



Objective

This laboratory will cover basic receiver structures and implementations. We will also show how to construct a series of orthonormal basis functions that can be combined to produce a wide range of signal waveforms. Then, we will study a specific digital transceiver implementation based on the multicarrier transmission concept called orthogonal frequency division multiplexing (OFDM). In the experimental part of this laboratory, you will implement two different receiver structures and then observe their performance during over the air transmission. This is followed by a Simulink implementation of an OFDM communication system. Finally, the open-ended design problem will focus on bi-directional communication.

Contents

1	Theoretical Preparation	3
1.1	Gram-Schmidt Orthogonalization	3
1.2	Optimal Detection	3
1.2.1	Maximum Likelihood Detection in an AWGN Channel	4
1.2.2	Maximum A Posteriori Detection	5
1.3	Basic Receiver Realizations	6
1.3.1	Matched Filter Realization	6
1.3.2	Correlator Realization	8
1.4	Multicarrier Transmission	8
1.4.1	Orthogonal Frequency Division Multiplexing	9
1.4.2	Dispersive Channel Environment	11
1.4.3	OFDM with Cyclic Prefix	11
1.5	Suggested Readings	13
1.6	Problems	14
2	Software Implementation	16
2.1	Observation Vector Construction	16
2.2	Maximum-likelihood Decoder Implementation	17
2.3	Correlator Realization of a Receiver in Simulink	18
2.4	Multicarrier Modulation	20
2.4.1	MATLAB Design of Multicarrier Transmission	20
2.4.2	Simulink Design of Multicarrier Transmission	20

- 3 USRP2 Hardware Implementation 23**
 - 3.1 Eye Diagram 23
 - 3.1.1 Discrete-Time Eye Diagram Scope 23
 - 3.2 Matched Filter Observation 23

- 4 Open-ended Design Problem: Duplex Communication 26**
 - 4.1 Duplex Communication 26
 - 4.2 Half-duplex 26
 - 4.3 Time Division Duplexing 26
 - 4.4 Hints 27

- 5 Lab Report Preparation & Submission Instructions 29**

1 Theoretical Preparation

Signals can be represented as either waveforms or vectors. Waveform representations are defined by the Fourier series of a signal, or the sum of sines and cosines that make a particular shape. As a vector, a signal is represented as series of orthonormal vectors. This part of the lab will show how to derive these vectors and their basis functions. It will then show two receiver designs and their implementations.

1.1 Gram-Schmidt Orthogonalization

In mathematics, particularly linear algebra and numerical analysis, the Gram-Schmidt process is a method for creating an orthonormal set of vectors in an inner product space such as the Euclidean space R^n . The Gram-Schmidt process takes a finite, linearly independent set $S = \{S_1(t), S_2(t), \dots, S_M(t)\}$ and generates an orthogonal set $S' = \{\phi_1(t), \phi_2(t), \dots, \phi_i(t)\}$ that spans the same subspace of R^n as S .

To derive a set of orthogonal basis functions $\phi_1(t), \phi_2(t), \dots, \phi_i(t)$ from set of energy signals denoted by $S_1(t), S_2(t), \dots, S_M(t)$ we will use the following functions:

$$g_i(t) = S_i(t) - \sum_{j=1}^{i-1} S_{ij}\phi_j(t), \quad (1)$$

where:

$$S_{ij} = \int_0^T S_i(t)\phi_j dt, j = 1, 2, \dots, i - 1. \quad (2)$$

Using these two functions, we define our basis function ϕ_i as:

$$\phi_i = \frac{g_i(t)}{\sqrt{\int_0^T g_i^2(t)dt}}, i = 1, 2, \dots, N. \quad (3)$$

In Equation (1), we are developing our basis functions by subtracting the projections of the other waveforms on S_i from S_i . Note that S_{ij} is the projection of S_j onto S_i and is defined in Equation (2), while Equation (3) is normalizing g_i by its energy. Consequently, in several circumstances it may be much simpler to compute this graphically than to work out the integral. Note that the order in which you conduct the Gram-Schmidt process with respect to the selection of the waveforms S_i will yield different sets of orthonormal basis functions.

1.2 Optimal Detection

Detection theory, or signal detection theory, is used in order to discern between signal and noise. Using this theory, we can explain how changing the decision threshold will affect the ability to discern between two or more scenarios, often exposing how adapted the system is to the task, purpose or goal at which it is aimed.

When a signal is received, the probability of correct detection:

$$P(\text{correct}) = \text{MAX} \rightarrow P(\hat{m}_i = m_i), \quad (4)$$

and the probability of an error:

$$P(\text{error}) = \text{MIN} \rightarrow P(\hat{m}_i \neq m_i). \quad (5)$$

Recall Bayes' Rule which gives the conditional probability of an event and is stated by:

$$P(A|B)P(B) = P(A \cap B) = P(B|A)P(A)$$

when all components are expressed as probabilities, or:

$$p(x, y) = p(x|y)p(y) = p(y|x)p(x)$$

when all components are expressed as PDFs, or:

$$P(S_i|r = \underline{p}) = p(\underline{p}) = p(\underline{p}|S_i)P(S_i)$$

when all components are expressed as a mixture. These expression result in the rule:

$$P(S_i|r = \underline{p}) = \frac{p(\underline{p}|S_i) \cdot P(S_i)}{p(\underline{p})}. \quad (6)$$

Recalling the rule for the optimal detector given in Equation (4), using the mixed form of Bayes' Rule, we may define the optimal detector as [1]:

$$\max_{S_i} P(S_i|r = \underline{p}) = \max_{S_i} \frac{p(\underline{p}|S_i) \cdot P(S_i)}{p(\underline{p})}. \quad (7)$$

Since $p(\underline{p})$ does not depend upon S_i , we can simplify the optimal detector to:

$$\max_{S_i} p(\underline{p}|S_i)P(S_i). \quad (8)$$

1.2.1 Maximum Likelihood Detection in an AWGN Channel

Maximum likelihood detection is a popular statistical method employed for fitting a statistical model to data, and for identifying model parameters. In general, for a fixed set of data and underlying probability model, the method of maximum likelihood selects values of the model parameters that produce the distribution that are most likely to have resulted in the observed data (i.e., the parameters that maximize the likelihood function).

To develop the maximum-likelihood detection algorithm, let us consider an example. The signal transmitted in each signal interval is binary PAM. Hence, there are two possible transmitted signals corresponding to the signal points $s_1 = -s_2 = \sqrt{\varepsilon_b}$, where ε_b is the energy per bit. The output of the matched-filter or correlation demodulator for binary PAM in the k th signal interval may be express as:

$$r_k = \pm\sqrt{\varepsilon_b} + n_k, \quad (9)$$

where n_k is a zero-mean Gaussian random variable with variance $\sigma_n^2 = N_0/2$. Consequently, the conditional PDFs for the two possible transmitted signals are:

$$\begin{aligned} p(r_k|s_1) &= \frac{1}{\sqrt{2\pi}\sigma_n} \exp\left[-\frac{(r_k - \sqrt{\varepsilon_b})^2}{2\sigma_n^2}\right], \\ p(r_k|s_2) &= \frac{1}{\sqrt{2\pi}\sigma_n} \exp\left[-\frac{(r_k + \sqrt{\varepsilon_b})^2}{2\sigma_n^2}\right]. \end{aligned} \quad (10)$$

Suppose we observe the sequence of matched-filter outputs r_1, r_2, \dots, r_k . Since the channel noise is assumed to be white and Gaussian, it follows that $E(n_k n_j) = 0, k \neq j$. Hence, the noise sequence n_1, n_2, \dots, n_k is also white. Consequently, for any given transmitted sequence $s^{(m)}$, the joint PDF of r_1, r_2, \dots, r_k may be expressed as a product of K marginal PDFs, i.e.:

$$\begin{aligned} p(r_1, r_2, \dots, r_K | s^{(m)}) &= \prod_{k=1}^K p(r_k | s_k^{(m)}) \\ &= \prod_{k=1}^K \frac{1}{\sqrt{2\pi}\sigma_n} \exp \left[-\frac{(r_k - s_k^{(m)})^2}{2\sigma_n^2} \right] \\ &= \left(\frac{1}{\sqrt{2\pi}\sigma_n} \right)^K \exp \left[-\sum_{k=1}^K \frac{(r_k - s_k^{(m)})^2}{2\sigma_n^2} \right], \end{aligned} \quad (11)$$

where either $s_k = \sqrt{\varepsilon_b}$ or $s_k = -\sqrt{\varepsilon_b}$. Then, given the received sequence r_1, r_2, \dots, r_K at output of the matched filter or correlation demodulator, the detector determines the sequence $s^{(m)} = s_1^{(m)}, s_2^{(m)}, \dots, s_K^{(m)}$ that maximizes the conditional PDF $p(r_1, r_2, \dots, r_K | s^{(m)})$. Such a detector is called the maximum-likelihood (ML) detector.

By taking the logarithm of Equation (11) and neglecting the terms that are independent of $p(r_1, r_2, \dots, r_K)$, we find that an equivalent ML detector selects the sequence $s^{(m)}$ that minimizes the Euclidean distance metric, namely:

$$D(r, s^{(m)}) = \sum_{k=1}^K (r_k - s_k^{(m)})^2. \quad (12)$$

For more information about maximum likelihood detection, please refer to Section 5.6 of the course textbook [4].

1.2.2 Maximum A Posteriori Detection

In contrast to the maximum-likelihood detector for detecting the transmitted information, we now describe a detector that makes symbol-by-symbol decisions based on the computation of the maximum a posteriori probability (MAP) for each detected symbol. Hence, this detector is optimum in the sense that it minimize the probability of a symbol error. The detection algorithm that is presented below is due to [3], who developed it as a detection algorithm for channels with intersymbol interference, i.e., channels with memory.

We illustrate the algorithm in the context of detecting a PAM signal with M possible levels. Suppose that it is desired to detect the information symbol transmitted in the k th signal interval, and let r_1, r_2, \dots, r_{k+D} be the observed received sequence, where D is the delay parameter which is chosen to exceed the signal memory, i.e., $D \geq L$, where L is the inherent memory in the signal. On the basis of the received sequence, we compute the posterior probabilities:

$$P(s^{(k)} = A_m | r_{k+D}, r_{k+D-1}, \dots, r_1), \quad (13)$$

for the M possible symbol values, and choose the symbol with the largest probability. Since we have:

$$P(s^{(k)} = A_m | r_{k+D}, r_{k+D-1}, \dots, r_1) = \frac{p(r_{k+D}, \dots, r_1 | s^{(k)} = A_m) P(s^{(k)} = A_m)}{p(r_{k+D}, r_{k+D-1}, \dots, r_1)}, \quad (14)$$

and since the denominator is common for all M probabilities, the MAP criterion is equivalent to choosing the value of $s^{(k)}$ that maximizes the numerator of Equation (14). Thus, the criterion for deciding on the transmitted symbol $s^{(k)}$ is:

$$\tilde{s}^{(k)} = \arg\{\max p(r_{k+D}, \dots, r_1 | s^{(k)} = A_m) P(s^{(k)} = A_m)\}. \quad (15)$$

When the symbols are equally probable, the probability $P(s^{(k)} = A_m)$ may be dropped from the computation.

The algorithm for computing the probabilities in Equation (15) recursively begins with the first symbol $s^{(1)}$. In general, the recursive algorithm for detecting the symbol $s^{(k)}$ is as follows: Upon reception of r_{k+D}, \dots, r_2, r_1 , we compute:

$$\begin{aligned} \tilde{s}^{(k)} &= \arg\{\max p(r_{k+D}, \dots, r_1 | s^{(k)}) P(s^{(k)})\} \\ &= \arg\{\max \sum_{s^{(k+D)}} \cdots \sum_{s^{(k+1)}} p_k(s^{k+D}, \dots, s^{(k+1)}, s^{(k)})\}, \end{aligned} \quad (16)$$

where by definition:

$$\begin{aligned} &p_k(s^{k+D}, \dots, s^{(k+1)}, s^{(k)}) \\ &= p(r_{k+D} | s^{(k+D)}, \dots, s^{(k+D-L)}) P(s^{(k+D)}) \sum_{s^{(k-1)}} p_{k-1}(s^{(k-1+D)}, \dots, s^{(k-1)}). \end{aligned} \quad (17)$$

Thus, the recursive nature of the algorithm is established by the Equations (16) and (17).

1.3 Basic Receiver Realizations

The fundamental challenge of digital communications is recovering what was transmitted after it has passed through a channel and been corrupted by noise. The first receiver structure we will examine is based on filtering the received signal with a static filter that maximizes the SNR of the channel. This receiver structure is ideal because it maximizes SNR. From Laboratory 2, it was shown that this minimizes the bit error rate. The major disadvantage to matched filtering is that it requires a priori knowledge of what was transmitted.

1.3.1 Matched Filter Realization

Often in digital transmissions, data fields are preceded by headers. These headers are often standardized and can be anticipated. If you know what was transmitted (such as the header, for example), you can design your receiver as a filter that exactly matches the transmission pulse shape. This shape maximizes the signal-to-noise ratio (SNR) and yields the best decoding results. When designing a receiver, we are interested in the optimal detection rule, or the receiver that yields the correct result more than any other receiver would.

Now that the basis for our detection rule has been outlined, define a signal $x(t)$ as:

$$x(t) = g(t) + w(t), 0 \leq t \leq T, \quad (18)$$

where $g(t)$ is a pulse signal and $w(t)$ is additive noise with $\mu = 0$ and $PSD = \frac{N_0}{2}$. We are assuming that the receiver knows the possible waveforms that $g(t)$ could take. The function of the receiver is to detect the pulse signal $g(t)$ in an optimum manner, given the received signal $x(t)$.

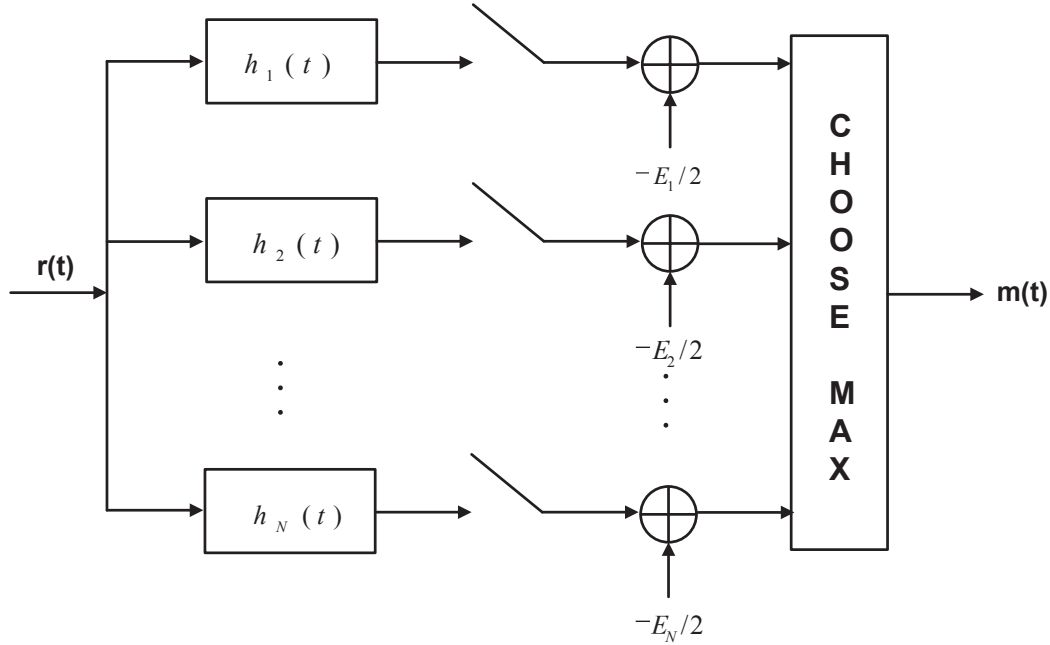


Figure 1: Matched Filter Realization. A matched filter is obtained by correlating a known signal, or template, with an unknown signal to detect the presence of the template in the unknown signal. This is equivalent to convolving the unknown signal with a conjugated time-reversed version of the template (cross-correlation).

The first step is to take the inverse Fourier Transform of $H(f)G(f)$, which is a convolution of $h(t)$ and $g(t)$:

$$g_0(t) = \int_{-\infty}^{\infty} H(f)G(f)e^{j2\pi ft} df. \quad (19)$$

Next, we find the energy:

$$|g_0(t)|^2 = \left| \int_{-\infty}^{\infty} H(f)G(f)e^{j2\pi ft} df \right|^2. \quad (20)$$

Furthermore, the peak pulse SNR is defined as:

$$\eta = \frac{|g_0(T)|^2}{E\{n^2(t)\}}, \quad (21)$$

where $|g_0(t)|^2$ is the instantaneous power of the output signal and $E(n^2)(t)$ is the average power of the output noise. Since $w(t)$ is Gaussian, we know its power spectral density (PSD) to be $\frac{N_0}{2}$. Applying the definition of η , we get:

$$\eta = \frac{\left| \int_{-\infty}^{\infty} H(f)G(f)e^{j2\pi fT} df \right|^2}{\frac{N_0}{2} \int_{-\infty}^{\infty} |H(f)|^2 df}. \quad (22)$$

We see that we need to solve for $H(f)$ such that it yields the largest possible η . Applying Schwarz's Inequality, it can be shown that $H_{opt} = kG^*(f)e^{-j2\pi fT}$ which is equal to:

$$H_{opt} = kg(T - t). \quad (23)$$

This result yields insight into the term ‘‘matched filter.’’ We are convolving the time flipped and time-shifted version of the transmitted pulse with the transmitted pulse itself. This attempt to match

the pulses is also *maximizing the SNR*.

Consider the realization shown in Figure 1, where $h_i(t) = S_i(T - t)$. Our decision rule here, derived from Bayes' procedure is given as:

$$\max_i (E\{P(S_i|r = \underline{p})\}). \quad (24)$$

For more information about matched filter, please refer to Section 5.1.4 of the course textbook [4].

1.3.2 Correlator Realization

A matched filter realization assumes knowledge of the transmitted data. A receiver based upon a correlator structure loosens this assumption by instead assuming only a knowledge of the waveforms.

Consider again the received signal given in (18). Since we possess a knowledge of the signal set and we want to know which S_i was transmitted, we can correlate $r(t)$ with $S_i(t)$. If we normalize the result by the energy with respect to E_i , we will have a good approximation of how close $r(t)$ is to S_i . This decision rule can be expressed mathematically as:

$$\max_i \left(\int_0^T r(t) S_i(t) dt - \frac{E_i}{2} \right). \quad (25)$$

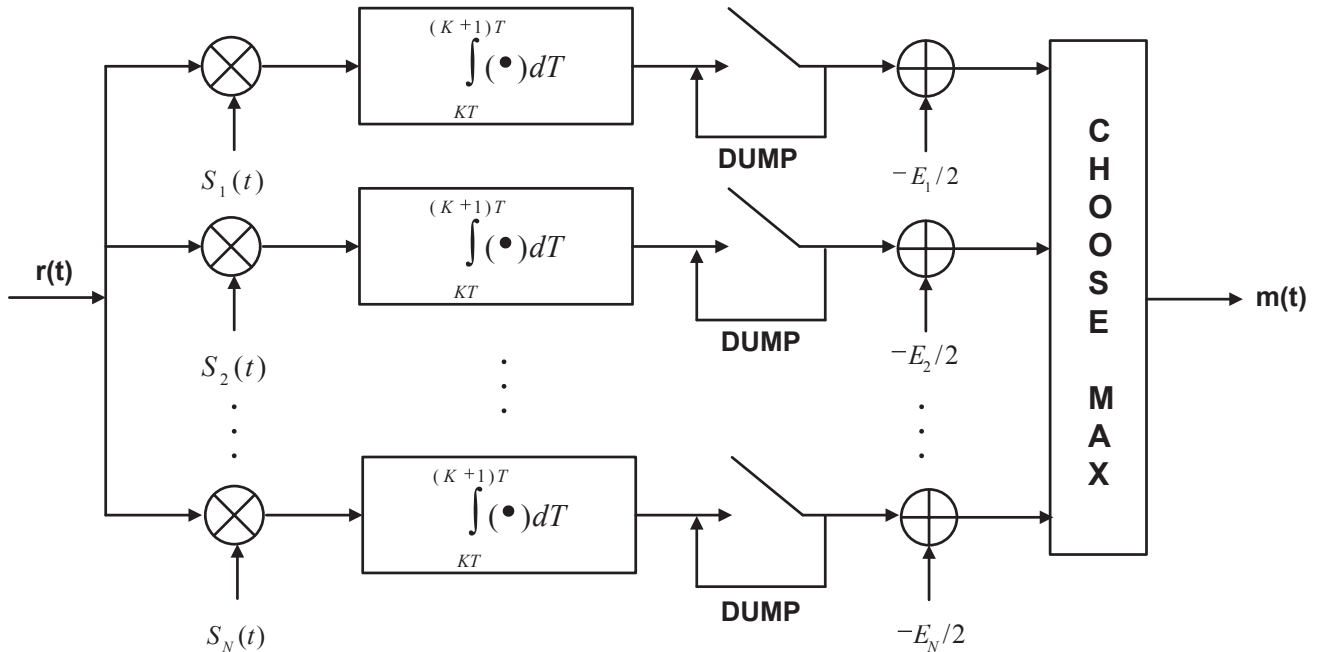


Figure 2: Correlator realization, which does not assume knowledge of the transmitted data.

1.4 Multicarrier Transmission

Instead of having one center frequency, f_c , Multi-Carrier Modulation (MCM) multiplexes serial input data into several parallel streams and transmits them over independent sub-carriers. These sub-carriers can be individually modulated and manipulated allowing for optimization with respect to

the channel. This is especially true for a wireline communications system where the channel is well defined.

Why MCM? What are the disadvantages of transmitting at a high data rate over a single carrier? What happens if part of your channel is severely attenuated? MCM offers a different approach to these problems. By sub-dividing the transmission over many carriers and tuning the modulation schemes and power levels of each carrier, a multi-carrier transmission scheme becomes quite robust to fast-fading channels and narrowband interference.

There are some important distinctions between OFDM in wired communications systems versus wireless communications systems. We will be focusing on wireless communications, but it is important to note that certain assumptions made here are not necessarily optimal solutions a wireline implementation.

1.4.1 Orthogonal Frequency Division Multiplexing

Orthogonal Frequency Division Multiplexing (OFDM) is an efficient type of multicarrier modulation, which employs the discrete Fourier transform (DFT) and inverse DFT (IDFT) to modulate and demodulate the data streams. The set-up of an OFDM system is presented in Figure 3. A high-speed digital input, $x[m]$, is demultiplexed into N subcarriers using a commutator. The data on each subcarrier is then modulated into an M -QAM symbol, which maps a group of $\log_2(M)$ bits at a time. For subcarrier k we will rearrange $a_k[\ell]$ and $b_k[\ell]$ into real and imaginary components such that the output of the “modulator” block is $p_k[\ell] = a_k[\ell] + jb_k[\ell]$. In order for the output of the IDFT block to be real, given N subcarriers we must use a $2N$ -point IDFT, where terminals $k = 0$ and $k = N$ are “don’t care” inputs. For the subcarriers $1 \leq k \leq N - 1$, the inputs are $p_k[\ell] = a_k[\ell] + jb_k[\ell]$, while for the subcarriers $N + 1 \leq k \leq 2N - 1$, the inputs are $p_k[\ell] = a_{2N-k}[\ell] + jb_{2N-k}[\ell]$.

The IDFT is then performed, yielding

$$s[2\ell N + n] = \frac{1}{2N} \sum_{k=0}^{2N-1} p_k[\ell] e^{j(2\pi nk/2N)}, \quad (26)$$

where this time $2N$ consecutive samples of $s[n]$ constitute an OFDM symbol, which is a sum of N different QAM symbols.

This results in the data being modulated on several subchannels. This is achieved by multiplying each data stream by a $\sin(Nx)/\sin(x)$, several of which are shown in Figure 4.

The subcarriers are then multiplexed together using a commutator, forming the signal $s[n]$, and transmitted to the receiver. Once at the receiver, the signal is demultiplexed into $2N$ subcarriers of data, $\hat{s}[n]$, using a commutator and a $2N$ -point DFT, defined as

$$\bar{p}_k[\ell] = \sum_{n=0}^{2N-1} \hat{s}[2\ell N + n] e^{-j(2\pi nk/2N)}, \quad (27)$$

is applied to the inputs, yielding the estimates of $p_k[\ell]$, $\bar{p}_k[\ell]$. The output of the equalizer, $\hat{p}_k[\ell]$, then passed through a demodulator and the result multiplexed together using a commutator, yielding the reconstructed high-speed bit stream, $\hat{x}[m]$.

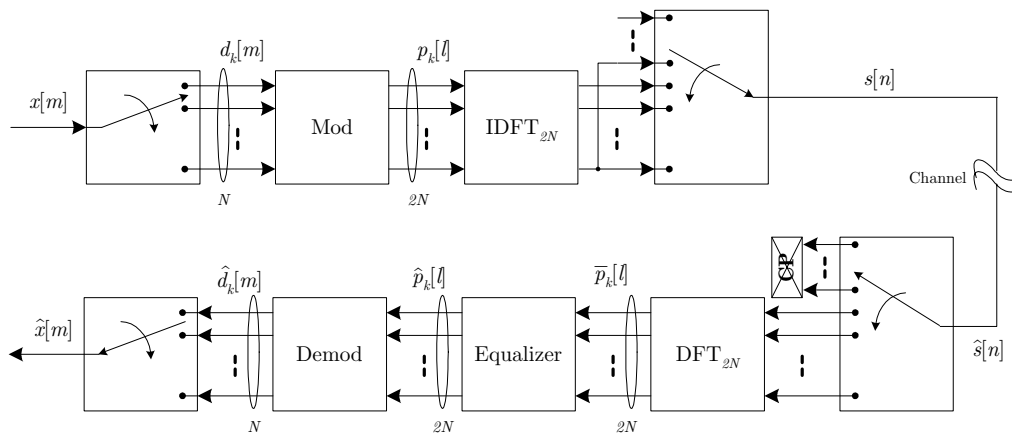


Figure 3: Overall schematic of an Orthogonal Frequency Division Multiplexing System

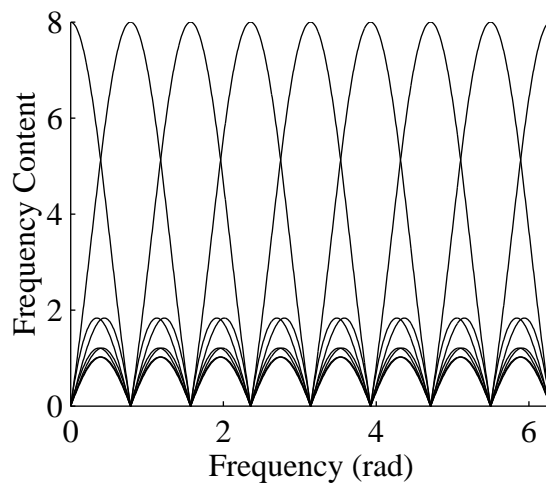


Figure 4: Characteristics of Orthogonal Frequency Division Multiplexing: Frequency response of OFDM subcarriers.

1.4.2 Dispersive Channel Environment

Until now, we have only considered an OFDM system operating under ideal conditions (i.e. the transmitter is connected directly to the receiver without any introduced distortion). Now we will cover channel distortion due to dispersive propagation and how it is compensated for in OFDM system.

From our high school physics courses, we learned about wave propagation and how they combine constructively and destructively. The exact same principles hold in high speed data transmission. For example, in a wireless communication system, the transmitter emanates radiation in all directions (unless the antenna is “directional”, in which case, the energy is focused at a particular azimuth). In an open environment, like a barren farm field, the energy would continue to propagate until some of it reaches the receiver antenna. As for the rest of the energy, it continues on until it dissipates.

In an indoor environment, as depicted in Figure 5(c), the situation is different. The line-of-sight component (if it exists), p_1 , arrives at the receiver antenna first, just like in the open field case. However, the rest of the energy does not simply dissipate. Rather, the energy is reflected by the walls and other objects in the room. Some of these reflections, such as p_2 and p_3 , will make their way to the receiver antenna, although not with the same phase or amplitude. All these received components are functions of several parameters, including their overall distance between the transmitter and receiver antennas as well as the number of reflections. At the receiver, these components are just copies of the same transmitted signal, but with different arrival times, amplitudes, and phases. Therefore, one can view the channel as an impulse response that is being convolved with the transmitted signal. In the open field case, the *channel impulse response* (CIR) would be a delta, since no other copies would be received by the receiver antenna. On the other hand, an indoor environment would have a several copies intercepted at the receiver antenna, and thus its CIR would be similar to the example in Figure 5(a). The corresponding frequency response of the example CIR is shown in Figure 5(b).

In an xDSL environment, the same principles can be applied to the wireline environment. The transmitted signal is sent across a network of telephone wires, with numerous junctions, bridging taps, and connections to other customer appliances (e.g. telephones, xDSL modems). If the impedances are not matched well in the network, reflections occur and will reach the devices connected to the network, including the desired receiver.

With the introduction of the CIR, new problems arise in our implementation which need to be addressed. In Section 1.4.3, we will look at how to undo the smearing effect the CIR has on the transmitted signal.

1.4.3 OFDM with Cyclic Prefix

The CIR can be modelled as a finite impulse response filter that is convolved with a sampled version of the transmitted signal. As a result, the CIR smears past samples onto current samples, which are smeared onto future samples. The effect of this smearing causes distortion of the transmitted signal, increasing the aggregate BER of the system and resulting in a loss in performance.

Although equalizers can be designed to undo the effects of the channel, there is a trade-off between complexity and distortion minimization that is associated with the choice of an equalizer. In particular, the distortion due to the smearing of a previous OFDM symbol onto a successive symbol is a

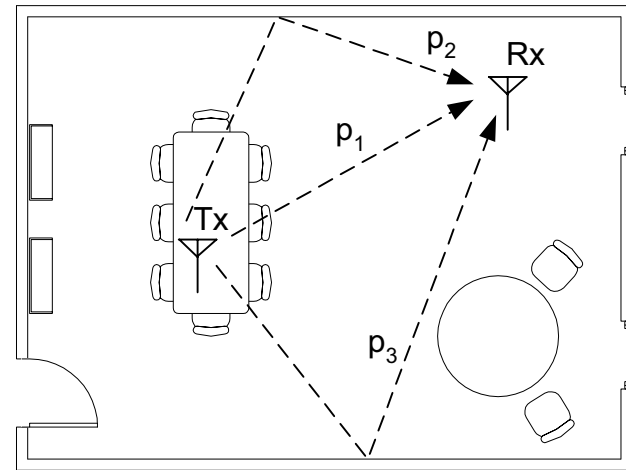
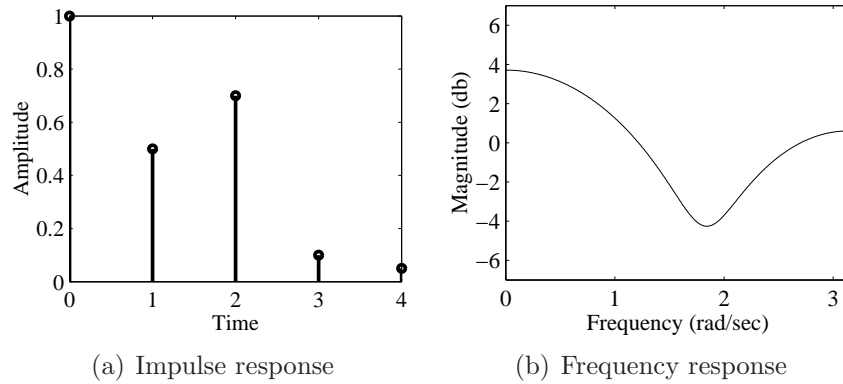


Figure 5: Example of a channel response due to dispersive propagation

difficult problem. One simple solution is to put a few “dummy” samples between the symbols in order to capture the intersymbol smearing effect. The most popular choice for these K dummy samples are the last K samples of the current OFDM symbol. The dummy samples in this case are known as a *cyclic prefix*, as shown in Figure 6(a).

Therefore, when the OFDM symbols with cyclic prefixes are passed through the channel, the smearing from the previous symbols are captured by the cyclic prefixes, as shown in Figure 6(b). As a result, the symbols only experience smearing of samples from within their own symbol.

At the receiver, the cyclic prefix is removed, as shown in Figure 6(c), and the OFDM symbols proceed with demodulation and equalization.

Despite the usefulness of the cyclic prefix, there are several disadvantages. First, the length of the cyclic prefix must be sufficient to capture the effects of the CIR. If not, the cyclic prefix fail to prevent distortion introduced from other symbols. The second disadvantage is the amount of overhead introduced by the cyclic prefix. By adding more samples to buffer the symbols, we must send more information across the channel to the receiver. This means to get the same throughput as a system

without the cyclic prefix, we must transmit at a higher data rate.

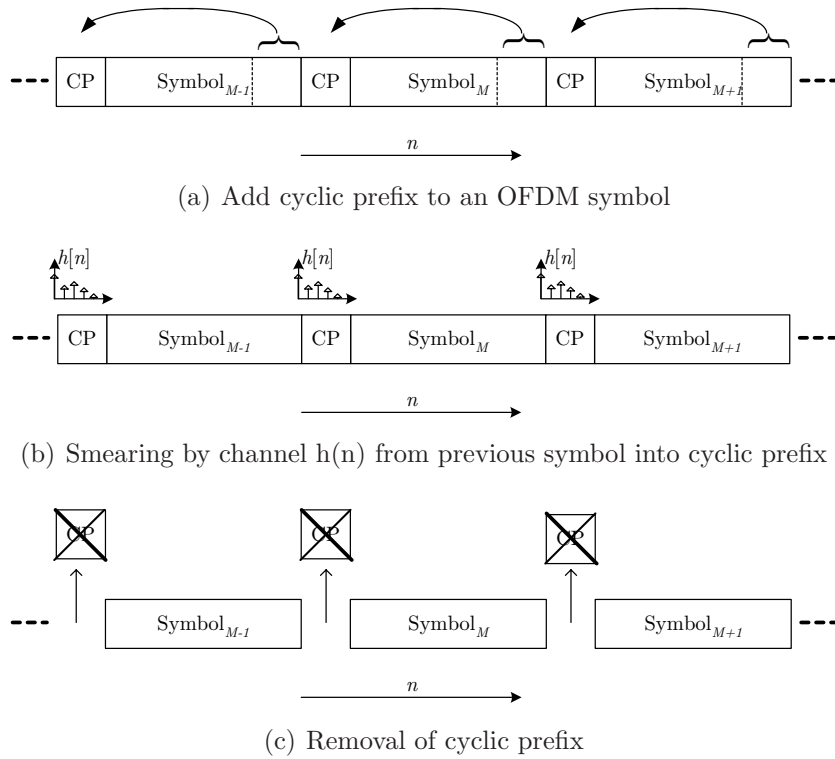


Figure 6: The process of adding, smearing capturing, and removal of a cyclic prefix

1.5 Suggested Readings

Although this laboratory handout provides some information about the fundamentals of digital modulation schemes, the reader is encouraged to review the material from the following references in order to gain further insight on these topics.

- Overview of Maximum Likelihood Detection
 - Section 5.6 in [4]
- Maximum Likelihood Detection in Synchronization
 - Section 7.8 and 8.8 in [4]
- Introduction to Simulation Techniques on Correlation Receiver
 - Section 2.4 and 3.1 in [5]
- Overview of Multicarrier Data Transmission
 - Section 2.1 in [6]

1.6 Problems

1. Carry out the Gram-Schmidt Orthogonalization procedure of the signals in Figure 7 in the order $s_3(t), s_1(t), s_4(t), s_2(t)$ and thus obtain a set of orthonormal functions $\{f_m(t)\}$. Then, determine the vector representation of the signals $\{s_n(t)\}$ by using the orthonormal functions $\{f_m(t)\}$. Also, determine the signal energies.

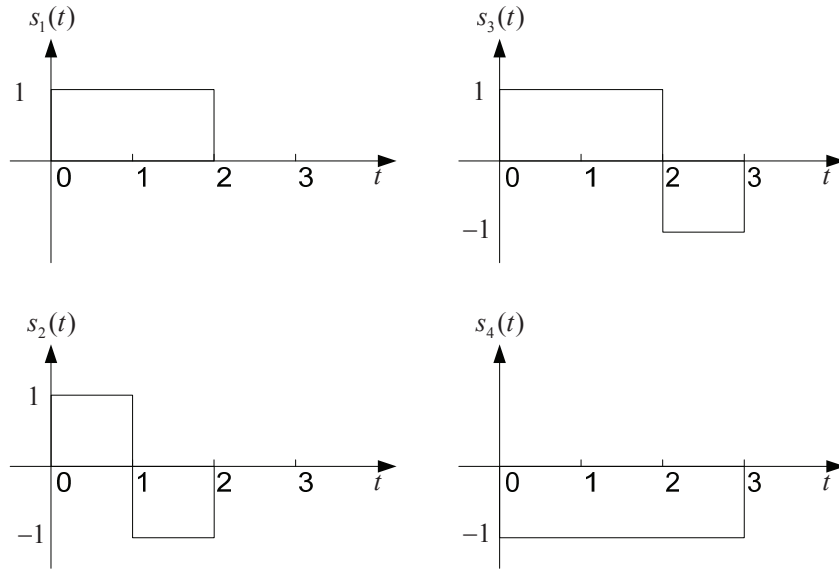


Figure 7: Four signal waveforms for the Gram-Schmidt Orthogonalization procedure.

2. Consider the signal: $s(t) = \frac{A}{T}t\cos(\omega_c t)$ for $0 \leq t \leq T$
 - (a) Determine the impulse response of the matched filter for the signal.
 - (b) Determine the output of the matched filter at $t = T$.
 - (c) Suppose the signal $s(t)$ is passed through a correlator that correlates the input $s(t)$ with $s(t)$. Determine the value of the correlator output at $t = T$. Compare your result with that in part (b).
3. A certain digital baseband modulation scheme uses the pulse shown in Figure 8 to represent binary symbol “1” and the negative of this pulse to represent binary “0”. Derive the formula for the probability of error incurred by the maximum likelihood detection procedure applied to this form of signaling over an AWGN channel.
4. Exercise 5.23 from course textbook [4].
5. Exercise 5.24 from course textbook [4].

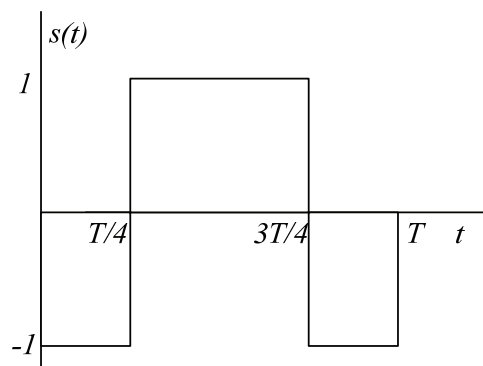


Figure 8: Pulse representing binary “1”.

2 Software Implementation

2.1 Observation Vector Construction

Referring to Figure 9, this part of the laboratory experiment will involve the construction of observation vector \mathbf{X} , that will be used for a correlator-based detector. This section employs the orthonormal functions $\{f_m(t)\}$ specified in Problem 1 of Section 1.6. This bank of correlators operates on the signal $x(t)$ to produce the observation vector \mathbf{X} .

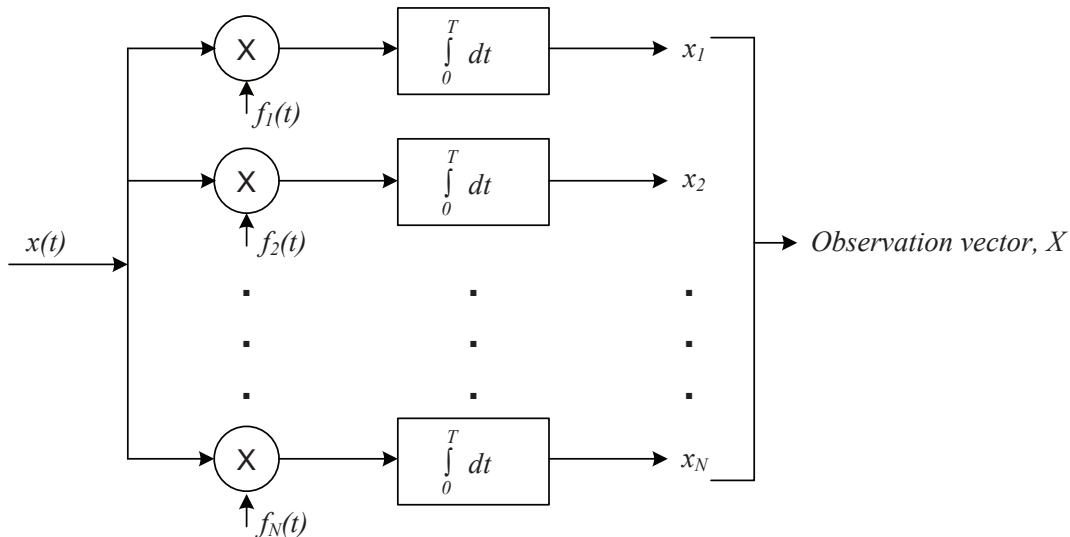


Figure 9: A bank of correlators operating on an input signal $x(t)$ to produce the observation vector \mathbf{X} , that will be used for a correlator-based detector.

Although the design presented here is for continuous time signals, we need to employ a discrete time approximation of 1000 samples per second in order for this to work in MATLAB. Also assume perfect synchronization between the received signal and the communications system.

Open `correlator.m` available from the course website. The code has been partially implemented, and you will need to finish the rest of the code with the following steps and plot the results.

- Define the four equiprobable signals $s_1(t)$, $s_2(t)$, $s_3(t)$, $s_4(t)$ shown in Figure 7. Since you need to choose one of the four signals randomly, a good way is to use `switch` and `case` in MATLAB. Plot a randomly generated stream of these waveforms generated in MATLAB.
- Define the orthonormal functions $\{f_m(t)\}$ that you have derived in Problem 1 of Section 1.6. This step is similar to the previous step, just make sure that the vector for $s_i(t)$ and $f_m(t)$ has the same length.
- Create an AWGN channel that will be used to add white Gaussian noise to the transmitted signal consisting of the four waveforms. You can use `randn` in MATLAB to define the noise. Plot the time domain representations of the input and output signals for the channel for several different noise variances. Explain how the noise could potentially impair the successful decoding of the intercepted signal at the receiver.

- Implement an “integrate-and-dump” block in order to obtain the observation vector \mathbf{X} . Since we are employing discrete time approximation for continuous time signals, integration could be realized by `sum` in MATLAB.
- Test your implementation with a transmission of duration 300 seconds, randomly consisting of one of the four equiprobable signals shown in Figure 7 of duration $T=3$ seconds each, with zero mean white Gaussian noise of variance 0.5 added.
- Plot each of the elements of the observation vector \mathbf{X} using the `stem` command in MATLAB.

2.2 Maximum-likelihood Decoder Implementation

Referring to Figure 10, this part of the laboratory experiment involves the implementation of the signal transmission decoder subsystem in the form of a maximum-likelihood (ML) decoder in order to accompany the subsystem described in Section 2.1. This decoder will operate on the observation vector \mathbf{X} from Section 2.1 to produce an estimate \hat{m} of the transmitted symbol m_i in a way that would minimize the average probability of symbol error.

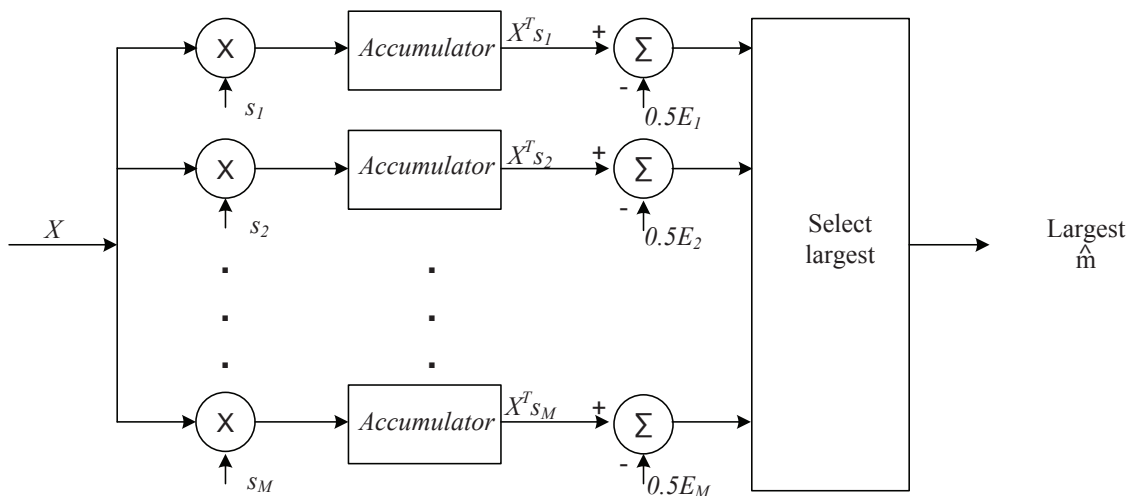


Figure 10: A maximum-likelihood (ML) decoder system operating on the observation vector \mathbf{X} from Section 2.1 to produce an estimate \hat{m} of the transmitted symbol m_i to minimize the average probability of symbol error.

Download and open `decoder.m` from the course website. Although this code is incomplete, it will serve as a starting point for the rest of this experiment.

- Define the four signal vectors that you have derived in Problem 1 of Section 1.6. Each signal vector should be a 1×3 vector.
- Calculate the energy of each signal vector defined in the previous step. $E_i = \sum_{j=1}^3 E_{ij}^2$, $i = 1, 2, 3, 4$, where E_{ij} is the component of each signal vector.
- Accumulator and subtraction as shown in Figure 10. Accumulator is realized by `sum` in MATLAB. Plot the output of the accumulator for each of the branches. Provide your observations and explain.

- Select the largest branch to produce an estimate \hat{m} . You can use `max` in MATLAB to select the largest branch.
- Remove the energy subtraction from each of the branches and repeat your experiment. Do you get the same results as before? What would be required for the results with and without energy subtraction to be equivalent?

Now try out the following parts with parameters in this structure:

- In Section 2.1, do not integrate the entire period T , integrate until $0.75T$. Plot both the estimate \hat{m} and the actual transmitted symbol m_i and compare it with the original plot.
- In Section 2.2, do not subtract the energy of $s_i(t)$ from each branch. Plot both the estimate \hat{m} and the actual transmitted symbol m_i and compare it with the original plot.
- Combine your implementation from Sections 2.1 and 2.2. Change the parameters of AWGN channel in Section 2.1 to make the signal to noise ratio (SNR) range from 10^{-5} to 10^{-1} . Compare the \hat{m} in Section 2.2 with the input x in Section 2.1 to get the bit error rate. Plot a BER curve. Note: In order to get an accurate BER curve, you may need a large number of transmitted signals, especially when the SNR value is high.

2.3 Correlator Realization of a Receiver in Simulink

In this part of the laboratory, you will be responsible for implementing a correlator realization of the receiver structure shown in Figure 2 using the signal waveforms shown in Figure 7. Consequently, you will be constructing a transmitter that is capable of randomly generating $s_1(t)$, $s_2(t)$, $s_3(t)$, and $s_4(t)$, devising an additive white Gaussian noise (AWGN) channel, and a four-branch receiver using correlation in each branch to help determine which symbol was transmitted every $T = 1$ second. Assume perfect synchronization in this system.

- Using `integrate_and_dump.mdl`, which is also shown in Figure 11, run the experiment for different pulse durations T (the default is $T = 1$ second). Note that the integrate-and-dump block accumulates samples according to the signal possessing the most samples per unit time. In this case, the Gaussian random number generator is producing 100 samples per 1 second duration. Explain the end-to-end operation of this basic Simulink model, from the binary PAM transmitter all the way across to the output of the integrate-and-dump block. Use time domain plots to help illustrate your explanation.
- Create a new Simulink model document that will contain your design for the four-branch correlator-based receiver. Implement a Simulink transmitter that can generate each of the four waveforms shown in Figure 7. Ensure that each waveform is equally likely to occur. Plot the output of this transmitter to show the randomly generated sequence of waveforms $s_1(t)$, $s_2(t)$, $s_3(t)$, and $s_4(t)$.
- Now pass this signal stream through the AWGN channel model in order to corrupt the signal with noise. Vary the amount of noise introduced to the signal stream and observe the shape of the signal and whether it matches the clean signal stream. How much noise must be added to make the signal stream unrecognizable?
- To create the receiver, you will need to implement the following steps:

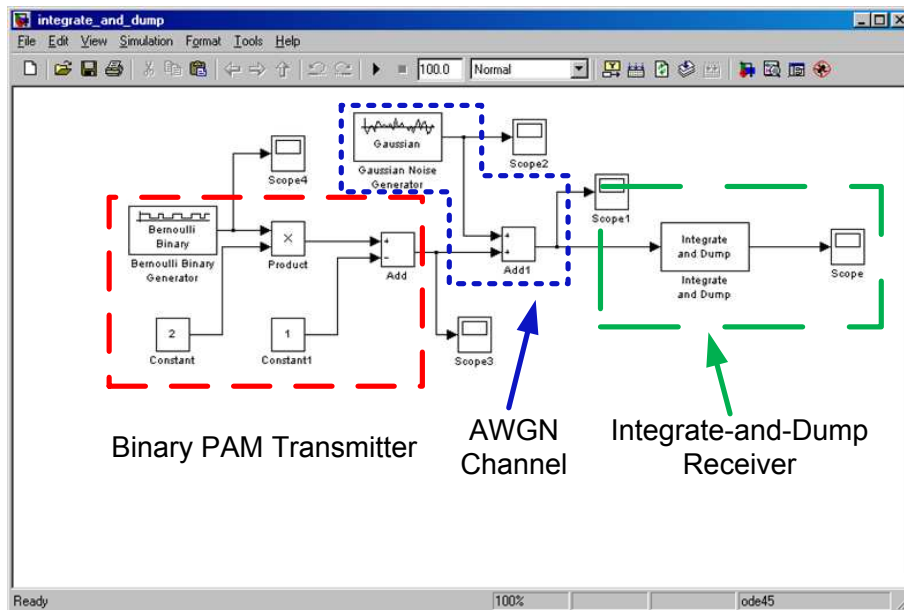


Figure 11: A simple Simulink model of a digital transceiver employing an integrate-and-dump block at the receiver.

- For each branch, you will need to repeat $s_i(t)$ indefinitely such that it aligns with each corresponding symbol period of the corrupted intercepted signal $r(t)$. One approach is to use the `repeating sequence` block in the basic Simulink sources blockset.
 - Multiply the repeating $s_i(t)$ waveforms in each of the branches with the intercepted signal $r(t)$. Take the output of this multiplication and input it into the integrate-and-dump block. What did you set as the number of samples to be integrated?
 - Take the output of the integrate-and-dump block and subtract from it the energy of the corresponding waveform $s_i(t)$. Since the receiver knows about all the waveforms ahead of time, simply calculate manually the symbol energy ahead of time and use the `constant` source block and the `addition` block to subtract off the energy from the integrated signal stream. What do you notice about the relative energy levels for each of the branches?
 - Finally, implement the decision making block that selects the maximum value at each time instant $T = 1$ seconds. Plot the output of the decision making process, indicating which of the four waveforms were selected. Why is there a delay in the decoded signal relative to the originally transmitted signal?
- Calculate the bit error rate of this receiver design for a range of noise variance values. Plot the BER curve of this model down to 10^{-3} .
 - Suppose you removed the energy subtraction for all the branches. Does the BER performance of the system change?

2.4 Multicarrier Modulation

2.4.1 MATLAB Design of Multicarrier Transmission

Implement Figure 3 according to the following hints. The size of the input/output should be a variable.

- Implement multiplex and demultiplex blocks using `reshape`.
- Implement IFFT and FFT blocks using `ifft` and `fft`.
- Each subcarrier equalizer should be a single complex value that is the opposite of the channel frequency attenuation corresponding to the center of the subcarrier.
- Modulation and demodulation should be implemented on subcarrier based.

Implement the channel filter, given as an input the channel impulse response, and add the noise generator. Test your implementation with the following parameters:

$$\begin{aligned}h_1 &= [1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0] \\h_2 &= [1 \ 0.1 \ 0.0001 \ 0 \ 0 \ 0] \\h_3 &= [1 \ 0.3 \ 0.4 \ 0.32 \ 0.2 \ 0.1 \ 0.05 \ 0.06 \ 0.02 \ 0.009]\end{aligned}\tag{28}$$

with $\mu = 0$ and $\sigma^2 = 0.001$. Hint: Use `filter` command in MATLAB

Implement the cyclic prefix add and remove blocks of the OFDM system. Make the length of the cyclic prefix an input variable.

With the system completely implemented, we can now study how it performs in a variety of conditions. The performance measure commonly used in digital transmission systems is the BER. Given an SNR value and channel conditions, the BER is the fraction of bits that are errors to the total number of bits transmitted. The only question is “how many bits need to be transmitted in order to obtain a reliable BER?” The rule-of-thumb answer is that one continues transmitting bits until 100 errors have been reached. [2]

With your implementation, plot the BER curves for the three channel realizations in Equation (28) between BER values of 10^{-2} and 10^{-4} . In particular, simulate your system when it uses $N = 16$, 32, and 64 subcarriers and employs a cyclic prefix of length $N/4$. The high speed input $x[m]$ is a uniformly distributed. Plot the BER results.

2.4.2 Simulink Design of Multicarrier Transmission

Open `ofdm.mdl` available from the course website. The example OFDM transmitter and receiver has already been implemented and provided as shown in Figure 12. You will need to finish the rest of the model with the following steps and play around with the parameters.

- Get the AWGN block from Simulink Library Browser. Connect the transmitter and receiver via AWGN block.

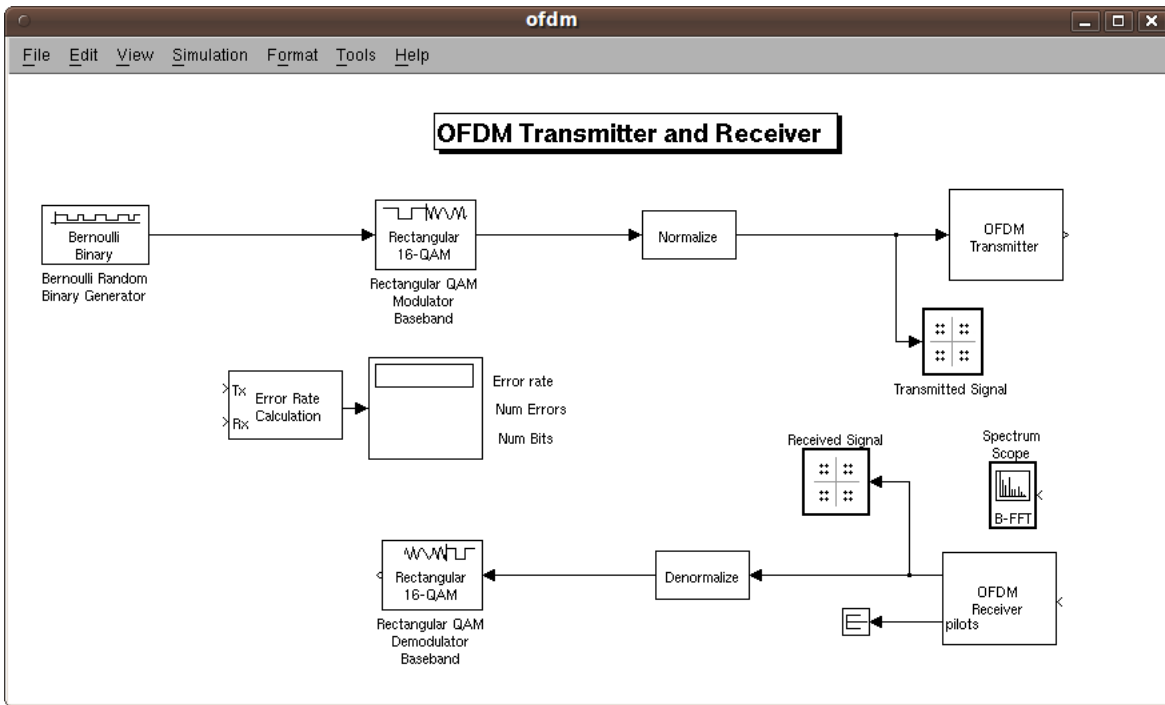


Figure 12: OFDM transmitter and receiver implementation in Simulink.

- Connect **Spectrum Scope** after the **AWGN** channel. This allows you to observe how the spectrum looks like after the OFDM transmitter and channel.
- Connect the transmitter and receiver to the inputs of **Error Rate Calculation** block. This allows you to examine how well the receiver performs with different fading characteristics and generate BER curve for varying SNR values.
- Change the SNR value of the **AWGN** channel and observe the effect in the frequency domain via **Spectrum Scope**. Plot your observation with several different SNR values and conclude how the spectrum change due to the channel.
- Open the **OFDM Transmitter** and **OFDM Receiver** subsystems, as shown in Figure 13 and Figure 14, change the number of subcarriers, and observe the effect in the frequency domain via **Spectrum Scope**. Plot your observation with several different number of subcarriers and conclude how the spectrum change due to subcarrier number.

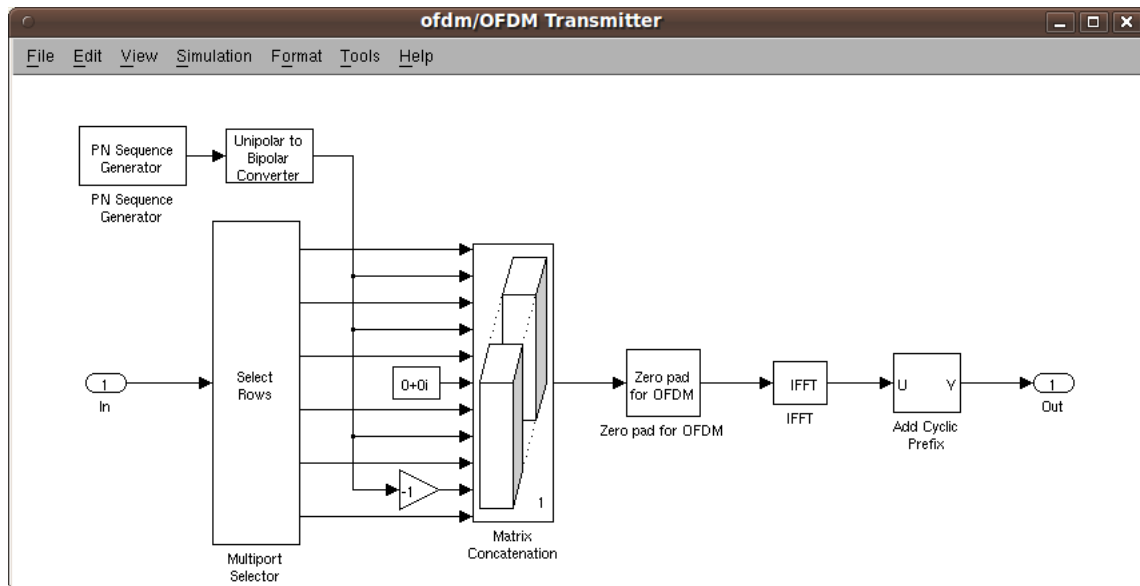


Figure 13: OFDM transmitter implementation in Simulink, where the subcarriers are connected by a Matrix Concatenator.

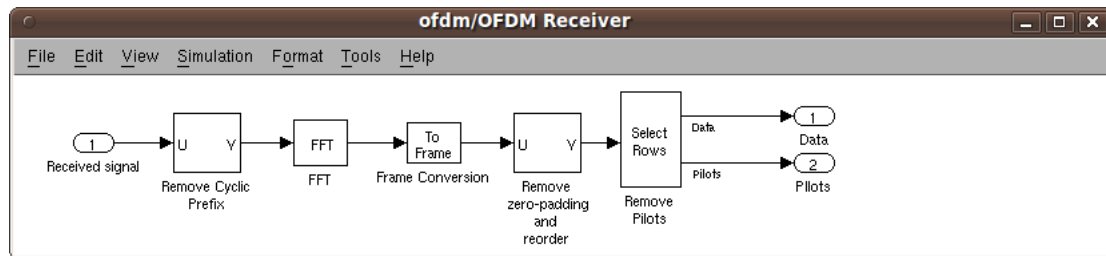


Figure 14: OFDM receiver implementation in Simulink, where the cyclic prefix is removed and the transmitted data is retrieved.

3 USRP2 Hardware Implementation

A *matched filter* is a theoretical framework and not the name of a specific type of filter. It is an ideal filter which processes a received signal to minimize the effect of noise. Hence, it maximizes the signal to noise ratio (SNR) of the filtered signal. It happens that an optimum filter does not exist for each signal shape transmitted and it is a function only of the transmitted pulse shape. Because of its direct relationship to the transmitted pulse shape, it is called a matched filter. The two most common matched filters are integrate and dump and root raised cosine. We have already studied integrate and dump in Section 2.3, so we will focus on root raised cosine in this section.

In this section, we use two models to illustrate a typical setup in which a transmitter uses a square root raised cosine filter to perform pulse shaping and the corresponding receiver uses a square root raised cosine filter as a matched filter. The receiver plots an *eye diagram* from the filtered received signal. Based on this diagram, we can learn the effect of matched filter in a communication system.

3.1 Eye Diagram

In telecommunication, an *eye diagram*, also known as an *eye pattern*, is an oscilloscope display in which a digital data signal from a receiver is repetitively sampled and applied to the vertical input, while the data rate is used to trigger the horizontal sweep. It is so called because, for several types of coding, the pattern looks like a series of eyes between a pair of rails.

Several system performance measures can be derived by analyzing the display. If the signals are too long, too short, poorly synchronized with the system clock, too high, too low, too noisy, or too slow to change, or have too much undershoot or overshoot, this can be observed from the eye diagram. For example, in Figure 15, D_A is the range of amplitude differences of the zero crossings and is a measure of distortion caused by *intersymbol interference* (ISI), J_T is the range of amplitude differences of the zero crossing and is a measure of the timing jitter, M_N is a measure of noise margin, and S_T is measure of sensitivity-to-timing error. In general, the most frequent usage of the eye pattern is for qualitatively assessing the extent of the ISI. As the eye closes, the ISI increases; as the eye opens, the ISI decreases.

3.1.1 Discrete-Time Eye Diagram Scope

The **Discrete-Time Eye Diagram Scope** block displays multiple traces of a modulated signal to produce an eye diagram. You can use the block to reveal the modulation characteristics of the signal, such as pulse shaping or channel distortions. This is a most important block that will be used in this section.

An *open* eye pattern corresponds to minimal signal distortion. Distortion of the signal waveform due to intersymbol interference and noise appears as *closure* of the eye pattern.

3.2 Matched Filter Observation

In this section, you can use the DBPSK transmitter model on the transmitter side. On the receiver side, download and open `rcfilter_receiver.mdl` from course website, as shown in Figure 16. The top level of this model looks similar to the DBPSK receiver model, however, its enabled subsystem

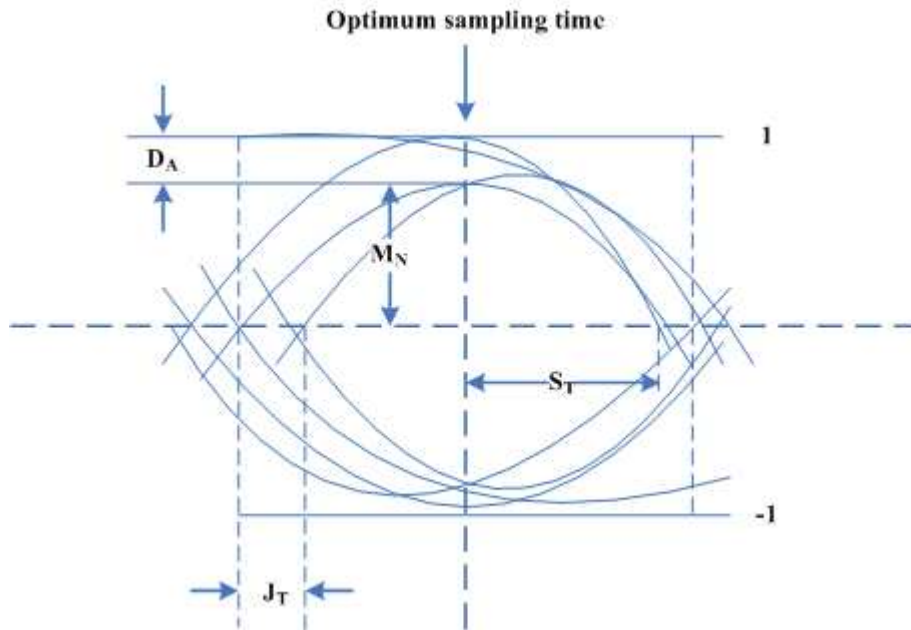


Figure 15: A typical eye pattern. The width of the opening indicates the time over which sampling for detection might be performed. The optimum sampling time corresponds to the maximum eye opening, yielding the greatest protection against noise. If there were no filtering in the system then the system would look like a box rather than an eye.

contains the information that we are interested in.

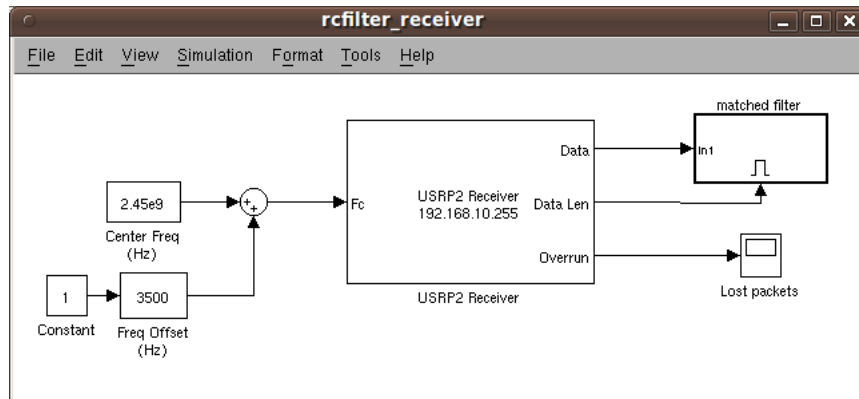


Figure 16: A model that illustrates a typical receiver setup, in which the receiver uses a square root raised cosine filter as a matched filter corresponding to the square root raised cosine filter on the transmitter.

Go into the matched filter subsystem, as shown in Figure 17, where there is a Raised Cosine Receive Filter and a Discrete-Time Eye Diagram Scope. Perform the following tasks and plot your observations:

- Compare Raised Cosine Receive Filter and Raised Cosine Transmit Filter on the transmitter side, see whether they match each other.

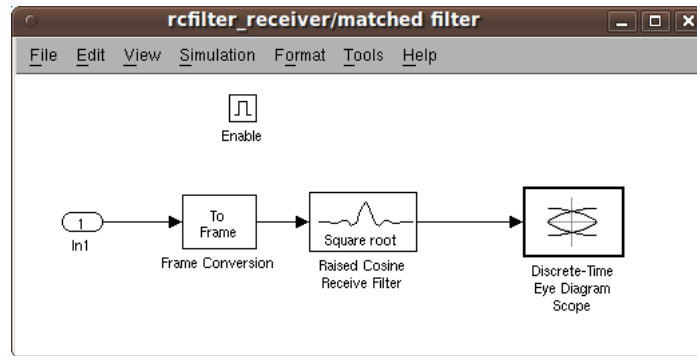


Figure 17: The subsystem of the receiver model, in which the receiver plots an *eye diagram* from the filtered received signal. Based on this diagram, we can learn the effect of matched filter in a communication system.

- Change the parameters (group delay, rolloff factor and etc.) of **Raised Cosine Receive Filter** to make it *match* the **Raised Cosine Transmit Filter**, how does the eye diagram look like?¹
- Change the parameters of **Raised Cosine Receive Filter** to make it *mismatch* the **Raised Cosine Transmit Filter**, how does the eye diagram look like?
- Delete **Raised Cosine Receive Filter** block, as shown in Figure 18, how does the eye diagram look like?

When the eye diagram has two widely opened “eyes”, it indicate appropriate instants at which to sample the filtered signal before demodulating. It also indicates the absence of intersymbol interference at the sampling instants of the received waveform. A large signal-to-noise ratio in the channel will produce a low-noise eye diagram. You can construct a Simulink-only model using **AWGN Channel** block and then change the SNR parameter in the **AWGN Channel** block to see how the eyes in the diagram change.

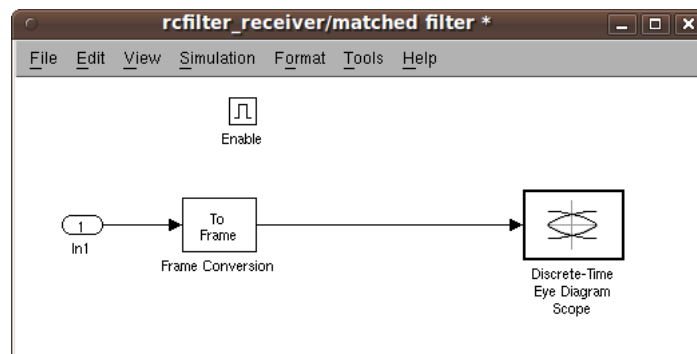


Figure 18: The subsystem of the receiver model without a **Raised Cosine Receive Filter**.

¹To get better observation, you need to use ‘Autoscale’ of eye diagram.

4 Open-ended Design Problem: Duplex Communication

4.1 Duplex Communication

A duplex communication system is a system composed of two connected parties or devices that can communicate with one another in both directions. Duplex systems are often employed in many communications networks, either to allow for a communication “two-way street” between two connected parties or to provide a “reverse path” for the monitoring and remote adjustment of equipment in the field.

Systems that do not need the duplex capability use instead simplex communication. These include broadcast systems, where one station transmits and the others just “listen”. Several examples of communication systems employing simplex communications include television broadcasting and FM radio transmissions.

Thus far, the models that we have been used in this course all use simplex communication, such as the DBPSK transmitter and receiver studied in previous laboratory experiments. Since we are using the XCVR2450 transceiver daughterboard, our USRP2 can be either a transmitter or a receiver. However, it cannot be a transmitter and a receiver at the same time. Consequently, in this open-ended design problem, you will need to implement a system that is capable of doing half-duplex communication.

4.2 Half-duplex

A half-duplex (HDX) system provides communication in both directions, but only one direction at a time (not simultaneously). Typically, once a party begins receiving a signal, it must wait for the transmitter to stop transmitting, before replying.

An example of a half-duplex system is a two-party system such as a “walkie-talkie” style two-way radio, wherein one must use “Over” or another previously-designated command to indicate the end of transmission, and ensure that only one party transmits at a time, because both parties transmit and receive on the same frequency.

A good analogy for a half-duplex system would be a one-lane road with traffic controllers at each end. Traffic can flow in both directions, but only one direction at a time, regulated by the traffic controllers.

There are several different ways to control the traffic in a half-duplex communication system, and we suggest you to use *time division duplexing* (TDD).

4.3 Time Division Duplexing

Time division duplexing (TDD) refers to a transmission scheme that allows an asymmetric flow for uplink and downlink transmission which is more suited to data transmission. In a time division duplex system, a common carrier is shared between the uplink and downlink, the resource being switched in time. Users are allocated one or more timeslots for uplink and downlink transmission, as shown in Figure 19.

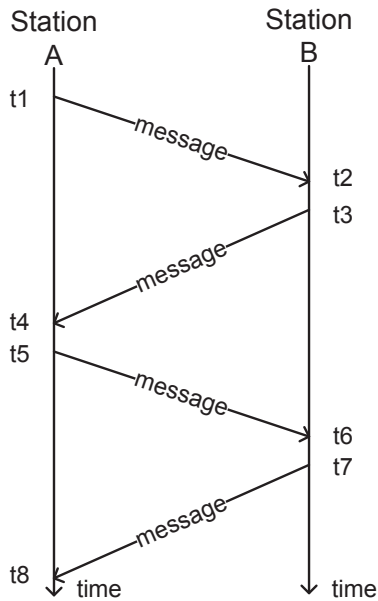


Figure 19: A half-duplex communication system using time division duplexing. A common carrier is shared between Station A and Station B, the resource being switched in time.

For radio systems that aren't moving quickly, an advantage of TDD is that the uplink and downlink radio paths are likely to be very similar. This means that techniques such as beamforming work well with TDD systems.

Examples of time division duplexing systems are:

- UMTS 3G supplementary air interfaces TD-CDMA for indoor mobile telecommunications.
- DECT wireless telephony.
- Half-duplex packet mode networks based on carrier sense multiple access, for example 2-wire or hubbed Ethernet, Wireless local area networks and Bluetooth, can be considered as Time Division Duplex systems.
- IEEE 802.16 WiMAX.

4.4 Hints

Suppose you have a radio station A and a radio station B, you are required to perform the following tasks in time sequence: Station A transmits “Hello world” to Station B, Station B transmits “Hello world” to Station A, Station A transmits “Hello world” to Station B, Station B transmits “Hello world” to Station A, and etc., , as shown in Figure 19.

- At the beginning, you may want to try something simple and see whether it works. For example, you can transmit ‘1010.....’ instead of “Hello world”, such that frame synchronization is not necessary. You can test one cycle only, that is Station A transmits “Hello world” to Station B and Station B transmits “Hello world” to Station A.

- In order to make it an automatic communication system, you need to write a MATLAB code for Station A and another MATLAB code for Station B, such that all you need to do is to click the “Run” button on two stations at the beginning of simulation.
- A MATLAB function called `sim` allows you to run your Simulink model from MATLAB code.
- A MATLAB function called `pause` allows you to specify the interval between two actions in your MATLAB code. This function is necessary due to the different computation ability of PCs. You want both of them get ready before continuing to the next step. It is suggested that you have at least 20 seconds buffer time.
- In order to specify the simulation time of your model, go to **Simulation** → **Configuration Parameters** → **Solver** tab, change **Stop time**.
- For example, a very simple piece of code is as follows: `sim('tx'); pause(10); sim('rx');`
According to this code, the radio station first runs a Simulink model called 'tx', then it pauses for 10 seconds, and in the end it runs a Simulink model called 'rx'.

5 Lab Report Preparation & Submission Instructions

Include all your answers, results, and source code in a lab report formatted as follows:

- Cover page: includes course number, laboratory title, names and student numbers of team, submission date
- Table of contents
- Pre-lab
- Responses to laboratory questions and explanation of observations
- Responses to open-ended design problem
- Source code

Please include images and outputs wherever possible, as well as insights on your laboratory.

Each group is to submit a single report electronically (in PDF format not exceeding 2MB) to alexw@ece.wpi.edu by scheduled due date. Reports that do not meet these specifications will be returned without review.

References

- [1] John A. Gubner. *Probability and Random Processes for Electrical and Computer Engineers*. Cambridge University Press, 2006.
- [2] Michel C. Jeruchim, Philip Balaban, and K. Sam Shanmugan. *Simulation of Communication Systems: Modeling, Methodology and Techniques*. Springer, 2nd edition, October 2000.
- [3] K.Abend and B.D.Fritchman. Statistical detection for communication channels with intersymbol interference. In *Proceedings of IEEE*, volume 58, pages 779–785, May 1970.
- [4] Michael Rice. *Digital Communications: A Discrete-Time Approach*. Pearson/Prentice Hall, Upper Saddle River, NJ, USA, 2009.
- [5] Dennis Silage. *Digital Communication Systems using MATLAB and Simulink*. Bookstand Publishing, Gilroy, CA, 2009.
- [6] Alexander M. Wyglinski. *Physical Layer Loading Algorithms for Indoor Wireless Multicarrier Systems*. PhD thesis, McGill University, Montreal, QC, Canada, November 2004.