Pertanika J. Sci. & Technol. 10(2): 269 - 286 (2002)
ISSN: 0128-7680

Wireless Spread Spectrum Communication Channel Modelling and Simulation Technical Area: Wireless Communication

*Sabira Khatun, Ashraf Gasim Elsid Abdalla & Borhanuddin Mohd Ali
Department of Computer & Communication System Engineering,
Faculty of Engineering, Universiti Putra Malaysia
43400 UPM, Serdang, Selangor, Malaysia
e-mail: sabira@eng.upm.edu.my

Received: 31 July 1999

ABSTRAK

Kertas ini berkait dengan pola Rangkaian Komunikasi Spektrum Sebaran Wayarles dan perancangan menggunakan teknik spektrum sebaran Jujukan Terus (DSSS) model-model teori dan matematik dibangunkan untuk simulasi dan penilaian persembahan. Sebab pensimulasian sistem adalah untuk memeriksa kesahan dan mengelak sebarang perubahan yang tidak perlu semasa pelaksanaan perkakasan sebenar sistem tersebut. Sebab lain pensimulasian sistem sebelum melaksanakan adalah untuk mencari cara yang sebaik mungkin atau kaedah utnuk membuatnya seperti teknik modulasi, lebar jalur, sekuriti dan sebagainya. Jenis rangkaian komunikasi ini memberikan kebolehan untuk mengelak sumber-sumber luaran lain daripada penyesakan dan gangguan dengan transmisi informasi disebabkan penggunaan teknik Jujukan Terus. Kertas ini menghuraikan pola, pembangunan dan simulasi saluran komunikasi wayarles digital dalam bangunan. Saluran komunikasi tersebut mengandungi unit penerimaan transmit beroperasi dalam frekuensi 900-915 MHz. Untuk rangkaian komunikasi wayarles yang selamat dan boleh diharap teknik Spektrum Sebaran Jujukan Terus (DSSS) digunakan. Bahagian yang paling mencabar adalah penerima di mana jujukan hingar pseudorawak perlu diselaraskan untuk memulihkan mesej sebenar yang dihantar. Lain-lain kawasan penting penyiasatan termasuklah kumpulan kod hingar pseudorawak (kod PN), Kekuncian Anjakan Amplitud (ASK), pemodulatan/penyahmodulatan Kekuncian Anjakan Fasa Perduaan Kebezaan (DBPSK), modulatan Kekuncian Anjakan Fasa Kebezaan (DPSK), pengesanan jelas lawan tidak jelas, dan sebagainya.

ABSTRACT

This paper deals with *Wireless Spread Spectrum Communication Link* design and planning using Direct Sequence Spread Spectrum (DSSS) technique. The theoretical and mathematical models are developed for simulation and performance evaluation. The purpose of simulating the system is to check validity and avoid unnecessary changes during the actual hardware implementation of the system. Another purpose of simulating the system before implementing it is to find the best possible way or method to fabricate it like the modulation technique, bandwidth, security etc. This type of

*Communicating Author

communication link gives the ability to prevent other external sources from jamming and interfering with the transmission of information due to the use of Direct Sequence technique. This paper describes the design, development and simulation of an indoor digital wireless communication channel. The communication channel consists of a transmit-receive unit operating in the 900-915 MHz frequency range. For a reliable and secure wireless communication link, the Direct Sequence Spread Spectrum (DSSS) technique is used. The most challenging part is the receiver where the pseudo-random noise sequences must be synchronized to successfully recover the original transmitted message. Other key areas of investigation include selection of the pseudo-random noise code (PN code), Amplitude Shift Keying (ASK), Differential Binary Phase Shift Keying (DBPSK) modulation/demodulation, *Differential Phase Shift Keying (DPSK)* modulation, coherent versus non-coherent detection, etc.

Keywords: Wireless, spread spectrum, direct sequence, differential phase shift keying

INTRODUCTION

Spread Spectrum Communication is a method to transmit information wirelessly and securely using the Direct Sequence Spread Spectrum (DSSS) technique (Wilheimsson and Zigangirov 1998; Chu and Mitra 1998; Qiao 1998; Host-Madsen and Cho 1999; Glistic *et al.* 1999; The American Radio Relay League 1996; Power Spectral Density Curve; Messier 1998). This paper describes the construction of a transmit / receive unit to operate in the range of 900 - 915 MHz band using the spread spectrum method (Prasad 1996; Peterson and Ziemer 1985; Freeman 1995). One such technique is direct sequence spread spectrum, which has become popular for many wireless communication systems (Wilheimsson and Zigangirov 1998; Chu and Mitra 1998; Qiao 1998; Host-Madsen and Cho 1999; Glistic *et al.* 1999; Glistic *et al.* 1999; The American Radio Relay Leaque 1996).

The communication channel involves the construction of a spread spectrum transmitter and receiver. In this paper two areas of investigation have been selected. The first stage is the simulation and modelling, such as channel modelling and Tx / Rx circuit design. The second stage involves the hardware implementation, which requires the determination of antenna impedance, radiation pattern, cost, durability and polarization, Tx / Rx switch for half duplex operation, amplifier impedance, bandwidth, gain, power, distortion, modulator/demodulator BPSK, QPSK, DPSK, local oscillator frequency, stability, power output and digital interference synchronization speed (Messier 1998). In this paper the main focus is on the system modelling and simulation.

The transmitter and receiver comply with self-defined standards and protocols to enable proper communication and data transfer described in detail in (Wilheimsson and Zigangirov 1998; Chu and Mitra 1998; Qiao 1998; Host-Madsen and Cho 1999; Glistic *et al.* 1999). The various blocks or program functions, which make up the overall system, are implemented by software simulation (Ong 1998; Proakis and Salehi 1998).

The challenge in implementing a working DSSS is in the receiver. In order to receive information successfully the receiver has to synchronize with the pseudo random noise (PN) sequence of the transmitter (Transmission Line Attenuation Chart). This involves extracting the sequence from the incoming signal, aligning the local sequence to the transmitted sequence, and then locking onto the incoming signal so that data can be correctly de-spread (Error Correction with Hamming Codes 1994; Blahut 1983). The Amplitude Shift Keying (ASK) and Differential Phase Shift Keying (DPSK) modulation and demodulation scheme have been selected (Peterson and Ziemer 1985). Because of its very low output power, it is not expected that the transmitted signal will interfere with any other communication equipment.

This paper is organised as follows. The next section describes the theoretical and mathematical models. The simulation model, the results and discussion and finally the conclusion follow this.

THE THEORETICAL AND MATHEMATICAL MODEL

The simulation has been carried out using MATLAB as the main simulation tool. This section details the concept of the simulation method and gives a description of the simulation model. In addition the simulation environment is presented.

The Concepts of the Method Applied

The primary advantage of a spread spectrum communication system is its ability to reject interference whether it is unintentional interference by another user simultaneously attempting to transmit through the same channel, or the intentional interference by a hostile transmitter attempting to jam the transmission. It also provides excellent narrow-band noise rejection characteristics. The fundamental concept of spread spectrum is to spread the baseband digital signal with a periodic binary sequence, noise-like in nature, called a pseudo random noise (PN) sequence.

In a DSSS system, a PN sequence is used to convert a narrow-band digital signal to a larger bandwidth signal, referred to as a spread signal. To transmit the spread signal through a channel such as the atmosphere, Amplitude Shift Keying (ASK) or Phase Shift Keying (PSK) techniques are applied to the spread signal. A sinusoidal carrier is multiplied by the spread data to produce ASK modulated data or the carrier is multiplied by differentially encoded spread data to produce DPSK modulated data. The received signal may be recovered by using coherent detection or a phase lock loop and a matched filter.

Synchronization is of concern with the recovery of the baseband digital signal. For proper operation, a spread spectrum system requires that the locally generated pseudo random noise sequence used to de-spread the received signal be synchronized with the pseudo random noise sequence used to spread the transmitted signal.

The locally generated pseudo random noise sequence is compared to an interval of the received signal, a measure of correlation is used to determine when the two signals are satisfactorily aligned. After alignment, the remaining received signal is then correlated with the pseudo random noise sequence and is properly de-spread using a matched filter after which the baseband digital data is properly recovered.

Pseudo Random Noise Sequences

Sequence Spread Spectrum applies the principle of spreading the spectrum through the use of pseudo random noise sequences. The bit sequence is not truly random since the sequence is periodic. However, it is referred to as pseudo random because the periodicity is so large that usually more than one thousand bits occur before the sequence repeats. A random bit sequence generator forms the pseudo random noise sequence. The generator is a set of feedback shift registers operated by a single clock. During a pulse of the clock, the state of each flip-flop is shifted to the next one and the result is fed back as the input to the first flip-flop. This sequence is then employed in the transmitter and receiver for spreading and de-spreading.

Processing Gain in spread spectrum system is defined as the ratio of transmitted bandwidth to information bandwidth. This parameter can also be defined as the difference between the signal-to-noise-ratio (in dB) of the transmitted bandwidth to the information bandwidth. The spread spectrum technique results in a message signal with a transmission bandwidth (B_t) that is much larger than the information bandwidth (B_i) of the original signal (Prasad 1996). This can also be expressed in terms of dB by:

$$G_p(dB) = 10\log_{10}(\frac{B_t}{B_i}) \tag{1}$$

where, (B_t/B_i) = L, is the length of PN code.

For a spread spectrum system the ratio of the transmission bandwidth to information bandwidth is often the length of the PN code length. By doubling the length of the PN code, a 3dB increase in signal to interference ratio is obtained. By increasing the processing gain, the system will be better able to reject interfering signals, and more users will be able to reuse the same frequency band using different PN codes.

Model Description

Two modulation techniques have been implemented, the PN modulation (or coding) and ASK or DPSK, respectively.

Pseudo Random Noise Modulation

First, the incoming data sequence is modulated with a pseudo random noise sequence code. This noise-like code transforms the narrowband data sequence into a noise-like wide-band signal. The function of pseudo random noise modulation is to spread each bit of binary data in the transmit packet which converts the original narrow-band digital signal into a wide-band spread spectrum signal. This requires the multiplication of every bit in the transmit packet by a predefined seven bit pseudo random noise sequence to form a noise-like spread digital signal. The bandwidth occupied by the pseudo random noise modulated data will be seven times larger than the original's transmit packet bandwidth. In general this process can be described as follows:

$$M(t) \times PN(t) = P(t) \tag{2}$$

where M(t) = the message signal, PN(t) = the PN code and P(t) = the PN modulated wave. However, the bit rate of this signal is $L * R_b$, where L is the length of the PN code and R_b is the bit rate of M(t).

PN demodulation takes place at the receiver. PN demodulation simply decodes the transmitted message by multiplying the transmitted message with the PN code. The demodulated signal, $\hat{M}(t)$, is a good approximation of the original message data and can be expressed as

$$\hat{M}(t) = P(t) \times PN(t) \tag{3}$$

ASK Modulation

In the second technique the resultant wide-band signal (PN modulated signal) is used to modulate a local carrier to produce an ASK signal. The ASK modulation is essential for conversion of the baseband signal into a radio frequency signal.

In ASK technique, the different amplitudes differentiate each binary sign example, 1 volt (or 'on') may represent a binary bit '1' and 0 volt (or 'off') may represent a binary bit '0'. In Differential Binary Phase Shift Keying (DPSK) the phase changes only at the transition from bit '1' to '0' or vice versa.

In ASK modulation, a baseband data signal P(t) is modulated by a complex envelop g(t) with carrier wave S(t) and the modulated signal Y(t) is

$$Y(t) = P(t) \times g(t)$$

where g(t) = A S(t), A is a sinusoidal amplitude for sending a binary bit '1' and $S(t) = \cos(2\pi f_c t)$, f_c is the operating frequency (915MHz).

$$\therefore Y(t) = AP(t) \cos(2\pi fct) \tag{4}$$

The radio frequency signal is then transmitted across the channel and received at the other end. Here two stages of demodulations are required. First, the received noise-like wide-band signal is passed through a ASK demodulator to demodulate the signal. A method of synchronization must be employed at this stage to ensure proper de-spreading of the demodulated signal.

ASK demodulation is used to recover the baseband signals. Multiplying the received waveform (that deviates from the desired values frequency and amplitude by Δf and ΔA respectively due to channel noise) with carrier wave S(t), the demodulated wave is

$$Y'(t) = \widehat{Y}(t) \times S(t) \tag{5}$$

where the received waveform

$$\widehat{Y}(t) = (A + \Delta A)\times(P(t)\times\cos 2\pi(fc+\Delta f)\, t.$$

Since ΔA and Δf are very small using an appropriate scaling factor 'K' the low pass filter (LPF) output can be represented as follows:

$$Y'(t) = \frac{1}{2}AP(t)\times\cos 2\pi\Delta f\, t \Rightarrow \frac{1}{2}A\,P(t)\times K \Rightarrow \widehat{P}(t) \tag{6}$$

which is an approximation of the original signal P(t).

Hence, the steps taken to ASK modulate and detect a spread spectrum signal are as follows:

At the Transmitter:	PN modulation
	ASK modulation
At the Receiver:	ASK demodulation
	PN demodulation
	Message retrieval

DPSK Modulation

In DPSK technique the implementation of the DSSS system can be broken down into two main sections: i. e. hardware section and a software section. The software portion of the implementation performs all the DSP (Digital Signal Processing) while the hardware part of the implementation performs the DPSK modulation and demodulation.

There are mainly two steps in DPSK modulation: 1) differential encoding of the signal and 2) BPSK modulation.

1) Differential Encoding:

If the data from the information source is denoted by D_n and the initial reference bit is Cn-1, the differentially encoded data sequence C_n is (Peterson and Ziemer 1985):

$$C_n = D_n \oplus C_{n-1} \tag{7}$$

where, '$\oplus$' represents the exclusive OR (XOR) operations. This operation is used to produce the differentially encoded data sequence $\{C_n\}$ from original data sequence $\{D_n\}$. The next step is PN coding using $\{C_n\} = M(t)$ to obtain P(t) followed by BPSK modulation.

The differential decoding is performed by forming the sequence $\{\hat{D}\}$ as follows:

$$\hat{D}_n = C_n \oplus C_{n-1} \tag{8}$$

2) BPSK Modulation

P(t), the PN modulated wave is used to binary phase shift the carrier wave S(t), resulting in a DPSK modulated signal $Y_1(t)$. So the transmitted signal

$$Y_1(t) = P(t) \text{ x } S(t) \tag{9}$$

Let $\hat{S}(t) = \cos(2\pi(f_c+f_\Delta)t+\phi)$ represent the local oscillator carrier wave at the receiver where f_Δ is the carrier frequency offset between the transmitter and receiver and ϕ is the carrier phase. Hence the demodulated wave is

$$Y_2(t) = Y_1(t) \text{x} \hat{S}(t) = P(t) \text{x} S(t) \text{x} \hat{S}(t) \tag{10}$$

Since f_Δ is very small, using an appropriate scaling factor 'R', the filter output is

$$Y_2(t) = \frac{1}{2}P(t) \times \cos(2\pi f_\Delta t - \Phi) \Rightarrow LPF \Rightarrow \frac{1}{2}P(t)\cos\Phi \times R \Rightarrow \widehat{P}(t) \tag{11}$$

Hence, the steps taken to DPSK modulate and detect a spread spectrum signal are as follows:

At the Transmitter:	Differential encoding
	PN modulation
	BPSK modulation
At the Receiver:	BPSK demodulation
	PN demodulation
	Differential decoding
	Message retrieval

The results obtained from the two different modulation techniques have been compared. The main reason to compare the two different modulation techniques is to find out what is the best and the most suitable technique for this particular system.

Transmitter Design for ASK Modulation

The spread spectrum transmitter is designed in two stages consisting of the radio frequency component and the intermediate frequency component. The radio frequency component is used to up-convert the intermediate frequency for wireless transmission at 915 MHz.

The primary purpose of the spread spectrum transmitter is to convert a narrow-band digital signal to a wide-band, noise-like, digital signal through pseudo random noise modulation. This wide-band digital signal is then used to modulate a sinusoidal carrier to produce an ASK modulated signal at the intermediate frequency. Before modulation occurs, the input string is first converted into binary and then a packet is created complete with preamble, start and stop characters, and the input string. Secondly, the packet is pseudo random noise modulated. ASK modulation is used since baseband information is not suitable for wireless transmission.

The spread spectrum transmitter design consists of multiple components that function together as shown in *Fig. 1.* Each component is designed using MATLAB. The user input on the transmit side is in the form of an ASCII string.

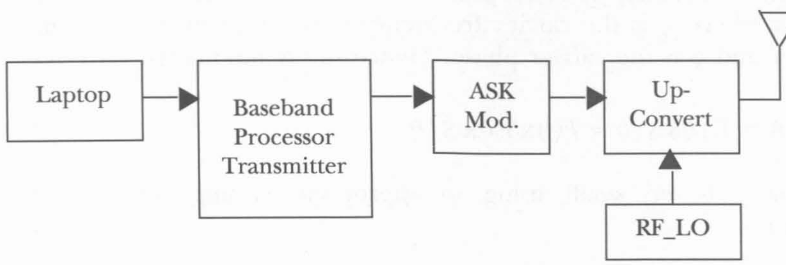

Fig. 1: Transmitter design model

This data is converted to binary and is included in the transmitted packet.

The mode of transmission employed in most spread spectrum systems is packet transmission. The general packet structure includes preamble, a start character, the message data, and a stop character (described in the Simulator section in *Fig. 5*). Preamble is used to ensure that the receiver has sufficient time to synchronise with the transmitted pseudo random noise sequence. The stop and start characters are added to ensure that the receiver does not need to know when transmission begins or ends.

Normally, the preamble consists of two hundred binary one bits. The start character consists of eight zero bits added at the front of the input binary message and after the preamble. The stop character consists of sixteen one bits and is appended to the end of the packet to form the packet for transmission.

Transmitter Design for DPSK Modulation

The simulation of this system is done by breaking up the whole system into various blocks. Each of these blocks has its own function and finally all these blocks are put together to form the simulation model. The various functions to represent these blocks are written in MATLAB M-files. The blocks of this system are shown in *Fig. 2.*

The first block is the **Text to ASCII** block. This block serves to convert the

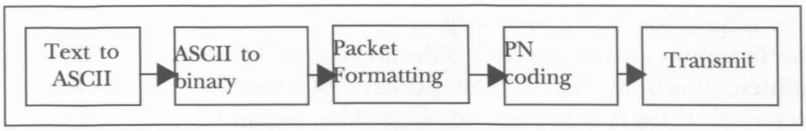

Fig. 2: Blocks in the transmission system

text data received into its ASCII code representation. To achieve this purpose, a function is written in MATLAB M-files with the name ***Te-xt2Asc***. This function uses a MATLAB built-in function ***abs***. This function changes string into a decimal representation. This representation is not equal to the ASCII code but for the simulation purpose, this representation would be sufficient. After changing the text to its ASCII format (for simulation purposes, a different decimal representation), the decimal numbers are converted into their binary equivalent. This is achieved by putting the information in the second block. This block, ***ASCII to binary***, converts any decimal number into its binary representation. This is to meet the requirement of a digital system. This block utilises the function ***de2bi***. The output of this function is a matrix with 7 columns and number of rows that are equal to the number of letters in the alphabet and/or numbers and/or punctuation. The output of this block is channelled to the third block.

This block would do all the necessary addition of bits to mark the header, tail and others. This block should actually add the headers, tail and also put the information into organised packets of fixed number of bytes / bits. Because all the previous functions have fixed the number of bits representing all possible data to 7 bits, formatting has been made simpler. So in this block only the header and tail are added. The header is placed at the start of the information being sent and the tail is appended at the back. For simulation purposes, the header has been set to 10 contiguous '1's and the tail to 6 contiguous '0's. It is used just to simplify the simulation process (since only one channel is used). There is no possibility that the information would contain these sequences of '1's and '0's because of the choice of representing the various letters in the alphabet, punctuation and numbers. The number of bits representing each data is 7 as stated earlier and thus '0's would not be sufficient to represent any of the sent data. Besides, all '0's would mean that the decimal equivalent is also '0' and there is no data represented by this value. As for the 10 '1', the equivalent value in decimal is '1023' and this also has no meaning. Additionally the decoding of this information does not look at the bits at all, for it counts the number of bits from the start of reception. This will be explained further in the reception section.

Receiver Design for ASK Modulation

The spread spectrum receiver is designed in two stages consisting of the radio frequency component and the intermediate frequency component. The radio frequency component is used to down-convert to the intermediate frequency for data acquisition and processing.

The functions in the receiver side are basically the inverse of that of the transmitter shown in *Fig. 3*. The primary purpose of the spread spectrum receiver is to convert a wide-band, noise-like, signal to a narrowband digital signal. The received wide-band digital signal is demodulated to produce a spread baseband digital signal. These spreads signal is then demodulated to obtain the original transmit message.

The received signal is acquired using analogue to digital conversion at the intermediate frequency. Based on the input signal level, the sampled signal is amplified accordingly to obtain a two-volt peak-to-peak waveform. This is a form of automatic gain control employed in many receiver designs. The signal is in the form of an ASK modulated waveform. By detecting the amplitude change, the ASK signal is demodulated and converted to a baseband digital waveform.

Amplitude modulation is the result of the variance of the instantaneous amplitude of the received signal with time. The next step in the recovery of the received message is pseudo random noise sequence synchronization.

In order to ensure the receiver has enough time to synchronise, a preamble is appended to the transmitted message. The digital waveform is sampled over one bit interval, composed of seven chips, and correlated to the locally generated pseudo random noise sequence. From this, the level of correlation is measured. If the level is below a set threshold, the data is discarded and a new sample of the received data is taken. Simultaneously, the locally generated pseudo random noise sequence is shifted forward to provide better probability so that the received signal correlates to the pseudo random noise sequence. This process continues until synchronization is achieved. At this point, pseudo random noise demodulation occurs and the de-spread digital data is recovered. The receiver then parses the data to look for the start and stop characters and converts the extracted binary data into a string. This string should be the string entered at the transmitter end earlier.

Receiver Design for DPSK Modulation

The reception model is relatively just the opposite of the transmission model. It has several blocks too and these blocks tend to reverse the process of its peer in the transmission block. All blocks are independent of each other. The preceding or proceeding blocks do not have any knowledge of what has been

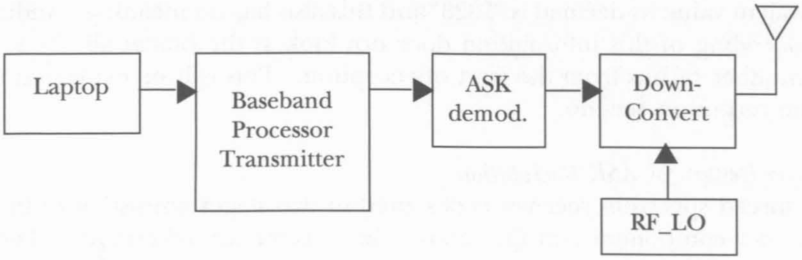

Fig. 3: Receiver design model

done at a particular block. This is done in order to ensure that if in any case a problem arises, the source of the problem could be easily located without having to go through every part of the whole system. It would suffice to just concentrate on the particular block that is causing the problem. A diagram of

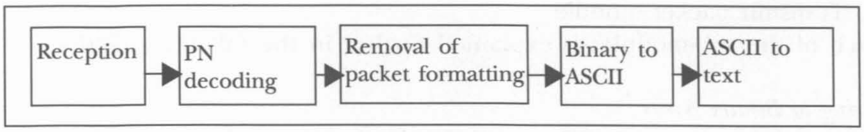

Fig. 4: Blocks in the reception system

the blocks in the reception model is shown in *Fig. 4* below.

As in the transmission model, this block - which is analogous to the Transmit block in the transmission model-is the most complicated of all the blocks in the reception model, though not as complex as the transmit block. The Reception block receives the transmitted signals from the channel and demodulates it to the baseband signal. This model represents a basic transmission and reception model.

In this block, several functions have been referred in order to achieve the desired results, Viz as Rx, Detect, and Diff dec. In fact, Rx constitutes the main function called in this block. Rx then calls the other functions, Detect and Diff dec, in order to demodulate the incoming signal. Diff dec performs differential decoding in accordance with the differential decoded incoming signal. This would then result in the signal being differentially decoded and this signal would then pass to the subsequent block, i.e. PN Decoding block.

THE SIMULATOR

Software development is broken down into two sub sections: transmitter and receiver. The receiving end is basically the opposite of the transmission end. The software is designed using the modular designing technique, i.e., a big task is subdivided into smaller tasks. Both transmitter and receiver have multiple modules, which work together to achieve the required functionality.

Transmitter Functions

The transmitter allows a user to input a text message to be transmitted. This text message is an array of ASCII character. Each character is converted into its ASCII representation. After this conversion, all the characters are stored in a binary array. This binary array of data is then converted into one or multiple packets depending upon the length of the entered message. Each packet is then PN coded and then stored in a two-dimensional array. The output two-dimensional array contains a packet in every row.

To achieve all the above-mentioned functionalities, the software is broken into the following sub modules:

* String to binary array module
* Arrangement of various matrix sizes $m \times n$ into $l \times (m*n)$ matrix form (matrix resizing) module
* Packet formatting module
* PN code data module
* Transmit packet module

Each of the sub-modules is explained further in the following sections.

String to Binary Array

This module takes an ASCII string input by the user and outputs a binary array. The output array contains binary representation of all the characters.

Matrix Resizing

This function resizes whatever size or shape of the matrix of the input binary array say $m \times n$ into $1 \times (m *n)$ matrix. This is done because all the input data at the latter stage will be required to be of this type.

Packet Formatting

The binary data array from the "String To Binary Array" module is broken into one or multiple packets. A packet consists of a header segment, data segment

Header segment	*Data segment*	*Stop Bit (s)*

and a stop bit segment.

The 'data length per packet' determines the number of data bits to be put in one packet. The 'stop bit(s)' sequence is set to six consecutive 0. The preamble is repeated in a packet ten times by default. The repetitions are used for packet synchronization in the receiver module. The output of this module is a double array, which contains a packet per row and number of packets made.

PN Coder transforms an input binary array into a PN coded array. The PN code forms the input to this module along with the binary data array.

Transmit

This function transmits the packetized data. It generates samples of a bandpass waveform using Amplitude Shift Keying (ASK) modulation technique. The carrier frequency modulates the baseband signal.

This module takes in a packet of information and creates a continuous waveform using the packet data bits as values for the waveform. This waveform is sent to the ASK modulator and then sent out to the channel.

The receiver module starts off by collecting a requested number of data packets. This large array of data is stored in the 'receive buffer array'. Once the array is full, the data is sent for processing. This processing includes non-coherent detection technique. Therefore, it needs to process the collected data through two – decoding/demodulation process.

The first is ASK demodulation and the second is PN decoding. Note that before PN decoding can take place, synchronization between the received data and the PN code, needs to be done. The synchronization module is written and tested for a non-coherent simulation.

Once the PN code synchronization has taken place, the PN code is applied to the received data. The output of the PN code demodulation is the packetized binary data. Once the received data is obtained and stored in an array, the search for the preamble begins. The user can control the data length variable.

The data bits, once found, are converted into the ASCII value they represent. The ASCII number is translated into an ASCII character and displayed to the user.

To achieve all the above-mentioned functionality the software is broken into the following sub modules:

* Receive packet module
* Matrix resizing module
* PN decode data module
* Packet deforming module
* Inversion of matrix sequence module
* Binary to decimal module
* ASCII to string module

Receive

This function receives the transmitted data. The received data is then BPSK demodulated to get the PN coded packetized data.

PN Decoder

The received data bits are PN coded. Before doing any further processing, PN code synchronization is essential for decoding. This module makes use of shift registers to achieve synchronization. A PN coded data bit when multiplied with the PN code will result in all the chips being in the same state. This property is used for synchronization in this module. The received data is compared bit by bit. When a match is found (all the chips are of the same state) the search ends and the index of the match is stored for the next module.

The decoding starts by collecting the specified number of chips into a bit. The number of chips corresponds to the length of the PN code array. Each chip in a bit is multiplied with its corresponding chip from the PN code. All of the data bits are decoded in this fashion.

Packet Deforming

This module expects data that has been completely decoded. The purpose of this module is to extract the data field out of the received and decoded data. The module starts by finding the first preamble match. When a match is found, the location of the data portion of the packet is still unknown. The only piece of information known to the “depackc 1 “ module is that the data section is immediately after the preamble.

Inversion of Matrix Sequence

This function is important for arranging the data back to its original order. This is because the data is received in the inverted order. So, in order to correct this, we need to invert it back to its original order.

Example:

A = (1 0 0) , "invert" A = (0 0 1)

Binary to Decimal

This function converts the binary data into its decimal equivalent. The decimal value actually represents the ASCII code that has been predetermined. This output is then used in the next module.

ASCII to Text/String

This function converts the ASCII code back to the original data first keyed in by the sender. The function can be found in the usual MATLAB function library.

RESULTS AND DISCUSSIONS

A) ASK Modulation

The system developed above shows good functionality. It is able to receive, without error, the original information transmitted on the transmitter end. This is due to the fact that this system is simulated using AWGN channel within indoor environment. With ASK, this particular modulation is susceptible to any form of noise and distortion.

For ASK to differentiate between two different bits ('0' and '1'), a comparison with a threshold value is requested. So when any form of noise or distortion sets upon, it affects the accuracy of the matched filter deciding in which bit ('0' or '1') that particular signal falls into.

The simulation program is basically divided into two major groups, the Transmitting End and the Receiving End. Generally, the processes in the receiver are the inverse of the transmitter.

At the transmitter, the first step is to convert the string or text that has been keyed in by the user into decimal (in this case, the word is TESTING). The decimal value is actually in the form of ASCII code. For instance, 1 is represented by ASCII code of **49** and **a** is represented in ASCII as 97. This ASCII code is then converted into its binary equivalent because we need to transmit and receive based on a digital system. After that, the binary data sequence is arranged into a single row matrix form. This is to allow easier interpretation and for facilitating the next process, which is to packetize the binary sequence. The packet is completed with header, data and tail (stop bits). The header is meant for the packet to recognize where the last packet stopped and this is where it has to continue. The stop bits are to tell where that particular packet ends. Next, the packetized binary data is spread by applying PN code for security. The spread data is the Amplitude Shift Keyed and transmitted.

At the receiver, the data is first ASK demodulated to get the spread data. Next, the data is arranged into the form of a single matrix to facilitate the next processes. Later, the data is Pseudo random noise decoded to get the packetized binary data. It is then depacketized to get the binary sequence data. At this point, the data has to be arranged again into the form of a single row matrix. Next, the data is rearranged into a 7-column matrix. This is because 7 bits represent each letter, and thus, this makes it easier to interpret and process. Next, the binary sequence data is converted into decimal, or in other words, the ASCII code. Later, the ASCII is converted into string or text. *Fig.*

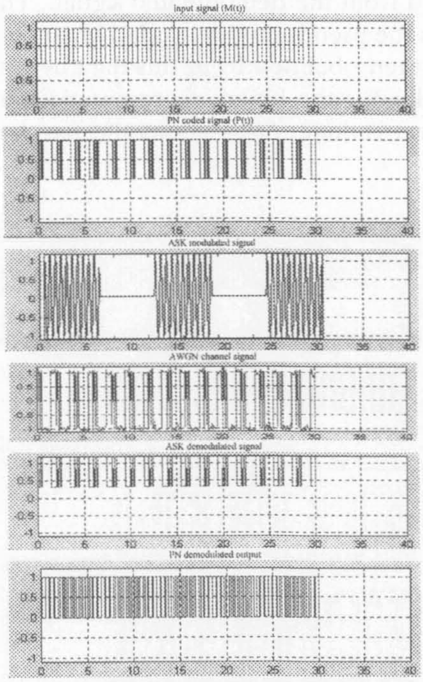

Fig. 6: The waveform results for ASK Modulation

6 shows the waveform results in time domain.

B) DPSK modulation

Simulation is done in the MATLAB environment. All functions are written in MATLAB M-files and the simulation progressed as a command line structure. A test with a few characters has been made and it was observed that transmitting and receiving blocks performed the expected operations on the data. The resulting output is found to be an approximation of the input because of the low noise and channel limitations.

It is clear from the simulation results that the expected objective has been achieved. In the simulation, differential coding is used before binary phase shift keying is done. This is to satisfy the condition of a DPSK system.

The simulation results showed that all the blocks in the transmission and reception models served their purpose well and each process is executed independently of the other processes. First, the data is converted from text to its ASCII code and then to the binary equivalent. This binary sequence is then rearranged into the form of single row matrix. This is to facilitate the transition into the next function. Next, packet formatting is done to the binary sequence. This is then coded with the pseudo-random noise sequence and finally modulated and transmitted.

At the receiving end, the transmitted signal is demodulated and the original binary data is retrieved from the demodulated signal. This demodulated signal is actually the same as the pseudorandom coded binary data in the transmission end. The payload is then obtained from this data by getting rid of the packet formatting. This payload is the binary representation and conversion back to decimal and finally text is done. *Fig.* 7 shows the waveform results in the time domain for DPSK.

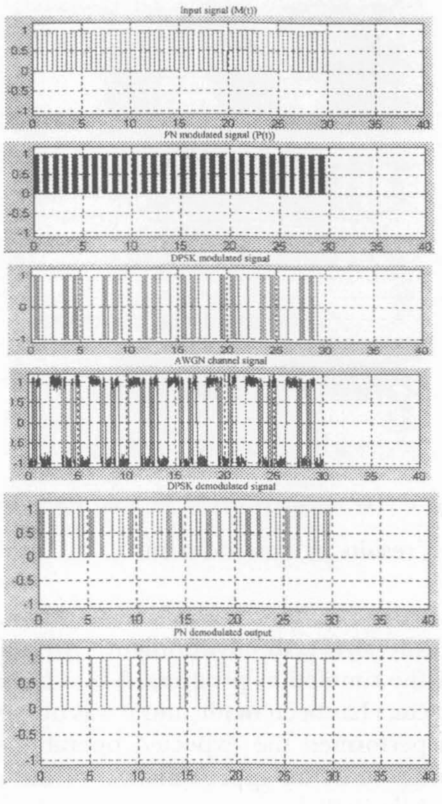

Fig. 7: The waveform results for DPSK Modulation

CONCLUSIONS

This paper describes the hardware model of the transmitter and receiver using Wireless Spread Spectrum Communication techniques using MATLAB communication toolbox. The aim of implementing such a system is to ensure a high security of the transmitted massage because spread spectrum signals are known to be noise-like and hard to detect. They are also hard to intercept and jam. These Low Probability of Intercept (LPI) and anti-jam (AJ) features are why the military personnel have been using Spread Spectrum for many years even until today. Presently IMT-2000 uses CDMA, a form of spread spectrum (SS). In future, an error control using block code (FEC) can be implemented to correct the channel error for both indoor and outdoor wireless spread spectrum communication system (WSSCS).

REFERENCES

BLAHUT R. E. 1983. *Theory and Practice of Error Control Codes.* Addison-Wesley Publishing Company, Inc.

CHEE-MUN ONG. 1998. *Dynamic Simulation of Electric Machinery: Using MATLAB/SIMULINK.* Prentice Hall.

CHU L. C. and U. MITRA. 1998. Performance analysis of an improved MMSE multiuser receiver for mismatched delay channels. *IEEE Transactions on Communications* **46** (**10**).

Error Correction with Hamming Codes, http://www-dos.uniine.msk.ru/tech1/1994/er_cont/hamming.html

FREEMAN R. L. 1995. *Practical Data Communications.* N.Y.: Wiley-interscience.

GLISTIC S.G., T.J. POUTANEN, W. W. WU, G. V. PETROVIC and Z. STEFANOVIC. 1999. New PN code acquisition scheme for CDMA network with low signal to noise ratio. *IEEE Transactions on Communications* **47**(**2**).

HAYKIN S. 1994. *Communication Systems.* N.Y.: John Wiley.

HOST-MADSEN A. and K. S. CHO. 1999. MMSE/PIC multiuser detection for DSCDMA system with inter- and intra-cell interference. *IEEE Transactions on Communications* **47**(**2**).

JOHN G. PROAKIS and MASOUD SALEHI. 1998. *Contemporary Communication Systems using MATLAB.* PWS Publishing Company.

MESSIER G. G. 1998. Private Communications, TRLabs, Calgary, March 12.

PETERSON R. L. and R. E. ZIEMER. 1985. *Digital Communications and Spread Spectrum Systems.* N.Y.: Collier Macmillan.

Power Spectral Density Curve, http:llwww.tapr.org/sslqexss .

PRASAD R. 1996. *CDMA for Wireless Personal Communications.* Norwood: Artech House Publisher.

QIAO C. 1998. A universal analysis model for photonic Banyan networks. *IEEE Transactions on Communications* **46(10)**.

The American Radio Relay League, The ARRL Handbook for Radio Amateurs, 1996.

Transmission Line Attenuation Chart, http://lwww.tapar.orgltaprlhtmlcoax.html.

WILHEIMSSON L. and K. S. ZIGANGIROV. 1998. Analysis of MFSK frequency-hopped spread spectrum multiple-access over a Rayleigh fading channel. *IEEE Transactions on Communications* **46(10)**.