SYNCHRONIZATION OF ULTRA-WIDEBAND SIGNALS
IN THE DENSE MULTIPATH CHANNEL

by

Eric A. Homier

A Dissertation Presented to the
FACULTY OF THE GRADUATE SCHOOL
UNIVERSITY OF SOUTHERN CALIFORNIA
In Partial Fulfillment of the
Requirements for the Degree
DOCTOR OF PHILOSOPHY
(ELECTRICAL ENGINEERING)

December 2004

Dedication

To my father, Jay, for his constant support and encouragement, as well as all my friends and family.

Acknowledgements

Firstly I would like to thank my advisor, Prof. Robert Scholtz, for all his help, suggestions, and advice regarding this thesis and the research that went into generating it. It has truly been a great privilege to work with him. I would also like to thank Prof. Urbashi Mitra of the Electrical Engineering Department and Prof. Larry Goldstein of the Mathematics Department for helping to review this thesis and for providing useful feedback. Additionally, I am thankful to Prof. Charles Weber and Prof. Keith Chugg of the Electrical Engineering Department for their help in reviewing my initial research.

Many thanks, also, go to the staff of the Communications Sciences Institute, specifically Milly Montenegro, Mayumi Thrasher, and Gerrielyn Ramos for all their help. I would like to thank the staff of the Electrical Engineering department, in particular, Diane Demetras for all her help. I would like to thank all of my colleagues at the Ultra-Wideband Radio Laboratory for their comments, suggestions, and insightful discussions regarding our various research efforts. All of these discussions, as well as, all of the various experiments we conducted greatly added to my understanding of ultra-wideband systems.

I am indebted to Northrop Grumman Space Technology (formerly TRW, Inc.) for their support, both financially through their fellowship program, and through the people I've encountered. Here, academic endeavors and graduate research are highly encouraged and supported. I would like to thank Linda Mastropaolo for all her help with the administrative details of the fellowship program. For all their help and support I would like to thank Tom Zeiller, Gina Keller, Renee Yazdi, Nick DiCamillo, Barry Sitek, Jeff Chou, Reggie Jue, Larry Clouse, and Kiet Ngo. I would like to thank Robert Golshan for all his advice about the doctoral program, for all the wonderful discussions about our various research efforts, and mostly for being a wonderful friend. His help has been invaluable. Thanks, also, go to Jean-Marc Cramer for all his advice along the way. Finally, I would like to thank Nick Parsons for reviewing some of this research, for the interesting discussions that resulted, and most importantly for being such a great friend.

Finally, I need to thank all my friends and family, although there are far too many to list out completely. I would like to thank Diane Heck for her support through this endeavor. Many thanks to my father, Jay Homier, for setting me on the path at an early age that has ultimately led me to this point. I would also like to thank my mother, Linda Wehr, and my step-mother, Darlene Homier. I am also indebted to my grandparents, Richard and Annie Homier, for all their love and support over the years, and Joe and Brenda Wehr, as well as Joyce Wehr. Finally, I would like to thank Tina Hsu for being so understanding and supportive and for all the happiness she has brought me.

Contents

List of Tables

List of Figures

Abstract

As data rates in wireless communications systems continue to increase, as networks become more populated with users, and as these networks of communications nodes are operated in increasingly harsh environments such as indoor and densely populated urban areas, the effects of multipath will become increasingly significant. Therefore a thorough understanding of its effects on communication systems will be necessary. One key element required in communication systems is synchronization, which produces alignment of transmitter and receiver clocks so that information can be accurately exchanged. Here the process of synchronization is studied in the presence of dense multipath for ultra-wideband (UWB) signals. The analytic framework developed for this purpose is applicable to a number of different problems involving synchronization in multipath and is not restricted to the case of the ultra-wideband signals considered herein. In fact, these analysis techniques are applicable to any problem involving a group of observers searching for a group of objects in some arbitrary fashion.

The mathematical framework developed here for search analysis is based upon graphical techniques. Specifically, a generalized signal flow graph is introduced which

is well suited for acquisition in multipath. This graph produces a complete statistical description of the search time, as well as the acquisition probability. Hybrid serial/parallel acquisition of UWB signals, as well as sequential detection schemes, can also be examined using these graphical techniques. Efficient search permutations are found for hybrid acquisition in multipath, specifically the bit reversal search is introduced. This search permuation significantly reduces the mean acquisition time in the presence of multipath without any additional complexity versus traditional linear search permutations. The process of acquisition is divided into two main parts, coarse acquisition followed by fine acquisition. The coarse acquisition process attempts to locate the group of paths which, because of the multipath channel, tend to cluster together. Fine acquisition attempts to locate the strongest paths within the multipath cluster of arriving paths, effectively combining the processes of verification and channel estimation. A combined graphical structure, termed a self-similar signal flow graph, has been developed to study the combined coarse/fine acquisition process.

Chapter 1

Introductory Material on UWB Signals and Systems

This chapter provides an overview of several key concepts in the area of communication systems employing Ultra-Wideband (UWB) signals in a wireless dense multipath channel. A definition of general UWB signals is given, along with the specific signal waveform to be considered. Also described in this chapter is a statistical model of the UWB multipath channel.

1.1 An Overview of Ultra-Wideband Signals

Ultra-Wideband (UWB) signals, those with large fractional bandwidths, are currently being investigated for use in communications systems where an advantage over more narrowband signals exists. These advantages are listed in a number of references and specifically include improved penetration through materials [29] as well as improved performance in dense multipath environments [64] where the UWB signals can be resolved in time making the use of a RAKE receiver possible [37]. Both of these advantages make UWB communication systems well suited for urban and indoor wireless applications where many local objects act as scatterers and absorbers of the transmitted electromagnetic energy. Also, these specific advantages allow for reduced transmitted signal power, which in turn result in low probability of detection or interception (LPD/LPI). To specifically define what is meant by an Ultra-Wideband signal, the following fractional bandwidth definition is employed:

$$B_f = 2\frac{f_H - f_L}{f_H + f_L} \tag{1.1}$$

where f_L and f_H are the lower and upper end (3 dB points) of the signal spectrum, respectively. UWB signals are then those signals that have a fractional bandwidth greater than 25 percent [33]. Narrowband signals are defined as those signals with fractional bandwidths less than 1 percent, while wideband signals are between 1 and 25 percent [57].

There are many conceivable signals which will have the required fractional bandwidth to be termed UWB signals. Specifically in this document, the signal choice for the UWB signal is a baseband pulse that is shaped as the 2^{nd} derivative of a Guassian pulse. This pulse derives from the application of a Gaussian pulse to the antenna. The electromagnetic wave radiated by an antenna is proportional to the time derivative of the antenna's driving current [57] while an additional derivative results from the receive antenna. In more narrowband systems employing carriers, this derivative is well approximated as a time-shift. The 2^{nd} derivative pulse shape is defined as

$$p(t) = \sqrt{\frac{4}{3\sigma\sqrt{\pi}}}\left(1-\left(\frac{t}{\sigma}\right)^2\right)\exp\left(-\frac{1}{2}\left(\frac{t}{\sigma}\right)^2\right) \tag{1.2}$$

The factor $\sqrt{4/(3\sigma\sqrt{\pi})}$ ensures that the signal is normalized to unit energy, i.e.,

$$\int_{-\infty}^{\infty} p^2(t)\,\mathrm{d}t = 1 \tag{1.3}$$

This allows all the energy in the received waveform to be stated explicitly, that is, the received energy in $\sqrt{E_p}\cdot p(t)$ is simply E_p. The scale factor, σ, determines the effective time width of the pulse shape and will be considered approximately $(2\sqrt{\pi})^{-1}\cdot 1$ nsec, resulting in an effective width on the order of one nanosecond. The pulse shape in (1.2) can be considered as the transmitted pulse shape by lumping both derivatives at the transmitter end of the system. This propagation model is very simplistic, but will suffice for the present purpose. For detailed propagation studies of UWB signals see [1] or [8] and the references therein.

1.2 UWB Modulation and Multiple Access Format

There are many possible modulation formats that can be considered when using Ultra-Wideband signals. An example of one such modulation format is time-hopped pulse position modulation on a per frame basis [46]:

$$x^{(k)}(t) = \sqrt{E_p^{(k)}} \cdot \sum_n p\left(t - nT_f - c_n^{(k)}T_c - \tfrac{\delta}{2}d_n^{(k)}\right) \tag{1.4}$$

Figure 1.1 below should help to explain this modulation format. The superscript (k) represents the k^{th} user in a multiple user system. The frame time, T_f, is the reciprocal of the pulse repetition frequency (PRF) and must be large enough to allow sufficient room for time-hopping and data modulation. The multipath channel also places a lower limit on the frame time, to be discussed below. In order to reduce the number of collisions in the multi-user system, each user has a unique time hopping code sequence, $\{c_n^{(k)}\}$, where each element in the sequence is an integer between 0 and N_g. The smallest time shift associated with successive elements of the time-hopping code is determined by the chip time, T_c. In order to prevent energy from spilling into the next frame when multipath is present, the maximum time shift, termed the guard time and given as $N_g \cdot T_c$, must be sufficiently small with respect to the frame time. The data symbol for the k^{th} user is $d_n^{(k)} \in \{\pm 1\}$. The difference between the two possible data locations is δ, and in general the decision error rate will be a function of this parameter so that an optimum value can be determined in the sense of minimum

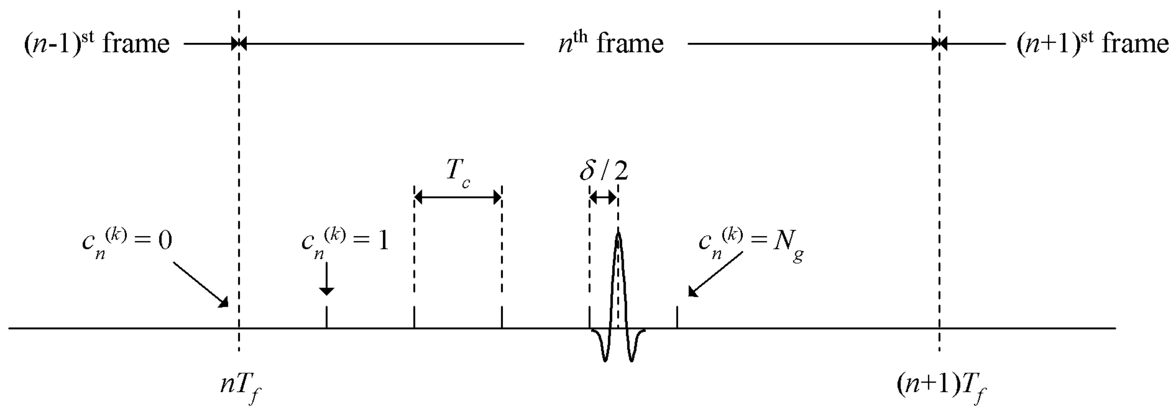

Figure 1.1: The n^{th} frame of $x^{(k)}(t)$ for $c_n^{(k)} = 4$, $N_g = 5$, and $d_n^{(k)} = +1$

decision error probability. The data modulation sequence may contain some form of error-correcting code in order to decrease this decision error probability, one example being to simply repeat the same data for some prescribed number of successive frames [46]. Other forms of data modulation are also possible [38],[39],[40],[41],[65].

1.3 UWB Multipath Channel

The dense multipath wireless channel has been examined for the urban environment [55], [59], the indoor environment [45], [50], and also for the UWB impulse radio channel [9], [63]. A common model for the impulse response of the dense multipath channel, first introduced in [45], is the clustering model. This model is based upon observations from experimental data where it was noticed that rays tended to arrive in closely spaced groups, or clusters. The inter-arrival times of the rays within a cluster are exponentially distributed, as are the cluster inter-arrival times, giving rise to a double Poisson arrival process. The amplitude of each ray can be either positive or negative with the magnitude being Rayleigh distributed with a mean-square value which decays with increasing ray and cluster arrival time.

$$h(t) = \sum_{l=0}^{\infty} \sum_{k=0}^{\infty} a_{kl} \delta \left(t - T_l - \tau_{kl}\right) \tag{1.5}$$

The amplitude of the k^{th} ray of the l^{th} cluster, a_{kl}, is the product of an equilikely random ± 1 with a Rayleigh random variable β_{kl} where the mean square values of these Rayleigh random variables are exponentially decaying functions of the arrival times, T_l and τ_{kl}:

$$E\left(\beta_{kl}^2\right) = E\left(\beta_{00}^2\right) e^{-T_l/\Gamma} e^{-\tau_{kl}/\gamma} \tag{1.6}$$

Here $E\left(\beta_{00}^2\right)$ is the mean square value of the first ray of the first cluster and is determined by the path loss that exists between the transmitter and receiver. This path loss is determined by the physical distance, d, between the transmitter and receiver and assumes that the received power is of the form [54]:

$$P_{(dB)}(d) = P_{(dB)}(d_0) - 10\beta \log_{10}(d/d_0) + \epsilon_{(dB)} \tag{1.7}$$

Here β is the path loss exponent which determines the rate at which the received signal amplitude decreases with distance and in free space, β is 2. The path loss exponent for the propagation experiment described in [63] was determined to be approximately 1.75 in [9]. Also, d_0 is the distance at which a reference measurement of the received signal amplitude is made and is typically 1 m for indoor environments. An explicit relationship between $E(\beta_{00}^2)$ and the path loss is given in [45]. As stated in [54], the $\epsilon_{(dB)}$ term is a zero mean, Gaussian random variable (in dB) which represents any measurement error in the path loss and arises because of shadowing.

As was mentioned above, the cluster inter-arrival time, $\Delta T_l = T_l - T_{l-1}$, and the ray inter-arrival time, $\Delta\tau_{kl} = \tau_{kl} - \tau_{(k-1)l}$, are each exponentially distributed with probability density functions:

$$f(\Delta T_l) = \Lambda \exp\left(-\Lambda \cdot \Delta T_l\right) \tag{1.8}$$

$$f(\Delta\tau_{kl}) = \lambda \exp\left(-\lambda \cdot \Delta\tau_{kl}\right) \tag{1.9}$$

As with the amplitude of the first arrival in the first cluster, β_{00}, the arrival time of this first path, τ_{00} must be explicitly given. If the distance between the transmitter and receiver is known, then τ_{00} can be computed. This is usually not the case, however, as this distance is not known, along with the location, number, and composition of obstacles between the transmitter and receiver. For the case of the pulsed UWB waveform of (1.4) the periodic nature of the signal will cause τ_{00} to be uniformly distributed over a code period, $[0, N_c \cdot T_f)$.

The impulse response in (1.5) can also be represented as a single summation by a one-to-one mapping of the amplitude coefficients, a_{kl}, into a new set of coefficients, a_m. Likewise the arrival time, $T_l + \tau_{kl}$, can be mapped into a new arrival time,

τ_m. It will be assumed that this mapping occurs so that the path arrival times are strictly increasing in their indices, that is $\tau_0 < \tau_1 < \tau_2 < \ldots$ where the inequality is strict because it is assumed that multiple paths arriving at the same time are lumped together as a single path. This yields a slightly simpler model which will be used in later sections of this document:

$$h(t) = \sum_{m=0}^{\infty} a_m \delta\left(t - \tau_m\right) \tag{1.10}$$

The models in (1.5) and (1.10) are known as specular multipath models. They assume that the effect of the channel is simply to sum up many scaled and time-shifted versions of the original transmitted pulse, i.e., there is no pulse waveform distortion. The appropriate multipath model when considering waveform distortion is the diffuse model. Such distortions can occur from the diffraction of electromagnetic waves around objects, the non-plane wave nature of these waves at near field distances, or when transmitter or receiver motion is involved. The diffuse model can be thought of in a couple different ways. First, the same form as (1.10) with the summation taken over an uncountable set can be considered. Secondly, the summation can remain countable with the output pulse waveform becoming a function of the index m. Of course, a combination of these two models might also be conceivable. The diffuse model is more accurate but comes with increased complexity. For the purposes of this work, the specular model will suffice as many observed channel response waveforms can be adequately modeled as such [9], [13], [63].

A typical waveform from the output of a dense multipath channel excited by a pulse shape as in (1.2) is shown in Figures 1.2 and 1.3. The clustering phenomenon is evident in Figure 1.2 as is the overall decaying nature with time in Figure 1.3. Both figures are of the same data but with different time axes. The source of this data is a UWB propagation experiment in an indoor office environment as described in [63]. In fact, the worst case multipath channel (Channel Model 4) in the IEEE 802.15.SG3a channel model final report [13] produces single realizations which are comparable to the one seen here in Figures 1.2 and 1.3. There exist several methods of experimentally determining the nature of the multipath channel as described in [42]. One possible method as discussed in [42] and employed in [63] is the direct pulse method whereby a narrow pulse in time is repeatedly transmitted over the channel with the received waveform at the receiver stored in a digital oscilloscope. Another method described in [42] is that of frequency domain channel sounding where a network analyzer is stepped through a discrete set of frequencies and the overall frequency response of the channel is determined. Both methods provide similar results when compared against one another as discussed in [60]. As an example of this, Figures 1.4 and 1.5 are provided which were taken from a UWB propagation experiment performed in the cargo hold aboard a Navy cargo ship. The first of these figures, 1.4, contains a direct pulse measurement as well as the inverse Fourier Transform of a frequency domain channel sounding measurement taken at the same transmit and receive locations. Notice that the measurements are very closely correlated giving some validation that the two channel measurement techniques provide similar results about the nature of the channel. The data collected from these channel sounding techniques can be used to generate estimates of the specular multipath parameters in (1.10). This procedure is discussed below in Section 1.4.

The square of the amplitude in either Figure 1.2, 1.3, 1.4, or 1.5 is known as the power delay profile. A single statistical parameter which quantifies the extent of the multipath signal in time is the rms delay spread, defined as the square root of the second central moment of the power delay profile [42]. Notice that the delay spread of Figure 1.3 is much shorter than the delay spread of Figure 1.5. This is due to the fact that the measurements were taken in significantly different environments.

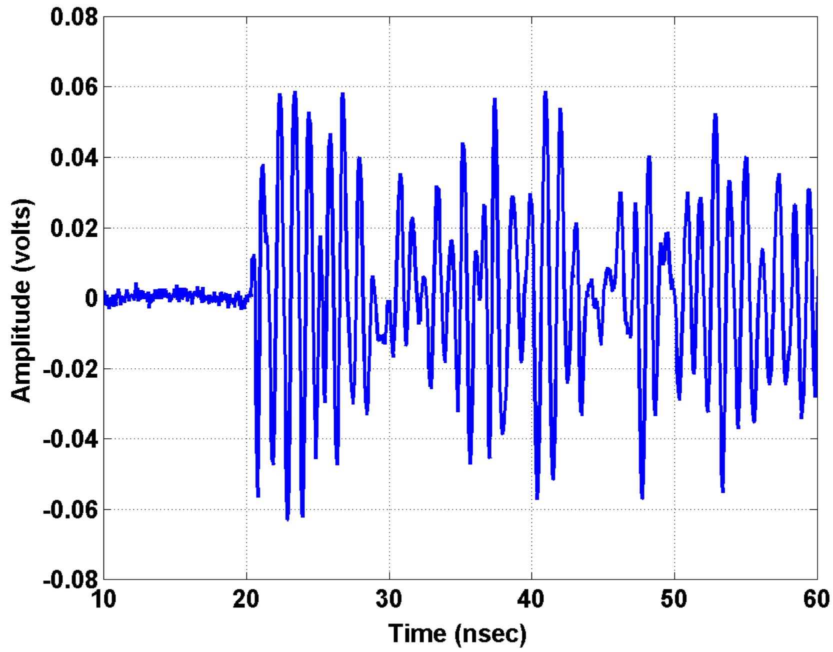

Figure 1.2: Output of a dense multipath channel excited by the 2nd derivative Gaussian pulse. Experimental data taken from [63].

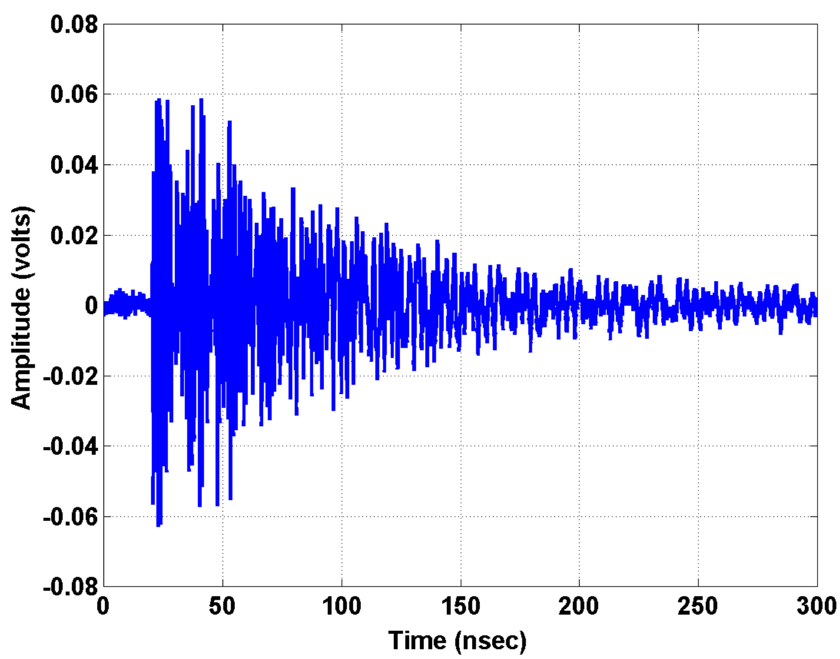

Figure 1.3: Output of a dense multipath channel excited by the 2nd derivative Gaussian pulse (Larger time axis). Experimental data taken from [63].

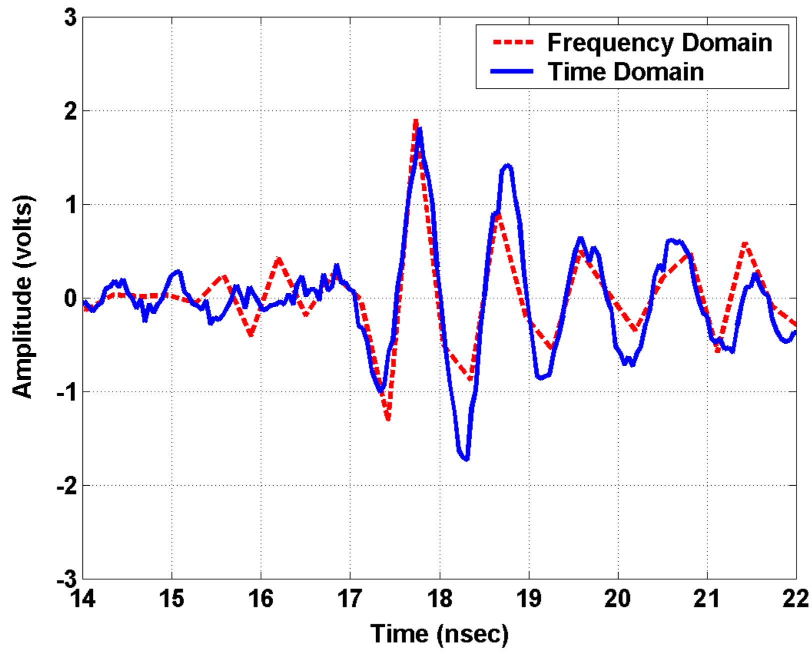

Figure 1.4: Output of a dense multipath channel excited by the 2nd derivative Gaussian pulse. Experimental data taken aboard Navy cargo ship. Direct pulse (time domain) measurement and frequency domain channel sounding measurement both shown.

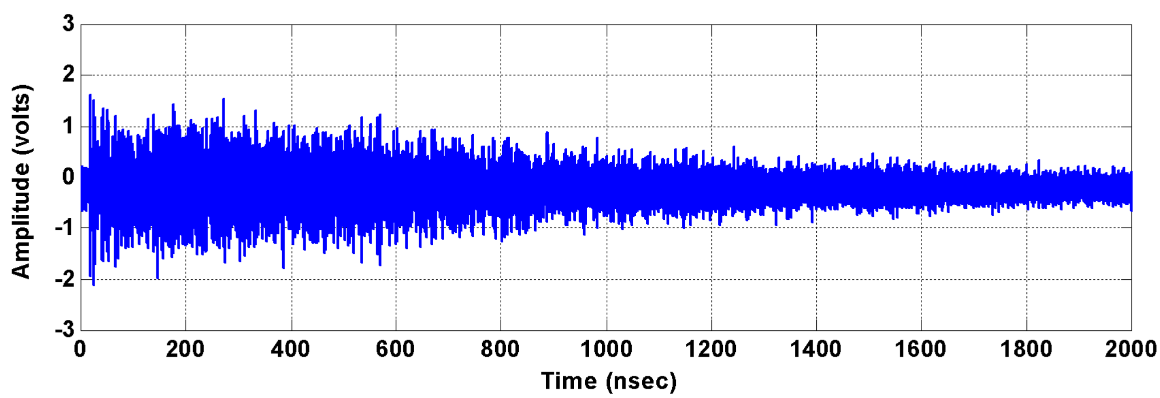

Figure 1.5: Output of a dense multipath channel excited by the 2nd derivative Gaussian pulse (Larger time axis). Experimental data taken aboard Navy cargo ship. Only the direct pulse measurement is shown.

The time varying nature of the channel is described by a parameter known as the coherence time [42] and is a measure of the time duration over which the channel remains statistically unchanged. Changes in the channel can result from one of two sources. First, if either the receiver or transmitter is moving, such as in the mobile channel, then the statistics of the multipath channel will change as a function of time. The a_m and τ_m terms in (1.10) will then become non-stationary and their distributions will vary with time. The degree with which these parameters change in time depends on the velocity involved relative to the coherence time. Secondly, objects in the local environment can move, e.g., people, vehicles, etc. Such movements of local scatterers will cause slight variations in the multipath channel.

It is known that the local environment of scatterers and absorbers determines the impulse response of the multipath channel. For this reason it would be expected that for a fixed transmitter location, two receiver locations which are very close to one another would have similar impulse responses. Likewise, if the receiver locations were far apart, drastically different impulse responses for the channel would be expected. The measure of similarity of the channel impulse response for any two locations is given by the spatial correlation function of the channel. For the purposes of this work, the users will be considered far enough apart so that the channels are statistically uncorrelated.

1.4 Specular Multipath Channel Estimation

This section describes the procedure used to estimate the a_m and τ_m parameters of the multipath channel impulse response in (1.10). The method described here is a form of subtractive deconvolution, also known as the CLEAN algorithm, as described in [9], [18], and [60].

The received signal from one of the channel sounding measurements discussed in the previous section will be denoted as $r(t)$. The signal estimate will be denoted as $\hat{r}(t)$ and will assume the following form:

$$\hat{r}(t) = \sum_{m=1}^{M} \hat{a}_m p\left(t - \hat{\tau}_m\right) \tag{1.11}$$

The UWB pulse, $p(t)$, is given in (1.2) and thus an estimate of the pulse width parameter, σ, will be required. One method of determining this estimate, $\hat{\sigma}$, is to select the value which maximizes the correlation between the received signal and the UWB pulse:

$$\hat{\sigma} = \arg \max_{(\tau,\sigma)} |R_{rp}(\tau)| \tag{1.12}$$

where $p(t)$ is a function of σ so that the correlation function, $R_{rp}(\tau)$, is also a function of σ. This correlation function will be an essential part of the channel estimation algorithm described below and is given explicitly as:

$$R_{rp}(\tau) = \int_{-\infty}^{\infty} r(t)p(t-\tau)\mathrm{d}t \tag{1.13}$$

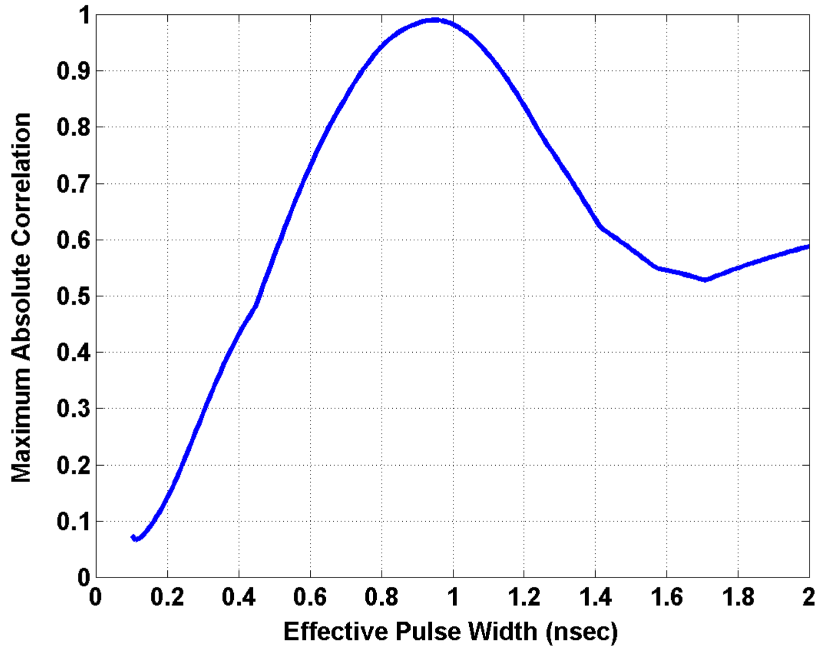

Figure 1.6: Maximum absolute correlation, $\max_\tau |R_{rp}(\tau)|$, versus the effective pulse width of $p(t)$. ($r(t)$ from Figure 1.2)

For the measurement shown in Figure 1.2, the function $\max_\tau |R_{rp}(\tau)|$ is plotted below in Figure 1.6. As mentioned earlier, the effective pulse width of $p(t)$ is $(2\sqrt{\pi}) \cdot \sigma$, and from Figure 1.6 is found to be 0.95 nsec.

The subtractive deconvolution technique based upon the CLEAN algorithm works on a *dirty map* of the received signal and produces a *clean map* of the estimated channel impulse response. The term *dirty map* is from [18] which referred to measurements taken on an array of interferometers which were distorted by the array sidelobes. The *clean map* was the resultant array output after removing the sidelobe affects, i.e., after 'cleaning' the received signal. At the n^{th} algorithm iteration, the dirty and clean maps will be denoted $d_n(t)$ and $c_n(t)$, respectively. The algorithm, given as follows, will be allowed to iterate until $n = N$.

1. Initialize the dirty map to $d_0(t) = r(t)$ and the clean map to $c_0(t) = 0$. Compute the UWB pulse energy, $R_{pp}(0)$. Initialize $n = 1$.

2. Compute the 'normalized' correlation between $d_{n-1}(t)$ and $p(t)$ as $f_{n-1}(\tau) = R_{pp}^{-1}(0)R_{d_{n-1}p}(\tau)$.

3. Compute $\hat{\tau}_n = \arg\max_\tau |f_{n-1}(\tau)|$ and $\hat{a}_n = f_{n-1}(\hat{\tau}_n)$.

4. Update the dirty map, $d_n(t) = d_{n-1}(t) - \hat{a}_n p(t - \hat{\tau}_n)$.

5. Update the clean map, $c_n(t) = c_{n-1}(t) + \hat{a}_n \delta(t - \hat{\tau}_n)$.

6. If $n = N$ then proceed to the next step, otherwise iterate, $n = n + 1$, and proceed to step (2).

7. The estimate of the channel impulse response is $\hat{h}(t) = c(t)$ and the received signal estimate is $\hat{r}(t) = \hat{h}(t) * p(t)$ where $*$ represents the convolution operator.

Although the UWB pulse in (1.2) is normalized to have unit energy, i.e., $R_{pp}(0) = 1$, the algorithm outlined above is given in its most general form for arbitrary $p(t)$. It should also be noted that at any given iteration, the possible location of $\hat{\tau}_n$ is unrestricted. Thus, it is possible for $\hat{\tau}_n$ to be exactly equal to $\hat{\tau}_k$ for $k \neq n$, in which case the amplitude estimate for that arrival time is simply the sum, $\hat{a}_n + \hat{a}_k$.

Two possible measures of estimator quality for the above algorithm are the normalized mean-square error and the fractional energy captured, both as functions of the iteration number, n. The normalized mean-square error at the n^{th} iteration, ϵ_n, is defined as $R_{rr}^{-1}(0) \cdot R_{\tilde{r}\tilde{r}}(0)$ where $\tilde{r}(t) = r(t) - \hat{r}(t)$. Written explicitly, this quantity becomes

$$\epsilon_n = \frac{\int_{-\infty}^{\infty}(r(t) - \sum_{m=1}^{n} \hat{a}_m p(t - \hat{\tau}_m))^2 \mathrm{d}t}{\int_{-\infty}^{\infty} r^2(t)\mathrm{d}t} \tag{1.14}$$

The fractional energy captured at each iteration, ϕ_n, is determined by the amount of energy removed from the dirty map at the previous iteration:

$$\phi_n = 1 - \frac{\int_{-\infty}^{\infty} d_n^2(t)\mathrm{d}t}{\int_{-\infty}^{\infty} r^2(t)\mathrm{d}t} \tag{1.15}$$

For $N = 300$ with an effective pulse width of 0.95 nsec as determined above, the algorithm results for the data given in Figure 1.2 are shown below. Figure 1.7 shows a portion of the original data and the reconstruction, $\hat{r}(t)$, while Figures 1.8 and 1.9 show the normalized mean-square error and the fractional energy captured, respectively, as just defined.

For the data in Figure 1.2, after 300 iterations of the algorithm, 296 distinct path arrival estimates were determined, indicating that anywhere from 4 to 8 iterations produced non unique path arrivals. Since almost all 300 iterations produced a unique arrival path, each iteration can be viewed as adding another unique separate path to a RAKE receiver, for example. Thus the plot in Figure 1.9 gives roughly the number of correlators required in such a RAKE receiver for a prescribed SNR degradation (determined as $10 \log \phi_n$). For roughly 300 correlators the total energy captured in $r(t)$ is 85%, resulting in an SNR degradation of 0.7 dB. Note that this degradation

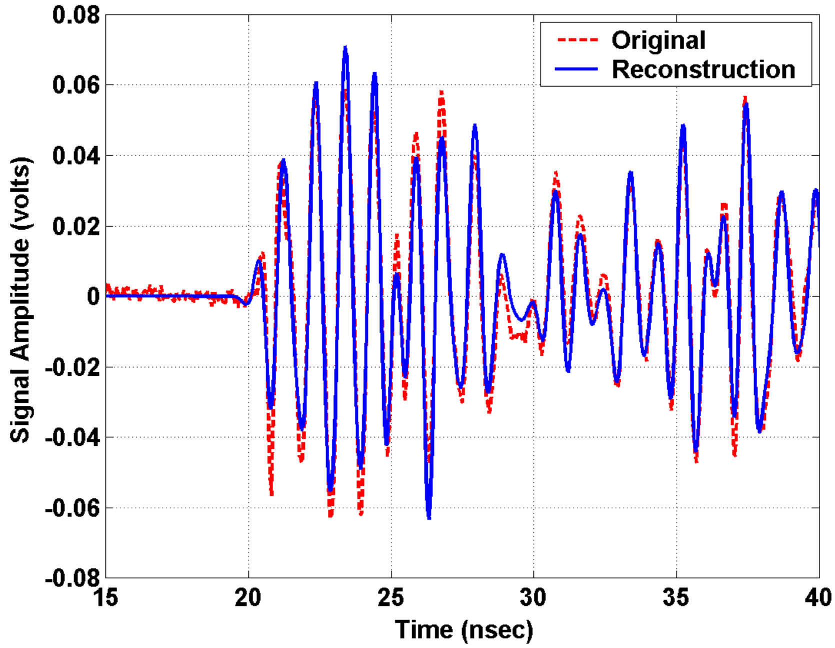

Figure 1.7: Original data from Figure 1.2 and reconstructed waveform

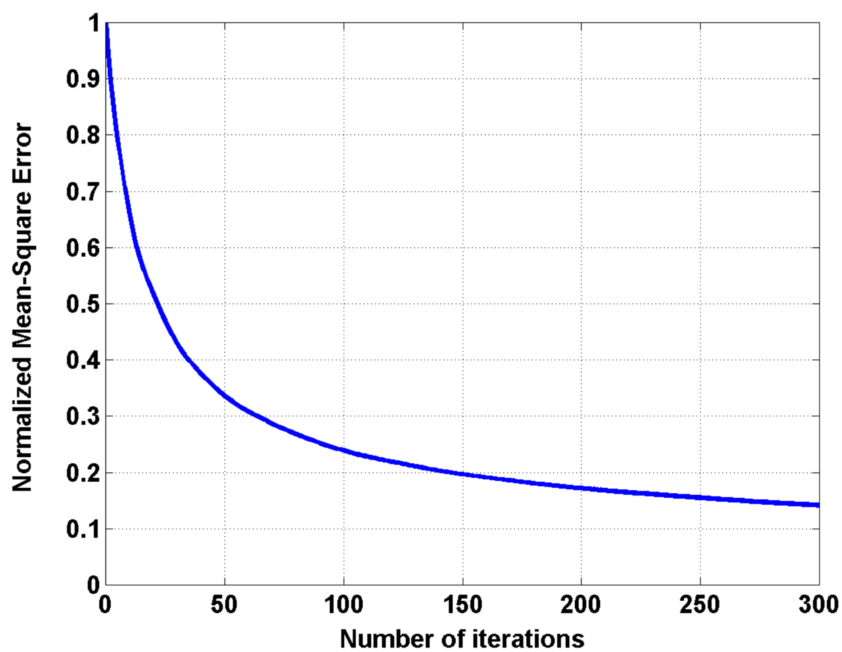

Figure 1.8: Normalized mean-square error for data in Figure 1.2

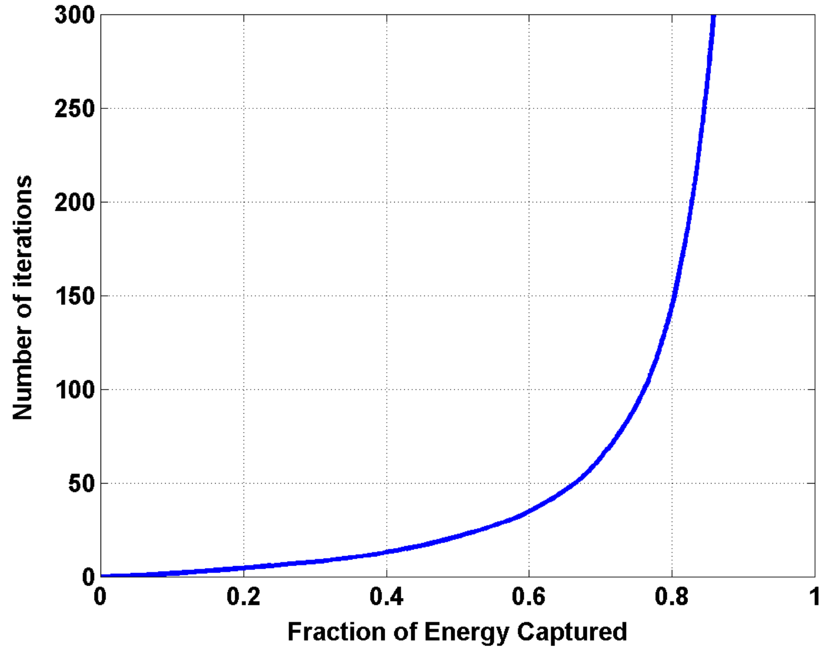

Figure 1.9: Energy capture curve for data in Figure 1.2

is merely an approximation since the received signal, $r(t)$, has been corrupted by noise (the data of Figure 1.2 is seen to be a relatively high SNR case so that this approximation is close).

As was mentioned above, the received data, $r(t)$, has passed through the multipath channel which, in actuality, is not purely specular. Also, the received signal has been corrupted by noise. These are the two primary reasons that the energy capture curve will never reach one (100% of the received energy accounted for with a specular model) and the normalized mean-square error curve will never reach zero.

A more efficient algorithm can be implemented which is identical to the one outlined above. Specifically, the number of operations required at each iteration can be substantially reduced by removing the correlation. This can be done by directly ‘cleaning’ the original correlation function. To see this, note that at each step the correlation function, $R_{d_{n-1}p}(\tau)$, is computed. However, this correlation can be determined recursively from the previous correlation function:

$$R_{d_{n-1}p}(\tau) = \int_{-\infty}^{\infty} d_{n-1}(t)p(t-\tau)\mathrm{d}t \tag{1.16}$$

$$= \int_{-\infty}^{\infty} (d_{n-2}(t) - \hat{a}_{n-1}p(t-\hat{\tau}_{n-1}))p(t-\tau)\mathrm{d}t \tag{1.17}$$

$$= R_{d_{n-2}p}(\tau) - \hat{a}_{n-1}R_{pp}(\tau - \hat{\tau}_{n-1}) \tag{1.18}$$

Thus, the only correlations required, $R_{pp}(\tau)$ and $R_{d_0p}(\tau) = R_{rp}(\tau)$, can be done outside the iteration loop. The updated algorithm is given as:

1. Initialize the ‘normalized’ correlation between $r(t)$ and $p(t)$ as $f_0(\tau) = R_{pp}^{-1}(0)R_{rp}(\tau)$ and the ‘normalized’ correlation of $p(t)$ with itself as $g(\tau) = R_{pp}^{-1}(0)R_{pp}(\tau)$. Initialize $n = 1$.

2. Compute $\hat{\tau}_n = \arg\max_\tau |f_{n-1}(\tau)|$ and $\hat{a}_n = f_{n-1}(\hat{\tau}_n)$.

3. Update the correlation function, $f_n(\tau) = f_{n-1}(\tau) - \hat{a}_n g(\tau - \hat{\tau}_n)$.

4. If $n = N$ then proceed to the next step, otherwise iterate, $n = n + 1$, and proceed to step (2).

5. The estimate of the channel impulse response is $\hat{h}(t) = \sum_{n=1}^{N} \hat{a}_n \delta(t - \hat{\tau}_n)$ and the received signal estimate is $\hat{r}(t) = \hat{h}(t) * p(t)$ where $*$ represents the convolution operator.

1.5 Single User Receiver Structure

The receiver structure to be considered here will simply be the matched filter, or equivalently, the correlator receiver. The optimum receiver structure in the presence of specular multipath has been determined as the RAKE receiver [37]. Instead of matching to the user's transmitted pulse as in (1.2), the RAKE receiver matches to the multipath channel output. In light of the fact that the multipath channel output is simply the sum of scaled, time-shifted versions of the pulse shape in (1.2), a correlator based receiver seems more attractive from a hardware implementation point of view. One additional advantage to the correlator based receiver versus the matched filter comes when a sub-optimum, sinusoidal local template waveform is considered. A sinusoidal template waveform has an advantage over the optimal template waveform of (1.2) in that it is much easier to generate locally at the receiver. The degradation from using such a template is shown below to be small, around 0.8 dB. This makes such a sub-optimum correlator receiver quite feasible. Such structures are currently being investigated [28]. Also shown below are several other attractive features of such a correlator design.

Figure 1.10 shows a correlator receiver for a single path without the frame time, time-hopping, or data modulation. The template waveform is $v(t)$. Because of the linear nature of the operations involved, the RAKE receiver can be implemented by working on the outputs of each correlator (Z in Figure 1.10) or by using a single integrator and correlating the receiver input directly with the multipath channel output. This means that $v(t)$ in Figure 1.10 can be either $p(t)$ as in (1.2) or it can be $p(t)*h(t)$

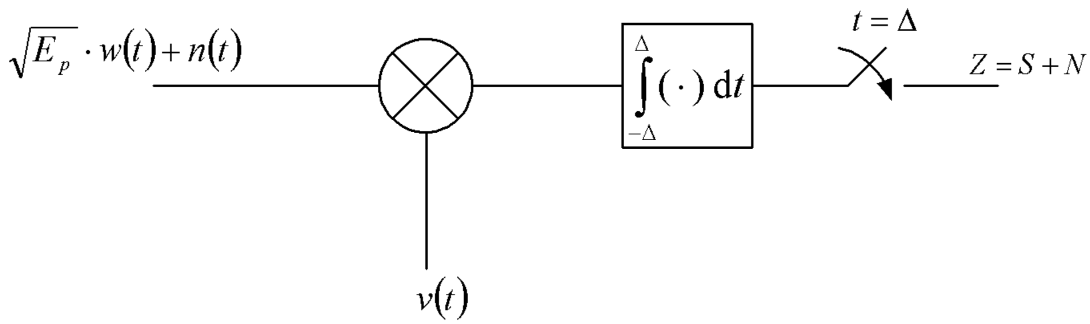

Figure 1.10: Single user correlator receiver

where $h(t)$ is the multipath channel impulse response of (1.10) and $*$ represents the convolution operator. In practice only a finite number of correlators can be built in the receiver giving rise to a suboptimum RAKE receiver structure known as the selective RAKE receiver. Energy capture curves show that a fairly large number of correlators are required to capture a significant portion of the received signal energy, e.g., for 50% energy capture roughly 25 correlators are needed and for 80% energy capture roughly 200 correlators are needed for the indoor environment as shown in [9], [63], and Section 1.4.

A general expression for the SNR at the output of the Ultra-Wideband correlator in Figure 1.10 is now examined. The results obtained will also be valid for a pulsed system with time-hopping and multipath, provided that the pulses are transmitted far enough apart so they don't overlap and the limits of integration and sampling instants are updated accordingly for each pulse or path.

The signal portion of the correlator output, S, is easily computed as

$$S = \sqrt{E_p} \cdot \int_{-\Delta}^{\Delta} w(t) \cdot v(t - \tau)\mathrm{d}t \tag{1.19}$$

Here the template waveform has a time shift of τ relative to the input pulse shape due to time asynchronism between the transmitter and receiver. The cross-correlation function will be explicitly defined as

$$R_{wv}(\tau) = \int_{-\Delta}^{\Delta} w(t) \cdot v(t-\tau)\mathrm{d}t \tag{1.20}$$

Thus the signal component of the correlator output becomes

$$S = \sqrt{E_p} \cdot R_{wv}(\tau) \tag{1.21}$$

The output noise component, N, is a random variable due to the AWGN, $n(t)$. The mean of $n(t)$ is zero while the autocorrelation function is $R_{nn}(t_1, t_2) = N_0 \cdot \delta(t_1 - t_2)$. Thus the mean of N is zero and the variance is:

$$\begin{aligned}
E\left(N^2\right) &= E\int_{-\Delta}^{\Delta}\int_{-\Delta}^{\Delta} n(t_1)n(t_2)v(t_1)v(t_2)\mathrm{d}t_1\mathrm{d}t_2 \\
&= \int_{-\Delta}^{\Delta}\int_{-\Delta}^{\Delta} E\left\{n(t_1)n(t_2)\right\} v(t_1)v(t_2)\mathrm{d}t_1\mathrm{d}t_2 \\
&= \int_{-\Delta}^{\Delta}\int_{-\Delta}^{\Delta} R_{nn}\left(t_1, t_2\right) v(t_1)v(t_2)\mathrm{d}t_1\mathrm{d}t_2 \\
&= N_0\int_{-\Delta}^{\Delta}\int_{-\Delta}^{\Delta} \delta\left(t_1 - t_2\right) v(t_1)v(t_2)\mathrm{d}t_1\mathrm{d}t_2 \\
&= N_0\int_{-\Delta}^{\Delta} v^2(t)\mathrm{d}t \\
&= N_0 R_{vv}(0)
\end{aligned}$$

The output SNR, computed as $S^2/E\left(N^2\right)$, is:

$$SNR = \frac{E_p}{N_0} \cdot \frac{R_{wv}^2(\tau)}{R_{vv}(0)} \tag{1.22}$$

1.5.1 UWB Correlator with Optimum Template

The optimum template is, of course, $v(t) = w(t)$ and the output SNR is:

$$SNR = \frac{E_p}{N_0} \cdot \frac{R_{ww}^2(\tau)}{R_{ww}(0)} \tag{1.23}$$

The maximum SNR is obtained when the timing error, τ, is zero:

$$SNR_{max} = \frac{E_p}{N_0} \cdot R_{ww}(0) \tag{1.24}$$

In this case, the optimum value of Δ can be found as that value which maximizes $R_{ww}(0)$. Examining $R_{ww}(0)$ reveals that its maximum is obtained as $\Delta \to \infty$. Figure 1.11 validates this assertion and is shown for $w(t) = p(t)$ as in (1.2) with the parameter σ approximately $(2\sqrt{\pi})^{-1} \cdot 0.8$ nsec. The maximum value of $R_{ww}(0)$ is unity since $p(t)$ was constructed to have unit energy as per (1.3).

As can be seen in Figure 1.11, any value of Δ above a certain level (approximately 0.7 nsec as shown for this example) will produce nearly the maximum SNR achievable, this maximum achievable value being:

$$\lim_{\Delta\to\infty} SNR_{max} = \frac{E_p}{N_0} \cdot \lim_{\Delta\to\infty} R_{ww}(0) = \frac{E_p}{N_0} \tag{1.25}$$

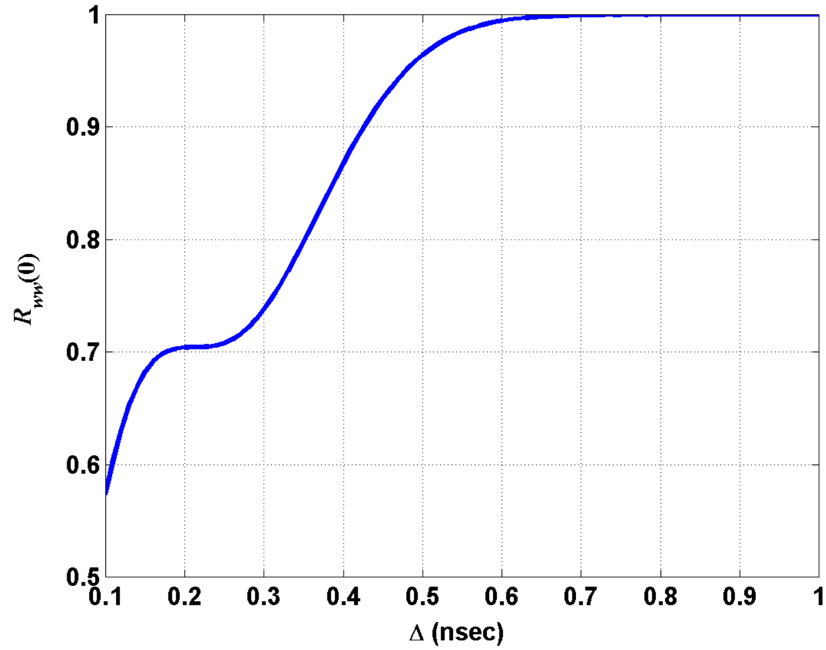

Figure 1.11: $R_{ww}(0)$ for the UWB pulse of (1.2)

Assuming Δ is sufficiently large such that $R_{ww}(0)$ is close to unity yields the following expression for the output SNR as a function of timing error:

$$SNR = \frac{E_p}{N_0} \cdot R_{ww}^2(\tau) \tag{1.26}$$

The function $R_{ww}(\tau)$ in (1.26) can be replaced with the limiting function since Δ is chosen sufficiently large, where this limiting function has a closed form expression. Recall that in this section $w(t) = p(t)$ which yields

$$\lim_{\Delta \to \infty} R_{ww}(\tau) = \int_{-\infty}^{\infty} p(t)p(t-\tau)\mathrm{d}t \tag{1.27}$$

Substitution of (1.2) into the previous equation eventually leads to the following expression where $\gamma(\tau)$ is defined as the limit of $R_{ww}(\tau)$:

$$\gamma(\tau) \triangleq \lim_{\Delta\to\infty} R_{ww}(\tau) = \frac{4}{3\sigma\sqrt{\pi}} \int_{-\infty}^{\infty} \left(\sum_{n=0}^{N} a_n \cdot t^n \right) \cdot \exp\left(-\alpha(t-\beta)^2\right) \mathrm{d}t \tag{1.28}$$

where

$$a_0 = 1 - \left(\frac{\tau}{\sigma}\right)^2, \quad a_1 = \frac{2\tau}{\sigma^2}, \quad a_2 = -\frac{2}{\sigma^2} + \frac{\tau^2}{\sigma^4}, \quad a_3 = -\frac{2\tau}{\sigma^4}, \quad a_4 = \frac{1}{\sigma^4} \tag{1.29}$$

and

$$N = 4, \quad \alpha = \frac{1}{\sigma^2}, \quad \beta = \frac{\tau}{2} \tag{1.30}$$

The following identity can then be employed:

$$\int_{-\infty}^{\infty} \left(\sum_{n=0}^{N} a_n \cdot t^n \right) \exp\left(-\alpha(t-\beta)^2\right) \mathrm{d}t = \sqrt{\pi} \sum_{n=0}^{N} a_n \alpha^{-\frac{n+1}{2}} (2i)^{-n} H_n\left(i\beta\sqrt{\alpha}\right) \tag{1.31}$$

where $H_n(t)$ is the Hermite polynomial of order n and $i = \sqrt{-1}$. This identity results in:

$$\gamma(\tau) = \frac{4}{3\sigma\sqrt{\pi}} \sqrt{\pi} \cdot \sum_{n=0}^{N} a_n \cdot \sigma^{n+1} \cdot (2i)^{-n} \cdot H_n\left(\frac{i}{2} \cdot \frac{\tau}{\sigma}\right) \tag{1.32}$$

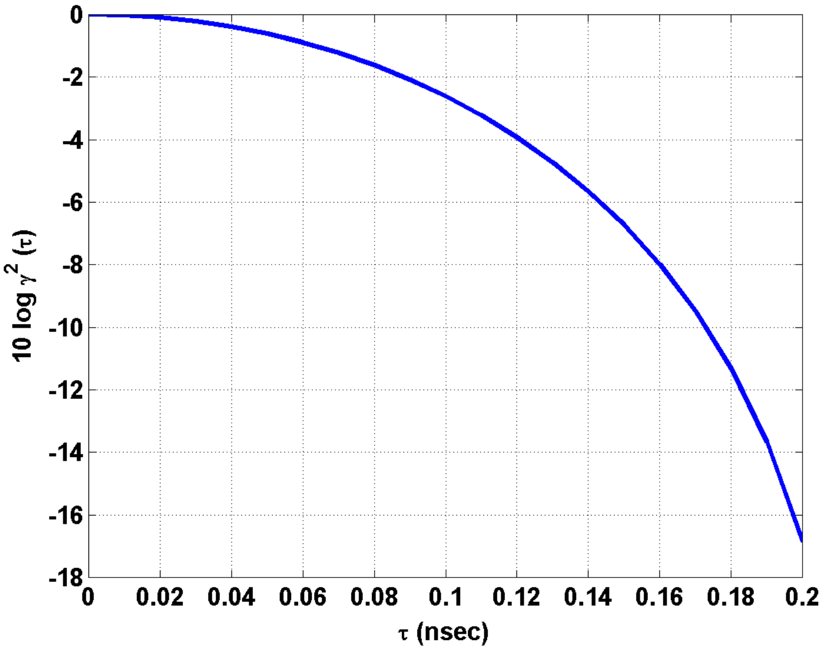

Figure 1.12: Normalized output SNR (dB) vs. timing error

This representation of the correlator output as a sum of Hermite polynomials simplifies to the following result:

$$\gamma(\tau) = \left(1 - \left(\frac{\tau}{\sigma}\right)^2 + \frac{1}{12} \cdot \left(\frac{\tau}{\sigma}\right)^4\right) \cdot \exp\left(-\frac{1}{4} \cdot \left(\frac{\tau}{\sigma}\right)^2\right) \tag{1.33}$$

The SNR of (1.26) can be represented as $\frac{E_p}{N_0} \cdot \gamma^2(\tau)$ and a logarithmic plot of this SNR, normalized by E_p/N_0, is shown in Figure 1.12 versus timing error (for the same σ as above). Of particular interest is the rapid decrease in output SNR for very small timing errors. This induces a very stringent timing requirement in a real system.

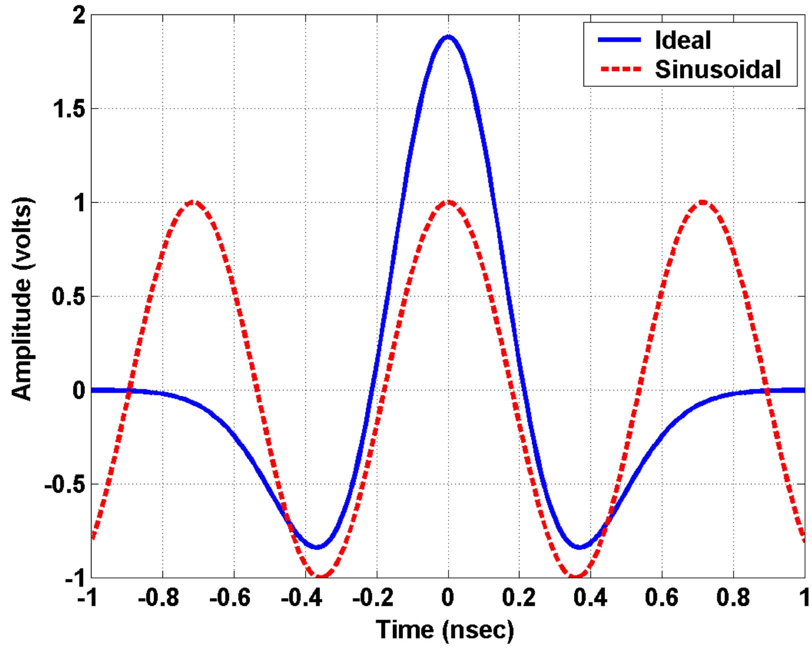

Figure 1.13: Ideal template versus sinusoidal template ($f_c = 1.4$ GHz)

1.5.2 UWB Correlator with Sinusoidal Template

In lieu of actually generating the ideal template at the receiver, the simpler sinusoidal template signal, $v(t) = \cos(2\pi f_c t)$, is examined. Of course, in a pulsed system the phase of the oscillator used to generate $v(t)$ needs to be adjusted for each pulse so as to accurately align the template with the incoming signal. Figure 1.13 shows the properly aligned oscillator template for one specific oscillator frequency, f_c. This oscillator frequency and the integration time, Δ, need to be chosen so as to maximize the output SNR of the correlator. The output SNR in (1.22) depends upon the functions $R_{wv}(\tau)$ and $R_{vv}(0)$ which are inherently functions of f_c and Δ. Rather than obtaining closed form expressions as was done for the ideal template, the expressions are left as integrals.

For the sinusoidal template, the output SNR of (1.22) becomes:

$$SNR = \frac{E_p}{N_0} \cdot \frac{\left(\int\limits_{-\Delta}^{\Delta} w(t) \cdot \cos\left(2\pi f_c (t-\tau)\right) \mathrm{d}t \right)^2}{\int\limits_{-\Delta}^{\Delta} \cos^2(2\pi f_c t) \mathrm{d}t} \tag{1.34}$$

The degradation to output SNR with respect to the ideal template's maximum achievable SNR is simply E_p/N_0 divided by the output SNR for the sinusoidal template as per (1.34):

$$D(\tau, \Delta, f_c) \triangleq SNR \;\; Degradation = \frac{\int\limits_{-\Delta}^{\Delta} \cos^2(2\pi f_c t) \mathrm{d}t}{\left(\int\limits_{-\Delta}^{\Delta} w(t) \cdot \cos\left(2\pi f_c (t-\tau)\right) \mathrm{d}t \right)^2} \tag{1.35}$$

Thus the optimum parameters are obtained by minimizing the SNR degradation at zero timing offset:

$$(\Delta, f_c)^{opt} = \arg \min_{(\Delta, f_c)} D(0, \Delta, f_c) \tag{1.36}$$

Equation (1.35) is shown in Figures 1.14 and 1.15 for various values of f_c and Δ. These plots provide rough estimates of the optimal parameters.

The lowest degradation achievable is roughly 0.72 dB when $\Delta = 0.6$ nsec and $f_c = 1.25$ GHz. Notice, however, that when $\Delta = 0.5$ nsec, the degradation in Figure 1.15 becomes fairly independent of f_c. This is a very beneficial property in lieu of oscillator frequency stability, i.e., a slight oscillator drift would not be catastrophic to the system. Choosing $\Delta = 0.5$ nsec because of this reason, a good choice of oscillator

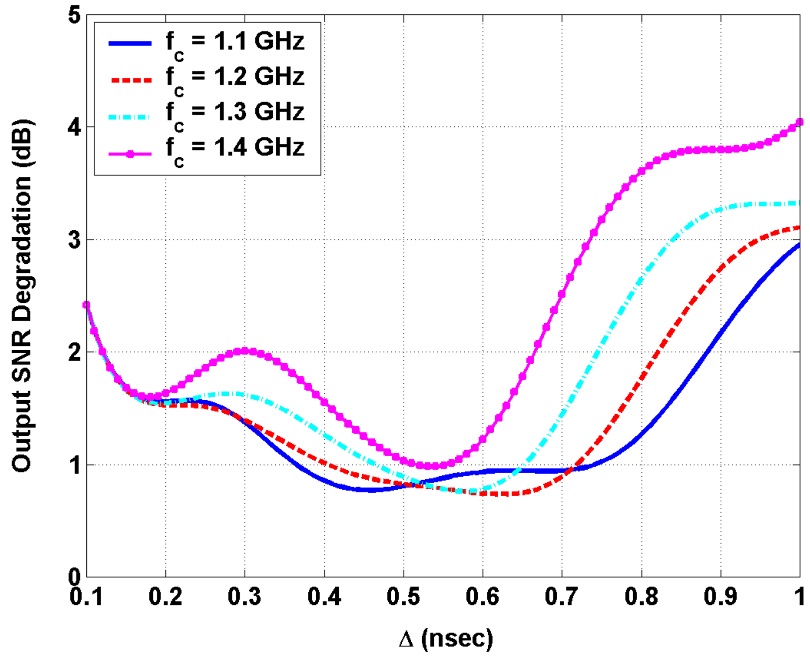

Figure 1.14: Output SNR degradation (dB) vs. Δ

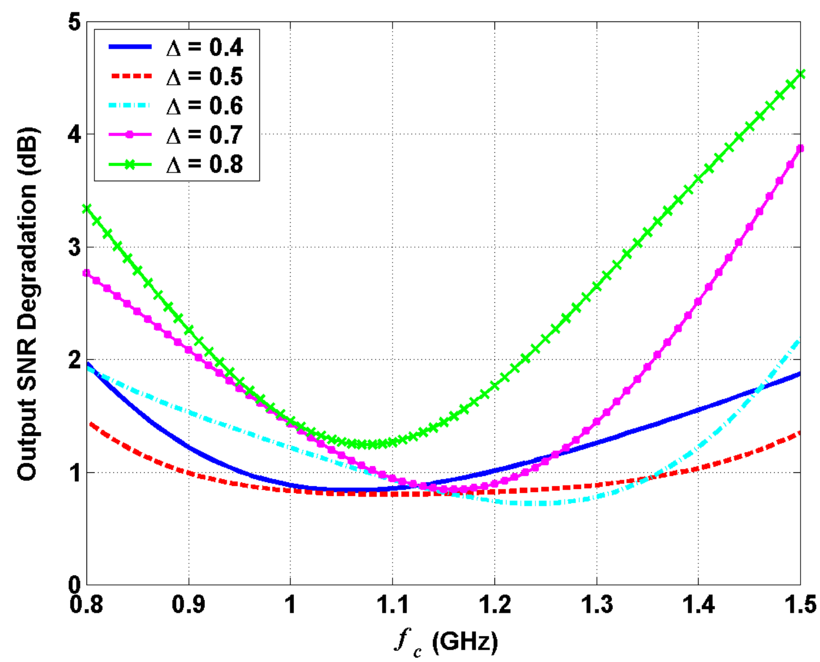

Figure 1.15: Output SNR degradation (dB) vs. f_c

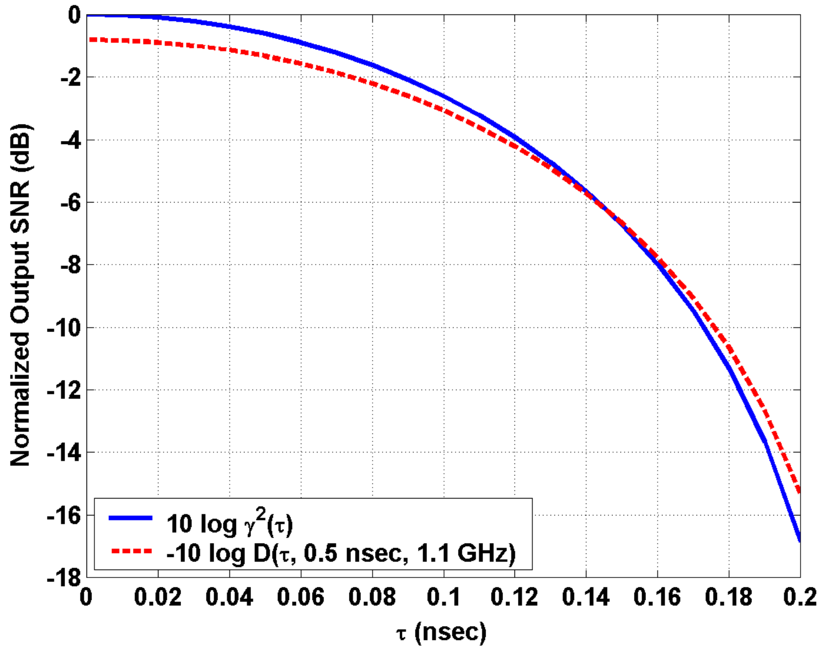

Figure 1.16: Normalized output SNR (dB) vs. timing error (for both the ideal and sinusoidal templates)

frequency from Figure 1.15 is roughly $f_c = 1.1$ GHz. Figure 1.16 shows the normalized output SNR versus timing offset for this choice of parameters. Also shown in Figure 1.16 is the normalized output SNR for the ideal template as in Figure 1.12. For the parameters chosen, an interesting observation is made, namely that the output SNR for the sinusoidal template is larger than the output SNR for the ideal template at large timing errors.

1.6 Selective RAKE Receiver

As mentioned earlier, a RAKE receiver is an optimum single-user detector in the presence of specular multipath [37]. In order for such a detector to be realizable, the number of paths considered in the receiver must be limited to a finite number, say L_p. For optimum RAKE performance, these L_p dominant paths will be those pairs (a_m, τ_m) in (1.10) corresponding to the L_p largest a_m values, with the set of indices corresponding to these pairs being represented by $\mathcal{M}$.

In actuality, however, the exact multipath channel will not be known and only estimates of the (a_m, τ_m) pairs will be available, possibly due to one of the channel sounding techniques discussed earlier or from the fine acquisition process described in Chapter 5. Denoting these estimates as $(\hat{a}_m, \hat{\tau}_m)$ and accounting for a time difference between transmitter and receiver of τ yields the correlator template, $v(t)$ in Figure 1.10.

$$v(t) = \sum_{m \in \mathcal{M}} \hat{a}_m p\left(t - \hat{\tau}_m - \tau\right) \tag{1.37}$$

The limits of integration in Figure 1.10 must be changed for the selective RAKE receiver and for the purposes of this section (to determine a mathematical formulation of the RAKE output) these limits will be set to $\pm\infty$. Straightforward calculation of the signal component of the correlator output, S, yields:

$$S = \sqrt{E_p} \cdot \sum_{m=0}^{\infty} \sum_{n \in \mathcal{M}} a_m \hat{a}_n \gamma(\tau_m - \hat{\tau}_n - \tau) \tag{1.38}$$

The correlation of the pulse waveform with itself, $\gamma(\tau)$, is given in (1.33). The signal component, S, can be divided into two terms based upon those a_m's for $m \in \mathcal{M}$ and $m \notin \mathcal{M}$.

$$S = \sqrt{E_p}\left(\sum_{m\in\mathcal{M}}\sum_{n\in\mathcal{M}} a_m \hat{a}_n \gamma(\tau_m - \hat{\tau}_n - \tau) + \sum_{m\notin\mathcal{M}}\sum_{n\in\mathcal{M}} a_m \hat{a}_n \gamma(\tau_m - \hat{\tau}_n - \tau)\right) \quad (1.39)$$

The signal portion of the correlator output, S, can also be written compactly as a vector-matrix product:

$$S = \sqrt{E_p}\left(\mathbf{a}^{\mathrm{T}}\mathbf{R}_s\hat{\mathbf{a}} + \zeta\right) \quad (1.40)$$

where ζ is the second term in (1.39), $\mathbf{a}$ is an $L_p \times 1$ column vector of the amplitude coefficients a_m for $m \in \mathcal{M}$, $\hat{\mathbf{a}}$ is an $L_p \times 1$ column vector of amplitude coefficient estimates and $\mathbf{R}_s$ is the $L_p \times L_p$ 'signal-only' correlation matrix. Letting the elements of $\mathcal{M}$ be denoted as $\{m_1, m_2, \cdots, m_{L_p}\}$, the vectors $\mathbf{a}$, $\hat{\mathbf{a}}$, and the matrix $\mathbf{R}_s$ can be explicitly stated:

$$\mathbf{a} = [a_{m_1}, a_{m_2}, \cdots, a_{m_{L_p}}]^{\mathrm{T}} \quad (1.41)$$

$$\hat{\mathbf{a}} = [\hat{a}_{m_1}, \hat{a}_{m_2}, \cdots, \hat{a}_{m_{L_p}}]^{\mathrm{T}} \quad (1.42)$$

$$\mathbf{R}_s = \begin{bmatrix} \gamma(\tau_{m_1} - \hat{\tau}_{m_1} - \tau) & \ldots & \gamma(\tau_{m_1} - \hat{\tau}_{m_{L_p}} - \tau) \\ \vdots & \ddots & \vdots \\ \gamma(\tau_{m_{L_p}} - \hat{\tau}_{m_1} - \tau) & \ldots & \gamma(\tau_{m_{L_p}} - \hat{\tau}_{m_{L_p}} - \tau) \end{bmatrix} \quad (1.43)$$

The noise component of the correlator output, N, is due solely to wide-sense stationary AWGN $n(t)$ present at the input. The mean of $n(t)$ is zero and the auto-correlation function, $E(n(t_1)n(t_2))$ is $N_0\delta(t_1 - t_2)$. As with the signal component, the noise component is computed in a straightforward manner as:

$$N = \int_{-\infty}^{\infty} n(t) \cdot \sum_{m \in \mathcal{M}} \hat{a}_m p(t - \hat{\tau}_m - \tau)\mathrm{d}t \tag{1.44}$$

The mean of N is found to be zero and the variance is found to be

$$\sigma_N^2 = N_0 \cdot \sum_{m \in \mathcal{M}} \sum_{n \in \mathcal{M}} \hat{a}_m \hat{a}_n \gamma(\hat{\tau}_m - \hat{\tau}_n) \tag{1.45}$$

As with the signal component, S, the variance of the noise component can be written in vector-matrix form:

$$\sigma_N^2 = N_0 \cdot \hat{\mathbf{a}}^{\mathrm{T}} \mathbf{R}_n \hat{\mathbf{a}} \tag{1.46}$$

where $\hat{\mathbf{a}}$ is given in (1.42) and the 'noise-only' correlation matrix, $\mathbf{R}_n$, is given by:

$$\mathbf{R}_n = \begin{bmatrix} \gamma(\hat{\tau}_{m_1} - \hat{\tau}_{m_1}) & \dots & \gamma(\hat{\tau}_{m_1} - \hat{\tau}_{m_{L_p}}) \\ \vdots & \ddots & \vdots \\ \gamma(\hat{\tau}_{m_{L_p}} - \hat{\tau}_{m_1}) & \dots & \gamma(\hat{\tau}_{m_{L_p}} - \hat{\tau}_{m_{L_p}}) \end{bmatrix} \tag{1.47}$$

The diagonal elements of $\mathbf{R}_n$ are always one since $\gamma(0) = 1$. The off-diagonal terms are not necessarily zero and depend on the path arrival time estimate differences as well as the signal correlation function. Finally, a general expression for the selective RAKE receiver signal-to-noise ratio, $S^2/E(N^2)$, is seen to be:

$$SNR_{sRAKE} = \frac{E_p}{N_0} \cdot \frac{(\mathbf{a}^{\mathrm{T}}\mathbf{R}_s\hat{\mathbf{a}} + \zeta)^2}{\hat{\mathbf{a}}^{\mathrm{T}}\mathbf{R}_n\hat{\mathbf{a}}} \tag{1.48}$$

The multipath channel is said to be *separable* if $|\tau_m - \tau_n| >$ 'width' of $p(t)$ for all $m \neq n$ so that $\gamma(\tau_m - \tau_n) = 0$ for all $m \neq n$. For perfect estimates of the multipath channel ($\hat{a}_m = a_m$ and $\hat{\tau}_m = \tau_m$ for all $m \in \mathcal{M}$), perfect synchronization ($\tau = 0$), and for a separable multipath channel as just defined, the correlation matrices, $\mathbf{R}_s$ and $\mathbf{R}_n$, both reduce to the $L_p \times L_p$ identity matrix, I, and the ζ term is *exactly* zero. Thus, for the channel which is *almost* separable, estimates which are *nearly* perfect, and *near* perfect synchronization ($\tau \approx 0$), the ζ term can be assumed to be *approximately* zero [1]. The selective RAKE output SNR of (1.48) for $\zeta = 0$ becomes

$$SNR_{sRAKE} = \frac{E_p}{N_0} \cdot \frac{(\mathbf{a}^{\mathrm{T}}\mathbf{R}_s\hat{\mathbf{a}})^2}{\hat{\mathbf{a}}^{\mathrm{T}}\mathbf{R}_n\hat{\mathbf{a}}} \tag{1.49}$$

[1] It can also be assumed that $\zeta \approx 0$ since the a_m's for $m \in \mathcal{M}$ are by definition the largest group of a_m's so that $\zeta \ll \mathbf{a}^{\mathrm{T}}\mathbf{R}_s\hat{\mathbf{a}}$.

If the channel is separable with perfect channel estimates and perfect synchronization then the above SNR expression in (1.48) reduces exactly to

$$SNR_{sRAKE} = \frac{E_p}{N_0} \cdot \sum_{m \in \mathcal{M}} a_m^2 \tag{1.50}$$

In order for the energy in the transmitted pulse to be properly conserved through the multipath channel, the SNR for an infinite RAKE receiver (one in which L_p approaches infinity) must be equivalent to the maximum attainable SNR in a perfect channel without multipath [2]. This maximum SNR is given in (1.25) as E_p/N_0. The infinite RAKE receiver SNR is found as L_p becomes unbounded:

$$SNR_{iRAKE} \triangleq \lim_{L_p \to \infty} SNR_{sRAKE} = \frac{E_p}{N_0} \cdot \sum_{m=0}^{\infty} a_m^2 \tag{1.51}$$

This infinite RAKE SNR reveals the following constraint on the multipath channel amplitude coefficients:

$$\sum_{m=0}^{\infty} a_m^2 = 1 \tag{1.52}$$

Thus it can be seen that a selective RAKE receiver is always a suboptimum detector suffering from a lower output SNR with respect to the infinite RAKE receiver since $\sum_{m \in \mathcal{M}} a_m^2 < \sum_{m=0}^{\infty} a_m^2 = 1$ which implies that $SNR_{sRAKE} < SNR_{iRAKE}$ in (1.50). This is evident in the energy capture curves of Figure 1.9.

[2]Here the multipath channel is assumed to be specular so that as L_p approaches infinity, 100% of the signal energy will indeed be accounted for in the selective RAKE receiver template.

Chapter 2

Search Analysis Techniques

The process of synchronization is to bring into alignment the transmitter and receiver time references. The majority of literature on synchronization, as discussed in [19], [30], [31], [32], for example, deals with narrowband sinusoidal carrier signals modulated in some fashion with a data stream. A shift in time of these signals results in a carrier phase shift of the sinusoid so that one can think of phase synchronization (at least within one wavelength of the carrier) as analogous to time synchronization. As a result, much of the synchronization literature discusses Phase-Locked Loop (PLL) architectures. There also exist PLL structures which use phase estimates to reduce the acquisition time [21] as well as modified 2nd order PLL structures which provide estimates of slowly varying frequency offset, such as Doppler shifts in satellite communications, while still tracking small user frequency offsets [20]. For UWB or 'carrier-less' systems, the Doppler shift of a carrier is no longer applicable. In this case the general theory of relativity is employed, and specifically, time-dilation is used to account for relative motion effects between the transmitter and receiver [58].

A significantly different form of signaling, known as pulse position modulation (PPM), requires a different framework when discussing synchronization. The UWB signals discussed earlier fall into the category of PPM signals as do signals used for optical communications. One significant difference between sinusoidal carrier modulated signals and PPM signals is that the carrier modulated signals are always present for any observed location in time. This is not true for PPM signals. The phase locking techniques rely on continuously attempting to drive an error term to zero through either feedback or feedforward systems, and can be done since a carrier is always present. For PPM signals there will exist a large portion of time where no signal is present. There are certain methods that can be employed to allow the use of a PLL for synchronization of PPM signals as discussed in [11].

The PLL techniques working on the carrier phase can only resolve the transmitter and receiver clocks to within one wavelength of the sinusoidal carrier signal. Because of the ranges involved in practical applications and the presence of sequential time events in the data stream, additional delay must be resolved and/or tracked. Direct sequence coding for multiple users is one such system. TDMA burst mode communications is another example where synchronization of the time references beyond a wavelength must be established. The UWB signals defined earlier will also require this level of synchronization because of the frame time and time-hopped coding (and because no carrier signal is employed.) This type of timing synchronization is usually accomplished via delay-locked loops [15], [51], [52] or tau-dither loops [17].

These delay-locked loops are typically used for signal *tracking* which resolves small timing errors. These small timing errors typically result from timing instabilities in the clocks. In contrast, the large initial timing errors are resolved during the *acquisition* portion of the synchronization process. Both *frame acquisition* and *code acquisition* will be addressed. Because of the multipath present in the channel and the presence of a RAKE receiver, for optimum detection the strongest paths in amplitude must be assigned to the available correlators. This process will be termed *fine acquisition* while the process of simply finding the multipath 'cluster' will be termed *coarse acquisition*.

Delay-locked loops typically use two local shifted versions of the received signal, one advanced in time and the other delayed in time (hence the term *early-late gate*). The early-late gate generates an error signal which is then used to update the local timing references accordingly. An analysis of the early-late gate proceeded by a memoryless nonlinearity (MNL) is given in [23] for a TDMA burst mode communication system employed root-raised cosine shaped QPSK. The same analysis without the MNL is given in [22].

An extension of the early-late gate which correlates with more than two time-shifted versions of the local signal allows a large expanse in time to be investigated. Actually, if a correlator could be placed at each possible time location, the correlator with the maximum value would yield the maximum likelihood estimate of the timing error. However, there are usually too many possible locations, perhaps an uncountable number, making the maximum likelihood receiver impractical. Generally the timing

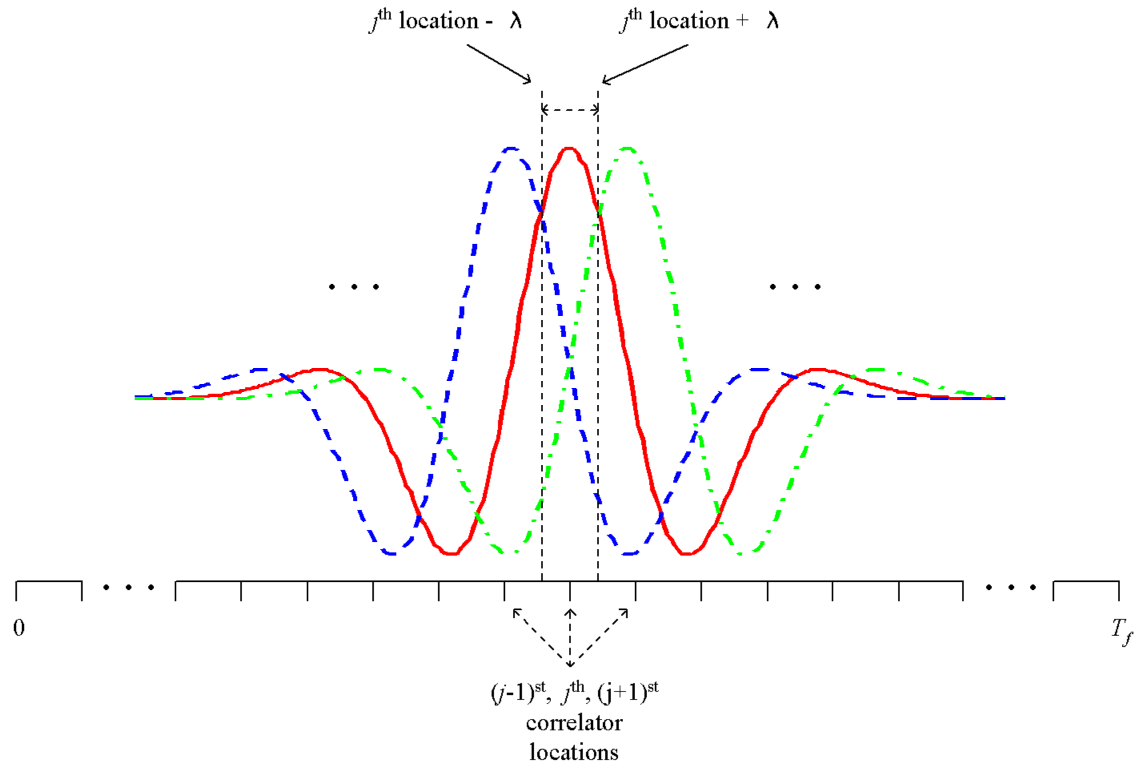

Figure 2.1: Equally spaced correlators across the frame time

error of the arriving signal is assumed random with some prior distribution. As discussed in Section 1.3, the timing error of the first multipath arrival is uniform over the frame time. Thus to make the problem tractable, the frame time is divided into discrete regions with the width of each region determined by some predetermined correlator spacing. Figure 2.1 shows an example with a specific spacing between correlators, where the correlator output function $\gamma(\tau)$ is shown.

The spacing between correlators is ultimately determined by the false alarm and detection probabilities, discussed here. Denoting the j^{th} correlator output as Z_j, and assuming a detection threshold of T, the probability of detection for the j^{th} correlator is

$$P_d = Pr(|Z_j| > T \,|H_j\,) \tag{2.1}$$

Here the hypothesis H_j represents the event that the timing error falls within $\pm\lambda$ of the j^{th} correlator peak, as shown in Figure 2.1. The notation $\bar{H}_j$ will represent the event that the timing error falls outside of a window $\pm\lambda$ around the j^{th} correlator. This means that $\bar{H}_j$ is simply one of the other hypotheses, H_k for $k \neq j$. Thus the false alarm probability is defined as

$$P_{fa} = Pr(|Z_j| > T \left|\bar{H}_j\,\right) \tag{2.2}$$

The correlator spacing sets the required number of correlators to cover a fixed frame time. Increasing the spacing decreases the number of correlators and thus reduces the overall receiver complexity. However, if the correlator spacing is too large then the correlator output for all of the correlators could be low resulting in a low detection probability. Conversely, if the correlators spacing is too small then an excessively large number of correlators outputs could exceed the detection threshold.

In light of the fact that an impractical number of correlators could be required to simultaneously investigate each hypothesis, $H_0, H_1, \cdots$, a type of hypothesis testing known as a *search* is employed. As discussed in [53], a search is the process of converting a multiple hypothesis test into a series of simpler binary hypothesis tests.

The trade-off inherent here is reduced complexity at the cost of increased time to reach a final decision. The first stage of the search tests the observed random variable (in this case the correlator output) against two hypotheses, say H_j and $\bar{H}_j$ as defined above. If the selected hypothesis is H_j then the search is terminated, otherwise the process is continued with another set of hypotheses, H_k and $\bar{H}_k$. To be as general as possible, k is unrestricted and can equal j, as will be the case for a truly random search to be defined below.

Traditionally, a search will properly terminate only with one correct hypothesis. This assumption is found in [53] and the results therein are useful only with this assumption. In order to properly analyze the UWB acquisition problem in Chapters 4 and 5 this assumption needs to be removed and the search needs to be reexamined when multiple hypotheses will correctly terminate the search. The following section, 2.1, provides background material for the single terminating hypothesis situation. The results are then extended to deal with multiple terminating hypotheses in the subsequent sections.

2.1 Search Performance for a Single Terminating Hypothesis

While the performance measures for a search ultimately depend upon the application, in most cases a predominant measure is the time required to complete the search, denoted by T_s. There are two distinct classifications of searches: A) a search which is allowed to run indefinitely until a hypothesis is selected and B) a search which needs to be completed before some specified time, say T_s^*. For example, the first scenario arises when the acquisition signal always has data present. The latter type of search arises when data transmission occurs only after the specified time, T_s^*, thus making it imperative that the search be completed with high probability before T_s^*.

Since the search time, T_s, will be random in nature, a statistical description will be required. A complete statistical description, i.e., a distribution function for T_s, is often too difficult to obtain. Thus, for the 'type A' search the mean and sometimes the variance of the search time will often suffice. For the 'type B' search the probability that the search terminates on time is of importance and is given as $\Pr(T_s \leq T_s^*)$, where it is desired that this probability be as large as possible.

An important parameter of either type of search is the probability that the search terminated by selecting the correct hypothesis, denoted as P_c. Often the goal of a well designed search algorithm is to maximize P_c while attempting to minimize the number of observations required to complete the search. A common practice used in this optimization is to fix P_c and then to minimize the search time. In order to attain a specific value for P_c a *verification* phase as described in [34] is employed.

Having described the appropriate performance measures, the single terminating hypothesis scenario will now be investigated. If there are N total hypotheses, each of which is a-priori equally likely to be the terminating hypothesis, then the expected search time (in number of hypotheses tested) for an *exhaustive* search is given in [53]. Denoting the search time as ν, the expected search time is shown to be

$$E(\nu) = \sum_{\nu=0}^{\infty} \nu \Pr(\nu) = \sum_{i=0}^{N-1} \sum_{j=0}^{\infty} (jN + i + 1) \Pr(\nu = jN + i + 1) \tag{2.3}$$

There are two summations above since the search quite possibly could examine all N hypotheses without selecting one of them. At this point the search is restarted, thus the j variable above represents the number of times the entire collection of hypotheses has been searched without termination. The i variable represents the number of hypotheses that have been tested on the current 'pass' of the entire collection. The probability of the search terminating on any given number is given as

$$\begin{aligned} \Pr(\nu = jN + i + 1) &= \frac{1}{N}(1-\beta)^{j(N-1)}\alpha^j \left[(1-\beta)^i(1-\alpha)\right. \\ &\quad \left. + i\alpha\beta(1-\beta)^{i-1} + (N-i-1)\beta(1-\beta)^i\right] \end{aligned} \tag{2.4}$$

Here the quantities $\alpha = 1 - P_d$ and $\beta = P_{fa}$, i.e., $1 - \alpha$ is the detection probability and β is the false alarm probability. A simplified expression for (2.3) is not given in [53] but can be computed as follows:

$$E(\nu) = \frac{1}{\beta} + \frac{1}{N\beta^2} \cdot \frac{1-(1-\beta)^N}{1-\alpha(1-\beta)^{N-1}} \cdot (\alpha + \beta - 1) \tag{2.5}$$

2.2 Search Performance for Consecutive Terminating Hypotheses

Given the hypotheses H_0, H_1, $\cdots$, H_{N-1} a 'search variable', Y_m for $m = 0,1,\cdots,\infty$, can be defined. Y_m represents the index of the current hypothesis being tested while the variable m represents the observation number, that is for the m^{th} observation the hypothesis under test is H_{Y_m}. The particular sequence of indices, Y_0, Y_1, Y_2, $\cdots$, determines the order in which the hypotheses are tested. The term 'search algorithm' is used to describe the general methodology for determining the search order, which may be completely deterministic or it may be a random sequence. Five different search algorithms will be investigated below: 1) linear search, 2) truly random search, 3) random permutation search, 4) 'look and jump by K bins' search and 5) bit reversal search. The indices of the 'true' hypotheses that will terminate the search are represented by the random variables $X_1, X_2, \cdots, X_K$, and are K consecutive indices. Note that these values will wrap from hypothesis H_{N-1} back to H_0 if need be. The general problem assumptions are listed explicitly below:

1. Y_m is the search variable for $m = 0, 1, \cdots, \infty$.

2. M is the stopping time associated with finding the 'cluster' of bins $X_1, X_2, \cdots, X_K$ and is given as $M = \inf(m : Y_m \in \{X_1, X_2, \cdots, X_K\}) + 1$.

3. $\Pr(X_1 = n) = 1/N$ for $n = 0, \cdots, N-1$, i.e., X_1 is discrete uniform.

4. $K \in \{1, 2, \cdots, N\}$ and is deterministic.

5. X_1 is independent of Y_m for all m.

6. $X_k = X_1 \oplus (k-1)$ for $k = 1, 2, \cdots, K$ where $\oplus$ is modulo N addition and allows for 'wrap-around'. By construction $X_2, \cdots, X_K$ are dependent upon X_1.

7. The observations are perfect such that the detection and false alarm probabilities are unity and zero, respectively.

Linear Search

For this particular search, $Y_m = m \bmod N$ for $m = 0, 1, \cdots, \infty$. As with all the searches to be examined, the stopping time, M, will be finite with probability one as long as the detection and false alarm probabilities are not *both* zero. For the ideal case at hand, the detection probability is one and the false alarm probability is zero, assuring that the search will indeed terminate with probability one. Each value that the stopping time can assume is listed explicitly below. These values will then be used to compute the expected stopping time.

- $M = 1$ with probability $\Pr(X_1 = 0$ or $X_1 = N - K + 1$ or $X_1 = N - K + 2$ or $\cdots$ or $X_1 = N - 1) = \frac{1}{N} + (K-1) \cdot \frac{1}{N} = \frac{K}{N}$

- $M = 2$ with probability $\Pr(X_1 = 1) = \frac{1}{N}$

  $\vdots$

- $M = N - K + 1$ with probability $\Pr(X_1 = N - K) = \frac{1}{N}$

- $M = N - K + 2$ with probability zero

$\vdots$

- $M = N - 1$ with probability zero
- $M = N$ with probability zero

The probability is zero that $M = N - K + 2, \cdots, N - 1$ since a 'wrap-around' will occur and the search will terminate with $M = 1$. As a check on the validity of the distribution of M, note that $\sum_{m=1}^{N} \Pr(M = m) = \frac{K}{N} + \sum_{m=2}^{N-K+1} \Pr(M = m) = \frac{K}{N} + \frac{1}{N}(N - K) = 1$. The expected stopping time is computed as

$$E(M) = \sum_{m=1}^{N} m \Pr(M = m) = \frac{K}{N} + \frac{1}{N} \sum_{m=2}^{N-K+1} m \tag{2.6}$$

$$= \frac{K}{N} + \frac{1}{N} \left(\frac{(N - K + 1)(N - K + 2)}{2} - 1 \right) \tag{2.7}$$

$$= \frac{(N - K)^2 + (3N - K)}{2N} \tag{2.8}$$

A plot of the normalized mean stopping time, $E(M)/N$, versus the parameter K/N is shown in Figures 2.2, 2.3, and 2.4 for the first four search algorithms listed earlier. The figures correspond to increasing values of N. The parameter K/N is the fraction of the total search area occupied by terminating hypotheses. In terms of the UWB acquisition problem discussed later, this parameter is related to the multipath delay spread normalized by the frame time, T_f. Of particular interest in these figures is the fact that the linear search algorithm performs the poorest for sufficiently large values of K/N. This is an intuitive result stemming from the fact that the hypotheses are consecutive and if the current bin does not terminate the search then the next bin to be searched should be sufficiently far from the current bin. This reasoning gives rise to the 'look and jump by K bins' and bit reversal search algorithms which are found to be optimum among the group. The three values of N in these figures are 25,50, and 100. These values are selected merely for illustrative purposes, as a typical value of N for the UWB frame time acquisition problem could be in the neighborhood of 5000 to 10000 or more.

One final note is given here. Recall for the single terminating hypothesis scenario that for ideal detection and false alarm probabilities the expected search time was shown in (2.42) to be $(N+1)/2$. If a single hypothesis properly terminates the search then $K = 1$. Substituting $K = 1$ into (2.8) yields the same result, $(N + 1)/2$.

Truly Random Search

The truly random search is one in which the history of the previously searched bins is ignored. Thus the search variable, Y_m, is selected at random from $0, 1, \cdots, N-1$ for each m. The probability of selecting any particular Y_m is $1/N$ and every Y_m is independent of every other Y_n for $m \neq n$. The distribution of M can be found by noting that the search terminates if the current search variable sees a terminating hypothesis and if *none* of the previous search variables saw a terminating hypothesis. Letting $\mathcal{X}$ represent $\{X_1, \cdots, X_N\}$ we see that:

$$\Pr(M = k) \quad = \quad \Pr(Y_{k-1} \in \mathcal{X}, Y_{k-2} \notin \mathcal{X}, \cdots, Y_0 \notin \mathcal{X}) \tag{2.9}$$

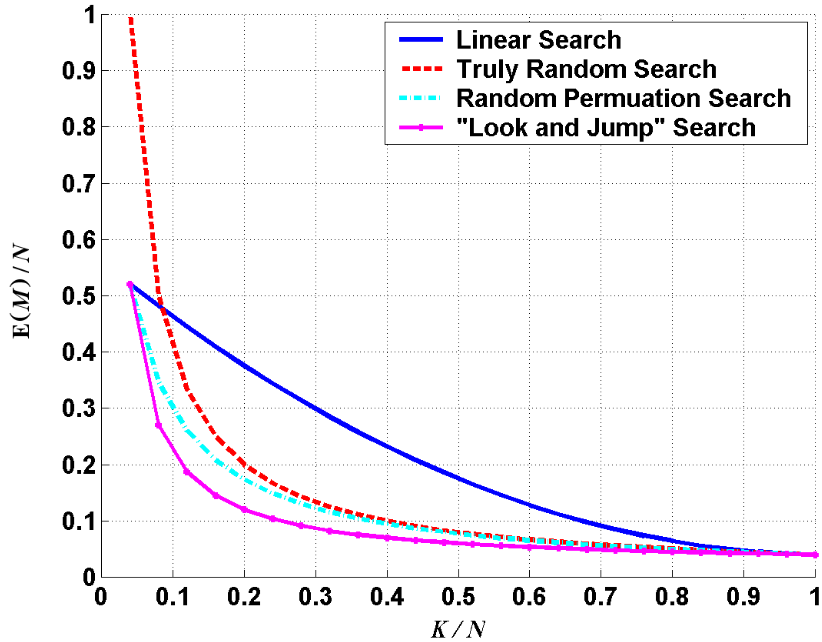

Figure 2.2: Normalized mean stopping time for $N = 25$.

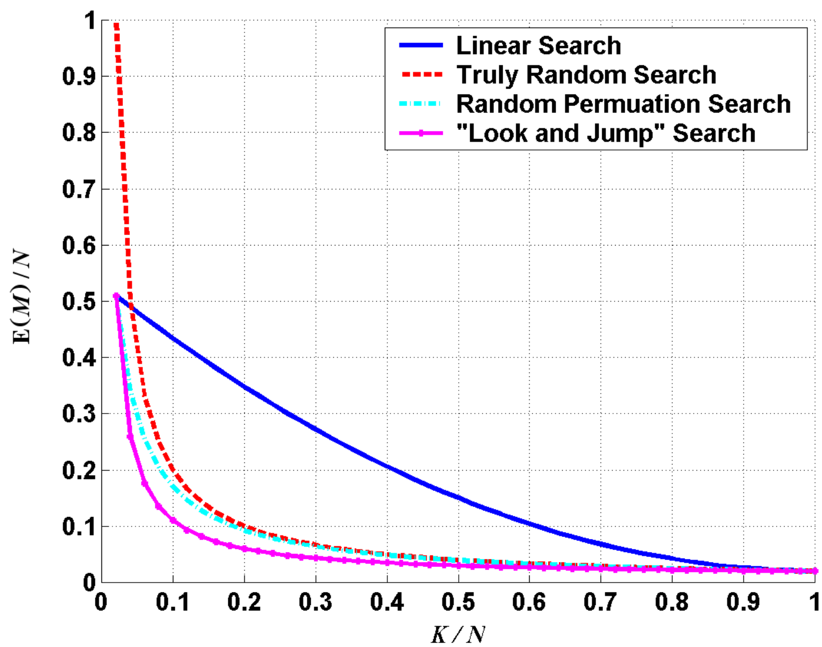

Figure 2.3: Normalized mean stopping time for $N = 50$.

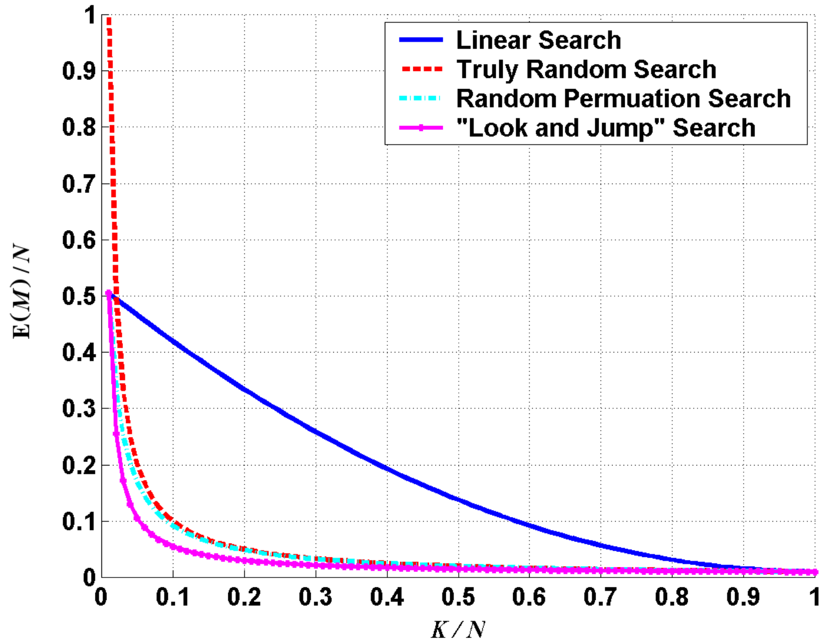

Figure 2.4: Normalized mean stopping time for $N = 100$.

$$= \frac{K}{N} \cdot \left(1 - \frac{K}{N}\right)^{k-1} \tag{2.10}$$

It is easily verified that the distribution sums to one, $\sum_{k=1}^{\infty} \Pr(M = k) = 1$. The expected stopping time is computed as

$$E(M) = \sum_{k=1}^{\infty} k \Pr(M = k) \tag{2.11}$$

$$= \frac{K}{N} \left(1 - \frac{K}{N}\right)^{-1} \sum_{k=1}^{\infty} k \left(1 - \frac{K}{N}\right)^{k} \tag{2.12}$$

$$= \frac{N}{K} \tag{2.13}$$

For a single terminating hypothesis, $K = 1$, the expected search time for a truly random search is seen to be N, which is roughly twice that of the linear search, $(N+1)/2$. However, as can be seen in Figures 2.2, 2.3, 2.3 the truly random search actually performs better than the linear search when K is only slightly larger than one. This occurs since the distance between successive search bins is much larger for the truly random search as compared to the linear search.

Random Permutation Search

For this particular search strategy, the integers $\{0, 1, \cdots, N-1\}$ are randomly permuted and the bins are searched according to this random permutation. More precisely, if σ_n is a random permutation of $\{0, 1, \cdots, N-1\}$ for $n = 0, 1, \cdots, N-1$, then the search random variable is simply $Y_m = \sigma_{m \bmod N}$ for $m = 0, 1, \cdots$. Two necessary facts are now pointed out. Firstly, the search variables $Y_0, Y_1, \cdots$ are no longer independent. Secondly, since the detection probability is one and the false alarm probability is zero the search will need to visit at most $N - K + 1$ bins. This implies that $\Pr(M = k) = 0$ for $k \geq N - K + 2$.

The distribution of M is now found inductively, where $\mathcal{X}$ is defined as the set of terminating hypotheses $\{X_1, X_2, \cdots, X_K\}$ as above.

- $M = 1$ with probability $\Pr(Y_0 \in \mathcal{X}) = \frac{K}{N}$
- $M = 2$ with probability $\Pr(Y_1 \in \mathcal{X}, Y_0 \notin \mathcal{X}) = \Pr(Y_1 \in \mathcal{X} | Y_0 \notin \mathcal{X}) \cdot \Pr(Y_0 \notin \mathcal{X})$ $= \frac{K}{N-1}(1 - \frac{K}{N})$

- $M = 3$ with probability $\Pr(Y_2 \in \mathcal{X}, Y_1 \notin \mathcal{X}, Y_0 \notin \mathcal{X}) = \Pr(Y_2 \in \mathcal{X} | Y_1 \notin \mathcal{X}, Y_0 \notin \mathcal{X}) \cdot \Pr(Y_1 \notin \mathcal{X}, Y_0 \notin \mathcal{X}) = \Pr(Y_2 \in \mathcal{X} | Y_1 \notin \mathcal{X}, Y_0 \notin \mathcal{X}) \cdot \Pr(Y_1 \notin \mathcal{X} | Y_0 \notin \mathcal{X}) \cdot \Pr(Y_0 \notin \mathcal{X}) = \frac{K}{N-2} \cdot \frac{N-1-K}{N-1} \cdot \left(1 - \frac{K}{N}\right) = \frac{K}{N-2} \cdot \left(1 - \frac{K}{N-1}\right) \cdot \left(1 - \frac{K}{N}\right)$

  $\vdots$

- By induction, $\Pr(M = k) = \frac{K}{N-k+1} \prod_{j=0}^{k-2} \left(1 - \frac{K}{N-j}\right)$ for $2 \leq k < N - K + 2$
- $\Pr(M = k) = 0$ for $k \geq N - K + 2$

Summing the distribution of M yields:

$$\sum_{k=1}^{N} \Pr(M = k) = \frac{K}{N} + \sum_{k=2}^{N-K+1} \frac{K}{N-k+1} \cdot \prod_{j=0}^{k-2} \left(1 - \frac{K}{N-j}\right) \tag{2.14}$$

Although difficult to compute, this sum can be shown to equal one. Likewise the mean stopping time is seen to be:

$$E(M) = \frac{K}{N} + \sum_{k=2}^{N-K+1} k \cdot \frac{K}{N-k+1} \cdot \prod_{j=0}^{k-2} \left(1 - \frac{K}{N-j}\right) \tag{2.15}$$

This is equally difficult to evaluate but does indeed simplify. The simplified result is found to be:

$$E(M) = \frac{N+1}{K+1} \tag{2.16}$$

It is noted that for $K = 1$, i.e., the single terminating hypothesis scenario, the performance of the random permutation search is identical to the linear search, namely the expected search time for both is $(N+1)/2$. This is actually the case for all the search algorithms except the truly random search as is evident in Figures 2.2, 2.3, and 2.4 by the fact that all the curves (except the truly random search) start at the same point.

'Look and Jump by K Bins' Search

The random permutation search algorithm previously analyzed gives the mean stopping time, averaged over all permutations of the integers, $0, 1, \cdots, N-1$. Since the linear search is one special case of such a permutation, and the mean stopping time of the random permutation search is lower than that of the linear search as seen in Figures 2.2, 2.3, and 2.4, this reveals that certain permutations must exist that give an even lower mean stopping time. The 'Look and Jump by K Bins' search analyzed here is one such permutation, as is the search analyzed in the next section.

The easiest way to compute the mean stopping time for the current search algorithm is to assume that N/K is an integer. The result obtained with this assumption is then valid even if N/K is not and integer. The basic idea for the current search is as the name suggests, e.g., starting in bin 0, the search continues on to bin K, then to $2K$, etc. As before, the probability of terminating the search after one observation is simply $\Pr(Y_0 \in \mathcal{X}) = K/N$, where again $\mathcal{X}$ is the set of terminating hypotheses. On the next observation, there still remain K bins that will terminate the search because it was not terminated on the first observation. The same reasoning applies to observations $3, 4, \cdots, N/K$, and hence the search will terminate with probability one after at most N/K observations. This argument leads to the distribution of M which is

$$\Pr(M = k) = \begin{cases} K/N & \text{for } k \in \{1, 2, \cdots, N/K\} \\ 0 & \text{for } k > N/K \end{cases} \tag{2.17}$$

Note that $\sum_{k=1}^{\infty} \Pr(M = k) = \sum_{k=1}^{N/K} (K/N) = 1$. The mean stopping time is found as

$$E(M) = \sum_{k=1}^{\infty} k \cdot \Pr(M = k) = \frac{K}{N} \sum_{k=1}^{N/K} k \tag{2.18}$$

$$= \frac{K}{N} \cdot \frac{(N/K)[(N/K) + 1]}{2} \tag{2.19}$$

$$= \frac{1}{2} \cdot \left(\frac{N}{K} + 1 \right) \tag{2.20}$$

As mentioned, the same result applies for N/K not an integer. This algorithm is simply the set of all random permutations which have the initial sequence of $\{0, K, 2K, \cdots, \}$ and thus is a specific subset of the random permutations which *all* happen to perform better than the average. This results because the search terminates after N/K observations with probability one, due to the ideal detection and false alarm probabilities. Additionally, the $K = 1$ case produces the expected result for the mean stopping time of $(N+1)/2$, the same result for all other searches (except the truly random search) when a single hypothesis terminates the search. Finally, it should be noted that the 'Look and Jump' mean search time with parameters N and K reduces to a linear search with parameters N/K and 1.

Bit Reversal Search

For UWB frame and code acquisition, the delay spread of the channel will not be known exactly or, quite possibly, at all. Also, due to the nature of the multipath channel, there will most likely not be K consecutive bins which terminate the search but a cluster of bins, some with high probability and some with low probability of terminating the search. For this reason, a search is desired which does not rely on the knowledge of K but has similar performance. The algorithm discussed in this section is such a search. In fact, as will be seen in Section 2.4 the performance is exactly equal to the 'Look and Jump' search when K is a power of 2. For other values of K the bit reversal search is approximated very well by the 'Look and Jump' search so that the mean search time given in the last section is useful.

The current search is first described by assuming that N is a power of 2, e.g., $N = 2^n$ for n a positive integer. The manner in which the bins are searched is then determined by *bit reversing* the linear search variable. For example, the integers for an $N = 16$ search can be represented in binary (base 2) as $0000, 0001, 0010, 0011, 0100, \cdots, 1101, 1110, 1111$. Obviously a linear search as per these indices will perform poor as shown above, due to the fact that the search does not sufficiently 'jump' far enough from the current location. One permutation of these integers which does maximize the distance between observations is obtained by 'bit reversing' the binary representation, e.g., searching as per

$0000, 1000, 0100, 1100, 0010 \cdots, 1011, 0111, 1111$. In decimal representation this is the sequence $Y_0, Y_1, Y_2, Y_3, \cdots, Y_{13}, Y_{14}, Y_{15} = 0, 8, 4, 12, \cdots, 11, 7, 15$. What this search is really doing is dividing the current group of bins in half and jumping to the halfway point of the current division, thus maximizing the distance between the current search point and the previous search point. If the number of bins N is not a power of 2, then dividing by 2 and rounding to the nearest integer produces a similar search.

2.3 Search Performance for Multiple Terminating Hypotheses

A general framework, based upon a Markov chain model, will now be used to compute the expected search time. The previous scenarios of a single terminating hypothesis and consecutive terminating hypotheses can be considered as a special case of the multiple terminating hypotheses scenario to be described here. As an example of this, the results from Section 2.1 will be compared to a special case of the results to be derived here, with exact agreement existing between the two sections.

Each hypothesis, H_n, for $n = 0, 1, \cdots, N-1$ can be assigned to a state in a Markov chain, e.g., state 0 represents H_0, state 1 represents H_1, etc. One additional state is added to the chain, state N, which represents that the search has terminated. Obviously this state will be *absorbing*, meaning that once entered there is no path leaving that state. Once the *verification* phase is introduced below, this state will be divided into multiple states, one *absorbing* and the rest *transient*. One of these

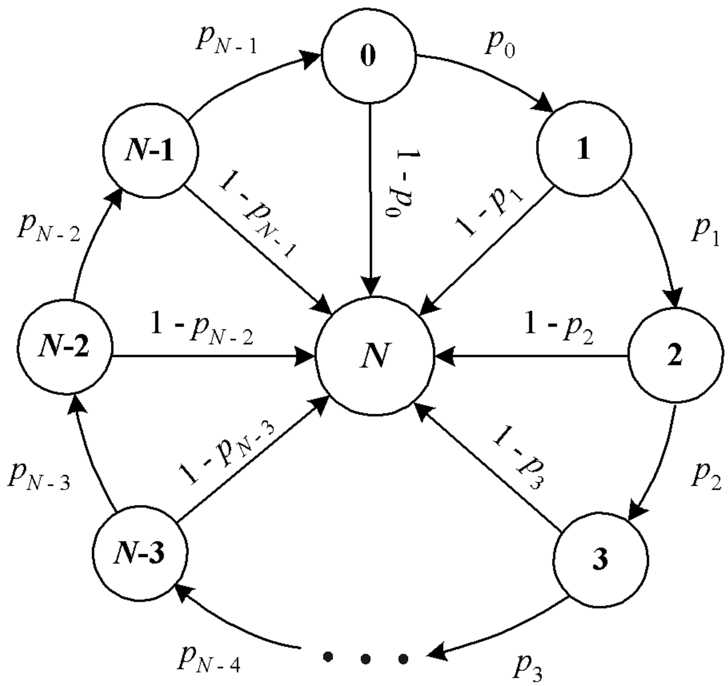

Figure 2.5: Markov chain model for the linear search of Section 2.3

transient states will also be termed the *false alarm* state as was done in [34] for the circular state diagram. For now, however, the Markov chain is shown in Figure 2.5.

The probabilities shown in Figure 2.5 are known as the *transition* probabilities and are defined for the linear search as

$$p_n = \Pr(Y_{m+1} = n \oplus_N 1 | Y_m = n) \tag{2.21}$$

$$1 - p_n = \Pr(Y_{m+1} = N | Y_m = n) \tag{2.22}$$

Here Y_m is the current state of the Markov chain at time index $m = 0, 1, \cdots, \infty$, i.e., the value of Y_m represents the current hypothesis being tested, H_{Y_m}. The operator

$\oplus_N$ represents modulo-N arithmetic. The transition probability matrix, denoted by $\mathbf{S}$, for the Markov chain in Figure 2.5 is

$$\mathbf{S} = \begin{bmatrix} 0 & p_0 & 0 & 0 & \ldots & 0 & 1-p_0 \\ 0 & 0 & p_1 & 0 & \ldots & 0 & 1-p_1 \\ 0 & 0 & 0 & p_2 & \ldots & 0 & 1-p_2 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \ldots & p_{N-2} & 1-p_{N-2} \\ p_{N-1} & 0 & 0 & 0 & \ldots & 0 & 1-p_{N-1} \\ 0 & 0 & 0 & 0 & \ldots & 0 & 1 \end{bmatrix} \tag{2.23}$$

Here the row index represents the current state of the Markov chain, Y_m, and the column index represents the next state of the Markov chain, Y_{m+1}. For example, the probability that the chain starts from state 0 and goes to state 1 is found by looking up the element at row 0 and column 1 which is p_0 as shown in $\mathbf{S}$. Starting from any state, the transition probabilities into any of the other states must sum to one, thus each row of the transition probability matrix must sum to one. Notice that starting from state N, at row N, the only nonzero element in that row is at column N, representing that once state N is entered, with probability one the chain will remain in that state forever. This is the only *absorbing* state in the chain, as described above.

The starting state, Y_0, of the Markov chain is determined by the initial distribution, $\boldsymbol{\pi}$, where each element of this $(N+1)\times 1$ vector is the probability that the chain starts in that state.

$$\begin{aligned}\boldsymbol{\pi} &= [\pi_0, \pi_1, \cdots, \pi_N]^{\mathrm{T}} \qquad (2.24)\\ \pi_n &= \Pr(Y_0 = n) \qquad (2.25)\end{aligned}$$

Obviously the sum of the elements of $\boldsymbol{\pi}$ must be one, $\sum_n \pi_n = 1$, since the Markov chain must start in one of the states.

The expected search time computed in the previous section can now be determined with the Markov chain model. Since state N is the only absorbing state of the chain and represents the fact that the search has terminated, the expected search time can be computed by finding the expected time it takes to enter this absorbing state. This is a commonly investigated problem from Markov theory as discussed in [56] and is given as

$$E(T_a) = \mathbf{1}^{\mathrm{T}}[\mathbf{I} - \mathbf{S}_{tr}^{\mathrm{T}}]^{-1}\boldsymbol{\pi}_{tr} \qquad (2.26)$$

T_a is the time (in number of states visited) to enter an absorbing state, $\mathbf{1}$ is the $N \times 1$ all ones vector, $\mathbf{I}$ is the $N \times N$ identity vector, $\boldsymbol{\pi}_{tr}$ is the initial distribution of the transient states, and $\mathbf{S}_{tr}$ is the reduced state transition matrix corresponding to only the transient states. This reduced matrix is formed by eliminating the rows and columns of $\mathbf{S}$ which correspond to the absorbing state of interest:

$$\mathbf{S}_{tr} = \begin{bmatrix} 0 & p_0 & 0 & 0 & \dots & 0 \\ 0 & 0 & p_1 & 0 & \dots & 0 \\ 0 & 0 & 0 & p_2 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \dots & p_{N-2} \\ p_{N-1} & 0 & 0 & 0 & \dots & 0 \end{bmatrix} \tag{2.27}$$

The mean time to absorption by state N, denoted as $E(T_a)$, given in (2.26) can be simplified for the reduced transition matrix, $\mathbf{S}_{tr}$ as just given. Specifically, it can be shown that

$$\mathbf{I} - \mathbf{S}_{tr}^{\mathrm{T}} = \begin{bmatrix} 1 & -p_0 & 0 & 0 & \dots & 0 \\ 0 & 1 & -p_1 & 0 & \dots & 0 \\ 0 & 0 & 1 & -p_2 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \dots & -p_{N-2} \\ -p_{N-1} & 0 & 0 & 0 & \dots & 1 \end{bmatrix}^{\mathrm{T}} \tag{2.28}$$

The determinant of this matrix is easily computed as

$$\Delta = \det(\mathbf{I} - \mathbf{S}_{tr}^{\mathrm{T}}) = 1 - \prod_{n=0}^{N-1} p_n \tag{2.29}$$

The inverse of $\mathbf{I} - \mathbf{S}_{tr}^{\mathrm{T}}$ can then be computed in terms of the above determinant:

$$\frac{1}{\Delta} \cdot \begin{bmatrix} 1 & p_0 & p_0 p_1 & \cdots & (p_0 p_1 \cdots p_{N-2}) \\ (p_1 p_2 \cdots p_{N-1}) & 1 & p_1 & \cdots & (p_1 p_2 \cdots p_{N-2}) \\ (p_2 p_3 \cdots p_{N-1}) & (p_2 p_3 \cdots p_{N-1} p_0) & 1 & \cdots & (p_2 p_3 \cdots p_{N-2}) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ p_{N-1} & p_{N-1} p_0 & p_{N-1} p_0 p_1 & \cdots & 1 \end{bmatrix}^{\mathrm{T}} \tag{2.30}$$

If the Markov chain starts in state 0 with probability one then the initial distribution is simply $\boldsymbol{\pi} = [1, 0, \cdots, 0]^{\mathrm{T}}$. For any matrix $\mathbf{A}$, the vector product $\mathbf{1}^{\mathrm{T}}\mathbf{A}[1, 0, \cdots, 0]^{\mathrm{T}}$ is simply the sum of the elements in the first column of $\mathbf{A}$. Thus for the initial distribution that always starts in state 0, the mean time to absorption in (2.26) becomes:

$$E(T_a) = \frac{1}{\Delta} \cdot [1 + p_0 + p_0 p_1 + \cdots + (p_0 p_1 \cdots p_{N-2})] \tag{2.31}$$

This expression can be represented as

$$E(T_a) = \frac{1 + \sum\limits_{m=0}^{N-2} \prod\limits_{n=0}^{m} p_n}{1 - \prod\limits_{n=0}^{N-1} p_n} \tag{2.32}$$

Although this result was computed for a linear search, it is also applicable to any search order which is a permutation of the integers $0, 1, \cdots, N-1$, as will be done in Section 4.1.

To demonstrate the use of the Markov result in (2.32), the results of Section 2.1 are now derived, where only one hypothesis, say H_k, would properly terminate the search. If the search was currently testing this hypothesis, the Markov chain would be in state k. Thus, the search would properly terminate as per the detection probability, P_d, and would improperly continue as per the miss probability, $1 - P_d$. For all other hypotheses, H_j with $j \neq k$, the probability of terminating the search by selecting that hypothesis would be determined by the false alarm probability, P_{fa}, and the search would properly continue as per the correct dismissal probability, $1 - P_{fa}$. Thus the transition probabilities are now functions of the true terminating hypothesis index, k, and can be denoted as $p_n(k)$.

$$p_n(k) = \begin{cases} 1 - P_d & \text{if } n = k \\ 1 - P_{fa} & \text{if } n \neq k \end{cases} \tag{2.33}$$

$$1 - p_n(k) = \begin{cases} P_d & \text{if } n = k \\ P_{fa} & \text{if } n \neq k \end{cases} \tag{2.34}$$

The mean time to absorption in (2.32) now becomes conditional on the true hypothesis, H_k. In order to find the unconditional expected search time, the conditional mean time is averaged by the a-priori distribution of the true hypothesis, $\Pr(H_k)$:

$$E(T_a) = \sum_{k=0}^{N-1} E(T_a|H_k)\Pr(H_k) \tag{2.35}$$

Assuming that any of the hypotheses is a-priori equally likely then $\Pr(H_n) = 1/N$ for all n. From this assumption and equations (2.32) and (2.35), the unconditional expected search time is found:

$$E(T_a) = \frac{1}{N}\sum_{k=0}^{N-1} \frac{1 + \sum\limits_{m=0}^{N-2}\prod\limits_{n=0}^{m} p_n(k)}{1 - \prod\limits_{n=0}^{N-1} p_n(k)} \tag{2.36}$$

This can be simplified by noting that $\prod_{n=0}^{N-1} p_n(k) = (1-P_d)(1-P_{fa})^{N-1}$ for any k so that the denominator in (2.36) is independent of k and can be brought outside the sum. The numerator can be simplified by noting that

$$\prod_{n=0}^{m} p_n(k) = p_0 p_1 \cdots p_m = \begin{cases} (1-P_{fa})^{m+1} & \text{if } k > m \\ (1-P_d)(1-P_{fa})^m & \text{if } k \le m \end{cases} \tag{2.37}$$

Relying on this fact, the numerator of (2.36) becomes

$$1 + \sum_{m=0}^{N-2}\prod_{n=0}^{m} p_n(k) \quad = \quad 1 + \sum_{m=0}^{k-1}(1-P_{fa})^{m+1} + \sum_{m=k}^{N-2}(1-P_d)(1-P_{fa})^n \tag{2.38}$$

After several straightforward calculations the above expression can be simplified. The resulting expression becomes

$$1 + \frac{1 - P_{fa}}{P_{fa}} \left[1 - (1 - P_{fa})^k + (1 - P_d)(1 - P_{fa})^{k-1} - (1 - P_d)(1 - P_{fa})^{N-2}\right] \tag{2.39}$$

Combining these simplified expressions for the numerator and denominator in (2.36) and simplifying yields:

$$E(T_a) = \frac{1}{P_{fa}} + \frac{1}{NP_{fa}^2} \cdot \frac{1 - (1 - P_{fa})^N}{1 - (1 - P_d)(1 - P_{fa})^{N-1}} \cdot (P_{fa} - P_d) \tag{2.40}$$

Recalling that $\alpha = 1 - P_d$ and $\beta = P_{fa}$ reveals that (2.40) is identical to the previous expected search time in (2.5). Also of interest is the ideal case as the false alarm probability vanishes to zero:

$$\lim_{P_{fa} \to 0} E(T_a) = \frac{N(1 - P_d)}{P_d} + \frac{N + 1}{2} \tag{2.41}$$

This is exactly equation 2.5.6 in [53]. If the detection probability is now allowed to approach one, the ideal case is achieved:

$$\lim_{\substack{P_{fa} \to 0 \\ P_d \to 1}} E(T_a) = \frac{N + 1}{2} \tag{2.42}$$

This result is simply the expected search time for a linear search of N bins, one of which contains a single object placed at random. Additional insight can be obtained into the search performance by considering such ideal situations, i.e., perfect detection and false alarm probabilities, as was done in Section 2.2.

Unless the false alarm probability is zero the possibility of incorrectly terminating the search always exist. Because of this fact, synchronization algorithms generally include some sort of verification phase which makes additional observations in an attempt to decrease the overall false alarm probability to a negligible level. This is usually done by simply dwelling for some extended period of time on the candidate hypothesis. The Markov analysis presented in this section did not include any verification, but can be adapted to include it. Rather than present such an analysis here, a signal flow graph analysis is presented in the next section which is better able to incorporate verification.

2.4 Generalized Signal Flow Graph Approach

The generalized signal flow graph considered in this section is shown in Figure 2.6. This signal flow graph, which is an extension of the basic signal flow graph of [35], will be used to analyze UWB acquisition. As can be seen from Figure 2.6, the generalized signal flow graph allows for an arbitrary search permutation, $\varepsilon(n)$. An arbitrary detection scenario is also now possible such that any of the states can terminate the search by entering the single trapping state in the middle of the graph, termed the ACQ state. Thus there are $N_s + 1$ states in the flow graph: one trapping state and N_s states representing the bins in the uncertainty region. These bins are labeled $0, 1, \ldots, N_s - 1$ and the specific order in which they are searched is determined by the permutation $\varepsilon(n)$ of the integers $0, 1, \ldots, N_s - 1$. The initial distribution of the states is given by $\pi_{\varepsilon(n)}$. The generating function into the acquisition state is defined as

$$P_{ACQ}(z) = \sum_{n=0}^{\infty} p_{ACQ}(n) z^n \tag{2.43}$$

Here z is a complex number and $p_{ACQ}(n)$ is the probability of entering the acquisition state in n transitions. This generating function can be found by various flow graph loop reduction techniques, Mason's gain formula, etc., and is given below, where $\oplus$ represents modulo N_s addition and $\prod_{j=0}^{-1}(\cdot)$ is defined to be unity:

$$P_{ACQ}(z) = \frac{\displaystyle\sum_{k=0}^{N_s-1} \pi_{\varepsilon(k)} \sum_{i=0}^{N_s-1} H_{\varepsilon(i \oplus k)}(z) \prod_{j=0}^{i-1} G_{\varepsilon(j \oplus k)}(z)}{\displaystyle 1 - \prod_{i=0}^{N_s-1} G_{\varepsilon(i)}(z)} \tag{2.44}$$

The path gains $H_{\varepsilon(n)}(z)$ and $G_{\varepsilon(n)}(z)$ are polynomials in the complex variable z and include the transition probabilities and the transition times. For example, a path gain of $0.9z^2$ between any two states means that with probability 0.9 that particular transition occurs and requires 2 'units' of time. The basic unit of time considered here is a single dwell-time, i.e., the amount of time an observer dwells on a particular bin when the search is underway. Dwelling longer on a particular bin increases the overall detection probability or decreases the overall false alarm probability depending on whether or not that bin leads to the acquisition state.

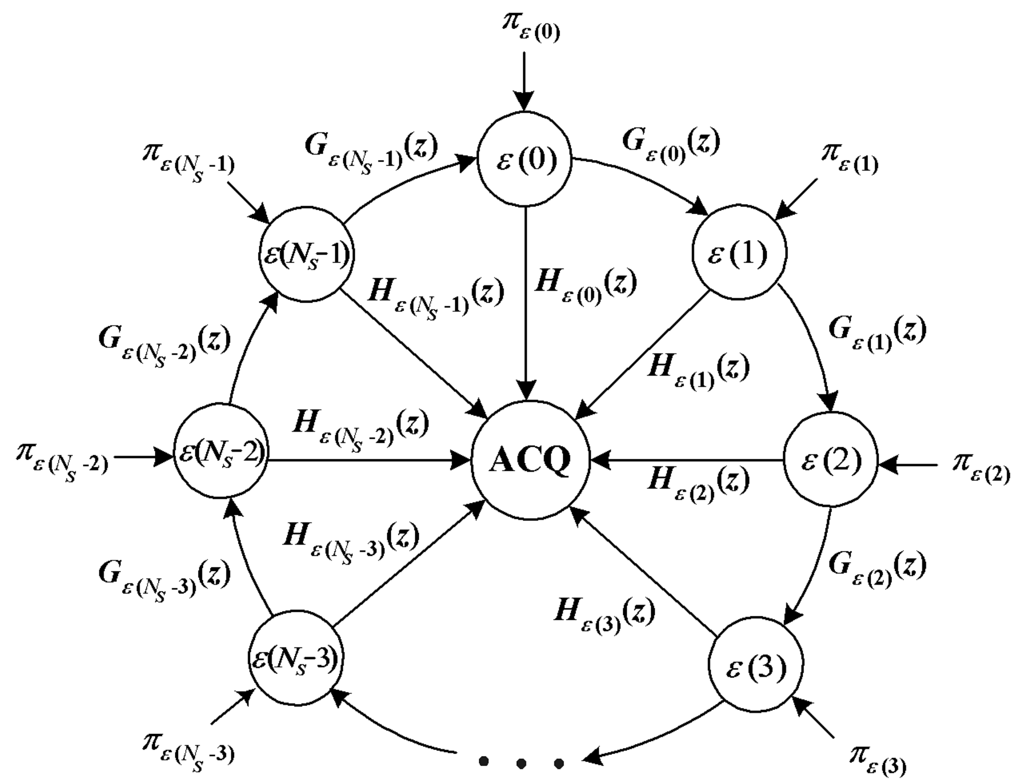

Figure 2.6: Generalized acquisition signal flow graph

The generating function in (2.44) can be used to determined the probability that the search correctly terminated simply by setting $z = 1$. It also yields a complete statistical description of the acquisition time through the inverse transform relation:

$$p_{ACQ}(n) = \frac{1}{2\pi j} \oint \frac{P_{ACQ}(z)}{z^{n+1}} \, \mathrm{d}z \tag{2.45}$$

where $p_{ACQ}(n)$ is the probability mass function of the acquisition time (in integer multiples of the dwell-time) and the contour of integration is a counterclockwise closed circular contour in the region of convergence of $P_{ACQ}(z)$ centered around the origin of the complex plane. Typically, only the first few moments of the acquisition time are analyzed for a specific problem and are related to the first few derivatives of the generating function. In this document only the mean search time is examined which is given as follows:

$$E(T_{ACQ}) = \frac{\mathrm{d}}{\mathrm{d}z} P_{ACQ}(z)\Big|_{z=1} \tag{2.46}$$

Note that the mean acquisition time just listed is in units of dwell-times, so that the mean time in seconds is simply the product of $E(T_{ACQ}) \cdot T_D$, where T_D is the dwell-time in seconds. A general sequence design problem can now be formulated, even though a solution in the general case seems, at best, very difficult to obtain. Namely, the minimum mean search time can be found from the right choice of search sequence:

$$\varepsilon_{min}(n) = \arg \min_{\varepsilon(n)} E\left(T_{ACQ} | \varepsilon(n)\right) \tag{2.47}$$

Several specific examples are now considered to demonstrate the applicability of the generalized signal flow graph and its associated generating function in (2.44). First consider the classical scenario of [35] where the search pattern is linear or consecutive, i.e., $\varepsilon(n) = n$, and there is only one state, say $N_s - 1$, leading to the acquisition state. This implies that the path gains, $H_n(z)$, are all zero except for $H_{N_s-1}(z)$ which is set equal to an arbitrary detection path gain $H_D(z)$. The path gains between states are set equal to $G_{N_s-1}(z) = H_M(z)$ and $G_n(z) = H_F(z)$ for $n = 0, 1, \ldots, N_s - 2$, where $H_M(z)$ is an arbitrary path gain associated with missed detection and $H_F(z)$

is an arbitrary path gain associated with a false alarm. This leads to the generating function in equation (4) of [35], Part I:

$$P_{ACQ}^{(LINEAR,1)}(z) = \frac{H_D(z)}{1 - H_M(z)H_F^{N_s-1}(z)} \sum_{i=0}^{N_s-1} \pi_i H_F^{N_s-i-1}(z) \tag{2.48}$$

Here the superscript on $P_{ACQ}(z)$ represents the search type and the number of detection states, i.e., (LINEAR,1) means a linear search and one detection state.

A second example is examined in [25] in which L consecutive states, 0, 1, $\cdots$, $L-1$, lead to the acquisition state and the search pattern is again linear, $\varepsilon(n) = n$. Thus the path gains are $H_n(z) = H_D(z)$ for $n = 0, 1, \cdots, L-1$ and zero for other n while $G_n(z) = H_M(z)$ for $n = 0, 1, \cdots, L-1$ and $G_n(z) = H_F(z)$ for all other n. The prior initial distribution of the states is uniform, $\pi_n = 1/N_s$ for all n. Using (2.44) with these path gains and initial distribution leads to the generating function in equation (3) of [25]:

$$\begin{aligned} P_{ACQ}^{(LINEAR,L)}(z) = \frac{1}{N_s} \cdot \frac{H_D(z)}{1 - H_M^L(z)H_F^{N_s-L}(z)} \Big[& \sum_{j=0}^{L-1} H_M^j(z) \sum_{i=0}^{N-L} H_F^i(z) \\ & + \sum_{i=1}^{L} \big[L - i + (i-1)H_F^{N_s-L}(z)\big] \cdot H_M^{i-1}(z) \Big] \end{aligned} \tag{2.49}$$

As one final example, it is noted that the generating function found in [48] can also be found using the generalized flow graph of Figure 2.6. The search permutation found in that particular reference, here termed the *Look and Jump* search as in [24], is discussed in more detail below.

The mean acquisition time can be found from the generating function of (2.44) as

$$
\begin{aligned}
E(T_{ACQ}) &= \left.\frac{\mathrm{d}}{\mathrm{d}z} P_{ACQ}(z)\right|_{z=1} \\
&= \frac{Num' \cdot Den - Num \cdot Den'}{Den^2} \qquad (2.50)
\end{aligned}
$$

where Num and Den are the numerator and denominator of (2.44), respectively, evaluated at $z = 1$. Num' is the derivative of the numerator evaluated at $z = 1$:

$$
\begin{aligned}
Num' &= \sum_{k=0}^{N_s-1} \pi_{\varepsilon(k)} \sum_{i=0}^{N_s-1} \Big(\prod_{j=0}^{i-1} G_{\varepsilon(j\oplus k)}(1) \Big) \cdot \\
&\Big[H'_{\varepsilon(i\oplus k)}(1) + H_{\varepsilon(i\oplus k)}(1) \sum_{l=0}^{i-1} \frac{G'_{\varepsilon(l\oplus k)}(1)}{G_{\varepsilon(l\oplus k)}(1)} \Big] \qquad (2.51)
\end{aligned}
$$

Here the summation $\sum_{l=0}^{-1}(\cdot)$ is defined as zero. Den' is the derivative of the denominator evaluated at $z = 1$:

$$
Den' = -\sum_{i=0}^{N_s-1} \frac{G'_{\varepsilon(i)}(1)}{G_{\varepsilon(i)}(1)} \cdot \prod_{j=0}^{N_s-1} G_{\varepsilon(j)}(1) \qquad (2.52)
$$

The factor $G'_{\varepsilon(i)}(1)/G_{\varepsilon(i)}(1)$ appears in the above expressions. Care should be exercised when computing the mean acquisition time if this factor approaches zero. One way around this, which is computationally less efficient, involves using the following alternate expression in the derivative of the numerator and the denominator:

$$\frac{\mathrm{d}}{\mathrm{d}z} \prod_{j=0}^{n-1} G_{\varepsilon(j)}(1) = \sum_{i=0}^{n-1} G'_{\varepsilon(i)}(1) \cdot \prod_{\substack{j=0 \\ j \neq i}}^{n-1} G_{\varepsilon(j)}(1) \tag{2.53}$$

Figure 2.7 gives an example of the mean search time, $E(T_{ACQ})$, for three different search patterns, K consecutive detection states, $N_s = 16$, and the path gains as follows:

$$H_{\varepsilon(i)}(z) = \begin{cases} P_D z & \text{if } i \in \mathcal{I} \\ 0 & \text{else} \end{cases} \tag{2.54}$$

and

$$G_{\varepsilon(i)}(z) = \begin{cases} (1 - P_D)z & \text{if } i \in \mathcal{I} \\ (1 - P_{FA})z + P_{FA} z^{J+1} & \text{else} \end{cases} \tag{2.55}$$

The index set, $\mathcal{I}$, represents those indices i of $\varepsilon(i)$ that lead to the acquisition state. Here these states are assumed to be $0, 1, \cdots, K-1$ so that the size of the set $\mathcal{I}$ is K. Since $\varepsilon(j)$ is simply a permutation of the integers, its inverse $\varepsilon^{-1}(j)$ exists and can be used to produce the index set. That is $\mathcal{I} = \{\varepsilon^{-1}(0), \varepsilon^{-1}(1), \cdots, \varepsilon^{-1}(K-1)\}$ for the example currently being considered, namely K consecutive detection states. The false alarm penalty time is shown in the path gains to be J dwell times where J is a known value. This represents a deterministic amount of time, via some level

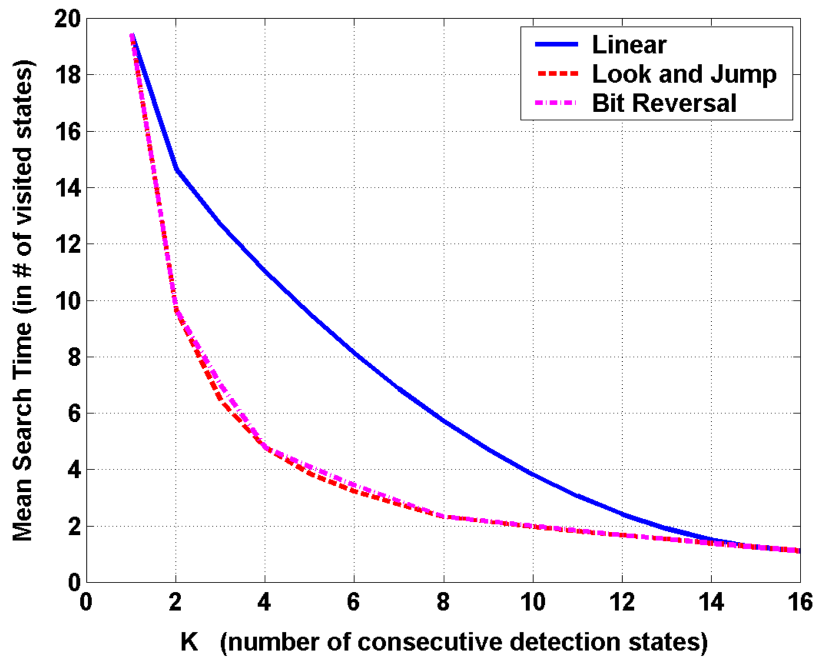

Figure 2.7: Mean acquisition time for $P_D = 0.9$, $P_{FA} = 0.1$, $N_s = 16$, and a false alarm penalty time of $J = 10$

of verification, that is added to the overall search time at every occurrence of a false alarm. In practice the verification phase produces a random penalty time but in order to simplify the acquisition analysis it is often assumed to be fixed. Chapter 5, which deals with fine acquisition, discusses the process of verification in more detail. Appendix A discusses the ($P_D = 1$, $P_{FA} = 1$) scenario for the path gains of (2.54) and (2.55). As will be discussed in Chapter 4 this scenario is equivalent to setting the detection threshold to zero.

The search permutations shown in Figure 2.7 are the *linear*, *look and jump*, and *bit reversal* searches. The linear search, as mentioned above, is simply a consecutive search with $\varepsilon(n) = n$. The index set for this linear case is $\mathcal{I} = \{0, 1, \cdots, K-1\}$. The look-and-jump search is the permutation $0, K, 2K, \cdots, 1, K+1, 2K+1, \cdots$ with the index set being computed as discussed earlier. For example, when $K = 3$ the index set is seen to be $\mathcal{I} = \{0, 6, 11\}$ for $N_s = 16$. As it turns out, the look-and-jump search is the optimum serial search permutation for K consecutive detection states. However, one issue with this type of search is that K, which must be known to generate this search permutation, is related to the number of detectable paths in the multipath channel for the UWB acquisition problem. As will be discussed in the next section, this quantity may not be known to the receiver.

In lieu of this fact, a class of searches is introduced that yield minimum mean search times but do not require knowledge of K. The searches are known as the *base-b reversal searches*. For example, the $b = 2$, or bit, reversal search is the specific permutation obtained from a bit reversal of the binary representation of the integers $0, 1, \cdots, N_s - 1$, assuming N_s is a power of 2. For $N_s = 16$, the bit reversal search pattern is (in binary) $0000, 1000, 0100, 1100, \cdots, 0111, 1111$ or (in decimal) $0, 8, 4, 12, \cdots, 7, 15$. The index set for the bit reversal search is the first K elements of the bit reversal search permutation, namely $\mathcal{I} = \{0, 8, 4, 12, \cdots\}$, since this permutation is its own inverse.

As can be seen in Figure 2.7, the bit reversal search and the look and jump search yield identical mean acquisition times when K is a power of 2 and the two search schemes yield very similar acquisition times for all other values of K. In fact, the mean acquisition time is linear in K between values where K is a power of 2. Similarly, it can also be shown that the base-b reversed indices yield minimum search times when K and N are powers of b. The $b = 2$ case is well suited for digital architectures and also has an advantage over base-b reversal searches for $b > 2$. Specifically, for a fixed N there are more points in the range $K = 1, \cdots, N$ that are powers of 2 than any other power. Thus, when K is not known an appropriate search permutation for minimizing the mean search time is the bit reversal search. As will be seen in the next section, the bit reversal search also yields an efficient hybrid search when multiple observers are introduced given that K is unknown.

2.5 Hybrid Search Analysis using the Generalized Signal Flow Graph

A fully parallel search would minimize the mean acquisition time but is often too complex to actually implement. For a discussion of optimal search techniques utilizing a fully parallel search see [43]. As will be seen in the next chapter, for a short code length of $N_c = 16$ with each frame time divided into $N = 256$ bins, $N_s = N \cdot N_c = 4096$ correlators would be required to implement a parallel search. A hybrid search offers a reasonable trade-off between acquisition time and receiver complexity. It is intuitively obvious that multiple correlators will always reduce the mean acquisition time versus a single correlator since each correlator can search a different location. It also seems reasonable that the individual correlator search patterns should not be independent of one another. In fact, dividing a single search permutation amongst multiple correlators provides an efficient method of searching the entire uncertainty region as quickly as possible with no redundancy.

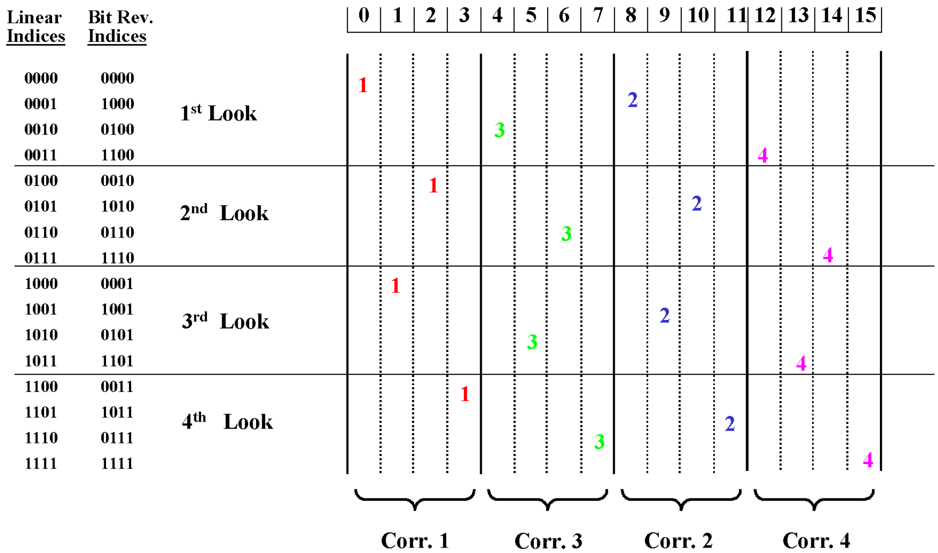

Figure 2.8: Hybrid bit reversal search example for $N_s = 16$ bins and $M = 4$ correlators

The bit reversal search pattern, being the optimum search permutation without knowledge of K, is now divided amongst M correlators. The permutation can be listed as $\beta_0, \beta_1, \beta_2, \cdots, \beta_{N_s-1}$. Then the first correlator is assigned to search as per $\beta_j^{(0)} = \{\beta_0, \beta_M, \beta_{2M}, \cdots\}$, the second correlator searches as per $\beta_j^{(1)} = \{\beta_1, \beta_{M+1}, \beta_{2M+1}, \cdots\}$, and so on. If M and N_s are both powers of 2, then M divides N_s into smaller regions of N_h bins, where N_h is also power of 2. Each correlator then performs a bit reversal search over this smaller region of N_h bins. Figure 2.8 shows an example of this phenomenon for $N_s = 16$ search bins and $M = 4$ correlators. This same phenomenon is also exhibited for other base-b reversal searches when both N_s and M are powers of b.

The generating function for the hybrid search can be computed from the generalized acquisition signal flow graph in Figure 2.6. Before this generating function is found, an alternate method of determining the mean acquisition time is given. Specifically, the initial distribution of the signal flow graph is set to $\pi_{\varepsilon(0)} = 1$ and $\pi_{\varepsilon(j)} = 0$ for all $j \neq 0$ so that the search always starts in state $\varepsilon(0)$. The resulting generating function is conditional on the set of states, $\boldsymbol{K} = [k_1, k_2, \cdots, k_K]^T$, that lead into the acquisition state:

$$P_{ACQ}(z|\boldsymbol{K}) = \frac{\sum_{i=0}^{N_s-1} H_{\varepsilon(i)}(z) \prod_{j=0}^{i-1} G_{\varepsilon(j)}(z)}{1 - \prod_{i=0}^{N_s-1} G_{\varepsilon(i)}(z)} \tag{2.56}$$

The dependence of this conditional generating function on the set $\boldsymbol{K}$ occurs via the path gains, $H_{\varepsilon(j)}(z)$ and $G_{\varepsilon(j)}(z)$, which are both inherently functions of $\boldsymbol{K}$. The mean search time found from the conditional generating function is also conditional on the set $\boldsymbol{K}$ and the overall mean search time is found as $E(T_{ACQ}) = E\left(E(T_{ACQ}|\boldsymbol{K})\right)$ where the outer expectation is with respect to $\boldsymbol{K}$. If the first component of $\boldsymbol{K}$ is uniform on the integers from 0 to $N_s - 1$ while the other values are simply some known offset away from the first random component then the mean search time can be computed as:

$$E(T_{ACQ}) = \frac{1}{N_s} \sum_{k_1=0}^{N_s-1} E(T_{ACQ}|k_1) \tag{2.57}$$

The conditional mean, $E(T_{ACQ}|k_1)$, is computed as the first derivative of the conditional generating function in (2.56) evaluated at $z = 1$. For the UWB acquisition problem examined in the next chapter, the uniform nature of the direct path arrival time, τ_0, must be incorporated into the mean acquisition time. It is fairly straightforward to show:

$$E(T_{ACQ}) = \frac{1}{N_c \cdot T_f} \int_0^{N_c \cdot T_f} E(T_{ACQ}|\tau_0)\, d\tau_0 \tag{2.58}$$

The first M search locations are $\varepsilon(0)$, $\varepsilon(1)$, $\cdots$, $\varepsilon(M-1)$, the next M locations are $\varepsilon(M)$, $\varepsilon(M+1)$, $\cdots$, $\varepsilon(2M-1)$, etc. It will be assumed that M divides N_s evenly into N_s/M regions. Only after the signal flow graph has exited one of these regions has one dwell-time elapsed since M correlation outputs are available every dwell-time. This can be expressed by defining a boundary set $\mathcal{B} = \{M-1, 2M-1, \cdots, N_s-1\}$ and redefining the path gains in terms of this new set. The signal flow graph for the hybrid search is seen in Figure 2.9. As an example, it is assumed that there are K consecutive detection states, starting at state k_1, which is uniformly random on the integers from 0 to $N_s - 1$. The path gains are:

$$H_{\varepsilon(i)}(z) = \begin{cases} P_D z & \text{if } i \in \mathcal{I} \\ 0 & \text{else} \end{cases} \tag{2.59}$$

and

$$
G_{\varepsilon(i)}(z) = \begin{cases} 1 - P_D & \text{if } i \in \mathcal{I} \quad \text{and} \quad i \notin \mathcal{B} \\ (1 - P_D)z & \text{if } i \in \mathcal{I} \quad \text{and} \quad i \in \mathcal{B} \\ 1 - P_{FA} + P_{FA}z^J & \text{if } i \notin \mathcal{I} \quad \text{and} \quad i \notin \mathcal{B} \\ (1 - P_{FA})z + P_{FA}z^{J+1} & \text{if } i \notin \mathcal{I} \quad \text{and} \quad i \in \mathcal{B} \end{cases} \tag{2.60}
$$

As in Section 2.4, the index set, $\mathcal{I}$, represents those indices i of $\varepsilon(i)$ that lead to the acquisition state. This set consists of the elements $k_1, k_1 + 1, \cdots, k_1 + K - 1$ so that $\mathcal{I} = \{\varepsilon^{-1}(k_1),\ \varepsilon^{-1}(k_1 + 1),\ \cdots,\ \varepsilon^{-1}(k_1 + K - 1)\}$. The addition performed here is modulo N_s. Some results, as computed with (2.57), for a hybrid bit reversal search are shown in Figure 2.10 for the scenario just described.

2.6 Sorted Hybrid Search

One method of decreasing the acquisition time for the hybrid case is to first sort the M correlator outputs based upon magnitude. The bins are then examined in order of decreasing magnitude starting with the largest correlator magnitude. A reduction in the mean search time occurs since the less likely bins are searched later, thus potentially reducing the number of false alarm penalties incurred. This idea was first introduced in [36]. The acquisition time cannot be analyzed using the particular signal flow graph discussed earlier since the specific search order now depends upon the outcome of the correlators, which are all random variables. It is possible that a more general framework, obtained by combining Markov Decision Theory and signal

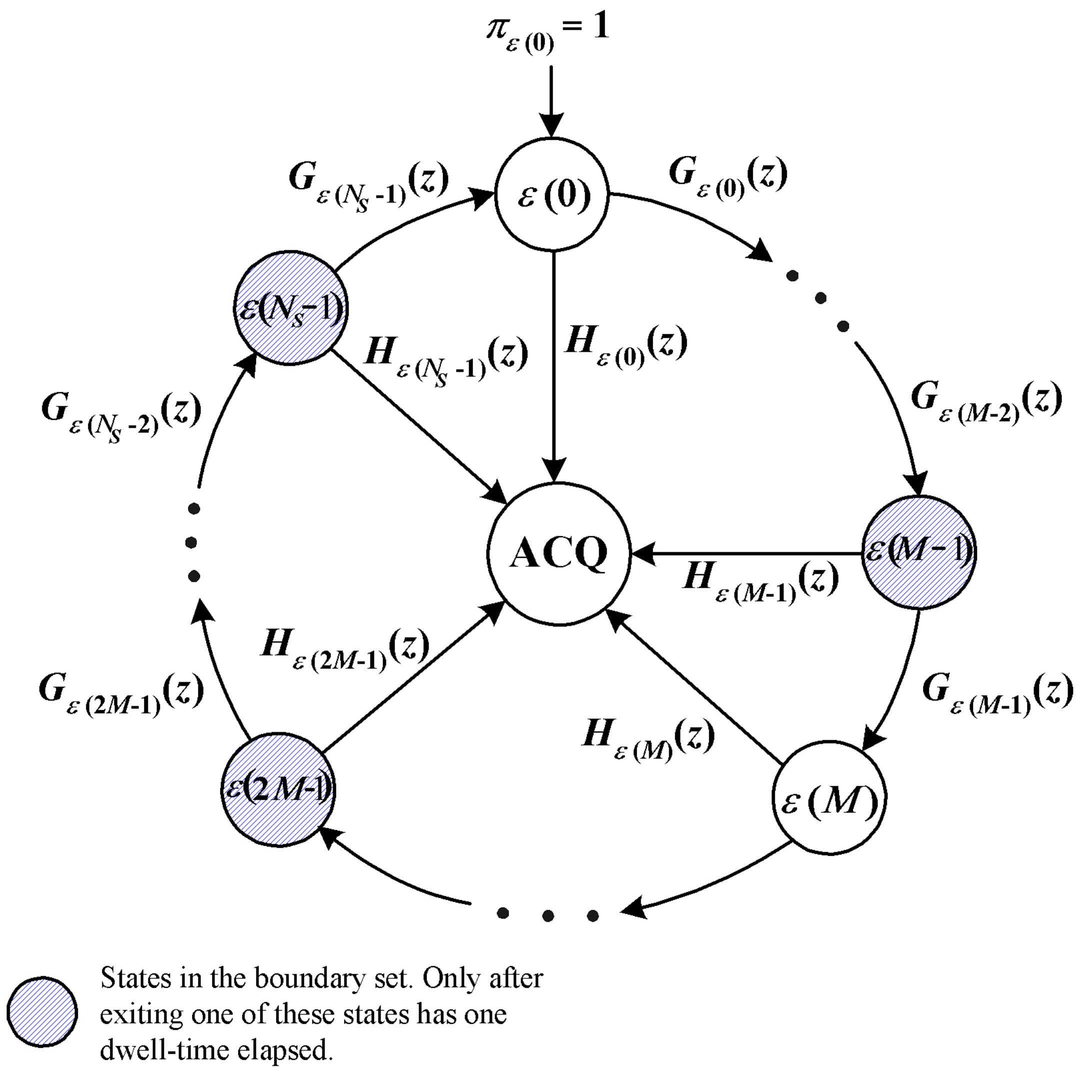

Figure 2.9: Signal flow graph for the hybrid search using M observers

flow graphs, could be used to study this type of search. Computer simulations have revealed, as expected, that sorting in this fashion does indeed reduce the acquisition time with respect to the unsorted hybrid search, as shown in Figure 2.11. As can be seen, there is a greater reduction in mean acquisition time for larger values of M. This is due to the fact that a false alarm is more likely to occur in the group of observations as M increases.

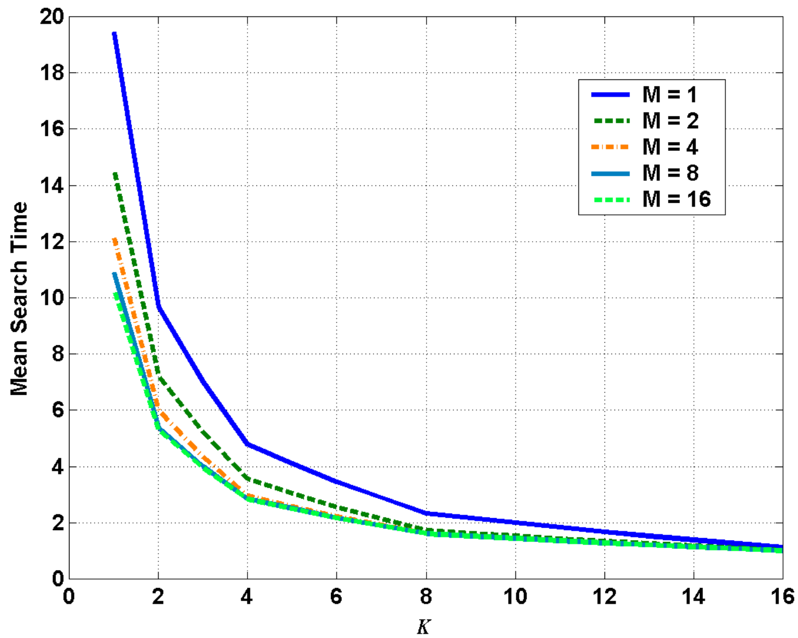

Figure 2.10: Bit reversal search mean acquisition time (in number of state transitions) for the hybrid case of M observers, K consecutive detection bins, $P_D = 0.9$ and $P_{FA} = 0.1$ for each bin, $N_s = 16$, and a false alarm penalty time of $J = 10$

A bound on the sorted hybrid search can be found using the generalized signal flow graph approach of Section 2.5. This bound is obtained by first searching those cells with the largest transition probabilities into the acquisition state, where the

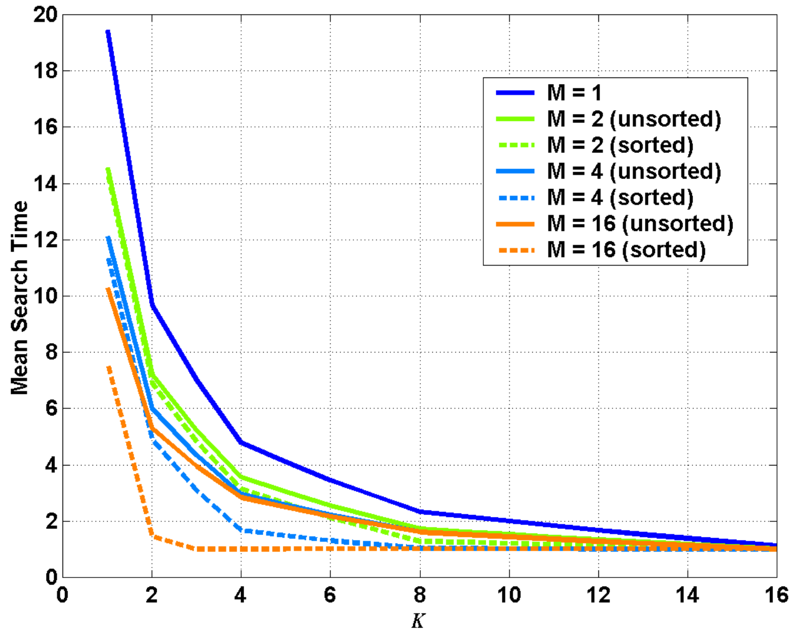

Figure 2.11: Bit reversal and sorted bit reversal search mean acquisition time for the hybrid case of M correlators, K consecutive detection bins, $P_D = 0.9$ and $P_{FA} = 0.1$ for each bin, $N_s = 16$, and a false alarm penalty time of $J = 10$

transition probability from state $\varepsilon(n)$ into the acquisition state is defined as $P_{\varepsilon(n)}$. Specifically this is done by defining a new search order $\varepsilon_s(j)$ where:

$$
\begin{aligned}
\varepsilon_s(0) &= \{\varepsilon(j_0) : P_{\varepsilon(j_0)} = \max_{i=0,\cdots,M-1} P_{\varepsilon(i)}\} \\
\varepsilon_s(1) &= \{\varepsilon(j_1) : P_{\varepsilon(j_1)} = \max_{\substack{i=0,\cdots,M-1 \\ i\neq j_0}} P_{\varepsilon(i)}\} \\
&\vdots \\
\varepsilon_s(M-1) &= \{\varepsilon(j_{M-1}) : P_{\varepsilon(j_{M-1})} = \min_{i=0,\cdots,M-1} P_{\varepsilon(i)}\} \\
\varepsilon_s(M) &= \{\varepsilon(j_M) : P_{\varepsilon(j_M)} = \max_{i=M,\cdots,2M-1} P_{\varepsilon(i)}\} \\
&\vdots \\
\varepsilon_s(2M-1) &= \{\varepsilon(j_{2M-1}) : P_{\varepsilon(j_{2M-1})} = \min_{i=M,\cdots,2M-1} P_{\varepsilon(i)}\} \\
&\vdots
\end{aligned}
\tag{2.61}
$$

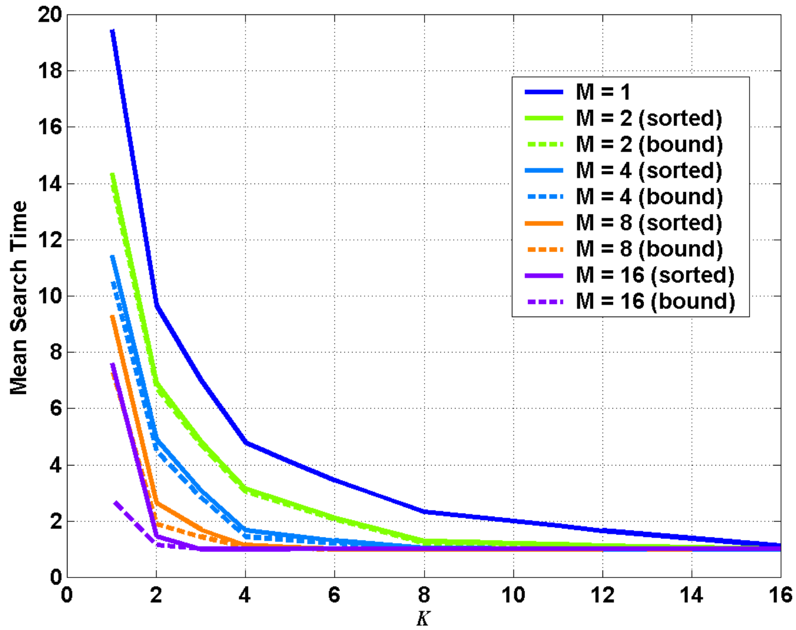

Figure 2.12: Mean acquisition time for the sorted bit reversal search and corresponding bound for the hybrid case of M correlators, K consecutive detection bins, $P_D = 0.9$ and $P_{FA} = 0.1$ for each bin, $N_s = 16$, and a false alarm penalty time of $J = 10$

The signal flow graph, and thus the generating function, from the previous sections are now updated with $\varepsilon_s(j)$ replacing $\varepsilon(j)$. Figure 2.12 shows some bounding results for the hybrid bit reversal search for a very simple case of $N_s = 16$ bins and consecutive acquisition bins as was done in Figures 2.7, 2.10, and 2.11. As can be seen in Figure 2.12, the bound is tighter for lower values of M.

2.7 Bound on Mean Search Time for a Single Observer

This section briefly discusses the possibility of a single observer nearly attaining the search performance of two observers. Recall that an observer is the object performing the search, e.g., the correlator in the previous sections. A single observer bound under perfect conditions ($P_D = 1$ and $P_{FA} = 0$) is derived based upon Markov's inequality and compared to an $M = 2$ hybrid bit reversal search for the same perfect conditions. It is seen that the single observer bound sits just above the two observer hybrid bit reversal search mean acquisition time.

Markov's inequality for a random variable, X, states that if $X \geq 0$ and $\alpha > 0$, then

$$\Pr(X \geq \alpha) \leq \frac{E(X)}{\alpha} \tag{2.62}$$

Here the random variable is assumed to be the search time, $X = T_{ACQ}$, and $\alpha = 1$, $2, \cdots, N_s - K + 1$. The probability $\Pr(T_{ACQ} \geq \alpha)$ can be computed directly as in Section 2.2 since it is also assumed here that $P_D = 1$ and $P_{FA} = 0$. The first few probabilities can be found as

$$\begin{aligned}
\Pr(T_{ACQ} \geq 1) &= 1 \\
\Pr(T_{ACQ} \geq 2) &= 1 - \frac{K}{N} \\
\Pr(T_{ACQ} \geq 3) &= \Pr(T_{ACQ} \geq 2) - \Pr(T_{ACQ} = 2) = \left(1 - \frac{K}{N}\right) \cdot \left(1 - \frac{K}{N-1}\right)
\end{aligned}$$

$$
\begin{aligned}
\Pr(T_{ACQ} \geq 4) &= \left(1 - \frac{K}{N}\right) \cdot \left(1 - \frac{K}{N-1}\right) \cdot \left(1 - \frac{K}{N-2}\right) \\
&\vdots
\end{aligned}
$$

By induction it is seen that for $k = 2, 3, \cdots, N_s - K + 1$:

$$
\Pr(T_{ACQ} \geq k) = \prod_{j=0}^{k-2} \left(1 - \frac{K}{N_s - j}\right) \tag{2.63}
$$

Repeated application of Markov's inequality then yields the greatest lower bound on the mean search time:

$$
E\left(T_{ACQ}\right) \geq \max_{1 \leq k \leq N_s - K + 1} \left(k \cdot \prod_{j=0}^{k-2} \left(1 - \frac{K}{N_s - j}\right)\right) \tag{2.64}
$$

As before $\prod_{j=0}^{-1}(\cdot)$ is defined to be unity, N_s is the number of search locations, or bins, and K is the number of consecutive locations that will terminate the search. The single observer bound is shown in Figure 2.13 along with the hybrid bit reversal search results for the case of $N_s = 16$. It is seen that indeed the bound on single observer performance is very near but slightly above the two observer case. In fact, this same phenomenon was observed for a wide range of scenarios involving different values of N_s. This raises the question: Does a single observer search pattern exist such that the mean search time approaches that of two observers, i.e., such that the bound given here is attained?

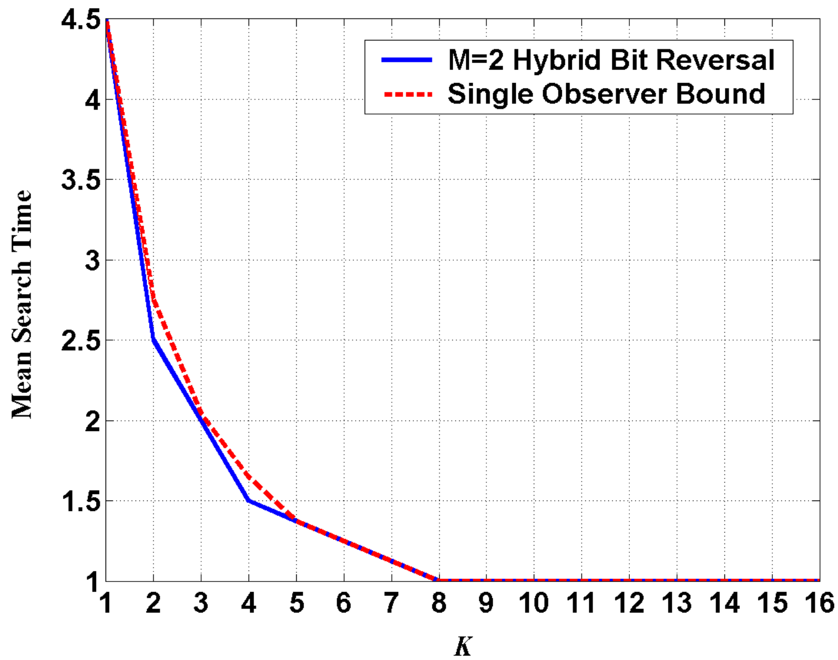

Figure 2.13: Comparison of the mean search time for the single observer bound vs. the two observer hybrid bit reversal search ($N_s = 16, P_D = 1$, and $P_{FA} = 0$)

Chapter 3

Extended Graphical Structures for Search Analysis

In the last chapter a generalized signal flow graph was presented as a very useful tool to analyze the search performance. This performance was given in terms of a complete statistical description of the search time as well as the probability of correctly terminating the search. In fact, these two performance parameters can be used to characterize any search algorithm for purposes of comparison. A search operating characteristic (SOC) curve can be formed by plotting these two basic performance measures against one another. Since the acquisition time is generally random, the mean search time can be used to determine the SOC curve. A general curve is shown in Figure 3.1 which is purely illustrative. Here the probability, β, is the a-priori probability of simply guessing correctly. As the mean time approaches zero, the actual value of the search time must approach zero almost surely since the search time is a non-negative random variable.

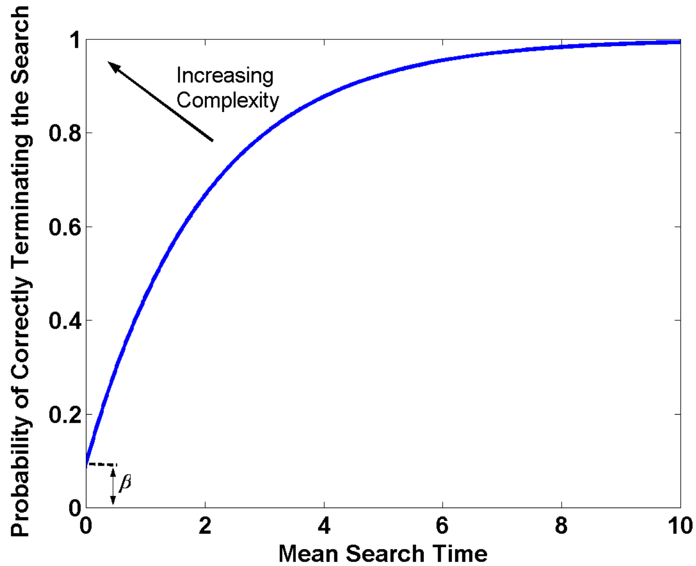

Figure 3.1: Search operating characteristic example

One inherent assumption for the SOC curve shown in Figure 3.1 is that the complexity is fixed. If a more complex search algorithm is allowed, then performance can be improved, essentially generating a completely new SOC curve. Assuming complexity can be quantified in some manner, e.g., the number of observers performing the search, it can be added as another dimension to the SOC curve. As shown in the figure, as complexity increases the curve is pushed toward the upper left corner as the mean search time decreases and the probability increases. If complexity is merely the number of observers then this graph approaches the curve specified by a fully parallel search.

While any type of search can be described with a curve similar to that of Figure 3.1, one specific example is a serial search which is widely used because of its low complexity. The generalized signal flow graph introduced in the last chapter can be used to analyze this type of search (as well the hybrid serial/parallel search), where it was inherently assumed that the serial search pattern was a permutation of the search locations. In the next section, 3.1, additional insight is given regarding this fact and search graphs are discussed which do not require the search pattern to be a permutation. Also discussed in this chapter are other graphical structures which find utility in analyzing search performance of arbitrary algorithms and/or methodologies. Specifically, the concept of a self-similar signal flow graph is introduced in Section 3.2. This concept arises if one assumes that instead of terminating the search a new search is initiated upon completion of the previous one. For the case of UWB acquisition, this type of graph represents the combined *coarse* and *fine* acquisition process described in Chapters 4 and 5, first locating the multipath cluster then locating the strongest paths within the cluster.

One highly complex type of search which tends to offer very good performance are random-dwell-time searches. These searches utilize concepts from the field of sequential analysis [61] and M-ary sequential hypothesis testing [3] [4] [5] [26] [27]. These types of searches have been used in many different areas, specifically the area of spread-spectrum code acquisition for DS-CDMA [2] [62], as well as UWB [66]. The graphical techniques introduced in the last chapter and expanded upon here will be applied to the analysis of these random-dwell-time schemes in Section 3.3.

3.1 Search Graphs

Given a *search space*, $\mathcal{S}$, consisting of N_s elements $\{0, 1, \cdots, N_s - 1\}$, a group of M observers searching for an *index set*, $\mathcal{I}$, of elements from the search space, one can

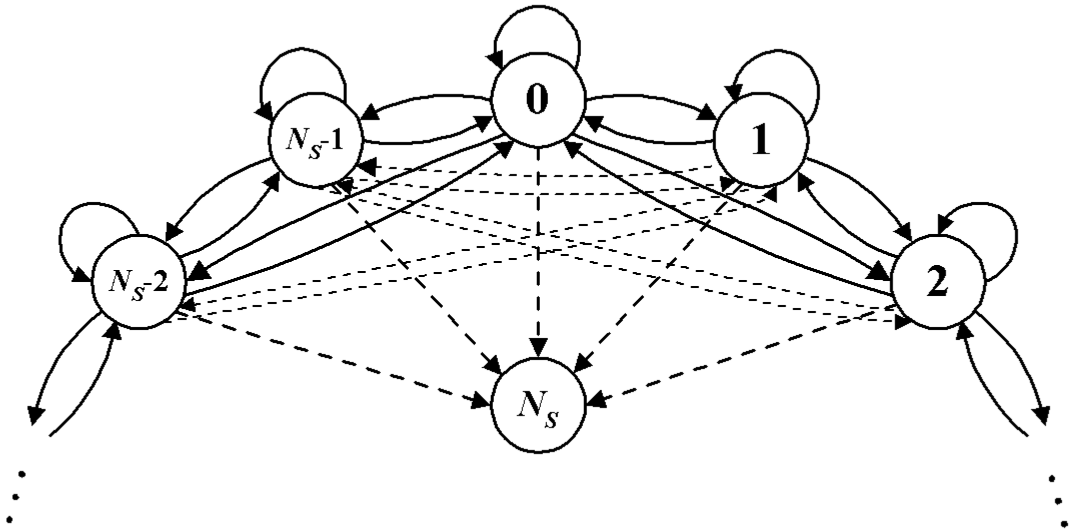

Figure 3.2: Search graph with $N_s + 1$ states.

construct an arbitrary *search graph.* Such a graph is shown in Figure 3.2. This graph is obtained by mapping each element of $\mathcal{S}$ into a state on the graph and, without loss of generality, element 0 becomes state 0, etc. There is an extra trapping state, N_s, which represents termination of the search.

Each branch between any two states of the graph in Figure 3.2 is some function of the search algorithm, the associated error probabilities, as well as the amount of time spent searching a particular state, and possibly other arbitrary information such as a vector of costs, rewards, probabilities, etc. In the most general form this function and its arguments are dependent upon the particular transition (or some number of past transitions) and are also dependent upon when that transition is made, so that the function may change over time. Each state in the graph connects to every other state as evident by the branches between states (although not all these branches are

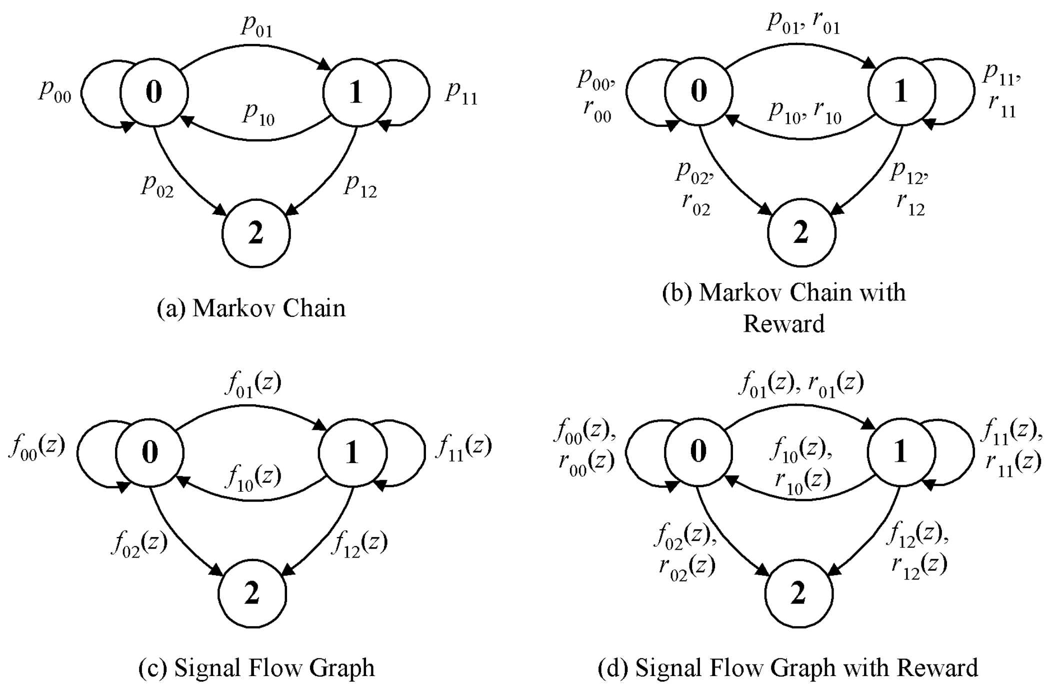

Figure 3.3: Specific examples of search graphs

illustrated in the figure). Several three-state examples are shown in Figure 3.3 which illustrate specific types of graphs.

The first example is a Markov chain as shown in Figure 3.3a. In this case the states are connected by the *transition probabilities* and p_{ij} represents, for i and j both integers between 0 and N_s, the probability of entering state j starting from state i. If the transition probabilities are independent of time then the chain is said to be *homogeneous*. The transition probabilities leaving any state, i, must sum to unity: $\sum_{j=0}^{N_s} p_{ij} = 1$. Section 2.3 presented a Markov chain analysis of the mean search time. One problem with this type of graph when used as to analyze a search is that the amount of time spent examining any particular state cannot be easily incorporated into the graph. The Markov chain inherently operates in discrete-time where transitions from one state to another occur at integer multiples of some basic, minimum unit of time. If each state has associated with it some examination time which is an integer multiple of this minimum unit of time, then extra states can simply be added to the graph. This produces an entirely different graph and doesn't allow these examination times to be easily parameterized. One way around this is to associate this examination time with the transition. This leads to the next type of graph discussed in this section, a Markov chain with reward.

As shown in Figure 3.3b, a Markov chain with reward has the associated transition probabilities between states as well as transition rewards, r_{ij}. A reward may also be associated with a single state, i, by the relationship:

$$r_i = \sum_j p_{ij} r_{ij} \tag{3.1}$$

The expected reward associated with a transition from state i is denoted r_i. The advantage of this type of structure, as alluded to earlier, is that the independent times associated with the various states can now be modeled as a transition reward. These times may include examination times (or dwell-times), false alarm penalty times, etc. A vast body of literature exists pertaining to Markov chains with rewards, e.g., see [14]. These graphical structures form the basic building block of Markov Decision Theory.

The third type of graph shown in Figure 3.3c is a signal flow graph. The branches between states are known as *path gains* and are functions of the complex variable z. As was seen in Section 2.4, the signal flow graph is a very useful tool for analyzing search performance. It will be assumed that the variable, z, is not a function of the transition between states. One example where z is a function of the transition, however, results in a Markov chain with transition probabilities which are complex. For instance, if the transition probability is p_{ik} and the transition reward is r_{ik}, then setting the path gain to $p_{ik} + jr_{ik}$ where j is the imaginary number, $\sqrt{-1}$, produces a signal flow graph in which the path gain variable z is indeed a function of the transition. The other possibility is that z remains independent of the transition and the path gains are indeed functions of this complex variable. For the same transition probability and reward just mentioned the path gain can be set to

$$f_{ik}(z) = p_{ik} z^{r_{ik}} \tag{3.2}$$

If the reward, r_{ik}, is the amount of time required to transition from state i to state k then the moment generating function of the search time can be determined from the graph, as described in Section 2.4.

Since Markov decision theory and the dynamic programming algorithm work on Markov chains with rewards, a signal flow graph with path gains given in (3.2) forms an alternate framework for Markov decision theory. Thus the dynamic programming algorithm can be generalized when (3.2) is not imposed, but the path gain is left as some arbitrary function of z and another function, $r_{ij}(z)$, is also associated with each transition. This type of graphical structure, shown in Figure 3.3d, is termed a *signal flow graph with rewards.*

Having briefly reviewed several specific types of graphs, we now return to the general search graph of Figure 3.2 and discuss possible applications of it as well as some general techniques for reducing the search time. Specifying a *search algorithm* sets restrictions and modifies the basic graphical structure as per the algorithm. One specific example would be to prune the branches of the graph such that each state only had one exit path (other than possibly a path to the termination state). In order to cover the entire search space, this pruning must be done such that all states are still connected. Thus, this pruning is done in such a way that the graph has only one cycle, which is of length N_s. This implies that the sequence mapped out by the state transitions is simply a permutation of the N_s states of the graph. This simple search algorithm makes sense intuitively since revisiting incorrect states only increases the search time. As was seen in Section 2.4, assuming the specific search permutation is $\varepsilon(n)$ for states $n = 0, 1, \cdots, N_s - 1$ leads to the graph shown in Figure 2.6. Minimization of the search time by selecting the best permutation, $\varepsilon(n)$, was investigated in Chapter 2.

3.2 Self-Similar Signal Flow Graphs

The graphical structure introduced here will be required for the search analysis of Chapter 5. This structure is known as a *self-similar signal flow graph* and one example is shown in Figure 3.4. This is an extension of the generalized signal flow graph of Section 2.4. This structure is useful when there exists an intermediate stage prior to terminating the search. This intermediate stage itself represents a search which, in

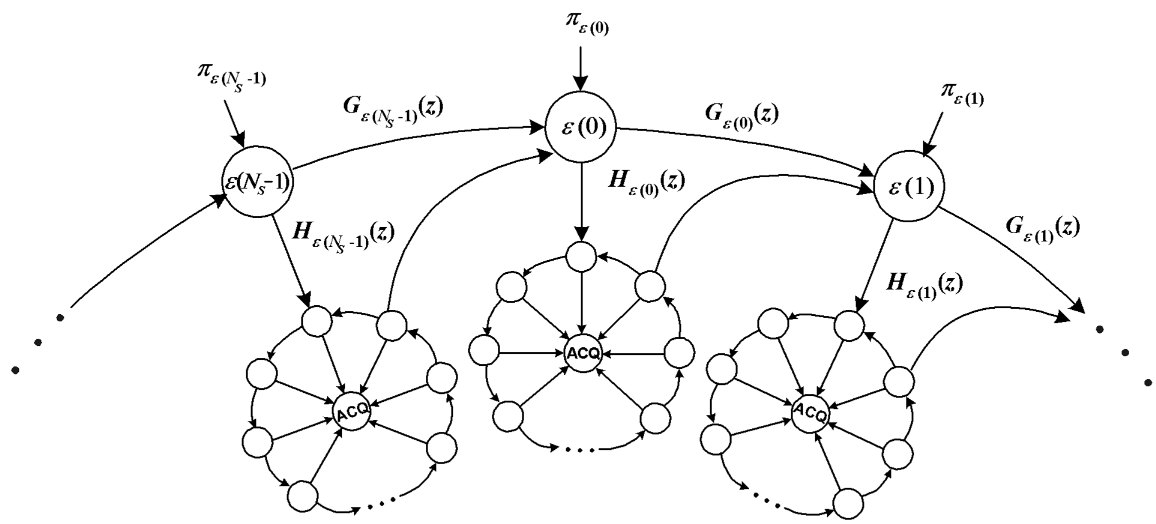

Figure 3.4: Self-similar signal flow graph with two rings

general, can be from a completely new search space. The example shown in Figure 3.4 consists of only an outer *ring*, which is defined as a circular arrangement of states as in a search graph, and N_s inner rings. In general, however, many more levels are possible and it is also possible for more than one state to connect into and out of the rings above and below the current ring. The final state which indicates search termination is labeled the acquisition state, ACQ. The two ring example shown in Figure 3.4 is sufficient for the analysis of Chapter 5 where the process of fine acquisition is examined. Now, however, the general subject of multi-ring self-similar signal flow graphs is briefly discussed.

If the outermost ring of the self-similar signal flow graph represents the search space $\mathcal{S}_0$ with $|\mathcal{S}_0| = N_0$ states, then the next ring consists of N_0 graphs, each representing a new search space. It will be assumed that the size of these new spaces is constant across this inner ring, so that each graph in the ring has N_1 states. The next ring will consist of N_1 graphs each of size N_2, and so on. This is depicted graphically in Figure 3.5. The total number of states, N_s, in this r-ring self-similar signal flow graph, not accounting for the termination state, is found to be:

$$N_s = \sum_{k=0}^{r-1} \prod_{i=0}^{k} N_i \tag{3.3}$$

If each of the termination states in the center of the final ring is considered to be the same state then there is obviously only one such state. However, if each of these states is distinct then the total number of these acquisition states, denoted as N_a, is as follows:

$$N_a = \prod_{i=0}^{r-2} N_i \tag{3.4}$$

As before $\prod_{i=0}^{-1}(\cdot)$ is defined to be unity. One interesting observation of (3.3) is the recurrence of this type of summation of a product, e.g., see (2.14), (2.32), or (2.44).

Under the assumption that only one acquisition state exists, one can compute an overall generating function into this state in precisely the same way that was done in Section 2.4. One straightforward way of producing this generating function is to simply reduce the self-similar signal flow graph to the generalized form of Figure 2.6. Such a reduction is always possible and reveals one of the main reasons the graph has been termed *self-similar*. The other reason comes from the fact that if one starts

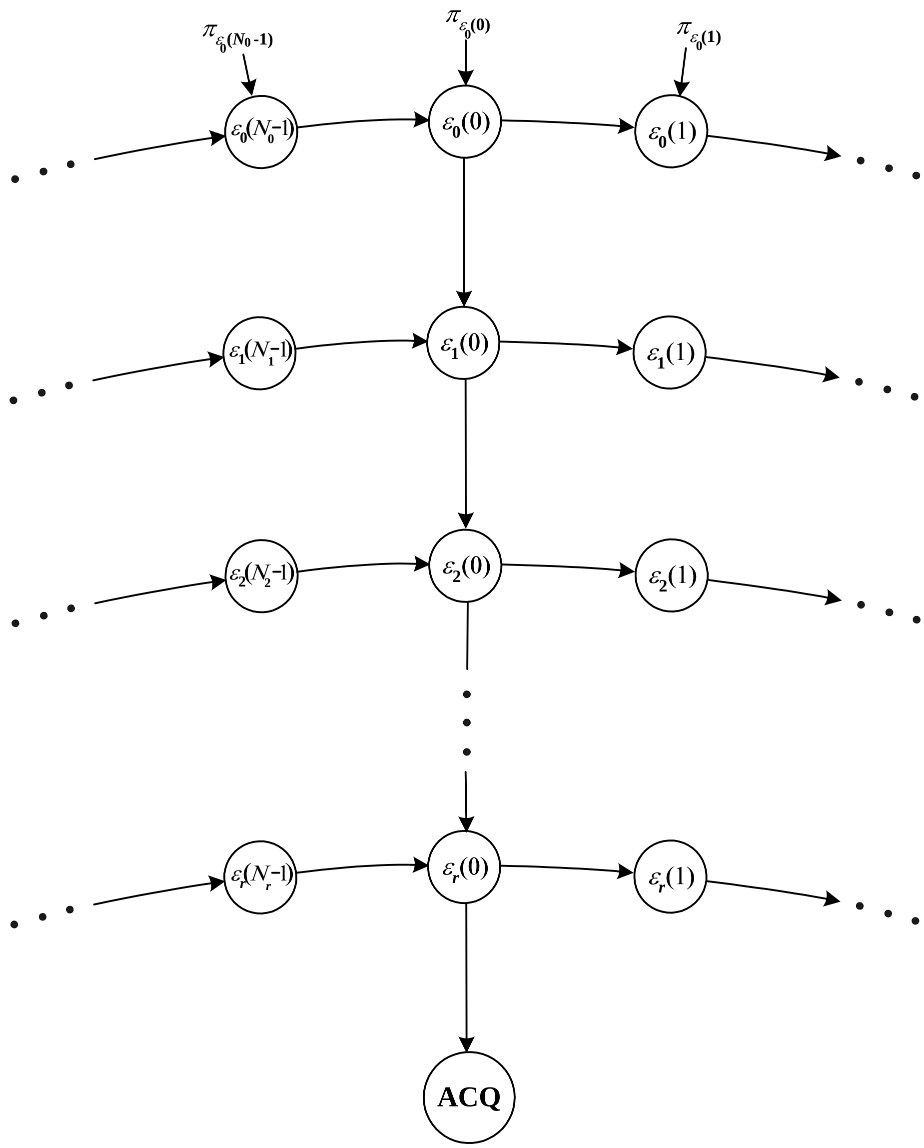

Figure 3.5: Self-similar signal flow graph with r rings

with the graph of Figure 2.6 then 'examination' of any path gain into the acquisition state reveals a new circular graph identical in structure to the outer graph.

The process of fine acquisition to be studied in Chapter 5 will use the graph introduced in this section to determine the search performance for a number of different stopping and verification criteria. Several specific stopping criteria are examined in the following sections. The first is a simple exhaustive search of the entire fine acquisition uncertainty region and leads to a stopping time which is deterministic. A second, more general, example is seen in Section 3.2.2 where random stopping time verification algorithms are analyzed. This leads to the general discussion of sequential search analysis in Section 3.3.

3.2.1 Deterministic Stopping Time

The graph for the example currently under consideration is shown in Figure 3.6. Recall from Section 1.3 that the multipath channel consists of arrival paths clustered into groups. The job of the coarse acquisition search is to locate this cluster while the fine acquisition process attempts to locate a set of useable paths within this cluster. The end result is basically a channel estimate that can be used for data detection and for optimum performance in the data detector the strongest paths should be estimated. One way in which these strongest paths can be located, or any set of paths for that matter, is to simply search the location immediately adjacent to the coarse search detection point, say a fixed amount in either direction. For the hybrid case there exists a total of M observers and here it is assumed that all M of these observers are diverted to fine search from coarse search. Thoughts about the general resource allocation problem of diverting some smaller number of observers to the fine search are discussed in Chapter 6.

For the example currently being considered, the kM bins to the right of the current bin will be searched followed by the kM bins to the left. The bins with the largest observed values (correlator magnitude) will be chosen as the strongest paths. If these largest observations satisfy some sort of verification criterion then the search is terminated, otherwise it continues along the outer ring. Since all $2kM$ bins surrounding the current bin will be searched, the order in which they are searched is irrelevant.

As can be seen from the graph of Figure 3.6 the coarse search exits the outer ring at state $\varepsilon(n)$ as per the path gain $\tilde{H}_{\varepsilon(n)}(z)$. This path gain is a function of the coarse search detection criterion and in its simplest form is

$$\tilde{H}_{\varepsilon(n)}(z) = P_{\varepsilon(n)} \tag{3.5}$$

Here the transition probability, $P_{\varepsilon(n)}$, is inherently conditional on whether or not state $\varepsilon(n)$ is a proper detection state. This will become more evident shortly when a specific example involving the detection and false alarm probabilities is examined. The function $\tilde{G}_{\varepsilon(n)}(z)$ determines when the next state in the coarse search is visited and keeping in line with the path gain just defined yields

$$\tilde{G}_{\varepsilon(n)}(z) = \begin{cases} 1 - P_{\varepsilon(n)} & \text{if } n \notin \mathcal{B} \\ \left(1 - P_{\varepsilon(n)}\right) z & \text{if } n \in \mathcal{B} \end{cases} \tag{3.6}$$

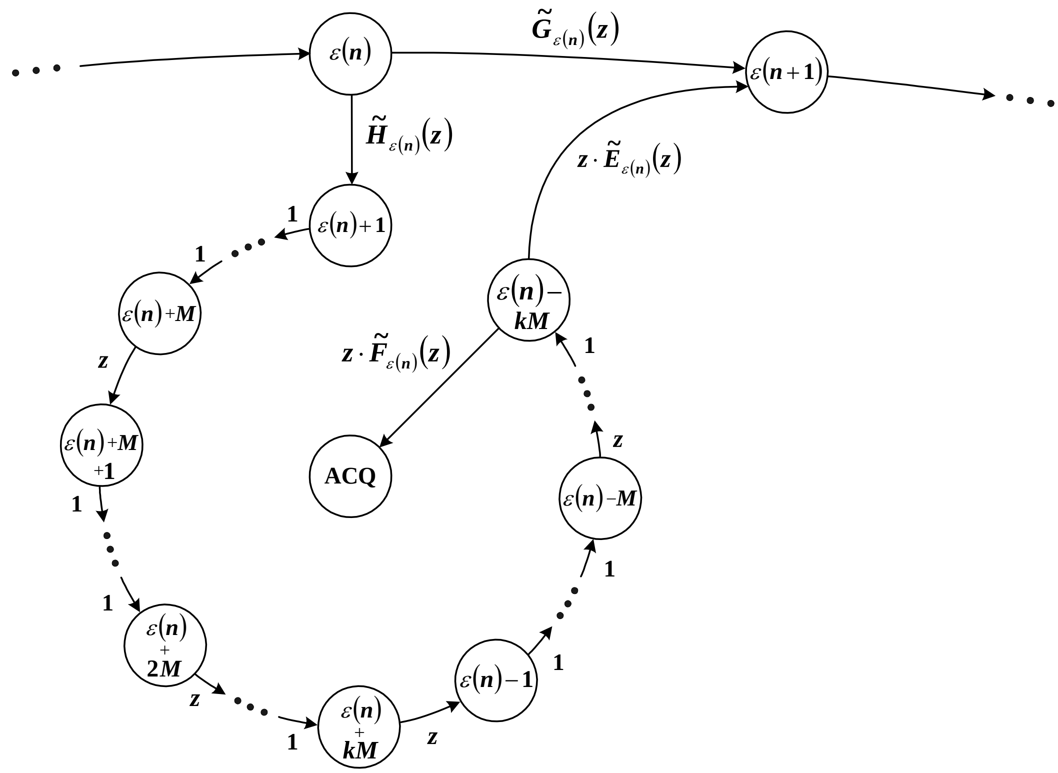

Figure 3.6: Self-similar signal flow graph representation of a fine acquisition search with a fixed, deterministic stopping time and M observers

.

As in Section 2.5, the boundary set is defined as $\mathcal{B} = \{M-1, 2M-1, \cdots, N_s-1\}$ and only upon exiting one of these states has a dwell-time elapsed in the coarse search. Again M is assumed to divide N_s evenly.

The states comprising the inner ring of Figure 3.6 are labeled $\varepsilon(n)+1, \varepsilon(n)+2, \cdots, \varepsilon(n)+kM, \varepsilon(n)-1, \cdots, \varepsilon(n)-kM$ where the addition and substraction is performed modulo N_s. Most of the path gains around this inner ring are simply unity and only after M consecutive states have been visited has a single dwell-time elapsed as evident by the path gain of z. Once all $2kM$ states have been visited in the inner ring, which will require $2k$ dwell-times, either the search terminates or the coarse search continues on the outer ring. This is determined by the path gains $z \cdot \tilde{F}_{\varepsilon(n)}(z)$ and $z \cdot \tilde{E}_{\varepsilon(n)}(z)$. The extra factor of z in these path gains is due to the dwell-time required for the last group of M observations, thus completing the search of the $2kM$ state uncertainty region about state $\varepsilon(n)$. $\tilde{F}_{\varepsilon(n)}(z)$ is a function of the verification criterion which, for this example, is simply

$$\tilde{F}_{\varepsilon(n)}(z) = \begin{cases} P_{\mathrm{v},\varepsilon(n)} \cdot z & \text{if } n \in \mathcal{I} \\ 0 & \text{else} \end{cases} \tag{3.7}$$

The verification probability, $P_{\mathrm{v},\varepsilon(n)}$, is set by the stopping criterion, properly selected to ensure that the strongest paths are indeed strong enough for data detection. The verification probability will be discussed in more detail shortly and will be left arbitrary for now so that the mean search time can be parameterized without selecting a particular verification criterion. The last function to be defined takes the search from the inner ring back to the outer ring. Here it is assumed, as was the case in Sections

2.4 and 2.5, that a final level of verification exists which confirms with high probability that the search should have indeed ended. This assumption will be removed shortly and the functions $\tilde{F}_{\varepsilon(n)}(z)$ and $\tilde{E}_{\varepsilon(n)}(z)$ will be redefined accordingly.

$$\tilde{E}_{\varepsilon(n)}(z) = \begin{cases} 1 - P_{\mathrm{v},\varepsilon(n)} & \text{if } n \in \mathcal{I} \quad \text{and} \quad n \notin \mathcal{B} \\ (1 - P_{\mathrm{v},\varepsilon(n)})z & \text{if } n \in \mathcal{I} \quad \text{and} \quad n \in \mathcal{B} \\ 1 - P_{\mathrm{v},\varepsilon(n)} + P_{\mathrm{v},\varepsilon(n)} z^{J} & \text{if } n \notin \mathcal{I} \quad \text{and} \quad n \notin \mathcal{B} \\ (1 - P_{\mathrm{v},\varepsilon(n)})z + P_{\mathrm{v},\varepsilon(n)} z^{J+1} & \text{if } n \notin \mathcal{I} \quad \text{and} \quad n \in \mathcal{B} \end{cases} \tag{3.8}$$

As was mentioned earlier, the self-similar nature of the graph of Figure 3.6 allows for a reduction to the generalized form of Figure 2.6. The path gains for this representation are:

$$H_{\varepsilon(n)}(z) = \tilde{H}_{\varepsilon(n)}(z) \cdot \tilde{F}_{\varepsilon(n)}(z) \cdot z^{2k} \tag{3.9}$$

$$G_{\varepsilon(n)}(z) = \tilde{G}_{\varepsilon(n)}(z) + \tilde{H}_{\varepsilon(n)}(z) \cdot \tilde{E}_{\varepsilon(n)}(z) \cdot z^{2k} \tag{3.10}$$

Substituting (3.5), (3.6), (3.7), and (3.8) into the above paths gain equations reveals that:

$$H_{\varepsilon(n)}(z) = \begin{cases} P_{\varepsilon(n)} \cdot P_{\mathrm{v},\varepsilon(n)} \cdot z^{2k+1} & \text{if } n \in \mathcal{I} \\ 0 & \text{else} \end{cases} \tag{3.11}$$

$$G_{\varepsilon(n)}(z) = \begin{cases} 1 - P_{\varepsilon(n)} + P_{\varepsilon(n)} \cdot (1 - P_{\mathrm{v},\varepsilon(n)}) \cdot z^{2k} & \text{if } n \in \mathcal{I},\ n \notin \mathcal{B} \\ (1 - P_{\varepsilon(n)}) \cdot z + P_{\varepsilon(n)} \cdot (1 - P_{\mathrm{v},\varepsilon(n)}) \cdot z^{2k+1} & \text{if } n \in \mathcal{I},\ n \in \mathcal{B} \\ P_{\varepsilon(n)} \cdot \left[1 - P_{\mathrm{v},\varepsilon(n)} + P_{\mathrm{v},\varepsilon(n)} \cdot z^{J}\right] z^{2k} + 1 - P_{\varepsilon(n)} & \text{if } n \notin \mathcal{I},\ n \notin \mathcal{B} \\ P_{\varepsilon(n)} \cdot \left[1 - P_{\mathrm{v},\varepsilon(n)} + P_{\mathrm{v},\varepsilon(n)} \cdot z^{J}\right] z^{2k+1} + (1 - P_{\varepsilon(n)}) \cdot z & \text{if } n \notin \mathcal{I},\ n \in \mathcal{B} \end{cases}$$

These path gains can then be used directly to find the mean acquisition time, e.g. see (2.50). The verification probability is computed directly based upon the specific verification criterion selected. For example, the scenario involving the M largest observations exceeding some predetermined value will be examined in Chapter 5. There, the verification probability will be determined from the order statistics of the non-IID collection of observations. As will be seen in Chapter 5, for the verification process to be accurate it is required that the verification probability be as large as possible for proper detection bins, $n \in \mathcal{I}$, and as small as possible for the other bins, $n \notin \mathcal{I}$. As can be seen from the path gains just listed above, this requirement also minimizes the mean acquisition time. Specifically, if $P_{\mathrm{v},\varepsilon(n)} = 1$ when $n \in \mathcal{I}$ and zero otherwise, then the acquisition time is minimized with respect to the verification probability. For now, however, the effect of verification on the acquisition time will be examined by directly varying the verification probability. Two example scenarios are considered below. The first case is no verification and the second is verification at low signal-to-noise ratios.

No Verification

For no verification the probability, $P_{\mathrm{v},\varepsilon(n)}$, is set to unity. The path gains for this situation become:

$$H_{\varepsilon(n)}(z) = \begin{cases} P_{\varepsilon(n)} \cdot z^{2k+1} & \text{if } n \in \mathcal{I} \\ 0 & \text{else} \end{cases} \tag{3.12}$$

and

$$G_{\varepsilon(n)}(z) = \begin{cases} 1 - P_{\varepsilon(n)} & \text{if } n \in \mathcal{I} \quad \text{and} \quad n \notin \mathcal{B} \\ (1 - P_{\varepsilon(n)}) \cdot z & \text{if } n \in \mathcal{I} \quad \text{and} \quad n \in \mathcal{B} \\ 1 - P_{\varepsilon(n)} + P_{\varepsilon(n)} \cdot z^{J+2k} & \text{if } n \notin \mathcal{I} \quad \text{and} \quad n \notin \mathcal{B} \\ (1 - P_{\varepsilon(n)}) \cdot z + P_{\varepsilon(n)} \cdot z^{J+2k+1} & \text{if } n \notin \mathcal{I} \quad \text{and} \quad n \in \mathcal{B} \end{cases} \tag{3.13}$$

For illustrative and comparative purposes, assume that the transition probabilities, $P_{\varepsilon(n)}$, are defined in terms of constant detection and false alarm probabilities, P_D and P_{FA}, respectively. Immediately it is seen that (3.13) is equivalent to (2.60) with the false alarm penalty time replaced by the sum of verification time and false alarm penalty time, i.e., J is replaced by $J + 2k$. In Section 2.5, the verification time was not accounted for, but if it had been then (2.59) would have simply been $H_{\varepsilon(n)}(z) = P_D z^{2k+1}$ for $n \in \mathcal{I}$ and zero otherwise. This is precisely the path gain in (3.12). Thus the mean search time results of Section 2.5, written as $MST_{coarse}(J)$, and the figures therein yield the mean search time here, denoted $MST_{fine}(J - 2k)$

for all other parameters held constant. Both these mean search times are functions of J and are related as follows:

$$MST_{fine}(J-2k) = MST_{coarse}(J) + 2k \tag{3.14}$$

Verification at Low Signal-to-Noise Ratios

As will be seen in Chapter 5, as the signal-to-noise ratio is decreased the verification probabilities for bins in the detection set approach the verification probabilities for bins outside the detection set. Thus for adequately small signal-to-noise ratios the verification probability can be assumed constant, say P_{v}. The mean search time can then be minimized with proper selection of P_{v}. One way to find the minimum mean search time is to solve for P_{v} in the equation:

$$\frac{\mathrm{d}}{\mathrm{d}P_{\mathrm{v}}}\left(\frac{\mathrm{d}}{\mathrm{d}z}\, P_{ACQ}(z|P_{\mathrm{v}})\big|_{z=1}\right) = 0 \tag{3.15}$$

Instead of solving this equation, which is a complicated endeavor, we simply plot several specific examples of the mean acquisition time versus P_{v}, as shown in Figures 3.7 through 3.10. These results are plotted for $N_s = 16$ bins in the search space and the mean search time is measured in terms of the number of states visited. The number of detection bins, K, is varied from 1 to 16. The state number of the first of these detection bins is assumed uniform on the discrete set $0, 1, \cdots, 15$ and the initial distribution is $\pi_{\varepsilon(0)} = 1$ so that (2.57) yields the mean search time. Notice from these graphs, that the $K = 16$ case is independent of the false alarm penalty time, J, since there are no false alarms if all bins in the search space are detection bins. Figures 3.7 and 3.8 show the results for a linear and bit reversal search, respectively, and for a small verification time of $2k = 4$. Figures 3.9 and 3.10 show results for a large verification time of $2k = 200$.

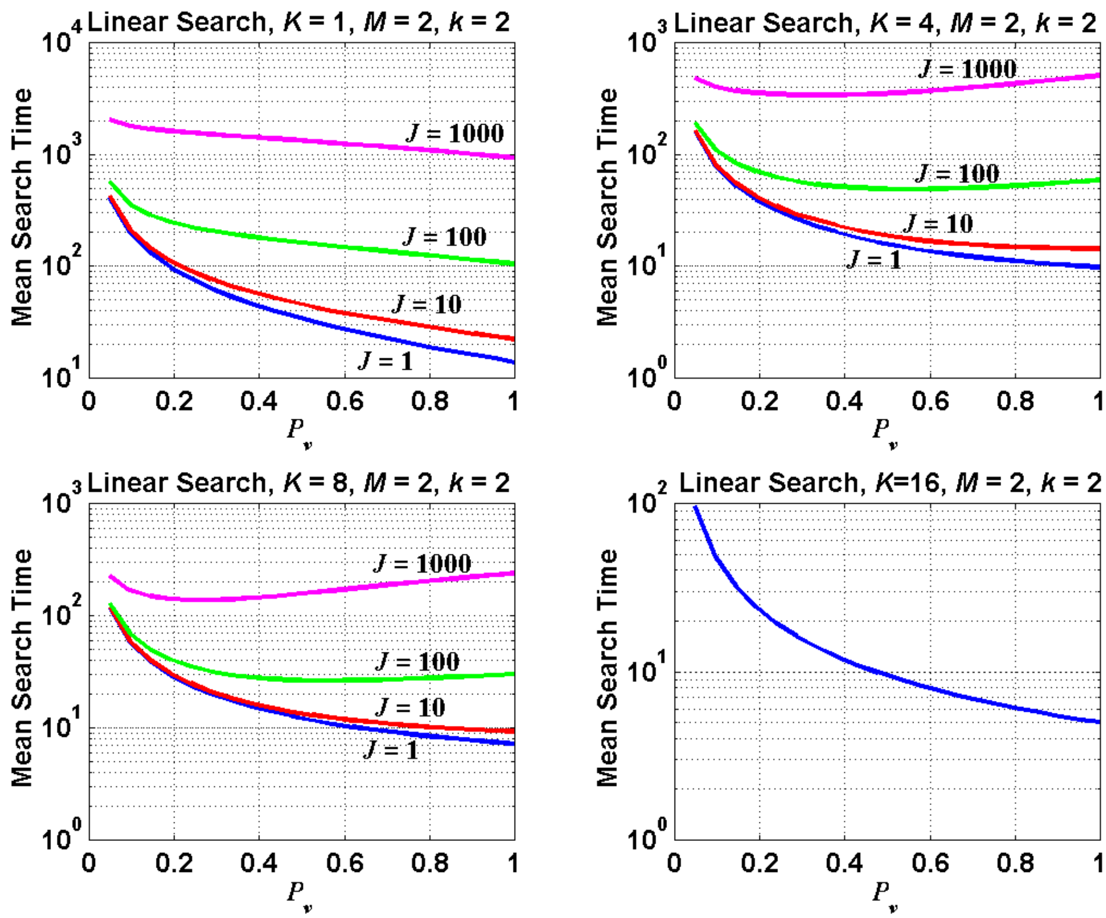

Figure 3.7: Mean search time versus constant verification probability, P_{v}, for a linear search with K consecutive detection bins out of $N_s = 16$ total bins, constant detection and false alarm probabilities $P_D = 0.9$ and $P_{FA} = 0.1$, false alarm penalty time as shown, a verification time of $2k = 4$, and $M = 2$ observers.

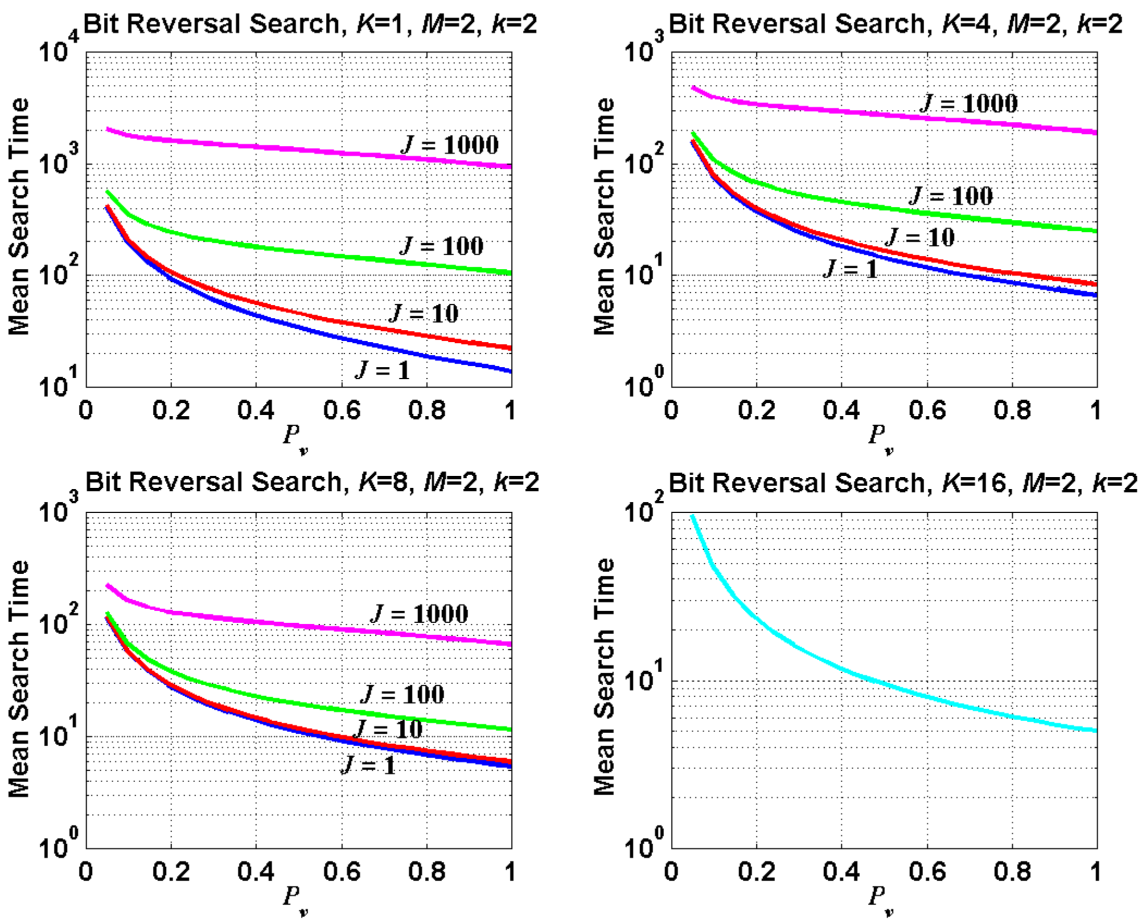

Figure 3.8: Mean search time versus constant verification probability, P_{v}, for a bit reversal search with K consecutive detection bins out of $N_s = 16$ total bins, constant detection and false alarm probabilities $P_D = 0.9$ and $P_{FA} = 0.1$, false alarm penalty time as shown, a verification time of $2k = 4$, and $M = 2$ observers.

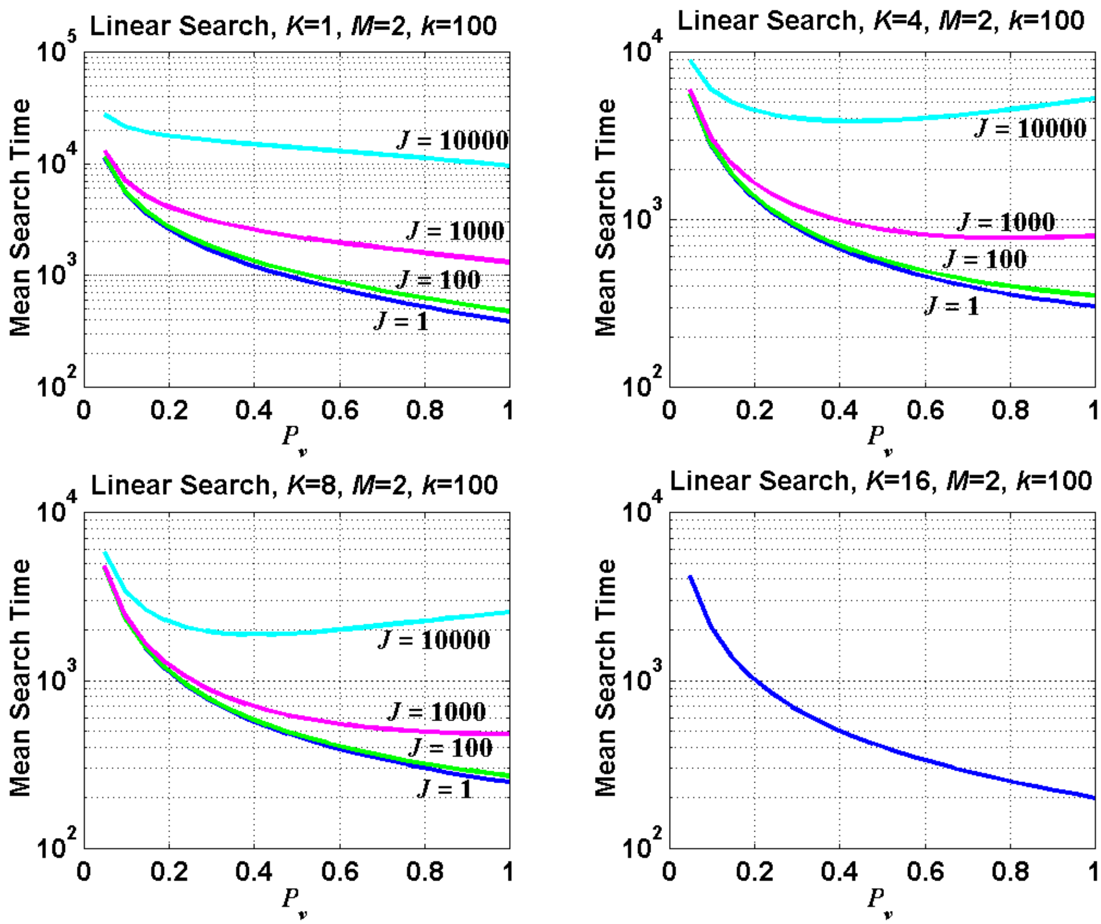

Figure 3.9: Mean search time versus constant verification probability, P_{v}, for a linear search with K consecutive detection bins out of $N_s = 16$ total bins, constant detection and false alarm probabilities $P_D = 0.9$ and $P_{FA} = 0.1$, false alarm penalty time as shown, a verification time of $2k = 200$, and $M = 2$ observers.

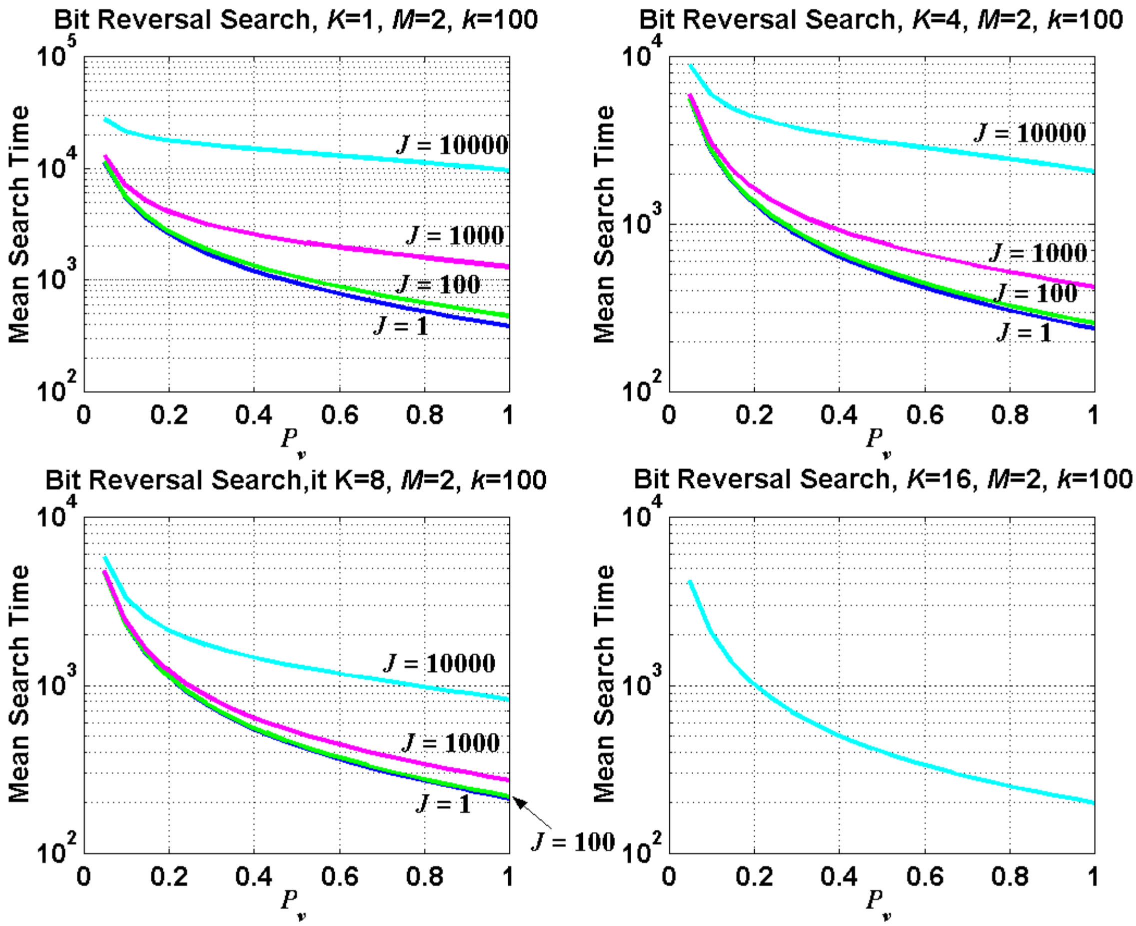

Figure 3.10: Mean search time versus constant verification probability, P_{v}, for a bit reversal search with K consecutive detection bins out of $N_s = 16$ total bins, constant detection and false alarm probabilities $P_D = 0.9$ and $P_{FA} = 0.1$, false alarm penalty time as shown, a verification time of $2k = 200$, and $M = 2$ observers.

The verification probability which minimizes the mean search time is seen from Figures 3.7 through 3.10 to be a function of the search type, the false alarm penalty time, and the verification time. In general this minimizing verification probability is a function of all the system parameters. However, a couple observations are apparent. Firstly, it is seen that for a sufficiently small false alarm penalty time the mean search time is minimized when $P_{\mathrm{v}} = 1$. Secondly, with only a single detection bin, $K = 1$, the mean search is also minimized for $P_{\mathrm{v}} = 1$. This scenario, however, is not of much interest when examining UWB acquisition in multipath. Lastly, it is seen for all cases examined that the bit reversal search produces a mean search time which decreases monotonically with increasing verification probability, independent of the various other parameters. Thus when using the bit reversal search, the verification process employed should attempt to drive P_{v} as close to unity as possible to minimize the average search time. As a general note, all four of the aforementioned graphs were generated for $M = 2$ observers. The same general results mentioned here were also seen for M ranging from 1 to 16.

As just discussed, the verification probability affects the mean search time. However, the accuracy of the overall decision is also affected by the selection of verification criterion and thus an inherent tradeoff exists between decision accuracy and mean search time, as discussed at the beginning of this chapter and illustrated in Figure 3.1. In order to see this tradeoff for the particular example under consideration we remove the false alarm penalty time and assume that no second level of verification exists. The probability of correctly terminating the search can be computed as a function of P_{v} which can then be related back to the mean search time producing the desired search operating characteristic curve. The path gains of Figure 3.6 are redefined as follows, allowing for incorrect decisions:

$$\tilde{F}_{\varepsilon(n)}(z) = \begin{cases} P_{\mathrm{v}} \cdot z & \text{if } n \in \mathcal{I} \\ 0 & \text{if } n \notin \mathcal{I} \end{cases} \tag{3.16}$$

$$\tilde{E}_{\varepsilon(n)}(z) = \begin{cases} 1 - P_{\mathrm{v}} & \text{if } n \notin \mathcal{B} \\ (1 - P_{\mathrm{v}}) \cdot z & \text{if } n \in \mathcal{B} \end{cases} \tag{3.17}$$

The path gains, $\tilde{G}_{\varepsilon(n)}(z)$ and $\tilde{H}_{\varepsilon(n)}(z)$, remain unchanged. Substituting these new path gains into (3.9) and (3.10) for constant detection and false alarm probabilities, P_D and P_{FA}, respectively yields:

$$H_{\varepsilon(n)}(z) = \begin{cases} P_D \cdot P_{\mathrm{v}} \cdot z^{2k+1} & \text{if } n \in \mathcal{I} \\ 0 & \text{if } n \notin \mathcal{I} \end{cases} \tag{3.18}$$

$$G_{\varepsilon(n)}(z) = \begin{cases} P_D \cdot (1 - P_{\mathrm{v}}) \cdot z^{2k} + 1 - P_D & \text{if } n \in \mathcal{I}, n \notin \mathcal{B} \\ P_D \cdot (1 - P_{\mathrm{v}}) \cdot z^{2k+1} + (1 - P_D) \cdot z & \text{if } n \in \mathcal{I}, n \in \mathcal{B} \\ P_{FA} \cdot (1 - P_{\mathrm{v}}) \cdot z^{2k} + 1 - P_{FA} & \text{if } n \notin \mathcal{I}, n \notin \mathcal{B} \\ P_{FA} \cdot (1 - P_{\mathrm{v}}) \cdot z^{2k+1} + (1 - P_{FA}) \cdot z & \text{if } n \notin \mathcal{I}, n \in \mathcal{B} \end{cases} \tag{3.19}$$

The probability of correctly terminating this hybrid search is found as $P_c = P_{ACQ}(1)$ which, because of the uniform nature of the the first detection bin, k_1, is:

$$P_c = \frac{1}{N_s} \sum_{k_1=0}^{N_s-1} P_{ACQ}(1|k_1) \tag{3.20}$$

Here $P_{ACQ}(1|k_1)$ is a function of k_1 through the path gain dependence on the index set, $\mathcal{I}$, as is also the case when computing the mean search time in this fashion. Figure 3.11 shows this probability as a function of the false alarm probability for the various parameters listed. Because of the method in which the hybrid search is performed, i.e., redistributing a single search permutation amongst the various observers, P_c is independent of the number of observers, M. Computing the mean search time for the same parameters as shown in Figure 3.11 allows the search operating characteristic curve of Figure 3.12 to be generated. The mean search time is in units of the number of states visited. Notice that the bit reversal search lies above the linear search for most values of P_{FA}. Also, if the number of observers, M, is considered to be a measure of complexity, Figure 3.12 reveals that as complexity increases the curves are pushed upward and to the left. This phenomenon was discussed at the beginning of this chapter and illustrated in Figure 3.1.

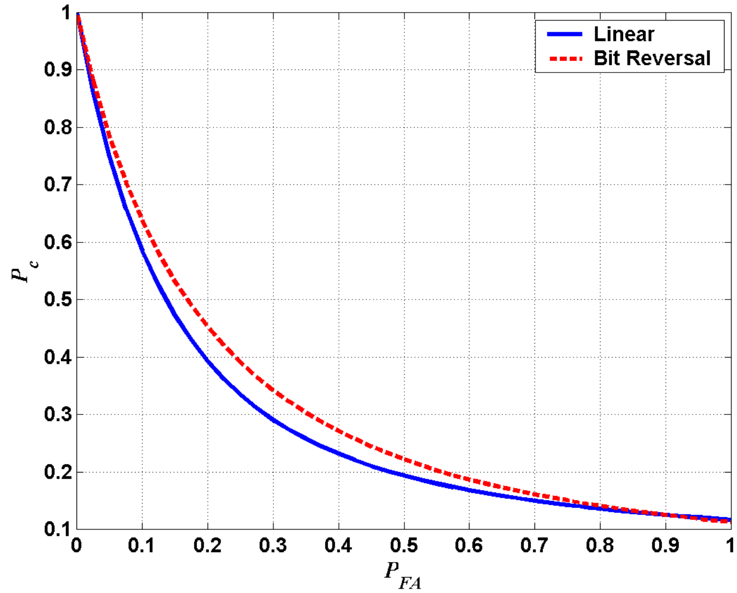

Figure 3.11: Probability of correctly terminating search, P_c, versus false alarm probability, P_{FA}, for $N_s = 16$, $P_D = 0.9$, $K = 2$, $P_v = 0.7$, $2k = 4$, and the search permutation as shown.

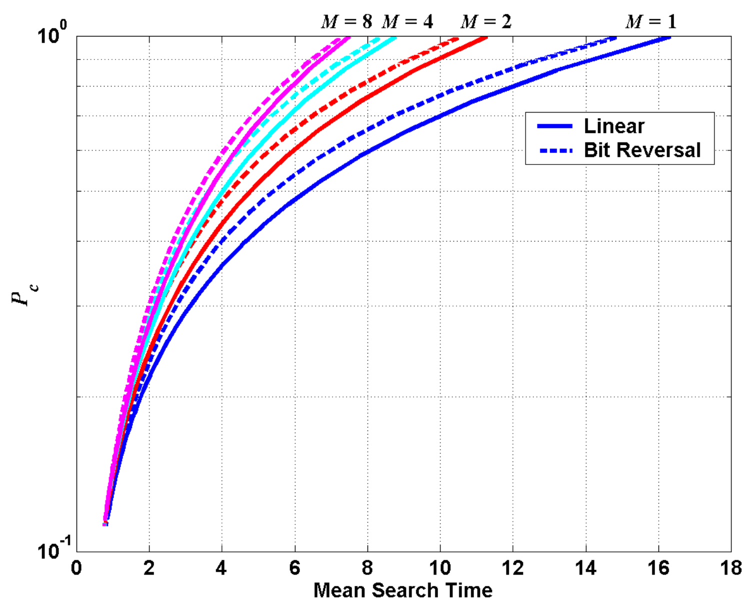

Figure 3.12: Search operating characteristic generated by varying P_{FA} for various values of M where $N_s = 16$, $P_D = 0.9$, $K = 2$, $P_v = 0.7$, $2k = 4$, and the search permutations are as shown.

3.2.2 Random Stopping Time

The end result of the fine acquisition process is a reliable set of estimates for M paths of the multipath channel. The previous section performed an exhaustive search of some fixed offset around the coarse acquisition termination point, $\varepsilon(n)$, by first searching some fixed amount to the right then by the same fixed amount to the left. Selection of this fixed amount assumes a certain amount of knowledge about the delay spread of the channel, not necessarily a good assumption. One slight modification, which alleviates this problem to some extent, would be to allow intermediate checks after each dwell-time, i.e., after each group of M observations has been made. For instance, while searching the fine acquisition uncertainty region if all M observations (assumed to be observations of consecutive bins) are sufficiently small then with high probability the group of observers has slid past the detection set, i.e., the multipath 'cluster'. At this point several options are possible. The option analyzed here is shown in Figure 3.13. As can be seen, at each dwell-time the acquisition state and the intermediate state, N, can be entered and both path gains are functions of the stopping and verification criteria. If the stopping criterion is satisfied but the verification criteria is not, then state N is entered which leads directly to the next state, $\varepsilon(n+1)$, in the coarse search. Changes to the search order of the fine acquisition uncertainty region have not been considered here. The moment generating function for an arbitrary search permutation of the fine acquisition uncertainty region can be found in Appendix D.

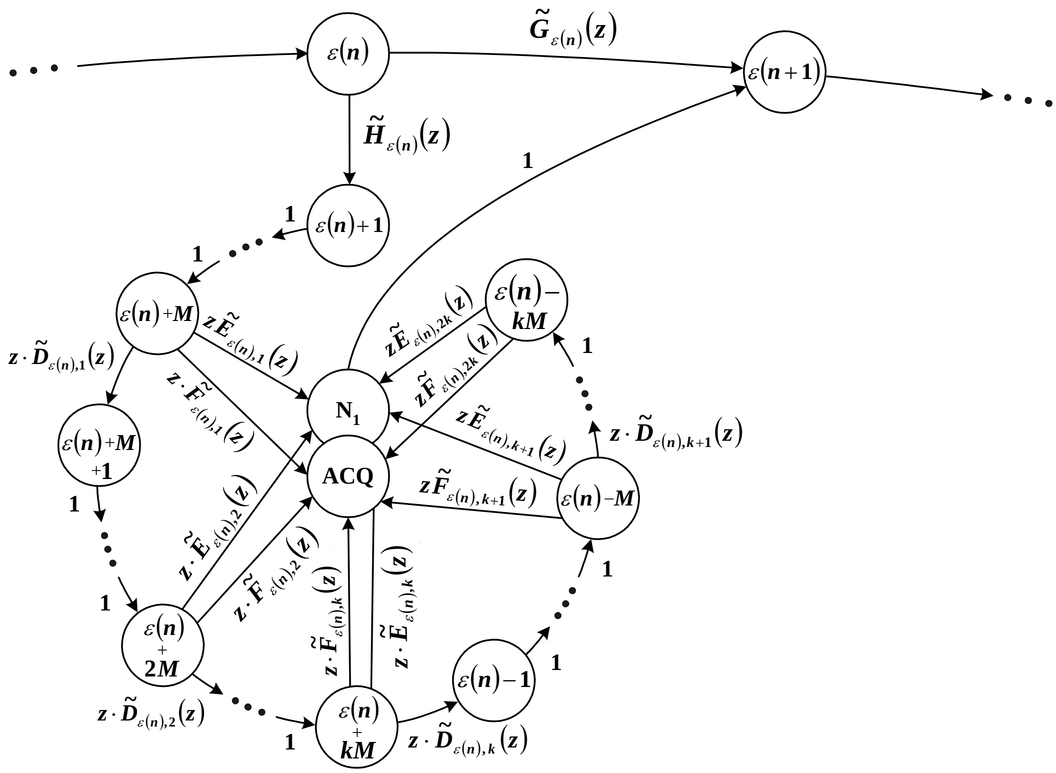

Figure 3.13: Self-similar signal flow graph representation of a fine acquisition search with a random stopping time and M observers

.

Collapsing the graph of Figure 3.13 into the generalized form of Section 2.4 produces the following overall path gains:

$$H_{\varepsilon(n)}(z) = \tilde{H}_{\varepsilon(n)}(z) \cdot \sum_{i=1}^{2k} z^i \cdot \tilde{F}_{\varepsilon(n),i}(z) \prod_{j=1}^{i-1} \tilde{D}_{\varepsilon(n),j}(z) \tag{3.21}$$

$$G_{\varepsilon(n)}(z) = \tilde{G}_{\varepsilon(n)}(z) + \tilde{H}_{\varepsilon(n)}(z) \cdot \sum_{i=1}^{2k} z^i \cdot \tilde{E}_{\varepsilon(n),i}(z) \prod_{j=1}^{i-1} \tilde{D}_{\varepsilon(n),j}(z) \tag{3.22}$$

The product $\prod_{i=0}^{-1}(\cdot)$ is again defined to be unity. Note that if $\tilde{D}_{\varepsilon(n),j}(z) = 1$ for all j, $\tilde{F}_{\varepsilon(n),i}(z) = \tilde{F}_{\varepsilon(n)}(z)$ and $\tilde{E}_{\varepsilon(n),i}(z) = \tilde{E}_{\varepsilon(n)}(z)$ both for $i = 2k$ and zero for $i = 0, 1, \cdots, 2k-1$ then the results of the previous section are obtained.

One alternate method of fine acquisition is now discussed. Previously, as indicated in Figure 3.13, all the bins to the right of $\varepsilon(n)$ were examined followed by all the bins to the left. At any dwell-time the search can exit the verification stage by either acquiring or returning to coarse acquisition. An alternate approach is shown in Figure 3.14. In this approach, the fine acquisition procedure searches to the right of $\varepsilon(n)$ until some stopping criterion is satisfied then searches to the left until some (possibly different) stopping criterion is satisfied. As can be seen in Figure 3.14, the intermediate state, N_1, is entered only by the bins to the right of $\varepsilon(n)$ and this intermediate state leads directly to the first bin to the left of $\varepsilon(n)$. The intermediate state, N_2, leads directly to the next state in the coarse search, $\varepsilon(n+1)$, and is entered only from those states to the left of $\varepsilon(n)$. Possible effects on the search performance due to first searching the bins to the left followed by searching to the right has not been considered here. The generalized path gains for this approach are as follows:

$$
\begin{aligned}
H_{\varepsilon(n)}(z) \quad &= \quad \tilde{H}_{\varepsilon(n)}(z) \cdot \left[\sum_{i=1}^{k} z^i \cdot \tilde{F}_{\varepsilon(n),i}(z) \prod_{j=1}^{i-1} \tilde{D}_{\varepsilon(n),j}(z) \right. \\
+ \quad & \left. \left(\sum_{i=1}^{k} z^i \cdot \tilde{E}_{\varepsilon(n),i}(z) \prod_{j=1}^{i-1} \tilde{D}_{\varepsilon(n),j}(z) \right) \cdot \left(\sum_{i=1}^{k} z^i \cdot \tilde{F}_{\varepsilon(n),i+k}(z) \prod_{j=1}^{i-1} \tilde{D}_{\varepsilon(n),j+k}(z) \right) \right]
\end{aligned} \tag{3.23}
$$

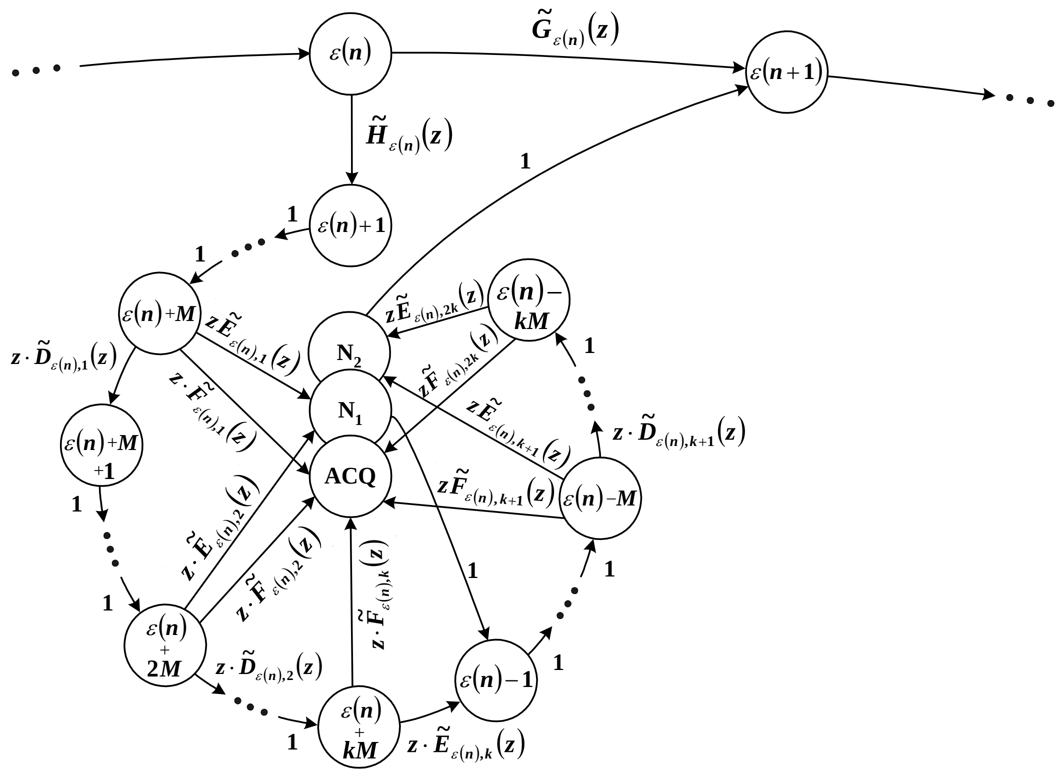

Figure 3.14: Self-similar signal flow graph representation of an alternate fine acquisition search with a random stopping time and M observers

.

For simplicity in representing this path gain $\tilde{D}_{\varepsilon(n),k}(z)$ and $\tilde{E}_{\varepsilon(n),k}(z)$ are defined to be equal.

$$G_{\varepsilon(n)}(z) = \tilde{G}_{\varepsilon(n)}(z) + \tilde{H}_{\varepsilon(n)}(z) \cdot \left(\sum_{i=1}^{k} z^i \cdot \tilde{E}_{\varepsilon(n),i}(z) \prod_{j=1}^{i-1} \tilde{D}_{\varepsilon(n),j}(z) \right) \cdot \left(\sum_{i=1}^{k} z^i \cdot \tilde{E}_{\varepsilon(n),i+k}(z) \prod_{j=1}^{i-1} \tilde{D}_{\varepsilon(n),j+k}(z) \right) \quad (3.24)$$

3.3 Sequential Analysis Using Graphs

Sequential analysis techniques applied to detection and search problems allow for a reduction in the average number of observations required to achieve a fixed detection probability. Such techniques have already been used for UWB acquisition [66]. Here the self-similar signal flow graph introduced in the previous section is used to outline the analysis of a single correlator search scheme involving sequential detection. The specific graph used for this analysis is shown in Figure 3.15 where an upper limit, k_d, on the amount of time spent dwelling at any state is assumed. This limit will be removed shortly by allowing k_d to increase toward $+\infty$.

Here the dwell-time between states is reduced from a complete code period, as in the previous section, to that of a single frame time, or single observation time. The generalized path gains can be computed as follows:

$$H_{\varepsilon(n)}(z) = \tilde{H}_{\varepsilon(n)}(z) \cdot \sum_{i=1}^{k_d} z^i \cdot \tilde{F}_{\varepsilon(n),i}(z) \prod_{j=1}^{i-1} \tilde{D}_{\varepsilon(n),j}(z) \tag{3.25}$$

$$G_{\varepsilon(n)}(z) = \tilde{H}_{\varepsilon(n)}(z) \cdot \sum_{i=1}^{k_d} z^i \cdot \tilde{E}_{\varepsilon(n),i}(z) \prod_{j=1}^{i-1} \tilde{D}_{\varepsilon(n),j}(z) \tag{3.26}$$

The path gains, $\tilde{E}_{\varepsilon(n),i}(z)$ and $\tilde{F}_{\varepsilon(n),i}(z)$, are functions of the i independent observations of state $\varepsilon(n)$. Specifically, assume that f_i is a sequence of real numbers generated from the set of i observations of the state $\varepsilon(n)$. One example is the sample mean of all i observations. The path gains $\tilde{E}_{\varepsilon(n),i}(z)$, $\tilde{F}_{\varepsilon(n),i}(z)$, and $\tilde{D}_{\varepsilon(n),i}(z)$ can now be defined

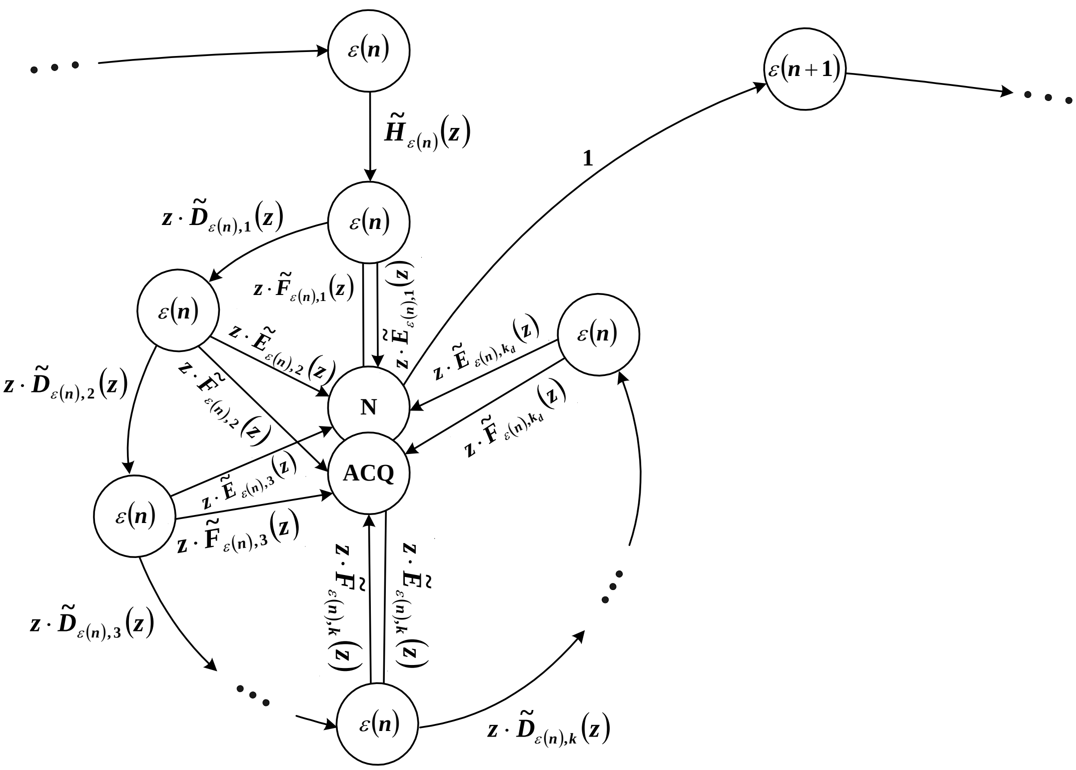

Figure 3.15: Self-similar signal flow graph representation of a search with sequential detection and $M = 1$ observer

in terms of two arbitrary thresholds, $T_0(i)$ and $T_1(i)$, which are shown as functions of i to be as general as possible.

$$\tilde{E}_{\varepsilon(n),i}(z) = \Pr(f_i \le T_0(i)) \tag{3.27}$$

$$\tilde{F}_{\varepsilon(n),i}(z) = \Pr(f_i \ge T_1(i)) \tag{3.28}$$

$$\tilde{D}_{\varepsilon(n),i}(z) = \Pr(T_0(i) < f_i < T_1(i)) \tag{3.29}$$

The resulting moment generating function, $P_{ACQ}(z)$ now yields a complete statistical description of the sequential detection problem. Recall that the amount of time spent dwelling on any particular state was upper bounded at k_d dwell-times. Taking the limit with respect to k_d yields the moment generating function, $P_{SEQ}(z)$, for the unbounded sequential detection problem:

$$P_{SEQ}(z) = \lim_{k_d \to \infty} P_{ACQ}(z) \tag{3.30}$$

The mean acquisition time and the probability of correct termination can then be determined using this moment generating function as discussed in Section 2.4. Although it is not discussed here, the hybrid case of $M > 1$ observers is also possible in the current framework.

Chapter 4

Coarse Acquisition of UWB Systems

In order for any type of data detector to properly function the frame time and the code hopping sequence, as discussed in Chapter 1, must first be synchronized. For the multipath data detector, such as the RAKE receiver of Section 1.6, some information about the multipath channel must also be known. In spread spectrum systems, synchronization typically occurs prior to channel estimation because of low SNR environments. This scenario will be assumed here. In fact, the final stage of acquisition, termed *fine acquisition* as discussed in the next chapter, produces channel estimates. These channel estimates are simply estimates of the strongest paths within the multipath cluster as discussed in Section 1.3. The current chapter deals with *coarse acquisition* which is simply the process of locating the multipath cluster within the frame time. Section 4.1 deals with acquisition when no time-hopping code is present. Since no time-hopping code is present only acquisition of the frame time needs to occur before data detection. Section 4.2 assumes a time-hopping code is present and discusses the process of joint frame and code acquisition. These scenarios are first examined for the single-user case with consideration given in Section 4.4 to acquisition in the presence of multiple users.

4.1 Single User Frame Acquisition of Uncoded UWB Systems

In this section, as in the next, it is assumed that the multipath channel does not vary significantly with time. For a UWB system with only a single user a time-hopping code would not be required thus eliminating the $c_n T_c$ term in (1.4). Also, one possibility is that the UWB system first acquires without any coding and then after acquisition begins communicating with a specific time-hopping code. At any rate, the current scenario helps to understand the situation discussed in the next section. Also removed is the data term, as it will be assumed that the acquisition occurs without any data modulation. Thus the UWB received signal becomes

$$s(t) = \sqrt{E_p} \cdot \sum_{n=0}^{\infty} \sum_{k=0}^{L_p-1} a_k \cdot p(t - nT_f - \tau_k) \tag{4.1}$$

For a fixed multipath channel, i.e., one that doesn't vary with time, the only random quantity will be the first path arrival time, τ_0. The a_k terms and the inter-arrival times, $\Delta\tau_k = \tau_k - \tau_{k-1}$, are assumed to be deterministic quantities. Here τ_0 will be assumed uniform from 0 to T_f. It should be noted that the received signal is truncated to the first L_p paths, where for the examples below L_p will be set to 300. The multipath channel to be considered will be determined from the sample data of Section 1.4, i.e., see Figure 1.7. Figure 1.9 reveals that for $L_p = 300$ roughly 85% of the energy in the measured waveform has been taken into account.

Shown in Figure 4.1 is a single correlator receiver. The value of Δ in the integrator limits is set by the effective time width of the pulse shape. The template waveform is

$$v(t) = \sum_{m=0}^{\infty} p\left(t - mT_f - \varepsilon_{(m \bmod N)}\right) \tag{4.2}$$

The limits of the integrator are only over the j^{th} frame's search location. This search location is determined by a search algorithm ε_n for $n = 0, 1, \cdots, N-1$, as discussed in Chapter 2 [1]. Thus, $\varepsilon_{(m \bmod N)}$ changes every frame time and represents the 'bin' center where the frame time is divided into N equally spaced bins. A simple acquisition algorithm for multiple correlators as in a selective RAKE receiver would be to divide the search among the correlators as discussed in Section 2.5. For now only the single correlator receiver structure will be considered.

The signal portion of the correlator input, as derived from (1.4) and (1.10), is

$$s(t) = \sqrt{E_p} \sum_{n=0}^{\infty} \sum_{k=0}^{L_p - 1} a_k p(t - nT_f - \tau_k) \tag{4.3}$$

[1] The truly random search would only need the indexing ε_m for $m = 0, 1, 2, \cdots$ since it would be random and independent of the previous frames.

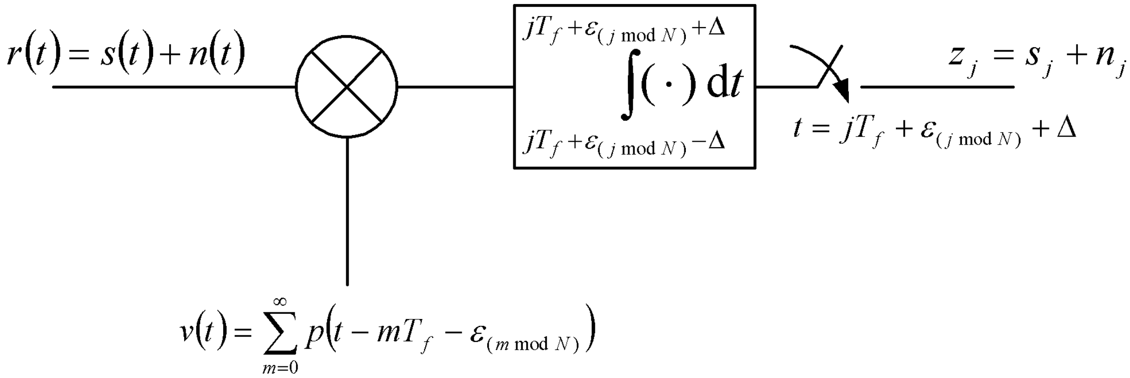

Figure 4.1: Single Correlator Receiver

This is an infinite energy, periodic signal with power E_p/T_f. The correlator output, z_j, has a signal and noise component, s_j and n_j respectively. The simplest, single-dwell, acquisition scheme terminates the first time the correlator magnitude exceeds some prescribed threshold, say $\sqrt{E_p}\Upsilon$ where Υ is some normalized threshold. This acquisition scheme was examined in Section 2.3 and is represented graphically in the Markov chain of Figure 2.5. For now only this Markov chain approach is used to analyze the acquisition time. The next section extends this analysis using the signal flow graph approach as discussed in Sections 2.4 and 2.5. The transition probabilities are seen to be

$$p_n = p_{(j \bmod N)} = \Pr(|z_j| \leq \sqrt{E_p}\Upsilon) \quad \text{for} \;\; j = 0, 1, 2, \cdots \tag{4.4}$$

As will be seen shortly, z_j is a function of the multipath parameters a_k and $\Delta\tau_k$, which are assumed deterministic, and τ_0 which is uniform over a frame time. Thus, the mean time to acquisition to be computed here is inherently conditioned on these multipath parameters. The mean time to acquisition is properly denoted as $E(T_{ACQ}|\mathbf{a}, \boldsymbol{\tau})$. If a statistical model is introduced for the multipath channel, the results obtained here are still useful. Namely, the unconditional mean acquisition time can be computed as

$$E(T_{ACQ}) = \int_{\mathbf{a}} \int_{\boldsymbol{\tau}} E(T_{ACQ}|\mathbf{a}, \boldsymbol{\tau}) f(\mathbf{a}, \boldsymbol{\tau}) \, \mathrm{d}\mathbf{a} \, \mathrm{d}\boldsymbol{\tau} \tag{4.5}$$

Obviously this assumes that the joint probability density of the amplitude coefficients and path delays, $f(\mathbf{a}, \boldsymbol{\tau})$, can be found.

The correlator output, z_j, is easily seen to be a Gaussian random variable with mean s_j and variance $\mathrm{var}(n_j)$. The noise component, n_j, is that portion of the output due solely to the AWGN, $n(t)$, at the input. The mean of this noise process is zero and the autocorrelation is simply $N_0\delta(t_1 - t_2)$. In all the following expressions, the substituted variable $\theta_j = jT_f + \varepsilon_{(j \bmod N)}$ is used. The noise component is now computed quite easily as

$$n_j = \int_{\theta_j - \Delta}^{\theta_j + \Delta} n(t)v(t)\mathrm{d}t = \sum_{m=0}^{\infty} \int_{\theta_j - \Delta}^{\theta_j + \Delta} n(t)p(t - mT_f - \varepsilon_{(m \bmod N)}) \, \mathrm{d}t \tag{4.6}$$

Here we are only integrating over one portion of a frame and since the frames do not overlap we see that the output noise sequence is independent and only one term in the above sum survives, i.e., the $m = j$ term.

$$n_j = \int_{\theta_j - \Delta}^{\theta_j + \Delta} n(t)p(t - jT_f - \varepsilon_{(j \bmod N)}) \, \mathrm{d}t \tag{4.7}$$

Using the substitution $u = t - jT_f - \varepsilon_{(j \bmod N)}$ and noting that the noise process is wide-sense stationary so that $n(t)$ and $n(t - jT_f - \varepsilon_{(j \bmod N)})$ are equivalent in the wide-sense [2] reveals the following:

$$n_j = \int_{-\Delta}^{\Delta} n(t)p(t)\,\mathrm{d}t \tag{4.8}$$

The mean and variance of n_j are found to be 0 and N_0, respectively. The output of the multiplier in Figure 4.1 when signal is present, $s(t) \cdot v(t)$, is seen to be

$$\sqrt{E_p}\left[\sum_{m=0}^{\infty} p(t - mT_f - \varepsilon_{(m \bmod N)})\right] \cdot \left[\sum_{n=0}^{\infty}\sum_{k=0}^{L_p-1} a_k p(t - nT_f - \tau_k)\right]$$

$$= \sqrt{E_p}\sum_{m}\sum_{n}\sum_{k} a_k p(t - mT_f - \varepsilon_{(m \bmod N)}) \cdot p(t - nT_f - \tau_k) \tag{4.9}$$

Several simplifying assumptions are now made. First, the frame time will be assumed long enough with respect to the 'delay spread' of the multipath channel so that only energy from the $(m-1)^{\mathrm{st}}$ frame will potentially spill into the m^{th} frame [3]. It is also assumed that $\varepsilon_{(m \bmod N)}$ only varies over the m^{th} frame. These assumptions, along with neglecting the edge effects and potential overlaps of search locations within a

[2] This means that the first two moments are exactly equal.

[3] For shorter frame times the effect of inter-frame interference needs to be considered.

frame and between frames, reveal that the above summation in n is limited to m and $m-1$. Thus we see that $s(t)v(t)$ is

$$\sqrt{E_p}\sum_m\sum_k a_k p(t - mT_f - \varepsilon_{(m \bmod N)}) \cdot [p(t - mT_f - \tau_k) \\ + p(t - (m-1)T_f - \tau_k)] \tag{4.10}$$

The integrator output is now seen to be

$$\sqrt{E_p}\sum_m\sum_k a_k \int_{\theta_j-\Delta}^{\theta_j+\Delta} p(t - mT_f - \varepsilon_{(m \bmod N)}) \cdot [p(t - mT_f - \tau_k) \\ + p(t - (m-1)T_f - \tau_k)]\,\mathrm{d}t \tag{4.11}$$

Here we are only integrating over one 'bin' of the j^{th} frame and since we have already accounted for any spillover of previous frames we see that m can only equal j in the above summation. Thus we have at the output of the integrator

$$\sqrt{E_p}\sum_k a_k \int_{\theta_j-\Delta}^{\theta_j+\Delta} p(t - jT_f - \varepsilon_{(j \bmod N)}) \cdot [p(t - jT_f - \tau_k) \\ + p(t - (j-1)T_f - \tau_k)]\,\mathrm{d}t \tag{4.12}$$

Again the substitution $u = t - jT_f - \varepsilon_{(j \bmod N)}$ is used and the correlator output is seen to be

$$s_j = \sqrt{E_p}\sum_k \int_{-\Delta}^{\Delta} p(t) \cdot \big[p(t + \varepsilon_{(j \bmod N)} - \tau_k) \\ + \ p(t + \varepsilon_{(j \bmod N)} - \tau_k + T_f)\big]\,\mathrm{d}t \tag{4.13}$$

Recalling the definition of $R_{pp}(\tau)$ as in (1.20) and the fact that Δ was chosen sufficiently large to yield a closed form expression, $\gamma(\tau)$, for $R_{pp}(\tau)$ as given in (1.33) yields the following expression for the signal portion of the correlator output:

$$s_j = \sqrt{E_p} \sum_k a_k \cdot \left[\gamma(\tau_k - \varepsilon_{(j \bmod N)}) + \gamma(\tau_k - \varepsilon_{(j \bmod N)} - T_f)\right] \tag{4.14}$$

The possibility does exist for both $\gamma(\cdot)$ terms in the above sum to be near zero. This indicates that that the search location ε_j is not near the multipath 'cluster'. Conversely, it is also possible for multiple terms to be significantly different from zero. However, since the frame time is assumed to be large, the situation cannot arise where both $\gamma(\tau_k - \varepsilon_{(j \bmod N)})$ and $\gamma(\tau_k - \varepsilon_{(j \bmod N)} - T_f)$ are non-zero. Recall from Figure 1.1 that the data modulation format may be designed so as to prevent the multipath 'cluster' from spilling into the next frame. However, prior to frame synchronization it is possible for this spillover of energy to occur. This is the reason that the two $\gamma(\cdot)$ terms are required in the above expression for the correlator signal component output.

As mentioned earlier, the mean of the correlator output is simply the signal component, $E(z_j) = s_j$. If the multipath parameters assume a statistical model then the amplitude coefficients and path delays will be random variables. The mean of the correlator output will then be random implying that the the correlator output has a *mixture distribution* arising from a hierarchical model [7]. This is exactly the reason that (4.5) is valid, i.e., for any two random variables X and Y the mean of X can be found as $E(X) = E(E(X|Y))$.

We have shown that the correlator output is Gaussian with mean s_j and variance N_0. The transition probability, however, requires the probability of the event $\{|z_j| \leq \sqrt{E_p}\Upsilon\}$. This probability can be computed in a straightforward manner without knowledge of the density function of $R = |z_j|$. For completeness, however, this density function is provided here. It can be shown (see Appendix 5A of [49], for example) that the correlator magnitude has the following density function:

$$f_R(r) = \sqrt{\frac{2}{\pi\sigma^2}} \cdot \exp\left(-\frac{r^2 + s^2}{2\sigma^2}\right) \cdot \cosh\left(\frac{s}{\sigma^2} \cdot r\right) \tag{4.15}$$

Here $s = s_j$ and $\sigma^2 = N_0$. The desired transition probability can be found by integrating the above density function, although a simpler method exists. Namely, one can directly compute this probability as

$$\begin{aligned} \Pr(|z_j| \leq \sqrt{E_p}\Upsilon) &= \Pr(-\sqrt{E_p}\Upsilon \leq z_j \leq \sqrt{E_p}\Upsilon) \\ &= F_{z_j}(\sqrt{E_p}\Upsilon) - F_{z_j}(-\sqrt{E_p}\Upsilon) \end{aligned} \tag{4.16}$$

The distribution function, $F_{z_j}(z)$, can be stated in terms of the Gaussian Integral Function, or Q function as it is also called. Namely, a mean μ and variance σ^2 Gaussian random variable, X, has the distribution function $F_X(x) = 1 - Q\left(\frac{x-\mu}{\sigma}\right)$ where the Q function is simply

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty \exp\left(-\frac{u^2}{2}\right) \,\mathrm{d}u \tag{4.17}$$

As mentioned earlier, the absolute values of the path arrival times are not known and thus only the *relative* path arrival times will be available. Rewriting the path arrival times τ_k in terms of these relative path arrival times also allows for an explicit relationship between the transition probabilities and the first path arrival τ_0. This is accomplished by noting that $\tau_k = \tau_0 + \Delta\tau_k$, where $\Delta\tau_k = \tau_k - \tau_0$. The relative path arrival times, $\Delta\tau_k$, are determined from the channel sounding procedure (here from the CLEAN algorithm on the measured results of a channel sounding experiment). Thus the transition probability can be found as follows, where $n = j \bmod N$:

$$
\begin{aligned}
p_n(\tau_0) \quad &= \quad \Pr(|z_j| \leq \sqrt{E_p}\Upsilon) \;= \qquad\qquad (4.18) \\
&1 - Q\left(\sqrt{\frac{E_p}{N_0}}\left(\Upsilon + \sum_k a_k \cdot [\gamma(\tau_0 + \Delta\tau_k - \varepsilon_n) + \gamma(\tau_0 + \Delta\tau_k - \varepsilon_n - T_f)]\right)\right) \\
&- Q\left(\sqrt{\frac{E_p}{N_0}}\left(\Upsilon - \sum_k a_k \cdot [\gamma(\tau_0 + \Delta\tau_k - \varepsilon_n) + \gamma(\tau_0 + \Delta\tau_k - \varepsilon_n - T_f)]\right)\right)
\end{aligned}
$$

These transition probabilities can then be used to compute the performance for a number of different acquisition algorithms as discussed in Chapter 2. For now the simplest acquisition scenario will be investigated, namely the single-dwell case as represented in Figure 2.5. Thus the mean acquisition time can be computed using (2.32). Since the first path arrival time, τ_0 is assumed uniform over the frame time, (4.5) must be used. The resultant mean acquisition time (in number of states visited) is seen to be

$$E(T_{ACQ}) = \frac{1}{T_f} \int_0^{T_f} \frac{1 + \sum_{m=0}^{N-2} \prod_{n=0}^{m} p_n(\tau_0)}{1 - \prod_{n=0}^{N-1} p_n(\tau_0)} \, \mathrm{d}\tau_0 \tag{4.19}$$

The transition probabilities for the above expression are given in (4.18). Plots of this result are shown below in Figure 4.4. The multipath channel used in computing the transition probabilities is determined from the CLEAN algorithm of Section 1.4 with the data as per Figure 1.7 limited to $L_p = 300$ paths. Two of the searches from Section 2.2 are investigated, the linear search and the bit reversal search. For simplicity in dealing with this bit reversal search, the total number of bins is set to $N = 2^{13} = 8192$. It should be noted that the probability of correct termination, P_c, may be rather poor for these scenarios due to the lack of any verification phase. The purpose of the figures is simply to demonstrate the inadequacies of the linear search versus the bit reversal search for the UWB acquisition problem. The probability P_c, which is the same for either of the two searches examined below, can be improved at the cost of increased acquisition time as discussed earlier. Due to a lack of verification, a large false alarm rate will give meaningless results for the mean acquisition time. To compensate for the current single-dwell example, the input E_p/N_0 is set to a large value, namely 50 dB. The normalized threshold, Υ, sets the detection and false alarm probabilities, as well as the overall probability of correct termination and the mean acquisition time. For the figures shown below, a normalized threshold of $\Upsilon = 0.05$ is assumed. A proper threshold can be determined by examining the mean correlator output, s_j, which is shown in Figure 4.2. Since the input SNR is large, this threshold can be lower and still produce a small false alarm rate. For illustrative purposes, the first path arrival time is set to $\tau_0 = 100$ nsec in Figure 4.2.

For the mean correlator output shown in Figure 4.2, the transition probabilities of (4.18) can be computed. Again for illustrative purposes, τ_0 is set to 100 nsec. The corresponding transition probabilities for the normalized threshold of $\Upsilon = 0.05$ are shown below in Figure 4.3. Notice the that since the input SNR is large, the transition probabilities appear to be mostly limited to either 0 or 1. The curve in Figure 4.3 can be used to illustrate the behavior of (2.32), i.e., the integrand of (4.19). Recall that the mean time from these expressions is given in terms of the *number* of bins searched, not absolute time itself. Thus for $N = 8192$, the bin spacing is $T_f/8192$ or 0.1221 nsec.

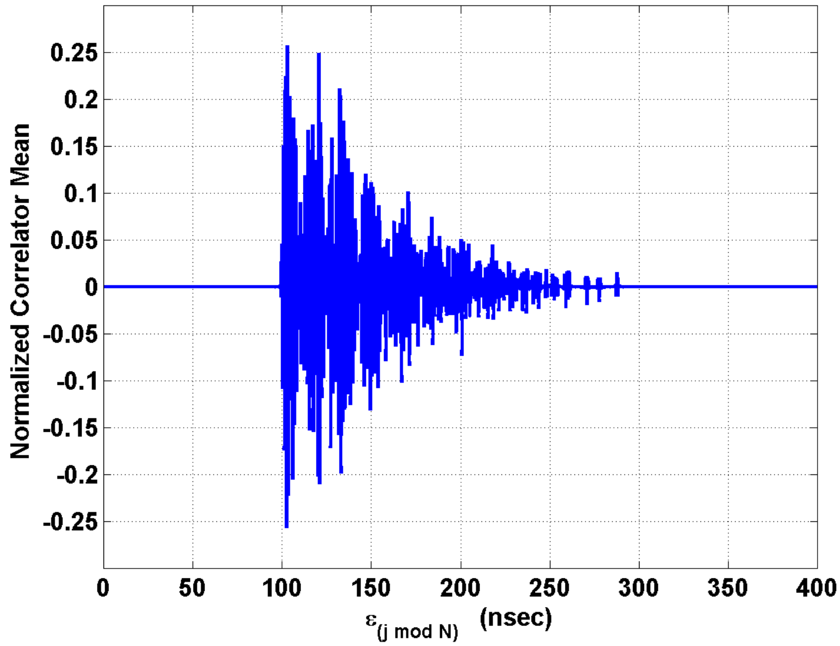

Figure 4.2: Normalized mean correlator output, $s_j/\sqrt{E_p}$, for the reconstructed signal of Figure 1.7, $\tau_0 = 100$ nsec, $T_f = 1000$ nsec, $N = 8192$ bins, and $\varepsilon_n = \frac{n}{N} \cdot T_f$ for $n = 0, 1, \cdots, N-1$.

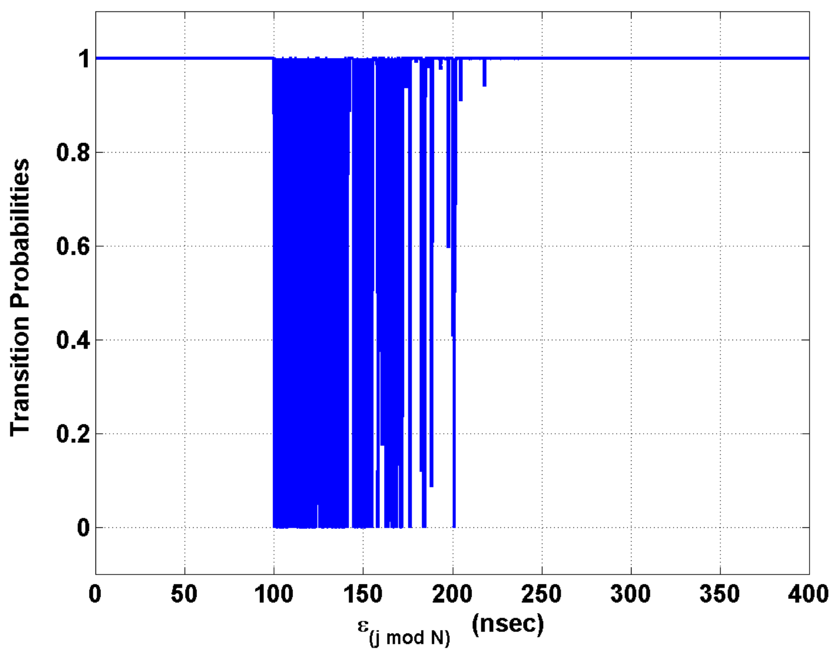

Figure 4.3: Transition Probabilities, p_n, for the reconstructed signal of Figure 1.7, $E_p/N_0 = 50$ dB, $\Upsilon = 0.05$, $\tau_0 = 100$ nsec, $T_f = 1000$ nsec, $N = 8192$ bins, and $\varepsilon_n = \frac{n}{N} \cdot T_f$ for $n = 0, 1, \cdots, N-1$.

Figure 4.3 reveals that since τ_0 is set to 100 nsec and the SNR is large, the transition probability is almost identically one for all bins up to approximately 100 nsec. It would be expected, then, that the mean acquisition time for a linear search conditional on $\tau_0 = 100$ nsec should be somewhere around 100 nsec. This is indeed the case, and can be easily explained using the integrand of (4.19). The denominator, $1 - \prod_{n=0}^{N-1} p_n$, is seen to be one since the product of the transition probabilities, p_n, will always be zero, as can be seen in Figure 4.3. The numerator will be the sum of the run of 1's in Figure 4.3 leading up to the first encountered zero. Thus the mean acquisition time is derived from the the number of bins searched up to approximately 100 nsec, as expected. After this value in Figure 4.3, the sum contributes no more terms because the product $\prod_{n=0}^{m} p_n$ will be zero for all remaining m.

The unconditional mean acquisition times as given in (4.19) are simply the means of the curves in Figure 4.4. The mean acquisition time of the linear search (in number of states visited) is found to be $E(T) = 3246.6$, while the mean acquisition time for the bit reversal search is much smaller at $E(T) = 28.7$. *Thus the bit reversal search is roughly 113 times faster that the linear search for the multipath channel and system parameters currently considered.* Recall that a new state is searched every frame time, so that multiplying these mean acquisition times by T_f yields the mean time in units of seconds. If one examines Figure 4.3, it will be noted that although the terminating hypotheses are not consecutive they can be viewed as approximately consecutive over about 100 nsec. This implies that the K/N parameter of Section 2.2 is more or less $100/1000 = 0.1$. Also the normalized mean acquisition times are seen

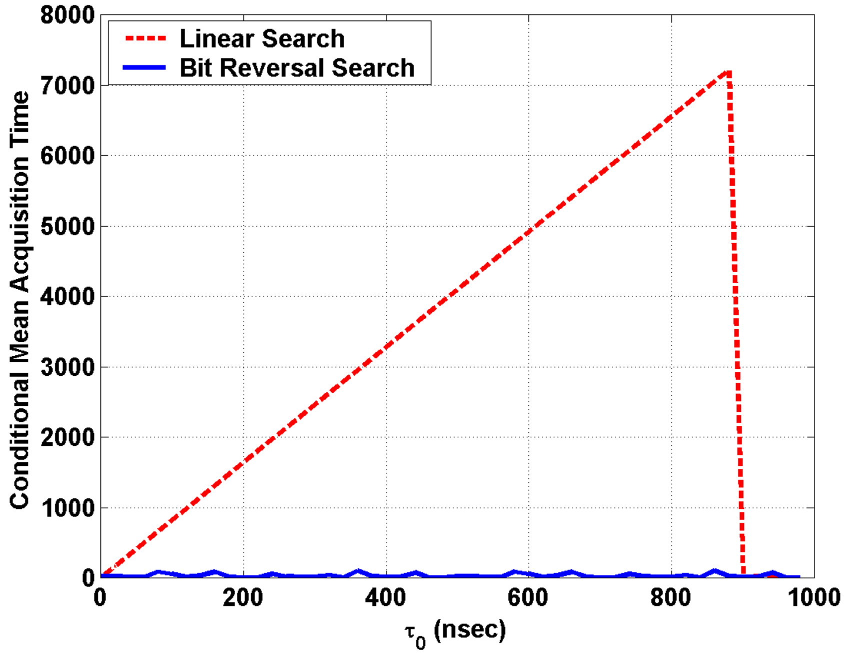

Figure 4.4: Conditional mean acquisition time, $E(T|\tau_0)$ in number of states visited, for the reconstructed signal of Figure 1.7, $E_p/N_0 = 50$ dB, $\Upsilon = 0.05$, $T_f = 1000$ nsec, $N = 8192$ bins.

to be $E(T)/N = 3246.6/8192 = 0.396$ and $E(T)/N = 28.7/8192 = 0.0035$ for the linear and bit reversal searches, respectively. A plot of the results of Section 2.2 is shown in Figure 4.5 for $N = 8192$. The curves are for ideal detection and false alarm probabilities, as well as the assumption of consecutive terminating hypotheses as explained in Section 2.2. These curves are labeled as 'Ideal' searches. Also shown are the normalized mean acquisition times for the results computed in this section, labeled as the 'with multipath' searches. Namely, the points $(K/N, E(T)/N) = (0.1, 0.396)$ and $(0.1, 0.0035)$ are plotted for the linear and bit reversal searches, respectively. Even though the true multipath does not lead to consecutive terminating bins, the results

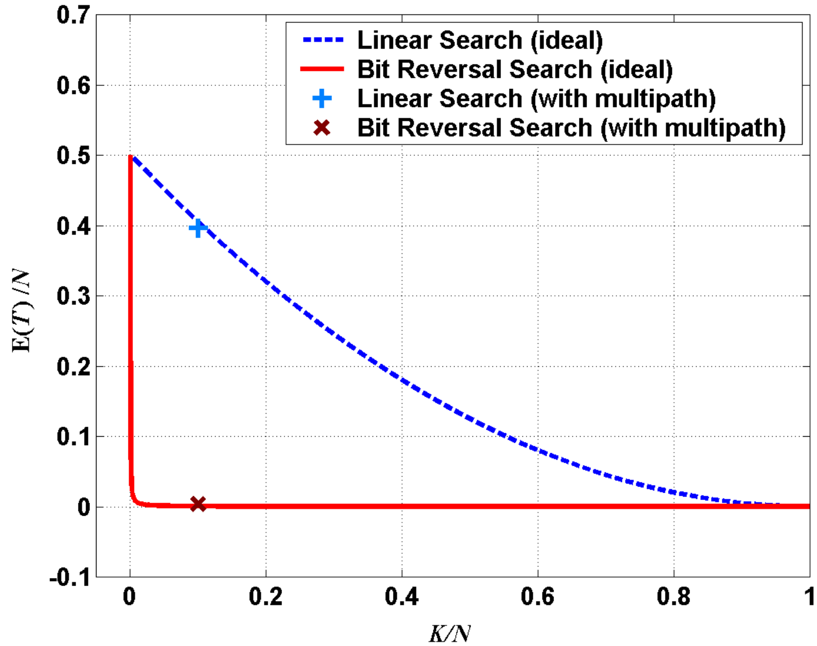

Figure 4.5: Comparison of Normalized Mean Acquisition Time for the Multipath Channel versus 'Idealized' Results of Section 2.2 ($N = 8192$).

from such an assumption are seen to approximate quite well the multipath scenario examined in this section.

4.2 Single User Acquisition of Coded UWB Systems

Here it will be assumed, as in the previous section, that the received pulse shape is that of Section 1.1 and the multipath channel is as described in Section 1.3. Also, a time-hopping code will be present, as discussed in Section 1.2, so that at the channel output the received signal, without any data modulation, is

$$r(t) = \sqrt{E_p} \sum_n \sum_{l=0}^{L_p-1} a_l \cdot p(t - nT_f - c_n T_c - \tau_l) + n(t) \tag{4.20}$$

The mean zero Guassian random process, $n(t)$, is additive noise with autocorrelation function $N_0\delta(t_1 - t_2)$. The time hopping code assumed here, c_n, is a length N_c sequence of nonnegative integers and T_c is the code chip time. Appendix B discusses one method for generating these time hopping codes. The frame time, T_f, is assumed to be an integer multiple of the code chip time so that $T_f = N_f T_c$. The receiver and transmitter frame times are assumed equal while the transmitter/receiver separation is not known. This gives rise to a uniformly-random direct-path arrival-time over the period of the received signal, thus implying that the direct path delay, τ_0, is uniform on $[0, N_c T_f)$.

At the receiver, a group of M correlators is present with received signal in (4.20) acting as the input to each of these correlators. It is assumed that there are at least as many arrival paths as there are correlators, i.e., $M \leq L_p$. The received signal is multiplied by an individual correlator template waveform, which for the m^{th} correlator is:

$$l^{(m)}(t) = \sum_j \sum_{k=0}^{N_c-1} p(t - \alpha_{j,k}^{(m)} - jN_c T_f) \tag{4.21}$$

The time offset for the m^{th} correlator template can can vary over a code period and is set by the term $\alpha_{j,k}^{(m)}$. This term is defined as

$$\alpha_{j,k}^{(m)} = \Big((kN_f + c_k + k_{\beta,j}^{(m)}) \bmod N_f N_c\Big)T_c + \beta_{r,j}^{(m)} \tag{4.22}$$

Proper code phase is accounted for with this term, as well as the proper timing offset within each frame for an arbitrary time shift of $\beta_j^{(m)} = k_{\beta,j}^{(m)} T_c + \beta_{r,j}^{(m)}$ which varies over $[0, N_c T_f)$. The integer term $k_{\beta,j}^{(m)}$ is a nonnegative integer and the remainder term $\beta_{r,j}^{(m)}$ varies over $[0, T_c)$. The frame time is divided into N bins so that $\beta_j^{(m)}$ can be selected from a set of $N \cdot N_c$ time offsets.

As discussed earlier, the codes considered here are short codes so that the correlator dwell time is one code period in length. Straightforward analysis reveals the output of the m^{th} correlator: $z_j^{(m)} = s_j^{(m)} + n_j^{(m)}$ where the correlator noise sequence, $n_j^{(m)}$, is an IID sequence of mean zero, variance $N_c N_0$ Gaussian random variables. The correlator output mean is

$$\begin{aligned} s_j^{(m)} = \sqrt{E_p} \sum_{n=(j-1)N_c}^{(j+1)N_c-1} \sum_{k=0}^{N_c-1} \sum_{l=0}^{L_p-1} a_l \cdot \gamma(\tau_l - \alpha_{j,k}^{(m)} \\ +c_{(n \bmod N_c)} T_c - (jN_c - n)T_f) \end{aligned} \tag{4.23}$$

where the pulse autocorrelation function is given in (1.33). Now an example of the correlation mean, normalized by $\sqrt{E_p}$, is shown where it is assumed that $T_c = 10$ nsec, $T_f = 1000$ nsec, $N_f = 100$, $N_c = 16$, and $N = 256$. The normalized correlation mean, computed at each of the bin centers for $\beta_j = j \cdot T_f/N$ with $j = 0, 1, \cdots, N \cdot N_c - 1$,

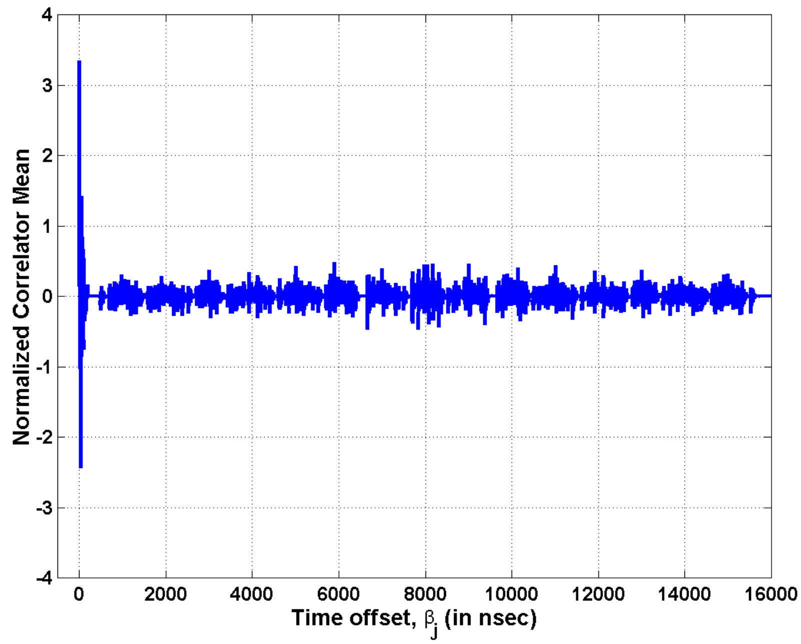

Figure 4.6: Normalized correlator mean for $N_c = 16$ and $\tau_0 = 0$ nsec

is shown in Figure 4.6 for $\tau_0 = 0$ nsec. The code sequence is $\{c_n\}_{n=0}^{N_c-1} = \{0,\ 13,\ 52,\ 43,\ 61,\ 30,\ 26,\ 48,\ 21,\ 21,\ 48,\ 26,\ 30,\ 61,\ 43,\ 52\}$. This code sequence is based upon techniques found in [47] and is a sequence of integers between 0 and 70 where the maximum value of 70 provides some guard time in each frame. A method of code design for rapid acquisition is studied in [12] for a slightly different modulation scheme. See Appendix B for further details on code selection.

The generalized signal flow graph approach of Section 2.4 is now used to study the acquisition of an ultra-wideband signal in dense multipath. The code length and the number of bins per frame are given above, thus there are $N_s = N \cdot N_c = 256 \cdot 16 = 4096$ bins in the uncertainty region. The code chip time and the frame time are $T_c = 10$ nsec and $T_f = 1000$ nsec, respectively, as above and the code sequence and multipath channel are also unchanged. Thus the normalized correlator mean of (4.23) and shown in Figure 4.6 is now used. The acquisition process analyzed here first assumes a single correlator and detection occurs when the correlator output crosses a predetermined threshold. This is then extended to use multiple correlators for the hybrid search using the results of Section 2.5.

The correlator output, $z_j = s_j + n_j$, is Gaussian with mean s_j and variance $N_c N_0$. The probability of the j^{th} code correlator output exceeding the threshold for a detection threshold of $\Upsilon \cdot \sqrt{E_p}$, where Υ is the normalized detection threshold, is:

$$\begin{aligned} P_j = \Pr(|z_j| > \Upsilon \cdot \sqrt{E_p}) = \\ Q\Big(\sqrt{\frac{E_p}{N_c N_0}} \cdot (\Upsilon + \frac{s_j}{\sqrt{E_p}})\Big) + \\ Q\Big(\sqrt{\frac{E_p}{N_c N_0}} \cdot (\Upsilon - \frac{s_j}{\sqrt{E_p}})\Big) \end{aligned} \tag{4.24}$$

The quantity s_j is given in (4.23) and the function $Q(x)$ is the Gaussian integral function. The permutation $\varepsilon(j)$ of the integers $0, 1, \cdots, N \cdot N_c - 1$ is related to the search variable β_j as:

$$\beta_j = \varepsilon(j) \cdot T_f / N \tag{4.25}$$

The initial distribution is set by the uniform nature of the direct path arrival time, τ_0, so that $\pi_{\varepsilon(j)} = 1/(N \cdot N_c)$. In the next section an alternate method of setting the initial distribution is examined. The signal flow graph path gains in Figure 2.6 are

$$H_{\varepsilon(j)}(z) = \begin{cases} P_j z & \text{if } j \in \mathcal{I} \\ 0 & \text{else} \end{cases} \tag{4.26}$$

$$G_{\varepsilon(j)}(z) = \begin{cases} (1 - P_j)z & \text{if } j \in \mathcal{I} \\ (1 - P_j)z + P_j z^{J+1} & \text{else} \end{cases} \tag{4.27}$$

The index set $\mathcal{I}$ is selected based upon the number of detectable paths in the multipath channel and for simplicity was selected as the first $K = 50$ bins. This assumption of K consecutive bins is a reasonable assumption since the arrivals are clustered, even though some paths within the cluster are small in amplitude. The mean search time in (2.50) is shown in Figure 4.7. Here the normalized detection threshold, Υ, is optimized for minimum mean acquisition time at each E_p/N_0. Again, J is the false alarm penalty time. Independent computer simulations have verified the results shown in Figure 4.7. Some of these simulation results are compared in Table 4.1.

Another UWB acquisition example is shown in Figures 4.8, 4.9, and 4.10. The code length for this example has increased to $N_c = 64$ while all of the other system parameters remain unchanged: $T_c = 10$ nsec, $T_f = 1000$ nsec, $N_f = 100$, $N = 256$, $J = 1000$. The code for this example is the 14^{th} code ($i = 14$) from the $p = 67$ family as described in Appendix B.

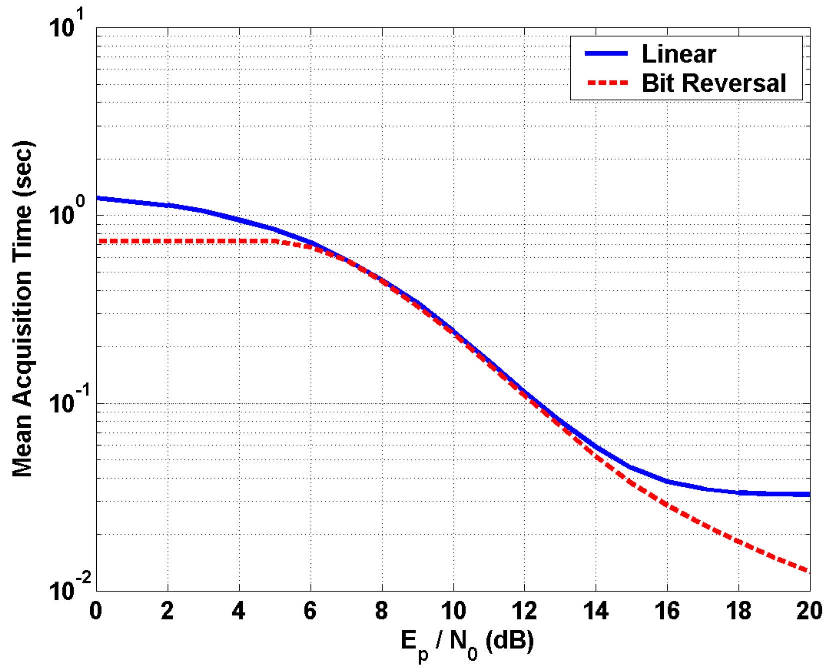

Figure 4.7: Mean acquisition time for a single correlator, T_f = 1000 nsec, T_c = 10 nsec, $N = 256$, $N_c = 16$, $J = 1000$, and optimized threshold, Υ

	$N_c = 16,\ J = 1000$ (Linear)		
E_p/N_0 =	0 dB	10 dB	20 dB
Sim Results:	1.287 sec	0.310 sec	0.0351 sec
Analytic Results:	1.238 sec	0.245 sec	0.0327 sec

	$N_c = 16,\ J = 1000$ (Bit Reversal)		
E_p/N_0 =	0 dB	10 dB	20 dB
Sim Results:	0.754 sec	0.299 sec	0.014 sec
Analytic Results:	0.729 sec	0.237 sec	0.0126 sec

Table 4.1: Comparison of the mean acquisition time with computer simulations for T_f = 1000 nsec, T_c = 10 nsec, $N = 256$, $N_c = 16$, $J = 1000$, and an optimized threshold, Υ

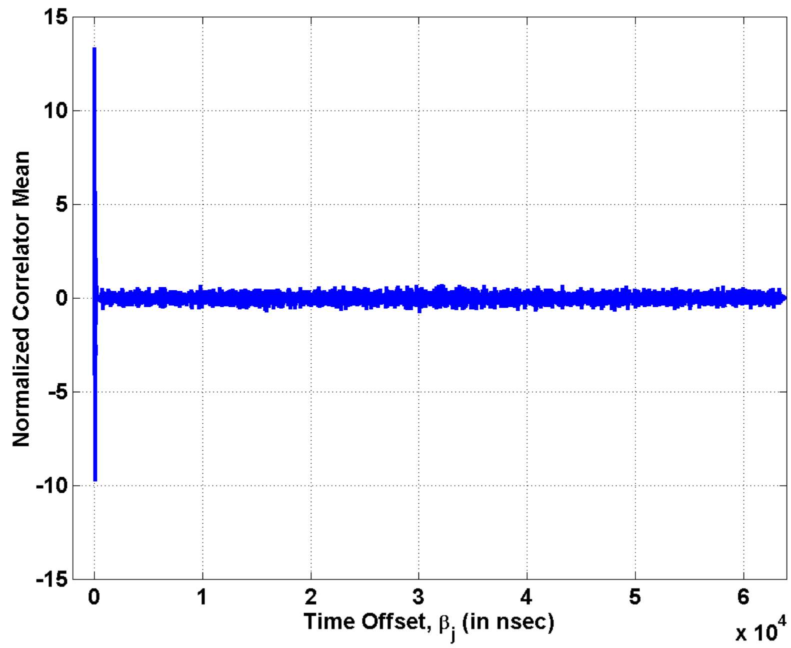

Figure 4.8: Normalized correlator mean for $N_c = 64$ and $\tau_0 = 0$ nsec

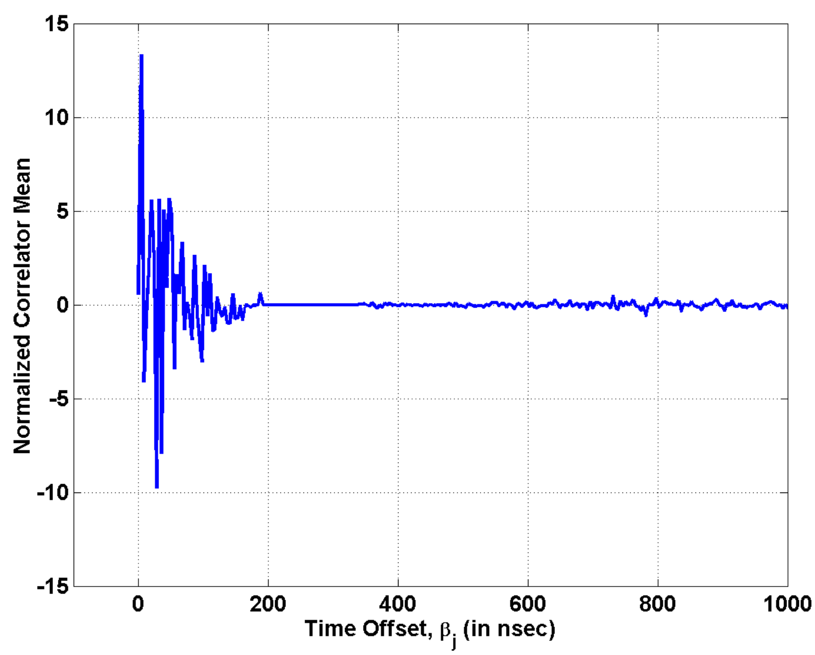

Figure 4.9: Normalized correlator mean for $N_c = 64$ and $\tau_0 = 0$ nsec (same graph as above shown over a narrower time scale)

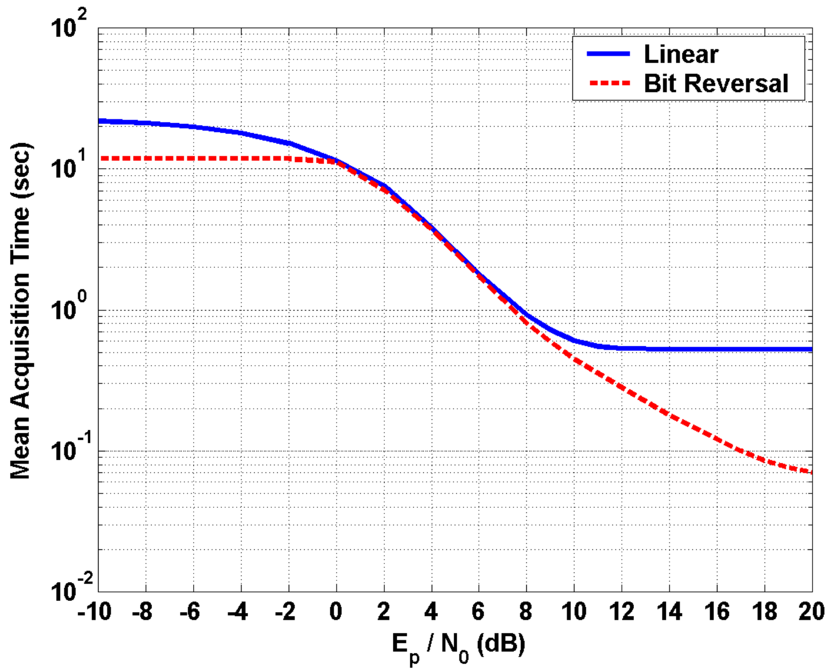

Figure 4.10: Mean acquisition time for a single correlator, $T_f = 1000$ nsec, $T_c = 10$ nsec, $N = 256$, $N_c = 64$, $J = 1000$, and optimized threshold, Υ

Another example is shown in Figures 4.11 and 4.12 where the code length has increased to $N_c = 256$. The false alarm penalty time for Figure 4.11 is $J = 1000$ while the penalty time for Figure 4.12 is $J = 100$. All of the other system parameters remain unchanged: $T_c = 10$ nsec, $T_f = 1000$ nsec, $N_f = 100$, and $N = 256$. The code for these examples is the 194^{th} code ($i = 194$) from the $p = 257$ family as described in Appendix B. The differences in the two graphs show the acquisition performance as a function of false alarm penalty time. As the signal-to-noise ratio increases the number of false alarms decreases and both graphs converge to the same mean acquisition time. At lower signal-to-noise ratios the acquisition time is dominated by the false alarm penalty time and the asymptotic ratio of the mean acquisition times for the two graphs (as the signal-to-noise ratio decreases) approaches approximately the ratio of the false alarm penalty times, $1000/100 = 10$. Although this asymptotic limit is not completely evident in the two figures, the table in Appendix A shows that this ratio is approximately 10.

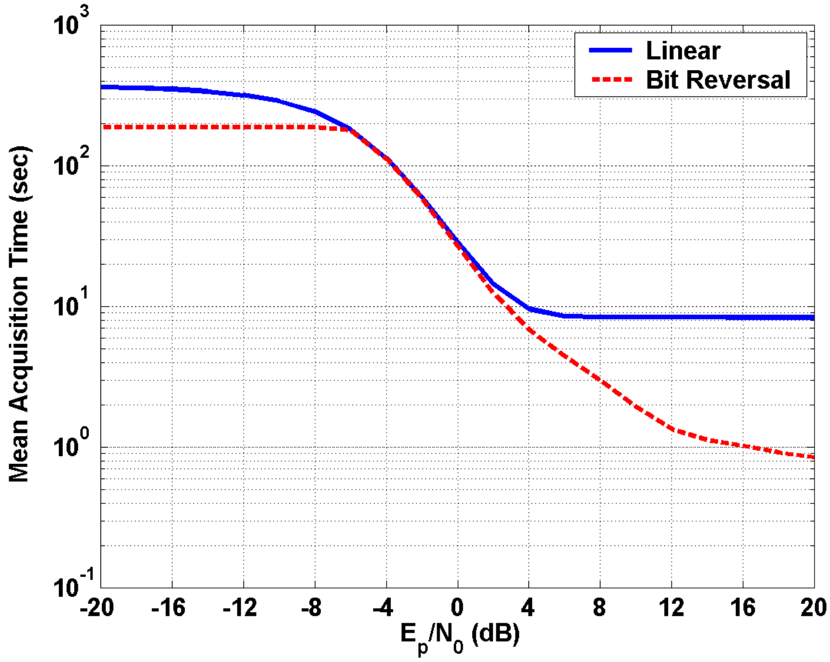

Figure 4.11: Mean acquisition time for a single correlator, $T_f = 1000$ nsec, $T_c = 10$ nsec, $N = 256$, $N_c = 256$, $J = 1000$, and optimized threshold, Υ

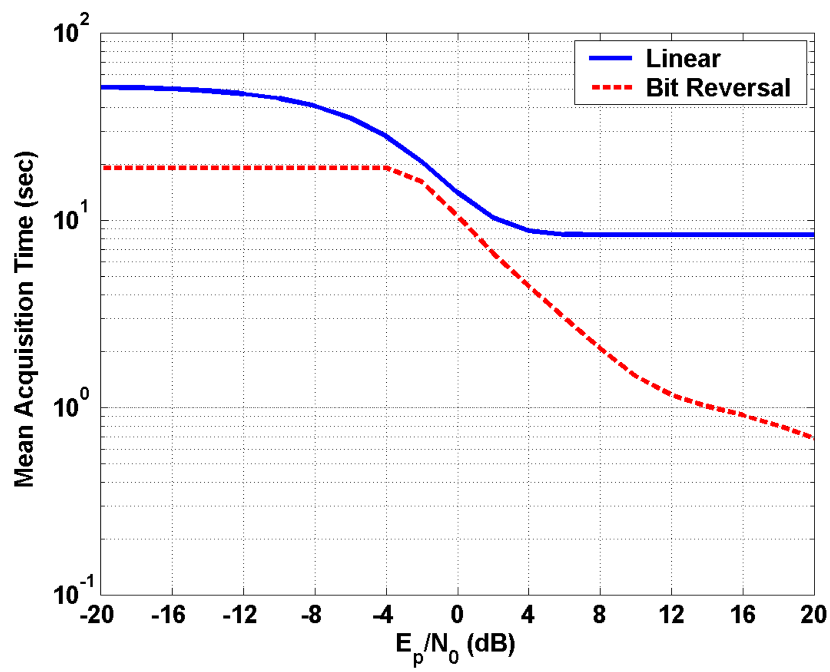

Figure 4.12: Mean acquisition time for a single correlator, $T_f = 1000$ nsec, $T_c = 10$ nsec, $N = 256$, $N_c = 256$, $J = 100$, and optimized threshold, Υ

A couple observations are evident from Figures 4.7, 4.10, 4.11, and 4.12. The first observation pertains to both the linear and bit reversal search. Specifically, the difference in code length between the various graphs increases by a factor of 4, i.e., $N_c = 16$, 64, and 256. In decibels this factor is equivalent to approximately 6 dB which, from the figures just mentioned, is precisely the amount each of the curves shifts with respect to E_p/N_0 as the code length is increased.

The second observation from Figures 4.7, 4.10, 4.11, and 4.12 pertains only to the bit reversal search. Specifically, at some point on each of the figures as the signal-to-noise ratio decreases the mean acquisition time reaches a limit. Once this limit has been reached, a decrease in the signal-to-noise ratio does not decrease the mean acquisition time. This limiting, low signal-to-noise ratio mean acquisition time can be approximated as discussed in Appendix A. The result is as follows:

$$E(T_{ACQ}) = 1 + \frac{1}{2} \cdot (J+1) \cdot \left(\frac{N \cdot N_c}{K} - 1 \right) \tag{4.28}$$

Again this is in units of dwell-times and is multiplied by the dwell-time, $N_c \cdot T_f$, to yield the mean acquisition time in seconds. This result is for a normalized detection threshold of zero, i.e., $\Upsilon = 0$. For values above the threshold signal-to-noise ratio, the optimum Υ is some non-zero value and for values below the threshold signal-to-noise ratio the optimum Υ is zero. A specific example is considered next, but first this interesting phenomenon is summarized as follows. As the observations become less and less reliable so that verification is required at each observation it is best to search the uncertainty region not in a linear fashion, but in such a way as to sufficiently spread the distance between observations. This, of course, applies when the detection bins are clustered together in some local region.

Taking the $N_c = 16$ case as a specific example, Figures 4.13 and 4.14 show the mean acquisition time for a linear and bit reversal search, respectively, for normalized detection threshold values ranging from $\Upsilon = 0$ to $\Upsilon = 15$. The lower bounds from both of these figures, as shown, give the minimum mean acquisition times of Figure 4.7. For the bit reversal search, Figure 4.14 shows that for signal-to-noise ratios below approximately 6 dB the mean acquisition time is minimized when $\Upsilon = 0$. For values 6 dB and above the mean acquisition time is minimum for some $\Upsilon > 0$. Figure 4.15 shows the bit reversal mean acquisition time for $N_c = 16$ as a function of Υ at a number of different signal-to-noise ratios.

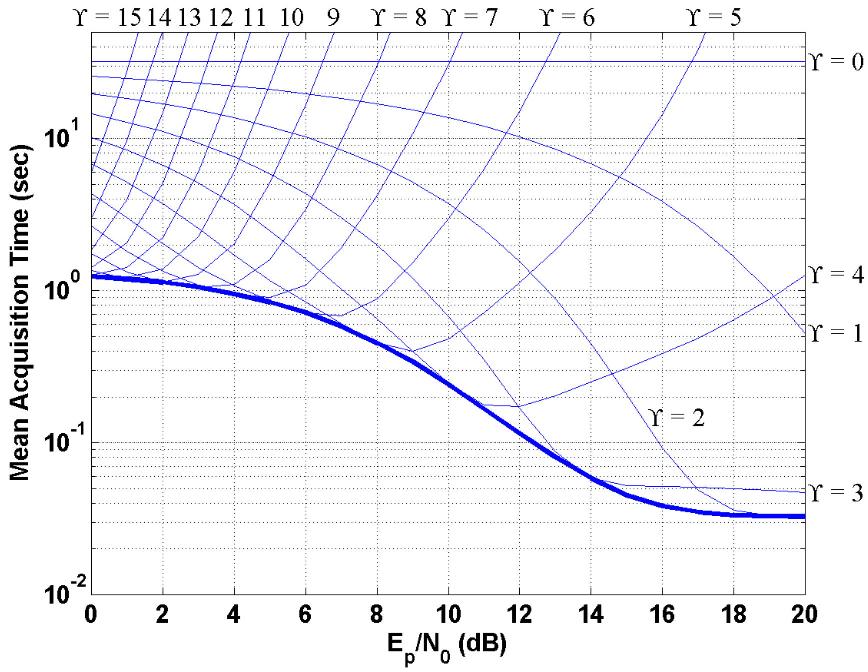

Figure 4.13: Mean acquisition time for a single correlator using a linear search, $N_c = 16$, $T_f = 1000$ nsec, $T_c = 10$ nsec, $N = 256$, $J = 1000$

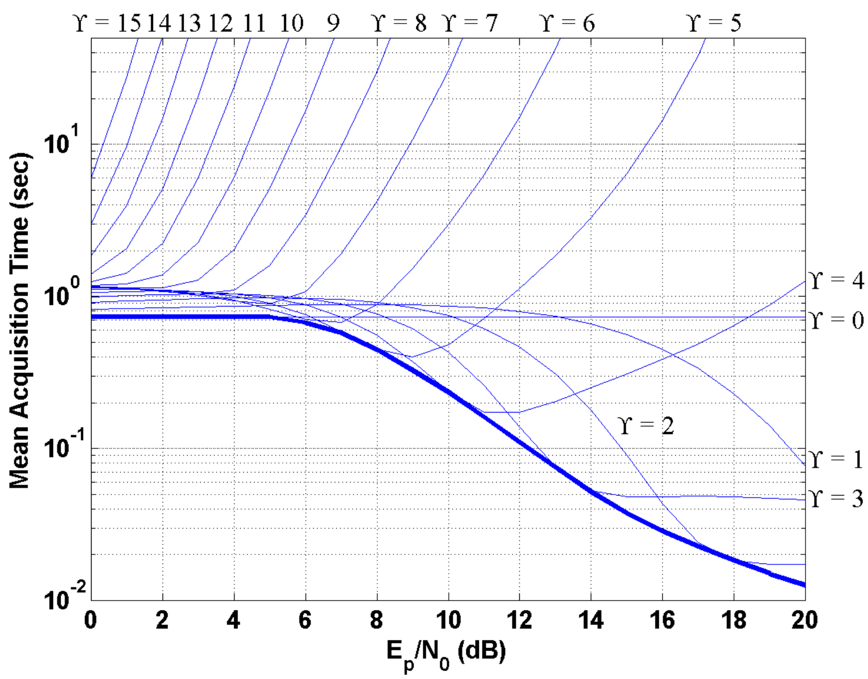

Figure 4.14: Mean acquisition time for a single correlator using a bit reversal search, $N_c = 16$, $T_f = 1000$ nsec, $T_c = 10$ nsec, $N = 256$, $J = 1000$

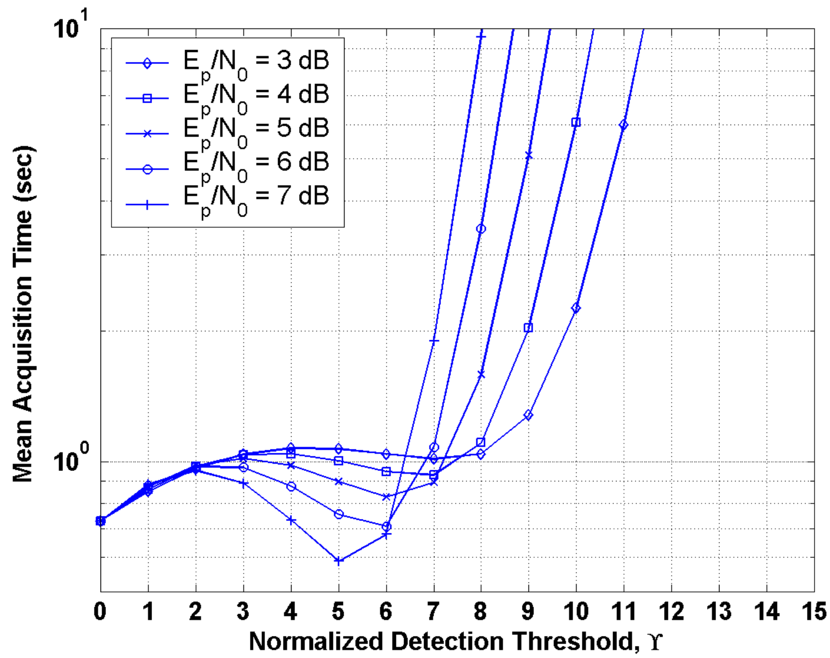

Figure 4.15: Mean acquisition time for a single correlator using a bit reversal search, $N_c = 16$, $T_f = 1000$ nsec, $T_c = 10$ nsec, $N = 256$, $J = 1000$

4.3 Single User Hybrid Acquisition of Coded UWB Systems

In this section coarse acquisition will be examined assuming multiple correlators are present in the receiver. The analysis techniques examined in Section 2.5 are employed here to determine the mean acquisition time of a UWB signal in dense multipath. The direct path arrival time, τ_0, is uniform over a code period so that the mean acquisition time is the average over this arrival time, as shown in (2.58). With adequate bin spacing, that is for sufficiently large values of N, it is reasonable to assume that τ_0 is uniform on the discrete set $\{k_1 \cdot T_f/N\}_{k_1=0}^{N_s-1}$. This significantly reduces the computational complexity associated with generating the mean acquisition time by allowing (2.57) to be used:

$$E(T_{ACQ}) = \frac{1}{N_s} \sum_{k_1=0}^{N_s-1} E(T_{ACQ}|k_1) \tag{4.29}$$

As before, $N_s = N \cdot N_c$ and the conditional mean acquisition time is

$$E(T_{ACQ}|k_1) = \frac{\mathrm{d}}{\mathrm{d}z} P_{ACQ}(z|k_1)\bigg|_{z=1} = \frac{Num' \cdot Den - Num \cdot Den'}{Den^2} \tag{4.30}$$

The conditional moment generating function is given in (2.56) so that the numerator, denominator, and the associated derivatives are:

$$Num = \sum_{i=0}^{N_s-1} H_{\varepsilon(i)}(1) \prod_{j=0}^{i-1} G_{\varepsilon(j)}(1) \tag{4.31}$$

$$Num' = \sum_{i=0}^{N_s-1} \left(\prod_{j=0}^{i-1} G_{\varepsilon(j)}(1) \right) \cdot \left[H'_{\varepsilon(i)}(1) + H_{\varepsilon(i)}(1) \sum_{l=0}^{i-1} \frac{G'_{\varepsilon(l)}(1)}{G_{\varepsilon(l)}(1)} \right] \tag{4.32}$$

$$Den = 1 - \prod_{i=0}^{N_s-1} G_{\varepsilon(i)}(1) \tag{4.33}$$

$$Den' = -\sum_{i=0}^{N_s-1} \frac{G'_{\varepsilon(i)}(1)}{G_{\varepsilon(i)}(1)} \cdot \prod_{j=0}^{N_s-1} G_{\varepsilon(j)}(1) \tag{4.34}$$

The dependence of the conditional mean acquisition time on k_1 occurs through the path gains:

$$H_{\varepsilon(i)}(z) = \begin{cases} P_{\varepsilon(i)\ominus k_1} z & \text{if } i \in \mathcal{I}(k_1) \\ 0 & \text{else} \end{cases} \tag{4.35}$$

$$G_{\varepsilon(i)}(z) = \begin{cases} 1 - P_{\varepsilon(i)\ominus k_1} & \text{if } i \in \mathcal{I}(k_1) \text{ and } i \notin \mathcal{B} \\ (1 - P_{\varepsilon(i)\ominus k_1})z & \text{if } i \in \mathcal{I}(k_1) \text{ and } i \in \mathcal{B} \\ 1 - P_{\varepsilon(i)\ominus k_1} + P_{\varepsilon(i)\ominus k_1} z^J & \text{if } i \notin \mathcal{I}(k_1) \text{ and } i \notin \mathcal{B} \\ (1 - P_{\varepsilon(i)\ominus k_1})z + P_{\varepsilon(i)\ominus k_1} z^{J+1} & \text{if } i \notin \mathcal{I}(k_1) \text{ and } i \in \mathcal{B} \end{cases} \tag{4.36}$$

As described in Section 2.5, a single search permutation, $\varepsilon(n)$ is divided amongst all M correlators and only after all M observations have been made has a dwell-time elapsed. This is reflected in the path gains by the boundary set $\mathcal{B} = \{M-1, 2M-1, \cdots, N_s-1\}$ where it assumed that N_s/M is an integer. In the above path gains, the operator $\ominus$ represents modulo N_s integer subtraction. When both arguments x and y are integers between 0 and $N_s - 1$, as is the case above, then $x \ominus y$ simply becomes $x - y$ when $x \geq y$ and $N_s + x - y$ when $x < y$. The detection set as a function of k_1, assuming K consecutive detection bins as before, is $\mathcal{I}(k_1) = \{\varepsilon^{-1}(k_1), \varepsilon^{-1}(k_1 \oplus 1), \cdots, \varepsilon^{-1}(k_1 \oplus K-1)\}$ where $\oplus$ represents modulo N_s integer addition. The transition probabilities, P_j, are computed as per (4.24). The correlator means, s_j, are computed a little differently. Computation of these means in (4.23) is based upon the local template timing offset term, $\alpha_{j,k}$, in (4.22). In the last section this term was a function of the search permutation $\varepsilon(n)$ via the expression $\varepsilon(j) \cdot T_f/N = k_{\beta,j}T_c + \beta_{r,j}$. In this section, however, the search permutation has already been incorporated into the path gains as shown above. Thus the correlator mean must be computed as per $j \cdot T_f/N = k_{\beta,j}T_c + \beta_{r,j}$, i.e., the correlator means are listed out linearly as a function of j.

Some results are shown in Figures 4.16 and 4.17 for a code length of $N_c = 16$ and for 1, 2, 4, 8, and 16 correlators. As can be seen from both graphs, the search time for multiple correlators is always reduced relative to a single correlator. In fact, as evident from the linear hybrid search of Figure 4.16, for sufficiently large values of the signal-to-noise ratio the mean acquisition time for M correlators is simply the mean acquisition time for a single correlator divided by M. As in the last section the graphs show the mean search time minimized with respect to the normalized detection threshold, Υ, and each correlator uses this same threshold.

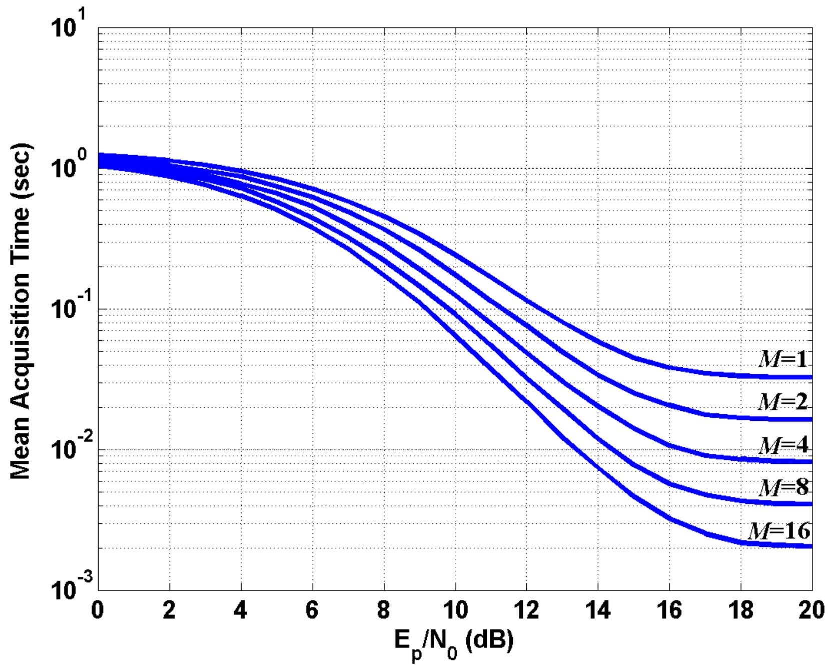

Figure 4.16: Mean acquisition time for M correlators using a hybrid linear search, $N_c = 16$, $T_f = 1000$ nsec, $T_c = 10$ nsec, $N = 256$, $J = 1000$

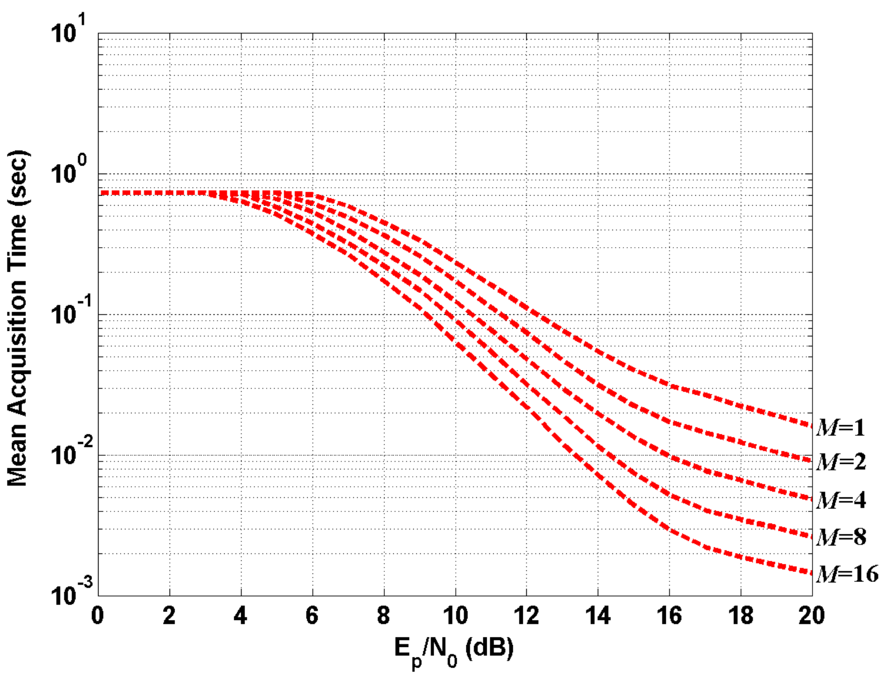

Figure 4.17: Mean acquisition time for M correlators using a hybrid bit reversal search, $N_c = 16$, $T_f = 1000$ nsec, $T_c = 10$ nsec, $N = 256$, $J = 1000$

4.4 Acquisition in the Presence of Multiple Users

Up until now a single transmitter has been assumed. This assumption will be removed in this section and N_u active transmitters will be present. One of these transmitters produces the signal of interest which is to be detected at the receiver while the remaining $N_u - 1$ signals will contribute to multiple access interference (MAI). Each user will have its own time-hopping code which is different than the user-of-interest. Also, all the users are assumed to be spatially separated giving rise to different multipath channels for each user. Here we will examine the output of a single correlator and not the combined RAKE output since this single correlator output is sufficient to study acquisition performance. Multi-user detection schemes exist which provide improved performance in the presence of MAI [44] [16]. These concepts will not be examined here, however, and the MAI is shown to be mean zero additive Gaussian noise which is simply added to the thermal noise at the correlator output [46].

The k^{th} user at the receiver input has the following form, where the superscript, (k), represents the user number for $k = 0, \cdots, N_u - 1$ as in Section 1.2.

$$s^{(k)}(t) = \sqrt{E_p^{(k)}} \sum_n \sum_{i=0}^{L_p-1} a_i^{(k)} \cdot p\left(t - nT_f - c_n^{(k)}T_c - \tau_i^{(k)}\right) \tag{4.37}$$

The overall received signal is the sum of all the users and thermal noise, which can be written as the sum of the signal-of-interest (SOI), multi-access interference (MAI), and additive, white, Gaussian noise (AWGN):

$$r(t) = \underbrace{s^{(0)}(t)}_{SOI} + \underbrace{\sum_{k=1}^{N_u-1} s^{(k)}(t)}_{MAI} + \underbrace{n(t)}_{AWGN} \tag{4.38}$$

Here it is assumed, without loss of generality, that the signal-of-interest is represented by the $k = 0$ transmitter. The local template will thus correlate the received signal using the code sequence, $c_m^{(0)}$. This template is defined as in (4.21):

$$l(t) = \sum_j \sum_{m=0}^{N_c-1} p(t - \alpha_{j,m}^{(0)} - jN_cT_f) \tag{4.39}$$

As before the search location and code phase is set by the time-offset parameter, $\alpha_{j,m}^{(0)}$. This parameter is a function of the search location, $\beta_j = k_{\beta,j}T_c + \beta_{r,j}$, which can vary over the entire code period.

$$\alpha_{j,m}^{(0)} = \Big((mN_f + c_m^{(0)} + k_{\beta,j}) \bmod N_fN_c\Big)T_c + \beta_{r,j} \tag{4.40}$$

The correlator output, $z_j = s_j^{(0)} + \sum_{k=1}^{N_u-1} s_j^{(k)} + n_j$, is determined as the product of the received signal in (4.38) and the template in (4.39) then appropriately integrated and summed over a code period as in Section 4.2. Analogous to (4.23), the signal portion of the correlator output corresponding to the k^{th} user is found to be

$$
\begin{aligned}
s_j^{(k)} = \sqrt{E_p^{(k)}} \sum_{n=(j-1)N_c}^{(j+1)N_c-1} \sum_{m=0}^{N_c-1} \sum_{i=0}^{L_p-1} a_i^{(k)} \cdot \gamma(\tau_0^{(k)} + \Delta\tau_i^{(k)} - \alpha_{j,m}^{(0)} \\
+ c_{(n \bmod N_c)}^{(k)} T_c - (jN_c - n)T_f)
\end{aligned}
\tag{4.41}
$$

As before, $\gamma(\tau)$ represent the pulse autocorrelation function of $p(t)$. The following assumptions are made with respect to the multipath channel parameters and time-hopping codes:

1. $\tau_0^{(k)}$ is a sequence of IID random variables, each uniform on $[0, N_c T_f)$ for $k = 0, \cdots, N_u - 1$
2. $a_i^{(k)}$ and $\Delta\tau_i^{(k)}$ are deterministic parameters which are unknown to the receiver (the multipath channels are assumed static over the acquisition time)
3. $c_n^{(0)}$ is deterministic and is the SOI time-hopping code being acquired at the receiver
4. $c_n^{(k)}$ is a group of IID random variables, each of which is discrete uniform on the set $\{0, 1, \cdots, N_g\}$ for $k = 1, \cdots, N_u - 1$ and $n = 0, \cdots, N_c - 1$
5. $c_n^{(k)}$ and $\tau_0^{(k)}$ are statistically independent for all k and n

The mean of $s_j^{(k)}$ for $k = 0$ is given in (4.23) and for $k \neq 0$ using the assumptions above we see that:

$$E\left(s_j^{(k)}\right) = \sqrt{E_p^{(k)}} \sum_{n=(j-1)N_c}^{(j+1)N_c-1} \sum_{m=0}^{N_c-1} \sum_{i=0}^{L_p-1} a_i^{(k)} \cdot \frac{1}{N_c \cdot T_f \cdot (N_g+1)} \cdot$$
$$\sum_{l=0}^{N_g} \int_0^{N_c T_f} \gamma(\tau_0^{(k)} + \Delta\tau_i^{(k)} - \alpha_{j,m}^{(0)} + l \cdot T_c - (jN_c - n)T_f) \mathrm{d}\tau_0^{(k)} \tag{4.42}$$

This mean was computed for $N_c = 16$, a variety of multipath channels and SOI time-hopping codes and seen to be very close to zero. The same result holds for longer code lengths. The main factors contributing to this zero mean situation are 1) the integral of the autocorrelation function is approximately zero and 2) the autocorrelation function is zero everywhere except a narrow window around zero so that most of the terms contribute nothing to the overall sum. Independent simulation also verified this zero mean calculation.

The variance of $s_j^{(k)}$ for $k \neq 0$ and conditioned directly on the interfering signal's time-hopping code is seen to be

$$E\left[\left(s_j^{(k)}\right)^2 | c_0^{(k)}, \cdots, c_{N_c-1}^{(k)}\right] = E_p^{(k)} \sum_{n_1=(j-1)N_c}^{(j+1)N_c-1} \sum_{m_1=0}^{N_c-1} \sum_{i_1=0}^{L_p-1} \sum_{n_2=(j-1)N_c}^{(j+1)N_c-1} \sum_{m_2=0}^{N_c-1} \sum_{i_2=0}^{L_p-1} a_{i_1}^{(k)} a_{i_2}^{(k)} \cdot$$
$$E\left[\gamma\left(\tau_0^{(k)} + \Delta\tau_{i_1}^{(k)} - \alpha_{j,m_1}^{(0)} + c_{n_1 \bmod N_c} \cdot T_c - (jN_c - n_1)T_f\right) \cdot\right.$$
$$\left.\gamma\left(\tau_0^{(k)} + \Delta\tau_{i_2}^{(k)} - \alpha_{j,m_2}^{(0)} + c_{n_2 \bmod N_c} \cdot T_c - (jN_c - n_2)T_f\right)\right] \tag{4.43}$$

This expression has an extremely large number of terms even for short code lengths, e.g., $N_c = 16$, thus making direct computation prohibitive. The following simplifying assumptions can be made which lead to a reasonable approximation to the variance, as verified via simulation. Firstly, the time-hopping code of the interfering user can be ignored, i.e., $c_n^{(k)} = 0$ for all n and for $k = 1, \cdots, N_u - 1$. Secondly, all off-diagonal terms can be ignored, i.e., those terms such that $n_1 \neq n_2$, $m_1 \neq m_2$, and $i_1 \neq i_2$. This leads to the following approximation to the variance for $k \neq 0$:

$$E\left[\left(s_j^{(k)}\right)^2\right] \approx E_p^{(k)} \sum_{n=(j-1)N_c}^{(j+1)N_c-1} \sum_{m=0}^{N_c-1} \sum_{i=0}^{L_p-1} \left(a_i^{(k)}\right)^2 \cdot \frac{1}{N_c T_f} \cdot \int_0^{N_c T_f} \gamma^2\left(\tau_0^{(k)} + \Delta\tau_i^{(k)} - \alpha_{j,m}^{(0)} - (jN_c - n)T_f\right) \mathrm{d}\tau_0^{(k)} \tag{4.44}$$

This summation was found via direct computation to have only ${N_c}^2$ surviving terms, all of which were equal to $\frac{1}{N_c} \cdot E_p^{(k)} \cdot \sigma_\gamma^2 \cdot \sum_{i=0}^{L_p-1} \left(a_i^{(k)}\right)^2$ where σ_γ^2 is given below. The multipath channels are assumed here to consist only of L_p paths and thus from the multipath coefficient normalization of Section 1.6 it is seen that $\sum_{i=0}^{L_p-1} \left(a_i^{(k)}\right)^2 = 1$. Thus the overall variance is reasonably well approximated by:

$$E\left[\left(s_j^{(k)}\right)^2\right] \approx E_p^{(k)} \cdot N_c \cdot \sigma_\gamma^2 \tag{4.45}$$

where

$$\sigma_\gamma^2 = \frac{1}{T_f} \int_{-\infty}^{\infty} \gamma^2(\tau) \mathrm{d}\tau \tag{4.46}$$

The above approximation to the variance of the MAI for a single interferer was seen via simulation to be quite accurate. For example, when $N_c = 16$, $T_f = 1000$

nsec, and $E_p^{(k)} = 1$ it is seen that $E_p^{(k)} \cdot N_c \cdot \sigma_\gamma^2 = 0.0078$ while simulations consisting of $100,000$ randomly generated time-hopping code sequences and direct path arrivals, $\tau_0^{(k)}$, yielded values averaging 0.0083 for the same multipath amplitude coefficients, $a_i^{(k)}$, and relative path arrival times, $\Delta\tau_i^{(k)} = \tau_i^{(k)} - \tau_0^{(k)}$. It was also seen by direct computation, and verified via simulation, that the statistics varied little for different values of j, thus removing any dependence of the search permutation on the MAI statistics.

Now that the individual interferer statistics have been computed, the total multi-access interference can be examined. The multi-access interference term, $\sum_{k=1}^{N_u-1} s_j^{(k)}$, at the correlator output is the sum of a number terms. Each of these terms has zero mean, so that the mean of the sum is also zero. The variance of each term in the sum is a function of signal energy, $E_p^{(k)}$. The signal energy of the k^{th} user can be rewritten in terms of the SOI energy, $E_p^{(0)}$, and a scale factor, A_k.

$$E_p^{(k)} = A_k \cdot E_p^{(0)} \tag{4.47}$$

The assumptions listed earlier ensure that the individual terms, $s_j^{(k)}$, are independent and thus the variance of their sum is simply the sum of the individual variances:

$$\sigma_{MAI}^2 = E_p^{(0)} \cdot N_c \cdot \sigma_\gamma^2 \cdot \sum_{k=1}^{N_u-1} A_k \tag{4.48}$$

Assuming that the MAI is Gaussian, an assumption that will be discussed in more detail shortly, the variance can simply be added to the thermal noise variance, $N_c N_0$, at the correlator output.

$$\begin{aligned}
\sigma_n^2 + \sigma_{MAI}^2 &= N_c N_0 + E_p^{(0)} \cdot N_c \cdot \sigma_\gamma^2 \cdot \sum_{k=1}^{N_u-1} A_k \\
&= N_c \cdot \left(N_0 + E_p^{(0)} \cdot \sigma_\gamma^2 \cdot \sum_{k=1}^{N_u-1} A_k \right) \qquad (4.49)
\end{aligned}$$

Thus it appears as if the thermal noise variance has increased by the term shown above. Denoting the ratio of SOI signal energy to total noise variance as a function of the number of users, $SNR(N_u)$, it is noted that $SNR(1) = E_p^{(0)}/N_0$ and

$$SNR(N_u) = \frac{1}{\frac{1}{SNR(1)} + \sigma_\gamma^2 \sum_{k=1}^{N_u-1} A_k} \qquad (4.50)$$

A plot of this signal-to-noise ratio is shown in Figure 4.18 for three different single user SNR's. Here it is assumed that perfect power control exists and that $A_k = 1$ for all k. The impact of additional users on the mean acquisition time is seen through the decrease in SNR. For example, it is seen if the initial signal-to-noise ratio is $SNR(1) = 10$ dB then for 2000 users $SNR(2000) = -0.33$ dB. From Figure 4.10, which is for a code length of $N_c = 64$, the mean acquisition time for a bit reversal search at an SNR of 10 dB is approximately 0.45 sec while at an SNR of -0.33 dB the mean acquisition time has increased to 11.7 sec. However, if $SNR(1) = 0$ dB then for 2000 users the SNR has dropped to around -3 dB. This time, from Figure 4.10, we see that the mean acquisition time changes very little, only by approximately 0.5

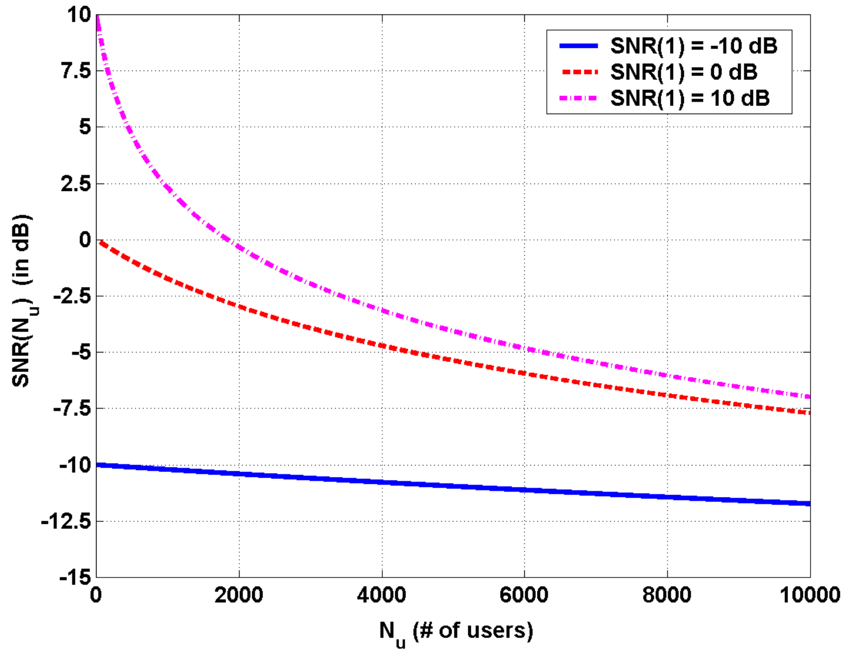

Figure 4.18: Change in signal-to-noise ratio for multiple users

sec. This is because the change in SNR for the latter case is much smaller. As can be seen from Figure 4.18, the SNR decreases more rapidly as users are added to the system for a larger SOI signal-to-noise ratio. This simply shows that for the lower SOI signal-to-noise ratio the total noise is dominated by the thermal noise.

If perfect power control exists then the individual terms of the MAI have equal variance and the central limit theorem can be used to see that the MAI converges in distribution to a Gaussian random variable. The density evolution with an increasing number of users is shown in Figure 4.19 assuming $E_p^{(k)} = 1$ for all k. Here histograms are shown for $N_u - 1 = 1$ through $N_u - 1 = 30$, where $N_u - 1$ is the number of additional users above and beyond the SOI.

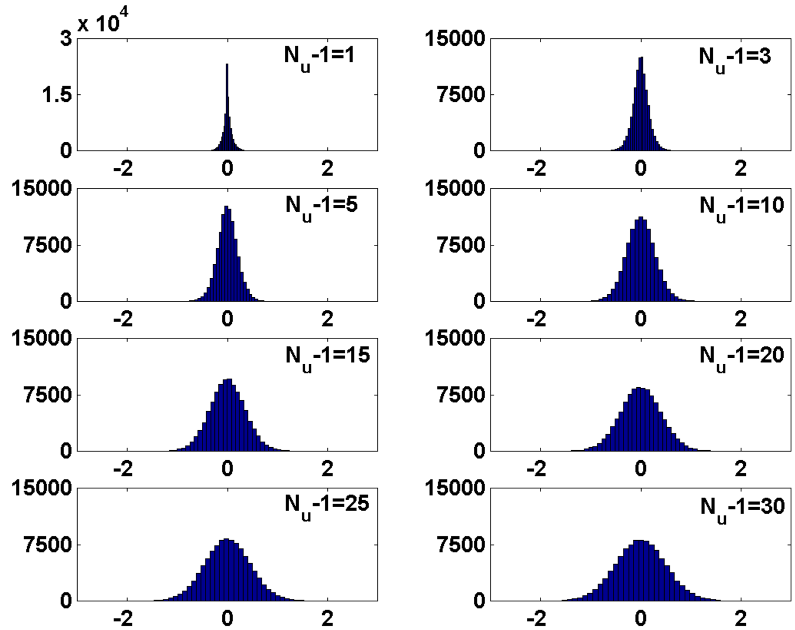

Figure 4.19: Histograms of multiple-access interference for 1 to 30 additional users

Perfect power control is one sufficient condition which insures that the multiple-access interference is Gaussian as discussed above. This, however, is not a necessary condition. A simple example will demonstrate this. Specifically, the case of imperfect power control was examined where each of the scale factors, A_k, were assumed to be an independent sequence of random variables, each uniform on the interval $[0.5, 1.5]$. The mean is obviously $E(A_k) = 1$ and as expected the MAI for a sufficiently large number of users was found via simulation to be Gaussian in nature. The histograms, in fact, are very similiar to the ones shown in Figure 4.19 and the density evolution with N_u occurs at the same rate for the two cases.

Chapter 5

Fine Acquisition of UWB Systems

As discussed in Section 1.3, the various arrival paths at the output of a UWB channel tend to 'cluster' into groups. The coarse acquisition process described in the previous chapter has only determined the location of this multipath cluster, with some associated probability. The *fine acquisition* process following coarse acquisition has two main purposes: verification and channel estimation. As will be seen below, fine acquisition can be performed in a manner that allows both of these objectives to be accomplished simultaneously to some extent. The desired output is a reliable set of M path estimates in the multipath cluster. The total number of arrival paths is assumed to be a finite number, L_p. As will be seen, exactly which M of the L_p paths are estimated is dependent on the algorithm employed. Ideally maximal ratio combining in a RAKE receiver architecture is desired. As described in Section 1.6, this requires the M strongest paths to be estimated in order to maximize the RAKE output signal-to-noise ratio.

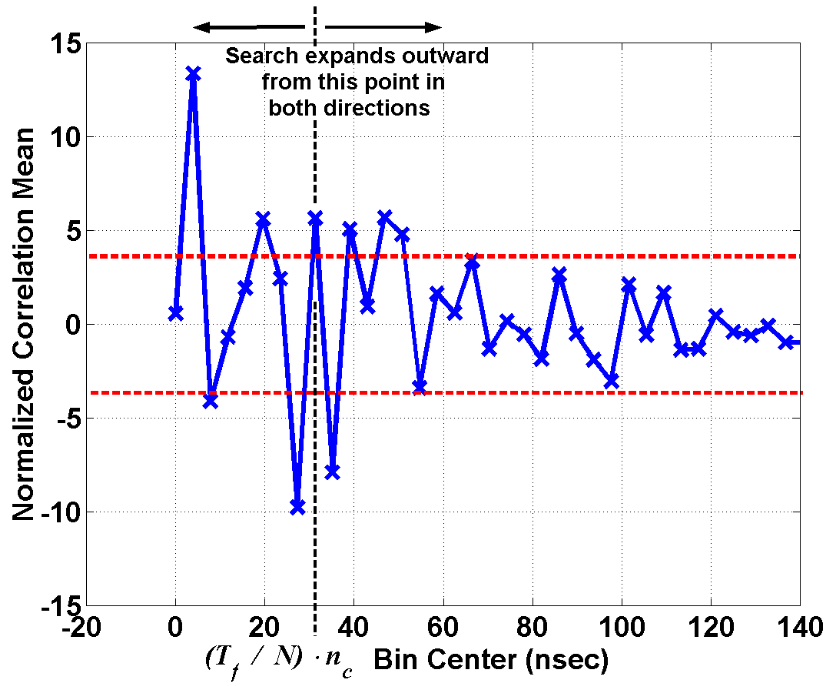

Figure 5.1: Normalized correlator mean for a code length of $N_c = 64$, $T_f = 1000$ nsec, $N = 256$, $T_c = 10$ nsec. The coarse acquisition termination point is denoted as n_c and is the starting point for fine acquisition.

The clustering phenomenon of the dense multipath channel suggests that when a single path has been located the immediate vicinity should be searched for other paths. The bin spacing and the amount of time spent dwelling at each bin are assumed to be identical to that of the coarse search, although this is not a requirement. Figure 5.1 shows an example of the fine acquisition process. Coarse acquisition has terminated at the bin, n_c, which corresponds to a time offset of $n_c \cdot \frac{T_f}{N}$. Some subset of the M correlators then search the neighboring bins outward away from n_c until some stopping criterion is satisfied. In all the algorithms to follow it is assumed that all M correlators are used for fine acquisition. The general problem of optimal resource allocation between coarse and fine acquisition has not been examined and is discussed briefly in Chapter 6.

Generally an upper limit is set on the distance that can be traveled relative to n_c before declaring a false alarm and returning to coarse acquisition. This upper limit directly affects the false alarm penalty time and is assumed to be an integer multiple of M, say $\pm kM$ with respect to n_c. For the m^{th} correlator at the j^{th} dwell-time, the output is defined as $z_j^{(m)} = s_j^{(m)} + n_j^{(m)}$ for $m = 0, 1, \cdots, M-1$. The correlator mean is given in (4.23) and the noise is Gaussian with zero mean and variance $N_c N_0$. This group of M correlator outputs can then be used to decide whether or not the search should terminate. Some example stopping criteria are listed below under the assumption that the uncertainty region is fixed by the upper limit $\pm kM$:

1. Search in both directions and stop after the entire uncertainty region has been examined

2. Search in one direction followed by the other direction and stop at any point if $|z_j^{(m)}| < T_1$ for all m

3. Search in one direction followed by the other direction and stop at any point if $\sum_m |z_j^{(m)}|^n < T_3$ for some constant n, e.g., $n = 1$ or 2

4. Search in one direction until $|z_j^{(m)}| < T_2$ for all m then search in the other direction and stop if $|z_j^{(m)}| < T_2$ for all m

Independent of the stopping criterion is a verification criterion, which basically determines if a false alarm has occurred. It is assumed that the largest correlator magnitudes are stored during the fine acquisition process in a vector $\mathbf{v} = [\mathrm{v}_1, \mathrm{v}_2, \cdots, \mathrm{v}_M]$ with the corresponding locations stored in a different vector. After every dwell-time the components of $\mathbf{v}$ are updated and v_1 represents the maximum magnitude, v_2 represents the second largest, and so on. Once the search has stopped, a decision must be made as to whether or not the estimates in $\mathbf{v}$ are adequate. Some example verification criteria include:

1. Fine acquisition is complete if $\mathrm{v}_M > T_4$

2. Fine acquisition is complete if $\sum_m \mathrm{v}_m^n > T_5$ for some n, e.g., $n = 1$ or 2

3. Coarse acquisition is resumed if $\mathrm{v}_m < T_6$ for some m

If the estimates in $\mathbf{v}$ pass the verification phase then the next phase of synchronization encountered is tracking. If the bin spacing is sufficiently small so that the estimates are within the tracking pull-in range then $\mathbf{v}$ can be passed directly to the tracking processor. Otherwise an intermediate stage is required which reduces the timing uncertainty on each path estimate to a sufficient level. This process, termed *coarse tracking*, is discussed briefly in Chapter 6. Basically the coarse tracking process either successfully locks onto M paths and begins continual signal tracking and data detection or it rejects the estimates and initiates either coarse or fine acquisition. It is possible for only a subset of the estimates to be rejected and in this case some paths are tracked while a number of correlators continue to scan the area near the tracked paths. This same procedure is followed if any of the paths fall out of lock once tracking has begun.

For the fine acquisition algorithms analyzed below, it will be assumed that acquisition is complete once the verification criterion has been satisfied for those bins in the detection set. If the verification criterion is satisfied for improper bins then a fixed penalty time will be incorporated and acquisition will be resumed. This is analogous to the false alarm penalty time seen in Chapter 4. In both cases this false alarm penalty time is random and in the interest of simplifying the analysis a fixed number has been used. This number could possibly be the mean of the false alarm penalty time or if a more worst case analysis is required a bound based upon the standard deviation could be used.

Selection of a stopping criterion and a corresponding verification criterion from the two lists above results in many different possibilities, all of which can be analyzed using the self-similar signal flow graphs introduced in Chapter 3. For example, the analysis of stopping criterion 1, which is examined in detail below, can be performed using the signal flow graph shown in Figure 3.6. Similarly, stopping criterion 2 and stopping criterion 3 require the use of the graph in Figure 3.13 while stopping criterion 4 requires the graph in Figure 3.14. From this point forward, use of verification criterion 1 is assumed. The analysis of the other verification criteria can be performed in a similar manner to the analysis shown in the next section.

5.1 Verification Probability Using Non-IID Order Statistics

The components of $\mathbf{v}$ are updated every dwell-time during fine acquisition as more of the search space is examined. Thus the total number of observed random variables increases each dwell-time. These random variables are independent Gaussian random variables with equal variance and differing means so that the components of $\mathbf{v}$ are order statistics of a group of non-IID random variables. Appendix C discusses an efficient, iterative method for computing the distribution function for these order statistics.

The probability of completing fine acquisition via verification criterion 1 requires the smallest component of $\mathbf{v}$ to be larger than some threshold. This is equivalent to all the components of $\mathbf{v}$ exceeding some threshold by virtue of the vector's construction. Here, the order statistics are drawn from the entire uncertainty region, which is assumed to consist of $2kM + 1$ bins. These bins include the coarse acquisition termination point, $\varepsilon(n)$, and the kM bins to the left and the kM bins to the right of $\varepsilon(n)$. The self-similar signal flow graph used to analyze this situation is shown in Figure 3.6. The verification probability associated with this graph for $n = 0, 1, \cdots, N_s - 1$ is

$$P_{\mathrm{v},n} = \Pr(\mathrm{v}_M > T) = 1 - \Pr(\mathrm{v}_M \leq T) \tag{5.1}$$

The probability $\Pr(\mathrm{v}_M \leq T)$ is given in (C.9) and thus the verification probability is:

$$P_{\mathrm{v},n} = 1 - \sum_{m=1}^{M} H^*_{2kM+2-m} \cdot \prod_{j=-kM}^{kM} (1 - R_{n \oplus j}) \tag{5.2}$$

The $\oplus$ operator represents modulo N_s integer addition. From Appendix C the terms, H^*_m, not to be confused with the signal flow graph path gains, are as follows for $m = 2kM - M + 2,\ 2kM - M + 3,\ \cdots,\ 2kM$:

$$H^*_m = \frac{1}{2kM + 1 - m} \cdot \sum_{i=1}^{2kM+1-m} (-1)^{i+1} L_{-i} H^*_{m+i} \tag{5.3}$$

The $m = 2kM + 1$ term is seen from Appendix C to be unity, i.e., $H^*_{2kM+1} = 1$. The factor L_{-i} above is defined in terms of the distribution function and the complementary distribution function of the random variables as shown in (C.8), which is adapted here in terms of the probabilities R_j.

$$L_{-i} = \sum_{j=-kM}^{kM} \left(\frac{1 - R_{n \oplus j}}{R_{n \oplus j}} \right)^{-i} \tag{5.4}$$

The probability, R_j, represents the probability that the correlator magnitude exceeds some given threshold. If the coarse acquisition and fine acquisition processes were to use the same detection threshold then R_j would simply be the transition probability, P_j, properly computed for hybrid acquisition as discussed in Section 4.3.

$$R_j = \Pr(|z_j| > \Upsilon_F \cdot \sqrt{E_p})$$

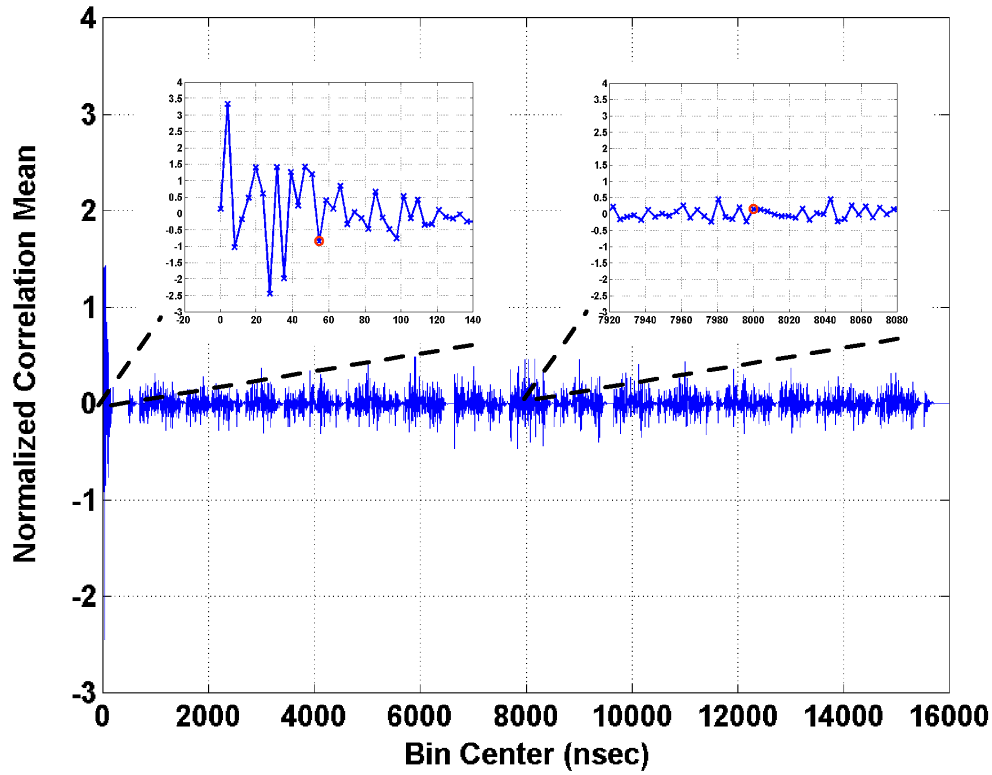

Figure 5.2: Normalized correlator mean for a code length of $N_c = 16$, $T_f = 1000$ nsec, $N = 256$, and $T_c = 10$ nsec. The two regions shown represent coarse acquisition termination points both inside and outside the multipath cluster.

$$= \quad Q\Big(\sqrt{\frac{E_p}{N_c N_0}} \cdot (\Upsilon_F + \frac{s_j}{\sqrt{E_p}})\Big) + Q\Big(\sqrt{\frac{E_p}{N_c N_0}} \cdot (\Upsilon_F - \frac{s_j}{\sqrt{E_p}})\Big) \qquad (5.5)$$

As in Section 4.3, the correlator mean, s_j, used in these probabilities is computed linearly across the uncertainty region, i.e., $j \cdot T_f/N = k_{\beta,j}T_c + \beta_{r,j}$ where $k_{\beta,j}$ and $\beta_{r,j}$ are used in (4.22) which is subsequently used to produce the correlator means, s_j, as per (4.23).

An example is now given which contrasts the verification probability for a detection bin within the multipath cluster with a false alarm bin outside of the multipath

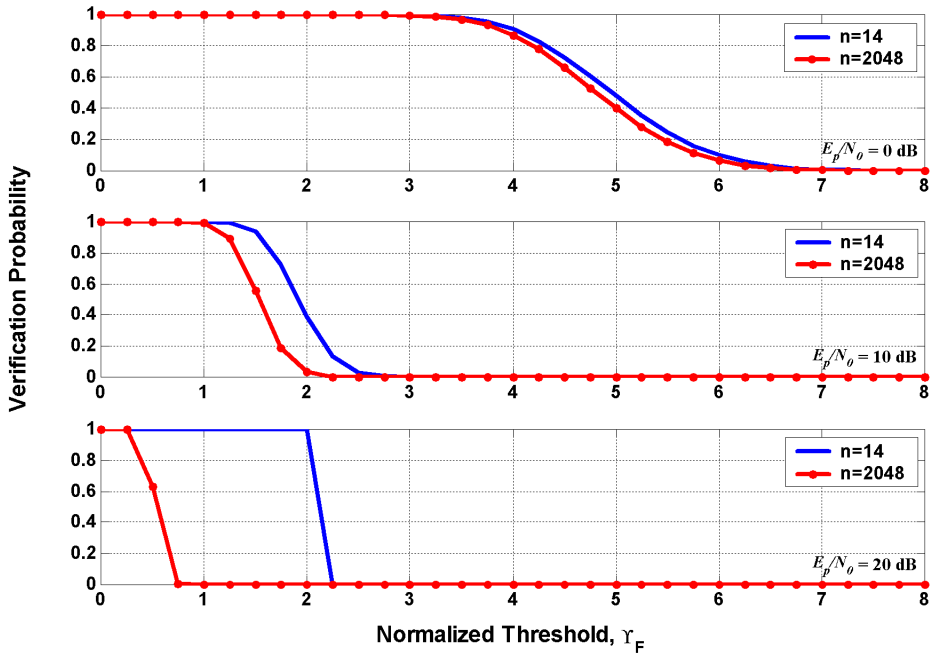

Figure 5.3: Verification probability versus normalized threshold for a code length of $N_c = 16$, $M = 8$ correlators, $k = 2$, and $E_p/N_0 = 0$ dB, 10 dB, and 20 dB.

cluster. Figure 5.2 shows the two regions of interest assuming coarse acquisition termination points of $14 \cdot T_f/N = 54.6875$ nsec and $2048 \cdot T_f/N = 8000$ nsec as highlighted in the figure. Assuming that there are $M = 8$ correlators available and that $k = 2$ sets the uncertainty region size for fine acquisition to $2kM + 1 = 33$ bins. The verification probability is the probability that, after examining all of these bins, the largest $M = 8$ observations exceed the normalized threshold, Υ_F. Figure 5.3 shows this verification probability as a function of the normalized threshold, Υ_F, at the two points $\varepsilon(n) = 14$ and $\varepsilon(n) = 2048$ for signal-to-noise ratios of $E_p/N_0 = 0$ dB, 10 dB, and 20 dB.

As can be seen from Figure 5.3, at a low signal-to-noise ratio the verification probabilities for the two locations ($n = 14$ and $n = 2048$) are very similar for all thresholds. Recall that the first location represents a bin in the middle of the multipath cluster while the second represents a bin far from the cluster. This indicates that verification will be poor at low signal-to-noise ratios for the particular verification criterion under consideration. As the signal-to-noise ratio increases, however, the distinction between the two locations becomes more apparent. As can be seen at the high signal-to-noise ratio of 20 dB, there is a wide range of thresholds around $\Upsilon_F = 1.5$ that provide high acceptance probabilities for the 'in-cluster' bin and low acceptance (high rejection) probabilities for the 'out-of-cluster' bin. The overall probability of correctly acquiring, as well as the mean acquisition time, both depend on the verification criterion's ability to distinguish these two types of bins. This distinction can be improved by increasing the 'post-correlation' signal-to-noise ratio by directly increasing E_p/N_0, by increasing the code length, N_c, or by increasing the dwell-time during verification above a single code period.

5.2 Mean Acquisition Time

Now that the verification probabilities have been computed, the mean acquisition time can be determined. Recall that here we examine the combination of stopping criterion 1 with verification criterion 1. The signal flow graph of Figure 3.6 represents this situation and can be used to determine the mean acquisition time. The path gains for this graph are given in (3.11). These gains are now modified slightly as per Section 4.3 to yield the hybrid search performance. These modified path gains are:

$$H_{\varepsilon(n)}(z) = \begin{cases} P_{\varepsilon(n)\ominus k_1} \cdot P_{\mathrm{v},\varepsilon(n)\ominus k_1} \cdot z^{2k+1} & \text{if } n \in \mathcal{I}(k_1) \\ 0 & \text{else} \end{cases}$$

$$G_{\varepsilon(n)}(z) = \begin{cases} 1 - P_{\varepsilon(n)\ominus k_1} + P_{\varepsilon(n)\ominus k_1} \cdot (1 - P_{\mathrm{v},\varepsilon(n)\ominus k_1}) \cdot z^{2k} & \text{if } n \in \mathcal{A}_1(k_1) \\ (1 - P_{\varepsilon(n)\ominus k_1}) \cdot z + P_{\varepsilon(n)\ominus k_1} \cdot (1 - P_{\mathrm{v},\varepsilon(n)\ominus k_1}) \cdot z^{2k+1} & \text{if } n \in \mathcal{A}_2(k_1) \\ P_{\varepsilon(n)\ominus k_1} \left[1 - P_{\mathrm{v},\varepsilon(n)\ominus k_1}(1 - z^J)\right] z^{2k} + 1 - P_{\varepsilon(n)\ominus k_1} & \text{if } n \in \mathcal{A}_3(k_1) \\ P_{\varepsilon(n)\ominus k_1} \left[1 - P_{\mathrm{v},\varepsilon(n)\ominus k_1}(1 - z^J)\right] z^{2k+1} + (1 - P_{\varepsilon(n)\ominus k_1})z & \text{if } n \in \mathcal{A}_4(k_1) \end{cases}$$

The conditions indicated by the sets $\mathrm{A}_i(k_1)$ correspond to the same conditions in (3.11), namely $\mathcal{A}_1(k_1) = \mathcal{I}(k_1) \cap \mathcal{B}^{\mathrm{c}}$, $\mathcal{A}_2(k_1) = \mathcal{I}(k_1) \cap \mathcal{B}$, $\mathcal{A}_3(k_1) = \mathcal{I}(k_1)^{\mathrm{c}} \cap \mathcal{B}^{\mathrm{c}}$, and $\mathcal{A}_4(k_1) = \mathcal{I}(k_1)^{\mathrm{c}} \cap \mathcal{B}$. Now the mean acquisition time can be computed using (4.29), which essentially averages the mean acquisition time over the uniform direct path arrival time. The verification probabilities to be used in these path gains are given in (5.2). Figure 5.4 shows the mean acquisition time for an $M = 8$ hybrid bit reversal search, a code length of $N_c = 16$, $k = 2$, and $J = 996$. The $M = 8$ hybrid bit reversal results of Figure 4.17 are also shown in the figure for comparison.

The detection thresholds, Υ and Υ_F are optimized for minimum mean acquisition time at each signal-to-noise ratio independently in Figure 5.4. The optimum value of Υ was found to be approximately equal to the optimum value used for the $M = 8$

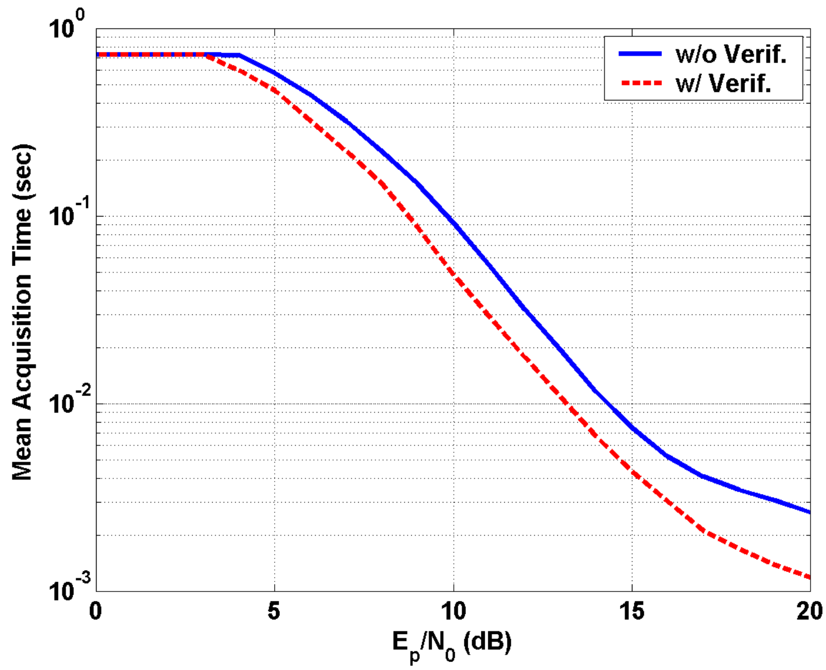

Figure 5.4: Mean acquisition time with and without verification for $M = 8$ correlators using a hybrid bit reversal search, $N_c = 16$, $T_f = 1000$ nsec, $T_c = 10$ nsec, $N = 256$, $k = 2$, $J = 996$, and stopping criterion 1 with verification criterion 1. The detection thresholds, Υ and Υ_F, are optimized for minimum mean acquisition time.

hybrid bit reversal search of Section 4.3. As expected, a certain amount of insight into the optimum value of Υ_F can be gained from the verification probabilities for 'in-cluster' versus 'out-of-cluster' bins as in Figure 5.3. That is, Υ_F should be chosen so that the verification probability is large for 'in-cluster' bins and small otherwise. For example, the optimum value of Υ_F was determined to be roughly 0.75 at $E_p/N_0 = 20$ dB which, from Figure 5.3, makes sense intuitively. For $E_p/N_0 = 10$ dB, it is not as obvious from Figure 5.3 what Υ_F should be, exactly, although the region near $\Upsilon_F = 2$ seems likely. Figure 5.5 shows the mean acquisition time versus Υ_F for a number of different coarse acquisition detection thresholds, Υ. As can be seen, the optimum threshold is indeed $\Upsilon_F = 2$.

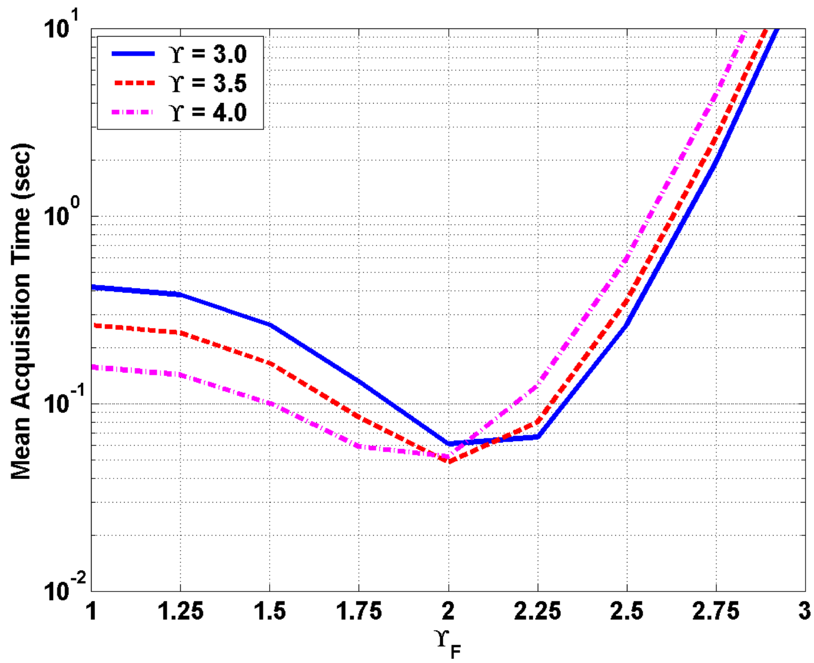

Figure 5.5: Mean acquisition time for $M = 8$ correlators using a hybrid bit reversal search, $E_p/N_0 = 10$ dB, $N_c = 16$, $T_f = 1000$ nsec, $T_c = 10$ nsec, $N = 256$, $k = 2$, $J = 996$, and stopping criterion 1 with verification criterion 1.

For $E_p/N_0 = 0$ dB, the distinction between a proper detection bin and every other bin is not as clear due to the large amount of observation noise. This is evident by the verification probability shown in Figure 5.3. At this low signal-to-noise ratio, however, insight into the optimum detection threshold can be gained from Section 3.2.1, specifically Figure 3.8. As was discussed in that section, the mean acquisition time is minimized when the verification probability is unity as the signal-to-noise ratio decreases toward zero ($-\infty$ dB). The optimum value of Υ_F was found to be 1.5, and the mean acquisition time for a number of detection thresholds is shown in Figure 5.6. At this point, the verification probability for both types of bins is indeed approaching unity as evident in Figure 5.3.

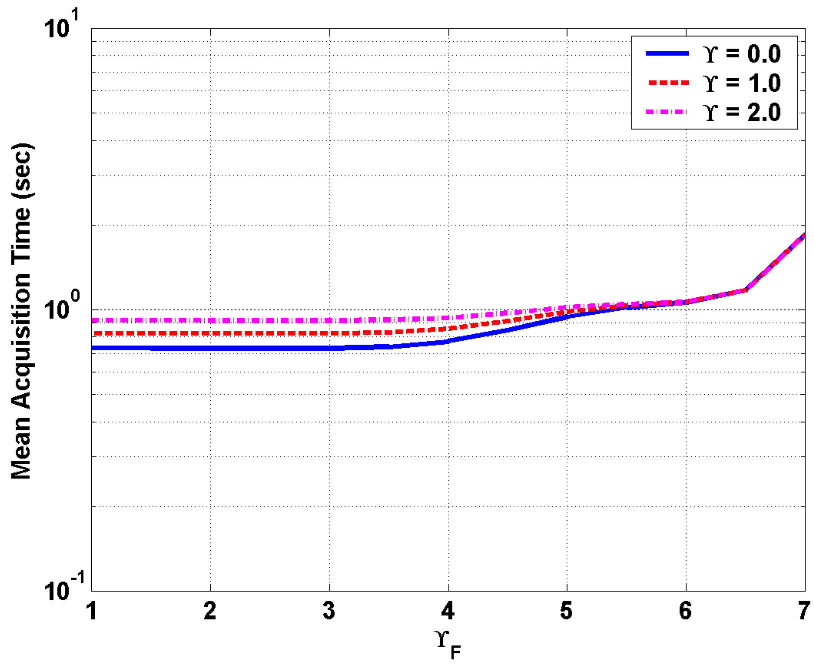

Figure 5.6: Mean acquisition time for $M = 8$ correlators using a hybrid bit reversal search, $E_p/N_0 = 0$ dB, $N_c = 16$, $T_f = 1000$ nsec, $T_c = 10$ nsec, $N = 256$, $k = 2$, $J = 996$, and stopping criterion 1 with verification criterion 1.

5.3 Acquisition Probability

Upon completion of the verification process, one of two possibilities exists. The hypothesis in question, namely the bin at which coarse acquisition was terminated, is either accepted as a proper terminating bin or it is not. Up until now a final level of verification has been assumed as evident by the false alarm penalty time, J, so that the probability of correctly acquiring was always unity. Here, this last stage of verification is removed so that the acquisition probability can be examined assuming only the verification criterion studied in the previous section. Analogous to (3.18) and (3.19), the hybrid path gains for this scenario are:

$$H_{\varepsilon(n)}(z) = \begin{cases} P_{\varepsilon(n)\ominus k_1} \cdot P_{\mathrm{v},\varepsilon(n)\ominus k_1} \cdot z^{2k+1} & \text{if } n \in \mathcal{I}(k_1) \\ 0 & \text{else} \end{cases}$$

$$G_{\varepsilon(n)}(z) = \begin{cases} 1 - P_{\varepsilon(n)\ominus k_1} + P_{\varepsilon(n)\ominus k_1} \cdot (1 - P_{\mathrm{v},\varepsilon(n)\ominus k_1}) \cdot z^{2k} & \text{if } n \in \mathcal{A}_1(k_1) \\ (1 - P_{\varepsilon(n)\ominus k_1}) \cdot z + P_{\varepsilon(n)\ominus k_1} \cdot (1 - P_{\mathrm{v},\varepsilon(n)\ominus k_1}) \cdot z^{2k+1} & \text{if } n \in \mathcal{A}_2(k_1) \\ P_{\varepsilon(n)\ominus k_1} \left[1 - P_{\mathrm{v},\varepsilon(n)\ominus k_1}\right] z^{2k} + 1 - P_{\varepsilon(n)\ominus k_1} & \text{if } n \in \mathcal{A}_3(k_1) \\ P_{\varepsilon(n)\ominus k_1} \left[1 - P_{\mathrm{v},\varepsilon(n)\ominus k_1}\right] z^{2k+1} + (1 - P_{\varepsilon(n)\ominus k_1}) z & \text{if } n \in \mathcal{A}_4(k_1) \end{cases}$$

The probability of correctly terminating this hybrid search is $P_c = P_{ACQ}(1)$. As in (3.20), due to the the uniform nature of the the first detection bin, this acquisition probability is:

$$P_c = \frac{1}{N_s} \sum_{k_1=0}^{N_s-1} P_{ACQ}(1|k_1) \tag{5.6}$$

Some results are now shown for a hybrid bit reversal search with $M = 8$ correlators, a code length of $N_c = 16$, and the signal-to-noise ratio of $E_p/N_0 = 10$ dB. Figures 5.7 and 5.8 show the mean acquisition time and acquisition probability, respectively, as a function of the verification threshold, Υ_F, for the coarse acquisition detection threshold, Υ, varying between 0 and 10. The search operating characteristic curve is obtained by plotting the acquisition probability versus the mean acquisition time, as shown in Figure 5.9.

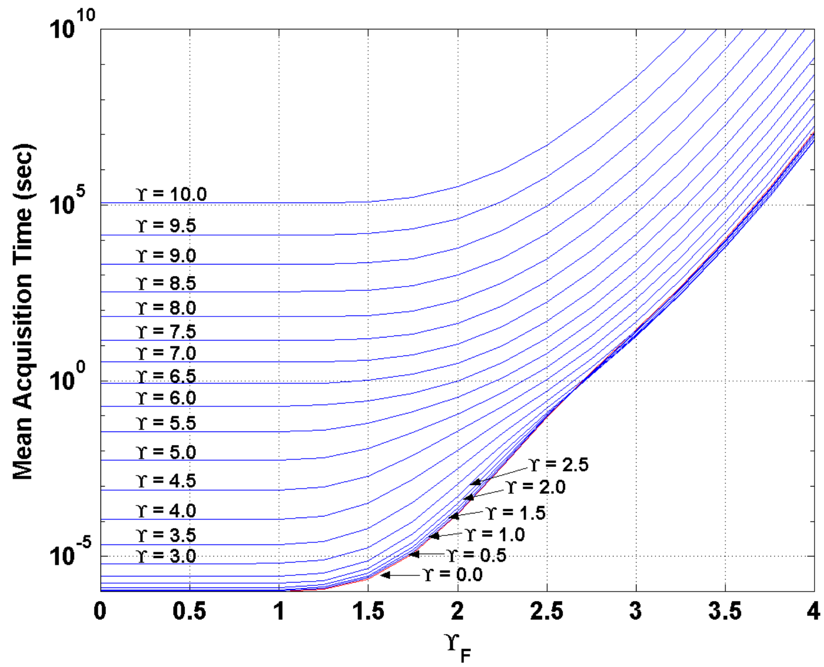

Figure 5.7: Mean acquisition time for an $M = 8$ hybrid bit reversal search, $E_p/N_0 =$ 10 dB, $N_c = 16$, $T_f = 1000$ nsec, $T_c = 10$ nsec, $N = 256$, and $k = 2$

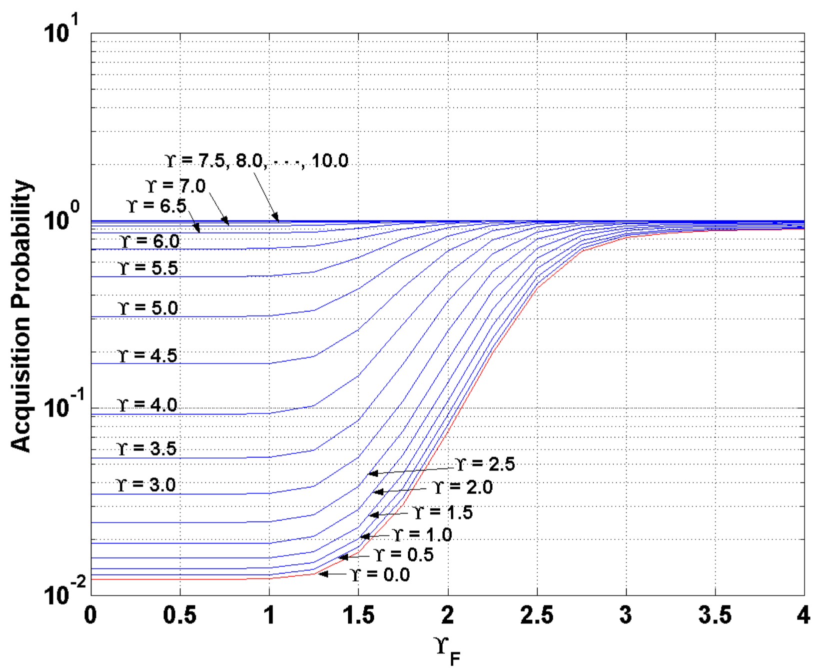

Figure 5.8: Acquisition probability for an $M = 8$ hybrid bit reversal search, $E_p/N_0 =$ 10 dB, $N_c = 16$, $T_f = 1000$ nsec, $T_c = 10$ nsec, $N = 256$, and $k = 2$

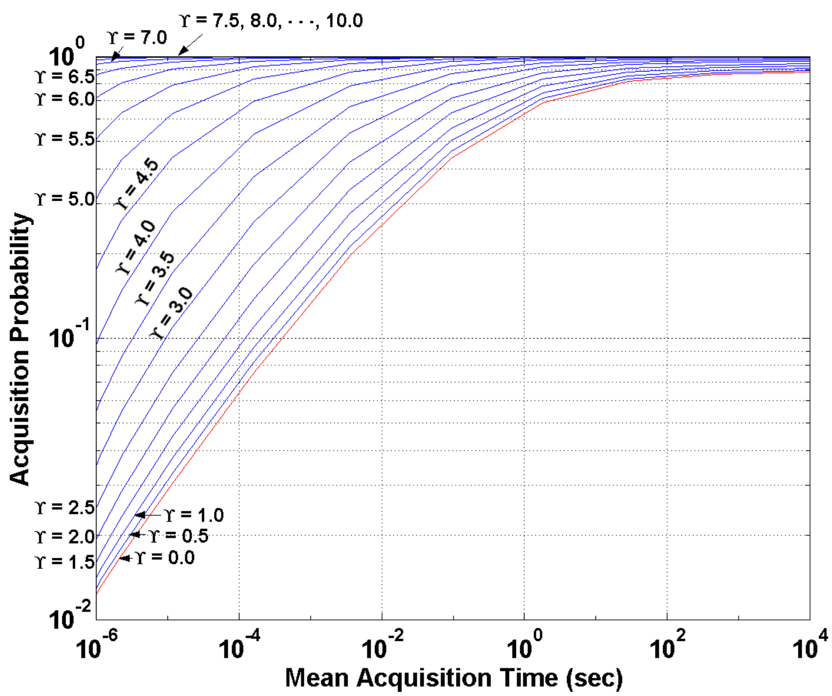

Figure 5.9: Search operating characteristic for an $M = 8$ hybrid bit reversal search, $E_p/N_0 = 10$ dB, $N_c = 16$, $T_f = 1000$ nsec, $T_c = 10$ nsec, $N = 256$, and $k = 2$

Chapter 6

Summary and Future Work

In any communication system, synchronization is of fundamental importance since without it, information cannot be reliably exchanged. Wireless communication systems designed to work indoors or in urban environments have the difficult task of synchronizing in the presence of dense multipath. For the ultra-wideband (UWB) signals discussed here, the indoor environment can induce a long delay spread, but typically these paths are resolvable. This fact actually aids the acquisition process since any number of paths can be located.

The general UWB acquisition process employed here was divided into two stages, *coarse and fine acquisition*. Because of the clustering phenomenon of the various paths in the multipath channel, the coarse acquisition process focused first on finding this cluster of paths. The search analysis techniques studied for this process included a Markov chain analysis, as well as a novel *generalized signal flow graph* analysis approach. This latter produces a complete statistical description of the search time, as well as the overall probability of correctly terminating the search. This generalized signal flow graph approach also allowed the hybrid serial/parallel search to be examined using the same mathematical framework. Additionally, arbitrary search patterns, as well as arbitrary detection scenarios, could also be examined with this new technique.

Once the cluster had been located, the fine acquisition process then attempted to locate the strongest paths within the multipath cluster, and in doing so combined the verification and channel estimation stages into a single process. The combined analysis of coarse and fine acquisition led to the idea of a *self-similar signal flow graph*. This new graphical structure allows for the analysis of a broad range of search problems. One specific application of this self-similar signal flow graph is in the area of sequential detection, producing a new analysis technique useful for studying an old problem. The number of problems suitable for analysis using these novel graphical structures and techniques is vast, extending well beyond just the acquisition of UWB or wideband-CDMA signals. Within the current framework, there are many new possibilities left unexplored. Thoughts on such future work are now given.

The frame clock at the receiver has been assumed throughout this work to run at the same rate as the frame clock in the transmitter. Clock instabilities, such as jitter or drift, and relative motion between the two radios will affect the acquisition and tracking processes. Future work should investigate the effect of the clocks running at different rates.

Throughout it has been assumed that there is no associated penalty time incurred when an observer transitions between search bins, e.g., state $\varepsilon(n)$ followed by $\varepsilon(n+1)$. This has been assumed since all that changes is a code sequence for a local template waveform, and one can basically switch to any other code sequence in the same amount of time. In other situations this may not be the case. That is, the amount of time required for an observer to switch from $\varepsilon(n_1)$ to $\varepsilon(n_2)$ may be depend on n_1 and n_2 and will directly affect the optimum selection of the search order, $\varepsilon(n)$. One example would be the scanning of a mirror in a free space optical communications system that is trying to acquire. There is a finite amount of time required to transition between search bins set by the amount of time required to steer the mirror. Incorporation of these penalty times between observations is straightforward in the current framework.

During coarse acquisition, if detection occurs then all of the available resources are diverted to fine acquisition, i.e., all the correlators. This allows the signal flow graph approach to be used for analysis purposes. The optimal resource allocation problem has not been examined, that is, how many correlators should be diverted to fine acquisition and how many should be left in the coarse search mode? Additionally, what if a detection occurs at the coarse search level while another bin is being verified during a fine search? Resources would again need to be diverted for verification purposes, so that the number of available resources fluctuates as a function of time. In addition, the problem can be constrained such that only those observers in coarse search mode which are 'close enough' to the detection point of one of the observers are diverted to search the region near that particular observer. This constraint makes a certain amount sense if a cost is associated with traveling to the new location (such as some amount of time) as just mentioned above.

At the start of the acquisition process, no a-priori knowledge is assumed of the channel, except possibly some knowledge of the delay spread as required by the optimum 'look and jump' search. Assuming prior knowledge of the channel (or a reasonable estimate), we can space the M correlators relative to one another based upon the spacing between the largest paths in the cluster. We could then hop the entire group around the search space in unison instead of letting each correlator search based upon its own pattern. Would this method perform better than the coarse/fine acquisition procedure described earlier? Obviously there would be a cost associated with doing this since the channel estimates must first be obtained, but the performance benefit may outweigh this cost.

Once the fine acquisition process is complete, the path estimates are passed to a tracking processor where each path is tracked. Methods which share information amongst the various path trackers have been shown to provide performance improvement relative to the situation involving no information sharing. Following fine acquisition, if all trackers obtain lock then acquisition is complete. If some number of them don't obtain lock then the available correlators are sent back to perform the fine acquisition process once again. If none of them obtain lock then either all the correlators perform fine acquisition in the immediate vicinity or coarse acquisition is reinitiated.

Finally, some thoughts are given on analyzing the tracking process using the graphical structures introduced in this work. Specifically, consider a self-similar signal flow graph with at least 3 rings. The last ring, which is completely circular, is simply a ring without an acquisition state in the middle. Instead a 'loss of lock' state is present. This state leads to the ring above the current one, back to precisely the state where tracking was first entered (or the state next to this state). Once fine acquisition has completed the innermost ring is entered. Each frame time, the states in this innermost circle are stepped through one at a time. At any point in time the state in the center can be entered which leads back to fine acquisition. The probability of entering this center state is the 'loss of lock' probability.

Reference List

[1] A. Alvarez, G. Valera, M. Lobeira, R. Torres, and J. Garcia. Ultra-wideband channel characterization and modeling. *Proc. International Workshop on Ultra-Wideband Systems*, June 2003.

[2] C. Baum and V. Veeravalli. Hybrid acquisition schemes for direct sequence CDMA systems. *Proc. International Conf. Communications*, pages 1433–1437, May 1994.

[3] C. Baum and V. Veeravalli. Sequential multihypothesis testing and nonuniform costs with applications in hybrid serial search. *Proc. IEEE Symposium on Information Theory*, page 256, June 1994.

[4] C. W. Baum and V. V. Veeravalli. A sequential procedure for multihypothesis testing. *IEEE Trans. Inform. Theory*, 40:1994–2007, November 1994.

[5] C. W. Baum and V. V. Veeravalli. Asymptotic efficiency of a sequential multihypothesis test. *IEEE Trans. Inform. Theory*, 41:1994–1997, November 1995.

[6] G. Cao and M. West. Computing distributions of order statistics. *Communications in Statistics*, 26, 1997.

[7] G. Casella and R. Berger. *Statistical Inference*. Duxbury Press, 1990.

[8] J. M. Cramer, R. A. Scholtz, and M. Z. Win. Evaluation of an ultra-wideband propagation channel. *IEEE Trans. Antennas and Prop.*, May 2002.

[9] R. Jean-Marc Cramer. *An Evaluation of Ultra-Wideband Propagation Channels*. PhD thesis, University of Southern California, Los Angeles, CA, December 2000.

[10] H. A. David. *Order Statistics*. Wiley, 2003.

[11] J. M. H. Elmirghani, R. A. Cryan, and F. M. Clayton. On the spectral estimation and synchronization of the cyclostationary optical fibre PPM process. *IEEE Trans. Commun.*, 43(2/3/4):1001–1012, Feb./ Mar./ Apr. 1995.

[12] R. Fleming, C. Kushner, G. Roberts, and U. Nandiwada. Rapid acquisition for ultra-wideband localizers. *Proc. IEEE Conference on Ultra-Wideband Systems and Technologies*, May 2002.

[13] J. Foerster. Channel modeling sub-committee report final. *IEEE 802.15.SG3a Study Group*, December 2002.

[14] R. G. Gallager. *Discrete Stochastic Processes*. Kluwer Academic Publishers, 1996.

[15] W. J. Gill. A comparison of binary delay-lock loop implementations. *IEEE Trans. on Aerospace Electr. Sys.*, 2:415–424, July 1966.

[16] A. R. Golshan. *Low Complexity Iterative Algorithms for Near-Optimal Detection in Like-Signal Interference*. PhD thesis, University of Southern California, Los Angeles, CA, May 2003.

[17] H. P. Hartmann. Analysis of a dithering loop for PN code tracking. *IEEE Trans. on Aerospace Electr. Sys.*, 10:2–9, January 1974.

[18] J. A. Högbom. Aperture synthesis with a non-regular distribution of interferometer baselines. *Astron. Astrophys. Suppl.*, 15:417–426, 1974.

[19] J. K. Holmes. *Coherent Spread Spectrum Systems*. John Wiley & Sons, 1982.

[20] E. Homier. Frequency estimation using a second order digital phase-locked loop. *TRW Space and Electronics Group Interoffice Correspondence*, C80-99-SE-058, August 1999.

[21] E. Homier. Initial phase estimate statistics. *TRW Space and Electronics Group Interoffice Correspondence*, C80-99-SE-041, May 1999.

[22] E. Homier. Timing probe processor acquisition analysis. *TRW Space and Electronics Group Interoffice Correspondence*, C80-99-SE-031, April 1999.

[23] E. Homier. Timing probe processor acquisition analysis with input normalization. *TRW Space and Electronics Group Interoffice Correspondence*, C80-99-SE-053, June 1999.

[24] E. A. Homier and R. A. Scholtz. Rapid acquisition of ultra-wideband signals in the dense multipath channel. *Proc. IEEE Conference on Ultra-Wideband Systems and Technologies*, pages 105–109, May 2002.

[25] J. Iinatti. Mean acquisition time of DS code acquisition in fixed multipath channel. *IEEE 5th International Symposium on Spread Spectrum Techniques and Applications*, 1:116–120, September 1998.

[26] C. Kose and D. Goeckel. Minimum complexity sequential multihypothesis detection: Weak sequential tests. *IEEE Trans. Commun.*, 46:129–133, May 1994.

[27] C. Kose, D. Goeckel, and S. Wei. Minimum complexity sequential multihypothesis detection. *Proc. IEEE Symposium on Information Theory*, 46:18, June 2001.

[28] S. Lee. *Design and Analysis of Ultra-Wide Bandwidth Impulse Radio Receiver.* PhD thesis, University of Southern California, Los Angeles, CA, April 2002.

[29] L. L. Li, L. S. Taylor, and H. Dong. Deeper penetrating waves in lossy media. In *Ultra-Wideband Short-Pulse Electromagnetics*, pages 275–284, Plenum Press, 1993.

[30] W. C. Lindsey and M. K. Simon. *Telecommunication Systems Engineering.* Prentice-Hall, Englewood Cliffs, New Jersey, 1973.

[31] H. Meyr and G. Ascheid. *Synchronization in Digital Communications.* Wiley, 1990.

[32] H. Meyr, M. Moeneclaey, and S. Fechtel. *Digital Communication Receivers.* Wiley, 1998.

[33] OSD/DARPA. *Ultra-Wideband Radar Review Panel, Assessment of Ultra-Wideband (UWB) Technology.* DARPA, Arlington, VA, 1990.

[34] A. Polydoros. *On the Synchronization Aspects of Direct-Sequence Spread Spectrum Systems.* PhD thesis, University of Southern California, Los Angeles, CA, August 1982.

[35] A. Polydoros and C. Weber. A unified approach to serial search spread-spectrum code acquisition - parts I and II. *IEEE Trans. Commun.*, 32(5):542–560, May 1984.

[36] E. C. Posner. Optimal search procedures. *IRE Trans. on Information Theory*, 11:157–160, July 1963.

[37] R. Price and P. E. Green. A communication technique for the multipath channel. *Proceedings of the IRE*, pages 555–570, March 1958.

[38] F. Ramirez-Mireles and R. A. Scholtz. N-orthogonal time-shift-modulated codes for impulse radio. In *IEEE Communication Theory Mini-Conference*, November 1997.

[39] F. Ramirez-Mireles and R. A. Scholtz. Performance of equicorrelated ultra-wideband pulse-position-modulated signals in the indoor wireless impulse radio channel. In *IEEE Pacific Rim Conference*, Vol. 2, pp. 640-644, 1997.

[40] F. Ramirez-Mireles and R. A. Scholtz. Time-shift-keyed equicorrelated signal sets for impulse radio M-ary modulation. In *WIRELESS Conference*, July 1998.

[41] F. Ramirez-Mireles, M. Z. Win, and R. A. Scholtz. Signal selection for the indoor wireless impulse radio channel. In *IEEE Vehicular Technology Conference*, Vol. 3, pp. 2243-2247, 1997.

[42] T. S. Rappaport. *Wireless Communications, Principles and Practice.* Prentice-Hall PTR, 1996.

[43] R. R. Rick and L. B. Milstein. Optimal decision strategies for acquisition of spread-spectrum signals in frequency-selective fading channels. *IEEE Trans. Commun.*, 46:686–694, May 1998.

[44] S. Verdú. Minimum probability of error for asynchronous Gaussian multiple-access channels. *IEEE Trans. Information Theory*, 32:85–96, January 1986.

[45] A. Saleh and R. Valenzuela. A statistical model for indoor multipath propagation. *IEEE J. Select. Areas Commun.*, 5:128–137, February 1987.

[46] R. A. Scholtz. Multiple access with time-hopping impulse modulation. In *Proc. IEEE Military Comm. Conf.*, Boston, MA, October 1993.

[47] R. A. Scholtz, P. V. Kumar, and C. J. Corrada-Bravo. Signal design for ultra-wideband radio. *Sequences and Their Applications (SETA'01)*, May 2001.

[48] Oh-Soon Shin and Kwang Bok Lee. Utilization of multipaths for spread-spectrum code acquisition in frequency-selective rayleigh fading channels. *IEEE Trans. Commun.*, 49:734–743, April 2001.

[49] M. Simon, S. Hinedi, and W. Lindsey. *Digital Communications Techniques.* PTR Prentice Hall, 1995.

[50] Q. H. Spencer, B. D. Jeffs, M. A. Jensen, and A. L. Swindlehurst. Modeling the statistical time and angle of arrival characteristics of an indoor multipath channel. *IEEE J. Select. Areas Commun.*, 18(3):347–360, March 2000.

[51] J. J. Spilker, Jr. Delay-lock tracking of binary signals. *IEEE Trans. on Space Electron. Telemetry*, 9:1–8, March 1963.

[52] J. J. Spilker, Jr. and D. G. Magill. The delay lock discriminator - an optimum tracking device. *Proc. IRE*, 49:1403–1416, 1961.

[53] J. J. Stiffler. *Theory of Synchronous Communications.* Englewood Cliffs, 1971.

[54] G. L. Stüber. *Principles of Mobile Communication.* Kluwer Academic Publishers, 1996.

[55] H. Suzuki. A statistical model for urban radio propagation. *IEEE Trans. Commun.*, 25(7):673–680, July 1977.

[56] H. M. Taylor and S. Karlin. *An Introduction to Stochastic Modeling.* Academic Press, 1984.

[57] J. D. Taylor. *Introduction to Ultra-Wideband Radar Systems.* CRC Press, Boca Raton, FL, 1995.

[58] Lai-King Anna Tee. Time-hopping spread spectrum system. Master's thesis, University of Southern California, 1998.

[59] G. L. Turin, F. D. Clap, T. L. Johnston, S. B. Fine, and D. Lavry. A statistical model of urban multipath propagation. *IEEE Trans. Veh. Tech.*, 21(1):1–9, February 1972.

[60] R. Vaughan and N. Scott. Super-resolution of pulsed multipath channels for delay spread characterization. *IEEE Trans. Commun.*, 47(3):343–347, March 1999.

[61] A. Wald. *Sequential Analysis*. Wiley, 1947.

[62] Huan-Chen Wang and Wern-Ho Sheen. Variable dwell-time code acquisition for direct sequence spread spectrum systems on multipath fading channels. *Proc. International Conf. Communications*, 3:1232–1236, June 1998.

[63] M. Z. Win. *Ultra-Wide Bandwidth Spread-Spectrum Techniques for Wireless Multiple-Access Communications*. PhD thesis, University of Southern California, Los Angeles, CA, May 1998.

[64] M. Z. Win and R. A. Scholtz. Ultra-wide bandwidth signal propagation for indoor wireless communications. In *Proc. International Conf. Communications*, Montreal, Canada, June 1997.

[65] M. Z. Win, R. A. Scholtz, and L. W. Fullerton. Time-hopping SSMA techniques for impulse radio with an analog modulated data subcarrier. In *IEEE ISSSTA*, 1996.

[66] H. Zhang, S. Wei, D. Goeckel, and M. Win. Rapid acquisition of ultra-wideband radio signals. *Proc. Asilomar Conf. Signals, Systems, Comp.*, November 2002.

Appendix A

Mean Search Time for Unity Detection and False Alarm Probabilities

From Section 2.4 the generating function into the acquisition state using the generalized signal flow graph is

$$P_{ACQ}(z) = \frac{\displaystyle\sum_{k=0}^{N_s-1} \pi_{\varepsilon(k)} \sum_{i=0}^{N_s-1} H_{\varepsilon(i\oplus k)}(z) \prod_{j=0}^{i-1} G_{\varepsilon(j\oplus k)}(z)}{\displaystyle 1 - \prod_{i=0}^{N_s-1} G_{\varepsilon(i)}(z)} \tag{A.1}$$

For detection and false alarm probabilities that are the constant across all states, (2.54) and (2.55) yield the following path gains:

$$H_{\varepsilon(i)}(z) = \begin{cases} P_D z & \text{if } i \in \mathcal{I} \\ 0 & \text{else} \end{cases} \tag{A.2}$$

and

$$G_{\varepsilon(i)}(z) = \begin{cases} (1 - P_D)z & \text{if } i \in \mathcal{I} \\ (1 - P_{FA})z + P_{FA}z^{J+1} & \text{else} \end{cases} \tag{A.3}$$

As before, the set $\mathcal{I}$ represents those indices, i_k for $k = 0, \cdots, K-1$, such that $\varepsilon(i_k)$ leads to the acquisition state. Since the detection and false alarm probabilities are assumed constant in (A.3) it is seen that the denominator of (A.1) becomes:

$$1 - \prod_{i=0}^{N_s-1} G_{\varepsilon(i)(z)} = 1 - ((1 - P_D)z)^K \cdot \left((1 - P_{FA})z + P_{FA}z^{J+1}\right)^{N_s-K} \tag{A.4}$$

Here it is assumed that P_D and P_{FA} are both unity which reveals that (A.4) is simply one for $K \geq 1$. Explicitly writing out the numerator of (A.1) in a similar manner for P_D and P_{FA} both equaling one and also assuming that $\pi_n = 1/N_s$ for all n reveals that:

$$P_{ACQ}(z) = \frac{1}{N_s} \cdot \left(z \cdot z^{(J+1)\cdot n_{\varepsilon(0)}} + z \cdot z^{(J+1)\cdot n_{\varepsilon(1)}} + \cdots + z \cdot z^{(J+1)\cdot n_{\varepsilon(N_s-1)}}\right) \tag{A.5}$$

The exponent, $n_{\varepsilon(k)}$, in the above equation is the clockwise distance around the circular state diagram from state $\varepsilon(k)$ to the nearest state, $\varepsilon(i_j)$ for $j = 0, \cdots, K-1$, which leads into the acquisition state. This distance can be written as:

$$n_{\varepsilon(k)} = \min_{l \in \mathcal{I}} d\left(\varepsilon(k), \varepsilon(l)\right) = \min_{0 \leq j < K} d\left(\varepsilon(k), \varepsilon(i_j)\right) \tag{A.6}$$

where $d(x, y)$ is the clockwise distance (in number of state transitions) from state x to state y:

$$d(x, y) = \begin{cases} (y - x) & \text{if } y \geq x \\ N_s + (y - x) & \text{else} \end{cases} \tag{A.7}$$

An example is now given which aids in verifying some of the results of Section 4.2. Namely when the normalized detection threshold, Υ, is set to zero, the transition probabilities in (4.24) become unity. Thus the generating function given in (A.5) is now applicable. If the first K states lead to the acquisition state and the search permutation is linear, $\varepsilon(n) = n$, then $\mathcal{I} = 0, 1, \cdots, K-1$ and $n_0 = n_1 = \cdots = n_{K-1} = 0$ and $n_{N_s-1} = 1$, $n_{N_s-2} = 2$, $\cdots$, $n_{K+1} = N_s - K - 1$, $n_K = N_s - K$. Substituting these distances into (A.5) yields

$$P_{ACQ}(z) = \frac{1}{N_s} \cdot \left(K \cdot z + \sum_{k=1}^{N_s - K} z \cdot z^{(J+1) \cdot k} \right) \tag{A.8}$$

Here we have $\sum_{k=1}^{0}(\cdot) = 0$ which occurs when $K = N_s$. The mean search time is then computed as

$$
\begin{aligned}
E(T_{ACQ}) &= \frac{\mathrm{d}}{\mathrm{d}z} P_{ACQ}(z)\bigg|_{z=1} \\
&= \frac{1}{N_s}\left(K + \sum_{k=1}^{N_s-K} 1 + (J+1)k\right) \\
&= \frac{1}{N_s}\left(N_s + (J+1)\cdot\frac{(N_s-K)(N_s-K+1)}{2}\right) \\
&= 1 + \frac{1}{2}\cdot(J+1)\cdot\left(1-\frac{K}{N_s}\right)\cdot(N_s-K+1) \qquad \text{(A.9)}
\end{aligned}
$$

The mean acquisition time given by the above equations is in units of dwell times, where one dwell time in Section 4.2 is equal to one code period, $N_c \cdot T_f$. For $J = 1000$, $N_s = N_c \cdot N = 16 \cdot 256 = 4096$, $K = 50$, and $T_f = 1000$ nsec we see that (A.9) yields $E(T_{ACQ}) \cdot N_c \cdot T_f = 32.01$ seconds. This agrees with the results of Section 4.2 for a normalized detection threshold, Υ, equal to zero. Similarly, as E_p/N_0 vanishes toward zero this same result is also seen since the transition probabilities for $E_p/N_0 = 0$ are the same as those probabilities for $\Upsilon = 0$. For $J = 1000$, $N_s = N_c \cdot N = 64 \cdot 256 = 16384$, $K = 50$, and $T_f = 1000$ nsec equation (A.9) yields $E(T_{ACQ})\cdot N_c \cdot T_f = 521.65$ seconds, which also agrees with the results of Section 4.2. Several such numerical comparisons are shown in Table A.1.

Use of the moment generating function in (A.5) for a bit reversal search is not as straightforward. However, as was seen in Chapter 2 the bit reversal search and the 'look and jump' resulted in exactly the same mean search time for proper values of K. Even for those values of K that did not produce exact agreement between the two types of searches, the observed difference is quite small. Thus an approximation to the bit reversal search performance can be obtained using the 'look and jump' search. Since here it is assumed that every bin is a-priori equally likely, the 'look and jump' search essentially divides the search space into K regions each containing N_s/K bins (assuming K divides N_s evenly). The mean time spent in any of these K regions is identical and the overall generating function reveals this:

$$P_{ACQ}(z) = \frac{1}{N_s} \cdot K \cdot z \cdot \sum_{k=0}^{N_s/K-1} z^{(J+1)\cdot k} \tag{A.10}$$

The mean acquisition time resulting from this generating function is:

$$E(T_{ACQ}) = 1 + \frac{1}{2} \cdot (J+1) \cdot \left(\frac{N_s}{K} - 1 \right) \tag{A.11}$$

As before this gives the mean search time in integer multiples of dwell-times and must be multiplied by $N_c \cdot T_f$ to give the mean time in seconds. Table A.1 compares the results determined here with the mean-time-to-acquisition (MTA) results from Section 4.2. Also shown is the other extreme, namely the large signal-to-noise ratio scenario. As the signal-to-noise ratio increases, the detection probability increases toward unity and the false alarm probability decreases toward zero (for the proper selection of

	Linear Search			
	$N_c = 16$ $J = 1000$	$N_c = 64$ $J = 1000$	$N_c = 256$ $J = 1000$	$N_c = 256$ $J = 100$
MTA Results from Section 4.2 at Low SNR	32.01 sec	521.65 sec	8384.32 sec	845.97 sec
MTA Results from Appendix A	32.01 sec	521.65 sec	8384.32 sec	845.97 sec
MTA Results from Section 4.2 at High SNR	32 msec	0.52 sec	8.38 sec	8.38 sec
MTA Results from Section 2.2	32 msec	0.52 sec	8.38 sec	8.38 sec
	Bit Reversal Search			
MTA Results from Section 4.2 at Low SNR	0.73 sec	11.76 sec	188.48 sec	19.02 sec
MTA Results from Appendix A	0.65 sec	10.46 sec	167.81 sec	16.93 sec
MTA Results from Section 4.2 at High SNR	3.33 msec	25.4 msec	322.1 msec	251.4 msec
MTA Results from Section 2.2	0.66 msec	10.52 msec	167.9 msec	167.9 msec

Table A.1: Comparison of the mean-time-to-acquisition (MTA) of Section 4.2 with the closed-form asymptotic expressions

detection threshold). Thus the results of Section 2.2, which were computed for $P_D = 1$ and $P_{FA} = 0$, are compared to the mean acquisition time results of Section 4.2. The 'High SNR' results from Section 4.2 shown in Table A.1 were computed at $E_p/N_0 = 20$ dB and $E_p/N_0 = 100$ dB for the linear and bit reversal search, respectively. As can be see from the table, the linear results are in exact agreement while the bit reversal results are in close agreement. Also, from the table it is noted that the ratio of the MTA at a high signal-to-noise ratio to the MTA at a low signal-to-noise ratio is approximately equal to the the false alarm penalty time (in dwell-times).

Appendix B

UWB Time-Hopping Codes for Multiple Users

The process presented here for generating the time-hopping codes used in Section 4.2 is derived from [47]. For a more detailed explanation of the relevant coding theory that particular reference should be consulted as only the procedural steps required to generate time-hopping codes are given here. Only a single code is required for the acquisition analyses given in Chapter 4 even though the procedure explained here generates a family of codes, each with low auto-correlation, as well as low cross-correlation amongst members in the family.

The first step in generating a UWB time-hopping code is to specify the code length N_c, the code chip time time, T_c, the frame time T_f, and the guard time, N_g, which sets the maximum hopping location, $N_g \cdot T_c$, within the frame time. All of these parameters were introduced in Section 1.2. It is assumed that the frame time is an integer number of code chip times, i.e., $T_f = N_f \cdot T_c$ where N_f is a positive integer.

Next, the smallest prime number, p, greater than or equal to N_c is chosen. A set of $p-1$ polynomials is generated as

$$f^{(i)}(x) = i \cdot x^2 \;\; \text{for} \;\; i = 1, 2, \cdots, p-1 \tag{B.1}$$

Each polynomial in the set is now associated with a $p \times p$ matrix $\boldsymbol{A}^{(i)}$ with elements defined by

$$a^{(i)}_{m,n} = \begin{cases} 1 \quad \text{if} \;\; f^{(i)}(n) = m, \;\; 0 \le m \le p-1, \; 0 \le n \le p-1 \\ 0 \qquad\qquad\qquad\qquad\qquad \text{else} \end{cases} \tag{B.2}$$

The matrix $\boldsymbol{A}^{(i)}$ has a single 1 in each of its columns. A sequence, a_n for $n = 0, 1, \cdots, p-1$, is generated which gives the row index between 0 and $p-1$ of this single 1 in each of the columns. The sequence, a_n, is now mapped into a new sequence, c_n, such that $0 \le c_n \le N_g$ via:

$$c_n = \left\lfloor a_n \cdot \frac{N_g}{p-1} \right\rfloor \tag{B.3}$$

This sequence, which is of length p, is truncated to the first N_c elements since p may be larger than N_c. This truncated sequence is the desired UWB time-hopping code.

As an example, consider the length $N_c = 16$ code used in Section 4.2. The system parameters chosen in that section were $T_f = 1000$ nsec, $T_c = 10$ nsec, $N_f = 100$, and

$N_g = 70$. The smallest prime number greater than or equal to 16 is $p = 17$. The matrix, $\boldsymbol{A}^{(3)}$, associated with this value of p is

$$\boldsymbol{A}^{(3)} = \begin{bmatrix} 1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&1&1&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&1&0&0&0&0&1&0&0&0&0&0 \\ 0&0&0&0&0&1&0&0&0&0&0&0&1&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0&0&0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&1&0&0&1&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&1&0&0&0&0&0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \end{bmatrix} \tag{B.4}$$

From this matrix it is seen that a_n=0, 3, 12, 10, 14, 7, 6, 11, 5, 5, 11, 6, 7, 14, 10, 12, 3. From (B.3) the time hopping code is c_n=0, 13, 52, 43, 61, 30, 26, 48, 21, 21, 48, 26, 30, 61, 43, 52. Here the sequence has been truncated so it is a length $N_c = 16$ code. This code was seen in Section 4.2. The other members of this code family can be generated by varying the superscript, i. For the code just given, $i = 3$ was seen from brute force comparison with all other values of $i = 0, \cdots, 16$ to yield the code with the best auto-correlation property.

For completeness, the $N_c = 64$ length code used in Section 4.2 is listed here. This code was generated with the same frame time, chip time, and guard time as above. For this longer code, however, $p = 67$ is the smallest prime greater than or equal to 64. The code selected corresponded to $i = 14$: c_n=0, 14, 59, 62, 24, 15, 37, 16, 26, 65, 63, 20, 6, 22, 67, 1, 35, 27, 49, 30, 41, 10, 9, 38, 25, 42, 18, 23, 58, 51, 4, 57, 68, 39, 39, 68, 57, 4, 51, 58, 23, 18, 42, 25, 38, 9, 10, 41, 30, 49, 27, 35, 1, 67, 22, 6, 20, 63, 65, 26, 16, 37, 15, 24.

Appendix C

Probability Distributions for Non-IID Order Statistics

Let $z_1, z_2, \cdots, z_N$ be a group of independent random variables, each with distribution function $F_j(z) = \Pr(z_j \leq z)$. This collection of random variables is often described as non-IID (where IID means independent, identically distributed) or INID (independent, not identically distributed) in the literature. The order statistics from this group of random variables is obtained by ordering the variables based upon their observed values. The n^{th} order statistic of the group is denoted as $z_{(n)}$ and are all listed below:

$$
\begin{aligned}
z_{(1)} &= \min_n z_n \\
z_{(2)} &= 2^{\text{nd}} \min_n z_n \\
&\vdots \\
z_{(N)} &= \max_n z_n
\end{aligned}
\tag{C.1}
$$

For the IID case, each random variable having distribution function $F_z(z)$, the distribution function for the j^{th} order statistic, denoted as $F_{(j)}(z) = \Pr(z_{(j)} \leq z)$, is found fairly easily for $j = 1, 2, \cdots, N$ to be [7]:

$$F_{(j)}(z) = \sum_{k=j}^{N} \binom{N}{k} F_z^k(z) \left(1 - F_z(z)\right)^{N-k} \tag{C.2}$$

For the non-IID case each random variable has its own distribution function, i.e., z_j has distribution $F_j(z)$. For this case the distribution of the j^{th} order statistic is quite complicated and computationally expensive in its general form [10]:

$$F_{(m)}(z) = \sum_{i=m}^{N} \sum_{S_i} \prod_{k=1}^{i} F_{j_k}(z) \prod_{k=i+1}^{N} \left(1 - F_{j_k}(z)\right) \tag{C.3}$$

The second summation is taken over all permutations S_i such that $j_1 < \cdots < j_i$ and $j_{i+1} < \cdots < j_N$. A simpler method is available which computes the distribution function of the order statistics recursively [6]. This method is much more efficient from a computational standpoint. For $n = 1, 2, \cdots, N-1$:

$$F_{(n)}(z) = F_{(n+1)}(z) + H_n^*(z) F_{(N)}(z) \tag{C.4}$$

where

$$F_{(N)}(z) = \prod_{j=1}^{N} F_j(z) \tag{C.5}$$

$$H_n^*(z) = \frac{1}{N-n} \cdot \sum_{i=1}^{N-n} (-1)^{i+1} L_{-i}(z) H_{n+i}^*(z) \;\text{ for }\; 1 \leq n \leq N-1 \tag{C.6}$$

$$
\begin{aligned}
H_N^*(z) &= 1 && \text{(C.7)} \\
L_{-i}(z) &= \sum_{j=1}^{N} \left(\frac{F_j(z)}{1 - F_j(z)} \right)^{-i} && \text{(C.8)}
\end{aligned}
$$

This recursion works most efficiently for computing the order statistics at the upper end, i.e., near $n = N$, although it is strictly valid for all n. A similar recursion is seen in [6] which is more efficient for computing the distribution of the order statistics near the lower end.

The order statistic $z_{(N-M+1)}$ will be useful in Chapter 5 for computing the verification probability. The distribution function for this particular order statistic from (C.4) is:

$$
\begin{aligned}
F_{(N-M+1)}(z) &= F_{(N-M+2)}(z) + H_{N-M+1}^*(z) F_{(N)}(z) \\
&= F_{(N-M+3)}(z) + H_{N-M+2}^*(z) F_{(N)}(z) + H_{N-M+1}^*(z) F_{(N)}(z) \\
&\vdots \\
&= \left(H_{N-M+1}^*(z) + H_{N-M+2}^*(z) + \cdots + H_N^*(z) \right) F_{(N)}(z) \\
&= \sum_{n=1}^{M} H_{N-n+1}^*(z) \cdot \prod_{i=1}^{N} F_i(z) && \text{(C.9)}
\end{aligned}
$$

Appendix D

Moment Generating Function for a General Two-Ring Self-Similar Signal Flow Graph

The verification procedures studied in Section 3.2.2 assumed a specific search permutation of the fine acquisition uncertainty region. The coarse acquisition termination point, here denoted $\varepsilon_1(n)$, is the starting point for the fine acquisition process. Thus the permutation on the inner ring of the self-similar signal flow graph, denoted $\varepsilon_2(m)$, will be a function of the state $\varepsilon_1(n)$. The self-similar signal flow graph studied here is shown in Figure D.1. The dependence of $\varepsilon_2(m)$ on $\varepsilon_1(n)$ has been suppressed in this figure. This graph represents a generalization of the verification processes studied earlier. The moment generating function of this graph is found to be:

$$P_{ACQ}(z) = \frac{\displaystyle\sum_{k=0}^{N_1-1} \pi_{\varepsilon_1(k)} \sum_{i=0}^{N_1-1} H_{\varepsilon_1(i\oplus k)}(z) \prod_{j=0}^{i-1} G_{\varepsilon_1(j\oplus k)}(z)}{\displaystyle 1 - \prod_{i=0}^{N_1-1} G_{\varepsilon_1(i)}(z)} \tag{D.1}$$

The operator $\oplus$ represents modulo N_1 integer addition and $\prod_{j=0}^{-1}(\cdot)$ is defined to be unity. The path gains for this generating function are:

$$H_{\varepsilon_1(n)}(z) = \tilde{H}_{\varepsilon_1(n)}(z) \cdot \sum_{m=0}^{N_2-1} \tilde{F}_{\varepsilon_2(m,\varepsilon_1(n))}(z) \prod_{k=0}^{m-1} \tilde{D}_{\varepsilon_2(k,\varepsilon_1(n))}(z) \tag{D.2}$$

$$G_{\varepsilon_1(n)}(z) = \tilde{G}_{\varepsilon_1(n)}(z) + \tilde{H}_{\varepsilon_1(n)}(z) \cdot \sum_{m=0}^{N_2-1} \tilde{E}_{\varepsilon_2(m,\varepsilon_1(n))}(z) \prod_{k=0}^{m-1} \tilde{D}_{\varepsilon_2(k,\varepsilon_1(n))}(z) \tag{D.3}$$

Here the dependence of the inner permutation on the outer permutation has been shown as $\varepsilon_2(k, \varepsilon_1(n))$.

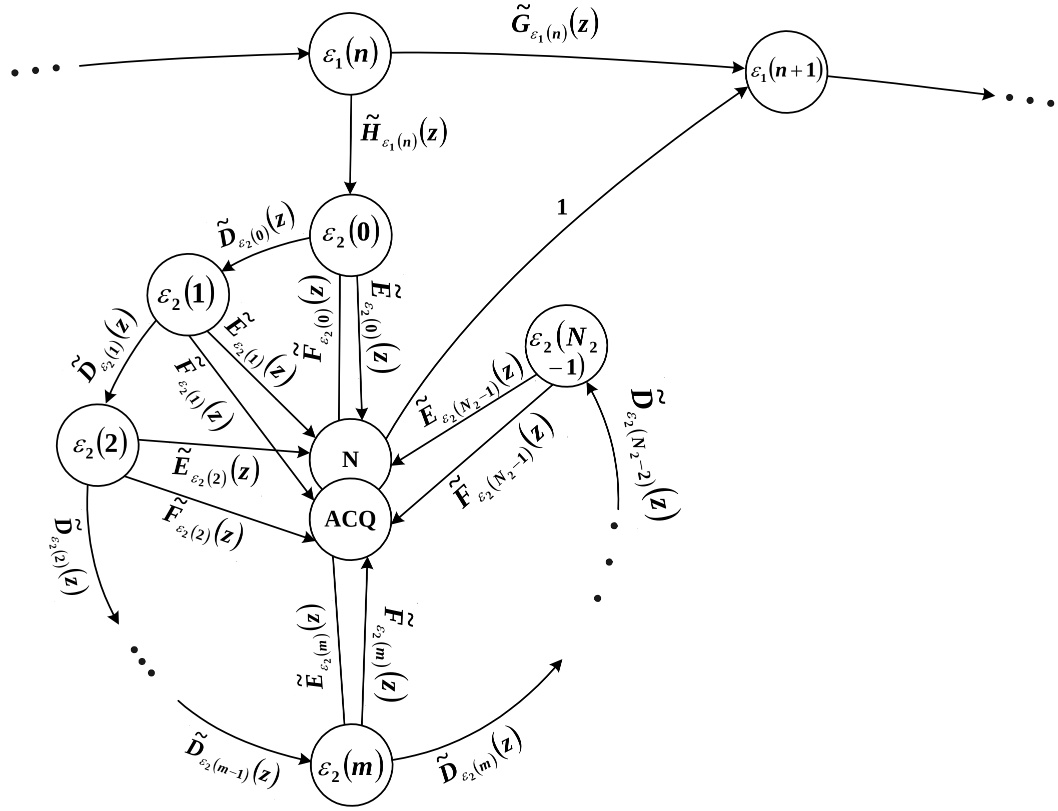

Figure D.1: A general two-ring self-similar signal flow graph

.