Chip Timing Recovery for Indoor Wireless Networks Employing Commutation Signalling

Paul Popescu

Department of Electrical Engineering

McGill University, Montreal

Quebec, Canada

July, 1996

A project submitted to the Department of Electrical Engineering in partial fulfillment of the requirements for the degree of Master of Engineering

Abstract

This project considers chip timing recovery for indoor wireless networks employing Commutation Signalling. First a general analysis is made, based on the *Maximum A Posteriori* Probability (MAP) concept, which leads to a synchronizer structure. Two cases are examined, one using a raised-cosine, the other using a half-sine as the chip pulse. It is shown that, once acquisition has taken place and therefore the difference between the incoming waveform and the locally generated clock is of no more than a half chip duration, the synchronizer will lock onto and track the phase of the incoming waveform.

The various blocks are implemented and the system is simulated using Simulink®, a system level simulator which is part of Matlab®.

Contents

List of Figures

1 Introduction

The word *synchronous* comes from the Greek *sunkhronos* (*sun*, with, and *khronos*, time), meaning two actions happening at the same time. Therefore, to synchronize means to cause two events to occur at the same time or to proceed at the same rate.

Synchronization is a very important aspect of a digital communication system. In a digital communication system, a set of symbols of predetermined duration and shape, representing the message to be conveyed, are transmitted from one point to another. In the process of transmission, the shape of the symbols may be altered, there may be various additive disturbances, or the time base may be unknown. The receiver has the task of processing these distorted symbols in such a way as to detect (recover) the original set of symbols.

Significant time and effort has been spent to investigate the problems related to optimal detection of the symbols; there is a great deal of literature studying the effects of fading, multipath and intersymbol interference in the context of white Gaussian additive noise. A common assumption of literature in this area is that the receiver has a proper time base (it knows the epoch and duration of the chips or symbols). But, in order for this to be so, synchronization must have been achieved.

Often, the information in a digital system appears serially in a time multiplexed form. The information can be encoded in the form of pulses representing chips, which are grouped to form bits; the bits can further be grouped to form words, which can in turn be grouped to form frames, etc.

For the receiver to properly process the received information, it is necessary that it be able to correctly identify the beginning and duration of each pulse, bit, word and frame. The process of identification is called timing synchronization of the receiver.

A type of channel that received — and continues to receive — increased attention due to its practical importance, is the wireless (multipath fading) channel. Unlike the wireline channel, the wireless channel is characterized by the existence of several propagation paths from the transmitter to the receiver: the transmitted wave is reflected by the buildings, trees, hills in the outdoor environment; or by walls, floors, ceilings, furniture in the indoor environment. The waves arrive at the receiver on different paths of unequal lengths and therefore with different delays and random phase shifts. This phenomenon is called "multipath", and the superposition of the waves reaching the receiver leads to either constructive or destructive addition, depending upon the relative phase shifts between the waves. Where the addition is destructive, the resulting signal can be very small, and this is called "fading".

Due to the unequal delays of propagation, replicas of one pulse may very well reach the receiver after the following pulse arrived at its destination, or at the same time, making it very hard to reliably determine whether this represents a new pulse, or a replica of a previous one, or even to distinguish them: this is called "intersymbol interference" (ISI). The time interval after which there are no more replicas of a transmitted pulse on the channel (ringing) is called "multipath delay spread"; in the outdoor environment it can be of the order of several tens of microseconds, while in the indoor environment it can be of the order of hundreds of nanoseconds. The main obstacle to achieving high transmission rates on such a channel is the intersymbol interference induced by the multipath phenomenon. When the multipath delay spread of the channel is larger than the duration of a channel symbol, ISI results. In the presence of ISI, the task of decoding the channel output symbol by symbol is difficult. For such a decoding strategy to work, the channel must be made to appear as if the transmission of each symbol occurs in isolation from all previous and successive symbols.

Among the approaches that have been considered for the purpose of overcoming ISI are adaptive equalization techniques and commutation signalling. Adaptive equalization is a subject widely treated in the literature and will not be discussed in this project. Commutation signalling was introduced by Turin in [6]; it is the approach examined in this project.

Commutation signalling is a bandwidth expanding modulation which is robust to multipath induced ISI, and makes possible the exploitation of time diversity inherent in this propagation medium. An $N \times M$ commutation signalling system uses M signal sets, each with N waveforms. Its functionality can be explained with the help of Figure 1: let

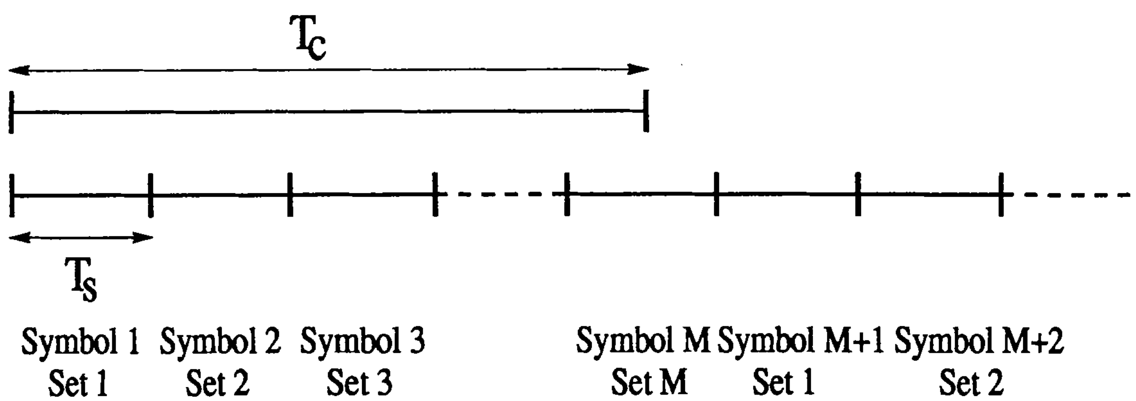

Figure 1: The Commutation Signalling Process

the channel multipath delay spread be T_c and the symbol duration, T_s, such that $M =$

$\left\lceil \frac{T_c}{T_s} \right\rceil$, where $\lceil(\cdot)\rceil$ denotes the nearest integer larger than $(\cdot)$; the bit rate of the data source is R, and N is chosen large enough to be able to support this bit rate. Provided that the M signal sets have very small inter-set partial correlations, a communication strategy which allows for the separation of successive symbols over the channel can be implemented: the source bit stream is divided into several lower rate bit streams, modulated using waveforms from different signal sets in a round-robin fashion — ideally those would be mutually orthogonal waveforms, uncorrelated under any time shift — and then is recombined, converted to passband and sent over the channel.

The $2 \times M$ commutation signalling scheme is optimal in the sense that it uses the minimum number of waveforms (actually, the number of signals used can be further halved by using bipolar modulation such as Binary Phase Shift Keying, BPSK, or Differential Binary Phase Shift Keying, DBPSK). To illustrate this optimality, let us consider two signalling schemes operating at the same bit rate, R: $2 \times M$ bipolar commutation signalling (e.g. DBPSK) and M-ary signalling (e.g. M-FSK). In the first case, since $R = \frac{1}{T_{s_1}}$, where T_{s_1} is the signal duration, it follows that the number of sets, M_1, necessary to transmit without ISI, is $M_1 = \left\lceil \frac{T_c}{T_{s_1}} \right\rceil = \lceil RT_c \rceil$; in the second case, again to avoid ISI due to channel ringing, $T_{s_2} = T_c$, where T_{s_2} is the signal duration; if the number of waveforms is now M_2, the bit rate is $R = \frac{\log_2(M_2)}{T_c}$. Therefore, in order to achieve the same bit rate R, M_2 must be equal to 2^{M_1}, i.e. we need now 2^{M_1} signals.

An advantage of $2 \times M$ commutation signalling using DBPSK is the fact that it does not require a phase reference, hence allowing for differentially coherent detection, which offers a 3dB signal-to-noise ration (SNR) advantage, compared to orthogonal M-FSK.

Indoor wireless systems have been proposed to operate at data rates of the order of 40Mb/s/channel. The receiver, and specifically the synchronizer, must be implementable in a combination of high-speed analog and high-speed digital circuitry for such a data rate (and an even higher chip rate). This means that the synchronizer structure should be chosen to be compatible with the IC technology.

The problem of synchronization at different levels (chip, bit, word, frame, etc.) has been studied in literature: Franks [12] and [2] studied a pulse amplitude modulation (PAM) timing recovery method based on the Maximum Likelihood (ML) criterion, which leads to a likelihood function to be maximized. Meyers and Franks [13] studied the joint carrier phase and symbol timing recovery for PAM systems. In [9], Pickholtz et al. studied both acquisition and tracking, at the chip level, of a classical spread-spectrum system. The same study can be found in [3] (pp. 562–570), and a similar one in [1] (pp. 159–165); also in [1] (pp. 428–435) a serial symbol synchronizer based on the MAP criterion is derived.

We have not seen a treatment of the problem of synchronization in commutation signalling in the standard literature. Although this technique was proposed in 1984, it is only recently that it has been studied in depth, after it was recognized that it offers a low complexity solution to the problem of ISI induced in the indoor wireless environments.

This project investigates the problem of chip synchronization appropriate for indoor wireless networks operating with commutation signalling. Its original contribution lies is the fact that, based on the MAP criterion, it derives a structure of a chip synchronizer for a modem operating with commutation signalling. The job of the synchronizer is to provide sample points once per chip, such that reliable decisions can be made from the sample values. The specific commutation signalling scheme considered here uses DBPSK with differential coherent detection at the receiver.

The project is structured in the following manner: in Section 2 a structure of synchronizer based on the MAP criterion is derived. In Section 3, the convergence of the proposed algorithm for pulses having a raised-cosine shaped spectrum and for half-sine shaped pulses is examined. In order to prove the concept, the synchronization system is implemented and simulated for half-sine pulses in Section 4. Two alternate synchronizer configurations, that do not need carrier phase tracking, are also presented in this section. Finally, details of some of the blocks used to build the system in Simulink® are presented in the Appendix.

2 Derivation of the Synchronizer Structure Based on the MAP Criterion

2.1 Block diagram of a DBPSK modem

In this section we will derive a structure for the synchronizer based on the MAP criterion. We start by describing the DBPSK modem in Section 2.1, then proceed with the mathematical derivations that lead us to an expression to be optimized in Section 2.2. We continue with some practical implementation considerations in Section 2.3, the analysis of convergence of the proposed algorithm in Section 2.4 and end up with a structure of synchronizer in Section 2.5.

The scheme considered in this chapter is an implementation of 2xM commutation signalling, using bandwidth spread DBPSK signals. Its block diagram is shown in Figure 2.

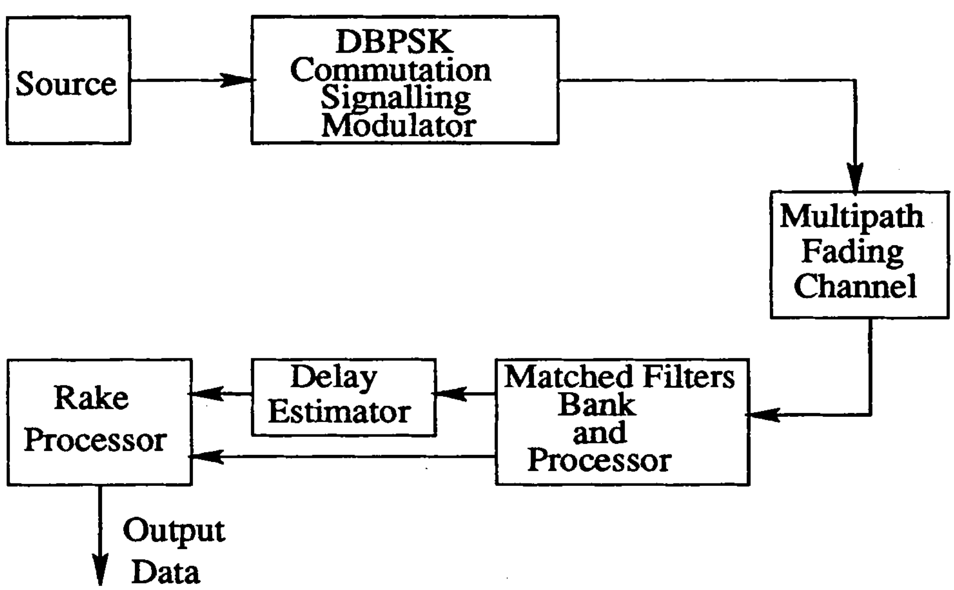

Figure 2: The DBPSK Commutation Signalling Block Diagram

The M commutation signal sets consist of M short chip sequences, selected in such a way as to have good shift and cross-correlation orthogonality.

In Figure 3 it is shown the DBPSK commutation signalling modulator. The source data is differentially encoded and then multiplied by the commutation signalling codewords, used in sequence (the commutation signalling codewords are, in turn, modulated by a spectrum formatting waveform, such as raised-cosine pulse or half-sine pulse, not shown in the figure); the resulting waveform is then converted to passband and transmitted over the channel.

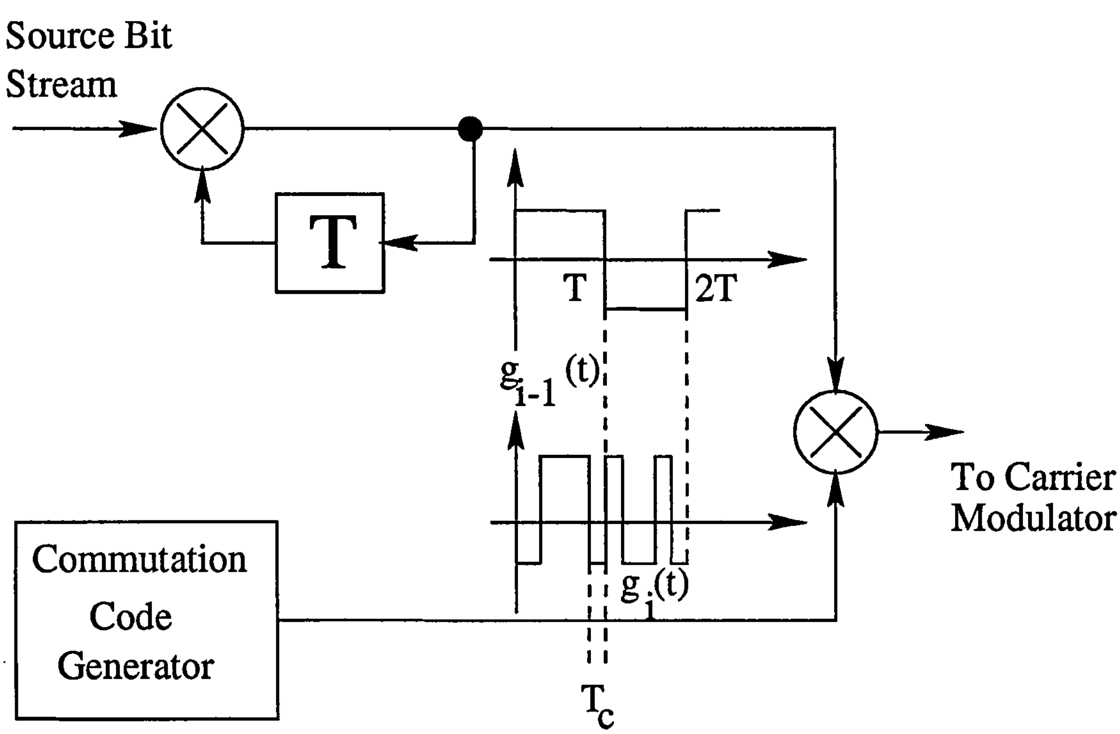

Figure 3: The DBPSK Commutation Signalling Modulator

Figure 4 shows the DBPSK commutation signalling coherent differential demodulator for the case of three shift-quasi orthogonal waveforms; it consists of a bank of filters matched to the commutation signals, followed by delay, conjugate and multiply blocks, the output of which are combined by a RAKE processor[1] with activated taps corresponding to the channel multipath delays (the multipath delays can be estimated from the matched filters outputs). The RAKE processor is usually implemented as a digital delay line which entails sampling its input. Synchronization provides the RAKE processor with the correct sampling instants and therefore with the correct sample values needed for making a decision.

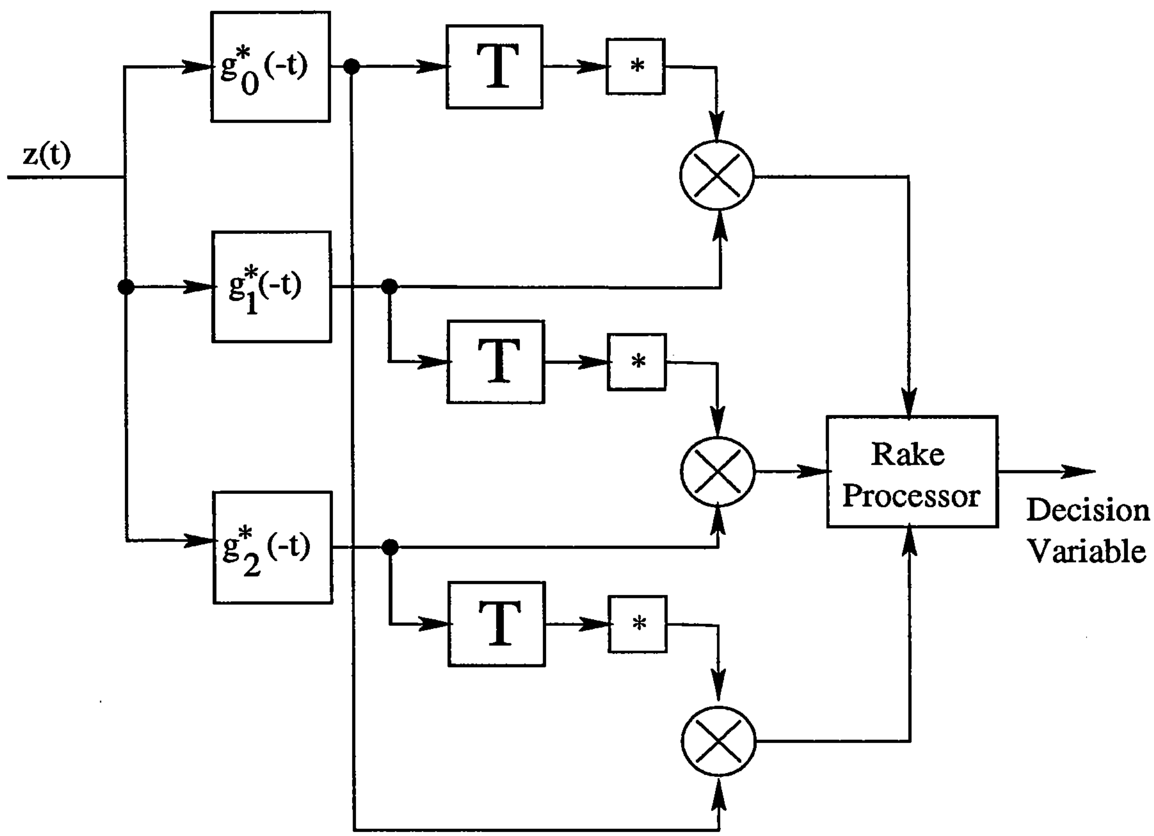

Figure 4: The DBPSK Commutation Signalling Demodulator

1 The "RAKE processor" is a structure that contains a tapped delay line, with taps spaced at intervals of a chip duration; it attempts to collect the signal energy from all the received signal paths that fall within the span of the delay line and carry the same information.

2.2 Application of the MAP criterion

At the beginning of our analysis, we will make the assumption that the carrier phase is known at the receiver (the receiver employs carrier phase tracking[2]); then, in Section 4.3, we will show how, with minor modifications, the scheme derived here can be used without carrier phase tracking.

Let us consider the wave present at the input of the receiver due to a single path. The incoming wave, in baseband representation, has the form

$$z(t) = x(t;\tau) + n(t)$$

where $n(t)$ is white Gaussian noise with (double sided) power spectral density N_0, $x(t;\tau)$ is the signal and where τ is the delay versus a certain reference (the same reference being used also by the locally generated clock). The signalling interval (known at the receiver) is T, and the observation interval, T_0, is of K symbols, $T_0 = K \cdot T$; during this interval it is assumed that the delay, τ , remains constant. We assume there are M signalling quasi-shift orthogonal waveforms, $g_0(t) \ldots g_{M-1}(t)$, used in this order, and that the information bits, b_k , are known during the preamble (training period). We can then write:

$$z(t) = \sum_{k=0}^{K-1} b_k g_{[k]_M}(t - kT - \tau) + n(t) \tag{1}$$

where $[k]_M$ means "k modulo M". By choosing a convenient complete orthonormal basis

[2] This assumption is needed in order for the synchronization problem to be mathematically tractable.

$\{\phi_i\}$ to represent $z(t)$ we have:

$$
\begin{aligned}
z_i &= \int_{T_0} z(t)\phi_i^*(t)dt \\
&= \int_{T_0} \left\{ \sum_{k=0}^{K-1} b_k g_{[k]_M}(t-kT-\tau) + n(t) \right\} \phi_i^*(t)dt \\
&= \sum_{k=0}^{K-1} b_k \int_{T_0} g_{[k]_M}(t-kT-\tau)\phi_i^*(t)dt + \int_{T_0} n(t)\phi_i^*(t)dt \\
&= \sum_{k=0}^{K-1} b_k g_{[k]_M i} + n_i \quad , \quad i = 1, 2, \ldots
\end{aligned}
\tag{2}
$$

Let $\mathbf{z} = [z_1 z_2 \ldots]^T$; then, according to the MAP criterion, the value of τ which maximizes the probability density function (pdf) $p(\tau|\mathbf{z})$ is the estimate of the delay. Making use of Bayes' relation, we have:

$$
p(\tau|\mathbf{z}) = \frac{p(\mathbf{z}|\tau) \cdot p(\tau)}{p(\mathbf{z})} \tag{3}
$$

Let $\mathbf{z_N} = [z_1 z_2 \ldots z_N]^T$; the variables z_i are Gaussian independent identically distributed (iid), having the mean $\sum\limits_{k=0}^{K-1} b_k g_{[k]_M i}$ and variance N_0, and therefore we have:

$$
p(\mathbf{z_N}|\tau) = \prod_{i=1}^{N} p(z_i|\tau) \tag{4}
$$

$$
p(z_i|\tau) = \frac{1}{(2\pi N_0)^{1/2}} \cdot \exp\left\{ -\frac{\left| z_i - \sum\limits_{k=0}^{K-1} b_k g_{[k]_M i} \right|^2}{2N_0} \right\} \tag{5}
$$

$$
p(\mathbf{z_N}|\tau) = \frac{1}{(2\pi N_0)^{N/2}} \cdot \exp\left\{ -\frac{\sum\limits_{i=1}^{N} \left| z_i - \sum\limits_{k=0}^{K-1} b_k g_{[k]_M i} \right|^2}{2N_0} \right\} \tag{6}
$$

and since $p(\tau)$ is constant over a symbol interval and

$$p(\mathbf{z_N}) = \frac{1}{(2\pi N_0)^{N/2}} \cdot \exp\left\{-\frac{\sum_{i=1}^{N} |z_i|^2}{2N_0}\right\}$$

it follows that:

$$p(\tau|\mathbf{z_N}) = \text{const} \cdot \exp\left\{-\frac{\sum_{i=1}^{N}\left|\sum_{k=0}^{K-1} b_k g_{[k]_M i}\right|^2}{2N_0}\right\} \cdot \exp\left\{\frac{\text{Re}\sum_{i=1}^{N} z_i \sum_{k=0}^{K-1} b_k g^*_{[k]_M i}}{N_0}\right\} \tag{7}$$

In the above expression the first exponential represents the energy of the transmitted signal, and does not depend upon τ , and thus only the second exponential influences the decision; we have:

$$\arg\max_{\tau} p(\tau|\mathbf{z_N}) = \arg\max_{\tau}\left(\frac{1}{N_0}\text{Re}\sum_{i,k} z_i b_k g^*_{[k]_M i}\right)$$

and thus, the likelihood function that must be maximized is:

$$\Lambda(\hat{\tau}) = \frac{1}{N_0} \cdot \text{Re}\sum_{i,k} z_i b_k g^*_{[k]_M i} \tag{8}$$

which further, by letting $N \to \infty$, becomes:

$$\Lambda(\hat{\tau}) = \frac{1}{N_0} \cdot \sum_{k=0}^{K-1} b_k \text{Re}\left[\int_{T_0} z(t) g^*_{[k]_M}(t - kT - \hat{\tau})dt\right] = \frac{1}{N_0} \cdot \sum_{k=0}^{K-1} b_k q_k(\hat{\tau}) \tag{9}$$

where:

$$q_k(\hat{\tau}) = \mathrm{Re}\left[\int_{T_0} z(t) g^*_{[k]_M}(t - kT - \hat{\tau}) dt\right]$$

2.3 Practical implementation approximations

In order to implement this relation in practice, the observation interval, T_0 , is assumed to be very long compared to the symbol duration, T , and then the integral is taken from $-\infty$ to ∞ . With those approximations, it follows that

$$\widetilde{q_k}(\hat{\tau}) = \mathrm{Re}\left[\int_{-\infty}^{\infty} z(t) g^*_{[k]_M}(t - kT - \hat{\tau}) dt\right]$$

$$\widetilde{\Lambda}(\hat{\tau}) = \frac{1}{N_0} \cdot \sum_{k=0}^{K-1} b_k \widetilde{q_k}(\hat{\tau}) \tag{10}$$

We can use the derivative of $\widetilde{\Lambda}(\hat{\tau})$ as a control voltage to adjust the delay:

1. $\widetilde{\Lambda}(\hat{\tau}) = \max$, $\dot{\widetilde{\Lambda}}(\hat{\tau}) = 0$, and the frequency (and phase) of the local oscillator remains unchanged;
2. $-\dot{\widetilde{\Lambda}}(\hat{\tau}) < 0$, which will determine a momentary decrease in the local oscillator's frequency, leading to an increase of $\hat{\tau}$;
3. $-\dot{\widetilde{\Lambda}}(\hat{\tau}) > 0$, which will determine a momentary increase in the local oscillator's frequency, leading to a decrease of $\hat{\tau}$.

Therefore, in the first case the local phase remains unchanged (i.e. the estimate is good) while in cases 2 and 3 the phase is adjusted so that the local oscillator gets synchronized with the incoming wave. The voltage controlled clock (VCC) by its integrating effect, will approximate the sum appearing in the formula of $\widetilde{\Lambda}(\hat{\tau})$; we can then use as control voltage v_k, where

$$\begin{aligned} v_k &= -b_k \frac{d}{d\hat{\tau}} \mathrm{Re} \left[\int_{-\infty}^{\infty} z(t) g^*_{[k]_M}(t - kT - \hat{\tau}) dt \right] \\ &= b_k \mathrm{Re} \left[\int_{-\infty}^{\infty} z(t) \dot{g}^*_{[k]_M}(t - kT - \hat{\tau}) dt \right] \end{aligned} \tag{11}$$

with the dot indicating differentiation with respect to the whole argument.

2.4 Analysis of the proposed convergence algorithm

In order to analyze the convergence of the algorithm based on the control voltage v_k , we consider the case where the channel noise is zero; then

$$z(t) = \sum_{i=0}^{K-1} b_i g_{[i]_M}(t - iT - \tau)$$

and

$$v_k = -\frac{d}{d\hat{\tau}} b_k \mathrm{Re} \int_{T_0} \sum_{i=0}^{K-1} b_i g_{[i]_M}(t - iT - \tau) \cdot g^*_{[k]_M}(t - kT - \hat{\tau}) dt$$

and since the sequences $g_{[k]_M}$ are shift-quasi orthogonal and real, it follows that

$$\begin{aligned} v_k &= -\frac{d}{d\hat{\tau}} b_k \mathrm{Re} \int_{T_0} b_k g_{[k]_M}(t - iT - \tau) \cdot g^*_{[k]_M}(t - kT - \hat{\tau}) dt \\ &= -\frac{d}{d\hat{\tau}} \int_{T_0} g_{[k]_M}(t - kT - \tau) \cdot g^*_{[k]_M}(t - kT - \hat{\tau}) dt \\ &\approx -\frac{d}{d\hat{\tau}} R_{g_{[k]_M}}(\hat{\tau} - \tau) \end{aligned}$$

where $R_{g_{[k]_M}}$ represents the time autocorrelation of the signalling function $g_{[k]_M}$[3].

Now, let us consider the clock locally generated by the VCC, delayed by $\hat{\tau}$ with respect to a certain reference (the incoming wave is supposed to be delayed by τ with respect to the same reference); the instantaneous phase of the clock is:

$$\hat{\Phi} = \omega_0(t - \hat{\tau})$$

which implies that

$$\dot{\hat{\Phi}} = \omega_0(1 - \dot{\hat{\tau}})$$

The VCC operates according to the equation:

$$\dot{\hat{\Phi}} = \omega_0 + K_v \cdot v_k$$

where $\omega_0 = 2\pi/T$ and K_v is the gain of the VCC. Therefore, for $t \in [kT, (k+1)T)$, we have:

$$-\omega_0 \cdot \dot{\hat{\tau}} = K_v \cdot v_k$$

$$\hat{\tau} = \hat{\tau}_k - \frac{K_v \cdot v_k}{\omega_0} \cdot t$$

$$\hat{\tau}_{k+1} = \hat{\tau}_k - \frac{K_v \cdot v_k}{\omega_0} \cdot T \tag{12}$$

---

3 The $\approx$ sign was used instead of the $=$ sign, since in order for this relation to be exact, the integral should extend from $-\infty$ to ∞.

Applying this last relation to our case, we have:

$$\begin{aligned}\hat{\tau}_{k+1} - \hat{\tau}_k &= -2\pi \frac{K_v \cdot v_k}{\omega_0^2} \\ &= 2\pi \frac{K_v}{\omega_0^2} \cdot \frac{d}{d\hat{\tau}} R_{g_{[k]_M}}(\hat{\tau} - \tau) \mid_{\hat{\tau}=\hat{\tau}_k} \end{aligned} \tag{13}$$

To demonstrate the convergence of the algorithm based on this expression is quite a difficult task; on the other hand, by using a simplified expression, the convergence near the point $\hat{\tau} = \tau$ can be demonstrated. For this, we use the fact that in the vicinity of their maximum, all three autocorrelation functions, $R_{g_{[k]_M}}$, vary almost identically; then, Eq. 13 simplifies to:

$$\hat{\tau}_{k+1} - \hat{\tau}_k = 2\pi \frac{K_v}{\omega_0^2} \cdot \frac{d}{d\hat{\tau}} R_g(\hat{\tau} - \tau) \mid_{\hat{\tau}=\hat{\tau}_k} \tag{14}$$

where we denote by R_g the unique autocorrelation function approximating the three autocorrelation functions. It is immediately apparent that in the cases where R_g is a smooth function of $\hat{\tau}$ in the vicinity of τ, it has a unique extremum (maximum) and provided the multiplicative constant in the right hand of the above equation is properly selected, this iterative method will converge towards the actual value of the delay, $\hat{\tau}_k \rightarrow \tau$.

2.5 Proposed structure for the synchronizer

In order to implement a synchronizer structure based on the expression obtained for v_k, we need to rewrite it in a different, but equivalent, manner; by noticing that[4]

$$-\frac{d}{d\theta}\left[z(\theta)\otimes g^*(-\theta)\right]=\int_{-\infty}^{\infty} z(t)\dot{g}^*(t-\theta)dt$$

and therefore

$$-\frac{d}{d\theta}\left[z(\theta)\otimes g^*(-\theta)\right]\Big|_{\theta=kT+\hat{\tau}}=\int_{-\infty}^{\infty} z(t)\dot{g}^*(t-kT-\hat{\tau})dt$$

we can rewrite the control voltage as

$$v_k=b_k\mathrm{Re}\left\{-\frac{d}{dt}\left[z(t)\otimes g^*_{[k]_M}(-t)\right]\Big|_{t=kT+\hat{\tau}}\right\} \tag{15}$$

which leads to the synchronizer structure illustrated in Figure 5.

The sample-and-hold blocks are needed to insure that the "ringing" (replicas of the same bit arriving delayed to the receiver due to multipath) does not influence the control voltage of the VCC during the symbol period, T. The decision block is the one that performs coarse synchronization (acquisition) up to an accuracy of $\pm T_c/2$ and then presets the timing block (a cyclic counter) to start its operation from the position corresponding to the specific code of the sequence arriving at that particular time.

---

4 The sign $\otimes$ means "convolution".

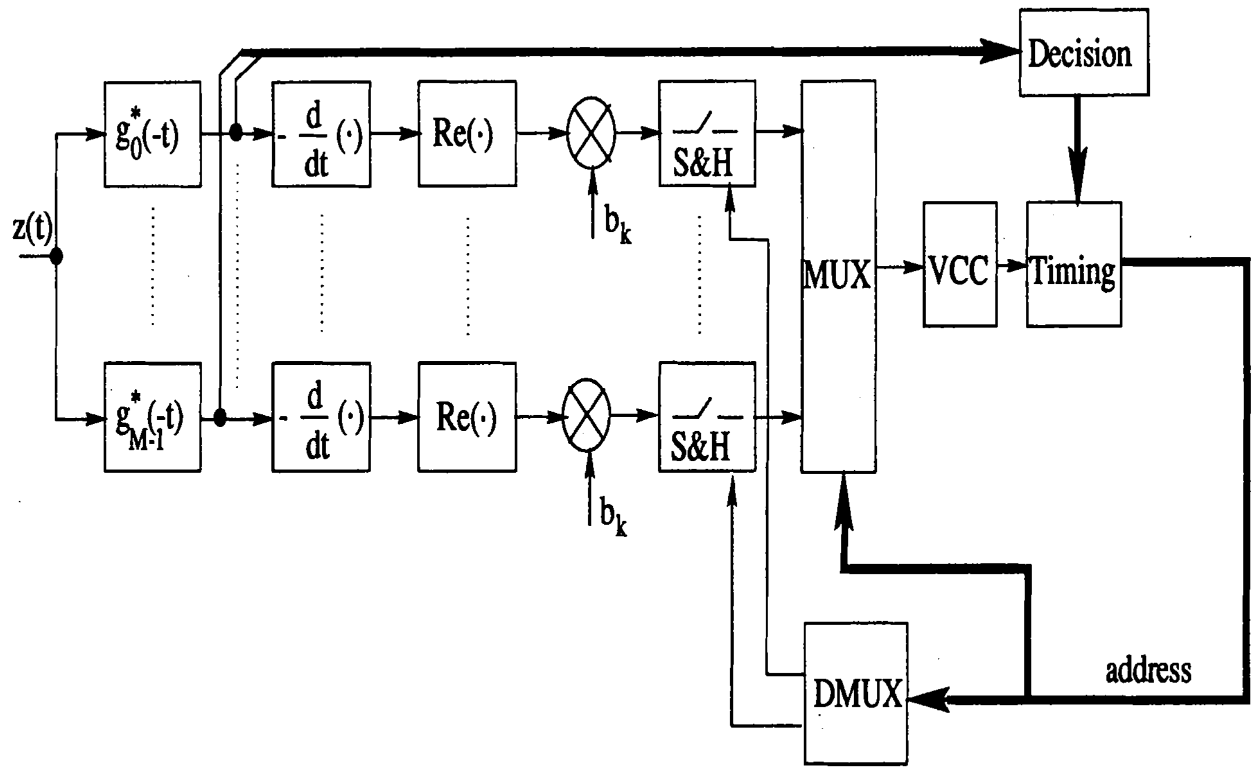

Figure 5: Synchronizer structure

3 Convergence for Specific Waveforms

3.1 Codewords and codewords autocorrelations

Prior to investigating the convergence of the algorithm for specific waveforms used to shape the chips, we need to calculate the general expression of the signalling waveform autocorrelation as a function of the pulse autocorrelation. In order to demonstrate the concept, three codewords of length five each (i.e. 5 chips per codeword) are used, which are sufficient to overcome the channel multipath delay spread; the codewords with the best properties in terms of orthogonality were found to be: 10110, 10001 and 11110.

Let $g_l(t) = \sum_{i=0}^{4} c_{i,l} \cdot p(t - iT_c)$ be one signalling waveform, where $l = 0, 1, 2$, $c_{i,l}$ are the chips and $p(t)$ is the chip shaping waveform; then

$$
\begin{aligned}
R_g(\theta) &= \int_{-\infty}^{\infty} g(t) \cdot g(t-\theta) dt \\
&= \int_{-\infty}^{\infty} \sum_{i=0}^{4} c_i p(t - iT_c) \cdot \sum_{j=0}^{4} c_j p(t - jT_c - \theta) dt \\
&= \sum_{i=0}^{4} \sum_{j=0}^{4} c_i c_j \int_{-\infty}^{\infty} p[\theta + (j-i)T_c + t] \cdot p(t) dt \\
&= \sum_{i=0}^{4} \sum_{j=0}^{4} c_i c_j R_p[\theta + (j-i)T_c] \qquad (16)
\end{aligned}
$$

The autocorrelations are, in order:

1. Codeword [10110]: $c_0 = -1, c_1 = 1, c_2 = c_3 = -1, c_4 = 1$, and after a few simple mathematical operations, we get:

$$R_{g_0}(\theta) = 5 \cdot R_p(\theta) - 2 \cdot [R_p(\theta + T_c) + R_p(\theta - T_c)]$$

$$
\begin{aligned}
&- \quad [R_p(\theta + 2T_c) + R_p(\theta - 2T_c)] + 2 \cdot [R_p(\theta + 3T_c) + R_p(\theta - 3T_c)] \\
&- \quad [R_p(\theta + 4T_c) + R_p(\theta - 4T_c)] \qquad (17)
\end{aligned}
$$

2. Codeword [10001]: $c_0 = -1, c_1 = c_2 = c_3 = 1, c_4 = -1$, leading to:

$$
\begin{aligned}
R_{g_1}(\theta) &= 5 \cdot R_p(\theta) - [R_p(\theta + 2T_c) + R_p(\theta - 2T_c)] \\
&- \quad 2 \cdot [R_p(\theta + 3T_c) + R_p(\theta - 3T_c)] \\
&+ \quad [R_p(\theta + 4T_c) + R_p(\theta - 4T_c)] \qquad (18)
\end{aligned}
$$

3. Codeword [11110]: $c_0 = c_1 = c_2 = c_3 = -1, c_4 = 1$, leading to:

$$
\begin{aligned}
R_{g_2}(\theta) &= 5 \cdot R_p(\theta) + 2 \cdot [R_p(\theta + T_c) + R_p(\theta - T_c)] \\
&+ \quad [R_p(\theta + 2T_c) + R_p(\theta - 2T_c)] \\
&- \quad [R_p(\theta + 4T_c) + R_p(\theta - 4T_c)] \qquad (19)
\end{aligned}
$$

Using the above results, we will further investigate the convergence when employing the following chip-shaping waveforms:

1. raised-cosine pulses;

2. half-sine pulses.

3.2 Codewords autocorrelation functions — raised-cosine case

We will start by deriving the codewords autocorrelations in Section 3.2, and then proceed to examine the convergence of the synchronization algorithm for the raised-cosine case in Section 3.3.

The expression of a pulse having a raised-cosine spectrum is:

$$p(t) = \frac{\sin(\pi t/T_c)}{\pi t/T_c} \cdot \frac{\cos(\beta\pi t/T_c)}{1-(2\beta t/T_c)^2}$$

where the "roll-off factor" $\beta \in [0,1]$; the Fourier transform of this pulse is:

$$P(f) = \begin{cases} T_c, & \text{for } 0 \le |f| \le \dfrac{1-\beta}{2T_c}; \\ \dfrac{T_c}{2}\left\{1 - \sin\left[\dfrac{\pi T_c\left(|f| - \frac{1}{2T_c}\right)}{\beta}\right]\right\}, & \text{for } \dfrac{1-\beta}{2T_c} \le |f| \le \dfrac{1+\beta}{2T_c}; \\ 0, & \text{else.} \end{cases}$$

and its power spectral density is:

$$S(f) = \begin{cases} T_c^2, & \text{for } 0 \le |f| \le \dfrac{1-\beta}{2T_c}; \\ \dfrac{T_c^2}{4}\left\{\dfrac{3}{2} - 2\sin\left[\dfrac{\pi T_c\left(|f| - \frac{1}{2T_c}\right)}{\beta}\right] - \dfrac{1}{2}\cos\left[\dfrac{2\pi T_c\left(|f| - \frac{1}{2T_c}\right)}{\beta}\right]\right\}, & \text{for } \dfrac{1-\beta}{2T_c} \le |f| \le \dfrac{1+\beta}{2T_c}; \\ 0, & \text{else.} \end{cases}$$

Since the autocorrelation is the inverse Fourier transform of the spectral power density,

$R_p(\theta) = \int_{-\infty}^{\infty} S(f) \exp(\jmath 2\pi f \theta) df$, after some straight, albeit tedious, calculations, we get:

$$\begin{aligned} R_p(\theta) &= T_c \cdot \left\{ \frac{5}{8} \cdot \frac{\sin\left[\pi(1-\beta)\frac{\theta}{T_c}\right]}{\pi\frac{\theta}{T_c}} + \frac{3}{8} \cdot \frac{\sin\left[\pi(1+\beta)\frac{\theta}{T_c}\right]}{\pi\frac{\theta}{T_c}} \right. \\ &+ \sin\left(\pi\frac{\theta}{T_c}\right) \cdot \cos\left(\pi\beta\frac{\theta}{T_c}\right) \cdot \frac{4\pi\frac{\theta}{T_c}}{\left(\frac{\pi}{\beta}\right)^2 - \left(2\pi\frac{\theta}{T_c}\right)^2} \\ &\left. - 0.25 \cdot \sin\left(\pi\beta\frac{\theta}{T_c}\right) \cdot \cos\left(\pi\frac{\theta}{T_c}\right) \cdot \frac{4\pi\frac{\theta}{T_c}}{\left(\frac{2\pi}{\beta}\right)^2 - \left(2\pi\frac{\theta}{T_c}\right)^2} \right\} \end{aligned} \tag{20}$$

The graph of this expression is plotted in Figure 6 as a function of $\frac{\theta}{T_c}$, where $\frac{\theta}{T_c} \in [-1, 1]$, for β taking the values $0.1, 0.3, \ldots 0.9$.

Based on Eq. 20, we can further calculate and plot the autocorrelation functions of the

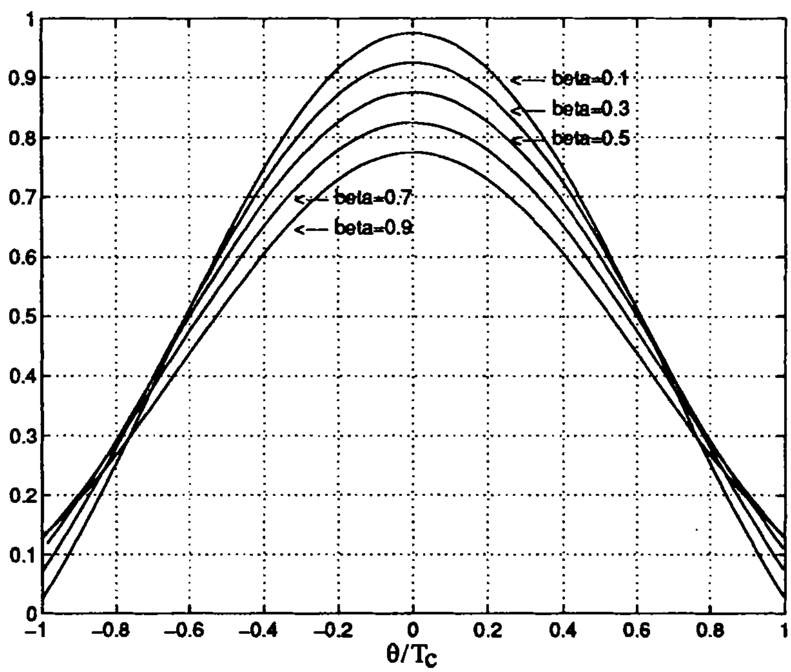

Figure 6: Raised-cosine autocorrelation

three codewords (Figure 7, Figure 8 and Figure 9 respectively):

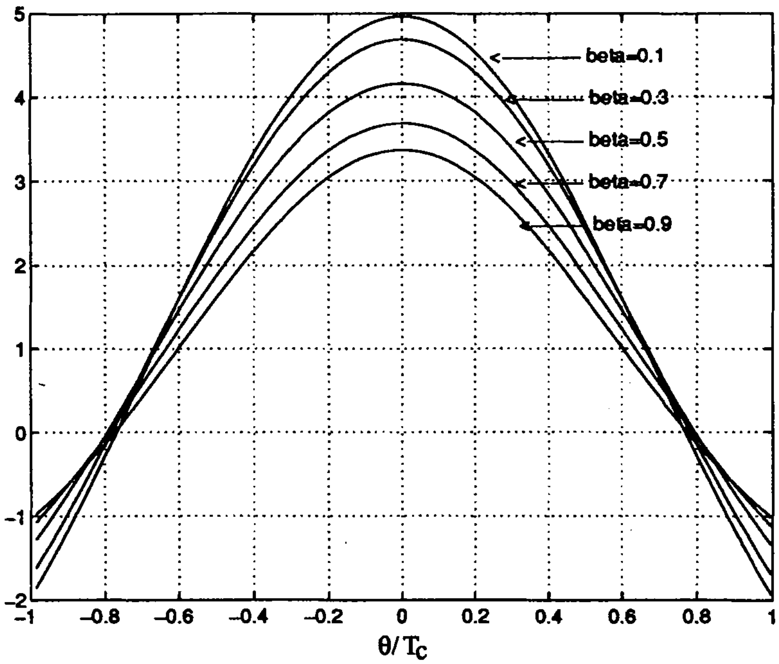

Figure 7: Codeword 10110 autocorrelation - raised-cosine chips

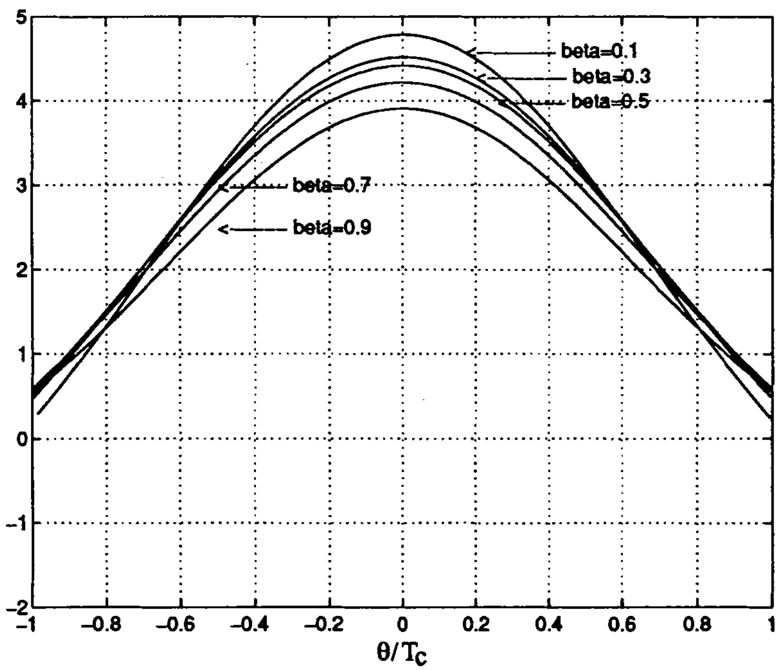

Figure 8: Codeword 10001 autocorrelation - raised-cosine chips

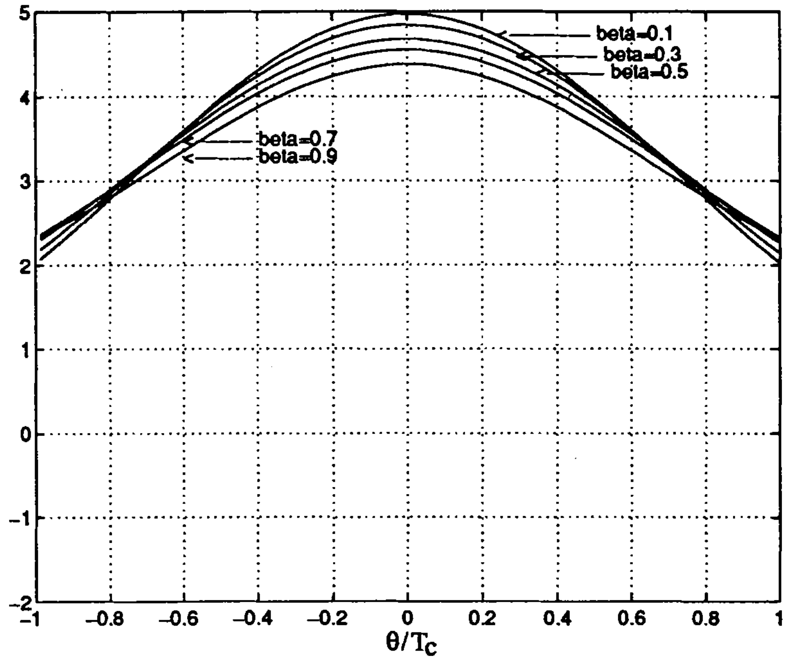

Figure 9: Codeword 11110 autocorrelation - raised-cosine chips

3.3 Convergence in the raised-cosine case

As can be seen, the three autocorrelation functions behave in a similar manner (i.e., their derivatives are almost equal) around the point $\theta = 0$, and they have only an absolute maximum, at $\theta = 0$, so the algorithm will converge. To demonstrate this, we will approximate $R_{[g]_M}(\theta)$ with $5 \cdot R_p(\theta)$; also, in the vicinity of $\theta = 0$, we will develop the sine and the cosine in Taylor series and keep only the relevant powers of $x \triangleq \frac{\theta}{T_c}$, with the result that:

$$R_p(x) \approx T_c \cdot \left\{ \frac{5}{8} \cdot \frac{\pi(1-\beta)x - \frac{\pi^3(1-\beta)^3}{6}x^3}{\pi x} + \frac{3}{8} \cdot \frac{\pi(1+\beta)x - \frac{\pi^3(1+\beta)^3}{6}x^3}{\pi x} \right.$$

$$+\pi x \cdot \frac{4\pi x}{\left(\frac{\pi}{\beta}\right)^2 - 4\pi^2 x^2} - \pi\beta x \cdot \frac{\pi x}{\left(\frac{2\pi}{\beta}\right)^2 - 4\pi^2 x^2}\Bigg\}$$

$$= T_c \cdot \left\{1 - 0.25\beta - \left[\frac{5}{48}\pi^2(1-\beta)^3 + \frac{1}{16}\pi^2(1+\beta)^3 - \frac{4}{1/\beta^2 - 4x^2} + \frac{\beta}{4/\beta^2 - 4x^2}\right] x^2\right\}$$

This relation can be further simplified by discarding the terms in x^2 from the denominators, leading to the following expression for the simplified raised-cosine pulse autocorrelation:

$$R_p(x) \approx T_c \cdot \left\{1 - 0.25\beta - \left[\frac{\pi^2}{24}(4 - 3\beta + 12\beta^2 - \beta^3) + \frac{\beta^3 - 16\beta^2}{4}\right] x^2\right\} \qquad (21)$$

It follows that

$$\frac{d}{d\hat{\tau}} R_p(\theta) \approx -T_c \cdot \left[\frac{\pi^2}{3} - \frac{\pi^2}{4}\beta + (\pi^2 - 8)\beta^2 + \frac{6 - \pi^2}{12}\beta^3\right] x = -f(\beta) \cdot \theta$$

where $f(\beta) \triangleq \frac{\pi^2}{3} - \frac{\pi^2}{4}\beta + (\pi^2 - 8)\beta^2 + \frac{6 - \pi^2}{12}\beta^3$, and therefore

$$\frac{d}{d\hat{\tau}} R_g(\hat{\tau} - \tau) \approx -10 \cdot \frac{f(\beta)}{T_c} \cdot (\hat{\tau} - \tau) \qquad (22)$$

which in turn leads to

$$\hat{\tau}_{k+1} - \hat{\tau}_k \approx -20\pi \frac{K_v}{\omega_0^2 \cdot T_c} \cdot f(\beta) \cdot (\hat{\tau}_k - \tau) = B \cdot (\tau - \hat{\tau}_k) \qquad (23)$$

where $B \triangleq 20\pi \frac{K_v}{\omega_0^2 \cdot T_c} \cdot f(\beta)$.

By applying the z-transform to Eq. 23, we get:

$$z\mathcal{T} - \mathcal{T} = -B \cdot \left(\mathcal{T} - \frac{\tau}{1 - z^{-1}}\right)$$

where $\mathcal{T} \triangleq Z(\hat{\tau}_k)$. For the above to converge, the condition is that the unit-circle be included in the region of convergence, i.e. $|1 - B| < 1$, which sets an upper limit on K_v, the VCC gain; the solution is then:

$$\hat{\tau}_k = \tau[1 - (1 - B)^k] \quad , \quad k = 1, 2, \ldots \tag{24}$$

Therefore, when the condition on B (or equivalently, on K_v), is fulfilled, $\hat{\tau}_k \to \tau$, which means that the algorithm is convergent for pulses with raised-cosine shaped spectrum and the system synchronizes, once capture has taken place.

3.4 Codewords autocorrelation functions — half-sine case

Using the same procedure as in the previous section, in Section 3.4 we will calculate the codewords autocorrelations in the half-sine case, followed by an examination of the convergence algorithm in Section 3.5.

We start from the mathematical expression of the half-sine pulse:

$$p(t) = \begin{cases} \cos\left(\pi \frac{t}{T_c}\right), & \text{for } t \in \left[-\frac{T_c}{2}, \frac{T_c}{2}\right]; \\ 0, & \text{else.} \end{cases}$$

Its autocorrelation function is:

$$
\begin{aligned}
R_p(\theta) &= \int_{-T_c/2}^{T_c/2} p(t) \cdot p(t-\theta) dt \\
&= \begin{cases}
\frac{T_c}{2} \cdot \cos\left(\pi \frac{\theta}{T_c}\right) + \frac{\theta}{2} \cdot \left[\frac{\sin\left(\pi \frac{\theta}{T_c}\right)}{\pi \frac{\theta}{T_c}} - \cos\left(\pi \frac{\theta}{T_c}\right) \right], & \text{for } \theta \in [0, T_c]; \\
\frac{T_c}{2} \cdot \cos\left(\pi \frac{\theta}{T_c}\right) - \frac{\theta}{2} \cdot \left[\frac{\sin\left(\pi \frac{\theta}{T_c}\right)}{\pi \frac{\theta}{T_c}} - \cos\left(\pi \frac{\theta}{T_c}\right) \right], & \text{for } \theta \in [-T_c, 0]; \\
0, & \text{else.}
\end{cases}
\end{aligned}
\tag{25}
$$

and is plotted on Figure 10.

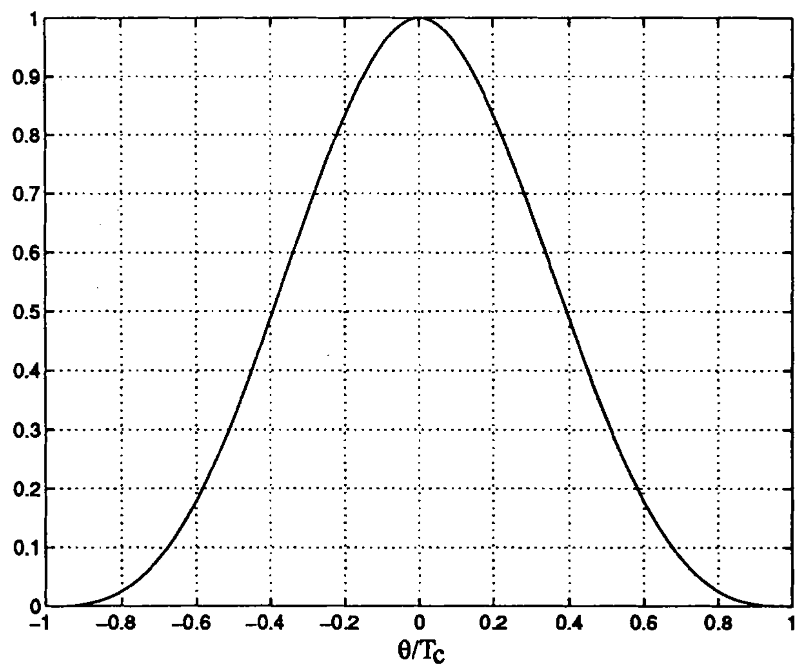

Figure 10: Half-sine autocorrelation

Since the pulses are time limited to a period of T_c, it follows that $R_p(\theta \pm n \cdot T_c) = 0$ for $n > 1$. Therefore, the only nonzero terms in the signaling waveforms' autocorrelations are, in this case, $R_p(\theta)$ and $R_p(\theta \pm T_c)$; we have:

$$R_p(\theta + T_c) = \begin{cases} \dfrac{\theta}{2} \cdot \left[\cos\left(\pi\dfrac{\theta}{T_c}\right) - \dfrac{\sin\left(\pi\dfrac{\theta}{T_c}\right)}{\pi\dfrac{\theta}{T_c}}\right], & \text{for } \theta \in [-T_c, 0]; \\ 0, & \text{else.} \end{cases}$$

$$R_p(\theta - T_c) = \begin{cases} \dfrac{\theta}{2} \cdot \left[-\cos\left(\pi\dfrac{\theta}{T_c}\right) + \dfrac{\sin\left(\pi\dfrac{\theta}{T_c}\right)}{\pi\dfrac{\theta}{T_c}}\right], & \text{for } \theta \in [0, T_c]; \\ 0, & \text{else.} \end{cases}$$

The waveforms' autocorrelations are:

1. Codeword [10110]: $c_0 = -1, c_1 = 1, c_2 = c_3 = -1, c_4 = 1$:

$$\begin{aligned} R_{g_0}(\theta) &= 5 \cdot R_p(\theta) - 2 \cdot [R_p(\theta + T_c) + R_p(\theta - T_c)] \\ &= \begin{cases} 5 \cdot \dfrac{T_c}{2} \cdot \cos\left(\pi\dfrac{\theta}{T_c}\right) + 3 \cdot \dfrac{\theta}{2}\left[\dfrac{\sin\left(\pi\dfrac{\theta}{T_c}\right)}{\pi\dfrac{\theta}{T_c}} - \cos\left(\pi\dfrac{\theta}{T_c}\right)\right], & \text{for } \theta \in [0, T_c]; \\ 5 \cdot \dfrac{T_c}{2} \cdot \cos\left(\pi\dfrac{\theta}{T_c}\right) - 3 \cdot \dfrac{\theta}{2}\left[\dfrac{\sin\left(\pi\dfrac{\theta}{T_c}\right)}{\pi\dfrac{\theta}{T_c}} - \cos\left(\pi\dfrac{\theta}{T_c}\right)\right], & \text{for } \theta \in [-T_c, 0]; \\ 0, & \text{else.} \end{cases} \end{aligned} \tag{26}$$

2. Codeword [10001]: $c_0 = -1, c_1 = c_2 = c_3 = 1, c_4 = -1$:

$$R_{g_1}(\theta) = 5 \cdot R_p(\theta)$$

$$
= \begin{cases} 5 \cdot \frac{T_c}{2} \cdot \cos\left(\pi \frac{\theta}{T_c}\right) + 5 \cdot \frac{\theta}{2}\left[\frac{\sin\left(\pi \frac{\theta}{T_c}\right)}{\pi \frac{\theta}{T_c}} - \cos\left(\pi \frac{\theta}{T_c}\right)\right], & \text{for } \theta \in [0, T_c]; \\ 5 \cdot \frac{T_c}{2} \cdot \cos\left(\pi \frac{\theta}{T_c}\right) - 5 \cdot \frac{\theta}{2}\left[\frac{\sin\left(\pi \frac{\theta}{T_c}\right)}{\pi \frac{\theta}{T_c}} - \cos\left(\pi \frac{\theta}{T_c}\right)\right], & \text{for } \theta \in [-T_c, 0]; \\ 0, & \text{else.} \end{cases} \tag{27}
$$

3. Codeword [11110]: $c_0 = c_1 = c_2 = c_3 = -1, c_4 = 1$:

$$
\begin{aligned}
R_{g_2}(\theta) &= 5 \cdot R_p(\theta) + 2 \cdot [R_p(\theta + T_c) + R_p(\theta - T_c)] \\
&= \begin{cases} 5 \cdot \frac{T_c}{2} \cdot \cos\left(\pi \frac{\theta}{T_c}\right) + 7 \cdot \frac{\theta}{2}\left[\frac{\sin\left(\pi \frac{\theta}{T_c}\right)}{\pi \frac{\theta}{T_c}} - \cos\left(\pi \frac{\theta}{T_c}\right)\right], & \text{for } \theta \in [0, T_c]; \\ 5 \cdot \frac{T_c}{2} \cdot \cos\left(\pi \frac{\theta}{T_c}\right) - 7 \cdot \frac{\theta}{2}\left[\frac{\sin\left(\pi \frac{\theta}{T_c}\right)}{\pi \frac{\theta}{T_c}} - \cos\left(\pi \frac{\theta}{T_c}\right)\right], & \text{for } \theta \in [-T_c, 0]; \\ 0, & \text{else.} \end{cases}
\end{aligned} \tag{28}
$$

The autocorrelations are plotted in Figure 11, Figure 12 and Figure 13, respectively.

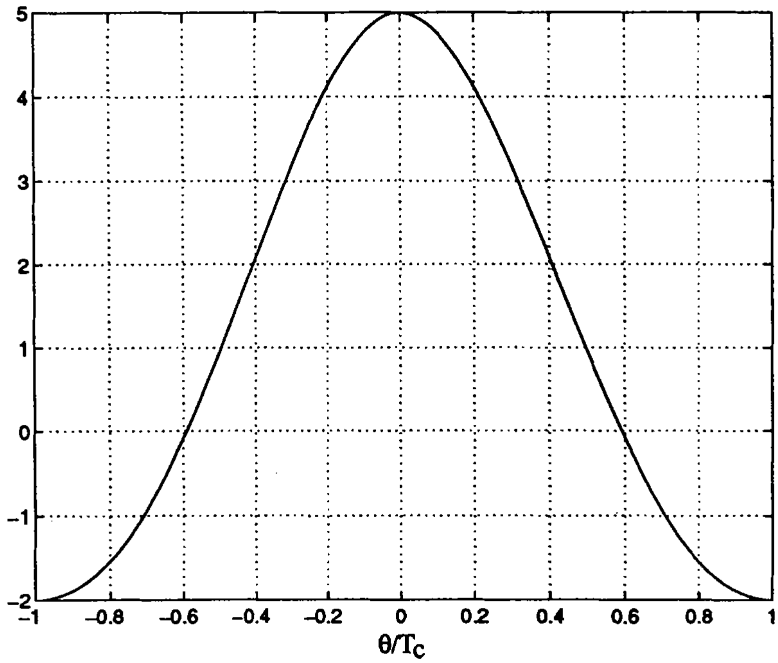

Figure 11: Codeword 10110 autocorrelation - half sine chips

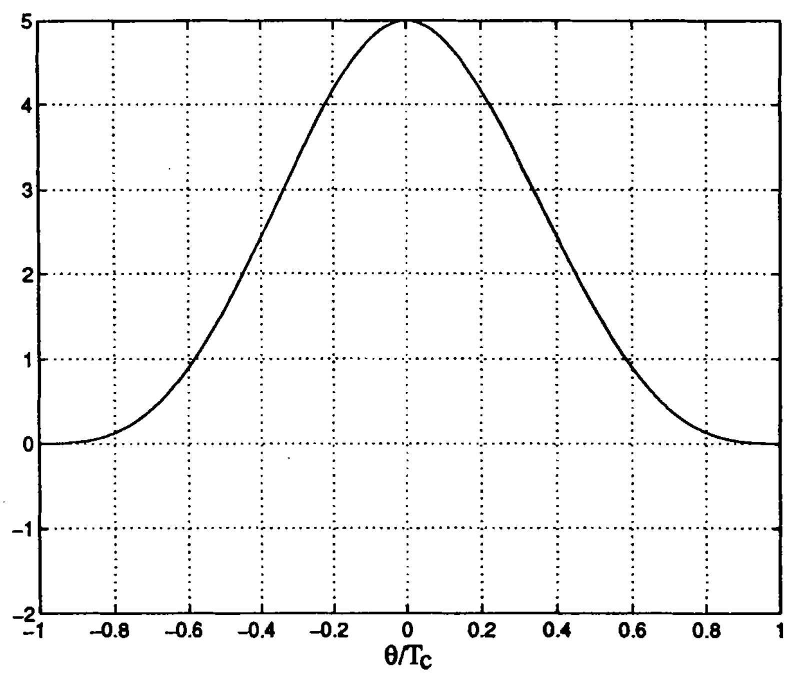

Figure 12: Codeword 10001 autocorrelation - half sine chips

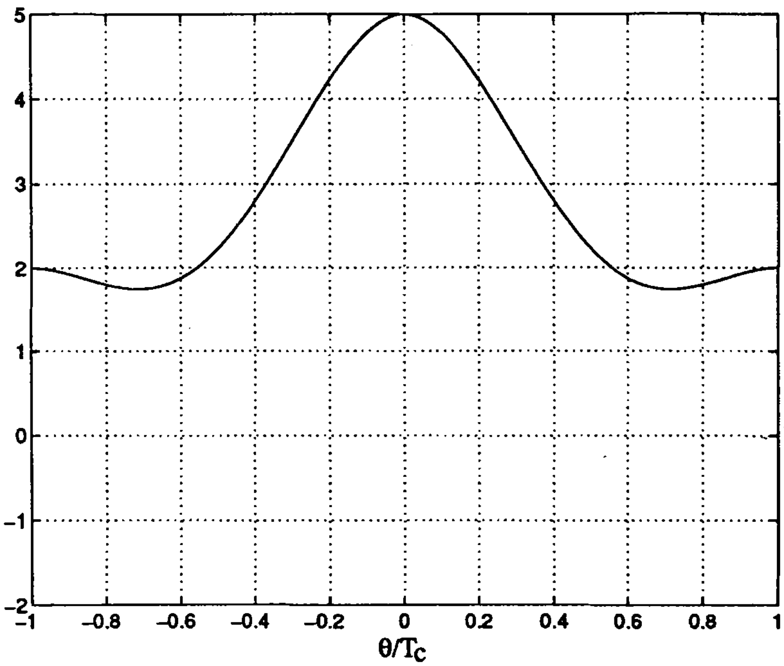

Figure 13: Codeword 11110 autocorrelation - half sine chips

3.5 Convergence in the half-sine case

In order to prove the convergence of the proposed algorithm around the point $\theta = 0$ (or $\hat{\tau} = \tau$), we will use the same kind of approximation that we used for the raised-cosine case, i.e. we'll approximate $R_g(x)$ with $5 \cdot R_p(x)$:

$$
\begin{aligned}
R_g(x) &\approx 5 \cdot R_p(x) \\
&= \begin{cases} 5 \cdot \dfrac{T_c}{2} \cdot \cos(\pi x) + 5 \cdot \dfrac{T_c}{2} \cdot x \cdot \left[\dfrac{\sin(\pi x)}{\pi x} - \cos(\pi x)\right], & \text{for } x > 0; \\ 5 \cdot \dfrac{T_c}{2} \cdot \cos(\pi x) - 5 \cdot \dfrac{T_c}{2} \cdot x \cdot \left[\dfrac{\sin(\pi x)}{\pi x} - \cos(\pi x)\right], & \text{for } x < 0; \end{cases}
\end{aligned}
$$

We further differentiate R_g with respect to $\hat{\tau}$, and get:

$$\begin{aligned}\frac{dR_g(x)}{d\hat{\tau}} &= \frac{1}{T_c} \cdot \frac{dR_p(x)}{dx} \\ &= -\frac{5}{2} \cdot T_c \cdot [\pi \sin(\pi x) \mp \pi x \cdot \sin(\pi x)]\end{aligned}$$

which, for θ close to 0 gives:

$$\frac{dR_g(\theta)}{d\hat{\tau}} \approx -\frac{5\pi^2}{2} \cdot \frac{\hat{\tau} - \tau}{T_c} \tag{29}$$

This in turn leads to:

$$\begin{aligned}\hat{\tau}_k - \hat{\tau}_{k-1} &= 2\pi \frac{K_v}{\omega_0^2} \cdot \frac{d}{d\hat{\tau}} R_g(\hat{\tau} - \tau)|_{\hat{\tau}=\hat{\tau}_{k-1}} \\ &= -\frac{5\pi^3}{\omega_0^2 \cdot T_c} \cdot K_v \cdot (\hat{\tau}_{k-1} - \tau) \\ &= A \cdot (\tau - \hat{\tau}_{k-1}) \end{aligned} \tag{30}$$

where $A \triangleq \dfrac{5\pi^3}{\omega_0^2 \cdot T_c} \cdot K_v$. Again, the condition of convergence imposes that $|1 - A| < 1$, therefore an upper limit on A, or, equivalently, on K_v.

4 Block-Level Implementation and Simulations

4.1 Circuit description

Based on the results obtained in the previous chapters, we used Simulink® to build a test-bench for the synchronizer. The simulation of the source, channel, receiver matched filters and synchronizer is described in Section 4.1, and the simulation results are discussed in Section 4.2. Two practical implementations are built and simulated in Section 4.3.

We start with the circuit that simulates the source of shift-quasi orthogonal waveforms and the channel; it uses a δ sequence to periodically excite a filter having the impulse response equal to the sequence of the three codewords (within the block called "3 Waveform Generator"), i.e. $g_0(t), g_1(t)$ and $g_2(t)$, where[5]:

$$g_0(t) = \begin{cases} -\sin\left(\frac{\omega_c}{2}t\right), & \text{for } t \in [0, 3T_c); \\ \sin\left(\frac{\omega_c}{2}t\right), & \text{for } t \in [3T_c, 5T_c]; \\ 0, & \text{else.} \end{cases}$$

corresponding to the codeword "10110";

$$g_1(t) = \begin{cases} -\sin\left(\frac{\omega_c}{2}t\right), & \text{for } t \in [0, 2T_c); \\ \sin\left(\frac{\omega_c}{2}t\right), & \text{for } t \in [2T_c, 3T_c); \\ -\sin\left(\frac{\omega_c}{2}t\right), & \text{for } t \in [3T_c, 5T_c]; \\ 0, & \text{else.} \end{cases}$$

[5] A time unit of 1ms was chosen for the chip period, T_c. The time interval between the δ pulses is then 15ms (5 chip periods/bit$\times$ 3 bits/sequence).

corresponding to the codeword "10001"; and

$$g_2(t) = \begin{cases} -\sin\left(\frac{\omega_c}{2}t\right), & \text{for } t \in [0, T_c); \\ \sin\left(\frac{\omega_c}{2}t\right), & \text{for } t \in [T_c, 2T_c); \\ -\sin\left(\frac{\omega_c}{2}t\right), & \text{for } t \in [2T_c, 3T_c); \\ \sin\left(\frac{\omega_c}{2}t\right), & \text{for } t \in [3T_c, 5T_c]; \\ 0, & \text{else.} \end{cases}$$

corresponding to the codeword "11110".

Two variable delay elements and two variable gain elements were used to generate two delayed and attenuated replicas of the initial waveform, all three being passed through a summing element. To this we further added white Gaussian noise resulting in the signal being fed to the synchronizer.

On the synchronizer side, three filters matched to the shift-quasi orthogonal waveforms used were built at the input, having the general expressions: $g_i^{'}(t) = g_i(5 \cdot T_c - t)$, where $i = 0, 1, 2$, followed by the rest of the building blocks of the synchronizer (Figure 14). Except for the gain and the derivative blocks, all the other had to be built from primitives, since they either do not exist as such in the standard libraries of Simulink®, or the usage of their model is restricted to certain specific applications only, as in the case of the D-flip-flops used to build the timing counter[6] (building details of the filters, timing

[6] We submitted the D-flip-flop and sample-and-hold models we built for this project (and their test circuits) to the support staff of Matlab® at The MathWorks, Inc.. They were accepted and are now retrievable via anonymous FTP from **ftp.mathworks.com**, under the directories **pub/contrib/simulink/dff** and **pub/contrib/simulink/sh**, respectively.

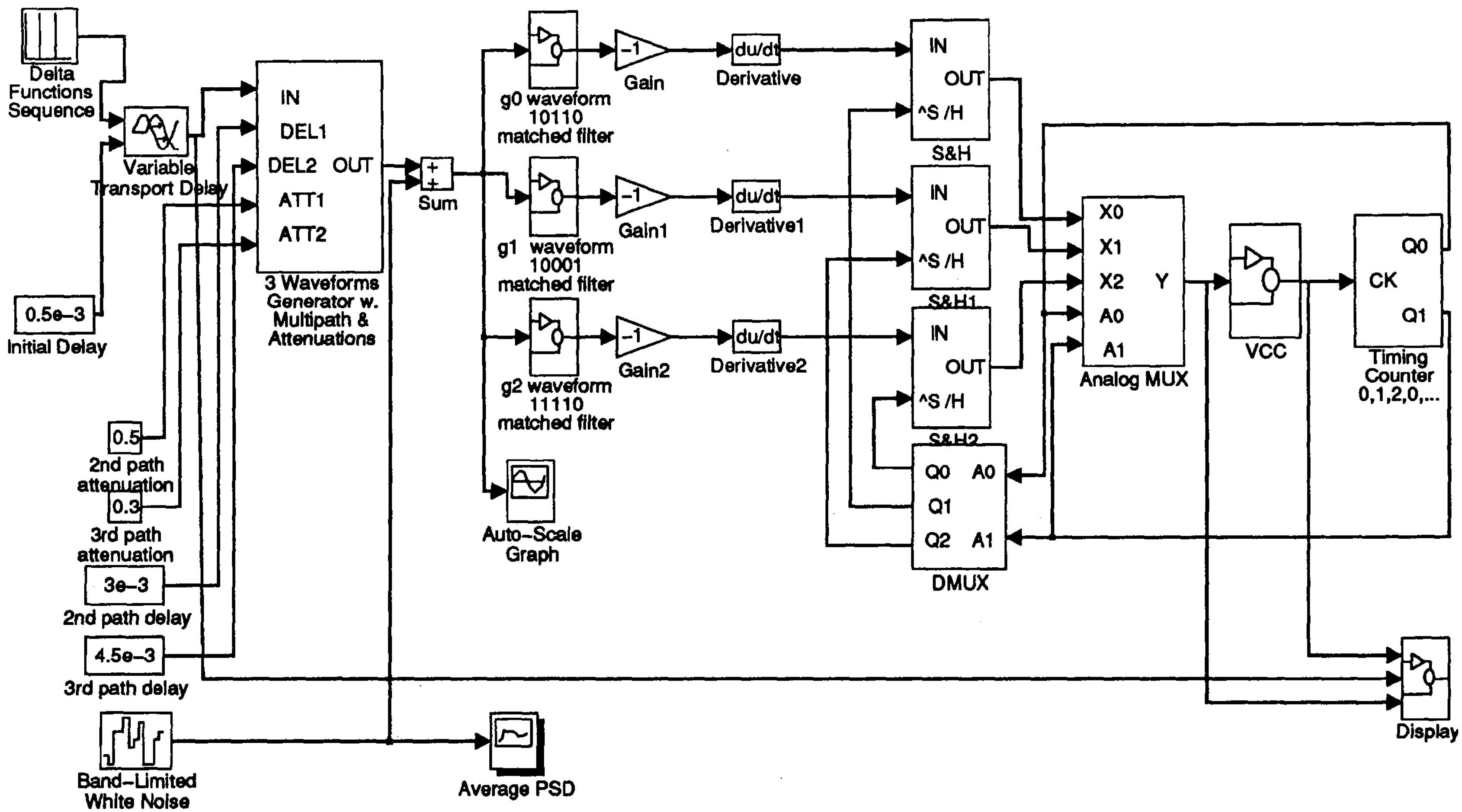

Figure 14: Testbench for Synchronizer

counter and sample-and-hold circuits are given in appendix A). The timing counter, activated by the raising edges of the clock pulses generated by the VCC, counts cyclically $(0, 1, 2, 0, \ldots)$, changing its address with every pulse. Its output is fed to a demultiplexer (DMUX) which will activate the sample-and-hold circuit that holds the proper VCC control voltage at the beginning of each symbol period.

4.2 Simulation results

The waveform at the input of the channel, being the result of a sequence of logical zeros (a sequence of ones in bipolar logic), spread and shaped by half-sine chips, is plotted in Figure 15. For comparison purposes, in Figure 16 is plotted the waveform corrupted by multipath (severe ISI produced by two replicas[7]), and in Figure 17 is plotted the waveform at the input of the receiver (the channel adds white Gaussian noise).

The system is analyzed during the training period, when the sequence of incoming bits of data is known. The VCC block incorporates a frequency divider, which divides the chip frequency clock by 5, outputting a clock with a period of 5 chip durations, i.e. 5ms.

We use three configurations to test the synchronizer (in all three the initial delay between the direct incoming wave and the locally generated clock being $T_c/2$):

1. direct wave, one replica attenuated by 50% and delayed by $3T_c$, the second attenuated by 70% and delayed by $4.5T_c$ (both with respect to the direct wave); the

---

[7] One replica unattenuated and delayed by $5T_c$ with respect to the direct wave, the second attenuated 20% and delayed by $9T_c$.

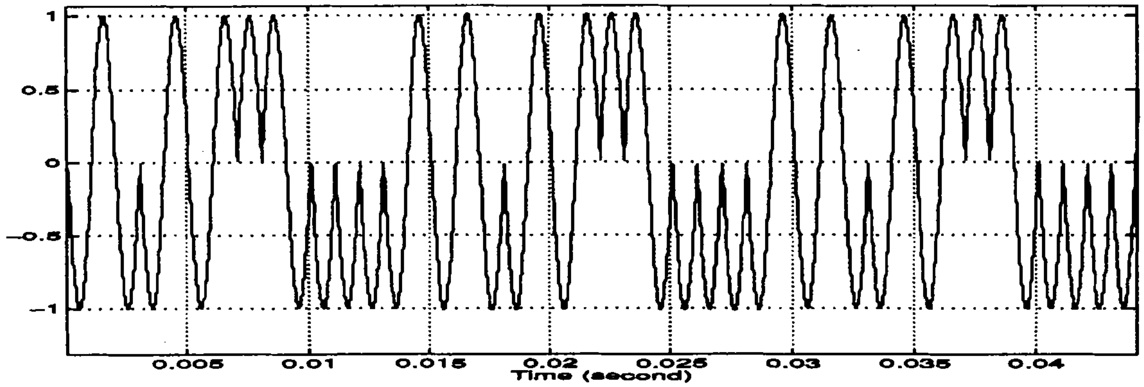

Figure 15: The wave at the input of the channel

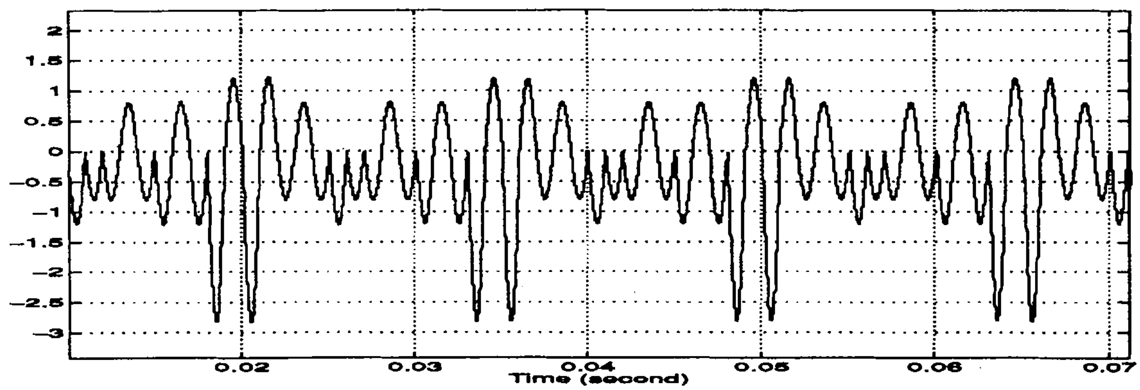

Figure 16: The direct wave and its replicas

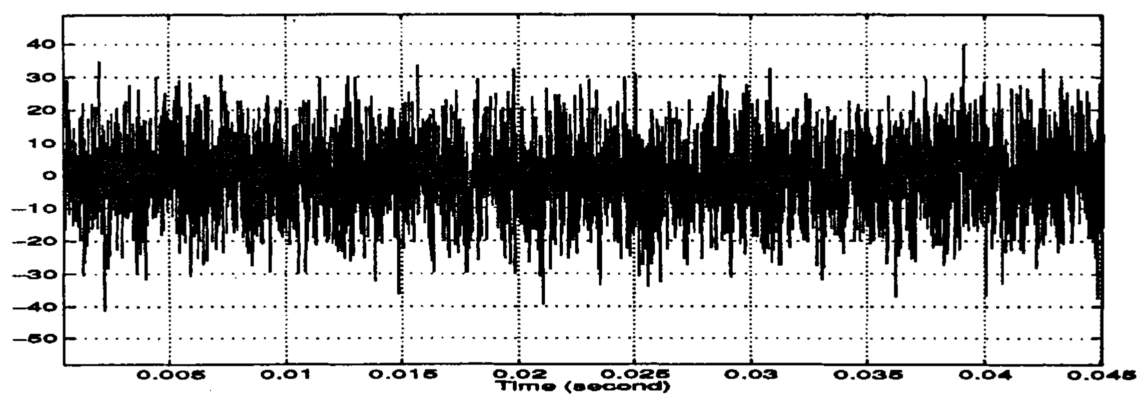

Figure 17: The wave at the input of the receiver

signal-to-noise ratio (SNR) is 10dB at the outputs of the matched filters[8]. The results of the simulation are plotted in Figure 18, where it is seen that synchronism is acquired after roughly 12 bits.

2. same parameters as before, but with an SNR of 3dB at the outputs of the matched filters. Now synchronism is acquired after 15 bits, as seen in Figure 19.

3. direct wave, one replica unattenuated and delayed by $5T_c$, the second attenuated 20% and delayed by $9T_c$ with respect to the direct path; the SNR is still of 3dB at the outputs of the matched filters. It is seen that synchronism is acquired after 18 bits (the synchronization process is plotted in Figure 20).

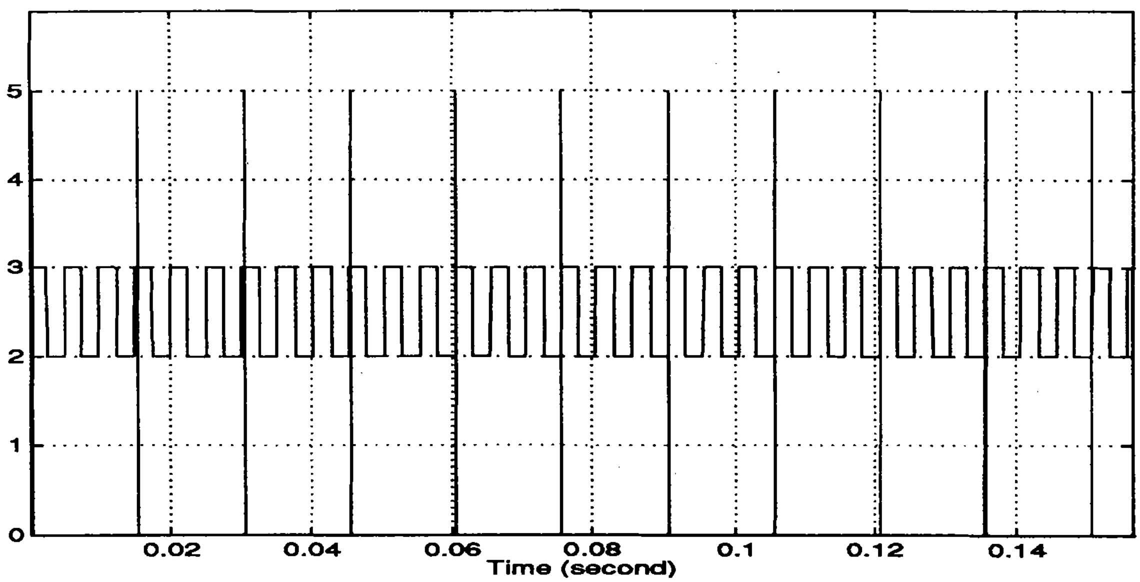

Figure 18: The synchronization process — case 1

As expected, the tracking process is slower and the jitter more pronounced in case 2

8 The SNR considered takes into account only the white Gaussian noise; if we consider the signal-to-interference ratio (SIR) as well, where the interference is due to the multipath effects, i.e. autocorrelations with delayed replicas and cross-correlations, the total S/(N+I) ratio is sensibly smaller.

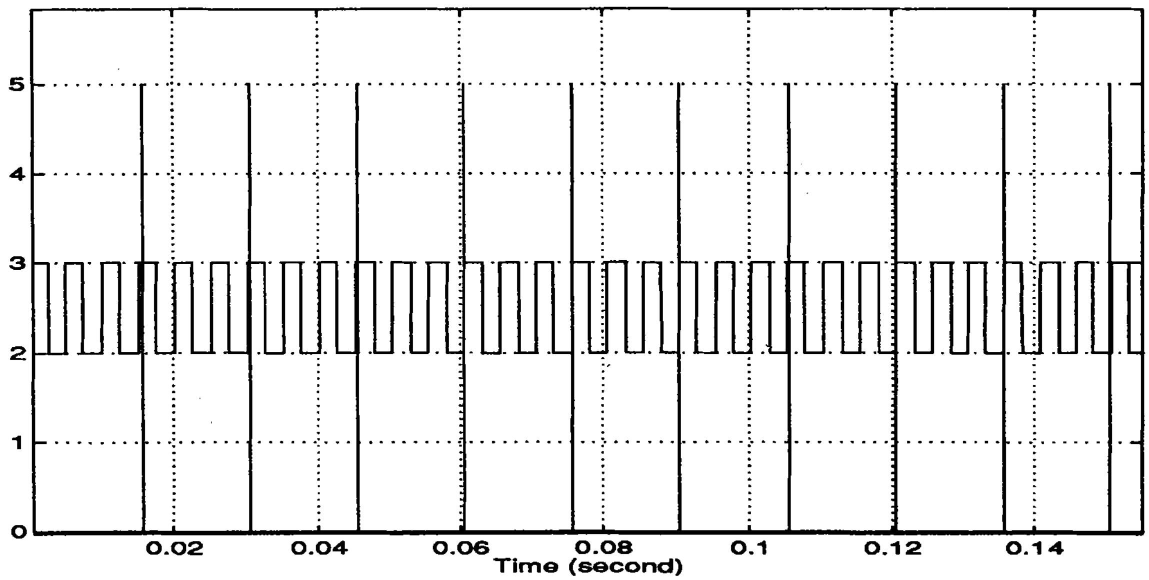

Figure 19: The synchronization process — case 2

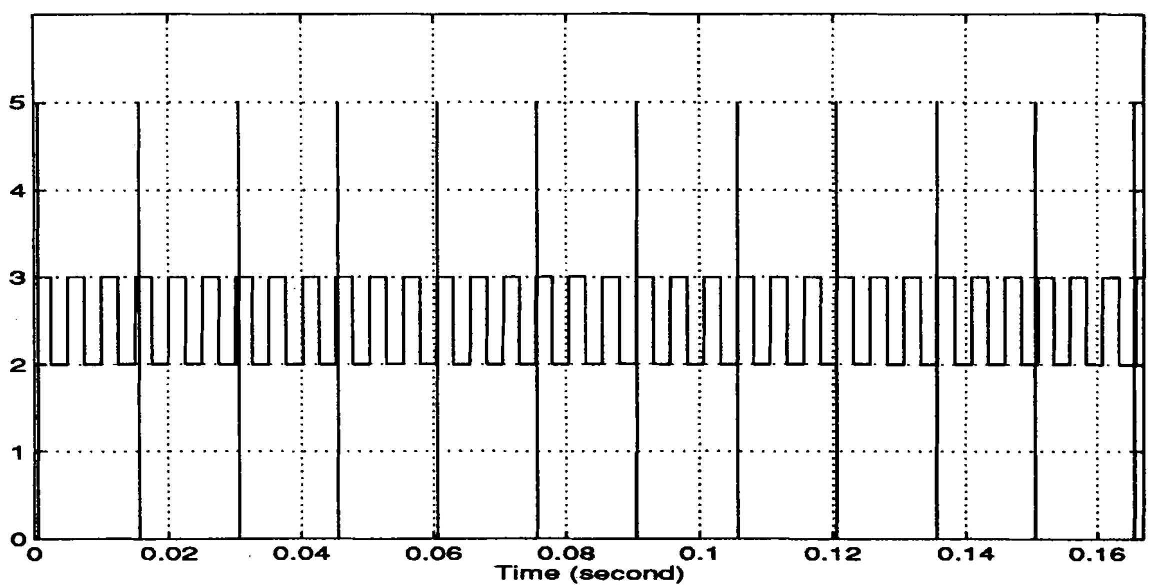

Figure 20: The synchronization process — case 3

versus case 1, due to the increase by a factor of 5 in the power spectral density of the noise at the input of the receiver. In case 3 tracking takes even longer (and the jitter becomes worse) when compared with case 2, this time due to an increase by a factor of about 4.8 in the power of the multipath induced interference at the input of the receiver.

4.3 Implementation without carrier phase tracking

Two practical implementations of the synchronizer are considered here; their advantage stems from the fact that they do not require knowledge of the carrier phase at the receiver (their functionality is based on signals that are not influenced by the carrier phase), and therefore the receiver structure is considerably less complex; they both can be used with differentially coherent demodulation.

In the first scheme, shown in Figure 21, the outputs of the matched filters are passed through absolute value blocks, and the resulting envelope of the signal, which is not influenced by the phase, is fed to the synchronizer; it is seen from Figure 22 that, for an SNR of 3dB, synchronism is reached after approximately 18 bits. It is interesting to observe that, for this scheme, the voltage controlling the VCC has, in the absence of noise, an expression very similar with the one derived in 2.4 for the case where the carrier phase was known; indeed, by using for the incoming wave (baseband representation) the expression

$$z(t) = \exp(\jmath\theta) \sum_{i=0}^{K-1} b_i g_{[i]_M}(t - iT - \tau) \tag{31}$$

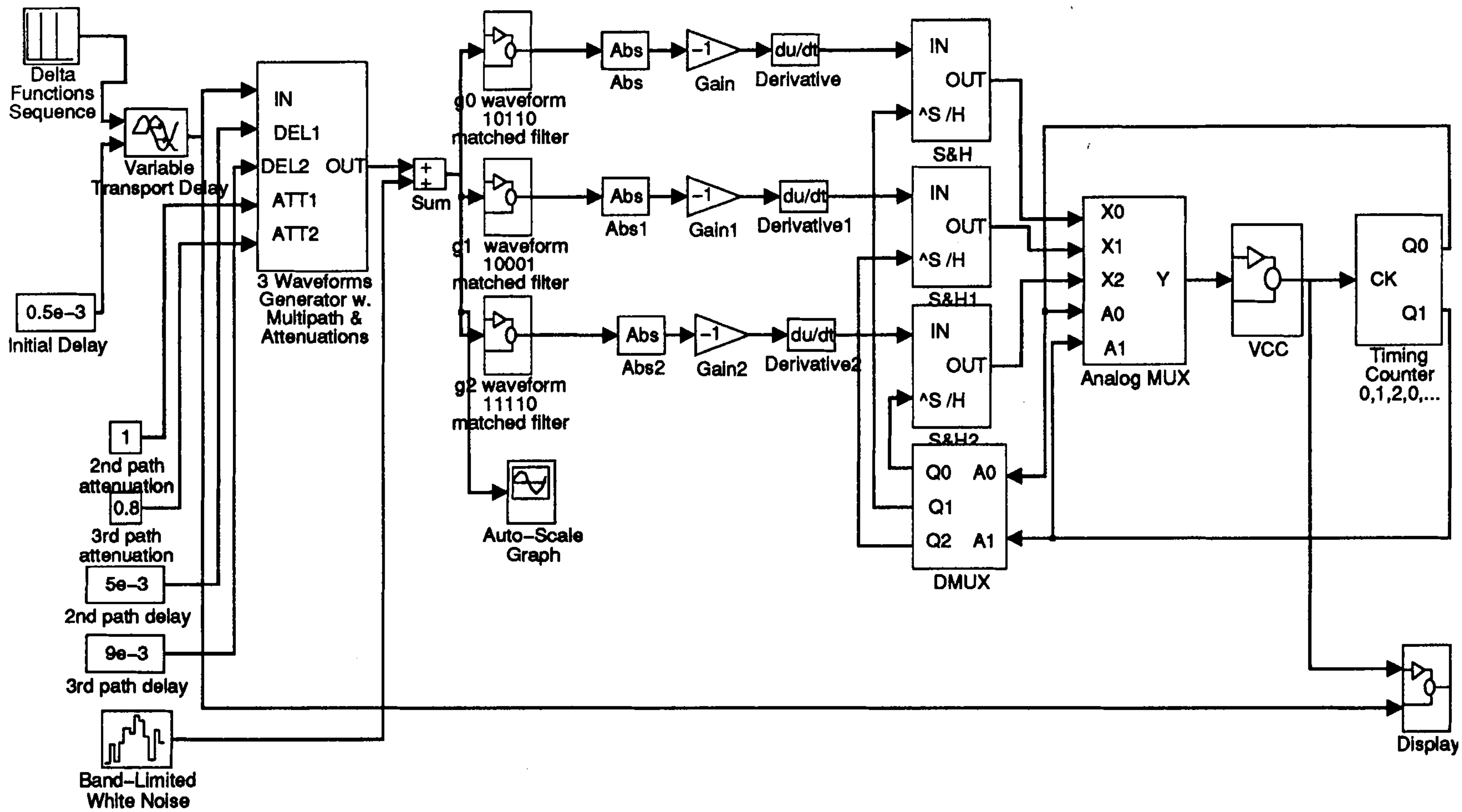

Figure 21: The synchronizer — envelope case

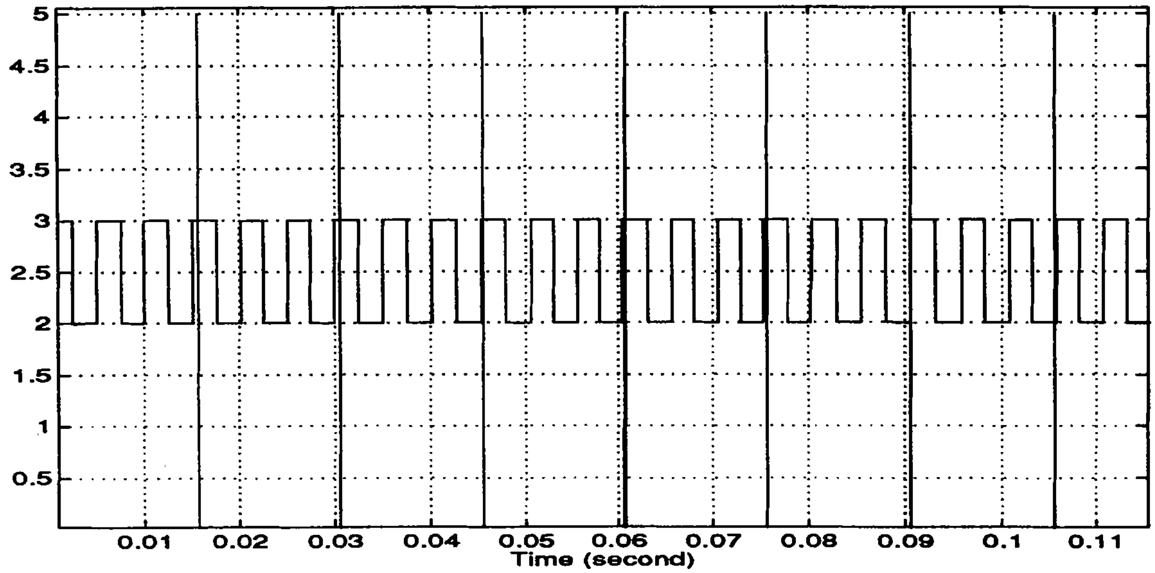

Figure 22: The synchronization process — envelope case

where θ is the carrier phase shift, it is very easy to show that, since $|\exp(\jmath\theta)| = 1$,

$$v_k \approx -b_k|b_k| \cdot \frac{d}{d\hat{\tau}}|R_{g_{[k]_M}}(\hat{\tau} - \tau)|$$

But, for $(\hat{\tau} - \tau) \in \left[-\frac{T_c}{2}, \frac{T_c}{2}\right]$, the waveforms autocorrelations are positive; also, since $b_k = 1$ during the training period, $|b_k| = b_k$. Thus, we would expect the two schemes to have similar performances in terms of tracking. The simulation performed confirms this expectation.

In the second scheme, shown in Figure 24, the input to the synchronizer is taken at the output of the differential coherent detectors, since the signals at those points are not influenced by the carrier phase. In this case, the VCC controlling voltage has the expression

$$v_k = -b_{k-1}b_k\frac{d}{d\hat{\tau}}\mathrm{Re}\Big\{\int_{-\infty}^{\infty} z^*(t)g_{[k-1]_M}(t - kT - \hat{\tau})dt \cdot \int_{-\infty}^{\infty} z(t-T)g^*_{[k]_M}[t-(k+1)T-\hat{\tau}]dt\Big\}$$

If again we analyze in the absence of noise, with $z(t)$ having the expression of Eq. 31, then, after considerations similar to those made for the previous case, we get

$$v_k = -b_{k-1} b_k \frac{d}{d\hat{\tau}} \left[R_{g_{[k-1]_M}}(\hat{\tau} - \tau) \cdot R_{g_{[k]_M}}(\hat{\tau} - \tau) \right]$$

This expression hints to the fact that the convergence rate of this scheme should be higher; the simulation, performed under identical conditions as in the previous case, shows indeed that the scheme tracks the incoming wave after 12–15 bits (Figure 23). In

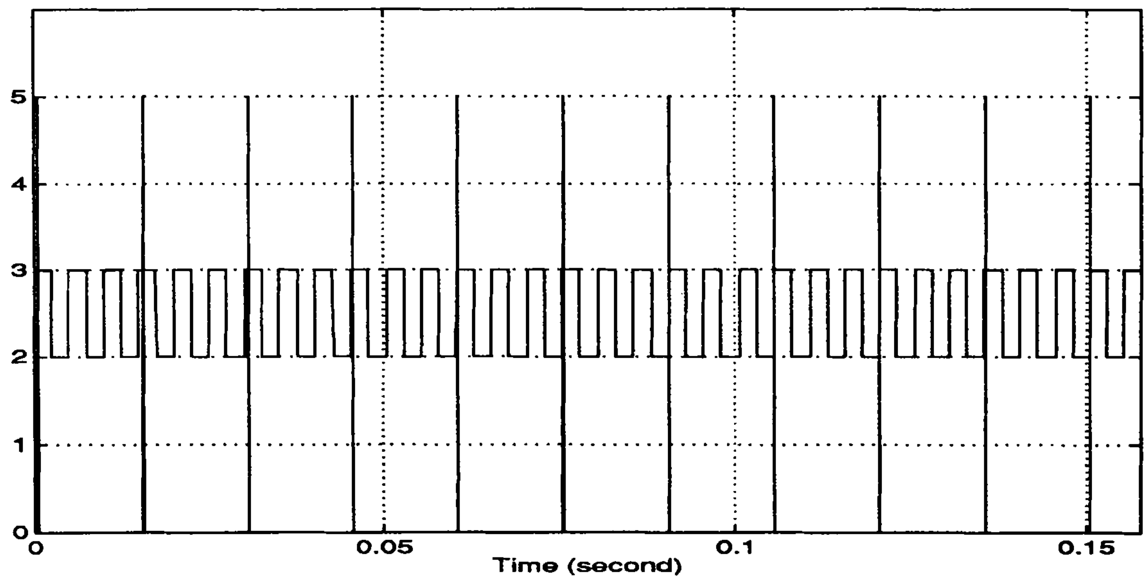

Figure 23: The synchronization process — multiplier case

the presence of noise and interference, due to the nonlinear operations performed on the incoming wave, it is very hard to analyze mathematically the performance of the two circuits discussed here. However, with the aid of computer simulations, we have been able to assess their performance in terms of speed of acquisition and jitter for specific conditions.

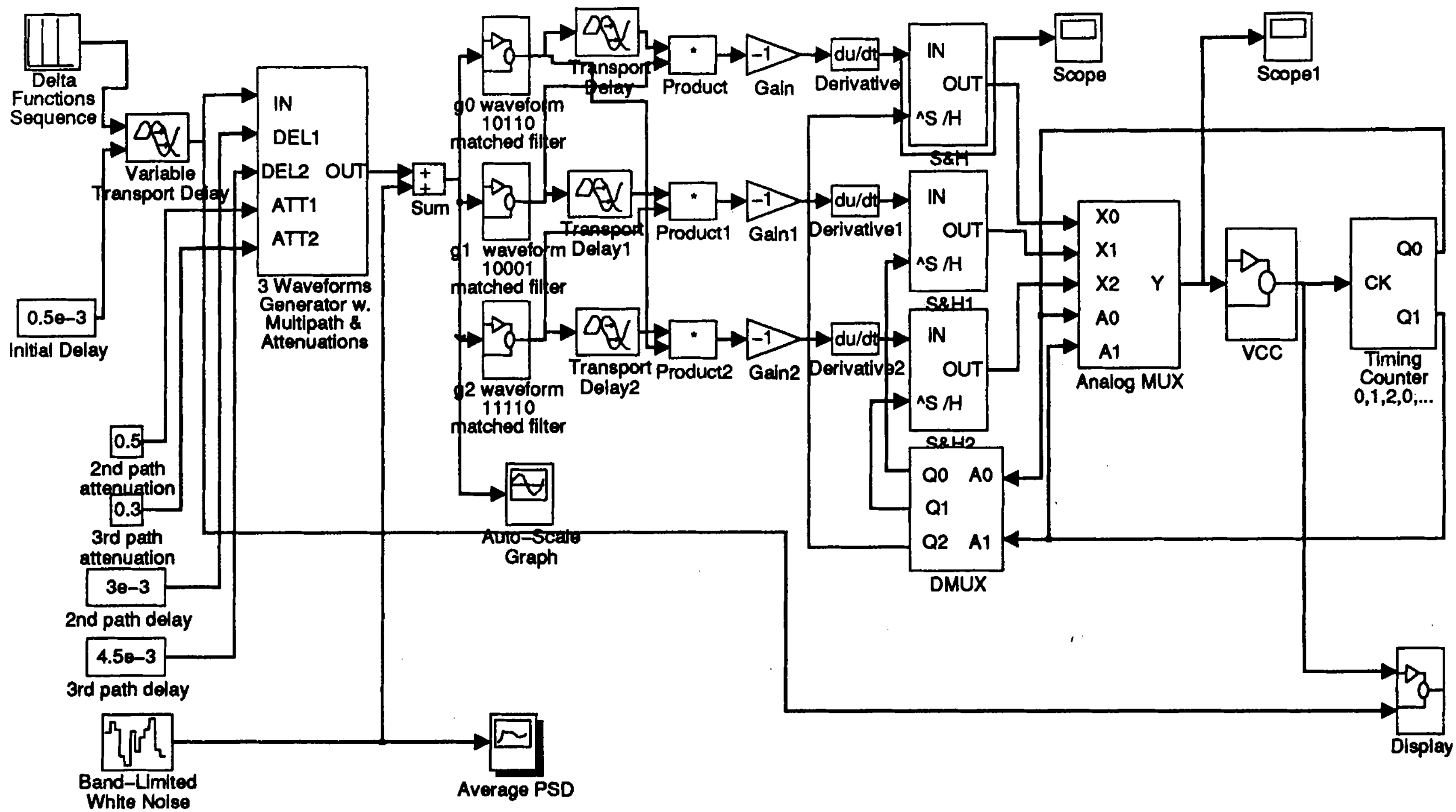

Figure 24: The synchronizer — multiplier case

5 Summary and Conclusions

In this project we proposed a method for chip timing recovery for indoor wireless networks employing Commutation Signalling. First it is assumed that the carrier phase in known at the receiver, so that the synchronization problem becomes mathematically tractable. The MAP criterion is used to derive a likelihood function, which indicates, by its point of maximum, the time of synchronization. This likelihood function is then approximated with a simplified expression (easier to implement in practice) and its derivative is used to form the control voltage which drives a voltage controlled clock (VCC) that provides the local chip timing. Based on this considerations, a synchronizer structure is proposed.

The convergence of the synchronization algorithm has been investigated, using three codewords of five chips each, that present very good orthogonality properties. We employed as shaping waveforms, in the first case, pulses having raised-cosine spectrum, and in the second case, pulses having half-sine form. In both cases it was shown that the synchronization is acquired, provided the VCC gain is chosen below a certain maximal limit. The simulation results are in concordance with the theoretical ones, showing that the local clock "tracks" the incoming waveform timing even when the SNR and SIR are small; they also show the influence of the noise and of the interference on the synchronism, by the presence of a jitter which is more pronounced in the cases of small SNR and SIR.

Two schemes that do not require carrier phase knowledge, and therefore can be used when the carrier phase is unknown, are proposed and simulated. Their mathematical analyses being extremely difficult, the performance of these schemes can be evaluated only by computer simulations.

The schemes proposed have been studied by simulation. However, a governing principle in the design has been to use building blocks that are easily implementable at high data rates. For such rates (chip rate of 200Mb/s), the proposed implementation for the matched filters would be surface-acoustic-wave (SAW) technology. The remaining parts of the synchronizer are fully compatible with the BiCMOS 0.8μ technology. We have had experience with the implementation of similar circuitry in this technology, and foresee no significant problems.

To achieve better performance, a summing window (memory) should be used to average over the result of several correlations before forming the control voltage, therefore attenuating the effects of the random factors (noise and cross-correlations). This could be a subject for further research. Also, a quantitative measure of the effect of noise and cross-correlations has not been treated in this project, and could also be a subject for further study.

A Details of the Building Blocks

The appendix presents details of some of the blocks used to build the system. The block used to generate the sequence of three waveforms, called "3 Waveforms Generator with Multipath and Attenuations", is plotted in Figure 26; it includes two transport delays and two attenuators, which will create replicas of the direct waveform and a filter called "Generating filter" used to generate the three quasi-shift orthogonal waveforms. Details of the filter are plotted in Figure 27. The counter's building details are plotted in Figure 28.

In order to explain the way the "Generating filter" was built, we start from its impulse response, which should be of the form:

$$h(t) = \begin{cases} g_0(t), & \text{for } t \in [0, 5T_c); \\ g_1(t), & \text{for } t \in [5T_c, 10T_c); \\ g_2(t), & \text{for } t \in [10T_c, 15T_c); \end{cases}$$

This impulse response is plotted in Figure 25:

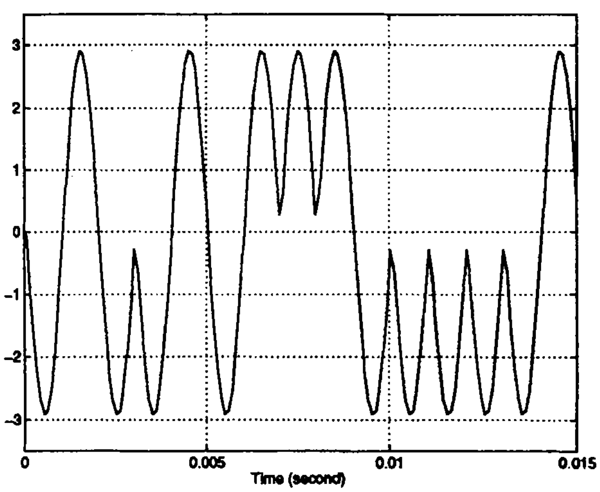

Figure 25: 3 Waveform Generator Filter Impulse Response

This filter generates, upon being excited by a sequence of δ pulses spaced by $15T_c$, the sequence of three shift-quasi-orthogonal waveforms, corresponding to the sequence of bits '10110,10001,11110'.

To build the filter in Simulink®, the starting point is provided by the Laplace transforms of the three waveforms:

$$G_0(s) = \frac{\omega_c'}{\omega_c'^2 + s^2} \cdot [\exp(-s \cdot 5T_c) - 2\exp(-s \cdot 3T_c) - 1]$$

$$G_1(s) = \frac{\omega_c'}{\omega_c'^2 + s^2} \cdot [-\exp(-s \cdot 5T_c) + 2\exp(-s \cdot 3T_c) + 2\exp(-s \cdot 2T_c) - 1]$$

$$G_2(s) = \frac{\omega_c'}{\omega_c'^2 + s^2} \cdot [\exp(-s \cdot 5T_c) - 2\exp(-s \cdot 3T_c) - 2\exp(-s \cdot 2T_c) - 2\exp(-s \cdot T_c) - 1]$$

where $\omega_c' = \frac{\omega_c}{2}$ and s is the complex variable. The exponentials, being the Laplace transforms of delays, were modeled in Simulink® with transport delays, and the factor $\frac{\omega_c'}{\omega_c'^2 + s^2}$ with a linear transfer function block. The same technique was used to build the matched filters, which have the transfer functions:

$$G_0'(s) = \frac{\omega_c'}{\omega_c'^2 + s^2} \cdot [-\exp(-s \cdot 5T_c) - 2\exp(-s \cdot 2T_c) + 1]$$

$$G_1'(s) = \frac{\omega_c'}{\omega_c'^2 + s^2} \cdot [-\exp(-s \cdot 5T_c) + 2\exp(-s \cdot 3T_c) + 2\exp(-s \cdot 2T_c) - 1]$$

$$G_2'(s) = \frac{\omega_c'}{\omega_c'^2 + s^2} \cdot [-\exp(-s \cdot 5T_c) - 2\exp(-s \cdot 4T_c) - 2\exp(-s \cdot 3T_c) - 2\exp(-s \cdot 2T_c) + 1]$$

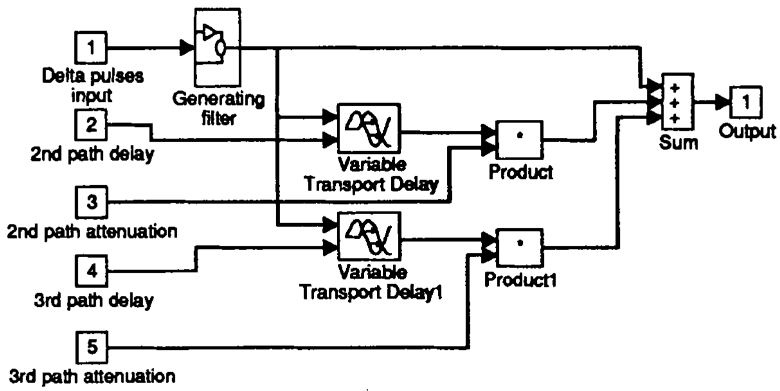

Figure 26: 3 Waveform Generator – detail

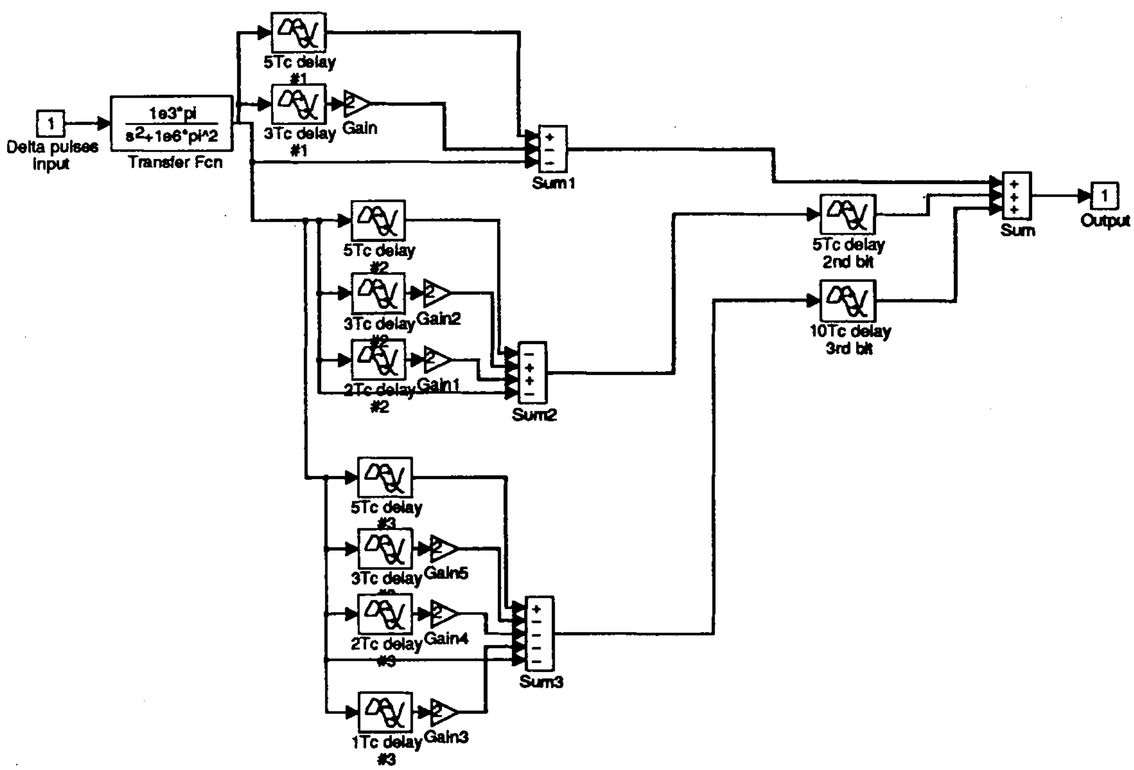

Figure 27: 3 Waveform Generator – Generating filter

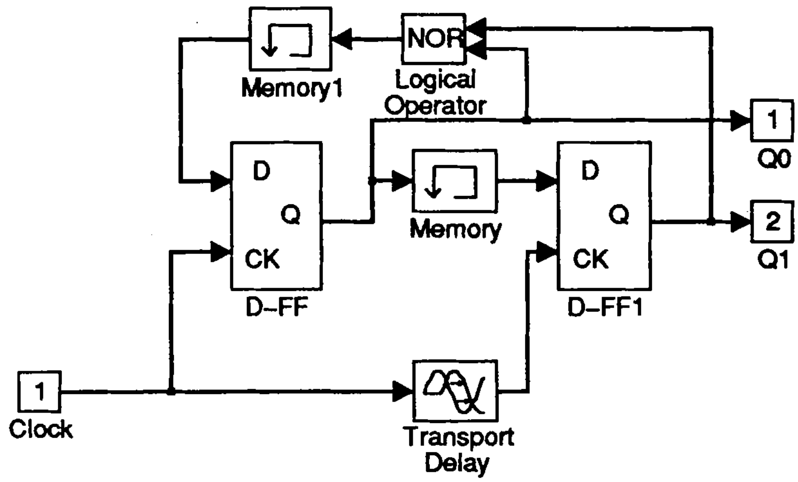

Figure 28: Counter block – detail

References

[1] W. C. Lindsey and M. K. Simon,*Telecommunication Systems Engineering.* New York: Dover Publications Inc., 1973.

[2] K. Feher, editor, *Digital Communications — Satellite/Earth Station Engineering.* Englewood Cliffs, N.J.: Prentice Hall, 1981; Chapter 7: L. E. Franks, *Synchronization Subsystems: Analysis and Design.*

[3] B. Sklar,*Digital Communications — Fundamentals and Applications.* Englewood Cliffs, N.J.: Prentice Hall, 1988.

[4] J. G. Proakis, *Digital Communications.* New York: McGraw-Hill Book Co., 1989, 2nd ed.

[5] G. L. Turin, "Introduction to Spread-Spectrum Antimultipath Techniques and Their Application to Urban Digital Radio", *IEEE Proc.*, vol. 68, pp. 328 – 353, Mar. 1980.

[6] G. L. Turin, "Commutation Signaling — An Antimultipath Technique", *IEEE JSAC*, vol. SAC-2, pp. 548–562, July 1984.

[7] K. M. Cheung, "Generalized Likelihood Commutation Signaling for Indoor Communications", *M.Eng. Thesis, Dept. of Electrical Engineering, McGill University*, Oct. 1992.

[8] P. V. Krishnamurthy, M. Alles, Y. Qian and H. Leib, "An Anti-Multipath Transmission Technique for Indoor Wireless Communications", *Dept. of Electrical Engi-*

neering, Telecommunications and Signal Processing Laboratory, McGill University, Oct. 1994 (unpublished).

[9] R. L. Pickholtz, D. L. Schilling and L. B. Milstein, "Theory of Spread-Spectrum Communications — A Tutorial", *IEEE Transactions on Communications*, vol. COM-30, pp. 855–884, May 1982.

[10] R. L. Pickholtz, D. L. Schilling and L. B. Milstein, "Spread-Spectrum for Mobile Communications", *IEEE Transactions on Vehicular Technology*, vol. 40, May 1991.

[11] L. E. Franks and J. P. Bubrouski, "Statistical Properties of Timing Jitter in a PAM Timing Recovery Scheme", *IEEE Transactions on Communications*, vol. COM-22, pp. 913–920, July 1974.

[12] L. E. Franks, "Carrier and Bit Synchronization in Data Communication — A Tutorial Review", *IEEE Transactions on Communications*, vol. COM-28, pp. 1107–1121, Aug. 1980.

[13] M. H. Meyers and L. E. Franks, "Joint Carrier Phase and Symbol Timing Recovery for PAM Systems", *IEEE Transactions on Communications*, vol. COM-28, pp. 1121–1129, Aug. 1980.

[14] R. D. Gitlin and J. Salz, "Timing Recovery in PAM Systems", *Bell System Technical Journal*, vol. 50, pp. 1645–1669, May–June 1971.