signer, a large number of 'blind' signatures would be needed to establish some sort of general consistency in the behaviour of the open loop plant. This however would be an unacceptable requirement to the users of the DSV system, since the users would be asked to sign large numbers of signatures with their eyes closed.

Given the above constraint on the estimation of L(s), I(s) then becomes similarly less accessible, since at least L(s) as well as T(s) must be sufficiently well known for the meaningful evaluation of I(s).

What transpired from the above analysis is that a straightforward method for the extraction of the controller in explicit form is unlikely, while only a few of the components within the signature production model can be directly assessed in a non-invasive and a user-friendly manner. This was in fact anticipated considering that the majority of the system components exist in an abstract form within the human body. Consequently, although a systems view of the signature process had been instrumental in gaining understanding and insight to the mechanisms of signature production, direct utilisation of the model may not be practically feasible.

A modelling approach as those presented by Plamondon and Lorette [2] in the evaluation of various handwriting generation models had also been considered, but such an approach was seen to be deficient in two aspects if applied to signature verification. Firstly, as with the problems encountered in the prior analysis, the inputs to the handwriting generation models are artificially defined, since the physical neural signals are almost impossible to quantify – the lack of a robust input definition would constitute an inherent weakness in the eventual DSV algorithm.

Secondly, even if it were assumed that the convention of a plant input model in the form of a train of step inputs of various amplitudes and durations was adopted, the DSV algorithm would remain incapable of dealing effectively with time distortions in the dynamic profiles of a signature. This dependence on signal shape was identified as a weakness in conventional DSV algorithms in the introduction of this document. To illustrate this point, consider the following example:

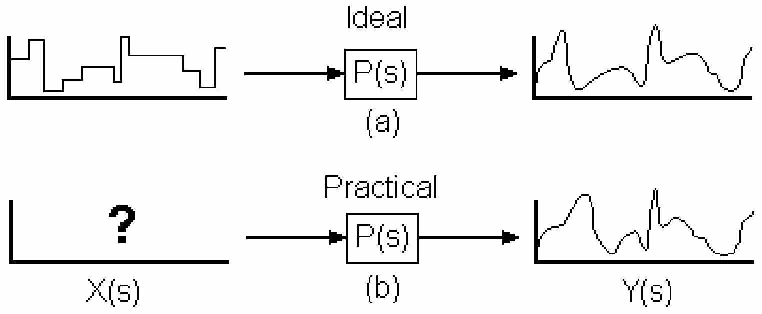


Figure 22 Solving a three variable system

To validate any one quantity in a three variable problem, such as shown in Figure 22, at least two of the variables must be known or 'clamped'. If the handwriting generation model P(s) is to be validated i.e. there is a template of P(s), then both Y(s) and X(s) would be needed. Now, given that Y(s) is the pen-tip trajectory and is thus always available, where from then should X(s), the modelled input, be retrieved? Should X(s) also be stored on a template or should it be generated on the run?

If X(s) is generated on the run, since the *real* X(s) is inaccessible, any X(s) could then be artificially manufactured to result in a P(s) that is close to the template, thereby defeating the purpose of verification. If X(s) were stored on a template, how would one then cater for the time distortion in Y(s) without resorting to time warping techniques as illustrated in Figure 22 (b)? These are some of the shortcomings that made this approach of plant modelling problematic in the research.

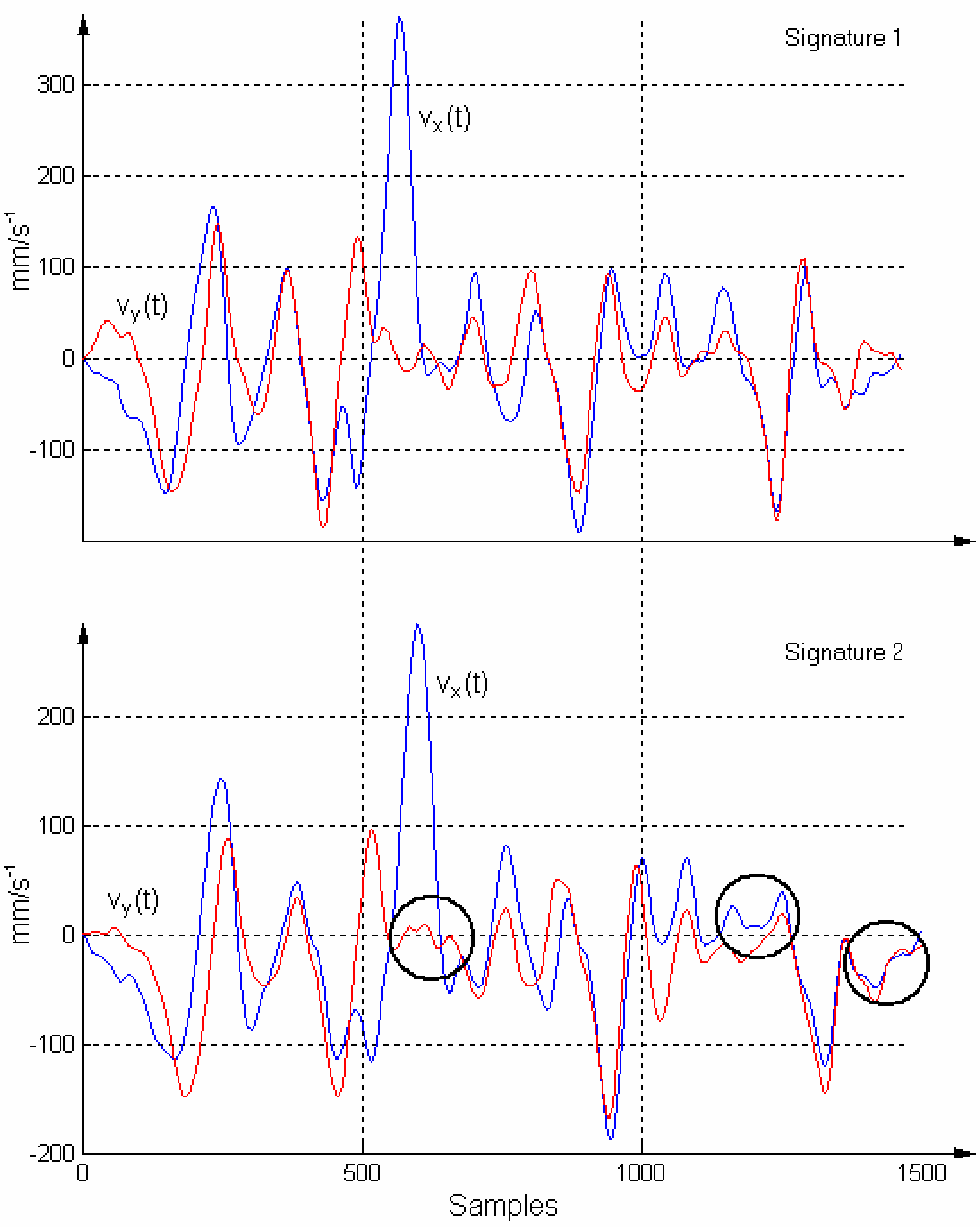


Figure 23 Time domain plots of $v_x[n]$ and $v_y[n]$

Fortunately, since ballisticity implies that the x and y components of pen-tip dynamics are strong functions of both the shape of region C and the signer's plant characteristics, attention was turned to the possibility of inferring information about such underlying consistencies from the plant outputs, which are easily accessible e.g. x(t), y(t), $v_x(t)$ and $v_y(t)$. Manifestation of such consistency was indeed observed, not specifically in the time domain profile of the plant outputs, but rather in the tight and consistent coupling between the behaviour of the x and y components of the pen-tip velocity. It was noted during the analyses of various signature profiles that albeit there may be local inconsistencies in the dynamic profiles of signatures, the x and y components of the pen-tip velocity exhibited strong 'correlations' between each other. Refer to Figure 23.

Figure 23 shows the dynamic profile of two signatures. Clearly, although the two signal pairs exhibit gross consistency, there are visible local inconsistencies in the graphs as highlighted by the circles, as well as slight misalignments of corresponding features within the graphs. However, although the time domain profiles of pen-tip dynamics may appear to be largely consistent, they inevitably suffer from *local time warping* as well as *distortion* due to affine scaling of signatures. The coupling or *covariation* relationship between the pen-tip velocity profiles on the other hand is however *invariant* to such occurrences for a given signature, and this can only be explained to be a manifestation of the underlying consistency embodied by the differentiating entity.

The covariance function between $v_x[n]$ and $v_y[n]$ would allow observers to look beyond a signal's surface attributes with the view that signals are self-contained entities, but rather into the underlying consistency in the generation of the signals. The gist of the argument is that a systems approach to DSV does not attach importance to the signals themselves, but rather to the characteristics of the underlying plant from which the signals originated.

Due to the strong co-varying relationship between $v_x[n]$ and $v_y[n]$, as well as the fact that derivative signals do not suffer from low-frequency trend problems as would be encountered with the use of x[n] and y[n], it was considered the most convenient to treat the $v_x[n]$ and $v_y[n]$ signals as an input-output pair to a plant G(z) i.e.

$$V_x(z) = G(z)V_y(z)$$

What this essentially points to is that although the physical plant and controllers may not be directly modelled, the plant dynamics can nevertheless be implicitly modelled as the characteristic covariation function between the two orthogonal vectors of pen-tip movement. In other words, this is an indirect and alternate manner of quantitative modelling of the controller-plant combination.

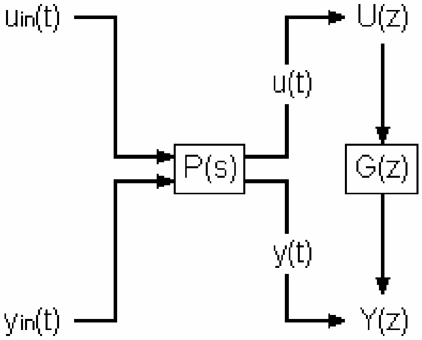


Figure 24 Alternate modelling of the controller-plant combination

$u_{in}(t)$ and $y_{in}(t)$ in the above diagram represents two simultaneous inputs to a plant P(s), which accordingly outputs u(t) and y(t). If P(s) is approximately LTI in the operating region corresponding to a specific range of values of the states u_{in} and y_{in} i.e. the instantaneous values of $u_{in}(t)$ and $y_{in}(t)$, then provided that the relationship between $u_{in}(t)$ and $y_{in}(t)$ is consistent, the relationship G(z) between U(z) and Y(z) would also, logically, be consistent. Conversely, if either the covariation between $u_{in}(t)$ and $y_{in}(t)$ or the plant characteristic in P(s) changes, then G(z) would also change. Recall that it was shown in Section 2.2.4 that it is highly unlikely that the application of a foreign objective function to a forger's controller-plant combination would result in u(t) and y(t) that is similar to those of a genuine signature.

With the above plant definition, the intended way to implement the system was to determine a template G(z) from a training set of signature specimens from each user, and in subsequent verification attempts, u(t) would be applied to G(z) to produce a corresponding simulated predicted output $y_{predicted}(t)$. If the same covariation function G(z) had been obeyed during the signing process, then $y_{predicted}(t)$ would be similar to the actual y(t), if not, then the signature was either badly signed, or is in fact a forgery. Consequently, with the above intended usage of G(z), the model G(z) then constitutes the signature representation in the system framework of DSV, which, by comparison to the representation methods described, is clearly a parameter based representation.

A number of inherent advantages can be reaped from this solution. Firstly, the plant model G(z) inherently caters for affine scaling of signatures, this is because G(z) is an LTI system and an increase in the signature size (and thus pen-tip velocity) would be automatically mapped to a corresponding increase in the output amplitude, thus no spatial normalisation would be needed.

Secondly, a phenomenon often found in signatures – as also noted by Nalwa [11] – is the occurrence of extra or missing loops, is also catered for by using G(z). In Figure 25, although all three signatures had been produced by the same person, parts (b) and (c) have missing and extra loops respectively, and a good algorithm should qualify all three signatures as genuine signatures. Since the covariation in the x(t) and y(t) corresponding to the extra and missing loops are similar, an additional feature in x(t) would produce a corresponding additional feature in the predicted output $y_{predicted}(t)$, thus the comparison between $y_{predicted}(t)$ and y(t) would not be affected.

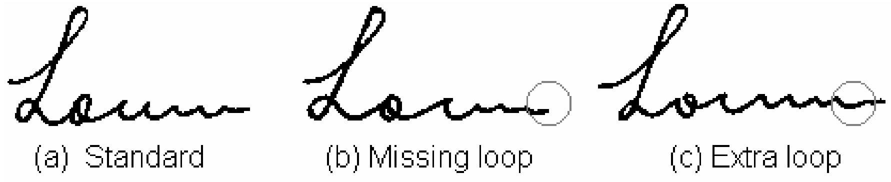


Figure 25 Extra and missing loops

Thirdly, because G(z) models the covariation function between the orthogonal pen-tip vectors, deviations from the normal form of a signature are easily detected. For example, if Figure 26 (a) represents a genuine signature, while Figure 26 (b) represents a forgery, then because the latter is more slanted than the former, the occurrences of maxima and minima in x and in y and thus in v_x and in v_y would be phase shifted compared to the $v_x[n]$ and $v_y[n]$ profile of the genuine signature. Such a phase shift would require a different description of G(z) from the genuine signature, thus deviations such as the slant of handwriting can be easily detected during comparison.

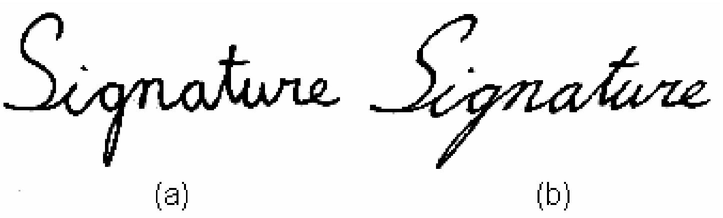


Figure 26 Slant in Handwriting

Fourthly, with a similar argument to the one presented above, no additional mechanisms would be required to cater for random forgeries and skilled forgeries with significant distortion in either shape or timing. Genuine signature with timing variations within the normal intra-personal range however would be easily accommodated, since it is the dynamic and not static relationship of the signatures that is used.

In terms of philosophy of approach, Hidden Markov Models (HMM) [25] and Linear Predictive Coding (LPC) are probably the closest to approaching the problem from a system's approach. In both approaches, an attempt is made in attaining the characteristics of underlying entity that produced the signature outputs. In the case of the former, a model is built for a Markov system that is likely to have generated observation sequence [26]. In the latter case, an equation is derived expressing the fundamental relationship between a signal's present and future values by regression, which has a corresponding Z or Cepstral transform [17]. The commonality here is that the approaches look at the fundamental generation mechanism beyond the surface characteristics of the signals.

Due to the definition of G(z), there are no theoretical grounds to guarantee the uniqueness of G(z) for a given sample of user population, yet due to the variety of signature forms, it is practically impossible for two users to have identical parameters for G(z). Furthermore, even the concurrent existence of two or more identical G(z) does not constitute a compromise to the overall system security, since the only threat would be to have one user with a particular G(z) to submit a random forgery in place of the signature of another user with an identical G(z) – the probability of having all system components to be in a state that is conducive to the occurrence of such a situation is practically zero[22].

In summary, Section 5 presented the function-based and parametric forms of signature representation found in literature, and discussed the implementation considerations pertinent to each format. Furthermore various examples of both function and parameter-based signature representations were also presented via brief reviews of individual algorithms in the first part of the section.

In the second half of the section, an analytical approach was presented in investigating possible and viable ways to derive explicit models of the signature process, which can be used as signature representations. Through a process of critical analysis and elimination, it was realised that direct modelling of the signature generation mechanism was not practically feasible. Nevertheless, an indirect manner of modelling was proposed, where models would be derived for the consistent covariation relationship between the x and y components of the pen-tip velocity. The benefits of such an approach were also explored, and it is clear that whatever the basis of the models used, the ultimate template would be that of a parameter-based one, since core information of signatures are captured by means of plant parameters.

[22] This is in the context of system error as a result of non-unique G(z), ignoring the effects of FRR and FAR.

The tasks of model selection, derivation, validation etc. necessary for the building of a signature representation are detailed in the following section.

6. Systems Identification

In the development of plant models, a distinction should be made between *plant modelling* and *systems identification*. In the former case, modelling begins with the modularisation of an overall system into subsystems whose properties are well-understood e.g. mathematical relationships and laws of nature derived from earlier empirical work. The models of these subsystems are in turn joined to obtain the dynamic model of the overall system. With systems identification however no understanding of the internal mechanisms of the process are necessarily in place, and a mathematical model of the process is inferred purely from input-output data sets based on experimentation [27, 28].

Clearly, the determination of G(z) from $v_x[n]$ and $v_y[n]$ is a systems identification problem, since for any given signature there is no prior knowledge regarding the covariation relationship between $v_x[n]$ and $v_y[n]$. The specific procedure followed for obtaining G(z) via systems identification is illustrated graphically in Figure 27, which forms the framework of discussion for the current section.

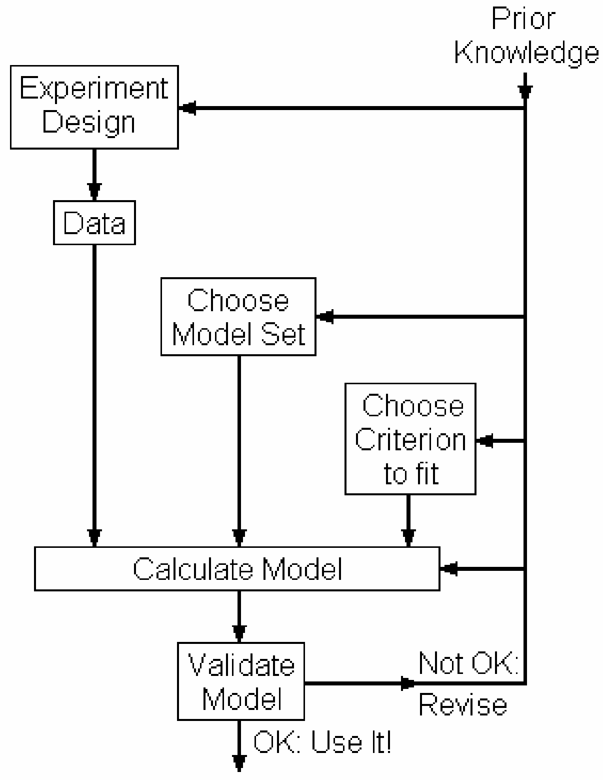


Figure 27 The system identification loop [28]

6.1. Application of Models to the DSV Problem

Although it was stated earlier that there is no a priori knowledge of the relationship between $v_x[n]$ and $v_y[n]$ for a given signature, the prior knowledge input of Figure 27 can nevertheless encompass other 'initial conditions' for the systems identification effort. One such condition adopted was the fundamental assumption that even though the covariation function **F**

$$y = \mathbf{F}(u) \qquad \text{or} \qquad v_y = \mathbf{F}(v_x)$$

may be non-linear, G(z) can still be deemed to be approximately LTI within given constraints[23]. This assumption was necessary to simplify the task of determining G(z) for each signature, since techniques in estimating LTI plant parameters are well established, while the freedom to allow for any format of non-linear plant format would be too divergent a problem to address given the constraints of the research.

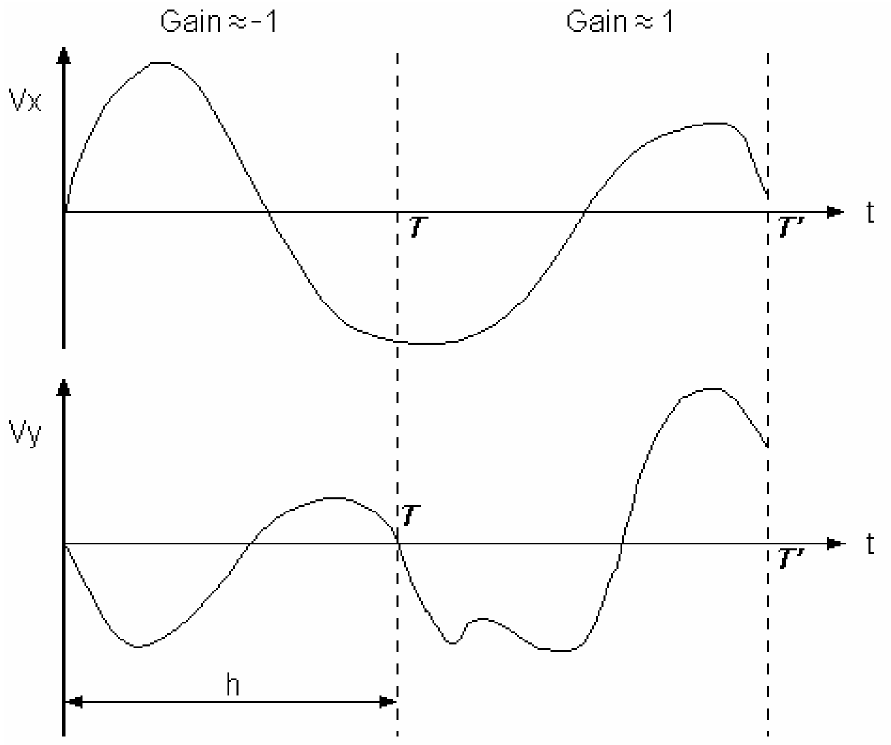


Figure 28 System identification constraints

To appreciate the plausibility of the assumption, consider a $v_x[n]$ and $v_y[n]$ signal pair as shown in Figure 28, assume that $v_x[n]$ is the input to G(z) and $y_x[n]$ the corresponding output. If a plant G(z) were to be inferred from the signals in their entirety, the resultant model would need to be moderately complex to account for all behaviour observed. Specifically, during time $0 < t < T$ the required gain of the plant is approximately –1 as indicated, since the movement in v_x and v_y are approximately directly opposite in direction, yet in time $T < t < T'$, the required gain is approximately 1 as the signals begins to fall into synchronisation.

Clearly these are two conflicting requirements for a single plant, where a 'best' solution may be a pole-zero combination that would result in initial strong negative gain dominance that decays with increasing t, and also persistent parts with positive gain to account for the behaviour during $T < t < T'$. The conclusion is that the problem may be a tricky one to solve, especially in view of the fact that there is no guarantee that a $v_x[n]$ and $v_y[n]$ pair in their entirety obeys a linear covariation relationship.

If, however, the signals were segmented into two smaller parts in the intervals $0 < t < T$ and $T < t < T'$ respectively, then the problem becomes very much simpler: the lessened amount of dynamic behaviour each of the plants $G_1(z)$ and $G_2(z)$, say, must now account for corresponds to less demanding constraints on both the magnitude and phase characteristics of the plants. This is

[23] To be elaborated upon shortly.

accompanied by a corresponding increase in the amount of dynamics accountable by an LTI model, since the magnitude and phase behaviours of an LTI system are bound by their inter-relationship as specified by Bodes laws [29].

Following the same reasoning, in the limit that the segmentation length h → 0 in Figure 28, all dynamic behaviour on $G_1(z)$ and $G_2(z)$ would disappear, and the plants reduce to single gains with their phases being 0° or -180°, in which case the plants are then exactly linear and the description of the signal becomes exactly accurate i.e.

$$G_1(z) = G_2(z) = \alpha K, \qquad \alpha \underset{.}{=} 1 \ \text{ or } \alpha = -1, K \geq 0$$

Equation 7 Plant form as segment length h → 0

On the other hand, as h → ∞, the unconstrained behaviour of the covariation between v_x and v_y would bring into scope increasing amount of unrealistic magnitude and phase requirements under the framework of Bodes laws. Under such conditions, a single *finite* order LTI plant description would accordingly become increasingly inadequate in accounting for the covariation behaviour.

Realising that it is unlikely that the relationship between $v_x[n]$ and $v_y[n]$ could be accurately modelled by a single plant description, it was decided to adopt signal segmentation as a standard procedure to facilitate more meaningful and accurate modelling i.e.

$$\mathbf{F'}(v_x) = \begin{cases} G_1(z) \approx \mathbf{F}(v_x) & t_0 < t < t_1 \\ G_2(z) \approx \mathbf{F}(v_x) & t_1 < t < t_2 \\ \vdots & \vdots \\ G_n(z) \approx \mathbf{F}(v_x) & t_{n\text{-}1} < t < t_n \end{cases}$$

Equation 8 Piece-wise modelling of covariation

where the overall template $\mathbf{F'}(v_x)$ is formed as a combination of the models $G_1(z)$ to $G_n(z)$ for an n segment signal. A number of tradeoffs amongst various design considerations also emerged from the necessity for segmentation, which are summarised in Table 2.

From the above discussions and also Table 2, an ideal h in each case would thus be the maximum h over a segment length where a plant would not have to account for 'significant' levels of non-linear behaviour that is unaccountable for by a LTI model.

Table 2 System identification trade off

	… with increasing h	Ideal design action
Plant complexity	Increases	Reduce h to reduce plant size and storage
Number of plants	Decreases	Increase h to reduce memory requirement
Prediction accuracy	Decreases	Reduce h to increase discriminancy
Dynamic information	Increases	Increase h to increase discriminancy

Having established the conditions surrounding the adequacy of LTI models for signature description, the next two steps according to Figure 27 were experiment design and data considerations. The primarily concern with design of experiments and data is in the yielding of experimental data that is rich in information and conducive to obtaining a good model. Factors of interest under this topic include conditions of experimentation, types of system inputs used, choice of sampling frequency, data manipulation, analysis of identifiability, convergence and consistency etc. In the context of the current application of system identification, these steps corresponds to the adoption of specific definition of the inputs, outputs and plant, which are (respectively) $v_x[n]$, $v_y[n]$ and the coupling thereof.

6.2. Choice of Model Structure

The two main families of models encountered in literature were the *transfer function* and *state-space* models. Linear transfer function models in the discrete time domain capture the dynamics of systems using polynomial coefficients of the shift operator *q* associated with the system input, outputs and disturbances. Under this framework, a number of variants can be identified. One simple discrete time transfer function models is the ARX or the AutoRegressive model with eXogeneous inputs [28], the format of which is given below:

$$y(t)+a_1y(t-1)+\cdots+a_ny(t-n)=b_1u(t-1)+\cdots b_mu(t-m)+e(t)$$

Equation 9 The ARX model structure

or

$$A(q)y(t) = B(q)u(t) + e(t)$$

where

$$A(q)=1+a_1q^{-1}+\cdots+a_nq^{-n}, \qquad B(q)=b_1q^{-1}+\cdots+b_mq^{-m}$$

with disturbance influence e(t) and discrete values of t. The above can alternatively be expressed as

$$y(t) = G(q)u(t) + H(q)e(t)$$

Equation 10 Alternative expression of the ARX model structure

where

$$G(q)=\frac{B(q)}{A(q)}, \qquad H(q)=\frac{1}{A(q)}$$

The model states that the output y(t) at an given instant is a linear combination of past inputs, outputs as well as any disturbance influences that are effected via H(*q*) i.e. the disturbances go through the same denominator dynamics as the input u(t) – an assumption which may be too simplistic. In order to allow for more freedom in the description of the disturbance term, an independent polynomial in the shift operator *q* may be assigned to e(t) to obtain

$$y(t)+a_1y(t-1)+\cdots+a_ny(t-n)=b_1u(t-1)+\cdots b_mu(t-m)+e(t)+c_1e(t-1)+\cdots c_ke(t-k)$$

Equation 11 The ARMAX model structure

This is known as an ARMAX model, where the MA refers to the Moving Average part associated with the coefficients of e(t). As before, the model can be rewritten as

$$\mathrm{A}(q)\mathrm{y(t)} = \mathrm{B}(q)\mathrm{u(t)} + \mathrm{C}(q)\mathrm{e(t)}$$

Equation 12 Alternative expression of the ARMAX model structure

where

$$\mathrm{C}(q) = 1 + \mathrm{c}_1 q^{-1} + \cdots + \mathrm{c}_\mathrm{k} q^{-\mathrm{k}}$$

and thus

$$\mathrm{y(t)} = \mathrm{G}(q)\mathrm{u(t)} + \mathrm{H}(q)\mathrm{e(t)}$$

where

$$\mathrm{G}(q) = \frac{\mathrm{B}(q)}{\mathrm{A}(q)}, \qquad \mathrm{H}(q) = \frac{\mathrm{C}(q)}{\mathrm{A}(q)}$$

Although the disturbance term can be described with varying levels of freedom with the above models, both u(t) and e(t) still undergo identical denominator dynamics – a restriction that may not be sufficiently flexible for practical applications. A more generic description of dynamics for both u(t) and e(t) is provided for by the well-known Box-Jenkins model structure [28]

$$\mathrm{y(t)} = \frac{\mathrm{B}(q)}{\mathrm{F}(q)}\mathrm{u(t)} + \frac{C(q)}{D(q)}\mathrm{e(t)}$$

Equation 13 The Box-Jenkins model structure

where the dependence of y(t) on the system inputs and disturbances are essentially described by two separate transfer functions in the shift operator *q*. The above three model structures represent the three main types of discrete transfer function models frequently encountered, but they remain special cases of the truly generic transfer function model structure [28]

$$\mathrm{A}(q)\mathrm{y(t)} = q^{-\mathrm{r}}\frac{\mathrm{B}(q)}{\mathrm{F}(q)}\mathrm{u(t)} + \frac{C(q)}{D(q)}\mathrm{e(t)}$$

Equation 14 The generic transfer function model structure

where each of A(q), B(q), C(q), D(q) and F(q) have no global common factors, and the q^{-r} term accounts for the possible delays of r samples from u(t) to y(t).

The main advantages of transfer function models are that their structures are simple, and the plot of the zeroes and poles constituting the numerator and denominator polynomials in the Z plane can often provide good clues on the dynamic behaviour of the system upon inspection. Furthermore, frequency domain methods are able to indicate relative stability directly, and provide a deep understanding of the system behaviour while requiring only simple calculations [30].

An alternative way of representing model structures is with the use of state-space models, where the relationship between the input, disturbance and output signals is written as a system of first order differential or difference equations using a supplementary state vector **x**. In the current context, the discrete time state-space model is more relevant, since all signals concerned are discrete in nature. The standard discrete-time state-space model is

$$\begin{aligned}\mathbf{x}[n+1] &= \mathbf{A}\mathbf{x}[n] + \mathbf{B}\mathbf{u}[n] + \mathbf{w}[n]\\ \mathbf{y}[n] &= \mathbf{C}\mathbf{x}[n] + \mathbf{D}\mathbf{u}[n] + \mathbf{v}[n]\end{aligned}$$

Equation 15 General discrete-time state-space model

where **A** is the state matrix, **B** the input matrix, **C** the output matrix and **D** the direct transition matrix. The vector **w** and **v** models process and measurement noises respectively, which are often assumed to be white noise with variance λ.

The major advantages of state-space representations on the other hand are that they allow models to be easily set up using equations representing laws of physics, so that the elements of the state vector **x** and of the matrices **A**, **B**, **C** and **D** are often physical variables of a real system – this makes the behaviour and relationships amongst various parameters transparent. Furthermore, state-space models are able to accommodate non-linear behaviour of systems, as opposed to transfer functions models, which are limited to the LTI domain.

The choice of a suitable plant for the piece-wise modelling of the covariation function **F** relies on the properties of the system to be modelled. Now, the process from $v_x[n]$ to $v_y[n]$ can be viewed as a single-input-single-output (SISO) one, for which no fundamental governing equations are known and no disturbance sources can be defined. These, in conjunction with the fact that no non-linearities are being considered imply that state-space models have no definitive advantages over transfer function models for the problem at hand. Given this, the remaining design considerations are then solely implementation based, specifically in terms of the complexity, data structuring and memory requirement implications of a given model.

Consider an n^{th} order transfer function model and a corresponding n^{th} state-space model describing a common process. On the one hand, a state-space model of a dynamic system depends entirely on the choice of state variables, thus an infinite number of state variable combinations and hence state-space models can be obtained for a single system. On the other hand, transfer function models are unique since the use of a different set of model parameters would describe an entirely different dynamic behaviour.

Now consider a typical n^{th} order SISO discrete time state-space model parametrised by the index k

$$\begin{aligned}\mathbf{x}[k+1] &= \mathbf{A}\mathbf{x}[k] + \mathbf{B}\mathbf{u}[k]\\ \mathbf{y}[k] &= \mathbf{C}\mathbf{x}[k]\end{aligned}$$

one is able to derive a Z transform transfer function model of the system from the discrete state-space model through a transformation matrix **T** given by [31]

$$\mathbf{T}=\begin{bmatrix}\mathbf{C}\\ \mathbf{CA}\\ \vdots\\ \mathbf{CA}^{n-1}\end{bmatrix}$$

With the matrix **T**, a new state-space model

$$\begin{aligned}\mathbf{x}^*[k+1]&=\mathbf{A}^*\mathbf{x}^*[k]+\mathbf{B}^*\mathbf{u}[k]\\ \mathbf{y}[k]&=\mathbf{C}^*\mathbf{x}^*[k]\end{aligned}$$

can obtained, where

$$\begin{aligned}\mathbf{x}^*&=\mathbf{Tx}\\ \mathbf{A}^*&=\mathbf{TAT}^{-1}\\ \mathbf{B}^*&=\mathbf{TB}\\ \mathbf{D}^*&=\mathbf{DT}^{-1}\end{aligned}$$

The coefficients of A(q) and B(q) of Equation 12 for the transfer function are then obtained using

$$\mathbf{A}^*=\begin{bmatrix}0 & & & \\ \vdots & & \begin{bmatrix} & \mathbf{I} & \end{bmatrix} & \\ 0 & & & \\ -a_n & -a_{n-1} & \cdots & -a_1\end{bmatrix}$$

$$\begin{bmatrix}b_1\\ b_2\\ \vdots\\ b_n\end{bmatrix}=\begin{bmatrix}1 & 0 & 0 & \cdots & 0\\ a_1 & 1 & 0 & & \vdots\\ a_2 & a_1 & 1 & \ddots & 0\\ \vdots & & \ddots & \ddots & 0\\ a_{n-1} & a_{n-2} & \cdots & a_1 & 1\end{bmatrix}\mathbf{B}^*$$

where for the SISO case, **B*** is a n × 1 column vector. Having determined a_1 to a_n and b_1 to b_n, the Z transform is

$$G(z)=\frac{b_1z^{-1}+\cdots+b_n}{1+a_1z^{-1}+\cdots+a_nz^{-n}}$$

Equation 16 Derivation of transfer function model from state-space model

In comparing the information content within state-space and transfer function models, recall firstly that since for any given system there are infinitely many possible state-space models while there is only one unique transfer function model – the mapping from state-space models to a transfer function is thus many-to-one. Secondly, since the number of roots in the denominator of G(z) in Equation 16 are directly related to the size of the state matrix **A**, as evident from the above derivation, there would always be as many poles as there are states in the state vector $\dot{\mathbf{x}}$ i.e. the order of a transfer function model and its corresponding state-space model are directly matched.

What these two points show is that for a given order model, the information content in a transfer function model and state-space model are equivalent, but the information density is much lower in a state-space model due to the redundancy arising from the many-to-one mapping relationship between the two model structure formats. Specifically, for a realisable nth order SISO system, a single transfer function model e.g. Equation 9 with normalised coefficients would need only 2n model parameters to be stored, while a state-space model would need the storage of $n^2 + 2n + 1$ model parameters (refer to Figure 29). In view of these and the properties of the problem described earlier, it was clearly that for a SISO system, a transfer function model was by far be the more suitable option for the storage of plant models in computer memory.

$$H(s) = \frac{b_{n-1}s^{n-1} + \cdots + b_1 s + b_0}{s^n + a_{n-1}s^{n-1} + \cdots + a_1 s + a_0}$$

n coefficients

+ n coefficients (a_n = 1)

2n

$$\dot{x}(t) = Ax(t) + Bu(t)$$
$$y(t) = Cx(t) + Du(t)$$

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \vdots \\ \dot{x}_n(t) \end{bmatrix} = \overset{n \times n}{\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & & a_{2n} \\ \vdots & & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}} \begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_n(t) \end{bmatrix} + \overset{n \times 1}{\begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}} u(t)$$

n^2+ n parameters

$$y(t) = \overset{1 \times n}{\begin{bmatrix} c_1 & c_2 & \cdots & c_n \end{bmatrix}} \begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_n(t) \end{bmatrix} + \overset{1 \times 1}{\begin{bmatrix} d_1 \end{bmatrix}} u(t)$$

+ n + 1 parameters

n^2 + 2n + 1

Figure 29 Storage requirement of transfer function and state-space models for a SISO system

Having decided upon the use of a transfer function model, MATLAB's System Identification Toolkit was used as a result of its rich set of model estimation functions.

6.3. Concepts of Model Calculation, Fitting Criterion and Model Validation

To investigate the effects of the various transfer function model on accuracy and algorithm performance, a logical step was to begin the study with a simple transfer function model, specifically the ARX model of Equation 9. Data from a number of signatures were retrieved from numerous signers and used as input-output pairs for the estimation and validation of model parameters. As mentioned before, $v_x[n]$ was arbitrarily chosen to be the plant input and $v_y[n]$ the plant output, a convention adopted in all subsequent system identification efforts. The choice of which data series to treat as input and which as output is really inconsequential, since the simple implication of a reversal of definition would be inversion of a transfer function model, and would not alter the algorithm fundamentally.

For the estimation of an ARX model, MATLAB's `arx` function was used. The derivation of a model using the `arx` function begins with data treatment. First, x[n] and y[n] signals such as the pair shown in Figure 30 are differentiated to obtain the velocity profile as shown in Figure 31. Clearly, at points of discontinuity in x[n] and y[n], spikes are found at the corresponding points in $v_x[n]$ and $v_y[n]$. Since such erratic behaviour would degrade the quality of a empirically derived model, the velocity signals are smoothed and off-set removed to obtained the final $v_x[n]$ and $v_y[n]$ signals of Figure 32, from which an `iddata` data object is then constructed.

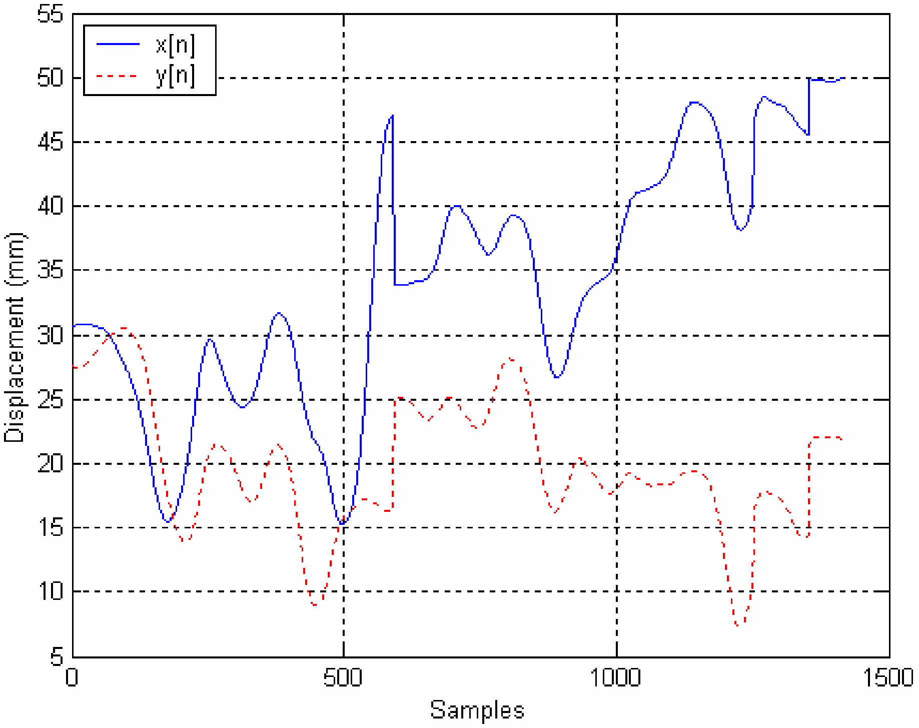


Figure 30 Plot of x[n] and y[n] with zero samples removed

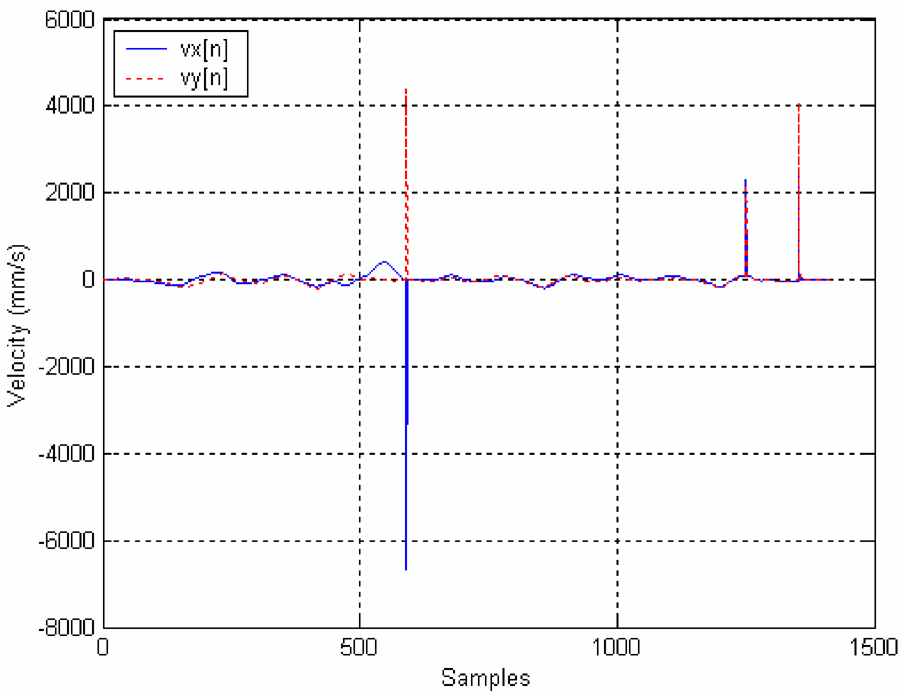


Figure 31 Plot of $v_x[n]$ and $v_y[n]$ before smoothing

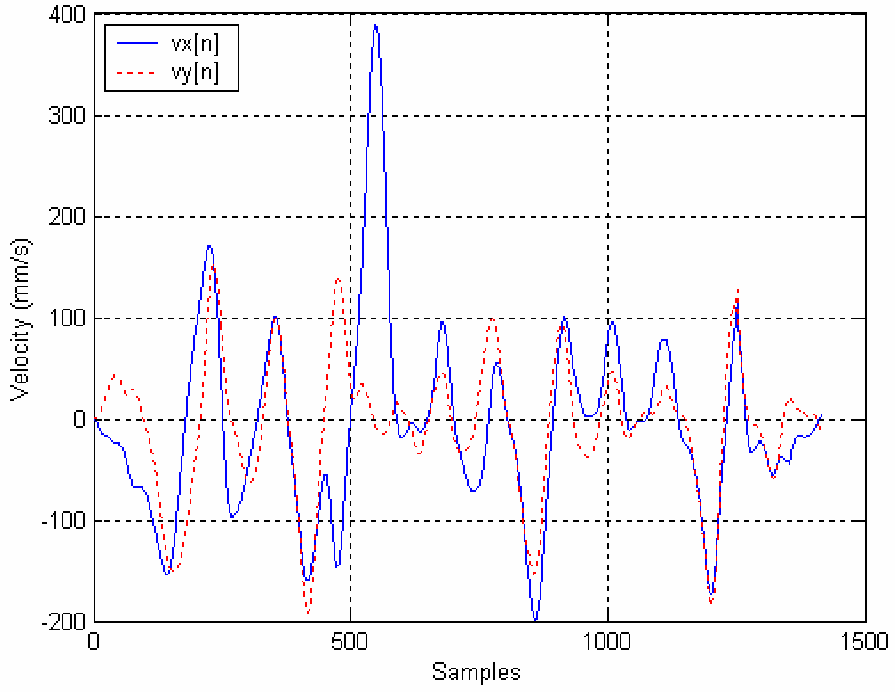


Figure 32 Plot of smoothed and offset removed $v_x[n]$ and $v_y[n]$

By recalling the discussion of Section 2.2.7, it is understandable that the removal of spikes from the original $v_x[n]$ and $v_y[n]$ signals in effect removes the information regarding the relative position of signature segments. This means that only the intra-segment or ballistic parts of a signature is modelled. Acknowledging the importance of the information contained in the spikes, the information lost via signal smoothing would need to be incorporated in the algorithm via additional complementary parameters.

A final point to be noted here is that the velocity profile of Figure 31 and Figure 32 show the actual velocities in the orthogonal directions i.e. at a sampling frequency of 600 Hz, where the derivative of x[n] and y[n] with respect to n are multiplied by 600 to obtain the true velocities. However, because the ARX model structure is LTI, in all subsequent efforts of model derivation the unscaled versions of $v_x[n]$ and $v_y[n]$ are be used, since scaling of inputs is inconsequential on the evaluation of LTI model parameter values.

The next step of model identification is submitting an `iddata` object to the `arx` function, where the model parameters for the ARX model structure are obtained using the well-known least squares method. The least square method (as well as QR factorisation) is presented in detail in Appendix 3 and also addresses the 'choose criterion to fit' box of Figure 27 [32, 33, 34].

In brief, when the number of data samples exceeds the desired order of the model, the least-squares estimation problem becomes an overdetermined set of linear equations with infinite number of solutions of which only one would result in minimum error. This minimum error solution is found using QR-factorisation in the `arx` function [35].

Once an ARX model is found, the accuracy of the resultant model was observed through both simulation and prediction methods. As a result it is instructive to call attention to the distinction between the two methods at this point. The most basic use of a system description is to simulate the system's response to various inputs i.e. the application of a user-defined input to a system description to compute the undisturbed output, mathematically this is shown in Equation 17.

$$A(q)y[n] = B(q)u[n], \qquad t = 1, 2, 3, \ldots, N$$

Equation 17 Model simulation

Disturbances can be incorporated into the simulation by adding to y[n] an e[n] term representing a source of white noise with zero mean and variance λ [28]. The flow of information in a simulation can be visualised as

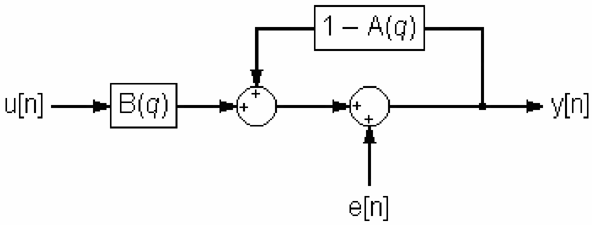


Figure 33 Information flow diagram of a simulation

where only the input u[n] is the *actually* measured signal, whilst both y[n] and e[n] are computationally generated. Note that due to the presence of the feedback loop, it is clear that memory exists in the scheme so that any error found in sample y[k] k = 1, 2, … , N – 1 would be fed back into the system in the next iteration to produce a further error in sample y[k+1]. Specifically, errors can result from inaccuracies in the A(*q*) and B(*q*)'s description of the 'true' system. Consequently, inaccuracies in the parameters A(*q*) and B(*q*) would contribute towards accumulated errors in the simulated y[n], so that the actual and simulated y[n] could be quite different indeed over time.

The key here is that the output y[n] is strictly a function of actual u[n] and simulated y[n], and there is no dependence of the simulated y[n] on the actual y[n].

The method of simulation is the typical and widely practiced use of system models, allowing experiments to be carried out on a model instead of the actual physical system, which is often the safer and more cost effective alternative.

In prediction however, both the actual input and output data of the real system are used to predict future values of the output. The information flow for a general k-step ahead predictor is shown in Figure 34, which shows that there is no feedback mechanism in a prediction scheme. Since both the actual data u[n] and y[n] instead of a feedback loop are used to excite the predictor, inaccuracies in A(*q*) and B(*q*) do not accumulate, and thus it is expected that the predicted output $\hat{y}[n]$ would be a closer approximation to the actual y[n] than the simulated output.

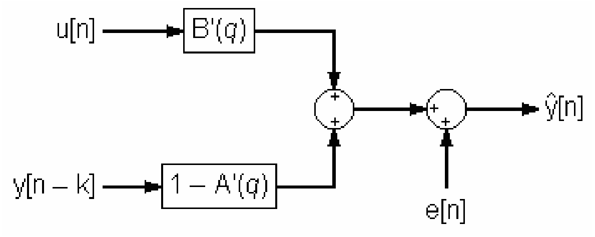


Figure 34 Information flow diagram for a prediction

In Figure 34 an output prediction $\hat{y}[n]$ at time corresponding to index n is generated based on information of all actual input data u[k] up to time k = n – 1, and all actual output data y[k] up to time k = n – k, where k is known as the *prediction horizon* (p.h.). Since A(*q*) and B(*q*) of the original model were not obtained under the conditions surrounding a predictor, the set of polynomials the predictor used are in fact modified versions of A(*q*) and B(*q*) i.e. A'(*q*) and B'(*q*).

Ljung [28] derived the general relationship of Equation 18 between a transfer function model and its k-step ahead predictor polynomial from first principles; the interested reader is referred to [28] for a full derivation. Essentially, for a general transfer function system of Equation 10, the k-step ahead predictor can be derived using

$$\hat{y}(n \mid n-k) = W_k(q)G(q)u[n] + [1 - W_k(q)]y[n]$$

Equation 18 k-step ahead predictor of transfer function models

where

$$W_k(q) = \overline{H}_k(q)H^{-1}(q), \qquad \overline{H}_k(q) = \sum_{m=0}^{k-m} h[m]q^{-m}, \qquad H(q) = \sum_{k=0}^{\infty} h[n]q^{-k}$$

The result is a predictor model with A'(*q*) and B'(*q*), where A'(*q*) have no terms in the most recent k samples of the actual output y[n]. Interestingly, because a simulation cannot use any actual output data, a simulation in fact corresponds to prediction with p.h. = ∞.

An important use of prediction is to evaluate a model's properties in the mid-frequency range. To appreciate this, recall that simulation conceptually corresponds to k = ∞, which due to the existence of e[n] can lead to levels that drift apart i.e. the low frequency behaviour is emphasized. A 1-step ahead prediction on the other hand is not a powerful test of the model's properties either, since the 1-step ahead predictor is a model of output dependence on the few samples *immediately* preceding it i.e. high frequency behaviour is stressed [36]. Where k is an intermediate value between 1 and ∞, the predictor models the dependence of an output sample on dynamics that occurred k samples ago in time, from which the mid-frequency characteristics of the original model can thus observed.

In MATLAB, the `sim`, `predict` and `compare` functions were used to perform the above tasks of simulation and prediction. In particular, the `compare` function also returns a model accuracy measure defined as:

$$P = 100 \times \left\{ 1 - \frac{\sqrt{\sum_{k=1}^{N} \left| \hat{y}[k] - y[k] \right|^2}}{\sqrt{\sum_{k=1}^{N} \left| y[k] - \frac{1}{N} \sum_{k=1}^{N} y[k] \right|^2}} \right\}$$

Equation 19 Definition of similarity distance in prediction

where P is taken as the percentage of dynamic behaviour accountable by the model.

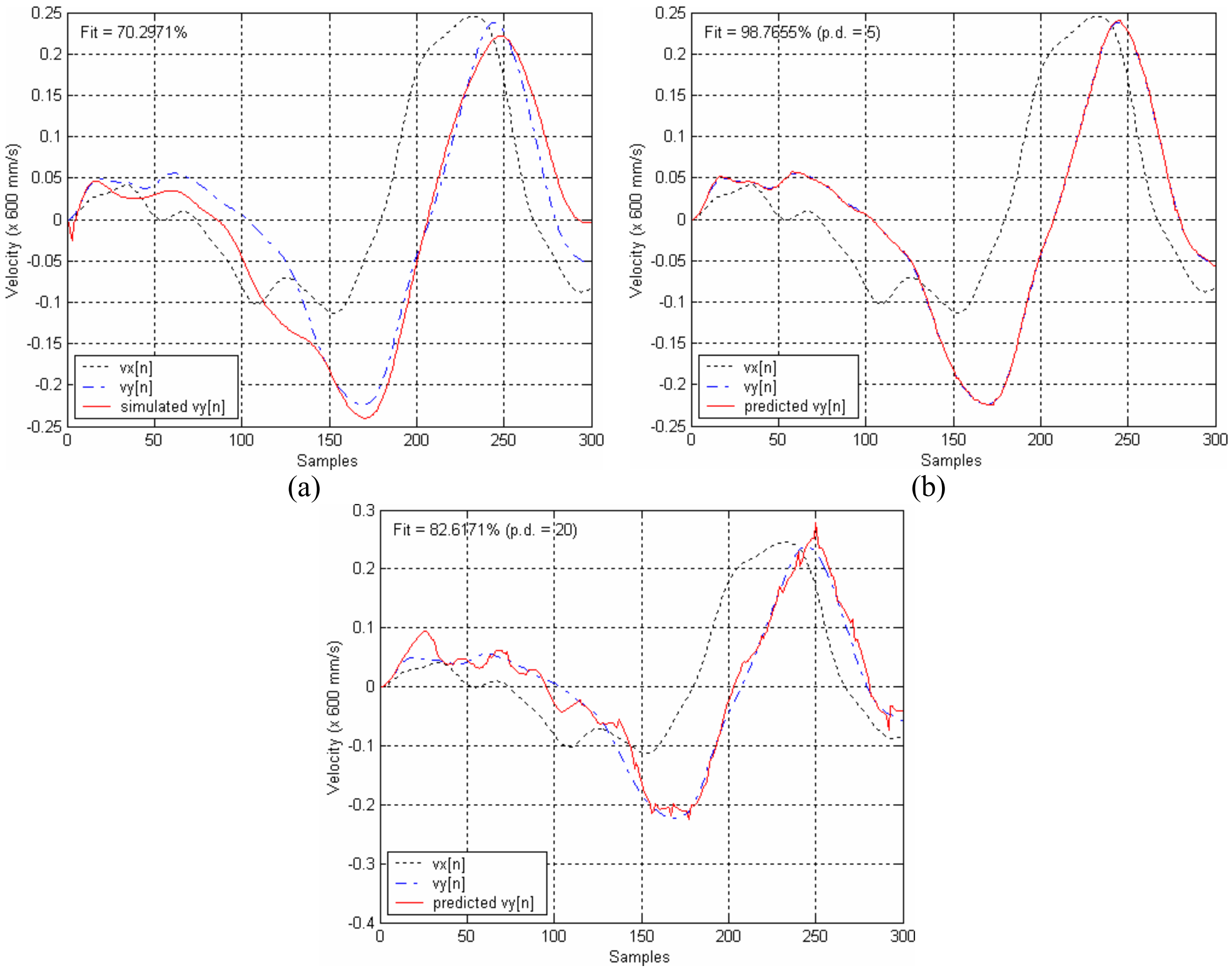
To understand Equation 19, note that the square root term in the numerator of the expression is the Frobenius norm of the N-element vector containing the difference function between the simulated or predicted output $\hat{y}[k]$ and the actual output y[k]. The norm is an indication of the overall magnitude of the difference function, which is assumed to be small for models of high accuracy. In the denominator is the Frobenius norm of the N-element vector containing the individual sample distances of the actual output y[k] from the mean thereof, and is an indication of the general pattern of distribution of y[k] samples.

Equation 19 in its entirety is therefore a measure of the deviation of the modelled behaviour from the actual behaviour by calculating the ratio between the norm of the difference function and the norm of the sample scatter function of the actual y[n]. Clearly, a greater norm of the difference function i.e. low accuracy, would result in a greater numerator and thus a lesser percentage of behaviour accountable by the model. As an example to complement the above explanation, suppose an ARX model with 6 terms in A(q) and 5 terms in B(q) with no input delay is to be derived using the `arx` function for the first 300 samples (0.5 seconds) of the $v_x[n]$ and $v_y[n]$ signals of Figure 32. The resultant plant is:

$$\begin{aligned} &y[n] - 2.388y[n-1] + 1.434y[n-2] + 0.1791y[n-3] - 0.08374y[n-4] - 0.1419y[n-5] \\ &= 0.632u[n] - 1.558u[n-1] + 0.9315u[n-2] + 0.3081u[n-3] - 0.3133u[n-4] + e[n] \end{aligned}$$

Equation 20 Example ARX model

Observe that since the coefficient of y[n] is always normalised to unity, the `arx` function in fact evaluates only five coefficients for the second to the sixth term in the above difference equation. The above model was both simulated and used to derive a predictor with different prediction horizons.



(a)

(b)

(c)

Figure 35 Simulation and prediction results

As expected, the simulation result of (a) yielded fit = 70.2971% according to Equation 19, and was not as accurate as the predicted results. The predicted output shown in (b) was obtained with p.h. = 5, and clearly by applying the predictor to actual observed data instead of simulated data, the outcome is notably more accurate. Using Equation 18, the 5-step ahead predictor derived from Equation 20 was

$$
\begin{aligned}
\hat{y}[n] = &-4.441\times10^{-16}\,y[n-2] - 3.331\times10^{-16}\,y[n-3] + 3.608\times10^{-16}\,y[n-4] \\
&+ 12.25y[y-5] - 13.77y[n-6] - 0.5017y[n-7] + 1.711y[n-8] + 1.315y[n-9] \\
&+ 0.632u[n] - 0.04869u[n-1] - 0.09129u[n-2] + 0.04679u[n-3] - 0.008993u[n-4] \\
&- 7.728u[n-5] + 9.321u[n-6] + 0.7906u[n-7] - 2.902u[n-8]
\end{aligned}
$$

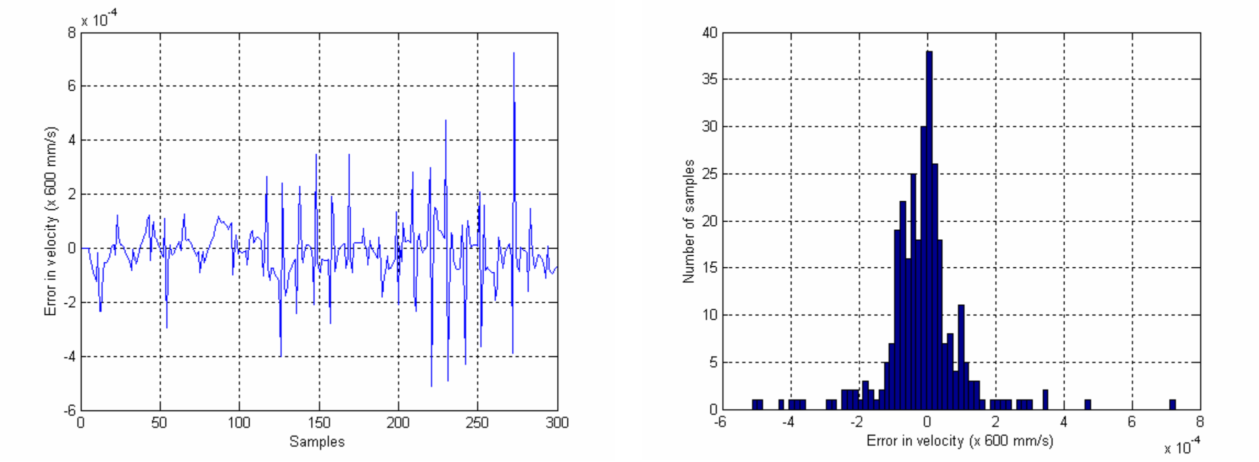
One may question the existence of output terms more recent than y[n – 5] in the 5-step ahead predictor above, but by observing the magnitude of the coefficients of the terms on the first line, it is apparent that they are a result of calculations with finite resolution and can be treated as practically non-existent. Nevertheless, this is something that has to be dealt with in practical implementations of the system.

For illustration, a prediction was also performed with p.h. = 20, and evidently as the p.h. increases i.e. as a prediction approaches a simulation, the accuracy of the prediction decreases. This is consistent with the prior expectation in view of the existence of the e[n] term in the original model. What this shows is that due to e[n], the prediction accuracy is bound to reduce as the dependence relationship of an output sample $\hat{y}$ on past inputs is stretched longer and longer into the past, which accounts for the 'noise' like component seen in the behaviour of predicted output of (c).

The deterioration of accuracy with increasing prediction horizon is an indication that there are errors or disturbances within the system that are not accounted for by the model, and which is lumped in the term e[n]. Now, because the ARX model parameters were evaluated using the least square method, e[n] should theoretically be a random variable, for if this were not so there would still be residual parts of the system's dynamics that have not been captured by the model parameters. To verify this, the e[n] term in Equation 20 was evaluated, plotted and the distribution of its values viewed using a histogram.

By comparing Figure 36 (a) with $v_x[n]$ in Figure 35, it appears that there is indeed no correlation between the behaviour of e[n] and $v_x[n]$. By looking at the histogram of the distribution of only 300 samples of e[n], it is already evident that the distribution is normal with zero mean. From these observations, it was confirmed that e[n] is indeed a random variable or white noise.

For completeness sake, it should be stated that if the model were 100% accurate, then the e[n] would fall away, and all simulation and predictions with different p.h. would produce identical results. However, since it is acknowledged that G(z) in the current context is only approximately linear, e[n] constitutes an integral part of the ARX model structure.



(a) (b)

Figure 36 Analysis of residual error in prediction

6.4. Effect of Model Parameters on Model Accuracy

From repeated experimentation with various model estimation parameter settings, a number of important observations were made, specifically in terms of the effects of model order, input delays, segment length and prediction quality as a function of the prediction horizon.

6.4.1. The Effect of Model Order

Using 10 different sample, signatures systems identification was performed on v_x [n] and v_y[n] signal segments of 100, 200, 300 and 400 samples long and at model orders of 5 to 50 in increments of 5. The similarity distance based on the Frobenius norm described previously was used to assess the accuracy of each model, the results of which are shown below.

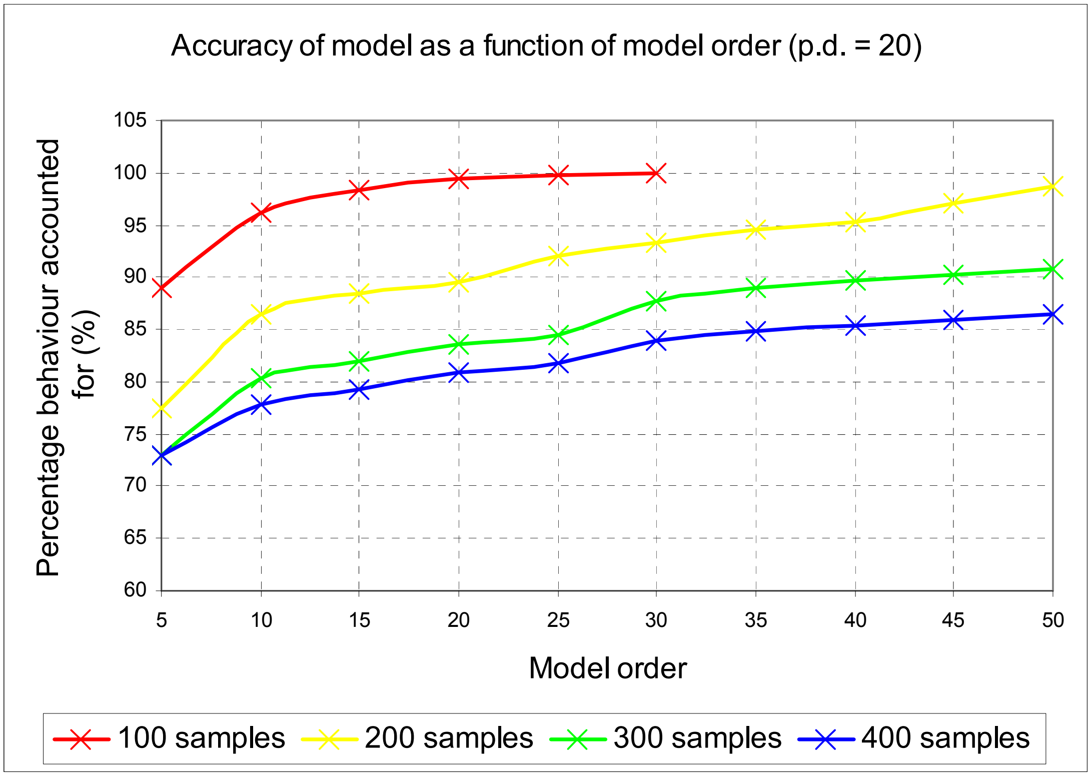


Figure 37 Accuracy of model as a function of model order

Clearly, regardless of the specific signature and the segmentation length used, the accuracy of the models' description of system dynamics is a non-linearly increasing function of model order. The result for the 100 sample long segments shows specifically that little improvement in model accuracy is achieved beyond a 15th order model, after which the accuracy approaches the 100% level almost asymptotically. The lines for the 200, 300 and 400 sample long segments however can be seen to approach the 100% accuracy level much more slowly, at an overall approach rate of approximately +1.74% per additional order.

Nevertheless, it is known from the least square method described in Appendix 3 that the graphs will eventually intersect the 100% line. This would happen when the order of the model used is equal to the number of sample points in the input v_x[n], at which time the LTI model description becomes exact. No plant of higher order than the number of samples in v_x[n] can however be obtained, since the amount of information available would be insufficient for such a quest.

An interesting observation of Figure 37 is that with a model order of 'only' 50, the 400-sample graph can be seen to already have attained an accuracy of approximately 86%, suggesting that the remaining 14% of accuracy has to be achieved using 350 additional orders. This would hardly be worth the extra memory required, and highlights the trade-off mentioned previously between model accuracy and memory requirement.

From the above, it was confirmed that an increase in the model order contributes to increased model accuracy, and a trade off exists between the desired level of accuracy and memory requirement of the template due to the non-linear order-to-accuracy relationship. A point to be borne in mind however is that Figure 37 merely illustrates the *overall* dependence of model accuracy on model order i.e. one cannot quote a generic accuracy value for a given model order by reading off the graph of Figure 37, since in individual cases the effect of signal behaviour on model accuracy far outweighs the effect of the model order chosen.

6.4.2. The Effect of Input Delays

The next consideration investigated was that of the effect of input delays on model accuracy. By input delay, it is meant that the most recent input sample that can affect the output has index n – k, where n represents the present time and k is the amount of delay specified e.g. for an ARX system of input delay k = 3, the system is

$$y[t] + a_1y[t-1] + \ldots + a_ny[t-n] = b_ku[t-k] + b_{k-1}u[t-k-1] + \ldots + b_mu[t-k-m]$$

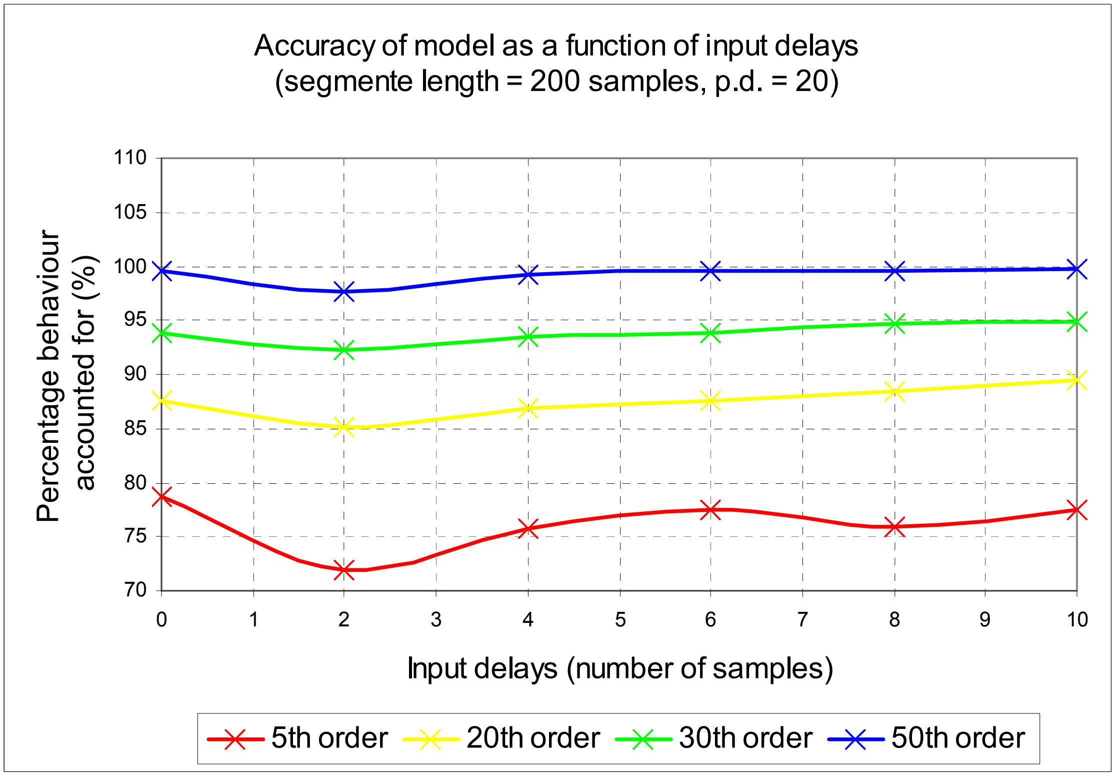


Figure 38 Accuracy of model as a function of input delays

In a similar manner as undertaken in the previous experiment, signatures of various individuals were used to generate various order models with different input delay specifications based on signal segments of 200 samples. Specifically, ARX models with input delays ranging from 0 to 10 in increments of 2 were generated.

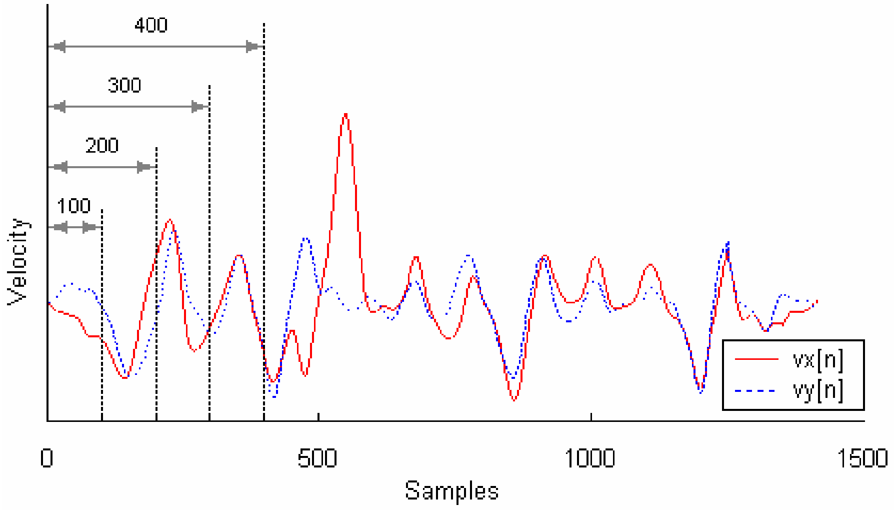
Graphs of model accuracy for 5, 20, 30 and 50th order models are plotted in Figure 38, and it is striking that each graph runs almost horizontally across the figure, suggesting that model accuracy is in fact unaffected by input delays with the definition of G(z) employed in the research. Nevertheless, the finding is understandable by recalling that G(z) is an abstract system, therefore the choice of any input delays can result in a reasonably accurate model, which is emphatically verified by Figure 38.

In view of the above result, it was clear that input delay is not a critical design factor that could significantly influence model accuracy and thus the overall DSV system performance. Consequently, it was decided unnecessary to include input delays in any of the subsequent system identification attempts.

6.4.3. The Effect of Segment Length

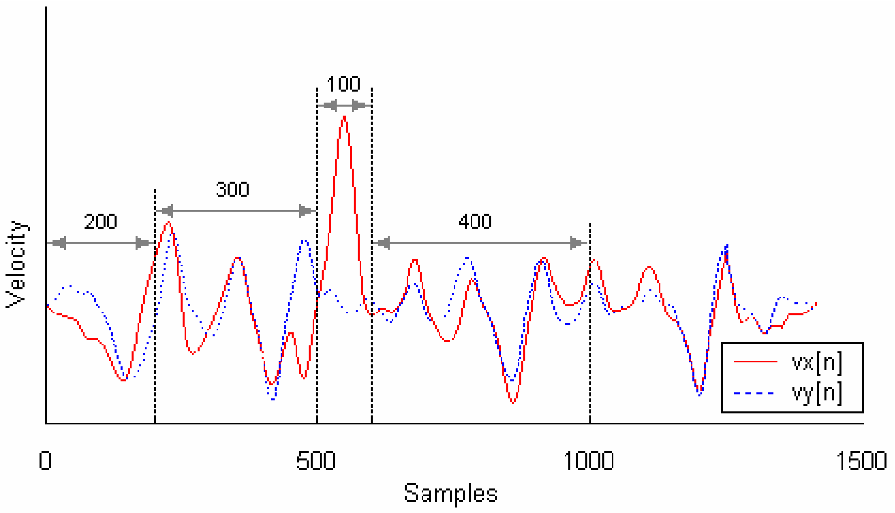
As explained in Section 6.1, the length of signal segment is related (referred to as h henceforth) to the overall absolute level of non-linearity present in the segment, where the LTI approximation G(z) can be exact if $h \rightarrow 0$ and cannot be derived if $h \rightarrow \infty$. The purpose of this section is to explicitly quantify the qualitative behaviour anticipated.

Using the same set of signatures used for the preceding investigations, signal segments were created as shown in Figure 39. The advantage of segmentation according to (a) and not (b) is that it ensured that a longer segment always contained a higher level of non-linearity than the shorter segment in absolute terms, since segmentation according to (b) may very well result in a 100 sample long segment that is highly non-linear and a 400 sample long segment that is unusually linear. This precaution was taken to ensure that the changes in accuracy seen in Figure 40 are purely due to increases non-linear behaviour, and not due to behaviour specific to particular signal segments.



(a)

Figure 39 The correct segmentation of signals



(b)

Figure 39 The correct segmentation of signals (continued)

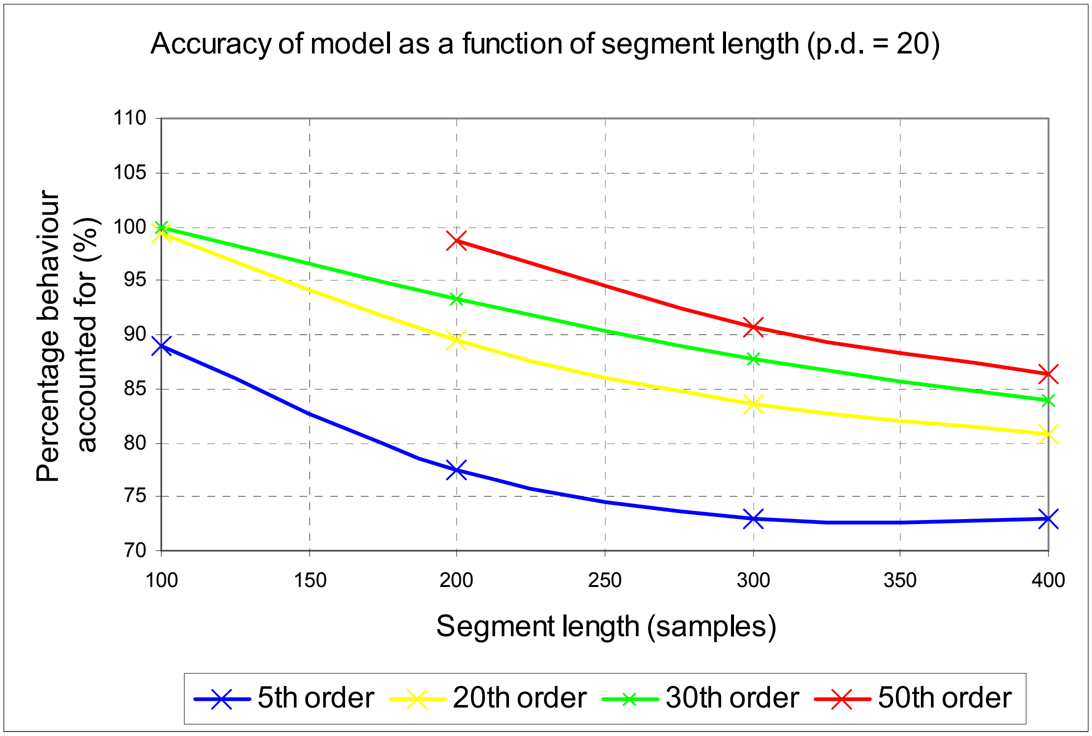


Figure 40 Accuracy of model as a function of segment length

As expected, the accuracy of all given orders of models decreased as h is increased. Also evident is that the initial increase of h from 100 samples to 200 samples resulted in the steepest reductions in accuracy of models of all orders, after which the reduction in accuracy is more gentle – almost resembling exponential decay in accuracy as a function of h. Again, the graphs merely show the *overall* dependency of model accuracy on segment length h i.e. the effect of signal characteristics on accuracy can often outweigh the effect of segment length in individual cases.

Note that a 30th order model can have almost 100% accuracy even for segments 100 samples long, this means that between h of 30 (minimum signal length for a 30th order system) and 100, a 30th order model would, for practical purposes, be an exact description of the system dynamics. The same reasoning can be applied to the models of other orders for other ranges of h.

In the limit of h → ∞ however, no generic statements can be made about the precise value of the fit with the definition of Equation 19. This is so because firstly ARX models are evaluated using the least square method, and the minimum error for different sets of signals would inevitably be different for a given model order. Secondly, there is no reason to assume that the mean of the fit value of all signatures should tend to a specific value when the signal is infinitely long i.e. whereas the fit value for $h = 0$ is clamped at 100%, the fit is not clamped for $h = \infty$, therefore the farther away one is from $h = 0$, the less generic our description of Figure 40 becomes.

6.4.4. The Effect of Prediction Horizon

A final design variable investigated was the suitable choice of a prediction horizon; this is especially relevant as the prediction horizon can be used as a variable threshold during verification. It should be pointed out that the phrase ‘the effect of prediction horizon’ is strictly a misnomer, since the prediction horizon does not influence the accuracy of a model. What is really meant here is that given that the effects of e[n] are acknowledged, and that A(*q*) with B(*q*) alone cannot account for all observed dynamics, how does the predictor quality deteriorate with increased p.h.?

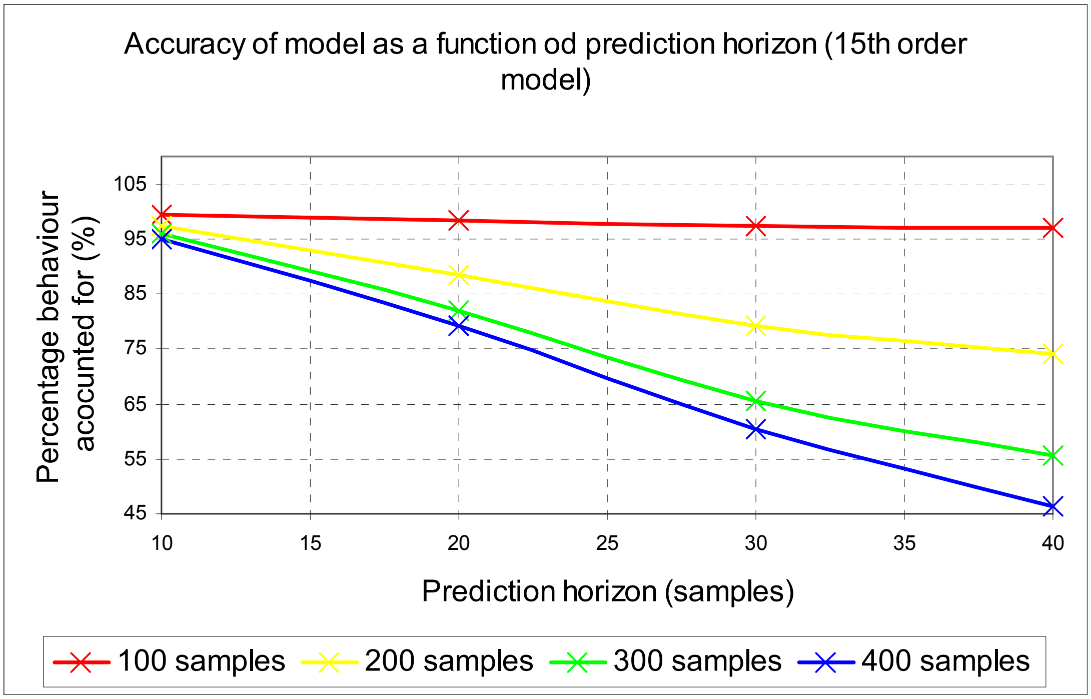


Figure 41 Accuracy of model as a function of prediction horizon

Models of various orders (based on signal segments of 100, 200, 300 and 400 samples long respectively) were evaluated using[24] $10 \leq$ p.h. ≤ 40 at increment of 10. For illustration purposes, only the results for a 15th order model is shown.

The behaviour shown in Figure 41 is consistent with and verifies what was observed in Figure 35. It is clear that as the prediction horizon are increased, the accuracy of the predictor decreased as the effect of e[n] on the predictor output became more pronounced. In spite of this, the close proximity of the 100-sample line to the 100% limit suggests that a 15th order model is able to capture the dynamics in a 100-sample segment sufficiently thoroughly so that the influence of e[n] becomes very limited indeed.

Accordingly, as a given order of model is required to capture more and more non-linear dynamics as the segment length increases, the LTI model description becomes increasingly inadequate, which means that the relative magnitude of e[n] also increases. This accounts for the observation that the graphs of longer segments dive away faster from 100% than those of shorter segments, since for a given prediction horizon, the relative influence of e[n] on $\hat{y}[n]$ is greater.
If the fit value of a prediction for a given signature were to be used as a DSV verification parameter, then a balance must be struck with the value of p.h., so that it is sufficiently ‘big’ to allow a model’s accuracy in the mid-frequency range to be observed, and yet sufficiently small so that a prediction based on a genuine signature would not be degraded to such an extent as to be classified as a forgery.

A final comment in this section is that given the simple model structure of an ARX model, it was found that, amazingly, it provided sufficient accuracy for almost all signatures encountered. This was a welcoming finding, since given the low memory requirement of an ARX model, it was realised that there was no need to explore the more elaborate versions of the transfer function family of model structures. Consequently, all subsequent design and development efforts took place within the framework of ARX model structures, and it will be seen that an ARX is in fact adequate for its role in a DSV algorithm.

6.5. Implementation Details

This section serves as the precursor to the actual investigation of the true feasibility of a systems approach to DSV. Before discussing the experimental results of Section 6.6 however, it is instructive to gain an understanding of the experimental set up used in the investigation, especially in terms of creation of templates and the verification process of signatures. Without knowledge of the topics covered in this section, the results presented in the next section would be of little value. Note that the MATLAB code to the template building and verification functions is given in Appendix 4.

6.5.1 Creation of Templates

The creation of templates consists of the following major steps, each of which is discussed in turn hereafter:

1. Retrieval of genuine signature data from four signatures

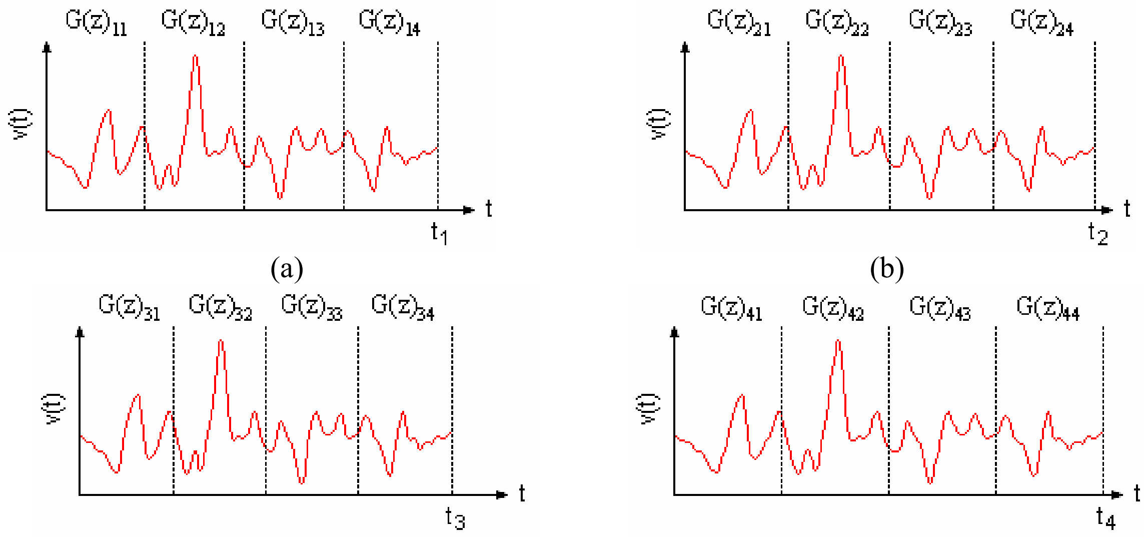
[24] It was known that accuracies corresponding to p.h. = 1 are inevitably in the range 98% - 100%, thus the lowest value of p.h. chosen was 10.

2. Normalisation of signal length to ensure that plants can be derived consistently for each signature given a chosen segmentation length
3. Segmentation of each signal pair
4. Perform systems identification
5. Save all derived systems model
6. Choose the model for each corresponding segment that gives the highest overall match for all genuine signatures.

Four specimens of a genuine signature were used to generate a template for each genuine signature. The number of specimen signatures was deliberately kept low since the level of user convenience would decrease with increasing number of genuine signatures required to derive a template.

Once the four pairs of $v_x[n]$ and $v_y[n]$ signal are retrieved, the system identification phase may commence. Graphs (a) to (d) of Figure 42 represent the pen-tip velocity profile corresponding to the 4 reference signatures. In acknowledging inconsistencies in the time duration of signatures, observe that the velocity profiles of each reference signature have different durations along the time axis i.e. t_1, t_2, t_3 and t_4. Also note that for the simplicity of illustration, the extended and compressed versions of the same velocity profile has been used to represent 4 separate reference signatures.

In dealing with differences in signature time from one genuine signature to the next, two methods were contemplated for template building so that that the maximum amount of information generic or common to a given set of genuine signatures can be extracted.



(a) (b)

(c) (d)

Figure 42 Template building with no signal length normalisation i.e. variable segment length

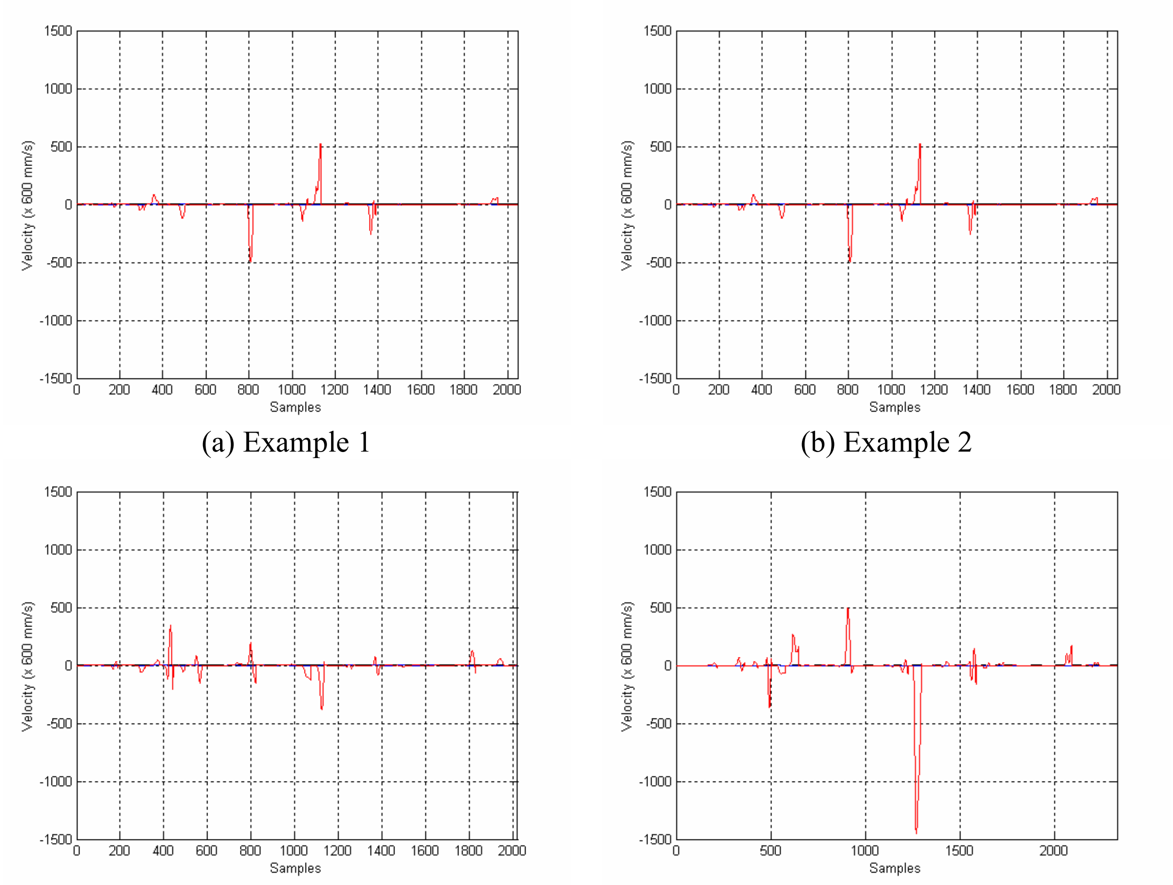
The one method contemplated is as shown in Figure 42, where the individual specimens of *varying* lengths are segmented into a predetermined number of segments, after which system identification is performed on these segments of *unequal* length. In order to find the best model for a given segment, each of the four models derived for the segment is applied to the corresponding segments in the remaining three signatures e.g. $G(z)_{11}$, $G(z)_{21}$, $G(z)_{31}$ and $G(z)_{41}$

for segment 1, and the one that yields the highest fit measure is chosen for the template. In this way, the information content of the template is optimised.

The second method was to normalise the three longer signals to have the *same* length as the shortest signal pair, so that for a given segmentation length an equal number of models can be derived for all four signature specimens. The optimisation process used in this case was identical to the one described for the prior method.

To choose between the above two methods, a test was performed to assess the overall accuracy of models generated via the two methods. In particular, 8 specimens were taken of each of the various signatures collected, of which 4 were then used to create individual templates. The models within the templates were in turn applied to all 8 specimens of each signature for assessing model accuracy. The segmentation and optimisation methods, the latter of which has been briefly described above, are described in detail in subsequent paragraphs of this section.

Through experimentation, it was found that when templates generated using the first method were applied to the 8 specimens of each genuine signature, instability such as shown in Figure 43 was often observed. In particular, each of the graphs (a) to (d) of Figure 43 shows the typical instability behaviour seen in the predicted velocity profile $\hat{v}[n]$ yielded by templates based on a variable segment length of method one.



(a) Example 1

(b) Example 2

(c) Example 3

(d) Example 4

Figure 43 Instability observed with template based on variable segment length

A explanation for this could be that a sufficiently high level of intra-personal inconsistencies was captured by the model during template building, so that when it is applied to another signature specimen, the inconsistency content of the model in conjunction with the e[n] of the new specimen are sufficient to cause significant instabilities – note the vertical scale of the graphs of Figure 43.

This is understandable considering the many degrees of freedom allowed in method one viz. inconsistencies in terms of differences in segment length, local inconsistencies, as well as local time warp distortions.

Method two on the other hand, being derived from signatures of fixed length, would only have to cater for local inconsistencies and local time warp distortions. The reduction of one degree of freedom was seen to have significant positive impact on the accuracy of models. Consequently, it was decided to adopt method two for the creation of templates, where the length of the $v_x[n]$ and $v_y[n]$ signal pairs are normalised to that of the shortest one of the four genuine signature specimens.

The third step is the segmentation of signals according to the following criterion for system identification: for the $v_x[n]$ and $v_y[n]$ signal pair of a given signature, the signature is divided into segments of length h, each of which then forms the data set for a system that is to be identified. However, since it is acknowledged that local signal warping in time is inevitable in handwritten signatures, the actual data set used is as shown in Figure 44.

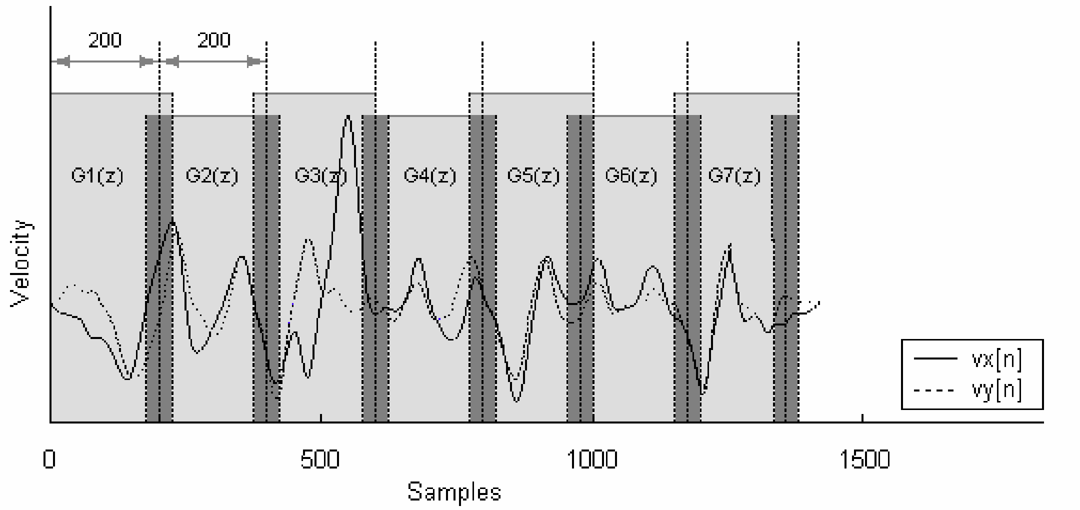


Figure 44 Segmentation scheme for systems identification

Firstly, a principle segmentation length is set to 200 samples for example. Realising that it is unlikely that such a segment length would be exactly divisible by the total signal length, the modulus or residue of the division

$$\text{Total length} \div \text{Principle length} = \text{Integer} + \text{Residue}$$

is then calculated. The residue would then be absorbed equally into each segment by increasing the segment length by the quotient of Residue ÷ Integer in the above equation. Consequently, the actual segmentation length used would be slightly longer than 200, which is necessary to account for the short segment of signal that protrudes from the shaded area in Figure 44. For simplicity sake however, we shall ignore this fact, and assume that the segmentation is exactly divisible by 200.

As already mentioned, due to the local signal warping in time, an overlap is introduced between each consecutive data segment. Specifically, the length of darker shaded region representing the overlap is 25% of the segment length i.e. 50 samples. As a result, the final segment length used is:

25 (extension into previous seg.) + 200 + 25 (extension into next seg.) = 250 samples

This is true for all except for the first and final segments, which is exactly 225 samples long, since they only overlap with one other segment. In the MATLAB code, the overlap is specified by the half overlap ratio e.g. for 25% total overlap, the overlap contribution due to one segment i.e.12.5%, is quoted.

The eventual length of the signals constituting the data set used for systems identification is therefore 225 samples long. If we incorporate the effect of the residue in signal length, the true length would be slightly longer than 225 samples.

During experimentation, the specific segment length and overlap used could be adjusted at will.

The task of system identification for an ARX model is as described in Section 6.3, and needs no further elaboration here. In MATLAB, the task is accomplished using the `arx` function, which returns an `idpoly` object for SISO systems. For four genuine signatures denoted S1, S2, S3 and S4, a cell array template is constructed as follows:

$$\begin{bmatrix} G(z)_{1,S1} & G(z)_{2,S1} & \cdots & G(z)_{n,S1} \\ G(z)_{1,S2} & G(z)_{2,S2} & \cdots & G(z)_{n,S2} \\ G(z)_{1,S3} & G(z)_{2,S3} & \cdots & G(z)_{n,S3} \\ G(z)_{1,S4} & G(z)_{2,S4} & \cdots & G(z)_{n,S4} \end{bmatrix}$$

where n is the number of segments found in a signature – this is step 5 in the creation of a template.

Clearly, there is considerable redundancy in the above cell array, since only one model for each segment is needed for the final template, step 6 therefore chooses an optimal model from amongst the four models in each column of the cell array for each segment. The specific criterion used was to emulate the verification condition using a k-step ahead predictor, and choose the model that yielded the best fit for a particular segment for all four genuine signatures i.e.

$$G(z)_r = \mathrm{Min}\left\{\mathrm{norm}(y - \hat{y}) \,\middle|\, \hat{y} = \mathbf{G}(G(z)_r, u, y, k)\right\} \qquad \text{for } r = 1, 2, \ldots, n$$

where r is the index of the r^{th} segment, norm is the Frobenius norm as defined in Section 6.3, **G** is the k-step ahead predictor function based on $G(z)_r$, $\hat{y}$ the predicted output, u and y the input data set and k the prediction horizon. Since the model that yields the minimum norm of $y - \hat{y}$ must be the one that has the highest common information content amongst the four models for each segment, step 6 is therefore a method of maximising the common information content of the template as a whole.

To appreciate information optimisation described above from a theoretical perspective, the reader should appreciate that a generic comment about the common characteristics of a series of observations can be only as generic as what the observer knows about the least informative observation. On the same token, since a signature template is a generic comment about common characteristics of a number of genuine signature specimens, the template should contain only generic or common information, and as little intra-personal variation (specific information) as possible.

At the end of step 6, the template exists as a 1 by n + 1 cell array of `idpoly` objects, n + 1 being due to the placement of a row vector containing implementational parameters as the first entry of the cell array. In practical terms, the template for a typical 3-second signature would require about 4 kByte of memory using a segmentation length of 60 samples and 20th order models. This is without compression and using floating-point numbers to store the coefficients, which is 4 bytes per number. At a sampling frequency of 600 Hz, this corresponds to an inherent compression ratio of around 56%, which is quite advantageous. Using fixed-point number storage of 2 bytes per number for example, even higher compression ratios can be realised.

Another topic of particular relevance here is concerned with the evolution of templates. Due to the fact that signatures tend to evolve over time, a template created now would likely to yield high number of false rejections in 10 years time. In a systems-based framework, this means that the coefficients of models need to be updated from time to time. One possible implementation of this may be to use the template to record the history of scores obtained for individual segments, so that when the score corresponding to a particular segment is noted to be consistently low, a new model can be derived for that segment.

Overall, the adjustable parameters in template creation includes:

- Principle segmentation length
- Amount of inter-segment overlap
- Order of ARX model
- Prediction horizon to use in step 6.

6.5.2. Verification

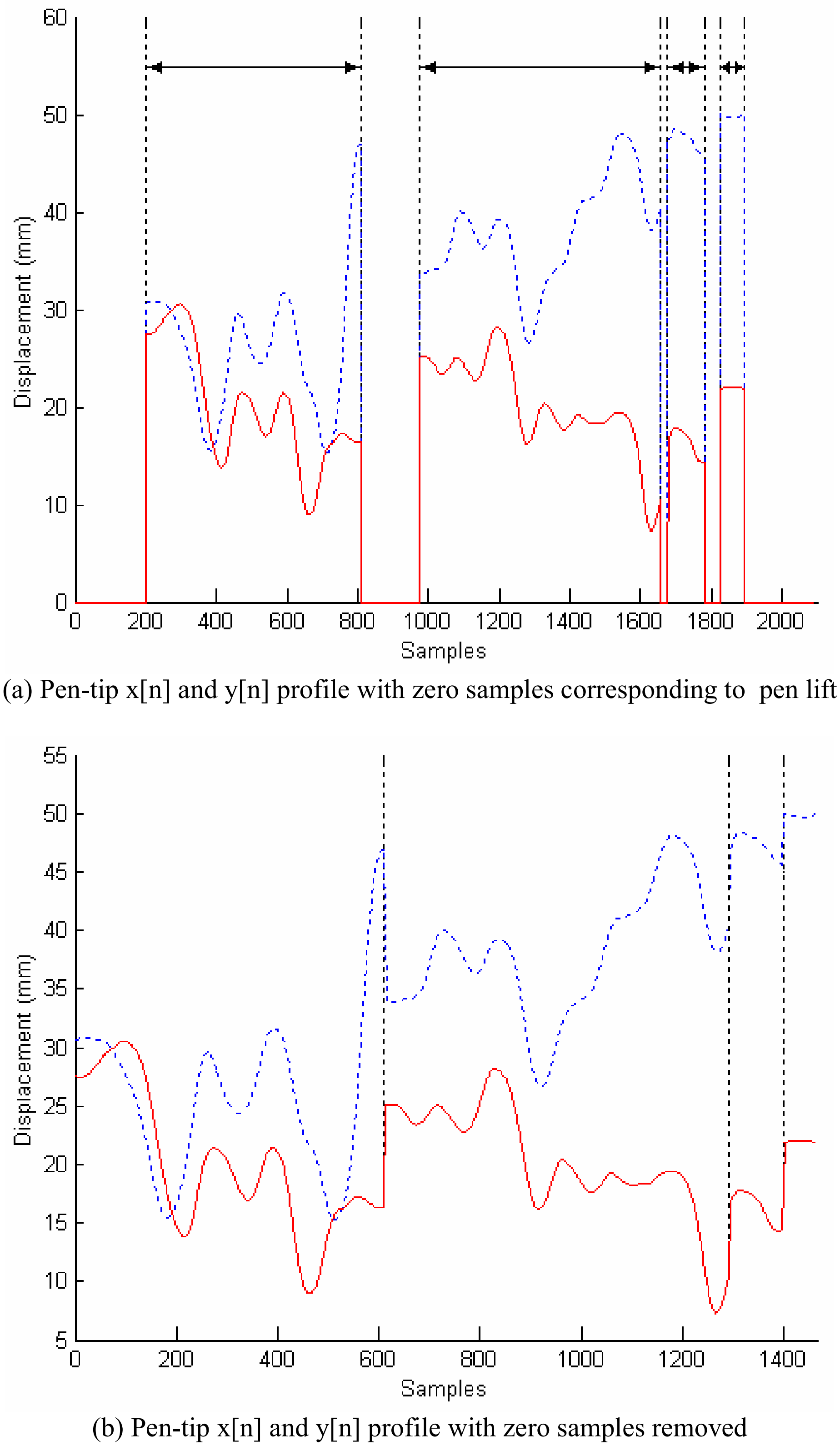
Dynamic signature verification is the task of determining whether a given handwritten signature was produced by the authentic owner by observation of the signature's dynamic characteristics. In the current context, verification implies decision making based on the similarity measure between the actual $v_y[n]$ and the predicted output $\hat{v}_y[n]$.

The input to the verification function includes the template, the signature to be verified, as well as any adjustable parameters that can influence the outcome of the verification process. Since the k-step predictor has been seen as an effective tool in assessing the correlation between signal dynamics and model description, while simulation results were adversely affected by e[n], the predictor was chosen as the basis of the verification procedure.

The major steps within the verification process are:

1. Retrieval of the test signature and the template against which it is to be compared
2. Determining appropriate segment length for the given signature
3. Piecewise construction of the predicted output $\hat{y}$

4. Calculate the similarity distance based on the Frobenius norm
5. Calculate other parameters of interest.



(a) Pen-tip x[n] and y[n] profile with zero samples corresponding to pen lift

(b) Pen-tip x[n] and y[n] profile with zero samples removed

Figure 45 Signal formatting

The test signature is submitted in the form of a column vector whose columns contain the x[n] and y[n] vectors of the signature captured at 600 Hz. The x[n] and y[n] are also 'zero-removed' as illustrated in Figure 45.

Figure 45 (a) shows the pre-formatted signal containing pen down strokes over which the arrows extend, as well as zero valued samples corresponding to pen lifts during a signature. (b) however shows the zero-removed version available. Due to the loss of essential timing information in (b), a complementary vector was also made available whose entries are the indices of discontinuities found in (a), so that (a) can be reconstructed from (b). Consequently, the full test signature is described by the combination of (b) and the discontinuity indices vector.

The x[n] and y[n] data series are then processed as described at the beginning of Section 6.3 to obtain the ARX regressors $v_x[n]$ and $v_y[n]$. The segmentation process that follows begins with evaluating the appropriate segmentation length, which involves the simple division of

$$\text{Total signal length} \div \text{Number of ARX models found in the template}$$

By recalling that the overlap facility has already been incorporated during the systems identification phase, the range of sample indices over which each model is applicable would thus always be greater than the evaluated segment length, as a result the effect of terminal inaccuracies is inherently moderated.

For each signal segment, the corresponding ARX model is retrieved from the template and the k-step ahead prediction performed using the MATLAB's `predict` function, which returns the predicted velocity $\hat{v}_y[n]$ given the actual observed velocities $v_x[n]$ and $v_y[n]$ for a particular segment. This is carried out for each segment, the concatenation of which produces the overall predicted output.

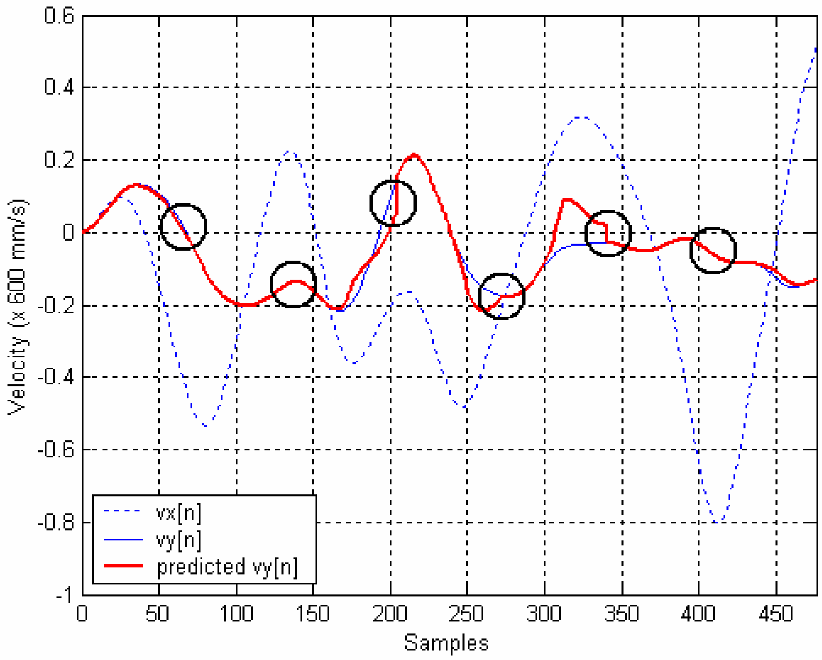


Figure 46 Verification example

For the example signature shown in Figure 46, the overall predicted output $\hat{v}_y[n]$ based on seven 20th order ARX models using half overlap ratio of 20% is shown. Clearly the actual and predicted y velocity profiles has a high degree of cross-correlation, this is expected since the test signature used in this case was a genuine signature of the signer. In fact, the fit value achieved was 70.537% – a good result for a test signature.

The bold line representing $\hat{v}_y[n]$ is a concatenation of 7 separate segments predicted using 7 different ARX models at a prediction horizon of 20 samples. The boundaries between each segment in $\hat{v}_y[n]$ are indicated by the circles, and clearly visible are the discontinuities that exists between segments 4 and 5 and segments 5 and 6. These features arise due to the fact that the gain and phase dynamics across the discontinuities are entirely different, since after all, different models describe the adjacent segments, and the models were not derived from the given test signature specimen.

In an attempt to make $\hat{v}_y[n]$ a closer approximation to $v_y[n]$ in view of the discontinuities, an investigation into two smoothing schemes was conducted.

The first scheme proposed was signal fading by linear combination of cosine-function-weighted samples. Graphically, the curve shows that at a segment boundary at index k, the effects of the former segment fade out according to the cosine function while the effects of the latter segment fade in proportionally. In this way, the behaviour of successive segments at the end points is merged at the boundaries, thereby reducing the appearance of discontinuities.

The prerequisite for this technique however is that each segment must extend into its neighbour (by 10% of their lengths say), so that in the region k – h×10% to k in the figure above the current segment fades out from 100% to 50% weighting, and in the region k to k + h×10% the next segment fades in from 50% to 100% weighting. Figure 48 shows the faded version of the signal of Figure 46, it can be seen that the discontinuities are moderated while the remaining parts are unaltered.

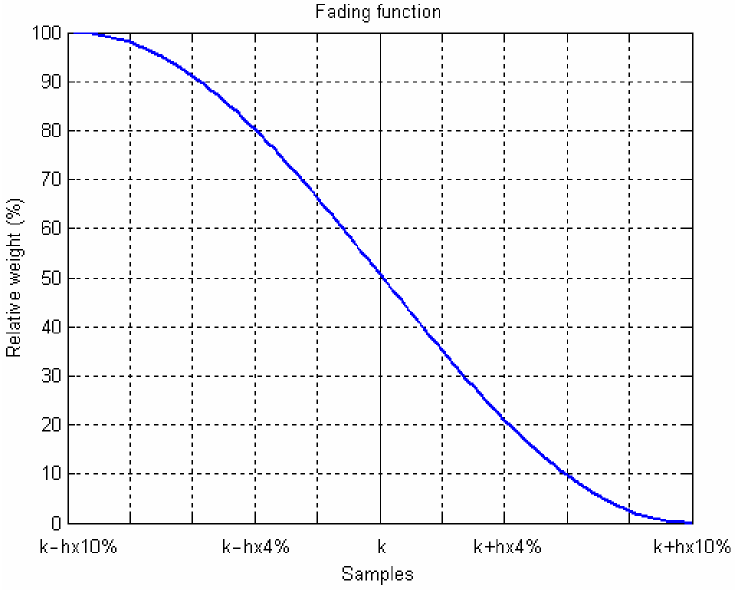


Figure 47 Cosine fading function

Surprisingly, the removal of discontinuities by fading in fact degraded the accuracy of the predicted signal, from 70.537% to 61.876%. To examine whether such degradation was a general result of the fading process, the technique was applied to various sets of genuine signatures, the results of which are shown below. Clearly, the unfaded signals yielded consistently higher fit values.

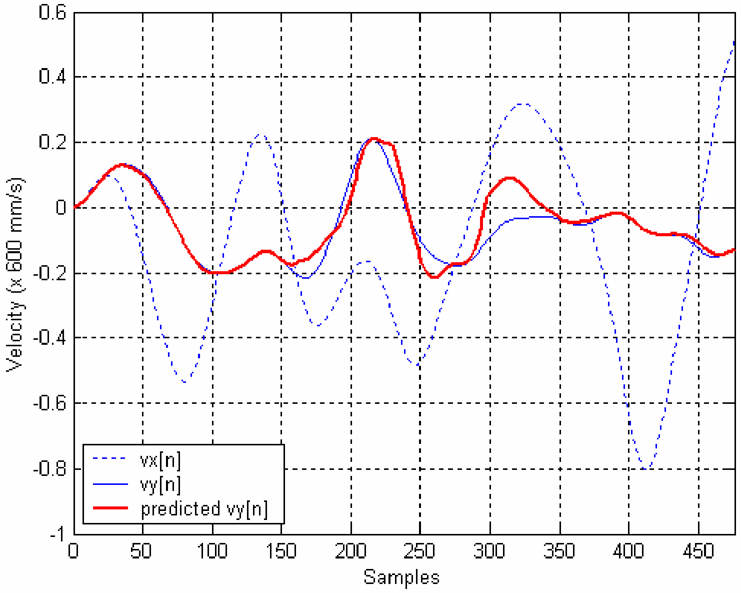


Figure 48 Verification example with fading

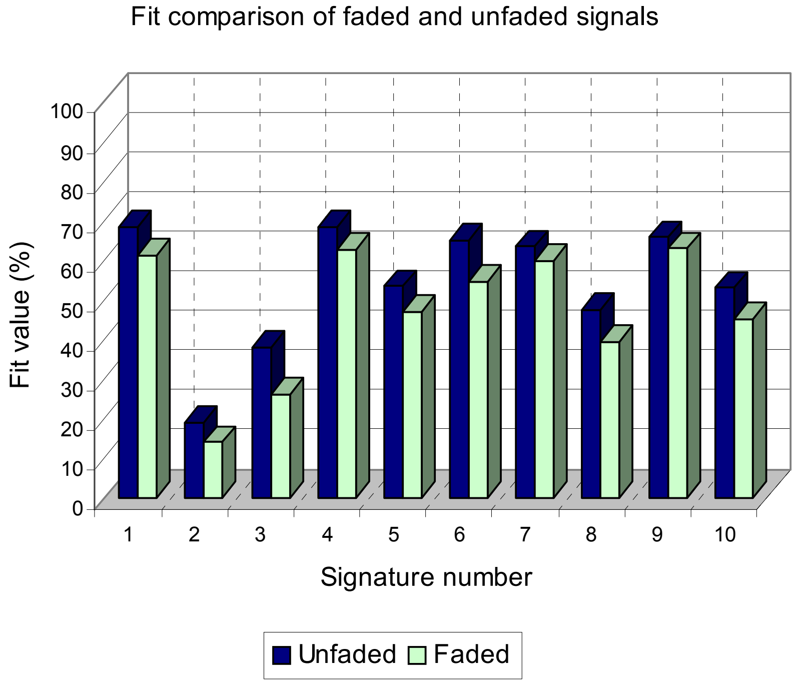


Figure 49 Fit comparison of faded and unfaded signals

By scrutinising the faded and unfaded signals, the mechanism of signal degradation was also realised. By noting that the signal accuracy is often unequal across a point of discontinuity e.g. at around samples 200, 270 and 390 in Figure 46, the fading process tended to propagate the error in the more inaccurate segments into the more accurate segments. Although accuracy is indeed improved in the originally less accurate side of the discontinuity, it turns out that the this improvement is almost always outweighed by a corresponding degradation on the originally more accurate side, thereby reducing the overall fit between $\hat{v}_y[n]$ and $v_y[n]$.

Consequently, a second method was considered involving the use of a zero phase distortion discrete low pass filter implemented by the `filtfilt` function of MATLAB. A 20th order FIR LPF was designed using the `fir1` function, where the cut off frequency was set to 5 Hz by trial and error. At the cut-off frequency, the gain is guaranteed to be below –6 dB. When the entire $\hat{v}_y[n]$ signal shown in Figure 46 was fed through the zero phase distortion filter the signal profile below was obtained.

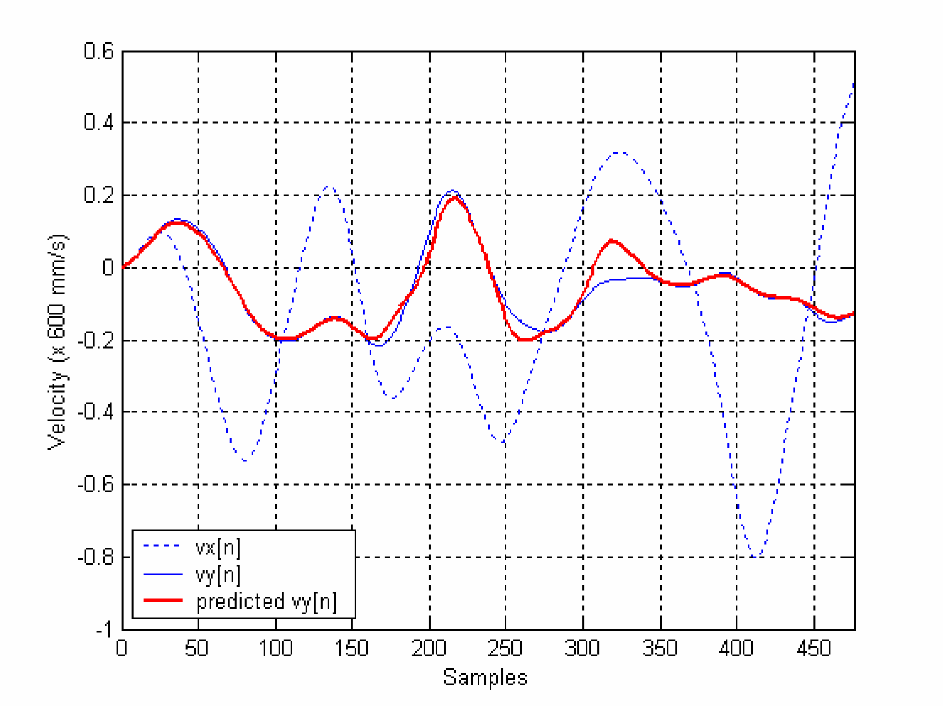


Figure 50 Verification example with FIR filtering

It is evident that all points of discontinuity have been smoothed compared to the profile of Figure 46, especially at around samples 260, 340 and 450. But this time the corresponding fit values actually improved – the fit value of the filtered version of Figure 46 improved from the previous value of 70.537% to 73.808%. As before, the bar graphs below serve to show that the observed result is a generally applicable one when applied to various sets of other genuine signatures.

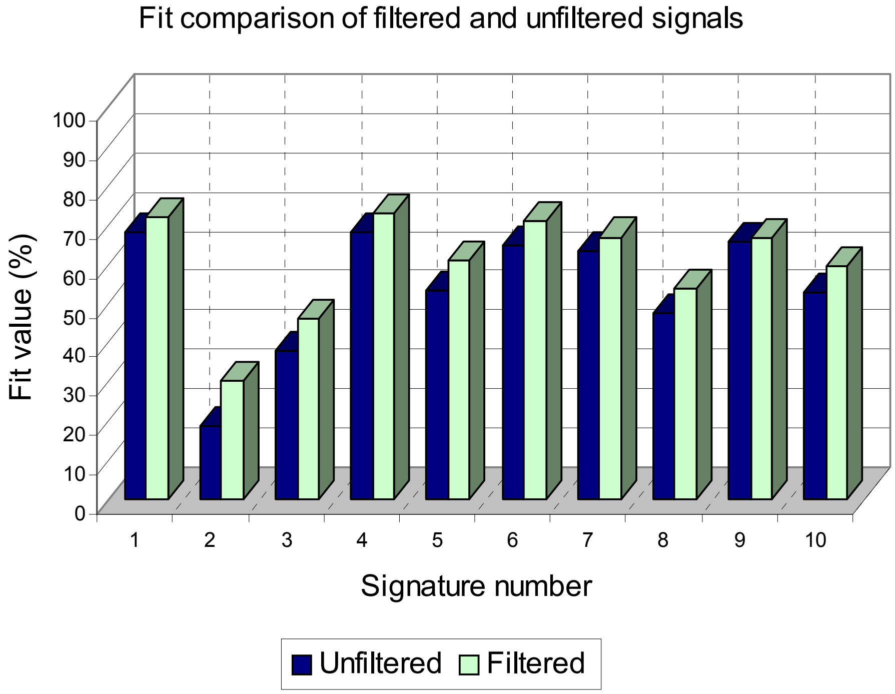


Figure 51 Fit comparison of filtered and unfiltered signals

In view of the positive effect filtering has on the fit value of genuine signatures, the zero-phase-distortion filtering was adopted as standard practice, so that the overall filtered $\hat{v}_y[n]$ along with the actual observed output formed the basis of the similarity calculation, which is step 4 of verification described by Equation 19 in Section 6.3.

In interpreting the left hand side of Equation 19, it is understood that if the correlation relationship between $v_x[n]$ and $v_y[n]$ as captured by the ARX models is obeyed in the test signature, then $\hat{v}_y[n]$ and $v_y[n]$ are expected to be similar. Consequently the greater the value of the similarity measure the more confidence one has in saying that the test signature is a genuine one. However, if the timing and correlation relationship is violated, the conflicting information in the dynamics of the test signals and the ARX models would yield a poor prediction, thereby giving a small value for the fit due to the larger magnitude of norm($y - \hat{y}$).

The above forms the crux of the verification process, however prior to finalisation of all implementation details, various other associated parameters were also calculated as a part of verification in search of a useful discriminant distance metric. Such variables included the sum of the absolute value of all errors, the number of turning points $\hat{v}_y[n]$, RMS, mean, maximum and minimum values of $v_x[n]$ and $\hat{v}_y[n]$ etc.

6.6. Effectiveness of Plant Models as a DSV Tool

The previous section provided the necessary understanding of the template creation and verification processes adopted during research; this section however focuses on the result of the experiments, which is an direct assessment of the true feasibility of a systems approach to DSV. Specifically the discussions are in terms of how experimental results correlate with prior expectations, whether a system approach provides sufficient discriminancy between genuine and forged signatures, if there are weaknesses inherent of the system, and finally design considerations pertinent to the overall system, which includes the use of additional parameters apart from the plant models.

From previous discussions it was noted that model order, segment length and prediction horizon were algorithm parameters that affected the accuracy of a model or predictor. Since model accuracy and discriminancy are closely related ideas in the given context, their effects on discriminancy from a practical implementation perspective was also examined using real genuine and forged signatures.

Notably, the same set of signatures that were used for the investigation of the previous subsections were again used, and to emphasise they were collected from people of different sex, age, handedness and cultural background, which was necessary to broaden the variety of signatures encountered in order to more closely approximate the universal set of all signatures encountered in practice. Furthermore, sets of forged signatures were also collected from a number of people who volunteered to conscientiously forge given signatures specimens. The forgeries were used to assess the discriminancy of a given signature model.

6.6.1. The Effect of Model Order on Discriminancy

The aim of the experiments carried out in this regard was to verify if increased model orders would contribute positively and notably towards model discriminancy. As already known from the results of Section 6.4.1, improvement in model accuracy is notable only up to a certain point, beyond which the gain in accuracy becomes only marginal with further increases of order. Since model accuracy is closely related to model discriminancy, the same behaviour is once more expected. The reader should also realise that for a signal segment of n samples, it would be unwise to use a model order in the region of n/2, since the memory requirement analysis of Section 6.2 indicates that such a model would require the storage of n model parameters i.e. no compression of information would then be achieved.

The parameter and verification settings used for the experiments are given below:

Table 3 Experimental parameter settings: model order – discriminancy

Segment length	:	50 samples to 75 samples 100 samples
Model order	:	4 to 20 in increments of 4 5 to 30 in increments of 5 5 to 40 in increments of 5
P.h. used for template optimisation	:	20 samples
Half overlap ratio	:	20%
P.h. used for verification	:	20 samples

In particular, 10 sets of 8 genuine signature specimens and 10 sets of 8 corresponding forged signature specimens were obtained. For each set of genuine signatures, a template is created based on the parameter values set out above using the first 4 signatures of a given set, so that no special preference is given to ‘better looking’ signatures during template building – this is to emulate conditions of practical implementation.

The templates generated are then applied to both the sets of genuine signatures and corresponding sets of forgeries so that their respective *mean fit values* can be calculated. As explained in Section 6.5.2, a higher value of the fit indicates a greater confidence of classifying a signature as a genuine, accordingly a forgery should yield low values of fit if the information captured in the template is discriminant. The greater the difference between the fit values achieved for the genuine and forged signatures, the more discriminant the information contained in the template.

In light of the above, the aim of this section is to investigate the effect of model orders on the level of discriminancy of the information contained in the template. For graphical illustration of experimental results, the definition of discriminancy used was simply

$$\text{Discriminancy} = \text{mean fit}_{\text{genuine}} - \text{mean fit}_{\text{forgery}}$$

Equation 21 Definition of discriminancy

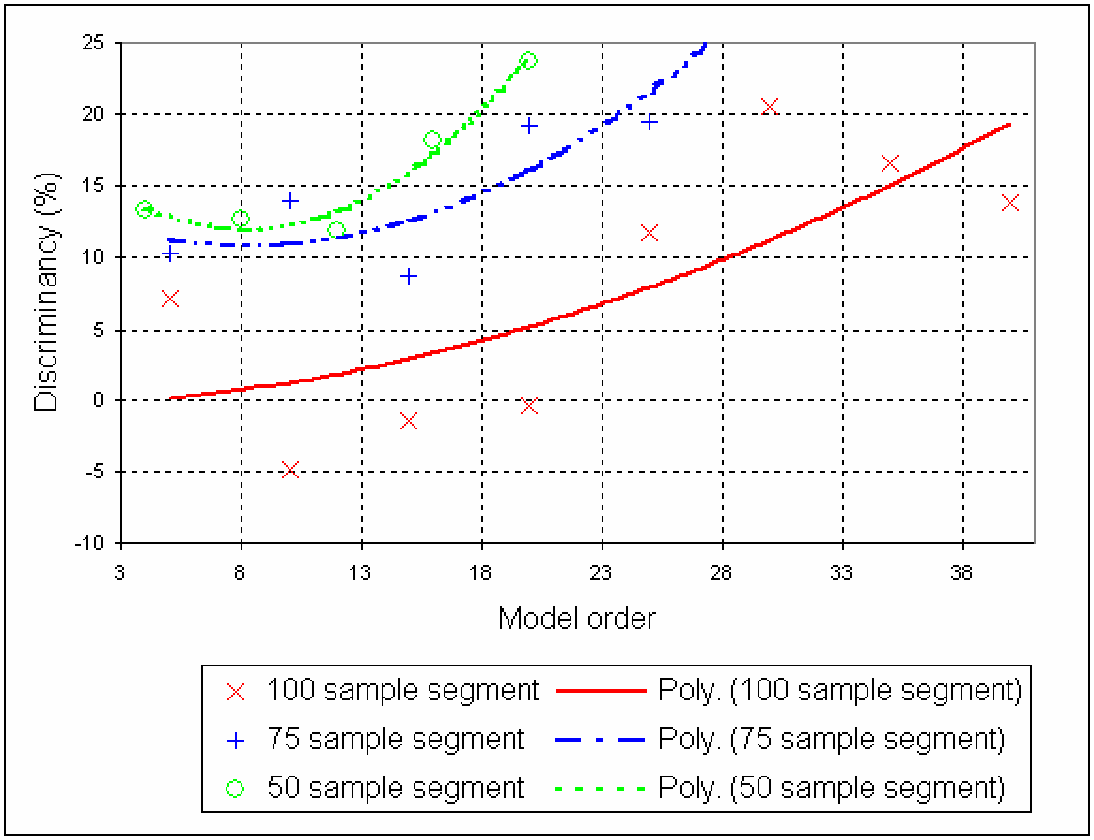


Figure 52 Model discriminancy as a function of model order

A number of important observations can be made from the results of Figure 52. Firstly and most visibly, the discriminancy of models based on all segment lengths increased with higher model orders. This is clear evidence that a higher model order does indeed increase model discriminancy as expected, with the upper discriminancy level being limited by the maximum practical model order i.e. half the segment length. The observed behaviour is understandable by considering that as the accuracy of model descriptions increases with model order, the e[n] contribution in the prediction outcome drops to very low levels for genuine signatures. However when a forgery is submitted for verification, the high levels of dynamics unaccounted for by the model amplifies the effect of e[n], which in turn degrades the quality of the predicted output. This is also the reason why the trends do not adhere to the expectation that they should taper out beyond a certain critical value, since the fit values for genuine and forged signatures diverge for higher model orders.

Secondly, the discriminant ability of the models increased most rapidly for models based on 50 sample segments and least rapidly for models based on 100 sample segments. This is an indication that given the same increase in model order and thus memory usage, a larger improvement in discriminancy can be reaped if shorter segments are used. This behaviour is a result of the shorter segments having less information content. Thus for a given increase in the model order, the relative increase in the amount of dynamics captured by the model is higher for shorter segments.

Thirdly, it is also clear that models based on shorter segments have higher discriminancy in general, which is evident from the initial offsets of each graph. This is no surprise and in fact reaffirms the findings of Sections 6.4.1 and 6.4.3, which indicates that higher order models for shorter segment lengths would in general result in higher model accuracy.

The final observation of Figure 52 is undoubtedly that the experimental data is more erratic than those found in Section 6.4, the origin of which can be traced back to the lack of a standard in the quality of forgeries. In other words, even if it is given that model accuracy increases consistently with model order for genuine signatures, the information content of forgeries are on the contrary not well correlated to the model. Consequently, the resultant fit values of forgeries are more erratic than the corresponding values yielded by genuine signatures i.e. the discriminancy as defined in Equation 21 would also be more erratic.

Nevertheless, the valuable finding of this section instrumental to making design decisions remains that of having established the positive effect of high model orders on discriminancy, as well as the confirmation of a correlation between model accuracy and discriminancy.

6.6.2. The Effect of Segment Length on Discriminancy

In this section, the relationship between the length of segments and the discriminancy of models is investigated. It was anticipated that longer segments would lead to lesser discriminancy in view of the results presented in Section 6.4.3. Using the same set of genuine and forged signatures as well as the definition of discriminancy as used for the previous section, the experimental parameters used were as follows:

Segment length	:	50 samples to 200 samples in increments of 10 samples
Model order	:	20
P.h. used for template optimisation	:	20 samples
Half overlap ratio	:	20%
P.h. used for verification	:	20 samples

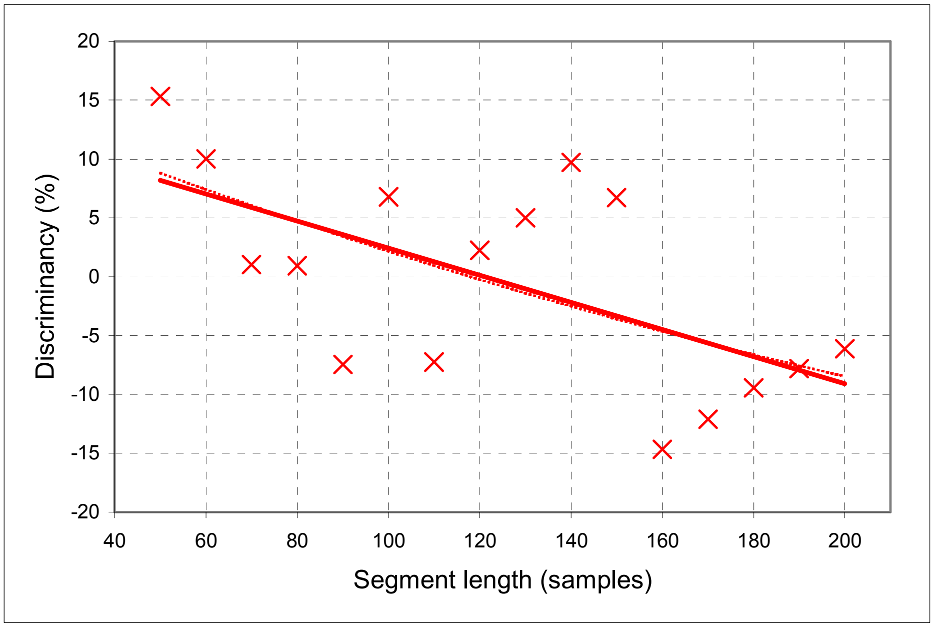
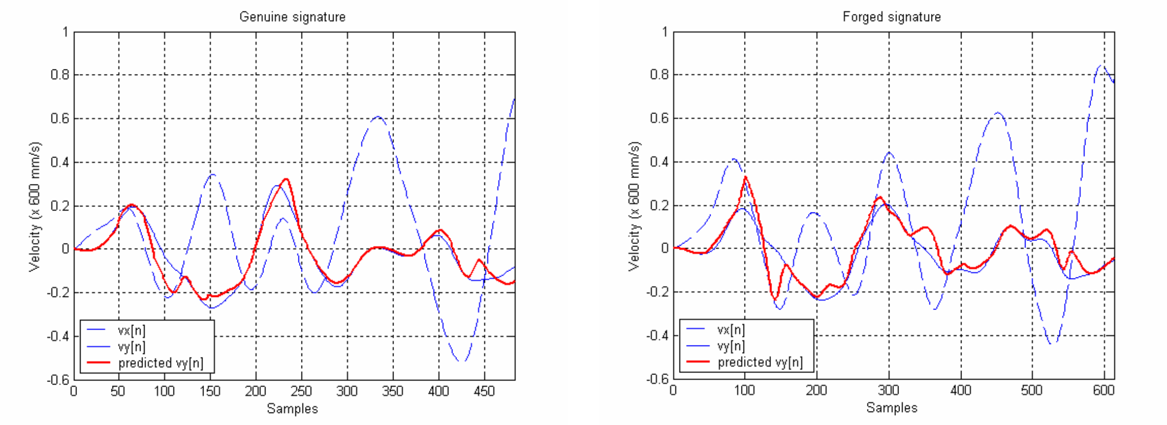


Figure 53 Model discriminancy as a function segment length

As expected the model discriminancy is a steadily decreasing function of segment length, what is striking though is that discriminancy seems to decrease approximately linearly with segment length. In particular, the solid trend line is a first order least square polynomial fit, while the dotted trend line is a second order least square polynomial fit – clearly both had resulted in almost identical trends.

Two reasons are cited here to account for the reduction of discriminancy with increasing segment length, the first of which has already been mentioned at the beginning of this section. The second and more subtle reason is that when models are developed for shorter segments, the information content of the models pertain specifically to the *local behaviour* of $v_x[n]$ and $v_y[n]$, which has been noted to be particularly difficult to imitate.



(a) Genuine signature (b) Forgery

Figure 54 Discriminancy within local behaviour of dynamic profile

In the signal pairs shown in Figure 54, it is clear that at a macroscopic scale, the dynamic profiles exhibit a gross correlation, with equal number of velocity peaks and troughs and even similar covariation relationship between $v_x[n]$ and $v_y[n]$. However, when one scrutinises the same set of characteristics at a microscopic level, focusing for example on the behaviour around one specific peak, then the dissimilarity between the genuine and the forgery becomes quite apparent.

In the example shown by Figure 54, eight 20th order ARX models were derived based on segment 50 samples long i.e. 83.3 ms segments. Due to the concentration of each model's information content on the local behaviour, the macroscopic similarity falls out of scope and the predicted output based on the template is thus able to emphatically distinguish the forged signature on the right (fit = -12.7124%) from the genuine on the left. Note that the bold solid line of figure (b) above is characterised by an overall lower quality fit and a higher number of turning points.

In view of the above, it is thus desirable to adopt short segments for systems identification in general. However, for some signatures classified as 'sinusoidal' in Section 6.6.5, a longer segment length may in fact be more favourable due to the lack of discriminant information on the local scale. Refer to Section 6.6.5 for more detailed explanations on this class of signatures.

6.6.3. The Effect of Overlap Ratio on Discriminancy

The experiments of this section were conducted to examine whether the overlap facility catering for local time warp distortions would improve the overall discriminancy of a template. The parameter settings used and the corresponding experimental results are shown in Table 4 and Figure 55 respectively.

Table 4 Experimental parameter settings: half overlap ratio – discriminancy

Segment length	:	60 samples
Model order	:	20
P.h. used for template optimisation	:	20 samples
Half overlap ratio	:	5% to 45% in increments of 5%
P.h. used for verification	:	20 samples

It is clear from Figure 55 that model discriminancy is a decreasing function of half overlap ratio, which suggests that over catering for local time warp distortions may in fact be counter productive in the verification process. This makes sense since an increase in the amount of overlap implies an increased segment length, which according to the findings of prior sections must result in reduced model accuracy and thus discriminancy.

Since the underlying mechanism for degradation of model discriminancy in the results of Figure 55 and Figure 53 are the same, one may ask why then are the trend lines in the two figures different. The reason is that in Figure 53 the segment length increases monotonically along the x-axis, while in Figure 55 the x-axis denotes the *half* overlap ratio i.e. for every unit increase on the x-axis the segment length increases by twice the specified amount so that the segment length does not increase monotonically in Figure 55.

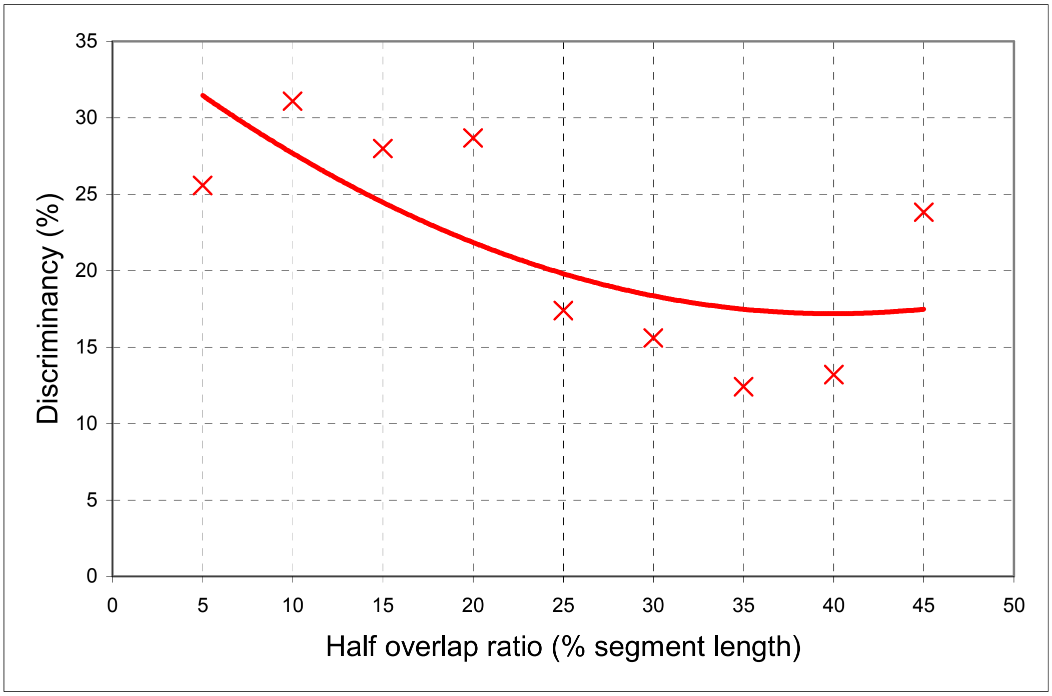


Figure 55 Model discriminancy as a function of half over lap ratio

Nevertheless, the result shown in Figure 55 also makes sense from a practical perspective, as it states that although the overlap facility benefits *only* the authentic signer and not the forger *at small values*, over catering would however help the forger by *overlooking time warping on a large scale*. In other words the reduction in model accuracy as a result of overlap far outweighs the benefit of thereof when the amount of overlap is extended beyond its region of benefit.

In view of the above, the author is of the opinion that had the research have had the means to a large pool of signatures, the 'true' graph of Figure 55 should show initial peaking for small values of half overlap ratio and then decrease steadily thereafter.

6.6.4. The Effect of Prediction Horizon on Discriminancy

The final relationship examined was that between the prediction horizon and model discriminancy. Previously, it was established that as the prediction horizon increases, the accuracy of the predictor based on the model decreases, but the rate at which such degradation occurs remains unknown. To find out, the following set of parameter values were used in experimentation.

Segment length	:	60 samples
Model order	:	20
P.h. used for template optimisation	:	5 samples to 60 samples in increments of 5 samples
Half overlap ratio	:	20%
P.h. used for verification	:	5 samples to 60 samples in increments of 5 samples

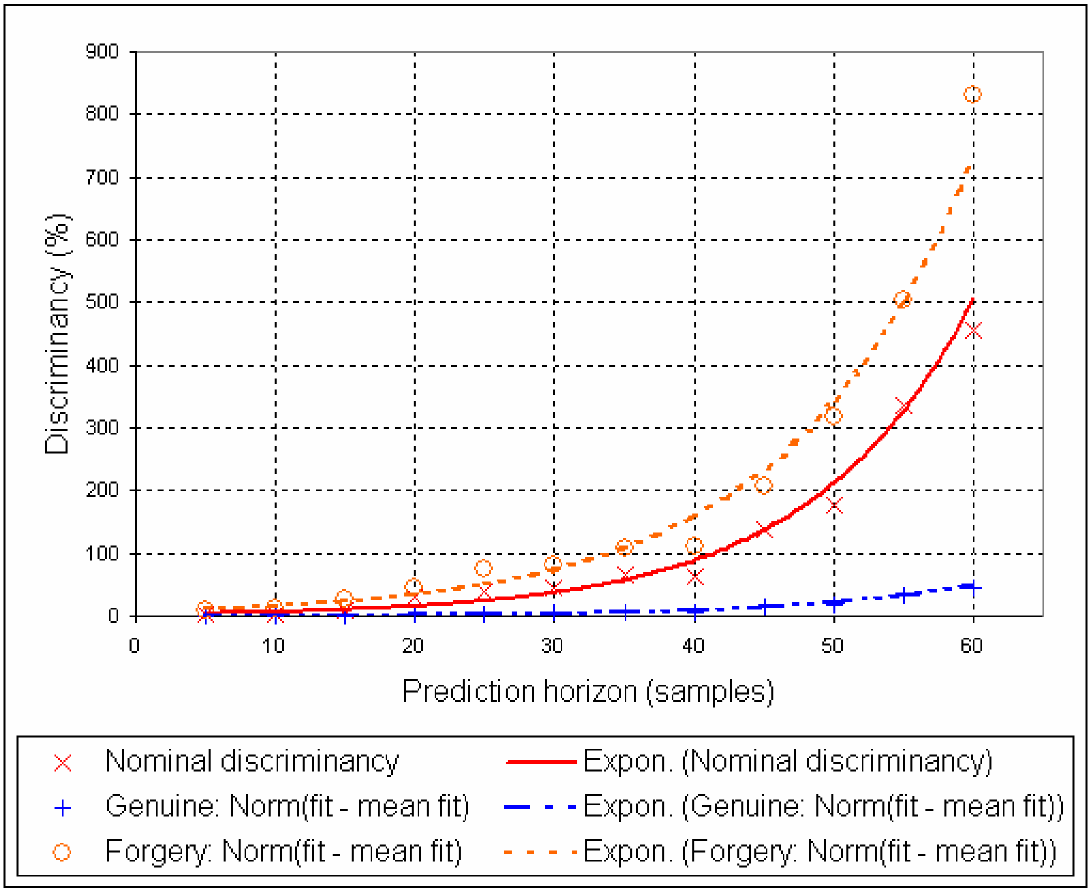


Figure 56 Predictor discriminancy as a function of prediction horizon

The graph in Figure 56 shows that discriminancy is an exponentially increasing function of prediction horizon, which is quite an interesting result considering that predictor accuracy reduces with p.h. What this reveals is that although predictor accuracy reduces with p.h. in general, this happens much more rapidly for forgeries than for genuine signatures.

The above seems to suggest that the greater the prediction horizon the better, but not really. Recall that the discriminancy measure was derived from differences between the *mean* fit values of genuine and forged signatures, thus for a more complete description we also need to look at the dash-dotted (genuine) and dashed (forgery) graphs of Figure 56. The points along each of these lines are defined as the Frobenius norm of the difference between individual fit values and the mean fit value, which gives an indication of the *scatter* of fit values.

These lines tells us that the scatter of fit values about the mean in fact increases simultaneously with the nominal discriminancy – at the same rate. What this means is that although the means of the fit values are further separated by an increase in the prediction horizon, the standard deviation is simultaneously increased at a similar rate so that the net effect is a no-gain situation[25]. More specifically, the dashed graph above indicates that

$$\text{norm}(\text{fit}_{\text{forgery}} - \text{mean}(\text{fit}_{\text{forgery}})\ ) > \text{fit}_{\text{genuine}} - \text{fit}_{\text{forgery}}$$

[25] Recall the discussion surrounding Figure 17 in Section 3.2.

which suggests that the scatter of fit values for forgeries is in fact larger than the separation distance between the genuine and forged signatures. Putting the former plainly, a forged signature may very well fall within the range of fit values of genuine signatures when the prediction horizon is too large, this in turn implies a higher false acceptance error rate. Nevertheless, practical experience has shown that a prediction horizon between 20 and 30 samples tend to provide a sufficient separation and minimal overlapping of fit values between the genuine and forged signatures.

This concludes the investigation of the relationship between the various algorithm parameters and the level of discriminancy. In summary, discriminancy is an increasing function of model order, decreasing function of segment length and overlap ratio, and the influence of changes in the prediction horizon on discriminancy is not a well-defined one due to the simultaneous change in fit value separation and scatter.

Although the above had formed an crucial part in establishing the knowledge base needed to fine tune the various algorithm parameters, a number of secondary findings of interest and importance were also made as a result of the experiments conducted. The following sections focus specifically on such findings, which in themselves form an integral part of the knowledge base.

6.6.5. Three Categories of Signatures

As has already been stated a number of times throughout the prior discussion, the results shown were derived from mean fit values. In individual cases however, it was noted that the accuracy and discriminancy of models were heavily dependent on specific signature examined, and in this regard three categories of signatures could be identified based on the effectiveness of a systems approach to their verification.

The first category of signatures are those that are formed by a relatively *few* number of predominantly *ballistic* strokes. The italicised letters in the previous sentence are especially important because both a few number of strokes and ballisticity contributes towards more consistency. Furthermore, in most signatures of this category the form of the signature tends to be ill-related to the name it is representing, which implies that the correlation between $v_x[n]$ and $v_y[n]$ in these signatures also tend to be lower i.e. what $v_x[n]$ does has little bearing on what $v_y[n]$ does. In summary $v_x[n]$ and $v_y[n]$ in this category are consistent but poorly correlated.

The fact that such signatures are both consistent and their dynamics profiles difficult to forge, the systems approach based on an ARX models performs especially well on them. Example specimens of such signatures are given below.

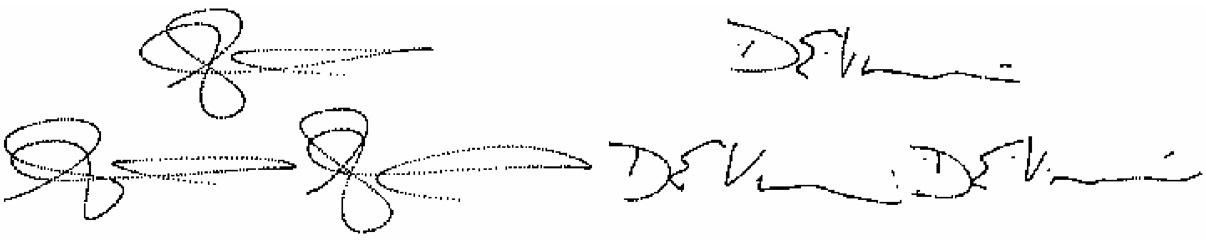


Figure 57 Examples of category 1 signatures

The second category of signatures includes those that are more similar to ordinary writing, where the signer tends to write out a signature rather than signing it. Such signatures usually have a few ballistic strokes accompanied by a large component of near-ordinary writing. Due to their similarity to ordinary writing, the $v_x[n]$ and $v_y[n]$ profile are usually centred around zero with a low RMS value and high numbers of peaks and troughs. Unfortunately, due to the lack of ballisticity the level of consistency within such signatures is also generally lower.

From experiments with signatures within this class, it has been noted that although the discriminancy of templates are compromised by a lack of emphatic ballisticity, the predictor can nevertheless distinguish between genuine and forged signatures. A pertinent observation in this regard is that forgeries in this category tend to carry considerably more turning points than the genuine signatures as a result of the more irregular behaviour of the forgery's predicted output. This is understood to be the result of a low yet fundamental level of consistency that exists within the signature, which is too obscure to be imitated by a forger but is captured by the template models. Examples of signatures in this category are shown below.

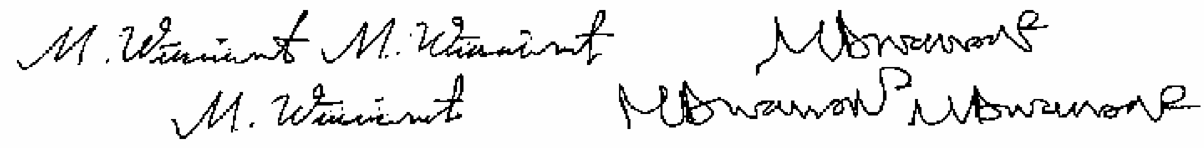
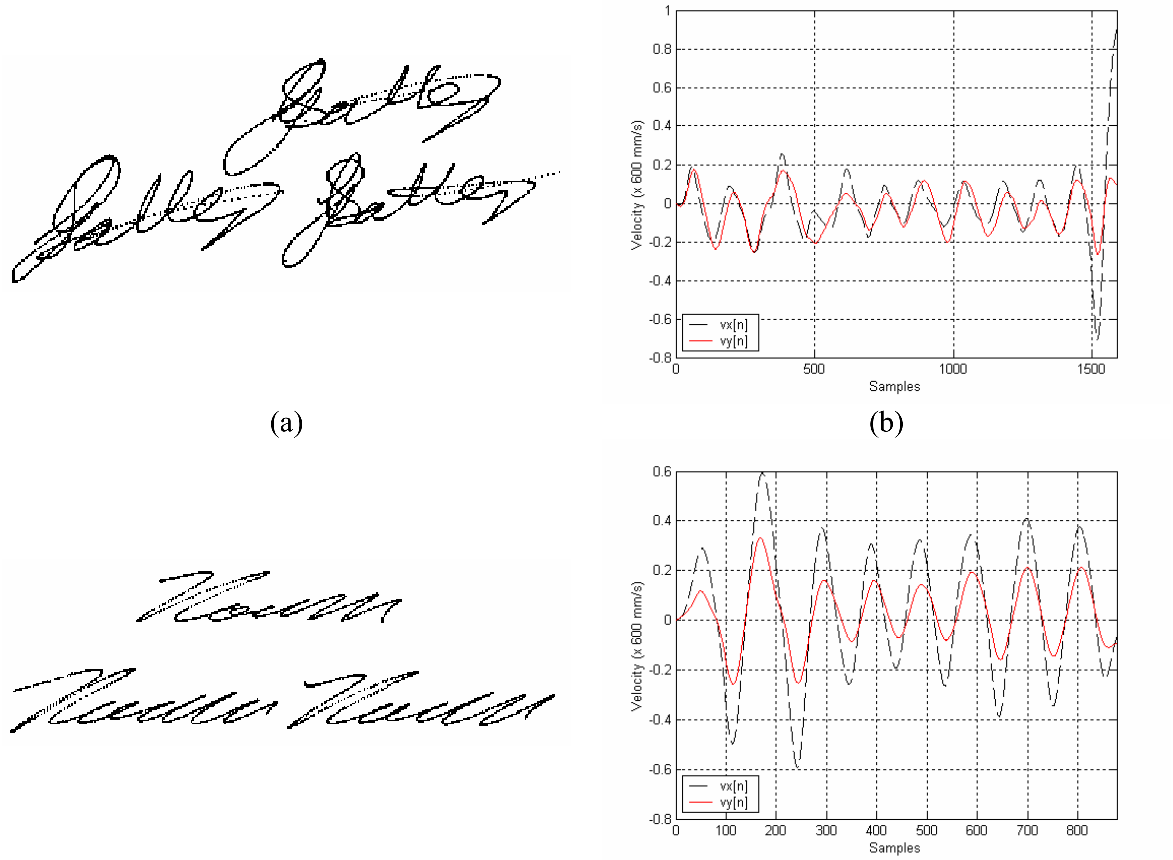


Figure 58 Examples of category 2 signatures



(a) (b)

(c) (d)

Figure 59 Examples of category 3 signatures

A final category of signatures includes those whose $v_x[n]$ and $v_y[n]$ profiles are almost purely sinusoidal in form – these present the toughest challenge for the systems approach to DSV. In Figure 59 the dynamic profiles (b) and (d) of the signatures shown in (a) and (c) are clearly almost purely sinusoidal, consequently a very simple yet tight correlation exists between $v_x[n]$ and $v_y[n]$. The implication hereof is that the information contained in the models are bound to be of little discriminancy, since they may very well be an elaborate version of a simple gain and phase shift.

Since 'sinusoidal' signatures tend to be more easily forgeable than others, the sinusoidal form of the $v_x[n]$ and $v_y[n]$ signals can also be easily imitated. Due to the in-phase sine-wave-in and sine-wave-out nature of these signatures, there is minimal information of pen-tip dynamics that can be captured by the template, thus the resulting fit values are often within the range of those of genuine signatures. This calls for the necessity of additional and conceptually separate parameters to complement the task of verification to ensure a low FAR, since it has been seen that it is unlikely that ARX alone would be sufficient for such signatures.

The above categorisation was instrumental in devising a comprehensive strategy that would be effective for the verification of all types of signatures given the framework of a systems approach to DSV.

Finally, it should be stated that regardless of the category of the signature, inconsistency is the greatest enemy of DSV. In the face of inconsistency, the fundamental philosophy of any DSV algorithm becomes counter-productive as a result of increased FRR. This is the focus of a plausible argument presented by Nalwa [11], where it was mentioned that the consistency in velocity profile that brings about security in DSV is a requirement that is absent in the traditional notion of a signature.

Furthermore, the increased FRR brought about by inconsistencies raises further questions into the 'true' meaning of false acceptances and false rejections. Perhaps, the most sensitive question to ask is whether society should accept inconsistent signatures i.e. why should a false rejection yielded by a DSV algorithm due to inconsistency be regarded as an 'inconvenience' or a 'nuisance'? This is the topic is presented in some more detail in Section 8.1.

6.6.6. List of Complementary Parameters

As mentioned in Section 6.5.2 on verification, various signal attributes were also calculated in search of useful complementary template parameters. In this section a number of parameters that the author found useful are listed, specifically in view of their discriminant power against skilled or random forgeries.

The first such parameter is the number of turning points in the predictor output $\hat{v}_y[n]$. It was observed that, in general, $\hat{v}_y[n]$ corresponding to a forgery tends to have significantly higher number of turning points due to the increased level of irregularity as a result of the e[n] term. Although this observation is generally true, it is especially useful during verification of category 2 signatures, since the fit values of forgeries corresponding to this category can at times yield similar fit values to those for genuine signatures, yet the number of turning points are almost always significantly more.

A second useful parameter identified was that of signing time. More often than not, a forger would practice the signature to be forged using a number of available specimens of the genuine signature. Since the dynamic information of a signature e.g. speed, rhythm, hesitations etc. is completely irrecoverable with such specimens, the sign time of a forgery is almost always significantly different from that of the genuine signature, especially in view of the fact that the forger is likely to produce the forgery relatively slowly in order to achieve a good visual match. Consequently, this dynamic attribute of the signature was chosen to be a crucial part of the verification process that is generically applicable for all three categories of signatures.

The third parameter used was the pen up-down ratio, defined as the percentage of total signing time that the pen tip is in the air, which can be considered to be a ballistic feature due to its consistency from one genuine signature to the next. Again, because such information is not derivable from a static image of a signature, it is likely to be very different to the actual value if a signature was not signed by the authentic owner. Experiments conducted have shown that, in general, the pen up-down ratio can be exceptionally helpful in distinguishing between genuine and forged signatures in general.

The next two parameters are concerned with the physical appearance of a signature, which would be useful for the detection of random forgeries as well as when the discriminancy offered by an ARX model-based template is insufficient e.g. for some of category 2 and all of category 3 signatures.

The one shape related attribute is the height to width ratio, which is an indication of the visual proportion of a signature. The ratio is defined as the quotient of the dimension of a signature in the x direction divided by the dimension of the signature in the y direction. During experimentation involving various genuine and forged signatures, it was observed that with signatures whose ending or dominant strokes are of high velocity, forgers tend to find it difficult to control both the velocity and length of the stroke simultaneously, thereby resulting in consistent yet inconspicuous distortions in the proportion of such a signature. Consequently, the height to width ratio was chosen to detect such telltale signs of forgeries.

The fifth and final parameter considered was the number of pen lifts in a signature. The number of pen lifts refers specifically to the explicit pen lifts found in a signature, and was chosen simply as a tool for the easy detection of random forgeries. Each of the five complementary parameters is, along with the mean and standard deviation of the fit value, stored as a part of the template.

This concludes Section 6, where we have determined the bounds of high discriminancy (Sections 6.6.1, to 6.6.4), established categories of signatures that require different emphases or verification strategies (Section 6.6.5), and identified a list of useful complementary parameters for a systems approach to verification (Section 6.6.6). From this point on, the author was able to fine-tune the various design parameters to make a systems-based algorithm generally applicable.

In adopting and combining the findings of all previous sub-sections of Section 6, the following set of parameter values were ultimately chosen as the standard parameter settings for template building and verification:

Model order	:	20
Segment length	:	60 samples
P.h. used for template optimisation	:	20 samples
Half overlap ratio	:	20%
P.h. used for verification	:	20 samples

The segment length was set to 60 samples in order to focus on the local behaviour of $v_x[n]$ and $v_y[n]$, while at the same time being sufficient to extend over a typical complete feature e.g. a peak or a trough. The minimisation of segment lengths has been previously explained to have the effect of affording the algorithm a higher level of discriminancy.

The highest derivable from a 60 sample long segment pair is 23rd order, however, to avoid possible boundary anomalies associated with 23rd order models (e.g. the last segment may not always be sufficiently long), 20th order models were chosen, which would also require less memory for storage. The act of maximising the model order can be explained by Section 6.6.1, where indisputable evidence was presented supporting the positive effect of higher model orders on the level of discriminancy. Next, a half overlap ratio of 20% was decided upon to provide a minimum amount of meaningful overlap facility according to the trend line of Figure 55. Again as previously shown, too large an overlap would lead to degraded model discriminancy.

A value of 20 was chosen for the prediction horizon as a compromise between the minimisation of the scatter of forgery fit values and fit value separation between genuine and forged signatures. According to Figure 56 a prediction horizon of 20 would give a typical separation of fit values in the tens of percents.

Clearly, the knowledge from experimentation described throughout Section 6.6 has formed the entire basis of decision-making concerning the value of the above algorithm parameters. Further experiments conducted using both used and unused signatures showed that the parameter values decided upon indeed yielded promising and satisfactory results, which can be seen in Section 8.

The investigations in this section have proven that a system approach to DSV is indeed a feasible practice, the results of which are the first available in terms of investigating the application of systems concepts and tools to the realm of signature verification.

7. Classifier

The classifier is the part of a DSV algorithm that, using various similarity measures, provides an indication of the overall similarity of a given signature representation to the genuine signature template used. The discussion of Section 3.2 provided instructive background for an appreciation of the challenges surrounding this function. From the author's review of the various algorithms in literature, two main groups of classifiers can be identified, in particular those based on neural networks and those based on statistical measures – both of which are powerful techniques in dealing with fuzzy inputs.

7.1. Classifiers in Literature

Examples of algorithms using neural network-based classifiers are the works of Keit *et al* [23], Wu *et al* [17], Martínez-R *et al* [18] and Martínez-R and Alcántara-S [19]. Without delving too deeply into the specifics, the first two algorithms had both adopted the widely used Multi Layer Perceptron (MLP) architecture trained using the back propagation algorithm. The specific number of nodes and hidden layers used in each case was however different.

In terms of usage of MLPs, the representation vectors or matrices are fed into the input layer, while a bounded value is retrieved at the output layer that is indicative of the level of confidence with which a signature can be classified as a genuine. By their nature, neural networks do not need the explicit input of templates, since they are intrinsically stored in the weighting of the synapses. Consequently, the 'definition' of the global measure of similarity distance is also inherent of the synapse weights in neural networks.

Both algorithms co-developed by Martínez-R involved the use of neural networks beyond that of direct 'feature value to decision' mapping. For example, in the algorithm of Martínez-R and Alcántara-S [19], decision-making forms a two-stage process where neural networks are used as sub-components to a greater Neural Network Driven Fuzzy Reasoning (NND FR) system. Recall that the representation matrix for the algorithm of [19] consists of a subset of shape-based characteristic functions proposed by Nalwa [11], as well as a subset of dynamic features. Accordingly, two FR modules (four neural networks) are implemented to separately evaluate the form and dynamics of a questioned signature. The final verification decision is made by a Mamdani-type fuzzy system that draws from the outputs of both NND FRs.

The remaining neural network-based classifiers cited hereunder were developed by Martínez-R, López-V and Luna Rosa [18]. The classifier constitutes of a three-layer neural network trained using supervised learning with the standard back propagation algorithm. In this case however, the output of the network is the *identity of the signer* expressed as a binary number. However because the output layer had only five neurons, the DSV system accommodated for a maximum of only 32 signers. Note that with such usage of neural networks, the objective has moved from signature verification onto writer identification.

In rationalising the use of neural networks for DSV problems, it immediately becomes apparent that the particular technique and the nature of problem are part and parcel of each other. On the one hand, the verification algorithms are faced with imprecise and varying inputs that possesses only gross consistencies, while on the other hand the user requires a quantitative measure of similarity distance – logically, neural networks form the natural choice for providing the qualitative to quantitative mapping required.

Next, statistical concepts form the other frequently adopted basis of classifiers, examples of which are the works of Dullink *et al* [16], Crane and Ostrem [12], Hastie *et al* [21], Liu *et al* [22] and Wu *et al* [14].

Specifically, the classifier used by Dullink *et al* used the most rudimentary of all statistical methods, where the deviation of features from template values are normalised by the standard deviation i.e. the z-score, after which the mean of the z-scores for all the features is evaluated as the similarity measure.

A more sophisticated method involves the evaluation of similarity measure based on a linear combination of individual z-scores i.e. a linear classifier. Common classifiers encountered include the linear discriminant function, Euclidean distance classifier etc. The classifier used by Crane and Ostrem [12] adopts this approach using Euclidean distance classifier defined by:

$$\mathrm{d}(\vec{s},\vec{t}) = \sqrt{\frac{1}{\mathrm{n}}\sum_{i=1}^{\mathrm{n}}\left(\frac{s_i - t_i}{\sigma_i}\right)^2}$$

Equation 22 Euclidean distance metric

where n is the total number of features, s_i is the ith component of the representation vector $\vec{s}$ containing all the features of the test signature, t_i is the ith component of the template vector $\vec{t}$ containing all the features of the template signature, and σ_I is the standard deviation of the ith feature as computed from a set of enrolment signatures. The value d constitutes the final similarity measure to which a threshold is then applied.

Although the use of the abovementioned statistical concepts so far have been only in instances where the similarity measure is evaluated from relatively few parameters e.g. tens of features, Hastie *et al* [21] had applied the general form of Euclidean distance measure (i.e. the least square difference) to entire distortion compensated signal segments to gauge the similarity between the corresponding parts of a signature. Hastie's classifier compares the least square distance between a test signature and a reference signature against the least square distances amongst the reference signatures themselves, and if they are comparable, then the test signature is deemed authentic.

When one expands the concept of similarity measure from a point-to-point comparison towards that of an overall correlation in time, the concept of cross-correlation comes into scope. For example algorithms of Liu *et al* [22] and Wu *et al* [14] had both used custom defined cross-correlation functions as classifiers, where the objective in either case has been to derive a single numeric indication of similarity between the test and template signatures as a whole.

Although also being reliant on statistical concepts, Nalwa's [11] classifier is however by far the most sophisticated of all statistics-based classifiers. Expressly, his classifier involves a serial process of optimising the cross-correlation of functions to their models by parametric warping, determining the individual errors between functions and their models, thresholding the z-score of errors and computing the root mean square of the individual biased and thresholded z-scores in obtaining a net local error. A conceptually similar process is also applied to four global static features, where the resultant net global error is combined with the net local error calculated from the characteristic functions by evaluating a weighted and biased harmonic mean value, and this final step yields the final similarity measure.

Apart from the neural network and statistic-based classifiers described above, there remain methods that depart from these more common approaches. For example, in McCabe's [15] algorithm where signature features are encoded into strings, the similarity distance is defined as the minimum cost in matching a test string to a template string using dynamic programming based on the Wagner-Fischer string distance algorithm. The technique of Dynamic Time Warping mentioned in Section 5.2.5 is exactly the same concept except the lowest cost route is not used to define a time-warp function, but a similarity measure – in fact this technique is called dynamic programming matching. During verification, the costs for matching the test signature to each of the 5 reference signatures are calculated, the minimum of which is compared to the mean and standard deviations values of the similarity distances amongst the reference signatures themselves – an approach conceptually similar to the one adopted by Hastie [21].

In the case of the classifier proposed by Nakanishi *et al* [20], an adaptive algorithm is placed in a feedback loop of a rather unconventional classifier.

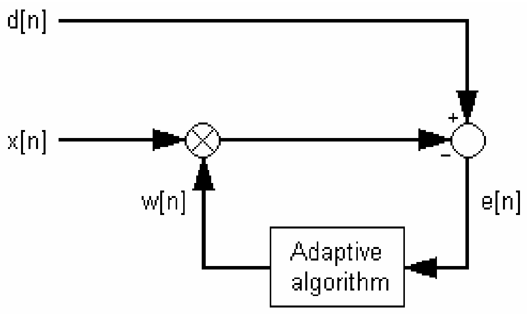


Figure 60 Nakanishi's classifier

Referring to Figure 60, d[n] is the template signal, x[n] is the signal generated from the test signature, e[n] is the error function and w[n] consists of running coefficients of multiplication, which are functions of e[n] via an adaptive algorithm. Conceptually, this is very similar to a controller-plant configuration where the controller strives to match the plant output to a given plant input as closely as possible. Analogously, if d[n] and x[n] are very similar, then the value of w[n] to achieve zero error is approximately unity, and e[n] is approximately zero. However, if x[n] differs significantly from d[n], the behaviour of w[n] would be more erratic in achieving a zero error. Based on this fact, the classifier makes a decision based on the historical values of w[n] i.e. the closer it is to unity, the greater the confidence in grading x[n] as a genuine signature.

With reference to the review of [2], the classifier types encountered above in fact constitute a fairly representative view of classifiers at large, with lower level differences in the specific methods and definitions used e.g. although other techniques such as histograms, likelihood ratios, square and 'city-block' distances etc. cited by [2] are not explicitly mentioned, they nonetheless remain housed under the category of statistics-based classifiers. Overall, the types of fundamentally different mechanisms used in classifiers encountered had not escaped the fields of neural networks, statistics, fuzzy logic, dynamic programming and certain clustering techniques, most of which have been represented in the current review.

7.2. Developing a Classifier

The first step in the formulation of a classifier suitable for the algorithm under development was looking at the set of inputs to the classification stage (Figure 61). The information available at the input of a classifier include the test, mean and standard deviations values of the fit, number of turning points, sign time, height to width ratio, number of pen lifts and the pen up-down ratio.

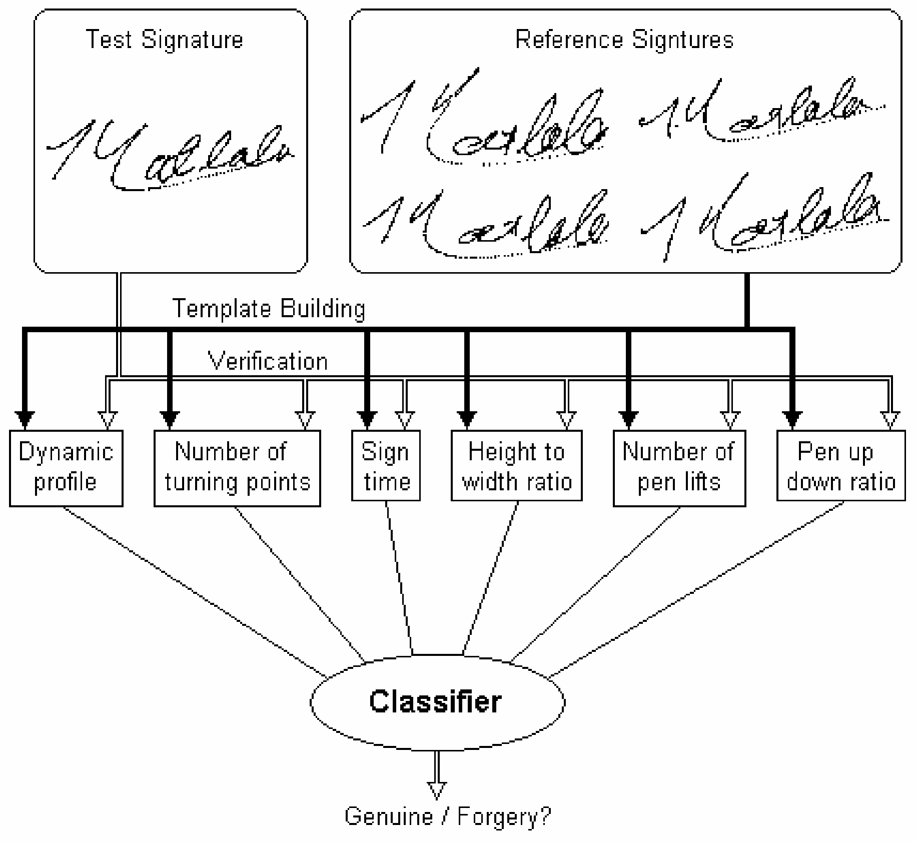


Figure 61 Development of classifier

In essence, the desired relationship between the input and output sets of the classifier must be such that when the fit value and the values of the 5 complementary parameters are substantially different from the template values, then the output of the classifier must indicate a high probability of a forgery. However, before a decision could be made between the adoption of either a neural network or a statistics based classifier, the behaviour of both the fit value as well as the complementary parameters need to be understood.

Firstly, during experiments conducted within the research, it was noted that although genuine signatures generally yielded positive fit values, the varying degrees of consistency associated with each signer meant that the general magnitude of fit values achievable by various set of genuine signatures differed. As a result, it was realised early on that classification could not be conducted on the pure basis of fit values directly, but rather on the mean of the fit value associated with a given genuine signature. This is the reason why both the mean and standard deviation of the fit values are also stored as a part of the template.

Secondly, the complementary parameters chosen have varying degrees of discriminant power. For example, although the sign time and pen up-down ratio can be highly consistent, the height to width ratio is, within stringent constraints, less so. It was therefore acknowledged that not all complementary parameters would be given an equal level of priority in view of the inherent consistency limit of each parameter. Nevertheless, a common set of priority settings would be adopted for all signers in the prototype algorithm.

Thirdly, the treatment of the number of turning points needs to be further elaborated. Ideally, the mean number of turning points stored in the template should be at its minimum value. The minimum value is defined as the number of turning points found in the actual $v_y[n]$ i.e. any additional turning points in $\hat{v}_y[n]$ must be the result of 'noise' due to increased influence of e[n] during prediction. Experience from repeated experimentation has shown however that the mean value of a number of turning points stored in the template may at times be non-minimum itself, which can be the result of intra-personal inconsistency etc. This means that when the template models are applied to a new specimen of a genuine signature, it is possible that the number of turning points in $\hat{v}_y[n]$ may in fact be fewer than that indicated by the template.

As a first approximation, the treatment of the number of turning points parameter can therefore be regarded as *zero-preferential* rather than *mean-value-preferential* i.e. a lower number of turning points should be rewarded for a genuine signature. One may question whether a random forgery with inherently fewer turning points in its $v_y[n]$ would also be rewarded. The answer is 'No', it is believed that by appropriate weighting, the penalisation given by the inevitably poor fit value would far out-weigh the reward for the fewer number of turning points.

Fourthly, recall that the parameter corresponding to the number of pen lifts was chosen as a convenient tool for the detection of random forgeries, and this is precisely how the parameter should be used. It is understood that since forgers invariably strive for signature shape similarity, the number of pen lifts within a forgery is inevitably similar to the genuine signature being forged. Without loss of generality, this fact would be helpful in a two-part classification scheme, where the first step in classification involves determining if the test signature is a *candidate for skilled forgery*.

The rationale hereof is that if the number of pen lifts in the test signature are considerably different from the template value, then the test signature must be significantly different from the genuine signature. In such cases the test signature can be regarded as a random forgery, which does not justify the use of further and more processing intensive verification functions. Should the number of pen lifts be similar however, the verification must then enter its second part, which could involve a classifier of any architecture.

Finally, experimental experience has also shown that templates generated by only 4 reference signatures tend to be over-stringent rather than over-tolerant. For example, it has been observed that the total sign time of a number of genuine signature sets often varied from the template mean by more than one standard deviation calculated from the 4 reference signatures. Consequently, the standard deviations of the fit value, sign time as well as the number of pen lifts were artificially increased by a factor of 1.5 in order to reduce the FRR. This means that the 'true' standard deviation for a given genuine signature is in fact larger than the value calculated based on only 4 signatures.

Given the above input characteristics, both a neural network and a statistics based approach would in principle be feasible. A natural choice was however to start experimenting with a conceptually simple approach, such as a simple linear classifier based on z-scores defined below:

$$d = \sum_{r=1}^{m} k_r \left(1 \pm \frac{|t_r - \mu_r|}{\sigma_r} \right)$$

Equation 23 Simple linear classifier

where the fraction after the ± sign constitutes the z-score, t_r is the rth parameter yielded by the test signature (or the rth test parameter), μ_r is the mean value of the rth parameter and σ_r the corresponding standard deviation. Clearly the z-score expresses the distance between a test parameter and the parameter's mean value in terms of the numbers of standard deviations. In the remainder of the above expression, k_r is a positive weight constant and d is the overall similarity measure whose positive extremity corresponds to a genuine signature. The use of the ± sign will be explained shortly.

Now, in the usual case that that the '–` part of the ± sign is used, it should be easy to see that the expression rewards similarity and penalises difference, with no bound being attached to the lower limit of d. Although this definition can be used for all the complementary parameters[26], the possible scenario of an authentic signer achieving a fit value that is much higher than the mean should not however be penalised in the same manner due to the high z-score. This is because the fit value is *positive-preferential* as opposed to *mean-value-preferential*. Consequently, only in the case of the fit value, is the ± sign conditional i.e.

if $(fit_{test} - fit_{mean}) \geq 0$ then the '+' sign is used
if $(fit_{test} - fit_{mean}) < 0$ then the '–` sign is used

With regards to the zero-preferential treatment of the number of turning points, a separate marking scheme needed to be formulated. A simple scheme of a parabolic curve was adopted whose marking schedule is shown by the graph below, where the horizontal axis is normalised by the mean number of turning points.

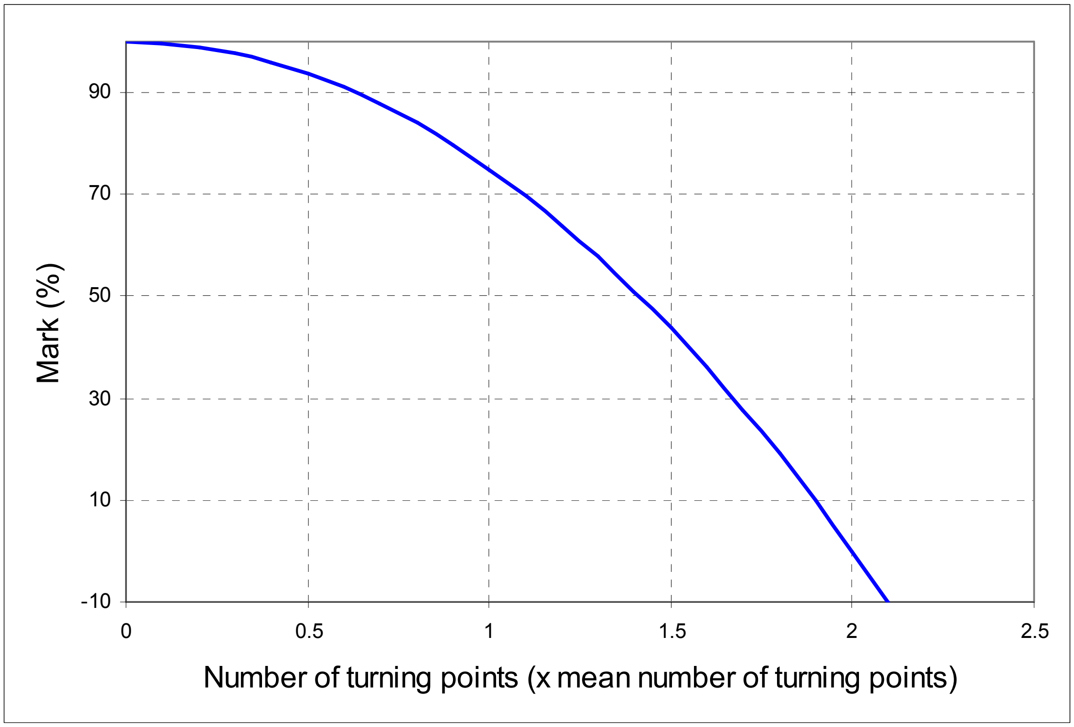


Figure 62 Zero-preferential marking scheme for the number of turning points

[26] With the exception of the number of pen lifts as explained previously.

The formula to the profile shown in Figure 62 is

$$\text{score} = -\left(0.5 \times \frac{\text{turning\_points}_{\text{test}}}{\text{turning\_points}_{\mu}}\right)^2 + 1$$

Equation 24. Marking formula for number of turning points

Specifically, when the number of turning points in $\hat{v}_y[n]$ is exactly the mean, the score obtained was set at 75%, while less turning points results in higher scores and vice versa.

The classifier outlined above is a simple one based on statistical measures, yet with repeated experimentation, using suitable values of k_r, the results achieved were promising (see Section 8). As a result the classifier presented above was adopted for the prototype algorithm, where a positive result of the summation is interpreted as a genuine signature while a negative result indicates a forgery.

As mentioned earlier, the same classification object can be achieved using a neural network, where the test and template parameters can be fed into the input layer, and a similarity measure retrieved at the output. However, unless neural network can realise significant improvements in the algorithm performance (which could be the topic of future research), the efforts required in the selection of network architecture, nodal function and training algorithm could not be sufficiently justified within the context of this research, where the focus was primarily on the feasibility of a systems approach to DSV.

Given the general framework of the classifier described above, the subsequent sections serves to provide more implementation details and consideration thereof.

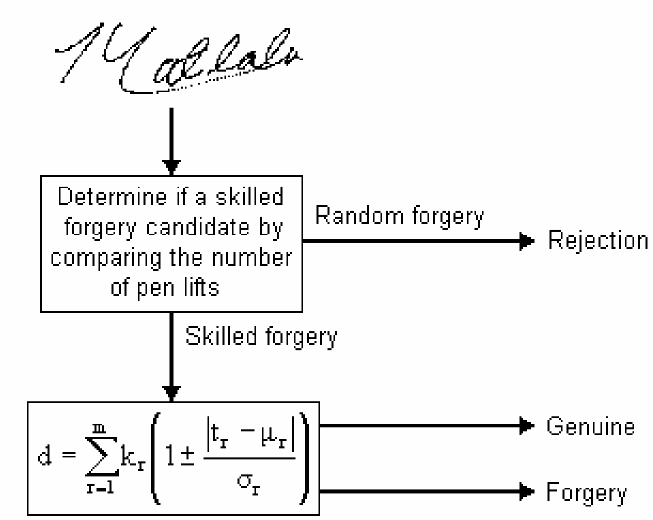

Determine if a skilled forgery candidate by comparing the number of pen lifts
Random forgery
Rejection
Skilled forgery
$d = \sum_{r=1}^{m} k_r \left(1 \pm \frac{|t_r - \mu_r|}{\sigma_r}\right)$
Genuine
Forgery

Figure 63 Implementation of the simple linear classifier

The first step of the classification is to determine the difference between the number of pen lifts in the test signature and template value. Generally, the number of pen lifts found in the 4 reference signatures is highly stable e.g. [4 4 5 4], and because of such behaviour the typical value of standard deviation for the number of pen lifts is in the range between 0 to 0.7. However, because the number of pen lifts can only differ by integer numbers in real terms, the condition for the detection of random forgeries is set to:

If abs(number_pen_lift – Numebr_pen_lift$_{mean}$) > ceil(pen_lift_std) × 1.5

then the test signature is a random forgery. The abs function in MATLAB evaluates the absolute value while the ceil function rounds a non-integer to the nearest integer in the positive direction e.g. ceil(0.231) = 1. In words, the above says that if the difference in the number of pen lifts in the test signature from the template is more than 1.5 times the standard deviation, then the test signature would be dismissed as a random forgery. The scaling of the standard deviation by 1.5 is in acknowledgement of that fact that due to the very few numbers of reference signatures used, the 'true' standard deviation may in fact be higher than mentioned previously[27]. More often than not, ceil(pen_lift_std) × 1.5 would evaluate to 1.5 which means a pen lift difference of 1 would be allowed. In practice such differences can arise due to a softly written stroke, which is sometimes captured as one continuous stroke and sometimes as two separate strokes using the pressure pad.

If the difference in the number of pen lifts is less than ceil(pen_lift_std) × 1.5 however, which is generally true of skilled forgeries, the signature enters the second part of the classifier. Given that only one fit value and four complementary parameters are handled in this stage, Equation 23 can be written in the complete and expanded form as below.

$$d = k_1\left(1 \pm \frac{\left|\text{fit}_{\text{test}} - \text{fit}_{\mu}\right|}{\text{fit}_{\sigma}}\right) + k_2\left(-\left(0.5 \times \frac{\text{turning\_points}_{\text{test}}}{\text{turning\_points}_{\mu}}\right)^2 + 1\right) + k_3\left(1 - \frac{\left|\text{sign\_time}_{\text{test}} - \text{sign\_time}_{\mu}\right|}{\text{sign\_time}_{\sigma}}\right)$$
$$+ k_4\left(1 - \frac{\left|\text{pen\_up\_down}_{\text{test}} - \text{pen\_up\_down}_{\mu}\right|}{\text{pen\_up\_down}_{\sigma}}\right) + k_5\left(1 - \frac{\left|\text{height\_width\_ratio}_{\text{test}} - \text{height\_width\_ratio}_{\mu}\right|}{\text{height\_width\_ratio}_{\sigma}}\right)$$

Equation 25 Simple linear classifier in expanded form

After heuristic tuning of the weighting constants, the eventual set of constants were chosen as follows:

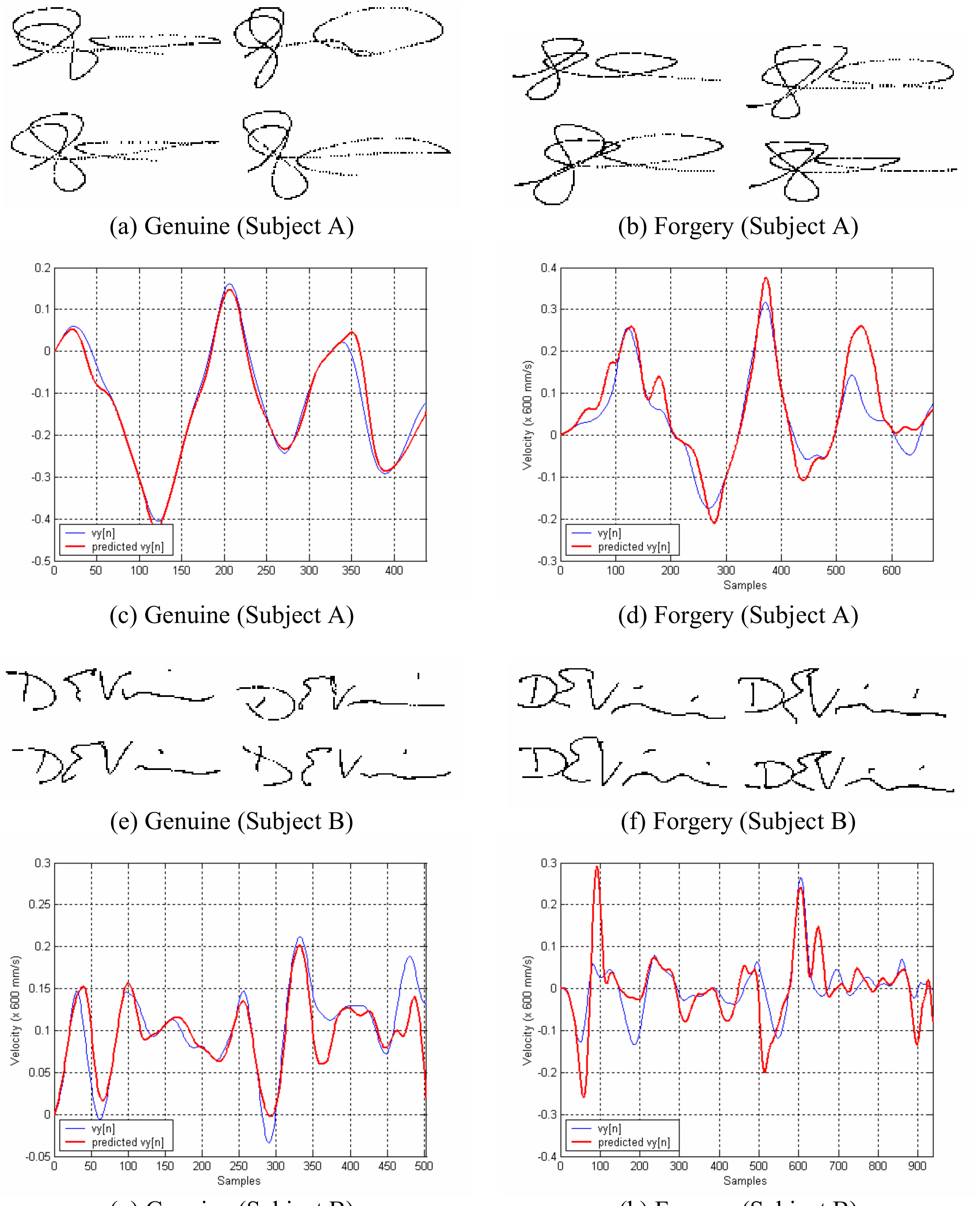
$k_1 = 100$ $k_2 = 50$ $k_3 = 50$ $k_4 = 20$ $k_5 = 15$

which gave the linear classifier a adequate level of discriminancy and accuracy. From the set of constants used, it is clear that the systems identification component of the algorithm is weighed at least twice as heavily as any other parameter towards the final similarity score. Remember that since scoring is based on the *mean* fit value (which is not necessarily 100%), the total achievable score for a signature may in fact be any number over 235 (the sum of the weight constants), consequently only the relative importance of each parameter can be quoted instead of an absolute one.

In terms of the performance characteristics of the classifier, its response to each category of signature also carries distinct attributes. For category 1 signatures, it was noted that the fit value alone would typically suffice in generating a correct classifier output, since the dominantly ballistic strokes would ensure a high level of inherent discriminancy, which can be well

[27] The 'true' standard deviation cannot in general be lower, since that would often imply a standard deviation of 0 due to the high level of consistency associated with the number of pen lifts.

exploited by the systems approach. In such cases, the roles of the complementary parameters are in fact more marginal. Examples of such signatures are shown below.



(a) Genuine (Subject A)

(b) Forgery (Subject A)

(c) Genuine (Subject A)

(d) Forgery (Subject A)

(e) Genuine (Subject B)

(f) Forgery (Subject B)

(g) Genuine (Subject B)

(h) Forgery (Subject B)

Figure 64 Classifier response to category 1 signatures

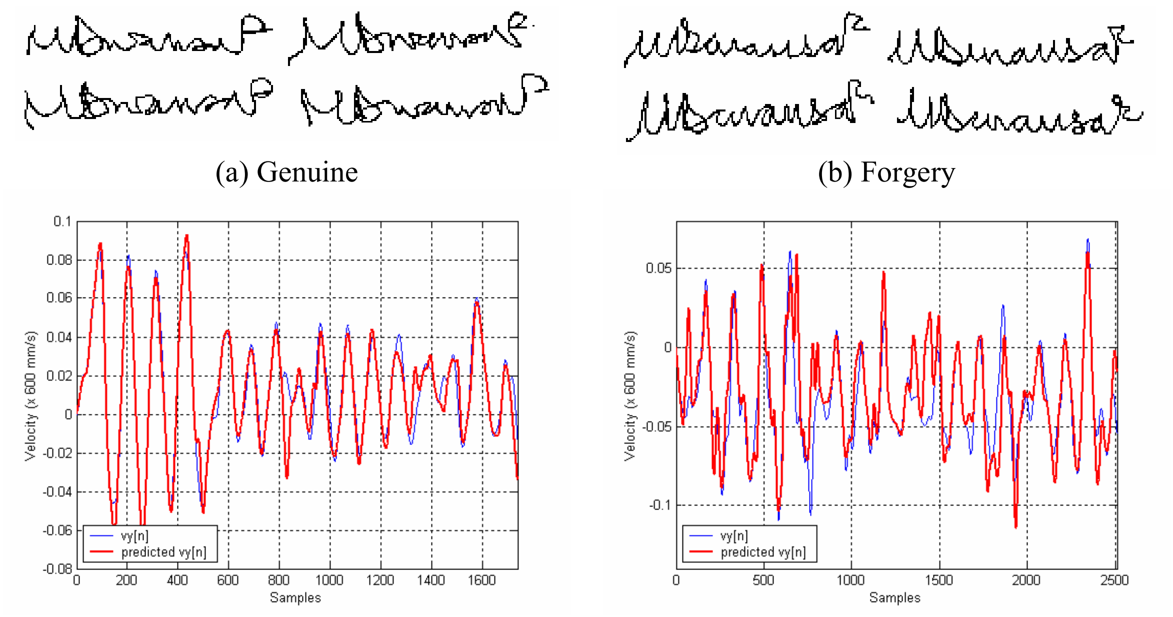
The appearance and velocity profile of two category 1 signatures and their forgeries are shown in Figure 64, while the corresponding fit value as well as the score contribution due to the fit values are shown in Table 5. Evidently, the fit values of both signatures are significantly different from their forged counterparts, and the corresponding fit values are therefore also polarised. Due to this fact, the fit values by themselves would have performed satisfactorily even without the aid of the complementary parameters. Although it is not implied that the forgeries used are of the highest quality, the above results nevertheless typify the similarity scores obtained based on the characteristics of category 1 ballistic signatures.

Table 5 Classifier response to category 1 signatures

Sig.	Subject A				Subject B			
	Genuine		Forgery		Genuine		Forgery	
	Fit (%) (μ = 79.473%)	Contribution (k = 100)	Fit (%)	Contribution (k = 100)	Fit (%) (μ = 33.658%)	Contribution (k = 100)	Fit (%)	Contribution (k = 100)
1	83.001	111.017	55.131	-74.788	29.593	51.805	-14.782	-165.467
2	82.029	104.535	66.505	1.039	33.580	71.328	-1.130	-98.623
3	75.953	64.028	56.352	-66.650	3.663	-75.160	19.714	3.435
4	81.599	101.669	58.179	-54.463	43.169	118.276	16.143	-14.050
5	80.926	97.178	63.590	-18.395	38.499	95.410	15.698	-16.229
6	81.012	97.753	62.224	-27.497	30.481	56.153	2.171	-82.462
7	82.497	107.654	52.058	-95.270	29.669	52.177	12.910	-29.880
8	87.127	138.520	51.467	-99.211	47.733	140.624	25.783	33.152
9	60.799	-37.002			37.435	90.200		
10	79.795	89.642			42.755	116.250		

For category 2 signatures, due to the lowered level of consistency and discriminancy as explained previously, the role of the complementary parameters becomes pivotal. The typical response of the classifier to category 2 signatures are shown in Figure 65. Although the fit values of these forgeries are of generally lower value, they are not necessarily significantly different from their genuine counterparts. But as mentioned before, observe that the predicted velocity $\hat{v}_y[n]$ for forgeries is notably more 'noisy' than that of a genuine signature – this telltale sign of a forgery is captured by the number of turning points, whose mean and standard deviation values are stored in the template. This is verified by looking at the corresponding parameter value in Table 6 below.

Note that although the fit values of category 2 genuine signatures are not as discriminant as for category 1 signatures, the number of turning points, however, are. With forgeries, it is unmistakable that the number of turning points yielded is of consistently higher value (observe Figure 65 (d)), where there can in fact be twice as many turning points as indicated by the template mean. Using this fact, the score contribution due to the number of turning points would be notably lower for forgeries of category 2 signatures.



(a) Genuine (b) Forgery

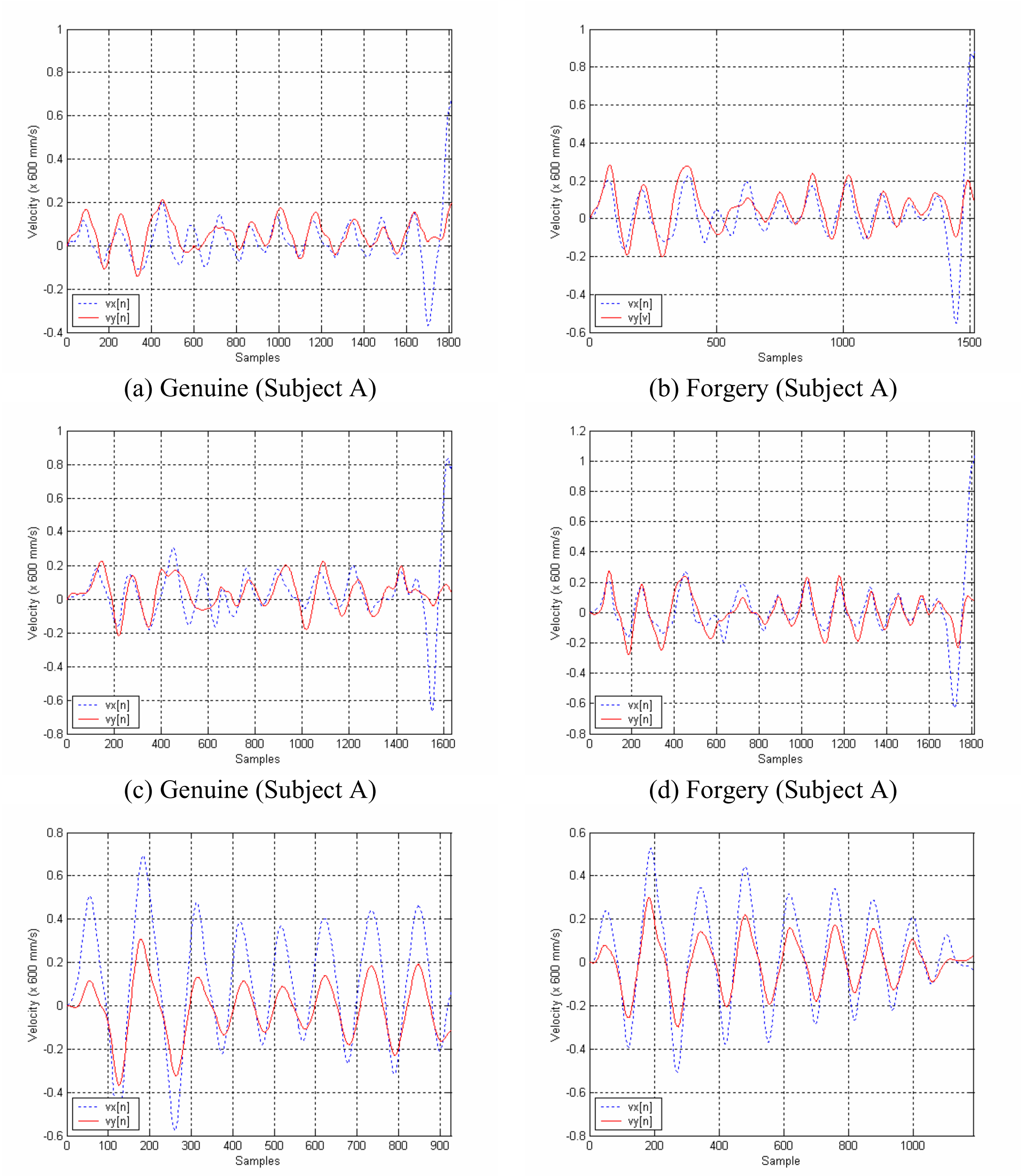
(c) Genuine (d) Forgery

Figure 65 Classifier response to category 2 signatures

Table 6 Classifier response to category 2 signatures

Sig.	Genuine				Forgery			
	Fit (%) (mean = 58.45%)	Contribution (k = 100)	Turn. Pts. (mean = 41.33%)	Contribution (k = 50)	Fit (%)	Contribution (k =100)	Turn. Pts.	Contribution (k = 50)
1	62.594	115.676	44	35.835	34.182	8.201	78	5.486
2	71.088	147.806	41	37.701	50.724	70.773	71	13.117
3	44.634	47.737	48	33.143	38.547	24.711	50	31.709
4	63.801	120.241	43	36.472	58.424	99.990	58	25.387
5	40.468	31.979	46	34.518	50.834	71.191	64	20.031
6	64.999	124.772	48	33.143	27.656	-16.485	60	23.660
7	38.248	23.580	45	35.184	22.281	-36.818	67	17.156
8	57.658	97.002	40	38.293	48.265	61.472	57	26.228
9	19.255	-48.265	39	38.871				
10	56.463	92.483	40	38.293				

Finally, the classifier response to category 3 or ‘sinusoidal’ signatures is presented. Recall that since the covariation relationship in category 3 signatures are highly simple, efforts to capture idiosyncratic features in the velocity profile of such signatures using models are often futile. The difficulty with category 3 signatures is compounded by the fact that unlike category 2 signatures, where the number of turning points for example can be discriminant against forgeries, no simple measures based on the correlation behaviour exist for category 3 signatures.



(a) Genuine (Subject A)

(b) Forgery (Subject A)

(c) Genuine (Subject A)

(d) Forgery (Subject A)

(e) Genuine (Subject B)

(f) Forgery (Subject B)

Figure 66 Velocity profile of sinusoidal genuine signatures and their forgeries

Now since the difference between the observed signal $v_y[n]$ and the predicted signal $\hat{v}_y[n]$ for both genuine and forgeries are often subtle for sinusoidal signatures as shown in above, there is often no grounds to undisputedly classify a forged signature as a forgery from a velocity profile point of view. Consequently, reliance is transferred from the template models almost entirely to the complementary parameters for correct verification, whose roles thus become especially crucial. This means that, for a generally effective algorithm, one has to include parameters and similarity measures that are conceptually separate from the systems approach – in the prototype algorithm just 4 such parameters are used. The results shown in Table 7 are for the signatures shown in Figure 59 (a) and (c), where each row shows the mean values of the contribution due to each parameter towards the overall score. It is clear that the combined result of all the parameters used were unable to emphatically distinguish between the genuine and forged signatures.

Table 7 Classifier response to category 3 signatures

	Subject A		Subject B	
	Genuine	Forgery	Genuine	Forgery
Fit value	57.931	74.815	-11.140	15.218
Turning points	36.329	38.244	33.988	31.999
Sign time	23.539	-5.495	43.971	38.934
Pen up/down ratio	-6.810	-11.500	3.012	3.012
Height width ratio	-16.026	-12.982	-2.022	-28.794
Score	**94.967**	**83.130**	**67.810**	**60.369**

Naturally further parameters would be required to more comprehensively complement this particular weakness of a model-based DSV algorithm, and such a search stands as a topic worthy of future research. Nevertheless, the crucial finding given the focus of this research is that it has been discovered and acknowledged that: firstly there is a category of signature for which the systems approach does not perform strongly, and secondly a set of mutually independent[28] parameters would be needed to fully complement a model-based DSV algorithm.

The above summarises the typical response of a simple linear classifier applied to the three categories of signatures. Based on these findings as well as experience gained from experimentation a further number of points worthy of mention remain.

It was mentioned previously that inconsistency is the enemy to a DSV algorithm in general, but ironically experiments have also shown that too much consistency may also give rise to false rejection of genuine signatures. This is because only four reference signatures are used to create the template, and if all four signatures are extremely similar the value of standard deviations stored in the template would be unduly low. The result is that when the template is used to verify subsequent genuine signatures, any previously unseen variations would be 'incorrectly' penalised thereby leading to false rejections[29]. In view of this, it would in fact be advisable to ensure that during the enrolment phase users be encouraged to sign variably within what the user deems to be reasonable range of intra-personal variation.

Furthermore, in order to address the problem of category 3 signatures, a number of parameters were identified during research that can be discriminant against forgeries of category 3 signatures, they were however not included as a part of the final classifier since their consistency, discriminancy and therefore effectiveness varied markedly from one signature to another. Nevertheless, signal features such as initial pen stroke direction, maxima, minima, mean value, root-mean-square values, average positive or negative values etc. can be incorporated into the classifier once their behaviour and characteristics are better understood.

The final aspect in terms of classifiers is the prospect of employing classifiers of other formats. Specifically it is recommended that future investigations be carried out into the possibility of

[28] Independent in the sense that they carry little redundant information, thereby describing a signature by features in separate metric domains.

[29] The same problem was also observed with the commercial DSV product used in Section 8.

adopting of a neural network classifier, which could be beneficial in realising higher resilience against forgeries of sinusoidal signatures and thus improved performance.

8. System Evaluation

The combination of the data capturing, systems identification, template building, verification and classifier functions developed in the prior sections means that all the necessary components of a DSV algorithm are in place for the implementation of prototype DSV algorithm. The point of interest in the next phase of the research was thus an assessment of the overall performance of the combination of the individual subsystems. This section therefore focuses on the concepts and considerations pertinent to the *evaluation* of a DSV algorithm, which is presented in conjunction with a brief overview of the performance characteristics of DSV algorithms encountered in literature.

The latter part of this section is dedicated to presenting the result of evaluations carried out for the resultant algorithm of the research under discussion. In order for the results obtained to be interpreted in context, the performance of the prototype algorithm is presented along with the results of a similar evaluation procedure applied to a commercially available DSV product, which was adopted as the benchmark standard in the accompanying discussions.

8.1. DSV Algorithm Evaluation Considerations

The core discussion of Section 8 shall begin with a review of the performance of various algorithms in literature, while at the same time highlighting certain considerations requiring acknowledgement when making comparisons of the various DSV algorithms.

Due to the variety of conditions under which research on DSV algorithms are carried out, it has to be stated that no universal standard has seen to be adopted for the evaluation of DSV algorithm performance. Consequently, numerous issues need to be considered when interpreting a given set of algorithm performance measures.

In terms of the specific system evaluation parameters, numerous factors need to be considered when citing the FRR or FAR of a system, the main ones of which include:

- The size of a signature database
- The diversity of signatures in a database
- The time period over which signatures in a database is collected
- The conditions surrounding the signing process
- The definition of FAR and FRR adopted
- The information a forger has access to
- The definition of a successful verification
- User enrolment criteria

each of which is discussed in turn in the subsequent paragraphs.

One of the most important aspects of DSV algorithm evaluation is its robustness when used by a large population of signers. In research this is often emulated using a representative sample of signatures stored in a database, the size of which ranges from just tens of signatures in some instances to thousands of signatures in others [2]. Naturally, the greater the sample size used for testing the more accurate the quoted error rates would be. Consequently, attention should always be paid to the number of subjects used in obtaining FRR and FAR values, since test results obtained by involving just 10 signers are far less reliable than those obtained by involving 1000 signers say.

Secondly, since most DSV algorithms are developed with the ultimate aim of being widely deployed in society, it is equally important for an algorithm to be able to accommodate a diverse spectrum of signers specifically in terms of cultural backgrounds, age, gender, handedness etc. Consequently, the quality of a signature database lies not only in its size, but also the mixture of the constituent. It is often the case that the signatures are collected from colleagues or volunteers in a research or working environment, where it is highly probable that the individuals have come from fairly similar backgrounds, which may or may not be a concern.

Thirdly, since it is widely acknowledged that people's signatures may evolve over time, the robustness of a DSV algorithm is thus also a function of time, in which case the handling of signature evolution should be examined as a important feature of an algorithm. Since some algorithms incorporate adaptive learning, the effectiveness of such mechanisms can only be evaluated if signature samples in a database are collected over a sufficiently long period of time so that the effects of signature evolution becomes notable, and this typically spans over months or years.

Next, the values of FRR and FAR are also strongly affected by the conditions surrounding the signing process. In some cases, the signers are restricted to a comfortable sitting position during signing, while in others the signers are allowed to assume any desired posture.

The fifth factor mentioned earlier was the definition of FAR and FRR adopted, as will be seen, the understanding of which is a problem in itself. Nalwa made a strong argument for the subjective, and perhaps impossible to define, nature of the terms FAR and FRR [11], to quote him:

> *The specification of a trade-off curve relating false accepts to false rejects assumes knowledge of, and agreement on, a ground truth. In reality, it is not always clear what a false reject is. In particular, is it an error to reject a signature that is produced by the original signer but that looks substantially different from that signer's specimen signatures? If not, who or what decides whether two signatures look different? If yes, are we affording opportunities to individuals to disown their signatures after the fact?*

The emphasis here is that firstly, the concepts of FAR and FRR are inherently subjective in the context of DSV. In fact, the purpose of the research described in this document is one step towards circumventing this problem based on a crude definition of FRR and FAR; recall that in the introduction of this dissertation it was stated that the aim of the current research was to develop methods of verification that are independent of the specific signal form, but rather based the characteristics of the fundamental signal generation mechanism.

Even when broad definitions are assigned to these terms however, the procedures adopted in evaluating them still differ widely. In particular, it has been noted that whereas some researchers had only used random forgeries in evaluating the FAR, others had used both random and skilled forgeries[30]. Likewise, bad but genuine signatures were discarded in some experiments but not in others – procedural inconsistencies such as these, in conjunction with different conceptual understandings of FAR and FRR, makes direct comparison of algorithms impossible.

[30] FAR obtained using only random forgeries is, in the opinion of the author, of little practical significance, since the information of real interest is the strength of a DSV algorithm when faced with a persistent forger determined to impersonate as the genuine signer.

A further major inconsistency observed in the literature relates to the privileges afforded by the forger. In cases where skilled forgeries are used in evaluating the FAR, the forgers, be they professional or amateur, were granted varying levels of privileges in different researches. In particular, the privileges are in terms of access to signature information, access to feedback from the DSV system during repeated trials, financial and time resources permitted etc. For example, Crane and Ostrem [12] permitted the forgers to view video tapes of the signatures being signed, allowed as much practice as needed and even set up prize money for the best forgery, while Nalwa [11] permitted forgers to observe feedback from the verification system, where he noted that forgers with the capacity to understand the feedback provided by the system were able to learn from such feedback [11].

Such practices may or may not be possible in practical terms, but the fact remains that they lead to forgeries of varying qualities, which in turn affects the FAR quoted.

The seventh factor to be taken into account is what developers define to be a successful verification. In most cases, researchers have allowed one signature per verification attempt, however Crane and Ostrem [12] viewed a maximum of three genuine signatures as one single verification attempt, in which case the true FRR would be higher than that quoted when the one-signature-one-trial definition is applied. If subtleties such as this are overlooked, the results of a study may be very misleading indeed.

The final factor mentioned is that of a user enrolment criterion. As noted in [12], it can happen that almost all the error rates associated with a system were the result of a very small percentage of users who were the most inconsistent in writing their signatures (referred to as 'goats' of the system). Consequently user enrolment criteria are sometimes put in place in order to reduce the influence of the 'goats', often with the result of improved FRR and FAR ratings. However, depending on the eventual application of a given DSV scheme, such an enrolment procedure may or may not be acceptable to the users. In the author's opinion, researchers should always aim to emulate the conditions surrounding the final implementation of the system as closely as possible during testing, especially in the absence of a standardised code of evaluation.

The preceding paragraphs listed and described the major factors surrounding the evaluation of DSV systems that may give rise to inconsistencies in the interpretation of test results. Irrespective of the specific gauging techniques used, the reader should recognise that the problem of signature verification is not a precise science and will always be an art to a greater or lesser extent. In view of these facts, it would nevertheless be informative to gain a feel of the magnitude of errors associated with DSV algorithms in general, which is briefly stated below.

Irrespective of the specific definitions or procedures adopted, the range of FAR and FRR encountered in the review of [2] are predominantly in the range of a few percent, with a number of algorithms having error rates below 1% and even fewer with errors rates above 10%. This is consistent with the findings of the current research, and in particular the error rates of the algorithms described in Section 4, 5 and 7 are listed in Table 8, where available:

Table 8 List of evaluation results of algorithms described

Developer	**FRR**	**FAR**	**Notes**
Martínez-R & Alcántara-S [19]	1.05%	0.27%	Global correct verification rate of 99.86%
Dullink *et al* [16]	14.3%	2.1%	10 sample signatures with threshold biased towards low FAR
Martínez-R [18]	–	–	Global correct verification rate of 95%
Wu, Jou & Lee [17]	–	–	Equal error rate of about 4.2% based on Chinese signatures
Liu, Herbst & Anthony [22]	1.7%	0.4%	User enrolment requiring user to successfully pass verification first
Crane & Ostrem [12]	–	–	Best error rates in the region of 0.5% – 0.75%, depending on enrolment criteria, but otherwise centred around 1.5%
Nalwa [11]	–	–	Equal error rate of about 3.6%
Keit *et al* [23]	2.13%	3.4%	
Wu, Jou & Lee [14]	0.12%	0.04%	Minimum error rates based on 10 signers only, using 4 levels of splitting depth
McCabe [15]	–	–	The equal error rate appears to be in the region 3% - 4%

So far it should be evident that most available methods can achieve reasonably accurate results when applied to the problem of DSV. The challenge however is in the improvement of the results beyond these levels, specifically when one aims to reduce the FRR and FAR by a further order of magnitude.

In conclusion, it seems that by the nature of the problem, the reduction of error rates to below 0.1% and under may not be possible with the current knowledge and understanding of the writing or signing process. However this is not the only limitation in the reduction of error rates, one also has to realise that the quality (resolution) of an error rate measurement is directly proportional to the size of the database of sample signatures used. When one attempts to accurately measure error rates in the order of 0.001% for example, the number of signatures used in generating such a result must by definition be in the order of 10,0000 or greater, which in most cases would be beyond the means of a DSV research team.

Practically speaking however, the availability of such a vast signature database would still be of limited value unless accompanied by a revolutionary breakthrough in understanding the neurophysiological and biomechanical aspects of the writing process.

8.2. Algorithm Evaluation Results

The first step of algorithm evaluation involved the collection of signatures. In total 350 signature specimens were collected from 21 subjects in the compilation of a moderately sized signature database, where most of the signers were research students who worked in the same laboratory as the author. Each subject was allowed to decide whether a signed signature should be saved into the database or not based on its quality, since an admittedly bad genuine signature would distort the evaluation results through unduly high FRR. All signatures were captured using the hardware described in Appendix 1, with the signer being in a seated posture. It is important to note that signatures were signed on paper cards placed above the writing area so that signers could see the signature during the signing process.

In terms of the mixture of signatures in database, the signers included people of both sexes and handedness, although it was somewhat unfortunate that the age of most of signers was between 20 and 30 since they were mostly students. The spectrum of cultural background encompassed was however especially diverse. Specifically, the pool of subjects included people of European, African and Asian decent, which added much value to the signature database.

The next attribute of a database is its time period of compilation, and in this case signatures were collected over a period of three months. It should be stated however that not all signatures were collected over the full duration and for those signatures that were, they were not collected at regular intervals over the three months period. Specifically, the signatures of 9 signers were collected in two separate sessions separated by at least 2 weeks, while the remaining signatures were all collected in a single session.

For the forgeries on the other hand, 3 volunteers were given specimens of genuine signatures to be forged without disclosing the identity of the signatures' owner. In order to emulate the most likely practical conditions surrounding forgeries, the forgers were only given the visual images of the signatures, and were not given or allowed access to either the dynamic characteristics of a genuine signature or feedback from the system with regards to the quality of a forgery.

No user enrolment criteria were established during evaluation to discriminate against certain types of signatures or signers, the reason for this was two fold. Firstly it is natural to maximise the exposure of the algorithm to various signatures in an effort to discover its strengths and weaknesses. Secondly, there was in fact insufficient understanding of the algorithm to justify the classification of certain signers or signatures as potential 'goats' of the system. Consequently, all signatures had to be accepted without bias.

The definition of a verification attempt adopted was that of 'one signature one verification attempt'. This is because the 'many signature one verification' such as the one adopted by Crane and Ostrem [12] was considered to be an over-demanding and unacceptable definition in practice, and would only contribute towards an artificially lowered FRR. Finally, the definition of false acceptance error and false rejection error adopted were as follows:

False acceptance error : any forgery, either skilled or random, that is deemed to be highly probably a genuine signature as indicated by a value of classification outcome above the threshold.

False rejection error : any genuine signature within the range of intra-personal variations deemed to be reasonable by the signer that is deemed to be highly probably a forgery as indicated by a classification outcome below the threshold.

The default threshold value used was 0.

Due to the fact that the signers were given the privilege of allowing admittedly bad genuine signatures to be discarded, the above interpretation of false rejection error used was acceptable. The reason hereof is that the set of database signatures are of sufficiently good quality, such that they are minimally affected by the argument of Nalwa [11] presented in the previous section.

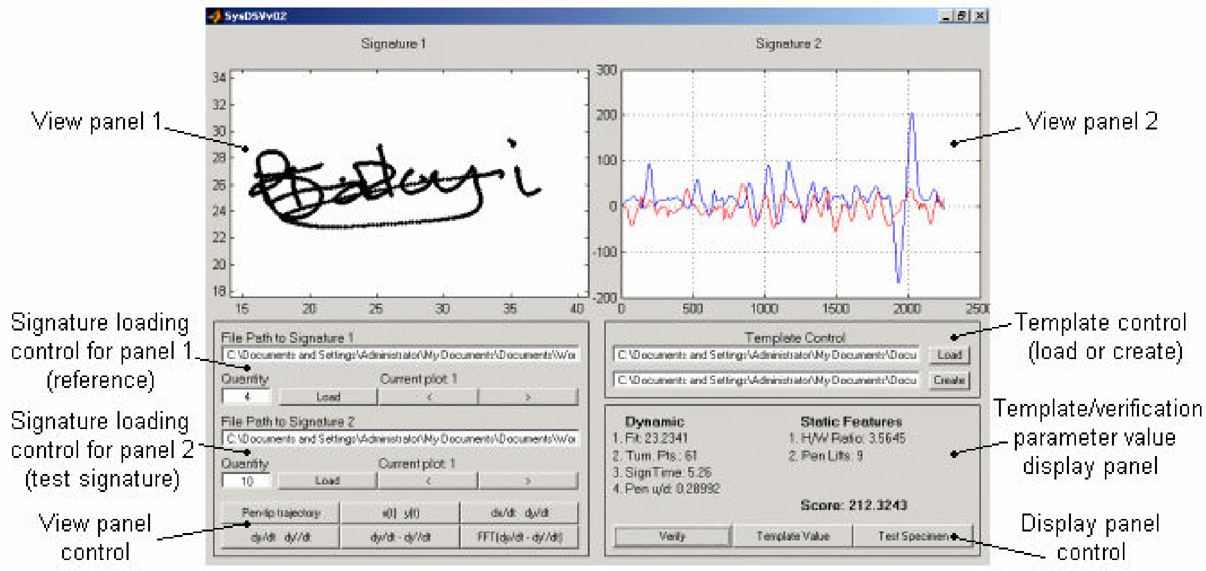
To facilitate easy viewing and comparison of signatures and their dynamic profiles, as well as template building and verification, the author built an integrated work environment shown below using the `guide` Integrated Development Environment (IDE) of MATLAB.

The integrated toolset is named SysDSV, and allowed simultaneous viewing and thus comparison of two signatures. Six possible views are also provided for the analysis of signatures, the views are:

- The pen-tip trajectory
- $x[n]$ with $y[n]$
- $v_x[n]$ with $v_y[n]$
- $v_y[n]$ with $\hat{v}_y[n]$
- $v_y[n] - \hat{v}_y[n]$
- FFT of $v_y[n] - \hat{v}_y[n]$.

SysDSV also allows instant creation of templates based on all signatures loaded into view panel 1, as well as viewing and comparison of verification results and template parameter values.

For each set of genuine signatures, the first 4 signatures are loaded into view panel 1 whilst the remaining signature specimens are loaded into view panel 2. Thereafter, a template is created from the 4 reference signatures of view panel 1, after which the template parameter values are shown in the parameter display panel. The final step of evaluation was the verification of each genuine signature specimen of view panel 2 against the derived template, followed by observation of both the classification score as well as the individual parameter values – this process is for the assessment of FRR. The parameter display panel was useful for identifying the reasons for false rejection.


File Path to Signature 1
C:\Documents and Settings\Administrator\My Documents\Documents\Wo
Quantity 4
Current plot: 1
File Path to Signature 2
C:\Documents and Settings\Administrator\My Documents\Documents\Wo
Quantity 10
Current plot: 1
Template Control
C:\Documents and Settings\Administrator\My Documents\Docu
C:\Documents and Settings\Administrator\My Documents\Docu

Dynamic
1. Fit: 23.2341
2. Turn. Pts.: 61
3. SignTime: 5.26
4. Pen u/d: 0.28992

Static Features
1. H/W Ratio: 3.5645
2. Pen Lifts: 9

Score: 212.3243

Figure 67 SysDSV version 0.2 graphical user interface

In determining the FAR, each template (created from genuine signatures) was loaded into the environment, while the corresponding forgeries were loaded into view panel 2. Each of the forgeries were then rotated through the view panel and verified against the corresponding template.

The experimental results are tabulated in Table 9, where each row represents a template of a genuine signature and each column represents sets of test signatures consisting of only genuine signatures. For ease of explanation on interpretation of the table, the first two columns of row 1 of the table are shown.

	1	2	
1	10	2	
	10	25	

Each row corresponds to one template and each column corresponds to a signer. The 10/10 in row 1 column 1 means that when a set of 10 signatures produced by subject 1 was verified using subject 1’s own template (therefore the grey shading), all 10 signatures were accepted as genuine signatures. In row 1 column 2 however, a set of 25 signatures signed by subject 2 were verified using the template of subject 1 i.e. *random forgeries*, and two were falsely accepted as genuine signatures. Lastly, the last column of Table 9 indicates both the total number of random signatures submitted (bottom) and the number of false accepts encountered (top).

Table 9 System evaluation results based on standard parameter settings (part 1)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	**Tot.**
1	10	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	**3**
	10	25	22	14	20	23	14	6	14	17	7	14	14	14	14	22	16	20	9	22	14	**321**
2	0	11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**
	14	21	22	14	20	23	14	6	14	17	7	14	14	14	14	22	16	20	9	22	14	**310**
3	0	0	11	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**8**
	14	25	11	14	20	23	14	6	14	17	7	14	14	14	14	22	16	20	9	22	14	**313**
4	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**
	14	25	22	10	20	23	14	6	14	17	7	14	14	14	14	22	16	20	9	22	14	**321**
5	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**
	14	25	22	14	11	23	14	6	14	17	7	14	14	14	14	22	16	20	9	22	14	**315**
6	3	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	**4**
	14	25	22	14	20	11	14	6	14	17	7	14	14	14	14	22	16	20	9	22	14	**312**
7	0	0	0	0	0	0	4	0	0	0	4	0	0	0	0	0	0	0	0	0	0	**4**
	14	25	22	14	20	23	10	6	14	17	7	14	14	14	14	22	16	20	9	22	14	**321**
8	0	0	0	0	0	0	0	2	0	3	0	0	0	0	0	0	0	1	0	0	0	**4**
	14	25	22	14	20	23	14	2	14	17	7	14	14	14	14	22	16	20	9	22	14	**329**
9	0	0	0	0	0	0	0	0	9	0	0	0	0	0	0	0	0	0	0	0	0	**0**
	14	25	22	14	20	23	14	6	10	17	7	14	14	14	14	22	16	20	9	22	14	**321**
10	0	0	0	0	0	0	0	0	0	7	0	0	0	0	0	0	2	0	0	0	0	**2**
	14	25	22	14	20	23	14	6	14	13	7	14	14	14	14	22	16	20	9	22	14	**318**
11	0	0	0	0	0	0	0	2	0	0	3	0	0	0	0	0	1	0	0	0	0	**3**
	14	25	22	14	20	23	14	6	14	17	3	14	14	14	14	22	16	20	9	22	14	**328**
12	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	1	0	**1**
	14	25	22	14	20	23	14	6	14	17	7	10	14	14	14	22	16	20	9	22	14	**321**
13	0	0	0	14	0	0	0	0	0	0	0	0	9	0	0	0	0	0	0	0	0	**14**
	14	25	22	14	20	23	14	6	14	17	7	14	10	14	14	22	16	20	9	22	14	**321**
14	0	0	0	0	0	0	0	5	0	7	0	0	0	9	0	0	2	0	0	0	0	**14**
	14	25	22	14	20	23	14	6	14	17	7	14	14	10	14	22	16	20	9	22	14	**321**
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	0	0	0	0	0	0	**0**
	14	25	22	14	20	23	14	6	14	17	7	14	14	14	10	22	16	20	9	22	14	**321**
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	**0**
	14	25	22	14	20	23	14	6	14	17	7	14	14	14	14	10	16	20	9	22	14	**313**
17	0	8	0	0	0	0	0	0	0	3	0	0	0	0	0	0	5	0	0	20	0	**31**
	14	25	22	14	20	23	14	6	14	17	7	14	14	14	14	22	10	20	9	22	14	**319**
18	0	1	0	0	0	0	0	5	0	0	0	2	0	0	0	0	1	5	0	2	0	**11**
	14	25	22	14	20	23	14	6	14	17	7	14	14	14	14	22	16	9	9	22	14	**315**
19	0	0	22	14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	**36**
	14	25	22	14	20	23	14	6	14	17	7	14	14	14	14	22	16	20	5	22	14	**326**
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	**0**
	14	25	22	14	20	23	14	6	14	17	7	14	14	14	14	22	16	20	9	11	14	**313**
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9	**0**
	14	25	22	14	20	23	14	6	14	17	7	14	14	14	14	22	16	20	9	22	10	**321**

Table 9 represents the FRR characteristics of the algorithm, as well as the FAR measure due to random forgeries only. From the shaded diagonal, the FRR is 27.05%[31], and from the last column the value of FAR based on random forgeries is 2.01%[32]. In order to include the effects of skilled forgeries too in the FAR, the results in Table 10 need to be included. The first row is the number of false acceptances and the second row the total number of skilled forgeries submitted.

Table 10 System evaluation results based on standard parameter settings (part 2)

	Subj. 1	Subj. 2	Subj. 3	Subj. 4	Subj. 5	Subj. 6	Subj. 7	Subj. 8	Subj. 9	Subj. 10
FA	0	0	0	7	0	0	0	0	7	0
Total	8	8	8	8	8	6	8	28	8	8

From 98 skilled forgeries, 14 false accepts were encountered, which implies a FAR due to skilled forgeries of 14.3%. This, in conjunction with the FAR quoted for random forgeries, produced a composite FAR of 2.19%. Note that both contributions of FAR are due to category 3 signatures.

Due to the moderate size of the signature database however, the FRR and FAR values presented provide only a moderately refined view of the algorithm, and in the absence of a greater pool of signatures, a correct understanding and interpretation of the results would in fact be more crucial than having the results themselves. Consequently the results are analysed below to draw information from the FAR and FRR evaluation data.

First the cause of the relatively high FRR was investigated. This was done by scrutinising both the test parameter values as well as the various dynamic profiles of the rejected genuine signatures using SysDSV. Three main mechanisms of false rejection were identified.

The first such mechanism is associated with the influence of the number of pen lifts. Consider the set of genuine signatures below. The set on the left (a) are the reference signatures from which a template was created, the set on the right (b) are the test signatures which are verified using the template based on set (a). In the template the mean number of strokes is 2.5 with a standard deviation of 0.5774 strokes, yet the genuine signature set on the right has between 3 to 5 pen lifts as highlighted by the circles.

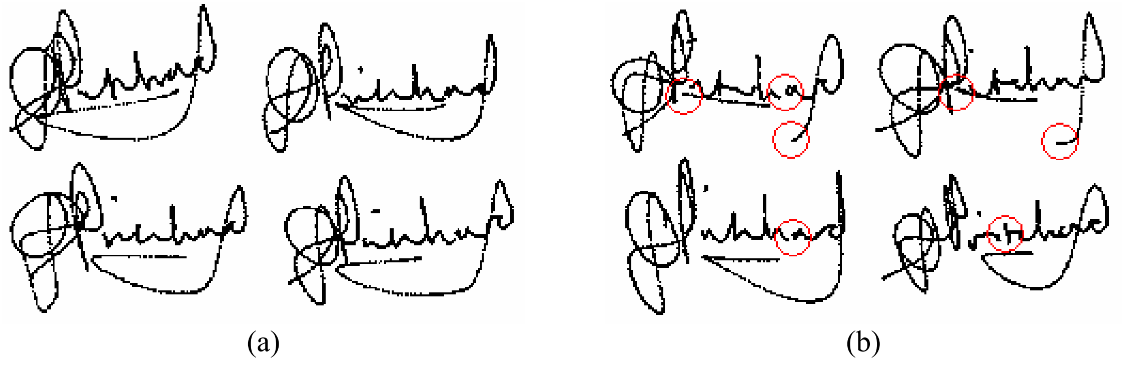


Figure 68 Inconsistent capture of lightly written strokes

[31] This was obtained by dividing the sum of all falsely rejected genuine signatures by the total number of genuine signatures tested, both of which can be calculated by observing the values along the diagonal.

[32] This was obtained by dividing the sum of all falsely accepted forgeries by the total number of random forgeries tested, both of which are shown in the last column of the table.

Such inconsistencies are almost always a result of the limitations of the hardware used. Due to the fact that the x[n] and y[n] co-ordinates are captured using a touch-pad type surface, a low but sufficient level of pressure needs to be exerted in order for the pen-tip movement to be sensed. This may however be a problem with lightly written strokes as shown in Figure 68, where a light stroke is sometimes sensed as a continuous stroke and at other times sensed as a pen lift. This problem however is not attributed to the algorithm, and in practical implementations where radio-based writing tablets are used, such disturbances would disappear altogether.

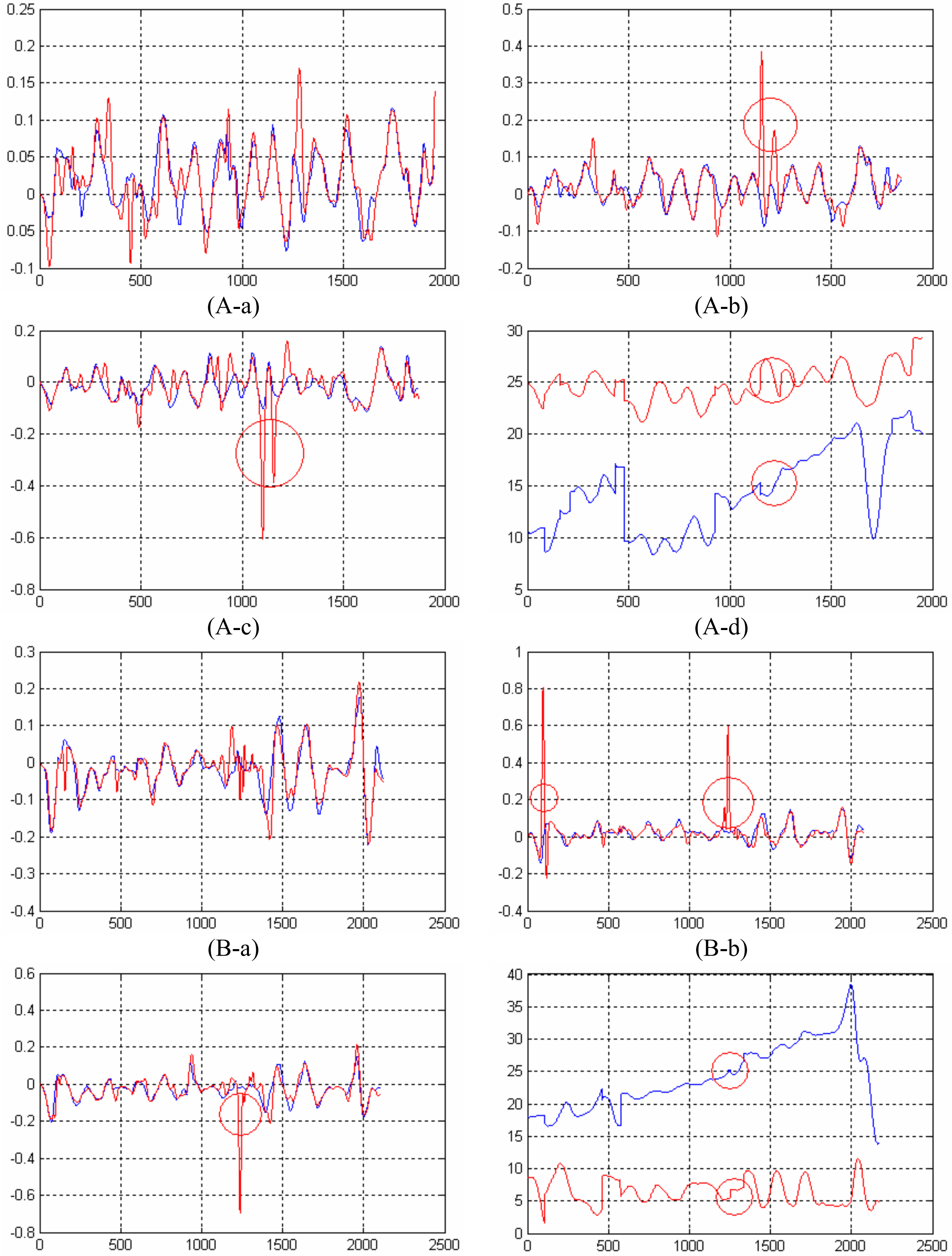
Nevertheless, by recalling that the comparison of the number of pen lifts constitutes the first phase of the classifier – in this case a variation of only '1.5' strokes is allowed, the genuine signature specimens with 5 strokes would be rejected as a forgery. An indirect effect of this is that the pen up-down ratio would also be affected by the length of strokes that are not sensed, and this contributes towards a lower overall classifier score even if the genuine signature were not rejected in phase one of the classifier. By inspection of the test parameters and template values this disturbance accounted for, either fully or partially, about 18 of the false rejections encountered, which means that the true FRR is in the region of 18% instead of 27.05%.

Another mechanism that contributed towards a higher FRR was the derivation of models across points of discontinuities in x[n] and y[n], for which no overt problems were observed initially. For illustration, four signal pairs are given in Figure 69 for genuine signatures sets belonging to two subjects A and B. Subplot (a) in each case is the normal and most frequently encountered behaviour of the $v_y[n]$ and $\hat{v}_y[n]$ signals, in (b) and (c) are shown the occurrences and local instabilities, while (d) contains the x[n] and y[n] signal pair.

It is evident that the occurrences of local instabilities coincide with the presence of discontinuities in x[n] and y[n]. To link these discontinuities to local unstable behaviour, realise that across the discontinuity, the correlation behaviour instantaneously goes from one set of dynamics to another set of dynamics. This has the implication of infinite bandwidth, which is impossible for a physical system and thus cannot be modelled as a physical plant model. In fact, models derived across discontinuities were often only conditionally stable, where a slight change in the inputs would immediately result in unstable outputs.

In view of the above, a more robust approach would be to detect all points of discontinuity within a given x[n] and y[n] pair, and develop system models only on signal segments between two pen lifts. The drawback of this approach is however that since the segment length between two consecutive pen lifts are highly variable, no standard segmentation length can be adopted. The implication hereof is that number of models, model order and prediction horizon would vary from segment to segment, and would need to be automatically determined in a robust way, which would necessitate the use of graphs presented in Section 6.6.

Overall, the local instabilities contributed either fully or partially towards the false rejection of about 20 genuine signatures.



(A-a) (A-b)

(A-c) (A-d)

(B-a) (B-b)

(B-c) (B-d)

Figure 69 Discontinuity related local instabilities

The final major cause of false rejections was not attributable to either the hardware or the algorithm used, but rather to gross inconsistencies in the sign time of the user. One particular user's sign time for example, varied by as much as more than 5 standard deviations or about 1 second, which is significant in the context of signatures. As a result, numerous genuine signatures were penalised by the vast differences in sign times to the template, resulting in a unduly high FRR – this signer is thus a 'goat' of the system. As will be seen, the result of the same signer (subject 12 in Table 9) using the commercial product also resulted in a higher percentage of FRR.

As a result of the evaluation process, a number of additional key findings of interest were also made. For example, since the time duration of signatures can differ significantly, the number of models in the template of each genuine signature would differ accordingly i.e. quick signatures require only few models while long signatures need many more models. Consequently, whenever signatures of relatively short duration were applied as random forgeries to templates with a large number of models, the results were emphatic rejects even without employing the classifier. This is because in order to accommodating all the models within the template, the short signature signal would have to be divided into very short segments, on which no prediction can be performed as their lengths became comparable to the prediction horizon used.

Furthermore, the values of FAR corresponding to random and skilled forgeries seems to indicate that the similarity distance between a genuine signature and its random forgeries can in fact be coincidently very much closer than a skilled forgery. This means that although the dynamic profile of a genuine signature and a skilled forgery is more than likely to be considerably different, the dynamic traits of unrelated signatures may in fact be quite similar to each other. What this suggests is that the coincidence of similarity between unrelated genuine signatures needs to be reduced, the logical implication of which is that more shape related parameters would be required.

This would also help towards the false acceptances suffered with skilled forgeries of subjects 4 and 9 in Table 10, which were in fact category 3 signatures as noted.

The next part of systems evaluation is focused on the influence of the classifier parameters on the evaluation output, specifically the tolerances allowed for the individual parameters and the threshold used between what is regarded as a genuine or a forged signature.

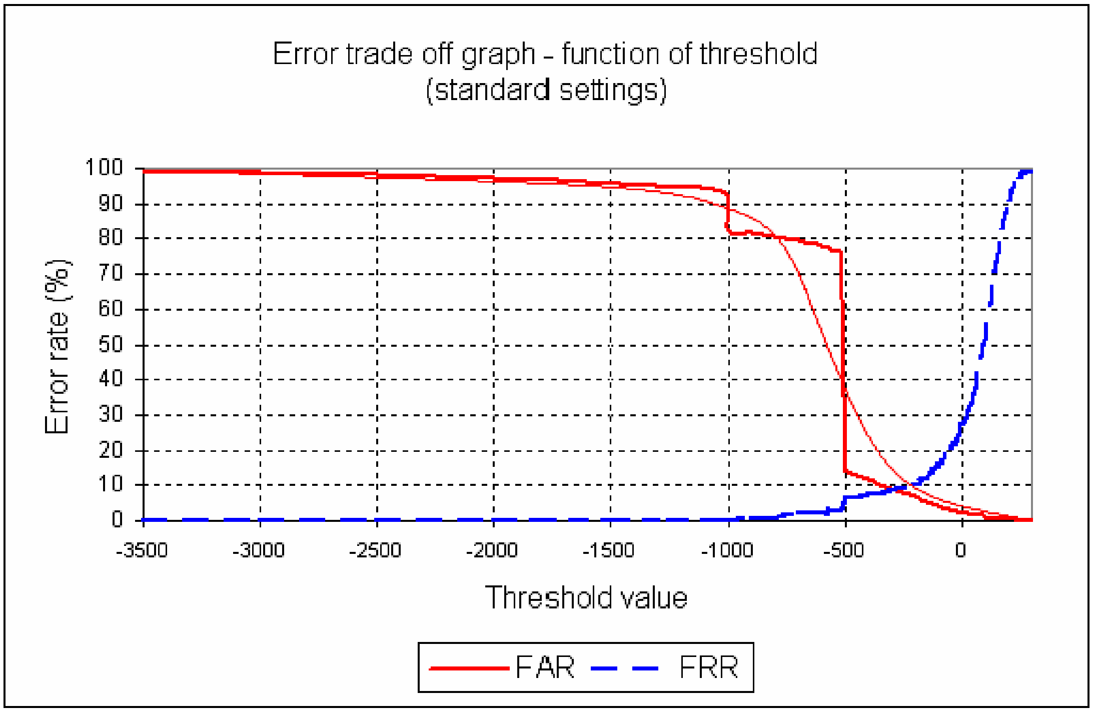


Figure 70 Error trade off graph

The error trade off graph[33] above, from which a number of striking features can be observed, was plotted by varying the threshold value used and noting the FAR and FRR value corresponding to each threshold value. The threshold is the value across which classifier output values would adopt different meanings i.e. genuine or forgery. None of the parameters within the classifier itself were varied.

Firstly, as expected from the discussion of Section 3.2, FAR is a steadily decreasing function of the threshold values and vice versa for the FRR which exhibited behaviour consistent with the typical smooth behaviour of an FRR graph. Secondly, two points of discontinuity can be observed, one at threshold = -500 and another at threshold = -1000, these knick points are in fact the result of two specific score assignments by the classifier.

In the one instance when signatures failed the skilled forgery candidacy test, an arbitrary negative value of –500 was assigned to indicate the occurrence. Also, recall that if the duration of a random forgery submitted is significantly different from the genuine signature the template represents, a prediction may not be possible due to the overly short segment lengths – in such cases another arbitrary mark of –1000 would be assigned. Without such actions, the FAR and FRR profiles would indeed be smooth, as indicated by the thin line in Figure 70 for example.

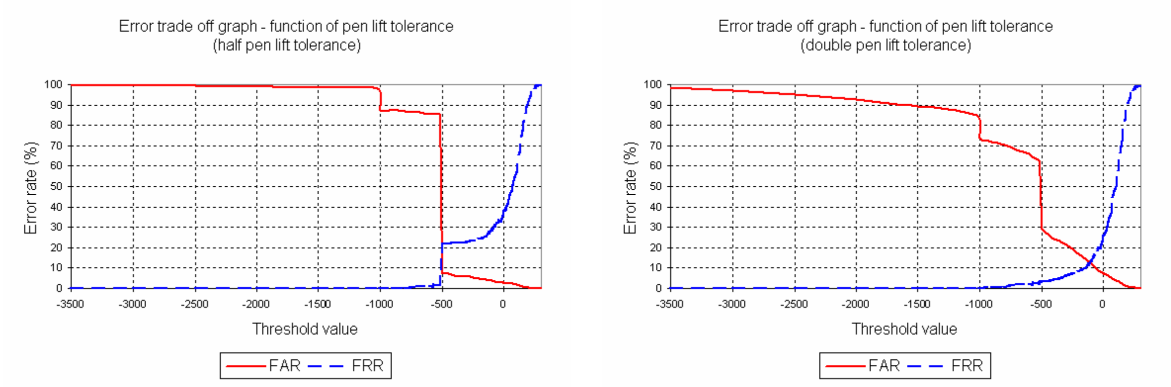
To relate the mark assignment to the knick points in the FAR profile however, consider the forgeries that are assigned –500 or –1000. Such assignment means that the distribution of scores due to forgeries are artificially increased at –500 and –1000 with a corresponding decrease in other regions. This would in turn give rise to the steep changes around the points –500 and –1000. It is also useful to view the trade off graph from a statistics point of view, where the FRR and FAR graphs are essentially probability density functions.

[33] Refer to Section 3.2 for an explanation of error trade off graphs.

The third major feature of the error trade off graph is that the threshold extends much further to the negative direction than it does in the positive direction. This can be appreciated by recalling that the maximum positive scores for most of the parameters are clamped, while the negative extremity aren't. It also suggests that there are significantly more 'bad' signatures relative to a template than there are good ones given an arbitrary pool of signatures. This makes logical sense, and is thus an indication of the strength of the algorithm.

Finally, the equal error rate was situated at a threshold value of approximately –280, where both the FRR and FAR are at 9%, which is relatively high compared to the results generally obtained in literature. However, remember that there are significant disturbance contributions from the limitations in hardware in capturing lightly written strokes, as well as the effects of local instabilities across points of discontinuities in x[n] and y[n]. The implication hereof is that the high FRR is not an inherent characteristic of a systems approach to DSV algorithms, and the true equal error rate is more likely to be in the region of 2% – 3%.

The above error trade off graph corresponds to a pure sweep of the threshold value, with no bias given to the tolerance levels of specific parameters. In the subsequent paragraphs however, the effect of variations in the stringency regarding particular parameters on the classification outcome is investigated.



(a) (b)

Figure 71 Error trade off graph – function of pen lift tolerance

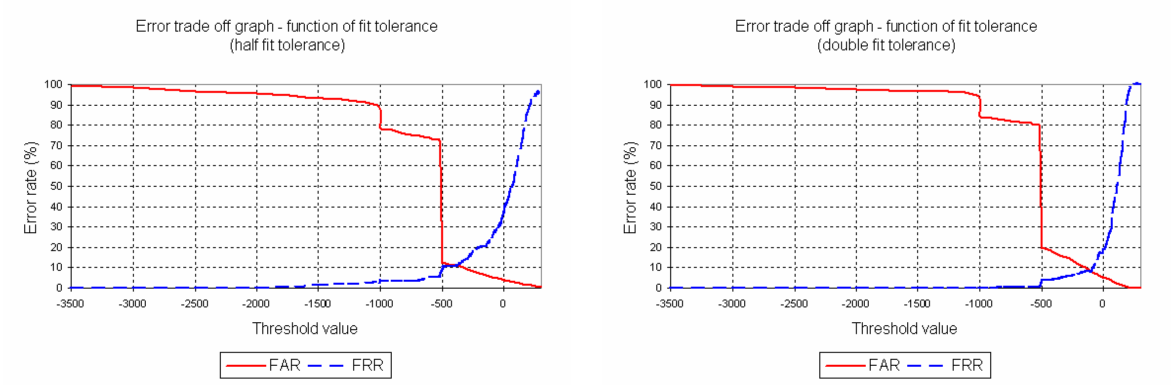
Part (a) of the figure above corresponds to the error trade off curve when the standard tolerance allowed for the number of pen lifts is halved, whilst the curves of part (b) corresponds to when the same tolerance is doubled. Just as expected, when the difference allowed in the number of pen lifts is reduced to essentially 0, the distribution of scores at –500 is further increased, with a corresponding decrease in the level of false acceptances to the right of the –500 point.

But the FRR is adversely affected more seriously, where the level of false rejections to the right of the –500 point in Figure 71 rose from 7% to 21%, which would be a practically unacceptable situation. Consequently, it is clear than a reduction in the tolerance of number of pen lifts is not an option in improving the performance of the algorithm.

Furthermore, the equal error rate of part (a) does not in fact exist in the true sense, since the equal error rate is associated strictly with the intersection of FRR and FAR curves at a *smooth* and *continuous* point, which in this case is obviously false. In fact the 'true' equal error point is buried to the left of the cliff at –500.

Likewise, an increase in the tolerance level in this regard would be equally unacceptable. This is because although the FRR is only slightly reduced at threshold = 0, the FAR has been increased to approximately 8% from 2% at the same point i.e. no real improvement in the algorithm's performance is reaped. Again, the tolerance of the number of pen lifts should be untampered with at all times.

Finally, note that the gradient of the FRR curve in Figure 71 levels out in the region –500 to 0. This is testimony to the fact that a major portion of false rejections can be attributed to the difference in the number of pen lifts, or more directly the inadequacies of the hardware used. The gentle gradient in the region suggests that signatures are not being rejected because of a low overall score, but rather due to the fact that they are unduly assigned the score of –500 due to the difference in the number of pen lifts.

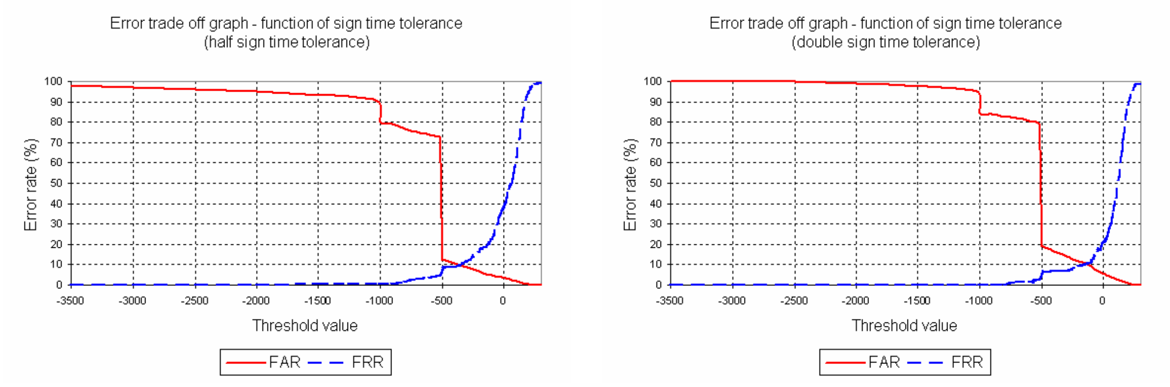


(a) (b)

Figure 72 Error trade off graph – function of fit tolerance

As before, subplot (a) of Figure 72 corresponds to the results with the fit tolerance halved, and subplot (b) with the tolerance doubled. An interesting observation here is that unlike the case with changes in the tolerance of the number of pen lifts (which affected the 'height' of the cliffs at –500 and –1000), the height of the cliffs in the graphs above remain almost identical to those shown in Figure 70. This means that the signatures that failed the skilled forgery candidacy test, and those that are 'too' different from the template in duration are still being treated as such. Therefore, the changes in the FAR and FRR curves seen above are purely a result of 'similar' random forgeries, good skilled forgeries and bad genuine signatures.

Apart from the above, no extraordinary observations can otherwise be made, apart to say that in subplot (a) the FRR is 37.20%, and in subplot (b) it is shifted to 18.38% – which is expected behaviour. This does seem to suggest that a relaxed fit tolerance may be beneficial seeing that the FRR is significantly reduced at 0 without much increase in FAR (5.05% at 0 in subplot (b)).



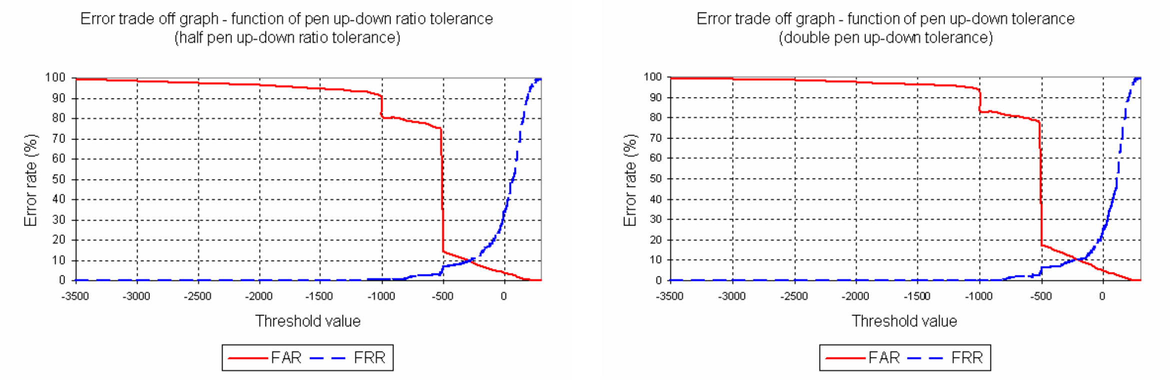
(a) (b)

Figure 73 Error trade off graph – function of sign time tolerance

The graph on the left of Figure 73 represents results obtained when the standard tolerance of sign time was halved, and the graph on the right corresponds to the standard tolerance being doubled. As is the case with changes in the tolerance of the fit value, the heights of cliffs are again unaltered by the changes in the tolerance of sign time. The FAR and FRR for half sign time tolerance at zero were at similar values to those seen in both subplots (a) and (b) of Figure 72.

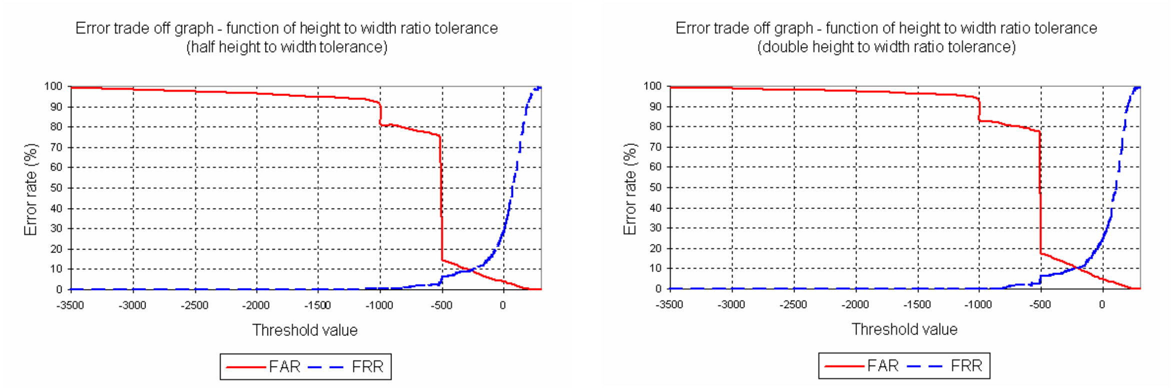
The above seems to suggest that the classifier response to the tuning of either the fit value tolerance or the sign time tolerance is similar, specifically with regard to its sensitivity to each unit of variation in tolerance introduced to the individual parameters.

A trend that is emerging from the graphs studied thus far, is that the equal error rate, whenever visible, remains in the region of 9% regardless of the specific tolerance changing and the amount by which it is changed. This was an unknown quality prior to the evaluation process, and may very well be the reason why the equal error rate is often quoted as a singular measure of an algorithm's performance.



(a) (b)

Figure 74 Error trade off graph – function of pen up-down ratio tolerance



(a) (b)

Figure 75 Error trade off graph – function of height to width ratio tolerance

The tolerances associated with the remaining two parameters of pen up-down ratio and the height to width ratio were also varied and their effects on the FRR and FAR curves were observed. Although slight changes in the FRR and FAR values were indeed observed, the changes are clearly not as pronounced as those observed for the prior three parameters. This is understandable since the weighting constants of the final two parameters are significantly smaller in magnitude than those for the former three. Consequently the sensitivity of the classifier to tolerances changes was seen as a function of the weighting constants.

The impression of the algorithms obtained form the above evaluation analysis is that the algorithm's FAR measure is competitive, while it seems to be slightly over stringent towards genuine signatures. As mentioned previously, the need for the use of more complementary parameters is further emphasised here, so that if the FAR is reduced by tolerance tuning the FRR can still be kept as low as possible.

8.3. Benchmarking

Since most of the information available in literature is based on academically researched algorithms, the performance of the proposed algorithm was evaluated against that of a commercial DSV product.

The specific product used for this purpose was the SignDoc version 0.1.23 software produced by SOFTPRO GmbH & Co., KG of Germany, who had permitted the use of the demo version of the software for the current research. The rationale hereof is that since SignDoc is a commercially available product that has been deployed in more than 200 business and financial institutions worldwide, the product must have a significantly high level of performance considering its commercial applications.

Specifically, the SignDoc© software is available to be installed as an integrated part of Microsoft® Word 2000 package which, through the use of suitable digital capturing devices, allows legally binding handwritten signatures to be embedded into electronic *.doc documents. This may be a result of an intent to authorise, approve or to gain exclusive editing right of the contents of a *.doc document.

Whatever the case, a signature reference or template is first created in each instance by the capturing of 3 reference signatures, followed by the insertion of a visual image of the written signature into the document. The template, which contains both static (or visual) and dynamic information about a signature, is embedded as a part of the *.doc document, so that it can be used in subsequent attempts to verify a test signature or unlock a secure document. In all transactions, encryption and time stamping is also applied to ensure security – these features are however non-essential for evaluating the proposed algorithm, which is not yet productised.

During signature collection, 18 of the 21 subjects whose signatures were used in the evaluation results presented earlier were asked to sign individual *.doc documents so that firstly they create a template in the document, and subsequently they verify themselves against their own templates to assess the FRR. The FAR was evaluated by using both skilled and random forgeries verified against the template of various subjects – the composite FAR evaluated to 0%.

Attention should also be drawn to the fact that under 'non-standard' conditions, both high FAR and FRR values have been encountered with SignDoc. Specifically, the signature of 2 subjects experienced FRR's as high as 86% and that of yet another subject had almost 100% false acceptance. These were however not included in the evaluation results shown in Table 11, since the same number of subjects exhibited the 'normal' range of FRR and FAR values when the test was repeated. Nevertheless such behaviour shows the universal vulnerability of DSV algorithms to external and instantaneous factors, such as posture, mental state, fatigue etc.

Finally, the conditions surrounding signature capturing were identical to those adopted for the results of Table 9, with the only difference being the capturing equipment used. For SignDoc, the equipment used was the WACOM ET-0405A-U radio tablet, which senses both the position of the pen tip as well as the normal pen pressure exerted during writing – it is understood that SignDoc does in fact exploit the pressure profile for verification. Due to the radio link between the stylus and the writing surface, the equipment had no problems sensing softly written strokes. However due to the fact that the stylus associated with the WACOM tablet was not an inking stylus by nature, the visual feedback came not from the writing surface but via a SignDoc graphical user interface, which provided a view of the signature signed in real time.

The FAR results are shown in Table 11, where each row represents a new subject and each column is a genuine signature submitted with Y indicating correct verification and N indicating false rejection. The FRR evaluated from the result is 9.3%. Of specific interest is that subjects 10 and 17 in Table 11 are subjects 12 and 20 in Table 9, which suggests that both DSV algorithms suffered unduly high FRR due to the inferior consistency originating from the signers – the system goats.

Another interesting observation is that a FAR of 0% and a FRR of 9.3% also seems to suggest that the algorithm is tuned towards slight over-stringency towards genuine signatures, as was also noted for the algorithm developed in the research. Essentially, such settings constitute a compromise between ensured security and slightly increased user inconvenience. After all it is only the inconsistent signers that would be inconvenienced, which accounts for a only small fraction of signers at large.

Table 11 System evaluation results of SignDoc version 0.1.23 software

	1	2	3	4	5	6	7	8	9	10	11	12	13	**Tot.**
Subject 1	Y	Y	Y	Y	Y	Y	Y	N	Y	Y	-	-	-	**1/10**
Subject 2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	-	-	-	**0/10**
Subject 3	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	-	-	-	**0/10**
Subject 4	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	-	-	-	**1/10**
Subject 5	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	-	-	-	**0/10**
Subject 6	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	-	-	-	**0/10**
Subject 7	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	-	-	-	**0/10**
Subject 8	Y	Y	Y	Y	Y	Y	N	Y	Y	Y	N	N	Y	**3/13**
Subject 9	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	-	-	-	**0/10**
Subject 10	Y	N	N	Y	N	N	N	Y	Y	Y	-	-	-	**5/10**
Subject 11	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	-	-	-	**0/10**
Subject 12	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	-	-	-	**0/10**
Subject 13	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	-	-	-	**0/10**
Subject 14	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	-	-	-	**0/10**
Subject 15	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	-	-	-	**0/10**
Subject 16	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	-	-	-	**1/10**
Subject 17	N	Y	Y	N	Y	N	N	N	Y	N	-	-	-	**6/10**
Subject 18	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	-	-	-	**0/10**

Unfortunately the error trade off graph for SignDoc was not accessible to the research, due to the commercial nature of the solution. Nevertheless, the FAR and FRR values alone are very in encouraging, since a FRR below 10% e.g. 4% or 5% is believed to be readily achievable by adopting the recommendations set forth prior to the analysis of the evaluation results of Table 9. Consequently, a systems approach based DSV can in fact be competitive as a commercial product, which after all is the ultimate aim of the research.

9. Conclusion

The idea of taking a systems and control approach to the problem of dynamic signature verification was explored, and the study conducted constitutes one of the very few, if any, works that focuses on the application of systems tools and concepts to DSV.

In essence the entire research was based on the combination of 2/3 power law and a proposed systems model of the handwriting process. One of the most important findings was a result of an analogy drawn between learning theory and controller synthesis, where it was seen that the idiosyncratic nature of handwriting habits could be attributed to the differences in the set of control strategies employed by each person for handwriting.

In addition, experimental results confirmed the applicability of the 2/3 power law to handwriting, and reveal that the law is obeyed by all individuals in the same manner with a universal k(v) profile. This, in conjunction with the fact that the 2/3 power law is obeyed for the case of velocity as a function of curvature of movement *and* vice versa, means that given any two-dimensional image of handwriting, the original pen-tip velocity profile can be completely and exactly recovered. From these findings it was realised that the underlying security in DSV technology is based on the fact that each individual has a unique set of motor coordination control strategies, which is the reason why a successful forgery must be written with ballistic motion.

In the implementation of a systems based DSV algorithm, the x and y coordinate profiles of the pen-tip were adopted as the algorithm inputs, since the coupling relationship between the x and y components of pen-tip velocity was seen to constitute an important invariant in handwritten signatures. Acknowledging that this coupling relationship is likely to be non-linear, signal segmentation was adopted to allow for localised linear approximation using ARX models.

In discovering the set of parameters that would result in higher discriminancy, experiments were conducted to investigate the effects of individual parameters on the accuracy and discriminancy of signature templates. Specifically, the eventual algorithm used ARX models of 20th order, segment lengths of 60 samples, half overlap ratio of 20% and a prediction horizon of 20 samples, the combination of which resulted in an optimised level of discriminancy.

In terms of a strategic view towards the implementation of a more robust algorithm, three categories of signatures were identified from behavioural patterns observed during experimentation, namely, ballistic, written and sinusoidal categories. Recognising that different verification strategies would be needed for each signature category, five complementary parameters were used to improve the verification results. Specifically, the parameters are the number of turning points in the predicted velocity profile, total sign time, the pen up-down ratio, the number of pen lifts and the signature height-to-width ratio.

During the design of a classifier it was realised that methods capable of handling fuzzy inputs would be needed for effective classification. Consequently, a hybrid classifier encompassing a skilled forgery candidacy test as well as a simple linear classifier was proposed. This classifier examines the distance between the individual parameters and their mean or preferred value. This rudimentary classifier did in fact yield satisfactory and promising results, and the use of a neural network as a classifier stands as a topic worthy of future research.

Evaluation of algorithm performance in terms of FRR and FAR indicated that characteristic verification outcomes could be observed for the three categories of signatures, and that the crude values of FRR and FAR were 27.05% and 2.19% respectively, while a constant equal error rate of 9% was also observed. These results included the effects of inconsistent hardware capturing of lightly produced strokes. In the absence of this problem the FRR is believed to be around 18% or lower. By further incorporating the effects of knick points in the velocity profile (which can be circumvented by implementation of dynamic segment length allocation), it is believed that the FRR can be reduced to values much lower than 10%.

In benchmarking the prototype algorithm, FRR and FAR values of 9.3% and 0% respectively were estimated for the commercially available SignDoc software. However much higher error rates (some in the region between 80% and 100%) were also observed with SignDoc under non-standard conditions. It is therefore believed that with further research, the performance of the prototype algorithm will be comparable, if not superior, to SignDoc, which proves that a systems approach can indeed be an effective alternative to dynamic signature verification.

To summarise, the value of this research is firstly in its validation of the idea that a systems and control approach to DSV can work. Secondly, by tying together previously loosely related fields of science, a plausible explanation and understanding was acquired with regards to the origin of idiosyncratic writing habits and the underlying security in DSV technology. Finally, the specific areas of possible future research are the implementation of dynamic segment length allocation, the use of neural networks as a classifier, and the incorporation of additional complementary parameters.

Appendix 1: Handwriting Capturing Device Summary

The hardware equipment used consists of two parts, the pressure sensitive surface and the pressure-sensing pen, which are shown together in the photo below with a wooden hand rest. This hardware set was designed and developed by the author for an undergraduate project [38].

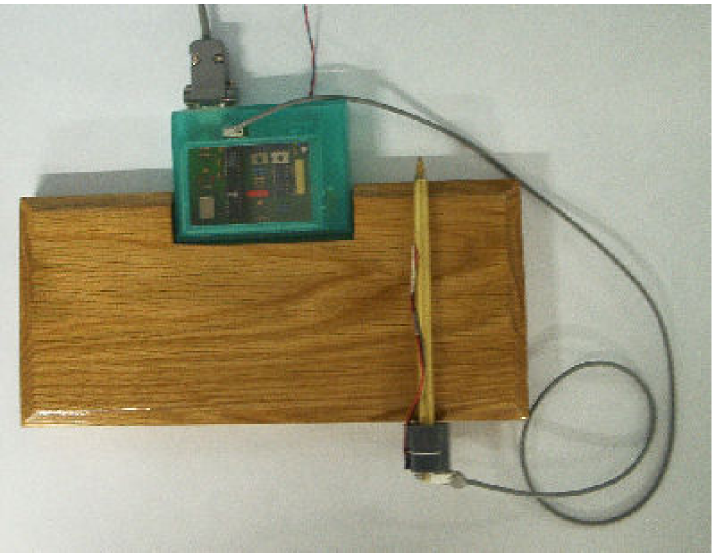


Figure 76 Signature capturing hardware

Housed in the green box is the transparent pressure sensitive surface (Figure 77) for capturing the x and y co-ordinates of the pen-tip motion, below which rests the control board that co-ordinates the various tasks of the hardware device. The pressure-sensing pen shown senses minute axial displacements of the pen tip using a stiff spring coupled with an infrared diode and phototransistor pair. The pressure signal is transmitted via the soft wire to the control board where it is sampled.

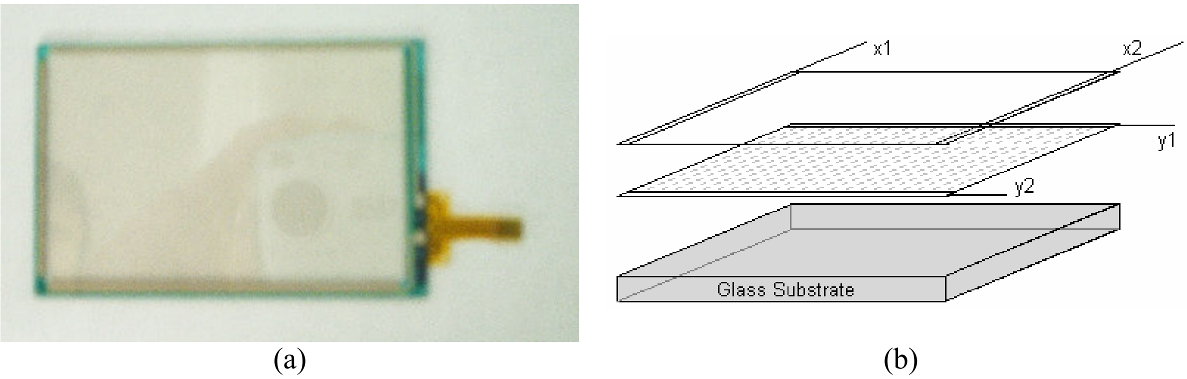


Figure 77 AMT 9503 4-wire resistive touch screen

Figure 77 depicts an actual image (a) and a cross-sectional view (b) of the AMT 9503 4-wire resistive touch screen. The rectangular sensing surface is placed on a glass substrate for support, and consists of 2 layers of transparent conductors, which have a matrix of finely spaced conducting metallic dots between them. The 2 layers of conductors are separated by a few microns, and are not in contact with each other in the relaxed state. On each layer, there are two conductive tracks running along opposite edges as indicated by x1, x2, y1 and y2 in the diagram.

By assuming that the top left corner of the touch surface has co-ordinates (0, 0), and the bottom right corner has co-ordinate (255, 255), the x co-ordinate of a touch is determined by setting up a voltage across the top surface where x1 = 0 V and x2 = 5 V (to be consistent with TTL voltage levels), while the voltage at y1 and y2 is undefined. When the screen is touched at a point, the pressure applied causes the top surface to make contact with the metal dots on the bottom surface, thereby giving the entire bottom surface a potential that is equal to the potential at the point of contact on the upper layer, so that one in effect have the top surface acting as an potentiometer, where a voltage reading can be made at either y1 or y2.

Since the closer the point of pen-tip contact is to x2, the closer the reading would be to 5 V, and the closer the point of pen-tip contact is to x1, the closer the reading would be to 0 V. The position of a touch can thus be deduced from the voltage value present on the bottom surface. The same is applicable when the y co-ordinate is to be determined, except that use of the top and bottom surfaces are interchanged.

A summary of the part specification of AMT 9503 is given in Table 12.

Table 12 Specification summary of AMT 9503

	Value in x direction	**Value in y direction**
Physical attributes		
Writing area dimension	73 mm	46 mm
Surface Resistance	859 ohm	211 ohm
Surface-ground capacitance	9.3 nF	37.9 nF
Rise time	25 us	28 us
Distance discretisation	0.286 mm / LSB_{10}	0.180 mm / LSB_{10}
Resolution	89 dpi	141 dpi
Transducer attributes		
Input range	0 mm – 73 mm	0 mm – 46 mm
Input span	73 mm	46 mm
Output range	0.281 V – 5 V	0.533 V – 5 V
Output span	4.719 V	4.467 V
Linearity	Full range	Full range
Gain	64.64 V/m	97.1 V/m
Sensitivity	64.64 V/m	97.1 V/m
Hysteresis	None	None

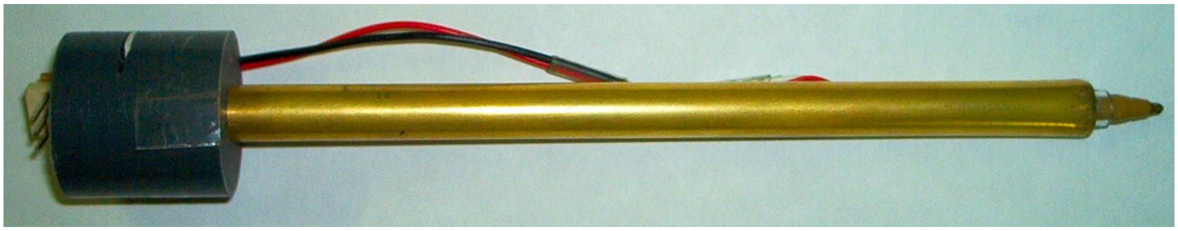


Figure 78 Photograph of the old optical pressure-sensing pen

The other input device in the system is the pressure-sensing pen shown in Figure 78. A slightly modified normal writing pen is placed inside a metal tube in line with a stiff spring, across which the PT381F Darlington pair phototransistor with a visible light cut off filter and the GL380 infrared emitter is placed. As a writer exerts variable levels of force along the pen's axis during writing, the stiff spring is displaced by small amounts so that the separation between the IR diode and phototransistor is varied accordingly thereby causing variations in the output voltage of the phototransistor.

The output voltage of the phototransistor is fed to an analogue to digital converter (ADC) via a soft wire, and in this way the axial writing pressure of a piece of writing is captured. A summary of the devices measurement specifications is given below.

Table 13 Specification summary of pressure-sensing pen

Transducer attributes Value	
Input range	0 Pa – 1333.33 Pa
Input span	1333.33 Pa
Output range	2.93 V – 3.15 V
Output span	0.22 V
Linearity	Limited range
Linear gain	165×10^{-6} V/Pa
Linear constant offset	2.93 V
Maximum non-linearity	28.75 mV
Hysteresis	None

Due to the fact that the pressure-sensing pen shown in Figure 76 was constructed from relatively crude components and was consequently relatively awkward to write with, a new prototype of the pen (refer to Figure 79 and Figure 80) was designed and machined during the course of the research. The new pen, which carries the same behavioural specifications as set out in Table 13, was constructed entirely out of aluminium for lightweight, and also ensured precision movements of the pen in warranting measurement accuracy.

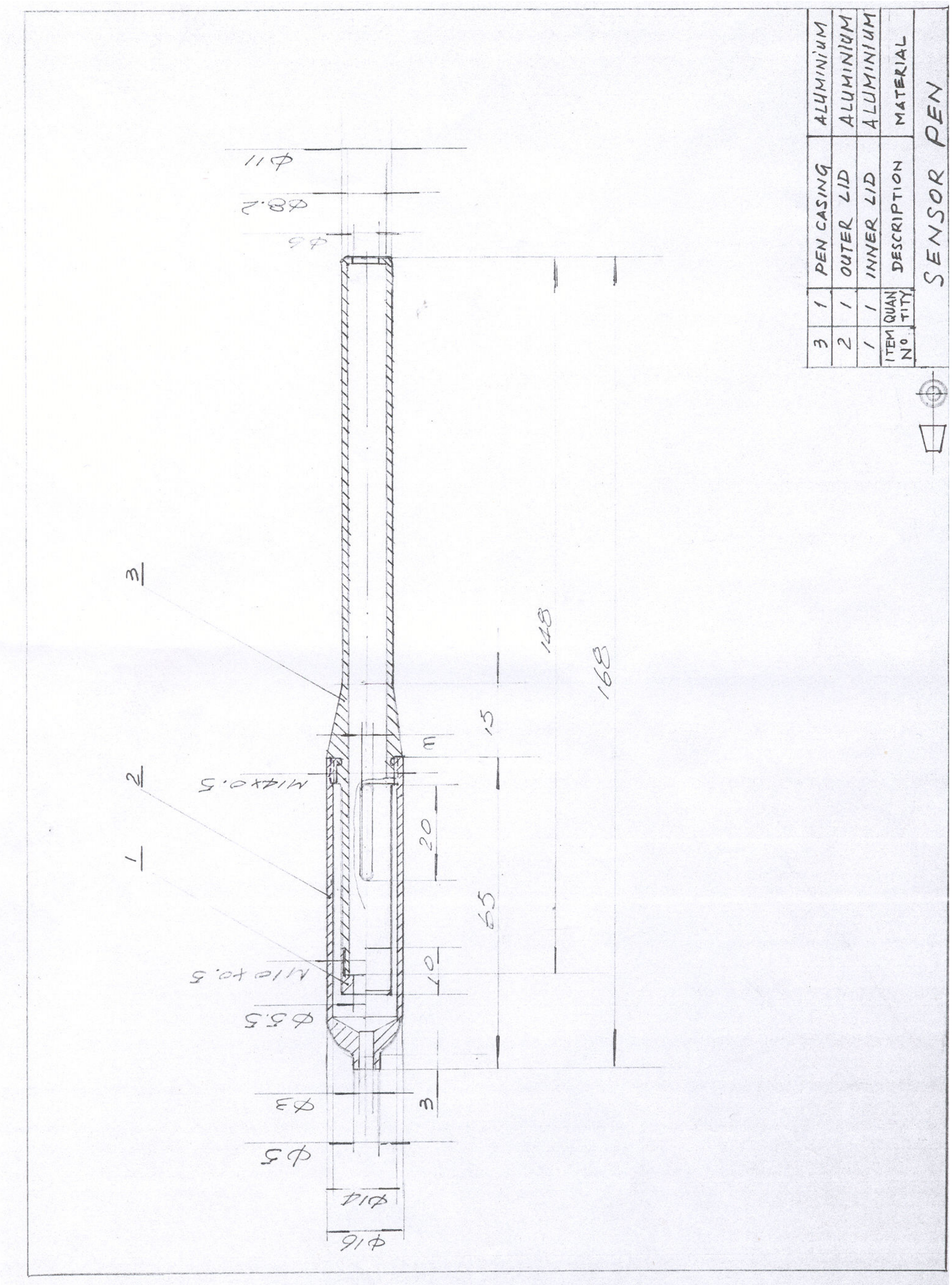


Figure 79 Design of the optical pressure-sensing pen

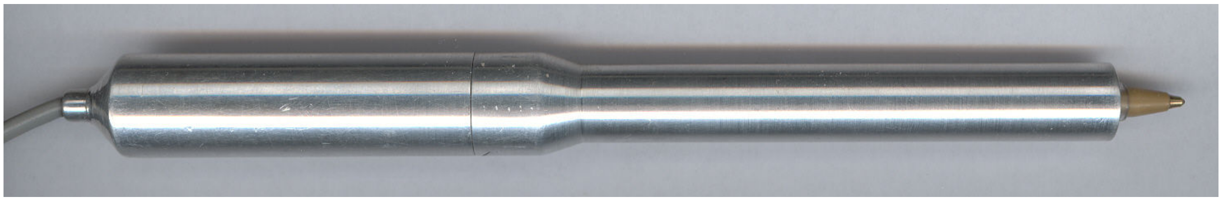


Figure 80 Photograph of the new optical pressure-sensing pen

Finally, the input signals captured using the devices described above needs to be processed, sampled and sent to a host computer for analysis; at the same time a voltage needs to be applied to the touch pad in a specific sequence to determine the x and y co-ordinates of the pen tip – to do all this, a control board was needed to co-ordinate the various tasks. The Motorola MC68HC908KX8 microcontroller was chosen to be the basis of such a controller.

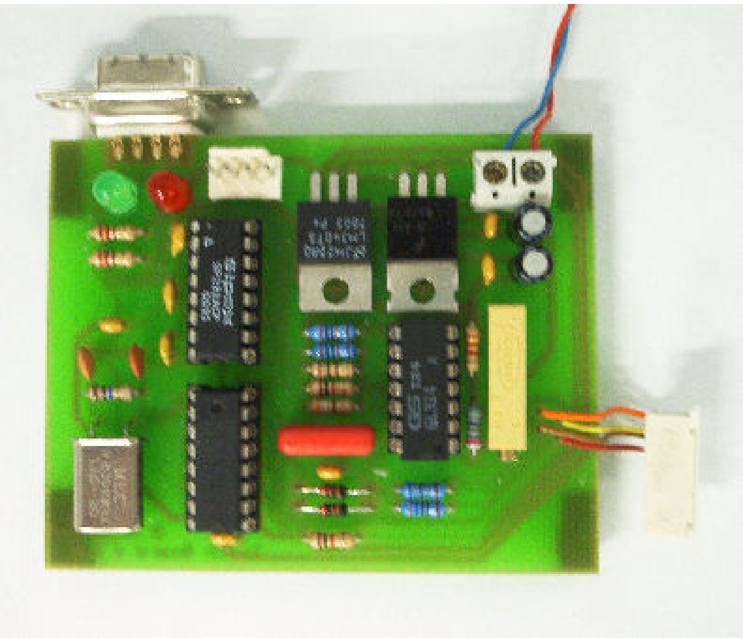


Figure 81 Motorola MC68HC908KX8 microcontroller-based control board

Specifically, 4 pins of the microcontroller are dedicated to the control of the voltage across the two surfaces of the touch board, and two of these pins are 8-bit ADC inputs for capturing the x and y co-ordinate values. Once the co-ordinate samples are taken, rudimentary signal checking and corrections are carried out within the microcontroller before being transmitted. For the processing of the pressure signal from the pen, the board also provides dc decoupling and high-pass filtering modules, the output of which is fed into yet another 8-bit ADC input. Each of the x and y co-ordinate values and pressure readings are sampled at 600 Hz

The final task of the control board is the transmission of all the 8-bit x co-ordinate, y co-ordinate and pressure reading samples to a host PC via the 9-pin RS232 serial interface set to 96800 baud, where the voltage shifting from TTL to RS232 standards is implemented by the MAX232 integrated chip.

Appendix 2: Calculation of the Running Radius of Curvature

In the analysis of the 2/3 power law, a MATLAB *.m file was written to calculate the running radius of curvatures in the pen-tip trajectories, the workings of the function is presented in this section.

The method involved the fitting of a uniform arc to the curve formed by three consecutive sample points. Consider sample points (x_{i-2}, y_{i-2}), (x_{i-1}, y_{i-1}) and (x_i, y_i) as shown in Figure 82, although it is quite unlikely that all three points would lie on a perfect arc of uniform radius, a good approximation to the curve can nonetheless be achieved by finding the radius to a segment (shaded in grey) whose area is equal to that of the triangle formed by joining the three consecutive sample points (x_{i-2}, y_{i-2}), (x_{i-1}, y_{i-1}) and (x_i, y_i).

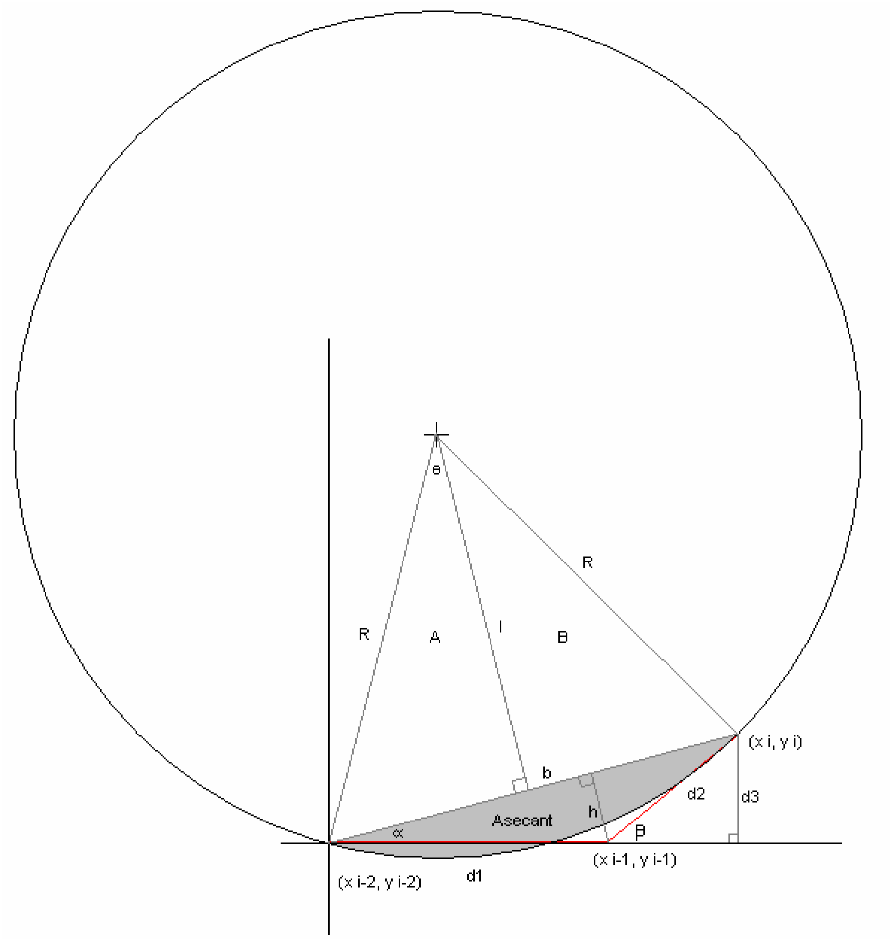


Figure 82 Running radius calculation

Given the co-ordinate of the three points, the area of the triangle with sides d1, d2 and b is to be found with the readily calculable parameters being b, d1 and d2. The entire exercise begins with the calculation of β, which is:

$$\beta = \left| \arctan\left(\frac{y_i - y_{i-1}}{x_i - x_{i-1}}\right) - \arctan\left(\frac{y_{i-1} - y_{i-2}}{x_{i-1} - x_{i-2}}\right) \right|$$

because the line segment d1 could be in any orientation. Now by inspection of the geometry, d3 is deducible from β using:

$$d3/d2 = \sin\beta \quad \Rightarrow \quad d3 = d2\times\sin\beta$$

With knowledge of the value of d3 and b, the angle α can then be calculated using:

$$d3/b = \sin\alpha \quad \Rightarrow \quad \alpha = \arcsin(d3/b)$$

In the triangle formed by d1, h and b, h can be evaluated using:

$$h/d1 = \sin\alpha \quad \Rightarrow \quad h = d1\times\sin\alpha$$

Finally, with knowledge of both b and h, the area of the triangle formed by the sides b, d1 and d2 can be computed using the well-known formula:

$$\text{Area } \Delta = 0.5\times b\times h$$

The next step is the calculation of the radius R of a circle that would result in the grey-shaded segment having the same area as the triangle, but because there is no analytical way of solving for R, the evaluation of R involves an iterative process.

The first step of the procedure is choosing an initial value of R, which is set to equal to half the length of b i.e. b forms the diameter of an initial circle of radius R. Secondly, θ is found using:

$$\theta = \arcsin(0.5\times b/R)$$

The knowledge of R and θ allows the sector area A_{sector} to be calculated:

$$\frac{2\theta}{2\pi} = \frac{A_{\sec tor}}{A_{circle}} \quad \Rightarrow \quad A_{\sec tor} = \frac{2\pi R^2\theta}{2\pi} = R^2\theta$$

If the areas of the two congruent triangles ΔA and ΔB are subtracted from A_{sector}, then the area of the segment is obtained. For calculation of the two triangles' areas the length of *l* is needed, which is calculated using:

$$b/2\div l = \tan\theta \quad \Rightarrow \quad l = b/(2\tan\theta)$$

Finally, the area of the shaded segment is given by:

$$A_{segment} = A_{sector} - 0.5\times l\times b$$

From an initial value of R = b/2, the length of R is gradually increased in 0.001 mm increments until the difference in Area Δ and $A_{segment}$ is within a set tolerance.

Appendix 3: The Least Square Estimation Method by QR Factorisation

Consider an ARX model

$$\mathrm{A}(q)\mathrm{y}(\mathrm{t}) = \mathrm{B}(q)\mathrm{u}(\mathrm{t}) + \mathrm{e}(\mathrm{t})$$

Equation 26 The ARX model structure

where

$$\mathrm{A}(q) = 1 + \mathrm{a}_1 q^{-1} + \cdots + \mathrm{a}_\mathrm{n} q^{-\mathrm{n}}, \qquad \mathrm{B}(q) = \mathrm{b}_1 q^{-1} + \cdots + \mathrm{b}_\mathrm{m} q^{-\mathrm{m}}$$

and e(t) models any disturbance influence, with q being the shift operator and integer values of t. For simplicity of analysis, let us define a regressor vector $\boldsymbol{\varphi}(\mathrm{t})$ and a parameter vector $\boldsymbol{\theta}$ so that the expression can be rewritten as

$$\mathrm{y}(\mathrm{t}) = \boldsymbol{\varphi}(\mathrm{t})^\mathrm{T}\boldsymbol{\theta} + \mathrm{e}(\mathrm{t})$$

Equation 27 An alternative expression for the ARX model structure

where

$$\boldsymbol{\varphi}(\mathrm{t}) = [\mathrm{u}(\mathrm{t}-1), \mathrm{u}(\mathrm{t}-2), \ldots, \mathrm{u}(\mathrm{t}-\mathrm{m}), \mathrm{y}(\mathrm{t}-1), \mathrm{y}(\mathrm{t}-2), \ldots, \mathrm{y}(\mathrm{t}-\mathrm{n})]^\mathrm{T}$$

$$\boldsymbol{\theta} = [\mathrm{b}_1, \mathrm{b}_2, \ldots, \mathrm{b}_\mathrm{m}, \mathrm{a}_1, \mathrm{a}_2, \ldots, \mathrm{a}_\mathrm{n}]^\mathrm{T}$$

The aim of the least square method is to find the parameter vector $\boldsymbol{\theta}$ such that the square of the observation error or noise e(t) in Equation 27 is minimised. In other words the maximum amount of system dynamics is captured by $\boldsymbol{\theta}$ i.e. minimise S where

$$\mathrm{S} = \sum_{\mathrm{t}=1}^{\mathrm{N}} [\mathrm{e}(\mathrm{t})]^2 = \sum_{\mathrm{t}=1}^{\mathrm{N}} \left[\mathrm{y}(\mathrm{t}) - \varphi(\mathrm{t})^\mathrm{T}\boldsymbol{\theta}\right]^2$$

and N is the total number of data samples available. Since S is to be minimised over all N data points, N linear equations must be set up in the form

$$\mathbf{y} = \boldsymbol{\Phi\theta} + \mathbf{e}$$

so that

$$\mathrm{S} = \mathbf{e}^\mathrm{T}\mathbf{e} = (\mathbf{y}^\mathrm{T} - \boldsymbol{\theta}^\mathrm{T}\boldsymbol{\Phi}^\mathrm{T})(\mathbf{y} - \boldsymbol{\Phi\theta})$$

where

$$\mathbf{y} = \begin{bmatrix} \mathrm{y}(1) \\ \mathrm{y}(2) \\ \vdots \\ \mathrm{y}(\mathrm{N}) \end{bmatrix}, \qquad \boldsymbol{\Phi} = \begin{bmatrix} \varphi(1) \\ \varphi(2) \\ \vdots \\ \varphi(\mathrm{N}) \end{bmatrix}, \qquad \mathbf{e} = \begin{bmatrix} \mathrm{e}(1) \\ \mathrm{e}(2) \\ \vdots \\ \mathrm{e}(\mathrm{N}) \end{bmatrix}$$

Now, the to find the value of **θ** that minimises S denoted **θ**$_{\mathbf{min}}$, we need to find the point where the multidimensional gradient of S with respect to **θ** is zero:

$$\frac{\partial \mathrm{S}}{\partial \boldsymbol{\theta}} = \begin{bmatrix} \frac{\partial \mathrm{S}}{\partial \theta_1} & \frac{\partial \mathrm{S}}{\partial \theta_2} & \cdots & \frac{\partial \mathrm{S}}{\partial \theta_\mathrm{p}} \end{bmatrix}^\mathrm{T} = 0$$

Equation 28 The least squares model fitting criterion

where p = n + m. By applying two standard results to the matrix partial derivative given in [34], **θ**$_{\mathbf{min}}$ can be solved by solving for

$$\frac{\partial \mathrm{S}}{\partial \boldsymbol{\theta}} = -2\boldsymbol{\Phi}^\mathrm{T}\mathbf{y} + 2\boldsymbol{\Phi}^\mathrm{T}\boldsymbol{\Phi}\boldsymbol{\theta} \Rightarrow \left[\boldsymbol{\Phi}^\mathrm{T}\boldsymbol{\Phi}\right]\boldsymbol{\theta}_{\mathbf{min}} = \boldsymbol{\Phi}^\mathrm{T}\mathbf{y}$$

Equation 29 Point of minimum least square error

to ensure that **θ**$_{\mathbf{min}}$ gives a true minimum and not a maximum or a saddle point, one must verify whether any small change Δ**θ** about **θ**$_{\mathbf{min}}$ would increase S.

In cases where N = p, a unique solution exists for Equation 29, and the parameter description would constitute an exact LTI model for the observed data. If N > p however, which is almost always the case, then the set of equations becomes an overdetermined problem that can be solved using QR factorisation.

QR factorisation is a very convenient and efficient tool that allows any full rank N × p matrix **A** to be factored in the form **A = QR**, where **Q** is an N × p orthonormal matrix i.e. $\mathbf{Q}^\mathrm{T}\mathbf{Q} = 1$, and R is an p × p upper triangular matrix with positive elements along the diagonal. If we let **Φ** = **QR** using QR factorisation, Equation 29 can then be rewritten as

$$\begin{aligned} \left[\boldsymbol{\Phi}^\mathrm{T}\boldsymbol{\Phi}\right]\boldsymbol{\theta}_{\mathbf{min}} &= \boldsymbol{\Phi}^\mathrm{T}\mathbf{y} \\ \mathbf{R}^\mathrm{T}\mathbf{Q}^\mathrm{T}\mathbf{Q}\mathbf{R}\boldsymbol{\theta}_{\mathbf{min}} &= \mathbf{R}^\mathrm{T}\mathbf{Q}^\mathrm{T}\mathbf{y} \\ \mathbf{R}^\mathrm{T}\mathbf{R}\boldsymbol{\theta}_{\mathbf{min}} &= \mathbf{R}^\mathrm{T}\mathbf{Q}^\mathrm{T}\mathbf{y} \\ \mathbf{R}\boldsymbol{\theta}_{\mathbf{min}} &= \mathbf{Q}^\mathrm{T}\mathbf{y} \end{aligned}$$

By backward substitution (since **R** is an upper triangular matrix), each element of **θ**$_{\mathbf{min}}$ can be solved for and hence the set of model parameters is obtained.

Appendix 4: MATLAB Template Building & Verification Function Code

```
function [template] = sysidcell3(sig, disc, sp, seg, order, ph, half_overlap_ratio)

% This function has format
%
% [template] = sysidcell3(sig, disc, sp, seg, order, ph, half_overlap_ratio) where
%
% it calculates the template (fixed segment length) for a set of signatures
% given the following list of parameters:
%
% sig: a 1 x N cell array of k x 3 signature vector (k samples).
% disc: a 1 x N cell array of index vector of discontinuities.
% sp: sampling period.
% seg: length of signal segments.
% order: order of the ARX model.
% ph: prediction horizon used to derive model.
% half_overlap_ratio: specifies the half overlap ratio between each segment.
%
% The output template is a 1 x (Num_segs + 1) cell array of a parameter vector
% and idpoly objects.


if order < 1 | order ~= round(order)
    error('The order selected must be a non-zero positive integer.');
end
if size(sig, 1) ~= 1 | size(disc, 1) ~= 1
    error('The signature and discontinuity cell arrays must be 1 x N in size.');
end
if size(sig, 2) ~= size(disc, 2)
    error('Dimensions of sig and disc inconsistent.');
end
% ==============================================================================
% Read in format data
% ==============================================================================
% Determine the shortest of all signatures
num1 = size(sig, 2);
shortest = [Inf, 0];
for i = 1:num1
    if size(sig{i}, 1) < shortest(1)
        shortest(1) = size(sig{i}, 1);
        shortest(2) = i;
    end
end
xall = [];
yall = [];
for i= 1:num1
    x = sig{i}(:, 1)';
    y = sig{i}(:, 2)';
    % Resample all signals to the shortest length
    if i ~= shortest(2) & length(x) ~= shortest(1)
        x = resample(x, shortest(1));
        y = resample(y, shortest(1));
    end
    xall(i, :) = x;
    yall(i, :) = y;
end
clear shortest;
[xall, yall] = preprocess(xall, yall);

% ==============================================================================
% Create template matrix
% ==============================================================================
% Determine the number of segments in each signal
x = xall(1, :);
seg = seg + floor(mod(length(x), seg)/(floor(length(x)/seg)));
half_ovrlp = round(seg*half_overlap_ratio);
stop = floor(length(x)/seg);
% Create template matrix
num = size(sig, 2);
template = cell(num, stop);

% ==============================================================================
% Fill the template matrix with plant models
% ================================================================================
% Find template for each of the signatures
for i = 1:num
        x = xall(i, :);
        y = yall(i, :);
    % Find ARX plant for each segment
    for j = 1:stop
        % Start and end points for all but the 1st and last segments
        if j ~= 1 & j ~= stop
            sta = (j - 1)*seg - half_ovrlp + 1;
            stp = j*seg + half_ovrlp;
        % The first segment
        elseif j == 1
            sta = 1;
            % If there are more than one segment
            if stop ~= 1
                stp = seg + half_ovrlp;
            % If there is only one segment, can't have overlap
            else
                stp = length(x);
            end
        % Last segment
        else
            sta = (j - 1)*seg - half_ovrlp + 1;
            stp = length(x) - 1;
        end
        sx = x(sta:stp);
        sy = y(sta:stp);
        dat = iddata(sy', sx', sp);
        m = arx(dat, [order order 0]);
        template{i, j} = m;
    end
end
clear dat sx sy;
% ================================================================================
% Assess and choose optimised model for each segment
% ================================================================================
temp = cell(1, stop + 1);
% Go through the models for all segments
disp('Choosing optimal model.');
for i = 1:stop
    disp(['Processing segment ' num2str(i) ' of ' num2str(stop) '.']);
    % Go through the segments of each signature
    fitrec = [-Inf 0];
    for j = 1:num
        m = template{j, i};
        fit = 0;
        % Apply the model to each signature
        for ind = 1:num
            simx = xall(ind, :);
            acty = yall(ind, :);
            sta = 1 + (i - 1)*seg;
            if i ~= stop
                              stp = i*seg;
            % The last segment
            else
                stp = length(x) - 1;
            end
            simx = simx(sta:stp);
            acty = acty(sta:stp);
            simx = iddata(acty', simx', sp);
            simy = predict(m, simx, ph);
            f = (1 - norm(acty - simy.OutputData')/norm(acty -mean(acty)))*100;
            fit = fit + f;
        end
        fit = fit/num;
        if fit > fitrec(1)
            fitrec(1) = fit;
            fitrec(2) = j;
        end
    end
    temp{1, i + 1} = template{fitrec(2), i};
end
clear m simx simy acty;
template = temp;
temp = [];
% Write the sampling rate
temp(1) = sp;
```

```
template{1, 1} = temp;
% =================================================================================
% Determine the mean and standard deviation of the remaining parameters
% =================================================================================
% Calculate the mean and standard deviation of:
fit = [];
for i = 1:num
    [y1, score, fit(i), turn(i), sign_time(i), pudr(i), hwr(i), pen_lift(i)] =
verifycell2(template, sig{i}, disc{i}, ph, [1 1 1 1 1], 1);
end
% =================================================================================
% fit
% bn = min(fit);
% bn = find(fit ~= bn);
% fit = fit(bn);
% Detect a fit value that is more than 50% less than its neighbour
bn = 0;
for i = 2:length(fit)
    if (fit(i) - fit(i - 1)) <= -50
        bn = i;
    end
end
if bn == 0 & (fit(2) - fit(1)) >= 50
    bn = 1;
end
if bn ~= 0
       bn = find(fit ~= fit(bn));
       fit = fit(bn);
end

temp(2) = mean(fit);
if mystd(fit, temp(2)) < 10
    temp(3) = 15;
else
    temp(3) = mystd(fit, temp(2))*1.5;
end
% =================================================================================
% turn
bn = min(turn):range(turn)/2:max(turn);
if isempty(bn) == 0
       hs = hist(turn, bn);
       if hs(1) == hs(2) | hs(2) == hs(3) | hs(1) == hs(3)
        hs = turn;
       else
        hs = [find(hs == max(hs)) range(turn)/4];
        hs = turn(find(bn(hs(1)) - hs(2) < turn & turn < bn(hs(1)) + hs(2)));
       end
else
    hs = turn;
end
temp(5) = mean(hs);
temp(6) = mystd(turn, temp(5));
% =================================================================================
% sign_time
bn = min(sign_time):range(sign_time)/2:max(sign_time);
if isempty(bn) == 0
       hs = hist(sign_time, bn);
       if hs(1) == hs(2) | hs(2) == hs(3) | hs(1) == hs(3)
        hs = sign_time;
       else
        hs = [find(hs == max(hs)) range(sign_time)/4];
        hs = sign_time(find(bn(hs(1)) - hs(2) < sign_time & sign_time < bn(hs(1)) + hs(2)));
       end
else
    hs = sign_time;
end
temp(8) = mean(hs);
temp(9) = mystd(sign_time, temp(8))*1.5;
% =================================================================================
% pudr
bn = min(pudr):range(pudr)/2:max(pudr);
if isempty(bn) == 0
       hs = hist(pudr, bn);
       if hs(1) == hs(2) | hs(2) == hs(3) | hs(1) == hs(3)
        hs = pudr;
       else
        hs = [find(hs == max(hs)) range(pudr)/4];
        hs = pudr(find(bn(hs(1)) - hs(2) < pudr & pudr < bn(hs(1)) + hs(2)));
        end
else
    hs = pudr;
end
temp(11) = mean(hs);
temp(12) = mystd(pudr, temp(11));
% =================================================================================
% hwr
bn = min(hwr):range(hwr)/2:max(hwr);
if isempty(bn) == 0
        hs = hist(hwr, bn);
        if hs(1) == hs(2) | hs(2) == hs(3) | hs(1) == hs(3)
         hs = hwr;
        else
         hs = [find(hs == max(hs)) range(hwr)/4];
         hs = hwr(find(bn(hs(1)) - hs(2) < hwr & hwr < bn(hs(1)) + hs(2)));
        end
else
    hs = hwr;
end
temp(14) = mean(hs);
temp(15) = mystd(hwr, temp(14));
% =================================================================================
% pen_lift
bn = min(pen_lift):range(pen_lift)/2:max(pen_lift);
if isempty(bn) == 0
        hs = hist(pen_lift, bn);
        if hs(1) == hs(2) | hs(2) == hs(3) | hs(1) == hs(3)
         hs = pen_lift;
        else
         hs = [find(hs == max(hs)) range(pen_lift)/4];
         hs = pen_lift(find(bn(hs(1)) - hs(2) < pen_lift & pen_lift < bn(hs(1)) + hs(2)));
        end
else
    hs = pen_lift;
end
temp(17) = mean(hs);
temp(18) = mystd(pen_lift, temp(17));
% =================================================================================
% Update template
template{1, 1} = temp;
% =================================================================================
% Assign weight
% =================================================================================
temp(4) = 100;
temp(7) = 50;
temp(10) = 50;
temp(13) = 20;
temp(16) = 15;
% =================================================================================
% Template complete
% =================================================================================
template{1, 1} = temp;
```

```
function [y1, score, fit, turn, sign_time, pudr, hwr, pen_lift] = verifycell2(t, sig, disc, ph,
tune, sw)

% This function has the form
%
% [y1, score, fit, turn, sign_time, pudr, hwr, pen_lift]
% = verifycell2(t, sig, disc, ph, sw)
%
% it calculates the distance 'fit' of the simulated y1 to the actual y, and
% returns the template parameter values where:
%
% t: template
% sig: a N x 3 signature matrix
% disc: a row vector containing indices of points of discontinuity in x
% ph: prediction horizon
% tune: the number of std tolerance in the order [pen_lift fit sign_time pudr hwr]
% sw: if sw == 0 the function is used in verification mode
%     if sw == 1 the function returns the parameter values without scoring
%     if sw == 2 the function returns in each parameter its z measure
% y1: predicted y
% score: overall score of a signature
% fit: percentage y1's behaviour accounted for by the model
% turn: number of turning points in y1
% sign_time: total signature time
% pudr: pen up/down ratio
% hwr: height to width ratio
% pen_lift: number of pen lifts

% ================================================================================
% Initialise variables
% ================================================================================
y1 = [];
score = [];
fit = [];
turn = [];
sign_time = [];
pudr = [];
hwr = [];
pen_lift = [];
x = sig(:, 1)';
y = sig(:, 2)';
[x, y] = preprocess(x, y);
% ================================================================================
% Piecewise prediction of y
% ================================================================================
% Determine order
order = t{2};
order = size(order.b, 2);
% Determine appropriate segment length
seg = floor(length(x)/(size(t, 2) - 1));
% Retrieve sampling rate
sp = t{1}(1);
stop = size(t, 2) - 1;
try
    for j = 1:stop
        sta = 1 + (j - 1)*seg;
        if j == stop
            stp = length(x);
        else
            stp = j*seg;
        end
        sx = x(sta:stp);
        sy = y(sta:stp);
        dat = iddata(sy', sx', sp);
        m = t{j + 1};
        simy = predict(m, dat, ph);
        y1 = [y1 simy.OutputData'];
    end

    % ================================================================================
    % Perform smoothing
    % ================================================================================
    coeff = fir1(20, 5/300);
    y1 = filtfilt(coeff, 1, y1);

    % ================================================================================
    % Calculate parameters
    % ================================================================================
        % Calculate how much of the co-variation is captured by the model
        fit = 100*(1 - norm(y - y1)/norm(y - mean(y)));
    % ================================================================================
        % Check the number of turning points in y1
        diffy = diff(y1);
        turn = 0;
        for j = 2:length(diffy)
            if sign(diffy(j)) ~= sign(diffy(j - 1))
                turn = turn + 1;
            end
        end
    % ================================================================================
        % Total sign time
        if isempty(disc) == 1
            sign_time = length(x)/600;
            total_time = sign_time;
        else
            tempx = fliplr(sig(:, 1)');
            con = 180/pi;
            len = length(tempx) - 1;
            i = 1;
            nexgradx = 0;
            curgradx = 0;
            % Check for discontinuity by means of gradient change
            while abs(nexgradx - curgradx) < 30 & i <= length(tempx) - 2
                curgradx = atan(tempx(i + 1) - tempx(i))*con;
                nexgradx = atan(tempx(i + 2) - tempx(i + 1))*con;
                i = i + 1;
            end
            total_time = disc(length(disc)) + i - 1;
            sign_time = total_time/600;
        end
    % ================================================================================
        % Determine pen up to down ratio i.e. percentage of time the pen is in
        % the air
        pudr = 0;
        for i = 1:length(disc)/2
            pudr = pudr + disc(i*2) - disc(i*2 - 1);
        end
        pudr = pudr/total_time;
    % ================================================================================
        % Determine height to width ratio
        hwr = range(x)/range(y);
    % ================================================================================
        % pen_lift
        pen_lift = length(disc)/2;
    % ================================================================================
    if sw == 1
        % Fit
        score = 0;
    else
        % Retrieve template values
        fit_mean = t{1}(2);
        fit_std = t{1}(3);
        turn_mean = t{1}(5);
        turn_std = t{1}(6);
        sign_time_mean = t{1}(8);
        sign_time_std = t{1}(9);
        pudr_mean = t{1}(11);
        pudr_std = t{1}(12);
        hwr_mean = t{1}(14);
        hwr_std = t{1}(15);
        pen_lift_mean = t{1}(17);
        pen_lift_std = t{1}(18);
        if sw == 2
            % Assign the z measures for eight calculation
            if turn_std ~= 0
                turn = abs(turn - turn_mean)/turn_std;
            else
                turn = 0.05;
            end
            if sign_time_std ~= 0
                sign_time = abs(sign_time - sign_time_mean)/sign_time_std;
            else
                sign_time = 0.05;
            end
            if pudr_std ~= 0
                pudr = abs(pudr - pudr_mean)/pudr_std;
            else
                pudr = 0.5;
            end
            if hwr_std ~= 0
                hwr = abs(hwr - hwr_mean)/hwr_std;
            else
                hwr = 0.05;
            end
            if pen_lift_std ~= 0
                pen_lift = abs(pen_lift - pen_lift_mean)/pen_lift_std;
            else
                pen_lift = 0.5;
            end
        else
    % ================================================================================
    % Calculate score
    % ================================================================================
            %  See if the number of pen lifts is significantly different
            if pen_lift_std == 0
                pen_lift_std = 0.05;
            end
            if abs(pen_lift - pen_lift_mean) > ceil(pen_lift_std)*1.5*tune(1)
                score = -500;
            else
    % ================================================================================
                % Determine overall fit
                if (fit - fit_mean) > 0
                    score(1) = t{1}(4)*(1 + abs(fit - fit_mean)/(fit_std*tune(2)));
                else
                    score(1) = t{1}(4)*(1 - abs(fit - fit_mean)/(fit_std*tune(2)));
                end
    % ================================================================================
                % Number of turning points
                k2 = 0.5/turn_mean;
                score(2) = t{1}(7)*(-(turn*k2)^2 + 1);
    % ================================================================================
                % Total signature time
                if sign_time_std == 0
                    sign_time_std = 0.05;
                end
                score(3) = t{1}(10)*(1 - abs(sign_time -
sign_time_mean)/(sign_time_std*tune(3)));
    % ================================================================================
                % Pen up down ratio
                if pudr_std ~= 0
                    score(4) = t{1}(13)*(1 - abs(pudr - pudr_mean)/(pudr_std*tune(4)));
                elseif pudr_std == 0 & pudr ~= pudr_mean
                    score(4) = -t{1}(13);
                else
                    score(4) = 0;
                end
    % ================================================================================
                % Height to width ratio
                if hwr_std == 0
                    hwr_std = 0.05;
                end
                score(5) = t{1}(16)*(1 - abs(hwr - hwr_mean)/(hwr_std*tune(5)));
    % ================================================================================
                % Final score
                score = sum(score);
            end
        end
    end
catch
    y1 = 0;
    score = -1000;
    fit = 0;
    turn = 0;
    sign_time = 0;
    pudr = 0;
    hwr = 0;
    pen_lift = 0;
end
```

References

1 R. Plamondon, S. N. Srihari, "On-Line and Off-Line Handwriting recognition: A Comprehensive Survey", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 1, pp. 63-84, January 2000.

2 R. Plamondon and G.Lorette, "Automatic Signature Verification and Writer Identification — The State of the Art", *Pattern Recognition*, Vol. 22, No. 2, pp. 107-131, 1989.

3 X. H. Xiao, G. Leedham, *Signature Verification by Neural Networks with Selective Attention and A Small Training Set*, Nanyang Technical University, Singapore, April 1998.

4 E.J.R. Justino, A. El Yacoubi, F. Bortolozzi, R. Sabourin, "An Off-Line Signature Verification System using Hidden Markov Model and Cross-Validation", *Proc. XIII Brizilian Symposium on Computer Graphics and Image Processing (SIBGRAPI'00)*, pp. 105, Gramado, Brazil, October 17-20, 2000.

5 R. Actis-Grosso, C. de' Sperati, N. Stucci, P. Viviani, "Visual Extrapolation of Biological Motion", *Proc. Of the Seventeenth Annual Meeting of the International Society of Psychophysics*, pp.261-266, Leipzig, October 19th-23rd 2001.

6 C. M. Harris and D. M. Wolpert, "Signal-dependent noise determines motor planning", *Nature*, Vol. 394, pp. 780 – 784, 20 August 1998.

7 S. Schaal and D. Sternad, "Origins and Violations of the 2/3 Power Law in Rhythmic 3D Arm Movements", *Experimental Brain Research*, Vol. 36, No. 1, pp. 60-72, June 2000.

8 A.B. Schwartz and D.W. Moran, "Motor cortical activity during drawing movements: Population representation during lemniscate tracing", *Journal of Neurophysiology*, Vol. 82, No. 5, pp. 2705-2718, 1999.

9 R Plamondon and F. J. Maarse, "An Evaluation of Motor Models of Handwriting", *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 19, No. 5, pp. 1060-1072, September/October 1989.

10 J E Smith. *Principles of Forensic Handwriting Identification and Testimony*, Charles C Thomas Publisher, Springfield, Illinois, U.S.A., pp. 3-33, 1984.

11 V. S. Nalwa, "Automatic on-line signature verification", *Proceedings of the IEEE*, vol. 85, no. 2, pp. 215-239, February 1997.

12 H. D. Crane, J. S. Ostrem, "Automatic Signature Verification Using a Three-Axis Force-Sensitive Pen", *IEEE Transactions on Systems, Man and Cybernetics*, vol. Smc-13, no. 3, May/June 1983.

13 L. R. B. Schomaker, R. Plamondon, "The Relation Between Pen Force and Pen-Point Kinematics in Handwriting", *Biological Cybernetics*, vol. 63, pp. 277-289, 1990.

14 Q. R. Wu, I. C. Jou and S. Y. Lee, "On-Line Signature Verification Based on Split-and-Merge Matching Mechanism," *Pattern Recognition Letters*, vol. 18, pp. 665-673, 1997.

15 A. McCabe, *Implementation and Analysis of a Handwritten Signature Verification Technique*, James Cook University of North Queensland, Australia, 1997

16 H. Dullink, B. van Daalen, J. Nijhuis, L. Spaanenburg, H. Zuidhof, "Implementing a DSP Kernel for Online Dynamic Handwritten Signature Verification Using the TMS320DSP Family," *Proc. DSP Solution Challenge 1995 European Team Papers*, EFRIE, France SPRA304, 1996.

17 Q. Z. Wu, I. C. Jou, S. Y. Lee, "On-Line Signature Verification Using LPC Cepstrum and Neural Networks", *IEEE Transactions on Systems, Man and Cybernetics – Part B: Cybernetics*, vol. 27, no. 1, February 1997.

18 J. Martínez-R, J. de J. López-V, F. J. Luna Rosa, "A low-cost system for signature recognition", *Proc. 3er Congreso Internacional sobre Investigacjon en Ingenieria*

Electrica y Electronica, Aguascalientes, Mexico, November 2002.

19 J. Martínez-R, R. Alcántara-S, "On-line signature verification based on optimal feature representation and neural-network-driven fuzzy reasoning", *Proc. Fifth International Conference on Advances in Infrastructure for e-Business, e-Education, e-Science, e-Medicine on the Internet*, L'Auila, Italy, http://galeb.etf.bg.ac.yu/~vm/cd1/papers/51.pdf, January 2003.

20 I. Nakanishi, N. Nishiguchi, Y. Itoh, Y. Fukui, "On-line Signature Verification Method Using Adaptive Algorithm in Wavelet Transform Domain," *Proc. 2002 International Technical Conference On Circuits/Systems, Computers and Communications*, pp.385-388, Phuket Arcadia Hotel & Resort, Phuket, Thailand, July 2002.

21 T. J. Hastie, E. Kishon, M. Clark, J. Fan, "A Model for Signature Verification." *AT&T Bell Laboratories Technical Report*, February 1992.

22 C. N. Liu, N. M. Herbst, N. J. Anthony, "Automatic Signature Verification: System Description and Field Tests", *IEEE Transactions on Systems, Man and Cybernetics*, vol. Smc-9, no. 1, January 1979.

23 T. H. Keit, R. Palaniappan, P. Raveendran, F. Takeda, "Signature Verification System using Pen Pressure for Internet and E-Commerce", *Proc. The 12th International Symposium on Software Reliability Engineering*, Hong Kong, November 2001.

24 D. Ellen, *The Scientific Examination of Documents: Methods and Techniques*, pp. 40-44, Ellis Horwood Limited, Chichester, U.K., 1989.

25 A. McCabe, "Hidden Markov Modelling with Simple Directional Features for Effective and Efficient Handwriting Verification", *Proc. Sixth Pacific Rim International Conference on Artificial Intelligence (PRICAI 2000)*, Melbourne, 2000.

26 L. R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition", *Proceedings of the IEEE*, vol. 77, No. 2, pp. 257-286, 1989.

27 L. Ljung and T. Glad, *Modeling of Dynamic Systems*, PTR Prentice Hall, Englewood Cliffs, New Jersey, 1994.

28 L, Ljung, *System Identification – Theory For the User*, 2nd ed, PTR Prentice Hall, Upper Saddle River, New Jersey, 1999.

29 C. J. Pritchard, *Determining Signal Levels in Robustly Controlled Plant using Frequency Domain Closed Loop System Specifications*, PhD Thesis, University of the Witwatersrand, Johannesburg, South Africa, 1995.

30 I. MacLeod, and G. Brown, *Control engineering, Feedback System Design* (Lecture Notes), University of the Witwatersrand, Johannesburg, South Africa, 1999.

31 D. Graupe, Identification of Systems, *Van Nostrand Reinhold Company*, New York, New York, 1972.

32 L. Ljung and T. Söderström, *Theory and Practice of Recursive Identification*, The MIT Press, Cambridge, Massachusetts, 1983.

33 T. Söderström and P. Stoica, *System Identification*, Prentice Hall International, 1989.

34 J. P. Norton, *An Introduction to Identification*, Harcourt Publishers Ltd, 1986.

35 MathWorks, Inc., "System Identification: arx", *MATLAB Help*, 2001.

36 MathWorks, Inc., "System Identification: predict", *MATLAB Help*, 2001.

37 D. Sankoff, J. B. Kruskal, *Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison*, Addison Wesley Publishing Company Inc., pp. 125-160, 1983.

38 Y. Gu, *Real-time Dynamic Signature Verification*, BSc (Electrical engineering) Laboratory Project Report, University of the Witwatersrand, Johannesburg, South Africa, 2002.