BRAFF, MARTIN

SYNCHRONIZATION, BANDWIDTH OCCUPANCY AND EFFECTS OF TRANSMISSION ERRORS IN DIGITAL COMMUNICATION SYSTEMS

City University of New York PH.D. 1981

SYNCHRONIZATION, BANDWIDTH OCCUPANCY AND EFFECTS OF TRANSMISSION ERRORS IN DIGITAL COMMUNICATION SYSTEMS

by

MARTIN BRAFF

A dissertation submitted to the Graduate Faculty in Engineering in partial fulfillment of the requirements for the degree of Doctor of Philosophy, The City University of New York.

1981

This manuscript has been read and accepted for the Graduate Faculty in Engineering in satisfaction of the dissertation requirement for the degree of Doctor of Philosophy.

9/2/81
date

Chairman of Examining Committee

9/10/81
date

Executive Officer

Prof. D.L. Schilling (Mentor)

Prof. H. Taub

Prof. S.J. Oh

Prof. N. Scheinberg

Prof. T. Saadawi

Dr. J. Garodnick

Supervisory Committee

The City University of New York

Abstract

SYNCHRONIZATION, BANDWIDTH OCCUPANCY AND EFFECTS OF TRANSMISSION ERRORS IN DIGITAL COMMUNICATION SYSTEMS

by

Martin Braff

Adviser: Professor Donald L. Schilling

This research is concerned with the following three areas: a) Synchronization of digital networks, b) Correlation and spectral density of a Convolutional Encoder, and c) Channel errors in Delta Modulation.

a) The increased use of digital transmission in telephone networks motivates attempts to perform the switching function in digital form also. Such a digital network, however, requires carefully synchronized clocks to perform correctly. In this dissertation we present a new technique for synchronizing such clocks called "Peak Synchronization". This technique will be studied and compared to others that have been previously investigated.

b) When a digital signal is encoded for channel error protection, the resulting encoded signal may have statistics differing greatly from the original signal. The correlation of the encoded signal is especially important as it allows

for the determination of the signal's power spectrum. The correlation also provides information about the density of transitions, which is important in bit synchronizer design. We will study the correlation of signals encoded by a convolutional encoder, with both purely random and first order Markov inputs.

c) Delta Modulators have been extensively studied to determine their quantizing and slope overload distortion. However, the distortion introduced by channel errors has received little attention. Here we will determine the effect of channel errors on the performance of a Delta Modulator. Both independent random errors and errors occuring in bursts will be considered.

ACKNOWLEDGEMENT

I would like to express my appreciation to Professor Donald L. Schilling, for his support, encouragement and valuable guidance during the course of this research.

I am greatly indebted to Dr. J. Kella, for originally suggesting the idea of 'Peak Synchronization', and for his encouragement to pursue this area of research. I am also indebted to Dr. N. Kuroyanagi for helpful discussions on the subject of Network Synchronization.

I would like to thank M. Simon for suggesting the problem on convolutional codes, and for many valuable discussions.

Last but not least I would like to thank my wife, Tzipora, for her support and concern in this endeavor.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

SYNCHRONIZATION OF DIGITAL NETWORKS

1.1 INTRODUCTION

Digital transmission in the form of Pulse Code Modulation (PCM) has been used in telephone communications since the early 1960's. This early introduction of PCM into telephony was due both to the more economical digital hardware that became available, and the inherent advantages of digital communications over the conventional analog techniques [1]. Specifically conversion to digital offered the following benifits:

a) The quality of a digital signal does not significantly deteriorate with transmission distance. This is because digital repeaters can remove noise and restore the signal provided that the noise level is below some threshold. Thus tandem digital lines do not accumulate noise, and a uniform voice quality can be available over the whole network. The digital "bit probability of error" decreases very rapidly with increasing signal power, and can easily be made negligible.

b) A single high speed piece of digital hardware can be effectively shared among many users, thus resulting in substantial savings. For example a high speed A/D converter followed by a digital multiplexer can Time Division

Multiplex many incoming analog signals. This is in contrast with the FDM case where a separate modulator is needed for each audio signal.

PCM was first introduced into telephony for point to point communication links. Here as illustrated in Fig. 1.1 a group of voice channels are each sampled at 8000 samples/sec., A/D converted and transmitted in TDM form over a digital link. Since all the voice channels must be sampled and transmitted 8000 times per second, the time available for each sample is 125/n usec., where n is the number of incoming channels. A single sample from a channel in PCM format is said to occupy a time slot. The time slots containing samples from all the channels collectively form a frame. Since for correct demultiplexing it is imperative to know where a frame begins, framing information is added to the voice samples.

In this usage of PCM, synchronization is achieved by the receiver clock, phase locking to the incoming digital signal. In addition frame synchronization is acquired by use of the built-in framing bits. Point to point digital transmission thus allows for the use of independent transmitting clocks with inaccuracies of up to 10^{-5} [2].

When PCM is used only for point to point transmission as described above, all switching must be done with signals in analog form. Thus as shown in Fig. 1.2 a signal may undergo several A/D and D/A conversions as it passes through

the network. It is apparent that the full advantages of digital transmission would be realized only if both the transmission and switching functions were performed digitally. This would eliminate the expense of repeated A/D and D/A conversion, the quantizing noise which accumulates each time an A/D conversion is performed would be reduced, and the network would be able to handle data communication much more effectively. However, instituting digital as opposed to analog switching greatly complicates the synchronization problem.

1.2 DIGITAL SWITCHING

A digital switch in general contains several incoming and outgoing TDM lines. Switching is accomplished by transferring digital signals appearing at a given time slot on an incoming line, to a preassigned time slot on an outgoing line. In order for the digital switch to perform correctly an incoming signal must be available at the exact time when it is needed to fill the appropriate outgoing time slot. Since the arrival rate of data on the incoming TDM lines will change with oscillator drift and delay time variations, it is necessary to provide an elastic buffer for each received bit stream to store the data until it is needed and to absorb incoming rate fluctuations. Data is read out of each buffer under the control of a local clock at the digital switch. At the same time data is arriving at

the buffers with a rate determined by the remote transmitting clock. Unless these clocks are properly controlled to maintain precisely the same average frequency, buffer overflow or underflow is sure to occur. This would result in the deletion or repetition of a frame of data, an occurrence which is called a "slip".

The subject of Network Synchronization is concerned with the development of techniques to control the clock frequencies in order to prevent slips. A great deal of research effort has been devoted to the study of many proposed techniques (see references). In this research we investigate a new method of synchronizing remote clocks. In the method to be studied, called Peak Synchronization, each switch selects the incoming line that contains the largest filled buffer, and phase locks its clock to that incoming line. In the discussion that follows we will present a mathematical model for a digital network, summarize some of the previous work done in Network Synchronization, and motivate the use of the Peak Synchronization technique.

1.3 MATHEMATICAL MODEL

A digital network can be represented by a directed graph. The nodes of the graph representing the switching centers, and the directed edges representing the transmission channels. Let N be the number of nodes and M the number of edges. If the network is connected, that is

there is a directed path from any node to every other node, then we must have:

$$N < M < N(N-1)$$

The lower limit representing the case of all nodes being connected in a single loop, and the upper limit representing an edge between every pair of nodes.

The switching activities at each node are governed by a local clock. Let $f_i(t)$ denote the frequency of the local clock at node i. Signals arriving at node i from node j, arrive with a frequency $f_j(t)$ determined by the transmitting clock, and are processed and retransmitted at the frequency of the receiving clock $f_i(t)$. We denote by $b_{ij}(t)$ the contents (in bits) of the buffer at node i from node j. Then $b_{ij}(t)$ is given by:

$$b_{ij}(t) = \int_0^t [\, f_j(s - \tau_{ij}) - f_i(s) \,]ds + b_{ij}(0) \qquad (1.1)$$

where τ_{ij} represent the transmission delay to node i from node j, and we have assumed that these delays are constant. In the more general case of time varying transmission delays, the received frequency would be modified by a doppler shift. In this case we would have:

$$b_{ij}(t) = \int_0^t [(1-\tau'_{ij}(s))f_j(s-\tau_{ij}(s)) - f_i(s)]ds + b_{ij}(0) \qquad (1.2)$$

Fig. 1.3 shows a node with several input buffers.

Clearly if a frequency difference between any two nodes persists a buffer will eventually overflow or underflow. In this situation a frame of information will be lost or repeated resulting in a "slip". It is the primary function of the synchronization control to prevent such slips. In addition, it would be advantageous to ensure that the buffer contents can be kept small. The obvious reason is that buffer memory is costly. Also important is that large buffers will increase the transmission delay of a signal, which can be critical in speech transmission. Note that buffer readings can really only take on integer values. However if the buffer range is large enough it is possible to treat the buffer reading as the continuous variable given by (1.1).

The network configuration will be described by a connection matrix $\{a_{ij}\}$, with a_{ij} equal to 1 if there is a transmission path to node i from node j, and zero otherwise. In addition, we will denote by S_i the set of nodes that transmit to node i.

1.4 SYNCHRONIZATION TECHNIQUES

The problem of devising a control scheme that will keep all local clocks at the same long term average frequency has received a great deal of attention. Basically the various approaches that have been studied fall into the following three categories: Independent Clocks, Mutual

Synchronization and Master Slave Synchronization.

Conceptually the simplest method is for each node to have a sufficiently accurate independent clock, so as to reduce the occurrence of slips to a tolerable level. This approach has become known as Plesiochronous[19]. The feasability of this approach is due to the extremely accurate and stable atomic clocks that are available. For example, if a system is operating at a frame rate of 8000 frames/sec. and the slip rate objective is 1 slip per 5 hours, then the clock accuracy requirement is:

$$(8000 \text{ frames/sec.})(18{,}000 \text{ sec./5 h})(\Delta f/f) = 1 \text{ slip}$$

therefore

$$\Delta f/f = 7 \times 10^{-9}.$$

If as is expected this objective is to be maintained over an end to end connection consisting of several switches [24], the required clock accuracy would become approximately 10^{-9}. While atomic clocks can easily meet this objective, they present problems of high cost, reliability, power consumption and frequent maintenance [17]. Nevertheless this seems to be the method of choice for international network interconnections [25].

In the Master Slave technique one node, presumably containing a highly reliable and stable oscillator, is designated as the master, and all other nodes phase lock

their oscillators to a signal derived from the master clock. Thus as is illustrated in Fig. 1.4, certain communication links are specified as being used for synchronization purposes. This subnetwork is a directed tree with every node except the master having one incoming link.

While attractive for its simplicity, the master slave technique suffers from being extremely vulnerable to a failure in either the master clock or a link being used for synchronization. Two schemes have been proposed to help mitigate these weaknesses. First in case of a failure in a synchronization path, an alternate path could be chosen by the receiving node on a preassigned basis. This method has been called Preselected Alternate Master Slave, and has the advantage of enhancing the reliability of Master Slave synchronization without requiring any internodal communication. It has been chosen for the switched digital network in the U.S.[16]. A more elaborate scheme proposed by Darwin and Prim[26] allows for the network to reorganize itself in an optimum fashion in the event of a failure in a line or clock. Though more reliable than the Preselected Alternative method, it requires that a channel be available for nodes to exchange synchronization information. It will be seen that Peak Synchronization has many of the characteristics of the self organizing Master-Slave approach.

Mutual Synchronization

Mutual Synchronization is a method that was proposed to completely eliminate the dependence on a master clock. Instead each node participates in the determination of the final system frequency. It has the advantage of maintaining network synchronization in the event of both oscillator and link failure, and does not require any internodal communication of synchronizing information, as is required by the self organizing Master-Slave approach.

The control scheme for Mutual Synchronization is as follows: Each node forms an average of the phase differences of all incoming clocks, relative to the phase of its own clock. This error signal is then filtered and used to modify the local clock's free running frequency. The control can be expressed mathematically as follows:

$$P_i'(t) = f_{io} + h_i(t) * \sum_{j=1}^{N} \hat{a}_{ij}[P_j(t-\tau_{ij}) - P_i(t)] \qquad (1.3)$$

Here $\hat{a}_{ij}$ are averaging coefficients that satisfy $\sum_j \hat{a}_{ij} = 1$, f_{io} is the free running frequency at node i, $h_i(t)$ is the impulse response of the filter at node i and * denotes convolution. The phase at node i is defined by,

$$P_i(t) = \int_0^t f_i(t)\, dt + P_i(0) \qquad (1.4)$$

It is assumed here that all clocks are free running for $t<0$, and that at time $t=0$ the controls are applied. From the definition of the buffer contents given in (1.1) we see that

$$b_{ij}(t) = P_j(t-\tau_{ij}) - P_i(t) + f_{jo}\tau_{ij} + P_i(0) - P_j(0) + b_{ij}(0)$$

Thus the buffer contents and phase differences are related by an additive constant. Therefore a formulation of Mutual Synchronization very similar to (1.3) would be

$$P_i'(t) = f_{io} + h_i(t) * \sum_{j=1}^{N} \hat{a}_{ij} b_{ij}(t) \qquad (1.5)$$

The filters $h_i(t)$ are assumed to possess the properties of causality, stability in the sense of a bounded input producing a bounded output, and positive D.C. gain, that is:

$$H_i(0) \equiv L_i > 0.$$

where $H_i(s)$ is the Laplace Transform of $h_i(t)$. A diagram of the control scheme implied by (1.5) is shown in Fig. 1.5.

The performance of systems governed by controls similiar to (1.3) and (1.5) have been extensively studied [3-12]. Gersho and Karafin[3] found a sufficient condition for such a system to be stable, that is for all clock frequencies to approach some common value. Specifically they show that stability is ensured provided that:

$$\left| \frac{H_i(jw)}{jw + H_i(jw)} \right| < 1, \quad w \neq 0. \qquad (1.6)$$

It is easy to see that for a flat filter response (i.e. no filtering), $H_i(jw) = L_i$, and (1.6) is always satisfied. If $H_i(s)$ is a simple one pole low pass filter defined by:

$$H_i(s) = L_i/(1 + G_i s)$$

then substituting into (1.6) yields,

$$\frac{L_i^2}{L_i^2 + (1-2L_iG_i+G_i^2w^2)w^2} < 1 \qquad (1.7)$$

which is satisfied provided $L_iG_i < 1/2$. We shall see that it is appropriate to choose L_i to be quite small, so that (1.7) is not difficult to satisfy. It has been shown[4] that the common frequency attained by all the oscillators is given by:

$$f_c = \frac{\sum_{i=1}^{N} b_i f_{io}}{\sum_{i=1}^{N} b_i(1 + L_i \tau_i)} \qquad (1.8)$$

Here b_i is the cofactor of any element in the ith row of the matrix $L(I - A)$. L is a diagonal matrix with elements L_i, τ_i is the average delay to node i, and A is the averaging matrix.

In order for the system frequency not to depend on the transmission delays, it is necessary to choose the L_i's such that

$$L_i \tau_i << 1 \qquad (1.9)$$

in which case we have from (1.8)

$$f_c \simeq \frac{\sum_i b_i f_{io}}{\sum_i b_i} \qquad (1.10)$$

In this case the system frequency becomes a simple weighted average of the free running clock frequencies.

Though the performance of the Mutual Synchronization scheme outlined above is satisfactory from the point of view of achieving a common system frequency, this is not sufficient to ensure a properly functioning digital switching network. In order to prevent the loss of information due to slips, it is necessary that the control scheme prevent the buffers from overflowing. While it is known that the buffer sizes in Mutual Synchronization will remain finite, the problem of finding a bound to the buffer size that can be easily computed has not been solved. There is no simple expression available, comparable to the one for system frequency, giving the final or asymptotic buffer sizes.

The problem of determining buffer sizes in Mutual Synchronization was considered by Moreland[15], who pointed out that unlike system frequency, buffer sizes were critically dependent on network configuration. Moreland conjectured that the "dumbell" configuration (Fig. 1.6), which consists of two equal size groups of nodes with full interconnection in each group and only a single intergroup link, would be the worst case for buffer size. He then

shows that the largest buffer in a dumbell network would have a value proportional to N^2, where N is the number of nodes. This dependence of buffer size on network configuration and network size is an important drawback of Mutual Synchronization. It means that extensive modification may have to be made to an existing network in order to allow for expansion.

In order to improve the performance of Mutual Synchronization with regard to buffer size, several authors[13,14] have suggested substituting non-linear control laws for the linear one given by (1.3). These control laws share the common property of giving more weight to the larger buffers. Karnaugh[14] approaches the problem by formulating a control law that will minimize a convex penalty function $R(\underline{B})$, where $\underline{B}$ is a vector with components b_{ij}, the buffer contents. In order to minimize $R(\underline{B})$ he chooses a control which has components in the direction of the negative gradient of $R(\underline{B})$. Specifically his control has the form:

$$P_i'(t) = f_{io} - \sum_j \hat{a}_{ij} \frac{\partial}{\partial P_j} R(\underline{B}) \qquad (1.11)$$

where

$$b_{ij} = [\, P_j(t) - P_i(t)] + b_{ij}(0)$$

A possible form for $R(\underline{B})$ which is suggested is:

$$R(\underline{B}) = \sum_{ij} (b_{ij})^{2n}$$

Unfortunately, stability for the above scheme is only demonstrated for the case of zero delays and no bounds are found for the buffer contents.

More pertinent to this research is the work done by Sandberg[13]. Sandberg considers a system of controls of the form

$$f_i(t) = f_{io} + \psi_i[\sum_j \theta_{ij}(b_{ij}(t)] \quad i= 1,2,\ldots,N \qquad (1.12a)$$

where as before

$$b_{ij}(t) = \int_0^t [\, f_j(s-\tau_{ij}) - f_i(s)]ds + b_{ij}(0) \qquad (1.12b)$$

The functions $\theta_{ij}(x)$ are monotonic functions chosen to reduce the likelihood of buffer overflow. For this purpose they should have moderate slope near the origin, and large slope for buffer values near overflow. Clearly in this way they emphasize the larger buffers. The functions $\psi_i(x)$ on the other hand are chosen to reduce the tuning range required of the oscillators. These functions therefore have moderate slope near the origin, and very small slope far from the origin. Sandberg shows that the system described by (1.12) is stable provided the following conditions are satisfied:

a) $0 < \psi_i'(x) < \infty$, (1.13a)

b) either $\theta_{ij}(x) \equiv 0$, or $0 < \theta_{ij}'(x) < \infty$ (1.13b)

c) The directed graph defined by $\theta_{ij}'(0)$ is connected (1.13c)

$\theta_{ij}'(0)$ defines a directed graph in the following sense. If $\theta_{ij}'(0) \neq 0$, then there is a directed edge to node i from node j. It is also shown that bounds on the buffer contents exist, but again no simple method for finding these bounds is presented.

Peak Synchronization

In this dissertation we will investigate the behavior of systems governed by the following control rule:

$$f_i(t) = f_{io} + L_i \max_{j \in S_i} \{ b_{ij}(t) \} \quad (1.14)$$

Here only the largest buffer appearing at node i is considered in determining the amount to offset the frequency of that node. This scheme has the obvious advantage that the nonlinear function required is easily implemented. We shall study the stability of the above system, its dynamic response, and determine a bound on the size of the buffers. The analytical work will be verified by computer simulation. We will show that in Peak Synchronization, we can find a very easily computable bound to the buffer size, that is totally independent of network configuration or size.

The concept of Peak Synchronization is clearly a natural extension of the nonlinear controls suggested by Sandberg and Karnaugh, that are aimed at emphasizing the larger buffers. An important distinction though is that in Peak Synchronization only the most positive buffer is considered rather than the buffer largest in magnitude. This approach was originally proposed by J. Kella[27] because of the observation that a large negative buffer representing a potential underflow is less damaging than a positive buffer with overflow likely. Underflow can be handled by inserting a special bit sequence that would be recognized and removed by the receiver. Overflow on the other hand represents an irreversible loss of information.

It is now apparent however that in most common situations preventing buffer overflow also prevents underflow. If we make the very reasonable assumption that all communication links are bilateral (the matrix $\{a_{ij}\}$ is symmetric), then for every buffer b_{ij} there will be a buffer b_{ji}. We shall show that these two will be approximately complementary ($b_{ij} \simeq -b_{ji}$) and therefore an underflow would imply an overflow and vice-versa. In this case ensuring that there is no overflow also ensures no underflows.

It is of interest to note that we can approximate Peak Synchronization, with arbitrarily small error, by controls that have the same form as those studied by Sandberg. To see this let us choose

$$\theta_{ij}(x) = a_{ij}M^{x} \quad , \quad \psi_i(x) = L_i \log_M x \tag{1.15}$$

where M is a large positive number and $\{a_{ij}\}$ is the connection matrix of the network. In this case (1.12a) becomes:

$$f_i(t) = f_{io} + L_i \log_M \left[\sum_j a_{ij} M^{b_{ij}(t)} \right] \tag{1.16}$$

To see that this approaches (1.14), let $b_{imax} = \max_{j \epsilon S_i} b_{ij}(t)$, then

$$\log_M M^{b_{imax}} < \log_M \sum_j a_{ij} M^{b_{ij}(t)} < \log_M N M^{b_{imax}}$$

$$b_{imax} < \log_M \sum_j a_{ij} M^{b_{ij}(t)} < \log_M N + b_{imax}$$

since

$$\lim_{M \to \infty} \log_M N = 0$$

we have

$$\lim_{M \to \infty} \log_M \sum_j a_{ij} M^{b_{ij}(t)} = b_{imax}$$

So by making M large enough we can make (1.16) arbitrarily close to (1.14). For any fixed M the conditions indicated by (1.13) are satisfied and Sandberg's proof of stability applies. However, if M goes to infinity (which is required to get Peak Synch), these conditions are no longer satisfied and a new stability proof is needed.

Peak Synchronization can also be regarded as a form of

self organizing Master-Slave Synchronization. Here as in the Master-Slave technique a subnetwork of the original network is used for synchronization purposes. In Master-Slave Synchronization we saw that the subnetwork was a directed spanning tree. A spanning tree with N nodes contains N-1 edges. In Peak Synchronization each node chooses one incoming link, and phase locks its oscillator to that signal. Therefore the synchronization subnetwork consists of N links. We shall see that this subnetwork is a directed tree, with one directed loop. The nodes which are part of the directed loop mutually synchronize, and all other nodes are slaved to the frequency established by the synchronization loop. In case of a failure in a transmission link (if that link happened to be used for synchronization), the network would automatically reconfigure itself, maintaining synchronization.

Hybrid Synchronization Techniques

It may be advantageous for a network to simultaneously employ several synchronization techniques [23]. An example of this is a scheme being adopted by the British Post Office [20]. Here the network nodes are divided into four hierarchical levels. Nodes within a given level mutually synchronize, whereas nodes at a higher level act as masters to nodes at a lower level. Fig. 1.7 shows the network configuration. In normal operation the level one clock is

master for the entire network. Should the level one clock fail, the nodes in level two would collectively take over control of all lower level clocks. This structure allows each of the mutually synchronizing parts to be kept small, thus averting the large buffers that can be associated with a large mutually synchronized network.

Parameters

In general we may express the frequency of node i as:

$$f_i(t) = F + E_i + g_i(t)$$

where F is the nominal or average frequency of all nodes, E_i is the frequency offset of node i due to oscillator inaccuracy, and $g_i(t)$ is the applied control signal. $g_i(t)$ has the following forms:

$$g_i(t) = L_i \sum_j a_{ij} b_{ij}(t) \quad ; \text{ Mutual Synch no filters.}$$

$$g_i(t) = L_i \max_{j \in S_i} b_{ij}(t) \quad ; \text{ Peak Synch.}$$

We wish to determine an appropriate value for the parameters L_i. Since the smallest amount the buffer readings can change is unity, the quantity L_i represents the smallest frequency shift applied by the controls. Clearly this amount should be much less than the natural inaccuracy of the clocks. If d represents the maximum fractional clock

inaccuracy, then $E_i < d \cdot F$, for all i. A reasonable value for L_i would be given by:

$$L_i = (d \cdot F)/10$$

If F is 2×10^6 Hz, then the values of L_i for several values of d are:

	d	L_i	$1/L_i$ sec.
crystal	10-7	2x10-2	50
oscillators	10-8	2x10-3	500 (8.3 minutes)
atomic clock	10-11	2x10-6	500,000 (138 hours)

The quantities $1/L_i$ will be seen to be proportional to the time constants of the system. We see therefore that the very stable clocks require slowly applied control signals.

Considering now the requirement given by (1.9) which is:

$$\tau_i L_i << 1$$

and noting that the transcontinental transmission time is approximately .02 seconds, then for $d = 10^{-7}$, $\tau_i L_i = 4 \times 10^{-4}$ which easily satisfies (1.9).

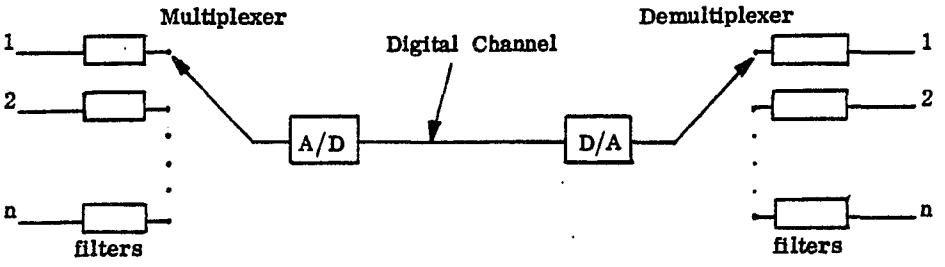

Figure 1.1 Point to point PCM transmission.

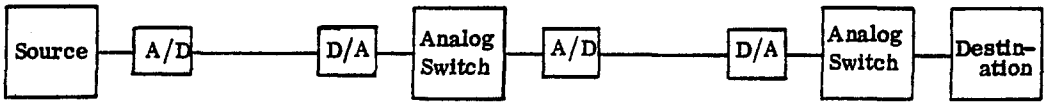

Figure 1.2 Analog switching and digital transmission. Repeated A/D and D/A conversions are necessary resulting in accumulated quantizing noise.

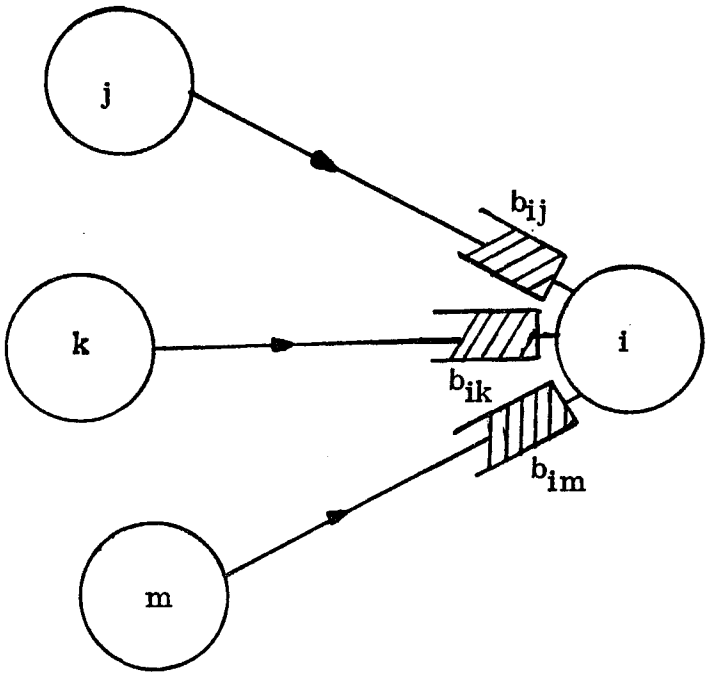

Figure 1.3 Elastic buffers at node i. Data is stored until needed for transmission.

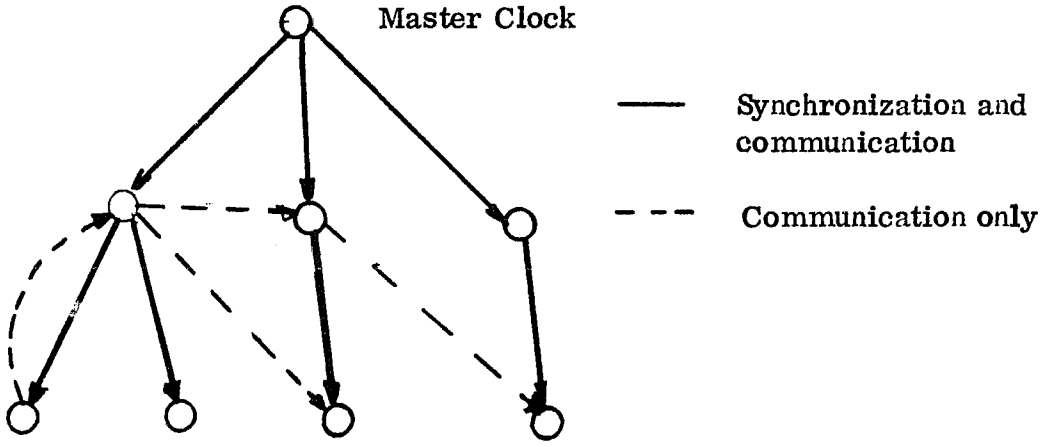

Figure 1.4 Master Slave synchronization. The synchronization links form a directed tree.

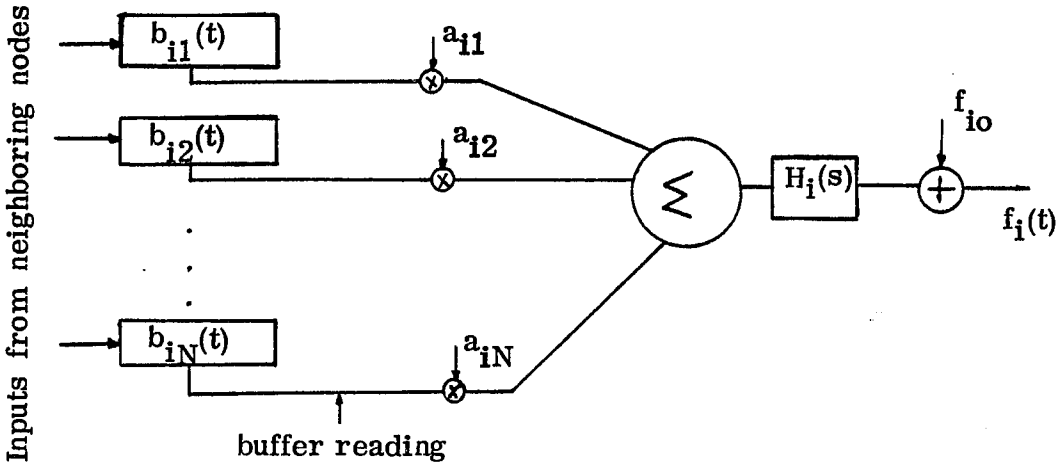

Figure 1.5 Block diagram of Mutual Synchronization

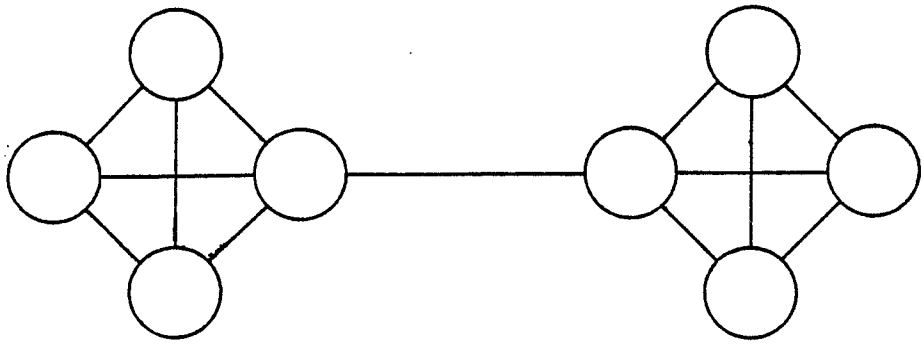

Figure 1.6 Dumbell Network

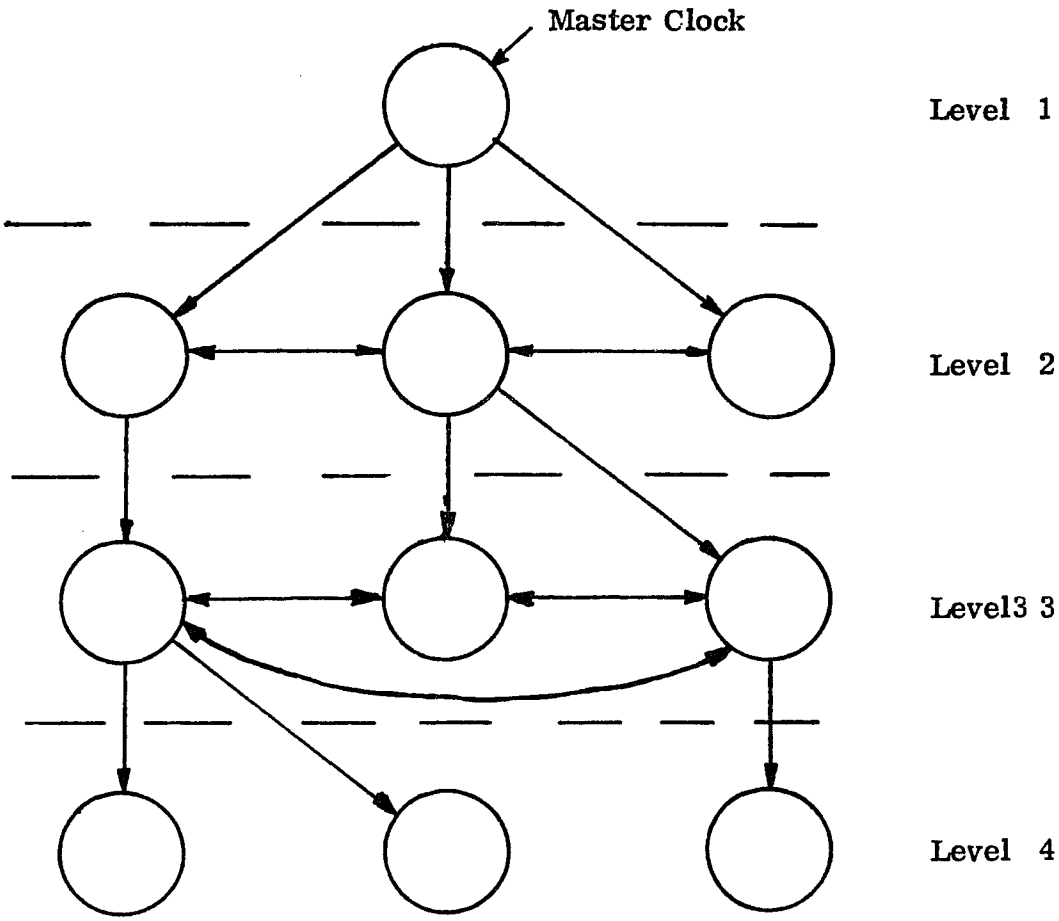

Figure 1.7 Hybrid Synchronization. Both Mutual and Master Slave Synchronization techniques are used.

Chapter 2

STABILITY OF PEAK SYNCHRONIZATION

2.1 INTRODUCTION

A synchronization technique is generally considered stable if it insures that all clock frequencies will eventually approach some common value. This common frequency must, however, be suitably restricted for proper network operation to result. First, the common frequency must lie within the tuning range of the oscillators and second, since the network must interface with external equipment, the common frequency must lie within a range suitable for that equipment. Additionally, in order to prevent the loss of information due to slips, a synchronization technique should limit buffer size to some predeterminable amount.

In this chapter we will study the performance of Peak Synchronization with regard to the above mentioned criteria. Specifically we will show that the following properties are satisfied by Peak Synchronization:

a) No oscillator frequency ever goes above the highest uncontrolled frequency, or below the lowest uncontrolled frequency.

b) The oscillators reach a common frequency which must clearly lie in the above mentioned range.

c) The buffer sizes are bounded by an amount that is

independent of network size or configuration.

2.2 MATHEMATICAL FORMULATION

The control rule for Peak Synchronization was stated in Chapter 1 as:

$$f_i(t) = f_{i0} + L_i \max_{j \in S_i} \{b_{ij}(t)\}, \tag{2.1}$$

where, as before, S_i is the set of nodes that sends to node i, $b_{ij}(t)$ is the contents of the buffer at node i from node j, and f_{i0} is the uncontrolled frequency at node i.

We begin by rewriting (2.1) in a more suitable form, namely:

$$f_i(t) = f_{i0} + L_i \sum_{j=1}^{N} A_{ij}(t)\ b_{ij}(t). \tag{2.2}$$

The $A_{ij}(t)$s now perform the operation of choosing the largest buffer at node i. Therefore the functions $A_{ij}(t)$ can be defined as follows:

$$\begin{aligned} A_{ij}(t) &= 1 \quad \text{if } b_{ij}(t) \text{ is the largest buffer at node i} \\ A_{ij}(t) &= 0 \quad \text{otherwise} \end{aligned} \tag{2.3}$$

If several buffers at node i are equal to the largest value, we will assume that one is arbitrarily chosen as the maximum. Therefore at any instant exactly one of the $A_{ij}(t)$, $1 \leq j \leq N$, is equal to one, all others being zero. Differentiating (2.2) we have:

$$\dot{f}_i(t) = L_i[\sum_j A_{ij}(t)\ \dot{b}_{ij}(t) + \sum_j \dot{A}_{ij}(t)\ b_{ij}(t)] \qquad (2.4)$$

We wish to show that the second summation is equal to zero. The $A_{ij}(t)$s are constants equal to 1 or 0 except at those instants when a new buffer becomes the largest. Only those instants therefore need to be considered. Suppose that T is such an instant, and that $A_{i1}(t) = 1$ for $t < T$ and $A_{i2}(t) = 1$ for $t > T$. That is, $b_{i2}(t)$ becomes the largest buffer at time T. This situation is shown in Figures 2.1a and 2.1b.

At time T we have for the second summation in (2.4)

$$\begin{aligned}\sum_j \dot{A}_{ij}(T)\ b_{ij}(T) &= \dot{A}_{i1}(T)\ b_{i1}(T) + \dot{A}_{i2}(T)\ b_{i2}(T) \\ &= -\delta(t-T)\ b_{i1}(T) + \delta(t+T)\ b_{i2}(T) \\ &= 0\end{aligned}$$

since $b_{i1}(T) = b_{i2}(T)$. Thus (2.4) reduces to:

$$\dot{f}_i(t) = L_i \sum_{j=1}^{N} A_{ij}(t)\ \dot{b}_{ij}(t) \qquad (2.5)$$

It is important to note that since $b_{i1}(T) = b_{i2}(T)$ the oscillator frequencies will not suffer any jump discontinuities when a new buffer is chosen as maximum. This situation is, however, somewhat of an idealization, since in a practical system the transition would not occur exactly at time T, but rather at some $T + \Delta T$. The actual and idealized frequency variation is shown in Fig. 2.2.

Using the definition of $b_{ij}(t)$ given in (1.1), we can

write (2.5) as:

$$\dot{f}_i(t) = L_i \sum_j A_{ij}(t)[f_j(t-\tau_{ij}) - f_i(t)] \qquad (2.6)$$

and using the fact that $\sum_j A_{ij}(t) = 1$, (2.6) becomes

$$\dot{f}_i(t) + L_i f_i(t) = L_i \sum_{j=1}^{N} A_{ij}(t) f_j(t-\tau_{ij}) \qquad (2.7)$$

This is the basic equation that will be used throughout this chapter to characterize Peak Synchronization.

2.3 MONOTONIC BEHAVIOR OF THE OSCILLATORS

The first part of our stability proof consists of showing that the oscillator frequencies have an important monotonic behavior. This section follows closely a similar argument used by Sandberg [13].

Let us denote by $\bar{f}(t)$ and $\underline{f}(t)$ the largest and smallest oscillator frequencies at time t, respectively. Thus

$$\bar{f}(t) = \max_i f_i(t)$$

$$\underline{f}(t) = \min_i f_i(t).$$

In addition, let

$$\bar{\tau} = \max_{ij} \tau_{ij}$$

and let $\hat{f}(t)$ and $\underline{\hat{f}}(t)$ denote the largest and smallest frequency in the interval $[t-\bar{\tau},t]$. That is:

$$\hat{\bar{f}}(t) = \max_{s\in[t-\bar{\tau},t]} \bar{f}(s)$$

$$\check{\underline{f}}(t) = \min_{s\in[t-\bar{\tau},t]} \underline{f}(s) \quad .$$

The monotonic behavior of the oscillators mentioned above can now be stated as follows:

Theorem 2.1

The function $\hat{\bar{f}}(t)$ as defined above is monotonically non-increasing, likewise $\check{\underline{f}}(t)$ is monotonically non-decreasing.

Proof

Let T be any fixed point in time. We first show that for any time $t_0 > T$ the following hold:

a) $\max_{[T,t_0]} \bar{f}(t) \leq \hat{\bar{f}}(T)$

b) $\min_{[T,t_0]} \underline{f}(t) \geq \check{\underline{f}}(T)$

That is for all time greater than T, all frequencies remain less than or equal to $\hat{\bar{f}}(T)$, and greater than or equal to $\check{\underline{f}}(T)$.

We first note that there must be a time $t_1 \in [T,t_0]$, and a node i such that

$$f_i(t_1) = \max_{[T,t_0]} \bar{f}(t)$$

The relationship between the time instances and the maximum frequencies with their respective intervals is shown in Fig. 2.3.

Integrating (2.7) we have for $f_i(t_1)$:

$$f_i(t_1) = f_i(T)e^{-L_i(t_1-T)} + L_i \int_T^{t_1} \sum_{j=1}^{N} A_{ij}(\tau) f_j(\tau-\tau_{ij}) e^{-L_i(t_1-\tau)} d\tau \tag{2.8}$$

Let

$$f_{max} = \max_{[T-\bar{\tau},t_0]} \bar{f}(t)$$

Then we have from (2.8)

$$f_i(t_1) \leq f_i(T)e^{L_i(t_1-T)} + L_i \cdot f_{max} \int_T^{t_1} \sum_{j=1}^{N} A_{ij}(\tau) e^{-L_i(t_1-\tau)} d\tau. \tag{2.9}$$

Making use of the fact that $\sum_j A_{ij}(\tau) = 1$, (2.9) becomes

$$f_i(t_1) \leq f_i(T)e^{-L_i(t_1-T)} + f_{max}\left[1 - e^{-L_i(t_1-T)}\right]. \tag{2.10}$$

We will prove this theorem by contradiction. Assuming therefore that part (a) is incorrect, we have

$$\max_{[T,t_0]} \bar{f}(t) = f_i(t_1) > \hat{f}(t)$$

which implies that $f_{max} = f_i(t_1)$. As a result (2.10) yields

$$f_i(t_1) \leq f_i(T).$$

From which it follows immediately that

$$f_i(t) \leq \hat{f}(T)$$

and this is a contradiction, so that (a) is proved.

The proof of part (b) is entirely analogous. In this

instance we let

$$f_i(t_1) = \min_{[T,t_0]} \underline{f}(t)$$

and

$$f_{min} = \min_{[T-\bar{\tau},t_0]} \underline{f}(t).$$

From (2.8) we can then write:

$$f_i(t_1) \geq f_i(T)e^{-L_i(t_1-T)} + f_{min}\left[1-e^{-L_i(t_1-T)}\right] \qquad (2.11)$$

Again by contradiction we assume that

$$f_i(t_1) < \underset{\sim}{f}(T).$$

Then $f_{min} = f_i(t_1)$, and (2.11) yields

$$f_i(t_1) \geq f_i(T)$$

which is a contradiction and (b) is proved.

The statements of the theorem now follow directly. For if

$$\max_{[T,t_0]} \bar{f}(t) \leq \hat{f}(T),$$

then clearly

$$\hat{f}(t_0) = \max_{[t_0-\bar{\tau},t_0]} \bar{f}(t) \leq \hat{f}(T)$$

and similarly for the second part of this Theorem.

Since the functions $\hat{f}(t)$ and $\underset{\vee}{f}(t)$ are monotonic and bounded it follows immediately that these functions must monotonically approach limiting values at $t\to\infty$. Therefore there must exist quantities f_a and f_b such that

$$\lim_{t\to\infty} \hat{f}(t) = f_a$$

$$\lim_{t\to\infty} \underset{\vee}{f}(t) = f_b$$

The remaining part of our treatment of stability will be concerned with showing that under certain restrictions these two limits must be identical. Thus all oscillators must asymptotically approach a common frequency.

In order to prove the equality of f_a and f_b, we first show that for an arbitrarily large number T, we can find a time interval of length T such that there will be an oscillator remaining arbitrarily close to f_a, and one that will be arbitrarily close to f_b for the entire time interval. Specifically we shall prove:

Theorem 2.2

Let T be a positive number no matter how large, then for every $\varepsilon > 0$ there exists:

1) a time t_1 such that for all $t \geq t_1$ there is a nodal frequency $f_i(\tau)$ that satisfies

$$|f_i(\tau) - f_a| < \varepsilon \quad \text{for} \quad \tau\varepsilon[t,t+T]$$

2) a time t_2 such that for all $t \geq t_2$ there is a nodal

frequency $f_k(\tau)$ that satisfies

Proof $\quad |f_k(\tau) - f_b| < \varepsilon \quad$ for $\quad \tau \varepsilon [t, t+T]$

The time instants pertinent to this proof are illustrated in Fig. 2.4.

Since

$$\lim_{t\to\infty} \hat{f}(t) = f_a$$

then from the definition of the limit we know that for every $\varepsilon_1 > 0$ there exists a t_1 such that for all $t \geq t_1$

$$|\hat{f}(t) - f_a| \leq \varepsilon_1 .$$

The fact that $\hat{f}(t)$ is monotonic further implies that

$$f_a \leq \hat{f}(t) \leq f_a + \varepsilon_1 , \quad \text{for } t \geq t_1 . \tag{2.12}$$

Recalling the definition of $\hat{f}(t)$ we can now write

$$\bar{f}(t) \leq f_a + \varepsilon_1 , \quad t \geq t_1 - \bar{\tau} \tag{2.13a}$$

or

$$f_j(t) \leq f_a + \varepsilon_1 , \; j = 1,2,\ldots,N, \quad t > t_1 - \bar{\tau}. \tag{2.13b}$$

Let us choose

$$\varepsilon_1 = \frac{\varepsilon}{e^{\bar{L}(T+\bar{\tau})}-1} \tag{2.14a}$$

where

$$\bar{L} = \max_i L_i . \tag{2.14b}$$

Assume that $t \geq t_1$. There must be a node i and a time

$t^* \varepsilon [t+T,\ t+T+\bar{\tau}]$ such that

$$f_i(t^*) = \hat{f}(t+T+\bar{\tau}).$$

Clearly then

$$f_a \lesssim f_i(t^*) \lesssim f_a + \varepsilon_1. \tag{2.15}$$

Let $t_0 \varepsilon [t, t+T]$. Then from (2.8) we have:

$$f_i(t^*) = f_i(t_0) e^{-L_i(t^*-t_0)} + L_i \sum_{j=1}^{N} \int_{t_0}^{t^*} A_{ij}(\tau) f_j(\tau-\tau_{ij}) e^{-L_i(t^*-\tau)} d\tau. \tag{2.16}$$

Making use of (2.13) and (2.15) the above becomes:

$$f_a \lesssim f_i(t_0) e^{-L_i(t^*-t_0)} + L_i(f_a+\varepsilon_1) \int_{t_0}^{t^*} e^{-L_i(t^*-\tau)} d\tau \tag{2.17}$$

where we have once again made use of the fact that

$$\sum_{j=1}^{N} A_{ij}(\tau) = 1.$$

Performing the integration in (2.17) we obtain:

$$f_i(t_0) e^{-L_i(t^*-t_0)} \gtrsim f_a - (f_a+\varepsilon_1)\left[1-e^{-L_i(t^*-t_0)}\right]$$

or

$$f_i(t_0) \gtrsim f_a - \varepsilon_1(e^{L_i(t^*-t_0)}-1)$$

or

$$f_i(t_0) \gtrsim f_a - \varepsilon_1 (e^{\bar{L}(T+\bar{\tau})} - 1).$$

Substituting from (2.14a) gives:

$$f_i(t_0) \gtrsim f_a - \varepsilon.$$

Also $f_i(t_0) \lesssim f_a + \varepsilon_1$, and since $\varepsilon > \varepsilon_1$ this implies that $f_i(t_0) \lesssim f_a + \varepsilon$. Therefore we can write

$$|f_i(t_0) - f_a| < \varepsilon \quad \text{for } t_0 \varepsilon [t, t+T]$$

which proves the first part of the Theorem, namely that within the interval [t, t+T] there is an oscillator $f_i(t)$, whose frequency is within ε of f_a. The proof of the second half of this theorem parallel's the first half.

Since $\lim_{t\to\infty} f(t) = f_b$, then for every $\varepsilon_1 > 0$ there exists a t_2 such that for $t > t_2$

$$|\underset{\sim}{f}(t) - f_b| \lesssim \varepsilon_1$$

or

$$f_b - \varepsilon_1 \lesssim f(t) \lesssim f_b.$$

This implies that for $t \gtrsim t_2 - \bar{\tau}$, $f_j(t) \gtrsim f_b - \varepsilon_1$ for each node j. Let ε_1 and $\bar{L}$ be defined as in (2.14) and let $t \gtrsim t_2$, then there must be a node k and time $t^* \varepsilon [t+T, t+T+\bar{\tau}]$ such that

$$f_k(t^*) = \underset{\sim}{f}(t+T+\bar{\tau}).$$

Analogous to (2.16) we now have:

$$f_k(t^*) = f_k(t_0)e^{-L_k(t^*-t_0)} + L_k \sum_j \int_{t_0}^{t^*} A_{kj}(\tau) f_j(\tau-\tau_{kj}) e^{-L_k(t^*-\tau)} d\tau \qquad (2.18)$$

where $t_0 \varepsilon [t, t+T]$.

Using $f_j(t) \gtrsim f_b - \varepsilon_1$ for $t > t_2 - \bar{\tau}$, and since $f_k(t^*) \lesssim f_b$ (2.18) becomes:

$$f_b \gtrsim f_k(t_0)e^{-L_k(t^*-t_0)} + (f_b-\varepsilon_1)\left[1-e^{-L_k(t^*-t_0)}\right] \qquad (2.19)$$

which reduces to

$$f_k(t_0) \lesssim f_b + \varepsilon_1[e^{L_k(t^*-t_0)} - 1]$$

or

$$f_k(t_0) \lesssim f_b + \varepsilon.$$

Since in addition $f_k(t_0) \gtrsim f_b - \varepsilon$, we can write

$$|f_k(t_0) - f_b| \lesssim \varepsilon \quad \text{for} \quad t_0 \varepsilon [t, t+T],$$

which completes the proof of the second part of this theorem.

We have shown that for every $\varepsilon > 0$ and $T > 0$, there are times t_1 and t_2 such that for all time intervals of length T beginning after t_1 an oscillator remains within ε of frequency f_a, and that for intervals beginning after t_2 an oscillator remains within ε of f_b. Combining these two results we have that if $t \gtrsim \max[t_1, t_2]$, then for all $\tau \varepsilon [t, t+T]$ there must be nodes i and j, not necessarily distinct, satisfying

$$|f_i(\tau) - f_a| < \varepsilon$$

and

$$|f_j(\tau) - f_b| < \varepsilon.$$

The above theorem states that we can find an arbitrarily long time interval during which some node remains arbitrarily close to frequency f_a, and some node remains arbitrarily close to frequency f_b. This fact plus the connectivity properties of the network will be used to prove that f_a must equal f_b.

2.4 BOUND ON THE BUFFER SIZE

In order to continue with the stability proof, we must first establish some bounds on the size of the buffer contents. As discussed in the introduction, the fact that we can compute simple buffer content bounds that are independent of network size or configuration is an important advantage of Peak Synchronization over other techniques.

We are assuming that for $t < 0$ all oscillators were free running. That is $f_i(t) = f_{i0}$ for $t < 0$. From this we see that

$$\hat{f}(0) = \max_{[-\bar{\tau},0]} \bar{f}(t) = \max_i f_{i0}$$

likewise,

$$\underset{\vee}{f}(0) = \min_{[-\bar{\tau},0]} \underline{f}(t) = \min_i f_{i0}.$$

Define

$$\bar{f}_0 = \max_i f_{i0}$$

and

$$\underline{f}_0 = \min_i f_{i0}.$$

Then from Theorem 2.1 it is clear that

$$\underline{f}_0 \leq f_i(t) \leq \bar{f}_0 \quad t > 0,\ i = 1,2,\ldots,N \tag{2.20}$$

From (2.1) we have that

$$b_{imax}(t) \triangleq \max_{j\varepsilon S_i} \{b_{ij}(t)\} = \frac{f_i(t) - f_{i0}}{L_i} \tag{2.21}$$

Using (2.20) and (2.21) we have:

$$b_{imax}(t) \leq \frac{\bar{f}_0 - \underline{f}_0}{L_i} = \frac{\Delta F}{L_i} \tag{2.22}$$

where $\Delta F = \bar{f}_0 - \underline{f}_0$ is the maximum uncontrolled frequency difference between two oscillators. Equation (2.22) provides an upper bound on how full a buffer at node i can get (a half full buffer has value zero). The largest buffer anywhere in the network is clearly bounded by

$$b_{max} \leq \frac{\Delta F}{\underline{L}} \tag{2.23}$$

where $\underline{L} = \min_i L_i$.

We would also like to specify a lower bound on the buffer size. That is a bound on how negative or depleted a buffer can become. Such a bound is however not as simple to arrive at. In fact, the existence of a lower bound depends on the network being strongly connected. A strongly connected network is one in which there is a directed path from every node to every other node. For simplicity, and because the

assumption would generally be satisfied in practice we will assume that all communication links are bilateral (the connection matrix $\{a_{ij}\}$ is symmetrical). In this case the network is automatically strongly connected if it is connected.

With the assumption that the network is connected, and that communication links are bilateral, we now establish a lower bound for the buffer contents. The lower bound follows from the upper bound and the fact that the sum $b_{ij}(t)+b_{ji}(t)$ remains small. Let us assume that $b_{ij}(0) = b_{ji}(0) = 0$. From (1.1) we have

$$b_{ij}(t) = \int_0^t [f_j(s-\tau_{ij}) - f_i(s)]ds \tag{2.24}$$

Using the fact that $f_j(t) = f_{j0}$ for $t \leq 0$ this becomes

$$b_{ij}(t) = f_{j0}\tau_{ij} + \int_0^{t-\tau_{ij}} f_j(s)ds - \int_0^t f_i(s)ds. \tag{2.25}$$

Since we have assumed that all links are bilateral there must be a buffer $b_{ji}(t)$ which is similarly given by:

$$b_{ji}(t) = f_{i0}\tau_{ji} + \int_0^{t-\tau_{ji}} f_i(s)ds - \int_0^t f_j(s)ds \tag{2.26}$$

Therefore we can write:

$$b_{ij}(t) + b_{ji}(t) = f_{j0}\tau_{ij} + f_{i0}\tau_{ji} - \int_{t-\tau_{ji}}^t f_i(s)ds - \int_{t-\tau_{ji}}^t f_j(s)ds \tag{2.27}$$

Using (2.20) we have:

$$b_{ij}(t) + b_{ji}(t) \gtrsim \underline{f}_0(\tau_{ij}+\tau_{ji}) - \bar{f}_0(\tau_{ji}+\tau_{ij})$$

or

$$b_{ij}(t) \gtrsim - b_{ji}(t) - 2\bar{\tau}\Delta F. \qquad (2.28)$$

Finally, since $b_{ji}(t) \lesssim \Delta F/\underline{L}$ we have

$$b_{ij}(t) \gtrsim - \frac{\Delta F}{\underline{L}} - 2\bar{\tau}\Delta F = -\Delta F(\frac{1}{\underline{L}} + 2\bar{\tau})$$

which is the desired lower bound. Referring back to the section on parameters in chapter 1, it is of interest to note that generally $1/\underline{L} >> 2\bar{\tau}$, and that therefore the positive and negative buffer bounds are approximately symmetric.

2.5 PROOF OF STABILITY

We are now in a position to complete the proof of stability. That is, we can now show that the limits of the functions $\hat{f}(t)$ and $\underset{\sim}{f}(t)$ are identical. From this it immediately follows that all oscillators approach a common frequency.

Theorem 2.3

If the network is strongly connected then

$$\lim_{t\to\infty} \hat{f}(t) = \lim_{t\to\infty} \underset{\sim}{f}(t)$$

Proof

The proof is by contradiction. We assume that the Theorem is false and that the limits are distinct. Therefore

$f_a \neq f_b$. We will show that as a consequence of this assumption the buffer bounds previously demonstrated cannot be satisfied.

From Theorem 2.2 we know that for every $\varepsilon > 0$ and $T > 0$ we can find a time interval $[t,t+T]$ and nodes i and j that satisfy

$$|f_i(\tau) - f_a| < \varepsilon$$

$$|f_j(\tau) - f_b| < \varepsilon, \text{ for } \tau\varepsilon[t,t+T] \qquad (2.29)$$

If nodes i and j are the same, then from (2.29) $|f_a - f_b| < 2\varepsilon$. Since ε is arbitrarily small we can conclude that $f_a = f_b$. Therefore we need only consider the case when $i \neq j$.

Since the network is connected there must be a directed path from node i to node j. Let k denote the numbers of intermediary nodes along one such path and call these nodes $n_1, n_2, \ldots, n_k$. This situation is illustrated in Fig. 2.5. To simplify notation, node i is called n_0, and node j is called n_{k+1}.

We now compute the sum of all the buffer contents along this path at time t+T. This sum is given by:

$$B_T = \sum_{j=1}^{k+1} b_{n_j,n_{j-1}}(t+T) \qquad (2.30)$$

Substituting from (1.1) gives:

$$B_T = \sum_{j=1}^{k+1} b_{n_j,n_{j-1}}(t+\bar{\tau}) + \sum_{j=1}^{k+1} \int_{t+\bar{\tau}}^{t+T} [f_{n_{j-1}}(s-\tau_{n_j,n_{j-1}}) - f_{n_j}(s)]ds \tag{2.31}$$

define $b_{T0} = \sum_{j=1}^{k+1} b_{n_j,n_{j-1}}(t+\bar{\tau})$ then from (2.31) we can write:

$$B_T = b_{T0} + \int_{t+\bar{\tau}}^{t+T} f_{n_0}(s-\tau_{n_1,n_0})\,ds - \int_{t+\bar{\tau}}^{t+T} f_{n_{k+1}}(s)ds$$

$$+ \sum_{j=1}^{k} \int_{t+\bar{\tau}}^{t+T} [f_{n_j}(s-\tau_{n_{j+1},n_j}) - f_{n_j}(s)]ds \tag{2.32}$$

Making use of (2.29) and (2.20), we can establish the following bound:

$$B_T \gtrsim b_{T0} + (T-\bar{\tau})\left[(f_a-\varepsilon) - (f_b+\varepsilon)\right] - k\bar{\tau}\Delta F$$

or

$$B_T \gtrsim b_{T0} + T(f_a-f_b) - 2\varepsilon T - \bar{\tau}[f_a-f_b-k\Delta F-2\varepsilon] \tag{2.33}$$

By assumption $f_a \neq f_b$, and therefore $f_a > f_b$. Since T and ε are completely arbitrary positive numbers, the right hand side of (2.33) can be made arbitrarily large. Therefore the sum of the buffer contents cannot be bounded. This is a contradiction and the theorem is proved.

The preceding theorem established the fact that

$$\lim_{t\to\infty} \hat{f}(t) = \lim_{t\to\infty} \underset{\sim}{f}(t) = f_c$$

where f_c is now used to represent the common value of f_a and

f_b. Since for each i

$$\underset{\sim}{f}(t) \lesssim f_i(t) \lesssim \hat{f}(t)$$

we must have that

$$\lim_{t\to\infty} f_i(t) = f_c$$

which establishes the stability of Peak Synchronization in the sense that all oscillators eventually achieve a common frequency.

2.6 COMPARISON OF PEAK SYNCHRONIZATION WITH MUTUAL AND MASTER SLAVE SYNCHRONIZATION

We have seen that in Peak Synchronization the buffer contents satisfy the bounds

$$-\Delta F(1/\underline{L} + 2\bar{\tau}) \lesssim b_{ij}(t) \lesssim \frac{\Delta F}{\underline{L}},$$

and that these bounds are independent of network size. We have already mentioned in chapter 1, that for a "dumbell network" mutual synchronization exhibits a growth in buffer size proportional to N^2, where N is the number of nodes in the network. We will now show that with Master-Slave synchronization, the maximum buffer size can increase linearly with N.

Consider the Master-Slave network illustrated in Fig. 2.6. In the figure the master node and all nodes on the left-hand side have uncontrolled frequency 100. The

nodes on the right-hand side have uncontrolled frequency 50. The frequency at each node is governed by the equation

$$f_i(t) = f_{i0} + L_i\, b_{ij}(t)$$

where $b_{ij}(t)$ represents the buffer being synchronized from. Therefore, assuming $L_i = 1$, each buffer on the right-hand side will develop a contents of 50 bits, whereas the buffers on the left-hand side will contain zero. For simplicity we will assume that the transmission links have zero delay, and that $b_{ij}(0) = 0$ for all i,j. In this case the sum of the buffer contents around a closed loop will always be zero. This is because every oscillator in the loop is filling one buffer at the same rate as it is emptying another. Thus the total contents of all the buffers cannot change, and must remain zero. Using this fact, we can compute the contents of the buffer labeled b_1. Namely:

$$b_1 + 4 \cdot 50 = 0$$

or

$$b_1 = -200.$$

Since buffers b_1 and b_2 form a loop they must sum to zero, which means $b_2 = +200$ bits. This value greatly exceeds the Peak Synchronization bound of $\Delta F/\underline{L}$, which in this case would equal 50 bits.

It is easy to see that increasing the size of the network, while maintaining the same structure, would make the

contents of b_1 (and b_2) increase linearly with the network size. Specifically the contents of b_1 would be given by

$$b_1 = \left(\frac{N-1}{2}\right) \cdot 50$$

where again N is the number of nodes.

To summarize, in Mutual Synchronization the maximum buffer size can increase as N^2, in Master-Slave Synchronization it can increase linearly with N, but for Peak Synchronization the maximum buffer size is bounded independent of network size.

2.7 CONCLUSION

In this chapter we have demonstrated some of the main features of Peak Synchronization that make it attractive as a synchronization technique. We have shown that the system is stable, independent of the size of the transmission delays or gain factors, and that the monotonic behavior of the oscillators insures that the final system frequency will lie within the range of the uncontrolled clock frequencies.

Concerning buffer sizes, we have proved that the buffers are bounded, and that this bound depends simply on the oscillator inaccuracies and gain factors. The positive and negative buffer bounds were shown to be approximately symmetric for the small delays expected in a terrestrial network. For two other common network synchronization schemes, Mutual

Synchronization and Master-Slave Synchronization, we noted that both had the potential for buffer size increasing with network size.

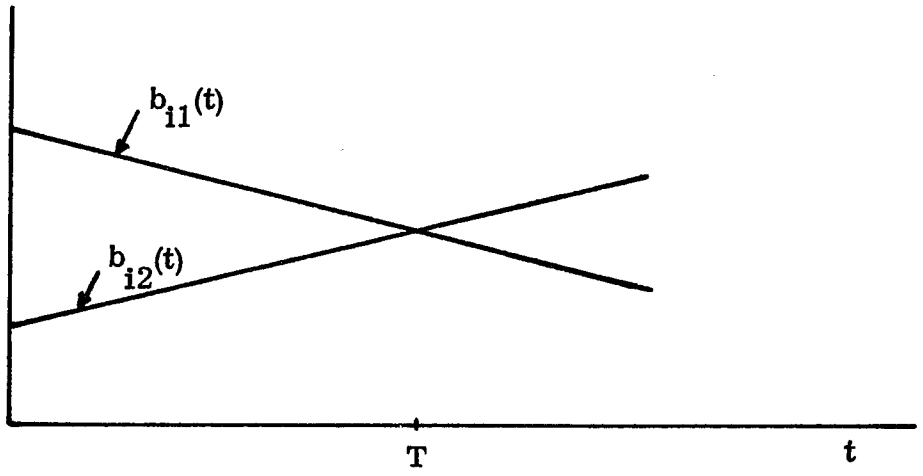

Fig. 2.1a At time T $b_{i2}(t)$ overtakes $b_{i1}(t)$.

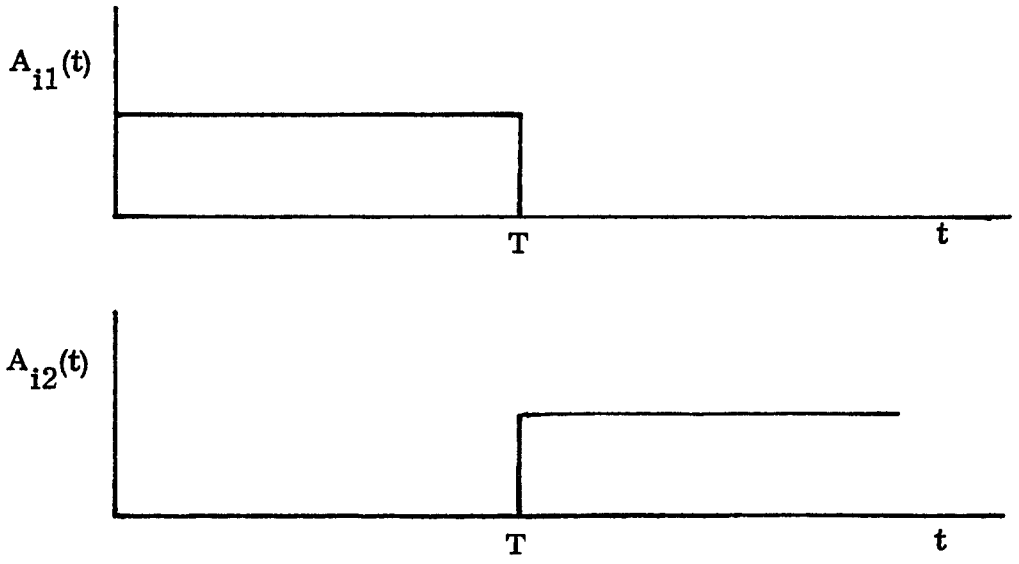

Fig. 2.1b Illustrates $A_{i1}(t)$ and $A_{i2}(t)$ at a switching instant.

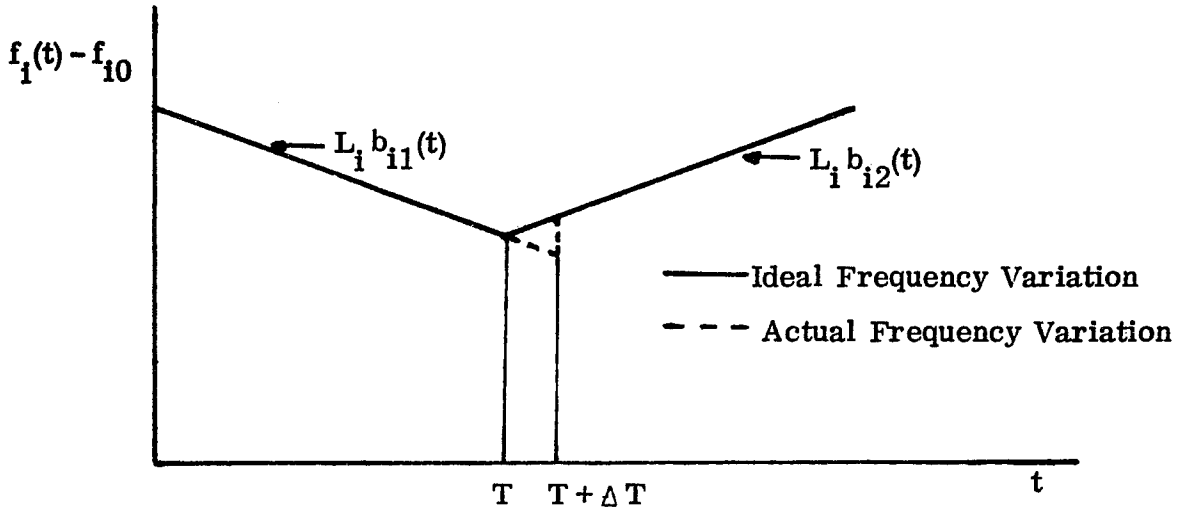

Fig. 2.2 A small frequency discontinuity that would result in a practical system.

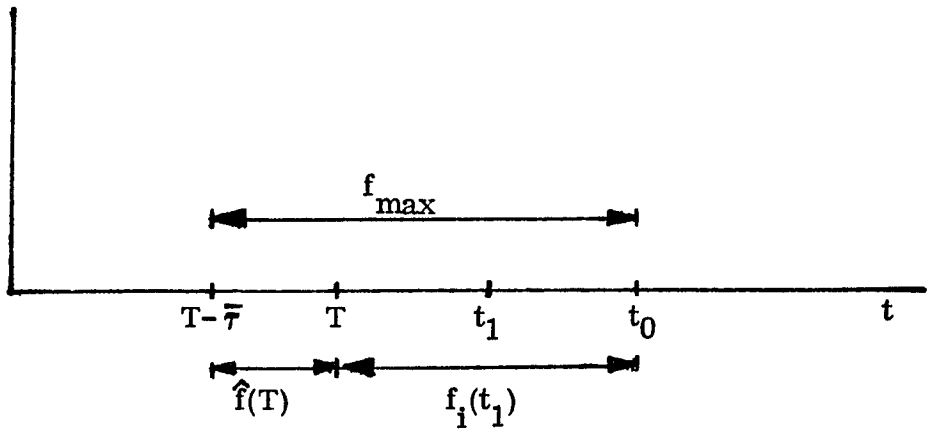

Fig. 2.3 Illustrates the time instances involved in Theorem 2.1, and the various maximum frequencies discussed there.

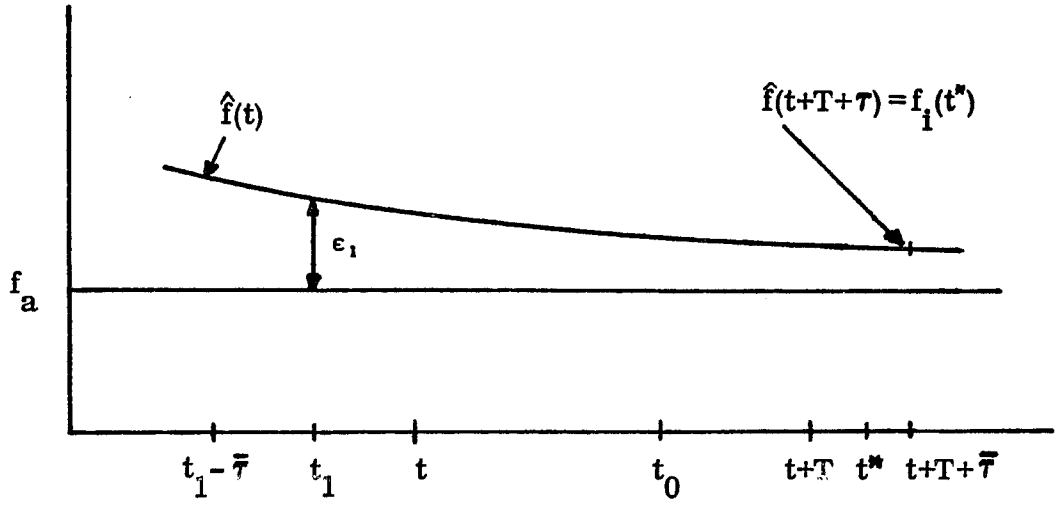

Fig. 2.4 Relationship between the time instances involved in Theorem 2.2.

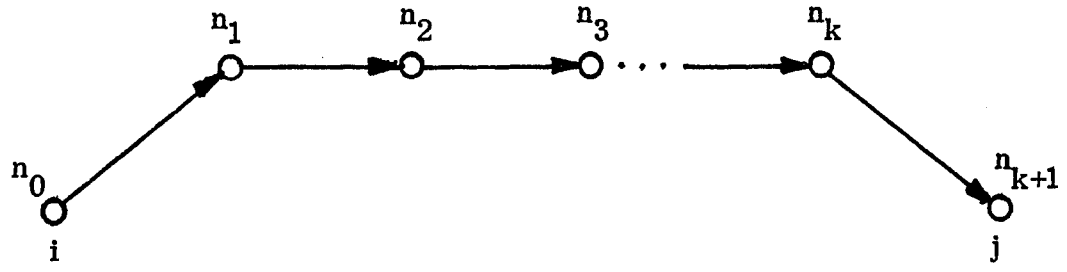

Fig. 2.5 A directed path from node i to node j.

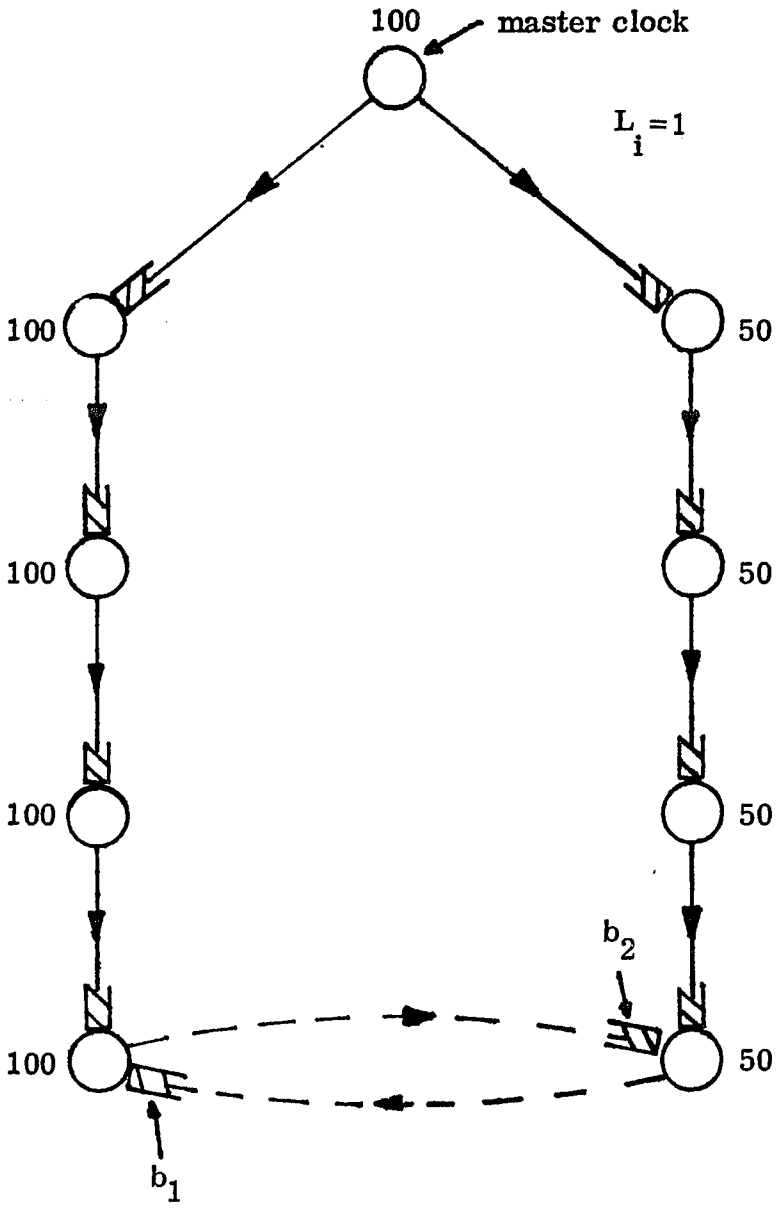

Fig. 2.6 Example of Master - Slave synchronization. Solid lines show the synchronization subnetwork.

Chapter 3

DYNAMIC RESPONSE OF PEAK SYNCHRONIZATION

3.1 INTRODUCTION

In the previous chapter we were able to conclude that in a network employing Peak Synchronization all oscillators will eventually settle down to some common frequency. This chapter will address the problem of describing the transient response, and of determining the value of the steady state frequency.

We are able to calculate the response of Peak Synchronization because a synchronization subnetwork is formed whose structure is independent of the nature of the original network. Thus the first part of this chapter will be devoted to determining the structure of the synchronization subnetwork. We shall show that this subnetwork always consists of a directed tree with an additional link forming a loop. The nodes in the loop mutually synchronize, and all other nodes are slaved to the loop nodes.

In studying the response of the nodes in the synchronization loop, we shall find that the problem is greatly complicated by the presence of transmission delays. Specifically, the time delays result in a characteristic equation that is transcendental and has an infinite number of poles. Fortunately, however, we will see that for the small values of delay expected, the response can be very accurately approximated by considering

only one non-zero pole. We will also present an exact time domain method for computing the response. This will be used to verify the accuracy of the approximate technique.

Finally, we will study the effect of oscillator jitter. This is done by assuming that a random "noise process" is added to the free running oscillator frequencies. We will then determine how much random variation is imposed on the controlled oscillator frequencies. It will be shown that the variance of the controlled frequencies is less than that of the input noise process.

3.2 STRUCTURE OF THE SYNCHRONIZATION SUBNETWORK

In Peak Synchronization a part of the original network is used for synchronization purposes. This synchronization subnetwork arises since each node chooses one of its incoming links to synchronize from. It is the collection of these "chosen" links that form the synchronization subnetwork. In order to study the dynamic response of Peak Synchronization, we first need to determine the structure of this subnetwork. We shall see that this structure is independent of the original network.

Before proceeding we will first state some definitions and theorems that apply to directed graphs. The proofs of these theorems can be found in Deo [39].

Definition 1: The Indegree of node i is equal to the number of edges incident into node i. Similarly, the Outdegree of

node i is the number of edges incident out of node i. The degree of node i equals the sum of its indegree and outdegree.

Definition 2: An Arborescence is a directed graph with no loops, and precisely one node i of zero indegree. Node i is called the root of the arborescence.

Theorem 3.1

An arborescence is a tree in which every node other than the root has indegree of exactly one.

Theorem 3.2

In an arborescence there is a directed path from the root to every other node.

We will also refer to an arborescence as a directed tree. An example of an arborescence is shown in Fig. 3.1.

The synchronization subnetwork may in general consist of several separate or unconnected parts. In light of the fact that the entire network must reach a common frequency, it is necessary for the synchronization subnetwork to evolve into a connected network, if it didn't start out as one. However, we will deal with the general situation in which separate parts may exist. The nodes in a single separate part are by definition connected. We now establish the following important result.

Theorem 3.3

Each separate part of the synchronization subnetwork

consists of a directed tree (arborescence) plus an additional link that forms a single directed loop.

Proof

Let nodes $a_1, a_2, \ldots a_r$ form a separate part. Clearly this separate part contains r links and therefore must contain a loop. To show that this is a directed loop, let us remove one edge from the loop. Say that the removed edge went from node a_i to node a_j. What remains is a connected network with r-1 edges, hence a tree. In addition, since each node except a_j has indegree of one, the remaining network is an arborescence with root a_j (Theorem 3.1). By Theorem 3.2 there must therefore be a directed path from node a_j to node a_i. Replacing the removed edge completes a directed loop by connecting a_i to a_j. Thus the theorem is proved.

An example of a synchronization subnetwork with two separate parts is shown in Fig. 3.2. As can be seen in this figure, each separate part contains a single loop. In addition, if all communication links are bilateral, then each loop will consist of eactly two nodes. This is because each separate part contains a highest frequency node, to which all neighboring nodes will synchronize. Also, the highest node will synchronize from its neighbor of highest frequency, thus establishing a two node loop.

If we restrict our consideration therefore to networks where all communication links are bilateral, we obtain a very

simple form for the synchronization subnetwork. Each separate part consists of a loop of two nodes, with these two nodes mutually synchronizing. All other nodes are slaved directly or indirectly to one of the two nodes in the loop.

3.3 DYNAMIC RESPONSE OF A TWO NODE LOOP

We are now in a position to study the dynamic response of Peak Synchronization. We will first study the behavior of the two nodes forming the loop. Following this, we will consider the response of the slave nodes.

Let us assume that nodes 1 and 2 form a loop as shown in Fig. 3.3. The behavior of these two nodes is governed by the equations:

$$f_1(t) = f_{10} + L_1 b_{12}(t) \tag{3.1a}$$

$$f_2(t) = f_{20} + L_2 b_{21}(t) \tag{3.1b}$$

Differentiating (3.1) we obtain:

$$\dot{f}_1(t) = L_1 [f_2(t-\tau_{12}) - f_1(t)] \tag{3.2a}$$

$$\dot{f}_2(t) = L_2 [f_1(t-\tau_{21}) - f_2(t)]. \tag{3.2b}$$

We assume that the loop above was formed at time t=0, and proceed to determine the response for $t > 0$. Clearly the calculated response is only correct for the time interval during which the loop persists. Taking the Laplace Transform of (3.3) yields:

$$sF_1(s)-f_1(0) = L_1\left\{e^{-s\tau_{12}}\left[\int_{-\tau_{12}}^{0} f_2(t)e^{-st}\,dt+F_2(s)\right] - F_1(s)\right\} \qquad (3.3)$$

In this instance, and in the future, we generally specify results only for node 1. The corresponding results for node two are obtained simply by changing subscript 1 to 2 and 2 to 1.

If we assume that the response time of the system is much greater than the time delay, then we may approximate

$$f_1(t) \approx f_1(0) \qquad t\varepsilon[-\tau_{21},0]$$
$$f_2(t) \approx f_2(0) \qquad t\varepsilon[-\tau_{12},0]$$

With this simplification we can obtain from (3.3)

$$sF_1(s) = f_1(0)+L_1\left[\frac{f_2(0)}{s}(1-e^{-s\tau_{12}})+e^{-s\tau_{21}}F_2(s)-F_1(s)\right] \qquad (3.4a)$$

$$sF_2(s) = f_2(0)+L_2\left[\frac{f_1(0)}{s}(1-e^{-s\tau_{21}})+e^{-s\tau_{21}}F_1(s)-F_2(s)\right] \qquad (3.4b)$$

These equations can be solved to yield:

$$F_1(s) = \frac{f_1(0)\left[s+L_2+\frac{L_1L_2}{s}(1-e^{-\tau_{21}s})e^{-\tau_{12}s}\right]+F_2(0)\left[L_1+\frac{L_1L_2}{s}(1-e^{-\tau_{12}s})\right]}{(s+L_1)(s+L_2) - L_1L_2e^{-(\tau_{12}+\tau_{21})s}} \qquad (3.5)$$

3.3.1 Asymptotic Behavior of the Oscillators

We can now employ the Laplace Transform final value theorem to find $f_1(\infty)$ and $f_2(\infty)$. Specifically this theorem

states that

$$f_1(\infty) = \lim_{s \to 0} sF_1(s) \tag{3.6}$$

Applying this theorem to (3.5), we get

$$f_1(\infty) = \frac{f_1(0)\,[L_2+L_1L_2\tau_{21}] + f_2(0)\,[L_1+L_1L_2\tau_{12}]}{L_1+L_2 + L_1L_2(\tau_{12}+\tau_{21})} \tag{3.7}$$

We note that $f_1(\infty) = f_2(\infty)$ as is required by stability. Substituting for $f_1(0)$ and $f_2(0)$ from (3.1) the above becomes:

$$f_1(\infty) = \frac{[f_{10}+L_1B_{12}(0)\,](L_2+L_1L_2\tau_{21}) + [f_{20}+L_2B_{21}(0)\,](L_1+L_1\tau_{12})}{L_1+L_2 + L_1L_2(\tau_{12}+\tau_{21})} \tag{3.8}$$

If we consider the simple case where $\tau_{12} = \tau_{21} = \tau$ and $L_1 = L_2 = L$, then the above reduces to

$$f_1(\infty) = \frac{f_{10}+f_{20}}{2} + \frac{L}{2}\,[B_{12}(0) + B_{21}(0)\,] \tag{3.9}$$

In the previous chapter we showed that $B_{12}(t) + B_{21}(t)$ always remains small. We can therefore conclude that

$$f_1(\infty) \simeq \frac{f_{10}+f_{20}}{2}\,. \tag{3.10}$$

Thus we see that the final steady state frequency is approximately the average of the free running frequencies of the two

oscillators forming the loop.

3.3.2 Approximate Calculation of the Dynamic Response

In order to determine the dynamic response of the two node loop we could, in principle, find the inverse transform of (3.5). This equation is, however, extremely complex, and so we will only consider the simpler case where the time delays and gains are equal. In this case (3.5) becomes:

$$F_1(s) = \frac{f_1(0)\left[s+L+\frac{L^2}{s}(1-e^{-s\tau})e^{-s\tau}\right]+f_2(0)[L+\frac{L^2}{s}(1-e^{-s\tau})]}{(s+L)^2 - L^2e^{-2s\tau}} \qquad (3.11)$$

If we turn our attention first to the simplest case of no time delay ($\tau=0$), this equation simplifies to

$$F_1(s) = \frac{f_1(0)[s+L] + Lf_2(0)}{s(s+2L)} \qquad (3.12)$$

In this case we can easily invert the Laplace Transform using a partial fraction expansion to obtain

$$f_1(t) = \frac{f_1(0) + f_2(0)}{2} + \frac{f_1(0) - f_2(0)}{2} e^{-2Lt} \qquad (3.13)$$

Thus we see that for zero time delay the two node loop has a simple exponential response, with time constant 1/2L. We found in chapter 1, that for crystal oscillators an appropriate value for L is $\approx 10^{-2}$. Therefore the time constant would be ≈ 50 seconds.

We will now consider the case of very small time delays. In this case we use the approximation

$$e^{-x} = 1 - x. \tag{3.14}$$

Substituting in (3.11) we get

$$F_1(s) = \frac{f_1(0)\left[s(1-L^2\tau^2) + L(1+L\tau)\right] + f_2(0)[L(1+L\tau)]}{s[s+2L(1+L\tau)]} \tag{3.15}$$

Performing a partial fraction expansion, we obtain for the corresponding time function

$$f_1(t) = \frac{f_1(0) + f_2(0)}{2} + \frac{f_1(0)[1-2L^2\tau^2]-f_2(0)}{2} e^{-2L(1+L\tau)t} \tag{3.16}$$

We note that the asymptotic value expressed above is correct, however, the initial value of $f_1(t)$ is somewhat in error due to the approximation in (3.14). It is of interest to note that the effect of the small time delay is to lower the time constant, and therefore hasten the response of the network.

In order to determine the dynamic response when the delays are not small we need to invert (3.11). One method for doing this would be to determine the zeros of the denominator (the poles), and perform a partial fraction expansion. The denominator however, is transcendental and may have an infinite number of zeros. We can, however, approximate the response

by locating and considering only the dominant poles (those that have the largest real part).

By factoring the denominator of (3.11) we have that the poles are given by the solutions of

$$(s+L-Le^{-st})(s+L+Le^{-s\tau}) = 0 \tag{3.17a}$$

or

$$s + L = \pm Le^{-s\tau} \tag{3.17b}$$

Suppose that the roots of (3.17) are $s_o, s_1, s_2, \ldots$ then the inverse transform of (3.11) can be written in the general form

$$f_1(t) = \sum_{r=0}^{\infty} p_r(t) e^{s_r t} \tag{3.18}$$

where $p_r(t)$ represents a polynomial whose degree is less than the order of the pole at s_r[40]. $p_r(t)e^{s_r t}$ is, in fact, the residue of $F_1(s)e^{st}$ at the pole s_r.

In order to simplify the discussion that follows we normalize (3.17) by substituting

$$s = \hat{s}\, L$$

and as a result (3.17) becomes

$$\hat{s} + 1 = \pm e^{-\alpha \hat{s}} \quad , \quad \text{where } \alpha \triangleq L\tau. \tag{3.19}$$

We now wish to show that the poles are, in fact, all simple.

To do this we use the fact that if f(s) has a zero of order 2 or greater at s_r then $f(s_r) = 0$, and $f'(s_r) = 0$. We consider the following three possibilities separately:

$$1)\quad g_1(\hat{s}_r) \triangleq \hat{s}_r + 1 - e^{-\alpha\hat{s}_r} = 0$$
$$g_1'(\hat{s}_r) = 1 + \alpha e^{-\alpha\hat{s}_r} = 0$$

Solving these two equations leads to

$$e^{(\alpha+1)} = -1/\alpha$$

which is impossible since α is real and positive. Therefore $g_1(\hat{s})$ has only simple roots.

$$2)\quad g_2(\hat{s}_r) \triangleq \hat{s}_r + 1 + e^{-\alpha\hat{s}_r} = 0$$
$$g_2'(\hat{s}_r) = 1 - \alpha e^{-\alpha\hat{s}_r} = 0$$

These two equations lead to

$$\hat{s}_r = -\left(\frac{\alpha+1}{\alpha}\right), \quad e^{(\alpha+1)} = 1/\alpha.$$

Solving the second equation for α yields

$$\alpha = \alpha_o = .27846$$

and the corresponding value of $\hat{s}_r$ is easily found to be $\hat{s}_r = -4.6$. Thus we see that as long as $\alpha \neq \alpha_o$, $g_2(\hat{s})$ also has only simple roots.

3) We now wish to consider whether a root of $g_1(\hat{s})$ can also be a root of $g_2(\hat{s})$. That is:

$$\hat{s}_r + 1 - e^{-\alpha \hat{s}_r} = 0$$
$$\hat{s}_r + 1 + e^{-\alpha \hat{s}_r} = 0$$

These equations lead to

$$e^{\alpha} = 0$$

which clearly cannot be satisfied.

We can summarize therefore by saying that provided $\alpha \neq \alpha_o$ the poles of (3.11) are all simple. Since α can be expected to be of the order of 10^{-2} or 10^{-3} the condition $\alpha \neq \alpha_o$ can safely be assumed. With all poles simple, the polynomials in (3.18) reduce to constants and therefore we have for the inverse of (3.11)

$$f_1(t) = \sum_{r=0}^{\infty} c_r e^{s_r t} \tag{3.20}$$

where c_r is the residue of $F_1(s)$ at the pole s_r. Further, since $F_1(s)$ is the quotient of two analytic functions, we have for c_r [40]

$$c_r = \frac{N(s_r)}{D'(s_r)} \tag{3.21}$$

where $N(s)$ and $D(s)$ are the numerator and denominator of $F_1(s)$. Since zero is a pole of $F_1(s)$ we have from (3.21)

$$c_o = \frac{N(0)}{D'(0)} = \frac{f_1(0) + f_2(0)}{2}$$

Therefore (3.20) becomes

$$f_1(t) = c_o + \sum_{r=1}^{\infty} c_r e^{s_r t} \tag{3.22}$$

Location of the Real Roots

In order to determine the real roots of (3.19) we let $\hat{s} = \hat{x}$, and therefore (3.19) becomes

$$\hat{x} + 1 = \pm e^{-\alpha \hat{x}} . \tag{3.23}$$

Fig. 3.4 contains a plot of the three functions $\hat{x} + 1$, $e^{-\alpha\hat{x}}$ and $-e^{-\alpha\hat{x}}$. Also $-e^{-\alpha\hat{x}}$ is plotted for $\alpha = .1$ and $\alpha = \alpha_o$. It is clear from this figure that $\hat{x} = 0$ is always a root. In addition, we see that for $0 < \alpha < \alpha_o$ there are two additional negative real roots. For example, for $\alpha = .1$ the roots are $\hat{x} = -2.25$ and $\hat{x} = -35.4$. For $\alpha = \alpha_o$ there is a second order root at $\hat{x} = -4.6$. For $\alpha > \alpha_o$ there are no real roots other than $\hat{x} = 0$.

It is clear from this figure that for the small values of α that are generally expected (in the order of 10^{-2} or 10^{-3}), there will be a real root close to $\hat{x} = -2$, and a second real root that is much more negative. For example, with $\alpha = .01$ the real roots are located at $\hat{x} = -2.02$ and $\hat{x} = -647.1$. Clearly the response due to the pole at $\hat{x} = -647$ will be negligible compared to the effect of the pole at $\hat{x} \approx -2$.

Location of the Complex Poles

Identifying the location of the complex poles is more difficult. If we let $\hat{s} = \hat{x} + j\hat{y}$, and equate the real and imaginary parts of (3.19), we get the following two relations:

$$\hat{x} + 1 = \pm e^{-\alpha\hat{x}} \cos \alpha\hat{y} \qquad (3.24a)$$

$$\hat{y} = \mp e^{-\alpha\hat{x}} \sin \alpha\hat{y}. \qquad (3.24b)$$

Dividing (3.24a) by (3.24b) yields

$$\hat{x} = -\hat{y} \cot \alpha\hat{y} - 1. \qquad (3.25)$$

Substituting this into (3.24b) and rearranging terms yields

$$f(u) \triangleq \frac{\sin u}{u} e^{u \cot u} = \pm \frac{1}{\alpha} e^{-\alpha}, \text{ where } u = \alpha\hat{y} \qquad (3.26)$$

Fig. 3.5 shows a plot of $\text{Log}|f(u)|$. The roots corresponding to $\alpha = .01$ are found graphically by equating the function $\log|f(u)|$ to $\log[100e^{-.01}] \approx 2$. Having found the roots $\{u_i\}$, the corresponding values of $\{\hat{x}_i\}$ can be found simply from (3.25). As seen from Fig. 3.5 the N^{th} root of u_N occurs in the interval $(N\pi, (N+\frac{1}{2})\pi)$. These roots tend to approach $(N+\frac{1}{2})\pi$ as N increases.

In Table 3.1 we list the values of u_i and $\hat{x}_i$ for the first ten roots corresponding to $\alpha = .01$. It can be seen that for this value of α the complex roots have a time constant over 300 times shorter than the dominant real root which was previously found to be $\hat{x} \approx -2$.

We can now summarize the preceding section by saying that when the gains and delays are small enough so that $\alpha \lesssim .01$, the response of the two node loop is strongly dominated by the single real pole located at $\hat{x} \simeq -2$.

3.3.3 Exact Solution for the Two Node Loop

The previous sections provide a simple and accurate solution for the behavior of the two node loop when the time delay-gain product (α) is small. For large values of α many terms in (3.22) would have to be computed making that method impractical. In this section we will use a "step-by-step" method to find a closed form solution for an arbitrary value of α.

We begin with equation (3.2), which assuming equal gains and delays is:

$$\dot{f}_1(t) = L[f_2(t-\tau) - f_1(t)] \tag{3.27}$$

This equation can be integrated to yield

$$f_1(t) = f_1(n\tau)e^{-L(t-n\tau)} + e^{-Lt}\int_{n\tau}^{t} Le^{Ls}f_2(s-\tau)\,ds \tag{3.28}$$

where $n\tau \lesssim t \lesssim (n+1)\tau$. Again we are not explicitly writing the corresponding equations for $f_2(t)$. Assuming as before that $f_1(t) = f_1(0)$ and $f_2(t) = f_2(0)$ for $t\varepsilon[-\tau,0]$, (3.28) can be evaluated for n = 0 yielding

$n = 0,\ 0 \leq t \leq \tau$

$$f_1(t) = f_2(0) + [f_1(0) - f_2(0)]e^{-Lt} \tag{3.29}$$

The corresponding expression for $f_2(t)$ in the interval $[0,\tau]$ can now be used to evaluate (3.28) for $n = 1$. The result is

$n = 1,\ \tau \leq t \leq 2\tau$

$$f_1(t) = f_1(0) + [f_1(0)-f_2(0)]\{e^{-Lt}-[1+L(t-\tau)]e^{-L(t-\tau)}\} \tag{3.29}$$

continuing the above procedure yields:

$n = 2,\ 2\tau \leq t \leq 3\tau$

$$f_1(t) = f_2(0) + [f_1(0) - f_2(0)]\cdot$$
$$\{e^{-Lt}-[1+L(t-\tau)]e^{-L(t-\tau)}+\left[1+\frac{L^2}{2}(t-2\tau)^2+(t+2\tau)\right]e^{-L(t-2\tau)}\}$$

Continuing in this manner, we can see that the general solution is:

$$f_1(t) = \frac{1}{2}\{f_1(0) + f_2(0) + [f_2(0)-f_1(0)](-1)^N\} +$$

$$(f_1(0)-f_2(0))\left[\sum_{i=0}^{N}(-1)^i\, e^{-L(t-i\tau)}\sum_{j=0}^{i}\frac{[L(t-i\tau)]^j}{j!}\right] \tag{3.30}$$

where $N = [t/\tau]$, the integer part of t/τ.

Fig. 3.6 shows a graph of the calculated exact response for $\alpha \triangleq L\tau = .1$. This is compared with the response for zero delay which is obtained from (3.13) It is seen that the

response is somewhat hastened but not substantially affected by a time delay of this magnitude. The response for larger values of α is illustrated in Fig. 3.7. We see there that for large time delays, the response takes the form of a dampened oscillation. This is in agreement with the previous section where we found that for $\alpha > \alpha_o = .2785$, the only real pole is at $x = 0$, all other poles being complex. Therefore the response must be oscillatory.

We have also evaluated the response for small values of α, using only the first two terms of (3.22). The accuracy of this approximation is illustrated in Table 3.2 where the approximate and exact calculations are compared for $\alpha = .1$. It is seen that there is a small initial error in the approximate formula, but that this quickly disappears. For smaller values of α the initial error becomes correspondingly smaller. This justifies our previous statement that the response is strongly dominated by the real pole located at $\hat{x} \approx -2$.

3.4 BEHAVIOR OF THE SLAVE NODES

In Fig. 3.8 we illustrate a slave node that is synchronizing from its immediate master. If $f_M(t)$ denotes the master frequency and $f_s(t)$ the slave frequency, then these two frequencies are related by:

$$\dot{f}_s(t) = L[f_M(t-\tau) - f_s(t)] \tag{3.31a}$$

or

$$f_s(t) = f_s(0)e^{-Lt} + e^{-Lt}\int_o^t Le^{L\lambda} f_M(\lambda-\tau)d\lambda \tag{3.31b}$$

If the master node is part of the synchronization loop its frequency is known and given by (3.30). We can therefore integrate (3.31b), and obtain the slave frequency response. By repeating the above procedure we can compute the response of slave nodes located further away from the synchronization loop.

We will restrict ourselves to carrying out the computation for the zero delay case only. For zero delay the master clock frequency, $f_1(t)$, is given by (3.13). Substituting from (3.13) into (3.31b), therefore gives the response of a first level slave node. The result is:

$$f_{s1}(t) = \frac{f_1(0)+f_2(0)}{2} + [f_{s1}(0)-f_2(0)]e^{-Lt} + \frac{f_2(0)-f_1(0)}{2} e^{-2Lt} \tag{3.32}$$

To obtain the response of a second level slave node, $f_{s1}(t)$ is used as the master in (3.31b). Performing the integration gives:

$$f_{s2}(t) = \frac{f_1(0)+f_2(0)}{2} + \frac{f_1(0)-f_2(0)}{2} e^{-2Lt} + \{f_{s2}(0)-f_1(0)+Lt[f_{s1}(0)-f_2(0)]\} e^{-Lt} \tag{3.33}$$

Clearly, the same procedure can be repeated to obtain the response of lower level slave nodes.

In Fig. 3.9 we illustrate a network containing a synchronization loop and two levels of slave nodes. The calculated

response of these oscillators for the shown free-running frequencies is illustrated in Fig. 3.10. We see there, as would be expected, that the lower level slave node responds more slowly than the higher level node.

3.5 OSCILLATOR JITTER

Until this point we have assumed that each oscillator has an uncontrolled frequency that is exactly constant. Practically, however, the frequency of every oscillator will exhibit some random variation or jitter. To account for this oscillator jitter we will assume that a zero mean random process is added to each uncontrolled frequency. Thus the uncontrolled frequency of node i becomes:

$$f_{i0}(t) = f_{i0} + N_{i0}(t)$$

where $N_{i0}(t)$ is a random process representing the jitter in oscillator i.

In this section we will study the effect of oscillator jitter on the response of peak synchronization. We will consider first the response of the two nodes that form the synchronization loop.

If we rewrite (3.1) including the oscillator jitter we have

$$f_1(t) = f_{10} + N_{10}(t) + Lb_{12}(t) \qquad (3.34a)$$

$$f_2(t) = f_{20} + N_{20}(t) + Lb_{21}(t) \qquad (3.34b)$$

Taking the Laplace Transform of the above we obtain

$$F_1(s) = \frac{f_{10}+Lb_{12}(0)}{s} + N_{10}(s) + \frac{L}{s}\left[\frac{f_2(0)}{s}(1-e^{-s\tau})+e^{-s\tau}\,F_2(s)-F_1(s)\right] \tag{3.35}$$

and a similar expression for $F_2(s)$.

We wish to consider only the random component of the response. Therefore, we will drop the constant terms $f_{10}, b_{12}(0)$, and $f_2(0)$. The resulting equations for $F_1(s)$ and $F_2(s)$ can now be solved to obtain:

$$F_1(s) = \frac{s(s+L)N_{10}(s) + sLe^{-s\tau}N_{20}(s)}{(s+L)^2-L^2e^{-2s\tau}} \tag{3.36}$$

Thus the response consists of a term due to $N_{10}(t)$, and a term due to $N_{20}(t)$. If we let $s = j\omega$ we can write (3.36) in the form

$$F_1(j\omega) = H_1(j\omega)N_{10}(j\omega) + H_2(j\omega)N_{20}(j\omega) \tag{3.37}$$

If we assume that $N_{10}(t)$ and $N_{20}(t)$ are independent, then the two terms that contribute to $f_1(t)$ will also be independent. The Power Spectrum of $f_1(t)$ is therefore the sum of the spectra of the individual terms. Thus we have for the spectrum of $f_1(t)$

$$G_1(\omega) = |H_1(j\omega)|^2 G_{n_1}(\omega) + |H_2(j\omega)|^2 G_{n_2}(\omega) \tag{3.38}$$

where $G_{n_1}(\omega)$ is the spectrum of $N_{10}(t)$, and $G_{n_2}(\omega)$ the spectrum of $N_{20}(t)$. If, as might be expected, $N_{10}(t)$ and $N_{20}(t)$ have the same statistics, and we denote their common spectrum by $G_n(\omega)$ then (3.38) becomes

$$G_1(\omega) = |H(j\omega)|^2 G_n(\omega) \tag{3.39}$$

where $|H(j\omega)|^2 = |H_1(j\omega)|^2 + |H_2(j\omega)|^2$

We can now evaluate $|H(j\omega)|^2$ by making use of (3.36), (3.37) and the preceding definition. The result is:

$$|H(j\omega)|^2 = \frac{N^4 + 2\omega^2L^2}{\omega^4+2L^2[\omega^2+L^2+(\omega^2-L^2)\cos 2\omega\tau+2L\omega \sin 2\omega\tau]} \tag{3.40}$$

Let us first consider the simple case when there is no time delay. Setting $\tau = 0$ above gives:

$$|H(j\omega)|^2 = \frac{\omega^2 + 2L^2}{\omega^2 + 4L^2} \tag{3.41}$$

It is apparent that in this case $|H(j\omega)|^2 < 1$. Thus we can conclude that the power spectrum of the random component of $f_1(t)$ is always less than the spectrum of $N_{10}(t)$. From this it is evident that the response, $f_1(t)$, has less total power or variance than the variance of $N_{10}(t)$.

For the case of arbitrary time delay $|H(j\omega)|^2$ may assume values greater than 1. However, in this case we still have $\lim_{\omega\to\infty} |H(j\omega)|^2 = 1$. If we restrict ourselves to small values of τ

and approximate $e^x \approx 1 + x$ in (3.36) we obtain for $|H(j\omega)|^2$:

$$|H(j\omega)|^2 = \frac{\omega^2 + 2L^2}{\omega^2 + 4L^2(1+L\tau)^2} . \qquad (3.42)$$

Numerical evaluation of (3.40) and (4.42) shows that the approximation is very accurate provided $L\tau \lesssim .1$. Comparing (3.41) and (3.42) it is clear that for $L\tau \lesssim .1$, the presence of time delay further decreases the amount of jitter in the response $f_1(t)$.

Jitter in Slave Nodes

Let us suppose now that node 1 is a slave node whose immediate master is node 2. In this case (3.35) still correctly describes the behavior of $f_1(t)$, with the understanding that $F_2(s)$ is now a known quantity.

Again, dropping the constant terms we have from (3.35)

$$F_1(s) = \frac{s}{s+L} N_{10}(s) + \frac{Le^{-s\tau}}{s+L} F_2(s). \qquad (3.43)$$

From this the power spectrum of the random component of $F_1(t)$ is

$$G_1(\omega) = \frac{\omega^2}{\omega^2 + L^2} G_n(\omega) + \frac{L^2}{\omega^2 + L^2} G_2(\omega) \qquad (3.44)$$

where $G_n(\omega)$ is the spectrum of $N_{10}(t)$ and $G_2(\omega)$ is the spectrum of the random part of $f_2(t)$. From (3.44) we see that

the spectrum of $f_1(t)$ consists of a high pass filtered $N_{10}(t)$ plus a low pass filtered $f_2(t)$. This situation is intuitively plausible for the following reasons.

If the slave node frequency varies rapidly the synchronization system will not be able to respond rapidly enough to undo these variations, slow variations in the slave node will however be compensated, and therefore these variations are attenuated in (3.44). Likewise a slow variation in the master frequency will allow the slave node to respond, and therefore this will be unattenuated. The slave node cannot, however, respond to rapid master frequency variations. We note that in both cases the transition frequency occurs at $\omega = L$, which indicates a response time of the order of $1/L$.

TABLE 3.1

LOCATION OF THE COMPLEX POLES FOR $\alpha = .01$.

u_i/π	$\hat{x}_i$
1.16	-664
2.25	-691
3.31	-715
4.34	-735
5.36	-752
6.38	-768
7.39	-781
8.40	-793
9.41	-803
10.42	-813

TABLE 3.2

COMPARISON OF EXACT AND APPROXIMATE RESPONSES OF A TWO NODE LOOP FOR $\alpha = .1$

Lt	$F_1(t)$ exact	$F_1(t)$ approx
Ø	1Ø.ØØØ	1Ø.Ø37
.1	9.524	9.526
.2	9.117	9.117
.3	8.791	8.791
.4	8.531	8.531
.5	8.323	8.323
.6	8.157	8.157
.7	8.Ø24	8.Ø24
.8	7.919	7.919
.9	7.834	7.834
1.Ø	7.767	7.767
2.Ø	7.528	7.528

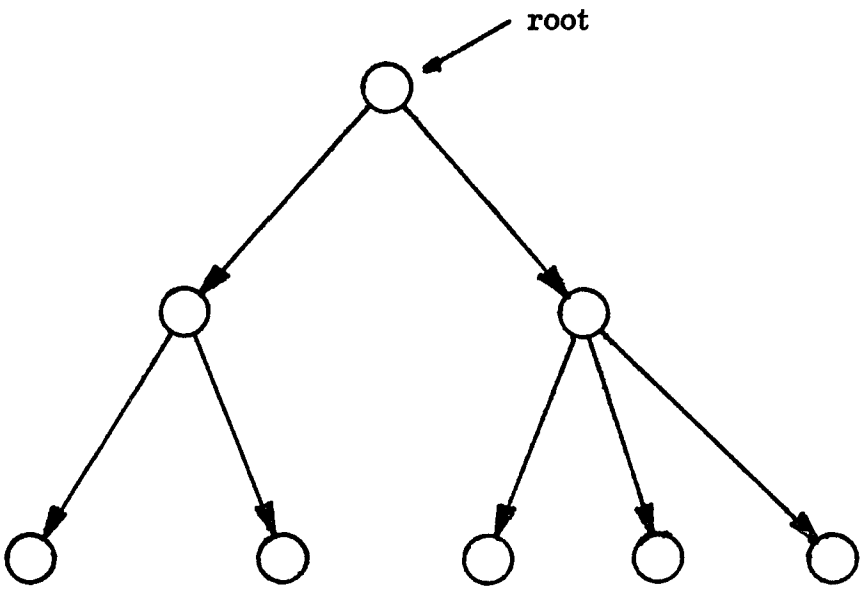

Fig. 3.1 An Arborescence or directed tree.

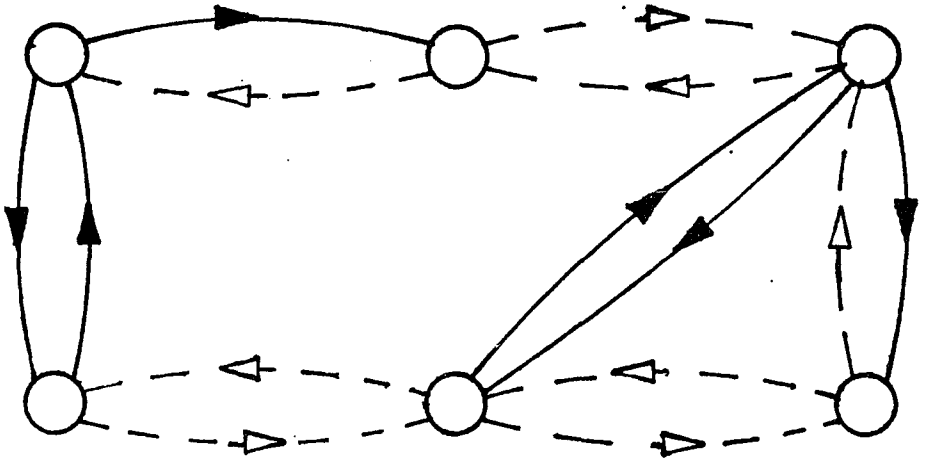

Fig. 3.2 Illustrates a synchronization subnetwork consisting of two separate parts. The solid lines show the synchronization subnetwork.

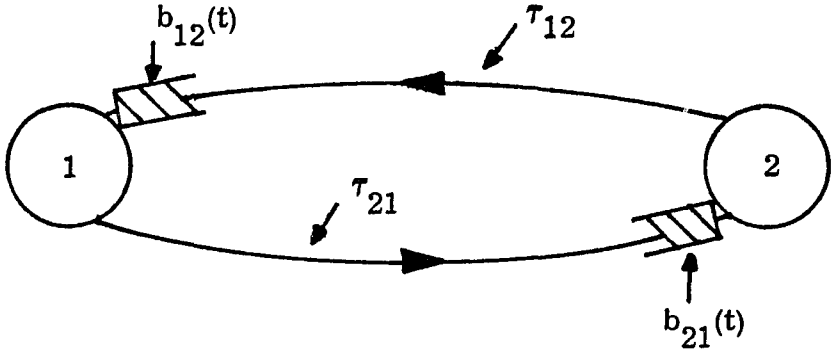

Fig. 3.3 A two node loop.

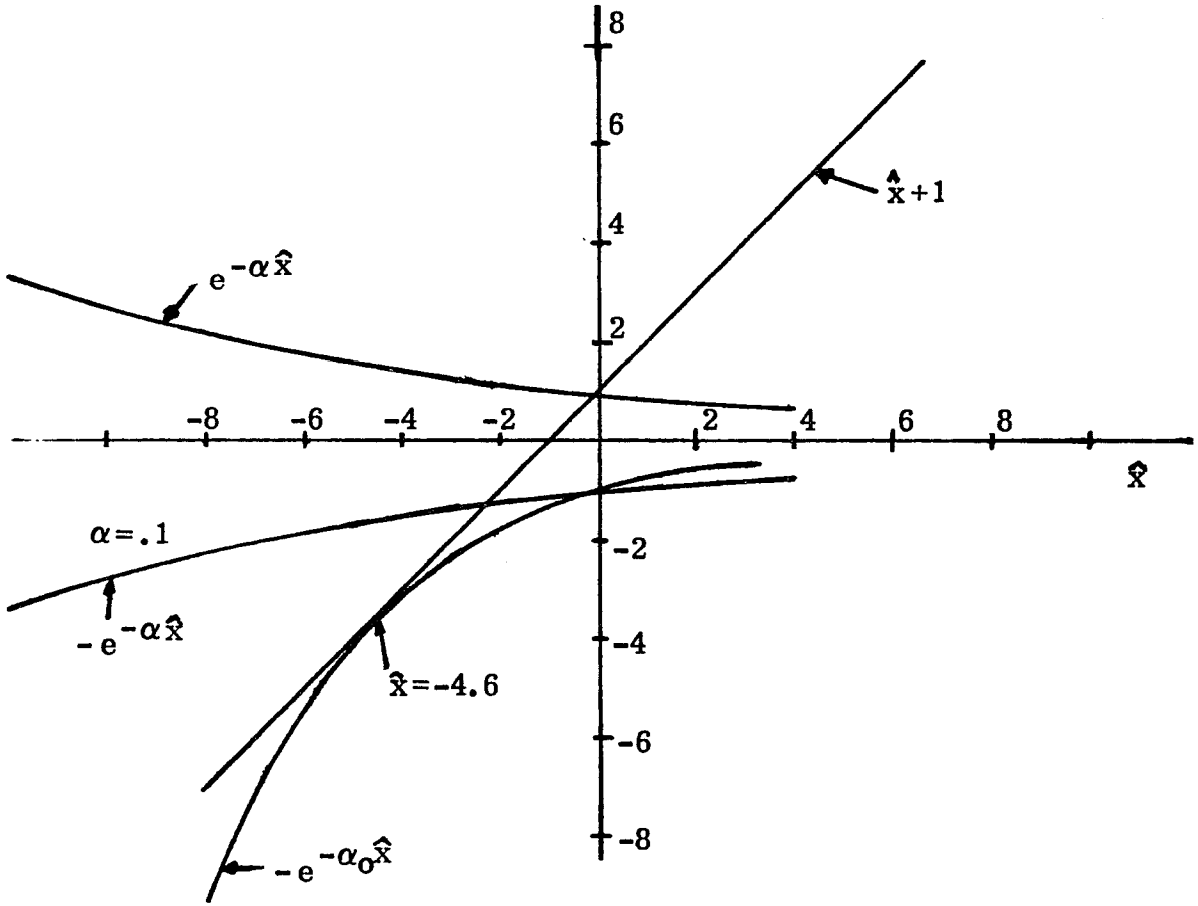

Fig. 3.4 Location of the real roots.

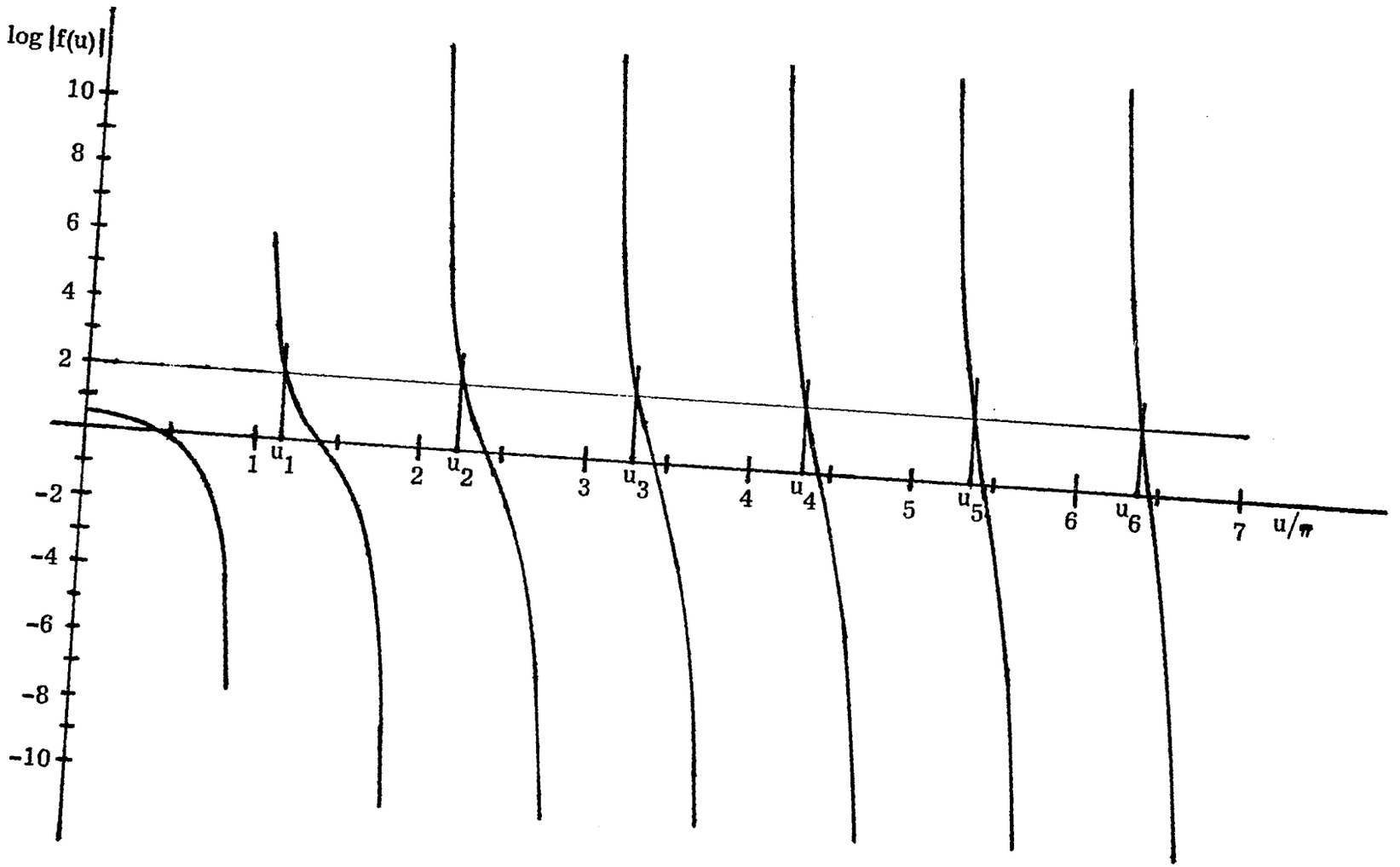

Fig. 3.5 Plot of $\log|f(u)|$.

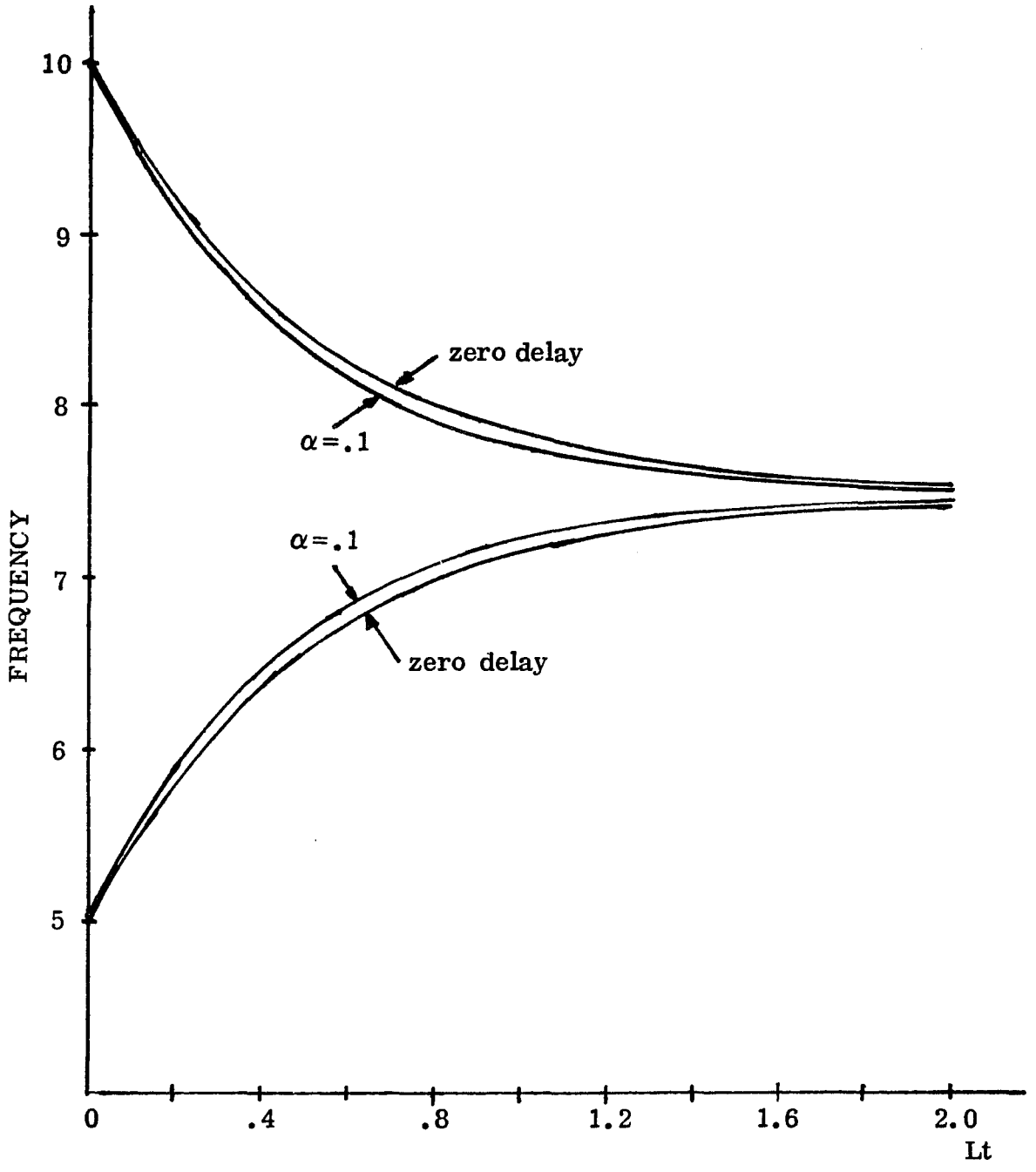

Fig. 3.6 Response of a two node loop for $\alpha = .1$ and for zero delay.

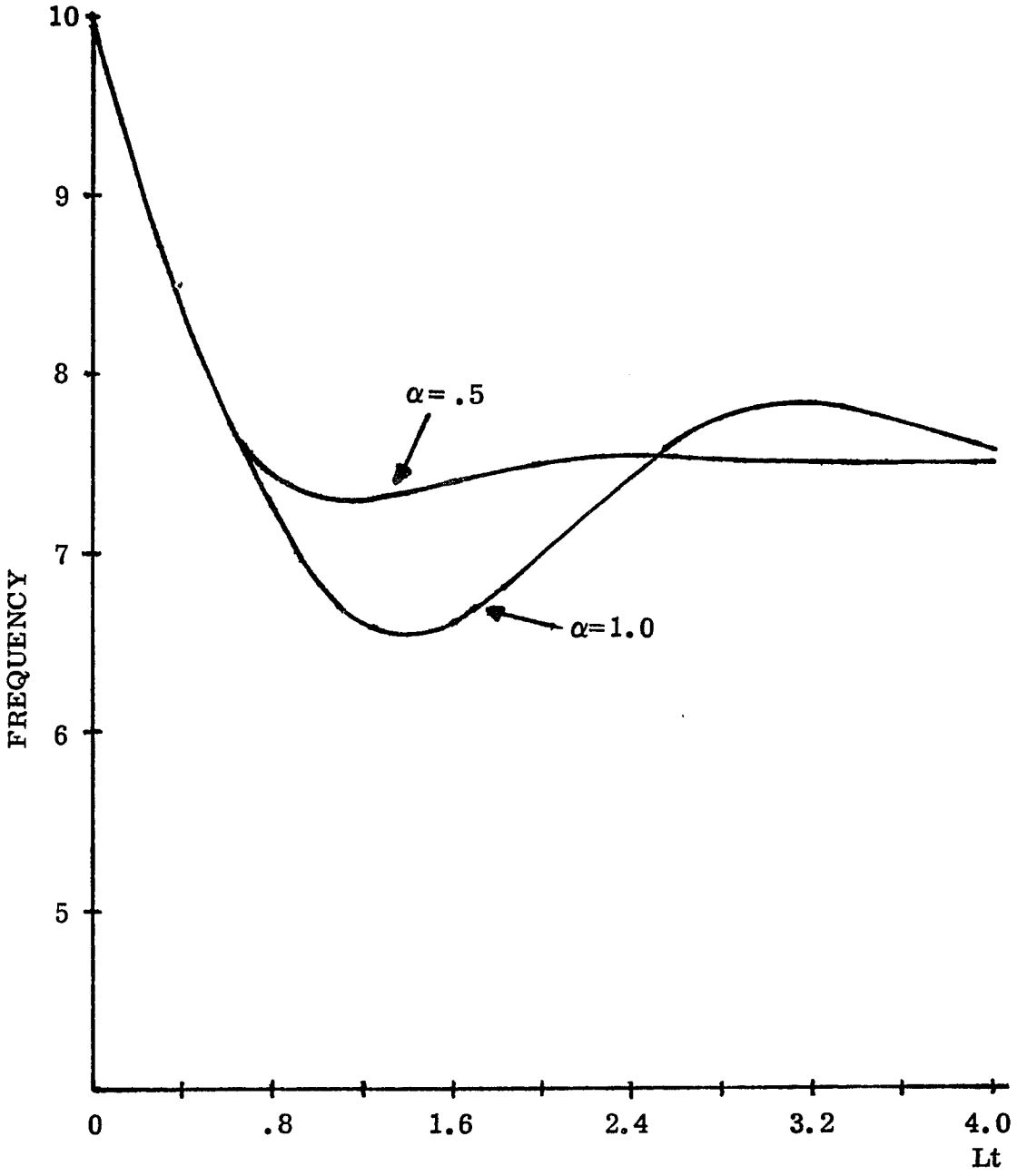

Fig. 3.7 Response of a two node loop when the delay is large.

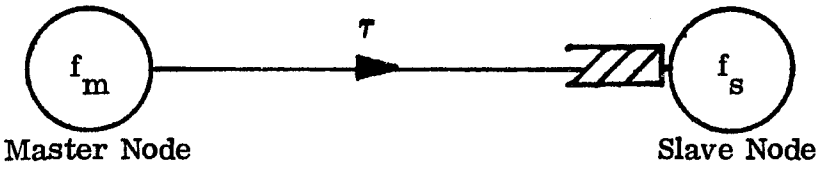

Fig. 3.8 A slave node synchronizing from its immediate master.

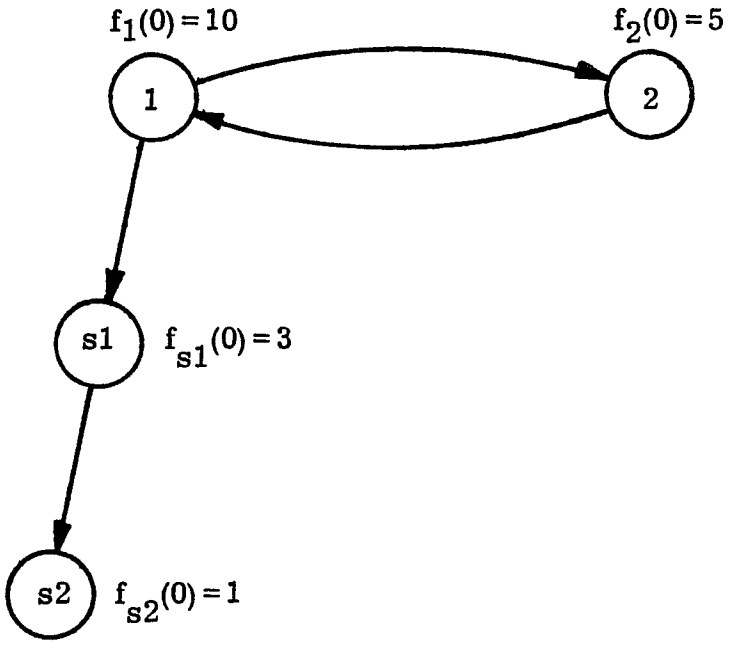

Fig. 3.9 A network with two levels of slave nodes.

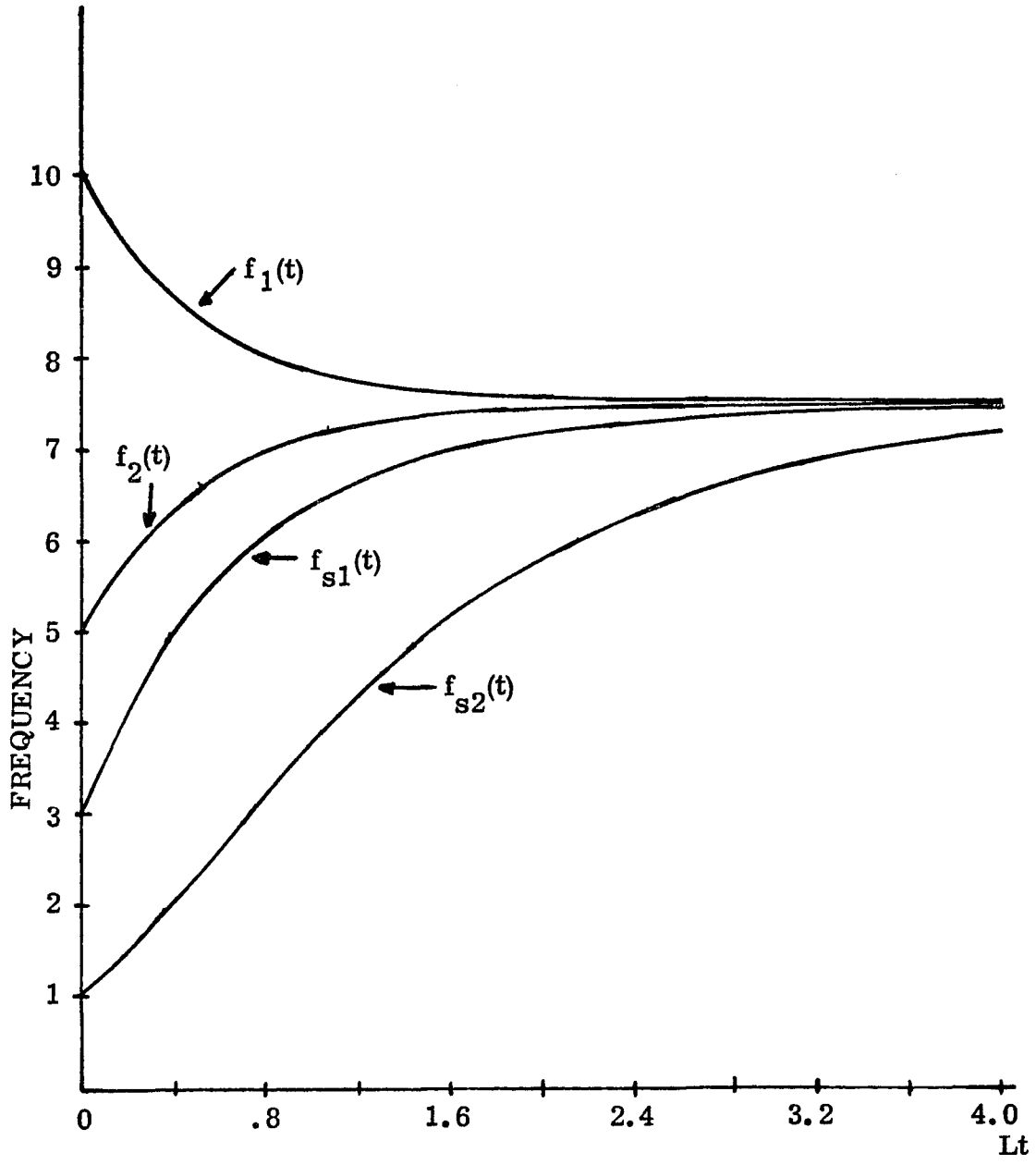

Fig. 3.10 Response of the slave nodes.

Chapter 4

SIMULATION OF PEAK AND MUTUAL SYNCHRONIZATION

4.1 INTRODUCTION

In our discussion of the dynamic response of Peak Synchronization we assumed that the structure of the synchronization subnetwork did not vary with time. For those time intervals during which this assumption holds, our calculations show the type of response to be expected. In the course of attaining synchronization, however, one buffer may overtake another, and this may cause the synchronization subnetwork to change. This subnetwork may also change due to oscillator drift or jitter. In this chapter we will use computer simulation to determine the dynamic response of Peak Synchronization. This will allow us to see how the dynamic response is affected by the changing synchronization subnetwork. We will also be able to verify some of the analytical results derived in the previous chapters. Finally, by also simulating Mutual Synchronization, we can compare the performance of the two systems.

We will consider two networks in this chapter. The first is a small five node network, and the second is a twelve node "dumbell" network. For the five node network we will find that Peak and Mutual Synchronization perform about equally well. They lead to about the same size buffers, and

noisy oscillators cause about the same amount of disturbance to the controlled frequencies and the buffer sizes. In the case of the dumbell network we know that Mutual Synchronization should perform badly. Specifically we expect that large buffer contents will result. We will find that the performance of Peak Synchronization with the dumbell network is not impaired, and that the buffer bounds derived in chapter 2 are satisfied. Though the dumbell network may seem unreasonable as a practical network, it demonstrates the vulnerability of Mutual Synchronization to network configuration.

In order to determine the effect of oscillator jitter we add to the uncontrolled oscillator frequencies computer generated low pass Gaussian noise. We then observe the resulting variations in both the controlled oscillator frequencies and the buffer sizes. We will also determine the effect of varying the correlation between successive samples of the Gaussian noise.

4.2 DESCRIPTION OF THE SIMULATION PROGRAM

The systems that we wish to simulate are governed by the following equations:

$$f_i(t) = f_{io} + N_i(t) + L_i \max_{j \in S_i} b_{ij}(t) \quad ; \text{ Peak Synch}$$

$$f_i(t) = f_{io} + N_i(t) + L_i \sum_{j=1}^{N} \hat{a}_{ij}\, b_{ij}(t) \quad ; \text{ Mutual Synch}$$

where $N_i(t)$ is the random jitter associated with node i, and $\{\hat{a}_{ij}\}$ is an averaging matrix that satisfies $\sum_j \hat{a}_{ij} = 1$.

For the purpose of the simulation the time axis is divided into intervals of lenth T. The value we have used for T is .01 seconds. The buffer contents are first initialized to zero and then computed recursively according to the following formula

$$\dot{b}_{ij}(kT) = f_j(kT - \tau_{ij}) - f_i(kT) \tag{4.1a}$$

$$b_{ij}((k+1)T) = T\,\dot{b}_{ij}(kT) + b_{ij}(kT) \tag{4.1b}$$

In order to simulate the time delays, we need to store both the current and past values of each oscillator frequency. For each node we form a circular array to store these frequency values, and a pointer is used to indicate the location of the current frequency. The location of any desired past frequency is computed relative to this pointer. The elements of each array are originally initialized to the free running frequency of the corresponding node. Clearly all delays must take on values that are integral multiples of T.

After computing the buffer contents at time (k+1)T using (4.1), we can then find the next frequency values as follows:

Peak Synch;

$$f_i((k+1)T) = f_{io} + N_i(k+1) + L_i \max_{j\epsilon S} \{b_{ij}[(k+1)T]\} \quad (4.2a)$$

Mutual Synch;

$$f_i[(k+1)T] = f_{io} + N_i(k+1) + L_i \sum_j \hat{a}_{ij}\, b_{ij}[(k+1)T] \quad (4.2b)$$

Our simulation therefore consists of repeatedly applying (4.1) and (4.2).

For the $N_i(k)$ we use a sequence of first order Markov Gaussian random variables. This corresponds to samples of RC filtered white Gaussian noise. The sequence $\{N_i(k)\}$ is generated recursively from the following relationships:

$$N_i(k) = \rho N_i(k-1) + \sqrt{1-\rho^2}\; X(k), \quad k > 1 \quad (4.3a)$$

$$N_i(0) = X(0) \quad (4.3b)$$

where $\{X(k)\}$ is a sequence of independent Gaussian random variables. A noise sequence generated in this fashion is stationary and has for its correlation function

$$R(n) = E[n(k)n(k+n)] = \rho^n \sigma_x^2 ,$$

with $\sigma_x^2 = E[X^2(k)]$. As mentioned previously $\{N(k)\}$ can be considered as samples of a continuous noise process $N_c(t)$, with $N(k) = N_c(kT)$. The process $N_c(t)$ corresponds to RC low pass filtered white Gaussian noise. The 3db cut off frequency of this filter is given by:

$$w_c = \frac{\ln 1/\rho}{T}$$

We will perform simulations for various values of w_c in order to see the effect of varying the frequency composition of the noise.

4.3 A FIVE NODE NETWORK

In Fig. 4.1a we illustrate the network to be considered in this section. Also shown are the free running oscillator frequencies and the synchronization subnetwork that is formed initially. This set of free running frequencies was chosen specifically so that the synchronization subnetwork would begin with two separate parts.

Fig. 4.2 illustrates the response of this network with no oscillator noise, L=.5, and τ =.01 seconds. We are using the same values for gain and delay throughout the network and therefore we have dropped the subscripts. This example illustrates how a synchronization subnetwork that starts out as two separate parts eventually becomes connected. We will now describe this process in detail. At the start of synchronization nodes 1 and 4 form a loop and begin to move exponentially towards a frequency of 60. Nodes 3 and 5 also form a loop and start to move towards 45. Node 2 is slaved to node 1 and therefore also heads towards 60. At time .6 seconds node 4 overtakes node 5 in frequency. This does not have an immediate effect, however it should be

clear that the buffer at 3 from 4 must eventually exceed the buffer at 3 from 5. This happens at 1.2 seconds and at that time node 3 starts to synchronize from node 4. A similar change occurs at time 2.4 seconds when node 5 also begins to synchronize from node 4. The final form of the synchronization subnetwork, which clearly must be connected, is illustrated in Fig. 4.1b.

We can describe the response shown in Fig. 4.2 as piecewise exponential, with the "breaks" occurring at the times when the synchronization subnetwork changes configuration. The largest buffers that developed during this simulation were b_{21}=100 and b_{12}=-100.2. These values are well within the bounds specified in chapter 2.

We next consider the effect of adding noise to the free running oscillator frequencies. Figures 4.3 and 4.4 show a section of the resulting response. In both figures the noise has a standard deviation of σ_N= 5, however in Fig. 4.3 the low pass filter cut off frequency is w_c = 1 rad/sec, whereas in Fig. 4.4 it is w_c = .1 rad/sec. Thus in Fig. 4.4 the noise is more predominately low frequency. The difference between the low frequency and high frequency noise is readily apparent from these figures. The high frequency noise causes the controlled oscillator frequencies to deviate more widely than the low frequency noise. The important facts concerning these two simulations and also

the no noise case are summarized in Table 4.1.

TABLE 4.1

SIMULATION RESULTS FOR PEAK SYNCHRONIZATION
FIVE NODE NETWORK

Type of simulation	τ	σ_N	w_c	Maximum freq. dev.	Maximum buffer size
No noise	.01	0	-	-	-100.2,100
Noise	.01	5	1	46-74 $=5.6\sigma$	-118.5,+118
Noise	.01	5	.1	49-70 $=4.2\sigma$	-125,+124

The maximum quantities listed are those observed over a simulation time of 1,000 seconds. Notice that the low frequency noise, although causing less frequency variation, causes more variation in the buffer contents.

For comparision, we now consider the performance of Mutual Synchronization using the same five node network. Fig. 4.5 shows the response for noise free oscillators. The largest buffers that develop in this case are ±114 bits. This compares to ±100 bits in the Peak Synchronization case. Comparing Fig. 4.5 to Fig. 4.2, we see that the response time for Peak and Mutual Synchronization are about the same. Table 4.2 summarizes the results obtained for this noise free simulation and two simulations with noisy oscillators.

TABLE 4.2

SIMULATION RESULTS FOR MUTUAL SYNCHRONIZATION
FIVE NODE NETWORK

Type of simulation	τ	σ_N	w_c	Maximum freq. dev.	Maximum buffer size
No noise	.01	0	-	-	-114, +114
Noise	.01	5	1	26-53 $=5.4\sigma$	-131, +131
Noise	.01	5	.1	31-49 $=3.6\sigma$	-137, +137

Again, as in Peak Synchronization, the lower frequency oscillator noise (w_c = .1) causes larger maximum buffer sizes, and smaller frequency deviations. On the whole we can say, that both systems perform about equally well for this small five node network. Peak Synchronization has a somewhat smaller maximum buffer size, but this difference is not very substantial. In the next section we will consider a twelve node dumbell network and again compare the performance of both systems.

4.4 A TWELVE NODE DUMBELL NETWORK

The dumbell network is known to cause Mutual Synchronization to generate very large buffer contents. Peak Synchronization, however, was shown in chapter 2 to have buffer bounds that are independent of network

configuration. In this section we will simulate the performance of both Peak and Mutual Synchronization on a twelve node dumbell network. This will serve to verify further the buffer bounds expected in Peak Synchronization.

The network to be used is illustrated in Fig. 4.6. In this network each node on the left hand side has an uncontrolled frequency of 25, whereas the nodes on the right hand side have an uncontrolled frequency of 50. The response of this network using Peak Synchronization and no oscillator noise is shown in Fig. 4.7. The time needed to achieve synchronization is about 11 seconds which is very close to the synchronization time of the five node network. The largest buffers that developed were -55 and +50. In this simulation we have used L = .5, and τ = .1. Since ΔF=25 it is clear that the above range of buffer values exactly satisfy the bounds of chapter 2, namely:

$$-\Delta F(\frac{1}{L} + 2\bar{\tau}) < b_{ij}(t) < \frac{\Delta F}{L}$$

The simulation results for Peak Synchronization including several cases of oscillator noise are summarized in Table 4.3. Once again we notice that the lower frequency noise (w_c =.1) causes less frequency deviation, and larger maximum buffers. Notice also that the buffer sizes here are smaller than for the five node network. That is because this network has a smaller value for ΔF. Thus, as expected, we find that the buffer sizes depend only on ΔF and the

gains (L_i), and not on the network size.

TABLE 4.3

SIMULATION RESULTS FOR PEAK SYNCHRONIZATION
TWELVE NODE NETWORK

Type of simulation	τ	σ_N	w_c	Maximum freq. dev.	Maximum buffer size
No Noise	.1	0	-	-	-55, +50
Noise	.1	5	1	(38,69) $=6.2\sigma$	-83, +77
Noise	.1	5	.1	(42,65) $=4.6\sigma$	-93, +87

Again for comparison we simulate the performance of Mutual Synchronization on the 12 node dumbell network, using the same values for uncontrolled frequencies, delay and gain. The resulting response for noise free oscillators is illustrated in Fig. 4.8. We see from this figure that the time needed to achieve synchronization is approximately 170 seconds. The largest buffers that developed during this simulation were ±802 bits. Both of these quantities greatly exceed the corresponding amounts that occurred using Peak Synchronization. Thus we have found, as expected, that the performance of Mutual Synchronization greatly deteriorates with large dumbell networks.

The simulation results for Mutual Synchronization with

the 12 node dumbell network are summarized in Table 4.4 below.

TABLE 4.4

SIMULATION RESULTS FOR MUTUAL SYNCHRONIZATION
TWELVE NODE NETWORK

Type of simulation	τ	σ_N	w_c	Maximum freq. dev.	Maximum buffer size
No Noise	.1	0	-	-	-802, +802
Noise	.1	5	1	(23,53) $=6\sigma$	-844, +844
Noise	.1	5	.1	(30,46) $=3.2\sigma$	-869, +869

In Table 4.4 we see again the same pattern that occurred in all the previous cases. The higher frequency noise causes smaller maximum buffers and larger frequency deviations.

4.5 CONCLUSION

We have studied by computer simulation the performance of Peak and Mutual Synchronization, and many of the expected properties of these two systems have been verified. For a small five node network with a fairly uniform structure, it was shown that both systems perform about equally. However, when a twelve node dumbell network was simulated we saw a very marked deterioration in the performance of Mutual

Synchronization. For both networks we were able to verify that Peak Synchronization satisfies the buffer bounds of chapter 2.

The maximum deviations in the controlled oscillator frequencies were found to be about 6 times the standard deviation of the noise. This is approximately equal to what we would expect for the maximum variation of the uncontrolled frequencies. Thus we verify that the random variation in the controlled frequencies is no more than that of the uncontrolled frequencies. Further, it was shown that when the noise is low frequency the variations in the controlled oscillator frequencies are reduced.

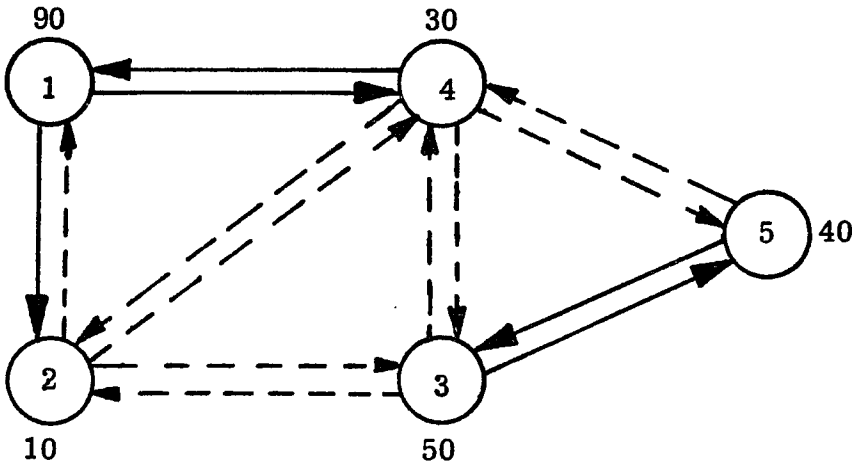

Fig. 4.1a A five node network used for simulation. Solid lines show the initial (unconnected) synchronization subnetwork. The uncontrolled frequencies are shown for each node.

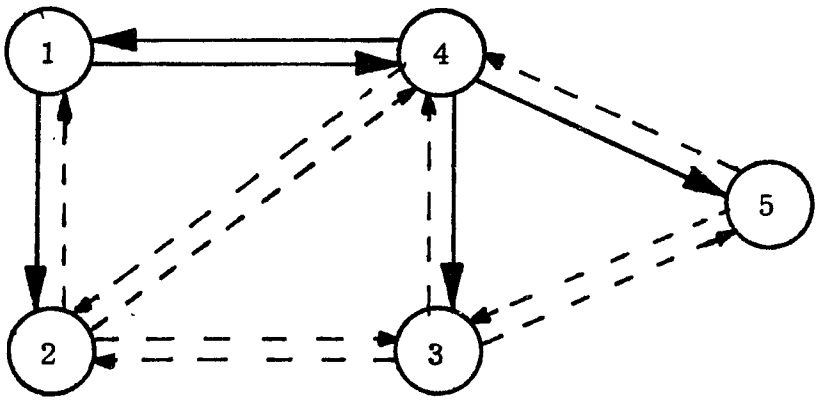

Fig. 4.1b The final form of the synchronization subnetwork is shown here by the solid lines.

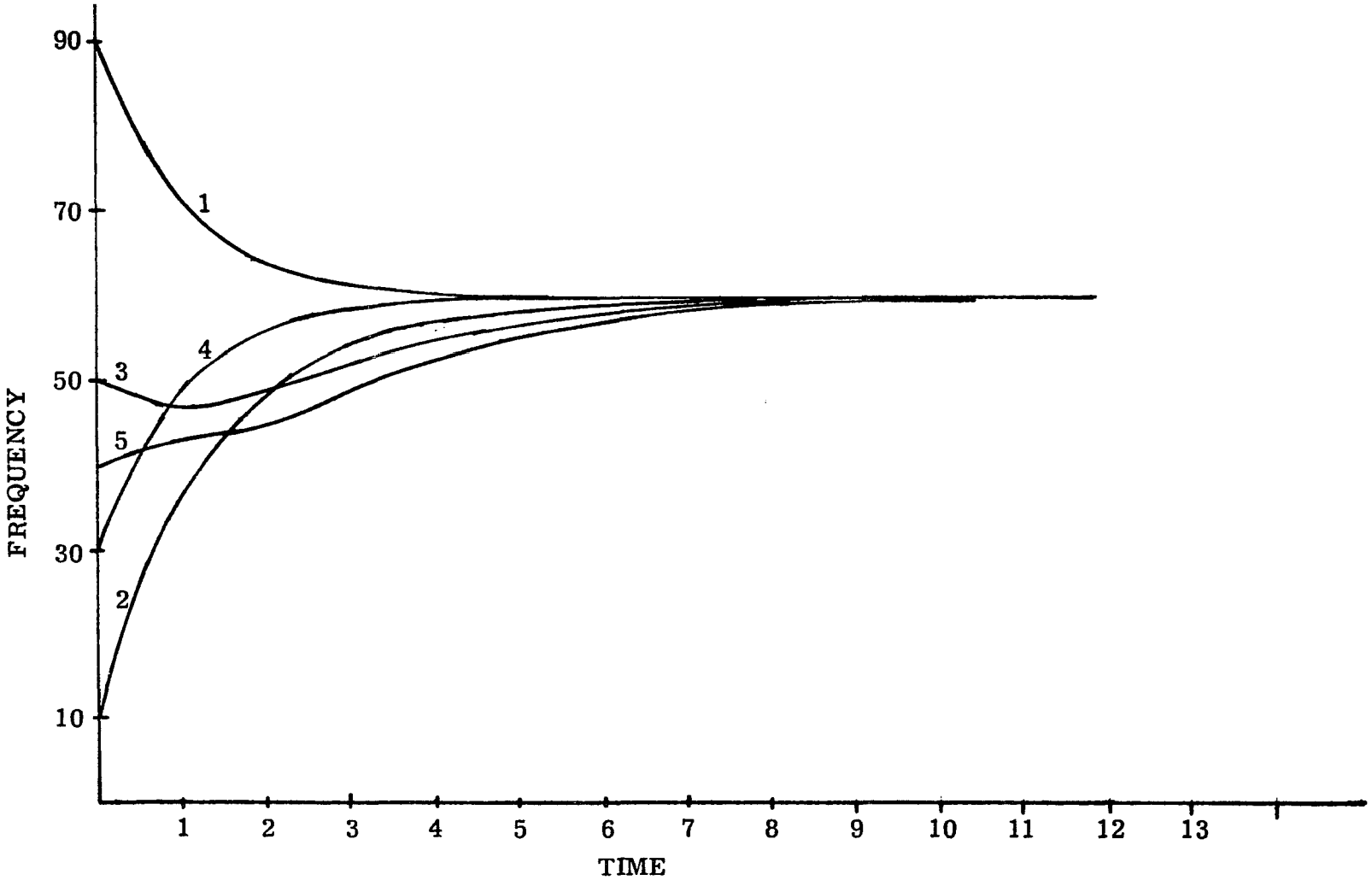

Fig. 4.2 Response of the 5 node network using Peak Synchronization and no oscillator noise.

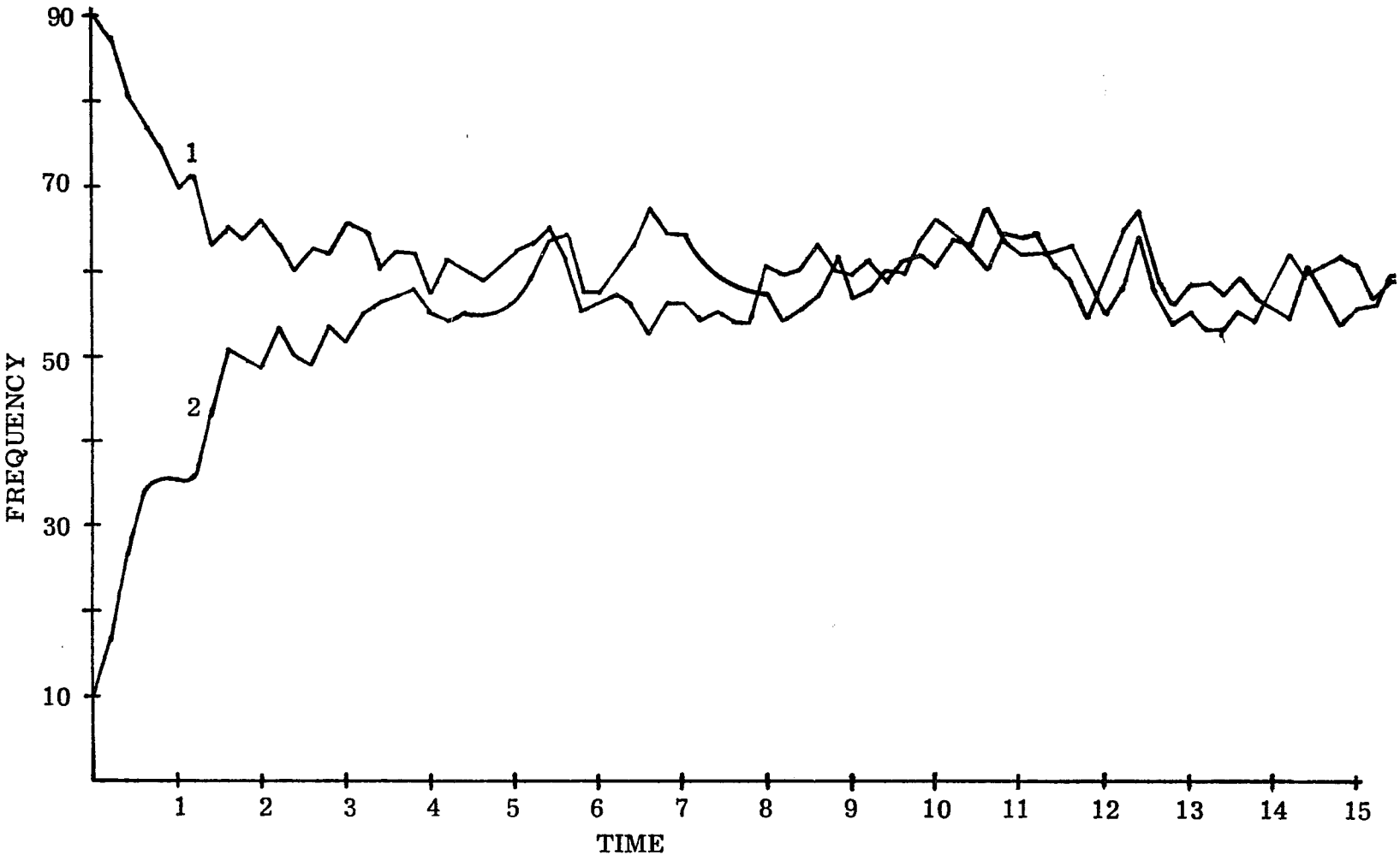

Fig. 4.3 Response of the five node network using Peak Synchronization. $\sigma_N = 5$, $w_c = 1$.

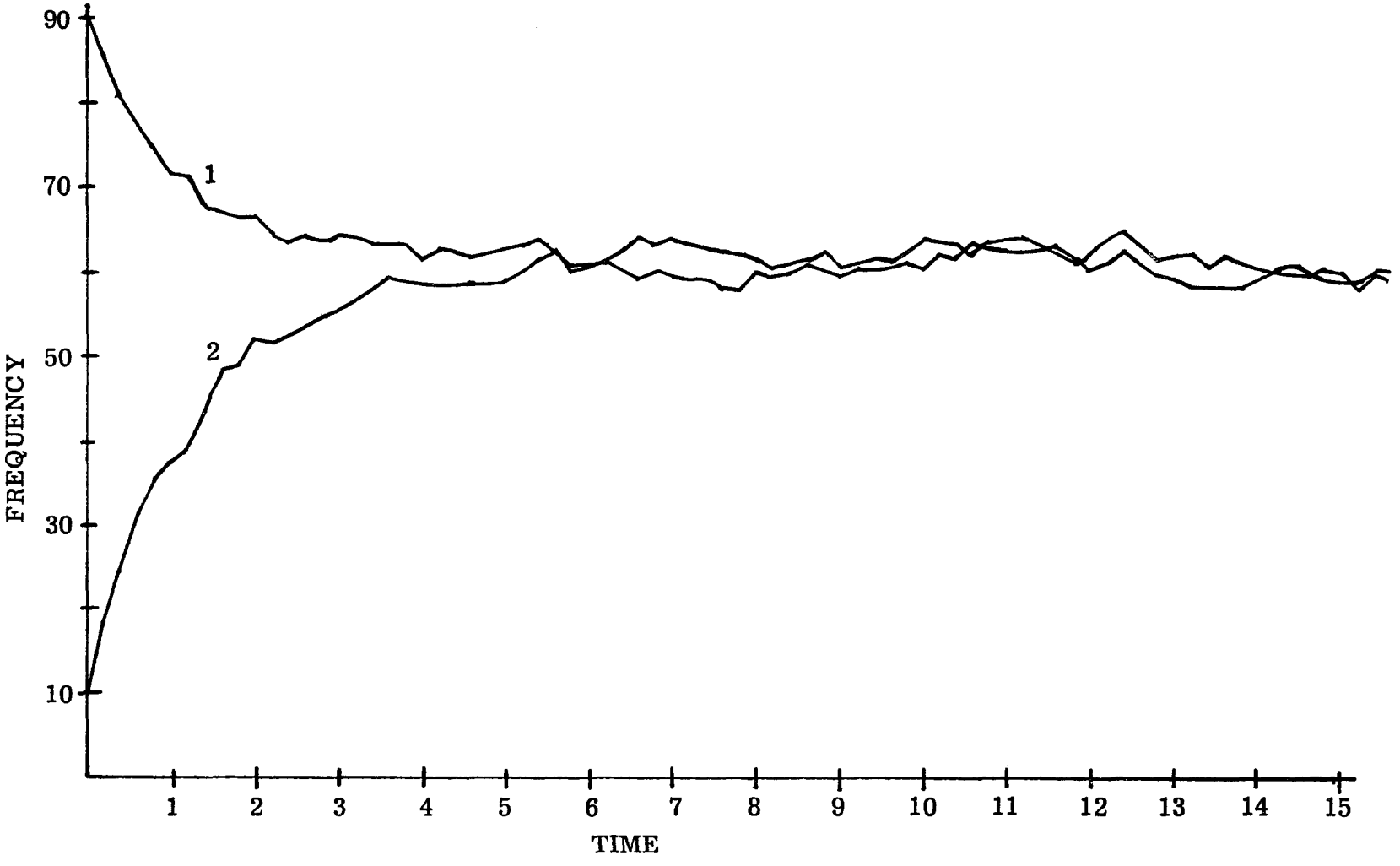

Fig. 4.4 Response of the five node network using Peak Synchronization. $\sigma_N=5$, $w_c=.1$.

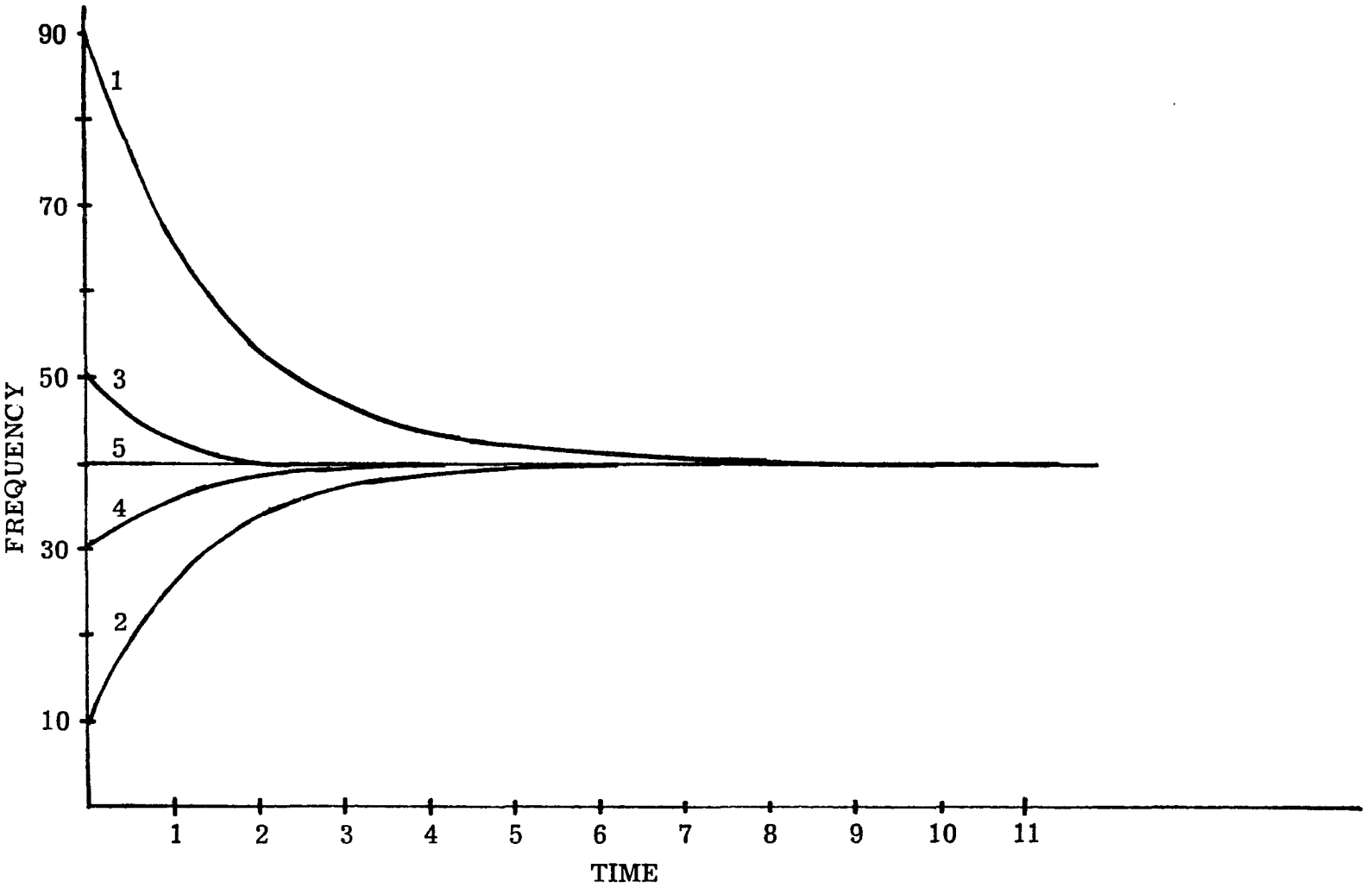

Fig. 4.5 Response of the five node network using Mutual Synchronization. No oscillator noise.

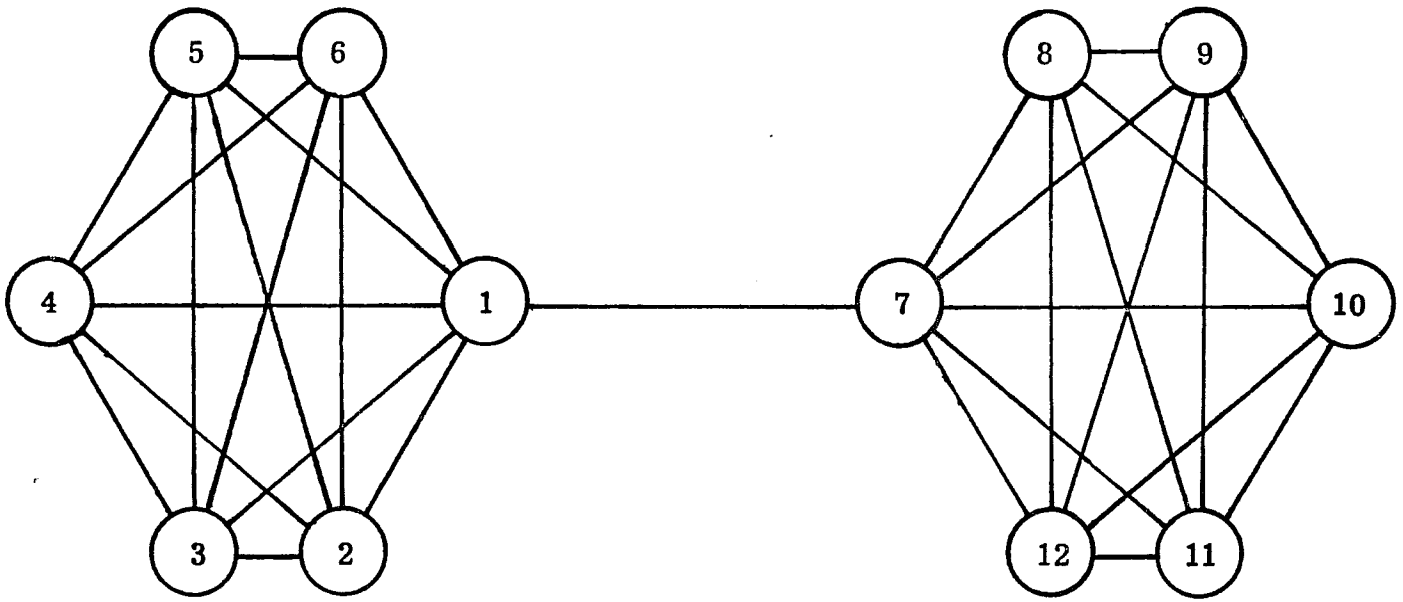

Fig. 4.6 A twelve node dumbell network.

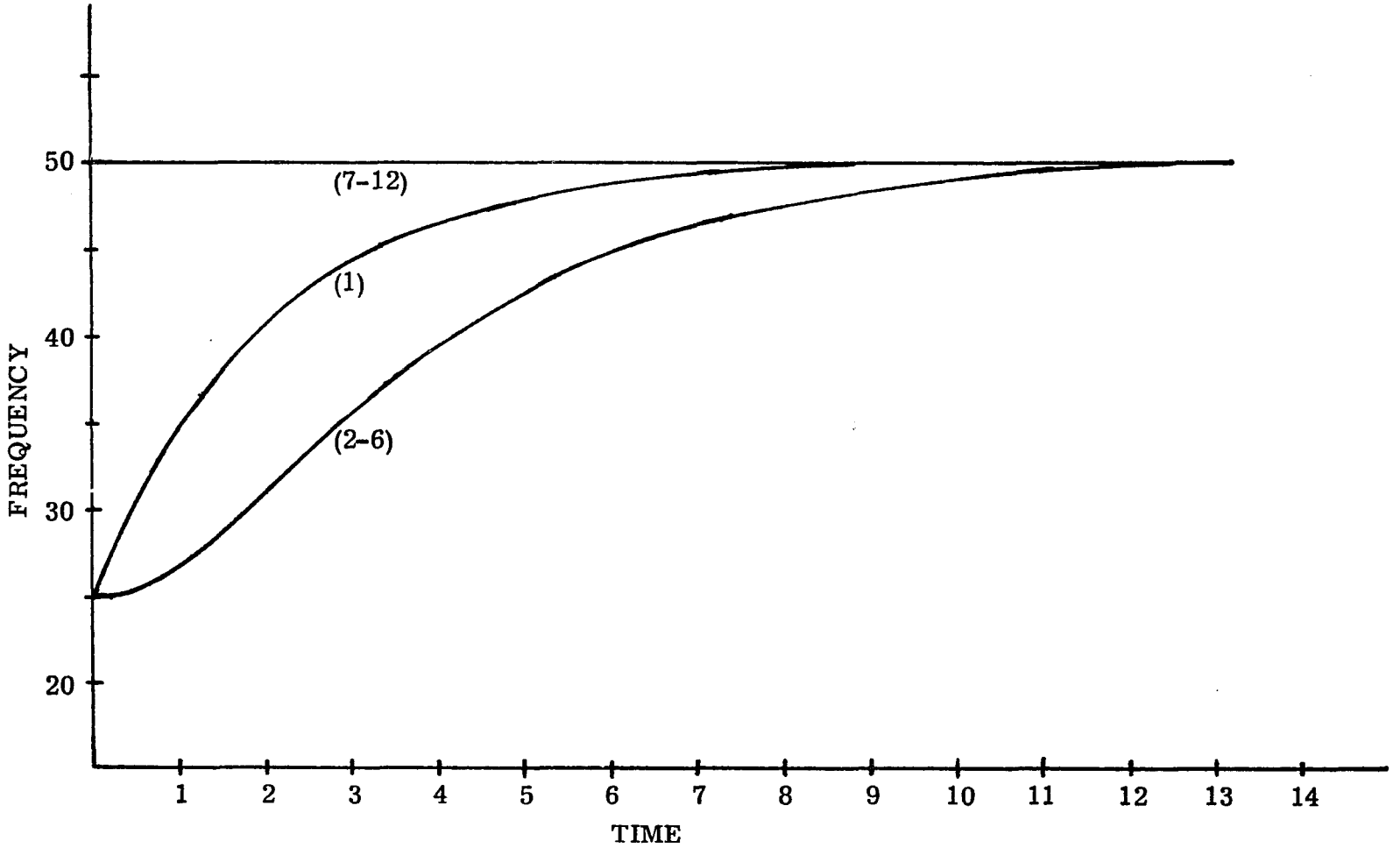

Fig. 4.7 Response of the twelve node dumbell network using Peak Synchronization.

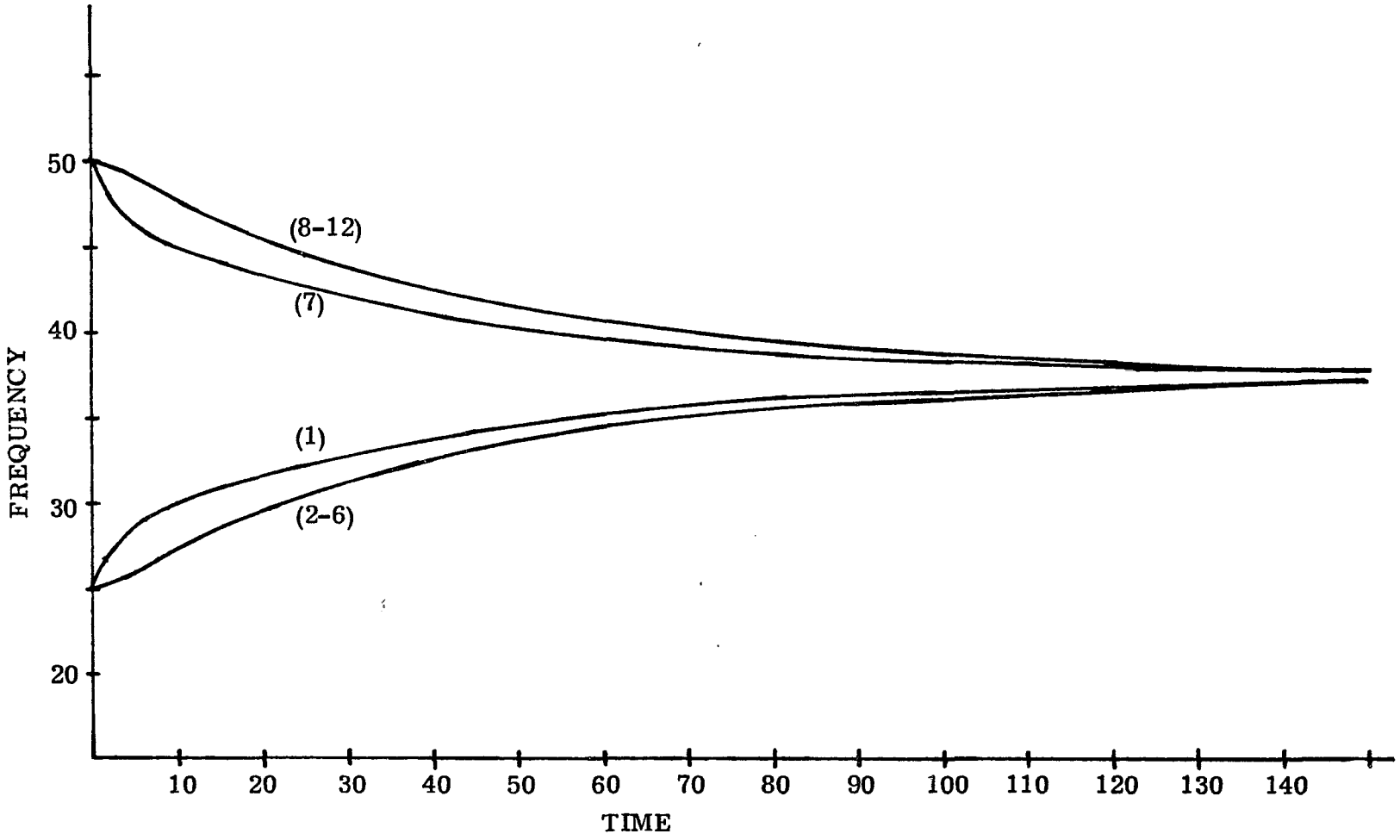

Fig. 4.8 Response of the twelve node dumbell network using Mutual Synchronization.

Chapter 5

CORRELATION OF A CONVOLUTIONAL ENCODER

5.1 INTRODUCTION

When a random binary data sequence is used to generate a digital signal consisting of a random sequence of pulses, the spectral density of the digital signal can be found from the correlation properties of the original data sequence[28]. The spectral density is an important parameter in system design as it allows one to determine the bandwidth necessary for transmission, and to estimate the interference that will result to other signals. In addition, the correlation of the original data sequence provides useful information for the design of bit synchronizers. High correlation indicates a tendency for long strings of identical bits, thereby hindering the performance of a bit synchronizer, whereas low correlation indicates a greater tendency for the occurrence of transitions.

In this study we consider the output data sequence produced by a binary convolutional encoder, and present a method for calculating the correlation of this output. We treat first the case of random statistically independent inputs, and then consider inputs generated by a first order Markov source.

A binary convolutional encoder consists of a K stage shift register and v modulo-2 adders. For a rate 1/v encoder the data is shifted in one bit at a time. More generally for a rate λ/v encoder the data is shifted in λ bits at a time. An example with K=4 and v=2 is shown in Fig. 5.1.

5.2 CORRELATION OF THE OUTPUT WHEN $\lambda = 1$

The output of the encoder for the case $\lambda = 1$ can be written as:

$$y_{ij} = \sum_{\ell=0}^{k-1} x_{i-\ell} g_{\ell j} \qquad 1 \leq j \leq v \tag{5.1}$$

where y_{ij} is the output of the jth mod-2 adder just after the ith input bit (x_i) has entered the shift register, and $g_{\ell j}=1$ if the ℓth stage of the shift register is connected to the jth mod-2 adder, and 0 otherwise.

We assume here that the input bits are statistically independent and take on the values 0 or 1 with probability 1/2. In this case an output bit y_{ij} is also equally likely to be 0 or 1. To see that such is the case, note that y_{ij} is the mod-2 sum of N input variables, where N equals the number of non-zero terms in the vector $\underline{g}_j = (g_{0j}, g_{1j}, \ldots, g_{K-1,j})$. Therefore we can write

$$y_{ij} = x_{a_1} \oplus x_{a_2} \oplus \ldots \oplus x_{a_N} \tag{5.2}$$

where x_{a_i}, $1 \leq i \leq N$, denotes the ith term of the sum in (5.1) that has a non-zero coefficient. The result is obvious for N=1, therefore by induction, set

$$z = x_{a_1} \oplus \cdots \oplus x_{a_{N-1}}.$$

Then

$$y_{ij} = z \oplus x_{a_N}$$

$$\Pr(y_{ij}=1) = \Pr(z=1, x_{a_N}=0) + \Pr(z=0, x_{a_N}=1).$$

Therefore

$$\Pr(y_{ij}=1) = 1/2$$

since z and x_{a_N} are independent.

To find the correlation of two output variables y_{ij} and y_{mn}, let us assume for convenience that the ouput symbols are (1,-1). That is we consider the output values (1,0) to be transformed into (1,-1) according to the rules $1 \to 1$, $0 \to -1$. Then the correlation coefficient ρ is given by:

$$\rho = E(y_{ij} \cdot y_{mn})$$

which can be written as

$$\rho = 1 - 2[\Pr(1,-1) + \Pr(-1,1)]$$

or equivalently

$$\rho = 1 - 2 \cdot \Pr[(y_{ij} \oplus y_{mn}) = 1]. \qquad (5.3)$$

Consider two arbitrary output variables y_{ij} and $y_{i+n,k}$.

Then from (5.1) we have

$$y_{ij} \oplus y_{i+n,k} = \sum_{\ell=0}^{K-1} x_{i+n-\ell}\, g_{\ell k} \oplus x_{i-\ell} g_{\ell j}$$

$$= \sum_{\ell=-n}^{K-n-1} x_{i-\ell} g_{\ell+n,k} \oplus \sum_{\ell=0}^{K-1} x_{i-\ell} g_{\ell j}. \qquad (5.4)$$

If $n \geq K$, then all the input variables appearing above are distinct, and since (5.4) has the same form as (5.2), we have

$$\Pr\{(y_{ij} \oplus y_{i+n,k}) = 1\} = 1/2,$$

substituting into (5.3) yields $\rho = 0$. For $n < K$, (5.4) can be written as

$$y_{ij} \oplus y_{i+n,k} = \sum_{\ell=-n}^{-1} x_{i-\ell} g_{\ell+n,k} \oplus \sum_{\ell=0}^{K-n-1} x_{i-\ell}(g_{\ell+n,k} \oplus g_{\ell j})$$

$$\oplus \sum_{\ell=K-n}^{K-1} x_{i-\ell} g_{\ell j}. \qquad (5.5)$$

In this case unless all the coefficients of the input variables in (5.5) are zero, we again have a situation as in (5.2), and as before $\rho = 0$. On the other hand, for all the coefficients to vanish we must have

$$g_{\ell k} = 0 \qquad 0 \leq \ell \leq n-1 \qquad (5.6a)$$

$$g_{\ell j} = 0 \qquad k-n \leq \ell \leq K-1 \qquad (5.6b)$$

$$g_{\ell+n,k} = g_{\ell j} \qquad 0 \leq \ell \leq K-n-1 \qquad (5.6c)$$

in which case (5.5) becomes

$$y_{ij} \oplus y_{i+n,k} = 0$$

which implies $\rho=1$. Note that for n=0, (5.6) reduces to

$$g_{\ell k} = g_{\ell j} \qquad 0 \leq \ell \leq K-1$$

We see from the above discussion that two output bits from a convolutional encoder, whose input is a statistically independent sequence, can only be completely uncorrelated ($\rho=0$) or perfectly correlated ($\rho=1$), with no intermediate values possible. In addition, the output bits y_{ij} and $y_{i+n,k}$ are correlated with $\rho=1$, only if $\underline{g}_j$ and $\underline{g}_k$ are a shifted version of one another in the sense of (5.6). Two output bits occurring at the same time, i.e. n=0, are correlated only if the vectors $\underline{g}_j$ and $\underline{g}_k$ are identical.

In examining the optimum convolutional codes found by Odenwalder[29], we find only one code that satisfies the conditions indicated in (5.6). This code has K=3, v=3 and is defined by

$$\underline{g}_1 = 1\ 1\ 1$$

$$\underline{g}_2 = 1\ 1\ 1$$

$$\underline{g}_3 = 1\ 0\ 1$$

In this case $\underline{g}_1 = \underline{g}_2$ and the outputs of $\underline{g}_1$ and $\underline{g}_2$ are correlated. Since only one of the optimum codes has

correlated outputs, we are justified in making the statement that for almost all good codes the output is uncorrelated for statistically independent inputs.

5.3 CORRELATION WHEN $\lambda > 1$.

For the more general case of a λ/v rate encoder, (5.1) becomes

$$y_{ij} = \sum_{\ell=0}^{K-1} x_{\lambda i-\ell}\, g_{\ell j} \tag{5.7}$$

from which we get

$$y_{ij} \oplus y_{i+n,k} = \sum_{\ell=0}^{K-1} x_{\lambda (i+n)-\ell} g_{\ell k} \oplus x_{\lambda i-\ell} g_{\ell j}. \tag{5.8}$$

Following the same argument as before it is readily shown that for $\lambda n \geq K$, $\rho = 0$. For $\lambda n < K$, $\rho = 0$ unless

$$g_{\ell k} = 0 \qquad 0 \leq \ell \leq \lambda n-1 \tag{5.9a}$$

$$g_{\ell j} = 0 \qquad K-\lambda n \leq \ell \leq K-1 \tag{5.9b}$$

$$g_{\ell+\lambda n,k} = g_{\ell j} \qquad 0 \leq \ell \leq K-\lambda n-1 \tag{5.9c}$$

in which case $\rho = 1$.

Note that condition (5.9c) implies that $\underline{g}_k$ and $\underline{g}_j$ must be shifted by a multiple of λ with respect to one another. Paaske[30] has found optimum convolutional codes for code rates 2/3 and 3/4. The examination of these codes shows that none of them satisfy the conditions in (5.9).

Therefore these optimum codes always produce uncorrelated outputs for random statistically independent inputs.

5.4 CORRELATION OF THE OUTPUT WITH MARKOV INPUTS

In this section we consider the correlation of the output bits when the input sequence is generated by a stationary first order Markov source. We treat only the case $\lambda=1$.

Define the state of the encoder at time i as the binary number formed by the contents of the shift register, that is:

$$Z_i = x_{i-K+1} \cdots x_{i-1}\, x_i \tag{5.10}$$

then

$$Z_i \in \{ 0, 1, \ldots, R-1 \} \quad \text{with } R = 2^K$$

Using this notation we have an input sequence $x_1x_2x_3\cdots$, generating a sequence of states $Z_1Z_2Z_3\cdots$, and a sequence of output vectors $\underline{y}_1\underline{y}_2\underline{y}_3\cdots$, where $\underline{y}_i = y_{i1}, y_{i2}, \ldots, y_{iv}$. The input sequence which is assumed to be first order Markov is characterized by its transition matrix

$$\begin{pmatrix} r_{00} & r_{01} \\ r_{10} & r_{11} \end{pmatrix}$$

where

$$r_{ab} = \Pr(x_i=b/x_{i-1}=a). \tag{5.11a}$$

The steady state probabilities are given by

$$r_0 = Pr(x_i=0) = \frac{r_{10}}{r_{10}+ r_{01}} \tag{5.11b}$$

and

$$r_1 = Pr(x_i=1) = \frac{r_{01}}{r_{10}+ r_{01}} \tag{5.11c}$$

For a symmetric source, $r_0= r_1= 1/2$ and $r_{10}=r_{01}$, $r_{00}=r_{11}$.

The sequence of states $\{Z_i\}$ resulting from a first order Markov input is also first order Markov. The transition matrix of this sequence is

$$\mathbb{T} = \{p_{jk}\}$$

where $p_{jk} = Pr(Z_i=k/Z_{i-1}=j)$ and $0 \leq j,k \leq R-1$. Since $Z_i = x_i+ (2Z_{i-1})$ mod-R, we have

$$p_{jk} = \begin{cases} r_{00} & \text{if } k=2j \text{ mod-R and } j=0 \text{ mod } 2 \quad (5.12a) \\ r_{10} & \text{if } k=2j \text{ mod-R and } j=1 \text{ mod } 2 \quad (5.12b) \\ r_{01} & \text{if } k=2j+1 \text{ mod-R and } j=0 \text{ mod } 2 \quad (5.12c) \\ r_{11} & \text{if } k=2j+1 \text{ mod-R and } j=1 \text{ mod } 2 \quad (5.12d) \\ 0 & \text{otherwise} \quad (5.12e) \end{cases}$$

Let $w_j^i= Pr(Z_i=j)$, that is w_j^i is the probability of being in state j at time i. Define the state distribution vector at time i as

$$W^i = (w_0^i \; w_1^i \; \cdots \; w_{R-1}^i).$$

Then in matrix form we can write

$$W^i = W^{i-1} \Pi$$

$$W^n = W^0 \Pi^n$$

Let p_{jk}^n denote an element of Π^n, that is $p_{jk}^n = Pr(Z_{i+n}=k/Z_i=j)$.

The steady state probability of a state α, will be denoted w_α. w_α can be calculated from the known input statistics as follows: Let the binary representation of α be

$$\alpha = \alpha_0 \; \alpha_1 \; \cdots \; \alpha_{K-1}$$

then

$$w_\alpha = Pr(x_i=\alpha_{K-1}/x_{i-1}=\alpha_{K-2}) \cdot Pr(x_{i-1}=\alpha_{K-2}/x_{i-2}=\alpha_{K-3}) \cdots$$

$$Pr(x_{i-K+2}=\alpha_1/x_{i-K+1}=\alpha_0) \cdot Pr(x_{i-K+1}=\alpha_0)$$

or

$$w_\alpha = r_{\alpha_0} \prod_{j=0}^{K-2} r_{\alpha_j,\alpha_{j+1}} \tag{5.13}$$

where the quantities r_{α_0}, and $r_{\alpha_j,\alpha_{j+1}}$ are defined in (5.11).

Let $A_j = \{k \mid Z_i = k \rightarrow y_{ij} = 1\}$ $j=1,2,\ldots,v$. Then A_j is the set of states for which the output of the jth mod-2 adder equals 1. Again let y_{ij} and $y_{i+n,k}$ represent two output variables, then we can write

$$\Pr(y_{i+n,k}=1, y_{ij}=1) = \Pr(Z_{i+n} \in A_k, Z_i \in A_j)$$

$$= \sum_{\beta \in A_k} \sum_{\alpha \in A_j} \Pr(Z_{i+n}=\beta, Z_i=\alpha)$$

$$= \sum_{\beta \in A_k} \sum_{\alpha \in A_j} \Pr(Z_{i+n}=\beta / Z_i=\alpha) \cdot \Pr(Z_i=\alpha)$$

$$= \sum_{\beta \in A_k} \sum_{\alpha \in A_j} p^n_{\alpha\beta} \; w_\alpha \qquad (5.14)$$

where we assume that the system at time i is in steady state. Likewise we have

$$\Pr(y_{i+n,k}=-1, y_{ij}=-1) = \sum_{\beta \in \bar{A}_k} \sum_{\alpha \in \bar{A}_j} p^n_{\alpha\beta} \; w_\alpha \qquad (5.15)$$

where $\bar{A}_k$ is the complement of A_k. In terms of the above quantities we have

$$E(y_{ij} \cdot y_{i+n,k}) = 2[\Pr(y_{ij}=1, y_{i+n,k}=1) + \Pr(y_{ij}=-1, y_{i+n,k}=-1)] - 1. \qquad (5.16)$$

Using (5.14), (5.15) and the relations[31]

$$\sum_\alpha p^n_{\alpha\beta} \; w_\alpha = w_\beta$$

$$\sum_\beta p^n_{\alpha\beta} \; w_\beta = 1$$

(5.16) can be written as

$$E(y_{ij} \cdot y_{i+n,k}) = 4 \sum_{\alpha \in A_j} \sum_{\beta \in A_k} p^n_{\alpha\beta} \; w_\alpha - 2\Pr(y_{ij}=1) - 2\Pr(y_{ik}=1) + 1, \qquad (5.17)$$

which gives the correlation of the two output variables y_{ij} and $y_{i+n,k}$. To see the behavior of this correlation for large values of n, note that

$$\lim_{n\to\infty} p_{\alpha\beta}^{n} = w_{\beta}. \tag{5.18}$$

Thus, using (5.18), we have from (5.17)

$$\lim_{n\to\infty} E(y_{ij}\cdot y_{i+n,k}) = (2Q_j-1)\ (2Q_k-1)$$

where Q_j is defined as $Pr(y_{ij}=1)$.

When the source is symmetric then by (5.13) every state has the same probability as its complement. In this case if the number of taps used to generate y_{ij} is odd, then y_{ij} will also be symmetric, that is $Pr(y_{ij}=1) = 1/2$. This statement cannot be made if the number of taps is even. Codes for which all modulo-2 adders are connected to an odd number of taps are called transparent.

The average correlation of two output variables that occur m output bits apart can be written as

$$R_y(m\tau_o) = 1/v \sum_{j=1}^{v} E(y_{ij}\cdot y_{i+n,k})$$

where

$$n = [(j+m-1)/v]$$

$$k = j + m - nv$$

here [y] denotes the integer part of y, and τ_o is the output

bit time = $(1/v)\tau_i$, the input bit time.

Plots of $R_y(m\tau_o)$, obtained by the evaluation of the preceding expressions for various code configurations are shown in Figs. 5.2 and 5.3. Also shown in these figures is $R_x(m\tau_i)$, the correlation of two input symbols that occur m input bits apart. For the symmetric sources considered the input correlation is given by[32]

$$R_x(m\tau_i) = (r_{11} - r_{01})^m.$$

As can be seen from the examination of these figures, for small values of m, the output of the convolutional encoder is much less correlated than the input. In the case of the transparent code of Fig. 5.2, it is seen that for values of $m \geq v(K-1)$, the input and output correlations are nearly identical, and both correlations approach zero at the same exponential rate. In the case of the non-transparent code of Fig. 5.3 the correlation of the output is zero when m is odd. In addition, since for the non-transparent code the output is not symmetric ($Q_2 = .39$), the correlation of the output does not approach zero. Note that when the correlation does not approach zero, the spectral density will have line components. Thus, we see from this example, that the output of a non-transparent encoder may have line spectral components, even though the input contains no such components.

5.5 CONCLUSIONS

We have presented a method of calculating the correlation of the output of a binary convolutional encoder. For statistically independent inputs we have shown that the output of a convolutional encoder is uncorrelated, for most good codes. For first order Markov inputs we have seen that for short sequence lengths, the convolutional encoder output is considerably less correlated than the input. For transparent codes we have seen that the correlation function goes exponentially to zero when the input is symmetric. On the other hand, for non-transparent codes the correlation function can have a periodic or D.C. component.

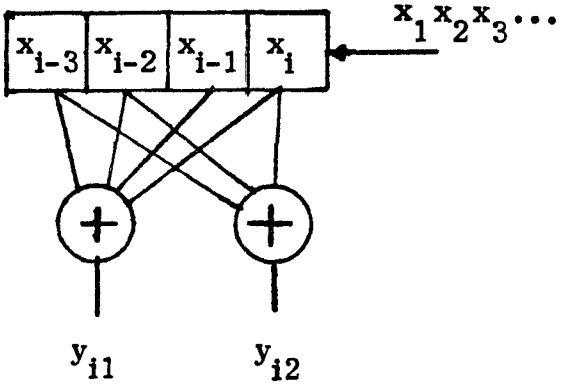

Fig. 5.1 A convolutional encoder for K=4, and v=2.

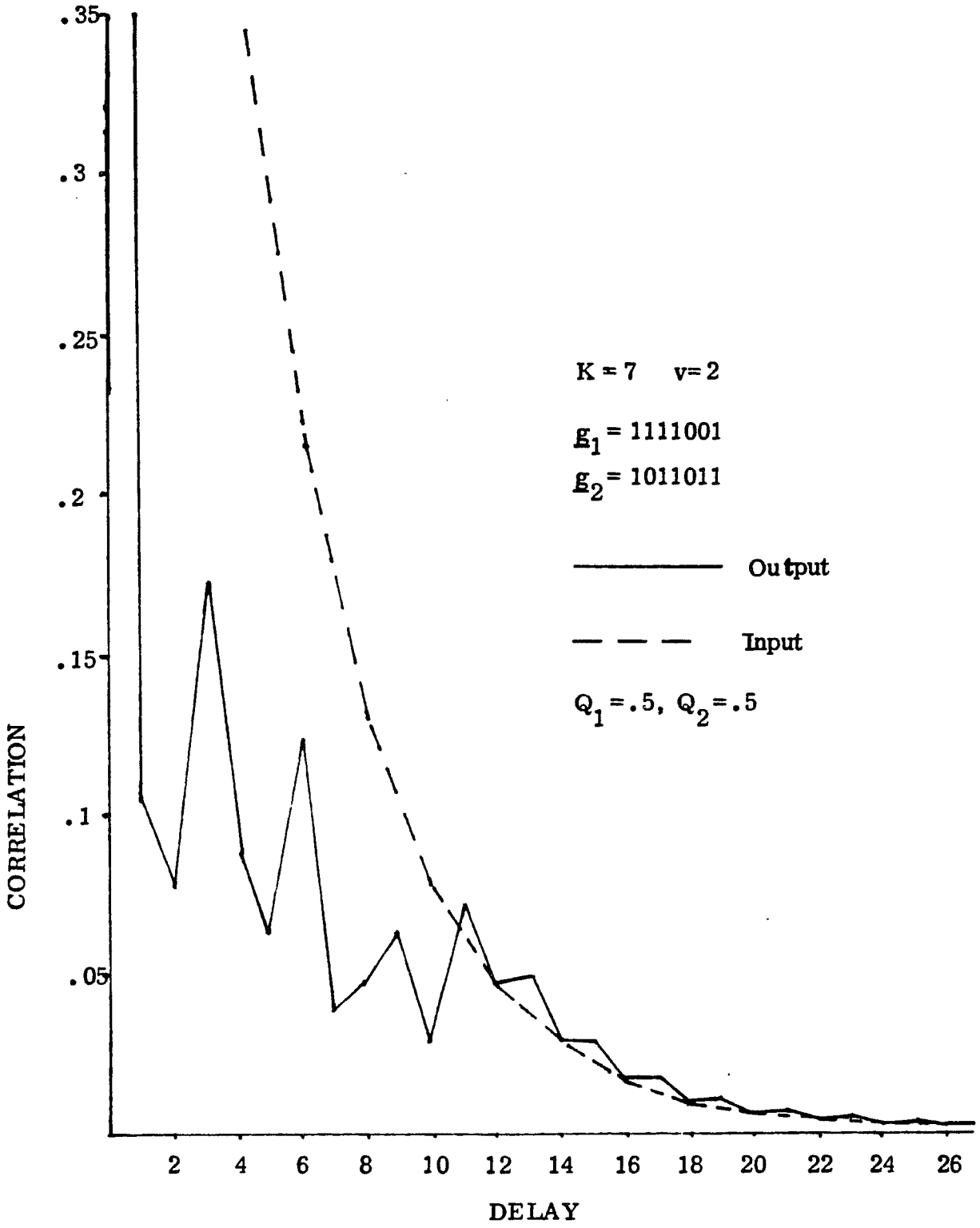

Fig. 5.2 Correlation of the output and input bit streams for a transparent code. Input is a Markov source with $r_{11} = r_{00} = .8$ and $r_{01} = r_{10} = .2$. $\tau_o = 1$ and $\tau_i = 2$.

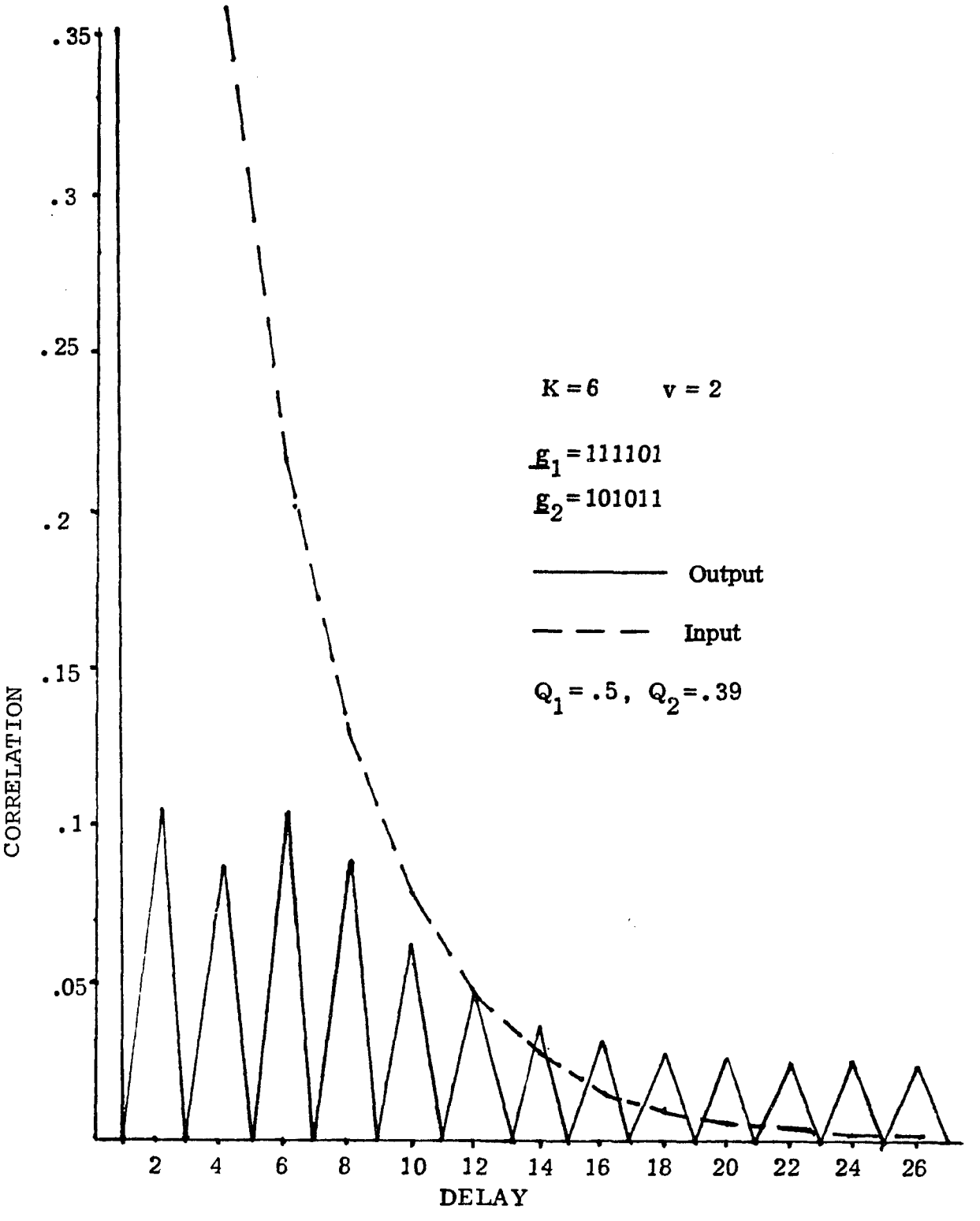

Fig. 5.3 Correlation of the output and input bit streams for a non-transparent code. $r_{11} = r_{00} = .8$, $r_{01} = r_{10} = .2$, $\tau_o = 1$, and $\tau_i = 2$.

Chapter 6

CHANNEL ERRORS IN DELTA MODULATION

6.1 INTRODUCTION

Delta Modulation is a method of analog-to-digital conversion in which samples of an analog waveform are compared to predicted values, and the difference is encoded by a 2 level (1 bit) quantizer. The Delta Modulator's estimate of the analog sample is then formed by combining the predicted value with the output of the quantizer. In Linear Delta Modulation the quantizer levels are fixed, whereas in Adaptive Delta Modulation the quantizer levels vary with local signal conditions. Compared to conventional PCM, Delta Modulation offers the advantages of conserving bandwidth by requiring lower bit rates, simpler implementation and greater tolerance to channel noise.

In studying the performance of Delta Modulator systems most authors have considered only the degradation due to the quantization process. In this study we will be concerned only with the noise which is caused by the presence of transmission errors.

The effect of channel errors in Linear Delta Modulation has received some attention in the past, however much of this has been of an approximate nature, and often analog as apposed to digital systems were studied. Typical of the approximate methods is the one presented by Johnson[33]. Here the received bit stream is considered as consisting of the original error free bit stream plus a random sequence of pulses due to the presence of errors. However, the error pulses are dependent on

the transmitted bit stream and this dependence is ignored. Wolf[34] and later Figueiras-Vidal[37] studied the effect of transmission errors in Delta Modulation taking into account the dependence of the error pulses on the statistics of the signal bit stream. However, to obtain results they had to assume unrealistic statistical properties for the signal bit stream. In addition, they do not predict a zero signal-to-noise ratio when the probability of error equals 1/2, as would be intuitively expected.

Channel errors not only introduce noise but also on the average reduce the output signal power. This fact although recognized by Wolf was not incorporated into his analysis. It turns out that when reduced signal power is taken into account, the characterization of the noise becomes much simpler and the S/N formula goes to zero for probability of error 1/2. The same phenomenon of signal suppression due to channel errors occurs also in PCM. An analysis of the performance of PCM systems including the effect of signal suppression was presented by Yates-Fish and Fitch[35], but no similar analysis seems to be available for Delta Modulation.

In this study we will derive exact formulas for the noise power in a Linear Delta Modulator, considering both independent and burst errors. We will also abtain approximate formulas for the case of Adaptive Delta Modulation.

6.2 RANDOM ERRORS IN LINEAR DELTA MODULATION

Noise Voltage

Consider a linear DM as showm in Fig. 6.1. The system is described by the following equations:

$$x[(k+1)T] = x(kT) + Se(kT) \tag{1a}$$

$$e(kT) = \text{sgn}[m(kT)-x(kT)] \tag{1b}$$

where m(kT) is the sampled input signal, x(kT) is the DM estimate of the sampled signal, S is the step size and $T = 1/f_s$ is the sampling period. From (1a) we can write:

$$x_k = x(kT) = x_o + S \sum_{i=1}^{k} e_i \tag{2}$$

where $e_i = e(iT)$. We now define x(t) such that:

$$x(t) = x_k , \quad kT < t < (k+1)T \tag{3}$$

where x(t) is the Delta Modulator's approximation to the input waveform. Using (2) and assuming for convenience that $x_o = 0$ we may write:

$$x(t) = \sum_{k=1}^{\infty} Se_k U(t-kT) \tag{4}$$

where U(t) is the unit step function.

Now let y(t) denote the received waveform at the output of the decoder. Then we have that:

$$y(t) = \sum_{k=1}^{\infty} Se_k' U(t-kt) \tag{5}$$

where e_k' represents the received bit when e_k was transmitted. We assume that the channel errors are independent of one another and define a sequence of independent identically distributed random variables $\{n_k\}$ as follows:

$$n_k = \begin{cases} -1, \text{ if } e_k \text{ is received incorrectly, Probability} = P_e \\ \\ 1, \text{ if } e_k \text{ is received correctly, Probablity} = 1-P_e \end{cases} \tag{6}$$

Thus we have

$$e_k' = n_k e_k \tag{7}$$

and $y(t)$ can be written as:

$$y(t) = \sum_{k=1}^{\infty} Se_k n_k U(t-kT) \tag{8}$$

The error voltage is simply the difference between the transmitted and received waveforms, and is given by:

$$E(t) = y(t)-x(t) \tag{9}$$

Channel errors are illustrated in Fig. 6.2. As shown each error contributes a step of magnitude 2S to the error voltage.

The expected value of the received waveform is:

$$E[y(t)] = \sum_{k=1}^{\infty} Se_k E[n_k] U(t-kT)$$

$$= \alpha x(t)$$

where

$$\alpha = E[n_k] = 1 - 2P_e.$$

Thus we see that on the average the effect of the channel errors is to decrease the magnitude of the received signal. We will consider $E[y(t)] = \alpha x(t)$ to be the signal component of the received waveform and the random deviations about this average to be the noise. Thus we may write:

$$y(t) = \alpha x(t) + N(t)$$

or

$$N(t) = y(t) - \alpha x(t)$$

$$N(t) = S \sum_{k=1}^{\infty} e_k(n_k - \alpha)U(t-kT) \tag{10}$$

With this definition for the noise we have

$$E[N(t)] = 0.$$

Other studies of Delta Modulation have considered the noise to be what we are calling the error waveform. However, using (8) and (9) gives for the expected value of the error waveform:

$$E[E(t)] = -2P_e x(t)$$

which shows that the error waveform has an average value proportional to the transmitted signal. This makes it unreasonable to treat the error waveform as noise.

Defining $b_k = n_k - \alpha$, it is clear that:

$$b_k = \begin{cases} 1-\alpha \text{ , with Probability } 1-P_e \\ -1-\alpha \text{ , with Probability } P_e \end{cases} \tag{11}$$

and from (10) N(t) becomes:

$$N(t) = \sum_{k=1}^{\infty} Se_k b_k U(t-kT). \tag{12}$$

Power Spectrum of the Noise

Having found a representation for the noise voltage due to channel errors we now proceed to determine its power spectral density (p.s.d.). From the p.s.d. we will compute the amount of inband noise power. The power spectrum of a signal can be calculated by one of two distinct methods. The first method is to find the autocorrelation function, $R(\tau)$, and then take its Fourier Transform. The second method which is often simpler to apply is as follows. First find the Fourier transform of a sample of the signal in a finite time interval, i.e.,

$$S_M(f) = \int_0^{MT} N(t)\, e^{-j2\pi ft}\, dt. \tag{13a}$$

Next evaluate the following:

$$G_M(f) = 2E[|S_M(f)|^2/MT] \tag{13b}$$

Finally the p.s.d. of the signal is given by

$$G(f) = \lim_{M\to\infty} G_M(f). \tag{13c}$$

It should also be noted that G(f) as defined above is the single sided p.s.d. due to the factor of 2 in (13b). In our analysis we will use the second approach. This method has been used by Bennett and Rice[36] to calculate the spectrum of waveforms resulting from frequency shift keying.

We begin by substituting for N(t) in (13a), and then interchanging the order of integration and summation. Next performing the integration we find

$$S_M(f) = \sum_{k=1}^{M} \frac{Se_k b_k}{j2\pi f} \left(e^{-j2\pi fkT} - e^{-j2\pi fMT}\right). \tag{14}$$

We next obtain the product $|S_M(f)|^2 = S_M(f) \cdot S_M^*(f)$, where * denotes complex conjugate, and then taking the expected value of both sides yields:

$$E[|S_M(f)|^2] = \sum_{k=1}^{M} \sum_{n=1}^{M} \frac{S^2}{4\pi^2 f^2} E(e_k e_n b_k b_n) \cdot \left(e^{-j2\pi fkT} - e^{-j2\pi fMT}\right)\left(e^{j2\pi fnT} - e^{j2\pi fMT}\right). \tag{15}$$

Note that the e_ks and b_ks are mutually independent. Also note that for $k \neq n$, $E(b_k b_n) = E(b_k) \cdot E(b_n) = 0$ and for $k=n$, $E(b_k b_n) = E(b_k^2) = E[(n_k - \alpha)^2] = 4P_e(1-P_e)$. Neglecting terms for which $k \neq n$ we have from (15)

$$E|[S_M(f)|^2] = \frac{2S^2 P_e(1-P_e)}{\pi^2 f^2} \left[M - \sum_{k=1}^{M} \cos 2\pi f(M-k)T\right]. \tag{16}$$

We now make the observation that:

$$\sum_{k=1}^{M} \cos 2\pi f(M-k)T = \sum_{k=0}^{M-1} \cos 2\pi fkT \tag{17a}$$

$$= \frac{1}{2} + \frac{\sin(2M-1)\pi fT}{2\sin \pi fT} \tag{17b}$$

Combining (17b), (16) and (13b) we obtain

$$G_M(f) = \frac{4S^2P_e(1-P_e)}{\pi^2 f^2 T}\left\{1-\frac{1}{M}\left[\frac{1}{2}+\frac{\sin(2M-1)\pi fT}{2\sin\pi fT}\right]\right\} \tag{18}$$

This is the power spectral density of the noise when we consider only the finite time interval (0, MT). We would now like to evaluate G(f) as the limit of $G_M(f)$ as $M\to\infty$. However we shall see that this limit does not exist for all values of f. We consider the following three cases separately:

Case 1: fT not an integer and $f\neq 0$. In this case the quantity in brackets in (18) remains finite and so we have:

$$G(f)=\lim_{M\to\infty} G_M(f)=\frac{4S^2 f_s P_e(1-P_e)}{\pi^2 f^2} \tag{19a}$$

where $f_s = 1/T$ is the sampling rate of the DM. It should be noted that for audio signals (19a) defines the inband noise spectral density since $fT=f/f_s < 1$ and $f > 0$ for all audio frequencies.

Case 2: fT is an integer, $f\neq 0$. In this case we have that $\cos 2\pi f(M-k)=1$ and so from (16) we have that

$$G(f) = 0. \tag{19b}$$

Case 3: f=0. To find $G_M(0)$ we apply L'hospital's rule to (18). The result is

$$G_M(0) = 8S^2P_e(1-P_e)T(2M^2-3M+1)/6 \tag{19c}$$

or for large M

$$G_M(0) = (8/3)S^2P_e(1-P_e)TM^2. \tag{19d}$$

Clearly $G_M(0) \to \infty$ as $M \to \infty$.

Total Average Noise Power

We now derive an expression for the total average power, P_T, which is given by

$$P_T = \langle E[N^2(t)] \rangle \tag{20}$$

where $\langle\ \rangle$ denotes time average. Using (12) we have

$$E[N^2(t)] = \sum_{k=1}^{\infty} \sum_{n=1}^{\infty} S^2 E[e_k e_n b_k b_n] U(t-kT)U(t-nT). \tag{21}$$

As before we can neglect terms for which $k \neq n$, and the above reduces to

$$E[N^2(t)] = \sum_{k=1}^{\infty} 4S^2 P_e(1-P_e)U(t-kT). \tag{22}$$

Averaging this on the interval [0,MT] yields

$$P_T = 2S^2 P_e(1-P_e)(M-1) \tag{23}$$

This last result indicates that in the presence of errors the total average noise power increases linearly with time (M). Clearly this situation is unacceptable, since hardware overload will inevitably occur. In the next section we describe a modification of the Delta Modulator that alleviates this difficulty.

Delta Modulation Using a Leaky Integrator

In order to prevent the continual increase in noise power, we modify the DM by introducing a leaky integrator. This is done by altering the encoder and decoder equations to read

$$x_k = Lx_{k-1} + Se_k \quad ; \text{ encoder} \tag{24a}$$

and

$$y_k = Ly_{k-1} + Se'_k \quad ; \text{ decoder} \tag{24b}$$

where L is the leak factor. It is important to choose L sufficiently large so as not to degrade the quality of the received signal. For audio signals with 256 quantization levels, L should be approximately 0.999.

Using (24) we obtain

$$x_k = L^k x_o + S \sum_{r=1}^{k} L^{k-r} e_r \tag{24c}$$

$$y_k = L^k y_o + S \sum_{r=1}^{k} L^{k-r} e'_r \quad , \quad e'_r = n_r e_r \tag{24d}$$

To see the effect of the leaky integrator assume that due to previous errors $x_o \neq y_o$ but that no new errors occur, then:

$$x_k - y_k = L^k(x_o - y_o)$$

and the difference between x_k and y_k decreases exponentially. To obtain an expression for the noise voltage due to channel errors assume $x_o=y_o=0$. We again find that the expected value of the received signal is:

$$E[y_k] = S \sum_{r=1}^{k} L^{k-r} E[n_r] e_r = \alpha x_k$$

Again this is the signal component of the received signal, the noise component is

$$N_k = y_k - \alpha x_k$$

$$N_k = S \sum_{r=1}^{k} L^{k-r} (n_r - \alpha) e_r \tag{25a}$$

$$N_k = S \sum_{r=1}^{k} L^{k-r} b_r e_r \tag{25b}$$

Power Spectrum of the Noise with Leak Factor

We now calculate the p.s.d. of the noise voltage for a DM using a leaky integrator. The procedure is similar to that used in the previous case. Using the fact that $N(t) = N_k$ for $kT<t<(kT+T)$ we substitute in (13a) for N(t). The result is

$$S_M^L(f) = \sum_{k=1}^{M-1} \int_{kT}^{(k+1)T} N_k e^{-j2\pi ft}\, dt. \tag{26}$$

Evaluating the integral, substituting from (25b) and then interchanging the order of summation we obtain

$$S_M^L(f) = \frac{S(1-e^{j2\pi fT})}{j2\pi f} \sum_{r=1}^{M-1} b_r e_r L^{-r} \sum_{k=r}^{M-1} \left(Le^{-j2\pi fT}\right)^k \tag{27}$$

The second summation is a geometric series, which can be evaluated to yield

$$S_M^L(f) = \frac{S(1 - e^{-j2\pi fT})}{j2\pi f(1 - Le^{-j2\pi fT})} \sum_{r=1}^{M-1} b_r e_r [e^{-j2\pi frT} - L^{M-r} e^{-j2\pi fMT}]. \tag{28}$$

We now form $|S_M^L(f)|^2$, take the expected value and note once again that only terms for which r=n need be retained. Thus we have

$$E|S_M^L(f)|^2 = \frac{S^2P_e(1-P_e)\ 2(1-\cos 2\pi fT)}{\pi^2f^2(1+L^2-2L\cos 2\pi fT)} \cdot \sum_{r=1}^{M-1} 1+L^{2(M-r)}-2L^{M-r}\cos 2\pi f(r-M)T \quad . \tag{29}$$

Evaluating the summation and substituting into (13b) we obtain

$$G_M^L(f)=\frac{4S^2P_e(1-P_e)}{T\pi^2f^2}\left[\frac{1-\cos 2\pi fT}{1+L^2-2L\cos 2\pi fT}\right]\frac{1}{M}\left\{M+\frac{L^{2M}-1}{L-1}\right. \tag{30}$$
$$\left.-2\left[\frac{L^{M+1}\cos 2\pi f(M-1)T-L^M\cos 2\pi fMT-L\cos 2\pi fT+1}{1+L^2-2L\cos 2\pi fT}\right]\right\}.$$

Considering the terms in the braces on the right hand side of (30), as M tends to infinity all terms except the first one remain finite provided L < 1. Thus letting M go to infinity we have

$$G_L(f)=\lim_{M\to\infty} G_M^L(f)=\frac{4S^2P_e(1-P_e)f_s}{\pi^2f^2}\left[\frac{1-\cos 2\pi fT}{1+L^2-2L\cos 2\pi fT}\right] \tag{31}$$

which exists at f=0 provided L≠1.

Equation (31) can be rewritten as follows

$$G_L(f)= 8P_e(1-P_e)S^2T\left[\frac{\sin \pi f/f_s}{\pi f/f_s}\right]^2 \frac{1}{(4L\sin^2\pi f/f_s)+(1-L)^2} \tag{32}$$

This is the p.s.d. of the noise resulting from a LDM with a leaky integrator. The p.s.d. and its integral have been numerically evaluated. Figures 6.3 and 6.4 show plots of these functions. From (32) we note that

$$G_L(0) = \frac{8P_e(1 - P_e)S^2T}{(1 - L)^2} \tag{33}$$

We now find the total noise power using a leaky integrator.

Total Noise Power with Leaky Integrator

Using (20) and (25b) we have for the noise power with leaky integrator

$$P_T^L = \lim_{M\to\infty} \frac{1}{MT} \sum_{k=1}^{M-1} \int_{kT}^{(k+1)T} E\left[S^2 \sum_{r=1}^{k} \sum_{n=1}^{k} b_r b_n e_r e_n L^{k-r} L^{k-n}\right] dt \tag{34}$$

Evaluating (34) we obtain

$$P_T^L = \frac{4S^2 P_e(1 - P_e)}{1 - L^2} \tag{35}$$

which remains finite provided $L < 1$. We next establish as a figure of merit the ratio of the inband noise power to total noise power.

Inband Noise Power

Consider the factor $[4L\sin^2\pi f/f_s + (1-L)^2]$ in (32). Since $(1-L) \approx 10^{-3}$, then within the audio band $4L\sin^2\pi f/f_s >> (1-L)^2$. This is true provided $f_s << 2\times10^6$Hz and the lower limit of the

audio band is assumed to be 300Hz. With these assumptions we can neglect the $(1-L)^2$ term in (32), and as a result (32) reduces to

$$G_L(f) = \frac{2\,P_e(1 - P_e)S^2 f_s}{L\pi^2 f^2} \,. \tag{37}$$

Integrating this over the audio range of 300 to 3000Hz yields the inband noise power

$$P_I = \frac{0.006\,P_e(1 - P_e)S^2 f_s}{L\,\pi^2} \tag{38}$$

From this we find the ratio of the inband noise power to total noise power, R, as

$$R = \frac{0.0015\,f_s(1 - L^2)}{L\,\pi^2} \tag{39}$$

Two values for R are shown in Table 6.1 for $f_s = 8 \times 10^5$. Thus we note the important fact that most of the noise power lies outside of the useful band.

6.3 RANDOM ERRORS IN ADAPTIVE DELTA MODULATION

We now investigate the characteristics of the inband noise power in an Adaptive Delta Modulator (ADM). The ADM we are considering here is shown in Fig. 6.5 [41]. Its operation is described by the following equations:

$$\underline{\text{Encoder}}: \quad e_k = \text{sgn}\,(m_k - x_k) \tag{40a}$$

$$x_k = x_{k-1} + S_k \tag{40b}$$

$$S_k = |S_{k-1}|\, e_k + S_o e_{k-1} \tag{40c}$$

Decoder: $$y_k = y_{k-1} + S'_k \tag{41a}$$

$$S'_k = |S_{k-1}|\, e'_k + S_o e'_{k-1} \,. \tag{41b}$$

Once again in the decoder equations the symbols e'_k and S'_k represent quantities that have been perturbed due to channel errors. It has been found experimentally that this algorithm is particularly suited to the encoding of audio signals.

For the LDM we found that a single channel error causes a deviation of magnitude 2S between the transmitted and received signal. This simple situation allowed us to calculate the p.s.d. of the noise. In the ADM the situation is much more complex. To appreciate this fact notice that when a channel error occurs, not only is the decoder output y_k different from the encoder estimate x_k, but the decoder step size S'_k is also affected. This step size error may not disappear in some cases and as a result the error voltage will continually fluctuate.

Fig. 6.6 illustrates x(t), y(t) and E(t)=y(t)-x(t) when a single error occurs at bit e_k. We consider here the eight possible values of the three bits e_{k-1}, e_k and e_{k+1}. As we see from Fig. 6.6, in the four cases for which $e_{k-1} \neq e_{k+1}$, the step size error disappears at time k+1, and the error voltage is identical to the LDM case. In the four other instances it is seen that the step size error will not disappear.

As a first order approximation to this highly complex problem we will neglect the effects of step size errors, and assume that the correct step size is available at the decoder. This would correspond to the situation where step size information is transmitted to the receiver in an error protected format. Such schemes have been suggested in the literature [38] for differential Codecs operating in a noisy environment.

The ADM equations can be written in an alternate form as follows:

$$\underline{\text{Encoder}}: \quad x_k = x_{k-1} + |S_k| e_k \tag{42a}$$

$$|S_k| = |S_{k-1}| + S_o e_k e_{k-1}, \quad |S_{k-1}| \neq 0 \tag{42b}$$

$$|S_k| = S_o \qquad |S_{k-1}| = 0 \tag{42c}$$

$$\underline{\text{Decoder}}: \quad y_k = y_{k-1} + |S_k| e'_k \tag{43}$$

From (42) and (43) we obtain:

$$x_k = x_o + \sum_{i=1}^{k} |S_i| e_i$$

$$y_k = y_o + \sum_{i=1}^{k} |S_i| e'_i \ .$$

When a leaky integrator is employed these equations become:

$$x_k = L x_{k-1} + |S_k| e_k \quad \text{;encoder}$$

$$y_k = L y_{k-1} + |S_k| e'_k \quad \text{; decoder}$$

or

$$x_k = L^k x_o + \sum_{r=1}^{k} L^{k-r} |S_r| e_r$$

$$y_k = L^k y_o + \sum_{r=1}^{k} L^{k-r} |S_r| e'_r$$

Following the same procedure as for the LDM we have:

$$E[y_k] = \alpha x_k \quad ; \quad \alpha = 1 - 2P_e$$

$$N_k = y_k - \alpha x_k$$

or

$$N_k = \sum_{r=1}^{k} L^{k-r} |S_r| b_r e_r \tag{44}$$

We see that (44) is identical to (25b) which applied for the LDM except that Se_r is replaced with $|S_r| e_r$. Referring to (29) we see that it is simply necessary to replace S^2 by $E[S_r^2]$ to find the p.s.d. of the noise for the ADM. We then have:

$$G_L^A(f) = 8P_e(1-P_e)E[S_r^2]T \left[\frac{\sin \pi f/f_s}{\pi f/f_s}\right]^2 \cdot \frac{1}{(4L\sin^2 \pi f/f_s)+(1-L)^2} \tag{45a}$$

where $G_L^A(f)$ is the p.s.d. of the noise for the ADM with a leaky integrator. The inband noise power is given by:

$$P_I^A = \int_{300}^{3000} G_L^A(f)\, df \ . \tag{45b}$$

It should be noted that the sampling rate of the ADM is much less than the sampling rate for a LDM. Therefore, the approximation that $4L\sin^2\pi f/f_s >> (1-L)^2$ is also valid over the audio band. The inband noise spectrum is then given by:

$$G_L^A(f) = 2P_e(1-P_e)E[S_r^2]f_s/L\pi^2f^2 \qquad (45c)$$

thus

$$P_I^A = 0.006\ P_e(1-P_e)\ E[S_r^2]f_s/L\pi^2 \qquad (45d)$$

We have experimentally determined the distribution of S_r^2. These results are shown in Table 6.2 and Figs. 6.7a and 6.7b. The ratio of the inband noise power to total noise power for the ADM is

$$R_A = 0.0015\ f_s\ (1 - L^2)/L\pi^2\ . \qquad (45e)$$

Two values for R_A are also given in Table 6.1 with f_s= 32KHz. Here we see again that relatively little of the noise is inband. If we form the ratio of the inband noise power in the ADM to that of the LDM we find

$$R_P = E[S_r^2]\ f_s^A\ /\ S^2 f_s^L\ . \qquad (45f)$$

For f_s^A = 32Khz, f_s^L = 800Khz and $E[S_r^2] = 32S^2$ (see Table 6.2) we find that R_P= 1.28 = 1.07 dB. Thus we see that the inband noise power in an ADM is about 1 dB greater than in an LDM. This is understandable since the expected value of the step size squared is 32 times as great in the ADM as compared to the LDM, but this is offset by the lower sampling rate for the ADM.

6.4 BURST ERRORS IN LINEAR DELTA MODULATION

In this section we discuss the effect of errors that do not occur independently but rather occur in bursts. As first approximation we assume bursts of fixed lengths occurring periodically in time. This situation is shown in Fig. 6.8. We assume that during the burst interval which lasts IT seconds, the probability of error is .5. The sequence $\{n_k\}$ which specifies the occurrence or nonoccurrence of errors is in the present case not a sequence of identically distributed random variables as was the case for random errors. Rather their distribution is as follows:

$$n_k = \begin{cases} 1, & \text{with probability } 1/2 \\ -1, & \text{with probability } 1/2 \end{cases} \qquad k=nD+i,\ 1<i<I \tag{46a}$$

$$n_k = 1, \quad \text{with probability } 1 \qquad \text{otherwise} \tag{46b}$$

Thus the n_k are -1 or +1 with equal probability during an error burst, and are always +1 during the error free period. It is reasonable to assume that the phase of the periodic errors is random with respect to the beginning of Delta Modulator transmission. Such an assumption is necessary in order for the received bit stream to be stationary. To provide for this random phase we introduce a random variable θ which can take on the values $0,1,\ldots,D-1$ with equal probability. When bit e_k is transmitted the received bit is $n_{k+\theta}e_k$. Thus the output of an LDM decoder without leaky integration is given by:

$$y(t) = S \sum_{k=1}^{\infty} n_{k+\theta} e_k \, U(t-kT) \tag{47}$$

The expected value of the decoder output is:

$$E[y(t)] = S \sum_{k=1}^{\infty} E[n_{k+\theta}]\, e_k\, U(t-kT) \tag{48a}$$

but

$$E[n_{k+\theta}] = \frac{1}{D} \sum_{i=0}^{D-1} E[n_{k+\theta} / \theta=i] = \frac{D-I}{D} = 1 - \frac{I}{D} \tag{48b}$$

The sum above involves an entire period of the sequence $\{n_k\}$ and thus will contain I terms for which $E[n_{k+\theta}]=0$ and D-I terms for which $E[n_{k+\theta}]=1$. Also the average number of errors in a period of D bits is I/2. We can therefore say that $P_e = I/2D$. Thus we have from (48)

$$E[y(t)] = (1-2P_e)x(t) = \alpha x(t) \tag{49}$$

From this point on the development proceeds as in the previous sections. We have for the noise voltage:

$$\begin{aligned} N(t) &= y(t) - \alpha x(t) \\ &= S\sum_{k=1}^{\infty} (n_{k+\theta} - \alpha)\, e_k\, U(t-kT) \end{aligned} \tag{50}$$

Define $b_k = n_{k+\theta} - \alpha$, then

$$N(t) = S \sum_{k=1}^{\infty} b_k e_k\, U(t-kT) \tag{51}$$

and

$$E[b_k] = 0, \quad E[b_k^2] = 4P_e(1-P_e) \tag{52}$$

Equation (51) is identical to equation (12) which was derived for random errors. Also (52) is identical to the conditions that held for random

errors. We can therefore conclude that the spectrum of the noise with burst errors would be given by the same formulas that held for random errors [i.e. (18) and (19)]. It is important to bear in mind however that for burst errors as defined here $P_e = I/2D$.

Burst Errors With Leaky Integration

Again the development here follows closely the one given for random errors. Refering back to (24e), we have for the output of the encoder:

$$x_k = S \sum_{r=1}^{k} L^{k-r} e_r \quad ; \text{ encoder} \tag{53}$$

Where we assume that $x_o = 0$. The decoder output in the presence of burst errors is then given by:

$$y_k = S \sum_{r=1}^{k} L^{k-r} n_{r+\theta} e_r \tag{54}$$

As shown in the previous section $E[n_{k+\theta}] = 1 - I/2D = \alpha$. Therefore:

$$E[y_k] = \alpha x_k \tag{55}$$

The noise at time k is therefore:

$$N_k = y_k - \alpha x_k$$

$$= S \sum_{r=1}^{k} L^{k-r} (n_{r+\theta} - \alpha) e_r$$

$$= S \sum_{r=1}^{k} L^{k-r} b_r e_r \quad , b_r = n_{r+\theta} - \alpha \ . \tag{56}$$

Once again this formula is identical to (25b) which was found for random errors. The spectrum of the noise for burst errors is thus given by (32), and all other results found for random errors hold also for burst errors provided P_e is set equal to I/2D.

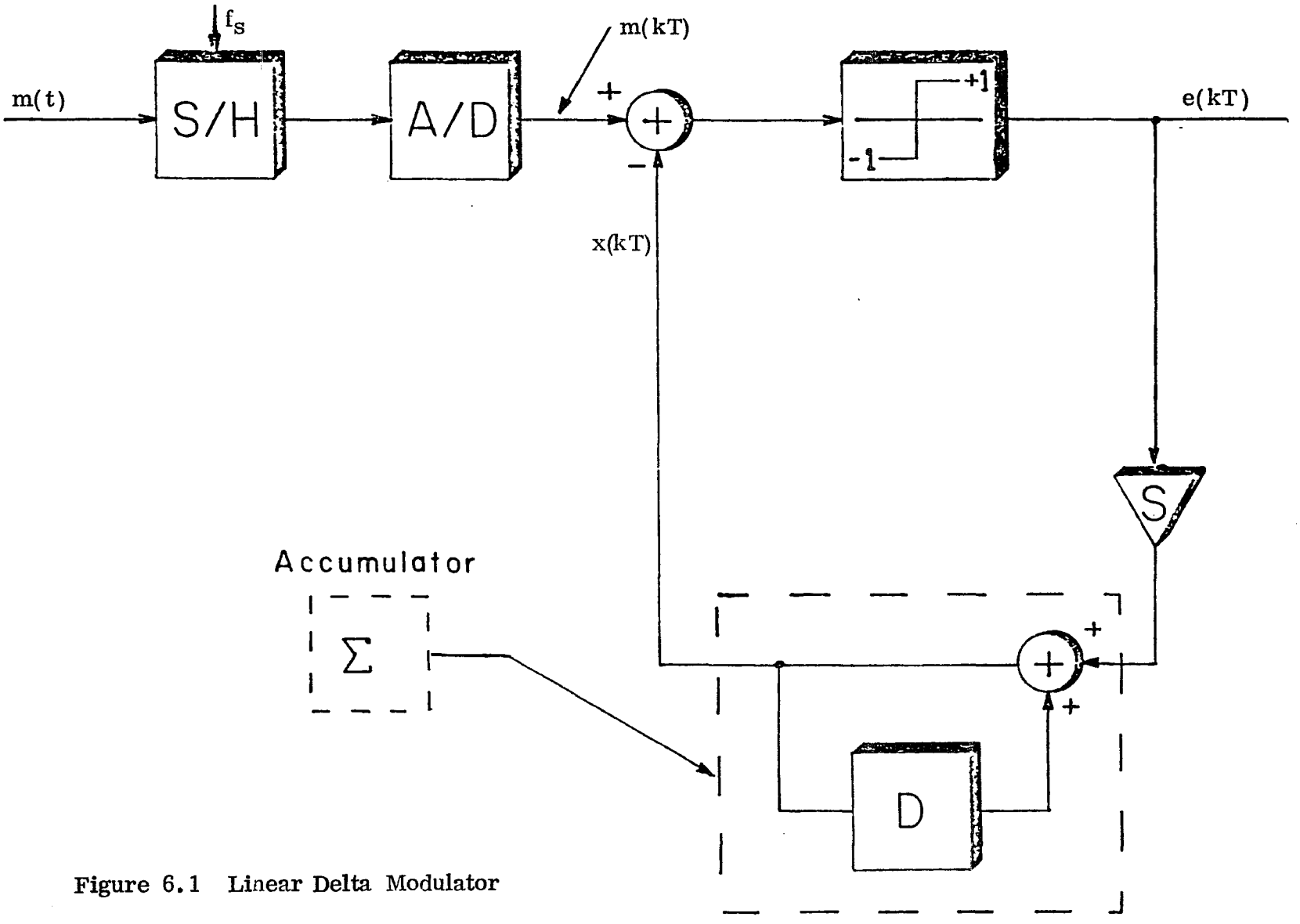

Figure 6.1 Linear Delta Modulator

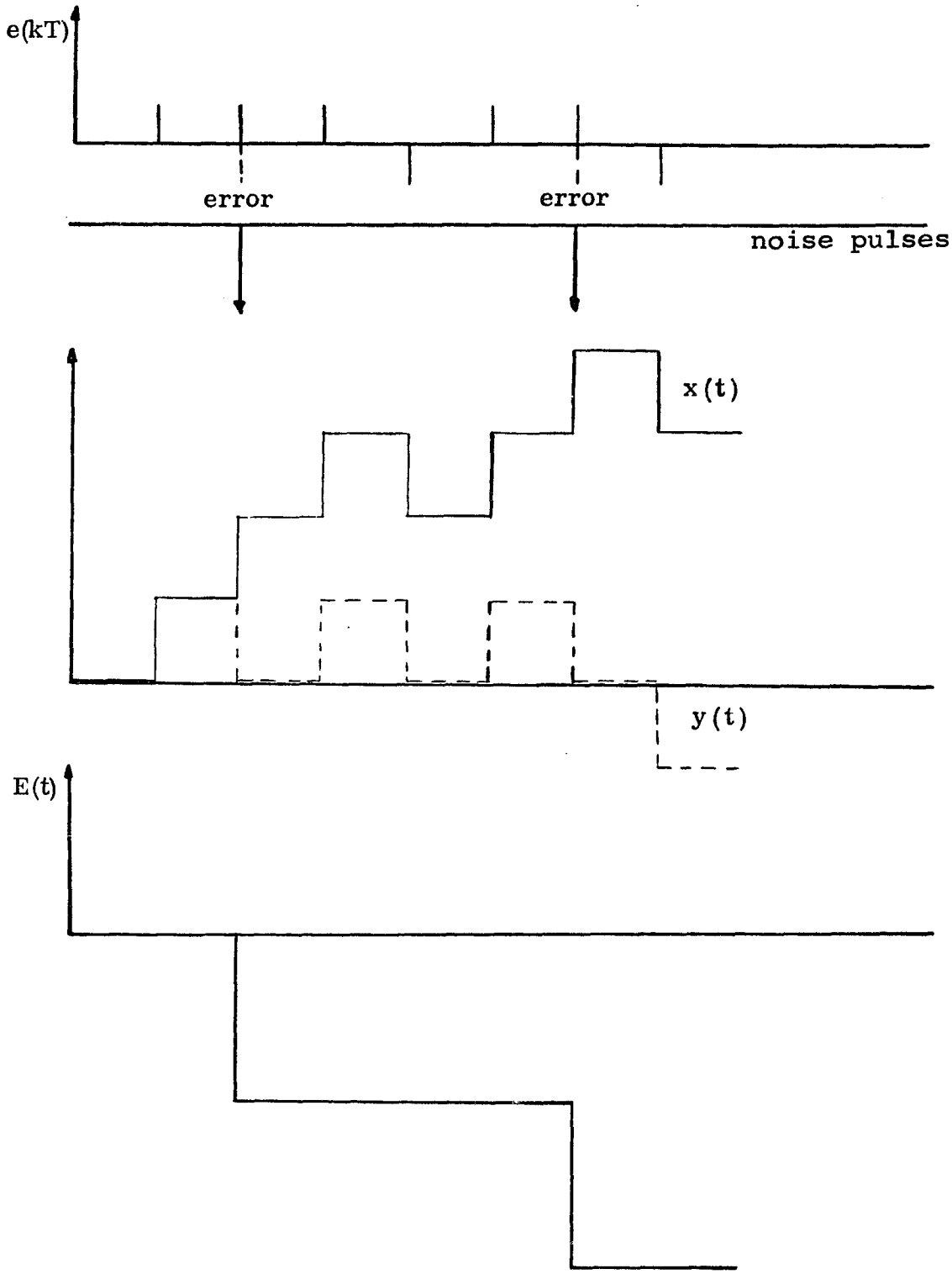

Figure 6.2 Illustrating the effect of channel errors on the error voltage in a LDM.

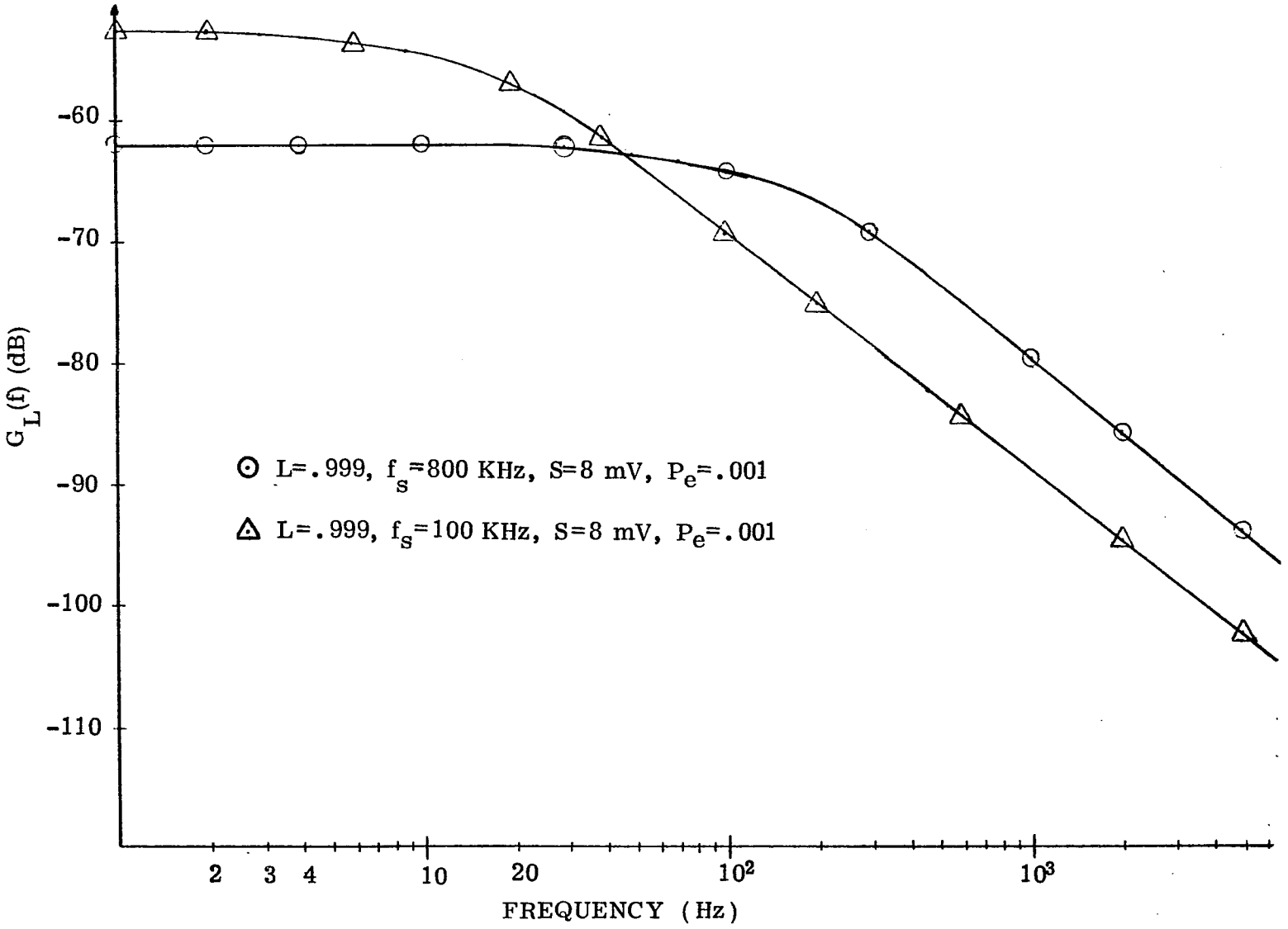

Figure 6.3 Noise Spectral Density for LDM with Leak Factor

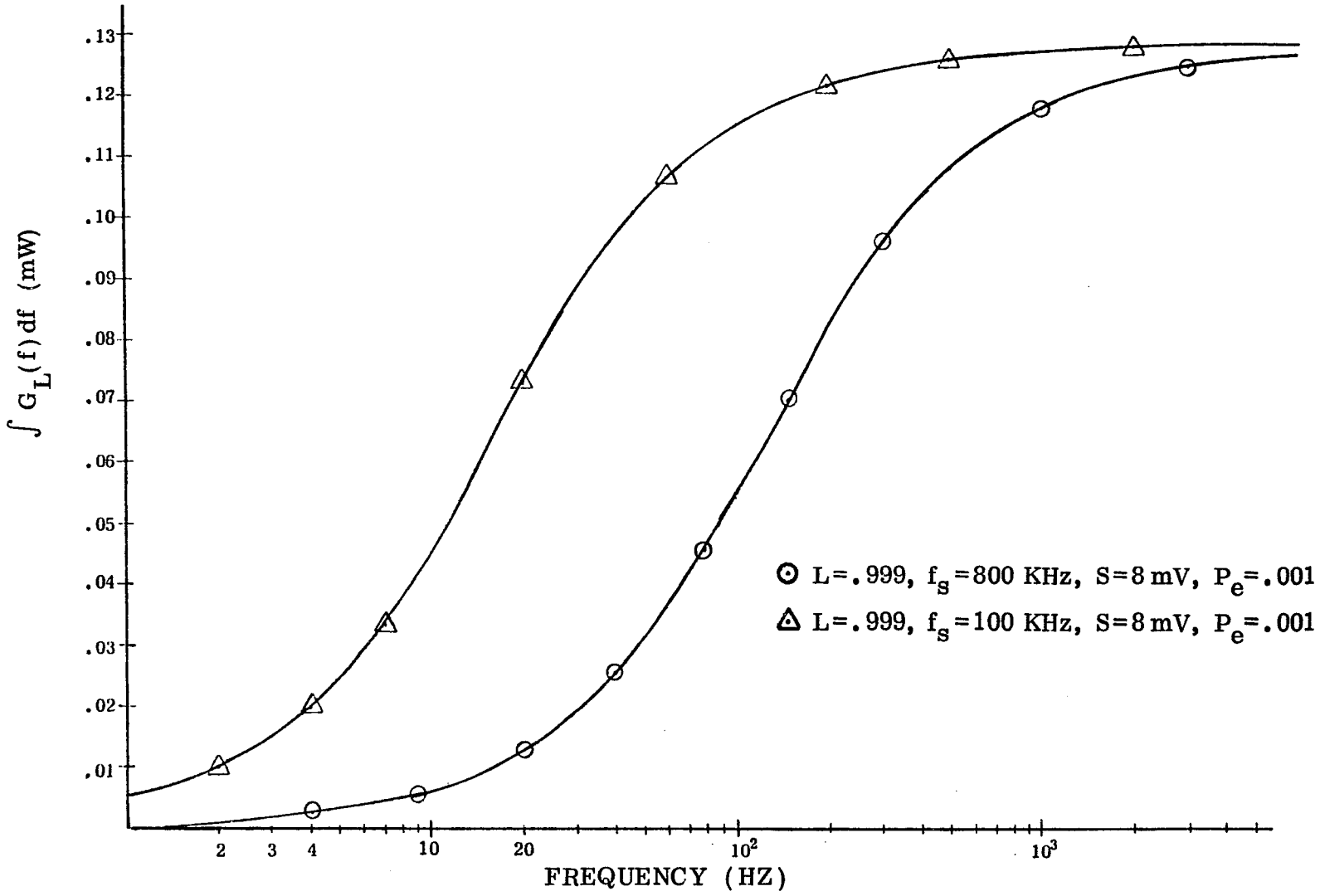

Figure 6.4 Integral of the Noise Spectrum

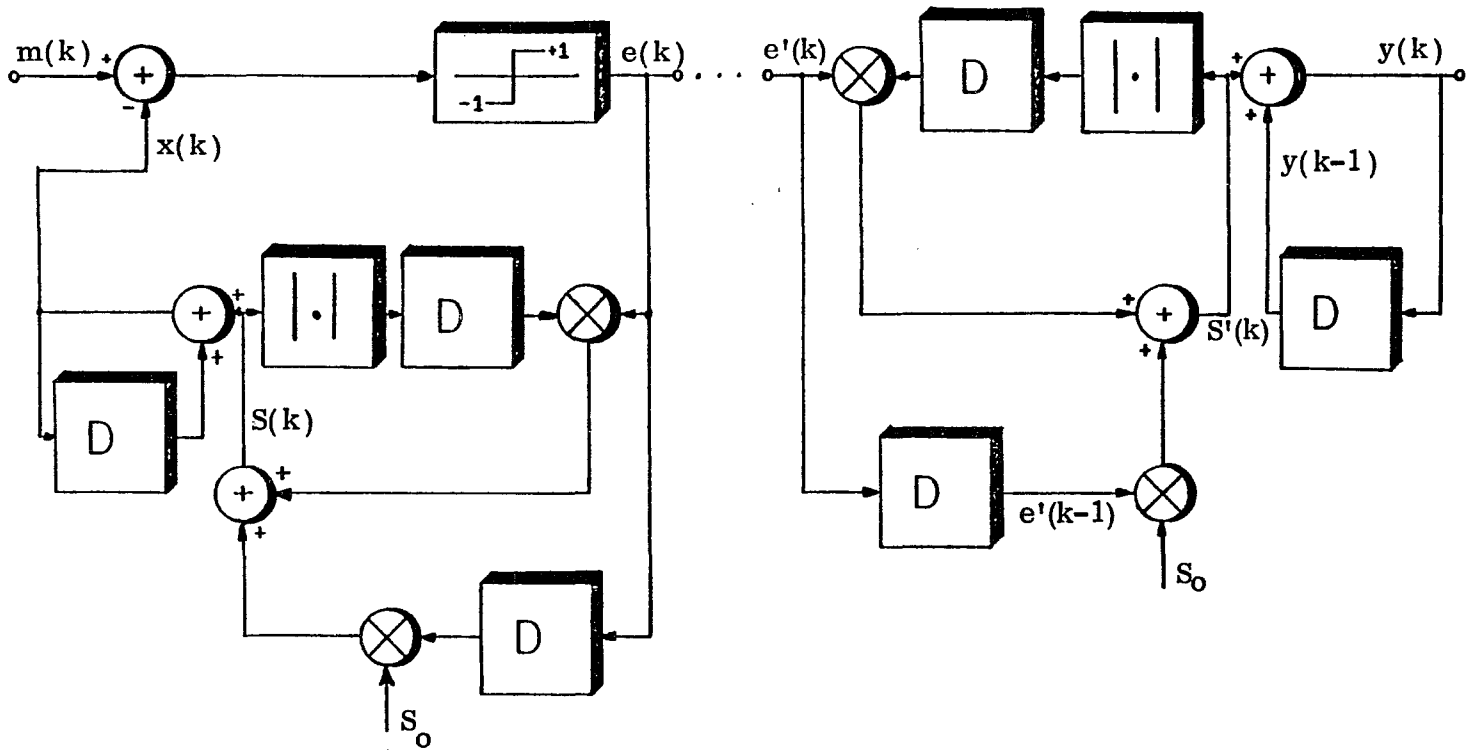

Figure 6.5 Adaptive Delta Modulator

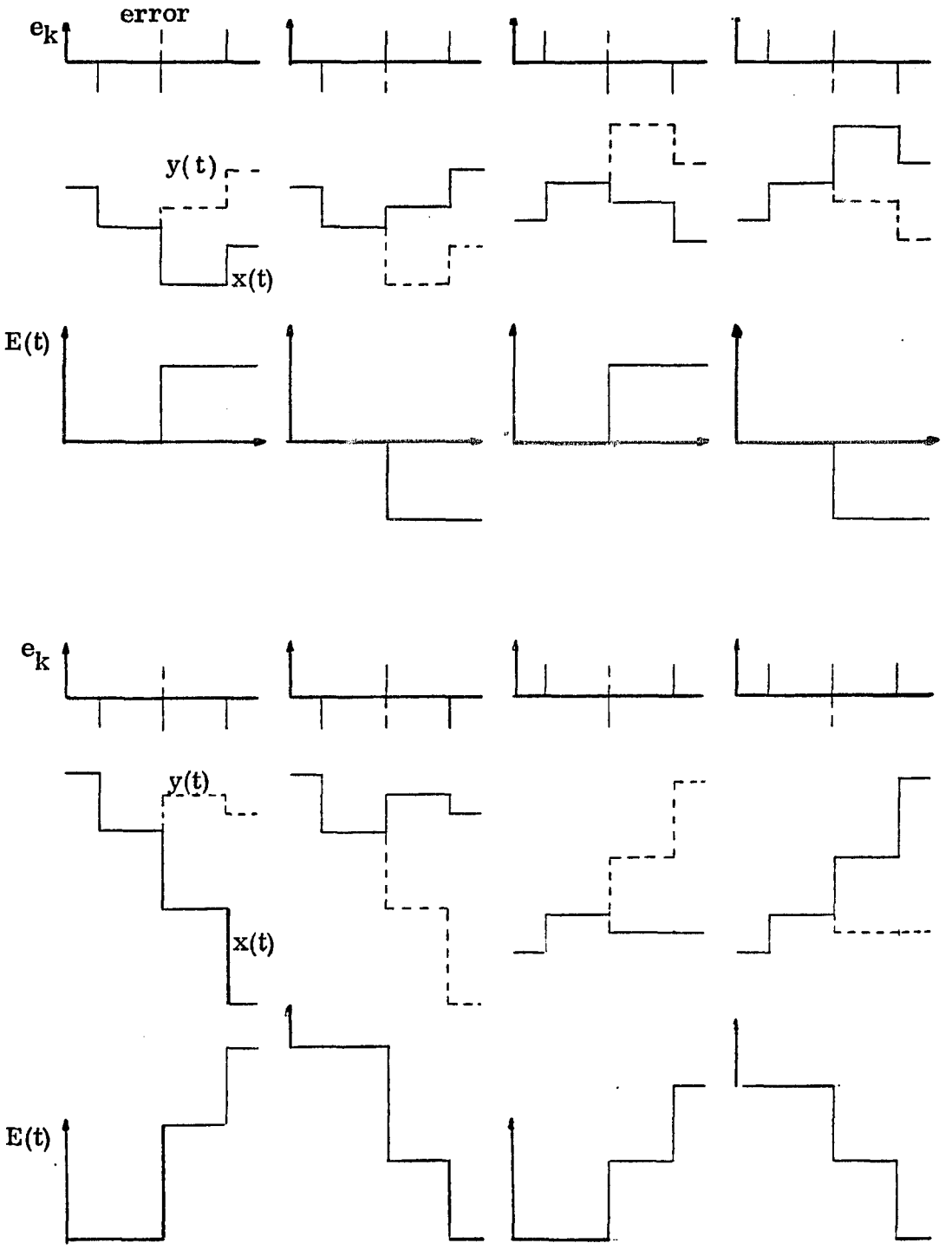

Figure 6.6 The 8 Possible Error Patterns in an ADM

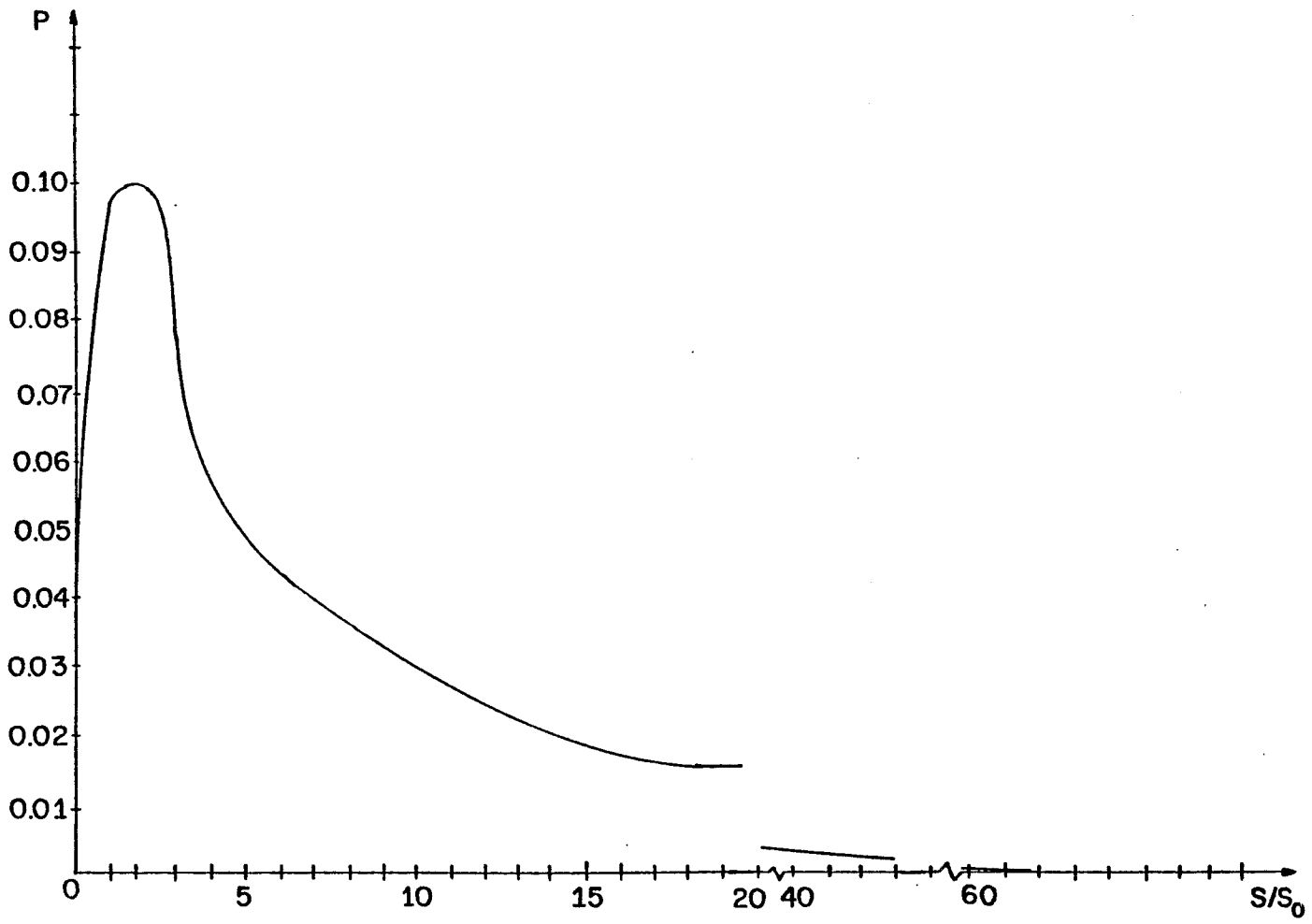

Figure 6.7a Probability Distribution of the Step Size for 1024 Levels, f_s = 32 KHz

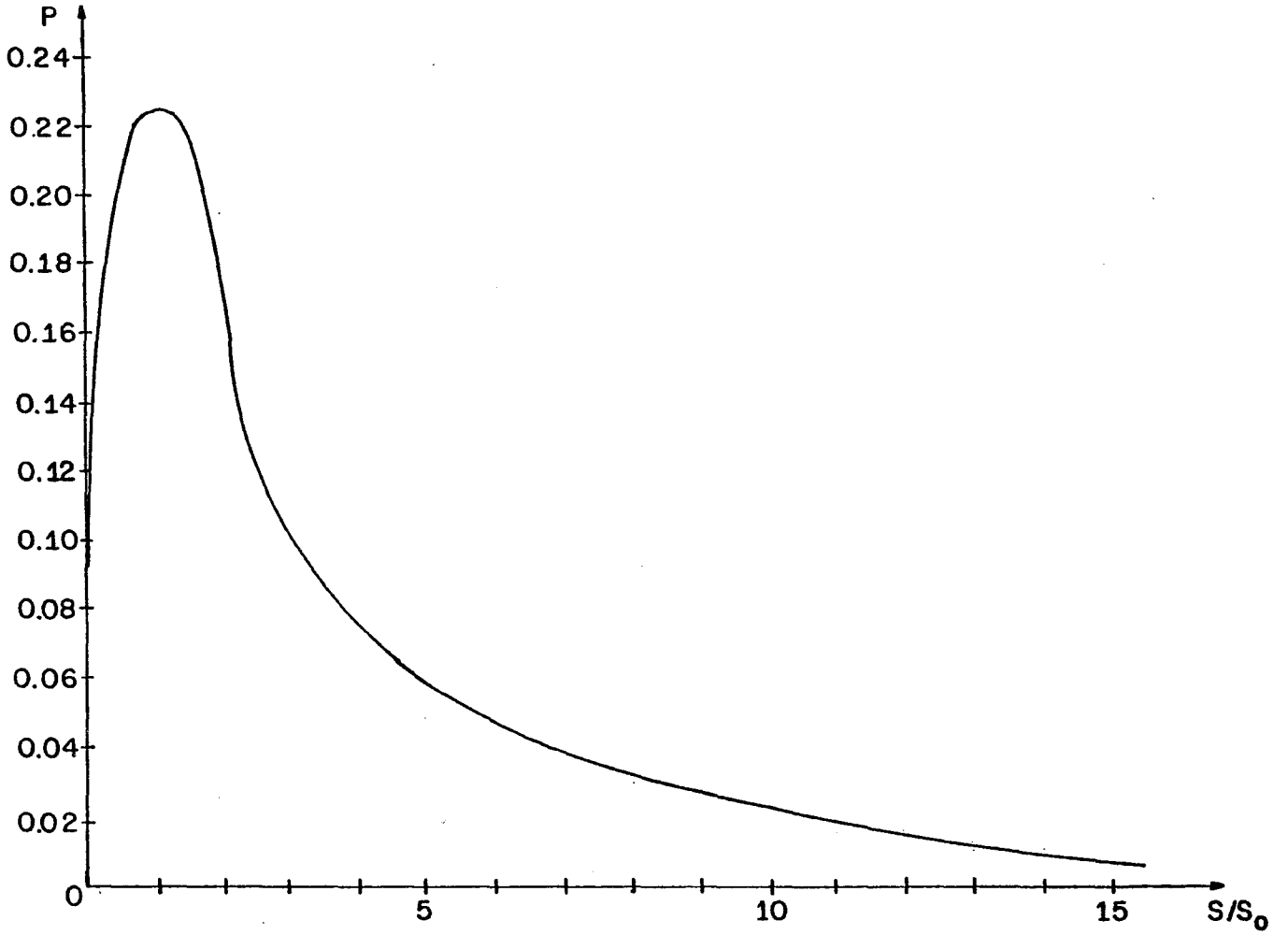

Figure 6.7b Probability Distribution of the Step Size for 256 Levels, $f_s = 32$ KHz

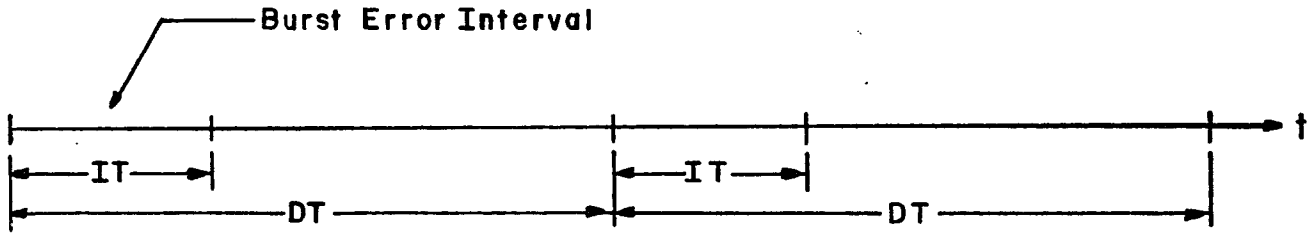

Figure 6.8 Illustrating Periodic Burst Errors

TABLE 6.1

RATIO OF INBAND TO TOTAL NOISE POWER

LDM			ADM	
f_s (KHz)	R	L	R_A	f_s (KHz)
800	0.24	0.999	0.00973	32
800	0.024	0.9999	0.000973	32

STEPSIZE DISTRIBUTION 1024 LEVELS

S/S0= 0	PROB=.035544	S/S0=41	PROB=.004337
S/S0= 1	PROB=.087056	S/S0=42	PROB=.004089
S/S0= 2	PROB=.091404	S/S0=43	PROB=.003870
S/S0= 3	PROB=.067035	S/S0=44	PROB=.003585
S/S0= 4	PROB=.049642	S/S0=45	PROB=.003346
S/S0= 5	PROB=.043603	S/S0=46	PROB=.003167
S/S0= 6	PROB=.041170	S/S0=47	PROB=.002918
S/S0= 7	PROB=.038451	S/S0=48	PROB=.002628
S/S0= 8	PROB=.035068	S/S0=49	PROB=.002359
S/S0= 9	PROB=.032176	S/S0=50	PROB=.002096
S/S0=10	PROB=.029616	S/S0=51	PROB=.001832
S/S0=11	PROB=.027279	S/S0=52	PROB=.001596
S/S0=12	PROB=.025439	S/S0=53	PROB=.001386
S/S0=13	PROB=.023784	S/S0=54	PROB=.001192
S/S0=14	PROB=.022476	S/S0=55	PROB=.001016
S/S0=15	PROB=.021128	S/S0=56	PROB=.000876
S/S0=16	PROB=.019705	S/S0=57	PROB=.000801
S/S0=17	PROB=.018659	S/S0=58	PROB=.000751
S/S0=18	PROB=.017724	S/S0=59	PROB=.000703
S/S0=19	PROB=.016742	S/S0=60	PROB=.000653
S/S0=20	PROB=.015874	S/S0=61	PROB=.000581
S/S0=21	PROB=.015176	S/S0=62	PROB=.000500
S/S0=22	PROB=.014411	S/S0=63	PROB=.000432
S/S0=23	PROB=.013691	S/S0=64	PROB=.000399
S/S0=24	PROB=.013055	S/S0=65	PROB=.000379
S/S0=25	PROB=.012501	S/S0=66	PROB=.000323
S/S0=26	PROB=.012102	S/S0=67	PROB=.000247
S/S0=27	PROB=.011749	S/S0=68	PROB=.000202
S/S0=28	PROB=.011347	S/S0=69	PROB=.000145
S/S0=29	PROB=.010711	S/S0=70	PROB=.000114
S/S0=30	PROB=.010108	S/S0=71	PROB=.000091
S/S0=31	PROB=.009636	S/S0=72	PROB=.000071
S/S0=32	PROB=.008998	S/S0=73	PROB=.000057
S/S0=33	PROB=.008153	S/S0=74	PROB=.000049
S/S0=34	PROB=.007450	S/S0=75	PROB=.000032
S/S0=35	PROB=.006942	S/S0=76	PROB=.000018
S/S0=36	PROB=.006322	S/S0=77	PROB=.000011
S/S0=37	PROB=.006543	S/S0=78	PROB=.000007
S/S0=38	PROB=.005202	S/S0=79	PROB=.000003
S/S0=39	PROB=.004868	S/S0=80	PROB=.000001
S/S0=40	PROB=.004596		

E[S/S0]=13.217 E[S/S0**2]=339.94

TABLE 6.2a

STEPSIZE DISTRIBUTION 256 LEVELS

S/S0= 0	PROB=.092666
S/S0= 1	PROB=.222241
S/S0= 2	PROB=.191297
S/S0= 3	PROB=.103111
S/S0= 4	PROB=.073478
S/S0= 5	PROB=.058510
S/S0= 6	PROB=.048192
S/S0= 7	PROB=.040111
S/S0= 8	PROB=.033615
S/S0= 9	PROB=.028030
S/S0=10	PROB=.023210
S/S0=11	PROB=.018949
S/S0=12	PROB=.015445
S/S0=13	PROB=.012515
S/S0=14	PROB=.009942
S/S0=15	PROB=.007804
S/S0=16	PROB=.006000
S/S0=17	PROB=.004477
S/S0=18	PROB=.003224
S/S0=19	PROB=.002267
S/S0=20	PROB=.001634
S/S0=21	PROB=.001174
S/S0=22	PROB=.000790
S/S0=23	PROB=.000540
S/S0=24	PROB=.000350
S/S0=25	PROB=.000202
S/S0=26	PROB=.000106
S/S0=27	PROB=.000051
S/S0=28	PROB=.000034
S/S0=29	PROB=.000020
S/S0=30	PROB=.000009
S/S0=31	PROB=.000005
S/S0=32	PROB=.000002

E[S/S0]= 4.016 E[S/S0**2]=32.360

TABLE 6.2b

REFERENCES

[1] B.M. Oliver, J.R. Pierce and C.E. Shannon, "The Philosophy of PCM," Proc. IRE, Vol. 36, pp 1324-1331, Oct. 1948.

[2] J.W. Pan, "Synchronization and Multiplexing in Digital Communication Networks," Proc. IEEE, Vol. 60, May 1972.

[3] A. Gersho and B.J. Karafin, "Mutual Synchronization of Geographically Separated Oscillators," Bell Syst. Tech. J., pp 1689-1704, Dec. 1966.

[4] M.B. Brilliant, "The Determination of Frequency in Systems of Mutually Synchronized Oscillators," Bell Syst. Tech. J., pp 1737-1748, Dec. 1966.

[5] J.R. Pierce, "Synchronization of Digital Networks," Bell Syst. Tech. J., pp. 615-636, March 1969.

[6] M.B. Brilliant, "Dynamic Response of a System of Mutually Synchronized Oscillators," Bell Syst. Tech. J., pp. 319-356, Feb. 1967.

[7] I.W. Sandberg, "On Conditions Under Which it is Possible to Synchronize Digital Transmission Systems," Bell Syst. Tech. J., pp. 1999-2022, July-August 1969.

[8] M.W. Willard, "Analysis of a System of Mutually Synchronized Oscillators," IEEE Trans. Commun. Technol., pp. 467-483, Oct. 1970.

[9] M.W. Willard and H.R. Dean, "Dynamic Behavior of a System of Mutually Synchronized Oscillators," IEEE Trans. Commun. Technol., August 1971.

[10] J. Yamato, M. Ono and S. Usuda, "Synchronization of a PCM Integrated Telephone Network," IEEE Trans. Commun. Technol., Feb. 1968.

[11] J. Yamato, S. Nakajima and K. Saito, "Dynamic Behavior of a Synchronization Control System for an Integrated Telephone Network," IEEE Trans. on Commun., June 1974.

[12] M.R. Miller, "Feasibility Studies of Synchronized Oscillator Systems for PCM Telephone Networks," Proc. IEE, July 1969.

[13] I.W. Sandberg, "Some Properties of a Nonlinear Model of a System for Synchronizing Digital Transmission Networks," Bell Syst. Tech. J., pp. 2975-2997, Nov. 1969.

[14] M. Karnaugh, "A Model for the Organic Synchronization of Communication Systems," Bell Syst. Tech. J., pp. 1705-1735, Dec. 1966.

[15] J.P. Moreland, "Performance of a System of Mutually Synchronized Clocks," Bell Syst. Tech. J., pp. 2449-2465, Sept. 1971.

[16] B.R. Saltzberg and H.M. Zydney, "Digital Data System: Network Synchronization," Bell Syst. Tech. J., pp. 879-892, May 1975.

[17] A. Datta and E. Mitwally, "Synchronization of Multi-Exchange Local Network by Bit by Bit Method," IEEE Trans. on Commun., July 1979.

[18] C.A. Cooper, "Synchronization for Telecommunications in a Switched Digital Network," IEEE Trans. on Commun., July 1979.

[19] E.A. Harrington, "Issues in Terrestrial/Satellite Network Synchronization," IEEE Trans. on Commun., pp. 1690-1696, Nov. 1979.

[20] R.A. Boulter and W. Bunn, "Network Synchronization," POEEJ, April 1977.

[21] H.L. Hartmann, "The Single Ended Principle of Digital Phase Averaging for Synchronization of Communication Networks," NTZ, pp. 421-426, Vol.28, No. 12, 1975.

[22] A.C. Davies, "The Effect of Clock Drift Upon The Synchronization of Digital Communication Networks," IEEE Trans. on Commun., pp. 1842-1844, Nov. 1974.

[23] A. Marlevi, "Direct Digital Controlled Network Synchronization," IEEE Trans. on Commun., Oct. 1979.

[24] J.E. Abate, L.H. Brandenburg, J.C. Lawson and W.L. Ross, "The Switched Digital Network Plan," Bell Syst. Tech. J., pp. 1297-1320, Sept. 1977.

[25] CCITT Rec. G.811, "Plesiochronous Operation of International Digital Links," Orange Book, Vol. III-2, pp. 485-488.

[26] G.P. Darwin and R.C. Prim, "Synchronization of a System of Interconnected Units," USA Patent 2,986,723, May 1961.

[27] J. Kella, Unpublished Work, 1977.

[28] W.R. Bennett, "Statistics of Regenerative Digital Transmission," Bell Syst. Tech. J., pp. 1501-1542, Nov. 1958.

[29] J.P. Odenwalder, "Optimal Decoding of Convolutional Codes," Ph.D. Dissertation, Dep. Syst. Sci. Sch. Eng. Appl. Sci. Univ. California, Los Angeles, 1970.

[30] E. Paaske, "Short Binary Convolutional Codes with Maximal Free Distance for Rates 2/3 and 3/4," IEEE Trans. Inform. Theory, pp. 683-689, Sept. 1974.

[31] R. Ash, Information Theory. New York: Wiley, 1965.

[32] W. Feller, An Introduction to Probability Theory and Its Applications. New York: Wiley, 1968.

[33] F.B. Johnson, "Calculating Delta Modulator Performance," IEEE Trans. Audio Electroacoust., pp. 121-129, March 1968.

[34] J.K. Wolf, "Effects of Channel Errors on Delta Modulation," IEEE Trans. Commun. Tech., pp. 2-7, Feb. 1966.

[35] N.L. Yates-Fish and E. Fitch, "Signal to Noise Ratio in Pulse Code Modulation," Instn. Elec. Engrs. Proc., pp. 204-210, March 1955.

[36] W.R. Bennett and S.O. Rice, "Spectral Density and Autocorrelation Functions Associated with Binary Frequency Shift Keying," Bell Syst. Tech. J., Sept. 1963.

[37] A.R. Figueiras-Vidal, J.B. Marino-Acebal and M.A. Lagunas-Hernandez, "Comments and Extentions of Wolf's Signal to Channel Noise Formulas for Delta Modulation Systems," IEEE Trans. on Commun., pp. 131-135, Jan. 1980.

[38] N.S. Jayant, "Step-Size Transmitting Differential Codecs for Mobile Telephony," Bell Syst. Tech. J., Nov. 1975.

[39] N. Deo, Graph Theory with Applications to Engineering and Computer Science. New Jersey: Prentice Hall, 1974.

[40] R.V. Churchill, Operational Mathematics. New York: Mcgraw Hill, 1958.

[41] V.R. Dhadesugoor, C. Ziegler, and D.L. Schilling, "Delta Modulators in Packet Voice Networks," IEEE Trans. on Commun. pp. 33-51, Jan. 1980.