# Signal Processing in Electronic Communications

**MICHAEL J. CHAPMAN**
**DAVID P. GOODALL**
**and NIGEL C. STEELE**
**School of Mathematics and Information Sciences**
**University of Coventry**



**Horwood Publishing**
**Chichester**

LGE EXHIBIT 1013
*Amazon v. Jawbone*
U.S. Patent 7,246,058
**Amazon Ex. 1013**

First published in 1997 by
**HORWOOD PUBLISHING LIMITED**
International Publishers
Coll House, Westergate, Chichester, West Sussex, PO20 6QL
England

**COPYRIGHT NOTICE**
All Rights Reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the permission of Horwood Publishing Limited, Coll House, Westergate, Chichester, West Sussex, PO20 6QL, England

© Michael J. Chapman, David P. Goodall, and Nigel C. Steele, 1997

**British Library Cataloguing in Publication Data**
A catalogue record of this book is available from the British Library

**ISBN 1-898563-23-3**

Printed in Great Britain by Hartnolls, Bodmin, Cornwall

# 9

# Aspects of speech processing

## 9.1 INTRODUCTION

There are many aspects of speech processing such as, for example: speech production, modelling of speech, speech analysis, speech recognition, speech enhancement, synthesis of speech, etc. All these aspects have numerous applications, including speech coding for communications, speech recognition systems in the robotic industry or world of finance (to name but a few), and speech synthesis, the technology for which has been incorporated in a number of modern educational toys for example. In this chapter, we introduce some models for the processing of speech signals, discuss some analysis techniques and consider some specific applications of speech processing.

## 9.2 SPEECH PRODUCTION MODEL

In this section, an appropriate system function is suggested for modelling the **vocal** and **nasal tracts** as shown in Figure 9.1 (for more details, see Deller *et al* [5] and Rabiner and Schafer [27]). The vocal tract produces a sound when the lungs contract, forcing air up through an air-passage known as the **trachea**. The increase in air pressure causes the **glottis** (a type of valve) to burst open, momentarily, before quickly closing. When this is repeatedly performed, a very nearly periodic waveform is produced. The resulting flow of air is then perturbed by the vocal tract (or vocal and nasal tracts) to produce a sound. Some factors that can affect the waveform are: tension in the vocal chords, lung pressure, and the shape of the vocal cavities. The mouth, nose and throat are the primary resonating cavities. Their resonant characteristics can be changed (thereby altering the speech waveform) by moving the lips, tongue, jaw, and soft palate at the back of the mouth.

Speech sounds can be classified into a number of distinct classes. Three important classes are:

(a) **voiced sounds**: produced when quasi-periodic (that is approximately periodic) pulses of air excite the vocal tract (for example 'u', 'i', 'd', 'w', etc.);

(b) **unvoiced sounds** (or **fricatives**): generated by forming a constriction in the vocal tract and forcing air through the constriction. This creates a noise source which excites the vocal tract (for example 'sh', 'f', 's');

(c) **plosive sounds**: obtained by making a complete closure of the vocal tract (with lips), building up pressure behind the closure, and abruptly releasing it (for example 'p', 'b').

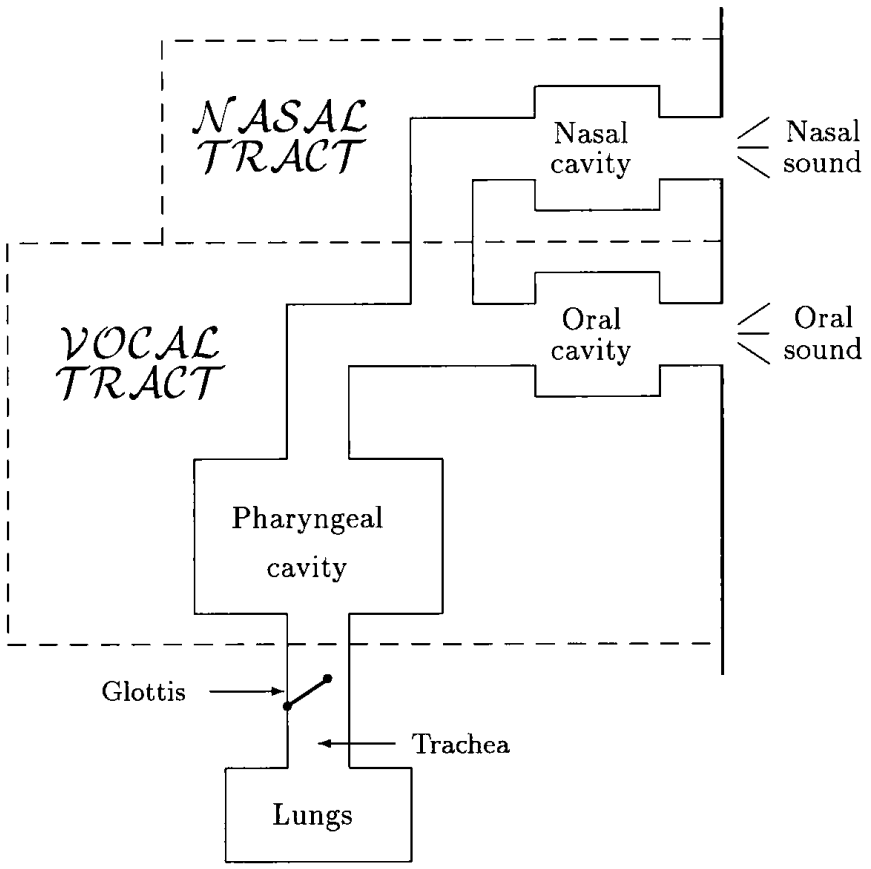



Figure 9.1: Schematic diagram of the vocal and nasal tracts.

Resonances in the vocal tract are characterized by resonant frequencies in the vocal tract spectrum, which are called **formant** frequencies (or simply **formants**). Tension of the vocal chords can affect **pitch**, that is an increase in tension causes the chords to vibrate at a higher frequency. The time between successive openings of the glottis valve, producing the quasi-periodic pulses of air, is called the **fundamental period**. By convention, see Deller et al [5] for more details, the fundamental period is often referred to as the **pitch period**. *Estimation of pitch periods and formants are two very important problems in speech processing.*

In order to analyse a speech (continuous-time) signal, the signal is sampled to obtain a discrete-time signal, $\{s(m)\}$. It is assumed that speech production can be modelled by a linear time-varying filter that is excited by an input, $\{u(m)\}$, consisting of a quasi-periodic train of impulses, in the case of voiced sounds, or

a random noise source, in the case of unvoiced sounds. Usually during excitation, vocal and nasal tract properties of a speech signal change relatively slowly with time and, for most speech signals, we may assume that these properties remain fixed over short time periods, such as 10-20 msec. Thus, in these short time periods, speech production is assumed to be modelled by *time-invariant* filters. A linear time-invariant model for speech production is usually taken to be of the form:

$$S(z) = G(z)V(z)L(z)U(z) \ ,$$

where $S(z) = \mathcal{Z}[\{s(m)\}]$, $U(z) = \mathcal{Z}[\{u(m)\}]$, $G(z)$ is the **glottal shaping model**, $V(z)$ a **vocal tract model**, and $L(z)$ a **lip radiation model**. Experimentally, it is found that a reasonable approximation to the speech production system function has the form:

$$H(z) = \frac{S(z)}{U(z)} = G(z)V(z)L(z) \approx \frac{G_s \left[1 - \sum_{i=1}^{q} b_i z^{-i}\right]}{\left[1 - \sum_{k=1}^{p} a_k z^{-k}\right]} \ ,$$

where the constant $G_s$ is known as the **speech gain parameter**. Therefore, the model

$$S(z) = H(z)U(z) = \frac{G_s \left[1 - \sum_{i=1}^{q} b_i z^{-i}\right]}{\left[1 - \sum_{k=1}^{p} a_k z^{-k}\right]} U(z)$$

is often used for the synthesis of speech. For non-nasal, voiced sounds an **all-pole** (namely **autoregressive**) filter, that is $b_j = 0$ for all $j$, is a good approximation for the system function. Thus, a further simplification of the speech production model (for non-nasal, voiced sounds) is:

$$S(z) = G_s \frac{U(z)}{A(z)} \ ,$$

where

$$A(z) \stackrel{\text{def}}{=} 1 - \sum_{k=1}^{p} a_k z^{-k}$$

and, in the literature, is known as the **inverse filter**.

## 9.3 CORRELATION FUNCTIONS

Correlation functions give a measure of 'correlation' between two signals, namely the dependence of the present value of a signal on the values of a second signal. Such correlation functions can be defined for both deterministic and stochastic

(that is random) signals. Here, we consider only deterministic signals. Given two continuous-time signals $f(t)$ and $g(t)$ the correlation function is defined as

$$R_{fg}(\tau) \stackrel{\text{def}}{=} \lim_{T\to\infty} \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} f(t)g(t+\tau)\,\mathrm{d}t \ , \tag{9.1}$$

and is a function of the time shift between two signals. If $f$ and $g$ have finite energy (see Chapter 3, §3.6), $R_{fg}(\tau)$ is zero and, in this case, (9.1) is not very useful. To overcome this problem, an alternative definition is given for the correlation function of two finite energy signals $f$ and $g$ as

$$R_{fg}(\tau) \stackrel{\text{def}}{=} \int_{-\infty}^{\infty} f(t)g(t+\tau)\,\mathrm{d}t \ . \tag{9.2}$$

Therefore, *in determining the correlation function of two signals,* (9.1) *is used when the signals have finite, non-zero average power, whilst for signals with finite energy, the correlation function is evaluated using* (9.2). If the two signals are sampled, the time shift, $\tau$, is an integer multiple of the sampling period and, thus, the correlation function becomes a discrete (or 'sampled') sequence. The concepts of (total) energy and average power, introduced in Chapter 3, can easily be extended to discrete-time signals, $\{x(m)\}$, as

$$\boldsymbol{E} = \sum_{m=-\infty}^{\infty} |x(m)|^2 \tag{9.3}$$

$$\boldsymbol{P}_{\text{av}} = \lim_{M\to\infty} \frac{1}{2M+1} \sum_{m=-M}^{M} |x(m)|^2 \ . \tag{9.4}$$

For discrete-time signals $\{x(m)\}$ and $\{y(m)\}$ the correlation function is defined by

(*i*) for signals with finite, non-zero average power,

$$R_{xy}(k) \stackrel{\text{def}}{=} \lim_{M\to\infty} \frac{1}{2M+1} \sum_{m=-M}^{M} x(m)y(m+k) \ ; \tag{9.5}$$

(*ii*) for signals with finite energy,

$$R_{xy}(k) \stackrel{\text{def}}{=} \sum_{m=-\infty}^{\infty} x(m)y(m+k) \ . \tag{9.6}$$

Only discrete-time signals are considered in the remaining part of this chapter.

In the case when $x$ and $y$ are different signals, $R_{xy}(k)$ (as defined in (9.5) and (9.6)) is known as the **cross-correlation** function, whilst, when the signals are the same,

(*i*) for signals with finite, non-zero average power,

$$R_{xx}(k) \stackrel{\text{def}}{=} \lim_{M\to\infty} \frac{1}{2M+1} \sum_{m=-M}^{M} x(m)x(m+k) \ ; \tag{9.7}$$

(*ii*) for signals with finite energy,

$$R_{xx}(k) \stackrel{\text{def}}{=} \sum_{m=-\infty}^{\infty} x(m)x(m+k) \ , \tag{9.8}$$

which is known as the **autocorrelation** function.

### *9.3.1 The autocorrelation function*

Properties of the autocorrelation function are given below (for more details see, for example, Stearns and Hush [29]):

*Properties*

For signals with non-zero, finite average power or finite energy:

(a) $R_{xx}(-k) = R_{xx}(k)$; $R_{xx}$ is an even function.

(b) $|R_{xx}(k)| \leq R_{xx}(0)$ ;

For signals with non-zero, finite average power:

(c) if $\{x(m)\}$ contains a periodic component, with period $M$, then $R_{xx}(k)$ also contains a periodic component, with period $M$.

(d) if $R_{xx}(k)$ has no periodic component, then

$$\lim_{k\to\infty} R_{xx}(k) = \mu_x^2 \ , \quad \text{where } \mu_x \stackrel{\text{def}}{=} \lim_{M\to\infty} \frac{1}{2M+1} \sum_{m=-M}^{M} x(m)$$

denotes the average value for the sequence $\{x(m)\}$;

(e) defining $\sigma_x^2 \stackrel{\text{def}}{=} \left[\lim_{M\to\infty} \frac{1}{2M+1} \sum_{m=-M}^{M} x^2(m)\right] - \mu_x^2$, then $R_{xx}(0) = \mu_x^2 + \sigma_x^2$ and

$$-\sigma_x^2 + \mu_x^2 \leq R_{xx}(k) \leq \sigma_x^2 + \mu_x^2 \ .$$

The term $\mu_x^2$ is known as the **dc component** of $R_{xx}(k)$.

As a consequence of (9.7) and in view of (9.4),

$$\boldsymbol{P}_{\text{av}} = R_{xx}(0) \ .$$

Similarly, for a discrete-time, finite energy signal $\{x(m)\}$,

$$\boldsymbol{E} = R_{xx}(0) \ ,$$

in view of (9.3) and (9.8). A typical graph of an autocorrelation sequence is illustrated in Figure 9.2.

*Example 9.1*

Using the result: $\displaystyle\sum_{r=0}^{\infty} w^r = \frac{1}{1-w}$ , for $|w| < 1$,

and given $x(k) = a^k\zeta(k)$, $|a| < 1$ :

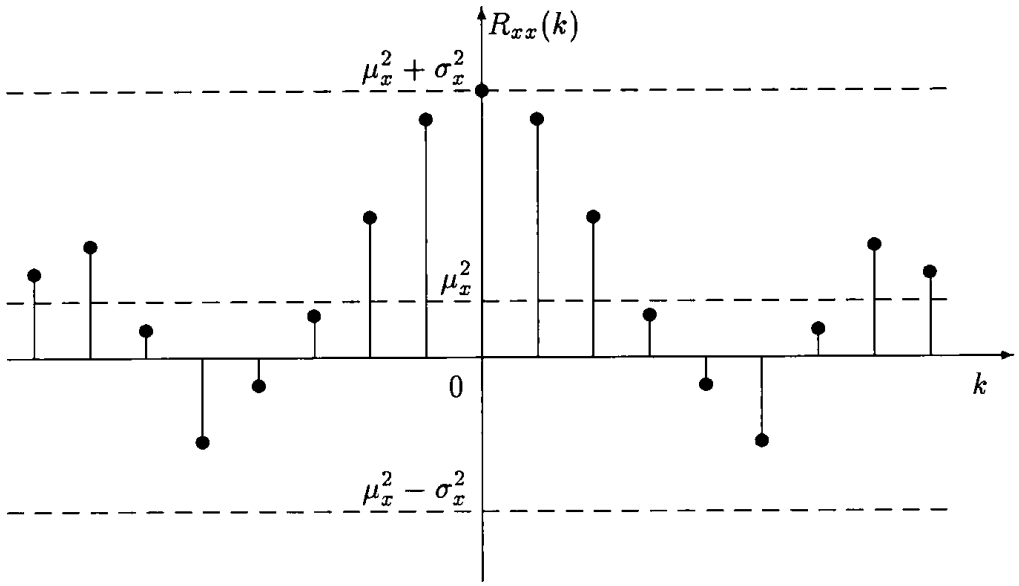



Figure 9.2: Typical graph of an autocorrelation function $R_{xx}(k)$ with no periodic component.

(a) show that the sequence $\{x(k)\}$ has finite energy;

(b) determine the autocorrelation sequence for $\{x(k)\}$.

(a) $\displaystyle\sum_{k=-\infty}^{\infty} x^2(k) = \sum_{k=0}^{\infty} a^{2k} = \frac{1}{1-a^2}$ , since $|a| < 1$,

and, hence, the sequence has finite energy.

(b) Suppose $k \geq 0$, then

$$\begin{aligned} R_{xx}(k) &= \sum_{m=-\infty}^{\infty} x(m)x(m+k) \\ &= a^k \sum_{m=0}^{\infty} a^{2m} \\ &= \frac{a^k}{1-a^2} \cdot \end{aligned}$$

Using the even symmetry property of $\{R_{xx}(k)\}$,

$$R_{xx}(k) = \frac{a^{-k}}{1-a^2} , \qquad \text{when } k < 0 .$$

Thus, for all $k$, $R_{xx}(k) = \dfrac{a^{|k|}}{1-a^2}$.

□

### *9.3.2 The cross-correlation function*

The cross-correlation function, defined in (9.5) and (9.6), also has some important properties, which are listed below.

*Properties*

For signals with non-zero, finite average power or finite energy:

(a) $R_{xy}(-k) = R_{yx}(k)$

(b) $|R_{xy}(k)| \leq \{R_{xx}(0)R_{yy}(0)\}^{\frac{1}{2}} \leq \frac{1}{2}\{R_{xx}(0) + R_{yy}(0)\}$

(c) If both $\{x(k)\}$ and $\{y(k)\}$ contain a periodic component, with period $M$, then both $R_{xy}(k)$ and $R_{yx}(k)$ contain a periodic component, with period $M$.

This function is very useful for certain aspects of system identification, as shown in, for example, Ljung [16].

### *9.3.3 An application of the autocorrelation function*

The autocorrelation function is very useful for estimating periodicities in signals (see Property (c) in §9.3.1), including speech. In practice, a segment of speech $\{s(m);\ \ m = 0, 1, \ldots, M-1\}$ is available for analysis. This sequence has finite energy. The **'short-time' autocorrelation function** (which is an estimate of the true autocorrelation function) is determined by

$$r_{ss}(k) = \sum_{m=0}^{M-1-k} s(m)s(m+k) \ , \tag{9.9}$$

for $k = 0,\ 1, \ldots,\ M-1$. It can be shown that $r_{ss}$ has the even symmetry property (see Property (a) in §9.3.1), if the speech segment is assumed to be identically zero outside the interval $0 \leq m \leq M-1$. Let $\{\tilde{s}\}$ denote the sequence

$$\{\ldots 0,\ 0,\ \underset{\uparrow}{s(0)},\ s(1), \ldots,\ s(M-1),\ 0,\ 0, \ldots\} \ ,$$

then

$$\begin{aligned} r_{ss}(-k) &= \sum_{m=0}^{M-1+k} \tilde{s}(m)\tilde{s}(m-k) \\ &= \sum_{i=-k}^{M-1} \tilde{s}(i+k)\tilde{s}(i) \ , \quad \text{with} \ \ i = m-k \ , \\ &= \sum_{i=0}^{M-1-k} \tilde{s}(i+k)\tilde{s}(i) \ , \quad \text{since } \tilde{s}(i) = 0 \text{ for } i < 0 \text{ and } i \geq M \ , \\ &= \sum_{i=0}^{M-1-k} s(i+k)s(i) \\ &= r_{ss}(k) \ . \end{aligned}$$

For voiced speech, peaks occur in $r_{ss}(k)$ approximately at multiples of the pitch period. Hence, *the pitch period of a speech signal can be estimated by finding the location of the first maximum in the autocorrelation function.* If there are no strong periodicity peaks, then this indicates a lack of periodicity in the waveform, implying that the section of the speech signal is unvoiced. This technique is illustrated for a segment $\{x(m)\}$ of a non-nasal, voiced speech signal from the spoken word 'should'. The speech signal for the word 'should' is sampled at a rate of 13.16 kHz (sampling period 0.076 msec. approximately) and its graph is plotted in Figure 9.3. The autocorrelation function is obtained using MATLAB with the command **xcorr**.

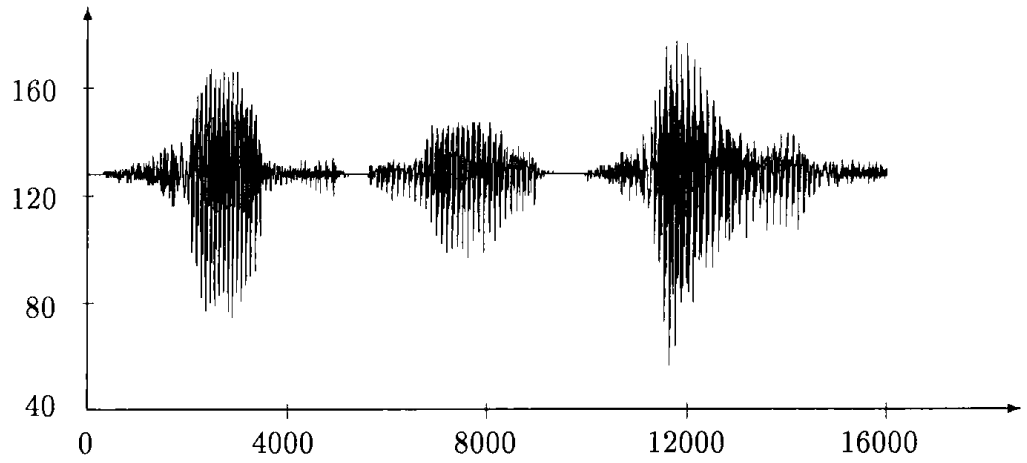



Figure 9.3: Graph of speech data for the word 'should'.

This calculates

$$\sum_{m=0}^{M-1-|k|} x(m)x(m+k) \ , \quad k = -(M-1), \ldots, \ M-1, \tag{9.10}$$

which is identical to the short-time autocorrelation function (9.9) when the time-lag $k$ satisfies $k \geq 0$. However, the sequence is constructed so that the time-lag $k = 0$ for this sequence occurs in the middle and, thus, some adjustment is necessary. Finally, each value in the autocorrelation sequence is divided by the energy in the sequence, that is the sequence is **normalized**. To illustrate this, a segment of speech, with 512 data values (its graph is plotted in Figure 9.4), is selected from the speech data for the word 'should' and its autocorrelation function, plotted in Figure 9.5, is obtained using the MATLAB commands:

```
r=xcorr(x);          % autocorrelation sequence evaluation
l=length(r);
a=r((l+1)/2:l);      % zero th time-lag occurs at a(1)
na=a/a(1);           % normalized sequence
plot(na);
```

As can be seen in Figure 9.5, the application of (9.10) using the given data generates a decreasing function curve which masks any periodicity. To overcome this problem

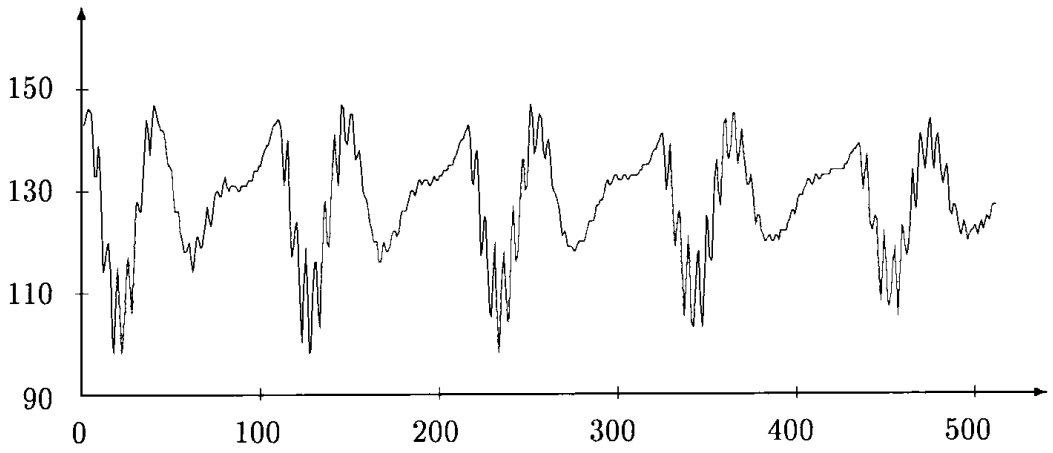



Figure 9.4: Segment of speech, 512 data values, for the word 'should'.

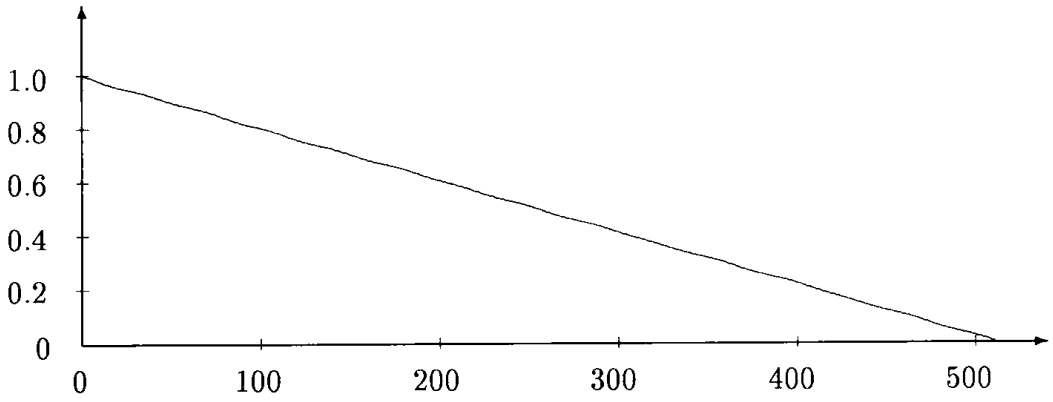



Figure 9.5: Normalized autocorrelation function for the speech segment.

the mean value of the data set is subtracted from each data value to form a new sequence. The autocorrelation function for this new sequence is shown in Figure 9.6. It is clear that the graph has dominant peaks at approximately 110, 220 and 330

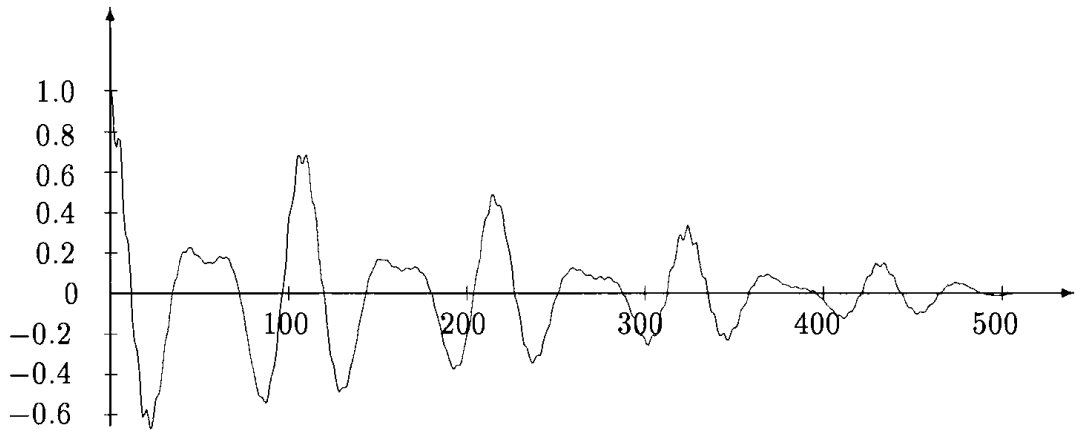



Figure 9.6: Normalized autocorrelation function for the speech data with zero mean.

which suggests that the pitch period is approximately $110 \times 0.076$ msec. $\approx 8.4$ msec. and the pitch frequency is approximately $13160 \div 110 \approx 120$ Hz (or 754 rads./sec.). One may ask the question "Why do we need to use this procedure, when the approximate periodicity of the signal can be seen graphically in Figure 9.4?". The main point, here, is that the use of the autocorrelation function together with a 'peak picking' algorithm can be implemented quite easily electronically and, hence, the process of estimating the pitch period (or frequency) of a non-nasal voiced speech signal can be automated.

## 9.4 LINEAR PREDICTIVE FILTERS

The method of **linear prediction** (or **linear predictive coding**, as it is sometimes known) is based on the classical least squares method and utilizes the concept of **autocorrelation matrices**. An informative tutorial paper on this technique has been presented by Makhoul [17], and a detailed text has been written by Markel and Gray [18] (see, also, Haddad and Parsons [8], Parsons [25], Rabiner and Schafer [27] and Saito and Nakata [28]). Based on this method and properties of the autocorrelation function, discussed in §9.3.1 and §9.3.3, some filtering techniques can be used to analyse speech signals.

Let us suppose that an appropriate *linear* model for speech production is the **autoregressive** (AR) model

$$s(m) = G_s u(m) + \sum_{k=1}^{p} a_k s(m-k) \ ,$$

where $a_k$ are the model parameters, $p \geq 1$, $u(m)$ is the model input, $G_s$ is the speech gain parameter and $s(m)$ is the model output. Our main objective is to estimate

the model parameters $a_k$. We introduce the concept of a **linear predictor**, which is a system whose output sequence $\{\tilde{s}\}$ is given by

$$\tilde{s}(m) = \sum_{k=1}^{p} \alpha_k s(m-k) \ ,$$

corresponding to the input sequence $\{s\}$. This system may be used to 'predict' $\{s(m)\}$, in the sense that, if $s(m-p)$, $s(m-(p-1)), \ldots,$ $s(m-1)$ are given, $\tilde{s}(m)$ can be determined and used as an approximation to $s(m)$. Here, $\alpha_k$ are known as the **predictor coefficients**, $p$ is the order of the filter, and $\tilde{s}(m)$ is the predicted value of $s(m)$. Suppose we wish to perform an analysis with a finite-length data set $\{s(m);\ m = 0, \ldots,\ M-1\}$, where $M$ is the number of signal samples. With sequences $\{s\}$ and $\{\tilde{s}\}$, we can define an error sequence $\{e\}$, where

$$e(m) = s(m) - \tilde{s}(m) \ .$$

For the present, it is assumed that $s(m) = 0$ if $m < 0$ or $m \geq M$, in which case $e(m) = 0$ when $m < 0$ or $m \geq p + M$. The predictor coefficients can be determined by minimizing the mean square error

$$e_{\rm ms} = \sum_{m=0}^{p+M-1} |e(m)|^2 \ .$$

A necessary condition for minimizing the mean square error, with respect to the predictor coefficients, is

$$\frac{\partial e_{\rm ms}}{\partial \alpha_i} = 0$$

$$\iff \quad 2 \sum_{m=0}^{p+M-1} (s(m) - \tilde{s}(m)) \frac{\partial}{\partial \alpha_i} (s(m) - \tilde{s}(m)) \ = \ 0 \ .$$

But

$$\frac{\partial}{\partial \alpha_i} (s(m) - \tilde{s}(m)) = -s(m-i)$$

and therefore

$$\sum_{m=0}^{p+M-1} (\tilde{s}(m) - s(m)) s(m-i) = 0 \ , \quad i = 1, \ldots,\ p \ .$$

This produces the relationship

$$\sum_{m=0}^{p+M-1} \sum_{k=1}^{p} \alpha_k s(m-k) s(m-i) = \sum_{m=0}^{p+M-1} s(m) s(m-i) \ ,$$

or, since $s(m-k) = 0$ when $m > M-1+k$ and $s(m) = 0$ when $m > M-1$,

$$\sum_{k=1}^{p} \alpha_k \sum_{m=0}^{M-1+k} s(m-k) s(m-i) = \sum_{m=0}^{M-1} s(m) s(m-i) \ , \quad i = 1, \ldots,\ p \ . \qquad (9.11)$$

With $n = m - i$, and since $s(n) = 0$ when $n < 0$, we have

$$\sum_{m=0}^{M-1+k} s(m-k)s(m-i) = \sum_{n=-i}^{M-1+k-i} s(n)s(n+i-k)$$
$$= \sum_{n=0}^{M-1-(i-k)} s(n)s(n+i-k) \ , \quad i,k = 1,\ldots,\ p \ ,$$

which is the short-time autocorrelation function evaluated at $(i-k)$ (see (9.9)). Define

$$r(i-k) \stackrel{\text{def}}{=} \sum_{m=0}^{M-1+k} s(m-k)s(m-i) \ , \tag{9.12}$$

where, for the sake of clarity, we have omitted the double subscript on the short-time autocorrelation function $r$. As a consequence of (9.12),

$$r(i) = \sum_{m=0}^{M-1} s(m)s(m-i) \ .$$

Hence, (9.11) may be rewritten in the form

$$\sum_{k=1}^{p} \alpha_k r(i-k) = r(i) \ , \qquad i = 1,\ldots,\ p \ .$$

As shown in §9.3.3, since $s(m)$ is zero outside the interval $0 \leq m \leq M-1$, $r$ is an even function. Thus, $r(k-i) = r(i-k)$ and so we may write

$$\sum_{k=1}^{p} \alpha_k r(|i-k|) = r(i) \ , \qquad i = 1,\ldots,\ p \ , \tag{9.13}$$

which are known as the **normal** equations. The matrix,

$$\begin{bmatrix} r(0) & r(1) & r(2) & \cdots & r(p-1) \\ r(1) & r(0) & r(1) & \cdots & r(p-2) \\ r(2) & r(1) & r(0) & \cdots & r(p-3) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ r(p-1) & r(p-2) & r(p-3) & \cdots & r(0) \end{bmatrix} \ ,$$

known as the **autocorrelation matrix**, is a **Toeplitz** matrix, that is the matrix is symmetric and the elements along diagonals, sloping downwards from left to right, are all equal. An efficient recursive method for solving Equation (9.13) is **Durbin's algorithm**, which, sometimes, is attributed to Levinson or both Durbin and Levinson, see Rabiner and Schafer [27]. The algorithm is as follows:

1. *Initialization*:

$$e^{(0)} = r(0)$$
$$k_1 = r(1)/e^{(0)}$$
$$\alpha_1^{(1)} = k_1$$
$$e^{(1)} = (1-k_1^2)e^{(0)}$$

2. (*For* $2 \leq i \leq p$)

$$k_i = \left\{ r(i) - \sum_{n=1}^{i-1} \alpha_n^{(i-1)} r(i-n) \right\} / e^{(i-1)}$$

$$\alpha_i^{(i)} = k_i$$

$$\alpha_n^{(i)} = \alpha_n^{(i-1)} - k_i \alpha_{i-n}^{(i-1)}, \qquad 1 \leq n \leq i-1$$

$$e^{(i)} = (1 - k_i^2) e^{(i-1)}.$$

The predictor coefficients are then given by $\alpha_k^{(p)}, \quad k = 1, \ldots, p.$

*Remarks*:

(a) The algorithm also generates predictor coefficients for predictors of order less than $p$, that is in the process of generating the prediction coefficients for a $p^{\text{th}}$ order predictor, at the $i^{\text{th}}$ stage ($i < p$), predictor coefficients are generated for an $i^{\text{th}}$ order predictor.

(b) Note that $e^{(i)}$ denotes the prediction error for a predictor of order $i$ and, thus, the prediction error can be monitored at each stage (as discussed in Rabiner and Schafer [27], Chapter 8 §8.3.2).

(c) It is known (see, for example, Parsons [25]) that $|k_i| \leq 1$, for every $i$, and that the linear predictive filter is stable.

(d) A MATLAB function m-file for this algorithm is presented in Appendix B.1.

*Example 9.2*
Show that Durbin's algorithm, for solving the normal equations, gives the exact values for the predictor coefficients of a second order predictor with autocorrelation matrix

$$\begin{bmatrix} 4 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 4 \end{bmatrix}.$$

We require to solve

$$r(0)\alpha_1 + r(1)\alpha_2 = r(1)$$
$$r(1)\alpha_1 + r(0)\alpha_2 = r(2)$$

with $r(0) = 4,\ r(1) = 2,\ r(2) = 1$, that is

$$4\alpha_1 + 2\alpha_2 = 2$$
$$2\alpha_1 + 4\alpha_2 = 1\ .$$

This system of equations has the exact solution $\alpha_1 = \frac{1}{2},\ \alpha_2 = 0$. Using Durbin's algorithm:

$$e^{(0)} = r(0) = 4$$

$$
\begin{aligned}
k_1 &= r(1)/e^{(0)} = \tfrac{1}{2} \\
\alpha_1^{(1)} &= k_1 = \tfrac{1}{2} \\
e^{(1)} &= (1-k_1^2)e^{(0)} = 3 \\
k_2 &= \left(r(2) - \alpha_1^{(1)} r(1)\right)/e^{(1)} = 0 \\
\alpha_2^{(2)} &= k_2 = 0 \\
\alpha_1^{(2)} &= \alpha_1^{(1)} - k_2\alpha_1^{(1)} = \tfrac{1}{2} \\
e^{(2)} &= (1-k_2^2)e^{(1)} = 3 \ .
\end{aligned}
$$

Thus, the predictor coefficients are $\alpha_1 = \alpha_1^{(2)} = \frac{1}{2}$, $\alpha_2 = \alpha_2^{(2)} = 0$.

□

In speech processing, the above method is known as the **autocorrelation method**. Another method, known as the **covariance method**, is based on the assumption that some signal values outside the interval $0 \leq m \leq M-1$ are available for analysis, specifically, $s(m)$ for $-p \leq m < 0$. For this case, the mean square error is minimized over the interval $0 \leq m \leq M-1$, that is we minimize

$$
e_{\text{ms}} = \sum_{m=0}^{M-1} [s(m) - \tilde{s}(m)]^2 \ ,
$$

which leads to

$$
\sum_{k=1}^{p} \alpha_k \sum_{m=0}^{M-1} s(m-k)s(m-i) = \sum_{m=0}^{M-1} s(m)s(m-i) \ . \tag{9.14}
$$

Let

$$
\rho(k,i) = \sum_{m=0}^{M-1} s(m-k)s(m-i)
$$

then

$$
\rho(0,i) = \sum_{m=0}^{M-1} s(m)s(m-i)
$$

and, therefore, (9.14) becomes

$$
\sum_{k=1}^{p} \alpha_k \rho(k,i) = \rho(0,i) \ . \tag{9.15}
$$

The equations given by (9.15) are also known as the **normal equations**. Since we can use values of $s(m)$ in the interval $-p \leq m \leq M-1$,

$$
\begin{aligned}
\rho(k,i) &= \sum_{m=0}^{M-1} s(m-k)s(m-i) \\
&= \sum_{n=-i}^{M-1-i} s(n)s(n+i-k) \ , \qquad \text{with} \quad n = m-i,
\end{aligned}
$$

for $1 \leq i \leq p$ and $0 \leq k \leq p$. The **covariance matrix**

$$\begin{bmatrix} \rho(1,1) & \rho(1,2) & \cdots & \rho(1,p) \\ \rho(2,1) & \rho(2,2) & \cdots & \rho(2,p) \\ \vdots & \vdots & \vdots & \vdots \\ \rho(p,1) & \rho(p,2) & \cdots & \rho(p,p) \end{bmatrix},$$

is symmetric, since $\rho(k,i) = \rho(i,k)$, but it is not Toeplitz. An efficient, recursive method for solving Equation (9.15) is the **Cholesky decomposition algorithm** (see Rabiner and Schafer [27]):

1. (*For* $i = 1, \ldots, p$)

$$\rho^{(1)}(i,k) = \rho(i,k) \qquad k = i, \ldots, p$$

2. (*For* $k = 1, \ldots, p$)

$$w(k,k) = \sqrt{\rho^{(k)}(k,k)}$$
$$w(k,n) = \rho^{(k)}(k,n)/w(k,k) \qquad n = k+1, \ldots, p$$

*For* $n = k+1, \ldots, p$

$$\rho^{(k+1)}(n,i) = \rho^{(k)}(n,i) - w(k,i)w(k,n) \quad i = n, \ldots, p$$

3. (*For* $k = 1, \ldots, p$)

$$a_k = \left( \rho(0,k) - \sum_{i=1}^{k-1} a_i w(i,k) \right) / w(k,k)$$

4.

$$\alpha_p = a_p / w(p,p)$$
$$\alpha_k = \left( a_k - \sum_{i=k+1}^{p} \alpha_i w(k,i) \right) / w(k,k) \quad k = p-1, \ldots, 1.$$

A MATLAB function m-file for this algorithm is presented in Appendix B.2. For the Cholesky decomposition algorithm there is no guarantee that the linear predictive filter is stable, as was the case for the autocorrelation method.

*Example 9.3*
Show that the Cholesky decomposition algorithm, for solving the normal equations, gives the exact values for the predictor coefficients of a second order predictor with covariance matrix

$$\begin{bmatrix} 10 & 4 \\ 4 & 7 \end{bmatrix}$$

and given $\rho(0,1) = 4$, $\rho(0,2) = 0$ .

We require to solve

$$\rho(1,1)\alpha_1 + \rho(2,1)\alpha_2 = \rho(0,1)$$
$$\rho(1,2)\alpha_1 + \rho(2,2)\alpha_2 = \rho(0,2) \ ,$$

that is

$$10\alpha_1 + 4\alpha_2 = 4$$
$$4\alpha_1 + 7\alpha_2 = 0 \ .$$

This system of equations has the exact solution $\alpha_1 = \frac{14}{27}$, $\alpha_2 = -\frac{8}{27}$. Using the Cholesky decomposition algorithm, with $p = 2$, we obtain:

Step 1: $i = 1$, $\rho^{(1)}(1,1) = \rho(1,1) = 10$
$\rho^{(1)}(1,2) = \rho(1,2) = 4$
$i = 2$, $\rho^{(1)}(2,2) = \rho(2,2) = 7$

Step 2: $k = 1$, $w(1,1) = \sqrt{\rho^{(1)}(1,1)} = \sqrt{10}$
$w(1,2) = \rho^{(1)}(1,2)/w(1,1) = \frac{4}{\sqrt{10}}$
$n = 2$, $\rho^{(2)}(2,2) = \rho^{(1)}(2,2) - w^2(1,2) = \frac{27}{5}$
$k = 2$, $w(2,2) = \sqrt{\rho^{(2)}(2,2)} = \sqrt{\frac{27}{5}}$

Step 3: $k = 1$, $a_1 = \rho(0,1)/w(1,1) = \frac{4}{\sqrt{10}}$
$k = 2$, $a_2 = [\rho(0,2) - a_1 w(1,2)]/w(2,2) = -\frac{8}{3\sqrt{15}}$

Step 4: $\alpha_2 = a_2/w(2,2) = -\frac{8}{27}$
$\alpha_1 = (a_1 - w(1,2)\alpha_2)/w(1,1) = \frac{14}{27}$ .

□

As indicated in §9.2 for speech production, an all-pole (AR) model is an appropriate model for non-nasal voiced sounds. The composite effects of lip radiation,vocal tract and glottal excitation can be represented by a time-varying digital filter, which can be approximated by a linear time-invariant system over short intervals of time whose system function has the form:

$$H(z) = \frac{G_s}{1 - \sum_{k=1}^{p} a_k z^{-k}} \ ,$$

where $G_s$ is the speech gain parameter. This model gives a good representation of speech sounds provided $p$ is large enough. The filter is assumed to be excited by a train of impulses:

$$u(m) = \sum_{n=n_1}^{n_2} \delta(m-n) = \begin{cases} 1 \ , & \text{if } n_1 \le m \le n_2 \ , \\ 0 \ , & \text{otherwise} \ , \end{cases}$$

for voiced speech. For this filter, the speech samples $\{s(m)\}$ are related to the excitation sequence $\{u(m)\}$ by the difference equation

$$s(m) = G_s u(m) + \sum_{k=1}^{p} a_k s(m-k) \ . \tag{9.16}$$

Using a linear predictor:

$$\tilde{s}(m) = \sum_{k=1}^{p} \alpha_k s(m-k)$$

for this system, the prediction error is given by

$$e(m) = s(m) - \sum_{k=1}^{p} \alpha_k s(m-k) \ .$$

A rough rule of thumb, as described in Rabiner and Schafer [27] and Parsons [25], is that the order of predictor satisfies

$$p = f_s/1000 + q,$$

where $f_s$ is the sampling frequency in hertz and $q$ is typically 2 or 3.

If the speech signal fits the model (9.16) exactly and if $\alpha_k = a_k$, then $e(m) = G_s u(m)$, that is for voiced speech, $\{e(m)\}$ would consist of a train of impulses. This suggests a method for the determination of the pitch period of a voiced speech segment using the method of linear prediction. First, the autocorrelation or covariance method may be used to find the predictor coefficients. The error signal $\{e(m)\}$ is then determined and the difference between consecutive peaks in $r_{ee}$ is an estimate of the pitch period. To illustrate this, the autocorrelation method, using Durbin's algorithm and with 20 prediction parameters, is performed on the same speech data used in §9.3. Graphs of the error signal and the autocorrelation function for the error signal are illustrated in Figure 9.7 and Figure 9.8, respectively. The graph,

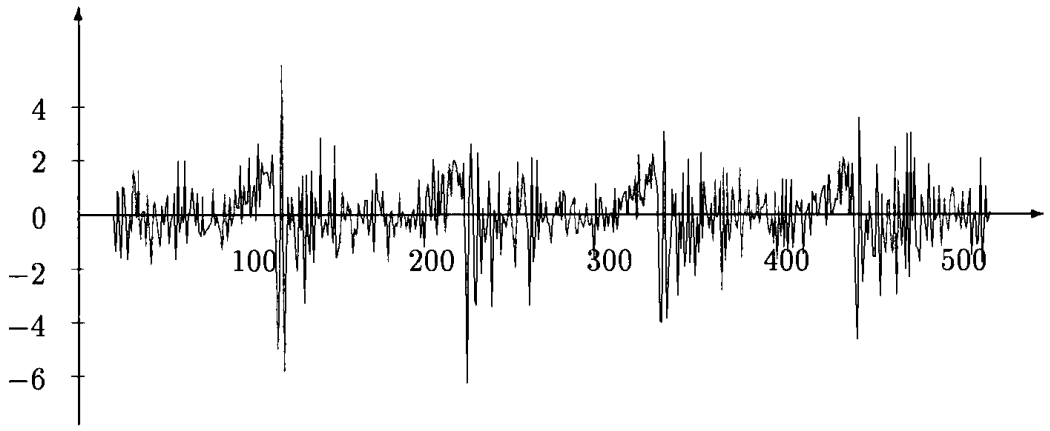



Figure 9.7: Error signal.

shown in Figure 9.8, has dominant peaks at approximately 110, 220 and 330 which confirms that the pitch period is approximately 8.4 msec. (see §9.3).

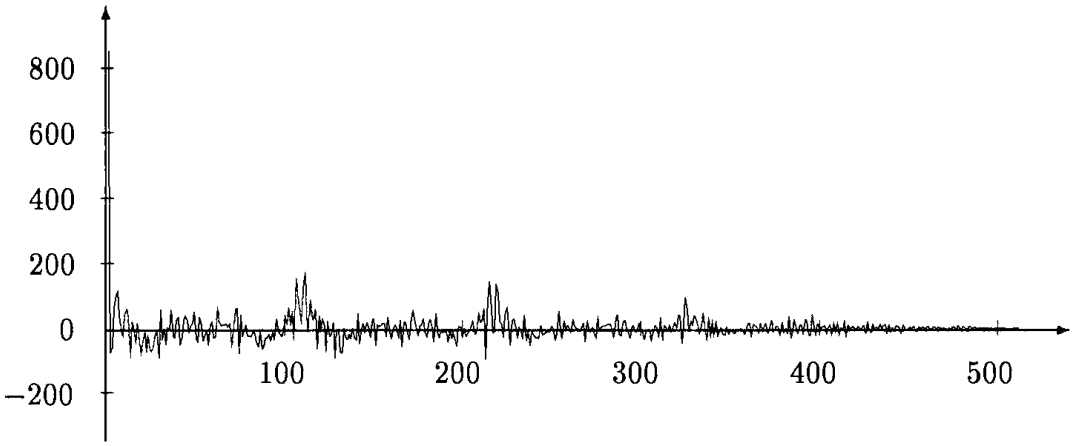



Figure 9.8: Autocorrelation function for the error signal.

## 9.5 ESTIMATION OF THE SPEECH GAIN PARAMETER

Suppose the excitation signal is a unit impulse at $m = 0$ and the interval for analysis is approximately the length of a pitch period, then the total energy in the error signal is $G_s^2$ and (9.16) holds with $u(m) = \delta(m)$. For voiced speech, since the input may be approximated by a train of impulses, then, when the interval is approximately the length of a pitch period and assuming $\alpha_k = a_k$ for all $k$, the impulse response sequence for the filter, $\{h(m)\}$, is described by the relationship

$$h(m) = G_s\delta(m) + \sum_{k=1}^{p} \alpha_k h(m-k) \ , \tag{9.17}$$

and $h(m) \approx s(m)$ in the analysis interval. Assuming the filter is causal, the short-time autocorrelation function for $h(m)$ can be approximated by

$$\tilde{r}(m) = \sum_{n=0}^{M-1-m} h(n)h(m+n) \ , \quad m = 0,\ 1, \ldots \tag{9.18}$$

$$\approx \sum_{n=0}^{M-1-m} s(n)s(m+n)$$

$$= r(m) \ , \quad m = 0,\ 1, \ldots,\ M-1 \ . \tag{9.19}$$

In addition, using (9.17) and (9.18),

$$\begin{aligned}
\tilde{r}(0) &= \sum_{n=0}^{M-1} h^2(n) \\
&= \sum_{n=0}^{M-1} h(n) \left[ G_s\delta(n) + \sum_{k=1}^{p} \alpha_k h(n-k) \right] \\
&= G_s h(0) + \sum_{k=1}^{p} \alpha_k \sum_{n=0}^{M-1} h(n)h(n-k)
\end{aligned}$$

$$= G_s h(0) + \sum_{k=1}^{p} \alpha_k \sum_{i=-k}^{M-1-k} h(i)h(i+k) \ , \quad \text{with } i = n-k,$$

$$= G_s h(0) + \sum_{k=1}^{p} \alpha_k \sum_{i=0}^{M-1-k} h(i)h(i+k) \ , \quad \text{since } h(i) = 0 \text{ for } i < 0,$$

$$= G_s^2 + \sum_{k=1}^{p} \alpha_k \tilde{r}(k) \ , \quad \text{since } h(0) = G_s \ .$$

By matching the energy in the signal, $r(0)$, with the energy in the predicted samples, $\tilde{r}(0)$, and utilizing (9.19), the speech gain parameter can be estimated from

$$G_s^2 \approx r(0) - \sum_{k=1}^{p} \alpha_k r(k) \ . \tag{9.20}$$

## 9.6 FORMANT ESTIMATION

As stated earlier, a simple model for voiced speech is excitations produced by the vocal chords and filtered by the vocal tract. For an adult male, the resonances produced by the vocal tract are approximately 1 kHz apart. However, the location of these resonances (or formants) can be changed by moving the tongue and lips. Normally, only the first three formants are of importance (in practice). Formant estimation is often performed using frequency-domain analysis. The segment of speech, obtained by windowing, must be long enough that the individual harmonics can be resolved, but short enough that the speech signal can assumed to be modelled by a linear time-invariant system. Normally, a window is chosen to be at least two or three pitch periods long. The pitch period varies from a minimum of approximately 80 Hz in adult males to a maximum of approximately 400 Hz in adult females. For example, three periods of 100 Hz voiced speech is approximately 30 msec. The signal sampling frequency is usually in the range $8 - 15$ kHz, and the number of data samples is chosen to be a power of two. In the spectrum, the narrow peaks are the harmonics of the excitation, whilst the peaks in the envelope of the peaks in the spectrum indicate the formants due to the vocal tract. To understand more clearly what is meant by an envelope, study the dashed-curve illustrated in the diagram of Figure 9.10.

Linear prediction may be used to determine the spectrum. Assuming an all-pole model,

$$H(z) = \frac{G_s}{1 - \sum_{k=1}^{p} a_k z^{-k}} \ ,$$

the model parameters can be estimated using the method of linear prediction and the speech gain parameter $G_s$ can be estimated as indicated in §9.5. Hence, the spectrum can be obtained by evaluating $H(z)$ on the unit circle $z = e^{j\theta}$. One method of achieving this is to construct a sequence $\{\tilde{a}\}$ which is composed of the sequence $\{1, -a_1, -a_2, \ldots, -a_p\}$ padded with zeros so that the length of the sequence, say

$N$, is a power of two, that is $\{\tilde{a}\} = \{1, -a_1, -a_2, \ldots, -a_p, 0, 0, 0, \ldots\}$, and divide $G_s$ by each element of the DFT of the sequence $\{\tilde{a}\}$ to produce the sequence

$$\left\{ G_s \left[ 1 - \sum_{k=1}^{p} a_k e^{-j\frac{2\pi kn}{N}} \right]^{-1} , \quad n = 0, 1, \ldots, N-1 \right\} .$$

An FFT algorithm can be used to determine the DFT of the sequence $\{\tilde{a}\}$. For sufficiently large $N$, the magnitude of the resulting sequence yields a high-resolution representation of the vocal tract amplitude spectrum

$$|H(e^{j\theta})| = \frac{G_s}{\left| 1 - \sum_{k=1}^{p} a_k e^{-j\theta k} \right|} .$$

The formants are estimated by detecting the peaks in the spectrum envelope.

To illustrate the above procedure, we apply the technique to the voiced speech data used in §9.3, using a 512-point FFT algorithm. In addition, to identify the formants more easily, the graph of the logarithm of the vocal tract amplitude spectrum is plotted. The graph of $20 \log_{10} |H(e^{j\theta})|$, with units measured in decibels, plotted against the frequency index $n$ is shown in Figure 9.9. The peak at approximately

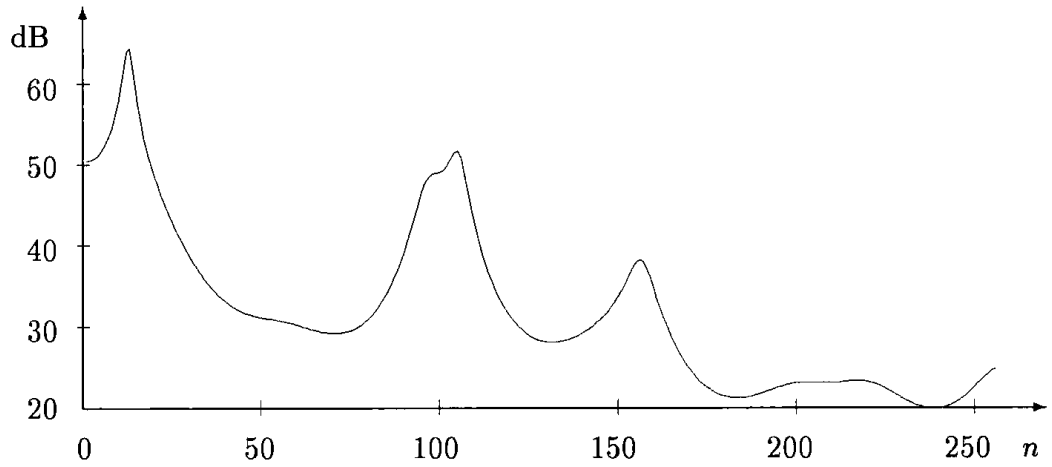



Figure 9.9: Log-amplitude spectrum using a linear predictive filter.

$n = 16$ corresponds to the first formant, denoted by F1. To obtain the frequency in Hertz, corresponding to values of the frequency index $n$, we recall from §6.6 that samples are $2\pi/(NT)$rads/sec. $= (NT)^{-1}$Hz apart, and so

$$f = \frac{n}{NT} ,$$

where $T$ is the sampling period and $N$ denotes the number of data values used for the FFT. Thus, since $T = 0.076$ msec. and $N = 512$, F1 has a frequency of approximately 386 Hz. Two other formants can easily be identified in Figure 9.9, and their frequencies are calculated to be approximately 2699 Hz and 4048 Hz.

The identification and estimation of formants for voiced speech can be made easier by *pre-emphasizing* the speech signal using a filter with impulse response $\delta(k) - a\delta(k-1)$, with $a \approx 1$ (see Deller *et al* [5]; Chapter 5). Such a filter has similar characteristics to those filters used to model lip radiation. If two formants are close to each other, the peaks in the log-amplitude spectrum can be accentuated by evaluating the spectrum on a circle with radius $\rho < 1$. This modification can be achieved quite easily by premultiplying the linear prediction parameters by $\rho$ before computing the DFT. For more details see, for example, Deller *et al* [5] and Parsons [25].

The graphs of the logarithm of the spectrum of the speech signal and its envelope are shown in Figure 9.10. Comparing the graph of the envelope of the logarithm of the spectrum with Figure 9.9, we see that the peaks in the envelope correspond, approximately, to the peaks in the vocal tract amplitude spectrum and, therefore, formants can be found using Figure 9.10. The **discrete power spectrum** is also

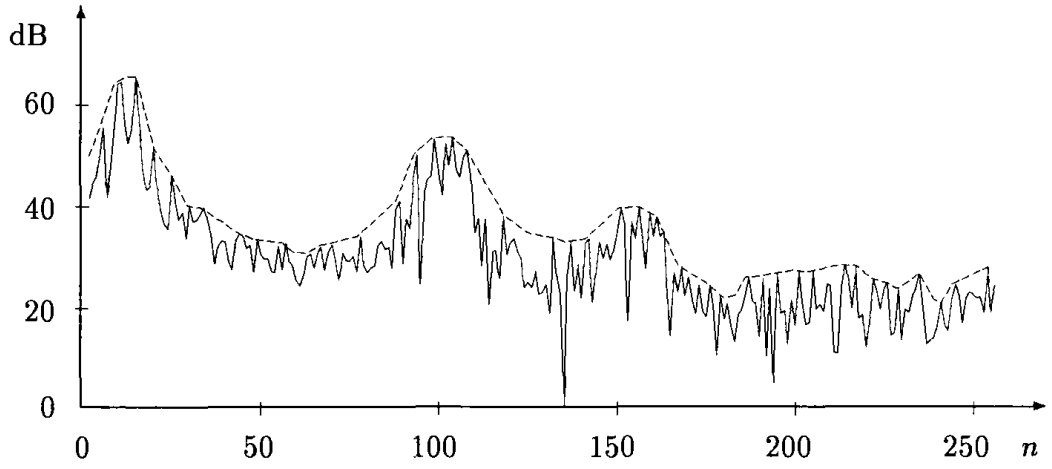



Figure 9.10: Log-amplitude spectrum and envelope (shown dashed) for the speech segment.

useful in locating formants. The power spectrum of a sequence, which is a function of frequency, is the DTFT of the autocorrelation function for the given sequence (see Stearns and Hush [29] for a more detailed discussion). A graph of the logarithm of the magnitude of the power spectrum for the zero-mean segment of speech, considered in §9.3, is shown in Figure 9.11. Comparing this graph with those illustrated in Figures 9.9 and 9.10, we notice that the two sets of peaks in the graph of the log-magnitude of the power spectrum correspond, approximately, to the first two peaks in Figure 9.9 and the first two peaks in the envelope illustrated in Figure 9.10. Thus, Figure 9.11 can also be used to identify the formant frequencies of the vocal tract spectrum.

## 9.7 LATTICE FILTER FORMULATION

In the $i^{\text{th}}$ stage of Durbin's algorithm, let $\{\alpha_n^{(i)},\ n = 1, 2, \ldots,\ i\}$ denote the coefficients of the $i^{\text{th}}$ order linear predictor. The prediction error, in this case, is

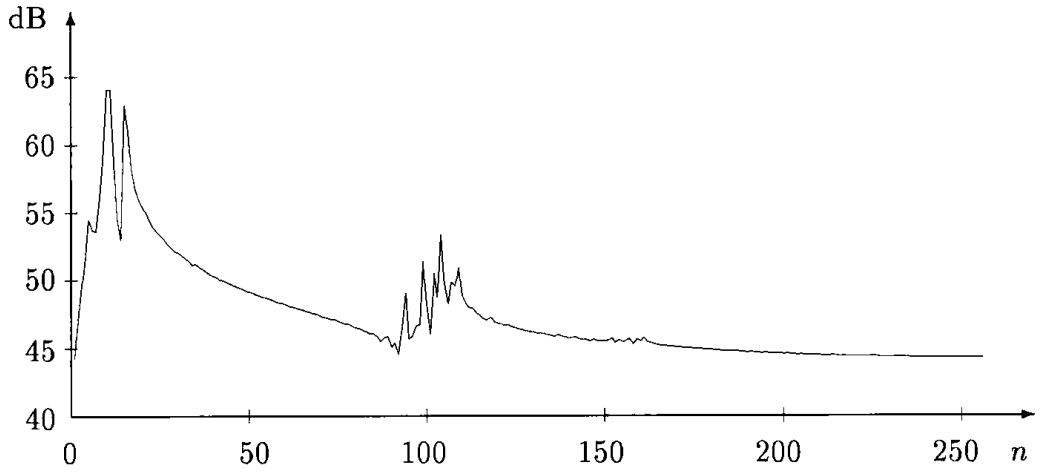



Figure 9.11: Log-magnitude of power spectrum for the speech data with zero mean.

then

$$f^{(i)}(m) = s(m) - \sum_{k=1}^{i} \alpha_k^{(i)} s(m-k) \ . \tag{9.21}$$

This can be interpreted as predicting $s(m)$ using the past $i$ samples $\{s(m-k),\ k = 1,\ 2, \ldots,\ i\}$. Since (9.21) represents a *forward* prediction using an $i^{\text{th}}$ order predictor, it is generally known as a **forward prediction error sequence**. Suppose, now, the $i$ future samples $\{s(m-i+k),\ k = 0, \ldots,\ i-1\}$ are used to predict $s(m-i)$. The sequence

$$b^{(i)}(m) = s(m-i) - \sum_{k=1}^{i} \alpha_k^{(i)} s(m+k-i) \ , \tag{9.22}$$

associated with the prediction of $s(m-i)$, is usually known as the **backward prediction error sequence**. Unfortunately, in this text, there is not sufficient space to elaborate on why this sequence does in fact represent an *error sequence*. For the reader interested in this particular aspect, it is suggested that the texts by Deller *et al* [5], Haddad and Parsons [8] and Rabiner and Schafer [27] be consulted. The concepts of forward and backward prediction are illustrated in Figure 9.12. It can be shown that (9.21) and (9.22) may be rewritten in the form:

$$f^{(i)}(m) = f^{(i-1)}(m) - k_i b^{(i-1)}(m-1) \tag{9.23}$$

$$b^{(i)}(m) = b^{(i-1)}(m-1) - k_i f^{(i-1)}(m) \ . \tag{9.24}$$

Initially, $f^{(0)}(m) = b^{(0)}(m) \stackrel{\text{def}}{=} s(m)$. The formulation (9.23)–(9.24) is known as a **lattice filter**, which can be illustrated diagrammatically as a lattice (as shown in Figure 9.13, where D represents delay), and the coefficients $k_i$ are known as **reflection** or **PARCOR** (partial correlation) **coefficients**. Lattice filters are used in many signal processing applications and are discussed in many signal processing texts, for example, Haddad and Parsons [8].

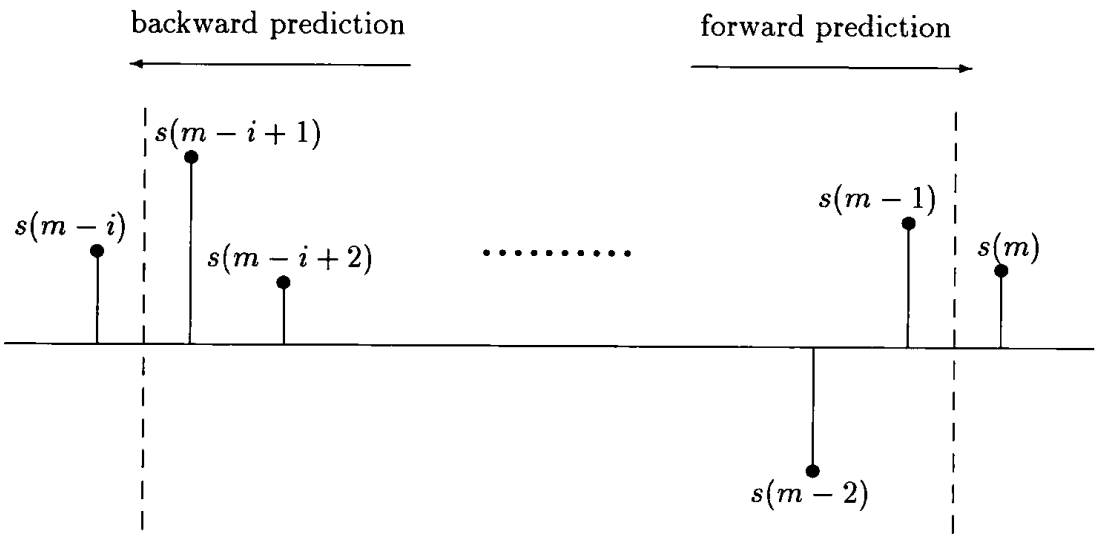



Figure 9.12: Forward and backward predictions.

This structure is known to be stable provided $|k_i| \leq 1$. Using this structure $e(m) = f^{(p)}(m)$. In particular, Burg used this lattice formulation, to develop a

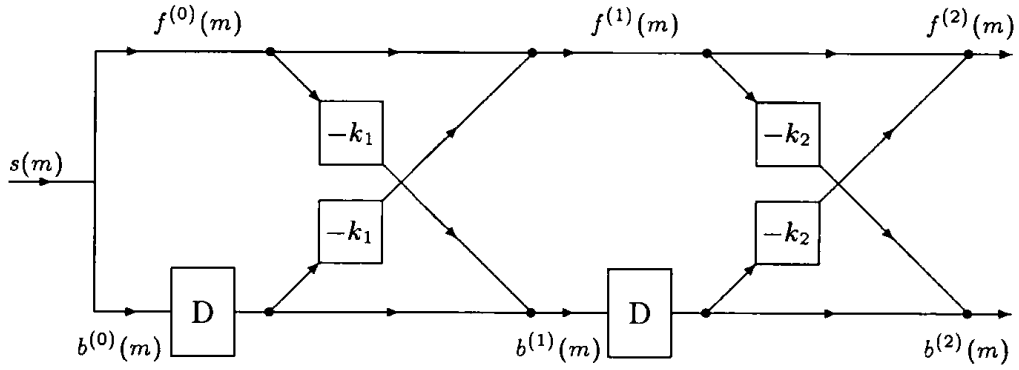



Figure 9.13: Lattice structure.

procedure based upon minimizing the sum of the mean square forward and backward prediction errors, that is Burg's method minimized

$$\sum_{m=0}^{M-1} [\{f^{(i)}(m)\}^2 + \{b^{(i)}(m)\}^2] \, .$$

In this case, the reflection (or PARCOR) coefficients are given by

$$k_i = \frac{2\sum_{m=i}^{M-1} f^{(i-1)}(m)b^{(i-1)}(m-1)}{\sum_{m=i}^{M-1}[\{f^{(i-1)}(m)\}^2 + \{b^{(i-1)}(m-1)\}^2]} .$$

**Burg's algorithm**, sometimes known as the **maximum entropy method** (see Parsons [25]), is as follows:

1. *Initialization*:

$$f^{(0)}(m) = b^{(0)}(m) = s(m), \qquad m = 0, \ldots, M-1$$

$$k_1 = \frac{2\sum_{m=1}^{M-1} f^{(0)}(m)b^{(0)}(m-1)}{\sum_{m=1}^{M-1}[\{f^{(0)}(m)\}^2 + \{b^{(0)}(m-1)\}^2]}$$

$$\alpha_1^{(1)} = k_1$$

2. (*For* $1 \le i \le p-1$)

$$\left.\begin{aligned} f^{(i)}(m) &= f^{(i-1)}(m) - k_i b^{(i-1)}(m-1) \\ b^{(i)}(m) &= b^{(i-1)}(m-1) - k_i f^{(i-1)}(m) \end{aligned}\right\} \qquad i \le m \le M-1$$

$$k_{i+1} = \frac{2\sum_{m=i+1}^{M-1} f^{(i)}(m)b^{(i)}(m-1)}{\sum_{m=i+1}^{M-1}[\{f^{(i)}(m)\}^2 + \{b^{(i)}(m-1)\}^2]}$$

$$\alpha_{i+1}^{(i+1)} = k_{i+1}$$

$$\alpha_n^{(i+1)} = \alpha_n^{(i)} - k_{i+1}\alpha_{i+1-n}^{(i)}, \qquad 1 \le n \le i .$$

As in Durbin's method, the predictor coefficients are given by $\alpha_k^{(p)}$, $k = 1, \ldots, p$.

*Example 9.4*
Given the data set {1, 2, 2, 1}, apply Burg's algorithm to determine the predictor coefficients for a second order predictor and compare these coefficients with those obtained using the autocorrelation method.

Step 1: With $s(0) = 1$, $s(1) = 2$, $s(2) = 2$, $s(3) = 1$,

$$f^{(0)}(0) = b^{(0)}(0) = 1 \qquad f^{(0)}(1) = b^{(0)}(1) = 2$$
$$f^{(0)}(2) = b^{(0)}(2) = 2 \qquad f^{(0)}(3) = b^{(0)}(3) = 1$$

$$k_1 = \frac{2(f^{(0)}(1)b^{(0)}(0) + f^{(0)}(2)b^{(0)}(1) + f^{(0)}(3)b^{(0)}(2))}{([f^{(0)}(1)]^2 + [f^{(0)}(2)]^2 + [f^{(0)}(3)]^2) + ([b^{(0)}(0)]^2 + [b^{(0)}(1)]^2 + [b^{(0)}(2)]^2)} = \frac{8}{9}$$

and so $\alpha_1^{(1)} = k_1 = \frac{8}{9}$.

Step 2: $i = 1$

$$f^{(1)}(1) = f^{(0)}(1) - k_1 b^{(0)}(0) = \tfrac{10}{9} \qquad b^{(1)}(1) = b^{(0)}(0) - k_1 f^{(0)}(1) = -\tfrac{7}{9}$$
$$f^{(1)}(2) = f^{(0)}(2) - k_1 b^{(0)}(1) = \tfrac{2}{9} \qquad b^{(1)}(2) = b^{(0)}(1) - k_1 f^{(0)}(2) = \tfrac{2}{9}$$
$$f^{(1)}(3) = f^{(0)}(3) - k_1 b^{(0)}(2) = -\tfrac{7}{9} \qquad b^{(1)}(3) = b^{(0)}(2) - k_1 f^{(0)}(3) = \tfrac{10}{9}$$

$$k_2 = \frac{2(f^{(1)}(2)b^{(1)}(1) + f^{(1)}(3)b^{(1)}(2))}{([f^{(1)}(2)]^2 + [f^{(1)}(3)]^2) + ([b^{(1)}(1)]^2 + [b^{(1)}(2)]^2)} = -\frac{28}{53}$$

and so $\alpha_2^{(2)} = k_2 = -\frac{28}{53}$. Finally, $\alpha_1^{(2)} = \alpha_1^{(1)} - k_2\alpha_1^{(1)} = \frac{72}{53}$.

Hence, the predictor coefficients are $\alpha_1^{(2)} = \frac{72}{53} \approx 1.3585$ and $\alpha_2^{(2)} = -\frac{28}{53} \approx -0.5283$.

Using the autocorrelation method,

$$r(0) = \sum_{m=0}^{3} s^2(m) = 10,\ r(1) = \sum_{m=0}^{2} s(m)s(m+1) = 8,\ r(2) = \sum_{m=0}^{1} s(m)s(m+2) = 4\,.$$

Solving
$$\begin{aligned} 10\alpha_1 \quad &+8\alpha_2 \quad = 8 \\ 8\alpha_1 \quad &+10\alpha_2 \quad = 4 \end{aligned}$$
gives $\alpha_1 = \frac{4}{3} \approx 1.3333$ and $\alpha_2 = -\frac{2}{3} \approx -0.6667$.

□

*Remarks*:

(a) Lattice methods do not give exact solutions to the normal equations of §9.4 defined by (9.13).

(b) The Burg algorithm uses only available samples with no assumptions about the unavailable data, whereas Durbin's method uses only available samples but assumes that unavailable data samples are zero.

(c) For 'small' sets of data, the Durbin algorithm has larger mean square errors and larger prediction errors than the Burg algorithm. However, for 'large' data sets, where end effects contribute little to the final results, the performance between the two algorithms is similar (see Giordano and Hsu [7] and, also, Parsons [25]).

(d) A MATLAB function m-file for the Burg lattice algorithm is presented in Appendix B.3.

Details of fast algorithms, including lattice methods, for autoregressive spectral estimation can be found in the text by Marple [19].

## 9.8 THE CEPSTRUM

Often signals are combined using operations of multiplication and convolution. The system models, representing the above operations, are examples of a class of systems called **homomorphic systems**, which are discussed in the next section (§9.9). It is shown that the output signal from a homomorphic system, for which the response of the system is a convolution of input signals, can be deconvolved using a process that involves determination of a function known as the **cepstrum** (this terminology is derived from the word 'spectrum').

Let $x(k)$ denote the convolution of two real signals $x_1(k)$ and $x_2(k)$, that is $x(k) = (x_1 * x_2)(k)$. Then, provided the corresponding **z**-transforms exist,

$$X(z) = X_1(z)X_2(z) \ .$$

*Remark*: In this section and subsequent sections, all references to **z**-transforms implicitly imply the use of the bilateral **z**-transform.

Using the complex logarithm, the product term can be separated to give

$$\log_e(X(z)) = \log_e(X_1(z)) + \log_e(X_2(z)) \ ,$$

where $\log_e(X(z)) = \log_e|X(z)| + j \arg(X(z))$. If the corresponding inverse **z**-transforms exist, the resulting sequence takes the form:

$$\{c_x^{(c)}(k)\} \stackrel{\text{def}}{=} \mathcal{Z}^{-1}[\log_e(X(z))] \ ,$$

where $c_x^{(c)}(k) = c_{x_1}^{(c)}(k) + c_{x_2}^{(c)}(k)$ and the superscript $^{(c)}$ notation denotes that the *complex* logarithm has been used to generate the sequence. Here, $c_x^{(c)}(k)$ is known as the **complex cepstrum** of $x(k)$. If $\log_e(X(z))$ has a convergent power series representation, with region of convergence

$$\{z \in \mathbb{C} : r_1 < |z| < r_2, \ 0 < r_1 < 1 \text{ and } r_2 > 1\} \ ,$$

then it can be shown (for more details see Oppenheim and Schafer [21]) that

$$\{c_x^{(c)}(k)\} = \frac{1}{2\pi}\int_{-\pi}^{\pi} \log_e(X(e^{j\,\theta}))e^{j\,\theta k}\,\mathrm{d}\theta \ , \tag{9.25}$$

which can be identified as an IDTFT as shown in §7.3, Equation (7.13b). With reference to (9.25), adding a multiple of $2\pi$ to $\arg X(z)$ does not produce a different sequence $\{c_x^{(c)}(k)\}$. However, if $\arg X(e^{j\,\theta})$ is not restricted to its principal values and it is assumed to be a continuous function of $\theta$, then, for each value of $\theta$, the sequence $\{c_x^{(c)}(k)\}$ is unique. If the DFT is used for the computation of the Fourier spectrum of $\{x(k)\}$, then sample values of $\arg X(e^{j\,\theta})$ contain only principal values (that is values between $-\pi$ and $\pi$). These are known as **wrapped** values. For continuity of $\arg X(e^{j\,\theta})$, the phase of $X(e^{j\,\theta})$ must be **unwrapped** before $\{c_x^{(c)}(k)\}$ can be computed. More details can be found in the texts by Oppenheim and Schafer [21] and Rabiner and Schafer [27].

Before presenting some properties of the complex cepstrum, we introduce the concept of a stable sequence and definitions of minimum-phase and maximum-phase sequences.

### *9.8.1 Stable sequences*

Recall from Chapter 6, §6.2, that a system is (BIBO) stable if its impulse response sequence $\{y^\delta(k)\}$ decays to zero, which implies that its impulse response is absolutely summable, *i.e.* $\sum_{k=0}^{\infty} |y^\delta(k)| < \infty$. This motivates the definition of a **stable sequence** $\{x(k) : k \in \mathbb{Z}\}$ as *a sequence which is absolutely summable*, that is $\sum_{k=-\infty}^{\infty} |x(k)| < \infty$.

Consider a sequence $\{x\}$ with **z**-transform $X(z)$. Depending on the poles of $X(z)$, there may be several regions of convergence, that is regions for which the power series defining $X(z)$ converges. For example, the three sequences

$$(i)\ x(k) = \begin{cases} 0\ , & k \geq 0 \\ \frac{2}{5}\left[(-2)^k - (\frac{1}{2})^k\right] & , k < 0 \end{cases} \qquad (ii)\ x(k) = \begin{cases} \frac{2}{5}(\frac{1}{2})^k\ , & k \geq 0 \\ \frac{2}{5}(-2)^k & , k < 0 \end{cases}$$

$$(iii)\ x(k) = \begin{cases} \frac{2}{5}\left[(\frac{1}{2})^k - (-2)^k\right] & , k \geq 0 \\ 0\ , & k < 0 \end{cases}$$

have **z**-transform $X(z) = \dfrac{z}{(z-\frac{1}{2})(z+2)}$, but with different regions of convergence:

$$(i)\ \{z \in \mathbb{C} : |z| < \tfrac{1}{2}\} \qquad (ii)\ \{z \in \mathbb{C} : \tfrac{1}{2} < |z| < 2\} \qquad (iii)\ \{z \in \mathbb{C} : |z| > 2\}\ ,$$

respectively. Since the absolute summability condition is equivalent to the condition

$$\sum_{k=-\infty}^{\infty} |x(k)z^{-k}| < \infty \quad \text{with } |z| = 1\ ,$$

the sequence $\{x\}$ is stable if the region of convergence of $X(z)$ includes the unit circle. Thus, for the above example, with regions $(i)$ $\{z \in \mathbb{C} : |z| < \frac{1}{2}\}$ and $(iii)$ $\{z \in \mathbb{C} : |z| > 2\}$, the corresponding sequence is unstable. However, the sequence associated with $X(z)$ and region of convergence $\{z \in \mathbb{C} : \frac{1}{2} < |z| < 2\}$ is stable.

Note that a causal sequence $\{x\}$ always has a region of convergence of the form $\{z \in \mathbb{C} : |z| > r_p\}$, where $r_p \geq |z_p|$ and $|z_p|$ is the greatest magnitude of the poles of $X(z)$. Therefore, if a sequence $\{x\}$ is both causal and stable, then all the poles of $X(z)$ must lie inside the unit circle of the $z$-plane.

### *9.8.2 Minimum-phase and maximum-phase sequences*

As stated in §9.8.1, if a real sequence $\{x\}$ is causal and stable, then all the poles of $X(z)$ must lie inside the unit circle. For some problems it may be desirable that the sequence with **z**-transform $[X(z)]^{-1}$ is also stable and causal. Since the poles of $[X(z)]^{-1}$ are the zeros of $X(z)$, this implies that the zeros of $X(z)$ must also lie inside the unit circle. A sequence with this property is known as a **minimum-phase sequence**, that is *a minimum-phase sequence is a causal sequence whose* **z***-transform is such that its poles and zeros are inside the unit circle in the z-plane.* An explanation of the derivation of the name 'minimum-phase' is given in

Oppenheim and Schafer [21]. An example of a minimum-phase sequence is given by the unilateral **z**-transform $X(z) = (1 - \frac{1}{4}z^{-1})(1 + \frac{1}{2}z^{-1})$. In this case $X(z)$ has poles $z = 0,\ 0$ and zeros $z = \frac{1}{4},\ -\frac{1}{2}$. The sequence $\{x\}$ is specified by $x(0) = 1$, $x(1) = \frac{1}{4}$, $x(2) = -\frac{1}{8}$, $x(k) = 0$ otherwise, which can be expressed in the form

$$x(k) = \delta(k) + \left(-\tfrac{1}{2}\right)^{k+1} [\zeta(k-1) - \zeta(k-3)] \ .$$

Clearly, the sequence is causal and stable.

A **maximum-phase sequence** is *a stable sequence whose* **z***-transform is such that its poles and zeros are outside the unit circle in the z-plane.* Such a sequence is **anticausal**, that is a sequence $\{x\}$ with the property that $x(k) = 0$ for all $k > 0$. As an example, consider the sequence defined by

$$x(k) = 25\delta(k) + 4(4k+9)[\zeta(-k-1) - \zeta(-k-3)] \ ,$$

*i.e.* $x(-2) = 4$, $x(-1) = 20$, $x(0) = 25$, $x(k) = 0$ otherwise. This sequence is stable and, since $X(z) = (5 + 2z)^2$ has zeros $z = -\frac{5}{2},\ -\frac{5}{2}$ and no poles, is a maximum-phase sequence.

***Properties of the complex cepstrum***

Consider stable sequences with rational **z**-transforms and no poles or zeros on the unit circle. For these sequences, the following properties hold (for a detailed analysis, see, for example, Oppenheim and Schafer [21]).

1. The complex cepstrum of a real sequence is a real sequence.
2. $\{c_x^{(c)}(k)\}$ is of infinite duration, even if $\{x(k)\}$ has finite duration.
3. $c_x^{(c)}(k)$ decays at least as fast as $\dfrac{1}{|k|}$.
4. $\{c_x^{(c)}(k)\}$ is a causal sequence if and only if $\{x(k)\}$ is minimum-phase.
5. $c_x^{(c)}(k) = 0$ for $k > 0$ if and only if $\{x(k)\}$ is a maximum-phase sequence.

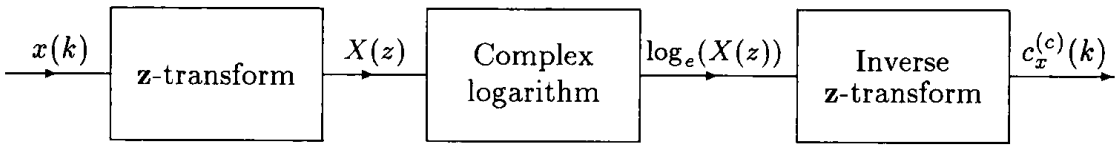



Figure 9.14: Process to obtain the complex cepstrum.

The processes required for the formation of $c_x^{(c)}(k)$ are illustrated diagrammatically in Figure 9.14 or, more simply as shown in Figure 9.15, where $\mathcal{C}_*$ is known as the **characteristic system**.

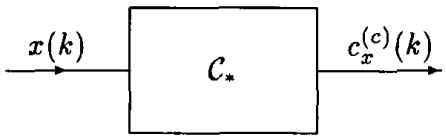



Figure 9.15: Characteristic system.

In some applications, the phase of $X(e^{j\theta})$ is unimportant, in which case the **real cepstrum**, $c_x^{(r)}(k)$, defined by

$$\{c_x^{(r)}(k)\} \stackrel{\text{def}}{=} \frac{1}{2\pi}\int_{-\pi}^{\pi} \log_e |X(e^{j\theta})| e^{j\theta k}\, \mathrm{d}\theta \ ,$$

is more useful. The relationship between $c_x^{(c)}(k)$ and $c_x^{(r)}(k)$ is given by

$$c_x^{(r)}(k) = \frac{1}{2}[c_x^{(c)}(k) + c_x^{(c)}(-k)] \ . \tag{9.26}$$

For real signals, $c_x^{(r)}(k)$ is the even part of $c_x^{(c)}(k)$ and, hence, $c_x^{(r)}(k)$ is an even function.

## 9.9 HOMOMORPHIC SYSTEMS AND DECONVOLUTION

**Homomorphic systems** are systems, with input-output transformation $\mathcal{T}$, that satisfy a generalized version of an application of the principle of superposition (see §1.4) in the form

$$\mathcal{T}\{u(k) \diamond v(k)\} = \mathcal{T}\{u(k)\} \circ \mathcal{T}\{v(k)\} \ ,$$

where $\diamond$ and $\circ$ represent rules for combining inputs and outputs, respectively. It is known that such systems can be represented by algebraically linear (homomorphic) mappings between input and output signal spaces; for a lucid explanation of homomorphic systems, see Oppenheim and Schafer [21].

As discussed in the previous section, if the input to the characteristic system is the convolution $(x_1 * x_2)(k)$, then the output is $c_x^{(c)}(k) = c_{x_1}^{(c)}(k) + c_{x_2}^{(c)}(k)$. This is an example of a homomorphic system, where $\diamond$ represents the convolution operation and $\circ$ represents addition. This property of the characteristic system may be utilized in an application to **deconvolution** of a signal, namely deconvolving (or separating) the signal into its constitutive components.

If the sequences $\{c_{x_1}^{(c)}(k)\}$ and $\{c_{x_2}^{(c)}(k)\}$ have non-overlapping regions of support, that is $c_{x_1}^{(c)}(k) = 0$ for all $k$ satisfying $c_{x_2}^{(c)}(k) \neq 0$, then a linear filter, $\{\ell(k)\}$, can be constructed so that

$$\ell(k) c_x^{(c)}(k) = c_{x_1}^{(c)}(k) \qquad [\text{ or, if desired, } \quad c_{x_2}^{(c)}(k)] \ .$$

Processing $c_{x_1}^{(c)}(k)$ (or $c_{x_2}^{(c)}(k)$) through the inverse characteristic system $\mathcal{C}_*^{-1}$, we are then able to recover $x_1(k)$ (or $x_2(k)$) from $x(k)$. Such filters are known as linear,

**frequency-invariant filters**. For example, if it is known that $x_1(k)$ is a minimum-phase sequence and $x_2(k)$ is a maximum-phase sequence with $x_2(0) = 0$, then a linear, frequency-invariant filter of the form:

$$\ell(k) = \zeta(-k-1) = \begin{cases} 1, & k < 0 \\ 0, & k \geq 0 \end{cases}$$

would remove $c_{x_1}^{(c)}(k)$ from $c_x^{(c)}(k)$, leaving $c_{x_2}^{(c)}(k)$. Hence, $x_2(k)$ can be recovered from $x(k)$. A fuller discussion of frequency invariant filters is given in Oppenheim and Schafer [21]. Thus, homomorphic deconvolution may be represented diagramatically by Figure 9.16. For the practical implementation of $\mathcal{C}_*$ or $\mathcal{C}_*^{-1}$, the operations of z-transform and inverse z-transform are replaced by DFT and inverse DFT, respectively.

A block diagram representation of $\mathcal{C}_*^{-1}$ is shown in Figure 9.17.

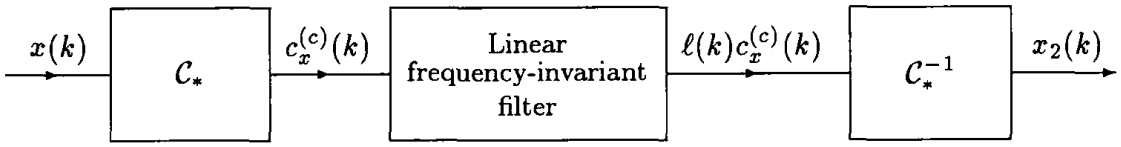



Figure 9.16: Homomorphic deconvolution.

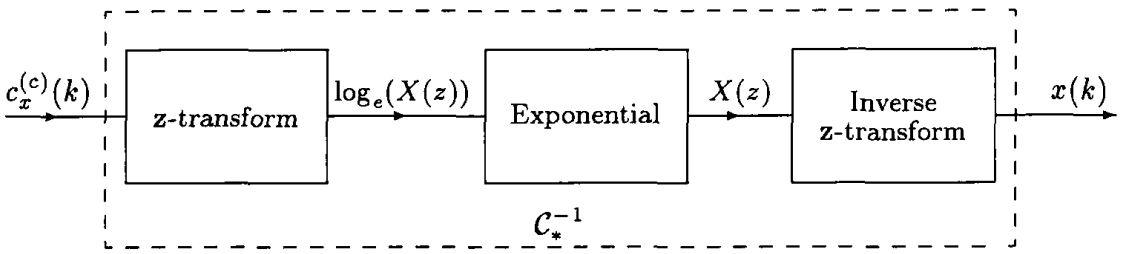



Figure 9.17: Inverse of characteristic system for homomorphic deconvolution.

## 9.10 CEPSTRAL ANALYSIS OF SPEECH SIGNALS

A simple model of the vocal tract system function is

$$V(z) = \frac{\sum_{n=0}^{N} \beta_n z^{-n}}{\sum_{m=0}^{M} \alpha_m z^{-m}}.$$

Since both numerator and denominator can be expressed as a product of linear factors, then, excluding poles and zeros on the unit circle, $V(z)$ may be rewritten in the form:

$$V(z) = \frac{Az^{-k_0}\prod_{n=1}^{N_1}(1-a_n z^{-1})\prod_{n=1}^{N_2}(1-b_n z)}{\prod_{m=1}^{M_1}(1-c_m z^{-1})\prod_{m=1}^{M_2}(1-d_m z)},$$

where $|a_n|$, $|b_n|$, $|c_m|$, $|d_m| < 1$ for all $n$, $m$. Hence, excluding $z = 0$ and $z = \infty$, there are $M_1$ poles and $N_1$ zeros inside the unit circle and $M_2$ poles and $N_2$ zeros outside the unit circle. It is assumed that the real constant $A$ is positive. Now

$$\log_e(V(z)) = \log_e(A) + \log_e(z^{-k_0}) + \sum_{n=1}^{N_1}\log_e(1-a_n z^{-1}) + \sum_{n=1}^{N_2}\log_e(1-b_n z) - \sum_{m=1}^{M_1}\log_e(1-c_m z^{-1}) - \sum_{m=1}^{M_2}\log_e(1-d_m z) .$$

The term $z^{-k_0}$ represents an advance or delay of the sequence $\{v(k)\}$ and contributes only to the imaginary part of $\log_e(V(e^{j\theta}))$. This term may be removed by either shifting the time origin of $\{v(k)\}$ or by subtracting the contribution to $\log_e(V(e^{j\theta}))$. Thus, we consider

$$\log_e(V(z)) = \log_e(A) + \sum_{n=1}^{N_1}\log_e(1-a_n z^{-1}) + \sum_{n=1}^{N_2}\log_e(1-b_n z) - \sum_{m=1}^{M_1}\log_e(1-c_m z^{-1}) - \sum_{m=1}^{M_2}\log_e(1-d_m z) ,$$

and assume that the region of convergence contains the unit circle. Hence, using the following two **z**-transform (bilateral) results:

$$\mathcal{Z}\left[\left\{-\frac{\alpha^k}{k}\zeta(k-1)\right\}\right] = \log_e(1-\alpha z^{-1}) , \quad |z| > |\alpha|, \tag{9.27}$$

and

$$\mathcal{Z}\left[\left\{\frac{\beta^{-k}}{k}\zeta(-k-1)\right\}\right] = \log_e(1-\beta z) , \quad |z| < |\beta|^{-1}, \tag{9.28}$$

and, since $\{c_v^{(c)}(k)\} = \mathcal{Z}^{-1}[\log_e(V(z))]$, $c_v^{(c)}(k)$ may be expressed as

$$c_v^{(c)}(k) = \log_e(A)\delta(k) - \frac{1}{k}\sum_{n=1}^{N_1}(a_n)^k\zeta(k-1) + \frac{1}{k}\sum_{n=1}^{N_2}(b_n)^{-k}\zeta(-k-1) + \frac{1}{k}\sum_{m=1}^{M_1}(c_m)^k\zeta(k-1) - \frac{1}{k}\sum_{m=1}^{M_2}(d_m)^{-k}\zeta(-k-1) ,$$

or

$$c_v^{(c)}(k) = \log_e(A)\delta(k) + \frac{1}{k}\left[\sum_{m=1}^{M_1}(c_m)^k - \sum_{n=1}^{N_1}(a_n)^k\right]\zeta(k-1) + \frac{1}{k}\left[\sum_{n=1}^{N_2}(b_n)^{-k} - \sum_{m=1}^{M_2}(d_m)^{-k}\right]\zeta(-k-1) .$$

*Remarks*:

(a) For $k \neq 0$, each term of $c_v^{(c)}(k)$ is a product of $\frac{1}{k}$ and a decaying exponential power and, hence, decays at least as fast as $\frac{1}{|k|}$ (see Property 3 of §9.8.2).

(b) If $\{v(k)\}$ is a minimum-phase sequence (namely $b_m = d_m = 0$ for all $m$), then its complex cepstrum is causal (see Property 4 of §9.8.2). For voiced speech, the contribution of the vocal tract will, in general, be non-mimimum-phase and thus the complex cepstrum will be nonzero for all $k$.

(c) The contribution of the vocal tract response to the complex cepstrum is mainly concentrated around the origin.

To model voiced speech, a window which tapers smoothly to zero at both ends, such as a Hamming window (see §8.6.2), is used to minimize end effects due to impulses that occur before the beginning of the analysis interval. We assume that the windowed speech segment, say $\{s(k)\}$, is modelled by

$$s(k) = (v * p_w)(k) ,$$

where $p_w(k) = w(k)p(k)$, $\{w(k)\}$ is a window sequence that tapers smoothly to zero at both ends, and $\{p(k)\}$ is the input sequence. For voiced speech, $p(k)$ is chosen to be a train of delayed and weighted impulses. The complex cepstrum of $\{s(k)\}$ is given by

$$c_s^{(c)}(k) = c_v^{(c)}(k) + c_{p_w}^{(c)}(k)$$

and, since the complex cepstrum of the vocal tract response is mainly concentrated around the origin, a frequency-invariant filter can be used to isolate $c_v^{(c)}(k)$ or $c_{p_w}^{(c)}(k)$.

*Example 9.5*
Using the results:

$$\sum_{r=0}^{n} q^r = \frac{1-q^{n+1}}{1-q} , \quad q \neq 1 , \tag{9.29}$$

$$\log_e(1-\alpha z^{-1}) = -\sum_{r=1}^{\infty}\frac{\alpha^r}{r}z^{-r} , \quad |z| > |\alpha| , \tag{9.30}$$

and given $p(k)$ has the form:

$$p(k) = \delta(k) + a\delta(k-\tau) + a^2\delta(k-2\tau) + a^3\delta(k-3\tau) ,$$

with $0 < a < 1$ and $\tau \in \mathbb{N}$, determine $c_p^{(c)}(k)$.

Now,

$$p(k) = \sum_{i=0}^{3} a^i \delta(k - i\tau) \ .$$

Thus, invoking (9.29),

$$\begin{aligned} P(z) &= \sum_{i=0}^{3} a^i z^{-i\tau} \\ &= \frac{1 - (az^{-\tau})^4}{1 - az^{-\tau}} \end{aligned}$$

and, therefore, utilizing (9.30) we have

$$\begin{aligned} \log_e(P(z)) &= \log_e\left(1 - a^4 z^{-4\tau}\right) - \log_e\left(1 - az^{-\tau}\right) \\ &= \sum_{r=1}^{\infty} \left( -\frac{a^{4r}}{r} z^{-4\tau r} + \frac{a^r}{r} z^{-\tau r} \right) \ . \end{aligned}$$

Hence,

$$c_p^{(c)}(k) = \sum_{r=1}^{\infty} \frac{1}{r} \left[ a^r \delta(k - \tau r) - a^{4r} \delta(k - 4\tau r) \right] \ ,$$

that is a train of impulses at multiples of $\tau$.

□

In the above example, in the context of a speech signal, $\tau$ would represent the pitch period. When $\{p(k)\}$ consists of an impulse train with impulses spaced at intervals of $\tau$, Oppenheim and Schafer [21] (see §12.9.1, Chapter 12) show that impulses, also spaced at intervals of $\tau$, appear in $\{c_{p_w}^{(c)}(k)\}$. As

$$c_s^{(c)}(k) = c_v^{(c)}(k) + c_{p_w}^{(c)}(k) \ ,$$

and since the vocal tract component of the complex cepstrum decays rapidly, $c_{p_w}^{(c)}(k)$ will be the dominant contribution to $c_s^{(c)}(k)$ for $k$ sufficiently large. Thus, peaks in the sequence $\{c_{p_w}^{(c)}(k)\}$ will surface in the sequence $\{c_s^{(c)}(k)\}$. This suggests that the pitch period can be estimated by determining the difference between two consecutive peaks in $\{c_s^{(c)}(k)\}$ for sufficiently large $k$. This technique is illustrated using the real cepstrum of the data used in §9.3. The real cepstrum may be determined in MATLAB using the function m-file `rceps(x)`. If this function m-file is not available in your version of the MATLAB Signal Processing Toolbox, use the MATLAB command: `y= real(ifft(log(abs(fft(x)))))`. The graph of the real cepstrum sequence is illustrated in Figure 9.18. Note that the voiced speech signal is not truly periodic and it will include some noise. Also, the cepstrum decays rapidly, as indicated by property 3 of §9.8. For these reasons, it is difficult to identify peaks in the graph of the cepstrum. One way to partly overcome this problem is to apply a frequency-invariant filter to the cepstrum in order to remove some

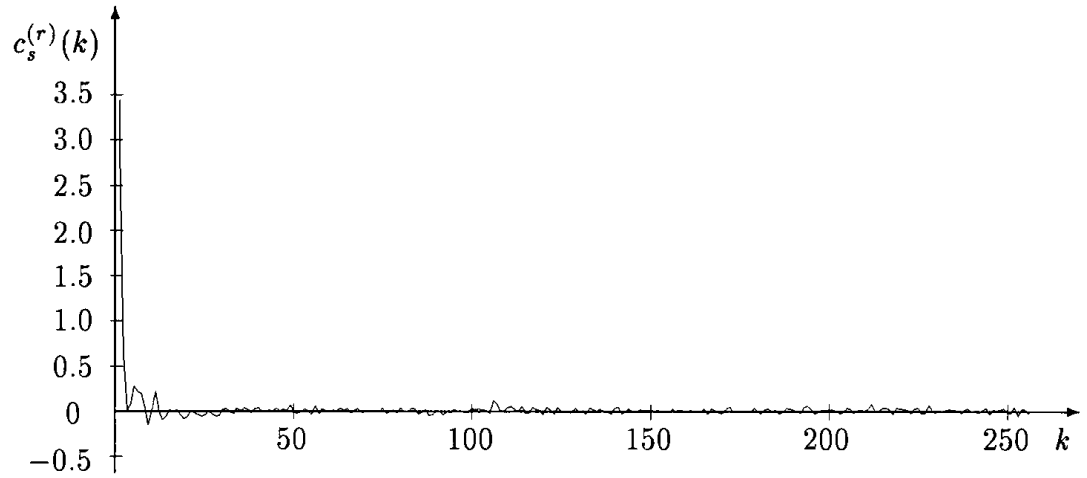



Figure 9.18: Real cepstrum incorporating vocal tract component.

of the vocal tract contribution with relatively large magnitude. Performing this technique with the frequency-invariant filter $\ell(k) = \zeta(k-6)$, a plot of the resulting modified real cepstrum is shown in Figure 9.19. A peak occurring near 110 is more easily distinguished; however, a dominant peak in the approximate range 210-220 is still difficult to locate. As indicated above, a linear, frequency-invariant filter

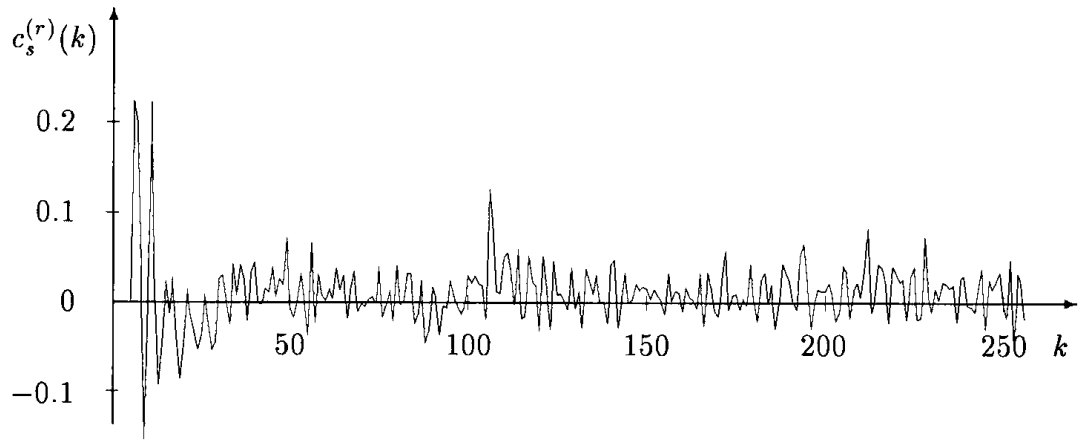



Figure 9.19: Real cepstrum with peaks corresponding to pitch period.

can be used to isolate either the vocal tract contribution or periodic component. The result is processed by the inverse characteristic system to recover the desired component.

## 9.11 FORMANT ESTIMATION USING CEPSTRAL ANALYSIS

A strongly periodic component in the log spectrum at a frequency $T^{-1}$, where $T$ is the pitch period, become periodic peaks in the cepstrum. In cepstral analysis, the

independent variable in the cepstrum has units of time and is referred to as **quefrency** (this terminology being derived from the word 'frequency'). For most voiced speech, a clear distinction can be made between the periodic excitation component in the cepstrum and the vocal tract contribution which appears as components at low values of quefrency.

A window function, that is a low quefrency **lifter** (**liftering** is the process of filtering in the quefrency domain), can be used to remove the high quefrency periodic excitations. The DFT of the resulting signal then gives a smoothed version of the log spectrum of the vocal tract filter. By investigating the peaks in the cepstrally smoothed log spectrum, the frequency, bandwidth and amplitude of the formants can be estimated from the spectral envelope. For more details on this technique see, for example, Deller *et al* [5] and Oppenheim and Schafer [21].

## 9.12 EXERCISES

1. Evaluate the autocorrelation function

$$R_{xx}(\tau) = \int_{-\infty}^{\infty} x(s)x(s+\tau)\,\mathrm{d}s$$

for the deterministic, finite-energy signal $x(t) \stackrel{\text{def}}{=} e^{-t}\zeta(t)$.

2. Test data for non-nasal, voiced speech can be constructed by passing an impulse train through a stable filter. For example, the MATLAB commands:

```
e = eye(16,16);              % generates 16 by 16 identity matrix
x = e(5:254);                % forms impulse train
a = [1 -0.2 -0.23 -0.05 -0.12];
b = [1];
s = filter(b,a,x);           % output from the stable filter
```

produce such a signal, with 250 data values and 14 impulses every 16 units. To make the signal more realistic, additive 'noise' may be added with the MATLAB command: `xs = s+k*randn(size(s))`, with $k$ suitably chosen (for example, $k = 1.2$). For the signal $xs$, use MATLAB to evaluate and plot $xs$ and its autocorrelation function. From the graph, estimate the periodicity in the signal $s$.

3. Given the autocorrelation matrices:

$$R_1 = \begin{bmatrix} 3 & 1 \\ 1 & 3 \end{bmatrix}, \qquad R_2 = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{4} \\ \frac{1}{2} & 1 & \frac{1}{2} \\ \frac{1}{4} & \frac{1}{2} & 1 \end{bmatrix}$$

and the equations

$$R_1\boldsymbol{\alpha} = \begin{bmatrix} 5 \\ 3 \end{bmatrix}, \qquad R_2\boldsymbol{\beta} = \begin{bmatrix} \frac{1}{2} \\ \frac{1}{4} \\ \frac{1}{8} \end{bmatrix},$$

show that the solutions obtained by an exact method and Durbin's algorithm are identical.

4. Given the data set $\{1\ 2\ 4\}$, determine the reflection coefficients $k_1$ and $k_2$ using Burg's algorithm and confirm that $|k_i| \leq 1$.

5. Determine the predictor coefficients for a linear predictor of order 2 given the normal equations:

$$\begin{bmatrix} 1 & 4 \\ 4 & 25 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix} = \begin{bmatrix} 5 \\ 2 \end{bmatrix} ,$$

using (a) the Cholesky decomposition algorithm
and (b) an exact method.

6. Suppose an estimate of a signal is required based on past data $x(k)$. If the signal is modelled by the linear representation

$$\tilde{x}(k) = \sum_{i=1}^{3} a_i x(k-i) ,$$

use Durbin's algorithm to obtain the estimate $\tilde{x}(k)$ assuming that the elements of the autocorrelation matrix for the process $x(k)$ are given by

$$r(i,k) = r(|i-k|) = \alpha^{|i-k|} ,$$

where $\alpha$ is a real constant satisfying $|\alpha| < 1$.

7. Suppose a signal $x(k)$ is estimated using the AR model

$$\hat{x}(k) = \sum_{i=1}^{p} \alpha_i x(k-i) .$$

By applying Burg's method, determine the parameters $\alpha_i$ with given data $\{x(1)\ x(2)\ x(3)\} = \{0\ \ 1\ \ a\}$, where $a \neq 0$ is a real constant, for the case $p = 2$.

Write down the estimate $\hat{x}(k)$ when $p = 1$.

8. If the sequence $\{w(k)\}$ is related to the sequences $\{x(k)\}$ and $\{y(k)\}$ by the relationship

$$w(k) = 100(x * y)(k) ,$$

where $*$ denotes convolution, obtain an expression for the complex cepstrum of $\{y(k)\}$.

9. Using the z-transform property (9.27), determine (a) the complex cepstrum and (b) the real cepstrum of $\{x(k)\}$, given by

$$x(k) = 2\delta(k) - 2\delta(k-1) + \frac{1}{2}\delta(k-2) .$$

10. If $p(k)$ is the train of impulses

$$p(k) = \sum_{m=0}^{M-1} a^m \delta(k - mN), \qquad N > 0 \text{ and } M > 1,$$

find $c_p^{(c)}(k)$, the complex cepstrum of $p(k)$, using the results (9.29) and (9.30). If $M = 3$, $N = 10$ and $a = \frac{1}{2}$, determine whether the sequence $\{p(k)\}$ is minimum-phase and sketch $c_p^{(c)}(k)$ for $-80 \leq k \leq 80$.

11. Using the test data for non-nasal, voiced speech $\{s\}$ generated in Exercise 2 above, introduce noise in the signal by convolving the sequence with a random noise sequence. The MATLAB command: `cs = conv(s,randn(size(s)))` will achieve this objective. For the signal $cs$, use MATLAB to evaluate and plot its real cepstrum. Using an appropriate frequency-invariant filter, estimate the periodicity in the signal. Does the value, so obtained, agree with your expectations?

12. If $u(k) = K\alpha^k \zeta(k)$, where $K$, $\alpha$ are real constants satisfying $K > 0$ and $|\alpha| < 1$, show, using (9.27), that

$$c_u^{(c)}(k) = \frac{\alpha^k}{k} \zeta(k-1) + \log_e(K)\delta(k) \ .$$

Suppose the impulse response of a linear, time-invariant filter satisfies

$$(h * u)(k) = \delta(k) + \beta\delta(k-1) \ ,$$

where $\beta \in \mathbb{R}$ satisfies $0 < \beta < 1$. Find $c_h^{(c)}(k)$ and, hence, determine whether the filter is minimum-phase.

13. Show that the complex cepstrum, $c_x^{(c)}(k)$, of a maximum-phase signal is related to its real cepstrum, $c_x^{(r)}(k)$, by

$$c_x^{(c)}(k) = c_x^{(r)}(k)\ell(k) \ ,$$

where $\{\ell(k)\}$ is the frequency-invariant, linear filter given by

$$\ell(k) \stackrel{\text{def}}{=} 2\zeta(-k) - \delta(k) \ .$$

# Index


Absolutely summable sequence, 257
Aliasing, 99, 176, 189, 190, 193, 194
All-pole filter/model, 233, 246, 249
Amplitude
    complex exponential, 59
    harmonic, 71
Amplitude modulation (AM), 82, 86
Amplitude response
    continuous-time, 60
    discrete-time, 174
Amplitude spectrum
    continuous-time, nonperiodic, 79
    continuous-time, periodic, 71
    discrete-time, nonperiodic, 186
    discrete-time, periodic, 182
    vocal tract, 250
Analogue filter, 105
Anticausal sequence, 258
Autocorrelation
    energy signal, 235
    function, 235–240, 251
    matrix, 240, 242
    method, 244
    normalized sequence, 238
    power signal, 234
    properties, 235
    short-time, 237
Autoregressive filter/model, 233, 240
Average power, *see* Power of a signal

Backward prediction error sequence, 252
Band limit, 99
Bandwidth, 99, 131
BIBO stability
    continuous-time, 42, 43
    discrete-time, 164, 165
Bilinear transform method, 214–218
Block diagram, 13
Burg's algorithm, 254–255, 279–280
Butterworth filter, 108–126, 209, 213, 217, 218
Butterworth polynomial, 113
    table of coefficients, 118

$C$–$R$ circuit, 6, 12–13, 25–26
Capacitor, 4
Carrier signal, 82, 86
Causal sequence, 142, 257
Causal signal, 15
Cepstral analysis, *see* Cepstrum
Cepstrum, 256–259
    cepstral analysis, 260–265
    complex, 256
    real, 259, 264
    unwrapped phase, 256
    wrapped phase, 256
Characteristic polynomial, 31, 38, 42
Characteristic system, 258
    inverse, 259
Chebyshev filter, 134–139
Chebyshev polynomial, 134
    table, 134
Cholesky decomposition algorithm, 245, 277–278
Comb signal, 95
Complex cepstrum, 256
Complex exponential, 268–275
    amplitude, 59
    argument, 272
    frequency, 59
    modulus, 272
    phase, 59
Complex exponential form, 272
Continued fraction expansion, 124
Convolution
    continuous-time, 24, 37, 50–58
    discrete-time, 169–172
    frequency domain, 96
Correlation function, 233–235
    energy signals, 234
    power signals, 234
Covariance
    matrix, 245
    method, 244

Critical frequency, 105
Cross-correlation
energy signals, 234
function, 234, 237
power signals, 234
properties, 237
Cut-off frequency, 105

Dc component, 235
Decibel (dB), 59, 137, 250
Delay block, 146
DFT, *see* Discrete Fourier transform
Difference equation, 143, 144, 156
Digital filter, 206–230
Digital filter design
bilinear (Tustin's) method, 214–218
direct design method, 219–222
impulse invariant method, 206–210
step invariant method, 210–214
Dirac $\delta$-function, *see* Impulse function
Direct design method, 219–222
Dirichlet conditions
nonperiodic signal, 76
periodic signal, 68
Discrete Fourier transform, 179–184
Discrete-time Fourier transform, 184–186
estimation of, 186–190
Discrete-time signal, *see* Sequence
Distribution, *see* Generalized function
DTFT, *see* Discrete-time Fourier transform
Durbin's algorithm, 242, 251, 255, 276–277

Energy of a signal, 88
Energy signal, 90
correlation function, 234–235
Envelope, 249
spectrum, 250, 265
Euler's formula, 269
Even function, 81

Fast Fourier transform, 183, 196–203
FFT, *see* Fast Fourier transform
FFT algorithm, 250
Filter
all-pole, 233, 246, 249
analogue, 105
autoregressive, 233, 240
Butterworth, 108–126, 209, 213, 217, 218
Chebyshev, 134–139
digital, 206–230
finite impulse response (FIR), 222
frequency-invariant, 260, 263–264
ideal band-pass, 107
ideal band-reject, 107
ideal high-pass, 105
ideal low-pass, 99, 105
inverse, 233
lattice, 251–255
linear prediction, 240–247
non-recursive, 222
vocal tract, 246
Filter transformations, 107, 127–132
First shift theorem, 24
Formant, 232, 249
estimation of, 249–251, 264
Forward prediction error sequence, 252
Fourier series, 67–73
estimation of, 190–191
Fourier spectrum, 79
Fourier transform, 73–78
discrete, 179–184
discrete-time, 184–186
estimation of, 192–196
fast, 183, 196–203
generalized, 92
properties, 81–88
table, 78
Frequency of complex exponential, 59
Frequency resolution, 193
Frequency response
continuous-time, 58–65
discrete-time, 172–176
Frequency-invariant filter, 260, 263–264
Fricatives, 232
Fundamental frequency, 68
Fundamental period, 232

Gain spectrum, *see* Amplitude response
Generalized function, 16–22
differentiation, 20
equivalence, 19
Geometric centre frequency, 131
Gibb's phenomenon, 221, 225
Glottal shaping model, 233, 246
Glottis, 231

Heaviside unit step function, 15, 48
Hertz (Hz), 59
Homomorphic
deconvolution, 259–260
system, 256, 259–260

Impulse function, 17, 91
Impulse invariant method, 206–210
Impulse response
    continuous-time, 44–48
    discrete-time, 166
Impulse sequence, 164
Impulse train, 52, 95, 161
Inductor, 4
Inverse characteristic system, 259
Inverse filter, 233

Kirchhoff's laws, 5

*L–C* ladder network, 120–126
*L–C–R* circuit, 4–9, 27–30
*L–R* circuit, 5
Laplace transfer function, 26, 30–36
Laplace transform, 15–16
    bilateral, 15
    modified definition, 22
    properties, 22–25
    table, 16
    unilateral, 15
Lattice filter, 251–255
Lifter, 265
Liftering, 265
Linear difference equation, 145
Linear operator, 9, 10, 145
Linear prediction, 247, 249
    filter, 240–247
Linear predictive coding, 240
Linear predictor, 241, 247
Linear system, 5, 9–12
Lip radiation model, 233, 246, 251
Log spectrum, 250, 265

Marginal stability
    continuous-time, 42, 43
    discrete-time, 165
Maximally-flat property, 119
Maximum entropy method, 254
Maximum-phase sequence, 257, 260
Mean value theorem
    differential calculus, 52
    first for integrals, 18
Minimum-phase sequence, 257, 260, 262

Nasal tract, 231
Nonperiodic signal, 73
Normal equations, 242, 244
Normalized autocorrelation sequence, 238
Normalized frequency, 172, 182
Nyquist condition, 99
Nyquist frequency, 99
Nyquist rate, 99

Odd function, 81
Orthogonality, 69

Parseval's formula, 104, 204
Periodic sequence, 179
Periodic signal, 67
Phase
    complex exponential, 59
    harmonic, 71
Phase response
    continuous-time, 60
    discrete-time, 174
Phase spectrum
    continuous-time, nonperiodic, 79
    continuous-time, periodic, 71
    discrete-time, nonperiodic, 186
    discrete-time, periodic, 182
Phase-shift spectrum, *see* Phase response
Pitch, 232
    period, 232, 247, 249, 263
Plosive sounds, 232
Poles, 42
Polynomial
    Butterworth, 113
    Butterworth–table of coefficients, 118
    characteristic, 31, 38, 42
    Chebyshev, 134
    Chebyshev–table, 134
Power of a signal, 89
Power signal, 90
    correlation function, 234–235
Power spectrum, 251
Pre-emphasized speech, 251
Pre-warping, 217
Prediction error sequence, 252
Predictor coefficients, 241
Principle of superposition, 10, 11
Proper transfer function, 31

Quefrency, 265

Real cepstrum, 259, 264
Realization problem, 108, 111
Reflection (PARCOR) coefficients, 252
Resistor, 4
Response
    amplitude, continuous-time, 60
    amplitude, discrete-time, 174

frequency, continuous-time, 58–65
frequency, discrete-time, 172–176
impulse, continuous-time, 44–48
impulse, discrete-time, 166
phase, continuous-time, 60
phase, discrete-time, 174
steady-state, 60, 173
step, continuous-time, 48–50
step, discrete-time, 167–169
zero-state, 11
Ripple, 134, 135, 137, 227
Ripple magnitude, 137

Sampled-data system, 162
Sampling, 2, 96, 156, 161, 182
Sampling frequency, 99
Second shift theorem, 25
Sequence, 141–143
absolutely summable, 257
anticausal, 258
average value, 235
causal, 142, 257
definition, 141
maximum-phase, 257, 260
minimum-phase, 257, 260, 262
nonperiodic, 184
periodic, 179
prediction error, 252
stable, 257
window, 249
Shift operator, 144
Sifting property, 17
Signal
band-limited, 98
causal, 15
continuous-time, 1
deterministic, 1
discrete-time, 1, *see* Sequence
energy, 90
input, 4
nonperiodic, 73
output, 4
periodic, 67
power, 90
stochastic, 1, 2
Simulation diagram
continuous-time, 12–13, 26
discrete-time, 146–147
Sinc, 78
Sinusoids, 59
Spectral leakage, 221, 225
Spectrum
envelope, 250, 265
Speech gain parameter, 233, 246, 249
estimation of, 248–249
Speech production model, 231
Stability
continuous-time, 40–44
BIBO, 42, 43
marginal, 42, 43
unstable, 42, 43
discrete-time, 164–166
BIBO, 164, 165
marginal, 165
Stable sequence, 257
State variables, 8, 32
State vector, 8
State-space representation, 9
Steady-state response, 60, 173
Step function, *see* Heaviside
Step invariant method, 210–214
Step response
continuous-time, 48–50
discrete-time, 167–169
Step sequence, 144
Strictly proper transfer function, 42
System
continuous-time, 4
definition, 4
discrete-time, 143
homomorphic, 256, 259–260
linear, 5, 9–12
time-invariant, 10

Testing function, 18
Toeplitz matrix, 242
Total energy, *see* Energy of a signal
Trachea, 231
Transfer function
Fourier, 60, 84
Laplace, 26, 30–36
proper, 31
strictly proper, 42
z, 160–161
Transform
Fourier, 73–78
discrete, 179–184
discrete-time, 184–186
fast, 183, 196–203
generalized, 92
Laplace, 15–16
z, 148–150
Tustin's method, 214–218
Twiddle factors, 199

Twiddling, 199

Unvoiced sounds, 232
Unwrapped phase, 256

Vocal tract, 231–232
    amplitude spectrum, 250
    filter, 246
    model, 233, 260–262
    resonances, 249
Voiced sounds, 231, 246

Wagon-wheel effect, 176
Window sequence, 190, 222–227, 249, 265
    Bartlett, 225
    Blackman, 225
    Hamming, 225, 262
    Hanning (von Hann), 196, 225
    rectangular, 219, 222–225
      spectrum, 224
      spectrum, main lobe, 224
      spectrum, main lobe width, 224
    table, 225
    triangular, 225
Wrapped phase, 256

Zero-state response, 11
z-transfer function, 160–161
z-transform, 148–150
    (unilateral), 149
    bilateral, 148
    properties, 150–156
    table, 153
    table of properties, 157