ensures that the column vector inputs to the next layer are of a similar size. For each utterance one column vector is generated which then acts as an input to the second layer.

To fully exploit the inherent time warp invariance property of the trajectories on the first layer it was decided not to repeat consecutively firing co-ordinates in the concatenation process. In this way, the shape of the trajectory and thus its time warp invariance property is preserved while the sequence in which the neurons fire is also maintained.

After the second layer has been trained, using the same procedure as before, different parts of the array become sensitive to different words. Inputting an utterance produces a trajectory on the first layer and causes a single neuron to fire on the second layer. By labelling a neuron to the word to which it becomes most excited, different words may be mapped to different parts of the array. Neurons, located next to each other, which are excited by similar words, may be grouped to form distinct clusters.

In Figure 2.9 an example of a labelled second layer is shown. It can be seen that the various digit sounds excite different parts of the second layer and clusters of the different sound types are clearly evident. However some neurons respond to more than one digit sound. In these cases it is the sound to which the neuron most frequently responds that the neuron is labelled with. Two clusters exist for some sounds, which indicates that global ordering has not
Figure 2.9 An example of a labelled second layer
occurred on the second layer. Global ordering will only occur if the inherent dimensionality of the input data is the same as the dimensionality to which it is mapped. However, global ordering is not necessary on the second layer for recognition purposes.
Chapter 3

Pre-processors: The Parallel Filter Bank, The Triangular Filter Bank, Linear Predictive Coding

3.1 Introduction

In this chapter, the development of two front-end processors, a parallel filter bank and a triangular filter bank is discussed. The output of both these front-ends is suitable for use with a neural network. Simulation results on the testing of these pre-processors with various inputs are presented. The inputs include sinusoids of different frequencies, impulse and step functions and some speech samples. Also included are the frequency response characteristics of some of the digital filters used in the parallel filter bank. The pascal code implementing the two front-end processors is given in Appendix B. Linear predictive coding (LPC), which is used to extract formant frequencies from speech samples, is also examined. A commercial speech processing package called "Hypersignal" was used to generate LPC outputs from the speech database. These outputs were later tested using Kohonen's two layer neural network.
3.2 Pre-processing of Speech Signals

Techniques most popularly used in the pre-processing of speech signals have, unsurprisingly, changed significantly in the last number of decades. In the 1950's and 60's use was made of features derived easily from the time domain waveform, such as amplitude and zero-crossing rate. Filterbanks using multiple channels, at the time, were impractical due to the limited processing power available. The increasing use of filterbanks in the 1960's and early 1970's can be partly attributed to the assimilation of Bekesy's experiments on frequency analysis in the ear, partly to the possibility of constructing analogue filterbanks from transistor electronics and partly to the increased processing power available to accommodate a large number of channels.

Today, there are a variety of front-end processors in use at present for the purpose of speech analysis, including Linear Predictive Coding (LPC), cochlear modelling, Mel Frequency Cepstral Coefficients (MFCC's), Fast Fourier Transforms (FFT's) and filter banks. Inherent in all of these techniques, at some point is a transformation of a time domain signal to its equivalent frequency representation. In the case of MFCC's this transformation is followed by a further one using a discrete cosine transform. In general, the end result of pre-processing is the production of an n-dimensional pattern vector which represents some particular feature that one wishes to extract from the inputted signal.
3.3 Development of the Parallel Filter Bank

A block diagram of the general structure of a parallel filter bank is shown in Figure 3.1. It consists of N separate and independent channels each of which is fed with the current speech input. The first stage in each channel is a band-pass filter (BPF) covering a certain bandwidth with a unique centre frequency. This is then followed by a non-linearity and low-pass filter section (NL & LPF). The final stage involves decimation and logarithmic encoding of the low-pass filter energy output. The resulting N-dimensional vector is suitable for later use with a neural network. In the following sections the various stages involved in the development of the parallel filter bank are explained in detail and any design decisions that were taken are justified.

Figure 3.1 Block diagram of the Parallel Filter Bank
3.3.1 Band-pass Filter Section

The parallel filter bank developed was composed of 19 separate channels producing a 19 dimensional vector every 16 ms. To this end it was necessary to develop 19 band-pass filters (BPF) that would span the frequency range 250 Hz to 4 kHz. This frequency range was chosen deliberately as it is the same as is used in most telephone systems. It was desirable that any recognition system developed could be used in an application that had telephoned speech as its input. The bandwidths and centre frequencies of these band-pass filters need to be chosen with care if one is attempting to attain a high recognition accuracy. As a result it was decided to use the centre frequencies based on the mel scale.

Essentially, the mel frequency scale is linear until 1 kHz and logarithmic thereafter (see Table 3.1). This kind of filter spacing has been shown to be more effective for recognition than a linear spacing [4].

<p>| | | | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>200</td>
<td>300</td>
<td>400</td>
<td>500</td>
<td>600</td>
<td>700</td>
<td>800</td>
<td>900</td>
<td>1000</td>
<td>1137</td>
<td></td>
</tr>
<tr>
<td>1292</td>
<td>1469</td>
<td>1671</td>
<td>1899</td>
<td>2159</td>
<td>2455</td>
<td>2791</td>
<td>3173</td>
<td>3607</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3.1 The Mel Frequency Scale given in Hz, from 100 Hz to 4 kHz

Infinite impulse response (IIR) filters were used to implement the band-pass filters. Fourth-order elliptical filters were selected
with a stop-band attenuation of 20dB. In Figure 3.2 the general canonical structure of these filters is displayed. From this diagram the transfer function of a fourth-order section can be easily calculated and is as follows:

\[
\frac{V_{out}}{X_{in}} = \frac{C.(a_{10} + a_{11}z^{-1} + a_{12}z^{-2}).(a_{20} + a_{21}z^{-1} + a_{22}z^{-2})}{(1 + b_{11}z^{-1} + b_{12}z^{-2})(1 + b_{21}z^{-1} + b_{22}z^{-2})}
\] (3.1)

Digital filter coefficients needed to be calculated that would give each band-pass filter its required frequency response. Table 3.2 summarises the characteristics of each of the 19 filters.

The filter coefficients were calculated using a "Hypersignal" software package. The package allows the design and construction of IIR bandpass, bandstop, lowpass, highpass filters using either classical methods such as Butterworth, Chebychev, Elliptical, or Bessel.

Using the software one can determine the maximum possible stop-band attenuation for a given order system once the start and end frequencies of the channel have been specified. In Figure 3.3 the different attributes of a band-pass filter which need to be specified when using the software package are displayed. The resulting digital filter coefficients are listed for reference in Appendix B.
### Table 3.2 Characteristics of the 19 Band-pass filters

<table>
<thead>
<tr>
<th>Filter No.</th>
<th>Centre Freq. (Hz)</th>
<th>Bandwidth (Hz)</th>
<th>Stopband Atten. (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>200</td>
<td>200</td>
<td>20</td>
</tr>
<tr>
<td>2</td>
<td>300</td>
<td>200</td>
<td>20</td>
</tr>
<tr>
<td>3</td>
<td>400</td>
<td>200</td>
<td>20</td>
</tr>
<tr>
<td>4</td>
<td>500</td>
<td>200</td>
<td>20</td>
</tr>
<tr>
<td>5</td>
<td>600</td>
<td>200</td>
<td>20</td>
</tr>
<tr>
<td>6</td>
<td>700</td>
<td>200</td>
<td>20</td>
</tr>
<tr>
<td>7</td>
<td>800</td>
<td>200</td>
<td>20</td>
</tr>
<tr>
<td>8</td>
<td>900</td>
<td>200</td>
<td>20</td>
</tr>
<tr>
<td>9</td>
<td>1000</td>
<td>237</td>
<td>20</td>
</tr>
<tr>
<td>10</td>
<td>1137</td>
<td>292</td>
<td>20</td>
</tr>
<tr>
<td>11</td>
<td>1292</td>
<td>332</td>
<td>20</td>
</tr>
<tr>
<td>12</td>
<td>1469</td>
<td>379</td>
<td>20</td>
</tr>
<tr>
<td>13</td>
<td>1671</td>
<td>430</td>
<td>20</td>
</tr>
<tr>
<td>14</td>
<td>1899</td>
<td>488</td>
<td>20</td>
</tr>
<tr>
<td>15</td>
<td>2159</td>
<td>556</td>
<td>20</td>
</tr>
<tr>
<td>16</td>
<td>2455</td>
<td>632</td>
<td>20</td>
</tr>
<tr>
<td>17</td>
<td>2791</td>
<td>718</td>
<td>20</td>
</tr>
<tr>
<td>18</td>
<td>3173</td>
<td>816</td>
<td>20</td>
</tr>
<tr>
<td>19</td>
<td>3607</td>
<td>827</td>
<td>20</td>
</tr>
</tbody>
</table>
Figure 3.2 Canonical description of the fourth order elliptical band-pass filter section
3.3.2 Frequency Response of the Band-pass filters

In Figure 3.4 the frequency response characteristics of the band-pass filter in channel 5 is displayed. This band-pass filter has a centre frequency at 600 Hz and a bandwidth of 200 Hz.
Figure 3.4 Frequency Response for channel 5

Figure 3.5 Frequency Response for channel 18
In Figure 3.5 the frequency response for filter number 18 is displayed, with centre frequency at 3173 Hz and a bandwidth of 816 Hz.

In Figure 3.6 the overall response for all 19 band-pass filters is displayed. It is desirable that this response should be reasonably flat.

![Overall Frequency Response of all 19 Band-pass filters](image)

**Figure 3.6 Overall Frequency Response of all 19 Band-pass filters**

### 3.3.3 Non-linearity and Low-pass Filter Section

In the parallel filter bank, the non-linearity and low-pass filter section accepts as its input the output from the previous band-pass filter section (Figure 3.1). The non-linearity has the effect of non-uniformly distributing the original band-limited signal energy over
the entire frequency spectrum. However, the signal energy at low frequencies is generally proportional to the total band-limited signal energy. Thus, when this non-linearity is followed by a low-pass filter, the output of the low-pass filter is a measure of the energy of the speech signal in the particular frequency band.

Half-wave Rectifier

\[ RC = e^{\frac{-30*2\pi}{f_s}} \] (3.2)

where \( f_s \) is the sampling frequency is 8 kHz.
The low pass filter outputs must be sampled at a rate of at least twice that of the low pass filter cut-off frequency if the Nyquist criterion is to be obeyed. In fact they were sampled after every 128 input samples, which is equivalent to every 16ms or a frequency of 62.5 Hz.

3.3.4 Logarithmic encoding and post processing

For the purposes of dynamic range compression, the sampled low-pass filter outputs or energy is encoded by a logarithmic transformation. The set of energy values, every 16 ms, constitute a 19 dimensional feature vector. These vectors are then post processed, to remove an energy offset caused by a low value of stop-band attenuation in the band-pass filters, chosen when designing the filters so that their order would not exceed four. This was achieved by subtracting the minimum entry in each vector from the other entries.

3.3.5 Mel Frequency Cepstral Coefficients (MFCC's)

Mel frequency cepstral coefficients were first defined and used by Davis [6]. Gagnoulet [7] compared MFCC's with a 12-channel filterbank and with LPC cepstral coefficients and found them to be superior on three different minimally paired vocabularies. MFCC's are generated by applying a Discrete Cosine Transform to the logged outputs of a mel frequency scale filterbank.
\[ \text{MFCCI}_i = \sum_{k=1}^{N} X_k \cos[i(k - \frac{1}{2}) \frac{\pi}{N}] \] (3.3)

where \( i = 1 \) to \( M \);

\( M = \) number of cepstral coefficients;

\( N = \) number of filters;

\( X_k = \) log magnitude output of the \( k^{th} \) filterbank channel ( \( k = 1 \) to \( N \)).

The Discrete Cosine Transform on the filterbank outputs has the effect of de-correlating the features. This can actually improve recognition performance especially in poor signal to noise conditions. The parallel filter bank program listed in pascal in Appendix B calculates mel frequency cepstral coefficients and displays them on screen for each frame.

3.3.6 Description of the Parallel Filter Bank Program

The program asks the user to enter in how many channels one wishes to implement (usually one selects 19). A particular channel may be viewed at two different points, one can choose between viewing either the output of the band-pass filter section or the output of the non-linearity and low-pass filter section. One can also select between observing the MFCC's for the current input or displaying the frequency response for the current channel. After this is done, one then decides what input is to be fed into the parallel filter bank. One can choose between a sinusoid, a step or impulse function or a speech input. If one chooses a speech input it is necessary to name what datafile holds the desired speech
utterance. Filterbank outputs are written to the hard disk after every input frame.

In Figure 3.8 a sinusoid at 1137 Hz has been fed into the parallel filter bank. The band-pass filter output of channel 10 with a centre frequency at 1137 shows the signal passing unhindered along this particular channel. If one observes the energy outputs across the 19 filters in the first window one can see a peak at the filter number whose centre frequency is closest to the frequency of the inputted sinusoid. In the adjacent channels the signal is attenuated as one moves away from the resonant frequency.

In Figure 3.9 the input is an impulse function. Displayed on this occasion is the output of the non linearity and low-pass filter section. Twelve mel frequency cepstral coefficients are calculated and may be seen in the top right window. The top left window displays the energy outputs of each channel.

In Figure 3.10 the input is a unit step function delayed by six samples. Any delay value required may be selected at the start of the program. The channel chosen this time is channel 2 and its frequency response is plotted. The output of the band-pass filter in channel two can be seen attempting to track the unit step function.

In Figure 3.11 a speech input is used corresponding to a "one" sound, spoken by a male speaker. The frame displayed in the input window is actually the sixth frame out of a total of fourteen. The
Figure 3.8 The parallel filter bank's display after inputting a sinusoid at 1137 Hz.
Figure 3.9 The parallel filter bank's display after inputting an impulse function.
Figure 3.10 A unit step function inputted to the parallel filter bank.
Figure 3.11 Sixth frame of the "one" sound inputted into the parallel filter bank
output of the non-linearity and low-pass filter section of channel 15 is plotted along with twelve mel frequency cepstral coefficients. Formant frequencies in the speech input may be seen in the energy output of the parallel filter bank.

In Figure 3.12 the time domain waveform for the word "one" is plotted in entirety. This is the same sample, whose sixth frame was used in Figure 3.11 and it consists of fourteen frames. In Figure 3.13, for comparison purposes, the output of the parallel filter bank to the same "one" sound is displayed. This consists of fourteen frames with nineteen energy values in each frame. These outputs were among those later used to train Kohonen's neural network.

3.4 Triangular Filter Bank

The triangular filter bank is similar in function to the parallel filter bank, in that it attempts to quantify the amount of energy in various fixed bandwidths in a set frequency spectrum. The manner in which this is brought about, however, is different. Implementation of the triangular filter bank begins by obtaining a Fast Fourier Transform (FFT) of the incoming time domain signal. This provides a frequency domain representation of the input signal.

Once this has been accomplished, a second stage is initiated which measures the energy in various frequency bands. This is done by selecting each FFT magnitude and deciding what frequency bands
Figure 3.12 Time domain waveform for the "one" sound
Figure 3.13 Output of the parallel filter bank for the "one" sound
the quantity is in. It may be in more than one as the bands overlap. The FFT magnitude is multiplied by a linear multiplier whose value increases the closer the FFT magnitude's frequency is to the centre frequency of the band.

All multiplied values within a frequency band are summed to provide a measure of energy in that particular frequency band (see section 3.4.2 for more details). The triangular filter bank outputs are the logarithm of these summed energy values.

3.4.1 Fast Fourier Transform

Speech samples used in this research were sampled at 8 kHz. Each 16ms frame of speech consists of 128 samples. To use the FFT routine listed as part of the triangular filter program in Appendix B, it is necessary that the number of samples in the input frame is a multiple of \(2^L\) where \(L\) is a natural number. The number of magnitude frequency outputs is then \(2^{L-1}\), and therefore in this case the output is a pattern vector with dimensionality 64 every 16ms. The Discrete Fourier Transform is given in equation 3.4

\[
Y_n = \frac{2}{M} \sum_{k=0}^{M-1} X_k \cdot e^{-j\pi nk} \quad (3.4)
\]

where \(n=1,2,3,\ldots,(M/2)\); \(k=0,1,2,\ldots,(M-1)\);

Number of time samples is \(M\);

Number of frequency samples (\(n\)) is \(M/2\);
Manipulation of this equation using a particular substitution reduces the computational effort required to calculate it considerably and the result is the Fast Fourier Transform. Once the FFT of each frame of the speech sample has been calculated it is then necessary to calculate the amount of energy in different frequency bands. This is done using triangular filters.

3.4.2 The Triangular Filters

The centre frequencies of the triangular filters are based on the mel scale in a similar fashion to the parallel filter bank. This ensures that when comparing results from both pre-processors that the comparison is more meaningful. Table 3.3 provides a summary of the characteristics of the 19 filters in the bank. Fe stands for centre frequency and bandwidth left and right are BWL and BWR respectively. The actual bandwidth is the sum of BWL and BWR.

In Figure 3.14 the triangular filters with their centre frequencies and bandwidths are displayed. Frequencies at the edge of each band are attenuated by 60 dB. Notice that the bandwidth for a channel with centre frequency $F_{c_i}$ extends from $F_{c_{i-1}}$ to $F_{c_{i+1}}$ and also that the bandwidth to the left of the centre frequency (BWL) is not always the same size as the bandwidth to the right of the centre frequency (BWR).

The linear equations which define the triangular filters may be derived as follows. Two equations are needed, one for the region
<table>
<thead>
<tr>
<th>Filter No.</th>
<th>Fc (Hz)</th>
<th>BW left</th>
<th>BW right</th>
<th>BW</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>200</td>
<td>100</td>
<td>100</td>
<td>200</td>
</tr>
<tr>
<td>2</td>
<td>300</td>
<td>100</td>
<td>100</td>
<td>200</td>
</tr>
<tr>
<td>3</td>
<td>400</td>
<td>100</td>
<td>100</td>
<td>200</td>
</tr>
<tr>
<td>4</td>
<td>500</td>
<td>100</td>
<td>100</td>
<td>200</td>
</tr>
<tr>
<td>5</td>
<td>600</td>
<td>100</td>
<td>100</td>
<td>200</td>
</tr>
<tr>
<td>6</td>
<td>700</td>
<td>100</td>
<td>100</td>
<td>200</td>
</tr>
<tr>
<td>7</td>
<td>800</td>
<td>100</td>
<td>100</td>
<td>200</td>
</tr>
<tr>
<td>8</td>
<td>900</td>
<td>100</td>
<td>100</td>
<td>200</td>
</tr>
<tr>
<td>9</td>
<td>1000</td>
<td>100</td>
<td>137</td>
<td>237</td>
</tr>
<tr>
<td>10</td>
<td>1137</td>
<td>137</td>
<td>155</td>
<td>292</td>
</tr>
<tr>
<td>11</td>
<td>1292</td>
<td>155</td>
<td>177</td>
<td>332</td>
</tr>
<tr>
<td>12</td>
<td>1469</td>
<td>177</td>
<td>202</td>
<td>379</td>
</tr>
<tr>
<td>13</td>
<td>1671</td>
<td>202</td>
<td>228</td>
<td>430</td>
</tr>
<tr>
<td>14</td>
<td>1899</td>
<td>228</td>
<td>260</td>
<td>488</td>
</tr>
<tr>
<td>15</td>
<td>2159</td>
<td>260</td>
<td>296</td>
<td>556</td>
</tr>
<tr>
<td>16</td>
<td>2455</td>
<td>296</td>
<td>336</td>
<td>632</td>
</tr>
<tr>
<td>17</td>
<td>2791</td>
<td>336</td>
<td>382</td>
<td>718</td>
</tr>
<tr>
<td>18</td>
<td>3173</td>
<td>382</td>
<td>434</td>
<td>816</td>
</tr>
<tr>
<td>19</td>
<td>3607</td>
<td>434</td>
<td>393</td>
<td>827</td>
</tr>
</tbody>
</table>

Table 3.3 Triangular filter's characteristics shown in Hertz
Figure 3.14 Individual frequency response of the triangular filters on a mel scale
from FCj-BWL to FCj and the other from FCj to FCj+BWR. Two points are chosen on the line and from these the slope and then the equation of the line is calculated (see Figure 3.14).

**For region 1.**

(x1,y1) is the point (FCj,1) \quad (x2,y2) is the point (FCj-BWL, 10^{-3})

Slope= \frac{(y1-y2)}{(x1-x2)} = \frac{(1-10^{-3})}{BWL} \quad (3.5)

The equation of a line is defined as \( y-y_1 = m(x-x_1) \). After substitution the equation for region 1 is

\[ y=1+ \frac{(1-10^{-3})*(x-Fcj)}{BWL} \quad (3.6) \]

**For region 2.**

(x1,y1) is the point (FCj,1) \quad (x2,y2) is the point (FCj+BWR, 10^{-3})

Slope= -(1-10^{-3})/BWR \quad (3.7)

Equation 2 is \[ y=1- \frac{(1-10^{-3})*(x-Fcj)}{BWR} \quad (3.8) \]

These two equations were used in a program written in pascal listed in Appendix B. To compensate for wider bands summing up a larger proportion of FFT magnitude values, some normalisation is required. The number of FFT magnitude values in every band was noted and the total summed energy output was divided by that
figure. After this was done the outputs were logarithmically encoded and stored on the hard disk.

3.4.3 Overall Frequency Response

The overall frequency response is shown in Figure 3.15.

Figure 3.15 Overall frequency response of triangular filter bank
One can see the reasoning behind using triangular bands that overlap. The overlapping sections of the triangular bands cancel each other out when summed together, thus producing a flat response across the whole frequency spectrum.

3.4.4 Example Inputs and Outputs

The triangular filter bank program listed in Appendix B is designed to read in one hundred utterances of single digit sounds, calculate
the triangular filter's output and write the results to the hard disk for later examination. In Figure 3.16 a snapshot is shown of what is displayed on screen during execution of the program. The log magnitude outputs of the fast fourier transform (FFT) is plotted versus frequency in the top window. 128 points in each incoming frame of speech sampled at 8 kHz are converted into 64 magnitude values in the frequency range 0 to 4 kHz. In the bottom window the log energy outputs of the 19 triangular filters are plotted versus filter number. The sixth frame of a "one" sound is the example used.

In Figure 3.17 the triangular filter bank's output to a complete utterance of the sound "one" is displayed. This may be compared with the parallel filter bank's output of the same input utterance in Figure 3.13. One of the advantages of filtering signals already in the frequency domain is that any desired characteristic can easily be achieved. For example, in the above derivation of the linear equations it was decided to attenuate magnitude values at the band edges by 60 dB or $10^{-3}$. This figure could have been easily altered, or indeed any of the other characteristics.

3.5 Linear Predictive Coding

Linear prediction coefficients became popular in the 1970's as a method of obtaining a good representation of a speech signal without any special hardware and less computational cost than required for a software filterbank. Many variants of LPC coefficients have been utilised for example autocorrelation,
Figure 3.16 The triangular filter bank's display when inputting the sixth frame of a "one" sound
Figure 3.17 The triangular filter bank's output for the entire "one" sound
PARCOR and reflection coefficients. LPC is normally used to extract formant frequencies as features. LPC is probably still the most widely used feature extraction technique for speech recognition.

3.5.1 Theory of Linear Predictive Coding

Linear predictive coding can be interpreted as a simplified model of part of the speech production apparatus, where the excitation is modelled as a pitch pulse for voiced sounds or "white" noise for unvoiced, and the vocal tract is approximated by a non-uniform lossless tube. The vocal tract approximation therefore acts as an all-pole spectral shaping filter with the transfer function given by:

\[ H(z) = \frac{1}{1 + \sum_{k=1}^{p} a_k z^{-k}} \]  

(3.9)

where \( p \) is the number of poles in the filter, and \( a_k \) are the filter coefficients. Speech can then be represented as a time varying all-pole shaping filter with an appropriate excitation source.

The filter coefficients \( a_k \) may be calculated in many different ways. One method called autocorrelation involves deriving filter coefficients by finding a solution to the following system of equations:

\[ \sum_{k=1}^{p} a_k R(i-k) = -R(i) \quad 1 \leq i \leq p \]  

(3.10)
where \( R(i) \) is the autocorrelation of the window of speech \( s_n \), and \( p \) is the order of the analysis. This is the approach employed by the software package Hypersignal that was used to calculate the LPC outputs.

3.5.2 Example LPC outputs from Hypersignal

The Hypersignal package allows fast, flexible linear predictive coding autocorrelation of time domain waveforms. Coefficient generation is accomplished by linear predictive analysis and output coefficients may be either reflection or analysis filter coefficients. Filter coefficients are output to a standard Hypersignal time domain file. FFT processing of these coefficients produces a highly smoothed frequency response of the original input time domain signal. An example of the frequency response of an autocorrelated LPC output of the sound "one" is displayed in Figure 3.18. For comparison the FFT of an unautocorrelated example of the same sound "one" is provided in Figure 3.19.

The autocorrelated LPC spectrum models the envelope of the FFT spectrum giving a good approximation to the vocal tract response. Autocorrelated LPC outputs were obtained using Hypersignal for all one hundred utterances in the speech database. Results as to how well these spectral patterns performed on recognition tests with the two-layer Kohonen net are documented in chapter 6.
Figure 3.18 An autocorrelated LPC output for the "one" sound using Hypersignal
Figure 3.19 FFT of an unautocorrelated "one" sound using Hypersignal
3.6 Collection of speech samples used

The speech utterances used in testing the pre-processors were all sampled at 8 kHz. Ten utterances of each of the single digit sounds (zero to nine) were recorded from one male speaker, thus creating a database of one hundred utterances. The speech sounds were manually endpointed and band-pass filtered in the range 250 Hz to 4 kHz before being applied to the front-end processors. The band-pass filter program used is listed in Appendix B.
The Cochlear Model

4.1 An Introduction to the Anatomy of the Cochlea

The ear is divided into three sections: the outer, middle and inner ear (Figure 4.1(a)). The outer ear is terminated by the eardrum (tympanic membrane). Sound waves entering the auditory canal of the outer ear are directed onto the eardrum and cause it to vibrate. These vibrations are transmitted by the middle ear, an air filled section comprising a system of three tiny bones, the malleus, incus and stapes, to the cochlea (the inner ear).

The cochlea is a spiral of about $2 \frac{3}{4}$ turns which unrolled would be about 3.5 cm long. The cochlea consists of three fluid-filled sections (Figure 4.1(b)). One, the cochlear duct, is relatively small in cross-sectional area, the other two, the scala vestibuli and the scala tympani are larger and roughly equal in area. The scala vestibuli is connected to the stapes via the oval window (Figure 4.1(c)). The scala tympani terminates in the round window which is a thin membranous cover allowing the free movement of the cochlear fluid.
Schematic diagram of the parts of the ear

Cross section of the cochlea (Zwislocki, 1981)

A longitudinal section of an uncoiled cochlea

Figure 4.1 The Anatomy of the Cochlea
4.2 The Basilar Membrane

Running the full length of the cochlea is the basilar membrane (BM) which separates the cochlea duct from the scala tympani. The Reissner membrane separates the cochlear duct from the scala vestibuli. When the vibrations of the eardrum are transmitted by the middle ear into movement of the stapes, the resulting pressure within the cochlea fluid generates a travelling wave of displacement on the basilar membrane. The location of the maximum amplitude of this travelling wave along the basilar membrane varies with the frequency of the eardrum vibrations.

By this mechanism, the basilar membrane transforms incoming time domain signals into their constituent frequency components. Each frequency within the auditory spectrum causes a maximum displacement to occur at a particular point on the basilar membrane.

This frequency dependent characteristic of the basilar membrane is most easily represented by treating the membrane as a linear transmission line model, whose individual sections are constructed to accurately reflect the known characteristics of the point of the membrane which they represent [2]. This in turn can be implemented using a cascade of digital filters with each filter tuned to a different frequency. After sound has been converted to basilar membrane displacement, it then undergoes a further transduction process, which converts the membrane displacement into electrical energy in the inner hair cells (Figure 4.2).
After this process, the electrical energy is converted into neural pulses by structures called primary neurons. These neural pulses carry information along the auditory nerve to higher processing centres in the brain.

4.3 The Physiologically-based Cochlear Model

The basic model used in this research is the transmission line model developed by Ambikairajah et al [2], in which the basilar membrane is modelled as a cascade of 128 second-order digital filters. Each filter has a different resonant frequency in the
auditory spectrum (Figure 4.2; the part enclosed within the dotted rectangle is the digital filter model of the basilar membrane). Each filter represents one section of the basilar membrane. A stimulus, representing sound pressure, is applied at the input of the transmission line as a travelling wave, corresponding to the pressure in the cochlear fluid. At each section of the transmission line, the pressure is converted to mechanical displacement of the basilar membrane. Each frequency within the auditory spectrum causes a maximum displacement to occur at a different section of the transmission line. The pressure transfer function in the z-domain, for a single section is given by:

$$\frac{V_o(z)}{V_i(z)} = \frac{K(1-a_0)(1-b_1+b_2)(1-a_1z^{-1}+a_2z^{-2})}{(1-a_0z^{-1})(1-b_1z^{-1}+b_2z^{-2})(1-a_1+a_2)}$$  \hspace{1cm} (4.1)$$

where $V_i$ is the pressure input to the section, $V_o$ is the pressure output from the section, $a_0$, $a_1$, $a_2$, $b_1$ and $b_2$ are the digital filter coefficients and $K$ is a gain factor.

The membrane displacement transfer function is given by:

$$\frac{V_m(z)}{V_i(z)} = \frac{K(1-a_0)(1-b_1+b_2)z^{-1}}{(1-a_0z^{-1})(1-b_1z^{-1}+b_2z^{-2})}.$$  \hspace{1cm} (4.2)$$

where $V_m$ is the membrane displacement.

Thus, the model performs the same frequency transformation as the cochlea itself, i.e. different sections of the model respond to the
different constituent frequencies of a stimulus. In this manner, the sound wave is resolved into its frequency components, first by conversion into pressure in the fluid of the cochlea, and thence into mechanical displacement of the basilar membrane. In Figure 4.3 a general filter section of the cochlear model is shown. One can observe from the diagram how the different transfer functions were calculated. The complete cochlear model is comprised of 128 of these filter sections. Each filter section’s characteristics depend on the digital filter coefficients used in that section.

4.4 Use of the Cochlear Model in this research

The original cochlear model developed by Ambikairajah et al. [2] operates at 48 kHz. It was this model that was used exclusively during development and testing of the two-layer Kohonen neural network. Later on in the research, an 8 kHz cochlear model was made available and it was the 8 kHz model that was used for the transputer implementation of one of the speech recognition systems. Both models consist of 128 digital filters, but the 48 kHz model spans the auditory spectrum range 0 Hz to 16 kHz while the 8 kHz model, due to the Nyquist criterion constraint, spans the frequency range 0 Hz to 4 kHz.

The frequency bandwidth of interest in this investigation was that of telephoned speech i.e. 250 Hz to 4 kHz. This meant that not all 128 digital filter outputs were required. For the 48 kHz model 60 digital filter outputs were sufficient to span the telephone bandwidth of speech. In the case of the 8 kHz model it was
Figure 4.3 Digital filter realization of a basilar membrane section

\[ G_0 = 1 - a_0 \]
\[ G_p = 1 - b_1 + b_2 \]
\[ G_z = \frac{1}{1 - a_1 + a_2} \]
necessary to use 90 digital filter outputs to cover the same bandwidth.

4.5 The 48 kHz Cochlear Model

In Appendix B, a pascal program implementing the 48 kHz cochlear model may be found. The speech samples used are the same as were used on the other pre-processors mentioned in chapter 3. Ten utterances of the single digit sounds "zero" to "nine" were collected from a male speaker. These word sounds were sampled at 8 kHz and were then pre-emphasised. Pre-emphasis helps boost the high frequency components of the signal and as a result preserves high formant frequencies which would otherwise be lost. This in turn improves recognition. A detailed account of pre-emphasis is provided in section 4.5.1.

After pre-emphasis, the word sounds which were sampled at 8 kHz need to be interpolated to 48 kHz before being applied to the cochlear model. An interpolation filter is used to achieve this and the process is described in section 4.5.2. In Figure 4.4 there is a block diagram of the principal features of the 48 kHz cochlear model. The basilar membrane is represented by a cascade of 128 digital filters. Out of these 128 filters, only 60 fall within the speech bandwidth of 250 Hz to 4 kHz. These are filters 47 to 108 (see Figure 4.4). Outputs from these filters are tapped off and sent through a model of the inner hair cells, which are located in the organ of the corti.
More information on the inner hair cells is provided in section 4.5.3. The outputs of the inner hair cells of the 60 filters are sampled every 16 ms. Some post-processing is done on these 60-dimensional output vectors using a smoothing algorithm. The algorithm used is described in section 4.5.4. After smoothing the 60-d pattern vectors are ready to be used with the neural network.

4.5.1 Pre-emphasis

The transfer function that performs pre-emphasis is as follows;

\[ H(z) = 1 - 0.95 \, z^{-1} \]  \hspace{1cm} (4.3)
Pre-emphasis is similar to differentiation, if one substitutes
\[ z = e^{sT} \text{ or } z = e^{j\omega T} = e^{j\theta} \]
where \( T \) is the sampling period
into the above equation to obtain a frequency response, the transfer
function will look like

\[ H(\theta) = 1 - 0.95 e^{-j\theta} \]  \hspace{1cm} (4.4)

The magnitude frequency response is

\[ H(\theta) = \sqrt{1.9025 - 1.8\cos\theta} \]  \hspace{1cm} (4.5)

In Figure 4.5 the magnitude frequency response for pre-emphasis is
plotted from \( \theta \) equal to zero to \( \theta \) equal to \( \pi \). From the diagram the
high-pass effect is obvious. At \( \theta = 0 \) the magnitude of \( H(\theta) = 0.32 \) and
at \( \theta = \pi \) the magnitude of \( H(\theta) \) is 1.92. One can easily see at low
values of \( \theta \) ie. low frequencies the output is attenuated while at
frequencies close to the sampling frequency \( \theta = \pi \) the output is
amplified.

4.5.2 Interpolation

The cochlear model operates at 48 kHz. However, as previously
stated the speech data was collected at 8 kHz. As a result the
speech samples must be interpolated, using an interpolation filter,
in order to increase the sampling rate by a factor of 6 (Figure 4.6).

The interpolation filter used was a low-pass FIR filter which has
almost the same frequency characteristics as an anti-aliasing
Figure 4.5 Magnitude frequency response for pre-emphasis
filter used by British Telecom in their telephone exchanges. The specifications are given below:

Filter type: Low-pass
Passband ripple < -0.1 dB
Gain at 3.4 kHz > -0.9 dB
Gain at 4.6 kHz < -30.0 dB

A filter-design software package was used to design the FIR filter to meet the above specifications. It was found that a 59th order FIR filter satisfied the requirements. The impulse coefficients required to implement the FIR filter are given in Appendix B.

The speech samples \( x(n) \) were derived by sampling \( x(t) \) at a sampling rate of 8 kHz (Figure 4.6). In order to obtain speech
samples which would approximate as closely as possible samples which would be obtained by sampling $x(t)$ at a rate of 48 kHz, it was necessary to insert an additional five samples between any two input samples $x(n)$ and $x(n-1)$. This interpolation process was done as follows.

The sequence $v(n)$ (Figure 4.6) at the input to the digital low-pass filter was obtained by simply inserting five zero's between any two successive samples $x(n)$ and $x(n-1)$. This was necessary, since the FIR filter was designed at a sampling rate of 48 kHz. The output $y(n)$ of the FIR filter were the interpolated speech samples.

4.5.3 Inner Hair Cells

The mechanical displacement to electrical energy transduction process takes place in structures termed inner hair cells. The inner hair cells have an innate potential difference, or receptor potential, across them. Bending of the inner hair cells cilia due to basilar membrane displacement produces a change in the overall resistance of the cell, thus modulating current flow through the hair cell, the modulation being directly proportional to the degree of bending of the cilia. This was originally proposed by Davis [5]. It is thought that modulation of current flow through the hair cell is produced by bending of the cilia in one direction only; in effect a half-wave rectification of the basilar membrane displacement takes place.

The model of the inner hair cell used in this research is a simplified version of the Davis model, in which the voltage input
corresponds to the spatially differentiated membrane displacement output of the auditory model. Spatial differentiation of the membrane displacement represents coupling (high-pass filter effect) between the cilia of the hair cells through the fluid in the subtectorial space. The remainder of the model consists of a half-wave rectifier and a low pass RC filter as shown.

![Diagram](image)

**Figure 4.7** Modified version of the Davis model for the inner hair cell

The digital implementation of the hair cell model involves a decision-making process to represent the half-wave rectifier, and a digital equation to represent the low-pass filter. A value for RC was chosen to accurately reflect the frequency dependent characteristics of the inner hair cells.
If the membrane displacement of each filter in the 128-filter auditory model is given as
\[ d(1), d(2), d(3), d(4), \ldots \ldots \ldots d(i), d(i+1), \ldots \ldots d(128) \]
then the spatially differentiated membrane displacement, \( s(i) \) of the \( i \)th section will be
\[ s(i) = \frac{[d(i+1)-d(i)]}{\Delta x_i} \]  
(4.6)
where \( x_i \) is the width of the \( i \)th section.

The cut-off frequency of the low-pass filter model is taken as 30 Hz. The digital filter equation of the inner hair cell model is given by
\[ v(n) = (1-C_0) \cdot s(n) + C_0 \cdot v(n-1) \]  
(4.7)
(i.e., charge the capacitor)

if \( s(n) > 0 \) (half-wave rectifier) and \( s(n) > v(n) \), otherwise,
\[ v(n) = C_0 \cdot v(n-1) \]  
(4.8)
(i.e., discharge the capacitor)

where \( C_0 = \exp(-30.0 \cdot 2\pi / F_s) \)
and \( F_s \) is the sampling frequency of the cochlear model.
(For the 48 kHz Cochlear model \( F_s=48000 \))
At the end of every 16 ms (62.5 Hz) frame the inner hair cell outputs are converted into dBs before being sent for smoothing.

4.5.4 A Smoothing Algorithm

Pattern vectors emerging from the cochlear model were filtered using a smoothing algorithm so that the formant content of the speech signal could be extracted. This was done as follows. If the energy outputs of each filter in the 128-filter model is given as

\[ X(1), X(2), X(3), X(4), \ldots, X(i), X(i+1), \ldots, X(128) \]

then the smoothed outputs are calculated by applying equation 4.9 to the outputs several times depending on the amount of smoothing required.

\[ [Y_{out}]_i = 0.25X_{i-1} + 0.5X_i + 0.25X_{i+1} \quad (4.9) \]

Filter outputs were first smoothed first in a forward direction (i.e. from \( i=1 \) to \( i=128 \)) and then in the reverse direction (i.e. from \( i=128 \) to \( i=1 \)). This helped prevent small shifts of frequency characteristics that would have otherwise occurred upon smoothing in one direction only. It was found that four applications of the smoothing equation (forward and reverse smoothing counting as one time) were sufficient to extract the formant frequencies. After smoothing the pattern vectors were scaled and converted to integers before being used as inputs to the neural network.
4.6 The 8 kHz Cochlear Model

The 8 kHz cochlear model operates in principal in the same way as the 48 kHz model. The frequency range covered is different and as a result the filter coefficients need to be recalculated for the new specifications. To prevent aliasing the highest allowable frequency is theoretically of course 4 kHz. In practise the centre frequencies of the 128 filters in the model range from 74.357 Hz to 3474.891 Hz. Centre frequencies and filter coefficients for all 128 filters are listed in Appendix B. Outputs from filters 1 to 90 were tapped off to provide input patterns in the telephone bandwidth range for the neural network. For the 8 kHz model pre-emphasis, the inner hair cell model and smoothing were all carried in the same manner as for the 48 kHz model but of course the interpolation stage was not required. The 8 kHz model was used exclusively for the transputer implementation and the Occam source code executing it may be found in Appendix E.

4.7 Simulation Results with the 48 kHz Cochlear Model

In Figure 4.8 the smoothed output of the cochlear model to the sound "one" is displayed. The output consists of 14 frames of 60-dimensional pattern vectors. This is the same input as was used to demonstrate the other pre-processors in chapter 3. One can observe the differences between the pre-processing techniques used by comparing Figure 4.8 with Figures 3.13, 3.17 and 3.18 in the previous chapter. Cochlear outputs, from the 48 kHz model, for the one hundred utterances in the speech database were calculated and
Figure 4.8 The smoothed output of the cochlear model for the one sound.
recognition results of how well the Kohonen neural network performed using them are presented in chapter 6.
CHAPTER 5

Cluster Analysis of the Cochlear Model Outputs

5.1 Introduction

Clustering algorithms can provide valuable information when analysing high-dimensional data vectors in any context. The process of speech recognition normally commences with a transformation of a time domain signal into an L-dimensional frequency representation. The ability of clustering algorithms to map these L-dimensional vectors to a lower dimensional representation (usually a 2-d representation) enables one to evaluate the effectiveness of different stages in the recognition process. Modifications can then be made to different stages enabling the overall performance of the system to be improved.

There are many clustering algorithms suitable for examining high-dimensional data. In this research two such clustering techniques were used, allowing a comparison to be made between them. The first is a non-linear mapping algorithm developed by Sammon [24]. This algorithm maps L-dimensional data to a lower dimensional
representation while preserving the Euclidean distance between the data vectors.

The second technique is called the K-means clustering algorithm or "nearest neighbour" method. This algorithm attempts to group together L-dimensional vectors that are close in the Euclidean sense in L-space. The final number of clusters is a parameter decided on by the user. It is possible to verify graphically the two-dimensional mappings made by Sammon's algorithm by examining vectors that are grouped together using the K-means approach and checking whether they correlate with the graphical output from Sammon's algorithm.

In this chapter the steps necessary to implement the above mentioned clustering algorithms are provided. Results produced by the algorithms using cochlear model outputs from the speech database are presented in graphical form. Also discussed are two time warping techniques, one linear the other dynamic, which were used on the cochlear pattern vectors before they had been mapped to a lower dimensional space with Sammon's algorithm. These time warping techniques are used to compensate for different articulation rates evident in the speaking of similar word sounds. Some background information on these techniques together with the necessary steps for software implementation is provided. The actual software executing the above functions was written in Pascal and may be found in Appendix C.
5.2 Sammon's Non-linear Mapping

Sammon's non-linear mapping algorithm is based on a point mapping of $N$ $L$-dimensional vectors to a lower dimensional space such that the inherent structure of the data is approximately preserved under the mapping. The approximate structure preservation is maintained by fitting $N$ points in the lower dimensional space such that their interpoint distances approximate the corresponding interpoint distances in the $L$-space. The lower dimension mapped to is normally two or three dimensions. This ensures easy evaluation of results by human observation. In this research mappings were made to two-dimensional space only.

5.2.1 Implementation of Sammon's algorithm

Let there be $N$ vectors in $L$-dimensional space and let us call them $X_i$, $i=1$ to $N$. Similarly let us define $N$ vectors in $d$-dimensional space ($d=2$ or $3$) called $Y_i$, $i=1$ to $N$. Let the distance between the vectors $X_i$ and $X_j$ in the $L$-dimensional space be defined by $d_{ij}^* =$dist $[X_i,X_j]$ and the distance in the $d$-dimensional space be defined $d_{ij} =$dist $[Y_i,Y_j]$.

$$d_{ij}(m) = \sqrt{\sum_{k=1}^{d} [y_{ik}(m) - y_{jk}(m)]^2}$$

(5.1)
It is necessary to choose an initial random d-dimensional space configuration for the Y vectors and this may be done as follows.

\[
Y_1 = \begin{bmatrix} y_{11} \\ \vdots \\ y_{1d} \end{bmatrix} \quad Y_2 = \begin{bmatrix} y_{21} \\ \vdots \\ y_{2d} \end{bmatrix} \quad Y_N = \begin{bmatrix} y_{N1} \\ \vdots \\ y_{Nd} \end{bmatrix}
\]  

(5.2)

One now computes all the d-dimensional space interpoint distances \( d_{ij} \), which are used to determine an error \( E \), which is representative of how well the present configuration of \( N \) points in d-dimensional space fits the \( N \) points in the \( L \)-dimensional space. The error at the \( m \)th iteration is defined as

\[
E(m) = \frac{1}{c} \sum_{i \neq j}^N \left[ \frac{d_{ij}^* - d_{ij}(m)}{d_{ij}} \right]^2
\]

(5.3)

where

\[
c = \sum_{i \neq j}^N [d_{ij}^*]
\]

(5.4)

In order to reduce the error it is necessary to adjust the \( y_{pq} \) variables (where \( p = 1 \) to \( N \) and \( q = 1 \) to \( d \)) so that the interpoint distance matrix for the d-dimensional space matches more closely its corresponding interpoint distance matrix in \( L \)-dimensional space.
A steepest gradient descent procedure is used to search for a minimum of the error. The new d-dimensional space configuration at time $m+1$ is given by

$$y_{pq}(m+1) = y_{pq}(m) - (M,F) \cdot \Delta_{pq}(m)$$

(5.5)

where

$$\Delta_{pq}(m) = \frac{\partial E(m)}{\partial y_{pq}(m)} \left/ \left( \frac{\partial^2 E(m)}{\partial y_{pq}^2(m)} \right) \right.$$

(5.6)

The "magic factor" (M.F.) was determined empirically by Sammon to be in the interval 0.3 to 0.4. The partial derivatives are given by

$$\frac{\partial E}{\partial y_{pq}} = -\frac{2}{c} \sum_{j \neq p}^{N} \left[ \frac{d_{pj}^* - d_{pj}}{d_{pj}^* d_{pj}} \right] \cdot [y_{pq} - y_{jq}]$$

(5.7)

$$\frac{\partial^2 E}{\partial y_{pq}^2} = \frac{2}{c} \sum_{j \neq p}^{N} \frac{1}{d_{pj}^* d_{pj}} \cdot \left[ (d_{pj}^* - d_{pj}) \cdot \frac{(y_{pq} - y_{jq})^2}{d_{pj}^*} \cdot \left(1 + \frac{d_{pj}^* - d_{pj}}{d_{pj}} \right) \right]$$

(5.8)

In order to ensure that the partial derivatives do not "explode" one should take precautions to prevent any two points in the d-dimensional space from becoming identical.
5.2.2 Linear Time Warping

When the same word is spoken more than once by the same speaker, it is found that different events within the utterances are seldom synchronised in time. This fluctuation is caused by variances in the speaking or articulation rate. Linear time warping eliminates the time difference between different speech pattern contours by compressing or stretching the pattern contours to a pre-computed average time length. In Figure 5.1 one can observe the effect that linear time warping has on a contour. In Figure 5.1(a) an unwarped contour is shown. In Figure 5.1(b) the contour has been linearly stretched to a length \( N_3 \), while in Figure 5.1(c) the contour has been linearly compressed. In this research, the average time length was chosen to be 30 frames sampled at 8 kHz. The duration of word lengths in the speech database varied between 14 and 50 frames.

Let \( L_a \) = average time length
\[ L_p = \text{unwarped time length} \]
\[ P(n) = \text{original unwarped contour} \]
\[ P'(n) = \text{linearly warped contour} \]

The time warping ratio \((w)\) is defined as
\[ w = \frac{L_p}{L_a} \] (5.9)
Figure 5.1 The effect of linear time warping

When \( w > 1 \) the \( P(n) \) contour is linearly stretched and when \( w < 1 \) the \( P(n) \) contour is linearly compressed. The equation which performs the linear time warping may be derived as follows:
The overall cost function $D$ is equal to the sum of the total amount of warp necessary to bring the input pattern into as close alignment as is possible with the reference pattern. The Euclidean distance is typically the measure used to calculate the cost at a given instant.

$$d[w(k)] = (a_{i(k)} - b_{j(k)})^2 \quad (5.12)$$

where $a$ is the input utterance and $b$ the reference utterance. There are $M$ frames in the input pattern and $N$ frames in the reference pattern.

The warp function attempts to minimise the overall cost function

$$D(w) = \sum_{k=1}^{K} d[w(k)] \quad (5.13)$$

subject to the following constraints

1. The function must be monotonic $i(k) \geq i(k-1)$ and $j(k) \geq j(k-1)$.
2. The function must match the end points of $a$ and $b$
   \[ i(1) = k(1) = 1 = j(1) \]
   \[ i(k) = M \quad j(k) = N \]
3. The function must not skip any points
   \[ i(k) - i(k-1) \leq 1 \text{ and } j(k) - j(k-1) \leq 1 \]
4. There is a global limit $Q$ on the maximum amount of warp allowable
   \[ \text{abs}(i(k) - j(k)) < Q \]
Computing the warp function $w(k)$ may be viewed as the process of finding a minimum cost path through the lattice of points $(1,1)$ to $(M,N)$ where cost is a function of the discrepancy between two points from each contour (see Figure 5.2).

![Figure 5.2 Dynamic time warping process](image)

Dynamic time warping algorithms are normally implemented using dynamic programming techniques. Dynamic programming states that the best path from $(1,1)$ to any given point is independent of what happens beyond that point. Hence the total cost of $[i(k),j(k)]$ is the cost of that point itself plus the cost of the cheapest path to it.

$$D(w_k) = d[w(k)] + \min_{\text{legal } w(k-1)} [D(w_{k-1})]$$  \hspace{1cm} (5.14)
where legal $w(k-1)$ means the minimum over all permissible predecessors of $w(k)$.

Each stage corresponds to a point on the $i$ axis and its corresponding column (i.e. $j$ value) in the lattice. For each value of $i$ it considers all possible points along $j$ axis. In each iteration there is a set of points to be considered for each point and a number of predecessors whose least cost path is already known. The best predecessor for each of the new possible points is found and then saved.

The total cost of the optimum path can be used directly as a distance measure between the input and the reference pattern. When comparing this with the total cost of other utterances of different frame lengths its important to normalise the measure in order to make a fair comparison.

$$\text{Average distance} = \frac{\text{Total Optimal Cost}}{\sum_{k=1}^{K} w(k)} \quad (5.15)$$

The average distance was the distance measure used to generate the distance matrix used in Sammon's algorithm.