FREQUENCY TRACKING AND ITS APPLICATION IN SPEECH ANALYSIS

A Thesis Presented to
The Faculty of the College of Engineering and Technology
Ohio University

In Partial Fulfillment
of the Requirements for the Degree
Master of Science

by

Pian Totarong
August 1983

OHIO UNIVERSITY
LIBRARY
ACKNOWLEDGEMENT

I owe special thanks to Dr. Kogi Ogino for his guidance and assistance in the preparation of this thesis.
<table>
<thead>
<tr>
<th>CHAPTER</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABSTRACT</td>
<td>1</td>
</tr>
<tr>
<td>1 INTRODUCTION</td>
<td>2</td>
</tr>
<tr>
<td>2 FREQUENCY TRACKING BY DISCRETE FOURIER TRANSFORM (DFT)</td>
<td>5</td>
</tr>
<tr>
<td>2.1 Fixed frequency time series signal (stationary)</td>
<td>5</td>
</tr>
<tr>
<td>2.2 Locally stationary (quasi-stationary)</td>
<td>6</td>
</tr>
<tr>
<td>2.3 Rapid changing frequency time series signal</td>
<td>7</td>
</tr>
<tr>
<td>3 FREQUENCY TRACKING BY AUTOREGRESSIVE (AR) MODEL</td>
<td>9</td>
</tr>
<tr>
<td>3.1 AR modeling</td>
<td>9</td>
</tr>
<tr>
<td>3.2 Fixed frequency or locally stationary</td>
<td>10</td>
</tr>
<tr>
<td>3.3 Rapid changing frequency time series signal</td>
<td>12</td>
</tr>
<tr>
<td>3.4 Another approaching of AR model</td>
<td>14</td>
</tr>
<tr>
<td>4 COMPOSITE SINUSOIDAL MODEL (CSM)</td>
<td>20</td>
</tr>
<tr>
<td>4.1 CSM for fixed frequency time series signal</td>
<td>20</td>
</tr>
<tr>
<td>4.2 CSM for rapid time varying frequency signal</td>
<td>24</td>
</tr>
<tr>
<td>4.3 Solution of Hankel Matrix system</td>
<td>25</td>
</tr>
<tr>
<td>4.4 Solution of Vander monde matrix system</td>
<td>31</td>
</tr>
<tr>
<td>5 NUMERICAL EXAMPLES</td>
<td>32</td>
</tr>
<tr>
<td>5.1 Data types</td>
<td>32</td>
</tr>
<tr>
<td>5.2 FFT method</td>
<td>33</td>
</tr>
<tr>
<td>5.3 AR method</td>
<td>48</td>
</tr>
<tr>
<td>5.4 CSM method</td>
<td>61</td>
</tr>
<tr>
<td>5.5 How to estimate the rate of change of frequency of the rapid time varying frequency signal</td>
<td>74</td>
</tr>
</tbody>
</table>
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>FIGURE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>Obtaining the first row of the inverse of a cyclic shift right matrix</td>
</tr>
<tr>
<td>4.2</td>
<td>Updating area at each stage of recursion without using Kumar's equation</td>
</tr>
<tr>
<td>5.0</td>
<td>Real spectrum of Kay&amp;Marple</td>
</tr>
<tr>
<td>5.1(a-e)</td>
<td>FFT spectrum of Kay&amp;Marple</td>
</tr>
<tr>
<td>5.2(a-c)</td>
<td>Tracking quasi-stationary frequency by FFT</td>
</tr>
<tr>
<td>5.3(a-c)</td>
<td>Tracking rapidly changing frequency by FFT</td>
</tr>
<tr>
<td>5.4(a-e)</td>
<td>AR spectrum of Kay&amp;Marple data</td>
</tr>
<tr>
<td>5.5(a-d)</td>
<td>Tracking quasi-stationary frequency by AR</td>
</tr>
<tr>
<td>5.6(a-c)</td>
<td>Tracking rapidly changing frequency by AR</td>
</tr>
<tr>
<td>5.7(a-f)</td>
<td>CSM spectrum of Kay&amp;Marple data</td>
</tr>
<tr>
<td>5.8(a-c)</td>
<td>Tracking quasi-stationary frequency by CSM</td>
</tr>
<tr>
<td>5.9(a-c)</td>
<td>Tracking rapidly changing frequency by CSM</td>
</tr>
<tr>
<td>6.1</td>
<td>Sliding window peak picking</td>
</tr>
<tr>
<td>6.2(a-f)</td>
<td>Spectrums and spectrograms of the utterance 'we were away' from the sound spectrograph machine 700 series</td>
</tr>
<tr>
<td>6.3(a-l)</td>
<td>FFT speech spectrums</td>
</tr>
<tr>
<td>6.4(a-l)</td>
<td>FFT speech spectrograms</td>
</tr>
<tr>
<td>6.6(a-o)</td>
<td>AR speech spectrums</td>
</tr>
<tr>
<td>6.7(a-k)</td>
<td>AR speech spectrograms</td>
</tr>
<tr>
<td>6.9(a-f)</td>
<td>CSM speech spectrums</td>
</tr>
<tr>
<td>6.10(a-d)</td>
<td>CSM speech spectrograms</td>
</tr>
</tbody>
</table>
Various spectral estimation methods are investigated for the purpose of tracking the frequencies of the given time series data including sampled speech signals. Analysis and numerical results yield that discrete Fourier transformation (DFT), widely-used autoregressive modeling (AR) and unmodified composite sinusoidal modeling (CSM) fail to track rapidly changing frequencies. A chirp Z transform (CZT) is used to overcome this problem. In addition, the AR and CSM are modified to improve tracking capabilities of the rapidly changing frequency. Development of the modified methods is described in the thesis. A fast algorithm to solve Hankel system of equations arisen from CSM is developed based on Kumar's equation and detailed analysis is also presented in the thesis. Numerical simulations of sinusoids with varying frequencies and applications to track the formant frequencies of speech signals are treated.
Frequency tracking of nonstationary time series has been applied in various fields of signal processing; such as tracking doppler shift of radar signals, studying the ocean waves, studying the oscillation of the earth in Geophysics and tracking the formants in speech analysis. There are some literatures in the past which discuss about frequency tracking. Most of them were based upon the DFT and the AR method, [1-6]. This thesis will analyze in detail three types of frequency tracking methods which include AR method, DFT method and a method called composite sinusoidal model (CSM).

Methods of manipulating the nonstationary time series can be broadly classified into two categories. One is so called an adaptive processing. Another is called a block data processing. In the adaptive processing, the frequency characteristic of the given time series is estimated at each time when the new data comes. On the other hand, in block data processing for the given time series sampled data,

\[ \{x(n)\} = \{x(0), x(1), \ldots, x(t), x(t+1), \ldots\} \]

it will be divided into many small intervals (blocks) of data, \( \{x_t(n)\} \);

\[ \{x_t(n)\} = \{x(t+n)\} = \{x(t), x(t+1), \ldots, x(t+N-1)\} \]

where \( t \) denotes time index \( t=0, N, 2N, \ldots \) and \( N \)
is the length of a data block, thereafter the data in this small blocks will be used to estimate the frequency characteristic of the given time series data at that particular time $t$. The thesis will be mainly concerned with the block data processing, since the result of the block data processing is more accurate than those of adaptive processing as one can see the examples in the later chapter, and the computation time of block data processing is less than those of adaptive processing. The thesis will be developed according to the methods used to track the frequency as following:

Chapter 2, the discrete Fourier transform (DFT) is shown to be able to track the frequency of stationary or locally stationary time series signal while in the case of rapid changing frequency time series DFT fail to track. In order to track this rapid changing frequency again, the DFT is modified. It is found that the modified DFT is indeed the CZT with the appropriate parameters, which was first developed by Rabiner, et al in [7].

Chapter 3, the time series is modeled by an Autoregressive Model (AR), where the present value of the given time series is modeled as the linear combination of the previous values. AR model is also known as Linear Prediction[8]. The AR spectrum estimation was also proven to be equivalent to the maximum entropy spectrum estimation method[9].

Tracking the frequency by using AR model, one has to
estimate the AR coefficient, many literatures had been published to obtain the AR coefficient starting from well-known Levinson's algorithm[10], Levinson-Durbin[11], Burg[12], Marple[13], and Scott&Nikias[14]. All of the above literatures were done in the block data processing. Obtaining the AR coefficient by an adaptive processing can be found in [3], [6], [15-18]. Analysis in chapter 3 will show that, in order to be able to track the rapid changing frequency by AR model, the AR coefficient is modified.

Chapter 4, the composite sinusoidal model (CSM) which was first proposed by Sayama & Itakura in 1981[19] is used to model the given time series signal. The CSM will be described in detail. The modified CSM is proposed to model the linear rapidly changing frequency time series. The technique to solve Hankel type system of equations which is arisen from CSM by applying Kumar's Toeplitz inversion fast algorithm[20] is derived in detail.

The numerical simulation examples and the speech examples are shown in chapter 5 and 6 respectively. In addition, the detail derivation of some mathematic properties and Fortran programs are attached in appendices.
CHAPTER 2

FREQUENCY TRACKING BY DISCRETE FOURIER TRANSFORM (DFT)

If a set of sampled data \{x(n)\} is given, its DFT is defined as

\[
X(w) = \sum_{n=0}^{N-1} x(n) \exp(-jwn) \tag{2.1}
\]

The Fourier Transformation defined above will be used to analyze the following type of data.

2.1 Fixed frequency time series signal (Stationary)

Assuming the given stationary signal is expressed by a complex exponential

\[x_t(n) = \exp(j\omega_0(t+n)) \quad ; n=0,1,\ldots,N-1\]

where \(\omega_0\), \(t\) and \(N\) denote fixed angular frequency, time index and the length of sampled data used to estimate frequency at time \(t\). Taking Fourier transform, frequency spectrum at time \(t\) is obtained by

\[
X_t(w) = \sum_{n=0}^{N-1} \exp(j\omega_0(t+n)) \exp(-jwn) \\
= \exp(j\omega_0 t) \sum_{n=0}^{N-1} \exp(j(\omega_0 - w)n) \tag{2.2}
\]

It can be easily seen that the magnitude of DFT, \(|X_t(w)|\), is maximum at \(w=\omega_0\). In another words, if the magnitude of the DFT is plotted versus frequency axis the peak will locate at \(w=\omega_0\). So, for stationary sampled data, DFT can be used to estimate the frequency by estimating the peak of the spectrum.
The efficient algorithm to compute DFT, known as FFT, was first developed Cooley & Turkey [21]. In direct DFT, the number of complex multiplication and addition is $N$, but in the FFT the computation can be reduced to $N \log_2 N$, where $N$ is the size of FFT. In real time application, where the rapid time varying frequency is being tracked, the fast algorithm is really necessary.

2.2 Locally stationary time series signal (Quasi stationary)

In this case, the frequency is not fixed but changing very slowly, so it can be assumed to be fixed within the small interval of time. In another words, the frequency is assumed to be fixed within the small period of sampled data, $\{x_{t}(n)\}$, used to estimate the frequency at time $t$.

Assuming the complex exponential signal is used

$$x_{t}(n) = \exp(j\omega_{t}(t+n))$$

where angular frequency $\omega_{t}$ is assumed to be fixed within the length of $N$ sampled data.

Corresponding DFT of the signal is given by

$$X_{t}(w) = \sum_{n=o}^{N-1} \exp(j\omega_{t}(t+n)) \exp(-jwn)$$

$$= \exp(j\omega_{t} t) \sum_{n=o}^{N-1} \exp(j(\omega_{t}-w)n) \quad (2.3)$$

In a similar manner as described in the fixed frequency case, the magnitude of DFT, $|X_{t}(w)|$, will be maximum at $w=\omega_{t}$.
2.3 Rapid changing frequency time series signal (nonstationary)

The frequency is assumed to be linearly changing within the small interval of sampled data, \( \{ x_t(n) \} \), used to estimate the frequency at time \( t \). Complex exponential signal with linearly changing frequency is expressed as

\[
x_t(n) = \exp(j(w_t+n\Delta w)(t+n))
\]

where \( \Delta w \) denotes the amount change of frequency from sample to sample.

Taking DFT, yields

\[
X_t(w) = \sum_{n=0}^{N-1} \exp(j(w_t+n\Delta w)(t+n)) \cdot \exp(-jwn) = \exp(jw_t t) \sum_{n=0}^{N-1} \exp(j(w_t+t\Delta w+n\Delta w-w)n) \quad (2.4)
\]

From eq. (2.3), \( |X_t(w)| \) will be maximum approximately at

\[
w = w_t + t\Delta w
\]

where \( t >> N \) and \( \Delta w << 1 \). As one can see that the DFT fail to track the rapid time varying frequency. To overcome the difficulty, the following technique is introduced.

Modifying eq. (2.1) above as follows

\[
X(w) = \sum_{n=0}^{N-1} x(n) \cdot \exp(-j\theta n) \cdot \exp(-jwn) \quad (2.5)
\]

Eq. (2.5) was known as CZT with parameter \( A_0 = \tilde{A}_0 = Q \). Applying the modified DFT eq. (2.5) on the linearly changing frequency sampled data, yields

\[
X_t(w) = \sum_{n=0}^{N-1} x_t(n) \exp(-j\theta n) \exp(-jwn)
\]

\[
= \sum_{n=0}^{N-1} \exp(j(w_t+n\Delta w)(t+n)) \exp(-j\theta n) \exp(-jwn) = \exp(jw_t t) \sum_{n=0}^{N-1} \exp(j(w_t-w)n) \exp(j(t\Delta w+n\Delta w-w)n) \quad (2.6)
\]

From eq. (2.6), \( |X_t(w)| \) will be maximum approximately at \( w = w_t \), if \( \theta = t\Delta w \), where \( t >> N \). So, with appropriate parameters,
the CZT can be used to track the frequency of the rapid linearly changing frequency time series. In chapter 5, some numerical examples will be simulated to confirm the analysis described above.
3.1 AR modeling

In order to model the given time series sampled data by M order AR model, the linear combination of M previous values of the time series sampled data are used to predict the present value, which can be expressed as,

\[ \hat{x}(n) = - \sum_{i=1}^{M} a_i x(n-i) \]  \hspace{1cm} (3.1)

where \( \hat{x}(n) \) = estimated value of \( x(n) \)

\( a_i \) = AR coefficient.

Furthermore,

\[ x(n) - \hat{x}(n) = \sum_{i=0}^{M} a_i x(n-i) = e_n \]  \hspace{1cm} (3.2)

where \( a_0 = 1 \)

\( e_n \) = prediction error (for more specific sometimes called forward error, since eq. (3.1) forms the forward prediction).

By minimizing the square of prediction error, \( (e_n)^2 \), the optimum AR coefficient, \( \{a_i\}^t \), for the given period of sampled data \( \{x_t(n)\} \), will be obtained, where \( t \) denotes time index.

This Thesis will not mention about how to obtain the optimum AR coefficient, since numerous papers have been published as already mentioned in the introduction. The examples in chapter 5 and 6 will use some of those
techniques[12], [13], [15]. to obtain the optimum AR coefficients.

3.2 Fixed frequency or locally stationary(Quasi-stationary) time series signal

Assuming the given time series signal is a complex exponential,

\[ x_t(n) = x(t+n) = \exp(jw_t(t+n)) \quad (3.3) \]

where \( w_t \) denotes the locally stationary frequency which is constant within a small block of sampled data, ie.

\[ w_{t-N/2} = w_{t-N/2+1} = \ldots = w_t = \ldots = w_{t+N/2} \]

where \((N+1)\) is the size of the small block of sampled data used to estimate \( w_t \).

Substituting eq. (3.3) into eq. (3.1),

\[ \hat{x}(t) = - \sum_{n=1}^{M} a_n \exp(jw_t(t-n)) \quad (3.4) \]

Assuming the estimated \( \hat{x}(t) \) is exactly equal to \( x(t) \), ie. \( e_n \) is zero; therefore

\[ \hat{x}(t) = x(t) \]

From eq. (3.3) and eq. (3.4), the above equation becomes

\[ - \sum_{n=1}^{M} a_n \exp(jw_t(t-n)) = \exp(jw_t) \]

Finally;

\[ \sum_{n=1}^{M} a_n \exp(-jw_t n) = -1 \quad (3.5) \]

One can easily see that a solution of eq. (3.5) is

\[ a_n = -\frac{1}{M} \exp(jw_t n) \quad (3.6) \]

To test this solution, let us substitute eq. (3.6) back into eq. (3.5)

\[ \sum_{n=1}^{M} -\frac{1}{M} \exp(jw_t n) \exp(-jw_t n) = -1 \]
\[-M = -1\]

So, eq. (3.6) is assured to be a solution of eq. (3.5).

To obtain the frequency from the AR coefficient \(a_i^t\), let us take Z transform of eq. (3.2)

\[
Z \left[ \sum_{i=0}^{M} a_i x(n-i) \right] = Z \left[ e_n \right]; \quad a_0 = 1
\]

\[
X(Z) + a_1 Z^{-1} X(Z) + \ldots + a_M Z^{-M} X(Z) = Z \left[ e_n \right]
\]

\[
X(Z) = \frac{Z \left[ e_n \right]}{1 + \sum_{i=1}^{M} a_i Z^{-i}} \quad (3.7)
\]

Furthermore, substituting eq. (3.6) into eq. (3.7)

\[
X(Z) = \frac{Z \left[ e_n \right]}{1 - \sum_{i=1}^{M} \frac{1}{M} \exp(j\omega t) Z^{-i}} \quad (3.8)
\]

and evaluating the magnitude of eq. (3.8), \(|X(Z)|\), along unit circle on Z-plane, i.e., by letting \(Z = \exp(j\omega)\). One can see that \(|X(Z)|\) will be maximum at \(\omega = \omega_t\), or eq. (3.8) will have a pole at \(\omega = \omega_t\). So, by evaluating the location of the pole of eq. (3.7) for a set of the optimum AR coefficient \(a_i^t\), which obtained by minimizing the prediction error of the given sequence of time series sampled data, the frequency of this given time series signal at the specific time \(t\) can be estimated. Since AR model is concerning with estimating the pole location, some literatures called it as 'All Pole' model [8], [17].

By using eq. (3.7) to estimate the frequency of the given time series signal, one problem is arisen when \(Z \left[ e_n \right]\)
equal to zero; in order to avoid this zero dividing by zero indeterminancy, eq. (3.7) is modified to

$$Q(w) = \left| X(Z) \right| = \frac{1}{1 + \sum_{i=1}^{M} a_i Z^{-i}}$$

(3.9)

where $Z = \exp(jw)$

The frequency estimator in the form of eq. (3.9) was first used by Griffith [3]. It also has another advantage as it is not affected by amplitude scale changes in the given time series sampled data.

3.3 Rapid changing frequency time series signal (non-stationary)

Assuming the frequency is linearly changing within the given block of time series sampled data; eq. (3.3) becomes

$$x(t+n) = \exp(j(w_t+n\Delta w)(t+n))$$

(3.10)

where $\Delta w$ is the amount of frequency change from sample to sample.

Substituting eq. (3.10) into eq. (3.1)

$$x(t) = \sum_{n=1}^{M} a_n \exp(-j(w_t-n\Delta w)(t-n))$$

(3.11)

Assuming the prediction error at time $t$, $e_n$, is equal to zero, i.e., $\hat{x}(t) = x(t)$; substituting eq. (3.10) and eq. (3.11) into this equation, yields

$$- \sum_{n=1}^{M} a_n \exp(j(w_t-n\Delta w)(t-n)) = \exp(jw_t t)$$

$$\sum_{n=1}^{M} a_n \exp(j(w_t+t\Delta w-n\Delta w)n) = -1$$

(3.12)

It can be shown that a solution of eq. (3.12) is

$$a_n = -\frac{1}{M} \exp(j(w_t+t\Delta w-n\Delta w)n)$$

(3.13a)

and

$$a_n \neq -\frac{1}{M} \exp(j(w_t+t\Delta w)n)$$

(3.13b)
where \( t \gg M \), and \( w \ll 1 \)

To test the existence of eq. (3.13), let us substitute eq. (3.13) into eq. (3.12)

\[
\sum_{n=1}^{M} \frac{-1}{M} \exp(j(w_t + t\Delta w - n\Delta w)n) \exp(-j(w_t + t\Delta w - w)n) = \frac{M}{M} = -1
\]

So, it is assured that eq. (3.13) is a solution of eq. (3.12). In order to estimate the frequency of the given rapid changing time series, let us substitute eq. (3.13) into the frequency estimator, \( Q(w) \), eq. (3.9)

\[
Q(w) = \frac{1}{1 - \frac{1}{M} \sum_{i=1}^{M} \exp(j(w_t + t\Delta w - i\Delta w)n) \exp(-j(w_t + t\Delta w - w)n)}
\]

Eq. (3.14) will have a pole at

\[
w = w_t + t\Delta w
\]

where \( t \gg M \), and \( w \ll 1 \)

One has already seen that by using the frequency estimator in the form of eq. (3.9) to estimate the frequency of the given rapid changing time series sampled data at time \( t \), the frequency is shifted by \( t\Delta w \). The numerical simulation examples to confirm the analysis above will be seen in chapter 5. In order to be able to correctly estimate the frequency of the given rapid changing time series signal the frequency estimator \( Q(w) \) in the form of eq. (3.9) should be modified as follows

\[
Q'(w) = \frac{1}{1 + \sum_{i=1}^{M} q_i \exp(-j(t\Delta w - i\Delta w)n) Z^{-i}}
\]

where \( Q'(w) \) notation was used to distinguish from unmodified frequency estimator \( Q(w) \).
Furthermore, if $t \gg M$, and $w << 1$, eq. (3.15) can be approximated as

$$Q'(w) = \frac{1}{1 + \sum_{l=1}^{M} a_i \exp(-j tw) Z^{-l}}$$

(3.16)

Comparing the modified frequency estimator $Q'(w)$ eq. (3.16) with those unmodified frequency estimator $Q(w)$ eq. (3.9), one can see that the modified frequency estimator $Q'(w)$ can be derived from the unmodified frequency estimator $Q(w)$ by modifying the AR coefficient $\{a_i\}^t$ as follows,

$$a'_i = a_i \exp(-j tw)$$

(3.17)

where ' notation used to indicate the modified coefficient.

Therefore eq. (3.16) can be rewritten as

$$Q'(w) = \frac{1}{1 + \sum_{l=1}^{M} d_i Z^{-l}}$$

(3.18)

3.4 Another approaching of AR model

As mentioned in the previous section, in order to obtain the optimum AR coefficient $\{a_i\}^t$, the square of prediction error $(e_n)^2$ have to be minimized. Let us define the residual error (average square error) of the given time series sampled date within the small interval used to obtain the optimum AR coefficient $\{a_i\}^t$ at time $t$ by

$$E_{M,N} = E[ (e_n)^2 ] = \frac{1}{N - M + 1} \sum_{n=M}^{N} e_n e_n^*$$

(3.19)

where $N =$ size of the small interval of data
M = order of AR model  
* = complex conjugate  
E[.] = expectation operator.

Substituting eq. (3.2) and its complex conjugate in eq. (3.19)

\[
E_{M,N} = \frac{1}{(N - M + 1)} \sum_{i=0}^{M} \sum_{j=0}^{M} a_i a_j^* x(n-i) x(n-j)
\]

\[
= \frac{1}{(N - M + 1)} \sum_{i=0}^{M} \sum_{j=0}^{M} a_i a_j^* r(m) \quad (3.20)
\]

where \( r(m) \triangleq \frac{1}{(N - M + 1)} \sum_{n=0}^{M-N} x(n)x(n+m) \)

\[
= \text{Autocorrelation function} \quad (3.21)
\]

Minimizing \( E_{M,N} \) by taking derivative of \( E_{M,N} \) with respect to \( a_j^* \) and equating them to zero; yields

\[
\begin{bmatrix}
  r(0) & r(-1) & r(-2) & \ldots & r(-N+1) \\
  r(1) & r(0) & r(-1) & \ldots & r(-N+2) \\
  \vdots & \vdots & \vdots & \ddots & \vdots \\
  \vdots & \vdots & \vdots & \ddots & \vdots \\
  r(N-1)r(N-2) & \ldots & r(0) 
\end{bmatrix}
\begin{bmatrix}
  a_1 \\
  a_2 \\
  \vdots \\
  \vdots \\
  a_N 
\end{bmatrix}
= 
\begin{bmatrix}
  r(1) \\
  r(2) \\
  \vdots \\
  \vdots \\
  r(N) 
\end{bmatrix}
\]

\[
R_x a = -r \quad (3.22)
\]
Eq. (3.22) forms the Toeplitz system of equations.

The same as in previous section, the analysis will be performed on both types of time series signals (stationary and non-stationary).

3.4.1 Fixed frequency time series signal (stationary)

By using eq. (3.21), the Autocorrelation function of time series signal expressed in the form of eq. (3.3) is

\[ r(m) = \frac{1}{(N - M + 1)} \sum_{n=0}^{N-M} \exp(-jw_n(t+m)) \exp(jw_n(t+n+m)) \]

\[ = \frac{1}{(N - M + 1)} \sum_{n=0}^{N-M} \exp(jw_n m) \]

\[ = \frac{1}{(N - M + 1)} \exp(jw_n m) \sum_{n=0}^{N-M} 1 \]

\[ r(m) = \exp(jw_n m) \]  (3.23)

Substituting eq. (3.23) into eq. (3.22), yields

\[
\begin{bmatrix}
\exp(0) & \exp(-jw_1) & \exp(-j2w_1) & \ldots & \exp(-j(N-1)w_1) \\
\exp(jw_1) & \exp(0) & \exp(-jw_1) & \ldots & \exp(-j(N-2)w_1) \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
\exp(jw_{N-1}) & \exp(0) & \ldots & \exp(-j(N-1)w_1) \\
\end{bmatrix}
\begin{bmatrix}
a_1 \\
a_2 \\
\vdots \\
a_{N-1} \\
\end{bmatrix}
= \begin{bmatrix}
\exp(jw_1) \\
\exp(j2w_1) \\
\vdots \\
\exp(j(N-1)w_1) \\
\end{bmatrix}
\]  (3.24)
Solving the above equation (3.24) by using
undertermined coefficient technique;

Let \( a_i = A_i \exp(j \omega t) i \) and substitutes back into
eq.(3.24), after some algebraic manipulation, it can be
seen that \( a_i = -\frac{1}{M} \exp(j \omega t i) \) is a solution of eq.(3.24).
It can be seen that the optimum AR coefficient \( \{a_i\} \)
just obtained above, has the same form as those AR
coefficient eq.(3.6) derived in previous section. As
already shown in previous section, the estimated frequency
will be equal to \( \omega t \).

3.4.2 Rapid changing frequency time series signal

Using the same expression for the rapid linear
changing frequency time series signal as in the previous
section eq.(3.10), by definition of the Autocorrelation
function in the form of eq.(3.21), the Autocorrelation
function of eq.(3.10) is

\[
\begin{align*}
R(m) &= \frac{1}{(N-M+1)} \exp(-j(w_t+n\Delta w)(t+n)) \exp(j(w_t+(n+m)\Delta w)(t+n+m)) \\
R(m) &= \frac{1}{(N-M+1)} \exp(j(w_t+t\Delta w+n\Delta w)m) \\
R(m) &= \frac{1}{(N-M+1)} \exp(j(w_t+t\Delta w)m) \quad (3.25)
\end{align*}
\]

where \( M = \) order of AR model
\( N = \) size of the sampled data interval used to
estimate the frequency
\( m = \) Autocorrelation lag and \( m \leq M \ll N \)

Substituting eq.(3.25) into eq.(3.22), yields
\[
\begin{bmatrix}
\exp(0) & \exp(-j(w_t + t\Delta w)) & \cdots & \exp(-j(M-1)(w_t + t\Delta w)) \\
\exp(j(w_t + t\Delta w)) & \exp(0) & \cdots & \exp(-j(M-2)(w_t + t\Delta w)) \\
\vdots & \vdots & \ddots & \vdots \\
\exp(j(M-1)(w_t + t\Delta w)) & \cdots & \exp(0)
\end{bmatrix}
\begin{bmatrix}
1 \\
a2 \\
\vdots \\
am_M
\end{bmatrix}
= -
\begin{bmatrix}
\exp(j(w_t + t\Delta w)) \\
\exp(j2(w_t + t\Delta w)) \\
\vdots \\
\exp(jM(w_t + t\Delta w))
\end{bmatrix}
\]
Solving eq. (3.26) by using undetermined coefficient method,

Let \(a_i = A_i \exp(j(w_t + t\Delta w)i)\), and substitutes it into eq. (3.26), yields

\[
MA \exp(j(w_t + t\Delta w)) = -\exp(j(w_t + t\Delta w)) \\
MA \exp(j2(w_t + t\Delta w)) = -\exp(j2(w_t + t\Delta w)) \\
eq \ldots \ldots \\
MA \exp(jM(w_t + t\Delta w)) = -\exp(jM(w_t + t\Delta w)) \quad (3.27)
\]

From eq. (3.27), it can be seen that

\[A_1 = A_2 = \ldots = A_M = \frac{-1}{M}\]

So, a solution of eq. (3.26) is

\[a_i = -\frac{1}{M} \exp(j(w_t + t\Delta w)i) \quad (3.28)\]

which has the same form as eq. (3.13b). As described in the previous section, to be able to accurately estimate the frequency of the given time varying frequency signal, the AR coefficients should be modified to be in the form of eq. (3.17); as the result, the modified frequency estimator \(Q'(w)\) eq. (3.18) is obtained. This is another approach to show that in the case of rapidly changing frequency the AR coefficients should be modified.
CHAPTER 4

COMPOSITE SINUSOIDAL MODEL

A procedure of tracking frequencies by composite sinusoidal MODEL (CSM) [19] is to assume that the time series is composed of the sum of multiple sinusoidal signals. And next, the modeled time series is used to derive the model Autocorrelation function; and by equating the model Autocorrelation function with the Autocorrelation function from the sampled data, it can be seen that the Hankel type and Van der monde type system of equation need to be solved [19]. Finally, by solving Hankel and Van der monde system of equations, we will obtain the frequencies and the amplitude of each frequencies respectively. This composite sinusoidal modeling (CSM) was first used by Sagayama & Itakura to analyze the speech signals [19], in their paper, they solved Hankel type system of equations by using the recursive algorithm similar to Levinson's algorithm [10]. In this thesis, Kumar's Toeplitz inversion fast algorithm will be applied to solve this Hankel type system of equations [20].

4.1 Composite Sinusoidal Modeling for fixed frequency time series signal

4.1.1 CS modeling

The given time series signal y(t) can be modeled by
the sum of multiple sinusoids

\[ y(t) = \sum_{i=1}^{n} s_i(t) + n_t \]  \hspace{1cm} (4.1)

where the multiple sinusoids is expressed by

\[ s_i(t) = \sqrt{2m_i} \sin(w_it + \phi_i) \]

and \( w_i, \phi_i, \sqrt{2m_i} \) and \( n_t \) are described as

- \( w_i \) = angular frequency
- \( \phi_i \) = phase
- \( \sqrt{2m_i} \) = amplitude
- \( n_t \) = white noise

The Autocorrelation function of eq. (4.1) can be shown by

\[ R_1 = \sum_{i=1}^{n} m_i \cos(w_i) \]  \hspace{1cm} (4.2)

where we assumed that \( \phi_i \) is a white noise with (uniform probability density between \(-T\) and \(T\)).

4.1.2 Autocorrelation function of sampled data and autocorrelation function of composite sinusoidal model

Autocorrelation function \( R_1 \) of sampled data can be defined by

\[ R_1 = \frac{1}{N} \sum_{i=1}^{N} Y_i Y_{i-1} \]  \hspace{1cm} (4.3)

where \( N \) denotes the length of sampled data.

The Autocorrelation function from sampled data should be equal to the CSM Autocorrelation \( r_1 \). By assuming we have \( 2n \) unknowns; \( (m_1, m_2, \ldots, m_n, w_1, w_2, \ldots, w_n) \), we are required to solve the following \( 2n \) simultaneous
equations.
\[ \sum_{i=1}^{n} m_i \cos(w_i l) = R_l \]  
(4.4)

where \( l = 0, 1, \ldots, 2n-1 \)

If we let \( x_i = \cos(w_i) \), eq. (4.4) will become
\[ \sum_{i=1}^{n} m_i T_l(x_i) = R_l \]  
(4.5)

where \( l = 0, 1, \ldots, 2n-1 \) and \( T_l(x) \) is called chebyshev polynomial[23]. By making use of the chebyshev polynomial properties, the following relation can be obtained
\[ x^l = \frac{1}{2^l} \sum_{k=0}^{2l} \binom{2l}{k} x^k \]  
(4.6)

where \( \binom{2l}{k} \) denotes the combination operator. The derivation of eq. (4.6) can be found in[22-23]. Now, if the sequence \( U_l \) is defined such that
\[ U_l = \frac{1}{2^l} \sum_{k=0}^{2l} \binom{2l}{k} R_{2k-l} \]  
(4.7)

Eq. (4.5) can be written as
\[ \sum_{i=1}^{n} m_i x_i^l = U_l \]  
(4.8)

where \( l = 0, 1, \ldots, 2n-1 \)

To solve eq. (4.8), let us define the following polynomial
\[ P_n(x) = x^n + p_1^{(n)} x^{n-1} + p_2^{(n)} x^{n-2} + \ldots + p_n^{(n)} \]  
(4.9)

where supper-script \( n \) in parenthesis indicate the order of polynomial.

Now, left hand side of eq. (4.8) is multiplied by the eq. (4.9), yields
\[ \sum_{j=1}^{n} m_j P_n(x) x_i^l = \sum_{i=1}^{n} m_i \sum_{k=0}^{n} p_{n-k}^{(n)} x_i^k x_i^l \]
\[ = \sum_{i=1}^{n} m_i \sum_{k=0}^{n} p_{n-k}^{(n)} x_i^k U_{k+1} = \]  
(4.10)
Eq. (4.10) can be written in the matrix form

\[
\begin{bmatrix}
U_0 & U_1 & \cdots & U_{n-1} \\
U_1 & U_2 & \cdots & U_n \\
\vdots & \vdots & \ddots & \vdots \\
U_{n-1} & U_n & \cdots & U_{2n-2}
\end{bmatrix}
\begin{bmatrix}
P_n \\
P_{n-1} \\
\vdots \\
P_1
\end{bmatrix}
= 
\begin{bmatrix}
U_n \\
U_{n+1} \\
\vdots \\
U_{2n-1}
\end{bmatrix}
\] (4.11 a)

the above expression is further converted to the following matrix equation.

\[
\begin{bmatrix}
U_0 & U_1 & \cdots & U_n \\
U_1 & U_2 & \cdots & U_{n+1} \\
\vdots & \vdots & \ddots & \vdots \\
U_n & U_{n+1} & \cdots & U_{2n}
\end{bmatrix}
\begin{bmatrix}
P_n \\
P_{n-1} \\
\vdots \\
P_1
\end{bmatrix}
= 
\begin{bmatrix}
0 \\
0 \\
\vdots \\
1
\end{bmatrix}
\] (4.11 b)

which can be briefly expressed by

\[
H \overrightarrow{P} = \overrightarrow{\sigma}
\] (4.11c)

where \( H \) = Hankel matrix

\[
\overrightarrow{P} = (P_n, P_{n-1}, \ldots, P_1, 1)^t
\]

\[
\overrightarrow{\sigma} = (0, 0, \ldots, \sigma)^t
\]

\( t \) = transpose operator.

Eq. (4.11a-c) form the Hankel type system of equations. By solving eq. (4.11b) the coefficient vector \( \overrightarrow{P} \) will be obtained. Substituting the elements of the coefficient vector \( \overrightarrow{P} \), one can obtain the frequencies \( \omega_i \) such that
\[ w_i = \cos^{-1}(x_i) \quad ; \quad i=1,2,\ldots,n \]

where \( x \) are the roots of the polynomial \( P_n(x) \). Amplitude of each frequencies, \( (m_1, m_2, \ldots, m_n) \), can be obtained by solving eq. (4.8), after substituting back the values of \( (x_1, x_2, \ldots, x_n) \) into eq. (4.8), which will form the Van der monde type system of equations as follows:

\[
\begin{bmatrix}
1 & 1 & \ldots & 1 \\
x_1 & x_2 & \ldots & x_n \\
x_1^2 & x_2^2 & \ldots & x_n^2 \\
\vdots & \vdots & \ddots & \vdots \\
x_1^{n-1} & x_2^{n-1} & \ldots & x_n^{n-1}
\end{bmatrix}
\begin{bmatrix}
m_1 \\
m_2 \\
m_3 \\
\vdots \\
m_n
\end{bmatrix}
= 
\begin{bmatrix}
U_0 \\
U_1 \\
U_2 \\
\vdots \\
U_{n-1}
\end{bmatrix}
\quad (4.12)
\]

4.2 CSM for rapid time varying frequency signal

In case of rapid time varying frequency, assuming linearly changing eq. (4.1) above should be changed to be in the following form

\[ y(t) = \sum_{i=1}^{n} s_i(t) + n(t) \]

where \( s_i(t) = \sqrt{2m_i} \sin(w_i(t) t + \phi_i) \)

\[ w_i(t) = w_i + t\Delta w_i \]

\( \Delta w = \) the amount of frequency changes from sample to sample.

The Autocorrelation function of the above equation can be shown to be

\[ r_1 = \sum_{i=1}^{n} m_i \cos((w_i + t\Delta w)1) \]
In similar manner as the fixed frequency case in previous section, in order to obtain 2n unknowns \((m_1, m_2, \ldots, m_n, w_1, w_2, \ldots, w_n)\) which now are time varying; the system of 2n simultaneous equations have to be solved. The same procedure as in the case of fixed frequency can be used, the only change is that in time varying frequency case \(x_i\) is defined as \(\cos(w_i + t\Delta w_i)\) instead of \(\cos(w_i)\). Therefore, after solving Hankel system of equations and finding the roots of polynomial eq. (4.9) the estimated frequencies can be obtained from

\[
\begin{align*}
  w_i + t\Delta w &= \cos^{-1}(x_i) \\
  w_i &= \cos^{-1}(x_i) - t\Delta w
\end{align*}
\]

4.3 Solution of Hankel Matrix System

The following steps are applied to solve the Hankel system of equations expressed in the eq. (4.11b).

4.3.1 Changing Hankel System to Toeplitz System

We can change Hankel system to Toeplitz system by premultiplying Hankel matrix by Exchange matrix \((J)\), which has 1's along its cross diagonal and 0's elsewhere, and has the same size as the Hankel matrix. This property was proven in appendix (1). Multiplying the Hankel system eq. (4.11c) by exchange matrix \((J)\), yields

\[
\begin{align*}
  JHP &= J\hat{\sigma}' \\
  TP &= \hat{\sigma}'
\end{align*}
\]

where \(T = a Toeplitz matrix\)

\[
\hat{\sigma}' = (\sigma', 0, 0, \ldots, 0)^t
\]
t = transpose operation.

4.3.2 Computing the first column of the Toeplitz matrix inverse

It is shown in appendix(2) that the solution of the Toeplitz system in the form of eq.(4.11b) is the first column of the inverse Toeplitz matrix dividing by its first element in the column. To obtain the first column of the inverse Toeplitz matrix, we will use Kumar's fast algorithm[20] as following;

4.3.2.1 Changing Toeplitz matrix to Cyclic Toeplitz matrix

By changing Toeplitz matrix to Cyclic Toeplitz matrix, we can take advantage of the fact that the inverse of Cyclic Toeplitz matrix still be in the form of Cyclic Toeplitz matrix, so the inverse of Cyclic Toeplitz can be obtained easily and efficiently by using FFT[20].

Let T is (n×n) Toeplitz matrix and P is (m+1×m+1) Cyclic Toeplitz matrix;

\[
\begin{bmatrix}
  t_0 & t_1 & \cdots & t_{n-1} \\
  t_n & t_0 & \cdots & t_{n-2} \\
  \vdots & \vdots & \ddots & \vdots \\
  t_0 & \cdots & \cdots & t_0 \\
\end{bmatrix}
\]
where \(c_1, c_2, \ldots, c_k\) are arbitrary constant to make the size \((m+1)\) of \(P\) to be the power of 2, and superscript in parenthesis indicate size of the matrix. Denoting the first row of the matrix \(P\) by the vector \(P\)

\[
P = (t_0, t_1, \ldots, t_{n-1}, c_1, c_2, \ldots, c_k, t_{2n-1}, t_{2n-2}, \ldots, t_n)
\]

Since \(P\) is a Cyclic Toeplitz matrix (cyclic shift right matrix) by knowing just its first row or first column, one can write the whole matrix \(P\). Next, let us denote the inverse of \(P\) by \(Q\), and the first row of \(Q\) by \(Q\). To obtain \(Q\), it is enough to find just the first row or first column, since \(Q\) still be in the form of the cyclic shift right matrix. In appendix (3), it is shown that the first row of the product of \(PQ\) is equal to circular convolution of \(P\) with \(Q\), and by definition of inverse matrix, yields

\[
P \ast Q = E
\]

where \(\ast = \text{circular convolution}\)

\[
E = (1 \ 0 \ 0 \ \ldots \ \ldots \ldots \ldots)
\]

Next, let us denote the FFT of \(P\) and \(Q\) by \(U\) and \(V\) respectively. From the above equation, one can see that

\[
V(k) = 1/U(k), \quad k=0,1,\ldots,n-1 \quad \text{where} \ n \ \text{is the size of the FFT.}
\]

Since the existency of \(Q\) is assumed, so all values of \(V(k)\) are finite. It is clear that \(Q\) can be obtained by taking the inverse FFT of \(V(k)\). The whole process of obtaining \(Q\) can be drawn on the figure (4.1).
\[ P \xrightarrow{\text{FFT}} U(k) \xrightarrow{\frac{1}{U(k)}} V(k) = \xrightarrow{\text{IFFT}} Q \]

Figure 4.1. Obtaining the 1st row of the inverse of a cyclic shift right matrix

(non-Toeplitz)

(toeplitz)

Figure 4.2. Updating area (shaded) of eq. (4.15)
4.3.2.2 Obtaining the Inverse Toeplitz matrix from the Inverse Cyclic Toeplitz

Matrices $P$ and $Q$ are partitioned as

$$P^{(m+1)} = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix}, \quad Q^{(m+1)} = \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix}$$

where $P_{11}, Q_{11}$ are $(1 \times 1)$ matrices. From matrix theory\[^{[37]}\]

$$P_{22}^{-1} = Q_{22} - Q_{21} Q_{11}^{-1} Q_{12}$$  \hspace{1cm} (4.14)

By using eq. (4.14) above, we can write the recursive algorithm to obtain the inverse matrix of $T$ as follows;

$$Q^{(m)} = Q_{22}^{(m)} - Q_{21}^{(m)} Q_{11}^{-1}^{(m)} Q_{12}^{(m)}$$

in scalar form,

$$Q_{i,j}^{(m)} = Q_{i+1,j+1}^{(m+1)} - Q_{i+1,1}^{(m+1)} Q_{1,j+1}^{(m+1)} / Q_{1,1}^{(m+1)}$$  \hspace{1cm} (4.15)

(Keep updating until $Q^{(m)}$ has the same size as $T$, ie. $T = Q^{(n)}$)

By using eq. (4.15), it still takes a lot of computation time, since we have to update almost the whole matrix at each recursion to get just the first row and first column of $Q$. In figure (4.2), the updating area at each stage of recursion are pictorially described.
It is seen in figure (4.2) that although the persymmetry property of inverse of Toeplitz matrix is used, by using eq. (4.15), one still has to update half of the matrix at each stage of recursion. In order to save computation time, Kumar applied the persymmetry of the inverse Toeplitz matrix to get the following updating equation [20];

\[
Q_{i,j}^{(m)} = Q_{i,j}^{(m+1)} - Q_{m+2-j,1}^{(m+1)} Q_{1,m+2-j,1}^{(m+1)} / Q_{1,1}^{(m+1)}
\] (4.16)

By using Kumar's equation [20], one can only update just the first row and first column of \( Q \) at each stage of recursion to get the first row of the inverse matrix of \( T \), so the computation time is much less than before.

### 4.4 Solution of the Van der Monde matrix system

Gaussian elimination [36] is the most efficient method available to solve the Van der Monde matrix system eq. (4.12).
In this chapter, test data are generated and used to demonstrate the behaviour of the frequency estimation technique described in the previous chapters. The data used in the simulations will be classified into 3 cases, which are fixed frequency data, locally stationary (quasi-stationary) and rapidly changing frequency.

5.1 Data types

5.1.1 Fixed frequency data

The Kay & Marple data [24] is used to test the algorithms. The real spectrum of Kay & Marple data is shown in Figure(5.0), and the data list is shown in appendix(4).

5.1.2 Locally stationary (quasi-stationary) data

The locally stationary data is the data sequence whose frequencies are fixed within the small intervals of data. The locally stationary is expressed as follows

\[ x_t(n) = \sum_{i=1}^{k} s_i(n) + \text{white noise} \quad (5.1) \]

where

\[ s_i(n) = A_i \sin(2\pi f_i (t+n)) \]

\[ f_i = f_c + f_d \cos(2\pi f_m t) \]

\[ n = 0, 1, 2, \ldots, N-1 \]

\[ t = 0, N, 2N, 3N, \ldots \]

and \( N \) is a size of the interval of data in which the
frequencies $f_m$ are assumed to be fixed. In the simulation, the following parameter values are used, $k=2$, $f_{m1}=1.0/1984.0$, $f_{m2}=2.0/1984.0$, $f_{cl}=0.35$, $f_{c2}=0.2$, $f_{dl}=f_{d2}=0.05$, $A_1=A_2=1.0$, $N=32$. Signal to noise ratio (SNR) is defined as

$$SNR = 20 \log_{10} \frac{A}{\sqrt{\sigma^2}}$$

where $A$ is a maximum amplitude of the signal and $\sigma$ is a variance of white noise.

5.1.3 Rapid changing frequency data

In this case the signal is expressed as

$$x_t(n) = A \sin(2\pi f_t(n) (t+n)) \quad (5.2)$$

where

$$f_t(n) = f_c + f_d \cos(2\pi f_m (t+n)) \quad (5.2a)$$

$$n = 0,1,2,\ldots,N-1$$

$$t = 0,N,2N,\ldots$$

In the simulation, the following parameter values will be used, $f_c=0.2$, $f_d=0.045$, $f_m=0.65/1984.0$, $N=32$.

5.2 FFT method

5.2.1 Fixed frequency case:

Kay & Marple data [24] is used in the simulation. Since the FFT is a block data processing, a set of sampled data whose size has to be the same as the size of the FFT, which is required as the input of the FFT process. But Kay & Marple data is a set of 64 numbers, so in order to be able to obtained the FFT spectrum from the FFT whose size is larger than 64, zeros have to be padded after the sequence of Kay & Marple data to make its size to be equal to the
size of the FFT being used. The FFT spectrum from various size of FFT are shown in Figures(5.1a-c). The comparison between the FFT spectrum with the real spectrum of Kay & Marple data Figure(5.0) shows that the FFT resolution is not worse, since the FFT seem to be able to separate the two close frequencies(0.2,0.21). The noisy characteristic of the FFT spectrum can be seen in Figures(5.1a-c). This FFT noisy characteristic can be reduced by applying the non-rectangular window on the time series data sequence. Figure(5.1c-e) show the FFT spectrum with rectangular window (no window), Hanning window and Hamming window respectively. One can see that, the FFT spectrums with non-rectangular window have less noisy characteristic, but the resolution is also reduced. The effects of non-rectangular window in spectrum estimation by block data processing will be discussed in some detail in chapter 6.

5.2.2 Locally stationary frequency case

The data expressed in eq.(5.1) with various SNR have been used in the simulation. Since the FFT spectrum has noisy characteristic, in order to plot the estimated frequencies versus time axis, the frequencies axis of the FFT spectrum will be divided into two intervals, thereafter the maximum peak in each interval is picked up to be the estimated frequency. The result are shown in Figures(5.2a-c).
5.2.3 Rapid changing frequency case

The data described in section 5.1.3 is used with various levels of SNR. The CZT with $A_0=0$, $W_0=0$, and $\theta=t(2\pi \Delta f)$ is used to track this rapid changing frequency time series signal, where $\Delta f$ denotes the amount change of frequency from sample to sample, which is the derivative of eq.(5.2a) at time $(t+N/2)$, where $t=0,N,2N,\ldots$; ie.

$$\Delta f = 2\pi fd. fm. \sin(2\pi fm. (t+N/2)) \quad (5.3)$$

Figure(5.3a-c) show the frequency tracking capability of the CZT of the data described in section 5.1.3 with various noise levels. One can see in Figure(5.3c) that although the noise is high (SNR=-3dB) the CZT still be able to track this rapid time varying frequency quite well.
Figure 5.0. Real spectrum of Kay&Marple data
Figure 5.1a. 128-point FFT spectrum of Kay&Marple data
Figure 5.1b. 256-point FFT spectrum of RayMarple data
Figure 5.1c. 1024-point-FFT spectrum of Kay&Marple data
Figure 5.1d. 1024-point-FFT with Hanning window spectrum of Kay & Marple data
Figure 5.1e. 1024-point-FFT with Hamming window spectrum of Kay & Marple data
FFT & HAMMING WINDOW

Figure 5.2a. quasi-stationary signal, SNR=17 dB.
+ estimated frequency, - real frequency
Figure 5.2c. quasi-stationary signal, SNR = -3dB
+ estimated frequency, - real frequency
Figure 5.3a. Rapid time varying frequency, SNR=17 dB
- real frequency, + FFT result, . CZT result
Figure 5.3b. rapid time varying frequency, SNR= 3 dB
- real frequency, + FFT result, . CZT result
Figure 5.3c. rapid time varying frequency, SNR = -3 dB
- real frequency, + FFT result, . CZT result
5.3 AR method

5.3.1 Fixed frequency case

The AR spectrum of Kay & Marple data with various AR order by various methods of obtaining the AR coefficients are shown in Figures(5.4a-e). Figure(5.4a-b) show the similarity of the AR spectrum when the AR coefficients have been obtaining by Burg's Algorithm and Marple's Algorithm. Figure(5.4c) shows the order which the AR method can be able to separate two close frequencies(0.2,0.21). Figure(5.4d) shows the AR spectrum obtaining by applying Hamming window and by applying no window (rectangular window). One can see that the AR spectrum has the smooth characteristic.

5.3.2 Locally stationary frequency case

The AR spectrograms (frequencies versus time axis) of the data, described in section 5.1.2 with various SNR are shown in Figures(5.5a-d). Since the AR spectrum has the smooth characteristic as one has already seen in section 5.3.1 above, therefore the simple peak picking appendix(10) can be used to obtain the estimated frequencies from the AR spectrum. A flow chart of AR process is shown in appendix(5).

5.3.3 Rapid changing frequency case

The modified frequency estimator eq.(3.16) with $\Delta f$ expressed in eq.(5.3) is used to track the rapid time varying frequencies of data described in section (5.1.3). The simulation results shown in Figures(5.6a-c) are agree with the analysis in section(3.3) and (3.4).
Figure 5.4a. AR spectrum of Kay&Harple data
AR coefficients by Burg, order= 15, no window
Figure 5.4b. AR spectrum of Kay\&Marple data
AR coefficients by Marple, order= 15, no window
Figure 5.4c. AR spectrum of Kay&Marple data
AR coefficients by Burg, no window
Figure 5.4d. AR spectrum of Kay & Marple data
AR coefficients by Burg, no window and Hamming window
order = 14
Figure 5.4e. AR spectrum of Kay & Marple data
AR coefficient by Marple, order = 20, no window
Figure 5.5a. quasi-stationary signal, SNR = 17 dB
order = 4, Hamming window, - real frequency,
. estimated frequency
AR COEF. BY BURG

Figure 5.5b. quasi-stationary signal, SNR=-0.5dB
order= 6, Hamming window, - real frequency,
. estimated frequency
Figure 5.5c. quasi-stationary signal, SNR = -3 dB
order = 4, Hamming window, - real frequency,
. estimated frequency
AR COEF. BY BURG

Figure 5.5d. quasi-stationary signal, SNR = -3 dB
order = 6, Hamming window, - real frequency,
. estimated frequency
Figure 5.6a. rapid time varying frequency, SNR = 9 dB
- real frequency, + unmodified AR, . modified AR
order = 3.
Figure 5.6b. rapid time varying frequency, SNR = 3 dB
- real frequency, + unmodified AR, . modified AR
order = 3.
Figure 5.6c. rapid time varying frequency, SNR=-0.5dB
- real frequency, + unmodified AR, . modified AR
order = 3.
5.4 CSM method

5.4.1 Fixed frequency case

The CSM method described in section (4.1) & (4.3) are used to estimate the frequencies of the Kay & Marple data and the amplitude of these frequencies will be obtained by solving the Van der monde type system of equations eq. (4.12), by using Gaussian elimination technique[36]. The CSM spectrum with various CSM order are shown in Figures (5.7a-f). One can see from the CSM spectrum that the CSM method seem to be able to detect high frequency better than low frequency.

5.4.2 Locally stationary case

Figure (5.8a-c) show the frequency tracking capability of the CSM method, where the data described in section (5.1.2) with various values of SNR are used in the simulation. It can be seen that when the SNR is low, the CSM method seem to be less sensitive to the frequency change.

5.4.3 Rapid changing frequency case

The modified CSM proposed in section (4.2) will be used to model the rapid changing frequency time series described in section (5.1.3), thereafter the estimated frequencies will be obtained by using eq. (4.13) with \( \Delta f \) expressed by eq. (5.3) above. Figure (5.9a-c) show the capability of the modified CSM to track the rapid time varying frequency.
5.4.4 How to select the order of the CSM

In order to match the CSM with the given sampled data, the appropriate CSM order is selected. Unfortunately, the theory about how to select the appropriate CSM order is not available yet. Therefore, trial and error technique is used in the thesis. In fixed frequency case (Kay & Marple data), the expected CSM spectrum is supposed to be close to the real Kay & Marple data where the order is higher and higher but the experimental results show that when the order is above the optimum order (order=10 which the CSM spectrum is closest to the Kay & Marple data), the CSM spectrums show just high frequency peaks (noise) and lost the expected low frequency peaks (0.1, 0.2 and 0.21). One reason which tries to explain this experimental result is that the length of Kay & Marple data is too short just 64 points.
Figure 5.7a. CSM spectrum of Kay&Marple data order 5, no window
Figure 5.7b. CSM spectrum of Kay&Márple data order = 5, with Hamming window
Figure 5.7c. GSM spectrum of Kay&Marpine data
order = 6, no window
Figure 5.7d. CSM spectrum of Kay&Marple data
order= 6, with Hamming window
Figure 5.7f. CSM spectrum of Kay&Marple data
order= 10, with Hamming window
COMPOSITE SINUSOIDAL MODEL

Figure 5.8a. quasi-stationary signal, \( \text{SNR}=17 \) dB
order = 2, - real frequency, . estimated frequency
COMPOSITE SINUSOIDAL MODEL

Figure 5.8b. quasi-stationary signal, SNR= 3 dB
order= 2, - real frequency, . estimated frequency
COMPOSITE SINUSOIDAL MODEL

Figure 5.8c. quasi-stationary signal, $\text{SNR}=-0.5\text{dB}$

order= 2, - real frequency, . estimated frequency
COMPOSITE SINUSOIDAL MODEL

Figure 5.9a. rapid time varying frequency, SNR = 17 dB
- real frequency, + unmodified CSM, . modified CSM,
order = 1
COMPOSITE SINUSOIDAL MODEL

Figure 5.9b. rapid time varying frequency, SNR = 9 dB
- real frequency, + unmodified CSM, . modified CSM,
order = 1
Figure 5.9c. Rapid time varying frequency, SNR = 6.1 dB
- real frequency, + unmodified CSM, . modified CSM,
order = 1
5.5 How to estimate the rate of change of frequency of the rapid time varying frequency signal.

In practical, the rate of change of frequency $\Delta w$, is an unknown parameter. The analysis in section (2.3), (3.3), (3.4) and (4.2) above show that in order to be able to track the rapid changing frequency, $\Delta w$ is required. The rate of change of frequency $\Delta w$ can be obtained as follows.

From section (2.3), (3.3), (3.4) and (4.2), the estimated frequency of rapid changing frequency by using the FFT, the unmodified AR or the unmodified CSM will be

$$ \hat{\omega}_t = \omega_t + t \Delta w $$

where $\hat{\omega}_t$ is an estimated angular frequency at time $t$,

$\omega_t$ is an expected angular frequency at time $t$

and $\Delta w$ is a rate of change of angular frequency at time $t$ (the amount of change of angular frequency from sample to sample).

Substituting $t = N$ and $t = 2N$ in the above equation, yields

$$ \hat{\omega}_N = \omega_N + N \Delta w $$  \hspace{1cm} (5.5a)

$$ \hat{\omega}_{2N} = \omega_{2N} + 2N \Delta w $$  \hspace{1cm} (5.5b)

Assuming, that frequency is linearly changing, therefore $\Delta w$ is constant within the interval of data used to estimate the frequency at time $t$, ie.

$$ \omega_{2N} = \omega_N + N \Delta w $$  \hspace{1cm} (5.6)

Substituting eq. (5.6) into eq. (5.5b)

$$ \hat{\omega}_{2N} = \omega_N + 3N \Delta w $$  \hspace{1cm} (5.7)

Subtract eq. (5.5a) from eq. (5.7), yields

$$ \hat{\omega}_{2N} - \hat{\omega}_N = 2N \Delta w $$

or

$$ \Delta w = (\hat{\omega}_{2N} - \hat{\omega}_N) / 2N $$
In this chapter, we will use the frequency tracking algorithm described in the previous chapters to track the frequency of formants (peaks of the envelop of speech spectrum). Since speech signal is locally stationary signal\[24\], so the fixed frequency algorithms will be used to track the formants. The results of various preprocessing conditions are presented.

6.1 Sampling the speech signal

The following steps was used to sample the speech signal.

step 1. Recording the speech signal by using cassette tape recorder.

step 2. Playing back the speech signal through A/D converter board of the apple II computer and then sampling the data by using sampling program written in Assembly language as shown in appendix(7). The sampling rate 10 Khz is used, and the data is sampled for 0.1 second, therefore 10 Kilobytes of memories are required. Then the sampled data(in binary numbers, 8bits per byte) are stored in a diskette.

step 3. Sending the sampled data from the diskette to IBM 370 main frame computer.
changing binary data file to text file (hexadecimal number) by using Bin-to-Text program, appendix (8).

-sending Text data file to IBM by using Apple II Data-capture Package.

step 4. Changing hexadecimal numbers to decimal number, appendix (9).

6.2 Preprocessing

6.2.1 Windowing

In block data processing, the given time series are divided into small blocks of data, which the slow varying frequencies can be assumed to be stationary, and use the data within each block to estimate the frequency. If a block of data is closely considered, it looks like applying the rectangular window on the sequence of time series sampled data. It was shown in many digital signal processing texts [25-26] that the spectral of rectangular window has the sidelobes effect which could cause the large bias (may cause a small spurious peak or lose a small peak) at the frequency within the vicinity of the sidelobes. The examples in figure (6.6f) and (6.6k) compare with figure (6.6g) and (6.6l) respectively show the possibility of losing a peak (at 0.18Khz in figure (6.6f-g) and at 0.46Khz in figure (6.6k-l)); on the other hand, comparison between figure (6.6a) and (6.6b) show the possibility of exaggerating spurious peaks at 0.35Khz and 0.43 Khz). To reduce the sidelobes error of the rectangular window, various
nonrectangular windows were proposed. In speech analysis, the Hamming or Hanning window is considered to be an appropriate window [27-28]. The choice between this two windows depends upon one's preference. The Hamming and Hanning are defined as follows,

\[
W_n^{(Ham)} = 0.54 - 0.46 \cos\left(\frac{2\pi n}{N}\right) \quad (6.1)
\]
\[
W_n^{(Han)} = 0.5 - 0.5 \cos\left(\frac{2\pi n}{N}\right) \quad (6.2)
\]

where \( n = 0, 1, \ldots, N-1 \)

The comparison of these two windows in speech analysis are shown in figure(6.6b-c), (6.6g-h) and (6.6l-m). It can be noted that the results are similar.

6.2.2 Preemphasis

Since speech spectrum has -6dB/octave slope as one can see in figure(6.3a,e,j), (6.6a,f,k) and (6.9a,c,e), it is difficult to detect the high frequency peaks. Therefore if the speech spectrum is flattened by applying 6dB/octave preemphasis on the speech signal, the high frequencies peaks should be more accurately detected[29]. The detail description why preemphasis is needed for speech analysis can be found in [27-34]. Preemphasis on speech data, also has advantages as described by Tierney [33].

If speech spectrum is obtained via AR process, by using the proper speech preemphasis the order of AR model used to match the speech signal is reduced, it means the computation time also has been reduced.

Also, the proper preemphasis will reduce numerical
errors, which are arisen from the truncation of the numbers. Therefore, if the fixed point analysis is necessary or the computation are performed on a small computer, the proper preemphasis is really needed.

We can preemphasis the speech signal by using hardware filter (preemphasis before sampling) or software filter (preemphasis after sampling). It was shown by Markel & Gray[32], that by using preemphasis filter $P(Z)$ expressed as

$$P(Z) = 1 - uZ^{-1}$$  \hspace{1cm} (6.3)$$

and let $u$ close to unity, $P(Z)$ will be an approximate 6dB/octave preemphasis filter. The easiest way to achieve 6dB/octave preemphasis speech sampled data is by differencing the data as follows:

$$y_i = s_i - s_{i-1}$$  \hspace{1cm} (6.4)$$

where $s_i$ is sampled speech data and $y_i$ is preemphasis speech data. The reason, why the differencing data is equivalent to the 6dB/octave prefILTERED data, is as following. Taking Z transform of eq.(6.4)

$$Y(Z) = S(Z) (1 - Z^{-1})$$  \hspace{1cm} (6.5)$$

Let us closely consider eq.(6.5), it can be seen that the spectrum of differencing data $Y(Z)$ is indeed the spectrum of the original speech data without differencing $S(Z)$ multiply by the preemphasis filter $P(Z)$ eq.(6.3) with $u=1$; as mentioned above when $u \neq 1 P(Z)$ will act as 6dB/octave
Finally, the preemphasis windowing sampled data will be in the form

\[ y_t = c \ W_t \ (s_t - s_{t-1}) \]  \hspace{1cm} (6.6)

where \( c \) is the constant to make the rms of the window equal to unity; (for Hamming Window \( c=1.5863 \), while \( c=\sqrt{8/3} =1.6329 \) for Hanning window [25]), since we do not include the window in the calculation, the preemphasis windowing sampled data in the form of eq.(6.6) was first used in speech analysis by Makel [32].
6.3 Formant frequencies tracking by FFT

The FFT was first used by Oppenheim[35] to produce the speech spectrograms where formant frequencies are plotted versus time axis. In his paper he displayed the spectrogram on the oscilloscope with z axis (intensity) input, (where z axis was used to represent the amplitude of the formants) and then he reproduced a hard copy of spectrogram by taking picture of the screen in the oscilloscope. But here, the sliding window peak picking will be used to pick up the formant frequencies from the speech spectrum after that the formant frequencies are plotted on the normal paper by using HP-7225A plotter (Hewlett-Packard). The reason why the sliding window peak picking technique has been used to estimate the formant frequencies from the FFT speech spectrum, because the FFT speech spectrum has a noisy characteristic as can be seen in figure (6.3a-1). The idea of the sliding window peak picking technique is as following, sliding the small window whose size is approximate 1/10 of the size of the FFT along the frequency axis of the FFT speech spectrum, the maximum peak in the window is picked up to be a formant of the speech signal.

Figure (6.1) demonstrates the idea of the sliding window peak picking.

The utterance 'We were away' (spoken by 85 years old women which have been sampling at 10 Khz by the procedure mentioned in section 6.1) is used in the experiments.
Figure (6.2a-f) show the analysis results of the utterance 'we were away' by using the sound spectrograph machine 700 series, (Voice Identification, inc.). The sampled data of this utterance will be divided into 62 small intervals, each interval has 128 points of data. The interval of 128 points of sampled data will be used in the experiments of every methods in order to be able to compare the results. Figure (6.3a-l) show the FFT speech spectrum of the tenth, thirtyfifth and sixtysecond interval of the utterance 'we were away'. The speech spectrograms which are produced by using 128-point FFT with various preprocessing conditions are shown in Figures (6.4a-l). Figure (6.4g,l) and (6.2a-b) show that the narrow sliding window and wide sliding window FFT speech spectrogram are equivalent to the narrow band (45Hz) and wide band (300Hz) speech spectrogram from the sound spectrograph machine 700 series, respectively.
Figure 6.1. Sliding window peak picking.
Figure 6.2a. Wide band spectrogram of the utterance 'we were away' from sound spectrograph machine 700 series

time scale: 9.5 ms/div.

wide band analysis (300hz)

1Khz
2Khz
3Khz

Figure 6.2b. Narrow band spectrogram of the utterance 'we were away' from sound spectrograph machine 700 series

time scale: 9.5 ms/div.

narrow band analysis (45hz)

1Khz
2Khz
3Khz
Figure 6.2c. Wide band contour spectrogram of the utterance 'we were away' from sound spectrograph machine 700 series

Figure 6.2d. Narrow band contour spectrogram of the utterance 'we were away' from sound spectrograph machine 700 series
Figure 6.2e. Wide band spectrum of 10th, 35th and 62nd interval of the utterance 'we were away'.

Figure 6.2f. Narrow band spectrum of 10th, 35th and 62nd interval of the utterance 'we were away'.

time scale: 9.5 ms/div.
Figure 6.3a. 128-FFT speech spectrum of 10th interval of the utterance 'we were away', no window
Figure 6.3b. 128-PRT speech spectrum of 10th interval of the utterance 'we were away', Hamming window
Figure 6.3c. 128-point speech spectrum of 10th interval of preemphasis pulse 'we were away', 6dB/octave
Figure 6.3d. 128-FFT speech spectrum of 10th interval of the utterance 'we were away', Hamming window, 6dB/octave preemphasis
Figure 6.3e. 128-FFT speech spectrum of 35th interval of the utterance 'we were away', no window
Figure 6.3f. 128-FFT speech spectrum of 35th interval of the utterance 'we were away', Hamming window
Figure 6.3g. 128-FFT speech spectrum of 35th interval of the utterance "we were away", 6dB/octave preemphasis.
Figure 6.3h. 128-FFT speech spectrum of 35th interval of the utterance 'we were away', Hamming window, 6dB/octave preemphasis
Figure 6.31. 128-FFT speech spectrum of 62th interval of the utterance 'we were away', no window
Figure 6.3j. 128-FFT speech spectrum of 62th interval of the utterance 'we were away', Hamming window
Figure 6.3k. 128-FFT speech spectrum of 62\textsuperscript{th} interval of the utterance 'we were away', 6dB/octave preemphasis
Figure 6.31. 128-FFT speech spectrum of 62nd interval of the utterance 'we were away', Hamming window, 6dB/octave preemphasis
FFT & RECTANGULAR WINDOW

Figure 6.4a. FFT spectrogram of the utterance 'we were away', sliding window width = 1660 hz
Figure 6.4b. FFT spectrogram of the utterance 'we were away', sliding window width = 1660 Hz, 6dB/octave preemphasis.
FFT & HAMMING WINDOW

Figure 6.4c. FFT spectrogram of the utterance 'we were away', sliding window width = 1660 hz
FFT & HAMMING WINDOW

Figure 6.4d. FFT spectrogram of the utterance 'we were away', sliding window width = 1660, 66dB/octave preemphasis
Figure 6.4e. FFT spectrogram of the utterance 'we were away', sliding window width = 1660 hz
FFT & HANNING WINDOW

Figure 6.4f. FFT spectrogram of the utterance 'we were away', sliding window width = 1660 hz, 6dB/octave preemphasis
Figure 6.4g. FFT spectrogram of the utterance 'we were away', sliding window width = 1660 hz, 6dB/octave preemphasis, only peaks above 0.5 are plotted.
FFT & RECTANGULAR WINDOW

Figure 6.4b. FFT spectrogram of the utterance 'we were away', sliding window width = 1162 hz
Figure 6.41. FFT spectrogram of the utterance 'we were away', sliding window width = 1162 hz, 6db/octave preemphasis
**FFT & HAMMING WINDOW**

Figure 6.4j. FFT spectrogram of the utterance 'we were away', sliding window width = 1162 hz
FFT & HAMMING WINDOW

Figure 6.4k. FFT spectrogram of the utterance 'we were away', sliding window width = 1162 hz, 6dB/octave preemphasis
FFT & HAMMING WINDOW

Figure 6.41. FFT spectrogram of the utterance 'we were away', sliding window width = 913 Hz, 6dB/octave preemphasis, only peaks above 0.5 are plotted.
6.4 Formant frequencies tracking by AR model

In order to track the formant frequencies by using AR model; first, the optimum AR coefficients have to be obtained, in the experiments both block data processing and adaptive processing are used. In block data processing, Burg's algorithm[12] and Marple's algorithm[13] will be used to estimate the AR coefficients, while Widrow & Hoff algorithm[15] is used as adaptive processing. Second, the AR coefficients are substituted into the frequency estimator, since the sampling frequency is not equal to one hz, the frequency estimator eq.(2.9) has to be changed to

\[ Q(w) = \frac{1}{1 + \sum_{i=1}^{M} a_i z^{-iT}} \]

where \( T \) denotes sampling period. Finally, the estimated formant frequencies is obtained by estimating the peaks of the frequency estimator \( Q(w) \) eq.(6.7). Since the speech spectrum which are obtained from \( Q(w) \) has the smooth characteristic as shown in Figures(6.6a-o); so the simple peak picking appendix(10) can be used to find the peaks of the AR speech spectrum. A flow chart of the AR process is shown in appendix(5). Figures(6.6a-o) and (6.7a-k) show the AR speech spectrums and the AR speech spectrograms of the same utterance 'we were away' with various preprocessing conditions respectively. The appropriate order of AR model (p) which fits with the given speech signal can be computed by following Markel & Gray suggestion [34] as follows
\[ p = f_s + (4 \text{ or } 5) \]

where \( p \) and \( f \) denote an appropriate AR order and sampling frequency in Khz, respectively. Here, the sampling is chosen to be 10 Khz, so the appropriate order \( p=14 \) and 15 has been used. The examples in Figures (6.7d-e) show the results using \( p=12 \) and \( p=15 \). It can be seen that when order is low the AR model lost its characteristic to separate two close frequencies. The similarity between the AR speech spectrogram and the wide band speech spectrogram from the sound spectrograph machine 700 series can be seen in Figures (6.7d,e,g,h) and (6.2a). Figure (6.7f) and (6.2b) show that the AR model can be also used to obtain the narrow band spectrogram by increasing the order of the AR model.
Figure 6.6a. AR speech spectrum of 10th interval of the utterance 'we were away', AR coeff. by Burg, order=14
Figure 6.6b. AR speech spectrum of 10th interval of the utterance 'we were away', AR coeff. by Burg, order=14, Hamming window
Figure 6.6c. AR speech spectrum of 10th interval of the utterance 'we were away', AR coeff. by Burg, order=14, Hanning window
Figure 6.6d. AR speech spectrum of 10\textsuperscript{th} interval of the utterance 'we were away', AR coeff. by Burg, order=14, 6dB/octave preemphasis
Figure 6.6e. AR speech spectrum of 10th interval of the utterance 'we were away', AR coeff. by Burg, order=14, Hamming window, 6dB/octave preemphasis
Figure 6.6f. AR speech spectrum of 35th interval of the utterance 'we were away', AR coeff. by Burg, order=14
Figure 6.6g. AR speech spectrum of 35th interval of
the utterance 'we were away', AR coeff. by Burg, order=14,
Hamming window.
Figure 6.6h. AR speech spectrum of 35th interval of the utterance 'we were away', AR coeff. by Burg, order=14, Hanning window
Figure 6.6i. AR speech spectrum of 35th interval of the utterance 'we were away', AR coeff. by Burg, order=14 6dB/octave preemphasis
Figure 6.6j. AR speech spectrum of 35th interval of the utterance 'we were away', AR coeff. by Burg, order=14, Hamming window, 6db/octave preemphasis
Figure 6.6k. AR speech spectrum of 62\textsuperscript{th} interval of the utterance 'we were away', AR coeff. by Burg, order=14
Figure 6.61. AR speech spectrum of 62\textsuperscript{th} interval of the utterance 'we were away', AR coeff. by Burg, order=14, Hamming window
Figure 6.6m. AR speech spectrum of 62th interval of the utterance 'we were away', AR coeff. by Burg, order=14, Hanning window
Figure 6. AR speech spectrum of 62th interval of 6 dB/octave preemphasis.

The utterance was away, AR coeff. by Burg, order=14.
Figure 6.60. AR speech spectrum of 62\textsuperscript{th} interval of
the utterance 'we were away', AR Coeff. By Burg, order=14,
Hamming window, 6dB/octave preemphasis
AR COEF. BY BURG

Figure 6.7a. AR spectrogram of the utterance 'we were away', order = 15
AR COEF. BY BURG

Figure 6.7b. AR spectrogram of the utterance 'we were away', order = 15, Hamming window, 6dB/octave preemphasis
AR COEF. BY BURG

Figure 6.7c. AR spectrogram of the utterance 'we were away', order = 12, Hamming window, 6dB/octave preemphasis
Figure 6.7d. AR spectrogram of the utterance 'we were away', order = 12, Hamming window, 6dB/octave preemphasis, only peaks above 0.5 are plotted.
Figure 6.7c. AR spectrogram of the utterance 'we were away', order = 15, Hann window, 60 dB/octave preemphasis, only peaks above 0.5 are plotted.
Figure 6.7f. AR spectrum of the utterance 'we were away', order = 30, Hamming window, 6dB/octave preemphasis, only peaks above 0.5 are plotted.
Figure 6.7g. AR spectrum of the utterance 'we were away', order = 15, Hamming window, 6dB/octave preemphasis, only peaks above 0.75 are plotted.
Figure 6.7b. AR spectrum of the utterance 'we were away', order = 15, Hamming window, 6dB/octave preemphasis, only peaks above 0.75 are plotted.
Figure 6.7i. AR spectrum of the utterance 'we were away', order = 15, alpha = 0.1
Figure 6.7j. AR spectrum of the utterance 'we were away', order = 15, alpha = 0.1, 6dB/octave preemphasis
Figure 6.7k. AR spectrum of the utterance 'we were away', order = 15, alpha = 0.1, 6dB/octave preemphasis, only peaks above 0.5 are plotted.
6.5 Formant frequencies tracking by CSM

The procedures which have been mentioned in section 4.1 and 4.3 will be used to track the formant frequencies of the given speech signal, which can be summarized into a flow chart as shown in appendix(6). The CSM speech spectrums and spectrograms of the same utterance 'We were away' with various condition of preprocessing are shown in Figures(6.9a-f) and (6.10a-d).

The appropriate order of the CSM which make the CSM fit to the given speech signal can be obtained from the experiment by increasing the order until the number of line of the estimated formant frequencies is less than the number of the CSM order. The maximum number of line of the estimated formant frequencies which does not change while increasing the number of the CSM order will be the appropriate order. The examples are shown in Figures(6.10b-c), in Figure(6.10c) the number of order is 6, the number of the estimated formant frequency lines is 6, but in Figure(6.10b) the number of order is 7 while the number of lines of the estimated formant frequencies is 6, so the appropriate CSM order which fit to the given speech signal in this experiment is 6. One can see from Figures(6.10d) and (6.2b) that the CSM speech spectrogram is equivalent to the narrow band analysis of the sound spectrograph machine 700 series.
Figure 6.9a. CSM speech spectrum of 10th interval of the utterance 'we were away', order = 6
Figure 6.9b. CSM speech spectrum of 10\textsuperscript{th} interval of the utterance 'we were away', Hamming window, 6dB/octave preemphasis, order = 6
Figure 6.9c. CSM speech spectrum of 35\textsuperscript{th} interval of the utterance 'we were away', order = 6
Figure 6.9d. CSM speech spectrum of 35th interval of the utterance 'we were away', order = 6, Hamming window, 6dB/octave preemphasis
Figure 6.9e. CSM speech spectrum of 62\textsuperscript{th} interval of the utterance 'we were away', order = 6
Figure 6.9f. CSM speech spectrum of 62th interval of the utterance 'we were away', order = 6, Hamming window, 6dB/octave preemphasis.
Figure 6.10a, CSM spectrogram of the utterance "we were away", order = 6

COMPOSITE SINUSOIDAL MODEL
**Composite Sinusoidal Model**

Figure 6.10b. CSM spectrogram of the utterance 'we were away', order = 7, Hamming window, 6dB/octave preemphasis.
COMPOSITE SINUSOIDAL MODEL

Figure 6.10c. CSM spectrogram of the utterance 'we were away', order = 6, Hamming window, 6dB/octave preemphasis
COMPOSITE SINUSOIDAL MODEL

Figure 6.10d. CSM spectrogram of the utterance 'we were away', order = 6, Hamming window, 6dB/octave preemphasis, only peaks above -2 are plotted.
In the thesis, three spectral estimation methods (FFT, AR and CSM) are used to track the frequencies of the given time series signal. It has been found that in order to be able to track the rapid time varying frequency accurately, all three methods have to be modified. The modified version of these three methods are proposed, and they are confirmed by the numerical simulation in chapter 5. The Hankel system of equations have to be solved during the process of the CSM method. Application of Kumar's equation to solve this Hankel system is developed and described in detail in section 4.3.

The advantages and disadvantages of these three methods can be summarized as follows. In the computational point of view, the number of multiplications will be used to compare the speed of each technique since multiplication takes time more than the other arithmetic operations. The number of multiplications at each estimation of the FFT, AR and CSM are approximately \(4M \times \log(M), \ M+n^2, \ M+p^2 +2p(N)\) respectively, where \(M, n, p\) and \(N\) denote the number of data used to estimate frequencies at time \(t\), the CSM order which is usually equal to the number of frequency lines being tracked, the AR order which is approximately twice of the number of frequency lines being tracked and the number of points of frequencies of AR spectrum being estimated,
respectively. When the number of frequency lines being tracked is small (e.g., one or two), the CSM is fastest, the FFT comes second; while the FFT will be the fastest and the CSM comes second when the number of frequency lines being tracked is large. If these three methods are compared in term of frequency resolution, the AR method has the highest frequency resolution. The FFT frequency resolution is a little bit better than the CSM. When the time series signal was interfered with high noise level (SNR is low), the FFT is the best technique which can still work well. The AR method is better than the CSM in term of noise resistivity.

In speech application, the FFT with narrow sliding window and with wide sliding window are equivalent to the narrow and wide band speech analysis by sound spectrograph machine 700 series, respectively. The AR method is fitted very well with wide band speech analysis. The narrow band speech analysis can also be achieved by the AR method by increasing the AR order with the cost of increasing computation time. The CSM method seems to be the best method to perform the narrow band speech analysis, since its computation time is less and its result Figure (6.10d) is matched very well with the narrow band spectrogram Figure (6.2b) from the spectrogram machine.
REFERENCES


36 IBM System/360 Scientific Subroutine Package, pp. 120-123.

APPENDIX 1

CHANGING HANKEL MATRIX TO TOEPLITZ MATRIX

**Definition** Exchange matrix \( J \) is a matrix whose element along across diagonal equal to unity and zero elsewhere. \( J \) can be written by

\[
J = \begin{bmatrix}
1 & 0 & 0 & \cdots \\
0 & 1 & 0 & \cdots \\
0 & 0 & 1 & \cdots \\
\vdots & \vdots & \vdots & \ddots
\end{bmatrix}
\]

Properties of Exchange Matrix are given as follows

1. If a matrix \( A \) is premultiplied by exchange matrix, it is equal to flipping matrix \( A \) up-down.

2. If a matrix \( A \) is postmultiplied by exchange matrix, it is equal to flipping matrix \( A \) left-right.

**Definition** Hankel matrix \( H \) is a matrix whose elements along the lines parallel to the cross diagonal are equal, in the other words the sum of row and column index of the elements along the lines parallel to the cross diagonal are the same (constant), ie.

\[
H = [H_{i,j}] = [H_{i+j}]
\]

**Definition** Toeplitz matrix \( T \) is a matrix whose elements along the lines parallel to the diagonal are equal, in the other words the difference between the row and column index of the elements along the line parallel to the diagonal are
the same (constant), ie.

\[ T = [T_{i,j}] = [T_{i-j}] \]

**THEOREM** If \( H \) is premultiplied by \( J \), the result matrix is a Toeplitz matrix.

**PROVE** Premultiplying Hankel matrix \( H \) by \( J \).

\[ [JH] = [H_{n-i+1}, j] \]

Subtracting the column index \( j \) from the row index \( n-i+1 \), and using the Hankel matrix properties (row index + column index = constant); yields

\[ n-i+1-j = n+1-(i+j) = \text{constant} \]

From the definition of Toeplitz above, it is obviously that the product of \( JH \) is a Toeplitz matrix. By the similar manner, it can be shown that the product of \( HJ \) is also a Toeplitz matrix.
Suppose that a following matrix equation is given

\[ PA = \hat{\mathbf{C}} \]  \hspace{1cm} (1)

where \( P \), \( A \) and \( \hat{\mathbf{C}} \) denote a \((nxn)\) matrix, a column vector whose first element is equal to unity and a column vector whose elements are all zero except first element, respectively. Vector \( A \) and \( \hat{\mathbf{C}} \) can be generally expressed by

\[
\begin{align*}
A &= (1 \ a_2 \ a_3 \ \ldots \ \ldots \ a_n)^t \\
\hat{\mathbf{C}} &= (\mathbf{C} \ 0 \ 0 \ \ldots \ \ldots \ 0)^t
\end{align*}
\]

where \( t \) denotes transpose operator. Denoting the inverse matrix of \( P \) by \( Q \), and the first column of \( Q \) by \( \bar{Q} \), where \( \bar{Q} = (q_1 \ q_2 \ q_3 \ \ldots \ \ldots \ q_n)^t \). From the matrix inversion properties

\[ PQ = \mathbf{E} \]

where \( \mathbf{E} \) is an unit vector \((1 \ 0 \ 0 \ \ldots \ 0)^t \). Dividing the above equation by \( q_1 \), yields

\[
(1/q_1) PQ = (1/q_1)(E)
= (1/q_1 \ 0 \ 0 \ 0 \ \ldots \ \ldots \ 0)^t
\]  \hspace{1cm} (2)

Comparision between eq. (1) and (2), one can conclude that the first column of the inverse matrix of \( P \) dividing its first element is a solution of the matrix equation expressed in eq. (1).
THEOREM The first row of the product of two cyclic shift right matrices is equivalent to the circular convolution of the first rows.

PROVE Let $P, Q$ denote $(n \times n)$ cyclic shift right matrices.

\[
P = \begin{bmatrix}
p_1 & p_2 & p_3 & \ldots & p_n \\
p_n & p_1 & p_2 & \ldots & p_{n-1} \\
\cdot & \cdot & \cdot & \ddots & \cdot \\
\cdot & \cdot & \cdot & \ddots & \cdot \\
\cdot & \cdot & \cdot & \ddots & \cdot \\
p_2 & p_3 & \cdot & \ldots & p_1
\end{bmatrix}
\]

\[
Q = \begin{bmatrix}
q_1 & q_2 & q_3 & \ldots & q_n \\
q_n & q_1 & q_2 & \ldots & q_{n-1} \\
\cdot & \cdot & \cdot & \ddots & \cdot \\
\cdot & \cdot & \cdot & \ddots & \cdot \\
\cdot & \cdot & \cdot & \ddots & \cdot \\
q_2 & q_3 & \cdot & \ldots & q_1
\end{bmatrix}
\]
let \[ R = PQ \]

where

\[
R = \begin{bmatrix}
    r_{11} & r_{12} & \ldots & r_{1n} \\
    r_{21} & r_{22} & \ldots & r_{2n} \\
    \cdot & \cdot & \ddots & \cdot \\
    \cdot & \cdot & \ldots & \cdot \\
    r_{n1} & r_{n2} & \ldots & r_{nn}
\end{bmatrix}
\]

From the matrix multiplication definition,

\[
\begin{align*}
    r_{11} &= p_{1}q_{1} + p_{2}q_{n} + p_{3}q_{n-1} + \ldots + p_{n}q_{2} \\
    r_{12} &= p_{1}q_{2} + p_{2}q_{1} + \ldots + p_{n}q_{3} \\
    \cdot & \cdot \cdot \cdot \\
    \cdot & \cdot \cdot \cdot \\
    r_{1n} &= p_{1}q_{n} + p_{2}q_{n-1} + \ldots + p_{n}q_{1}
\end{align*}
\]

From the above equation and the definition of circular convolution, it is clear that the circular convolution of the first row of two cyclic shift right is equivalent to the first row of the product of these two matrices.
<table>
<thead>
<tr>
<th>X(1)</th>
<th>1.291061</th>
<th>X(13)</th>
<th>0.289648</th>
</tr>
</thead>
<tbody>
<tr>
<td>X(2)</td>
<td>-3.806368</td>
<td>X(14)</td>
<td>1.212952</td>
</tr>
<tr>
<td>X(3)</td>
<td>-1.691316</td>
<td>X(15)</td>
<td>-0.119905</td>
</tr>
<tr>
<td>X(4)</td>
<td>1.243138</td>
<td>X(16)</td>
<td>-0.441688</td>
</tr>
<tr>
<td>X(5)</td>
<td>1.641872</td>
<td>X(17)</td>
<td>-0.879723</td>
</tr>
<tr>
<td>X(6)</td>
<td>-0.095528</td>
<td>X(18)</td>
<td>0.355181</td>
</tr>
<tr>
<td>X(7)</td>
<td>-1.659398</td>
<td>X(19)</td>
<td>0.735431</td>
</tr>
<tr>
<td>X(8)</td>
<td>-1.11467</td>
<td>X(20)</td>
<td>0.195958</td>
</tr>
<tr>
<td>X(9)</td>
<td>0.985886</td>
<td>X(21)</td>
<td>-0.342332</td>
</tr>
<tr>
<td>X(10)</td>
<td>1.991779</td>
<td>X(22)</td>
<td>-0.529788</td>
</tr>
<tr>
<td>X(11)</td>
<td>-0.046613</td>
<td>X(23)</td>
<td>0.197961</td>
</tr>
<tr>
<td>X(12)</td>
<td>1.649269</td>
<td>X(24)</td>
<td>0.671779</td>
</tr>
<tr>
<td>X(13)</td>
<td>1.049118</td>
<td>X(25)</td>
<td>0.185931</td>
</tr>
<tr>
<td>X(14)</td>
<td>1.854663</td>
<td>X(26)</td>
<td>-0.328923</td>
</tr>
<tr>
<td>X(15)</td>
<td>1.858916</td>
<td>X(27)</td>
<td>-0.366692</td>
</tr>
<tr>
<td>X(16)</td>
<td>-0.551182</td>
<td>X(28)</td>
<td>0.369647</td>
</tr>
<tr>
<td>X(17)</td>
<td>-1.476497</td>
<td>X(29)</td>
<td>-0.191325</td>
</tr>
<tr>
<td>X(18)</td>
<td>-4.212242</td>
<td>X(30)</td>
<td>0.519116</td>
</tr>
<tr>
<td>X(19)</td>
<td>0.768282</td>
<td>X(31)</td>
<td>0.885232</td>
</tr>
<tr>
<td>X(20)</td>
<td>1.415883</td>
<td>X(32)</td>
<td>-0.425946</td>
</tr>
<tr>
<td>X(21)</td>
<td>0.195882</td>
<td>X(33)</td>
<td>0.851478</td>
</tr>
<tr>
<td>X(22)</td>
<td>-0.027026</td>
<td>X(34)</td>
<td>-0.639979</td>
</tr>
<tr>
<td>X(23)</td>
<td>-0.463577</td>
<td>X(35)</td>
<td>-0.344389</td>
</tr>
<tr>
<td>X(24)</td>
<td>1.664913</td>
<td>X(36)</td>
<td>0.814138</td>
</tr>
<tr>
<td>X(25)</td>
<td>0.614114</td>
<td>X(37)</td>
<td>-0.356169</td>
</tr>
<tr>
<td>X(26)</td>
<td>-0.731469</td>
<td>X(38)</td>
<td>0.854213</td>
</tr>
<tr>
<td>X(27)</td>
<td>-1.195311</td>
<td>X(39)</td>
<td>-0.388088</td>
</tr>
<tr>
<td>X(28)</td>
<td>0.119881</td>
<td>X(40)</td>
<td>-0.163839</td>
</tr>
<tr>
<td>X(29)</td>
<td>0.607636</td>
<td>X(41)</td>
<td>1.189961</td>
</tr>
<tr>
<td>X(30)</td>
<td>0.092536</td>
<td>X(42)</td>
<td>0.114536</td>
</tr>
<tr>
<td>X(31)</td>
<td>-0.812734</td>
<td>X(43)</td>
<td>-0.667526</td>
</tr>
<tr>
<td>X(32)</td>
<td>-1.763842</td>
<td>X(44)</td>
<td>-0.814997</td>
</tr>
</tbody>
</table>
APPENDIX 5
AR PROCESS

start

read data (one interval)

compute optimum AR coefficients

estimate frequencies by finding peak locations of frequency estimator, \( Q(W) \)

end of data file

plot

stop
APPENDIX 6

CSM PROCESS

1. Start

2. Read data (one interval)

3. Compute autocorrelation function eq. (4.3)

4. Compute $U(1)$ eq. (4.7)

5. Rearrange $U(1)$ to form Toeplitz system eq. (4.11b)

6. Solve Toeplitz system

7. Find the root $(x_i)$ of polynomial eq. (4.9)

8. Estimate frequencies by $w_i = \cos^{-1}(x_i)$
I estimate amplitude by solving eq. (4.12).
### APPENDIX 7

#### SAMPLING DATA

```
<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0000</td>
<td>INPUT</td>
<td>EQU $C0DF</td>
</tr>
<tr>
<td>2</td>
<td>0000</td>
<td>STORE</td>
<td>EQU $0006</td>
</tr>
<tr>
<td>3</td>
<td>0000</td>
<td>M</td>
<td>EQU $4000</td>
</tr>
<tr>
<td>4</td>
<td>0000</td>
<td>M1</td>
<td>EQU $4001</td>
</tr>
<tr>
<td>5</td>
<td>0000</td>
<td>M2</td>
<td>EQU $4002</td>
</tr>
<tr>
<td>6</td>
<td>0000</td>
<td>T</td>
<td>EQU $4003</td>
</tr>
<tr>
<td>7</td>
<td>4010</td>
<td></td>
<td>ORG $4010</td>
</tr>
<tr>
<td>8</td>
<td>4010 A9 00</td>
<td></td>
<td>LDA =$00</td>
</tr>
<tr>
<td>9</td>
<td>4012 85 06</td>
<td></td>
<td>STA STORE</td>
</tr>
<tr>
<td>10</td>
<td>4014 A9 41</td>
<td></td>
<td>LDA =$41</td>
</tr>
<tr>
<td>11</td>
<td>4016 85 07</td>
<td></td>
<td>STA STORE+1</td>
</tr>
<tr>
<td>12</td>
<td>4018 AD DF CO</td>
<td></td>
<td>LDA INPUT</td>
</tr>
<tr>
<td>13</td>
<td>401B AD DF CO</td>
<td></td>
<td>LDA INPUT</td>
</tr>
<tr>
<td>14</td>
<td>401E 8D 00 40 AAA</td>
<td></td>
<td>STA M</td>
</tr>
<tr>
<td>15</td>
<td>4021 EA</td>
<td></td>
<td>NOP</td>
</tr>
<tr>
<td>16</td>
<td>4022 EA</td>
<td></td>
<td>NOP</td>
</tr>
<tr>
<td>17</td>
<td>4023 EA</td>
<td></td>
<td>NOP</td>
</tr>
<tr>
<td>18</td>
<td>4024 EA</td>
<td></td>
<td>NOP</td>
</tr>
<tr>
<td>19</td>
<td>4025 AD DF CO</td>
<td></td>
<td>LDA INPUT</td>
</tr>
<tr>
<td>20</td>
<td>4028 CD 00 40</td>
<td></td>
<td>CMP M</td>
</tr>
<tr>
<td>21</td>
<td>402B 30 03</td>
<td></td>
<td>BMI BBB</td>
</tr>
<tr>
<td>22</td>
<td>402D 4C 42 40</td>
<td></td>
<td>JMP CCC</td>
</tr>
<tr>
<td>23</td>
<td>4030 8D 01 40 BBB</td>
<td></td>
<td>STA M1</td>
</tr>
<tr>
<td>24</td>
<td>4033 AD 00 40</td>
<td></td>
<td>LDA M</td>
</tr>
<tr>
<td>25</td>
<td>4036 8D 02 40</td>
<td></td>
<td>STA M2</td>
</tr>
<tr>
<td>26</td>
<td>4039 AD 01 40</td>
<td></td>
<td>LDA M1</td>
</tr>
<tr>
<td>27</td>
<td>403C 8D 00 40</td>
<td></td>
<td>STA M</td>
</tr>
<tr>
<td>28</td>
<td>403F AD 02 40</td>
<td></td>
<td>LDA M2</td>
</tr>
<tr>
<td>29</td>
<td>4042 A2 00</td>
<td>CCC</td>
<td>LDX =$00</td>
</tr>
<tr>
<td>30</td>
<td>4044 CD 00 40</td>
<td></td>
<td>CMP M</td>
</tr>
<tr>
<td>31</td>
<td>4047 F0 05</td>
<td></td>
<td>BEQ AAA</td>
</tr>
<tr>
<td>32</td>
<td>4049 EE 00 40</td>
<td>DDD</td>
<td>INC M</td>
</tr>
<tr>
<td>33</td>
<td>404C CD 00 40</td>
<td></td>
<td>CMP M</td>
</tr>
<tr>
<td>34</td>
<td>404F F0 CD</td>
<td></td>
<td>BEQ AAA</td>
</tr>
<tr>
<td>35</td>
<td>4051 E8</td>
<td></td>
<td>INX</td>
</tr>
<tr>
<td>36</td>
<td>4052 E0 15</td>
<td></td>
<td>CPX =$15</td>
</tr>
<tr>
<td>37</td>
<td>4054 DC F3</td>
<td></td>
<td>BNE DDD</td>
</tr>
<tr>
<td>38</td>
<td>4056 AD 00</td>
<td></td>
<td>LDY =$00</td>
</tr>
<tr>
<td>39</td>
<td>4058 AD DF CO EEE</td>
<td></td>
<td>LDA INPUT</td>
</tr>
<tr>
<td>40</td>
<td>405B 91 06</td>
<td></td>
<td>STA (STORE)+Y</td>
</tr>
<tr>
<td>41</td>
<td>405D 20 6F 40</td>
<td></td>
<td>JSR DLAY</td>
</tr>
<tr>
<td>42</td>
<td>4060 EA</td>
<td></td>
<td>NOP</td>
</tr>
<tr>
<td>43</td>
<td>4061 EA</td>
<td></td>
<td>NOP</td>
</tr>
<tr>
<td>44</td>
<td>4062 EA</td>
<td></td>
<td>NOP</td>
</tr>
<tr>
<td>45</td>
<td>4063 CB</td>
<td></td>
<td>INY</td>
</tr>
<tr>
<td>46</td>
<td>4064 D0 F2</td>
<td></td>
<td>BNE EEE</td>
</tr>
<tr>
<td>47</td>
<td>4066 E6 07</td>
<td></td>
<td>INC STORE+1</td>
</tr>
<tr>
<td>48</td>
<td>4068 A5 07</td>
<td></td>
<td>LDA STORE+1</td>
</tr>
<tr>
<td>49</td>
<td>406A C9 42</td>
<td></td>
<td>CMP =$42</td>
</tr>
<tr>
<td>50</td>
<td>406C D0 EA</td>
<td></td>
<td>BNE EEE</td>
</tr>
<tr>
<td>51</td>
<td>406E 00</td>
<td></td>
<td>BRK</td>
</tr>
<tr>
<td>52</td>
<td>406F A9 07</td>
<td>DLAY</td>
<td>LDA =$07</td>
</tr>
<tr>
<td>53</td>
<td>4071 8D 03 40</td>
<td></td>
<td>STA T</td>
</tr>
<tr>
<td>54</td>
<td>4074 CE 03 40</td>
<td>LOOP</td>
<td>DEC T</td>
</tr>
<tr>
<td>55</td>
<td>4077 D0 FB</td>
<td></td>
<td>BNE LOOP</td>
</tr>
<tr>
<td>56</td>
<td>4079 60</td>
<td></td>
<td>RTS</td>
</tr>
<tr>
<td>57</td>
<td></td>
<td>END</td>
<td></td>
</tr>
</tbody>
</table>
```

; read input signal, store input signal if it is higher than the given level.

; set level (start storing input signal)

; store input signal

; delay subroutine approximate 0.1 ms.
APPENDIX 8

BIN TO TEXT

20 PRINT "MAXFILES3"
25 D$ = CHR$ (4)
30 DIM H$(16)
40 DATA 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F
50 FOR I = 0 TO 15: READ H$(I): NEXT I
100 HOME
110 INVERSE : HTAB 2: PRINT "BINARY TO TEXT FILE CONVERSION"
120 NORMAL : PRINT
130 PRINT
140 INPUT "BINARY FILE TO CONVERT?";B$
150 INPUT "TEXT FILE TO CREATE?";T$
160 PRINT
170 INPUT "INSERT DISK AND PRESS RETURN";A$
172 POKE 34,6: POKE 35,20
174 VTab 22: PRINT "PRESS"; : INVERSE: PRINT "SPACE"; :
175 NORMAL : PRINT "TO STOP"
176 HOME
180 PRINT D$; "BLOAD"; B$; "", A$1000"
190 A = 0
200 L = PEEK (43616) + PEEK (43617) * 256
210 ONERR GOTO 230
220 PRINT D$; "DELETE"; T$
230 POKE 216,0
240 PRINT D$; "OPEN"; T$
245 PRINT D$; "MONO"
250 PRINT D$; "WRITE"; T$
300 N = A
310 N1 = INT(N/256); N2=N-N1*256
320 Z = N1: GOSUB 1000
330 Z = N2: GOSUB 1000
340 PRINT ";-";
350 Z = PEEK(N-A+4096); GOSUB 1000
360 N = N + 1
365 IF PEEK (-16384) = 160 THEN POKE -16368,0: GOTO 400
370 IF N > = A + L THEN 400
380 IF N = INT(N/16)*16 THEN PRINT: GOTO 310
390 GOTO 350
400 PRINT: PRINT D$; "CLOSE"
405 PRINT D$; "NOMONO"
410 PRINT "DONE"
420 TEXT: HOME: NEW: END
1000 Z$ = H$(INT(Z/16)) + H$(Z-INT(Z/16)*16)
1010 PRINT Z$; : RETURN
READ 128 POINTS OF HEXADECIMAL DATA
CHANGE HEXADECIMAL TO DECIMAL NUMBER
X = DATA OUTPUT
A = HEXADECIMAL REFERENCES NUMBER (0 1 ... F)

SUBROUTINE DATA(A,X)
DIMENSION A(16),X(256),OUT(16)
DO 1 I=1,8
L1=(I-1)*16
CALL BIN(A,OUT)
DO 2 L=1,16
LL=L1+L
X(LL)=OUT(L)
2 CONTINUE
1 CONTINUE
RETURN
END

SUBROUTINE BIN(A,OUT)
DIMENSION A(16)
DIMENSION OUT(16)
READ(5,6)
(32Z1)
CALL BTD(A,B,C,OUT)
RETURN
END

SUBROUTINE BTD(A,B,C,OUT)
DIMENSION OUT1(16),OUT0(16),OUT(16)
DIMENSION A(16),B(16),C(16)
DO 11 N=1,16
DO 1 I=1,16
IF(A(I).NE.C(N)) GO TO 1
OUT0(N)=FLOAT(I-1)
1 OUT0(N)=FLOAT(I-1)
11 CONTINUE
DO 22 N=1,16
DO 2 I=1,16
IF(A(I).NE.B(N)) GO TO 2
OUT1(N)=FLOAT(I-1)
2 CONTINUE
22 CONTINUE
DO 3 I=1,16
OUT(I)=16.0*OUT1(I)+OUT0(I)
OUT(I)=(OUT(I)-128.0)*5.0/128.0
3 CONTINUE
RETURN
END
C
C FIND PEAK LOCATION ON X-AXIS
C INPUT X(I), Y(I), N
C N = LENGTH OF X AND Y
C OUTPUT = PK(I), IPOINT
C PK = PEAK LOCATIONS
C IPOINT = NUMBER OF PEAKS
C
SUBROUTINE PEAK(N,X,Y,PK,IPOINT)
DIMENSION X(201), Y(201), PK(20)
N2=N-2
IPOINT=0
DO 1 I=1,N2
   DS1=Y(I+1)-Y(I)
   DS2=Y(I+2)-Y(I+1)
   DS=DS1*DS2
   IF(DS).LT.0,1,1
   IF(DS1).LT.0,1,3
   IPOINT=IPOINT+1
   PK(IPOINT)=X(I+1)
1 CONTINUE
RETURN
END
COMPUTE AR COEFFICIENT BY BURG ALGORITHM
THE ALGORITHM WAS REDERIVED BY MARPLE IN
IEEE ASSP-28, PP. 441-454

INPUT X(I), N, MORD
OUTPUT G(I)

X = INPUT DATA
N = LENGTH OF X
MORD = AR ORDER
G = OPTIMUM AR COEFFICIENT

SUBROUTINE BURG(N, X, MORD, G)
DIMENSION X(256), G(40)
DIMENSION F(500), B(500), GN(256)
M = 0
E = 0.0
DO 11 I = 1, N
11 E = E + X(I) * X(I)
E = E * 2.0
D = E
Q = 1
DO 22 I = 1, N
F(I) = X(I)
22 B(I) = X(I)

M = M + 1
A = 0.0
NM = N - M
DO 33 K = 1, NM
33 A = A + B(K) * F(K + 1)
D = (D * Q) - (B(N - M + 1) ** 2) - (F(1) ** 2)
G(M) = -2.0 * A / D
Q = 1.0 - G(M) ** 2
E = E * Q
IF (M .EQ. 1) GO TO 2
M1 = M - 1
DO 44 I = 1, M1
44 GN(I) = G(I) + G(M) * G(M - I)
DO 55 I = 1, M1
55 G(I) = GN(I)
2 IF (M .EQ. MORD) GO TO 3
NM = N - M
DO 66 K = 1, NM
66 F(K) = F(K + 1) + G(M) * B(K)
B(K) = B(K) + G(M) * F(K + 1)
GO TO 1

RETURN
END
COMPUTE AR COEFFICIENT BY MARPLE ALGORITHM
MODIFIED FROM MARPLE'S SUBROUTINE
IEEE ASSP-28, PP. 441-454
PARAMETERS DESCRIPTION SEE ABOVE REFERENCE

```fortran
SUBROUTINE LS(N,M,MMAX,X,A,E,TOL1,TOL2,STATUS,E0)
DIMENSION X(N),A(N)
DIMENSION C(500),D(500),R(500)
INTEGER STATUS
E0=0.
DO 10 K=1,N
  EO=E0*X(K)*X(K)
  EO=2.*E0
  Q1=1./EO
  Q2=Q1*X(1)
  G=Q1*X(1)*X(1)
  W=Q1*X(N)*X(N)
  DEN=1.*G-W
  Q4=1./DEN
  Q5=1.*G
  Q6=1.*W
  F=X(1)
  B=X(N)
  H=Q2*X(N)
  S=Q2*X(N)
  U=Q1*X(N)*X(N)
  V=Q2*X(1)
  E=E0*DEN
  Q1=1./E
  C(1)=Q1*X(1)
  D(1)=Q1*X(N)
  M=1
  SAVE1=O.
  N1=N+1
  NM=N-1
  DO 20 K=1,NM
    SAVE1=SAVE1*X(K+1)*X(K)
    R(1)=2.*SAVE1
    A(1)=-Q1*R(1)
    E=E+(1.-A(1)*A(1))
  20 CONTINUE
  IF(M.LT.MMAX) GO TO 40
  STATUS=1
  RETURN

40 E0LE=E
  M1=M+1
  F=X(M1)
  B=X(NM)
  DO 50 K=1,M
    F=F*X(M1-K)*A(K)
  50 CONTINUE
  B=B*X(NM*K)*A(K)
  Q1=1./E
```

Q2=Q1*F
Q3=Q1*B
DO 60 K=1,M
   KK=M+1-K
   K1=KK+1
   C(K1)=C(KK)+Q2*A(KK)
   D(K1)=D(KK)+Q3*A(KK)
60
Q7=S*S
Y1=F*F
Y2=V*V
Y3=B*B
Y4=U*U
G=G+Y1*Q1+Q4*(Y2*Q6*Q7*Q5+2*V*H*S)
W=W+Y3*Q1+Q4*(Y4*Q5*Q7*Q6+2*S*H*U)
H=0.
S=0.
U=0.
V=0.
MM=M+1
DO 70 K=1,MM
   K1=K
   NK=N-K+1
   H=H*X(NM*K-1)*C(K1)
70
Q5=1.-G
Q6=1.-W
DEN=Q5*Q6-H*H
IF(DEN.GT.0.) GO TO 80
STATUS=2
RETURN

C
80
Q4=1./DEN
Q1=Q1/Q4
ALPHA=1.*(Y1/Q5+Y3/Q5+2.*(H*F+B))\Q1
E=ALPHA*E
C1=Q4*(F*Q6+B*H)
C2=Q4*(B*Q5+H*F)
C3=Q4*(V*Q6+H*S)
C4=Q4*(S*Q5+V*H)
C5=Q4*(S*Q6+H*U)
C6=Q4*(U*Q5+S*H)
DO 90 K=1,M
   K1=K+1
90
A(K)=ALPHA*(A(K)+C1+C(K1)+C2*D(K1))
M2=M/2+1
DO 100 K=1,M2
   MK=M+2-K
   SAVE1=C(K)
   SAVE2=C(K1)
100
C
C
RETURN
SAVE2=D(K)
SAVE3=C(MK)
SAVE4=D(MK)
C(K)=C(K) + C3\times SAVE3 + C4\times SAVE4
D(K)=D(K) + C5\times SAVE3 + C6\times SAVE4
IF(MK\cdot EQ\cdot K) GO TO 100
C(MK)=C(MK) + C3\times SAVE1 + C4\times SAVE2
D(MK)=D(MK) + C5\times SAVE1 + C6\times SAVE2

100 CONTINUE

C

M=M+1
NM=N-M
M1=M-1
DELTA=0*
C1=X(N1-M)
C2=X(M)
DO 110 K=1,M1
KK=M1+1-K
R(KK+1)=R(KK)-X(N1-KK)\times C1-X(KK)\times C2
110 DELTA=DELTA*R(KK+1)\times A(KK)
SAVE1=0*
DO 120 K=1,NM
120 SAVE1=SAVE1 + X(K+M)\times X(K)
R(1)=2\times SAVE1
DELTA=DELTA+R(1)
Q2=-DELTA/E
A(M)=Q2
M2=M/2
DO 130 K=1,M2
MK=M-K
SAVE1=A(K)
A(K)=A(K) + Q2\times A(MK)
IF(K\cdot EQ\cdot MK) GO TO 130
A(MK)=A(MK) + Q2\times SAVE1
130 CONTINUE

Y1=Q2\times Q2
E=E\times (1.-Y1)
IF(Y1.LT.1.) GO TO 140
STATUS=3
RETURN

C

140 IF(E.GE.E0\cdot TOL1) GO TO 150
STATUS=4
RETURN

C

150 IF((EOLD-E).GE.EOLD\cdot TOL2) GO TO 30
STATUS=5
RETURN
END
COMPUTE U(L) EQ. (4.7)

INPUT R, N
R= AUTOCORRELATION FUNCTION
N= LAG NUMBER

SUBROUTINE UI(R,U,N)
DIMENSION R(32),CB(32,32),U(32)
DO 1 L=1,N
LF=1
DO 10 I=1,L
10 LF=LF*I
  DO 11 K=1,L
  KF=1
  DO 12 I=1,K
12 KF=KF*I
  LK=L-K
  IF(LK.EQ.0) LK=1
  LKF=1
  DO 13 I=1,LK
13 LKF=LKF*I
  CB(L,K)=LF/LKF/KF
11 CONTINUE
1 CONTINUE

DO 2 L=2,N
  U(L)=R(L)
  L1=L-1
  DO 20 K=1,L1
  KKK=2*K-L1
  AK=FLOAT(KKK)
  K2L=IFIX(ABS(AK))
  K2L=K2L+1
20  U(L)=U(L)+CB(L1,K)*R(K2L)
  U(L)=U(L)/2**L1
2 CONTINUE
U(1)=R(1)
RETURN
END
SOLVE TOEPLITZ SYSTEM EQ. (4.11B)

XR(I) = FIRST ROW OF CYCLIC TOEPLITZ MATRIX

NORD = ORDER OF EQ. (4.11B)

NUM = A NUMBER (POWER OF 2) AND GREATER THAN NORD

FCOL = SOLUTION OF EQ. (4.11B)

SUBROUTINE TINV(NUM, XR, FCOL, NORD)

DIMENSION XI(32), Q12(32), XR(32)

DIMENSION PI(32,32), Q22(32,32), Q21(32), X(64)

DIMENSION FCOL(32)

N=NUM

NSTOP=NORD

DO 1 I=1,N

XI(I)=0.0

1 CALL FFT842(0, NUM, XR, XI)

DO 2 I=1,N

XX=XR(I)*XR(I) + XI(I)*XI(I)

IF(XX .NE. 0.) GO TO 10

WRITE(6,1110)

FORMAT(1X,'ERROR')

GO TO 1111

10 XR(I)=XR(I)/XX

XI(I)=-XI(I)/XX

CONTINUE

2 CALL FFT842(1, NUM, XR, XI)

56 FORMAT(/)

DO 5 I=1,N

5 X(I)=XR(I)

DO 6 I=1,N

6 X(N+I)=XR(I)

DO 4 I=1,N

DO 41 J=1,N

41 PI(I,J)=X(N-I+1+J)

4 CONTINUE

98 N1=N-1

DO 7 I=1,N

71 Q22(I,J)=PI(I,J)

7 CONTINUE

Q11=1.0/PI(1,1)

DO 8 I=1,N1

81 Q12(I)=PI(1,N+1-I)

8 DO 81 I=1,N1

81 Q21(I)=PI(N+1-I,1)

90 PI(1,J)=Q22(1,1)-Q12(1)*Q21(J)*Q11

91 PI(I,1)=Q22(I,1)-Q12(I)*Q21(1)*Q11

IF(N1 .EQ. NSTOP) GO TO 999

N=N-1

GO TO 98

APPENDIX 14
C

999  DO 222 I=1,N1
      FCOL(I)=PI(I,1)/PI(N1,1)
   222  CONTINUE

1111  RETURN

   END