THE CITY UNIVERSITY

DEPARTMENT OF SYSTEMS SCIENCE

"DEFECT RECOGNITION IN AUTOMATED SURFACE INSPECTION"

Walter John HILL

A thesis submitted for the
award of the degree of
Doctor of Philosophy in Systems Engineering

March, 1977

" ..... Faults observ'd, set in a notebook,
learn'd and conn'd by rote ....."

Shakespeare,
Julius Caesar,
Act 3, Part 2, 96.

DEFECT RECOGNITION IN AUTOMATED SURFACE INSPECTION

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF FIGURES (continued)

LIST OF FIGURES (continued)

LIST OF FIGURES (continued)

ACKNOWLEDGEMENTS

Throughout the work reported in this thesis, I have been most fortunate to have as my supervisor Professor Ludwik Finkelstein. His constant encouragement and guidance have had a profound effect on the development of this work, and on its presentation herein.

I am also indebted to my colleagues, past and present, in the team researching into automatic inspection at the Instrument Systems Centre: L. [redacted], for his enthusiastic and untiring efforts on the project, and V. [redacted], for many stimulating and helpful discussions.

The collaboration and assistance given by the SIRA Institute and the British Steel Corporation have been essential components of this work. Special thanks are due to [redacted], project leader in the Optical Systems Department of the SIRA Institute, and to the research staff of the B.S.C. Welsh Laboratories, Port Talbot.

Finally, I am very grateful to Mrs. Audrey [redacted] for the care and skill which she has devoted to this typescript.

ABSTRACT

Scanning systems for the optical inspection of flat surfaces, moving at high speed, have been developed and are in use in many industries. These systems produce an electrical signal describing the inspected surface, and incorporate signal processing capable of detecting many of the signals which arise from surface defects.

The work reported in this thesis is concerned with the possibility of identifying the defect type from an analysis of the profile of the signal generated by the defect as it is scanned.

The variation inherent in this signal, due to both the characteristics of the scanning systems and to variation between individual examples of the same defect, lead to the conclusion that the statistically based methods of feature space pattern recognition hold the most promise for defect identification. These techniques are reviewed, and those best suited to the system requirements are selected for further study. Most prominent among these requirements are those of fast processing and acceptable cost of implementation. The selected techniques are combined and extended, where necessary, into a set of programs for system design and comparative evaluation.

A data base from sheet tinplate is acquired on magnetic tape, using scanners developed by the SIRA Institute, and used to evaluate the selected techniques. With a suitable combination of these, 80% correct identification is achieved over five defect classes. However, this requires manual intervention in the processing chain so as adequately to delineate (define the limits of) each defect signal, so that measurements can then be made upon it. The systems available for detecting the signals were found to be ineffective for their delineation. A system is therefore developed, based on a bank of matched filters, and shown to provide signal detection and delineation as good as, or better than, that achieved with manual invervention.

A hardware design for the preferred system is developed in detail.

0. INTRODUCTION

For many products produced in the form of flat sheet or strip, the quality of their surface finish is of the utmost significance. A good example is cold-rolled steel strip for use in the manufacture of car bodies. With modern techniques for the application of a paint finish, even apparently minor blemishes on the steel surface can become clearly visible in the finished product. Other defects visible on the surface can cause the steel to break open during pressing operations. Another example is tinplate to be used in the manufacture of tin cans of various kinds. Defects in the tin coating are clearly not acceptable for many such applications. Similar examples are readily available for products such as aluminium, stainless steel, plastics, paper, etc.

Surface defects can arise at many stages of a production process. It is not often feasible so to control the process that an acceptable product invariably results. It is therefore necessary to inspect at various stages of manufacture so as to provide feedback for adjustment of the plant and to direct the product to those customers for whom its surface quality is acceptable. Currently, the inspection of such products for surface quality relies almost exclusively on human inspectors. These inspectors must undergo an extensive period of training before they are able to detect and identify surface defects, in an on-line situation and with an acceptable level of performance. Even so, the performance is often far from perfect. The task demands a high level of concentration and yet can be very boring. These factors tend to be mutually exclusive. The resulting uncertainty associated with manual inspection leads to a significant quantity of acceptable material being rejected, so as to leave an adequate safety margin against customer complaints and the associated loss of prestige, etc. Furthermore, the operating speed of production lines is continuously increasing, making the inspection task ever more difficult. On cold-rolled steel strip, for example, lines are planned for installation before 1980 which will be unable to operate at their maximum speed with manual inspection.

There is, therefore, a pressing need to automate this inspection process. Automation promises a solution to the problem of high production

speeds, greater consistency of inspection, and possibly a higher quality of inspection. In turn, this would lead to a better quality of product reaching the customer, and less wasted material due to delayed detection of process malfunction, and to unnecessary rejection.

Automatic optical inspection can be considered as a three-stage process:

(1) the surface is interrogated by illumination with visible light energy;

(2) the surface interacts with the illuminating energy, so as to change one or more of its characteristics;

(3) these changes are sensed and processed to extract the desired information about the surface.

Several organisations, and in particular the SIRA Institute, have developed high-speed optical scanners which implement the processes of surface interrogation and response sensing in the on-line environment. In addition, signal processing systems have been developed to provide detection of surface defects on suitable materials (ref. 1). Such inspection systems have established themselves as "inspector-aids", whereby they serve to alert a human inspector to potentially defective material, but also to reject automatically material which is grossly defective. These systems suffer from an inability to distinguish the various different kinds of defect on the inspected surface, and similarly to distinguish genuine defects from innocuous surface marks. This latter aspect of defect recognition encroaches upon their capability for defect detection. It is because of this that they cannot function autonomously, but require instead a human inspector for back-up. Furthermore, this shortcoming renders them virtually unusable on products such as aluminium, in which liberal surface lubrication is part of the production process. These systems respond to the lubrication as though it were a defect, and thereby generate an intolerable number of "false alarms" (ref. 1).

The broad aim of this project is to investigate the possibility of endowing such systems with a capability for defect recognition. This aim is linked to potential industrial application, in terms of cost and the constraints of the on-line situation. The work reported in this thesis is part of an ongoing research effort in the Instrument Systems Centre of The City University. It has been carried out in close

collaboration with the SIRA Institute and the British Steel Corporation. Specific objectives of this work were to identify suitable techniques for recognition processing, to evaluate these techniques in computer simulation on data gathered from a suitable material, and to develop the most promising into an outline design for an on-line system.

1. SURFACE INTERROGATION AND DEFECT DETECTION

1.1 Introduction

As previously mentioned, automatic optical inspection involves the interrogation of a surface by illuminating it with visible light energy. The surface interacts with the energy, changing one or more of its characteristics. These changes must be sensed and processed to yield the required information about the surface. The primary difficulty in this task is that a suitable model of the energy-surface interaction is not available. This leads, inevitably, to an empirical approach to the selection of suitable signal processing techniques. The implications of this for recognition processing will be discussed in Chapter 2. The purpose of this chapter is to describe the means whereby:

(1) the surface is interrogated;

(2) the energy-surface interaction is sensed;

(3) the sensed output is processed to detect surface defects.

The scope of this chapter is restricted primarily to systems developed by the SIRA Institute, the collaboration of which has been an essential part of this project. These systems will be described and analysed with special emphasis on those aspects most relevant to defect detection and recognition.

1.2 Interrogating the surface and sensing its response

For this project, interest centres on materials produced in strip form, such as cold-rolled steel strip and tinplate strip. Inspection of such materials involves at least one scanning process due to the material moving past the inspection device. In addition, the strip may be scanned across its width to produce a scanned raster pattern on the surface from the interaction of across strip scanning with strip movement. The scanning process due to strip movement applies inevitably to both surface interrogation and to response sensing. If across strip scanning is used it may involve either or both of these. Inspection systems may therefore be

classified into two main groups - those which scan across the strip surface and those which do not. The former will be referred to as sequential scan systems, and the latter as parallel scan systems (ref. 2). In all cases to be discussed, interaction between the illuminating energy and the surface is sensed only in terms of light absorption, scattering or deflection. Interaction through other characteristics of the illuminating energy, such as colour or polarisation, has, to the author's knowledge, not been used.

1.2.1 Parallel Scan Systems

In parallel scan systems, the entire width of the inspected strip must be illuminated simultaneously, for example by a strip light source positioned above the surface. As the strip passes beneath the illuminating source, simultaneous scanning of many individual points across the surface is achieved by an array of photosensors (for example, semiconductor photodiodes). Between 50 and 500 sensors per metre width might typically be used, with each sensor responding to light reflected or scattered from the surface, or both. Each sensor may be provided with an independent processing channel, or sensor outputs may be combined in a simple way to reduce costs. With the latter option, problems can arise from varying sensitivities from sensor to sensor.

Generally, these systems need to operate in close proximity to the inspected surface where they are liable to mechanical damage. They offer advantages over sequential scan systems in that they can be more sensitive to linear surface defects oriented across the strip, and in that they allow parallel signal processing with its attendant high speed capability.

No experience has been gained with such systems in this project, and they will not be considered further.

1.2.2 Sequential Scan Systems

Sequential scan systems can have a single channel of signal processing, and can therefore be simpler and cheaper than parallel scan systems. This advantage, however, is mitigated by a higher bandwidth requirement.

As previously mentioned, sequential scan systems allow options to scan either for illumination or for response sensing, or both,

in the across strip direction. To the author's knowledge, a system which does both has not been produced. The SIRA Institute has produced systems which scan for response sensing, with simultaneous illumination of the entire strip width, and systems which scan the illuminating energy, with simultaneous sensing of the entire strip width. Data from both has been used in this project, and it is necessary to understand their operating principles.

With all sequential scan systems, the system itself provides scanning motion across the strip width, with the strip itself moving perpendicular to the scan direction, so that successive scans cover successive strips of the surface. 100% surface coverage is achieved when each scan just fails to overlap the preceding one, as shown in Figure 1.1. If the instantaneously viewed area in the direction of strip motion has dimension ℓ metres, and the scan rate is N scans.second^{-1}, then 100% surface coverage will be achieved at a strip velocity of (N x ℓ) metres.second^{-1}. Typical figures are N = 400 to 4800 scans.second^{-1} and ℓ = 1 cm, leading to corresponding strip velocities between 4 and 48 metres.second^{-1}.

1.2.2.1 The Flying-Field Scanner

This system (ref. 1) incorporates a strip light source to illuminate evenly the entire strip width. Response scanning is achieved by a rotating lens drum with eight identical lenses equally spaced around its periphery (Figure 1.2). At any instant, only one lens will be viewing the surface, and will form an image of the entire illuminated area on the axis of rotation of the drum. As the drum rotates, images are therefore scanned sequentially across this axis.

The scanning mechanism is set up at the specular (mirror) angle with respect to the illuminating source. Surface blemishes cause a reduction (either by scattering or absorption) in the amount of light specularly reflected from the surface, and give rise to correspondingly darkened areas in the image. A fine slit on the axis of rotation of the lens drum allows light from a small portion of the image to pass to a photomultiplier, which yields an electrical signal

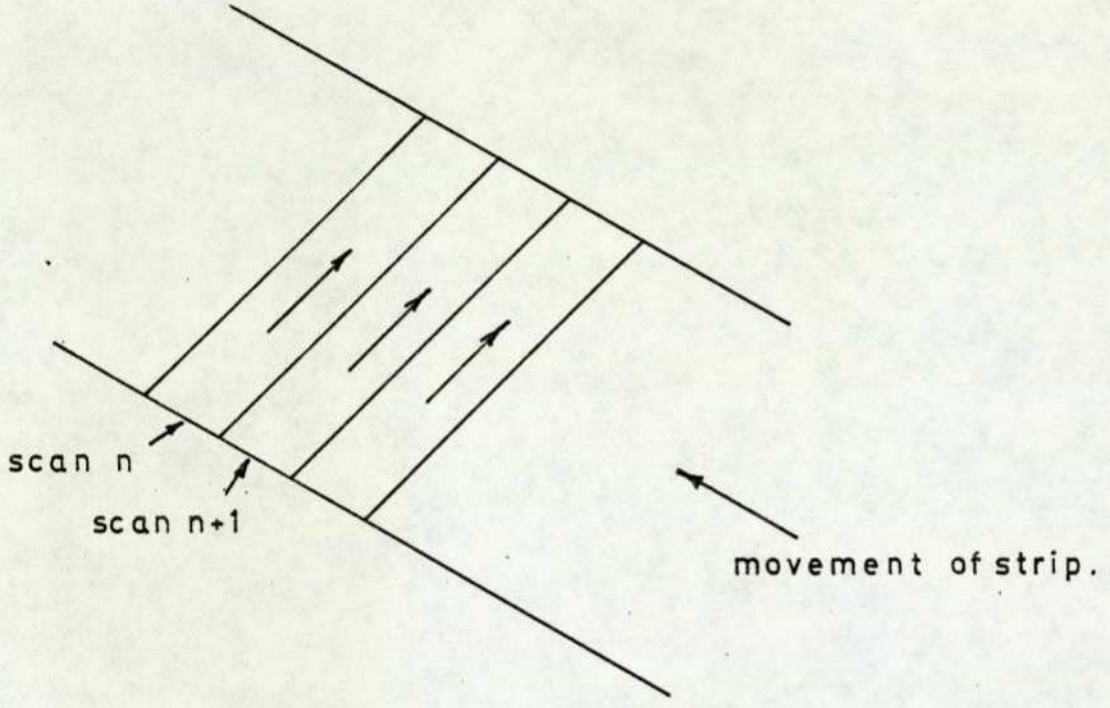


FIGURE 1.1 - SEQUENTIAL SCANNING.

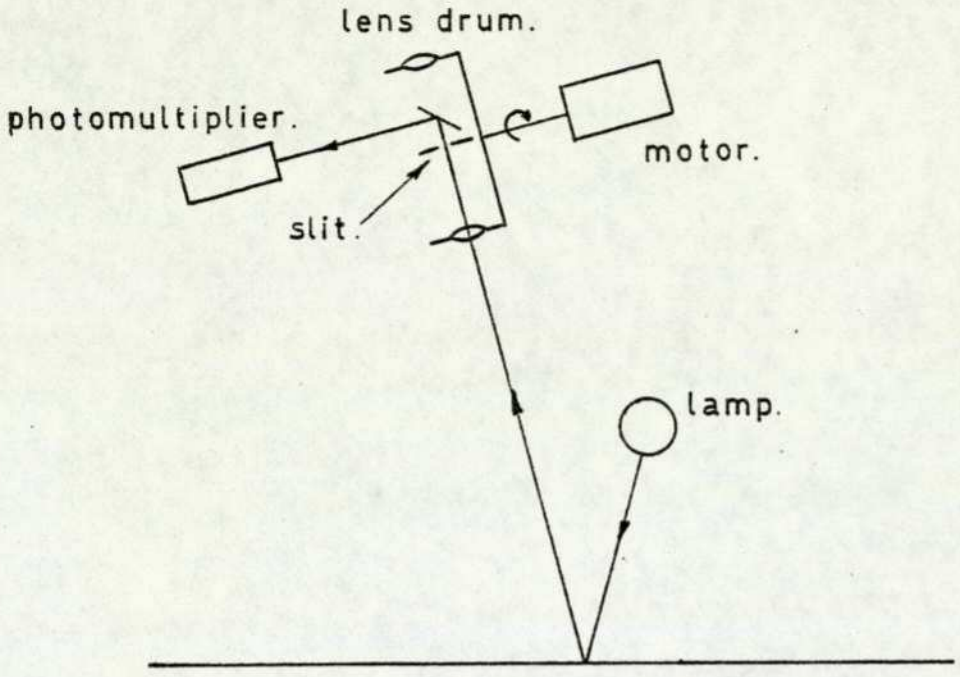


FIGURE 1.2 - THE FLYING-FIELD SYSTEM.

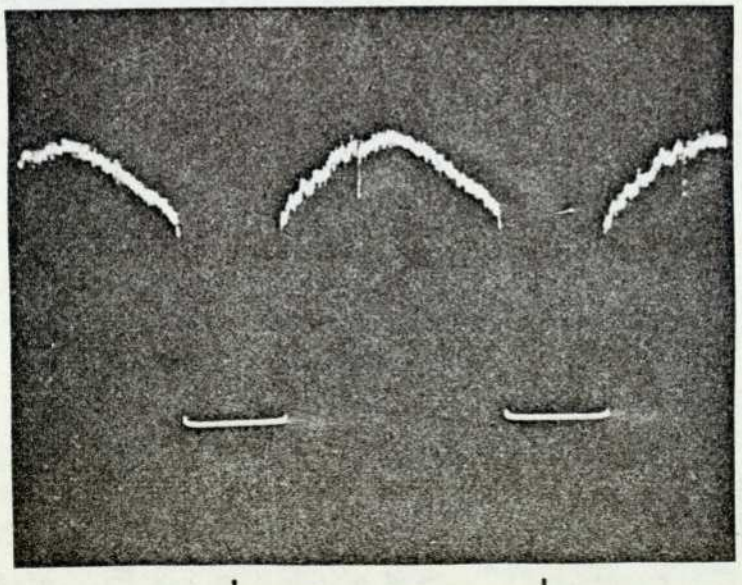


FIGURE 1.3 - A TYPICAL 'SCAN PEDESTAL' FROM THE FLYING-FIELD SYSTEM.

proportional to the total amount of light falling on its cathode. As the lens drum rotates, the image is scanned across the slit and the photomultiplier therefore generates a signal which is an analog of surface reflectivity. For each scan of a single lens across the surface, this signal takes the form of a "pedestal" (Figure 1.3), in which the edges correspond to the edges of the strip. In addition, the scanning geometry imposes a curvature on to the top of the pedestal which complicates subsequent signal processing.

An important aspect of this system is the instantaneously viewed area on the inspected surface. This is determined primarily by the dimensions of the slit located on the axis of rotation of the lens drum, through which light is allowed to pass to the photomultiplier. On the surface, this area takes the form of a rectangle, oriented as shown in Figure 1.4. As previously mentioned, the dimension in the direction of strip motion determines the necessary scan rate for 100% surface coverage at a given strip velocity. The dimension in the across strip direction needs to be sufficiently small that the smallest defects of interest give rise to an adequate change in the total reflected light from the area. Two factors combine to set a lower limit to this dimension:

(1) The signal from the photomultiplier is corrupted by electrical shot noise, at a level proportional to the square root of the total illumination reaching the cathode (ref. 3). The desired signal is directly proportional to this illumination, so that the electrical signal-to-noise ratio falls with the level of illumination reaching the photomultiplier. That level is proportional to the area of the surface instantaneously viewed, and therefore to its across strip dimension.

(2) The optical signal reaching the photomultiplier is affected by random fluctuations of the surface profile, even in regions of the surface which are defect-free. The signal due to a surface defect must be large enough, compared to this "structure noise", for it to be detectable. The structure

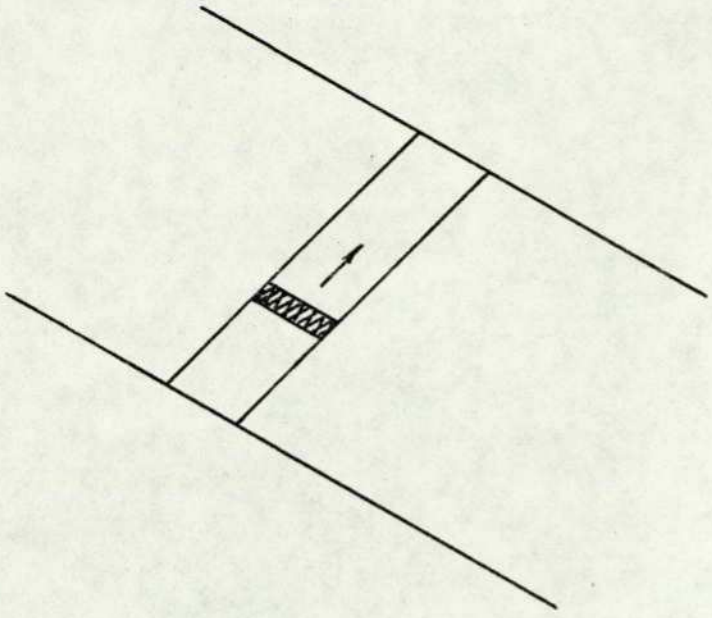


FIGURE 1.4 - THE INSTANTANEOUSLY VIEWED AREA.

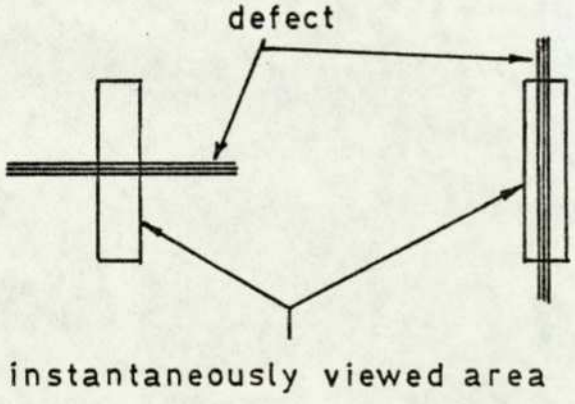


FIGURE 1.5 - SENSITIVITY TO DEFECT ORIENTATION.

noise amplitude increases as the instantaneously viewed area decreases, so long as this area remains "large" with respect to the structure.

Typical dimensions for the instantaneously viewed area are 1 mm across the strip and 1 cm along it. With lens drum rotating at 50 revolutions.second^{-1}, these dimensions provide 100% surface coverage at a strip velocity of 4 metres.second^{-1}.

An important consequence of this rectangular shape is a preferential sensitivity of the system to linear defects (such as scratches) oriented parallel to the strip edges. As illustrated in Figure 1.5, this orientation provides maximal darkening of the viewed area. This characteristic has been found to be acceptable on many strip products, since defects tend to be so oriented as a result of the rolling process.

With this scanner, three causes can be identified which introduce variation into the electrical signal from a defect, according to the location of the defect across the strip width:

(1) The surface close to the edges of the strip is illuminated and viewed at a different angle from that close to the centre.

(2) Since the surface is flat, whereas each lens describes an arc of a circle, no more than two points on the surface can be perfectly focussed on to the axial slit.

(3) The angular velocity of the lens drum is constant, so that the linear velocity of the instantaneously viewed area across the surface is not, being higher at the centre of the strip than at the edges. Therefore, the time duration of the signal from a defect of given width will be shorter when that defect is near the strip centre, than when it is close to the edge.

In addition, this scanner is sensitive to light scattered from regions of the surface outside the instantaneously viewed area. This effect is illustrated in Figure 1.6, and can greatly reduce (or even eliminate) the contrast generated by

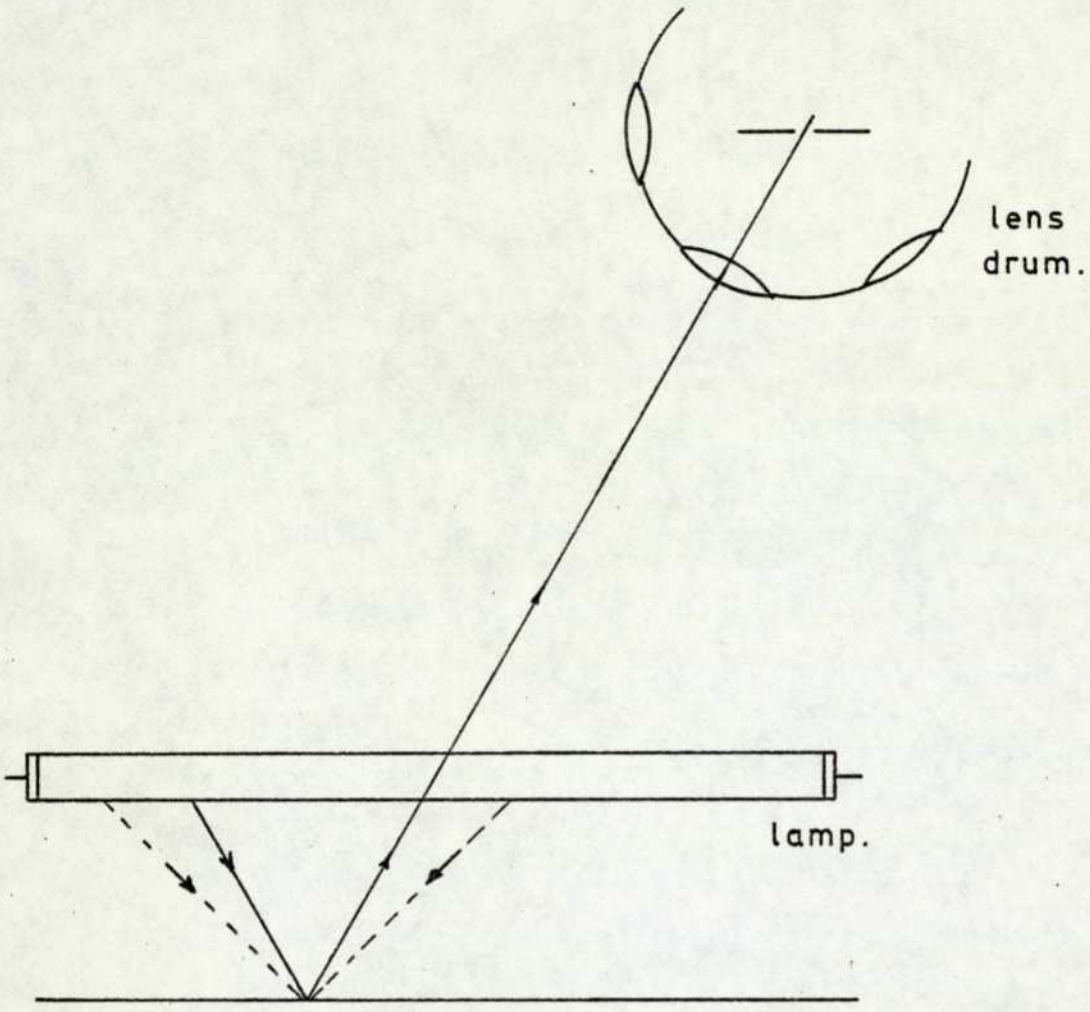


FIGURE 1.6 - ACROSS-STRIP SENSITIVITY OF THE FLYING-FIELD SYSTEM. (from ref. 2)

a defect.

1.2.2.2 The Flying Spot (Laser) Scanner)

This system (ref. 4) incorporates a large optical mirror system to focus the entire strip width (up to 1.5m) on to a photomultiplier cathode. A helium-neon laser provides a source of illuminating energy which is scanned across the strip width by a rotating, multi-faceted prism. Figure 1.7 shows the essential aspects of the system. Both the illuminating source (the rotating prism) and the receiver (the photomultiplier cathode) are situated on the focal axis of the curved mirror, when allowance is made for the various plane mirrors in the path. The source is offset to one side of the focus and the receiver to the other, so that in the case of a perfectly reflecting surface, light is simply transmitted from one to the other. To this extent, the surface is treated as an imperfect plane mirror, and viewed at the specular angle with respect to the illuminating source. As before, defects which cause a reduction (either by scattering or absorption) in the amount of light specularly reflected from the surface, may be detected. The signal is similar to, but better than, that produced by the flying-image scanner.

A laser is used as a convenient source of a concentrated beam of high intensity light. The polarisation and coherent properties of the light are not currently exploited. The intensity is sufficiently high that normal ambient illumination of the surface makes only a negligible contribution to the photomultiplier output signal. Essentially, the instantaneously viewed area is defined by the laser spot on the surface. As in the flying-field system, and for similar reasons, this area is rectangular in shape - typically 1 mm by 1 cm - and is achieved by simple optical shaping of the beam.

This system has many significant advantages over the flying-field system:

(1) It does not suffer from light scattered from regions of the surface outside the instantaneously viewed area.

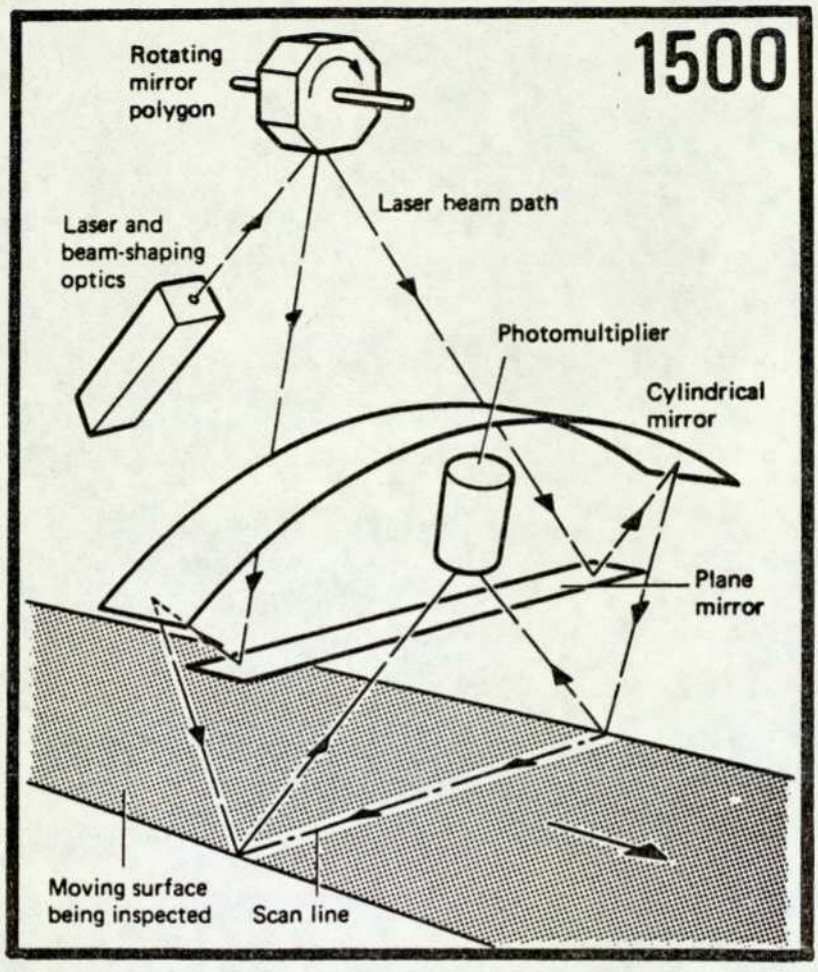


FIGURE 1.7 - THE FLYING-SPOT LASER SCANNER.

(2) The total light intensity reaching the photomultiplier is greater than in the flying-image scanner. In fact, it is sufficiently high that electrical noise from the photomultiplier itself is rendered negligible, as evidenced by repeated scans over the same surface area yielding substantially identical results. Further, the system currently uses a 5 mW laser, and more powerful sources are readily available. The limit is likely to be set by considerations of operator safety. A comparable extension of the flying-field system demands a high-intensity light source in strip form, and this is not easily achieved.

(3) Higher scan rates are possible. In the flying-field system, the rotating mass consists of a substantial lens drum, in which each lens is provided with independent adjustment for focus. In the laser scanner, this is reduced to a small prism. The flying-field system operates at 400 scans.second^{-1}, and any significant increase is thought to be impractical. In contrast, the laser scanner operates at 600 scans.second^{-1} and will shortly be upgraded to 4,800 scans.second^{-1}. The likely upper limit is thought to be higher than 24,000 scans.second^{-1}.

(4) Several extensions to the laser scanner are likely in the future, which would be impractical or impossible with the flying-image system:

(a) The laser light is inherently plane-polarised and coherent. The interaction of these properties with surface defects is currently being studied, with a view to improved detection and recognition.

(b) Additional photomultipliers can be positioned in the scattered light field to provide additional information about the surface/defect. With the laser scanner this is a much simpler task than with the flying-field scanner.

(c) The laser spot size on the surface can be reduced by adjusting the optics which shape the beam, down to a lower limit set by diffraction effects. This would cause no reduction in the amount of light reaching the photomultiplier, since the same amount of light would be concentrated on to a smaller area. In the flying-field scanner, it would be necessary to reduce the area of the slit through which the photomultiplier receives light from the image, with an attendant reduction in the signal-to-noise ratio.

On page 1.5 three causes of unwanted signal variation were described for the flying-field scanner. Similar causes can be identified for the laser scanner. The scanner described is designated type 1500 by the SIRA Institute. Alternative types are produced which use a lens instead of the large mirror. These cover a smaller strip width, but have the advantage of illuminating and viewing all regions of the surface uniformly. No experience has been gained with such systems in this project.

1.3 Defect detection

The sequential scan systems described in Section 1.2.2 are sensitive to energy-surface interaction in terms of changes in the amount of light specularly reflected from the surface, for a particular angle of incidence.

Surface defects give rise to absorption, scattering or deflection of the incident light energy, and thereby reduce the amount specularly reflected. Such defects therefore manifest themselves in the response signal as negative-going pulses. Figure 1.3 shows such a pulse, corresponding to a small pit in the steel surface. Occasionally, a defect may give rise to a positive-going pulse, corresponding to increased reflectivity or "surface glint", but this is rare. The systems developed by the SIRA Institute incorporate signal processing electronics to detect such pulses, and it is the purpose of this section to describe and analyse this processing.

The scanner response signal is first normalised to a constant mean level by means of a slow-acting automatic gain control on the photomultiplier anode voltage. This normalisation is designed so as not to respond during the period of a single pedestal, but rather to average over several such periods. It does not, therefore, alter the pedestal shape but only its mean value. Thus, for example, the curvature on the pedestal remains unchanged.

Following this normalisation, the signal is passed through a low-pass filter, appropriately switched at the pedestal edges, and then attenuated slightly. This produces a "reference pedestal" as shown in Figure 1.8. If the normalised response signal falls below the reference signal, a defect is indicated. This cross-over is detected by a straightforward comparator. Additional reference signals, at increasing levels of attenuation, may be provided to give a measure of the severity of the defect, to the extent that this is reflected in the pulse amplitude. In this respect, it is perhaps worth emphasizing that these scanners do not respond, directly ,to the surface profile. Instead, they respond to its reflectivity, which is not necessarily related to profile. Loss of reflectivity may be caused by an absorbing

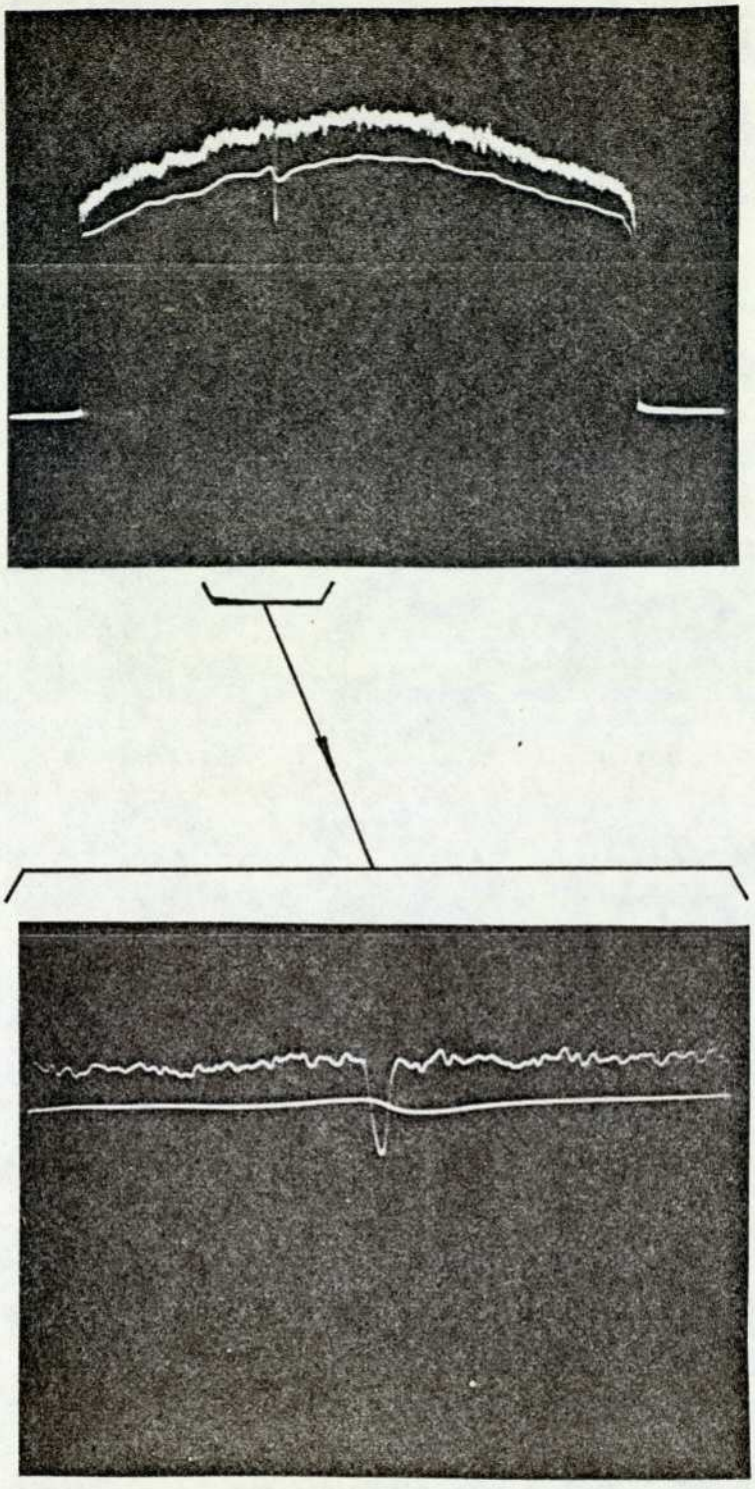


FIGURE 1.8 - OPERATION OF THE SIRA DEFECT DETECTION SYSTEM.

defect, such as a stain, or by an irregularity in the surface profile, such as a gouge, and the system cannot necessarily distinguish between the two. Multiple threshold processing should not, therefore, be seen as giving a measure of the "depth" of a defect.

The basic principles of the single threshold system can be described with reference to Figure 1.9, in which:

$x(t)$ is the normalised response signal applied to the filter

$y(t)$ is the response of the filter

k is the attenuation factor

$z(t) = x(t) - k.y(t)$
is the composite difference signal.

The criterion for defect detection is that $x(t)$ becomes less than $k.y(t)$, or, equivalently, that $z(t)$ becomes less than zero.

The fundamental principle underlying this scheme is to low-pass filter a signal and to subtract that filtered signal from the original. This can be equivalent to high-pass filtering the original signal. This is readily seen for the SIRA system if the attenuation factor, k, is taken to be unity (i.e. no attenuation), as follows:

In Laplace transform analysis, we have for Figure 1.9:

$$Y(s) = X(s)H_1(s)$$

$$\text{and } Z(s) = X(s) - k.X(s)H_1(s)$$

$$= X(s)\left[1 - k.H_1(s)\right]$$

where $X(s)$, $Y(s)$ and $Z(s)$ are the transforms of $x(t)$, $y(t)$ and $z(t)$ respectively,

and $H_1(s)$ is the transfer function of the low-pass filter.

So that

$$Z(s) = X(s)H(s)$$

where $H(s) = 1 - k\,H_1(s)$.

$H(s)$ is the overall transfer function of the system, which we wish

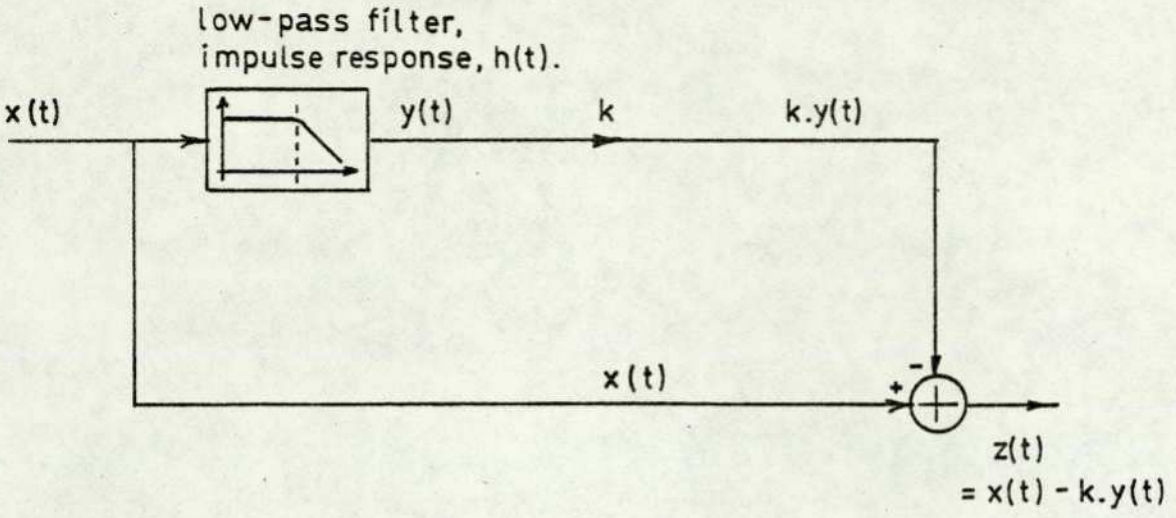


FIGURE 1.9 - THE SIRA DEFECT DETECTION SYSTEM.

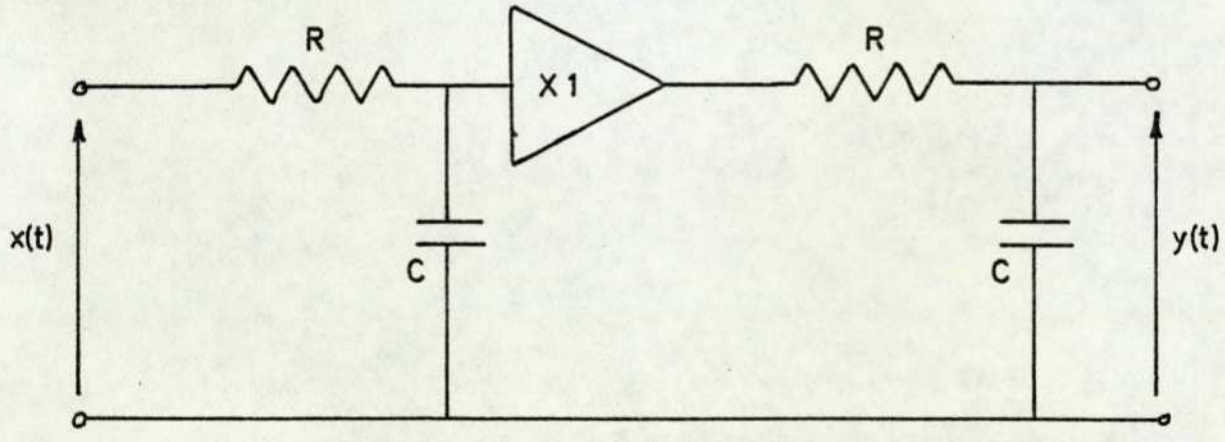


FIGURE 1.10 - THE SIRA LOW-PASS FILTER.

to show is a high-pass function when k = 1.

Figure 1.10 shows the circuit of the SIRA low-pass filter. From this circuit, we have:

$$H_1(s) = \frac{1}{(1 + s\tau)^2}, \text{ where } \tau = CR.$$

Substituting into the expression for H(s), we have:

$$H(s) = 1 - k\,H_1(s)$$

$$= 1 - k\left[\frac{1}{(1 + s\tau)^2}\right]$$

$$= \frac{1 - k}{(1 + s\tau)^2} + \frac{s\tau(2 + s\tau)}{(1 + s\tau)^2}$$

$$= (1 - k)H_1(s) + H_2(s)$$

where $H_2 s = \frac{s\tau(2 + s\tau)}{(1 + s\tau)^2}$

$H_2(s)$ is the transfer function of a high-pass filter, and with k equal to unity, H(s) reduces to

$$H(s) = H_2(s)$$

which is purely high-pass, as required.

The situation when k is less than unity is slightly more complex. We have:

$$Z(s) = X(s)H(s)$$

$$= X(s)\left[(1 - k)H_1(s)\right] + X(s)H_2(s)$$

which may be written as

$$Z(s) = Z_1(s) + Z_2(s)$$

with $Z_1(s) = X(s)\left[(1 - k)H_1(s)\right]$

and $Z_2(s) = X(s)H_2(s)$.

In the time domain:

$$z(t) = z_1(t) + z_2(t).$$

Since $H_1(s)$ is purely low-pass, and $H_2(s)$ is purely high-pass, we can say that $z_1(t)$ corresponds to the attenuated low-frequency

content of $x(t)$, and $z_2(t)$ to the high-frequency content.

The criterion for defect detection is that $z(t)$ should become less than zero. Thus:

$$z(t) < 0$$

$$\text{therefore } z_1(t) + z_2(t) < 0$$

$$\text{therefore } z_2(t) < - z_1(t).$$

$z_1(t)$ is always greater than (or equal to) zero, so that $z_2(t)$ must be less than zero. This reflects the fact that the system will detect negative-going pulses when $k < 1$, and positive-going pulses when $k > 1$. Furthermore, $z_2(t)$ must be less than zero by an amount determined by $z_1(t)$, for a defect to be detected. This means that detection is achieved by thresholding the high-frequency content of $x(t)$, but with a threshold which is proportional to the low-frequency content. This dependence on the low-frequency content allows the system to operate on proportional changes in the signal, and thereby provides inherent compensation for surfaces of differing average reflectivity.

1.4 Summary

In this chapter automatic surface inspection has been introduced as a process whereby a surface is interrogated by illuminating it with visible light energy, the surface interacts with that energy to change one or more of its characteristics, and these changes are sensed and processed to yield the desired information about the surface. Different procedures for illumination and response sensing lead to different inspection systems.

Systems produced by the SIRA Institute have been the source of data used in this project, and these have been described within the general framework above. The particular characteristics of these systems relevant to subsequent signal processing for defect detection and recognition have been emphasized. A processing scheme devised by the SIRA Institute to detect surface defects has been described and analysed in some detail.

It was believed, at the start of this project, that this detection system would be used as a buffer between the scanner and subsequent recognition processing. Thus, signals would be gated through to the recognition sub-system only when a defect was believed to be present, as indicated by the SIRA detection system. As work proceeded, it became clear that the system devised by the SIRA Institute did not meet the rather special requirements of such a procedure. The process of defect detection has therefore been examined more closely, and an alternative processing scheme has been developed. This work will be described in Chapter 5 of this thesis.

2. DEFECT RECOGNITION

2.1 Introduction

The processes of surface interrogation and response sensing have been discussed in Chapter 1, with emphasis on the sequential scan systems developed by the SIRA Institute. These systems produce a representation of the inspected surface in the form of a succession of "single-scan" signals, each of which corresponds to a narrow strip of the surface in a direction transverse to the rolling direction. Such a signal is shown in Fig. 2.1, containing several negative-going defect pulses.

Recognition processing need operate only on those sections of each signal which relate to surface defects. This pre-supposes some means of detecting and delineating those sections on each scan. This problem is the subject of Chapter 5 of this thesis, where a solution is developed. For the purposes of this chapter, the problem will be taken as solved. In other words, we shall assume that we have available, for each scan signal, a binary delineation signal which identifies those sections of the scan which arise from surface defects. This is illustrated in Fig. 2.1. The delineation signal will serve as a "gating" waveform, to allow the defect signals alone to pass to a recognition sub-system. The details of this gating process will be developed in Chapter 6 of this thesis.

Each scan signal relates to a strip of the surface which is typically about 1 cm wide in the rolling direction. Some small defects will be covered completely by such a strip, so that only a single scan section will then be available for recognition processing. In many cases, however, a number of successive scan sections will be available, all from the same defect. For best recognition performance, processing should be based on all the available scan sections from each defect, taken together. This processing should take account of the shape of each section, the way in which that shape varies from section to section, as well as the positional relationships between sections (referred to the inspected surface). To do this, a sub-system must first be developed to associate those scan sections which arise from the

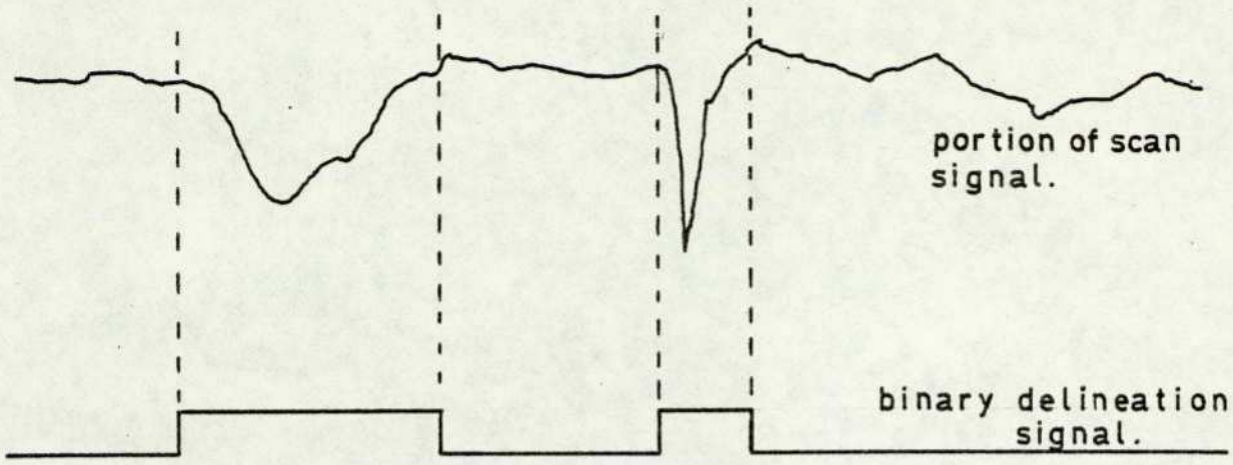


FIGURE 2.1 - DETECTION AND DELINEATION.

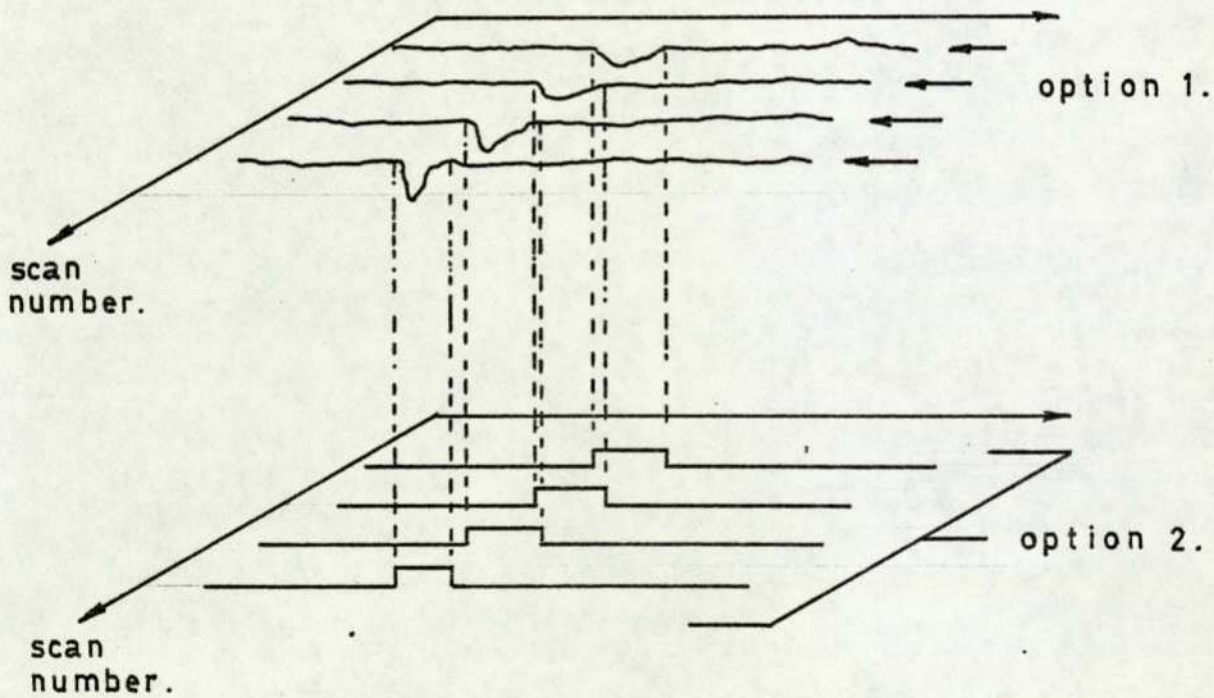


FIGURE 2.2 - SIGNAL PROCESSING OPTIONS.

same defect. This sub-system would need to cope with such problems as vee-shaped defects and linear defects along the strip which are discontinuous. Furthermore, a recognition system which used all of the available data in this way would be relatively complex and costly - in both development and implementation. At least two sub-optimum approaches can be identified which offer substantial simplifications, hopefully without undue loss of performance. These are:

(i) Apply recognition processing to each scan section, as it arises, taking no account of its relationship with other sections from the same defect

(ii) Apply recognition processing to the full set of delineation signals arising from each defect, taking into account their spatial relationships referred to the inspected surface, but taking no account of the detailed form of the scan sections underlying each one.

Fig. 2.2 illustrates these two options.

With option (i) there is no need to associate scan sections from the same defect, or to store and cross-reference information from scan to scan. With option (ii) these processes are necessary, but the recognition sub-system need operate only on binary data. It can be said that option (i) would function on a microscopic level, with recognition based on the fine structure of the defect, as reflected in the detailed shape of the scan section, whereas option (ii) would operate on a macroscopic level, making use only of the overall defect shape. With the former, for example, high contrast defects could be separated from low contrast defects, and wide defects from narrow ones. With the latter, vee-shaped defects could be separated from linear defects such as scratches. Each option needs to be evaluated, with the ultimate possibility of combining the results at a higher level of processing. This thesis is concerned exclusively with the first option, isolated scan sections, and a related thesis (ref. 5) is concerned with the other.

2.2 System Requirements and the Recognition Strategy

To derive the maximum benefit from this project, several requirements need to be defined to guide the research. These are:

(1) Recognition processing should be based on the signals generated by scanning systems of the kind developed by the SIRA Institute. Techniques cannot be realistically evaluated in a vacuum. Representative data from a specific problem area is necessary, because the difficulties of the task can be revealed only by processing such data. The SIRA systems are available and well established in several industries for defect detection and as operator aids. Furthermore, close contact between The City University and the SIRA Institute was already established, via the Instrument Systems Centre.

(2) Recognition processing should be "on-line", with a data processing rate suited to the fastest production lines currently in use, or envisaged for the near future.

(3) The cost of endowing currently available scanners with a capability for defect recognition should be commercially acceptable. Although this requirement is extremely difficult to quantify, a reasonable rule-of-thumb seems to be that recognition processing should add no more than about £40,000 to the cost of the scanner. This implies a component cost of no more than about £10,000.

(4) A system developed for, say, tinplate should be adaptable for other products, such as plastic, aluminium, steel, etc. - even though the defects to be recognised may be quite different - at a cost significantly less than the original development.

The first requirement has been inherent in the discussion to date. The second is of the utmost significance. The rate at which data must be processed in an on-line system can be calculated from

four factors:

(1) The dimension of the instantaneously viewed area in the direction of strip motion (see section 1.2.2)

(2) Strip velocity

(3) Strip width

(4) The spatial data sampling rate, referred to the inspected surface (for a digital processor - analogous to signal bandwidth for an analogue processor).

Although these factors are not independent, and will vary considerably from line to line, reasonable target figures seem to be:

(1) area dimension = 5 mm
(2) strip velocity = 25 $m.s^{-1}$
(3) strip width = 1 m
(4) sampling rate = 1 sample per mm.

The following relations apply, for 100% surface coverage:

$$\text{Scan period} = \frac{\text{area dimension}}{\text{strip velocity}}$$

$$= \frac{5.10^{-3}}{25} \text{ secs} = 200.10^{-6} \text{ secs.}$$

Number of samples per scan

= sampling rate x strip width

= 1000

$$\text{Sampling frequency} = \frac{\text{number of samples per scan}}{\text{scan period}}$$

$$= \frac{1000}{200.10^{-6}}$$

$$= 5.10^{6} \text{ samples per second.}$$

Thus, for example, if each sample is of 8 bits, this corresponds to a data rate of 40.10^6 bits per second. This is, unquestionably, a very high data rate. It implies parallel processing with special purpose hardware, rather than a general purpose computer. There is, however, a saving grace. The calculated data rate is the one at which raw data would be generated from the inspected surface, and this need be processed in its entirety only for defect detection

and delineation.

Recognition processing need operate only on those portions of the data relating to surface defects, and most of the surface should be defect-free. The attendant data rate reduction is difficult to quantify, but an average figure of 1 m s between "recognitions" seems reasonable.

It is necessary to decide upon the general form which recognition processing should take. One possibility is an ad-hoc approach in which known characteristics of each defect are embodied in a special purpose processing sub-system matched exclusively to that defect. Thus, for example, scratches might be identified from their long, thin outline, usually oriented along the rolling direction. "Chevrons" on tinplate have a characteristic "vee" shape which could be exploited for recognition. The result would be a collection of recognition sub-systems, probably one for each defect. Each sub-system would be fairly complex, depending upon the defect for which it was designed, and the overall complexity could well prove unacceptable. More importantly, the kind of a-priori knowledge required for this approach is generally not available within the framework of isolated scan sections. The characteristics mentioned are macroscopic, rather than microscopic. Comparable data on the fine structure of defects, as reflected in each scan section, is largely unavailable. Given this, perhaps a study might be undertaken to establish such characteristics. Such a study would need to encompass the following tasks:

(1) to establish for each defect class a characteristic structure on the microscopic level

(2) to establish the relationship between such structure and the scanner response signal.

For the first, the problem of considerable structure variation within each class would need to be overcome, and for the second, the lack of a suitable model of the interaction between the surface and the interrogating light energy would hamper the investigation. These considerations suggest a largely empirical approach. However, it is likely that no single characterising feature of isolated scan sections will serve to identify many of the defects.

Instead, it may be necessary to exploit the information contained in many such features, each of which may seem worthless in isolation. How can such a study encompass this possibility?

To the author's knowledge, the only consistent methodology for this problem is embodied in the techniques of automatic pattern recognition. With these techniques, this kind of information can be extracted by computer processing from a collection of data, and used to establish recognition strategies for the classes represented therein. The techniques do not demand a-priori definition of the class characteristics, aiming instead to extract these from collected data. It should not be inferred, however, that these techniques offer an easy solution to the recognition problem. There is first the task of gathering a representative data set to define the classes of interest. Each scan section must then be represented by a set of characterising measurements chosen by the designer. The quality of these measurements is of the utmost importance. Finally, it is necessary to select from the vast array of techniques available in pattern recognition, those most suitable for this application. The constraints of a high rate of data processing, coupled with an acceptable cost of implementation, bear heavily on this selection.

Accordingly, a study of the literature on pattern recognition has been made and the results of this study will be summarised in the next section. In general, only the author's conclusions will be presented and justified, with the reader referred to the literature for further details of the techniques. However, those techniques which were selected for simulation and evaluation will be more fully discussed. The study was confined to so-called feature space techniques and did not embrace the linguistic or structural approach. With the latter, items for recognition must be represented as strings in a suitable formal grammar, and a separate grammar discovered for each pattern class. For recognition, an item must first be represented by its corresponding string and the grammar most likely to have generated that string must be determined. Unlike feature space techniques, this approach is heavily dependent on a-priori knowledge of the pattern classes . Techniques for inferring automatically the class grammars from a

collection of data are not currently well developed. In consequence, the linguistic approach seems unsuitable for single scan analysis. The companion project previously mentioned, which is concerned with processing the delineation signals alone, is based on linguistic pattern recognition, and confronts the problem of grammar inference.

2.3 Feature Space Pattern Recognition

2.3.1 Basic Concepts (refs. 6, 7, 8)

Pattern recognition is concerned with two sets - a set of "objects" to be recognised, and a set of pattern classes to which the objects collectively belong. The process whereby an object is assigned to a particular class constitutes recognition of that object. Although formally incomplete, this definition is appropriate to the surface inspection problem. In this work, the object set consists of isolated scan sections, and the pattern classes are the defect classes.

Feature space techniques are concerned not with the objects themselves, but with points in a multi-dimensional space. Each point represents one or more objects, and the mapping from objects to points is via a set of tests or measurements. Each measurement must yield a real number, and must be applicable to any object in the object set. By associating each measurement with one co-ordinate of the space, the object-point mapping is defined (Fig. 2.3). Established terminology states that each measurement is of a single "feature" of the object and that the resulting space is therefore a "feature space". Similarly, the ordered set of numbers which results from applying the set of measurements to any object, is referred to as the "feature vector" for that object.

By this means, the object set is represented by a set of points in the feature space. Assuming that no two objects belonging to different classes map into the same point, there is associated with each point a single class name, i.e. the correct classification of the corresponding object (however that

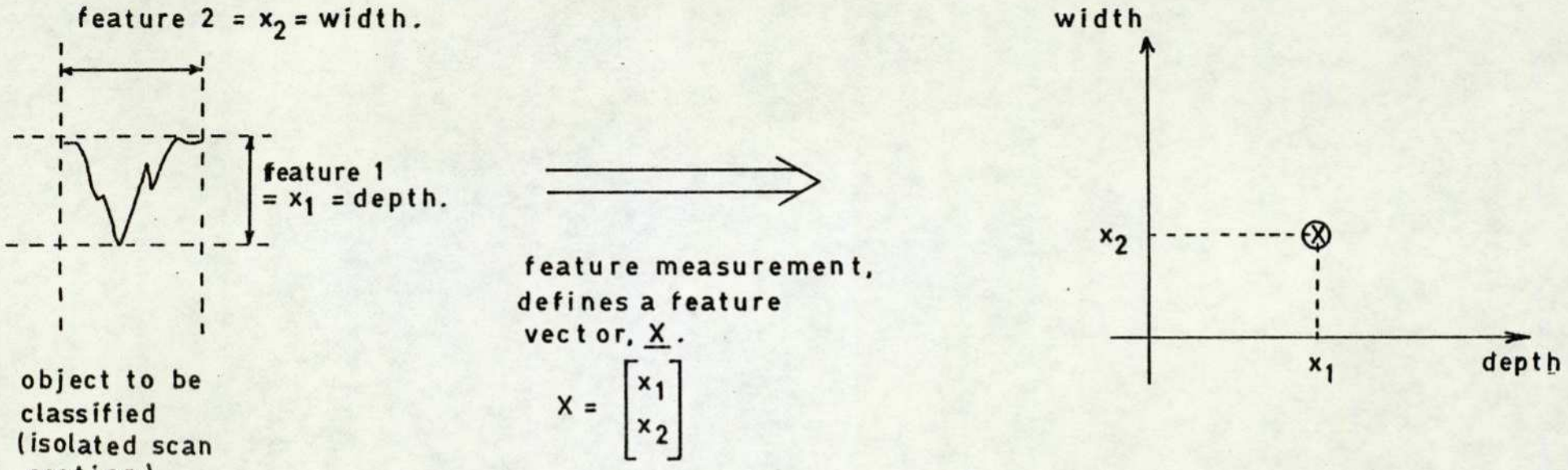


FIGURE 2.3 - THE FEATURE SPACE CONCEPT.

classification may be determined). In principle, therefore, there exists a partition of the space into mutually exclusive regions, such that each region contains points from only one class. This partition, together with a means for determining the region into which any object is mapped, is a complete solution to the recognition problem.

The practical design problem is to determine a partition which adequately approximates the true partition, given only a representative collection of objects from each class (a design set). These objects map into a finite number of points, each with an associated class name, and the design process must extrapolate from these to a suitable partition of the entire space (Fig. 2.4).

The literature on feature space techniques contains a vast selection of procedures for this process of extrapolation. They range from ad-hoc procedures based on a two- or three-dimensional picture of the space, to procedures firmly based on multivariate statistics. With the former, it is usually difficult or impossible to compare one technique with another, because of the lack of a common underlying theory. The designer must therefore base his selection on intuition, perhaps supported by empirical evaluations. Such a selection is difficult, time-consuming and suspect in its conclusions. In selecting techniques for this work, therefore, preference has been given to those having a sound theoretical basis. This basis is derived from statistical decision theory, and this will be discussed in the next section. Besides this, two further considerations have influenced the selection. The first was the potential for implementing the technique at reasonable cost and at high speed, as previously discussed. The second was the power of the technique to cope with complex distributions, in the feature space, for each class. Such distributions demand a complex partition of the space, in order adequately to separate the defect classes. Techniques which were strictly limited in this respect were therefore viewed with some scepticism. The assumption that complex distributions would arise, no matter what features were used, was based on three observations:

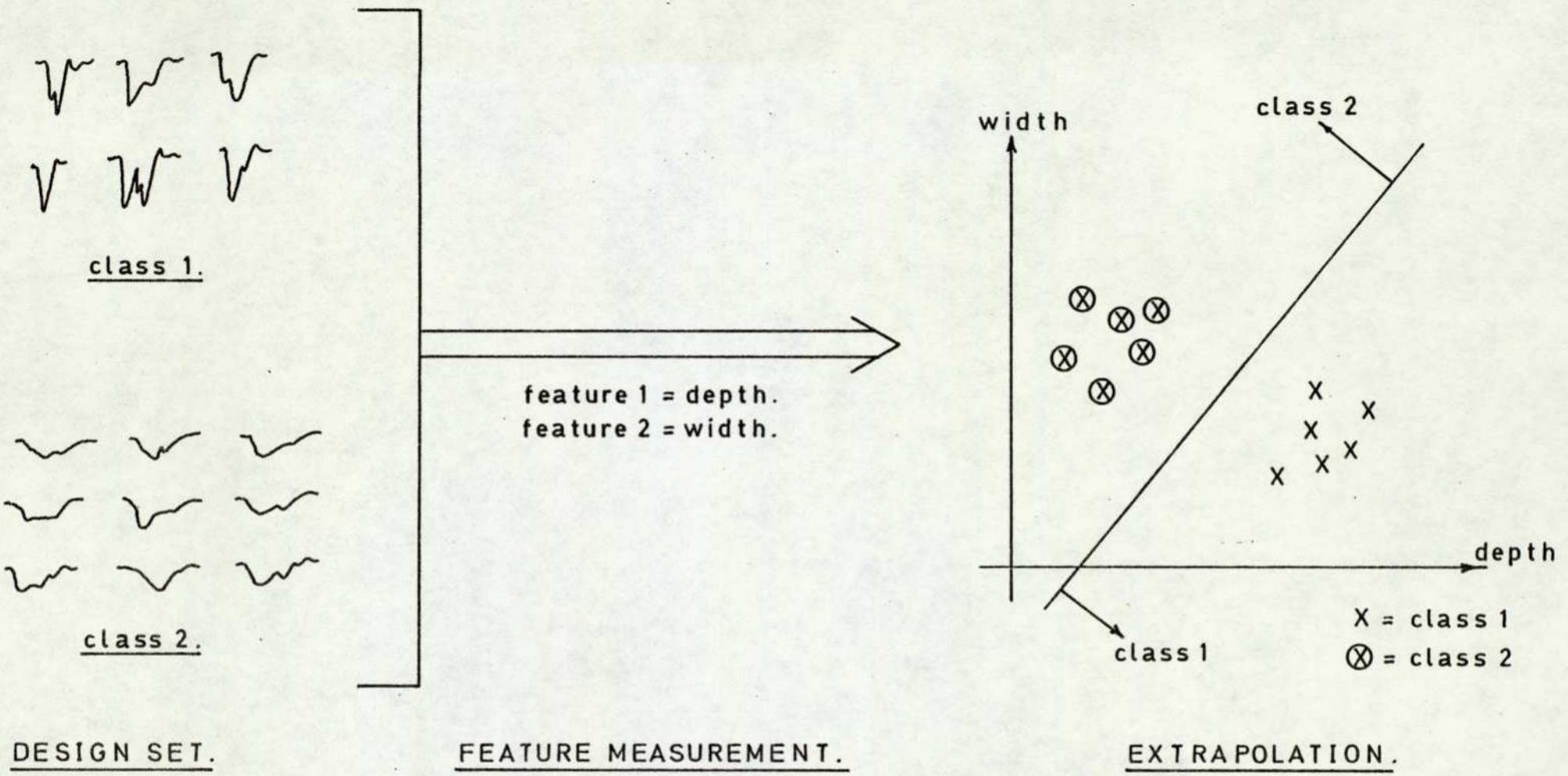


FIGURE 2.4 - EXTRAPOLATION FROM A DESIGN SET.

(1) There is very substantial variation in structure between defects from the same class. For example, there does not exist a "prototype scratch", from which all others can be derived with minor variations.

(2) The signal which results from a single scan across a defect will vary considerably with the location of the scan with respect to that defect. For example, if the defect is roughly circular, a scan which covers only a small arc of the circle will generate a signal quite different from one which passes over the centre.

(3) The characteristics of the scanning systems, discussed in Chapter 1, introduce further variation into the scan signals; for example, according to whether the defect is close to the centre of the strip, or close to the edge.

2.3.2 Statistical Decision Theory (ref. 9)

Statistical decision theory leads to the establishment of optimal decision strategies and rules. These tell us precisely what information must be made available for the decision process, and how that information should be used to reach a decision, given a set of feature values measured on the object to be classified. A decision rule is equivalent to a partition of the feature space, since both ultimately specify a decision to be associated with each point in the space.

Consider that a set of measurements has been chosen to define the mapping from objects to points in the feature space. Let an object be chosen at random from the set of all objects belonging to the j^{th} class, and let that object be characterised by the chosen measurements. This constitutes a single trial of a random experiment, in the framework of probability theory. The outcome (i.e. the set of feature values) is uncertain, but the set of all possible outcomes may be characterised by a probability distribution on the feature space. For continuous feature values, this is a probability density function (pdf),

and since the experiment is confined to objects from the j^{th} class, the pdf will be conditioned upon this restriction.

Let $\underline{X} = \left[x_1, x_2, -, x_d\right]^T$ be the (vector-valued) random variable defined by the experiment; so that x_1 is the result of the first measurement, x_2 the second, and so on.

Let $\{c_i;\ i = 1, 2, -, n_c\}$ denote the set of class labels. Then we may write the class-conditional pdf as $p(\underline{X}/c_j)$.

If the restriction to the j^{th} class is removed, so that an object is chosen at random from the complete object set, the experiment will be governed by the unconditional pdf $p(\underline{X})$. The law of total probability (ref. 10) allows us to write

$$p(\underline{X}) = \sum_{j=1}^{n_c} p(\underline{X}/c_j)\, P(c_j) \quad \ldots\ldots\ldots\ldots \quad (2.3.1)$$

where $P(c_j)$ is the a-priori probability that the object will be drawn from the j^{th} class.

Denote by $P(c_j/\underline{X})$ the <u>a posteriori</u> probability that the object <u>was</u> drawn from the j^{th} class, conditioned upon the measurement results. Bayes Rule (ref. 6) states

$$P(c_j/\underline{X}) = \frac{p(\underline{X}/c_j)P(c_j)}{p(\underline{X})} \quad \ldots\ldots\ldots\ldots \quad (2.3.2)$$

The set of a posteriori probabilities, $P(c_i/\underline{X})$, $i = 1, =, -, n_c$, defines a decision rule which minimises the probability of error. It is readily shown (ref. 6) that this rule is:

"Decide class c_i, such that

$$P(c_i/\underline{X}) \geq P(c_j/\underline{X}),\ j = 1, =, -, n_c"$$

- in other words, choose the most probable class, given the measurement results.

Two significant extensions can be made to this development. The first of these is to allow different classification errors to be weighted differently. To see the significance of this, consider that on tinplate there is a class of defects (arcing

spots) which can lead to the rupturing of tin cans under pressure. Superficially these defects appear as small black dots on the surface, and could be easily confused with relatively innocuous defects such as grease spots. Misclassification of a grease spot as an arcing spot would lead only to unnecessary re-inspection or rejection, whereas misclassification of an arcing spot as a grease spot would be considerably more serious. Minimum error rate classification would, of course, treat both errors equally.

Information on the relative significance of the various possible errors must be introduced in the form of a loss function, $\ell(c_i/c_j)$, $i, j = 1, 2, -, n_c$, where

$\ell(c_i/c_j)$ is the loss (cost/penalty) incurred by deciding class c_i, when c_j is the correct decision.

This loss function must be specified as part of the problem definition. With it, a decision rule can be defined to minimise the overall loss, averaged over the entire set of objects to be encompassed by the recognition process. It can be shown (ref. 6) that this decision rule is

"Decide c_i, such that $R(c_i/\underline{X}) \leq R(c_j/\underline{X})$, $j = 1, 2, -, n_c$"

where $R(c_i/\underline{X})$, $i = 1, 2, -, n_c$, is the expected loss (risk) associated with decision c_i, conditioned upon the measured feature vector, $\underline{X}$. This is given by

$$R(c_i/\underline{X}) = \sum_{j=1}^{n_c} \ell(c_i/c_j)P(c_j/\underline{X}).$$

The second extension is to allow decisions other than deciding class membership. For example, a useful decision might be to reject an object as un-classifiable, when the evidence is inconclusive. Such a rejection might be appropriate when the object lies close to a decision boundary in the feature space, so that a small change in the measured feature values would lead to a different classification.

This extension requires only an extended notation in the development already made.

Let $\{d_i, i = 1, =, -, n_d\}$ denote the set of possible decisions. (Usually n_d will be greater than n_c, with the decisions $d_i, i = 1, 2, -, n_c$ corresponding to the classes).

Then, as before, the optimum (Bayes) decision rule is (ref. 6):

"Decide d_i, such that $R(d_i/\underline{X}) \leqslant R(d_j/\underline{X})$, $j = 1, 2, -, n_d$"

$$\text{where } R(d_i/\underline{X}) = \sum_{j=1}^{n_c} \ell(d_i/c_j)P(c_j/\underline{X}) \quad \ldots\ldots\ldots\ldots \quad (2.3.3)$$

and $\ell(d_i/c_j)$ is the loss incurred in choosing decision d_i when the object belongs to class c_j.

Under the assumption that minimum expected loss is a suitable decision criterion, we therefore have the optimum decision rule defined in terms of the following quantities:

The loss function - $\ell(d_i/c_j)$, $i = 1, 2, -, n_d$
$j = 1, 2, -, n_c$.

The a-posteriori probabilities - $P(c_j/\underline{X})$,
$j = 1, 2, -, n_c$.

In turn, the latter may be calculated from the equation:

$$P(c_j/\underline{X}) = \frac{p(\underline{X}/c_j)P(c_j)}{p(\underline{X})}, \quad j = 1, 2, -, n_c.$$

Since the term $(p(\underline{X}))^{-1}$ appears multiplicatively in each conditional risk $(R(d_i/\underline{X}), i = 1, 2, -, n_d)$, it does not affect the decision processes and can be neglected. The loss function $(\ell(d_i/c_j), i = 1, 2, -, n_d;\ j = 1, 2, -, n_c)$, as mentioned, must be specified as part of the problem definition. The a-priori class probabilities $(P(c_i), i = 1, 2, -, n_c)$ can be estimated as relative frequencies in the design set, if this is representative in this respect. If not, they can be estimated by the designer or simply set equal if no class is to be preferentially treated in the decision process.

This leaves the class-conditional pdf's ($p(X/c_j)$, $j = 1, 2, -, n_c$) to be estimated from the design set. In essence, this is the extrapolation problem previously discussed. The design set gives a finite number of points in the space for each class, and the only certain inference is that $p(\underline{X}/c_j)$ is non-zero at these points. To go beyond this, it is necessary to make certain assumptions about the pdf's, $p(\underline{X}/c_j)$, $j = 1, 2, -, n_c$. For example, a very common assumption is that each one is multivariate Gaussian, so that the design set can be used to estimate the mean vector and the covariance matrix for each class. The assumptions on which any technique is based are the key factors in determining the power and generality of that technique. As previously discussed, the defect classes can be expected to generate fairly complex pdf's, perhaps with multiple modes within each class, and considerable overlap between classes. Accordingly, a technique has been sought for pdf estimation which can cope with these characteristics. Such a technique is that of Parzen estimation, and this will be discussed in the next section.

One further point must be made which is fundamental to all practical approaches to the estimation problem. This is the assumption that there exists a suitable distance measure, defined upon the feature space, such that, over most of the space, points which are close together belong to the same class. In most cases, this is assumed to be Euclidean distance. Although the basic assumption seems eminently reasonable, it is, in fact, an assumption that suitable features have been found for the problem at hand. As we shall see, it is not difficult to choose a feature set for which a suitable distance measure is extremely elusive.

2.3.3 Parzen Estimators

The following is a brief overview of Parzen estimators for pdf estimation, based primarily on refs. 11, 12, 13.

We shall assume that a design set is available which contains samples (objects) from each class of interest. Only those samples from class c_j will be used to estimate the class-conditional pdf

for that class, $p(\underline{X}/c_j)$, so that the estimation problem can be treated separately for each class.

Let there be N_j samples in the design set from class c_j, and let $\underline{X}_{ij}$ denote the feature vector for the i^{th} sample. The estimation problem, then, is to estimate $p(\underline{X}/c_j)$ from the data $\{\underline{X}_{ij}, i = 1, 2, -, N_j\}$. As discussed, each feature vector defines a point in the feature space for each sample. The estimation process is based upon the location, at each such point, of a "window function", which defines the contribution of that sample to the estimate, throughout the space. This is a scalar-valued function of a vector-valued argument, which we shall denote by $\gamma(\underline{X}, \underline{X}_{ij})$. Fig. 2.5 shows a typical window function in a two-dimensional space.

The pdf estimate at any point, $\underline{X}$, in the space, is formed as the average value of the complete set of window functions, evaluated at that point.

$$\hat{p}(\underline{X}/c_j) = \frac{1}{N_j} \sum_{i=1}^{N_j} \gamma(\underline{X}, \underline{X}_{ij}) \quad \ldots\ldots\ldots\ldots \quad (2.3.4)$$

where $\hat{p}(\underline{X}/c_j)$ denotes the estimate of $p(\underline{X}/c_j)$. Fig. 2.6 illustrates this estimate in a one-dimensional space. It can be seen that, with a suitable window function, the pdf estimate can assume high values in regions of the space where the design samples cluster, and low values elsewhere. This is, intuitively, a minimum requirement for pdf estimation.

In the work reported in this thesis, the multivariate, spherical gaussian distribution is used for the window function, so that:

$$\gamma(\underline{X}, \underline{X}_{ij}) = \frac{1}{(2\pi)^{d/2}\,\sigma^d} \exp\left[- \frac{(\underline{X} - \underline{X}_{ij})^T (\underline{X} - \underline{X}_{ij})}{2\sigma^2} \right] \ldots\ldots \quad (2.3.5)$$

where d is the dimensionality of the space.

Given that the true pdf, $p(\underline{X}/c_j)$, is continuous, it can be shown (ref. 13) that the estimate defined by equations (2.3.4) and (2.3.5) converges (in mean square) to the true pdf as the number (N_j) of design samples tends to infinity, provided that

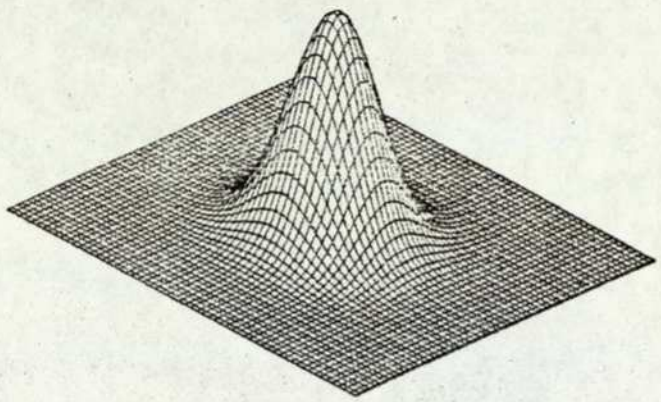


FIGURE 2.5 - A POTENTIAL FUNCTION OF TWO VARIABLES.

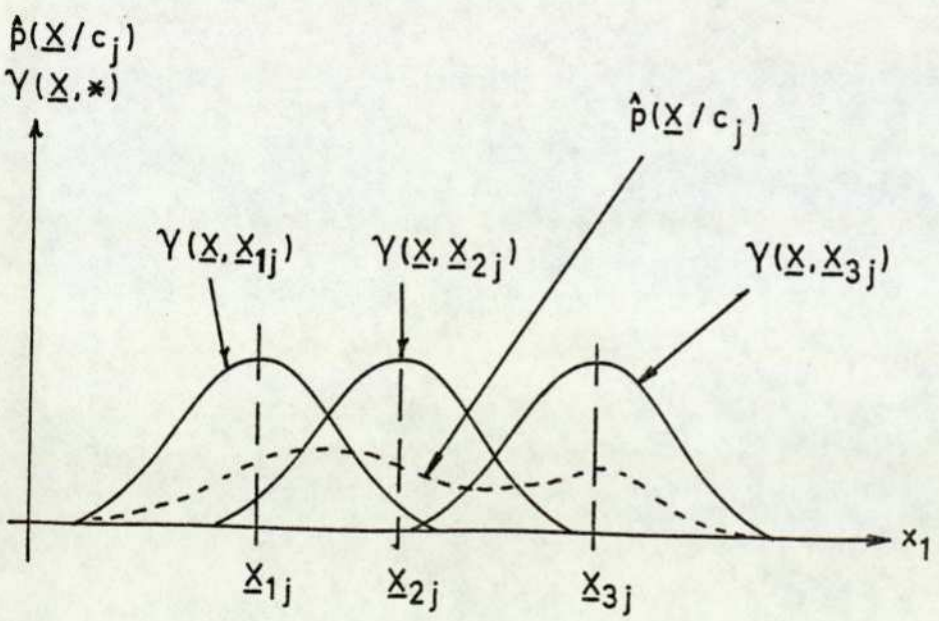


FIGURE 2.6 - A 1-DIMENSIONAL PARZEN ESTIMATOR.

σ varies with the number of samples so as to meet the following conditions:

(1) $\lim_{N_j \to \infty} \sigma^d = 0$

(2) $\lim_{N_j \to \infty} N_j \sigma^d = \infty$

These conditions state that σ^d (and therefore σ) must approach zero as N_j approaches infinity, but at a rate slower than $(N_j)^{-1}$.

The parameter σ is the standard deviation of the gaussian distribution, and therefore determines the "peakiness" of this window function. Figs. 2.7, 2.8 and 2.9 illustrate the effect of this parameter on the estimated pdf, in the practical situation of a finite number of design samples. In this situation, it is clear that the estimator embodies a wide range of quite different pdf estimates, according to the value assigned to this "smoothing parameter". Unfortunately, the theory can offer no guidance for this assignment, since the best value will be wholly dependent on the true pdf to be estimated.

The technique adopted in the work reported in this thesis is based upon the observation that the true pdf, when incorporated into the Bayes decision rule (equation (2.3.3)) would yield optimum performance. It is important to realise that "performance" here refers to performance over the complete object set. To estimate this performance, the so-called "leave-one-out" technique was used. This involves designing a classifier on the complete design set, less one member, and then using that design to classify the omitted member. This procedure is repeated for each member of the design set, so as to give a performance estimate based on the whole set. The smoothing parameter was adjusted so as to optimise that estimate. Any attempt to achieve this optimisation without omitting the member to be classified would fail, since perfect classification can always be achieved with a sufficiently small value of the smoothing parameter, when the member to be

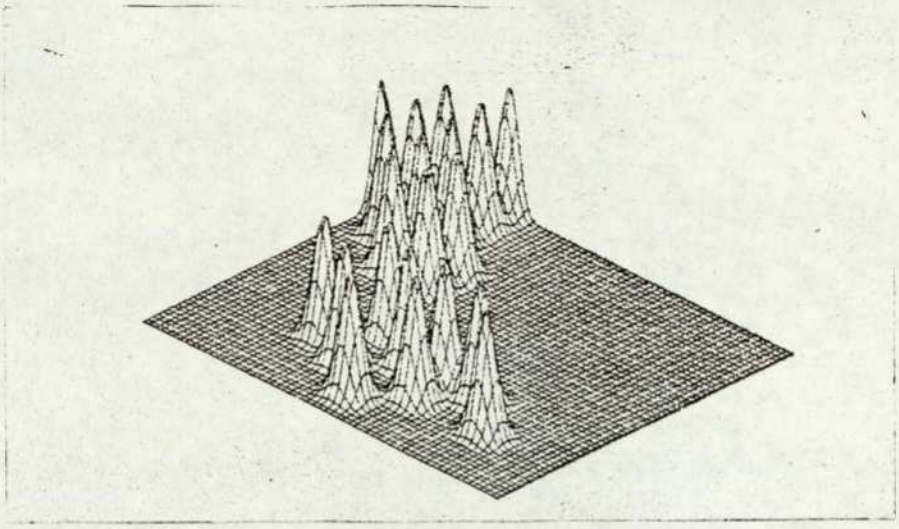


FIGURE 2.7 - A PARZEN ESTIMATE WITH LITTLE SMOOTHING.

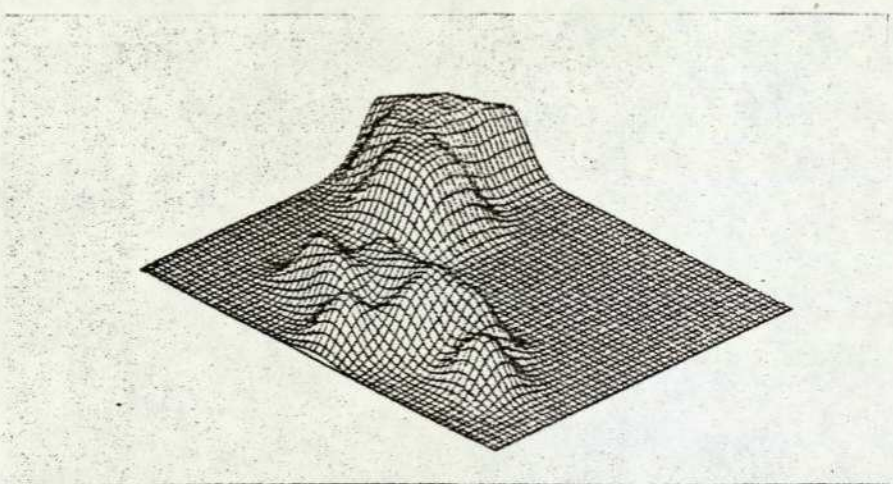


FIGURE 2.8 - WITH MORE SMOOTHING.

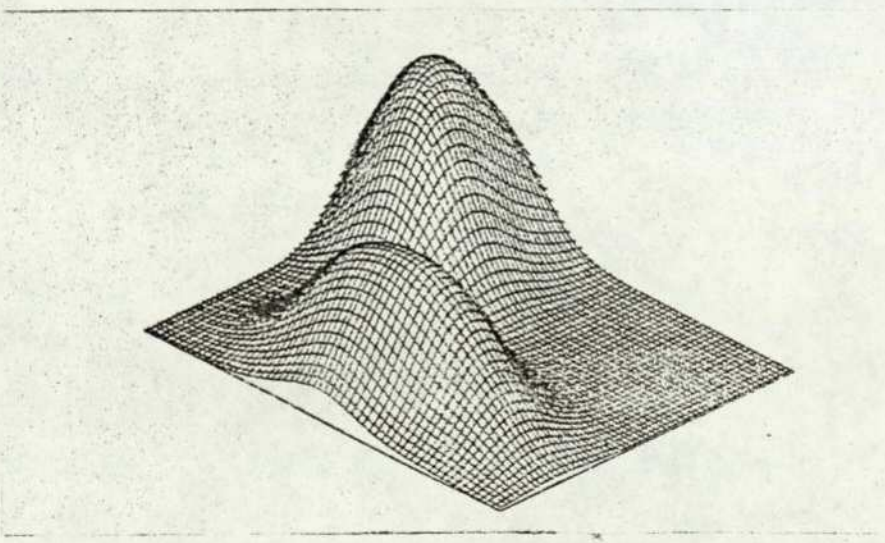


FIGURE 2.9 - WITH STILL MORE SMOOTHING.

(Figs. 2.7, 2.8, 2.9 from ref. 7)

classified contributes to the estimate (ref. 12).

2.3.4 The Polynomial Discriminant Method

The Parzen estimator, presented in the previous section, requires that a summation be performed over all members of the design set from a given class to arrive at the pdf estimate, for that class, at any point in the space (equation (2.3.4)). For Bayes decision rule (equation (2.3.3)) the estimate must be calculated for each class. In its basic form, therefore, the Parzen estimator requires storage of the complete design set and substantial computation to reach a decision. Both of these requirements clash with the system requirements of reasonable cost and fast processing. However, for a particular class of window functions, the estimator can be expressed in such a way as to eliminate the need to store the design set, and possibly greatly to reduce the on-line computational load.

We have (equation (2.3.4)):

$$\hat{p}(\underline{X}/c_j) = \frac{1}{N_j} \sum_{i=1}^{N_j} \gamma(\underline{X}, \underline{X}_{ij})$$

Let $\gamma(\underline{X}, \underline{X}_{ij})$ be expressible in the following form:

$$\gamma(\underline{X}, \underline{X}_{ij}) = \sum_{k=1}^{K} a_k \phi_k(\underline{X}) \psi_k(\underline{X}_{ij}) \quad \ldots\ldots\ldots \quad (2.3.6)$$

Substituting this into equation (2.3.4) gives:

$$\hat{p}(\underline{X}/c_j) = \sum_{k=1}^{K} b_k \phi_k(\underline{X}) \quad \ldots\ldots\ldots\ldots\ldots \quad (2.3.7)$$

where:

$$b_k = \frac{a_k}{N_j} \sum_{i=1}^{N_j} \psi_k(\underline{X}_{ij}) \quad \ldots\ldots\ldots\ldots\ldots \quad (2.3.8)$$

With this form of the estimator, the set of coefficients (b_k, k = 1, 2, -, K) can be calculated from the design set (for each class) and stored, instead of storing the design set itself. If K can be made much less than N_j, this will be a significant saving. Similarly, the on-line computational load is limited to the evaluation of equation (2.3.7) for each class.

If the set of functions ($\phi_k(\underline{X})$, k = 1, 2, -, K) is significantly easier to evaluate than N_j window functions ($\gamma(\underline{X}, \underline{X}_{ij})$, i = 1, 2, -, N_j), this can lead to further savings.

Specht (ref. 14) has suggested expanding $\gamma(\underline{X}, \underline{X}_{ij})$ of equation (2.3.5), via a Taylor's series expansion, to yield a polynomial representation of this window function. Such a representation is, of course, of the form defined by equation (2.3.6). The final expression which results is therefore of the form defined by equations (2.3.7) and (2.3.8), with K being the number of polynomial terms retained in the expansion. Specifically, the relevant equations are as follows:

$$\hat{p}(\underline{X}/c_j) = \frac{1}{\sigma^d (2\pi)^{d/2}} \cdot \exp\left(-\frac{\underline{X}^T\underline{X}}{2\sigma^2}\right) \cdot P^{(j)}(\underline{X}) \quad \ldots \quad (2.3.9)$$

where: $\underline{X} = [x_1, x_2, -, x_d]^T$

$$P^{(j)}(\underline{X}) = D^{(j)}_{0---0} + D^{(j)}_{10---0} \cdot x_1 + D^{(j)}_{010---0} \cdot x_2$$

$$+ \text{----} + D^{(j)}_{0---01} \cdot x_d$$

$$+ D^{(j)}_{20---0} \cdot x_1^2 + D^{(j)}_{110---0} \cdot x_1 \cdot x_2$$

$$+ \text{----} + D^{(j)}_{k_1 k_2 -- k_d} \cdot x_1^{k_1} \cdot x_2^{k_2} \text{-----} x_d^{k_d}$$

$$+ \text{----------}$$

and: $$D^{(j)}_{k_1 k_2 \text{----} k_d} = \frac{1}{k_1!\, k_2! \text{ --- } k_d!\, \sigma^{2h}} \cdot \frac{1}{N_j}$$

$$\cdot \sum_{i=1}^{N_j} x_{ij1}^{k_1} \cdot x_{ij2}^{k_2} \text{ --- } x_{ijd}^{k_d} \cdot \exp\left(-\frac{\underline{X}_{ij}^T \underline{X}_{ij}}{2\sigma^2}\right)$$

where: $h = k_1 + k_2 + \text{----} + k_d$

$$\underline{X}_{ij} = [x_{ij1}, x_{ij2}, \text{----}, x_{ijd}]^T$$

In later chapters, these equations will be referred to as the "polynomial form" of Specht's classifier, with equations (2.3.4) and (2.3.5) referred to as the "exponential form". It should be noted that the expression

$$\frac{1}{\sigma^d (2\pi)^{d/2}} \cdot \exp\left(-\frac{\underline{X}^T\underline{X}}{2\sigma^2}\right)$$

in equation (2.3.9) is, in fact, redundant, since it is class-independent and will therefore appear multiplicatively in each conditional risk (equations (2.3.3) and (2.3.2)). It will therefore have no effect upon the decision strategy, and can be ignored. This leaves $\hat{p}(\underline{X}/c_j)$ of equation (2.3.9) as a pure polynomial, $P^{(j)}(\underline{X})$.

The following theorems (ref. [14]) lend credibility to this classifier:

(1) If the magnitudes of the feature vectors from the design set $\{\underline{X}_{ij}, i = 1, 2, -, N_j\}$ are finite, and $\sigma > 0$, then each polynomial coefficient, $D^{(j)}_{k_1 k_2 \text{----} k_d}$, approaches zero as the degree, h, of the term approaches infinity.

(2) In the two-class problem ($n_c = 2$) Bayes decision rule (equations (2.3,2), (2.3.3))with the estimator of equations (2.3.4) and (2.3.5), becomes the nearest-neighbour decision rule as $\sigma \to 0$. (The nearest-neighbour rule classifies an unknown point in the space as being the same as that of the point from the design set for which the Euclidean distance to the unknown point is the smallest. It has been shown that the error rate for the nearest neighbour rule, as the number of design samples tends to infinity, is never greater than twice the Bayes error rate, using the true class-conditional pdf's - ref. 15).

(3) In the two-class problem, Bayes decision rule (equations (2.3.2), (2.3.3)) with the estimator of equations

(2.3.4) and (2.3.5), yields a hyperplane (linear) decision boundary as $\sigma \to \infty$, which is perpendicular to the vector $[\underline{\mu}_1 - \underline{\mu}_2]$, where $\underline{\mu}_j$ is the sample mean vector of class c_j, calculated from the design set.

To quote from ref. 14 , " ..... the polynomial discriminant method ..... converges to suboptimal but acceptable methods, even for the extreme cases of $\sigma \to \infty$ and $\sigma \to 0$. Similarly, the separating boundary ranges from strictly linear for $\sigma \to \infty$ to highly non-linear for $\sigma \to 0$.". Furthermore, the method demands only very mild assumptions on the underlying distributions, and should cope well with such difficulties as multi-modality within each class.

2.3.5 Linear Classifiers and Φ Machines

Any partition of a feature space into n_d mutually exclusive regions can be defined via a set of "discriminant functions":

$$g_i(\underline{X}) \; ; \quad i = 1, 2, -, n_d$$

such that:

$$g_j(\underline{X}) = \max. \; g_i(\underline{X}) \quad \ldots\ldots\ldots\ldots\ldots\ldots \quad (2.3.10)$$

for all $\underline{X}$ in the j^{th} region, R_j.

This form of definition arises naturally out of the Bayes decision strategy (equation (2.3.3)), for which

$$g_i(\underline{X}) = - R(d_i/\underline{X}).$$

In a linear classifier, the discriminant functions are constrained to be linear, with one for each class:

$$g_i(\underline{X}) = \underline{w}_i^T \cdot \underline{X} + w_{i0} \; ; \quad i = 1, 2, -, n_c \quad (2.3.11)$$

where $\underline{w}_i = [w_{i1}, w_{i2}, -, w_{id}]^T$

is a "weight vector" for class c_i.

The decision boundary between two contiguous regions R_i and R_j is defined by:

$$g_i(\underline{X}) = g_j(\underline{X})$$

and, for a linear classifier, this becomes:

$$(\underline{w}_i - \underline{w}_j)^T \cdot \underline{X} + (w_{i0} - w_{j0}) = 0.$$

This surface is also linear, and in a multidimensional space, is known as a "hyperplane". It is worth noting that it is the differences between weight vectors which determine the separating boundaries, rather than the weight vectors themselves.

It can also be shown (ref. 16) that the decision regions $\{R_i, i = 1, 2, -, n_c\}$ are each strictly convex. Thus, for example, multimodal class distributions, which demand multiple, disconnected regions for a single class, cannot be accommodated by a linear classifier. Fig. 2.10 shows a two-dimensional example of linear classification. The problem of extrapolating from a design set to an exhaustive partition of the space, previously discussed, becomes that of estimating the set of weight vectors and constant terms, $\{\underline{w}_i, w_{i0} ; i = 1, 2, -, n_c\}$, from the design set, and we shall shortly discuss the techniques available for this.

It is clear that linear classifiers are severely limited in power, no matter how the weights may be determined. In view of the discussion in previous sections, it is reasonable to question their relevance to this project. In this light two factors need to be considered.

First, the linear classifier can be implemented with very simple hardware. For example, if the feature values are available as a set of analogue voltages, the equation (2.3.11) can be implemented by the simple operational amplifier scheme shown in Fig. 2.11. In this case, the complete classifier would consist of just one set (n_c) of these, followed by a simple maximum value detector (equation (2.3.10)). Similarly, such an implementation could be made to operate at typical on-line processing rates with ease.

Second, the basic structure of the linear classifier can be extended to include classifiers of, in principle, unlimited power. These are the so-called "Φ machines" (ref. 16) defined

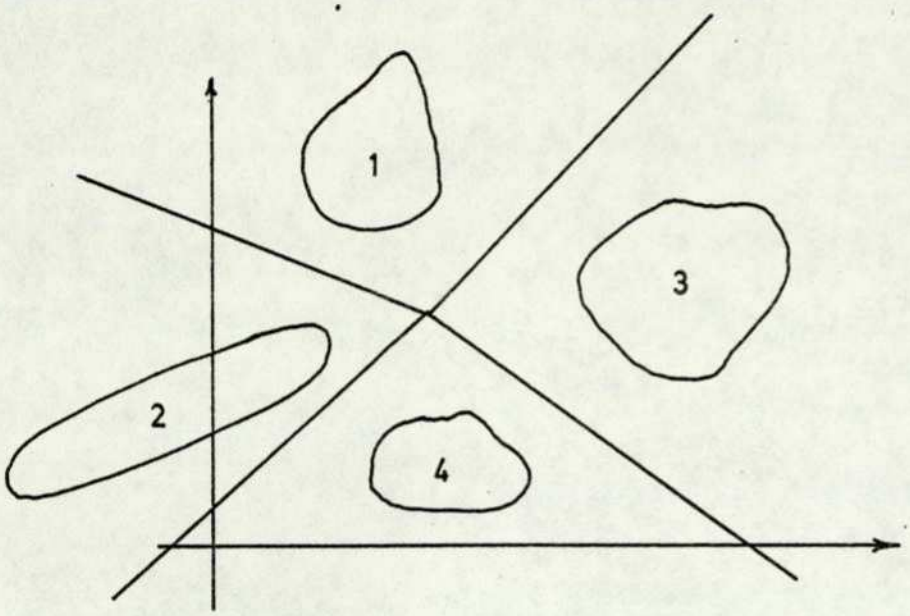


FIGURE 2.10 - LINEAR CLASSIFICATION.

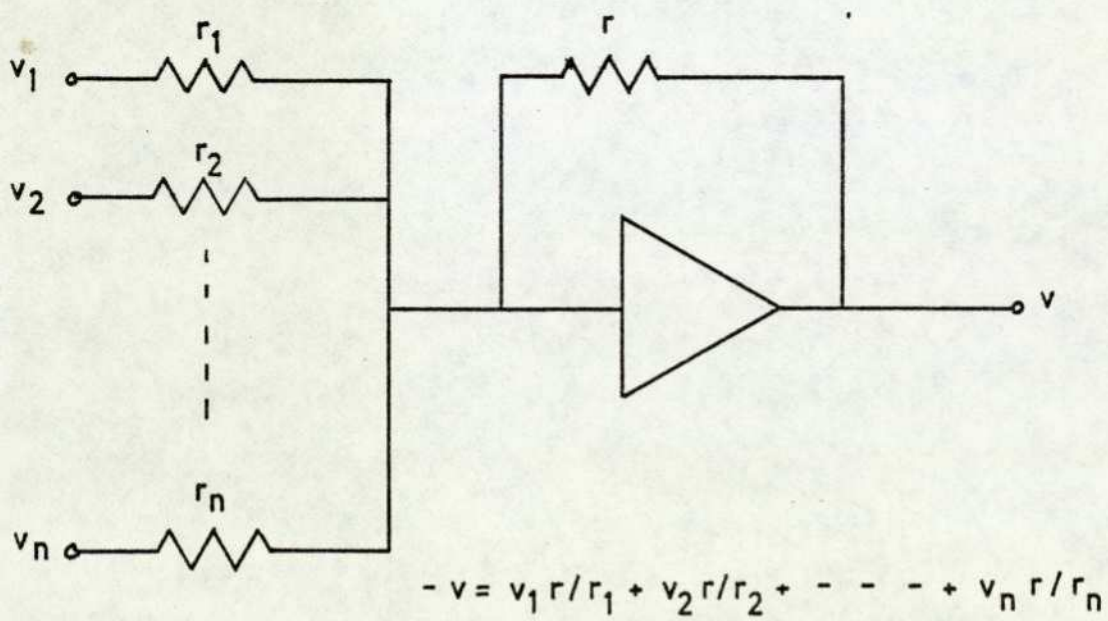


FIGURE 2.11 - LINEAR WEIGHTING HARDWARE.

by the following discriminant functions:

$$g_i(\underline{X}) = \underline{w}_i^T \cdot \underline{\Phi}(\underline{X}) + w_{i0} \; ; \quad i = 1, 2, -, n_c,$$

where

$$\underline{w}_i = \left[w_{i1}, w_{i2}, -, w_{ik}\right]^T$$

$$\underline{\Phi}(\underline{X}) = \left[\phi_1(\underline{X}), \phi_2(\underline{X}), -, \phi_k(\underline{X})\right].$$

In other words, each discriminant function is a linear combination of a set of k functions of the basic features. A typical example might be the following:

$$\underline{X} = \left[x_1, x_2\right]^T$$

$$\phi_1(\underline{X}) = x_1^2$$

$$\phi_2(\underline{X}) = x_1 x_2$$

$$\phi_3(\underline{X}) = x_2^2$$

so that $g_i(\underline{X}) = w_{i1} x_1^2 + w_{i2} x_1 x_2 + w_{i3} x_2^2 + w_{i0}$,

i.e. a general second-order polynomial.

The assumption underlying the Φ machine is that a set of functions of the basic features can be found, such that the classes can be adequately separated by a linear machine in the function space. The problem,of course, is to determine a suitable set of functions.

Finally, attention was naturally directed towards linear machines in the course of the simulation work with recorded data, and this will be discussed in Chapters 3 and 4.

2.3.5.1 Determining a Set of Weights

Many procedures have been proposed to determine the set of weights for a linear machine from a design set. Before discussing these, the notation will be simplified as follows:

We have (equation (2.3.11))

$$g_i(\underline{X}) = \underline{w}_i^T \cdot \underline{X} + w_{i0} \; ; \quad i = 1, 2, -, n_c,$$

where $\underline{w}_i = [w_{i1}, w_{i2}, -, w_{id}]^T$.

Define $\underline{Y} = [1, \underline{X}]^T$

$= [1, x_1, x_2, -, x_d]^T$

and $\underline{W}_i = [w_{i0}, \underline{w}_i]^T$

$= [w_{i0}, w_{i1}, w_{i2}, -, w_{id}]^T$

Then equation (2.3.11) can be written as:

$$g_i(\underline{X}) = \underline{W}_i^T \cdot \underline{Y} \; ; \quad i = 1, 2, -, n_c \quad \ldots\ldots\ldots \quad (2.3.12)$$

The design problem, then, is to determine the set of (augmented) weight vectors $\{\underline{W}_i \; ; \; i = 1, 2, -, n_c\}$ from a design set. Procedures for this will be discussed under two main headings: Error correction procedures and Least Mean Square Error procedures.

Error correction procedures (refs. 6, 7, 16)

These begin with an arbitrary set of weight vectors and examine each member of the design set in turn. If that member is correctly classified by the current set of weight vectors, so that:

$$\underline{W}_j^T \cdot \underline{Y} = \max_i \underline{W}_i^T \cdot \underline{Y} \quad \text{for } \underline{Y} \text{ from class } c_j,$$

then no further action is taken, and attention passes to the next member. If, on the other hand, the member is incorrectly classified, so that:

$$\underline{W}_\ell^T \cdot \underline{Y} = \max_i \underline{W}_i^T \cdot \underline{Y} \text{ for } \underline{Y} \text{ from class } c_j \text{ and } \ell \neq j,$$

then either $\underline{W}_j$ is increased, or $\underline{W}_\ell$ is decreased, or both. Typically, the changes would be:

$$\underline{W}_i \leftarrow \underline{W}_i + \rho \underline{Y}$$

or $$\underline{W}_\ell \leftarrow \underline{W}_\ell - \rho \underline{Y}$$

or both, where ρ is scalar (and may vary from member to member).

Attention then passes to the next member, and so on. When the design set is exhausted, and if corrections have been made for any member, the cycle is repeated. The procedure terminates if all members of the design set are correctly classified by the current set of weight vectors. Such procedures have been justified both intuitively, and as steepest-descent routines for minimising a suitable error criterion. In the latter case, the criterion would be such as to assume zero value for any set of weight vectors yielding zero error rate.

If the design set is such that a set of weight vectors exists which will yield zero error rate, the design set is said to be "linearly separable". In this case, it can be shown (ref. 6) that the error-correction procedures will converge to a zero error rate solution, in a finite number of iterations. It is not possible, however, to place a useful upper bound on this number. If the design set is not linearly separable, the error-correction procedures will not converge. Instead, the weight vectors will oscillate continuously as a solution is sought, not necessarily about the minimum error rate solution. In this case, the procedure must be terminated arbitrarily. For this project, it seems likely that a representative design set will not be linearly separable, and this is the primary reason why error-correction procedures have not been used. There are other reasons, however.

First, these procedures will terminate with <u>any</u> zero error rate solution, without necessarily finding the best. For example, it can be shown (ref. 6) that for two identically distributed, spherical gaussian classes, the optimal separating surface is that hyperplane which perpendicularly

bisects the line joining the two means. This hyperplane will yield optimum performance on unseen data. Error-correction procedures may well terminate with a quite different solution, as shown in Fig. 2.12.

Secondly, these procedures are iterative, rather than direct, and may therefore prove somewhat expensive in computation. Whilst this would not normally pose any serious difficulties, since a large computer can be used in the design stage, we shall see that a technique is to be used for feature selection which requires a succession of designs to be carried out, so as to evaluate different possible feature sets. As many as 2.10^5 separate designs will be required for some selection cycles, and it is therefore prudent to require a reasonably fast design procedure.

Least Mean Square Error procedures (refs. 6, 17, 18, 19)

In contrast with error-correction procedures, least mean square error (LMSE) procedures do not guarantee a zero error rate solution, even if one exists. Instead, they offer a reasonable compromise solution in both separable and non-separable cases. In addition, we shall see that their theoretical relationship to the Bayes discriminants can be closely defined.

We have a design set, comprising a total of N (augmented) feature vectors:

$$\{\underline{Y}_{ij} \; ; \; i = 1, 2, -, N_j \; ; \; j = 1, 2, -, n_c\}$$

so that $\underline{Y}_{ij}$ is the i^{th} vector from the j^{th} class:

$$\underline{Y}_{ij} = \left[y_{ij0}, y_{ij1}, - , y_{ijd}\right]^T$$

with $y_{ij0} = 1$ for all i, j,

and $$N = \sum_{j=1}^{n_c} N_j .$$

We shall represent the complete design set as a matrix of dimension (N x d*), where d* = d + 1, as follows:

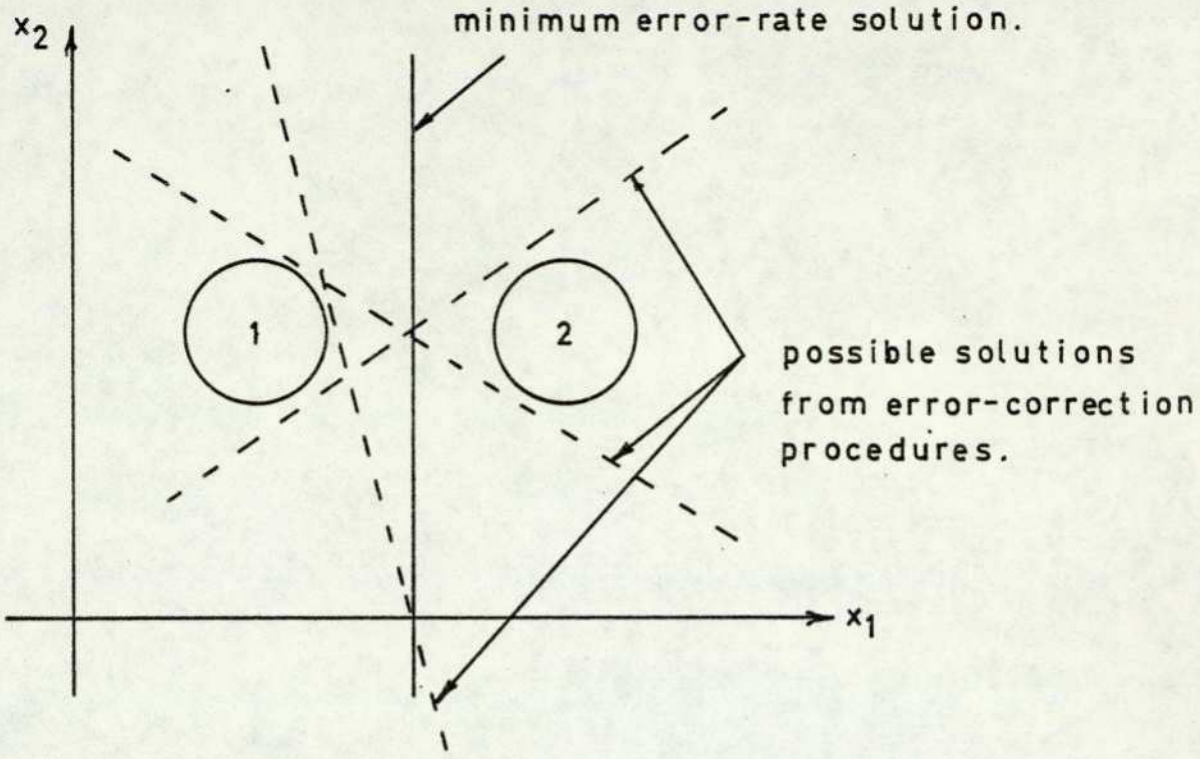
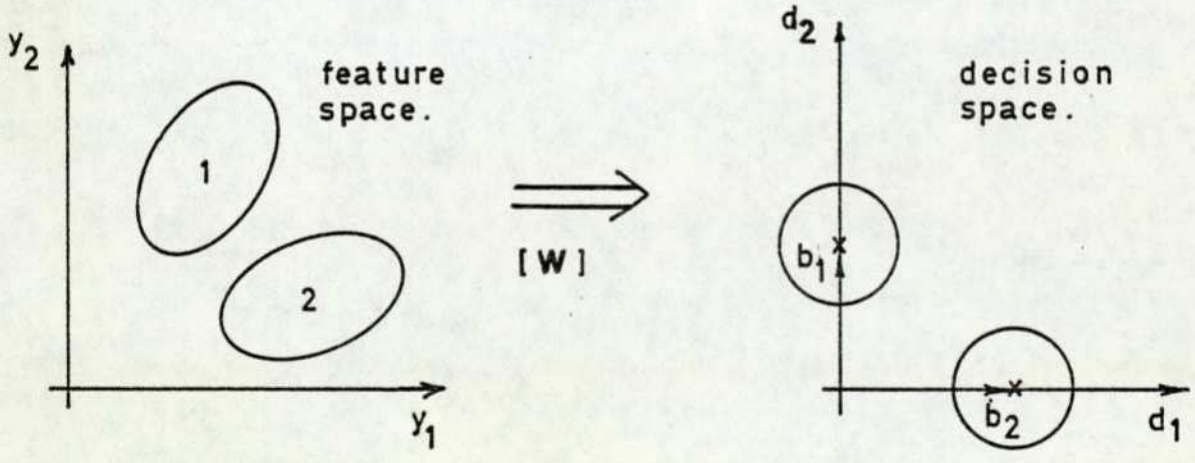


FIGURE 2.12 - ERROR-CORRECTION SOLUTIONS.



$b_1 = [0,1]^t$

$b_2 = [1,0]^t$

FIGURE 2.13 - THE LEAST-MEAN-SQUARE TRANSFORMATION.

$$A = \begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \\ \vdots \end{bmatrix}$$

where Y_j is a matrix of dimension (N_j x d*) containing the vectors from the j^{th} class:

$$Y_j = \begin{bmatrix} \underline{Y}_{1j}^T \\ \underline{Y}_{2j}^T \\ \underline{Y}_{3j}^T \\ \vdots \end{bmatrix} \quad ; \quad j = 1, 2, -, n_c$$

We shall define a matrix B, of dimension (N x n_c), as follows:

$$B = \begin{bmatrix} B_1 \\ B_2 \\ B_3 \\ \vdots \end{bmatrix}$$

where B_j is a matrix of dimension (N_j x n_c) consisting of N_j

identical rows:

$$B_j = \begin{bmatrix} \underline{b}_j^T \\ \underline{b}_j^T \\ \underline{b}_j^T \\ \vdots \end{bmatrix} \quad ; \quad j = 1, 2, -, n_c$$

where $\underline{b}_j = [b_{j1}, b_{j2}, -, b_{jn_c}]^T$.

Similarly, we shall represent the complete set of weight vectors as a matrix, of dimension ($d^* \times n_c$)

$$W = [\underline{W}_1, \underline{W}_2, \underline{W}_3, \text{-------}]$$

where $\underline{W}_j$ is the weight vector for the j^{th} class (equation (2.3.12)):

$$\underline{W}_j = [w_{j0}, w_{j1}, w_{j2}, -, w_{jd}]^T \quad ; \quad j = 1, 2, -, n_c.$$

Finally, the weight matrix will be determined so as to transform the A matrix into the B matrix, with least mean square error:

Determine W to minimise

$$\| AW - B \|$$

where $\| \cdot \|$ is the norm of a matrix, defined as

$$\| C \| = [\text{trace } C^T C]^{\frac{1}{2}} .$$

It can be shown (ref. 17) that the solution is given by:

$$W = (A^T A)^{-1} A^T B \quad \ldots\ldots\ldots\ldots\ldots\ldots \quad (2.3.13)$$

when $(A^T A)^{-1}$ exists.

This procedure requires some explanation. In essence, the weight matrix, W, transforms the set of design samples from the feature space into a "decision space" of dimension n_c. In this decision space there is a set of n_c "decision points", defined by the vectors

$$\underline{b}_j, \; j = 1, 2, -, n_c.$$

The transformation is such that each design sample from the j^{th} class is transformed into the decision point for that class, with least mean square error over all samples. This is illustrated in Fig. 2.13. Given such a transformation, a logical decision procedure for an unseen sample is to transform the feature vector into the decision space, and to determine the closest decision point. The reader may satisfy himself that, when $b_{jk} \alpha \delta_{jk}$ (as in Fig. 2.13), this is equivalent to determining the largest component of the transformed vector:

$$\underline{d}^T = [d_1, d_2, ----] = \underline{Y}^T W$$

Since the columns of W are the required weight vectors, this is the decision rule of equations (2.3.10) and (2.3.12).

A particular choice for the vectors $\underline{b}_j$; $j = 1, 2, -, n_c$ links this scheme to the Bayes strategy.

Let $b_{jk} = - \ell (c_k/c_j)$; $j, k = 1, 2, -, n_c$,

where $\ell(c_k/c_j)$ is the loss incurred by deciding class c_k, when c_j is the correct decision. Then it can be shown (ref. 17) that the discriminant functions $\underline{Y}^T \underline{W}_i$ from the LMSE procedure provide a minimum mean square error approximation to the optimal Bayes discriminants

$$g_i(\underline{X}) = -R(c_i/\underline{X})$$

$$= -\sum_{j=1}^{n_c} \ell(c_i/c_j)\, P\,(c_j/\underline{X})$$

- regardless of the underlying distributions, as the number of design samples tends to infinity.

This procedure therefore has both intuitive appeal, as well as a reassuring theoretical basis. Unlike the error-correction procedures, it incorporates the various error penalties assigned by the designer. Further, it yields the closed-form solution of equation (2.3.13), with acceptable computational demands.

This approach has one further advantage when a "reject" decision is to be incorporated. With error-correction procedures, such a decision can be incorporated only arbitrarily - for example, by rejecting a sample whenever the largest discriminant is not sufficiently dominant. This approach has also been taken with the LMSE procedure (e.g. ref. 20), but it is difficult to apply when unequal error costs are to be used. For this project, the basic technique has been extended, in a more consistent way, as follows:

Let: $p(\underline{Y}) = \sum_{j=1}^{n_c} p(\underline{Y}/c_j)P(c_j)$

$$\underline{F} = \left[P(c_1)p(\underline{Y}/c_1),\ P(c_2)p(\underline{Y}/c_2),\ \text{-------}\right.$$

$$\left.\text{-------},\ P(c_{n_c})p(\underline{Y}/c_{n_c})\right]^T$$

$$\underline{L}_i = \left[\ell(c_i/c_1),\ \ell(c_i/c_2),\ \ell(c_i/c_3)\ \text{------}\right.$$

$$\left.\text{-------},\ \ell(c_i/c_{nc})\right]^T;\quad i = 1,\ 2,\ -,\ n_c$$

$$L = \left[\underline{L}_1,\ \underline{L}_2,\ \underline{L}_3,\ -\ ,\ \underline{L}_{n_c}\right]$$

Then the set of Bayes discriminants (without indecision) can be written as:

$$\underline{D}^T = -\left[d_1, d_2, d_3, -, d_{n_c}\right]$$

$$= \frac{1}{p(\underline{Y})} \cdot F^T \cdot L$$

i.e. $$d_i = -R(c_i/\underline{Y})$$

$$= -\sum_{j=1}^{n_c} \ell(c_i/c_j) \frac{p(\underline{Y}/c_j)P(c_j)}{p(\underline{Y})}$$

Similarly, if a reject decision is to be allowed, we must add one more column to the matrix L, to give

$$L^* = \left[\underline{L}_1, \underline{L}_2, -, \underline{L}_{n_c}, \underline{L}_0\right]$$

$$= \left[L, \underline{L}_0\right]$$

where $$\underline{L}_0 = \left[\ell(c_0/c_1), \ell(c_0/c_2), -, \ell(c_0/c_{n_c})\right]^T$$

and c_0 denotes the reject decision.

We can therefore write the Bayes discriminant for the reject decision as

$$d_0 = -\frac{1}{p(\underline{Y})} \cdot F^T . \underline{L}_0 .$$

The LMSE procedure yields a set of discriminants with minimum mean square error from the Bayes discriminants, without a reject option.

Thus $\underline{Y}^T W = \underline{D}^T$ with minimum mean square error

therefore

$$\underline{Y}^T W = \frac{1}{p(\underline{Y})} \cdot F^T \cdot L$$

therefore

$$\frac{1}{p(\underline{Y})} F^T = \underline{Y}^T . W . L^{-1}$$

therefore

$$d_0 = \frac{1}{p(\underline{Y})} \cdot F^T \cdot \underline{L}_0 = \underline{Y}^T . W . L^{-1} . \underline{L}_0$$

Therefore, the required weight vector for a reject decision is simply:

$$\underline{W}_0 = W \cdot L^{-1} \cdot \underline{L}_0 .$$

This makes use of the complete loss function, as required.

Various extensions to the basic LMSE procedure have been proposed. In particular, the Ho-Kashyap extensions (ref. 21) combine the basic procedure with iterative adjustment of the B matrix, so as to produce a guaranteed zero error rate solution, if one exists. However, the performance on non-separable data is then less well defined, and in particular, minimum mean square error from the Bayes discriminants (in the limit) is sacrificed.

2.3.6 Feature Normalisation and Selection

The discussion so far has assumed that a set of features has been chosen, suitable for the problem at hand, and that a corresponding set of measurement procedures has been defined to yield numerical values of those features on any sample from the object set. In practice, the problem of finding a suitable set of features is both crucially important and, usually, very difficult. It is possible for the designer to specify a set of "candidate" features and to use a computer program to sift through this candidate set, so as to select a good subset. This allows the designer considerable freedom in specifying the candidate set, since he need not be too concerned with the possibility of including features which contribute more to inter-class confusion, than to class separation. Such features should be rejected by the selection procedure. We shall discuss the problems involved in subset selection, and the methods which have been proposed to overcome them.

Feature normalisation is a necessary pre-requisite for both feature selection and classifier design. It is, essentially, a

final stage in the measurement procedures and is intended to yield numerical values for each feature which are mutually compatible, and suited to the particular classifier to be used.

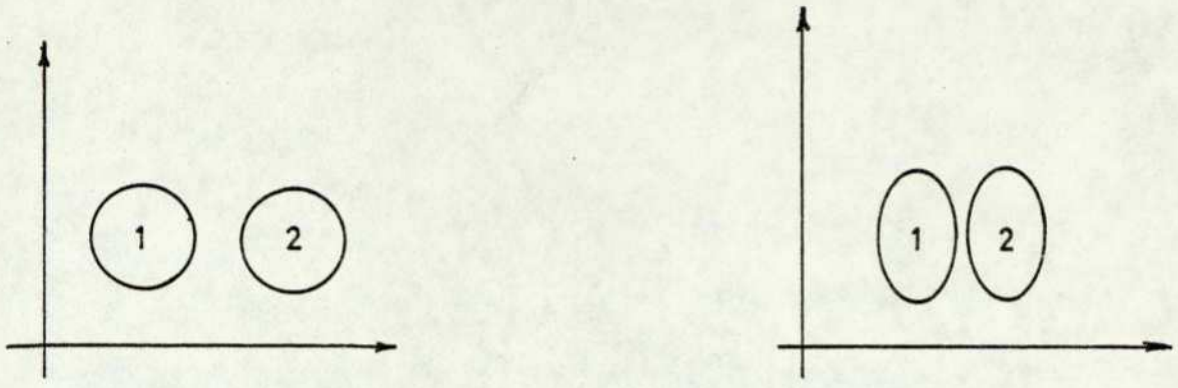
2.3.6.1 Feature Normalisation

We have seen that feature space classifiers make use of a distance measure defined on the space, usually Euclidean distance. If one feature yields numerical values orders of magnitude larger than other features, then this feature will dominate the distance calculations. This is a primary reason why features must be normalised for compatibility.

Almost invariably, features will be first normalised to have zero mean value over the design set. This is a sensible precaution for any classifier, but has special significance for Specht's polynomial discriminant method. The Taylor series expansions used to derive the polynomials are expansions about the origin of the space, and their approximation accuracy is therefore highest close to the origin. It is therefore desirable to locate the origin in the region of highest sample density, as reflected in the design set. Normalisation of each feature to zero mean value will often achieve this.

Normalisation for range can be most simply achieved by scaling each feature to have a range from -1 to +1 on the design set. This procedure is naturally sensitive to atypical, outlying members, and so a more common approach is to scale each feature to have unit variance on the design set. Other measures, such as mean absolute deviation, can be used, but these seem to have no outstanding advantages.

Normalisation to unit variance, over all classes, can lead to unwanted distortions of the class distributions. Fig. 2.14 illustrates this. In this example, normalisation of x_1 is determined largely by the empty space between the classes, and the distributions after normalisation are likely to be more difficult to separate than before. Again, this kind of distortion is most significant for Specht's



a) before normalisation.

b) after normalisation.

FIGURE 2.14 - NORMALISATION OF VARIANCE OVER ALL CLASSES.

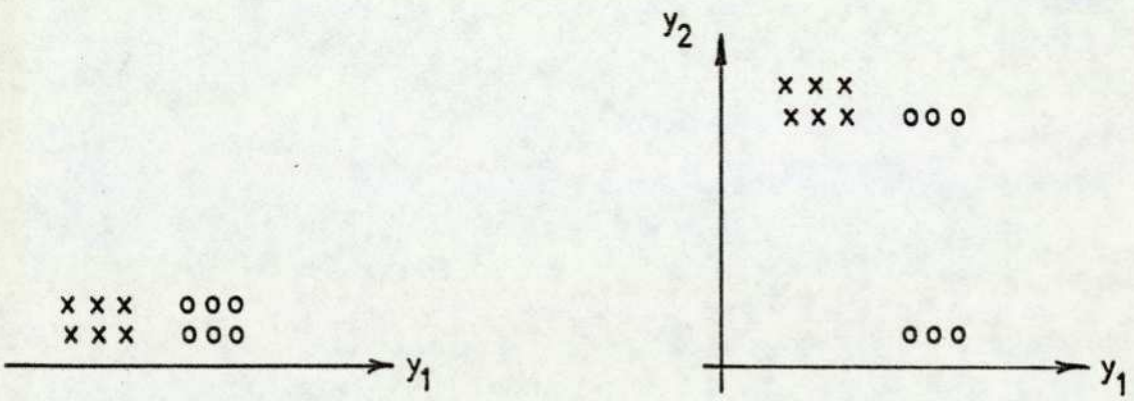


FIGURE 2.15 - INCREASED CONFUSION WITH ONE MORE FEATURE.

classifier. This uses the spherical Gaussian distribution as a potential function, and is therefore best suited to class distributions which are themselves roughly spherical. The problem can be avoided by scaling each feature to have unit average standard deviation per class. This normalisation would leave the space of Fig. 2.14(a) essentially unchanged.

Feature normalisation is, in fact, of little significance with the LMSE linear classifier. This is because of the optimising property of the transform from the feature space to the decision space. Normalisation can itself be represented as a linear transformation from the feature space to a normalised space. Without normalisation, the LMSE procedure would yield the optimum transformation from the feature space to the decision space. With prior normalisation, the optimum transformation from the normalised space to the decision space would be provided. Since optimality holds in both cases, the product of the normalising matrix and the corresponding transformation matrix in the second case must be equivalent to the transformation matrix alone in the first. In essence, the optimum transformation will automatically incorporate any normalisation required. In practice, however, some such prior normalisation is usually necessary to prevent overflow/underflow problems during computation.

2.3.6.2 Feature Selection

The importance of subset selection rests on two considerations. First, if a set of, say, 50 candidate features can be reduced to a sufficient subset of, say, 10 features, the on-line computational saving can be tremendous. Second, performance can be better, rather than merely sufficient, with the subset, as against the full candidate set. This possibility arises because a feature can be "worse than useless", by increasing confusion between classes, rather than merely failing to reduce it. Fig. 2.15 illustrates this.

Subset selection requires the specification of:

(1) a search procedure, to generate alternate subsets for evaluation;

(2) an evaluation criterion, applicable to any subset generated by the search procedure.

Usually, the search procedure must be something other than merely to generate all possible subsets in sequence, since, for N candidate features, 2^N subsets are possible. If N = 50, for example, this number is truly astronomical. Procedures such as Dynamic Programming are not applicable because a structured evaluation criterion, related to the worth of a subset, and such as to allow interpolation between subsets, does not exist for arbitrary class distributions. Consider, for example, Fig. 2.16, in which two features are shown which provide excellent class separation together, but poor separation in isolation. It follows that there can be no guarantee of finding the best subset. In these circumstances, a suitable procedure is the "without-replacement" search. In this procedure, the candidate features are all evaluated singly, to find the best subset of dimension one. The best single feature is then combined with all other features, to find the best subset of dimension two which includes the best of dimension one, and so on. An alternative scheme is to evaluate all subsets of dimension (N-1) from a candidate set of N features, so as to select the best. The excluded feature is then rejected from further consideration, and the cycle repeated with (N-1) features as candidates, and so on. The former scheme is often referred to as "forward sequential" and the latter as "backward sequential". With the backward sequential search, a significant disadvantage is that the initial, non-reversible rejections are made from a high-dimensional space. We shall see when we discuss the problems of performance estimation in such spaces, that these rejections are likely to be ill-founded.

Various suggestions have been made to ease the restrictions imposed by the basic without-replacement search (e.g.

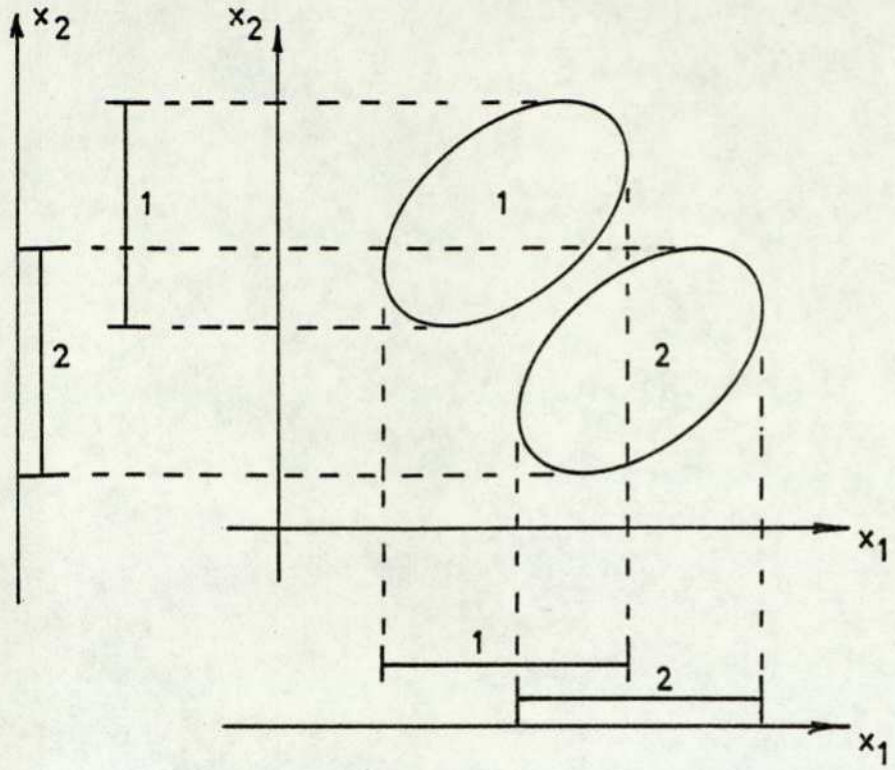


FIGURE 2.16 - MISLEADING MARGINAL DISTRIBUTIONS.

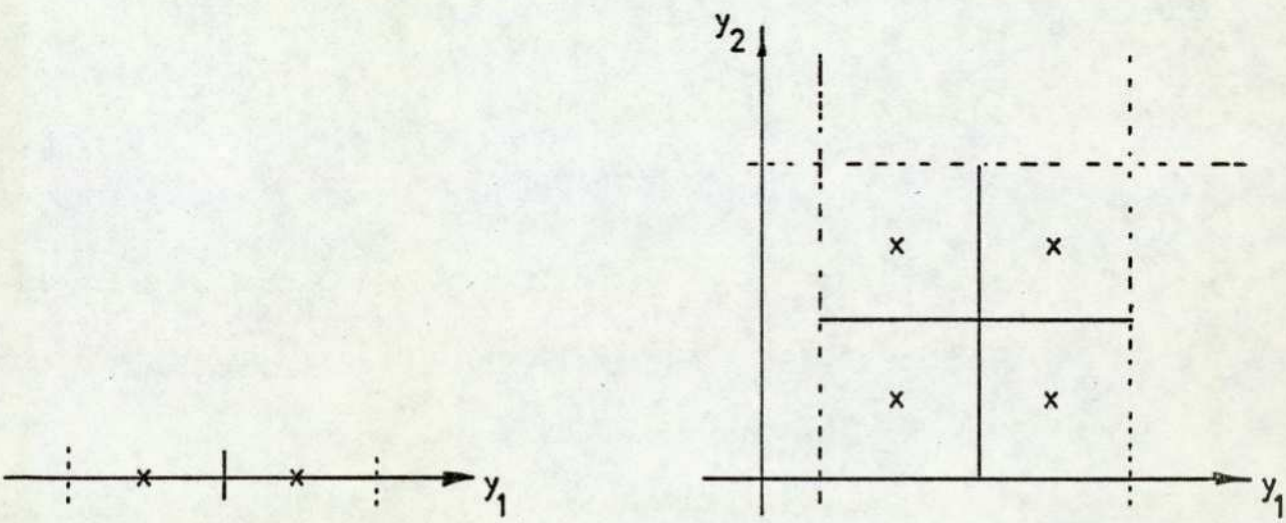


FIGURE 2.17 - THE DIMENSIONALITY PROBLEM.

ref. 22). In the forward sequential scheme, for example, the primary restriction is that the subset of dimensionality k must contain the subset of dimensionality (k-1). It is not difficult to construct examples where this restriction makes it impossible to arrive at the best subset. To overcome such restrictions, the price of an increased computational load must be paid. For example, the forward sequential scheme can be merged with the backward sequential scheme, so that additional features are selected by "two steps forward and one step backward", rather than simply "one step forward". Other combinations are readily constructed.

An evaluation criterion should, ideally, reflect the class separation induced by a feature subset. For arbitrary class distributions this is very difficult, and consequently many criteria have been proposed which are applicable only to particular types of distributions. The vast majority of these assume a Gaussian distribution for each class, and include measures based on Information Theory, Entropy, Divergence, etc. (refs. 23, 24, 25). In many cases, these criteria can be monotonically related to the Bayes error rate, and can provide upper and lower bounds upon that error rate, given the assumption of Gaussian distributions. Unfortunately, this assumption seems quite inappropriate to this project, and so these measures will not be considered further.

In practice, it is misleading to discuss the worth of a feature subset, without considering the classifier with which it is to be used. For example, a subset may yield perfect separation between two classes, but with an inherently bimodal distribution in each. This would lead to good classification performance with a potential function classifier, but probably abysmal performance with a linear classifier. The subset would therefore be good for the former, and bad for the latter.

The evaluation criterion should therefore measure the expected classification performance of a particular classifier,

with the specified subset. This is most simply done by designing the classifier upon that subset, and observing its performance. Since we are ultimately interested in performance on unseen data, we should try to estimate this from the design set. A suitable procedure would be to form the "leave-one-out" estimate described in section 2.3.3, and later.

To conclude this discussion, it seems that for the surface inspection problem, in which complex class distributions can be anticipated, a suitable procedure for feature selection is a without-replacement search (forward sequential) coupled with subset evaluation using the classifier itself. Although this option is likely to be computationally expensive, it should prove acceptable, given a powerful computer for the design process, and careful choice of classifiers.

2.3.7 Performance Estimation

A data set must serve two purposes. It must be used to design a classifier, and it must then be used to evaluate the performance of that design. We have discussed the first in some detail, and we shall now discuss, briefly, the second.

Performance evaluation is, in fact, a problem of prediction or estimation, since we wish to know how the classifier will perform on new, unseen data. In the surface inspection context, we must estimate the performance "on-line". It is fairly obvious that a performance estimate based on the same data as was the design will be optimistically biased - in many cases, disastrously so. We must therefore set aside some portion of the available data as a test set, design the classifier on the remainder, and then evaluate its performance on the unseen test set. Since, in this case, the test set will actually be unseen data, we can expect a reasonable performance estimate. If a large quantity of data is available, our problem is solved, for our design set and test set can then be of a reasonable size. In most cases, however, our data will be limited by the cost and effort involved in gathering and identifying it. In this case, we face something of a dilemma. A large design set implies a

good design, but a small test set and therefore suspect evaluation, and vice versa.

One approach to this problem is to repeat the design-test procedure several times, each with a different partition, and to average the results. In the limit this becomes the "leave-one-out" procedure. As the name implies, each member of the data set is omitted, in turn, from the design process, and the resulting design used to classify that single member. The average of the results thereby achieved for each member is taken as the required performance estimate. The penalty to be paid for multiple design-test partitions is, of course, computational effort. This is a further incentive to choose classifiers which are relatively simple and quick to design. The LMSE procedure is especially attractive in this respect, since it is possible to modify a design based on the complete data set, so as to produce a design based on the complete set less one member, without repeating the entire design process.

A certain amount of work has been carried out on the question of the number of design samples needed to produce a useful classifier. It is clear that a design set can be so small as to yield a meaningless design, and if this is so, multiple partitions of similar character will yield an equally meaningless estimate. For arbitrary data, no firm conclusions can be drawn, but it seems that two factors must be considered:

(1) the dimensionality of the space in which the classifier will operate;

(2) the number of classes to be distinguished.

To see the relevance of dimensionality, consider the following argument (ref. 7). Let a d-dimensional space be divided into hypercubes (d-dimensional cubes) by means of a single threshold on each dimension, located at the mean. Fig. 2.17 illustrates this for one and two dimensions, giving two and four cells respectively. Suppose the design set is uniformly distributed throughout the space, with a sample density of one per cell (Fig. 2.17). In general, therefore, we have 2^d samples. Certainly, in one, two and three dimensions

this is a sparse distribution. By implication, therefore, 2^d samples in d dimensions is equally sparse. Yet, for example, 2^{20} is greater than one million!

As another example, consider that a d-dimensional space is to be partitioned into two classes by a linear classifier. Let the design set consist of n samples in general position (i.e. such that no subset of (d+1)samples lies wholly within a (d-1)-dimensional subspace). There exist 2^n possible dichotomies of the design set between the two classes, and a certain fraction of these will be linearly separable. Denote this fraction by f(n,d). It can be shown (ref. 6) that this fraction is given by:

$$f(n,d) = \begin{cases} 1 & n \leq (d+1) \\ \dfrac{2}{2^n} \sum_{i=0}^{d} \binom{n-1}{i} & n > (d+1) \end{cases}$$

Fig. 2.18 shows this function for various values of d. It can be seen that if n = 2(d+1), for example, there is a probability of 0.5 that <u>any</u> design set will be linearly separable. This does not necessarily imply similar separability on unseen data.

The relevance of the number of classes to be distinguished is fairly clear, since we must have sufficient samples from each class to characterise the distribution of that class. As mentioned, firm rules cannot be derived for arbitrary distributions, but a reasonable rule of thumb seems to be no less than 10d samples per class,where d is the dimensionality of the space (ref. 26).

2.4 Related Work

The importance of the surface inspection problem is underlined by the evidence of two research projects on related topics, and these will now be briefly summarised.

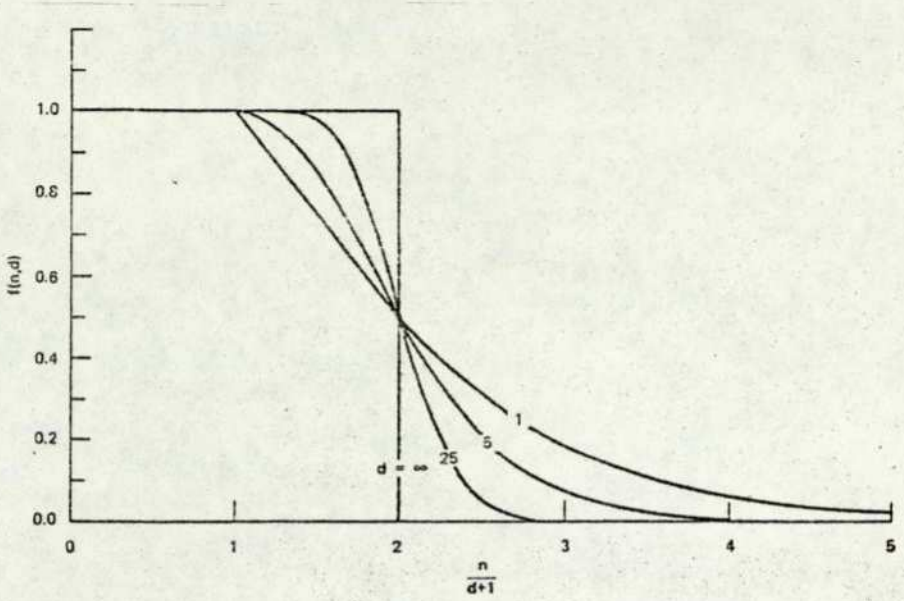


FIGURE 2.18 - THE FRACTION OF DICHOTOMIES OF n POINTS IN d DIMENSIONS THAT ARE LINEAR.

(from ref 6)

2.4.1 Videoprint Analysis on Steel Strip

In the early stages of the work reported in this thesis, a related (but independent) research programme was in progress at the University of Glasgow (ref. 27). This programme was concerned with defect recognition on the surface of cold-rolled steel strip, using videoprint data from the SIRA flying-field scanner and defect detection system, described in Chapter 1. This data was recorded on punched paper tape, and covered 134 distinct defects from five classes - laminations, black dots, stains, fleck and rust. Measurements were made of the shape of each defect, including area, circumference, width, etc. To extract these measurements automatically, algorithms were developed to associate individual trigger samples arising from the same defect. Essentially, these algorithms examined the distance between trigger samples so as to combine those which were sufficiently close. A different distance measure was used along the strip to across, on the grounds that many defects tend to be elongated in the direction of strip movement. These algorithms were apparently not intended for on-line application.

The measurements so derived were combined, using feature space pattern recognition, so as to identify the defect class. Results were encouraging, with correct recognition rates between 87% and 97%, depending upon the classifier. These results support our enthusiasm for feature space techniques in the surface inspection context. The work should be extended to encompass analogue data, as well as binary (videoprints), and the results confirmed on a larger data base, particularly one covering more defect classes. It will be interesting to compare these results with those of the companion project, mentioned in section 2.2, which uses the techniques of linguistic pattern recognition.

2.4.2 Optical Polar Diagram Analysis

A laser scanner, fundamentally similar to the SIRA scanner, has been developed at the Axel-Johnson Institute in Sweden. A system has also been produced for defect recognition (ref. 28).

This system differs fundamentally from the work reported

in this thesis in two respects:

(a) Specific defect identification, of the kind available from human inspection, was not a design aim. Instead, defects were sorted into one of 13 general types, including "large, spot-formed, light scattering", "small, non-spot-formed, light-absorbing", "large area", etc. Further, in the work reported, these classifications were used solely to grade the material into good, second-rate or scrap.

(b) Classification was achieved solely from an analysis of the optical polar diagram produced by the defect as the scanning spot passed over it. Each defect type was considered to produce a characteristic intensity distribution in that polar diagram. This was detected and recognised with multiple detectors (photo-multipliers), together with an optical filter. The latter was designed to block most of the light from a defect-free surface, and to transmit that scattered by defects to one or more receivers.

A minicomputer was incorporated but this was apparently used solely for data logging and subsequent derivation of surface quality grading. It did not participate in the recognition process.

The system is claimed to detect scratches down to 20.10^{-6} m in width, 2.10^{-6} m in depth, and black dots down to 4.10^{-4} m in diameter, at line speeds up to 20 $m.s^{-1}$.

System design was carried out in the laboratory, with samples from several sources. About 7% of the defects were found to be undetectable, and this was considered satisfactory.

The system was then installed on-line to inspect steel strip. Sections of the surface were graded into good, second-rate or scrap, according to the number, size and type of defects detected by the system. Results were compared with human inspection and the system adjusted for optimum correspondence.

Several kinds of comparison were made, and Table 2.1 typifies the results. This table shows the percentage differences in the amount of material assigned to each of the three categories, both between two human inspectors and between the inspectors and the automatic system, with one inspector as reference. These results are based on 1500 sample areas, each 0.1 m by 0.25 m, and a line speed of approximately 3.10^{-3} $m.s^{-1}$, the latter clearly favouring human inspection.

The concept of sorting defects according to their optical polar diagram is important. It is intuitively appealing, and holds the promise of high-speed operation. The project, of which the work reported in this thesis forms a part, is currently involved in a study of this aspect of defect recognition, in collaboration with the British Steel Corporation. Measurements extracted from the polar diagram could be incorporated as additional features in the current scheme, or the analysis could be carried out optically as a pre-sorting exercise. It remains to be seen whether the defects of interest do actually exhibit sufficiently unique scattering characteristics.

2.5 Summary

This chapter has discussed the defect recognition problem on isolated scan sections, assuming adequate delineation of those sections. A discussion of the requirements from such a recognition system, together with the expected difficulties of the problem, lead to the conclusion that Feature Space techniques of automatic pattern recognition hold the best promise of a solution, but that these will need to be carefully selected from those available.

A selective literature survey is presented, with emphasis on those techniques potentially suitable for this problem. This is based upon the requirements of fast data processing at reasonable cost, coupled with an expectation of considerable within-class variability. Preference is given to methods well-founded in theory, since "practice without theory is blind". The survey includes the problems of feature normalisation and selection, and

performance estimation; the latter sounding a cautionary note on the peculiarities of high-dimensional spaces.

Finally, a brief summary is given of two related research projects, both of which give some grounds for optimism in this work.

Inspector \ Assignment	Good	Second-rate	Scrap
A	+ 6.7%	- 6.2%	- 0.5%
B	Reference		
System	+ 3.0%	- 2.4%	- 0.6%

Table 2.1

Comparative results for the Axel-Johnson system and two human inspectors

(modified from ref. 28)

3. EXPLORATORY WORK

3.1 Introduction

A number of techniques have been identified as being potentially useful for on-line surface inspection, as described in Chapter 2. The actual usefulness of these techniques can be evaluated only with real data from a particular inspection problem, but the acquisition of such a data set - large enough for a definitive evaluation - requires substantial effort, not only by the researcher, but also by production, inspection and research staff of the manufacturer. Work aimed at acquiring such a data set was started close to the beginning of this project, and is described in Chapter 4 of this thesis, together with the data set and the work carried out with it.

In parallel with the gathering of this data, a fair amount of work was carried out in preparation for its processing. Much of this work was the development and testing of programs to implement the techniques discussed in Chapter 2, but it was also possible to explore these techniques further by running the programs on a small data set which was already available. In many respects the results obtained in this way were quite unexpected, and seemed to contradict certain preconceived beliefs about the problem and the techniques. Of course, results from limited data must be treated with extreme caution. Nonetheless, the questions which were raised have proved invaluable both as a spur to further consideration and development of the techniques, and as a guide to the most important analyses to be applied to the larger data set. For these reasons, it seems that this exploratory work should be reported, and this is the purpose of this chapter.

3.2 The Data Set

This data set was gathered prior to the beginning of the work described in this thesis by scanning sheets of cold-rolled steel strip (each approximately 1 m square) with the flying-image system of the SIRA Institute (see Chapter 1). The resulting scanner output was digitised (8 bits), punched on to paper tape and finally

stored on magnetic tape at the University Computer Centre.

The scan signal was sampled at a rate corresponding to one sample per millimetre (referred to the inspected surface) and to do this economically it was necessary to reduce the scan rate to only a fraction of the typical on-line rate. Such a reduction affects the various signal components differently. In particular, the component due solely to variations of surface reflectivity (including defects) suffers a downward shift of its frequency spectrum, whereas the component due to photomultiplier shot noise is unaffected. In such relationships, therefore, the data does not represent the on-line situation. Although photomultiplier shot noise is significant in the signal produced by the flying image scanner, this distortion should not seriously influence the results from the kind of processing to be described.

Figure 3.1 shows a number of scans from the data set. In some of these a defect signal is clearly visible, and in others it is not. Unfortunately, the defect detection signal described in Section 1.3 was not recorded with this data set. This signal is required to locate and define the limits of the signals to be analysed. In some cases, it could be closely estimated from a purely visual examination of the data. With a fair degree of confidence, this was found to be possible for 39 scans, and the remainder have not been used in this work.

The relevant sections of the 39 usable scans are shown in Figures 3.2A-E, together with the estimated detection threshold. Although this threshold is only an estimate, these figures highlight a disturbing feature of such a detection system - namely, that a portion of each defect pulse invariably lies above the detection threshold. By using this threshold to gate defect signals into a defect classifier, this portion of the signal would be lost. This probably matters little on some of the larger signals, but up to 50% of smaller signals can be lost in this way.

In the 39 scan sections, five defect classes are represented:

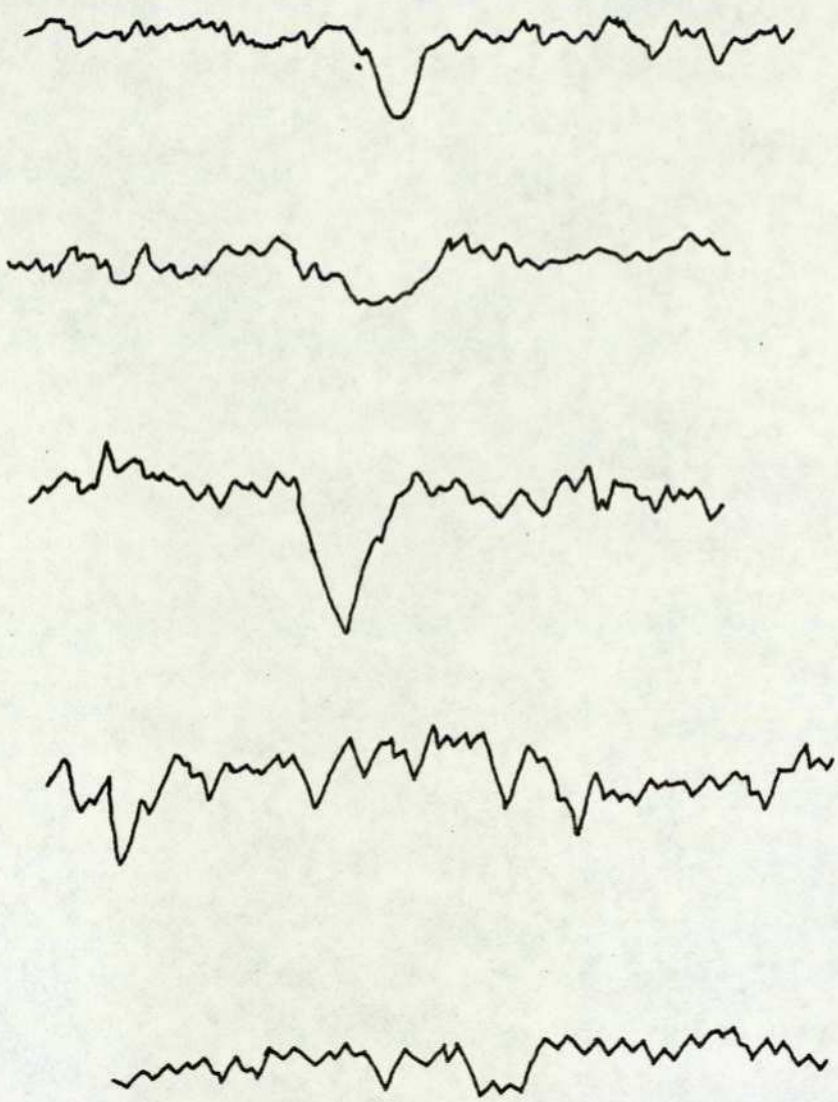


FIGURE 3.1 - TYPICAL DEFECT SCANS.

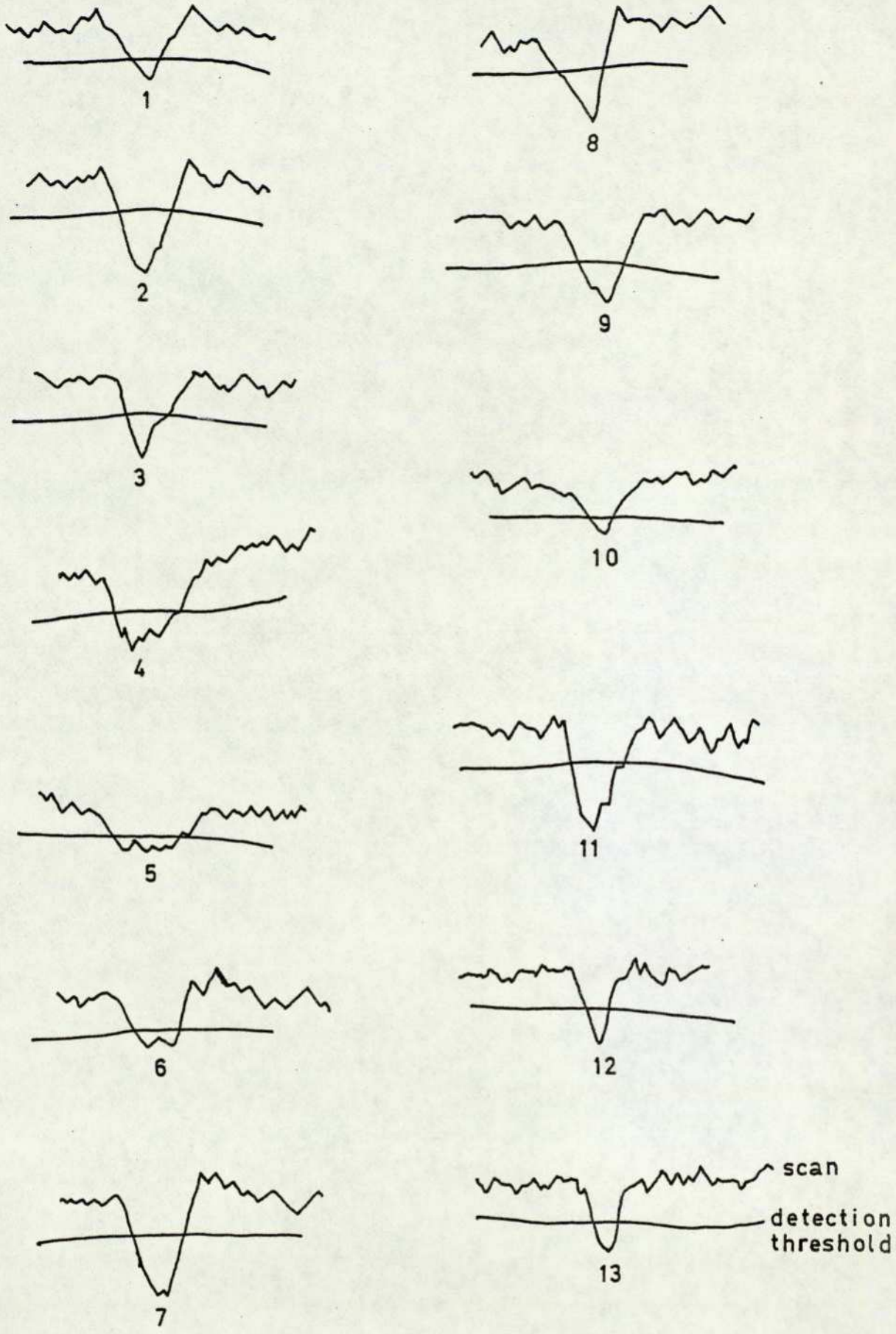


FIGURE 3.2A - PITS.

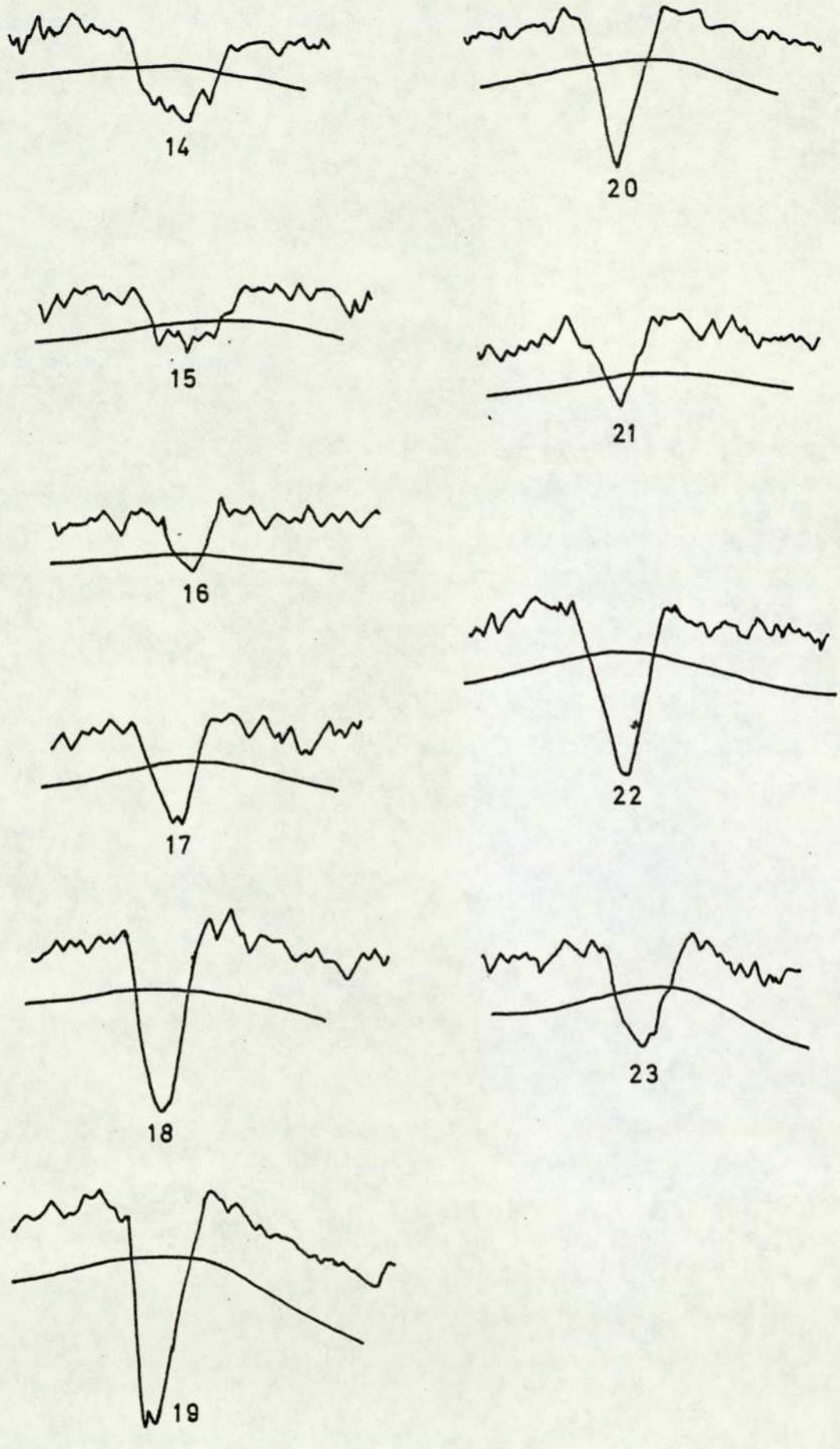
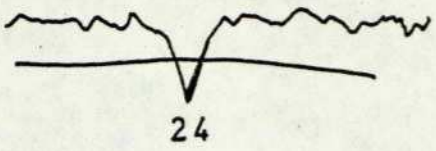
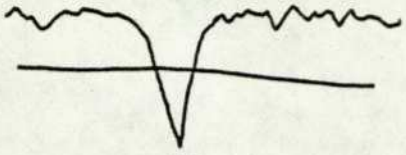


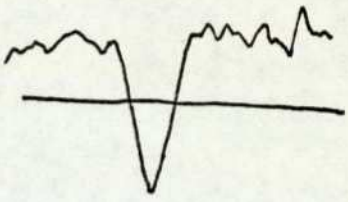
FIGURE 3.2B - GOUGES.



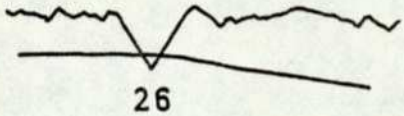
24



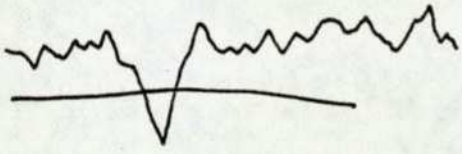
25



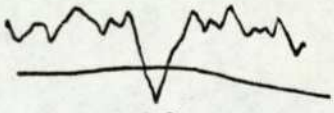
29



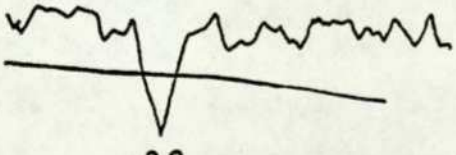
26



27



30



28

FIGURE 3.2C - RUST SPOTS.

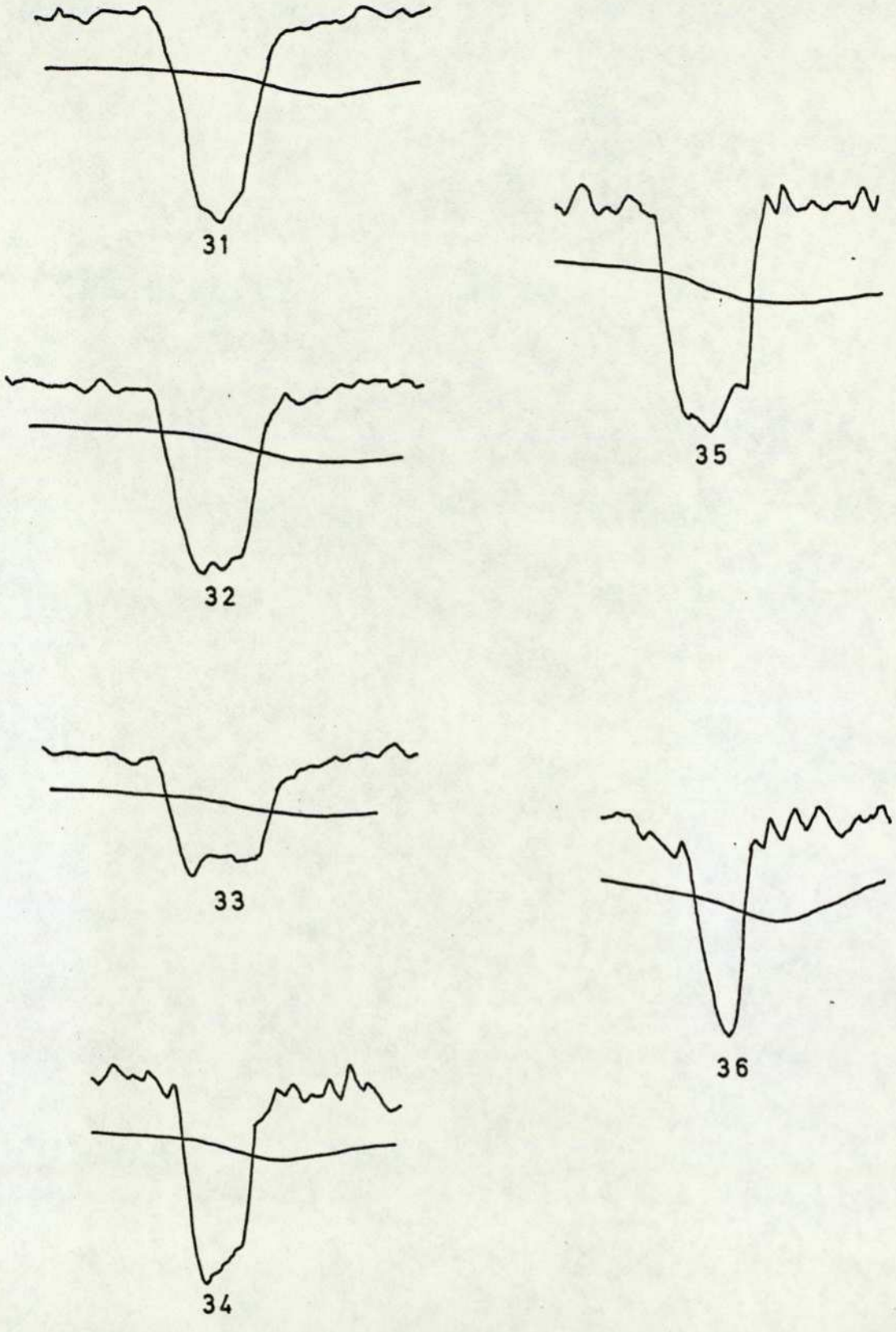
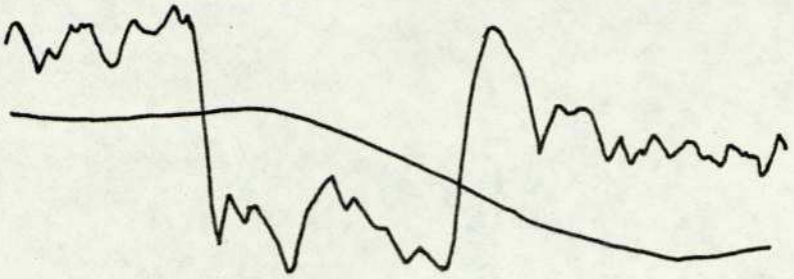
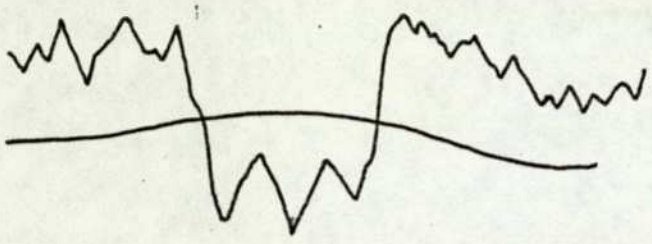


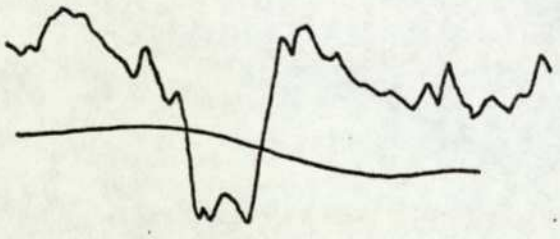
FIGURE 3.2D - SCALE.



37



38



39

FIGURE 3.2E - HEAVY LAMINATION.

(1) Pits (13)

(2) Gouges (10)

(3) Rust spots (7)

(4) Scale (6)

(5) Heavy lamination (3)

- where the numbers in parentheses are the number of scan sections from each class.

The limited size of this data set is evident. Furthermore, although there are 39 scan sections, these are derived from only 14 distinct defects. This is because several successive scans often cover adjacent portions of a single defect.

3.3 Signal Characterisation

In Figures 3.2A-E clear generic differences are evident between the defect signals, although it is difficult to define these differences precisely. For example, the signals from rust spots tend to be of short duration, of low amplitude (optical contrast) and usually triangular in shape. Scale produces a bigger signal (in both aspects) and one which tends to be "flat-bottomed".

To apply feature space techniques, as described in Chapter 2, the signals must be characterised by the results of a set of tests or measurements performed upon them - each of which yields a scalar value. In this application, it is clear that the measurements must, in some sense, characterise the shape of the defect signals. There are a number of measurements which can do this, and a few have been selected for ease of implementation at typical on-line processing speeds. These will now be described in increasing order of their complexity and cost of implementation.

3.3.1 Samples-as-Features

The defect signals are available as a sequence of digitised 8-bit samples across each waveform. The simplest possible feature set, therefore, consists simply of this sequence of sample values.

For any waveform, the number of samples available depends on the time duration of the waveform, which, in turn, depends on the defect size and the location of the scan across the defect. This number varies from as little as 4 on some rust waveforms up to 35 on heavy lamination. Since every signal must be represented in the same feature space, the same number of features must be measured on each one. This is most simply achieved by setting unused values to zero on the smaller signals.

With this proviso, a set of 50 features was defined. A verbal description of the Kth feature is "the value of the Kth sample on the waveform, following the initial crossing of the detection threshold, or zero if the final crossing has already occurred". Figure 3.3 illustrates this.

This feature set is undeniably crude. It offers no reduction of the information content of the signal. It incorporates no normalisation, such as for the signals mean value. Nonetheless, it requires no calculation whatsoever, and can be implemented with a standard analogue-to-digital converter at high speed and at moderate cost. To this extent, any competing feature set must justify itself by an improved performance. Essentially, samples-as-features can be expected to provide a "base-line" performance level against which other feature sets may be judged.

3.3.2 Geometric Features

The term "geometric" is used here to describe measurements which arise naturally in descriptions of geometric figures such as circles, squares and triangles. These include area, perimeter, width and length. Equivalent measures can be defined for the defect waveforms, and these are illustrated in Figure 3.4. For each measure, the limits of the waveform are taken to be the initial and final crossings of the detection threshold.

X_1 is the base width of the waveform

X_2 is the waveform amplitude, defined as the positive difference between the maximum and minimum signal levels

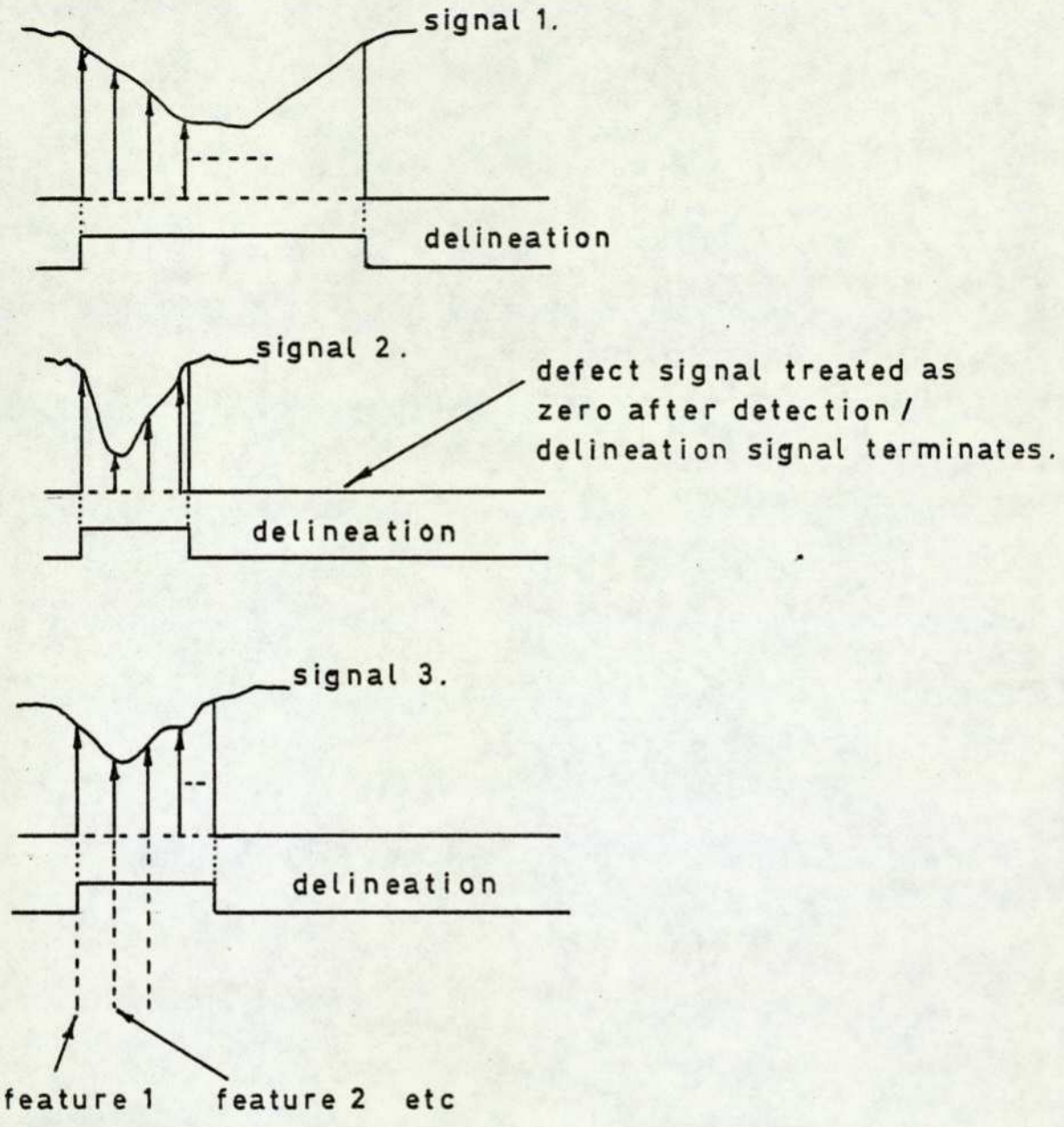
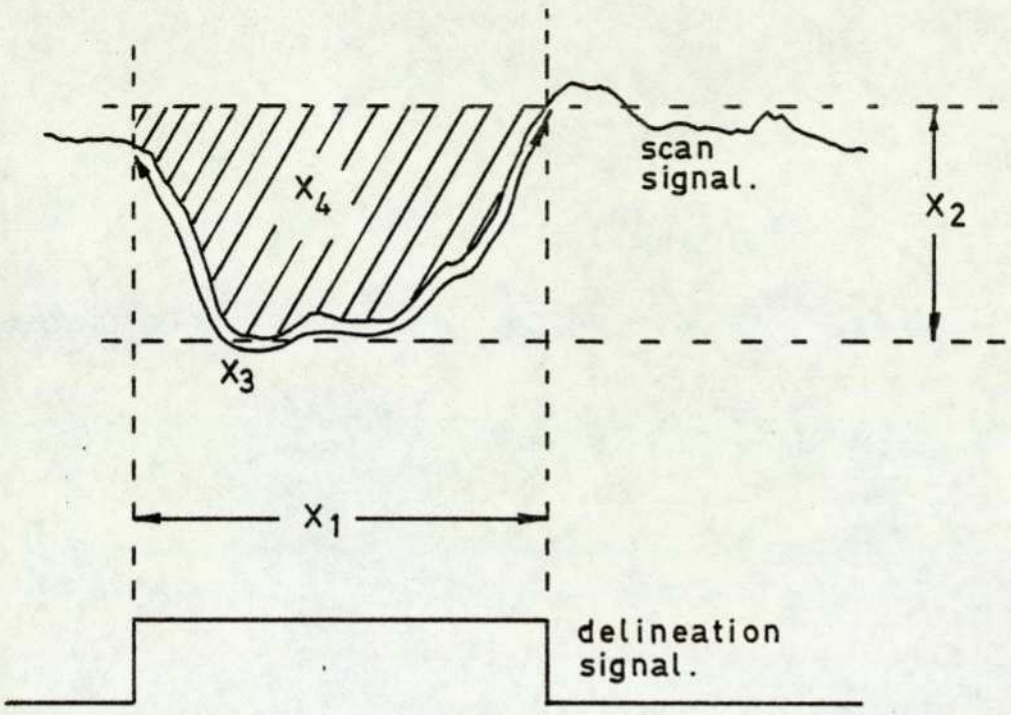


FIGURE 3.3 - SAMPLES-AS-FEATURES.



X_1 = base width

X_2 = depth (amplitude)

X_3 = perimeter

X_4 = area

FIGURE 3.4 - GEOMETRIC FEATURES.

X_3 is the perimeter or curve-length of the defect waveform

X_4 is the area enclosed between the defect waveform and a conceptual horizontal line passing through its maximum value.

Each of these measures has physical significance. X_1 measures the defect size in the scan direction. X_2 measures the change of optical contrast across the defect. X_3 reflects the monotonicity of the signal variation, and hence the fine structure of the defect. X_4 might be termed the "integrated optical contrast".

For their measurement, X_1 and X_2 are self-evident and may be implemented in hardware by a gated clock and peak detectors respectively. For X_3 and X_4, the waveform was taken to be the piecewise linear approximation which results when the waveform samples are linked by straight lines. This allowed X_3 to be calculated as a sum of line segment lengths (each of the form $\sqrt{x^2 + y^2)}$, and X_4 to be calculated via Simpson's Rule. In a hardware implementation the waveform need not be sampled. X_3 may then be measured with differentiators and integrators, and X_4 by a single integrator. Hardware implementation will be discussed more fully in Chapter 6.

A note of caution must be sounded with the feature X_3, the waveform perimeter. In an important respect this feature is not defined. Consider in Figure 3.4 that the horizontal dimension corresponds to time or distance, whereas the vertical dimension corresponds to voltage or optical contrast. To calculate numerical values for the four geometric features, units of measurement must be assigned to the horizontal and vertical dimensions. For X_1, X_2 and X_4 the assignments are unimportant because their scale will reflect <u>linearly</u> in the results, and can therefore be removed by a standard "shift-and-scale" normalisation. For X_3 this is not so. If, for example, the horizontal axis is measured in microseconds rather than milliseconds, the effect on the calculated perimeter value is <u>non-linear</u> and effectively irreversible. To this extent the calculation of X_3 is arbitrary.

Upon reflection it seems that the significant factor in this choice of units is the ratio between the horizontal and the vertical. For example, if the horizontal scale yields values around 1000 times as large as the corresponding vertical values, X_3 will reflect little more than the waveform's base width. This implies that the average vertical sample-to-sample variation should be numerically about equal to the horizontal sample-to-sample increment. In the event, the four geometric features were calculated on a vertical scale from 0 (corresponding to a zero value sample) to 1 (corresponding to a sample value of $2^8 - 1 = 255$), and a horizontal scale in which the sample-to-sample increment was 4.10^{-3} (approximately equal to $1/255$).

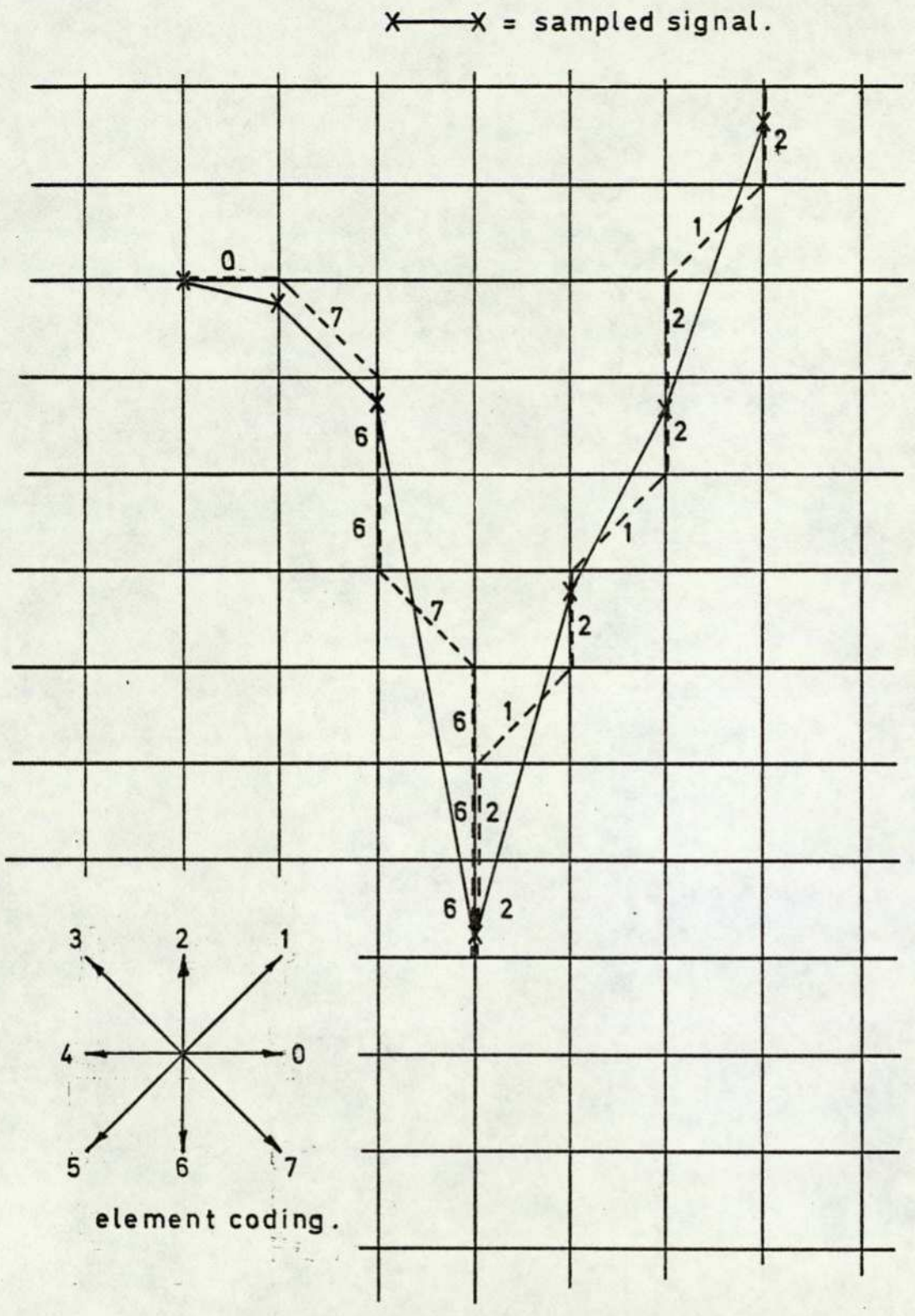
3.3.3 Chain-Encoding

3.3.3.1 The Encoding Process

Defect waveforms are recorded as a sequence of 8-bit digital samples. As described in reference 38, Freeman's chain-encoding process can be used to produce a representation in terms of a sequence of symbolic chain elements, of which only five different types are required.

Conceptually, the encoding process involves superimposing a rectangular grid upon the waveform and recording the sequence of grid-waveform intersections in terms of the symbolic code. Conventionally, a "square" grid is used, but this concept seems meaningless in this application (cf. the calculation of waveform perimeter discussed in Section 3.3.2).

There are strong arguments for locating successive vertical grid lines at successive sampling instants across the waveform, and this has been done. Figure 3.5 illustrates the encoding process under this constraint, for a particular spacing between horizontal grid lines. It remains to determine a reasonable spacing between horizontal grid lines. It seems that the "best" value for the spacing must depend on the spread of sample-to-sample variation in the waveforms of interest, and will probably be close to the average



chain code = 07667666221212212 .

FIGURE 3.5 - CHAIN-ENCODING.

value of that variation. The choice therefore is problem dependent. Initially, the encoding process was evaluated (via recognition processing) with two different spacings, 10^{-3} and 10^{-2}, on waveforms in which the sample values were scaled to lie between zero and 1. The coarser grid was found to be superior in every way, yielding a simpler code, a simpler classifier and better recognition accuracy. This suggested that the spacing should be chosen with some care. An analytical solution, based on averaging sample-to-sample variations, would be suspect for two reasons:

(1) average values will differ between waveforms, and especially between defect classes

(2) different portions of a waveform are likely to vary in their significance for defect classification, suggesting a weighted average approach.

In consequence, an empirical evaluation of different grid spacings was undertaken, based on a visual examination of the resulting encoded waveforms. Figure 3.6 shows some waveforms encoded with grid spacings between 5.10^{-4} and 1.10^{-1}, together with the original (sampled) waveforms. The smallest spacings and the largest clearly yield poor representations of the original waveform, and the best spacing seems to lie somewhere between 1.10^{-2} and 5.10^{-2}. It is, perhaps, worth reiterating here that the spacing between only the horizontal grid lines is being varied, with vertical grid lines fixed at the sampling instants. Figure 3.7 shows the same waveforms coded with spacings between 1.10^{-2} and 5.10^{-2} , and on balance, best representation seems to result with a grid spacing of 3.10^{-2} . It is of interest that the code strings produced with this spacing contain about the same number of elements (symbols) as the original number of digital samples.

This accords well with the suggestion that the grid spacing should be comparable with the average sample-to-sample variation. Furthermore, since each chain element

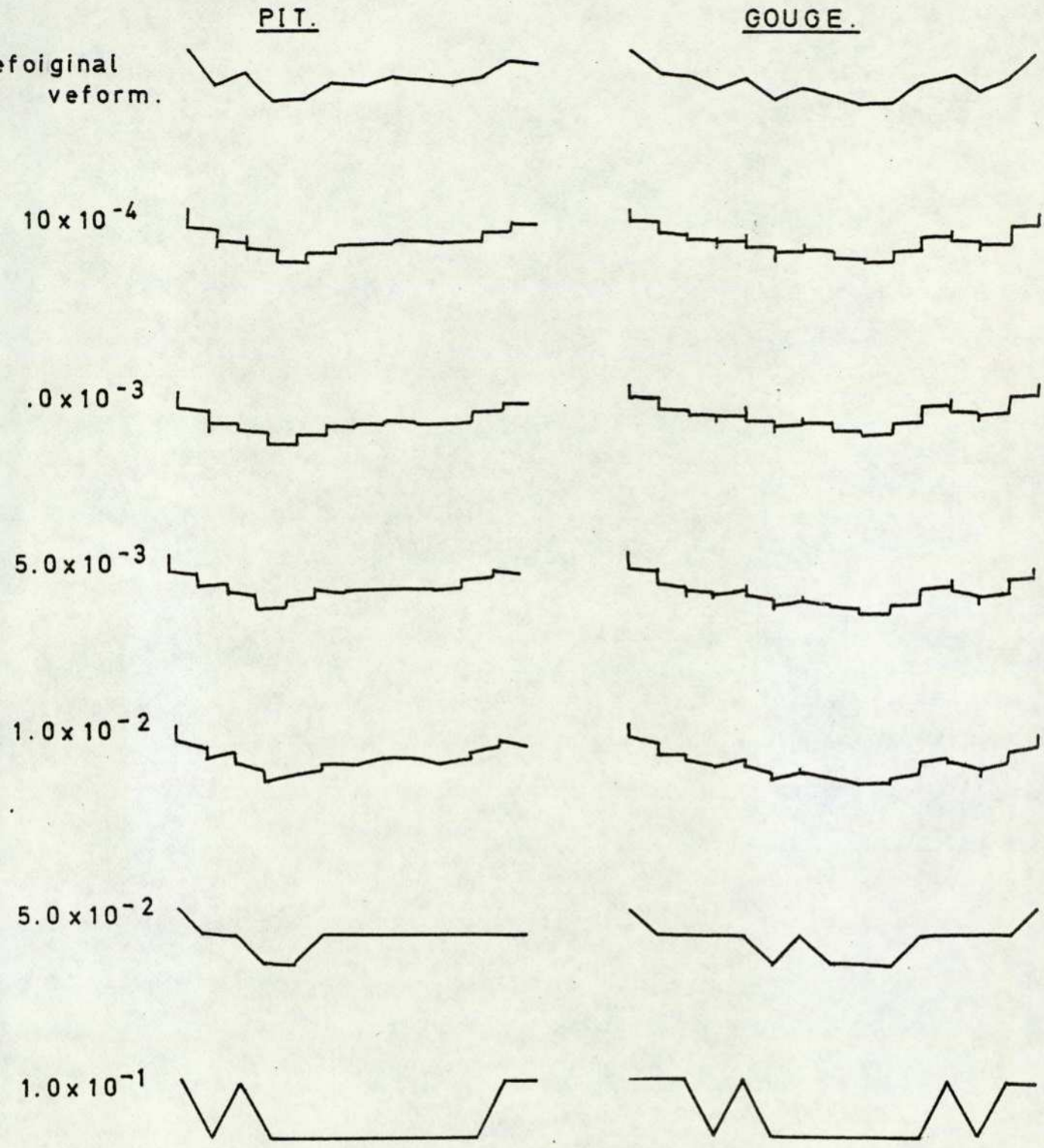


FIGURE 3.6A - CHAIN-ENCODING ON DIFFERENT GRIDS.

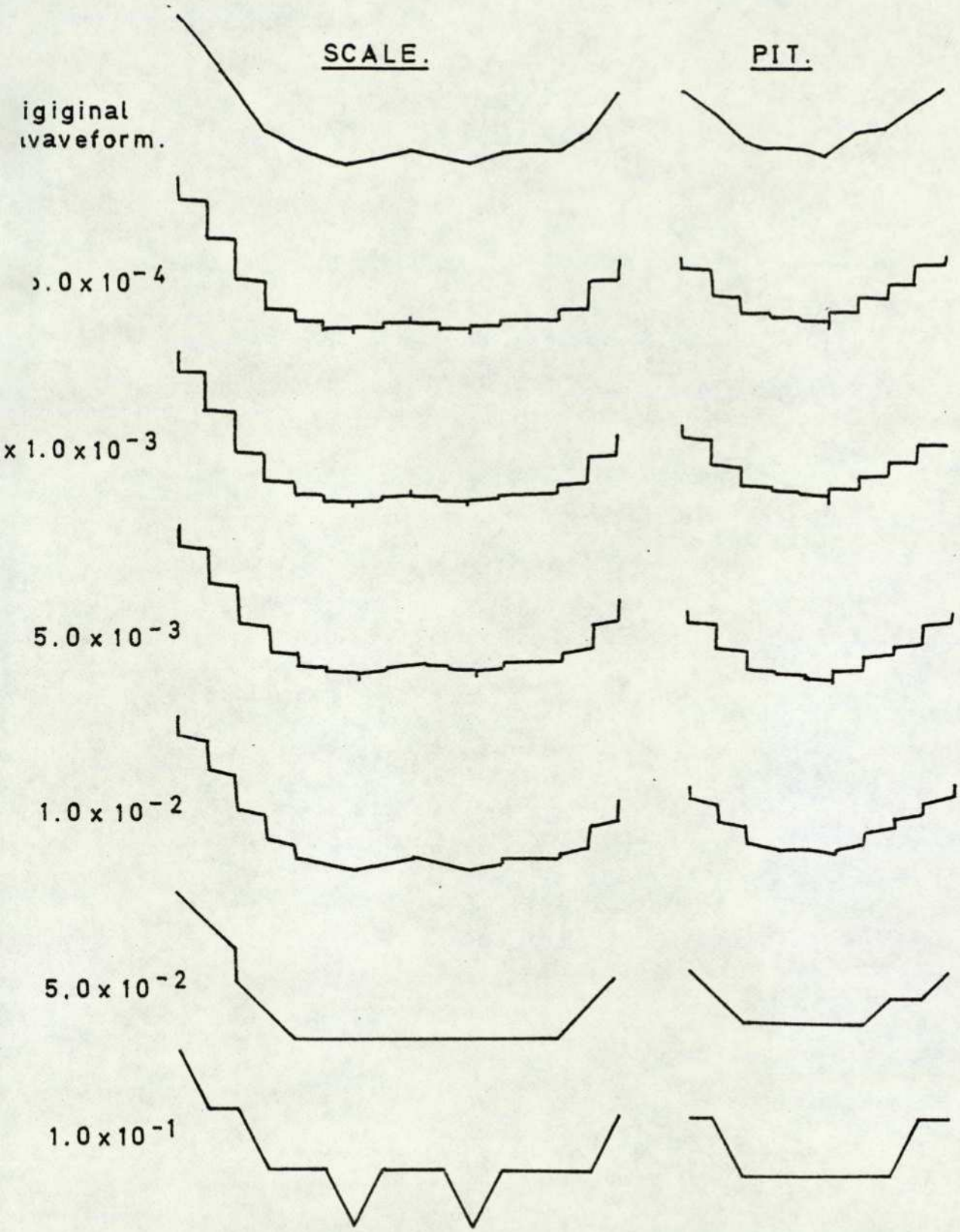


FIGURE 3.6 B - CHAIN-ENCODING ON DIFFERENT GRIDS.

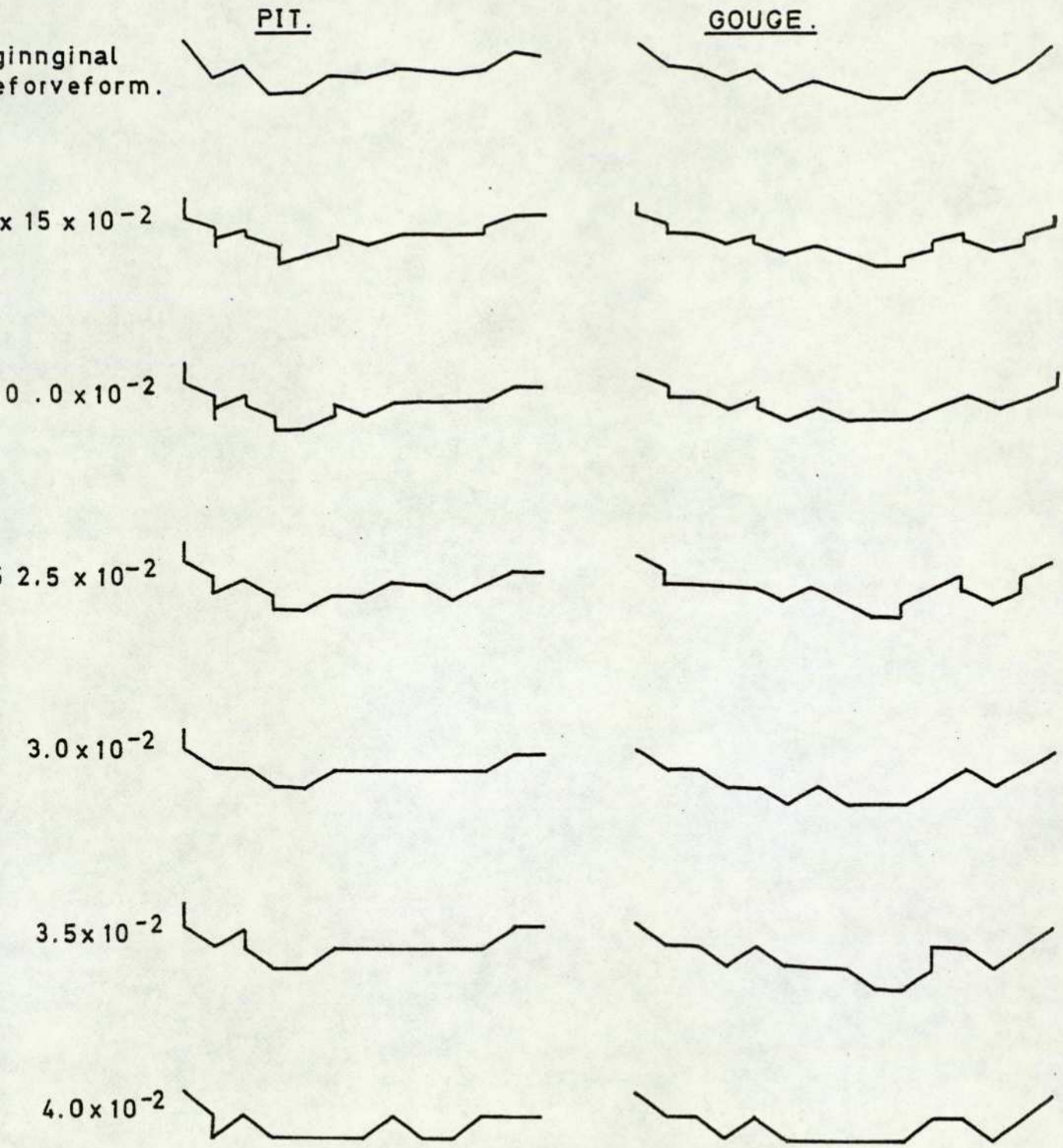


FIGURE 3.7A - CHAIN-ENCODING ON DIFFERENT GRIDS.

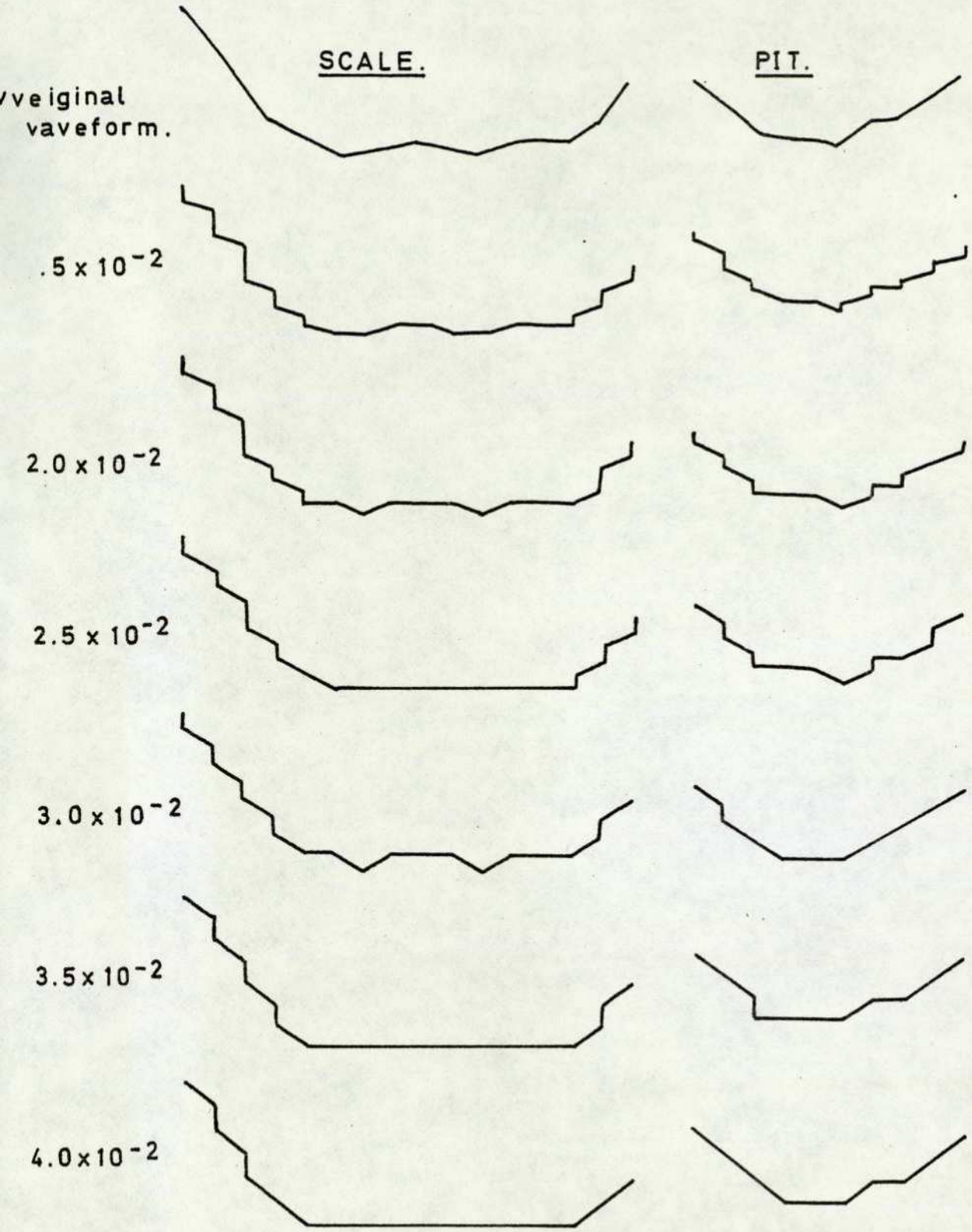


FIGURE 3.7B - CHAIN-ENCODING ON DIFFERENT GRIDS.

can take on one of only five different "values", compared with 256 possible values for each digital sample, the encoding process achieves a considerable reduction of information.

3.3.3.2 Feature-Extraction from Encoded Waveforms

Given a chain-encoded representation of each defect waveform, it is necessary to define a set of features to characterise that representation. As described in ref. 38, a set of five "slope-densities" can be extracted for this purpose, one for each type of chain element. Each slope-density is simply the proportion of elements of each type in the encoded representation. As a proportion, it includes normalisation for waveform size. Since waveform size is certainly an important distinguishing feature of waveforms from different classes, such a normalisation is undesirable. It was decided, therefore, to use "slope-counts" which, as their name implies, are simple counts of the number of chain elements of each type in the encoded representation.

Slope-counts represent a considerable information reduction over the original encoded representation, and such a reduction may well be too severe. Information on the element sequence within the code string is wholly discarded in this way, so that, for example, an element of type 6 occurring near the beginning of the waveform has the same effect as one which occurs near the end - although the latter would be less common and potentially more significant. To alleviate this problem, slope-counts may be extracted independently from successive sections of the waveform code. The size of such sections then determines the degree of information reduction, from zero when each section is just one element long, to the maximum already discussed when the complete waveform is a single section. For this exploratory study, slope-counts were extracted independently from each half of the waveform, for comparison with the global slope-counts extracted from the complete waveform.

Without chain-encoding, a feature set based simply on the sequence of 8-bit sample values has been described in Section 3.3.1. It seemed that the sequence of element codes might be used in the same way, so that the value of the Kth feature would be simply the numerical value of the code assigned to the Kth chain element. To this end, the chain elements were re-coded so as to reflect more closely their geometric significance, as shown in Figure 3.8. As with "samples-as-features", the shorter code strings were then padded with a unique symbol to make up 50 features.

In summary, then, three sets of features have been extracted from the chain-encoded waveforms:

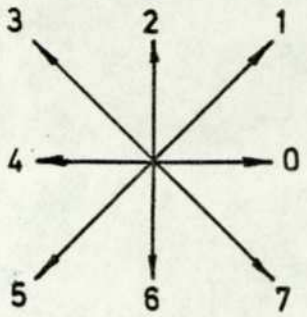
(1) Global slope counts (5)
(2) Semi-global slope-counts (10)
(3) Elements-as-features (50).

3.4 Feature Space Classification

In the main, results will be presented in approximately chronological order, since their unfolding was the primary guide to further program development. This development was in two main stages:

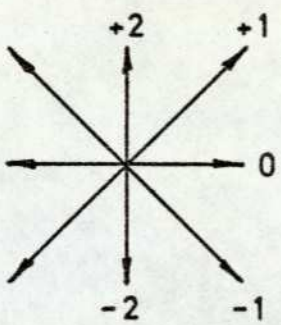
(1) A program was written to implement Specht's potential function classifier (Section 2.3.4) in exponential form, with a pre-defined set of features. With this program, the various feature sets described in Section 3.3 were evaluated in terms of the "leave-one-out" performance estimate from the 39 available waveforms.

(2) A feature selection program was developed to sift through a supplied set of up to 50 candidate features, and to select a subset yielding good performance. Initially, performance was evaluated as the leave-one-out estimate from Specht's classifier, but this was subsequently extended to allow



3, 4, 5 cannot occur with waveforms.

original assignment.



assignment to reflect geometric significance.

FIGURE 3.8 - CHAIN ELEMENT CODING.

the least-mean-square linear classifier (Section 2.3.5) to be used instead.

3.4.1 Results without Feature Selection

As mentioned, these results are with Specht's potential function classifier in exponential form, in a pre-defined feature space. Data input to the program consists primarily of a data matrix, in which each row corresponds to a waveform, and each column to a feature whose value is supplied on each waveform. This matrix defines both the design set and the space in which the classifier must operate. In addition, the true class membership for each waveform is supplied. The program first normalises each feature to have zero mean and unit variance over all waveforms, as discussed in Section 2.3.6.

Program output consists of a succession of confusion matrices, corresponding to successively increasing values of the smoothing parameter, σ, between pre-defined limits. Each confusion matrix is derived via the leave-one-out technique of performance estimation.

Five different feature sets were evaluated:

(1) Samples-as-features (Section 3.3.1)

(2) Geometric features plus Global Slope Counts with a grid spacing of 10^{-3} (Sections 3.3.2 and 3.3.3)

(3) As (2), but with a grid spacing of 10^{-2}.

(4) Semi-global Slope Counts, with a grid spacing of 10^{-2} (Section 3.3.3)

(5) Elements-as-features (Section 3.3.3)

In each case, the smoothing parameter was varied between 0.1 and 10.0. Figure 3.9 summarises the best results obtained with the first four feature sets.

Surprisingly, the most effective feature set seems to be samples-as-features, with which 64% of the waveforms may be correctly classified. However, this is in a space of 50

True class \ Assigned class	1	2	3	4	5
1	10	1	2	0	0
2	5	3	1	1	0
3	1	1	5	0	0
4	0	1	0	5	0
5	1	0	0	0	2

(A) Samples-as-features

Dimensionality = 50

Smoothing par. = 0.4

64% correct classifications

True class \ Assigned class	1	2	3	4	5
1	7	3	3	0	0
2	4	2	3	0	1
3	1	0	6	0	0
4	0	3	0	3	0
5	0	1	0	1	1

(B) Geometric + global slope counts. Grid = 10^{-3}

Dimensionality = 9

Smoothing par. = 0.5

49% correct classifications

True class \ Assigned class	1	2	3	4	5
1	9	1	3	0	0
2	4	2	2	0	2
3	0	0	7	0	0
4	0	3	0	3	0
5	0	1	0	1	1

(C) Geometric + global slope counts. Grid = 10^{-2}

Dimensionality = 9

Smoothing par. = 0.7

57% correct classifications

True class \ Assigned class	1	2	3	4	5
1	10	3	0	0	0
2	5	4	1	0	0
3	0	0	7	0	0
4	2	3	0	1	0
5	0	1	0	0	2

(D) Semi-global slope counts. Grid = 10^{-2}

Dimensionality = 10

Smoothing par. = 0.2

62% correct classifications

Class 1 = Pits,
Class 2 = Gouges,
Class 3 = Rust,
Class 4 = Scale,
Class 5 = Heavy lamination

Figure 3.9 Confusion matrices without feature selection (Specht's Classifier)

dimensions and with a relatively low value of the smoothing parameter. The smoothing parameter is, of course, the common standard deviation of the gaussian potential functions used by the classifier, and can be compared with the normalised standard deviation per feature of unity.
In this context, a value of 0.4 implies rather peaky discriminant functions, which are not likely to be adequately approximated by low order polynomials. With 50 variables, high order polynomials are unthinkable.

Of the two feature sets composed of the geometric features plus global slope counts, the grid spacing of 10^{-2} produces substantially better results than the spacing of 10^{-3} (as already mentioned in Section 3.3.3.1). The advantages of the coarser grid are:

(1) a simpler code is generated for each waveform (containing typically one-tenth the number of chain elements)

(2) class separation is somewhat better (57% against 49% correct classifications)

(3) the optimal value of the smoothing parameter is somewhat larger (0.7 against 0.5). In contrast, the finer grid yielded only 44% correct classifications with the smoothing parameter set to 0.7.

The feature set composed of the semi-global slope counts alone yields a performance second only to samples-as-features, but, again, with a low value of the smoothing parameter.

Finally, all four confusion matrices indicate similar patterns of inter-class confusion, with pits and gouges being the two classes most confused. If the distinction between these two classes was not necessary, results would be numerically much better. For example, samples-as-features would then yield 80% correct classification, and semi-global slope counts 82%. Since, in practice, these two classes are confused by trained human inspectors, these results are encouraging.

Elements-as-features

For this feature set, the grid spacing was set to 3.10^{-2}, as discussed in Section 3.3.3.1, and the chain elements were recoded as shown in Figure 3.8. The best result achieved was with the smoothing parameter set to 0.9, yielding 36% correct classifications. This is easily the worst result amongst all five feature sets, and merits closer analysis. Notice that it is very much worse than that achieved with the semi-global slope counts, although the latter feature set is derived from this one with a substantial information reduction.

It has been pointed out in Chapter 2, that feature space techniques almost invariably rely upon a distance measure defined on the space. With Specht's classifier, Euclidean distance is used. It seems that the poor results achieved with elements-as-features are attributable to the poor correspondence between Euclidean distance in this space and a meaningful measure of the difference between chain-encoded defect signals.

Consider first isolated chain elements coded as shown in Figure 3.8 (second assignment). The one-dimensional distance between element 0 and element +1 is then equal to that between +1 and +2. This seems reasonable in the context of a square grid, but "square" has no meaning in this work, as already discussed. It seems wholly unreasonable in the context of grid cells of height 1 mm and width 1 cm.

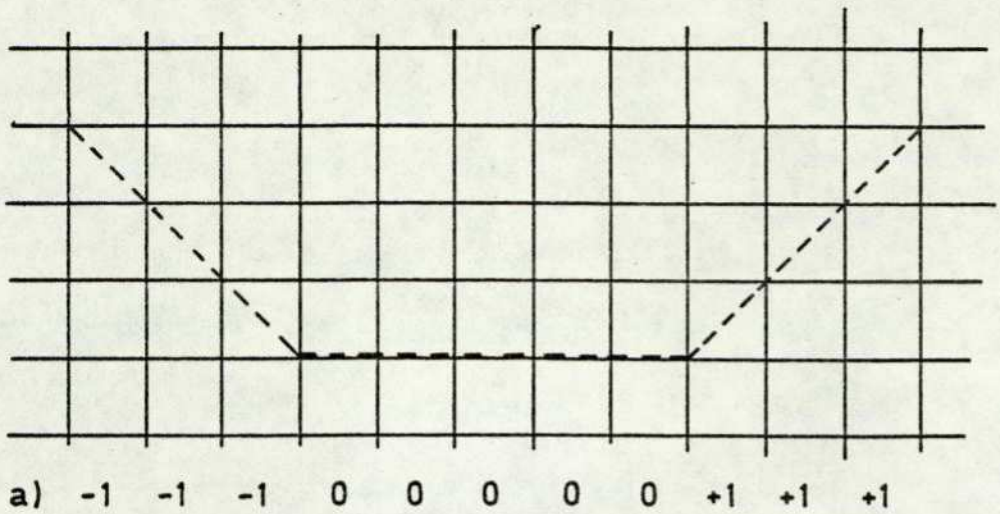
With isolated chain elements, then, the concept of Euclidean distance is suspect, but these suspicions are compounded when complete chain-encoded waveforms are considered. Figure 3.10 shows three such waveforms. Taking each chain element as a feature defines an 11-dimensional space in which the three waveforms are represented by the following three vectors:

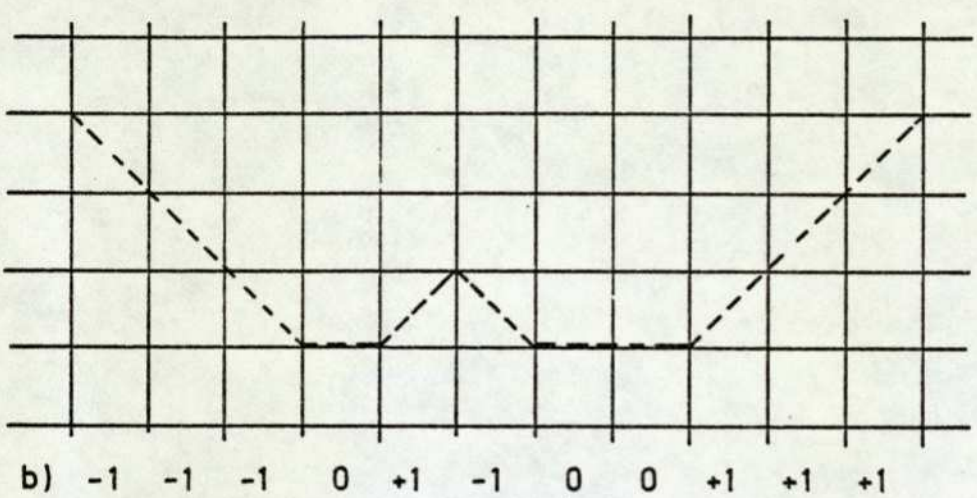
(-1, -1, -1, 0, 0, 0, 0, 0, 1, 1, 1) - (a)

(-1, -1, -1, 0, 1, -1, 0, 0, 1, 1, 1) - (b)

(-1, -1, -1, 0, 1, 1, 0, 0, 1, 1, 1) - (c)

The Euclidean distance between (a) and (b) is then equal to





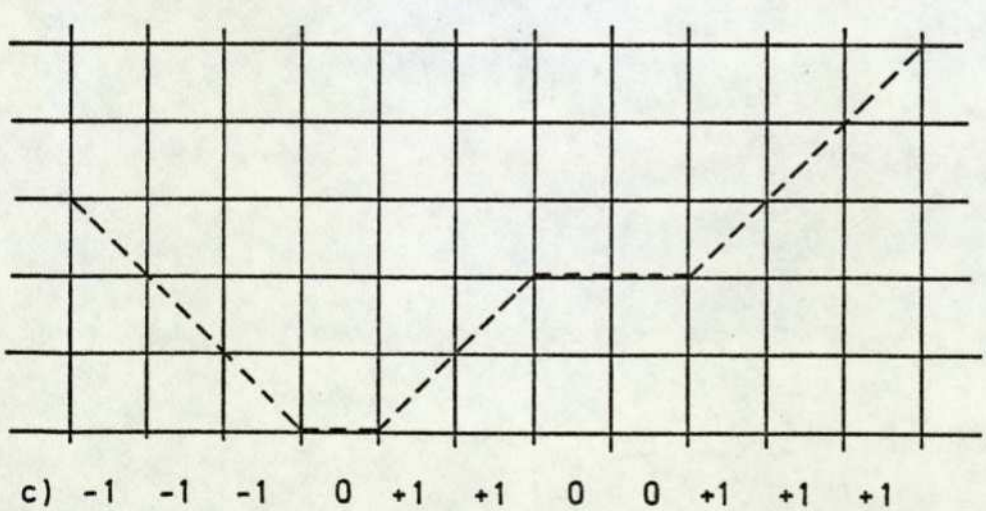


FIGURE 3.10 - CHAIN CODE SIMILARITIES.

that between (a) and (c) (= $\sqrt{2}$). This clearly does not reflect the real differences between the three waveforms, since (a) is certainly more like (b) than (c).

A suitable distance measure, which reflects intuitive notions of waveform similarity, is not easy to discover. Because of this, an alternative formalism for classification has been sought, which is capable of using chain-encoded waveforms directly. This will be discussed in Section 3.5.

3.4.2 Results with Feature Selection

The previous section has reported the results achieved with Specht's classifier with various pre-defined sets of features. These results were encouraging but, in some cases, involved spaces of high dimensionality. With samples-as-features, for example, a 50-dimensional space was involved. Such high dimensionality is undesirable for two main reasons:

(1) It leads, inevitably, to a complex realisation of the classifier

(2) The number of waveforms required to estimate reliably the class distributions in the space increases with dimensionality, as discussed in Section 2.3.7.

It seems likely that such high dimensionality is not, in fact, necessary for practical problems such as this. Further, such a large feature set is almost sure to contain a number of features which actually increase confusion between classes (Section 2.3.6).

These considerations point to the need for feature selection. In this context, feature selection will mean the process of sifting through a supplied set of "candidate" features, so as to determine a subset of these which yields good performance. The difficulties involved in such a process are discussed in Section 2.3.6, and the main ones are:

(1) A combination of two or more features or sets of features, each of which yields poor performance

in isolation, may yield good performance when used together

(2) For a candidate set containing N features, $(2^N - 1)$ subsets exist. In most cases, evaluation of all subsets is not feasible, and a non-exhaustive search strategy must be used

(3) In general, a structured performance function over the possible subsets, such as to allow interpolation between subsets, does not exist.

In these circumstances, a "without-replacement" search is applicable (Section 2.3.6) and this has been used.

Various techniques have been tried for feature normalisation before selection, and for optimising the smoothing parameter throughout the selection process, and these will be described as they occur.

3.4.2.1 Specht's Classifier with Samples-as-Features

In the results without feature selection, samples-as-features proved to be most effective. Since this feature set is simultaneously the most simple and cheap, most of the feature selection work has been based upon it.

In Section 2.3.6, different possibilities for feature normalisation have been discussed. Attention centres on two of these:

(1) Normalise each feature (independently) to zero mean and unit standard deviation over all waveforms from all classes (Normalisation 1)

(2) Normalise each feature (independently) to zero mean and unit average standard deviation per class (Normalisation 2).

Of the two, it was suggested that the second was the more appropriate for Specht's classifier. In the work without feature selection, only the first normalisation was used. In the work to be described now, however, the opportunity has

been taken to compare the results attainable under the two normalisations. We shall see that the second does, indeed, seem to be preferable.

Before presenting these results, one further variation must be discussed - namely, the strategies adopted to optimise the value of the smoothing parameter throughout the selection process. Initially, this parameter was optimised before every selection (Optimisation 1), but for later results optimisation was based on simply comparing the results for a number of preset values, which were maintained constant throughout the entire selection process (Optimisation 2). The reasons for this change of strategy will be discussed within the results.

Normalisation 1, Optimisation 1

Over a variety of different techniques for optimising the smoothing parameter before every selection, the best result achieved was 77% correct classifications, using samples 2, 3, 4, 5, 6, 10, 14 and 16, and a smoothing parameter of 1.3. Figure 3.11 shows the corresponding confusion matrix. The equivalent result without feature selection has been presented as Figure 3.9A. Two points are worth making:

(1) Performance, at 77% as compared to 64%, is significantly better

(2) The classifier is substantially simpler, both because the space is 8-dimensional instead of 50-dimensional, and because the smoothing parameter value is 1.3 instead of 0.4.

Such a two-fold gain, with no apparent loss, underlines the value of feature selection processing.

Normalisation 2, Optimisation 1

With the second normalisation, the best result achieved was again 77% correct classifications, but in a space of only 4 dimensions (samples 2, 5, 6 and 14) and a smoothing parameter of 1.0. The corresponding confusion matrix was

True class \ Assigned class	1	2	3	4	5
1	13	0	0	0	0
2	5	4	0	1	0
3	2	0	5	0	0
4	0	0	0	6	0
5	1	0	0	0	2

Samples-as-features

Dimensionality = 8

Smoothing par. = 1.3

77% correct classifications

Class 1 = Pits,
Class 2 = Gouges,
Class 3 = Rust,
Class 4 = Scale,
Class 5 = Heavy lamination

Figure 3.11 Confusion matrix with feature selection (Specht's Classifier)

substantially the same as that of Figure 3.11 and will not be reproduced.

The difference between the two normalisations, then, is reflected in half the number of features being required with the second. This represents a simplification which is not insignificant.

Normalisation 2, Optimisation 2

To recapitulate, Optimisation 2 refers to a procedure wherein the value of the smoothing parameter is preset to a certain value and the entire process of feature selection carried out without varying that value. On completion, the value is reset and the procedure repeated, and so on. Two considerations motivated this change of strategy:

(1) Optimisation before the selection of each feature, as with Optimisation 1, is computationally expensive. Consider that for N features, $N(N + 1)/2$ feature combinations must be evaluated for a complete without-replacement search. For each of these combinations, around ten different values of the smoothing parameter should be considered, and for each of these ten values, the leave-one-out performance estimate must be computed. Such a scheme is computationally tolerable with a small data set, but not with one of realistic size.

(2) At each stage of the selection process, with Optimisation 1, the "best" value of the smoothing parameter is available. It is not certain, however, that this "best" value is acceptable in terms of the associated complexity of the classifier. An important question which the procedure leaves unanswered is "what performance is possible with a smoothing parameter of value 4.0, say, and how does this performance vary with different pre-determined values?"

Such questions are, of course, answered directly by the second optimisation strategy.

For these reasons, the second strategy was implemented. Figure 3.12 summarises the results. Notice two aspects in particular:

(1) The relatively small performance variation (77% to 74%) over a wide range of preset values of the smoothing parameter

(2) The reduction in the number of features necessary for peak performance, as the value of the smoothing parameter is increased.

These results suggest that a significant simplification can be made in the classifier, with only a modest loss of performance.

3.4.2.2 Specht's Classifier in Polynomial Form

All the results presented in the previous section have been with Specht's classifier in exponential form. This form, however, would not be practical for on-line application, and the real promise of Specht's technique lies in its eventual realisation in polynomial form.

The formulae by which the polynomial coefficients may be calculated as sums over the design set, have been presented in Section 2.3.4. Accordingly, these calculations have been embodied in a program which may be used to compare the performance of the exponential form with the polynomial form, calculating and printing the polynomial coefficients in the process.

Program input consists, as before, of a data matrix in which rows correspond to waveforms and columns to features. In addition, a feature subset must be specified (since not all features need be used) together with a value for the smoothing parameter. This input constitutes a complete specification for the exponential form of the classifier. The program will then implement the exponential form and

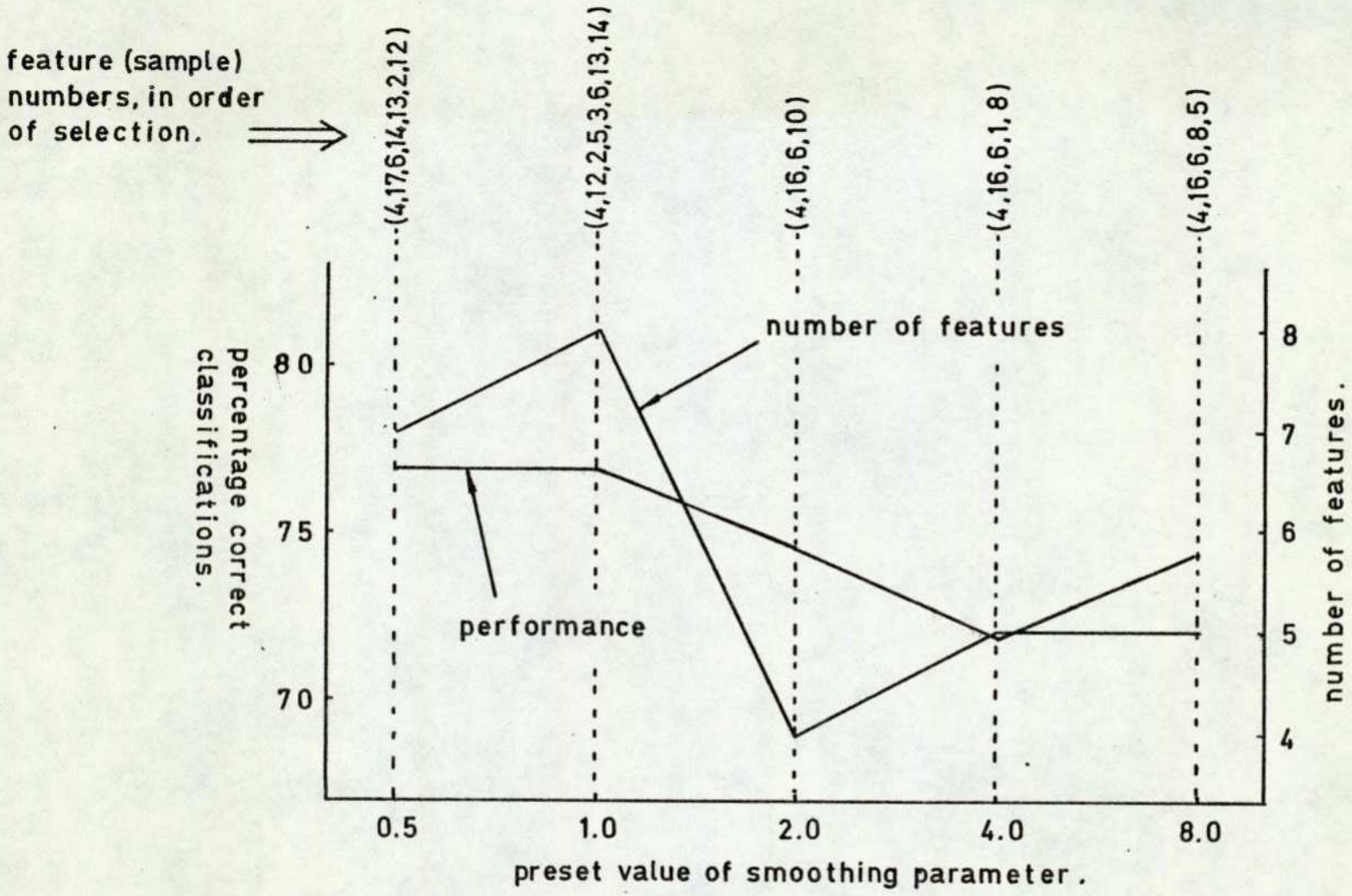


FIGURE 3.12 - FEATURE SELECTION WITH PRESET SMOOTHING PARAMETER AND SAMPLES-AS-FEATURES.

compute its performance:

(a) directly on the design set;

(b) "leave-one-out" on the design set.

(Although with this data set a separate test set does not exist, the program allows such a test set to be specified. In this case, performance on the test set would also be computed.)

The program continues by computing and printing all polynomial coefficients up to and including third order. With these coefficients, the polynomial form of the classifier is implemented and its performance computed, as before:

(a) directly on the design set;

(b) "leave-one-out" on the design set

(and if a separate test set were specified, polynomial performance on the test set would also be computed).

To provide an indication of how far the polynomials might be truncated without significant loss of performance, the details described above are computed for the polynomials truncated to first order and to second order, as well as with the complete polynomials of third order.

This program follows naturally the feature selection program, since results from that program specify a feature subset and a corresponding value for the smoothing parameter. The set of three results presented in Section 3.4.2.1 were therefore examined further with the polynomial program. These are:

(1) Normalisation 1, Optimisation 1
(Figure 3.11)

(2) Normalisation 2, Optimisation 1

(3) Normalisation 2, Optimisation 2
(Figure 3.12).

Results are summarised in Figure 3.13.

		Percentage correct classifications	
		On the design set	Leave-one-out on the design set
(i)	Smoothing parameter = 1.3 Dimensionality = 8 (Normalisation 1, Optimisation 1)		
	Exponential form	82	77
	Polynomial form (1st order	46	44
	Polynomial form (2nd order	69	59
	Polynomial form (3rd order	77	64
(ii)	Smoothing parameter = 1.0 Dimensionality = 4 (Normalisation 2, Optimisation 1)		
	Exponential form	85	77
	Polynomial form (1st order	49	44
	Polynomial form (2nd order	44	39
	Polynomial form (3rd order	61	54
(iii)	Smoothing parameter = 8.0 Dimensionality = 5 (Normalisation 2, Optimisation 2)		
	Exponential form	74	74
	Polynomial form (1st order	74	74
	Polynomial form (2nd order	74	74
	Polynomial form (3rd order	74	74

Figure 3.13 Performance of the polynomial classifier with samples-as-features

Two points are significant in these results. First, the performance of the exponential form does not extrapolate to the polynomial form, except in the final case with a smoothing parameter of value 8.0. Second, in this final case the polynomials can be truncated to be linear without degrading the performance at all. These results are unexpected and disturbing. In particular, Specht's technique seems to be a particularly circuitous route by which to arrive at a simple linear classifier. Simpler and more direct methods are available.

3.4.2.3 Specht's Classifier with other Feature Sets

All of the results presented so far with the feature selection process havebeen with the single feature set, samples-as-features. This has been so because this feature set had yielded the best performance without feature selection. The selection process has been seen to enhance its performance considerably. It remains to be seen whether features based on geometric measures and/or on the chain-encoding process can yield a similarly enhanced performance when combined with the feature selection process.

To this end, the four geometric features were combined with the five global slope counts (grid spacing 10^{-2}) for processing by the feature selection program, with Normalisation 2, Optimisation 1 (Section 3.4.2.1). The best performance achieved was 67% correct classifications, using the six features base width, amplitude, area, and three global slope counts. The corresponding confusion matrix is shown in Figure 3.14, which may be compared to the equivalent result without feature selection, Figure 3.9C. As before, the selection process allows better performance to be achieved with a simpler system. Even so, the result is worse than that achieved with samples-as-features (Figure 3.11).

True class \ Assigned class	1	2	3	4	5
1	10	3	0	0	0
2	5	2	1	0	2
3	0	0	7	0	0
4	0	1	0	5	0
5	0	1	0	0	2

Geometric features + global slope counts

Dimensionality = 6

Smoothing par. = 0.4

67% correct classifications

Class 1 = Pits,
Class 2 = Gouges,
Class 3 = Rust,
Class 4 = Scale,
Class 5 = Heavy lamination

Figure 3.14 Confusion matrix with feature selection (Specht's classifier)

True class \ Assigned class	1	2	3	4	5
1	13	0	0	0	0
2	4	6	0	0	0
3	2	0	5	0	0
4	0	2	0	4	0
5	0	1	0	0	2

Samples-as-features

Dimensionality = 7

77% correct classifications

Class 1 = Pits,
Class 2 = Gouges,
Class 3 = Rust,
Class 4 = Scale,
Class 5 = Heavy lamination

Figure 3.15 Confusion matrix with feature selection (linear classifier)

3.4.2.4 The Least-Mean-Square Linear Classifier with Samples-as-Features

The performance of Specht's classifier in exponential form seems not to extrapolate to its polynomial form, unless the value of the smoothing paramater is not less than approximately 8.0. With such a value, the polynomials need be only first order, with no improvement to be gained by including second or third order terms. In other words, we have a simple linear classifier, arrived at by an indirect and computationally expensive route.

Direct design methods for linear classifiers were discussed in Section 2.3.5, and the advantages of the least-mean-square technique were emphasized. This technique has been programmed and incorporated into the feature selection program, in place of Specht's technique. The program was run with samples-as-features and Normalisation 1 (as explained in Section 2.3.6, the particular normalisation used with this classifier is unimportant, and serves only to prevent overflow/underflow problems during computation). Best performance was 77% correct classifications in a space of 7 dimensions (samples 4, 5, 6, 10, 15, 16 and 19). The corresponding confusion matrix is shown in Figure 3.15. This result is as good as the best performance of Specht's classifier in exponential form, and better than the polynomial form.

3.5 A Tree-Classifier for Chain-Encoded Waveforms

Work so far reported with chain-encoded waveforms has been based on feature space techniques, with three feature sets:

(1) Global slope counts
(2) Semi-global slope counts
(3) Elements-as-features.

Reasonable results were obtained with (1) and (2), but not with (3). In Section 3.4.1 it was argued that the element codes are ill-suited to feature space techniques, and that an alternative

formalism was required. It is the purpose of this section to develop such a formalism and to evaluate it on the data available. The motivation for this work lies in the substantial reduction of information offered by the encoding process, and a belief that this is exploitable.

The generation of a chain code of N elements for a particular defect waveform may be seen as an N-level sequential process, with five possibilities at each level (the five possible chain elements). Such a process may be represented by a decision-tree, as shown in Figure 3.16. Every possible chain code then corresponds to a unique path within such a tree, and all possible defect waveforms are therefore represented by the complete tree of appropriate depth.

This concept can be used for pattern recognition if a separate tree is defined for each pattern class, and for each tree a weight is defined for each path within it. This corresponds to deriving a "class-weight" for each possible chain-encoded defect signal, and the weights can be such that the largest identifies the correct pattern class.

If such identifications are to be optimum, in the sense of minimising the expected loss due to errors, statistical decision theory (Section 2.3.2) shows that the weight assigned to any sequence of chain elements $(e_1, e_2, -, e_N)$must be the class-conditional probability of that sequence occurring (assuming equal a-priori class probabilities and equal error costs). This probability may be written as:

$$Pr(E/W_i)$$

for class W_i, where E denotes the sequence of chain elements $(e_1, e_2, -, e_N)$.

Although, in principle, these probabilities may be estimated from a design set, the number of possible chain sequences renders this approach impractical. Consider that for 50 chain elements, with five possible values for each one, 5^{50} different sequences can occur. This number is greater than 10^{30}, and with less than 10^{16} microseconds in a century these optimum weightings are clearly not attainable.

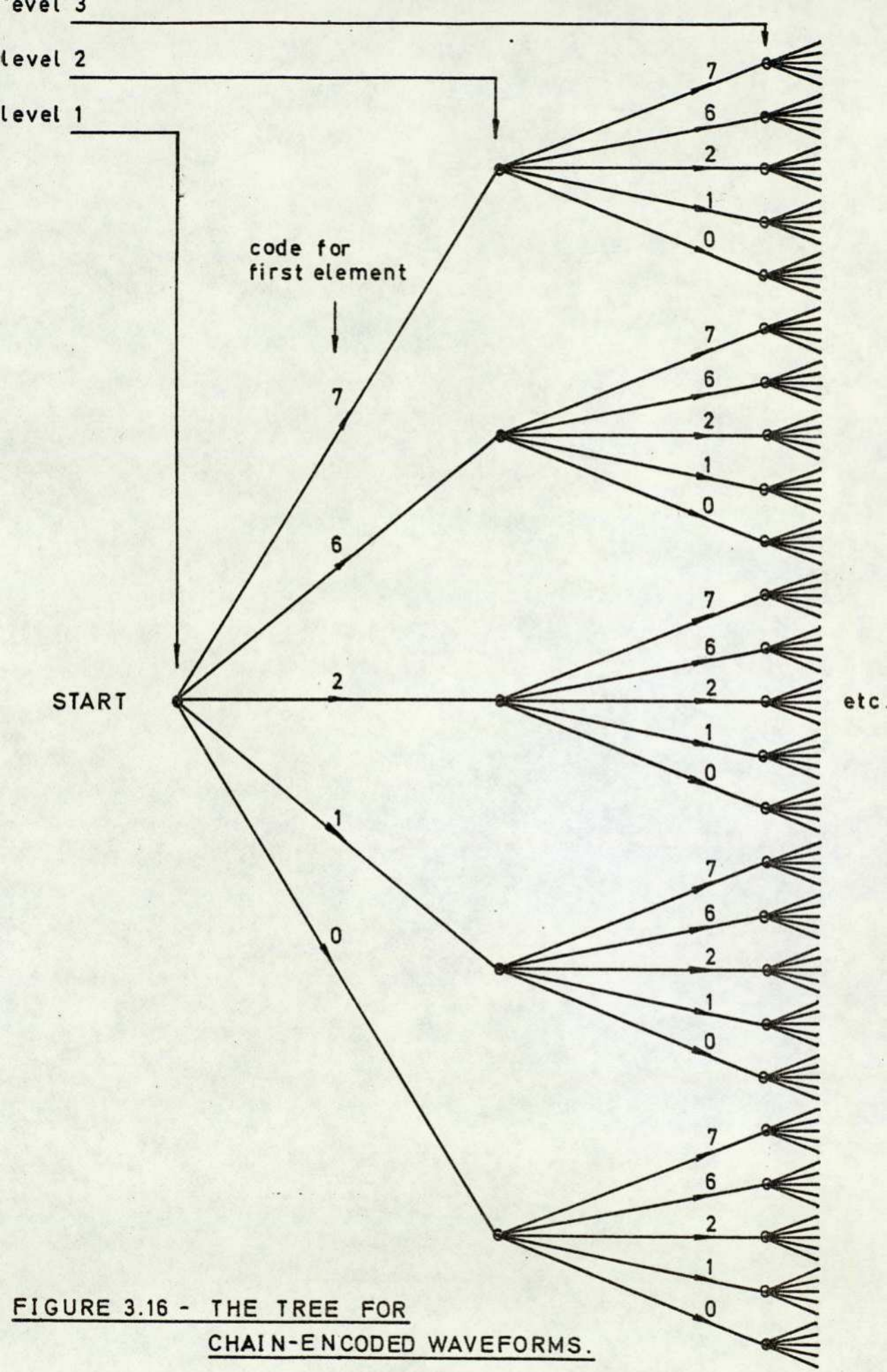


FIGURE 3.16 - THE TREE FOR CHAIN-ENCODED WAVEFORMS.

Pruning of the trees is therefore required, and this can be achieved by assuming some form of reduced dependency among the sequence of elements. In reality, the probability that any chain element in a sequence will take on a particular value, is almost certainly dependent upon the whole sub-sequence preceding (and possibly succeeding) that element, and this leads to the explosive estimation problem referred to above. It is not unreasonable, however, to expect dependencies to be stronger for near-neighbours of the element, than for elements further removed. Given this, the theory of Markov processes provides a framework within which the dependence assumptions may be relaxed.

3.5.1 First-order, Non-homogeneous Markov Dependence

assumes that the chain code generation process for defect signals is such that the class-conditional probability distribution over the five possible codes for any chain element, depends only on the code of the preceding element (first-order) and the element position in the sequence (non-homogeneous).

i.e. $\Pr(e_K/e_{K-1}, e_{K-2}, -, e_1, W_i)$

$= \Pr(e_K/e_{K-1}, W_i)$

and is a function of K.

Thus: $\Pr(E/W_i)$

$= \Pr(e_1, e_2, -, e_N/W_i)$

$= \Pr(e_1/W_i) \cdot \Pr(e_2/e_1, W_i) \cdot$

$\Pr(e_3/e_2, W_i) \cdot - \cdot \Pr(e_N/e_{N-1}, W_i)$

and the probability of any chain sequence may be computed from a limited number of transition probabilities:

$$\Pr(e_K/e_{K-1}, W_i)$$

plus the initial probability:

$$\Pr(e_1/W_i).$$

For a maximum sequence length of 50 elements, therefore, this scheme requires that 1,230 probabilities be estimated and stored for each class.

3.5.2 First-order, Homogeneous Markov Dependence

as 3.5.1, except that the transition probabilities are assumed independent of the element position in the sequence.

i.e. $\Pr(e_K/e_{K-1}, W_i)$ is independent of K.

Thus:
$$\Pr(E/W_i) = \Pr(e_1/W_i) \prod_{K=2}^{N} \Pr(e_K/e_{K-1}, W_i)$$

and only 30 probabilities need to be estimated and stored for each class, regardless of the maximum sequence length.

3.5.3 Second-order, Homogeneous Markov Dependence

as 3.5.2, except that the probability distribution for any chain element is assumed to be dependent on not only the previous element but also on the one preceding that.

i.e. $\Pr(e_K/e_{K-1}, e_{K-2}, -, e_1, W_i)$

$$= \Pr(e_K/e_{K-1}, e_{K-2}, W_i)$$

and is independent of K.

Thus:
$$\Pr(E/W_i) = \Pr(e_1/W_i) \cdot \Pr(e_2/e_1, W_i) \prod_{K=3}^{N} \Pr(e_K/e_{K-1}, e_{K-2}, W_i)$$

giving 155 probabilities to be estimated and stored for each class, regardless of the maximum sequence length.

Notice that with each one of these assumptions, the probability assigned to any path within a tree is evaluated as the product of probabilities assigned to each individual transition making up that path. This renders the estimation and storage problem feasible. The assumption of homogeneity throughout each path reduces complexity still further by reducing the number of individual transition probabilities involved.

3.5.4 Simulation and Results

The two schemes of first and second order homogeneous Markov dependence require the least number of transition probabilities to be estimated and stored. These have therefore been programmed and evaluated on the available data.

Program input consists of the code sequences for the waveforms, together with the correct classification of each one, and a specification of the order of Markov dependence to be assumed.

The design process consists of estimating the necessary transition probabilities as relative frequencies in the design set. Leave-one-out estimation was implemented by modifying the probabilities estimated on the complete design set, so as to remove the contribution of the waveform to be classified. Results are shown in Figure 3.17, for a grid spacing of 3.10^{-2}. In both cases they are very poor, being little better than a random guess at class membership.

The clue to this poor performance lies in the number of "ambiguous classifications" which occurred. There were 6 of these with first order dependence, and 23 with second order. Such ambiguity arises when a waveform yields equal class membership probabilities for two or more classes, and is arbitrarily resolved by assigning the waveform to the class considered last. Thus, if such ambiguity holds over all five classes, a waveform would be assigned to the final class considered - "heavy laminations". From Figure 3.17B, this seems to have occurred for every waveform from "gouges" and "scale", and probably for eight waveforms from "pits". Such ambiguity should properly be regarded as an inability to reach a decision, rather than a wrong classification.

A detailed examination of the data set reveals an explanation. This is that many of the waveforms yield a code sequence which contains a unique three element sub-sequence, i.e. a sub-sequence which does not occur for any other waveform. With the leave-one-out technique, the transition probability for such a sub-sequence will be estimated as zero. With the multiplicative

True class \ Assigned class	1	2	3	4	5
1	6	3	1	2	1
2	4	0	1	1	4
3	2	3	2	0	0
4	1	2	0	1	2
5	1	0	0	0	2

(A) First-order dependence

28% correct classifications

True class \ Assigned class	1	2	3	4	5
1	3	2	0	0	8
2	0	0	0	0	10
3	1	1	2	0	3
4	0	0	0	0	6
5	0	0	0	0	3

(B) Second-order dependence

20% correct classifications

Class 1 = Pits,
Class 2 = Gouges,
Class 3 = Rust,
Class 4 = Scale,
Class 5 = Heavy lamination

Figure 3.17 Confusion matrices with the tree-classifier (leave-one-out estimates)

calculation of class membership probabilities, this ensures a zero product for each class, and hence the ambiguous classifications observed. This effect is simply the result of insufficient design data, and should disappear with more data. Notice that the feature space techniques avoid such problems by extrapolating between design samples, using the appropriate distance measure.

To investigate this effect further, the programs were rerun, but with all 39 waveforms used to estimate the required probabilities, including the waveform being classified. Results are shown in Figure 3.18. These support the analysis above, since under these circumstances such unique subsequences guarantee the correct classification of the waveform.

Overall, then, these results are inconclusive, and a proper evaluation of the classifier must await more data.

3.6 Summary and Conclusions

This chapter has reported work with two feature space classifiers on a few selected feature sets, and one tree-classifier on symbolic data.

The work began with quite firm opinions as to the nature of the signal processing problems involved and clear ideas as to the most promising techniques to solve them. Programs have been developed to implement these techniques and they have been explored on a very small data set. It would certainly have been possible, and in some ways easier, to explore the techniques through the processing of artificial data. To a large extent, however, the results would have been predictable. The critical results arise from the interaction of these techniques with the inspection problem itself. This interaction cannot be explored with artificial data. It cannot be adequately explored even with real data, when so little is available. Nonetheless, one can hope for some useful indications. Apart from this, the work has resulted in a useful collection of programs and some familiarity with the techniques and the programs which implement them.

True class \ Assigned class	1	2	3	4	5
1	10	2	0	1	0
2	2	6	1	1	0
3	0	0	7	0	0
4	0	0	0	6	0
5	0	0	0	0	3

(A) First-order dependence

82% correct classifications

True class \ Assigned class	1	2	3	4	5
1	13	0	0	0	0
2	0	10	0	0	0
3	0	0	7	0	0
4	0	0	0	6	0
5	0	0	0	0	3

(B) Second-order dependence

100% correct classifications

Class 1 = Pits,
Class 2 = Gouges,
Class 3 = Rust,
Class 4 = Scale,
Class 5 = Heavy lamination

Figure 3.18 Confusion matrices with the tree-classifier (re-substitution estimates)

One conviction quite firmly held at the beginning of this work was that complex class distributions were sure to be encountered, no matter what features were used, simply because of the variation observable within each defect class. It was for this reason that Specht's classifier was selected for evaluation, since it is expressly designed to cope with such distributions, while retaining reasonable simplicity in its implementation. On this data, these expectations are not fulfilled. The expansion of the classifier into polynomial form, as required for its implementation, results in a significant loss of performance, unless the smoothing parameter has a very large value. In this case, the power of the classifier is severely constrained, and only first order polynomials are required for its approximation.

This observation prompted a search for a more direct method to design a linear classifier, and the least-mean-square technique was chosen. Results obtained with this technique were found to be superior to the best from Specht's classifier in its polynomial form.

As to the different feature sets, a similar pattern has emerged. The simplest, samples-as-features, has yielded the best results. The technique adopted for feature selection seems to be very valuable indeed, producing a simpler classifier with improved performance. With a larger data set, the ability to use a smaller number of features would also allow more confidence to be placed in the results.

The dismal performance achieved with elements-as-features led to the development of a tree-classifier operating under the assumption of Markov dependence within symbol strings. Unfortunately, the limited data set did not allow this classifier to be properly evaluated, and this must await more data.

At this stage, these results must all remain open to dispute. In the next chapter, work will be described involving the acquisition of a larger data set, and the pursuit of the main questions raised by this exploratory phase of the investigation.

4. VALIDATION ON A LARGER DATA SET

4.1 Introduction

Chapter 3 has described exploratory work based on a small data set from cold-rolled steel strip. This work has generated certain unexpected results and suggested particular areas which merit further attention. At the same time, a degree of insight has been gained which suggests the elimination of certain lines of attack, so as to examine more thoroughly the most promising.

Accordingly, the work reported in this chapter is concerned primarily with the following topics:

(1) The relatively simple linear classifier has been shown, on a small data set, to be as effective as, or more effective than, the more complex polynomial scheme of Specht - given suitable feature selection in each case. Is this unexpected result merely an anomaly due to insufficient data, or will it hold true on a more substantial data base?

(2) Similarly, the very simple feature set which comprised merely the raw digital samples across the defect waveform, has been shown to be more effective than the more complex set of geometric features - again, on a small data set. Will this carry over to a larger data base?

(3) The process of chain-encoding a defect waveform achieves a significant reduction in its information content. The encoded waveform, however, cannot be used directly for feature space recognition, in the same way as the original sampled waveform can. The tree-classifier can classify directly the chain-encoded waveform, but the results on a small data set are equivocal. How will this classifier perform on a larger data base?

(4) With each classifier, the system can be extended to allow indecision (rejection of the waveform as unclassifiable). If this is done, a reduction can be achieved in the error-rate. For each classifier,

what trade-off is possible between the error-rate and the reject-rate? In the limit, what percentage of the waveforms must be rejected as unclassifiable, if no waveforms are to be wrongly classified?

For topics (1) and (3), the study will be limited to the geometric features and samples-as-features. The definition and calculation of the geometric features can, however, be rationalised and this will be discussed in Section 5.2. In addition, the smoothing parameter in Specht's classifier will be optimised through the approach typified by Fig. 3.18; this being the most informative and versatile of the various strategies explored in Chapter 3.

4.2 The Data Gathering Process

The problems involved in setting up a suitable data base for the work described in this chapter have proved to be among the most difficult and frustrating of those encountered in this project. In retrospect, it seems that such problems are inherent in work of this nature, and therefore merit a brief discussion.

The first task is to select and define the problem to be tackled. This involves selecting a material for which good surface inspection is important, for which existing (largely manual) methods of inspection are not wholly satisfactory, and for which a useful solution to the automatic inspection problem seems possible. After consultation with staff of the SIRA Institute and the British Steel Corporation, it seemed that these criteria were met by several materials, but that sheet tinplate was probably the most suitable.

Perhaps the most important, and the most difficult, part of defining the problem to be tackled is to agree on the defects to be recognised, and the required accuracy of recognition. A whole range of opinions is held by different interested parties, ranging from 100% accuracy on every defect (about 30 types on tinplate) to sorting the material into just two classes - accepted or rejected, however these might be defined. Eventually, it was decided to concentrate on no more than ten of the most common defect types.

For recognition accuracy, a suitable standard was taken to be the performance of a human inspector in a typical on-line environment. Assigning a figure to this performance is fraught with difficulties, not least for reasons of commercial secrecy. Nonetheless, a reasonable estimate seems to be between 60% and 80% of the defects on the surface detected and correctly identified.

Having defined the problem, it becomes necessary to gather a suitable set of material to define the defects of interest, and on which to evaluate candidate techniques. In this context, a suitable set of material must contain a representative selection of each defect type. The defect samples must not be especially atypical, and must be sufficient in number. Naturally, to do this job properly requires considerable effort, not only by the research worker, but also by production, inspection and research staff of the manufacturer of the product. Such collaboration can be difficult to achieve, especially when the manufacturing process is fully stretched to meet demand, and must accordingly take priority over research and development. Despite these difficulties, approximately 30 sheets of tinplate, each about 1 metre square, were diverted from the production line and identified by BSC inspectors. Five main defect types were represented, with each sheet containing several examples of, typically, two or three different defects. These sheets were transported to the SIRA Institute to be scanned.

The sole aim of the scanning process is to generate data as closely representative as possible of the on-line situation. The laser scanner described in Chapter 1 represents state-of-the-art technology, and so it was decided to use this and to operate it as near as possible to the on-line scan rate. To this end, the scanner was connected via a fast analogue-to-digital convertor and a direct-memory-access interface unit, to a PDP-11 minicomputer. This allowed the scanner analogue output signal to be digitised at a 1 MHz sampling frequency and held in the PDP-11 core store. With this sampling frequency, the scan rate was set to 10^3 scans/second, allowing approximately 1000 samples in each scan, i.e. one sample per mm on the surface. The binary "trigger signal" from the SIRA defect detection system was simultaneously sampled

and stored. At the completion of a scan, the data was punched on to paper tape for subsequent transfer to the University of London Computer Centre (ULCC).

The scanner was mounted above a "rolling table" on which the tinplate sheet was placed. The table was first positioned so that the scan line was along the sheet leading edge, and the computer instructed to digitise that scan and record it on paper tape. The table was then manually adjusted to the next scan position, according to a scale along its edge, and that scan digitised and recorded. This process was repeated until the entire sheet had been scanned. The scanning spot size was set to about 1 mm by 1 cm and the table was moved by 5 mm between scans. This gives, nominally, 50% overlap of successive scans, but since the spot intensity distribution is gaussian, rather than rectangular, this measure is ill-defined.

With a sheet 1 m square, 200 scans were therefore necessary to cover it completely. This required some three hours' effort, and resulted in three full paper tapes being generated. It soon became apparent that the paper tape input facilities at ULCC were not suited to this quantity of data and that another input medium was necessary. The obvious candidate was magnetic tape, and so the paper tape data was transferred to magnetic tape using the large ICL computer at SIRA, before being input to ULCC.

A certain amount of processing was necessary at ULCC to complete the data gathering process. The magnetic tapes produced were non-standard to the ULCC operating systems, and it was advantageous to transfer the data on to standard ULCC tapes. Simultaneously, certain checks and corrections were applied. In particular, it was necessary to correct for a non-constant sampling frequency during digitisation, by normalising the number of samples in each scan of a sheet to the number in the first scan of that sheet. This was achieved by an interpolation process between the data samples actually recorded.

Each sheet (1 m square) yielded approximately 2.10^5 samples of data, and some form of validation, to detect failures in the overall data gathering process, was found to be necessary. A

suitable visual presentation of the data from each sheet was therefore devised, using the microfilm output facilities at ULCC. This took the form of a pair of pictures produced from the recorded data. The first picture of the pair displays the set of digitised analogue scans, in a pseudo-perspective mode - produced by plotting each scan shifted slightly with respect to its neighbour. Figure 4.1 shows such a plot. The second picture displays only the binary trigger signals derived from the analogue data. This picture has come to be known as a videoprint, and Figure 4.2 shows the videoprint derived from the data of Figure 4.1. Each line of the videoprint corresponds to a single scan, and the two levels of the resulting delineation signal are represented as black and white, with black for defects.

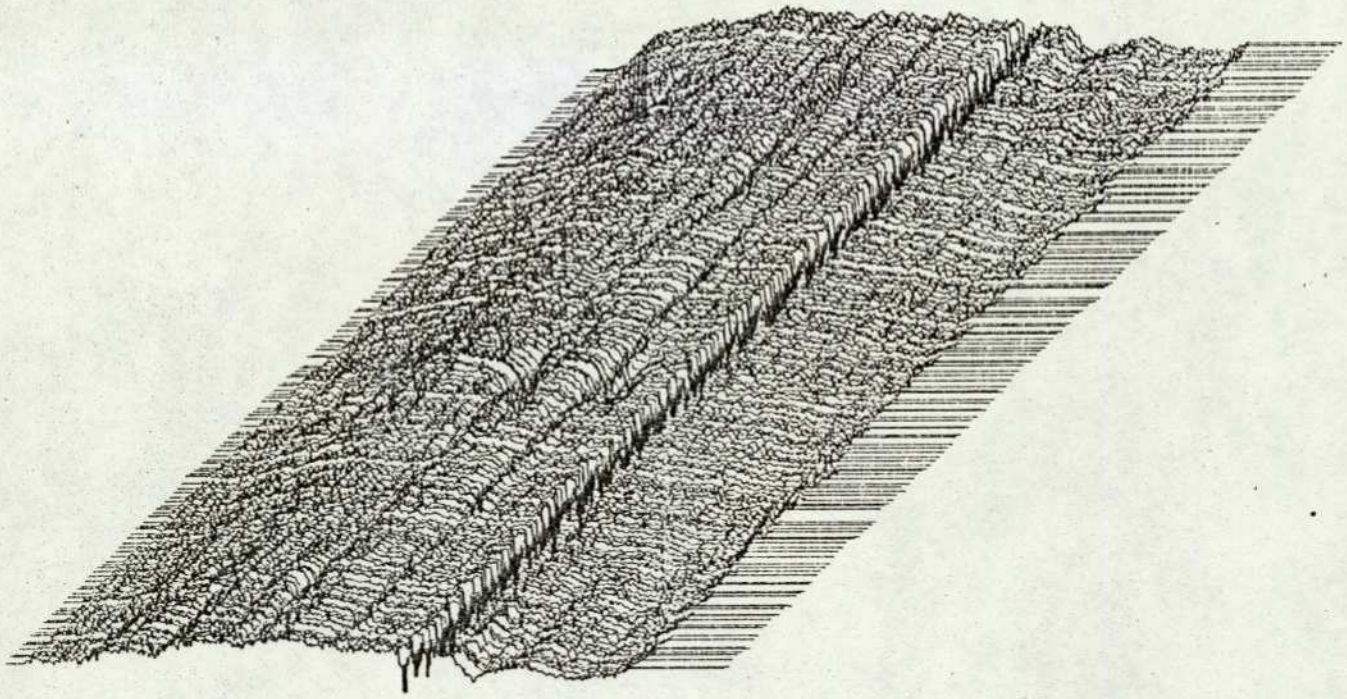
These pictures have proved invaluable for debugging and maintaining the overall system. Beyond this, however, they form an essential aid to the identification of the defect signals within the whole data set. This is achieved, primarily, by comparing the videoprint to the actual tinplate sheet, in collaboration with a person familiar with the defects. This allows the videoprint to be annotated as shown in Figure 4.3. With this information, the signals representing the various defect classes can be located and extracted for processing.

4.3 The Data Set

The data set for recognition studies consists of isolated defect signals, and it is necessary to locate, delineate and identify these signals in the totality of data available.

In Chapter 3 doubts were expressed about the SIRA detection system, insofar as it might be used to delineate (i.e. define the limits of) the defect waveform. These doubts could not be resolved then, because the detection signal was not available with that data set. With the tinplate data, this signal was available, and its effectiveness as a delineation signal could be examined.

Figure 4.4 shows a few defect signals, together with the response of the SIRA detection system to these signals. It is clear

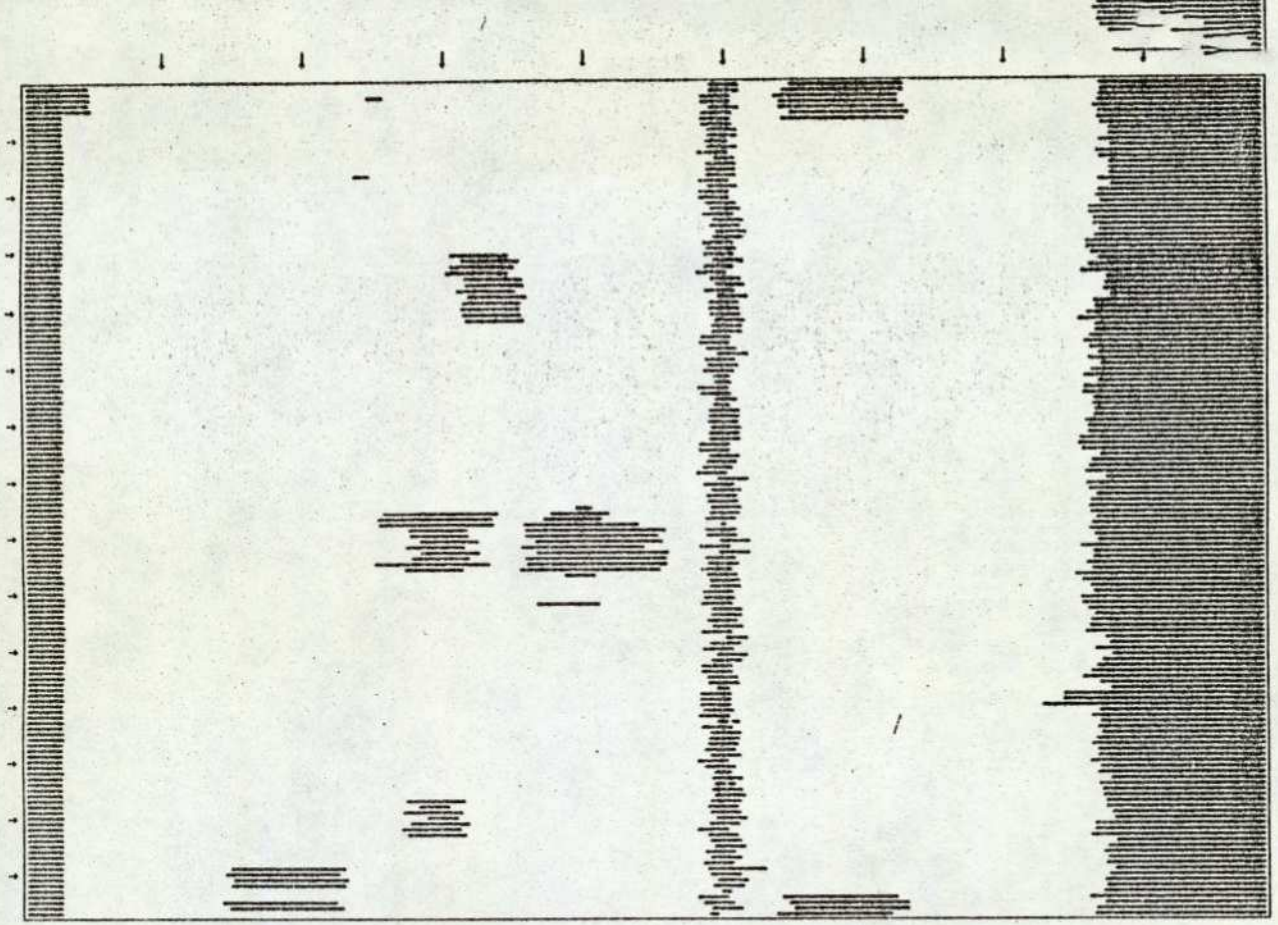


SHEET E4 - SPECULAR PERSPECTIVE.

D.C. LEVELS ADJUSTED.

VERTICAL SCALE FACTOR = 200. UNITS/INCH. 147 SCANS WITH 880 SAMPLES PER SCAN (MAX).

FIGURE 4.1 - ANALOGUE DATA.



SHEET E4 - SPECULAR VIDEOPRINT.

FIGURE 4.2 - THE VIDEOPRINT.

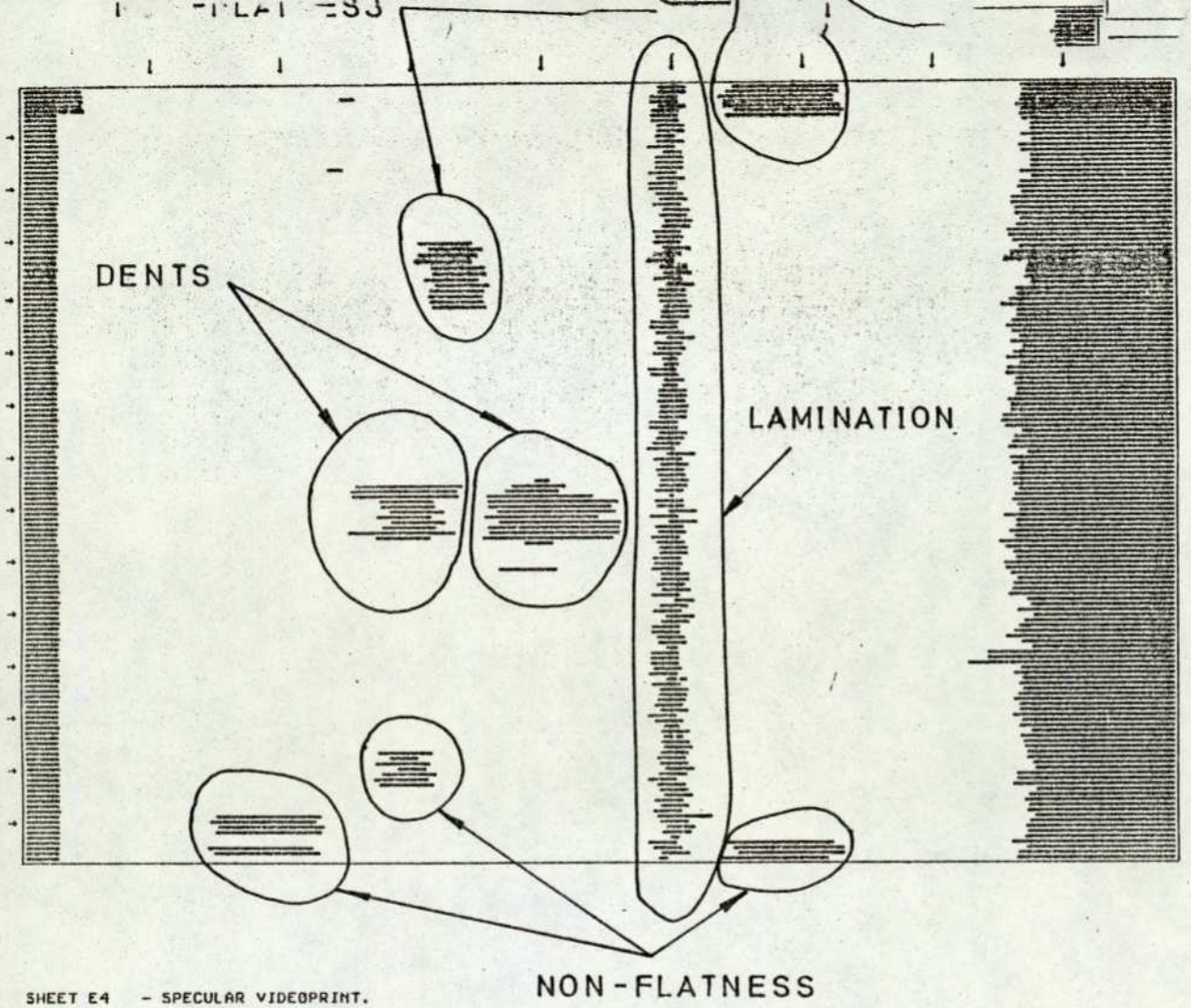


FIGURE 4.3 - MARKED-UP VIDEOPRINT.

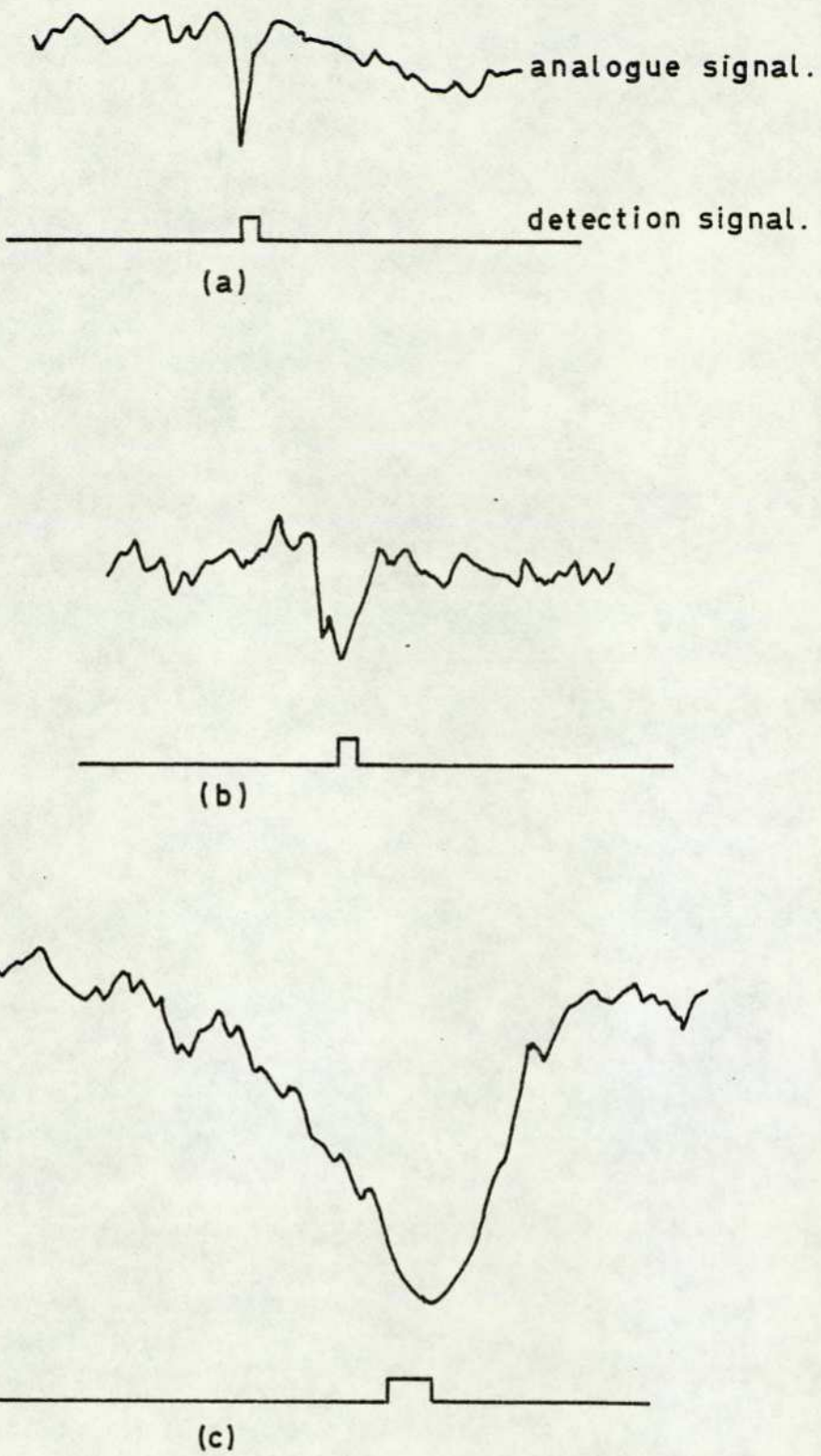


FIGURE 4.4 - THREE EXAMPLES OF THE RECORDED DETECTION SIGNAL.

that the detection signal has serious shortcomings for delineation. This matter will be pursued in Chapter 5, but for now the problem must be overcome, since the defect signals cannot be characterised without prior delineation. To this end, it was decided to make use of the SIRA detection signal simply to locate the defect waveforms in the data set, with the limits of each one subsequently determined from a visual examination of the waveform.

In this way, a total of 500 waveforms were located, delineated and identified by class. The identification process was based on the marked-up videoprints (Figure 4.3) already discussed. The 500 waveforms covered five defect classes, as follows:

Defect class name	Number of waveforms
Surface depression	58
Sand spots	55
Five-stand-ring	128
Lamination	153
Black dots	106
Total:	500

Surface depression is simply a "dent" in the surface, and yields a fairly large, slowly varying signal.

Sand spots are medium size surface marks (dullness), comet-shaped with a tail, and are caused by sand deposited on the underlying steel surface during batch annealing. They yield signals of fairly small duration and amplitude.

Five-stand-ring is a large linear defect, parallel to the edge of the sheet, and caused by a deposit built up on the rolls of the five-stand-mill (wherein the base steel is reduced in thickness). Signals vary widely, and at one extreme resemble those from black dots, whilst at the other they resemble those from mild laminations.

Lamination is also a large linear defect, parallel to the sheet edge, and present in the base steel. It is caused by air bubbles trapped in the molten steel ingot. Lamination signals tend to be of high amplitude with many subsidiary "spikes".

Black dots are literally small black dots on the tinplate surface. Their origin is obscure. They yield signals which are short in duration, but sometimes of high amplitude.

Figures 4.5 and 4.6 show a selection of these 500 waveforms.

4.4 Signal Characterisation

As already described, two feature sets have been chosen for this investigation, as well as the chain-encoded representation scheme for the tree classifier.

4.4.1 Feature Set 1

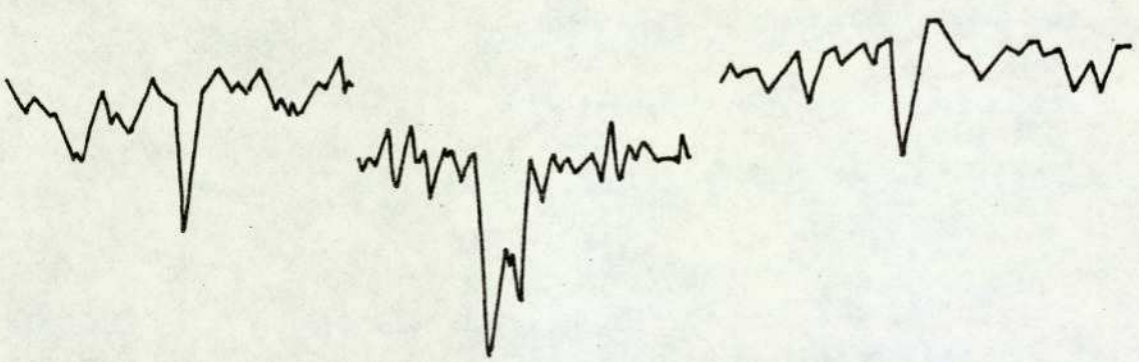
the raw digital samples across the defect waveform.

This feature set was originally introduced as the simplest, cheapest, fastest, etc., so as to provide a base-line datum against which the more costly schemes could be gauged. In fact, on the small data set, no other scheme yielded a performance even as good as this one, and this result clearly merits further attention. The features have been slightly refined for this main investigation, in that instead of taking simply the value of the nth sample as the nth feature value, the deviation of that sample from the mean of the first five samples is used instead. This provides a level normalisation and allows the feature vector to reflect more closely the waveform shape. The cost to a hardware implementation of this extension is small, since the local mean value can be derived with a simple low-pass filter.

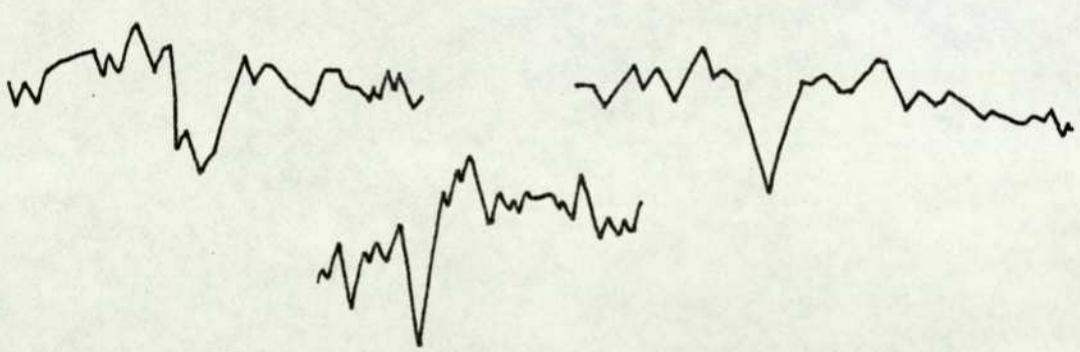
Two further features have been added in an attempt to cope with the interclass variation which arises in scans which cover different parts of the same defect. In this situation, scan sections will make up an unbroken area on the videoprint and the two features "consecutive scan number" and "first sample shift" can be defined for each scan section. Figure 4.7 illustrates these. The measure "first sample shift" is simply the along-scan offset of the first sample of a scan section with respect to the first section of the block. For isolated scan sections the two measures reduce naturally to one and zero, respectively.

The sample deviations are zero-padded (as in the exploratory study) where necessary, to make up 48 features, and the two additional features produce a vector of dimensionality 50.

LAMINATION

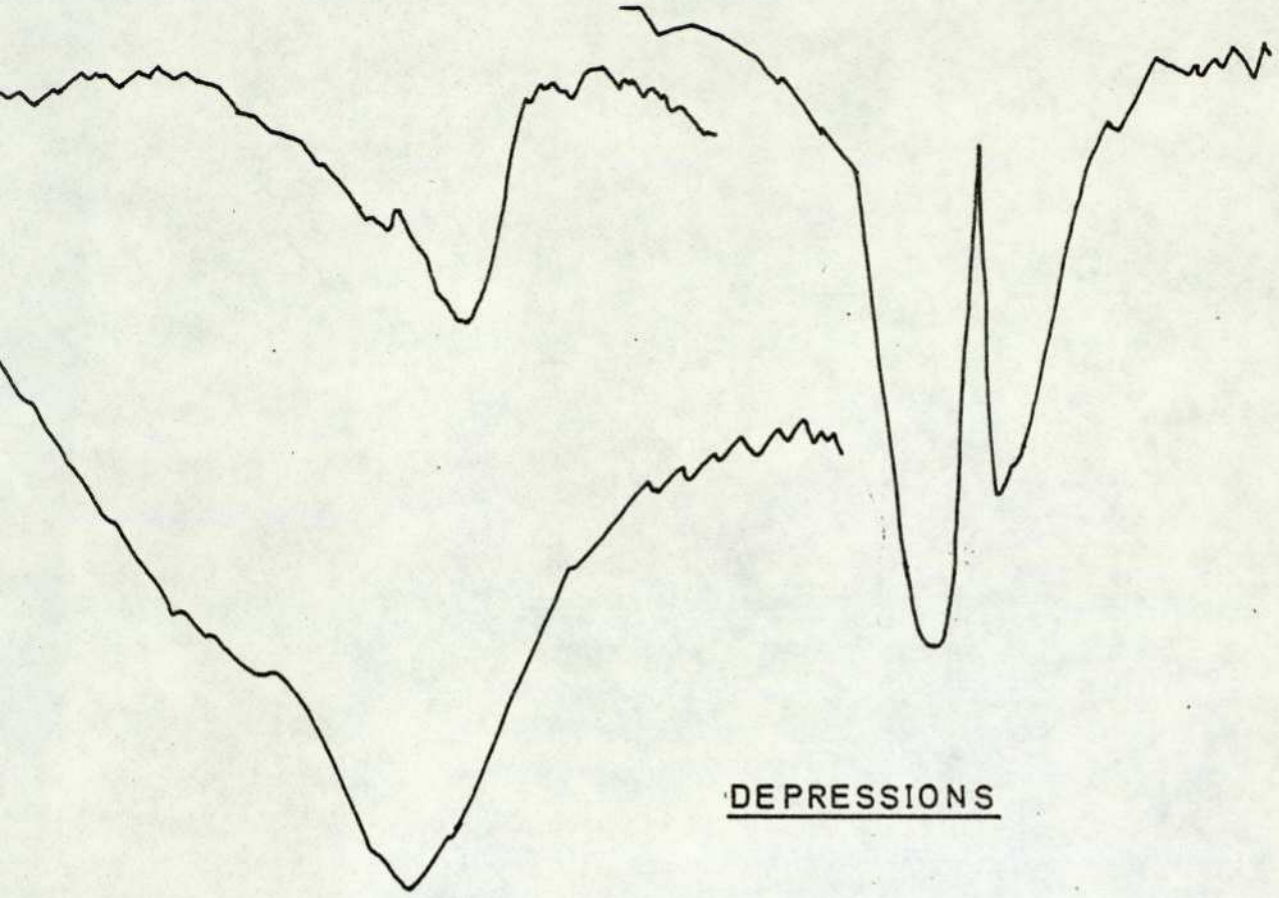


FIVE-STAND RING



BLACK DOTS

FIGURE 4.5 - WAVEFORMS FROM THE LARGER DATA SET.



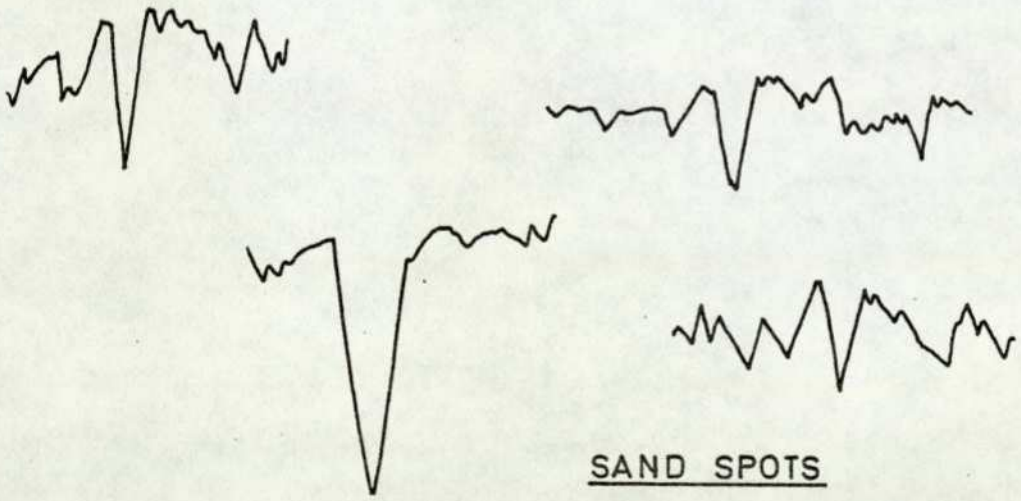


FIGURE 4.6 – MORE WAVEFORMS FROM THE LARGER DATA SET.

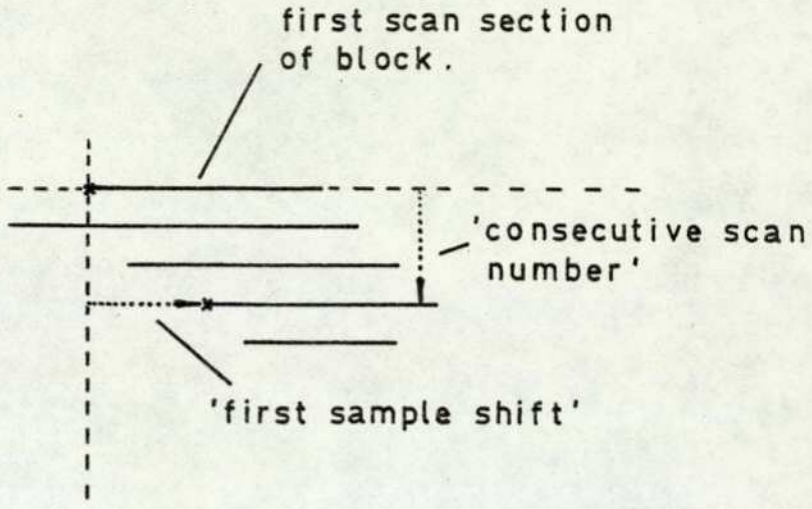


FIGURE 4.7 - SCAN SECTION FEATURES FROM THE VIDEOPRINT.

4.4.2 Feature Set 2

geometric features of the waveform shape.

As described in Chapter 3, the four geometric features of base width, amplitude, perimeter and area were used to characterise a defect pulse, and these features were chosen primarily for the ease with which they could be implemented using standard analogue hardware. Of the four, the worst in this respect is pulse perimeter, and this measure also suffers from a certain ambiguity in its definition, as described in Section 3.3.2. For these reasons, a "pseudo-perimeter" measure has been defined and used in its place. For a sampled waveform, the pseudo-perimeter is especially simple, and is merely the sum of the absolute differences between successive samples.

$$\text{i.e. pseudo-perimeter} = \sum_{k=1}^{n-1} \left| x_{k+1} - x_k \right|$$

where x_k, $k = 1, 2, -, n$ are the waveform samples. For an analogue realisation, the waveform must be differentiated, the absolute value of the differential derived (full-wave rectification), and that absolute value integrated over the waveform. Notice that this pseudo-perimeter is the limiting value of the true perimeter as the nominal value assigned to the intersample gap tends to zero, and that the two measures reflect similar properties of the waveform shape.

As well as this modification to the geometric feature set, a number of extensions have been made:

(1) all possible ratios of the four basic features have been added, as well as all possible "dimensionless" ratios such as area/(width)^2. Although this extension should be of little significance to the polynomial classifier, it is of potentially great significance to the simpler linear classifier, since it effectively allows non-linear decision surfaces to be realised in the basic, four-dimensional space (see Section 2.3.5);

(2) the location (sample number) of the minimum value sample on the waveform has been added as a feature. This is possibly of little significance, but would allow waveform shapes such as $\bigvee$ and $\bigvee$ to be distinguished, which is otherwise not possible with the geometric feature set;

(3) the two features of "consecutive scan number" and "first sample shift", as described for feature set 1, have been added.

With these extensions this feature set has dimensionality 25.

4.4.3 Chain Encoding

In the exploratory study, chain-encoding of the defect waveforms was used to produce measures for feature space analysis (slope counts) as well as being used directly with a tree-classifier operating under the assumption of Markov dependence between chain elements. The feature space analyses did not reveal any striking benefits from the features so derived, and such features are among the most costly of those considered. For this investigation, therefore, the application of the chain-encoding scheme has been limited to the tree-classifier - to which it is naturally suited.

A grid must be defined for the encoding process, and the constraint that each waveform sample should lie on a vertical grid-line has been retained. To choose the spacing between horizontal grid lines, the same approach was taken as in the preliminary study, i.e. the coded waveforms were plotted and compared to the original. Figures 4.8 and 4.9 show two such plots. In the preliminary study, a spacing of 3.10^{-2} was chosen on a similar basis. That figure relates to scan data scaled to lie between zero and one. In this main investigation, this scaling has been discarded, so that the data lies between zero and 255 (8 bits). The spacing of 3.10^{-2} therefore translates into 7.65, and the plots support this figure. A spacing of 8.0 was therefore used.

gingiiriginal
eáivefaveform.

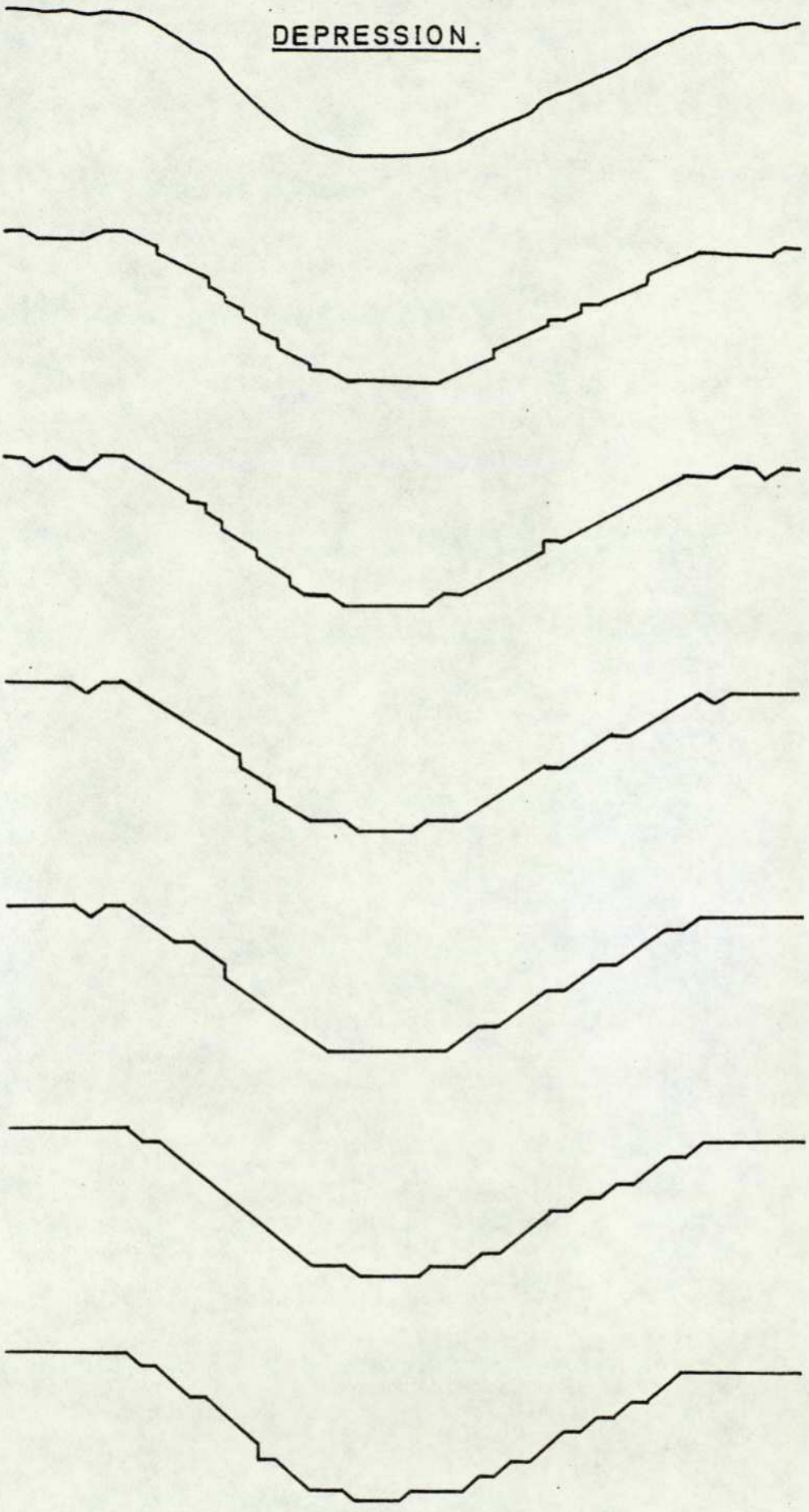


FIGURE 4.8 -CHAIN-ENCODING ON DIFFERENT GRIDS.

l
oginmiginal
rvefraveform.

rm.0

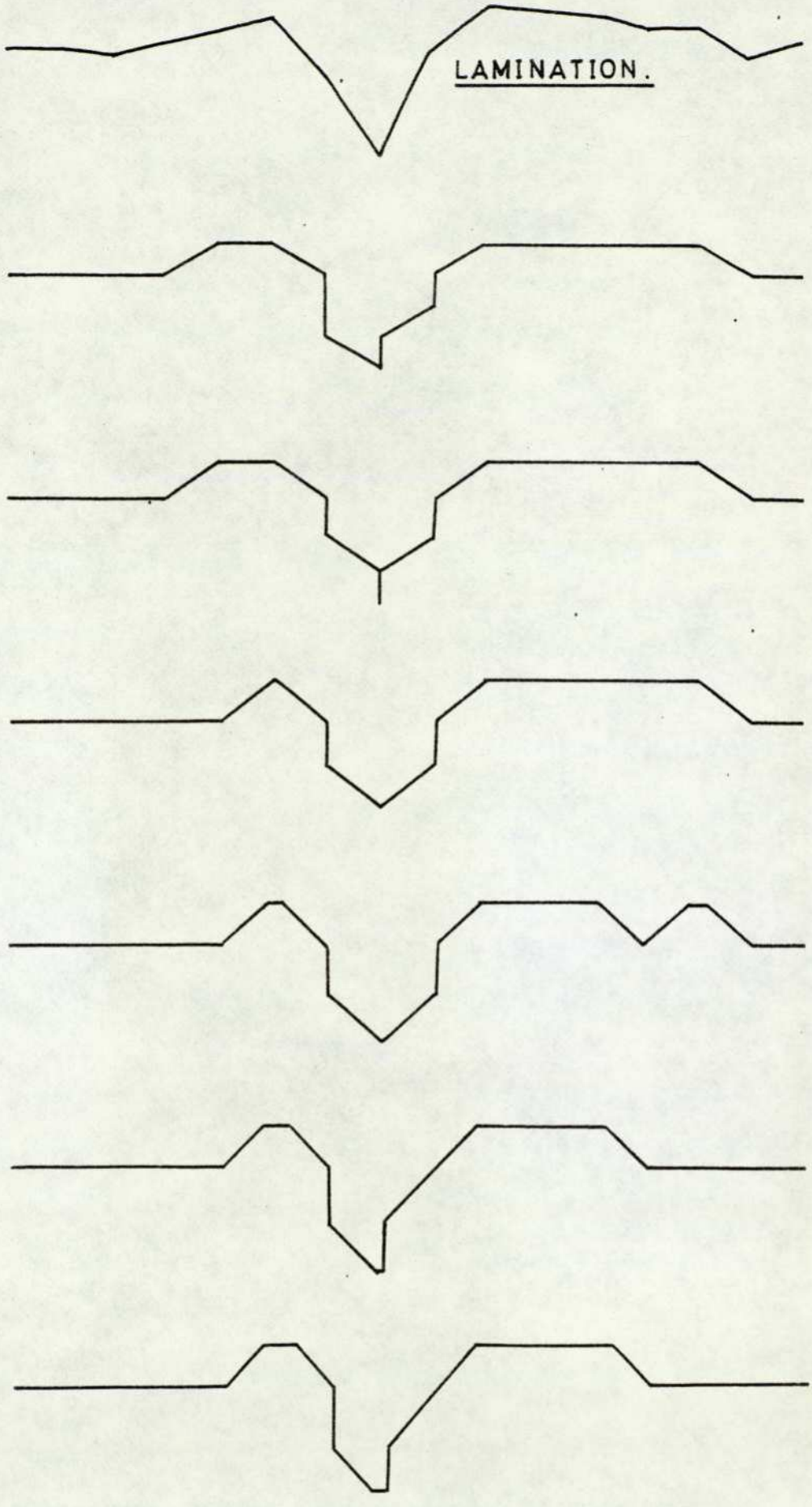


FIGURE 4.9 - CHAIN-ENCODING ON DIFFERENT GRIDS.

4.5 Feature Space Classification

With the 500 defect waveforms available, it is necessary to estimate the classification performance on unseen data, for the various classifiers and the various feature sets. To this end, the data can be divided into a design set and a test set, as discussed in Section 2.3.7. For this work, two separate design/test set partitions have been used, giving two sets of performance estimates. In general, the average of the two estimates will be presented for each particular situation. For each partition, the design set comprised 210 waveforms and the test set the remaining 290.

The results fall into two primary blocks - those with feature set 1 and those with feature set 2. Within each of these blocks a further important sub-division is made - results with indecision prohibited and results with indecision allowed. Indecision is prohibited by setting the cost of indecision, relative to the error costs, so high that it can never become the optimal choice. In this situation, the classifier is forced to assign the waveform to one of the five defect classes, no matter how inconclusive is the evidence. Throughout this work, uniform error costs have been used, so that each class is treated equally for such assignments.

In each of the sub-blocks, results will be presented for Specht's potential function classifier and for the least-mean-square linear classifier. In each case, the 210 sample design set was used in the feature selection programs (without-replacement-search), with selections based on the leave-one-out performance estimates from that design set. This identified a "best" feature subset, a classifier design for use with that subset, and an estimate of the performance to be expected on unseen data. The design was then tested on the separate test set, and the results compared. This procedure ensures that the test set is totally hidden from the design processes, and thereby results in the most reliable final estimate of classifier performance.

4.5.1 Feature Set 1 (Samples-as-Features)

4.5.1.1 Indecision Prohibited

Table 4.1 summarises the performance of Specht's potential function classifier. The figures presented are average values over the two partitions. Feature normalisation to zero mean and unit average standard deviation per class, prior to the design process, was applied. Figure 4.10 presents the same results graphically.

The following points are worthy of note:

(1) Unless the smoothing parameter is preset to a very large value, performance on the design set does not extrapolate to "unseen data", as exemplified by the test set. This is true for both the exponential and the polynomial form of this classifier.

(2) The polynomial form of the classifier does not produce a performance comparable to that of the exponential form, unless the smoothing parameter is, again, preset to a very large value.

(3) The performance figures converge to a value of approximately 80% correct classifications, if the smoothing parameter is preset to a value of 16.0.

Comparative results obtained with the least-mean-square linear classifier are presented in Table 4.2. These are superior to those of Specht's classifier in every respect.

These results, therefore, support the tentative conclusions drawn from the exploratory work of Chapter 3. They suggest an interesting general conclusion: namely, that a very flexible classifier, such as Specht's, poses severe and possibly intolerable demands on the amount of data needed to define a useful design. Such a classifier is able, by its very nature, to accommodate an arbitrarily complex distribution for each class. The problem is that such complexity may be simply a reflection of the necessarily

Smoothing parameter	Number of features selected	Percentage correct classifications			
		Design		Test	
		Exp. form	Poly. form	Exp. form	Poly. form
0.5	5	82	37	49	35
1.0	6	85	41	66	38
2.0	10	89	44	72	37
4.0	9	84	65	79	60
8.0	7	79	72	78	73
16.0	9	79	78	76	76

Table 4.1 Performance of Specht's Classifier with indecision prohibited (Feature set 1)

Number of features selected	Percentage correct classifications	
	Design	Test
8	81	83

Table 4.2 Performance of the Linear classifier with indecision prohibited (Feature set 1)

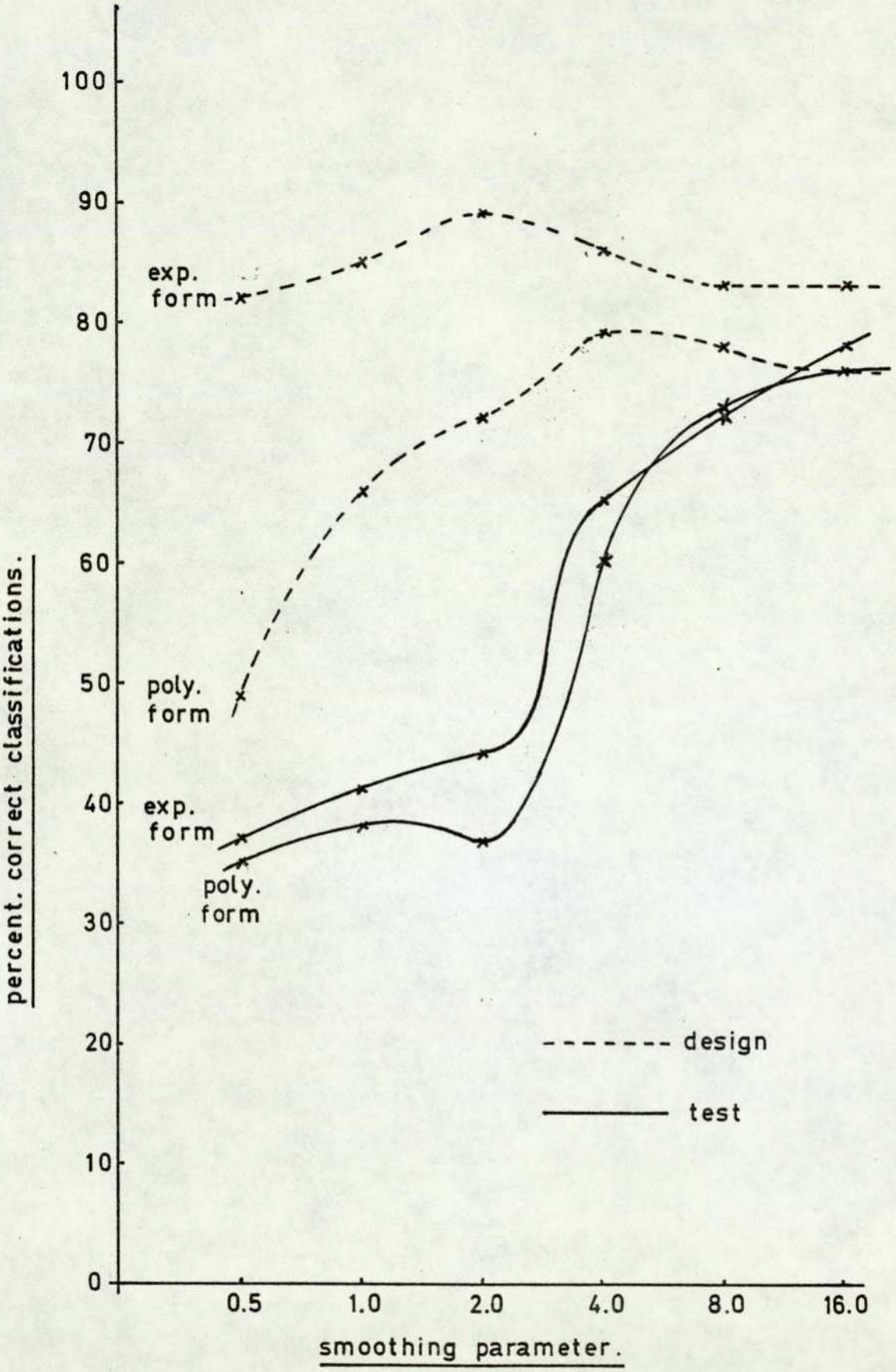


FIGURE 4.10 - FEATURE SET 1. PERFORMANCE OF SPECHT'S CLASSIFIER WITH INDECISION PROHIBITED.

limited size of the design set. If this is so, then the whole design process (including the selection of suitable features) will have no validity in general, and performance will not extrapolate to "unseen data". The effect of pre-setting the smoothing parameter to a very large value is to limit the power of the classifier to accommodate complex distributions. The linear classifier is, of course, inherently limited in this respect. In fact, the polynomials produced by expanding the exponential form of Specht's classifier, with the largest values of the smoothing parameter, could be truncated to be linear without affecting their classificatory power. The higher order terms had coefficients which were so small as to be effectively zero.

The Feature Subsets Selected: An essential part of the design process is to select a suitable feature subset from the set of candidate features presented, using a without-replacement search. Two distinct design/test set partitions have been used, and consequently two different feature subsets were chosen in each situation. These two subsets differ, in each case, only because a different partition has been used.

If this design process is to be meaningful, then the two subsets must be equally useful. Thus, either they must be closely similar, or else their selection must be non-critical. The latter proposition would imply the existence of many suitable subsets, any one of which would yield satisfactory performance. If neither proposition were valid, the implication would be that each design was peculiar to the particular partition used, and carried no validity in general.

Consistency between any two subsets can be evaluated by a simple count of the number of features which appear in both, expressed as a percentage of the mean subset size. This figure varies between zero and 33% for Specht's classifier, for different preset values of the smoothing parameter, and is 50% for the linear classifier. Close similarity

between the two subsets cannot, therefore, be claimed.

The remaining possibility is that the selection is non-critical. If this is so, a feature subset selected for any one design/test set partition should have general validity, even though it may not be unique. It should therefore yield comparable results if used with the other design set, to classify the corresponding test set. To test this, the subsets selected with the linear classifier were evaluated, in this way, on their "alien" partitions. Results were 75% and 88% correct classifications with alien partitions, as compared to 78% and 87% normally. These differences are held to be insignificant, and to support the proposition that many feature subsets exist, all of which are equally useful.

For the linear classifier, the feature numbers, in the order of selection, were as follows:

Partition 1 - [11, 20, 6, 13, 5, 10, 18, 49]
(where feature 49 is "consecutive scan number")

Partition 2 - [11, 6, 22, 13, 15, 9, 16, 10]

These subsets are shown diagrammatically in Figure 4.11, and the corresponding confusion matrices (on the test sets) in Figure 4.12. The former figure shows a relationship between the two subsets which is not obvious from their numerical description, and the confusion matrices strengthen this relationship, insofar as they reveal similar patterns of inter-class confusion. It therefore seems that the two subsets are, indeed, functionally similar, although formally distinct.

4.5.1.2 Indecision Permitted

The results presented in the preceding section were produced by setting the cost of indecision, relative to the uniform error cost, so high that each waveform was necessarily assigned to a defect class. As the indecision cost is reduced from this level, several changes can be anticipated:

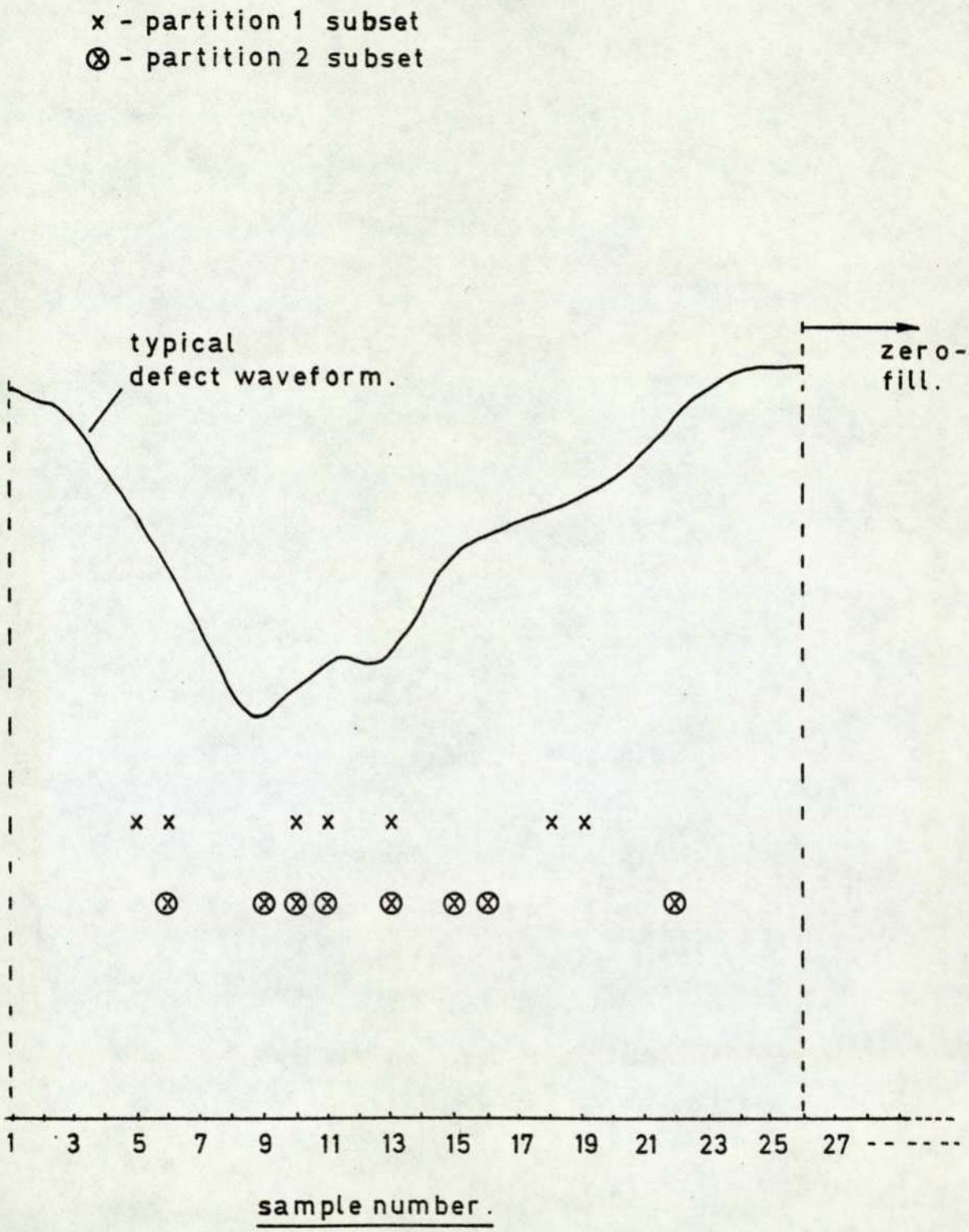


FIGURE 4.11 - FEATURE SET 1. FEATURE SUBSETS SELECTED WITH THE LINEAR CLASSIFIER.

True class \ Assigned class	1	2	3	4	5
1	103	0	0	0	0
2	0	36	0	18	2
3	0	0	27	1	0
4	0	5	0	73	0
5	2	4	0	6	13

(A) Partition 1

87% correct classifications

True class \ Assigned class	1	2	3	4	5
1	100	0	0	0	3
2	0	26	0	30	0
3	0	1	27	0	0
4	0	12	0	66	0
5	1	5	0	13	6

(B) Partition 2

78% correct classifications

Class 1 = Lamination,
Class 2 = Black dots,
Class 3 = Depressions,
Class 4 = Five-stand-ring,
Class 5 = Sand spots

Figure 4.12 Confusion matrices with the Linear Classifier on the test set (Feature set 1)

(1) some waveforms will be rejected as "undecidable" or borderline cases;

(2) the number of incorrect classifications will fall;

(3) the number of correct classifications will fall.

Essentially, the system is permitted to indicate uncertainty in the classification process, rather than merely indicating the most probable class. The important gain is that more confidence can then be placed in those classifications which still occur.

Figure 4.13 summarises the trade-offs obtained for the two classifiers. With Specht's classifier, the smoothing parameter was pre-set to 8.0, and for both classifiers, these results are estimates from the design set, averaged over the two partitions.

Two points should be noted:

(1) the error rate for Specht's classifier can be below that of the linear classifier, for a given indecision cost. This is a mixed blessing, however, because it is associated with a higher reject rate and a lower correct classification rate;

(2) with either classifier, if an error rate of virtually zero is necessary, a correct classification rate of less than 40% must be accepted also. Accordingly, more than 60% of the waveforms will be rejected as undecidable. On the other hand, an error rate of less than 10% can be achieved with 60% correct classifications, and this may be a better balance.

As already mentioned, Figure 4.13 presents the leave-one-out performance estimates from the design set. With the cost of indecision set greater than 8, the classifier is forced to classify each waveform, no matter how inconclusive is the evidence. In this case, therefore, extrapolation to the test

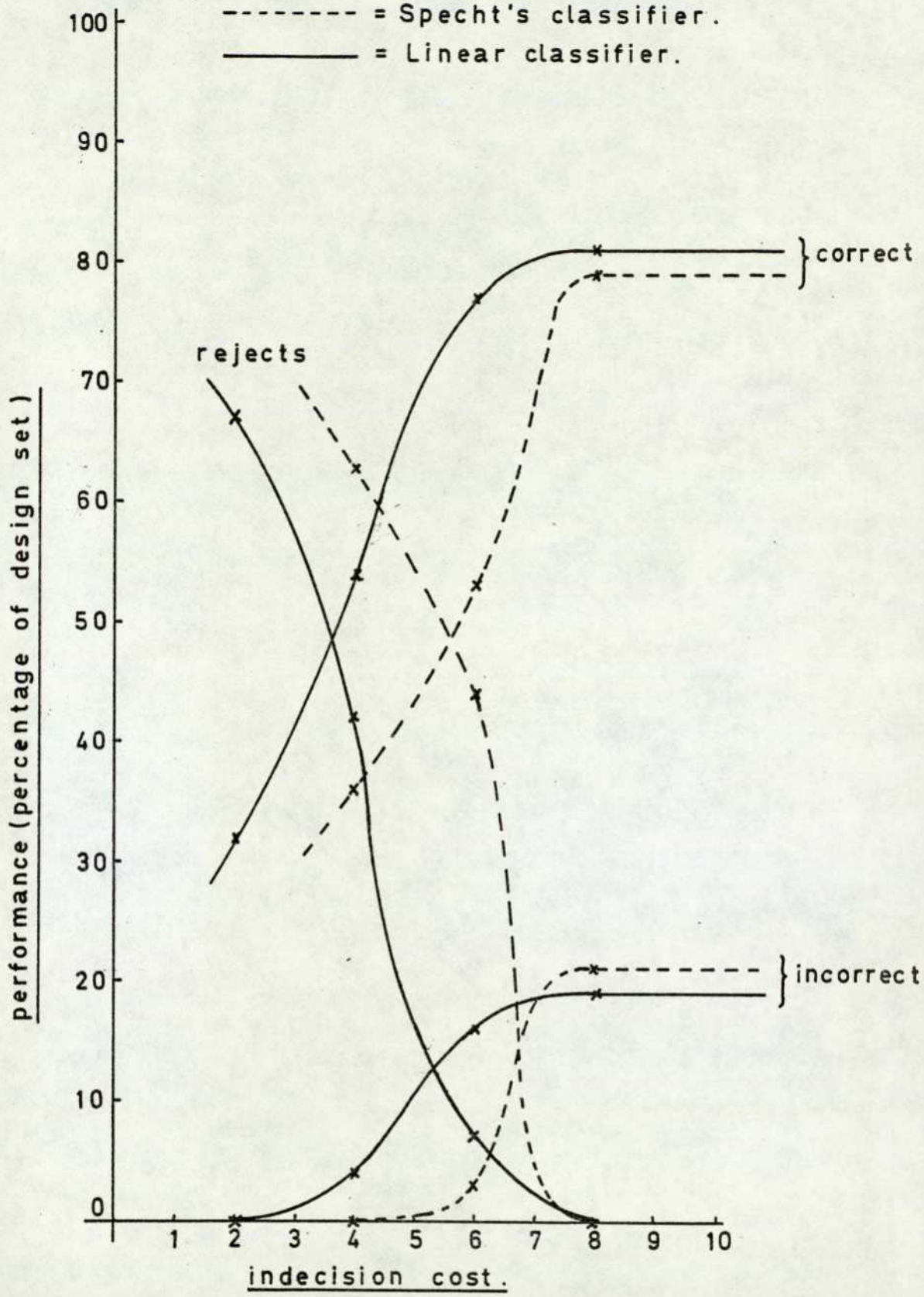


FIGURE 4.13 - FEATURE SET 1. PERFORMANCE VERSUS THE COST OF INDECISION.

set has already been evaluated in Section 4.5.1.1. A similar extrapolation can be expected for other values of the indecision cost. As a spot check, the performance of the linear classifier on the test set was determined with the cost of indecision set to 4. Results were 50% correct classifications, 4% incorrect classifications, and 46% rejects. These figures do not differ substantially from those presented in Figure 4.12.

4.5.2 Feature Set 2 (Geometric Features)

4.5.2.1 Indecision Prohibited

Table 4.3 summarises the performance of Specht's potential function classifier, in the same way as with feature set 1, and Figure 4.14 presents this performance graphically.

These results exhibit the same general characteristics as those with feature set 1 (Figure 4.10), and the comments made there will not be repeated. There are, however, a few significant differences:

(1) with the smoothing parameter preset to a value less than 8.0, the performance of the polynomial form is worse with this feature set than with feature set 1;

(2) performance extrapolation from design set to test set is, perhaps, somewhat better with this feature set, implying better waveform clustering.

Table 4.4 presents the comparative results obtained with the least-mean-square linear classifier. These are no worse than those with Specht's classifier.

The Feature Subsets Selected: The discussion presented for feature set 1 applies equally here, and the same evaluation can be applied.

Consistency between feature subsets, evaluated as the percentage of common features, varies between 20% and 33%

Smoothing parameter	Number of features selected	Percentage correct classifications			
		Design		Test	
		Exp. form	Poly. form	Exp. form	Poly. form
0.5	5	90	24	68	23
1.0	6	88	28	77	24
2.0	9	87	25	67	17
4.0	9	86	54	79	58
8.0	8	84	75	80	76
16.0	9	82	81	80	79

Table 4.3 Performance of Specht's Classifier with indecision prohibited (Feature set 2)

Number of features selected	Percentage correct classifications	
	Design	Test
10	86	80

Table 4.4 Performance of the Linear Classifier with indecision prohibited (Feature set 2)

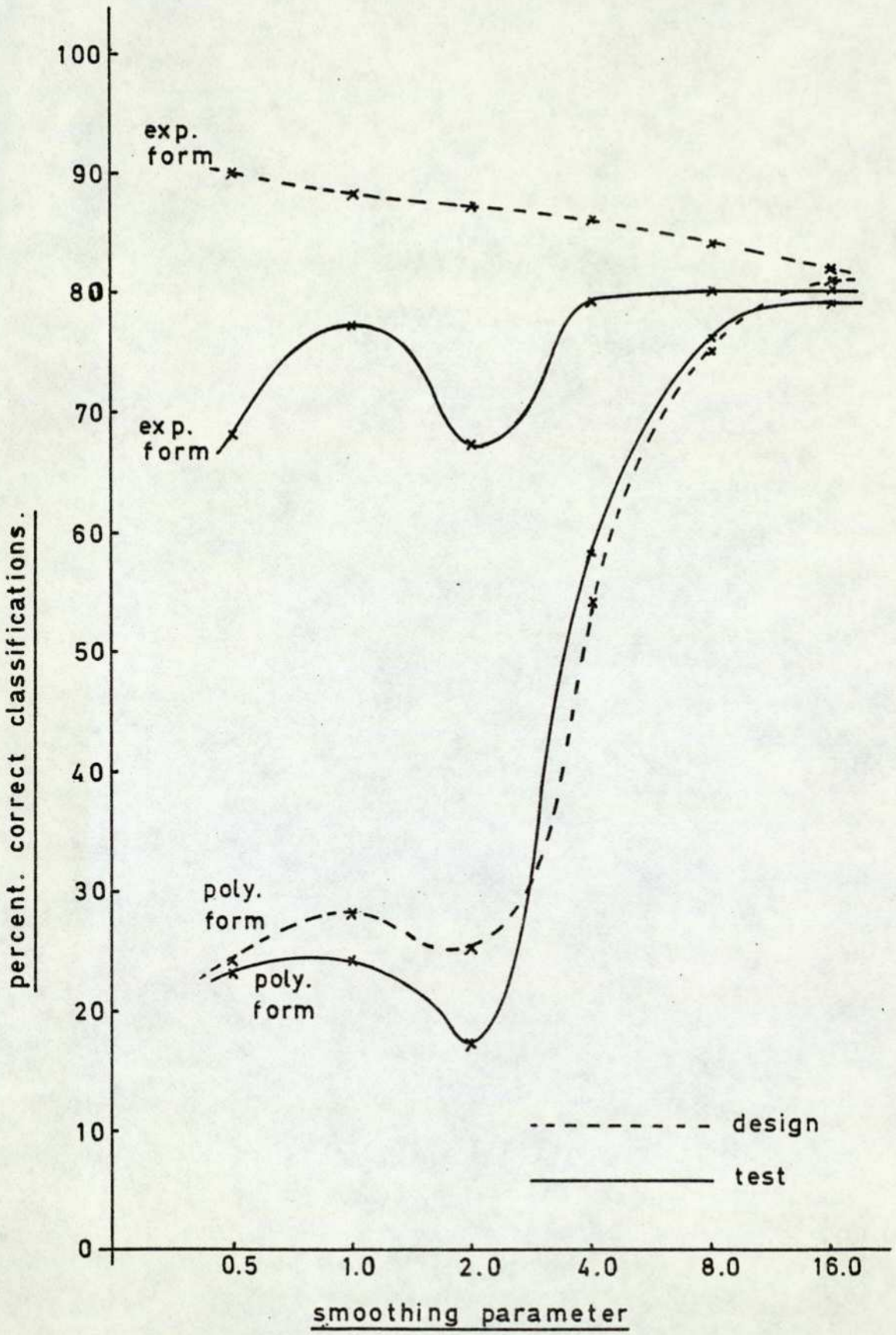


FIGURE 4.14 - FEATURE SET 2. PERFORMANCE OF SPECHT'S CLASSIFIER WITH INDECISION PROHIBITED.

for Specht's classifier, and is 20% for the linear classifier. As before, these figures do not reveal much similarity between the selected subsets.

To see whether the selection is critical, evaluation on alien partitions was used, as before, with the linear classifier. The results were 85% and 78% correct classifications, as compared to 83% and 77% normally. Again, these differences are held to be insignificant.

For the linear classifier, the subsets themselves were:

Partition 1: P/Ar, Am/Ar, Am, Am^2/Ar, BW/Ar, P^2/Ar, Min, Ar/BW, BW/Am

Partition 2: Ar/BW, Ar/BW^2, Ar/P^2, Ar/Am, Am/Ar, Am/BW, Ar/Am^2, Am/P, BW^2/Ar, Ar

where BW = Base Width
Am = Amplitude
P = Pseudo-perimeter
Ar = Area
Min = Location of minimum sample.

Notice the clear preference for <u>functions</u> of the four basic features, rather than simply the basic features themselves. This preference was less marked in the subsets selected for Specht's classifier. For example, on Partition 1 with the smoothing parameter preset to 4.0, nine features were selected for Specht's classifier, including all four basic features plus the consecutive scan number and the first-sample-shift. This accords well with the underlying theory of the two classifiers, and emphasizes the importance of including features in the candidate set which allow for any inherent limitations of the classifier which is to be evaluated.

Figure 4.15 shows the confusion matrices generated by the two feature subsets. As with feature set 1, the two subsets generate similar patterns of inter-class confusion, although, with this feature set, the performance achieved on Partition 2 (77% correct) is significantly worse than on Partition 1 (83% correct).

True class \ Assigned class	1	2	3	4	5
1	103	0	0	0	0
2	0	48	0	8	0
3	0	3	25	0	0
4	0	22	0	49	7
5	0	8	0	0	17

(A) Partition 1

83% correct classifications

True class \ Assigned class	1	2	3	4	5
1	102	0	0	1	0
2	0	23	0	32	1
3	0	0	27	1	0
4	3	10	0	64	1
5	0	5	0	14	6

(B) Partition 2

77% correct classifications

Class 1 = Lamination,
Class 2 = Black dots,
Class 3 = Depressions,
Class 4 = Five-stand-ring
Class 5 = Sand spots

Figure 4.15 Confusion matrices with the Linear Classifier on the Test set (Feature set 2)

4.5.2.2 Indecision Permitted

The discussion presented at the beginning of Section 4.5.1.2 applies equally to this feature set, and will not be repeated here.

Figure 4.16 summarises the trade-offs obtained with this feature set, under the same conditions as with feature set 1 (Figure 4.13). With both feature sets, the results exhibit the same general characteristics. There is, however, a curious difference between the two classifiers which, although evident with feature set 1, is more pronounced with this feature set. This is that the correct classification rate falls more rapidly with Specht's classifier, as the cost of indecision is reduced, than with the linear classifier. A close study of Figure 4.16 shows that this difference springs from a pre-disposition, on the part of Specht's classifier, to reject waveforms as unclassifiable. It seems likely that this is related to the preset value of the smoothing parameter (8.0 in this case).

With the linear classifier, performance with this feature set is significantly better than with feature set 1. For example, with an indecision cost of 4, 68% correct classifications result with 28% rejects. With feature set 1, the corresponding figures are 54% and 42%.

As a spot check on the extrapolation of these results to the test set, the performance of the linear classifier was determined with an indecision cost of 4, as for feature set 1. Results were 65% correct classifications, 9% incorrect classifications, and 26% rejects. These figures are substantially the same as the estimates from the design set presented in Figure 4.16.

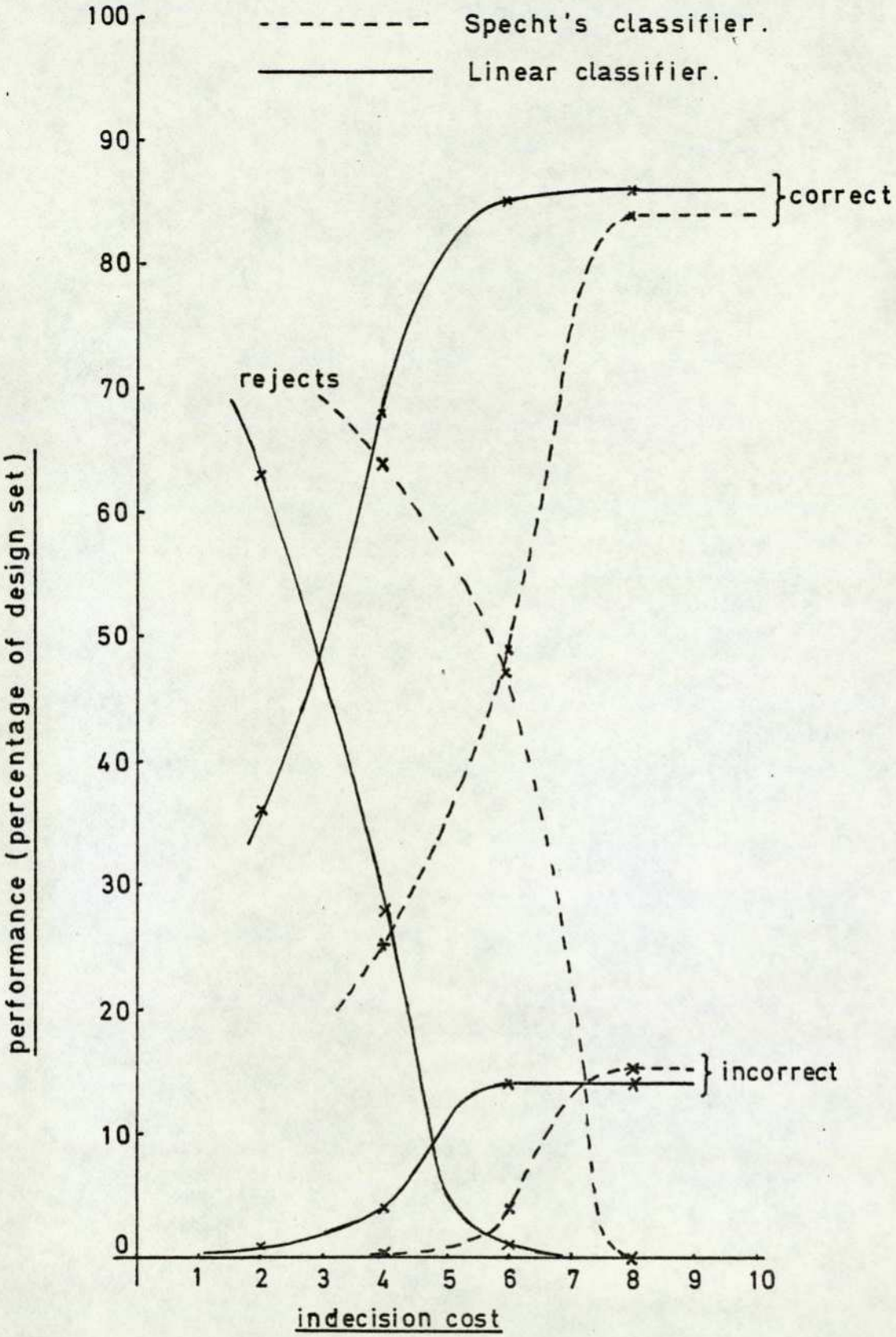


FIGURE 4.16 - FEATURE SET 2. PERFORMANCE VERSUS THE COST OF INDECISION.

4.6 Classification of Chain-Encoded Waveforms

To evaluate the tree-classifier, based on the assumption of Markov dependence between the elements of a chain-encoded defect waveform, the same two design/test set partitions as were used with the feature space classifiers, have been used. For this classifier, the design process is one of estimating the required transition probabilities for each class, as simple relative frequencies in the design set. The testing process uses the estimated probabilities to classify the test set waveforms. As before, the test set is completely hidden from the design process, so as to yield the most reliable estimate of classifier performance on "unseen" data. Where performance figures are presented for the design set, these are the "leave-one-out" estimates, and unless otherwise stated, are average values over the two partitions.

As in the exploratory work of Chapter 3, two versions of the tree classifier have been studied. Both of these versions assume homogeneity of the various transition probabilities throughout the waveform. In the first version, successive elements of a chain-encoded waveform are assumed to have been generated by a first order Markov process, and in the second version by a second order Markov process. For a fuller discussion of these principles, the reader should consult Section 3.5.

Table 4.5 summarises the results obtained under the assumption of first order dependence, and Table 4.6 the results for second order dependence. Figures 4.17 and 4.18, respectively, present the test set figures graphically.

For both versions of the classifier, the design set performance estimates are close to the corresponding performance figures on the test set, which lends credibility to both sets of figures as predictors of future performance.

There are clear differences between the performances of the two versions of the classifier, particularly for higher values of the cost of indecision. These are most clearly revealed in the graphical presentations of Figures 4.17 and 4.18. The second order classifier rejects 18% of the waveforms as un-classifiable, even for indecision costs of 8 and higher. For values greater than 8,

Cost of indecision	Percentage classifications					
	Design			Test		
	Correct	Incorrect	Reject	Correct	Incorrect	Reject
8	64	36	0	68	32	0
7	64	36	0	68	32	0
6	62	34	4	66	29	5
5	56	23	21	59	23	18
4	48	15	37	52	16	32
3	40	9	51	45	10	45
2	35	5	60	39	7	54
1	30	2	68	33	4	63

Table 4.5 Performance of the Tree-Classifier (first order dependence, indecision permitted)

Cost of indecision	Percentage classifications					
	Design			Test		
	Correct	Incorrect	Reject	Correct	Incorrect	Reject
8	57	34	9	53	29	18
7	57	34	9	53	29	18
6	57	34	9	53	28	19
5	54	29	17	50	26	24
4	51	26	23	47	22	31
3	46	22	32	43	18	39
2	42	17	41	39	13	48
1	37	13	50	34	11	55

Table 4.6 Performance of the Tree-Classifier (second order dependence, indecision permitted)

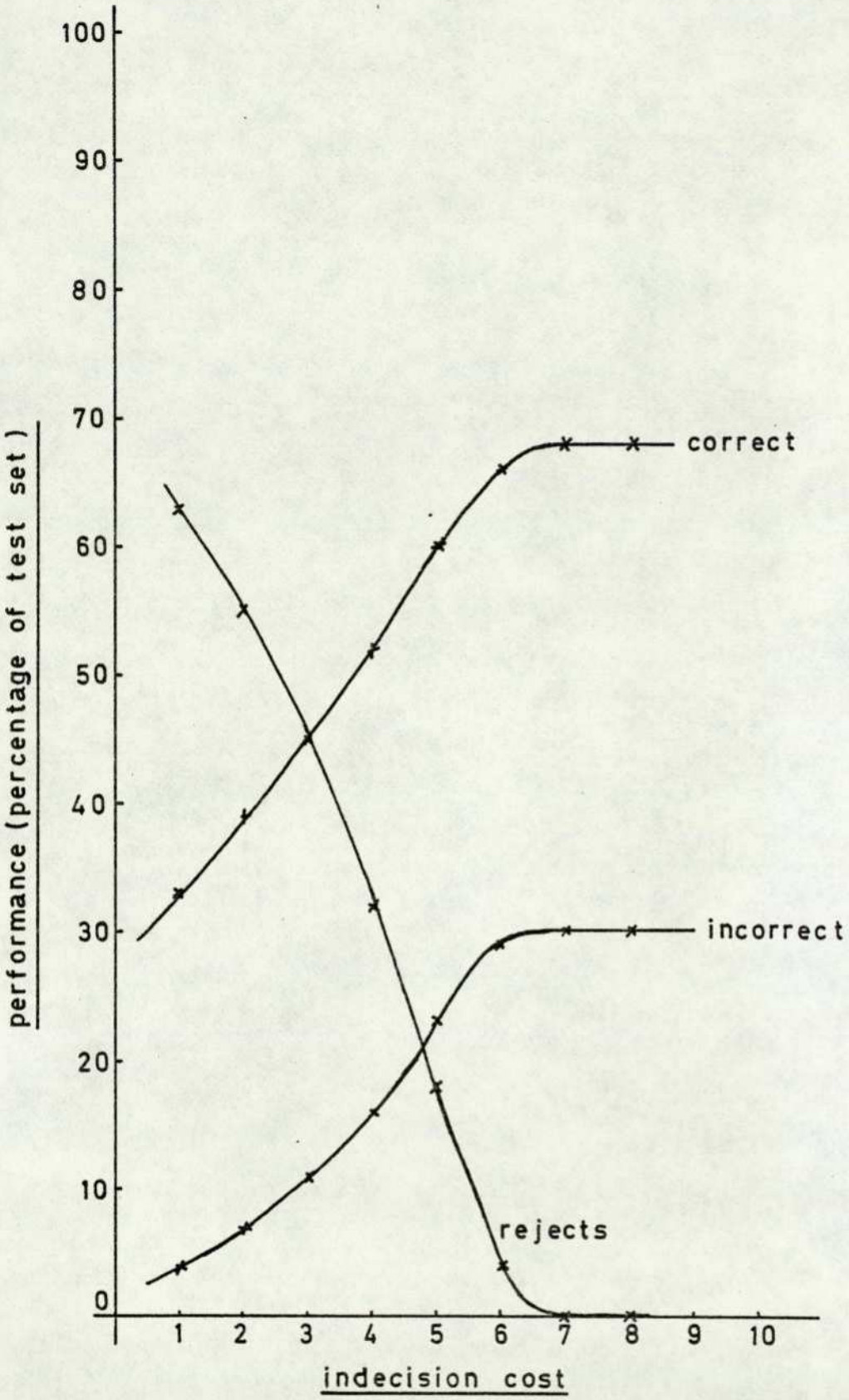


FIGURE 4.17 - FIRST-ORDER MARKOV CLASSIFIER. PERFORMANCE VERSUS THE COST OF INDECISION.

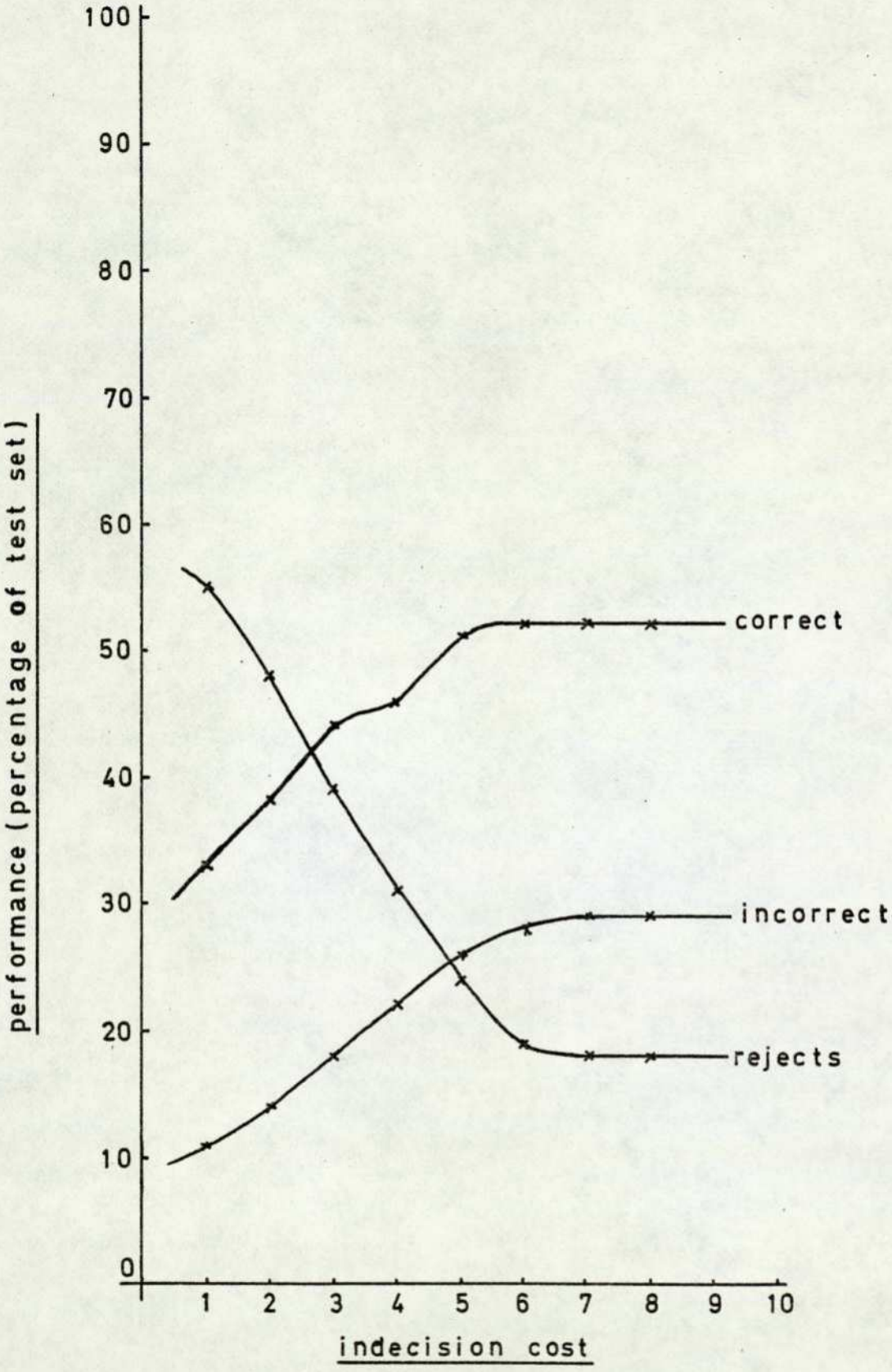


FIGURE 4.18 - SECOND-ORDER MARKOV CLASSIFIER PERFORMANCE VERSUS THE COST OF INDECISION.

a waveform can be rejected in this way only if the computed class-conditional probabilities ($P(E/W_i)$ of Section 3.5) are strictly zero for all classes. These probabilities are computed as the product of the appropriate transition probabilities for the coded waveform, and a zero value for any one therefore ensures a zero product. The transition probabilities,as already mentioned, are estimated as relative frequencies in the design set, and a zero value means simply that such a transition has not been observed. We therefore have the same problem as was encountered in the exploratory work of Chapter 3; namely, insufficient data for the estimation process. This problem does not seem to be present with the first order classifier, since all waveforms are classified with an indecision cost greater than, or equal to, 7. In this context, it is worth restating the figures presented in Section 3.5 for the number of transition probabilities which need to be estimated for each version of the classifier. These are:

first order - 30 per class, or 150 in all;
second order - 155 per class, or 775 in all.

The second order scheme is substantially more demanding in this respect, and this problem is closely related to the problems of Specht's feature space classifier, as compared to the linear classifier.

Although the first order version of the tree-classifier does not suffer from these difficulties, its best performance remains below that of the linear feature space classifier.

Figure 4.19 shows the confusion matrices on the test set, for both versions of the tree-classifier, with the cost of indecision set to 8. These give the general impression of substantial inter-class confusion, even though the first order matrix represents 67% correct classifications. In both cases, the two classes most confused are Black Dots and Five-stand-ring, which is in accordance with the feature space results.

True class	Assigned class 1	2	3	4	5	Rejects
1	90	0	0	2	11	0
2	1	30	9	14	1	1
3	0	6	22	0	0	0
4	3	26	3	39	7	0
5	3	4	1	5	12	0

(A) First-order dependence

67% correct classifications

True class	Assigned class 1	2	3	4	5	Rejects
1	53	0	0	1	6	43
2	0	30	2	15	2	7
3	2	2	16	0	5	3
4	7	19	1	35	14	2
5	4	4	1	3	9	4

(B) Second-order dependence

49% correct classifications

Class 1 = Lamination,
Class 2 = Black dots,
Class 3 = Depressions,
Class 4 = Five-stand-ring
Class 5 = Sand spots

Figure 4.19 Confusion matrices with the Tree-Classifier on the Test Set (Partition 1, Indecision cost = 8)

4.7 Summary and Conclusions

The work reported in this Chapter has been concerned with the further investigation of several lines of enquiry suggested by the exploratory work of Chapter 3. The results obtained in the exploratory work were unexpected but strictly tentative, because of the small data set on which they were based. Nonetheless, the results obtained in this chapter support, almost in their entirety, those tentative results.

Three classifiers are involved, two based on feature space techniques and a tree-classifier devised specifically for symbolic, non-quantitative data. The feature space classifiers have been evaluated with two distinct feature sets (which will be referred to as "samples" and "geometric", respectively) and the tree-classifier with symbol strings derived by chain-encoding the defect waveforms. All three classifiers were extended from the form used in the exploratory study, so as to allow indecision.

The evaluations have been based on a data set composed of 500 defect waveforms from 5 defect classes on sheet tinplate. Two distinct design set/test set partitions were used with, in each case, 210 waveforms in the design set and 290 waveforms in the test set. In general, evaluations are based on performance figures averaged over the two partitions. In all cases, care has been exercised to ensure that the test set data was completely hidden from the design processes. In particular, the temptation to modify a design after seeing results on the test set, and thereby to "try again", has been resisted. The main results will now be summarised.

Specht's classifier, with the smoothing parameter preset to a value less than about 8.0, does not yield useful results. Despite very promising performance estimates from the design set, test set performance was universally poor. With the smoothing parameter preset to a value of around 16.0, better results were achieved, although in this case, the resulting polynomials were essentially linear. This seems to discard the basic advantage of Specht's technique; namely, the ability to cope with complex, multimodal class distributions in the feature space.

With the smoothing parameter preset to 8.0, and with indecision made possible by reducing its cost, Specht's classifier shows a strong tendency simply to reject waveforms as unclassifiable. Thus, compared with the linear classifier, fewer waveforms are correctly classified, fewer incorrectly classified, and more rejected, for indecision costs less than about 7.

In all circumstances, the linear classifier performed as well as, or better than, Specht's classifier. Figure 4.20 brings together test set results for the two classifiers and the two feature sets, with indecision prohibited. When these results are coupled with the inherent simplicity of the linear classifier, in both software and hardware realisations, it seems overwhelmingly to be the first choice for this application.

Consider now the two feature sets which have been evaluated. With the linear classifier, Figure 4.20 suggests a slight preference for samples-as-features. When indecision is allowed, however, a stronger preference emerges for geometric features. For example, with the cost of indecision set to 4, and using samples-as-features, test set results are 50% correct classifications and 4% incorrect. With geometric features, equivalent results are 65% correct classifications, and 9% incorrect. On balance, considering only the performance achievable, the geometric feature set has the advantage. The question of a practical implementation, however, must also be considered. For samples-as-features, this need be no more than a simple sampling scheme, whereas for geometric features, differentiators, integrators and function generators are required. In this light, there seems to be no clear preference, overall, for either set of features.

Despite these results, it may still be difficult to accept the effectiveness of a feature set which does little more than sample the waveform at a few selected locations. To this end, two aspects of this procedure are worth discussing:

(1) Because of the "zero-fill" technique applied to short-duration waveforms, so as to make up to 48 sample values, the base-width (duration) of such waveforms

	Samples	Geometric
Specht's Classifier (polynomial form, smoothing par. = 16.0)	78% / 76%	81% / 79%
Least-mean-square Linear Classifier	81% / 83%	86% / 80%

Design / Test

Figure 4.20 Test Set results with indecision prohibited

is accessible through these zero values. Thus, if sample number K is found to be zero for a particular waveform, it is likely that the waveform has a base width of less than K samples. Furthermore, since each feature is measured as a deviation from the average value of the first five samples, a value close to zero carries a similar implication.

(2) Two or more samples, spaced not too far apart on a waveform, can reflect the rate of change of the signal in that region. Such rates of change would seem to be powerful characterisers of waveform shape.

Turning now to the tree-classifier for chain-encoded waveforms, the first order scheme has yielded better classification than the second order alternative. Even so, the results are substantially inferior to those of the linear feature space classifier. Since the encoding process would require complex, special-purpose hardware for an on-line implementation, this observation is particularly damaging. The second order scheme suffers from having insufficient data for the design process, and a parallel can be drawn with the problems encountered with Specht's feature space classifier.

It seems that complex classifiers should be viewed with some scepticism. The idea that a design based on one set of data will extrapolate to another (unseen) set of data assumes the existence of some underlying structure which has been captured by the design process. This assumption becomes less tenable as the complexity of the design increases.

5.1 IMPROVED DEFECT DETECTION AND DELINEATION

5.1 Introduction

In previous chapters the shortcomings of the processing system used by the SIRA Institute for defect detection have been described, insofar as that processing is to be followed by recognition processing. In particular, the SIRA system does not provide a good indication of the limits of each defect waveform. Since the features which have been considered depend, for their measurement, upon these limits, it has proved impossible to use the SIRA system to delineate (i.e. define the limits of) the defect waveforms. In the work described, this problem has been avoided by the simple expedient of using the SIRA system to detect and locate the defect waveforms within the data set, but then delineating those waveforms by hand, after a visual examination of each one. Figure 5.1 shows a selection of the 500 waveforms with their delineation so determined. If this delineation could be achieved without human intervention, the entire inspection process, from surface interrogation to defect recognition, would be fully automatic.

The work to be described does not aim for "optimum" delineation, primarily because such a concept is not well-defined. Consider, for example, a defect caused by surface discoloration or staining. For such defects, a well-defined boundary does not exist. Instead, a boundary region exists, within which the limits determined by delineation processing must fall. Variation within this region must be expected, and subsequent recognition processing must be tolerant of such variation. The manual delineation upon which recognition processing has been based, was naturally subject to similar variation, and the recognition results already presented suggest that sufficient tolerance can be achieved.

Instead of aiming for "optimum" delineation, the work has aimed to reproduce, as closely as possible, the results of the manual process described. Results have therefore been judged against those delineations previously determined for recognition processing, and selectively illustrated in Figure 5.1.

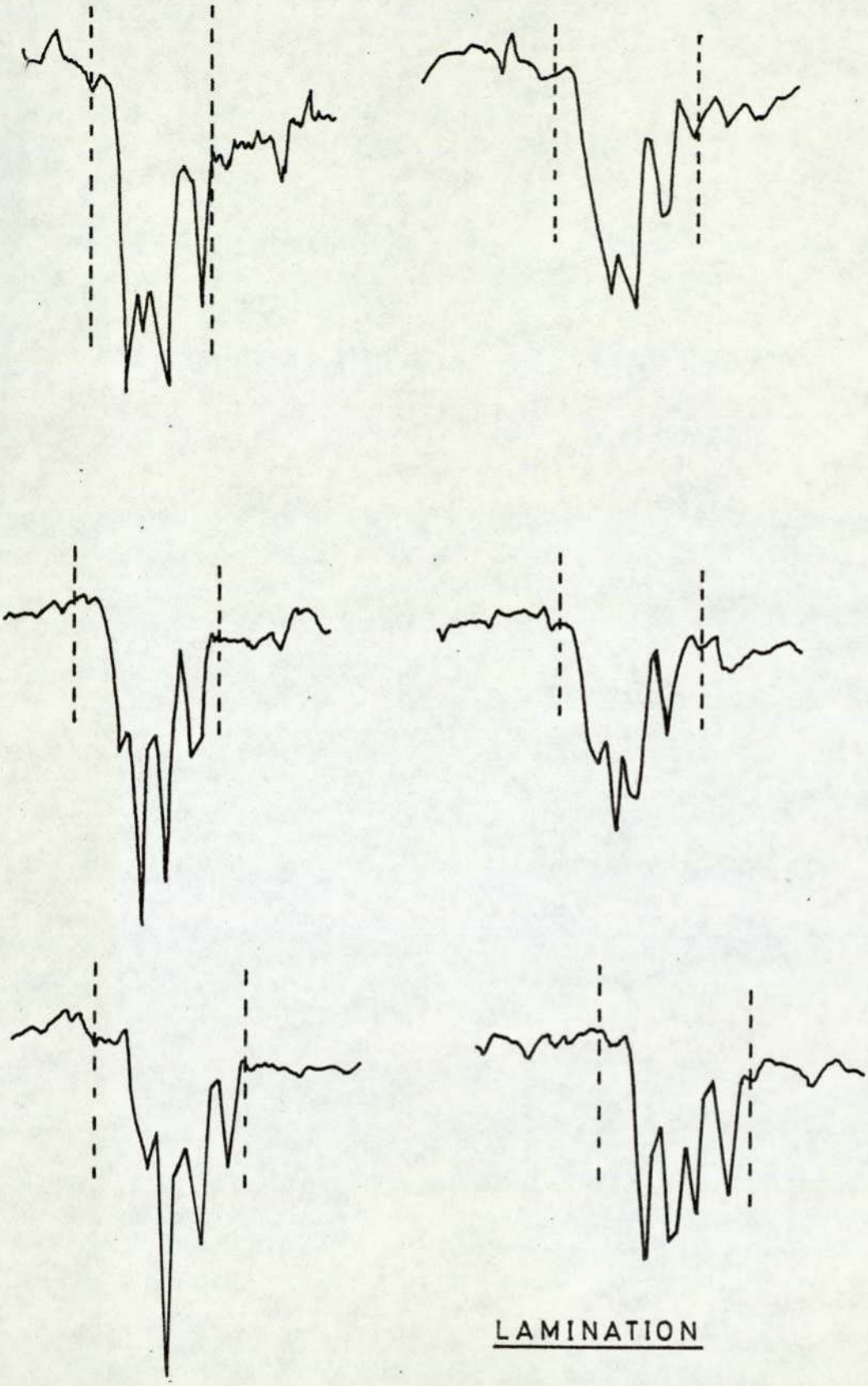
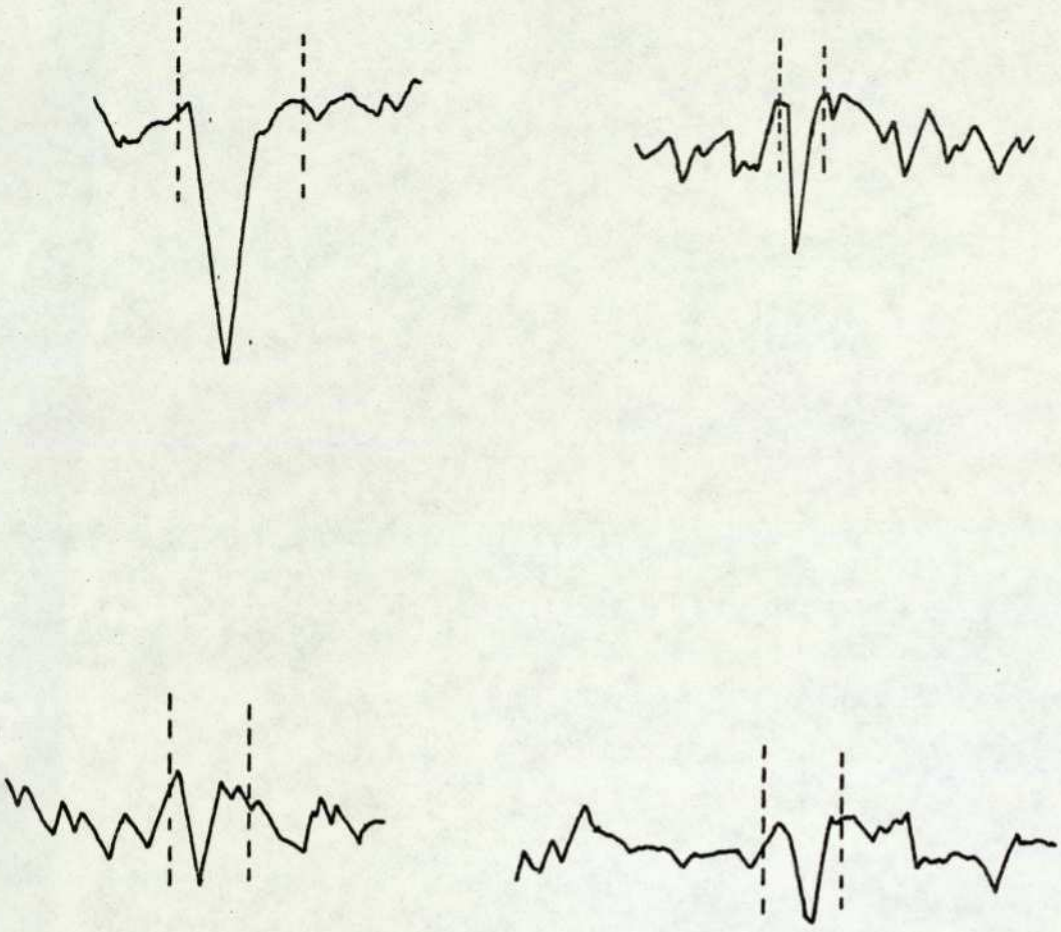


FIGURE 5.1A - EXAMPLES OF MANUAL WAVEFORM DELINEATION.



SAND SPOTS

FIGURE 5.1B - EXAMPLES OF MANUAL WAVEFORM DELINEATION.

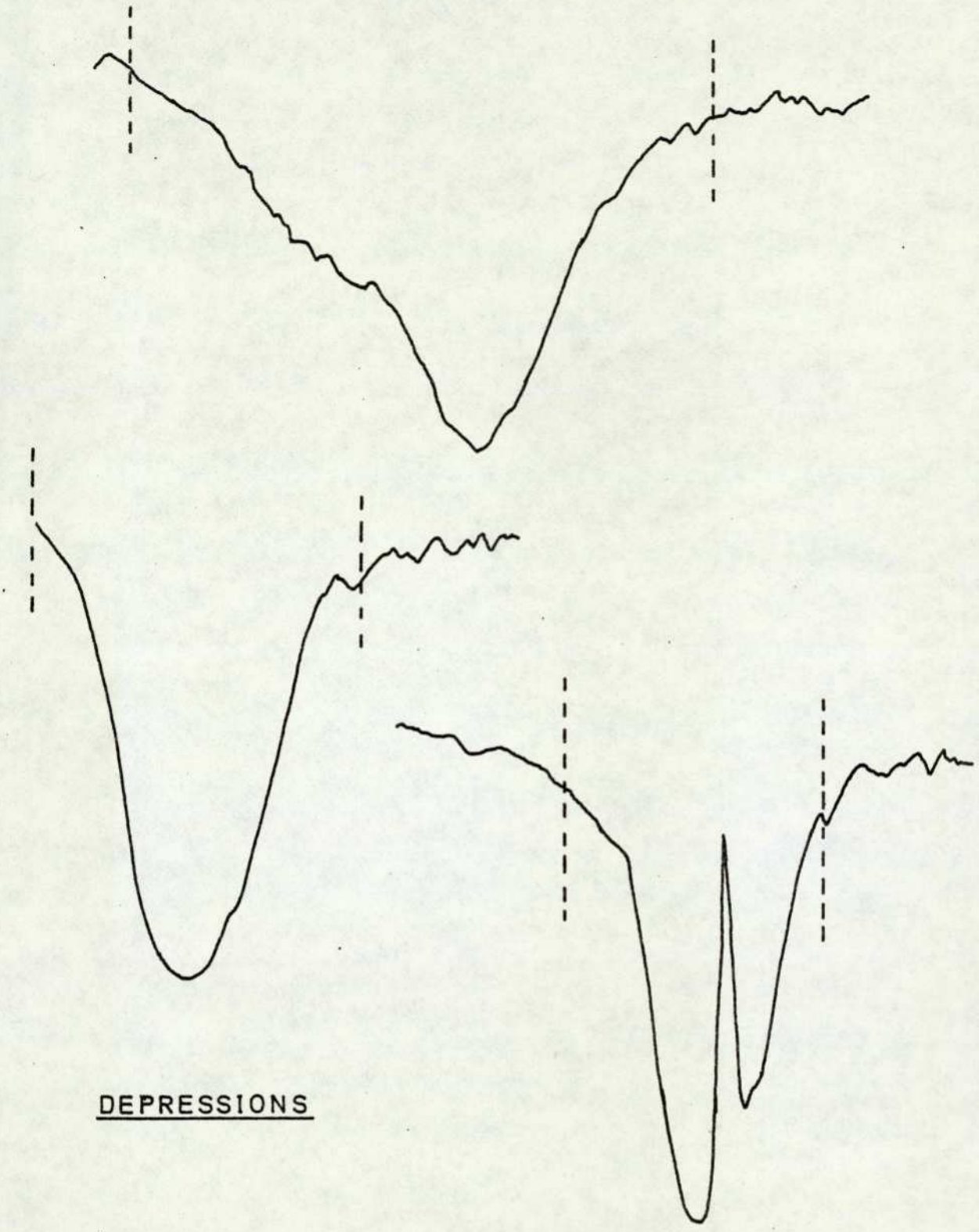
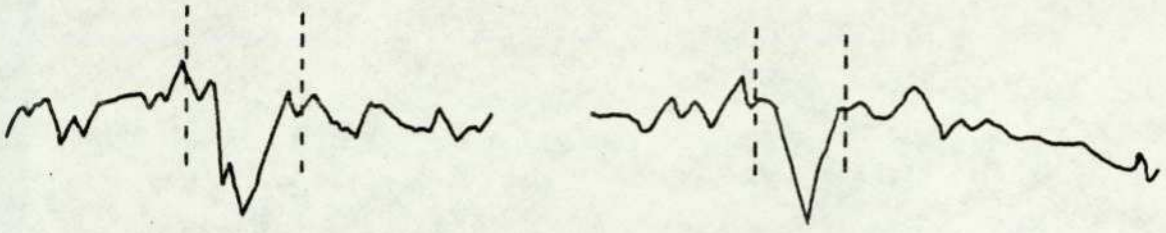
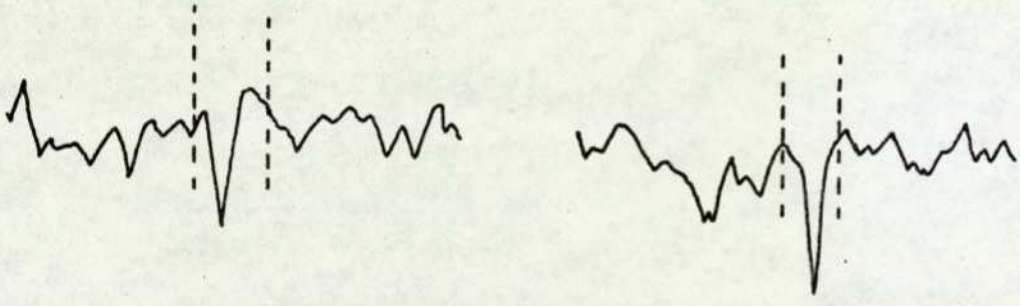


FIGURE 5.1C - EXAMPLES OF MANUAL WAVEFORM DELINEATION.



BLACK DOTS



FIVE-STAND RING

FIGURE 5.1D - EXAMPLES OF MANUAL WAVEFORM DELINEATION.

Although the primary aim of this work is to automate the delineation process, the strategies considered provide inherently for defect detection as well. Because of this, they have been evaluated for their detection performance, as well as for their delineation performance. The latter has been judged, as already described, against the results of the manual process, but an attempt has been made to provide a quantitative evaluation of their detection performance. The problems encountered in this attempt are fundamental to the detection problem, and will be discussed in Section 5.3.

5.2 Signal and Noise Characteristics

The detection and delineation problem can be treated as one of detecting and extracting a signal (the defect waveform) from noise. As such, its solution must be based upon the characteristics of the signal and the characteristics of the noise.

Of the signal, little can be said. It takes the form of a negative-going "pulse" caused by the deflection, scattering or absorption of the incident light energy. In this context, the term "pulse" is used simply to denote that the signal level falls and then rises again. This signal variation, referred to the inspected surface, may occur over a fraction of a millimetre at one extreme or over tens of centimetres at the other. Pulse amplitude is equally variable, although if it is too small it becomes wholly indistinguishable from the noise.

Concerning the noise, three substantially independent sources can be identified:

(i) Electrical - arising from photo-multiplier shot-noise and from the electronic circuitry. Photo-multiplier noise is a serious problem with the flying image scanner, but with the laser scanner it has been reduced to a negligible level. Similarly, noise from the electronic circuits is not significant.

(ii) Optical - arising from the vagaries of the optical system. In particular, the mean signal level along the scan is not constant, being higher at the scan centre than at the edges. This variation is most significant with the flying-image scanner, but is still present with the laser scanner. It rules out, for example, a simple comparison of the scan against a constant threshold, so as to detect "significant" deviations.

(iii) Surface structure - arising from innocuous variations of the inspected surface. This noise is generated by any surface which is not "mirror-perfect", and is a primary source of signal variation. It can be said that the inspection process is essentially one of distinguishing unacceptable variations of surface structure from acceptable variations, and the boundary between the two is frequently ill-defined.

For this problem, an important characteristic of the composite noise signal is its spectral content, coupled with any variation of that content in time. The power spectrum of a signal can be calculated via the Fourier Transform, and its variation in time would be to some extent revealed by scan-to-scan variations of that spectrum. However, any consistent variation during a scan would be of interest, and this can be revealed by simply dividing the scan into a few sections and calculating the spectrum independently for each section. Accordingly, scans were divided into five sections each and the power spectrum calculated for each one. Although many such scan sections are not free from defect signals, these usually occupy only a small fraction of the section and the resulting spectra can therefore be considered to be predominantly due to the noise. Typical results are shown in Figures 5.2 and 5.3.

In these results, the full-line spectrum is the raw estimate and the dotted line spectrum is the estimate after applying a Hanning smoothing filter. This filter reduces spurious effects due to the finite length data records, and the smoothed estimate can be considered the most reliable (ref. 29).

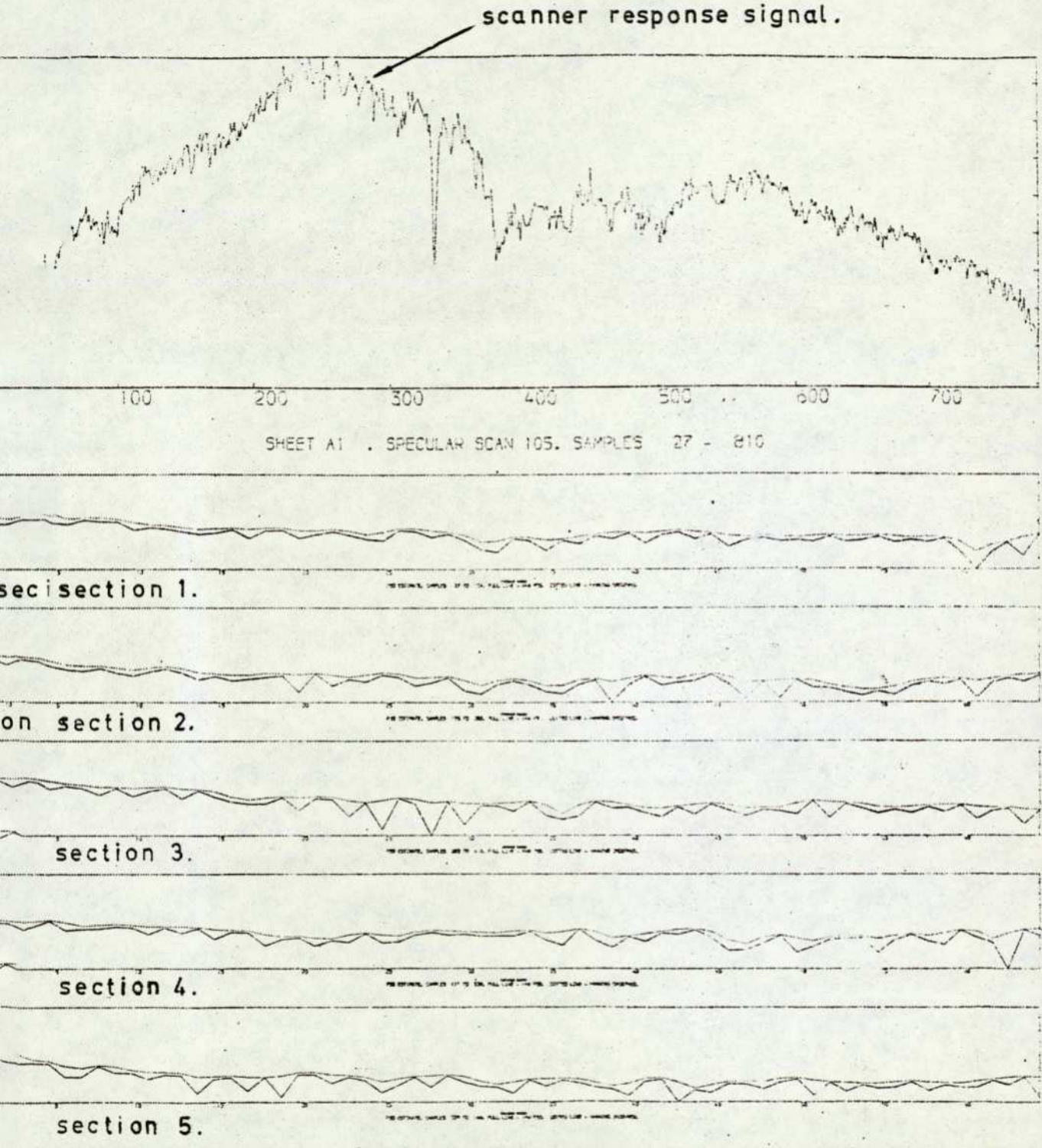


FIGURE 5.2 - POWER SPECTRUM ESTIMATES.

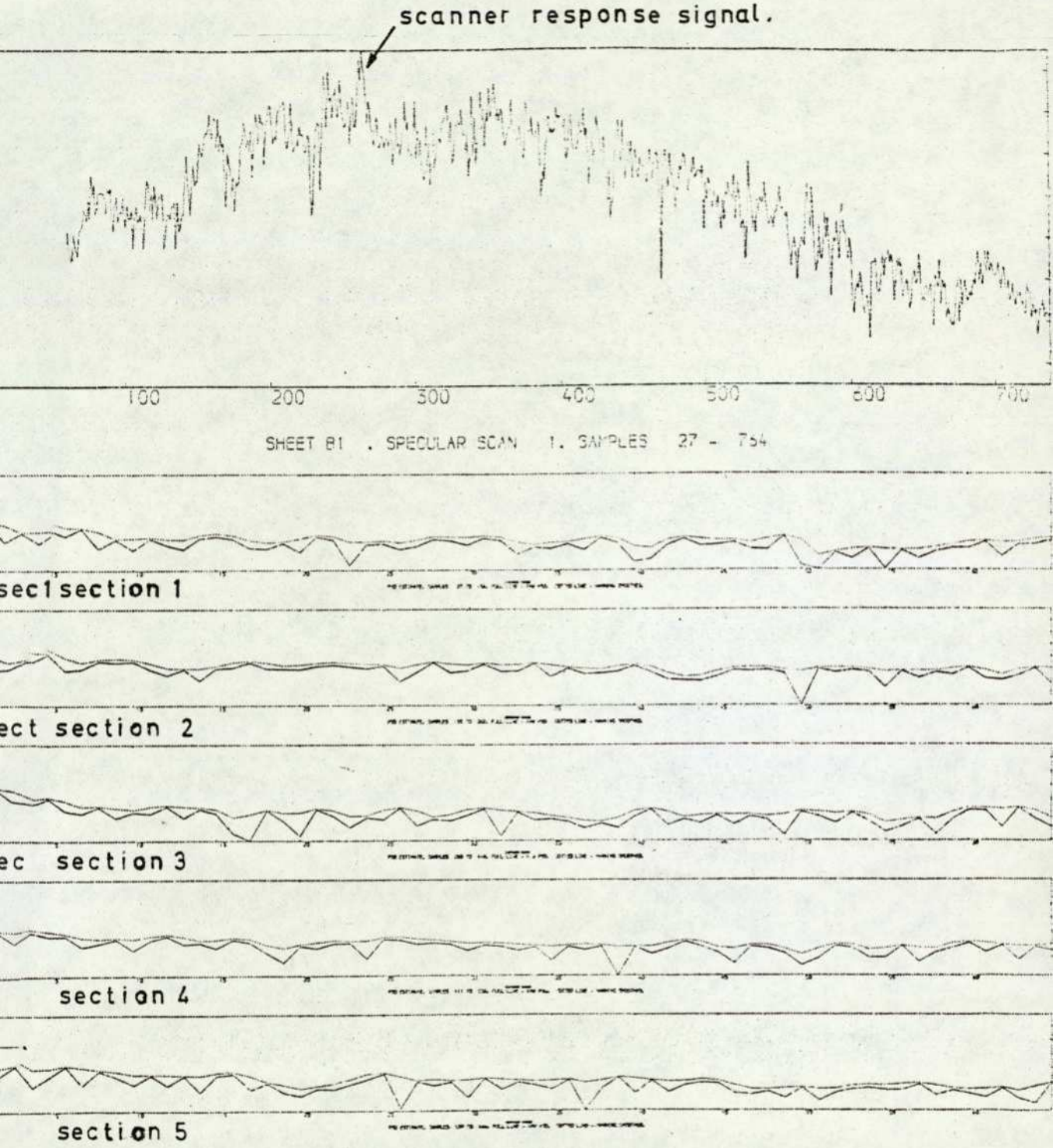


FIGURE 5.3 - POWER SPECTRUM ESTIMATES

Because the spectra are plotted to a logarithmic y-scale, some care is necessary in their interpretation. However, apart from a large and variable low-frequency content, these results suggest that the noise is substantially white and time-invariant. Equivalently, apart from the low-frequency content, successive samples of the scan are essentially independent. Considering the scanning process, this is not too surprising.

5.3 Alternative Strategies

Three strategies were considered for defect detection and delineation:

(1) a modification of the current SIRA system so as to improve its delineation performance. As described in Chapter 1, the current system compares the response signal with a reference signal, derived by low-pass filtering and then attenuating the response signal. The poor delineation performance of this system can be attributed to two factors:

(a) the reference signal tends to follow the signal due to a defect, especially on a large area, low contrast defects yielding a slowly varying signal;

(b) because of the attenuation, the crossing points between the response signal and the reference can never enclose the complete defect waveform. This is especially significant on those defect signals which only just cross the reference.

The second factor can be eliminated by comparing the response signal against the un-attenuated reference signal, as well as against the normal attenuated reference. Crossings of the un-attenuated reference will, of course, occur frequently in each scan, but may be ignored until they bracket crossings of the attenuated reference.

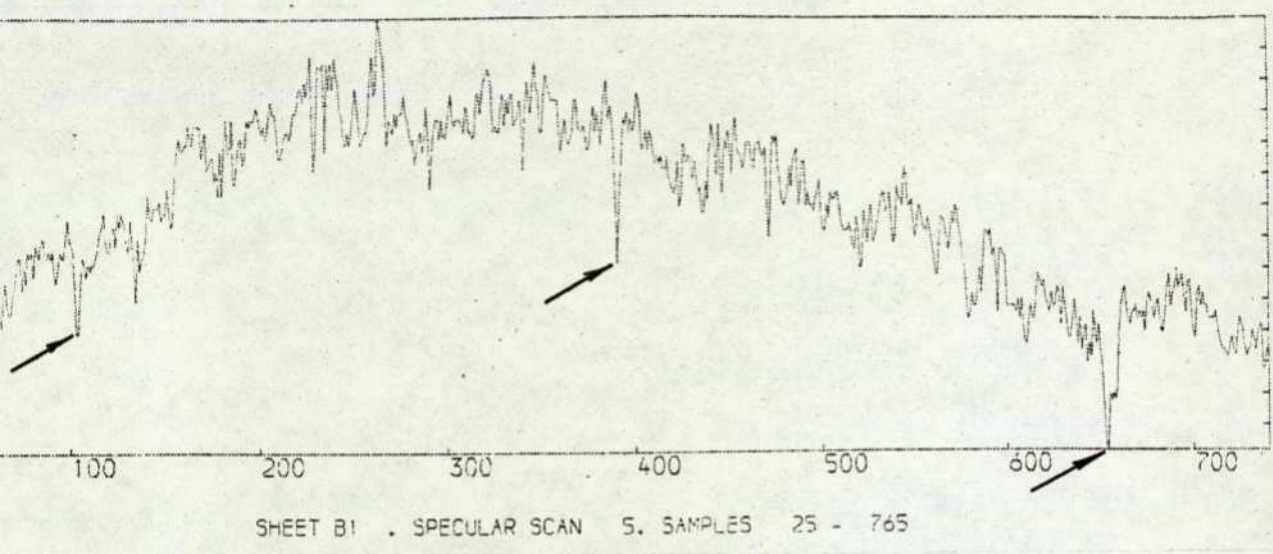
(2) The SIRA system can be described as one in which the response signal is low-pass filtered to yield a prediction of that signal a short time ahead, under the assumption that the inspected surface is defect-free. If the response signal then deviates significantly from its predicted value, a defect is indicated. The techniques of time series prediction can be applied to the sampled response signal to predict future values (ref.30). By monitoring the prediction error, the presence of a defect may be detected by an unusually large value of that error. A common approach is to predict future values as a linear weighted sum of past values of the time series (ref.30). This provides a link with the current SIRA system, since a low-pass filter in digital form is also a linear weighted sum of past values, with the weights exponentially related. A low-pass filter is therefore a particular example of this prediction technique. The potential of such systems for defect delineation is analogous to that of the SIRA system, with the extension already discussed.

(3) A quite different approach based on the techniques of matched filtering. A bank of filters may be designed to match the range of defect waveforms encountered. Each filter would yield a significant response only to waveforms similar to the waveform to which that filter was matched. By comparing the filter responses to a detection threshold, such waveforms could be detected. By further comparing the filter responses against each other, it should be possible to decide which of the various match waveforms was most closely represented in the input signal. Knowing this, and the time-relationship between the filter response and the input signal, the detected waveform could be located and delineated in time.

These three strategies were simulated digitally and evaluated on the recorded data from the laser scanner. To obtain a quantitative evaluation of their capability for defect detection, a set of 100 signals was identified in a subset of the data, as signals which should be detected by such systems. Failure to detect any one of these 100 signals was therefore recorded as a "false negative". Conversely, any signal which was detected, but which was not included in those 100, was recorded as a "false positive". This approach falls short of an objective evaluation, because of uncertainty in the a-priori selection of the 100 defect signals. In reality, there is no clear distinction between defects and non-defects. Instead, there is a continuous spectrum of surface marks (and corresponding response signals) ranging from the most severe defects through to normal, innocuous surface structure. The point in this spectrum at which a defect detection sub-system should cease to indicate a defect, is far from being well-defined. Because of this, a large proportion of the signals which were identified as having arisen from a defect were barely distinguishable from signals which were not so identified. Figure 5.4 illustrates the problem.

This problem cannot be resolved by appealing to a functional definition of what constitutes a defect, or to the judgement of trained human inspectors. Consider, for example, that some materials are produced with liberal surface lubrication. From either a functional viewpoint, or from an inspector's judgement, the resulting surface contamination does not constitute a defect. Nonetheless, this contamination is likely to produce clear response signals when scanned, such that no reasonable detection sub-system could ignore them. Instead, these signals would need to be passed on for recognition processing, recognised as having arisen from surface lubrication, and then rejected. Thus, whilst overall system performance can be judged against such criteria, the performance of internal sub-systems cannot.

Nonetheless, on the grounds that even questionable quantitative results are better than none at all - provided the limitations are clearly understood - this approach was pursued.



signals arrowed are taken to be defect signals.

FIGURE 5.4 - DEFECT SIGNALS AND OTHERS.

The digital simulations were set up as follows:

(1) The modified SIRA system was simulated using the design technique of "impulse invariance" (ref. 31).
The two parameters, filter time-constant and attenuation factor, were adjusted so as to optimise the detection performance.

(2) The time-series predictor was simulated as a first-order system, with parameters updated at each observation. With this predictor, the current observation is multiplied by a weight factor to predict the next observation:

$$\hat{Z}_{k+1} = \phi_k Z_k,$$

where $Z_1, Z_2, \ldots$ is the time-sequence of observations, and $\hat{Z}_k$ is the predicted value of Z_k.

The weight factor, ϕ_k, is determined such that if it had been used over all observations up to and including the current one, it would have yielded minimum discounted sum of squared prediction errors.

i.e. set ϕ_k to minimise

$$\overline{e^2} = \sum_{r=1}^{k} (Z_r - \hat{Z}_r)^2 \beta^{k-r},$$

where $\beta(\leq 1)$ is the discount factor.

As before, the predicted values were attenuated to provide a detection threshold. The two parameters, discount factor and attenuation factor, were adjusted so as to optimise the detection performance.

(3) The bank of matched filters was simulated as a bank of correlation detectors, as discussed in the following sections of this chapter.

For defect detection, the following results were thereby obtained:

(1) Modified SIRA system -

11 false negatives
14 false positives
24 "border line" false positives.

(2) First-order time-series predictor -

25 false negatives
77 false positives
79 "border line" false positives.

(3) Bank of matched filters -

6 false negatives
6 false positives
65 "border line"false positives.

The inclusion of "border line" false positives in these results requires an explanation. These were signals which, although they had not been included in the a-priori selection of 100 defect signals, could easily have been (i.e. they were very close to the dividing line). In this respect, they were quite distinct from those detections recorded as definite false positives, and should be so regarded.

Within their limitations, these figures indicate the superiority of the matched filter bank for defect detection. Moreover, for defect delineation, the matched filter bank was the only one of the three strategies to show any real promise. The time-series predictor failed because the sequence of first-order predictions did not resemble, in any degree, a smoothed version of the signal. It seems that a higher order predictor would be required for this, and this would greatly increase the on-line computation required to update parameters. The modified SIRA system was better, but could not adequately separate defect signals from noise with a single low-pass filter.

Consequently, the matched filter system was selected for further investigation, and the remainder of this chapter is devoted to it.

5.4 Matched Filter Theory

The matched filter occupies a prominent position in the theory of signal detection. It is the optimum linear filter for detecting a signal of known form in the presence of additive white noise (ref. 32). The theoretical development may be summarised as follows:

Let $s(t)$ be the signal to be detected, with Fourier Transform $S(\omega)$.

Let $r(t)$ be a composite signal of the form

$$r(t) = s(t) + n(t)$$
$$\text{or } r(t) = n(t),$$

where $n(t)$ is white noise with power spectral density, p_n.

The detection problem is that of distinguishing between these two possibilities for $r(t)$. This is to be done by applying $r(t)$ to a linear filter, and examining the filter response. An optimum filter is taken to be one which maximises the "signal to noise ratio" in the filter response.

Let $h(t)$ be the impulse response of the filter, with Fourier Transform $H(\omega)$.

Then the response of the filter to the signal alone is

$$g_s(t) = \int_{-\infty}^{\infty} s(\tau)h(t - \tau)d\tau$$

and the response of the filter to the noise alone, in mean square, is

$$\overline{g_n^2(t)} = \int_{-\infty}^{\infty} \frac{p_n \cdot |H(\omega)|^2}{2} \, d\omega$$

The signal to noise ratio is defined as

$$R(t) = \frac{g_s^2(t)}{\overline{g_n^2(t)}}$$

and is to be maximised at some arbitrarily chosen observation time, $t = T_1$.

This maximisation is achieved, over all $h(t)$, when

$$\underline{h(t) = \alpha\, s(T_1 - t)} \quad \ldots\ldots\ldots\ldots\ldots\ldots \quad (5.4.1)$$

where α is an undetermined gain factor.

In the frequency domain, we have

$$H(\omega) = \alpha\, S^*(\omega) e^{-j\omega T_1} \quad \ldots\ldots\ldots\ldots\ldots\ldots \quad (5.4.2)$$

where $S^*(\omega)$ is the complex conjugate of $S(\omega)$.

If $h(t)$ is to be physically realisable, we must have

$$h(t) = 0 \text{ for } t < 0.$$

From equation (5.4.1), therefore, the observation (delay) time, T_1, must be chosen so that $s(t) = 0$ for $t > T_1$.

This development is not concerned with preserving the signal waveform in the filter response. On the contrary, it is concerned with optimally distorting the waveform, so as to compress its energy into the response at time $t = T_1$. This distortion allows the signal to be detected by thresholding the filter response at time $t = T_1$, but complicates the delineation problem.

For a linear filter with impulse response $h(t)$, the convolution integral gives the response to an arbitrary input, $x(t)$, as

$$y(t) = \int_{-\infty}^{\infty} x(\tau) h(t - \tau)\, d\tau.$$

For the filter matched to a signal, $s(t)$, we have

$$h(t) = \alpha\, s(T_1 - t)$$

therefore

$$h(t - \tau) = \alpha\, s(T_1 - t + \tau)$$

therefore

$$y(t) = \int_{-\infty}^{\infty} x(\tau)\ \alpha\ s(T_1 - t + \tau)d\tau$$

therefore

$$y(T_1) = \alpha \int_{-\infty}^{\infty} x(\tau)s(\tau)d\tau$$

and if $s(t) = 0$ for $t > T_1$

and $x(t) = 0$ for $t < 0$

we have

$$y(T_1) = \alpha \int_{0}^{T_1} x(\tau)s(\tau)d\tau,$$

i.e. a correlation detector.

In other words, the matched filter may be implemented by multiplying the incoming signal, x(t), by the signal to be detected, and integrating the result.

The discussion so far is appropriate to a waveform which is completely known. In practice, we are interested in a defect signal of indeterminate amplitude which may be located anywhere within the scan. Only the signal waveform can be assumed known, say s(t), as before. The actual defect signal, therefore, has unknown parameters of amplitude and time location. Denoting this signal by d(t), we have

$$d(t) = \beta.s(t - t_o).$$

Consider the filter matched to s(t). The response of this filter to the defect signal is:

$$y_s(t) = \alpha \int_{-\infty}^{\infty} d(\tau)s(T_1 - t + \tau)d\tau$$

$$= \alpha.\beta \int_{-\infty}^{\infty} s(\tau - t_o)s(T_1 - t + \tau)d\tau$$

$$= \alpha.\beta\, R_{ss}(t - t_o - T_1),$$

where $R_{ss}(\tau)$ is the autocorrelation function of the signal s(t). $R_{ss}(\tau)$ reaches a maximum at $\tau = 0$, and we can therefore search $y_s(t)$ for such a maximum, say at $t = t_m$. We then have

$$t_m - t_o - T_1 = 0$$

$$\text{therefore} \quad t_o = t_m - T_1$$

$$\text{and} \quad \beta = \frac{y_s(t_m)}{kR_{ss}(0)}$$

This locates the defect signal in time, and since its waveform is known, this also determines its limits (delineation).

These expressions for t_o and β do not, however, allow for the defect signal, d(t), being corrupted by noise. Assuming this noise is additive, we have the filter response to a noisy defect signal as simply

$$y(t) = y_s(t) + y_n(t),$$

with $y_s(t) = \alpha.\beta.R_{ss}(t - t_o - T_1)$, as before, and $y_n(t)$ due to noise alone. The effect of $y_n(t)$ will be to introduce errors into the location and measurement of the peak of $y_s(t)$. The magnitude of these errors will, of course, depend on the relative energy of $y_s(t)$ and $y_n(t)$, or, in other words, the signal to noise ratio in the filter response.

The correspondence between this theory and the surface inspection problem is limited by two factors:

(1) the noise is not precisely white, and may not be additive

(2) the defect signals are not precisely known, even in their waveform.

Noise characteristics have been discussed in section 5.2, and an attempt to cope with the large and variable low-frequency content

described there will be discussed in section 5.5.

The second factor demands an extension to the basic matched filter scheme. Although all defect signals are roughly "pulse-like", they are known to vary in duration by an order of magnitude, at least. Any single waveform, therefore, can be expected to represent only a limited subset of these signals, and a number of different waveforms will be necessary to represent the whole range. This situation is analogous to those communication problems where any one of a number of different signal waveforms may be transmitted, and the receiver must determine which one, if any, has been received (M-ary communication, as opposed to binary). This suggests the use of a bank of filters, with each one matched to one of the possible signals. The filter responses may be compared with a detection threshold to determine if any signal is present, and if so, against each other to determine which one (ref. 33). Having done this, delineation can proceed as though only that filter was in use.

The validity of this scheme for surface inspection will depend upon the quality of approximation to actual defect signals, offered by a small number of precisely defined pulse waveforms. This can only be determined empirically, and a simulation was therefore set up for further investigation.

5.5 Simulation and Results

As discussed in section 5.4, a matched filter can be implemented as a correlation detector. To detect a signal of known waveform located anywhere within a scan, the signal must be correlated with all parts of the scan and peaks in the correlation detected. Essentially, the signal is used to "search for itself" in the incoming data.

In a digital simulation, this is a straightforward procedure.

Let s_k, $k = 1, 2, -, n_s$ be the sampled values of the signal to be detected (the match waveform).

Then the response of the filter to any n_s samples of the input signal

$$x_{\ell+1},\ x_{\ell+2},\ -\ ,\ x_{\ell+n_s}$$

is is simply

$$y_j = \sum_{k=1}^{n_s} x_{\ell+k}\, s_k \qquad \text{..... (5.5.1)}$$

A simple analysis reveals the following important properties:

(i) if the match waveform has zero mean, so that $\sum_{k=1}^{n_s} s_k = 0$, then the filter response is zero for any constant input, $x_k = c$, and is unchanged for any input by a shift in that inputs mean level;

(ii) if the match waveform is symmetrical and has zero mean, then the filter response is zero for any linear ramp input, $x_k \alpha k$, and is unchanged for any input by superimposing a ramp upon that input (see Appendix 5.1).

These two properties ensure that the filter output is essentially independent of the low-frequency content of the input signal. The signal from the laser scanner, due primarily to the scanner optics, is known to exhibit a large and variable low-frequency content, as shown by the spectral analysis of section 5.2. These properties are therefore especially significant, and symmetrical zero-mean match waveforms have been used.

Initial experiments with different shapes of match waveform suggested that the precise shape used was not particularly significant, primarily because any shape can only approximate to that of the defect signals actually encountered. A set of triangular waveforms was therefore adopted, and these are shown in Figure 5.5. With these six waveforms, a satisfactory response was found to be produced from at least one filter, for virtually the entire range of defect signals available. In this context, a response was held to be satisfactory if it was significantly above the general noise level at the filter output. Figure 5.6 shows a typical result. (In this figure, each filter output has been shifted in time so that each output value aligns with the centre of the section of the input signal which has generated it.)

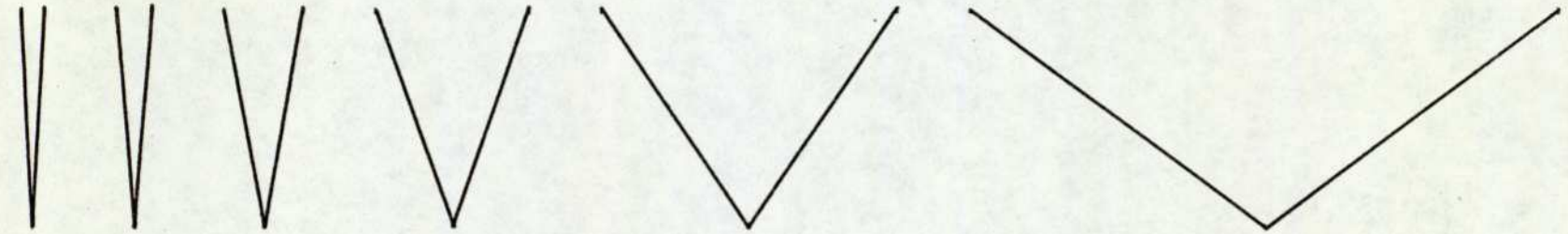


FIGURE 5.5 - THE SIX TRIANGULAR MATCH WAVEFORMS.

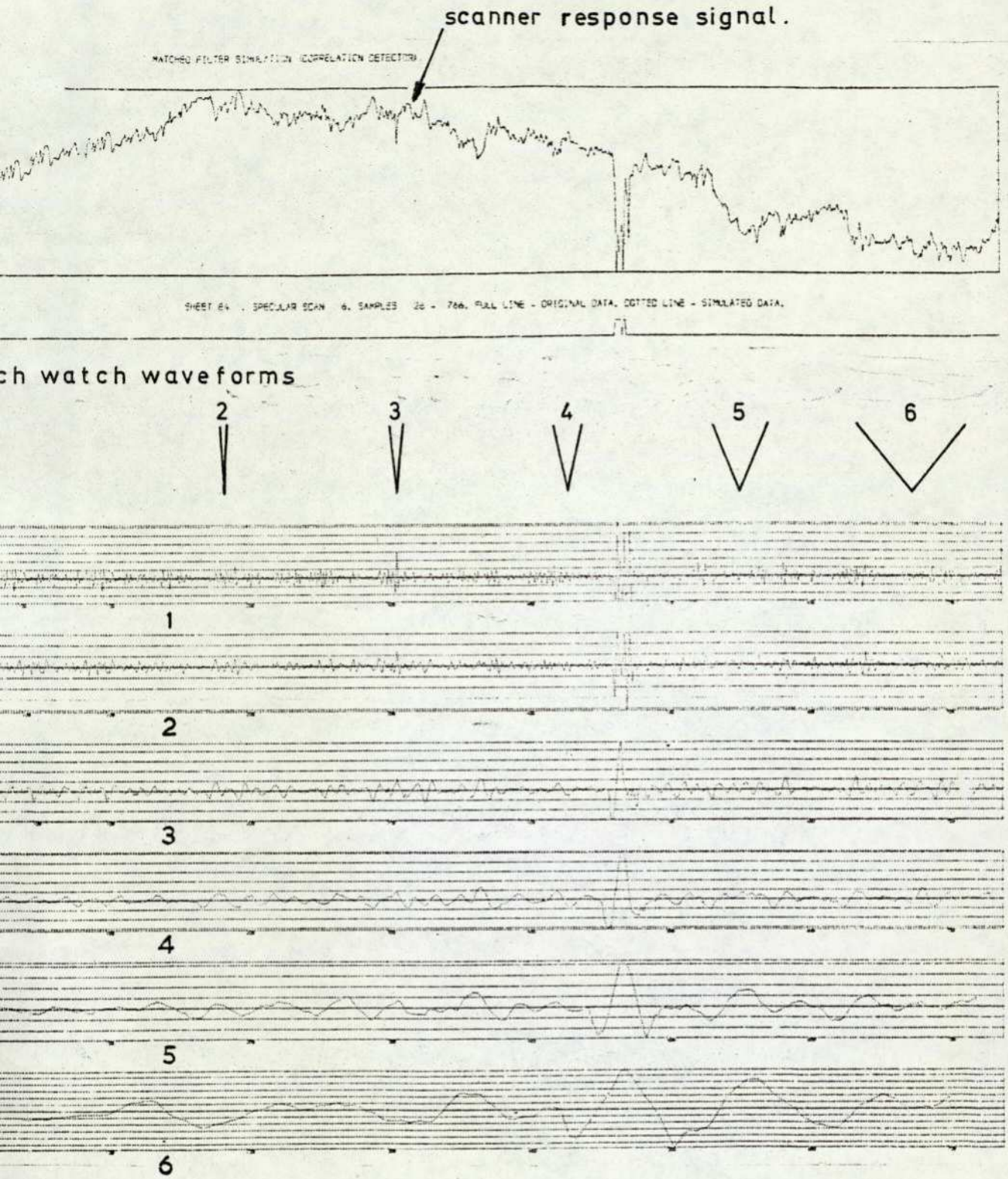


FIGURE 5.6 - MATCHED FILTER RESPONSES.

Whilst this is sufficient for defect detection, it does not test the capability of the system for defect delineation. For detection, it is enough that any filter yields a response above its detection threshold. For delineation, the most appropriate (best match) filter must be selected by comparing the filter outputs against each other. This comparison will be critically affected by the relative filter gains. With a correlation detector, gain is determined simply by the scale of the stored waveform, s(t). So far, the only conditions imposed on this waveform have been those of symmetry and zero mean value, which do not, of course, determine its scale.

It can be shown (ref. 32) that the response of a matched filter to its match waveform is proportional to the waveform energy, and this is evident for the correlation detector realisation. For the initial work, therefore, the stored match waveform, s(t), in each filter, was adjusted in scale to have unit energy, so as to ensure a filter response of unity to a perfect match. As already discussed, the results were evaluated in terms of a response significantly above the general noise level at the filter output. Unfortunately, however, the noise level from each filter was different, and a different detection threshold was therefore necessary for each one. In effect, the filter gains were not compatible in this respect, despite the common normalisation for unit energy.

This lack of compatibility became still more obvious when the filter responses were compared to find the best match for delineation. It was immediately apparent that a simple comparison to find the largest output would not be satisfactory. After some experimentation, it was found that good delineation could be achieved if each filter output was first divided by its corresponding detection threshold, before being compared to the others to find the largest. This implied that the empirically determined thresholds could be used to scale the gain of each filter, so as to allow a single common threshold thereafter, together with simple comparisons to find the best match. Such simplicity would be very valuable in an on-line system, but this empirical procedure is an unsatisfactory way of achieving it. In particular, if any of the

filters were changed, or if further filters were added, the scale factors would need to be determined again empirically. A standard normalisation procedure, applicable to any filter, would be more satisfactory.

Eventually, such a normalisation procedure was found. This was to adjust the stored match waveform for each filter so as to yield a response of unit amplitude to a waveform which was perfectly matched, but itself of unit amplitude. The normalisation parameters for this procedure are derived in Appendix 5.2.

With such a normalisation, the amplitude of each filter output is equal to the amplitude of its input, providing that the input waveform is a (scaled) perfect match. For such inputs, a common threshold at the filter outputs can be interpreted as a common threshold on the input signal amplitude. With the triangular match waveforms, Fig. 5.7 shows the set of signals which would be treated identically for detection purposes, under this normalisation.

The normalisation was implemented and the filter outputs compared with those obtained with the original (unit energy) normalisation. The ratio between the two outputs was used to scale the original detection thresholds, so as to give equivalent thresholds with the new normalisation. The results are shown in Table 5.1.

Considering the empirical basis of the original thresholds, the equivalent values are remarkably uniform. They suggest that a common setting of between 15 and 20, coupled with simple selection of the largest filter output to determine the best match, will yield equivalent results.

With this scheme, and a common threshold of 20, delineation signals were generated. The procedure at each sample point of the scan was as follows:

(1) Compute the output value for each matched filter, and select the largest(best match).

(2) If that value exceeds the common detection threshold, and is a peak value, set the delineation signal

FILTER NUMBER	WIDTH OF MATCH WAVEFORM (NUMBER OF SAMPLES)	THRESHOLD WITH ORIGINAL NORMALISATION	SCALED THRESHOLD WITH NEW NORMALISATION
1	3	-	-
2	5	12	15
3	11	15	13.9
4	21	20	14.7
5	41	30	16.4
6	81	50	19.6

TABLE 5.1 - THRESHOLD SCALING FOR THE NEW NORMALISATION

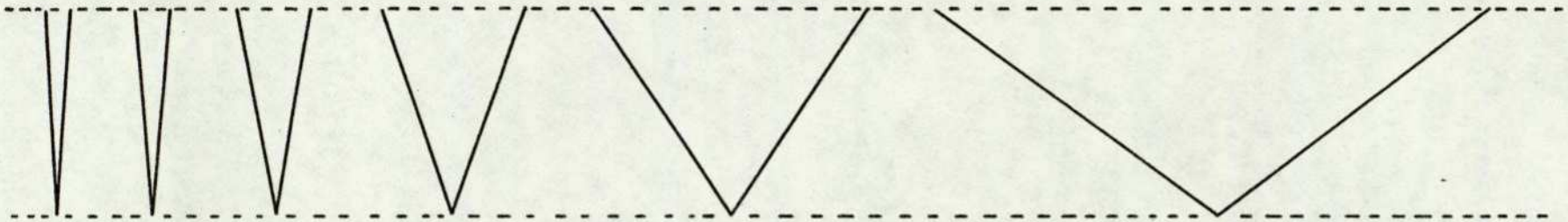


FIGURE 5.7 — DEFECT SIGNALS AT THE DETECTION THRESHOLD WITH THE TEXT NORMALISATION.

according to the match waveform of that filter - where a "peak value" is one which is greater than, or equal to, both the preceding and the succeeding value.

As already described, the filter outputs in the graphical results are so aligned with the input data that each value lies at the centre of the input signal which has generated it. Step (2) will then set the delineation signal on K samples before and after the point in question,where K is the half-width of the match waveform of the best-match filter. Figs. 5.8 and 5.9 show results for complete scans, and Fig. 5.10 shows the results on the same scan sections as Fig. 5.1.

Fig. 5.10 therefore represents the automatic process and Fig. 5.1 the manual process. The correspondence between the two is satisfactory. Where they differ, it can be argued that the automatic delineation is more "correct" than the manual. The single exception is the lamination signal upon which the delineation, determined automatically, encloses just a single spike of the overall waveform. This has occurred as a result of that spike generating a large response from the filter matched to the smallest triangle. This response was large enough to override the correct filter response, so taking precedence over it. For this to happen, such a spike must be larger in amplitude than the pulse upon which it is superimposed, and located close to the centre of that pulse. In the example shown both of these conditions are met, but this is a rare occurrence.

5.6 Summary and Conclusions

Prior to the work described in this chapter, systems for defect recognition had been devised and evaluated, which required for their input defect waveforms adequately delineated in the scanner response. These systems were fully automatic, as required for on-line application, and had been shown to yield good results on the data available.

No automatic method was available, however, to effect the

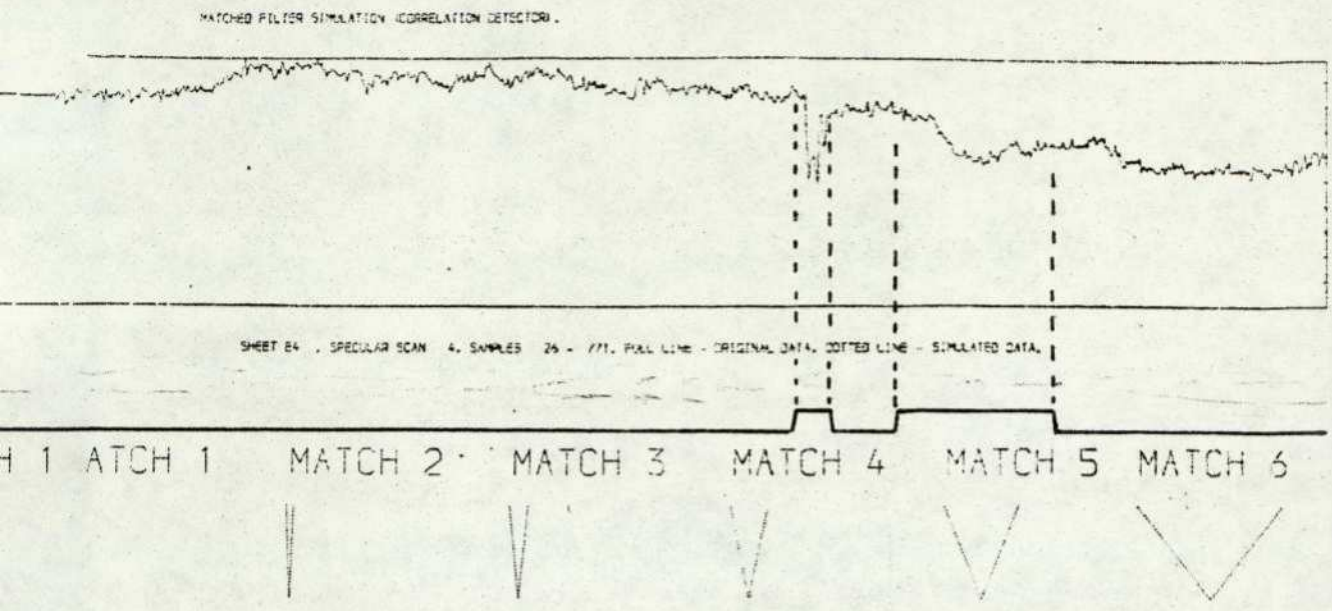


FIGURE 5.8 - DETECTION AND DELINEATION WITH THE MATCHED FILTER BANK.

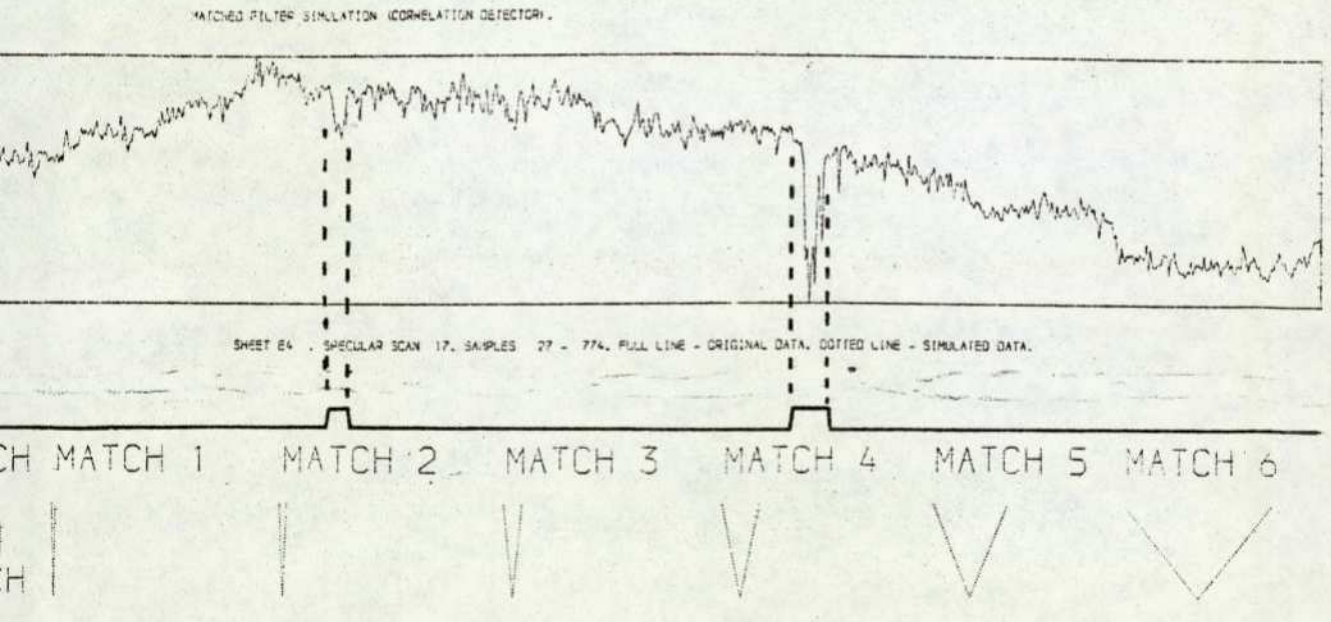


FIGURE 5.9 - DETECTION AND DELINEATION WITH THE MATCHED FILTER BANK.

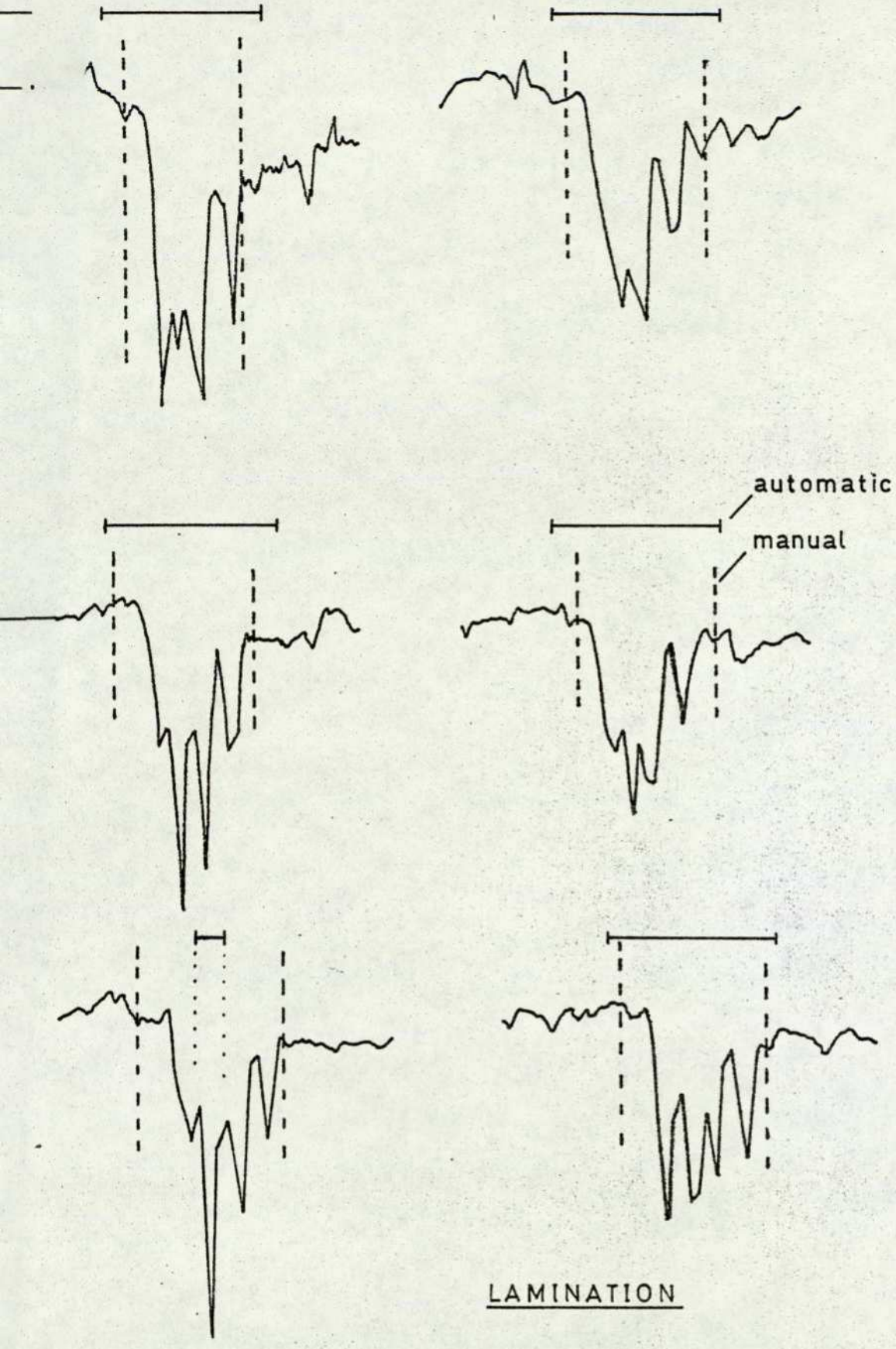
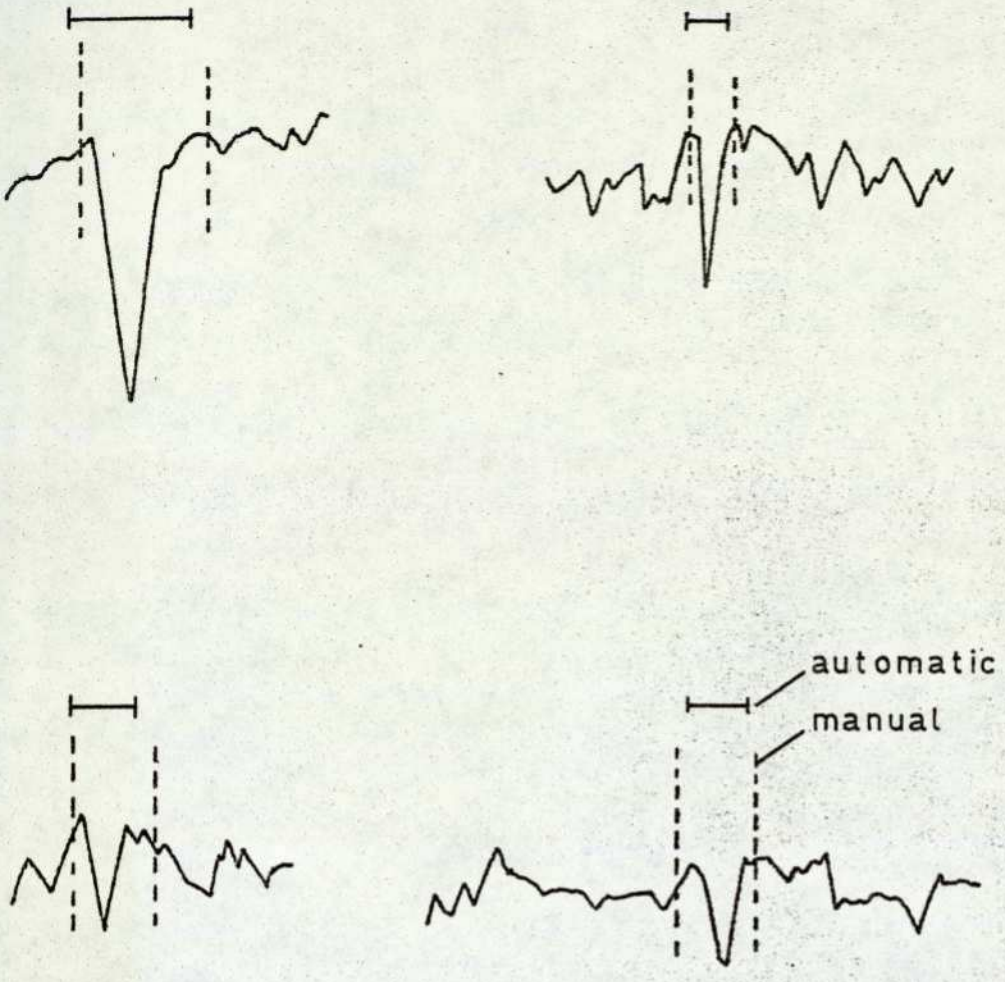


FIGURE 5.10A - AUTOMATIC VS MANUAL WAVEFORM DELINEATION.



SAND SPOTS

FIGURE 5.10 B - AUTOMATIC VS MANUAL WAVEFORM DELINEATION

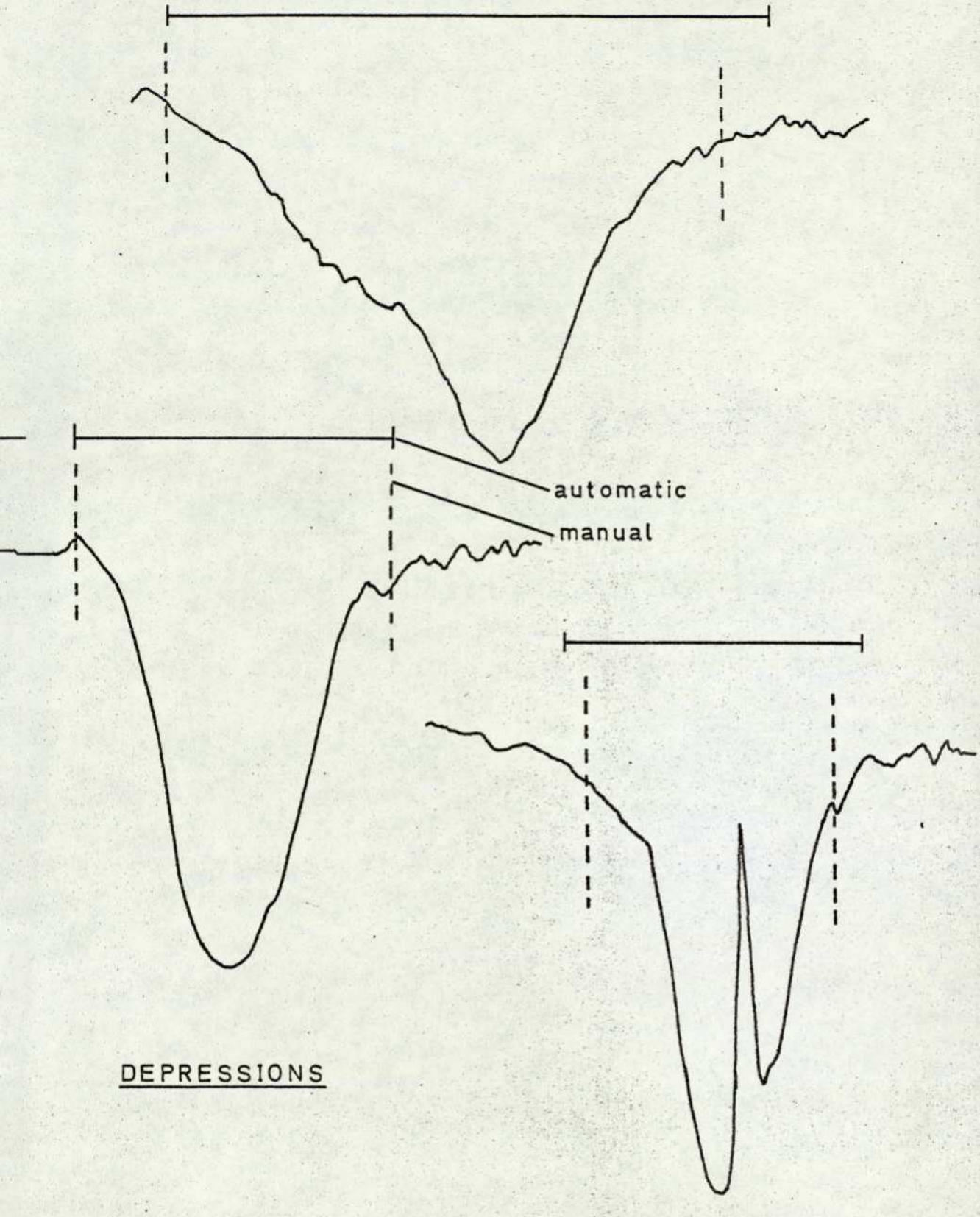
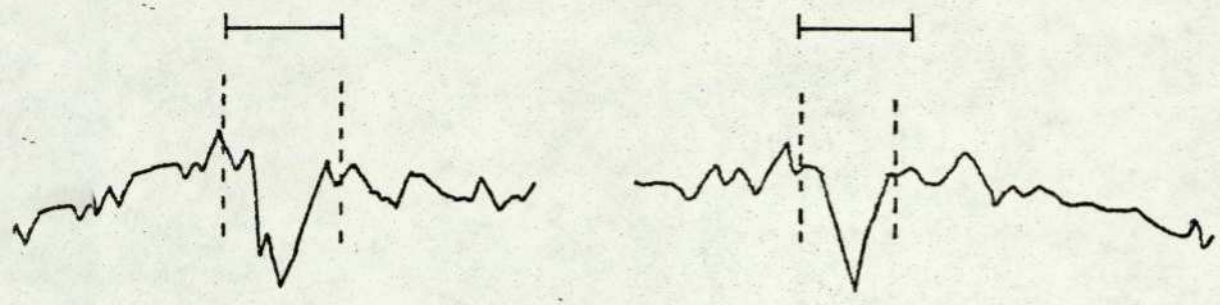
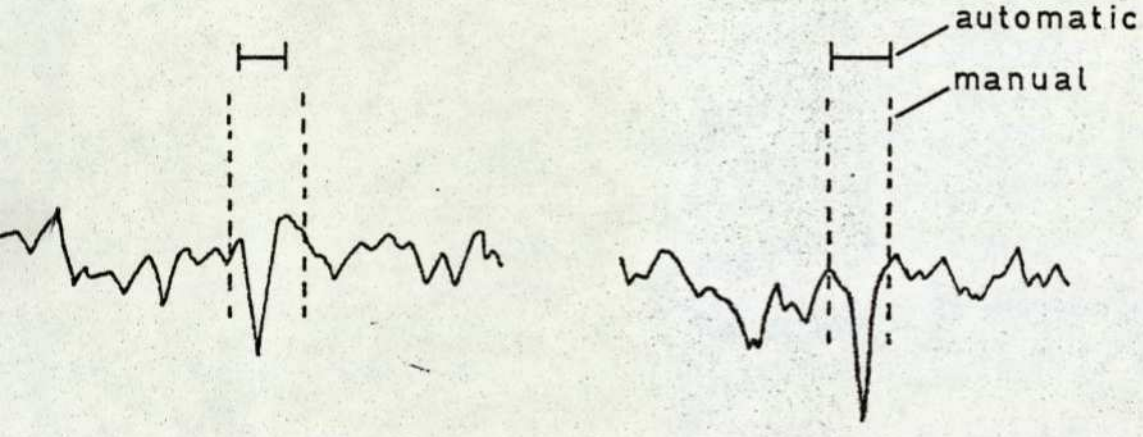


FIGURE 5.10C - AUTOMATIC VS MANUAL WAVEFORM DELINEATION



BLACK DOTS



FIVE-STAND RING

FIGURE 5.10D - AUTOMATIC VS MANUAL WAVEFORM DELINEATION

necessary waveform delineation in the scanner response signal, and this had therefore been carried out manually. The work reported here was intended to achieve this delineation automatically.

Three techniques were identified for this task, each of which provided also for prior detection of the defect waveform. These were:

(1) a modification of the detection system used by the SIRA Institute;

(2) a technique based on time series prediction, and similar to (1);

(3) a technique based on the concept of matched filtering, with a bank of filters chosen to cover the waveforms of interest.

These techniques were simulated and an attempt made to compare their detection capabilities in a quantitative way. This comparison indicated a preference for the matched filter bank. More importantly, however, the simulation showed that only this technique offered any real promise for waveform delineation. Consequently, the matched filter scheme was investigated in detail.

Implementation was as a bank of correlation detectors, and a straightforward analysis showed the value of symmetrical, zero mean match waveforms in coping with the low frequency content of the scanner response signal. A set of six simple triangular waveforms was found to be satisfactory, and a normalisation was derived which allowed a single detection threshold to be used on all filters, and simple amplitude comparisons to determine the best match. The resulting delineation algorithm was implemented and the results compared with those determined manually for the work on defect recognition. With few exceptions, close correspondence was observed between the two. There is little doubt in the author's mind that, if the recognition processing was to be repeated with delineation from the matched filter bank, closely similar results would be achieved. Indeed, because of the greater consistency attached to automatic delineation, better recognition performance could easily result.

It seems that the essential characteristic of the matched filter is that it makes use of the a-priori knowledge that defect waveforms are "pulse-like", to the extent that they fall and then rise again over a certain interval. No great significance should be attached to the triangular waveforms adopted, except insofar as they conveniently embody this characteristic. Actual defect waveforms are often far from triangular in shape, without unduly disturbing the technique.

Especially interesting to this work is the fact that a bank of matched filters, in digital form, is identical to a linear classifier operating with successive waveform samples as features. The stored match waveforms determine the linear weights, and the process of selecting the best match is identical to the maximum selector of the linear classifier. In these terms, the detection threshold is analogous to incorporating a reject capability into the classifier. This equivalence suggests that the filter outputs might well be used as features for the recognition system proper, to provide a so-called "layered" system overall.

Finally, the question of implementing a matched filter bank at typical on-line processing speeds has yet to be discussed. This will be taken up in Chapter 6, where it will be shown that a close approximation to the digital system can be achieved with primarily analogue hardware.

APPENDIX 5.1 - Symmetrical, Zero-mean Match Waveforms

As in equation (5.5.1), we can relate the response, y_j, of a matched filter (realised as a correlation detector) to any sequence of input samples $(x_{\ell+1}, x_{\ell+2}, -, x_{\ell+n})$ through the stored match waveform $(s_1, s_2, -, s_n)$:

$$y_j = \sum_{i=1}^{n} x_{\ell+i}\, s_i \qquad \text{..... (A5.1.1)}$$

Let the match waveform be symmetrical with zero mean value, so that:

(1) $s_i = s_{n-i+1}$, $i = 1, 2, -, m$,

where $n = 2m$ if n is even,

or $n = (2m + 1)$ if n is odd;

and (2) $\sum_{i=1}^{n} s_i = 0.$

We shall show that, under these constraints, the response of the filter is zero for any input sequence of constant value ($x_i = K$, $i = 1, =, -, n$) and for any input sequence which follows a linear ramp ($x_i = Ki$, $i = 1, 2, -, n$).

For a constant input, the proof is simple. Substituting $x_i = K$ into equation (A5.1.1) gives

$$y_j = K \sum_{i=1}^{n} s_i$$

and since $\sum_{i=1}^{n} s_i = 0$, $y_j = 0$ as required.

For a ramp input, we must treat separately the case when n is even, and the case when n is odd.

n even

In this case n = 2m.

We have $\sum_{i=1}^{n} s_i = 0$ (zero mean)

therefore

$$\sum_{i=1}^{m} s_i + \sum_{i=m+1}^{2m} s_i = 0.$$

We have $s_i = s_{2m-i+1}$ (symmetry)

therefore

$$\sum_{i=1}^{m} s_{2m-i+1} + \sum_{i=m+1}^{2m} s_i = 0.$$

Substituting p = 2m - i + 1:

$$\sum_{p=2m}^{m+1} s_p + \sum_{i=m+1}^{2m} s_i = 0$$

and the two summations are identical.

Therefore

$$\sum_{p=2m}^{m+1} s_p = 0 \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (A5.1.2)$$

For a ramp input $x_i = Ki$. Substituting into (A5.1.1), we have:

$$y_j = \sum_{i=1}^{n} K.i.s_i$$

$$= K\left[\sum_{i=1}^{m} i.s_i + \sum_{i=m+1}^{2m} i.s_i\right]$$

Substituting $s_i = s_{2m-i+1}$ (symmetry)

$$y_j = K\left[\sum_{i=1}^{m} i.s_{2m-i+1} + \sum_{i=m+1}^{2m} i.s_i\right]$$

Substituting $p = 2m - i + 1$:

$$y_j = K\left[\sum_{p=2m}^{m+1} (2m+1-p)s_p + \sum_{i=m+1}^{2m} i.s_i\right]$$

$$= K\left[(2m+1)\sum_{p=2m}^{m+1} s_p - \sum_{p=2m}^{m+1} p\, s_p + \sum_{i=m+1}^{2m} i\, s_i\right]$$

The last two summations are identical, and therefore cancel.

Therefore

$$y_j = K(2m+1)\sum_{p=2m}^{m+1} s_p$$

and substituting (A5.1.2), we have

$$y_j = 0, \text{ as required.}$$

n odd

In this case $n = 2m + 1$.

We have $\sum_{i=1}^{n} s_i = 0$ (zero mean)

therefore

$$\sum_{i=1}^{m} s_i + \sum_{m+2}^{2m} s_i + s_{m+1} = 0$$

We have $s_i = s_{2m-i+2}$ (symmetry)

therefore

$$\sum_{i=1}^{m} s_{2m-i+2} + \sum_{i=m+2}^{2m} s_i + s_{m+1} = 0.$$

Substituting $p = 2m - i + 2$:

$$\sum_{p=2m+1}^{m+2} s_p + \sum_{i=m+2}^{2m} s_i + s_{m+1} = 0$$

and the two summations are identical.

Therefore

$$2 \sum_{p=2m+1}^{m+2} s_p + s_{m+1} = 0$$

therefore

$$\sum_{p=2m+1}^{m+2} s_p = \frac{-s_m + 1}{2} \quad \ldots\ldots\ldots\ldots\ldots\ldots \quad (A5.1.3)$$

For a ramp input $x_i = Ki$. Substituting into (A5.1.1), we have:

$$y_j = \sum_{i=1}^{n} K.i.s_i$$

$$= K\left[\sum_{i=1}^{m} i.s_i + \sum_{i=m+2}^{2m+1} i.s_i + (m+1)\, s_{m+1}\right]$$

Substituting $s_i = s_{2m-i+2}$ (symmetry),

$$y_j = K\left[\sum_{i=1}^{m} i.s_{2m-i+2} + \sum_{i=m+2}^{2m+1} i.s_i + (m+1)\, s_{m+1}\right]$$

Substituting $p = 2m - i + 2$:

$$y_j = K\left[\sum_{p=2m+1}^{m+2} (2m+2-p)\, s_p + \sum_{i=m+2}^{2m+1} i.s_i + (m+1)\, s_{m+1}\right]$$

$$= K\left[2(m+1) \sum_{p=2m+1}^{m+2} s_p - \sum_{p=2m+1}^{m+2} p.s_p + \sum_{i=m+2}^{2m+1} i.s_i + (m+1)\, s_{m+1}\right]$$

The last two summations are identical and therefore cancel.

Therefore

$$y_j = K\left[2(m+1) \sum_{p=2m+1}^{m+2} s_p + (m+1)\, s_{m+1}\right]$$

and substituting (A5.1.3), we have

$$y_j = K\left[2(m+1)\left(\frac{-s_{m+1}}{2}\right) + (m+1)\, s_{m+1}\right]$$

therefore

$$y_j = 0, \text{ as required.}$$

APPENDIX 5.2 - Filter Normalisation Parameters

Let s_i, i = 1, 2, - , n be a match waveform prior to normalisation.

Let T_i, i = 1, 2, - , n be the same match waveform after normalisation.

Let the normalisation be of the form:

$$T_i = \frac{s_i - A}{B} .$$

We wish to determine the normalisation parameters, A and B, so that the normalised waveform will yield a filter response of unity to an input waveform which is perfectly matched, but scaled to unit amplitude.

After normalisation, the perfectly matched input waveform is simply the match waveform itself,

$$T_i, \; i = 1, 2, -, n.$$

Let $$T_{max} = \text{maximum } T_i \text{ w.r.t. } i$$

and $$T_{min} = \text{minimum } T_i \text{ w.r.t. } i$$

Then the input waveform which is perfectly matched, but scaled to unit amplitude, is U_i, i = 1, 2, -, n, where:

$$U_i = \frac{T_i}{(T_{max} - T_{min})} , \; i = 1, 2, -, n.$$

As in equation (5.5.1), we can relate the response, y_j, of a matched filter (realised as a correlation detector) to any sequence of input samples $(x_{\ell+1}, x_{\ell+2}, - , x_{\ell+n})$ through the stored match waveform $(T_i, i = 1, 2, -, n)$:

$$y_j = \sum_{i=1}^{n} x_{\ell+i} \, T_i$$

We require that $y_j = 1$ when $x_{\ell+i} = U_i$, i=1, 2, -, n.

i.e. $$1 = \sum_{i=1}^{n} U_i T_i$$

Therefore

$$1 = \sum_{i=1}^{n} \left(\frac{T_i}{T_{max} - T_{min}} \right) T_i$$

therefore

$$\sum_{i=1}^{n} T_i^2 = T_{max} - T_{min} \quad \ldots\ldots\ldots\ldots \text{ Condition 1.}$$

In accordance with Appendix 5.1, we shall also require the normalised match waveform to have zero mean:

$$\sum_{i=1}^{n} T_i = 0 \quad \ldots\ldots\ldots\ldots\ldots\ldots \text{ Condition 2.}$$

From these two conditions, the normalisation parameters A and B can be determined, in terms of the match waveform prior to normalisation, s_i, i = 1, 2, -, n.

Condition 2 determines A, as follows:

$$\sum_{i=1}^{n} T_i = 0$$

therefore

$$\sum_{i=1}^{n} \left(\frac{s_i - A}{B} \right) = 0$$

therefore

$$\sum_{i=1}^{n} (s_i - A) = 0$$

therefore

$$A = \frac{1}{n} \sum_{i=1}^{n} s_i \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \text{ (A5.2.1)}$$

i.e. A is simply the mean value of the match waveform, prior to normalisation.

Given this value of A, Condition 1 determines B, as follows:

$$\sum_{i=1}^{n} T_i^2 = T_{max} - T_{min}$$

therefore

$$\sum_{i=1}^{n} \left(\frac{s_i - A}{B} \right)^2 = \left(\frac{s_{max} - A}{B} \right) - \left(\frac{s_{min} - A}{B} \right)$$

since $$T_{max} = \frac{s_{max} - A}{B}$$

and $$T_{min} = \frac{s_{min} - A}{B}$$

where s_{max} = maximum s_i w.r.t. i

s_{min} = minimum s_i w.r.t. i

Therefore

$$\sum_{i=1}^{n} (s_i^2 - 2A\, s_i + A^2) = B\ (s_{max} - s_{min})$$

therefore

$$\sum_{i=1}^{n} s_i^2 - 2A \sum_{i=1}^{n} s_i + nA^2 = B\ (s_{max} - s_{min})$$

Substituting equation (A5.2.1):

$$\sum_{i=1}^{n} s_i^2 - \frac{2}{n} \left[\sum_{i=1}^{n} s_i \right]^2 + \frac{1}{n} \left[\sum_{i=1}^{n} s_i \right]^2$$

$$= B\ (s_{max} - s_{min})$$

therefore

$$B = \frac{\sum_{i=1}^{n} s_i^2 - \frac{1}{n} \left[\sum_{i=1}^{n} s_i \right]^2}{(s_{max} - s_{min})} \qquad \text{(A5.2.2)}$$

Equations (A5.2.1) and (A5.2.2) are the required relations.

HARDWARE IMPLEMENTATION

6.1 Introduction

This chapter will be devoted to suggestions for implementing the signal processing schemes developed and evaluated in this thesis. Only those schemes which have shown promise will be treated: namely, defect detection and delineation with a bank of matched filters, feature extraction from a defect pulse, and defect recognition with a linear feature space classifier. Designs will be discussed for special-purpose analogue and digital hardware, capable of meeting the system requirements of processing speed and cost. The matched filter bank presents the most severe difficulties, in this respect, and most of this chapter is therefore devoted to this.

Figure 6.1 shows the primary system components in block diagram form, and should be self-explanatory. We shall develop each component in turn.

6.2 Defect Detection and Delineation

This component must accept as input the analogue scan signal and produce as output a corresponding binary detection and delineation signal denoting the location and extent of any defect signals present. Paralleling the simulation described in Chapter 5, the following operations must be realised:

(1) Processing of the analogue signal by a bank of filters matched to the range of defect waveforms.

(2) Comparison of the filter responses, both amongst themselves and against a preset detection threshold, so as to determine which filter yields the largest response (best match) and whether that response is significant (greater than the detection threshold).

(3) Processing of the filter responses, so as to detect peaks.

(4) Combination of the results of (2) and (3) so as to generate a binary detection and delineation signal of

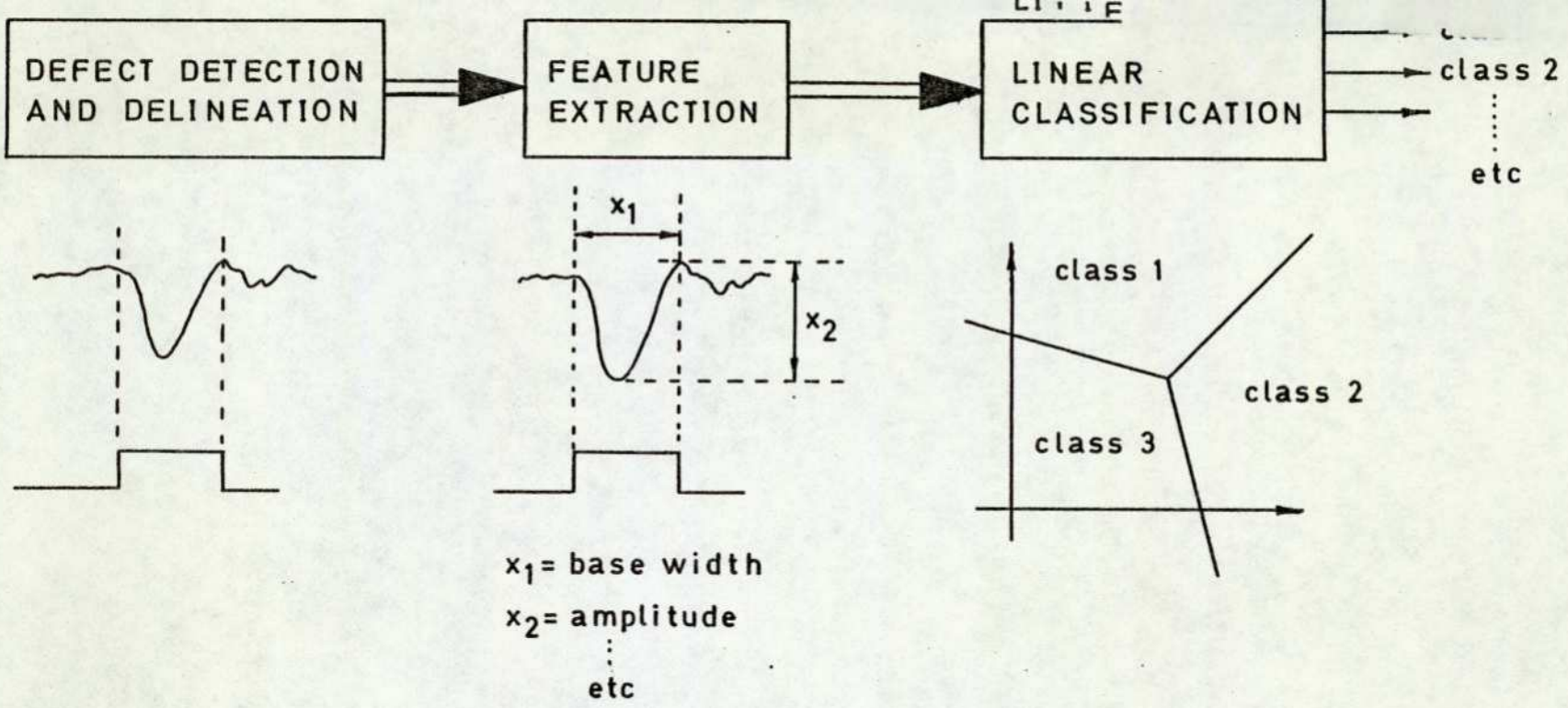


FIGURE 6.1 - THE SYSTEM COMPONENTS.

appropriate duration (according to the best-match filter) and appropriately located in time (according to the peak in the response of the best-match filter).

A suitable system is shown in block diagram form in Fig. 6.2. The elements of this system are as follows:

(a) Signal delay line - introducing pure delay into the analogue signal.

(b) Matched filters - each with the same observation delay, τ, so that their responses are in time synchronism for subsequent processing.

(c) Largest value block - has, in this example, four analogue inputs and four corresponding binary outputs (of which only three are used). The output corresponding to the largest analogue input must assume the value binary 1, whilst all other outputs assume binary 0.

(d) Peak detectors - generate a short binary pulse each time the input signal passes through a peak value, where a peak value is defined by the first derivative of the signal passing through zero, from +ve to -ve.

(e) AND gates - standard logic elements.

(f) Pulse generators - standard edge triggered monostables, each matched in its pulse width to the match waveform of the corresponding filter.

Although Fig. 6.2 shows only three filters, the extension to six is straightforward. Fig. 6.3 shows an associated waveform timing diagram, cross-referenced to Fig. 6.2 via the encircled letters. Operation is best explained via the waveform timing diagram.

Signal A represents an idealised response signal from a SIRA scanner. The negative-going defect pulse excites a response from each filter, to give the signals C, D and E. In this example, the second filter is most closely matched to the defect pulse, and, accordingly, yields the largest peak response. This is reflected in the corresponding outputs of the largest value block, signals

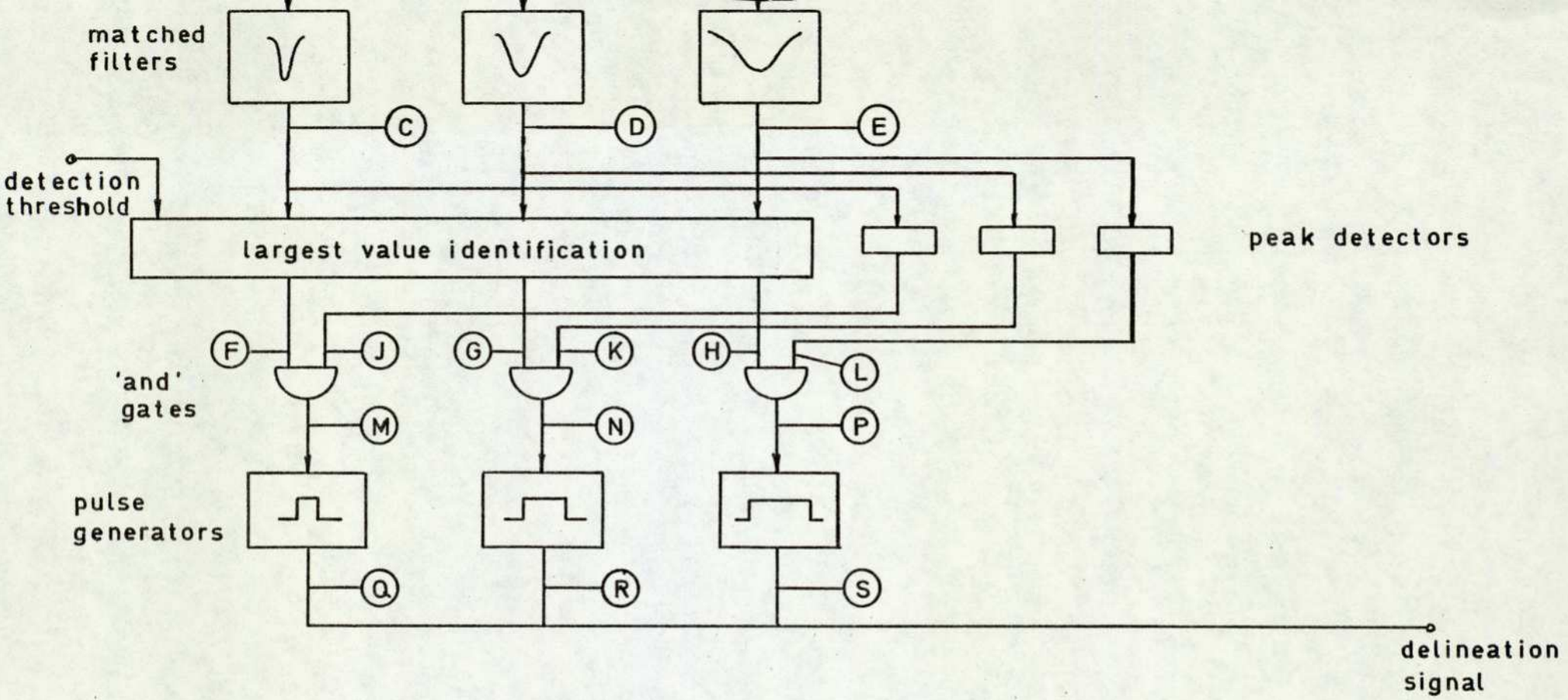


FIGURE 6.2 - MATCHED FILTER BANK : HARDWARE.

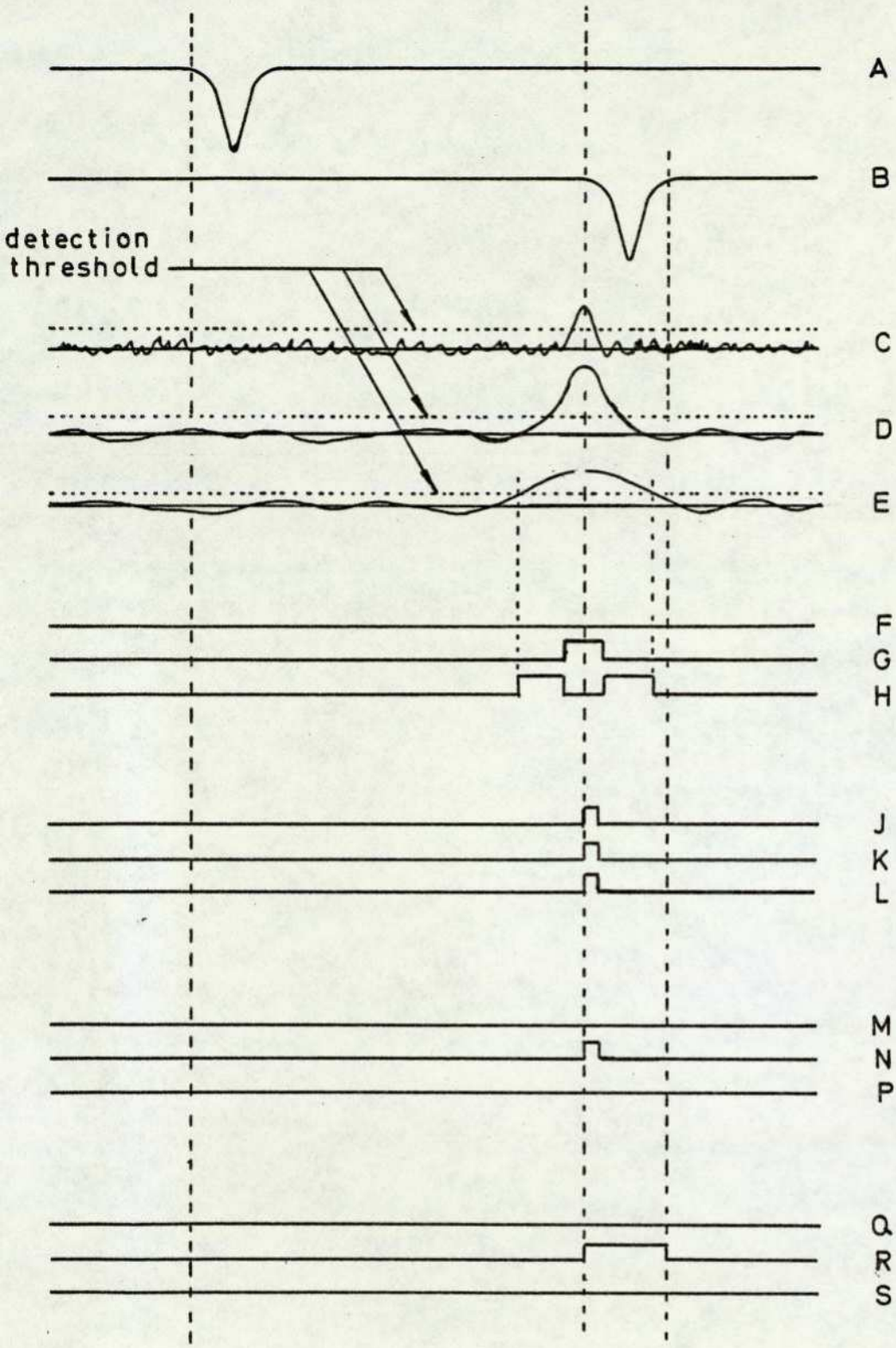


FIGURE 6.3 - MATCHED FILTER BANK : TIMING.

F, G and H, at the waveform peak. Notice that the detection threshold is treated simply as an additional input to this block, and will therefore hold the three output signals to zero when it is, itself, the largest input.

Signals J, K and L are the peak detector outputs in response to the signals C, D and E. Each is gated with the corresponding output of the largest value block, to give the signals M, N and P. By this means, only that peak detector pulse from the best-match filter is passed on to the corresponding pulse generator, and this responds with the detection and delineation signal, R, of appropriate duration.

Because of the inherent delay in the signal processing which generates this signal (primarily in the filters themselves) the scanner response signal, A, must be correspondingly delayed to give the signal B, which is synchronous in time with the delineation signal.

Of the six elements of this system previously listed, the last three (peak detectors, AND gates and pulse generators) are standard items and need not be discussed further. Signal delay lines are also standard items, although we shall see that this element must meet a rather demanding performance specification. Two elements, therefore, remain to be discussed - the matched filters and the largest value block.

6.2.1 The Matched Filters

In the simulation reported in Chapter 5, the filters were realised as correlation detectors. Such a realisation is relatively slow and expensive (although new technologies such as surface acoustic wave and charge coupled devices promise to change this in the near future). The discussion to follow will show that there are good grounds for believing that very similar results can be achieved with fast, cheap and fairly simple analogue filters.

We know that the impulse response, $h(t)$, of a filter matched to a waveform, $f(t)$, is a scaled, time-reversed, delayed replica of that waveform:

$$h(t) = k.f(\tau - t),$$

where k is an arbitrary scaling factor,
and τ is a chosen observation delay time, after which the filter output is to be examined.

Fig. 6.4 illustrates this relationship, with k equal to one, for the triangular waveforms used in the simulation. (Notice that τ must not be less than b for physical realisability.)

The discussion to follow will be simplified for waveforms which are symmetrical in time, and h(t) in Fig. 6.4 is a delayed version of such a waveform:

$$h(t) = h'(t - T)$$

where $$T = \tau - \frac{b}{2}.$$

h'(t) has the required symmetry (h'(t) = h'(-t)) and therefore has a Fourier Transform, H'(ω), which is wholly real.

H'(ω) is the transform of a symmetric triangle, centred at the time origin, and is given by (ref. 34):

$$H'(\omega) = \frac{A.b}{2}\left[\frac{\mathrm{Sin}(\omega b/4)}{\omega b/4}\right]^2$$

Fig. 6.5 shows this relationship, and defines A and b.

It would be possible to realise H'(ω), with delay, by a low-pass filter plus a bank of band-pass filters connected in parallel - the former for the main lobe and the latter for the side-lobes. But there are infinitely many side-lobes, of successively decreasing amplitude, and it is natural to question their significance.

With this in mind, we shall consider the transform of a symmetric Gaussian pulse, centred at the time origin (ref. 34):

$$g(t) = A.\exp(-t^2/2\sigma_t^2) \quad \ldots\ldots\ldots \quad (6.1)$$

$$G(\omega) = \sqrt{2\pi}.A.\sigma_t.\exp(-\omega^2\sigma_t^2/2) \quad \ldots\ldots \quad (6.2)$$

This transform pair is shown in Fig. 6.6. Both members are Gaussian, and so G(ω) has no side-lobes. It can be realised

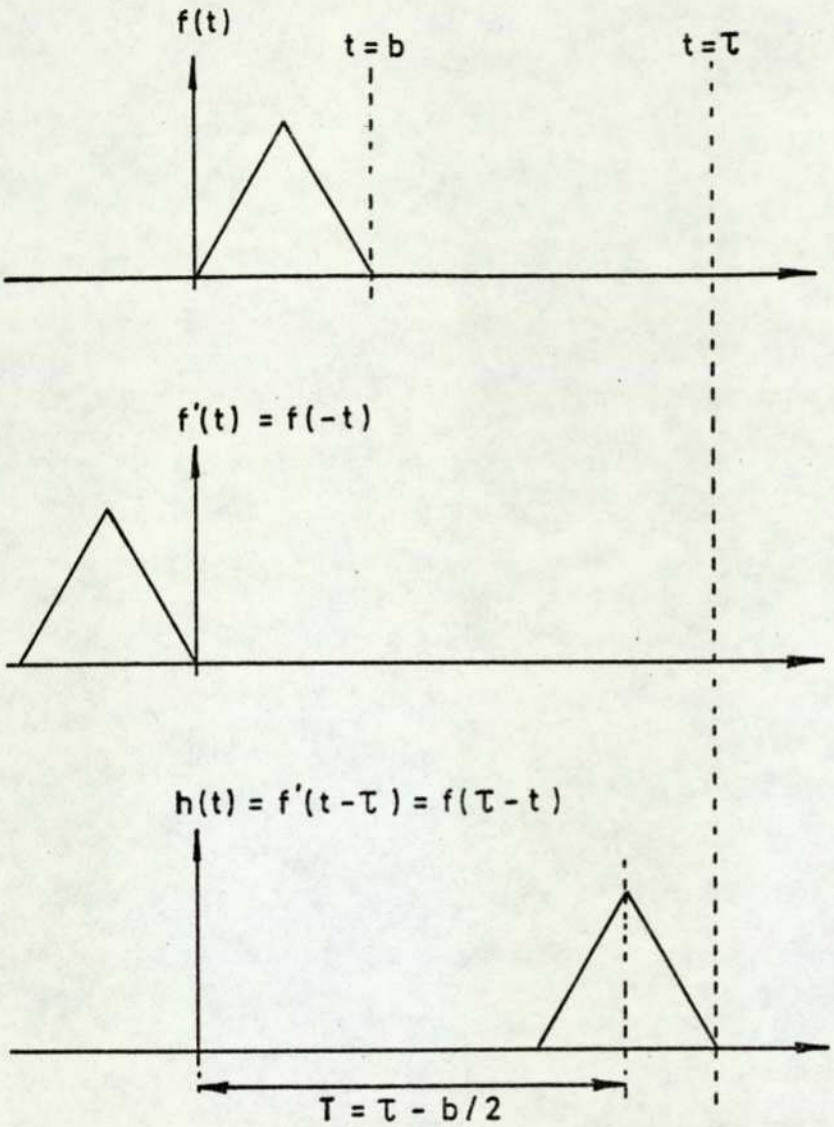


FIGURE 6.4 - TIME - REVERSAL AND DELAY.

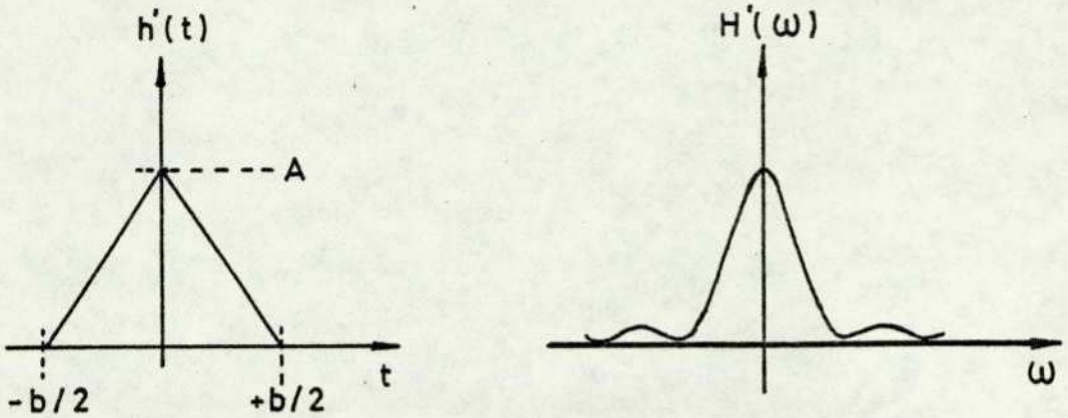


FIGURE 6.5 - THE FOURIER TRANSFORM OF A TRIANGULAR PULSE.

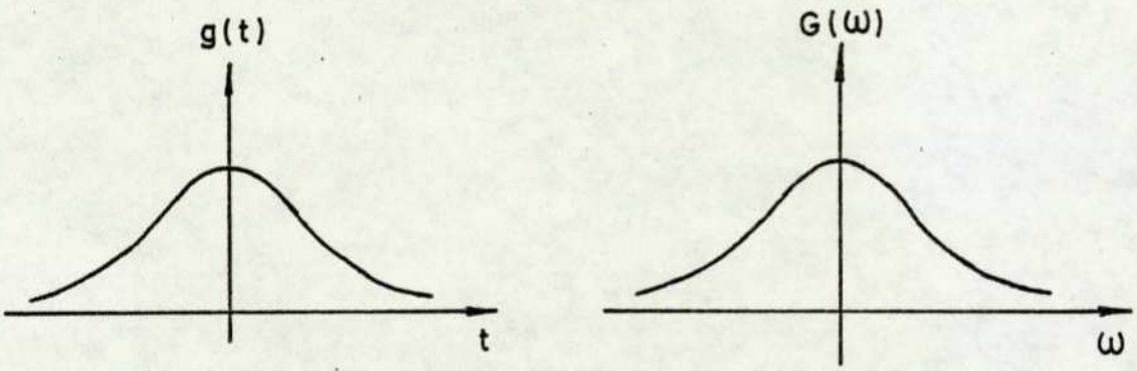


FIGURE 6.6 - THE GAUSSIAN TRANSFORM PAIR.

with a single low-pass filter. Such a filter would, of course, be matched to the Gaussian pulse of Fig. 6.6a, and we shall therefore consider the effect of using Gaussian match waveforms, rather than triangular.

In Section 5.5 it was noted that the precise shape of the match waveforms was not particularly significant, primarily because any shape could only approximate to that of actual defect waveforms. The essential feature seemed to be that the match waveform should rise to a peak and then fall again (or vice versa) over some pre-determined time period. Clearly, Gaussian waveforms are as good as triangular in this respect. Furthermore, many large area, low contrast defects yield signals for which a Gaussian pulse is a better approximation.

Reference 35 contains an analysis of the matched filter design problem for signals of rectangular form. The response of the optimum matched filter is compared with that of a Gaussian filter, amongst others. It is shown that the peak signal-to-noise ratio for the Gaussian filter is just 0.5dB below that of the optimum filter. But a Gaussian waveform is probably a poorer approximation to a rectangular pulse than to a triangular pulse. We can therefore expect even less difference.

6.2.1.1 Equivalent Gaussian Filters

If Gaussian match waveforms are to be used, a relationship must be established between the parameters which define a triangular pulse, and those which define an "equivalent" Gaussian pulse. For the triangular pulse, these are base width and amplitude, and for the Gaussian pulse, standard deviation and amplitude. Amplitude is simply a matter of filter gain, and the essential requirement is to match the two pulses in their base width. Since this is ill-defined for a Gaussian pulse, the following indirect approach can be taken:

Let the amplitude of the two pulses be the same, and let the parameters b (the triangle base width) and σ_t (the Gaussian standard deviation) be adjusted for equal area, as follows:

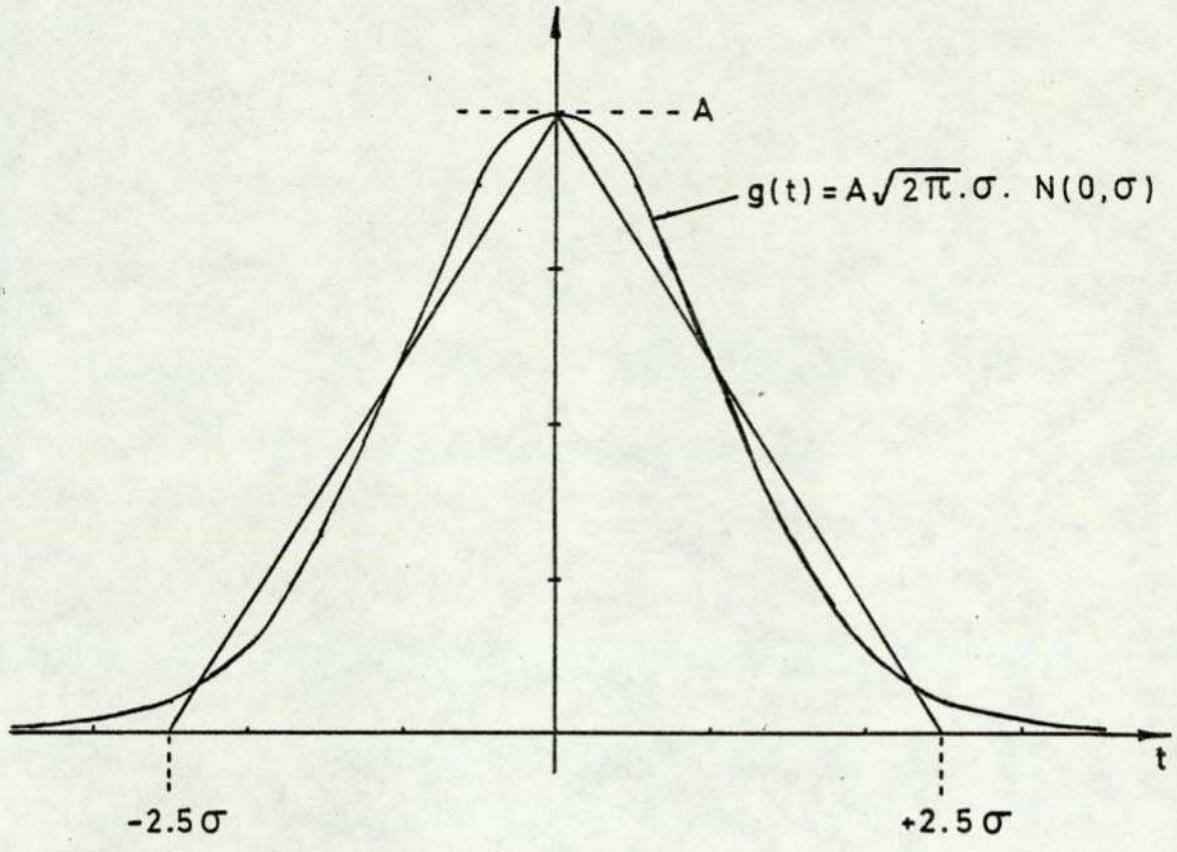


FIGURE 6.7 - THE 'EQUIVALENT' GAUSSIAN PULSE.

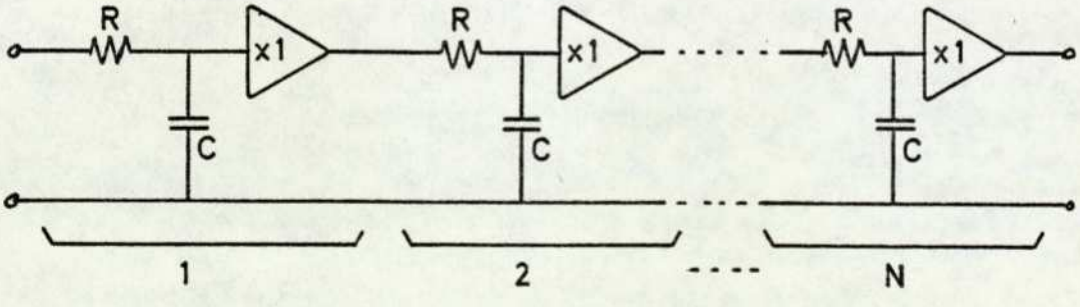


FIGURE 6.8 - CASCADE LOW-PASS CIRCUIT.

The triangular pulse of amplitude A and base width b, has area A_T, where

$$A_T = \frac{A.b}{2} .$$

The Gaussian pulse of amplitude A and standard deviation σ_t, is defined by

$$g(t) = A.\sqrt{2\pi}.\sigma_t.N(0, \sigma_t)$$

where $N(0, \sigma_t)$ is the standard Gaussian pdf, of amplitude $(\sqrt{2\pi}.\sigma_t)^{-1}$.

Since $N(0, \sigma_t)$ has unit area, f(t) has area A_G, where

$$A_G = A.\sqrt{2\pi}.\sigma_t.$$

Equating A_G and A_T yields

$$\frac{A\,b}{2} = A\,\sqrt{2\pi}\,\sigma_t.$$

Therefore $$\sigma_t = \frac{b}{2\sqrt{2\pi}} = \frac{b}{5.132}$$

For practical purposes, we can take

$$\sigma_t = \frac{b}{5}$$

and Fig. 6.7 illustrates this choice.

Equations (6.1) and (6.2) show that the standard deviation in the frequency domain is the reciprocal of that in the time domain,

i.e. $$G(\omega) = A' \exp(-\omega^2/2\sigma_\omega^2) \quad \ldots\ldots\ldots \quad (6.3)$$

where $$A' = A.\sqrt{2\pi}.\sigma_t$$

and $$\sigma_\omega = \frac{1}{\sigma_t}$$

We can therefore relate the triangle base width to the frequency transfer function of the filter, using

$$\sigma_\omega = \frac{1}{\sigma_t} = \frac{5}{b} .$$

To complete the frequency domain specification of the Gaussian matched filter, we must determine the gain factor, A', in equation (6.3). In the simulation work of Chapter 5, the filter gains were set so as to yield a response of unit amplitude to an input waveform perfectly matched, but itself scaled to unit amplitude. We must therefore determine A' in the same way.

Consider, therefore, a Gaussian pulse of unit amplitude:

$$f(t) = e^{-t^2/2\sigma_t^2}$$

with Fourier transform:

$$F(\omega) = \sqrt{2\pi}.\sigma_t.\exp\ (-\omega^2\ \sigma_t^2/2)$$

The corresponding matched filter has the frequency transfer function:

$$G(\omega) = A'\ \exp\ (-\omega^2/2\sigma_\omega^2)$$

or

$$G(\omega) = A'\ \exp\ (-\omega^2\ \sigma_t^2/2).$$

The filter response in the frequency domain, $Y(\omega)$, is therefore given by:

$$Y(\omega) = F(\omega)\ G(\omega)$$

$$= A'.\sqrt{2\pi}.\sigma_t\ \exp\ (-\omega^2\ \sigma_t^2)$$

which transforms into the time domain, to give $y(t)$:

$$y(t) = A'.\sqrt{2\pi}.\sigma_t.\left\{\frac{1}{\sqrt{2\pi}.\sqrt{2}.\sigma_t}\ .\ \exp\ (-\ t^2/2(\sqrt{2}\ \sigma_t)^2)\right\}$$

$$= \frac{A'}{\sqrt{2}}\ .\ \exp\ (-\ t^2/2(\sqrt{2}\ \sigma_t)^2)$$

i.e. $y(t)$ remains Gaussian, but with an increased standard deviation (by a factor $\sqrt{2}$), and with amplitude $\frac{A'}{\sqrt{2}}$.

Our requirement is for a response of unit amplitude, so we must set:

$$A' = \sqrt{2}.$$

Notice that the filter gain is independent of σ_t, so that all filters in the bank will have the same (zero frequency) gain. Because of this, the actual gain is not important, and we can just as well set A' = 1 for all filters.

This gives the very simple result:

$$G(\omega) = \exp(-\omega^2/2\sigma_\omega^2) \quad \ldots\ldots\ldots\ldots \quad (6.4)$$

with $\sigma_\omega = \frac{5}{b}$, for equivalence with a triangular waveform of base width, b.

6.2.1.2 Circuit Configuration and Parameter Values

It can be shown that the circuit configuration of Fig. 6.8 has a frequency transfer function which tends to the Gaussian form, as the number of stages tends to infinity (ref. 36). More sophisticated design methods are available for Gaussian filters (e.g. ref. 37), but we shall consider just this one.

More precisely, the circuit of Fig. 6.8 has a frequency transfer function, G(ω), which tends to (ref. 36):

$$G(\omega) = \exp(-0.35\ \omega^2/B^2)\exp(-j\omega\tau)$$

where B is the -3dB frequency of the complete filter (rads. sec^{-1}),

and τ is the inherent filter delay.

Comparing this to equation (6.4), we have:

$$\frac{-\omega^2}{2\sigma_\omega^2} = \frac{-0.35\ \omega^2}{B^2}$$

therefore $B^2 = 0.7\ \sigma_\omega^2$

therefore $B = 0.84\ \sigma_\omega$

Substituting $\sigma_\omega = \frac{5}{b}$ gives:

$$B = 4.2/b \text{ rads. sec}^{-1} \quad \ldots\ldots\ldots \quad (6.5)$$

or

$$B = 4.2/2\pi b \text{ Hz} \quad \ldots\ldots\ldots\ldots\ldots \quad (6.6)$$

This expression relates the -3dB frequency of the multi-stage filter to the base width of the corresponding triangular match waveform.

In practice, of course, the number of stages must be finite. We must therefore consider the response of the circuit of Fig. 6.8 under this restriction.

For n identical stages, we have (ref. 36):

$$g(t) = \left(\frac{t}{T}\right)^{n-1} \exp\left(\frac{-t}{T}\right) \quad \ldots\ldots\ldots \quad (6.7)$$

and

$$B = \frac{1}{T}\left(2^{\frac{1}{n}} - 1\right)^{\frac{1}{2}} \quad \ldots\ldots\ldots\ldots\ldots \quad (6.8)$$

where $g(t)$ is the impulse response of the complete filter,

B is the -3dB frequency of the complete filter (rads. sec^{-1}),

and T is the time constant (= CR) of each stage.

Equation (6.7) is plotted in Fig. 6.9 for a few small values of n, with T adjusted in each case (according to equation (6.8)) to maintain the same overall -3dB frequency. Also shown is the Gaussian response for $n = \infty$, and all are scaled to the same amplitude. We can see that the approximation to the Gaussian form improves rapidly for small n, and more gradually as n increases. Even for n = 4, the approximation is remarkably good. As an example, Table 6.1 gives the necessary design parameters for the six filters used in the simulation work of Chapter 5. This table is calculated from equations (6.5) and (6.8), with n = 4.

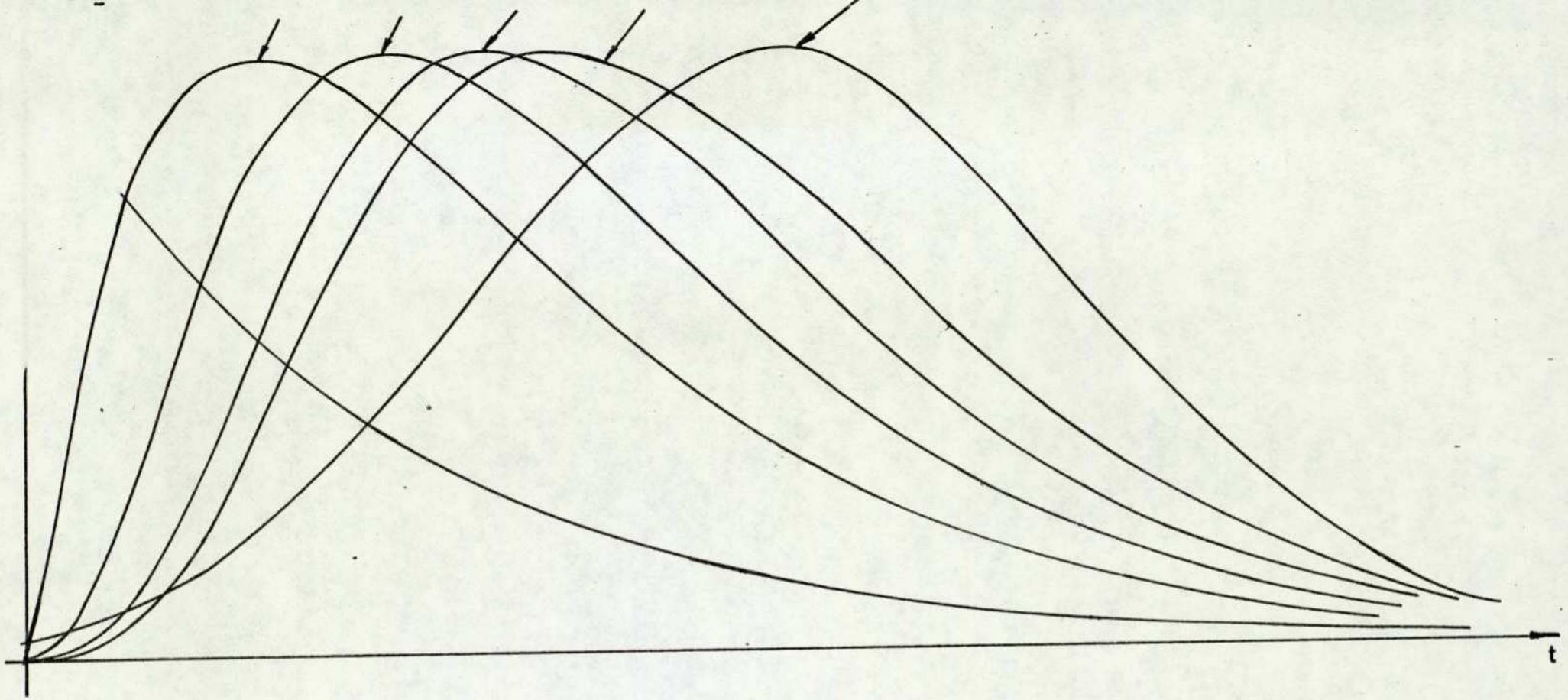


FIGURE 6.9 - IMPULSE RESPONSE OF THE CASCADE LOW-PASS CIRCUIT.

Filter number	Triangle base width		Filter -3dB frequency (rads. s^{-1})	Time constant (seconds)
	Number of samples b	Seconds b	B	T
1	3	3.10^{-6}	$1.\ 4.10^{6}$	$0.31.10^{-6}$
2	5	5.10^{-6}	$0.84.10^{6}$	$0.52.10^{-6}$
3	11	11.10^{-6}	$0.38.10^{6}$	$1.14.10^{-6}$
4	21	21.10^{-6}	$0.\ 2.10^{6}$	$2.17.10^{-6}$
5	41	41.10^{-6}	$0.\ 1.10^{6}$	$4.35.10^{-6}$
6	81	81.10^{-6}	$0.05.10^{6}$	$8.69.10^{-6}$

Table 6.1 Design Parameters for 4-stage R-C filters

6.2.1.3 Zero-Mean Match Waveforms

The development so far has assumed match waveforms with a base-line of zero. For the simulation work, however, the waveforms were adjusted to have zero mean value, so as to render the responses independent of the "low frequency" content of the input signal. It is clear that the bank of low-pass filters so far discussed does not exhibit this crucial characteristic.

The true Gaussian pulse cannot be adjusted to zero mean value, because the "tails" extend to $\pm\ \infty$. It must therefore be truncated, and we shall impose this truncation at the limits of the equivalent triangular waveform, i.e. $\pm\ 2.5\ \sigma_t$. With this proviso, we can simulate the shift to zero mean value by subtracting, from the Gaussian pulse, a rectangular pulse of appropriate height. This procedure is illustrated graphically in Fig. 6.10, with the corresponding frequency domain interpretation. Since all signals are symmetrical in time, their transforms are wholly real, and can therefore be subtracted algebraically.

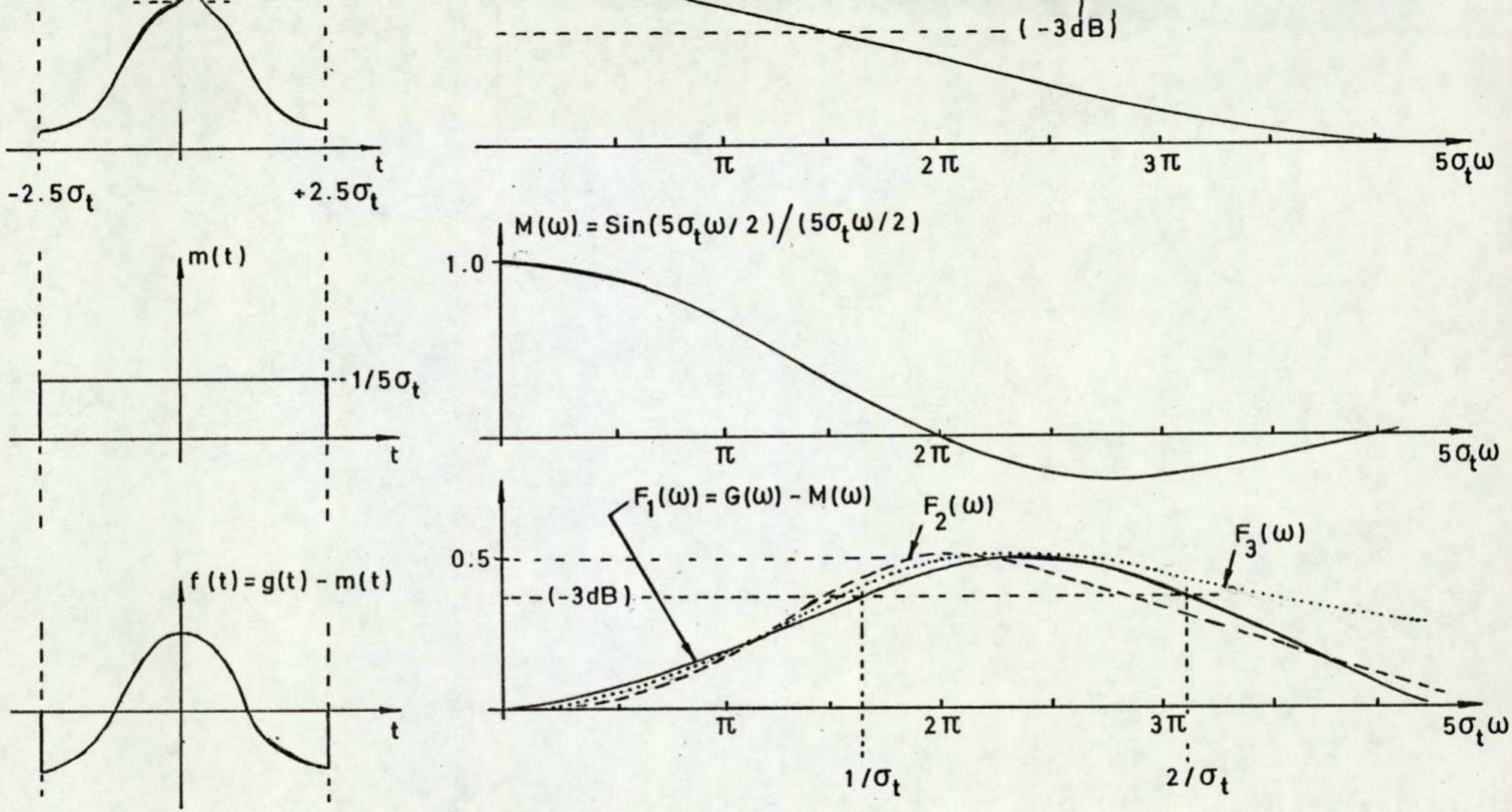


FIGURE 6.10 - THE SHIFT TO ZERO MEAN VALUE.

As might be expected, the resulting frequency transfer function, $F_1(\omega)$, is band-pass, rather than low-pass. The interesting point is that the lower and upper -3dB frequencies of $F_1(\omega)$ are, almost exactly, given by:

$$\omega_1 = \frac{1}{\sigma_t}, \quad \omega_2 = \frac{2}{\sigma_t}.$$

The shift to zero mean value has therefore modified the frequency transfer function in two significant respects:

(1) The upper -3dB frequency has been increased from

$$\omega_2 = 0.84\,\sigma_\omega = \frac{0.84}{\sigma_t}$$

to

$$\omega_2 = \frac{2}{\sigma_t}.$$

(2) The low frequency response rolls-off from a lower -3dB frequency of one-half the upper, so that

$$\omega_1 = \frac{\omega_2}{2} = \frac{1}{\sigma_t}.$$

Instead of a bank of low-pass filters, we therefore have a bank of band-pass filters. Further, since the triangular match waveforms approximately double in their base width, from one filter to the next, the upper -3dB frequency of one band-pass filter will be approximately equal to the lower -3dB frequency of the next. Fig. 6.11 illustrates this situation. In other words, the filter bank turns out to be a constant-Q spectrum analyser, albeit with coarse frequency resolution.

We can convert our bank of low-pass filters to band-pass by preceding each one with a suitable high-pass filter. Ideally, the high-pass filter should yield the response defined by $F_1(\omega)/G(\omega)$, of Fig. 6.10. This ratio is plotted in Fig. 6.12. In practice, we need not reproduce the fall in this response beyond $5\sigma\omega = 7\pi/2$, because this affects only the "tail" of $F_1(\omega)$. With this proviso, the dotted curve shown in

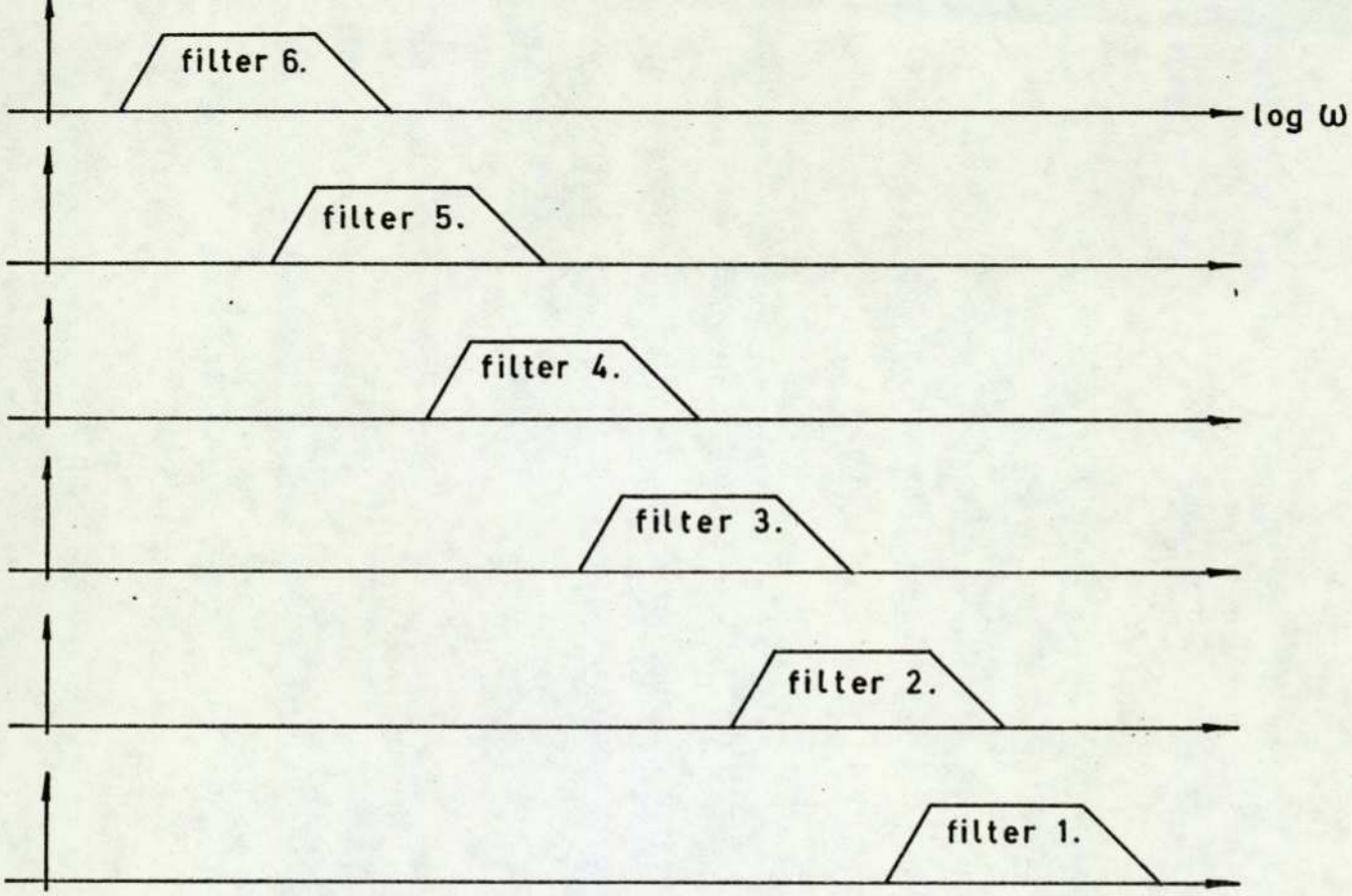


FIGURE 6.11 - THE MATCHED FILTERS IN THE FREQUENCY DOMAIN.

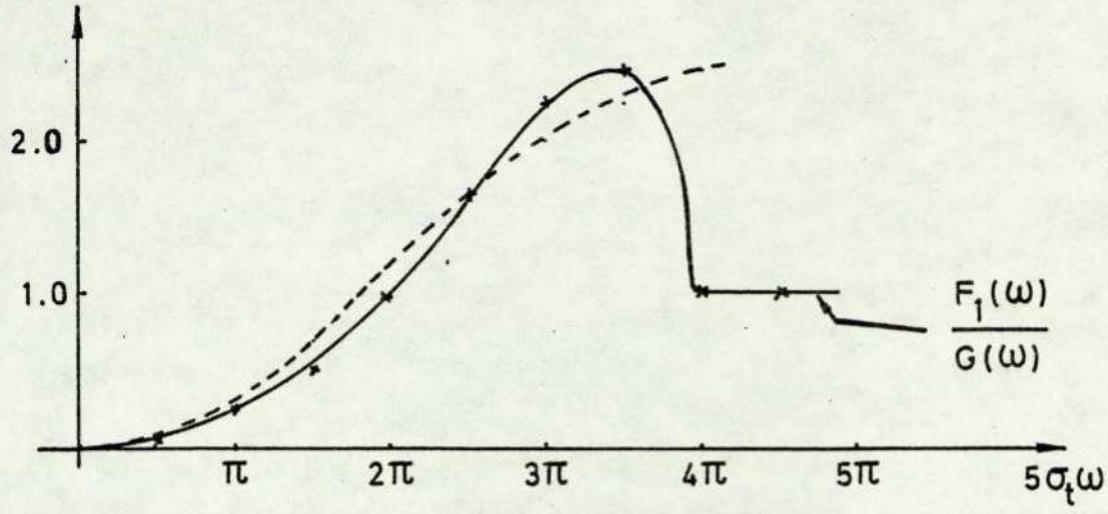


FIGURE 6.12 - THE EXACT HIGH-PASS FUNCTION AND IT'S APPROXIMATION.

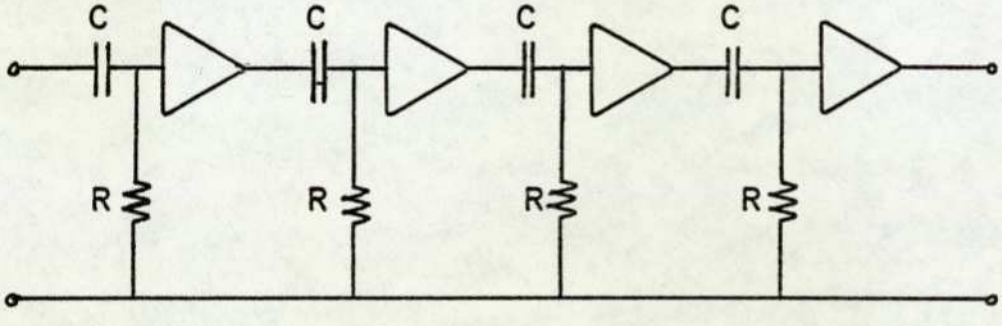


FIGURE 6.13 - CASCADE HIGH-PASS CIRCUIT.

Fig. 6.12 is a reasonable approximation, and this is derived from the 4-stage R-C filter of Fig. 6.13, with a -3dB frequency of

$$\omega = \frac{2.4}{\sigma_t} \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (6.9)$$

Multiplying this response by $G(\omega)$ yields the response $F_2(\omega)$, shown dotted in Fig. 6.10, which is a reasonable approximation to $F_1(\omega)$. More realistically, we can multiply the 4-stage high-pass response by the 4-stage low-pass approximation to $G(\omega)$, to give the (magnitude) response of the band-pass cascade. This yields the response $F_3(\omega)$, which is also shown dotted in Fig. 6.10. Again, the approximation is reasonable.

For the circuit of Fig. 6.13, the time-constant of each stage, T, is related to the overall -3dB frequency, ω, by:

$$T = (2^{\frac{1}{4}} - 1)^{-\frac{1}{2}} \cdot \frac{1}{\omega}$$

therefore $$T \doteq \frac{2.3}{\omega}$$

Substituting for ω from equation (6.9) gives:

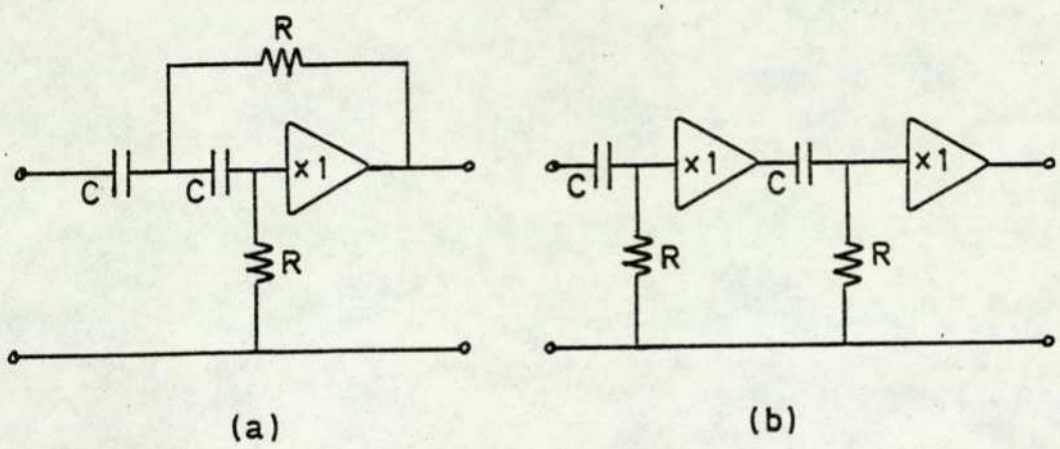
$$T \doteq \frac{2.3}{2.4}\,\sigma_t$$

therefore $$T \doteq 0.95\,\sigma_t.$$

Considering the approximation of Fig. 6.12, $T = \sigma_t$ would seem to be sensible.

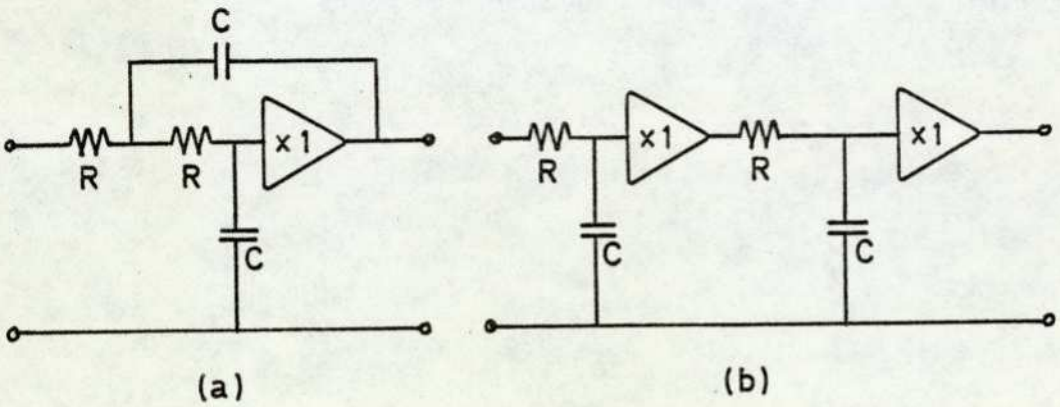
Our final solution, then, consists of a bank of band-pass filters, each composed of a 4-stage high-pass section, followed by a 4-stage low-pass section, and all with the same centre frequency gain. Although this design is particularly simple, it does call for eight buffer amplifiers. However, the circuit pairs shown in Fig. 6.14 have identical transfer functions, so that we need use only four amplifiers. No doubt a more sophisticated design could reduce this still further.

It is convenient to summarise here the necessary design equations developed in the text.



(a) (b)

HIGH-PASS CIRCUITS.



(a) (b)

LOW-PASS CIRCUITS.

FIGURE 6.14 - EQUIVALENT CIRCUIT PAIRS.

Let b be the base-width (seconds) of the triangular match waveform used in the simulation.

Then:

(1) $\sigma_t = \frac{b}{5}$; where σ_t is the standard deviation of the "equivalent" Gaussian pulse;

(2) $\omega_1 = \frac{1}{\sigma_t}$, $\omega_2 = \frac{2}{\sigma_t}$; where ω_1 and ω_2 are the lower and upper -3dB frequencies, respectively, of the corresponding band-pass filter (rads. sec^{-1});

(3) $\omega_3 = \frac{4.2}{b}$, $\omega_4 = \frac{12}{b}$ $\left\{ = \frac{2.4}{\sigma_t} \right\}$;

where ω_3 is the -3dB frequency of the 4-stage, high-pass section, and ω_4 the -3dB frequency of the 4-stage low-pass section, which are cascaded to produce the band-pass filter (both in rads. $sec.^{-1}$);

(4) $T_1 = \frac{(2^{\frac{1}{4}} - 1)^{-\frac{1}{2}}}{\omega_3} \cdot \left\{ \doteq \frac{2.3}{\omega_3} \doteq 0.55\ b \right\}$

$T_2 = \frac{(2^{\frac{1}{4}} - 1)^{\frac{1}{2}}}{\omega_4} \cdot \left\{ \doteq \frac{0.44}{\omega_4} \doteq 0.036\ b \right\}$

where T_1 is the common time constant of the high-pass section, and T_2 of the low-pass section (both in seconds).

Table 6.2 summarises this design for the six filters used in the simulation.

Filter number	b (seconds)	ω_1	ω_2	ω_3	ω_4	T_1	T_2
		(radians.sec^{-1})				(seconds)	
1	3.10^{-6}	$1.67.10^{6}$	$3.34.10^{6}$	$1.\ 4.10^{6}$	4.10^{6}	$1.65.10^{-6}$	$0.11.10^{-6}$
2	5.10^{-6}	$1.\ 0.10^{6}$	$2.\ 0.10^{6}$	$0.84.10^{6}$	$2.\ 4.10^{6}$	$2.75.10^{-6}$	$0.18.10^{-6}$
3	11.10^{-6}	$0.45.10^{6}$	$0.\ 9.10^{6}$	$0.38.10^{6}$	$1.\ 1.10^{6}$	$6.05.10^{-6}$	$0.39.10^{-6}$
4	21.10^{-6}	$0.24.10^{6}$	$0.48.10^{6}$	$0.\ 2.10^{6}$	$0.57.10^{6}$	$11.55.10^{-6}$	$0.76.10^{-6}$
5	41.10^{-6}	$0.12.10^{6}$	$0.24.10^{6}$	$0.\ 1.10^{6}$	$0.29.10^{6}$	$22.55.10^{-6}$	$1.48.10^{-6}$
6	81.10^{-6}	$0.06.10^{6}$	$0.12.10^{6}$	$0.05.10^{6}$	$0.15.10^{6}$	$44.55.10^{-6}$	$2.92.10^{-6}$

Table 6.2 Design Parameters for (4 + 4) Stage, R-C, Band-Pass Filters

6.2.1.4 Time Delay and Phase Response

We have so far discussed waveforms symmetrical in time, having transforms which are therefore wholly real (zero phase). In practice, of course, such waveforms are not realisable as impulse responses. A simple delay of $T = \tau - b/2$ (Fig. 6.4) is required, and this corresponds to a linear phase response:

$$\theta(\omega) = -\omega T .$$

τ is the observation time for a filter, and must be the same for all filters in the bank . Further, for each filter, τ must be not less than b, for physical realisability. The minimum value of τ is therefore determined by the maximum value of b in the bank. Fig. 6.15 illustrates this. In this figure, $f_1(t)$, $f_2(t)$ and $f_3(t)$ are the match waveforms of the three filters in the bank, and $h_1(t)$, $h_2(t)$ and $h_3(t)$ are the corresponding impulse responses (the waveforms are intended to represent 4-stage, low-pass R-C filters - see Fig. 6.9. They are drawn with zero base-line, rather than zero mean, for clarity).

Fig. 6.16 shows the phase response of the (4 + 4)-stage band-pass filter already discussed, together with the ideal response, $\theta(\omega) = -\omega T$, with $T = b/2 = 2.5\sigma$. The two responses are clearly different, although it is difficult to judge the significance of that difference for the task at hand. The question can be most easily resolved by constructing the filter and observing its impulse response.

Fig. 6.16 is directly applicable to the filter with the lowest pass band in the bank ($h_3(t)$ in Fig. 6.15). Other filters, however, will require additional delay of $(\tau - b)$, for a match waveform of duration b seconds. This will modify the overall phase response of these filters by adding a term $(-\omega(\tau - b))$ to that shown in Fig. 6.16. These delays could be achieved with a separate delay line for each of these filters, as implied in Fig. 6.2. Alternatively, the delay line which is shown there as delaying only the incoming signal, could be tapped and the filters supplied from those taps. In any case, this delay line must produce a delay

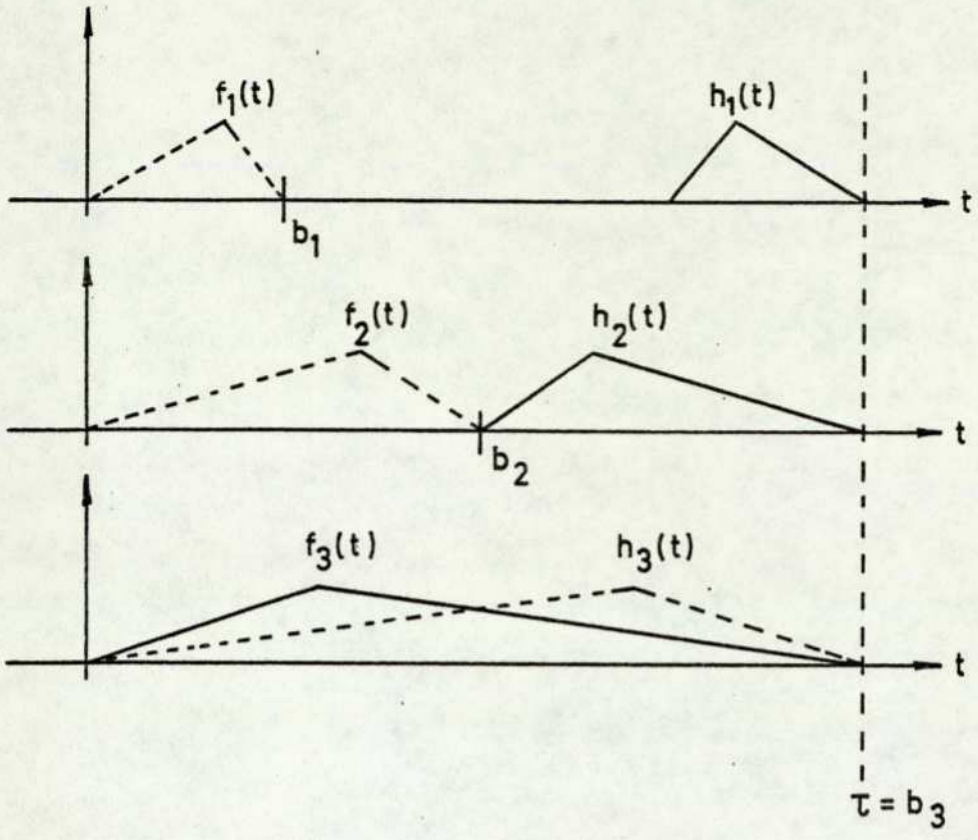


FIGURE 6.15 - FILTER IMPULSE RESPONSES DELAYED FOR COMPATIBILITY.

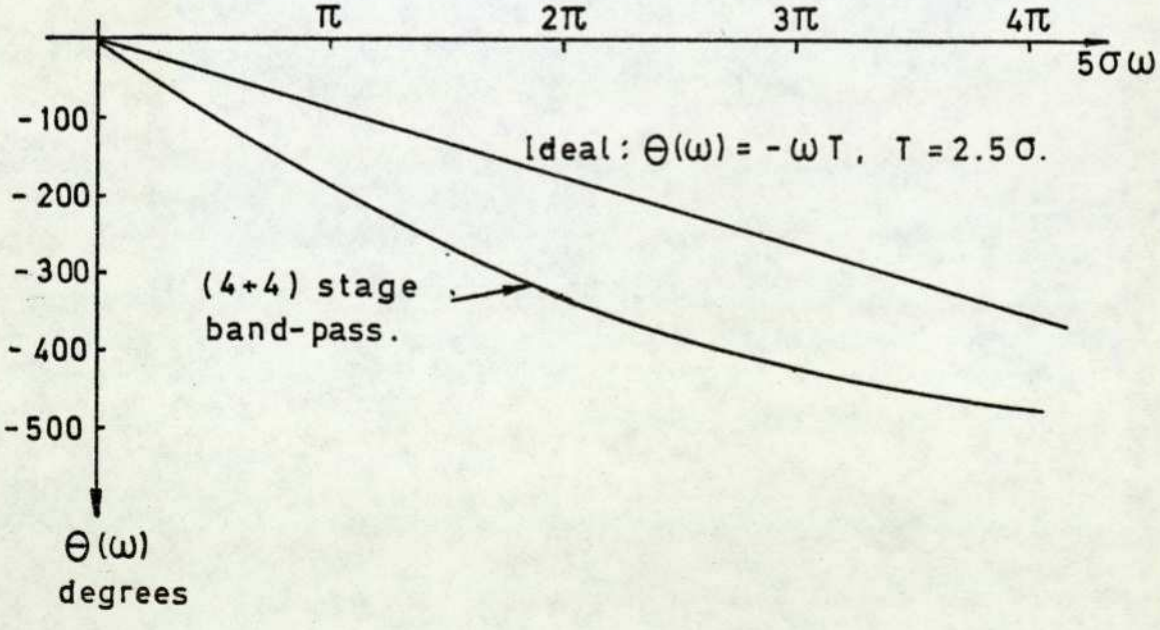


FIGURE 6.16 - PHASE CHARACTERISTICS.

determined, primarily, by the filter with the lowest pass-band (80.10^{-6} seconds in this work), with a signal bandwidth determined by the filter with the highest pass-band (greater than 3.10^6 rads. sec^{-1}, or 500 kHz). This is a very demanding specification, which can probably not be met by "lumped-constant", L-C delay lines. Other technologies, such as charge-coupled devices, may pose problems if a tapped line is to be used.

6.2.2 The Largest Value Block

It should be recalled that this block must accept n analogue inputs (comprising the filter outputs plus the detection threshold) and produce n corresponding binary outputs (one of which will not be used), such that the output corresponding to the largest input takes on the value logical one, and all other outputs logical zero. This can be achieved with a network of comparators connected between the inputs, followed by a logic network to decode the comparator responses. For three filters, this scheme is shown in Fig. 6.17.

Each comparator yields a logical output: +1 if the positive input is greater than or equal to the negative, and zero otherwise. Comparisons are made between all pairs of analogue inputs, and the logic network decodes the set of results to yield the required outputs. In fact, the network shown uses logical NAND gates to yield the complement of each desired output: $\overline{Y}_A$, $\overline{Y}_B$, $\overline{Y}_C$, $\overline{Y}_D$; so that $\overline{Y}_A$, for example, will be logical zero if A is the largest input, with $\overline{Y}_B$, $\overline{Y}_C$ and $\overline{Y}_D$ logical one. These can be inverted if necessary, although standard monostable chips frequently incorporate invertors and gates which permit the desired combination of these outputs with the peak detector outputs, to yield the monostable trigger pulses. The gates enclosed in dotted boundaries in Fig. 6.17 are redundant if $\overline{Y}_D$ is not required. This will be so when D is the detection threshold.

Such networks can be designed for any number of inputs, using the standard design methods for combinational logic. However, this is not necessary, as the network exhibits a systematic

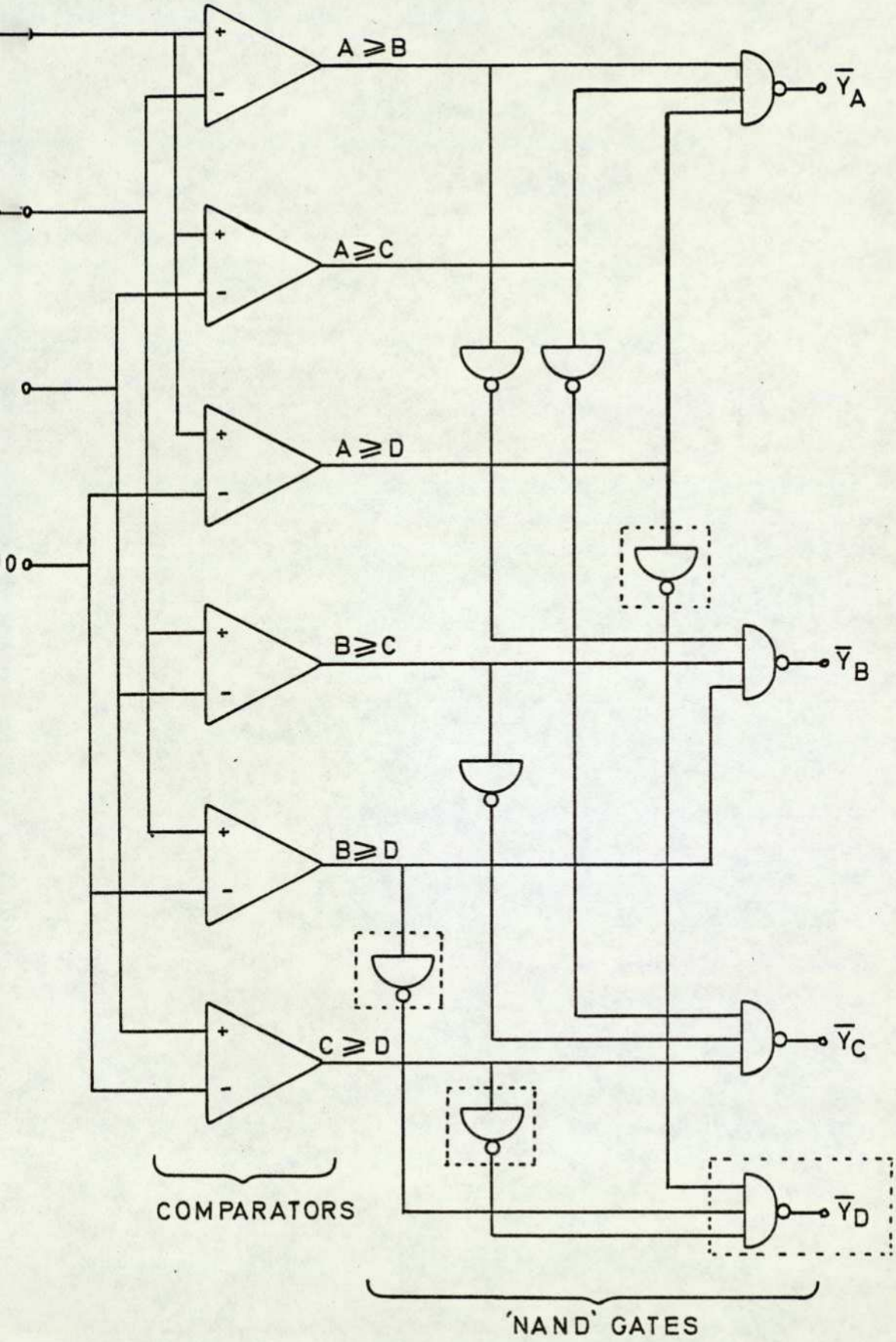


FIGURE 6.17 - LARGEST VALUE IDENTIFICATION.

structure. Consider a 5 input example:

Let A, B, C, D, E be the five analogue inputs.

Let $X_1, X_2, -, X_{10}$ be the comparator outputs, as follows:

$$X_1 = A \geq B \qquad X_6 = B \geq D$$

$$X_2 = A \geq C \qquad X_7 = B \geq E$$

$$X_3 = A \geq D \qquad X_8 = C \geq D$$

$$X_4 = A \geq E \qquad X_9 = C \geq E$$

$$X_5 = B \geq C \qquad X_{10} = D \geq E$$

The desired outputs are then defined by the following logical equations:

$$Y_A = X_1 \; X_2 \; X_3 \; X_4$$

$$Y_B = \overline{X}_1 \; X_5 \; X_6 \; X_7$$

$$Y_C = \overline{X}_2 \; \overline{X}_5 \; X_8 \; X_9$$

$$Y_D = \overline{X}_3 \; \overline{X}_6 \; \overline{X}_8 \; X_{10}$$

$$Y_E = \overline{X}_4 \; \overline{X}_7 \; \overline{X}_9 \; \overline{X}_{10}$$

The right-hand side of this set of equations forms a matrix, which divides into an upper triangle and a lower. The former contains only comparator outputs, and the latter only their negation. These run in numerical sequence along rows of the upper triangle, and down columns of the lower. These equations can therefore be written down immediately for any number of outputs.

6.3 Feature Extraction

This component must accept as input the analogue scan signal, together with the binary detection/delineation signal from the matched filter bank, and produce as output a set of feature values measured upon the analogue signal between the limits indicated by the delineation signal. In effect, the rising edge of the delineation

signal can be interpreted as a command to start the measurement processes, and the falling edge as a command to stop.

For "samples-as-features", this component is particularly simple, being merely an analogue to digital convertor, clocked at a suitable rate, and controlled by the delineation signal. It may be necessary to store the resulting samples, and even to re-convert them into analogue values, depending upon the classifier implementation. Alternatively, analogue values could be measured directly with a repetitive sample-and-hold scheme. In any case, no special difficulties arise, and so this feature set will not be discussed further.

For geometric features, we must implement four basic measurements: base width, amplitude, pseudo-perimeter and area. Base width is nothing more than the time duration of the delineation signal, and such a measurement is standard technique (e.g. charge a capacitor from a constant current source). Amplitude has been defined as the maximum signal level minus the minimum, within the limits of the delineation signal. Again, such a measurement follows standard practice, requiring only a maximum peak detector and a minimum, with their outputs feeding a simple operational amplifier subtraction circuit. Pseudo-perimeter is more difficult. It has been defined as the sum of absolute differences between successive samples, in the digital simulation. In analogue form, we have:

$$\text{Pseudo-perimeter} = \int_{t_1}^{t_2} \left| \frac{ds(t)}{dt} \right| dt$$

This requires differentiation, full-wave rectification and integration. Once again, these can be achieved with standard techniques, although the time constants will need to be carefully chosen to yield results comparable to the simulation. Finally, pulse area can be measured with a simple integrator.

The speed at which these measurements can be carried out will be governed primarily by the response time of the various operational amplifiers involved. With the shortest pulse duration of interest being 3.10^{-6} seconds, and amplifiers readily available with band-

widths of 10 MHz and higher, no difficulty should be encountered with real time operations.

The measurements are cheap enough to be implemented in parallel, and their outputs can be hardwired to the classifier. The delineation signal can then be used to inhibit classification when it is high, and we shall see that this can be done very easily. With this arrangement , the classification will become available almost immediately after the defect signal terminates.

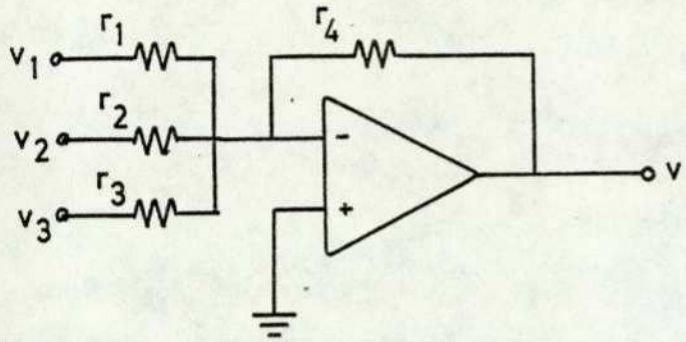
6.4 Linear Classification

This component must accept as input the measured feature values, form an appropriate linear weighted sum of these values for each defect class, and determine the largest sum for classification.

A linear weighted sum of a set of analogue voltages is most easily computed by the circuit of Fig. 6.18. We shall therefore provide one such circuit for each class (plus, perhaps, one for the reject class). The outputs can feed into a largest value block identical to that described in Section 6.2.2. To inhibit classification until the end of the delineation signal, it is a simple matter to inhibit the largest value block so that, for example, all outputs are held at logical zero when the delineation signal is at logical one.

6.5 Conclusions

This chapter has discussed the problems of implementing the various signal processing schemes developed and evaluated in simulation, at on-line speeds and at reasonable cost. Special-purpose hardware designs have been presented for defect detection and delineation, feature extraction from a defect pulse and linear classification. The designs have been analysed and discussed in sufficient detail to demonstrate their feasibility. Taken together, they constitute a complete system for defect recognition which, if supplied with signals from the SIRA scanning system, should yield results closely similar to those achieved in simulation. Furthermore,



$$-v = v_1 \frac{r_1}{r_4} + v_2 \frac{r_2}{r_4} + v_3 \frac{r_3}{r_4}$$

FIGURE 6.18 - LINEAR WEIGHTING.

the system structure is general, in that it should prove equally suitable for materials other than tinplate, requiring only appropriate detailed modification to the primary components.

7.0 SUMMARY, CONCLUSIONS, AND SUGGESTIONS FOR FURTHER WORK

The defect recognition problem is the primary impediment to further progress in automatic surface inspection. The work reported in this thesis is one step towards its solution.

The work has formed part of a team effort between the Instrument Systems Centre of The City University, the SIRA Institute, and the British Steel Corporation. It has involved the application of novel techniques of signal processing to the response signals generated by optical scanners of the kind developed by the SIRA Institute.

Signal processing has been applied, so far independently, to two distinct kinds of data available from such scanners:

(1) isolated, analogue scan sections;
(2) binary videoprint data

This thesis has been concerned exclusively with data of the first kind, and a related thesis (ref. 5) with the second.

A processing system has been developed which encompasses the following operations:

(1) Detection and delineation of the defect signals in the presence of a noise signal due to surface structure and other effects. This operation is a pre-requisite to recognition processing, and was previously unsolved (especially for large area, low contrast defects).

(2) Extraction of a set of pre-selected feature values from the signals so detected and delineated.

(3) Processing of the resulting set of feature values, using the methods of automatic pattern recognition, so as to identify the defect type.

For each of these three operations, a number of techniques were identified from the literature as potentially suitable for the surface inspection problem, with particular reference to the system requirements of fast on-line processing at reasonable cost. These techniques were extended where necessary and evaluated in computer simulation on a data set gathered from samples of sheet tinplate. With a suitable combination of techniques, acceptable performance was achieved, with

about 80% correct identification over five defect classes.

An implementation with special-purpose analogue and digital hardware has been presented, capable of operating at on-line speeds at reasonable cost. This implementation constitutes a complete recognition system for isolated scan sections, fully automatic in its operation. The general structure can be expected to be equally applicable to materials other than tinplate.

Considerable effort was devoted to the setting up of a data base for evaluating the various techniques considered. The resulting 500 signals from 5 defect classes, although substantial in comparison with many pattern recognition exercises, cannot be said to represent fully the inspection problem on tinplate. This has led to the implementation of a fully automatic data gathering system with which a task formerly requiring some three hours of continuous effort can be accomplished in less than five minutes. This will allow the data base to be enlarged considerably. In particular, some 30 different kinds of defect might need to be included.

More work needs to be devoted to deciding exactly what the inspection requirement is, and to specifying the performance required from an automatic system. The availability of automatic inspection should lead to a radical re-structuring of production and marketing processes, and this needs to be carefully considered when defining the required performance. This is a major exercise in its own right, and the work reported in this thesis should be seen as a feasibility study intended to justify such further work. To this extent, a degree of success can be claimed, since the work will now be progressed further, with the assistance of the British Steel Corporation and the European Coal and Steel Community.

Several extensions can be made to the signal processing. First, the system can be extended to encompass data from whole defects, rather than dealing with each scan in isolation. Many of the classification errors made by the proposed system can be attributed to this restriction (e.g. confusion between Five Stand Ring and Black Dots - Figures 4.12, 4.15, 4.19). This extension will require a means of associating those scan sections arising from the same defect. Second, more elaborate features can be extracted from each scan section, by

recognising that the matched filter bank provides a fairly complete analysis of each one. This is especially evident when the filter bank is viewed as a spectrum analyser. It follows that features extracted from the filter outputs, rather than from the scan section itself, should be a richer source of information. Finally, the classification techniques themselves merit further investigation. The results presented show the simple linear classifier to be as powerful on this data set as any other. This is at odds with the known limitations of this classifier, and is therefore difficult to explain. It may be a consequence of the feature selection procedure adopted, to the extent that this procedure searches for a feature subset with which inherent weaknesses of the classifier have least effect. Alternatively, as suggested in the text, it may be that theoretically powerful classifiers have so many "degrees of freedom" that an impossibly large data set is needed for their design.

REFERENCES

1 Brook, R. A. "An experimental automatic surface inspection system", Metron, vol. 4, no. 8, pp. 219-223 (Aug., 1971)

2 Bohlander, P. "Investigations of the automation of optical surface inspection of cold-rolled strip", VDEh-Inst. for Applied Research GmbH, Report no. 482 (1974)

3 Bell, D. A. "Electrical noise: fundamentals and physical mechanism", Van Nostrand (1960)

4 Brook, R. A. et al "Automatic inspection of flat strip in the steel industry", 7th IMEKO Congress, London, Paper no. AML/154/1 (1976)

5 Popovici, V. "Application of syntactic pattern recognition to defect classification in optical surface inspection", Ph.D. Thesis, The City University, London (1976)

6 Duda, R. O. and Hart, P. E. "Pattern classification and scene analysis", John Wiley (1973)

7 Meisel, W. S. "Computer-oriented approaches to pattern recognition", Academic Press (1972)

8 Arkadev, A. G. and Braverman, E. M. "Teaching computers to recognise patterns", Academic Press (1967)

9 Fu, K. S. "Sequential methods in pattern recognition and machine learning", Academic Press (1968)

10 Parzen, E. "Modern probability theory and its applications", Wiley (1960)

11 Parzen, E. "On estimation of a probability density function and mode", Ann. Math. Statist. vol. 33, pp. 1065-1076 (1962)

12 Meisel, W. S. "Potential functions in mathematical pattern recognition", IEEE Trans. Compt., vol. C-18, no. 10, pp. 911-918 (Oct., 1969)

13 Murthy, V. K. "Nonparametric estimation of multivariate densities with applications", in Multivariate Analysis, ed. P. R. Krishnaiah, Academic Press (1966)

14 Specht, D. F. "Generation of polynomial discriminant functions for pattern recognition", Ph.D. Thesis, Stanford University (1966)

15 Cover, T. M. and Hart, P. E. "Nearest Neighbour pattern classification", IEEE Trans. Inf. Th., vol. IT-13, no. 1, pp. 21-30 (Jan., 1967)

16 Nilsson, N. J. "Learning machines", McGraw-Hill (1965)

17 Wee, W. G. "Generalised inverse approach to adaptive multiclass pattern classification", IEEE Trans. Comp., vol. C-17, no. 12, pp. 1157-1164 (Dec., 1968)

18 Yau, S. S. and Chuang, P. C. "Statistical properties of linear, multicategory pattern classifiers based on the least-mean-square error criterion", IEEE Trans. Inf. Th., Sept., 1968, pp. 778-780 (correspondence)

19 Wee, W. G. "Generalised inverse approach to clustering, feature selection and classification", IEEE Trans. Inf. Th., vol. IT-17, no. 3, pp. 262-269 (May, 1971)

20 Smith, S. E. and Yau, S. S. "Linear sequential pattern classification", IEEE Trans. Inf. Th., Sept., 1972, pp. 673-678 (correspondence)

21 Ho, Y. C. and Kashyap, R. L. "A class of iterative procedures for linear inequalities", J. SIAM CONTROL, vol. 4, pp. 112-115 (1966)

22 Stearns, S. D. "On selecting features for pattern classifiers", Proc. 3rd Int. Joint Conf. on Pattern Recognition, Nov. 8-11, 1976, Coronado, California

23 Vilmansen, T. R. "Feature evaluation with measures of probabilistic dependence", IEEE Trans. Comp., vol. C-22, no. 4 (1973)

24 Toussaint, G. T. "Feature evaluation with quadratic mutual information", Information Processing Letters 1, pp. 153-156 (1972)

25 Backer, E. and Jain, A. K. "On feature ordering in practice and some finite sample effects", Proc. 3rd Int. Joint Conf. on Pattern Recognition, Nov. 8-11, 1976, Coronado, California

26 Kanal, L. N. "Patterns in pattern recognition", IEEE Trans. Inf. Th., vol. IT-20, no. 6, pp. 697-722 (Nov., 1974)

27 Logan, I. G. and Macleod, J. E. S. "An application of pattern recognition algorithms to the automatic inspection of strip metal surfaces", Proc. 2nd Int. Joint Conf. on Pattern Recognition, Aug. 13-15, 1974, Copenhagen, Denmark

28 Nordqvist, K. and Millgard, L. "A new system for surface inspection and classification", Internal Report, The Axel Johnson Inst. for Industrial Research, Sweden (1973)

29 Bendat, J. S. and Piersol, A. G. "Random data: analysis and measurement procedures", John Wiley (1971)

30 Brown, R. G. "Smoothing, forecasting and prediction of discrete time series", Prentice-Hall (1962)

31 Rabiner, L. R. and Gold, B. "Theory and application of digital signal processing", Prentice-Hall (1975)

32 Goodyear, C. C. "Signals and Information", Butterworths, London (1971)

33 Whalen, A. D. "Detection of signals in noise", Academic Press (1971)

34 Connor, F. R. "Signals", Edward Arnold, London (1972)

35 Schwartz, M. "Information transmission, modulation and noise", McGraw-Hill (1959)

36 Cherry, C. "Pulses and transients in communication circuits", Chapman & Hall, London (1949)

37 Zverev, A. I. "Handbook of Filter Synthesis", John Wiley (1967)

38 Freeman, H. "On the encoding of arbitrary geometric configurations", IRE Trans. on El. Comps., June, 1961, pp. 260-268