

Digicomm: A MATLAB-Based Digital Communication System Simulator

C.Romila¹

¹Project Student, Department of E.C.E, Vasireddy Venkatadri Institute of Technology, Nambur, AP, India

Abstract: A simple digital communication virtual tool, *DigiComm*, is introduced. This tool visualizes the transmission of binary data in different channels using frequency-shift keying (FSK) and accessing the shared media via frequency-division multiple access (FDMA). The transmitter part generates binary data, and modulates the frequencies of both carriers of each channel. The signals of each channel with noise added are distinguished from each other using bandpass filters (BPF) at the receiver, and converted to binary data by counting the zero crossings at the receiver part. Signals at the transmitter and receiver can be displayed in the time or frequency domains. Bit-error rates (BER) are calculated for a given signal-to-noise ratio (SNR) by comparing the received bits with the transmitted bits.

Keywords: Frequency shift keying (FSK); frequency division multiple access (FDMA); modulation; noise; digital communication system; simulation; visualization; bit error rate (BER); MATLAB; graphical user interface (GUI); signal-to-noise ratio (SNR).

I. Introduction

Technical challenges posed by system complexities in engineering today require a range of multidisciplinary, physics-based, problem-matched analytical and numerical skills. Physics-based modeling, computer-based simulations, and experimentation via real and virtual labs have become the key issues of current engineering education. These necessitate the establishment of an intelligent balance between real and virtual experimentation [1, 2]. Triggered by these thoughts, we have developed and hosted several virtual tools (simulation packages) to the use of readers in the *Magazine* for the last decade or so [3-13]. These packages can be used as effective engineering tools, as well as virtual lab environments in Under graduate- and graduate-level electromagnetics lectures.

The virtual tool *SNELL_GUI* [3] was designed to visualize ray paths through a propagation medium with variable atmospheric refractivity over simple terrain profiles. The package *RAY_GUI* [4] can be used to visualize individual rays and modes, and their field contributions inside a two-dimensional non-penetrable parallel-plate waveguide. The package *ANTEN_GUI* [5] enables the user to investigate beamforming and beam-steering abilities, and also radiation patterns of planar arrays of isotropic radiators. *SSPE_GUI* [6] simulates short-and long-range radio wave propagation over non-smooth Earth surfaces and through an inhomogeneous atmosphere. The *TDRMeter* package [7] was intended to visualize pulse propagation and reflections from various terminals and faults (discontinuities). The package can be used as a time-domain reflectometer to test different types of terminations and/or faults by analyzing the time history of the simulated pulses.

Another *MATLAB*-based virtual tool, *BESSEL_GUI* [8], computes the values of the Bessel function by asymptotically evaluating different types of Bessel integrals using high-frequency methods such as the stationary-phase method, steepest-descent-path evaluations, and even asymptotics. *MWFilter Designer* [9] is another visualization package designed to automate the design process of transmission-line, microstrip-line and lumped-element filters. A *MATLAB*-based package *ComplexGUI* [10] was developed to visualize elementary multi-valued complex functions, elementary multi-valued complex functions, $\sqrt{w} \ln w$, $\arcsin w$, and their mappings, Riemann surfaces, branch points and cuts, etc.

Besides the above-mentioned *MATLAB*-based packages, two *LabVIEW*-based virtual tools were also prepared. The first *LabVIEW* virtual instrument, *DOGUS_FFT* [11], was designed for numerical Fourier-transform calculations, and to understand Fourier-transform difficulties such as aliasing, spectral leakage, and scalloping loss. The second *LabVIEW*-based package, *AnalogMod* [12], was prepared to visualize elementary analog-modulation techniques. This tool investigates the time- and frequency-domain behaviors of modulated signals, the effects of noise, and the effect of different types of hardware- or software-implemented filters.

A novel *MATLAB*-based package, *DigiComm*, completes the virtual lab set. It can be used to simulate a digital

communication system, and is introduced in this paper. In this package, digital baseband signals in four different channels use frequency-shift keying (FSK) in modulating frequencies of analog carriers. All four channels use frequency-division multiple access (FDMA) and share the transmission medium, where the total bandwidth is alienated into four discrete bands, and each band is allocated to a single channel. Noise is added to the transmitted signal in order to simulate a realistic system. Non-ideal filters are used at the receiver in distinguishing channels from each other. All of these MATLAB- and LabVIEW-based packages can be downloaded from <http://modsim.dogus.edu.tr>. This Web site also includes Web-based versions of some of these packages [13] that can be accessed on the Internet online from anywhere in the world, without having to have installed MATLAB or LabVIEW, by only using a Web browser. The Web-based version of the DigiComm package introduced here will appear there soon.

Note that the tutorials introduced in [14-17] discuss fundamental issues, such as the meanings of the numbers we use, Fourier transforms in the analog and digital worlds, stochastic modeling of communication and interference signals, as well as noise and clutter, which are all casual and necessitate stochastic modeling. The reader is strongly advised to revisit these issues for a better understanding of this tutorial, and to use *DigiComm*.

II. Digital Communication Systems

Digital communication is the physical transfer of a digital bit stream over a communication channel. If the baseband signal expresses the digital bits transmitted without any modulation, as a sequence of electric or light pulses, it is called a baseband transmission. It is called a pass-band transmission if the digital baseband signal is converted from digital to analog using a digital modulation method.

Digital modulation is the modulation of an analog carrier by a digital baseband signal. The most fundamental digital modulation techniques are amplitude-shift keying (ASK), frequency-shift keying (FSK), phase-shift keying (PSK), and quadrature amplitude modulation (QAM) [18, 19]. In amplitude-shift keying, the amplitude of an analog carrier is modulated by a binary signal. In frequency-shift keying and phase-shift keying, the binary signal modulates the frequency and phase of an analog carrier, respectively. Quadrature amplitude modulation is a grouping of amplitude-shift keying and phase-shift keying, where an infinite number of at least two amplitudes and at least two phases is used. The simulation package introduced in Section 3 uses the digital modulation technique of frequency-shift keying. A block diagram of frequency-shift keying modulation with two levels is given in Figure 1.

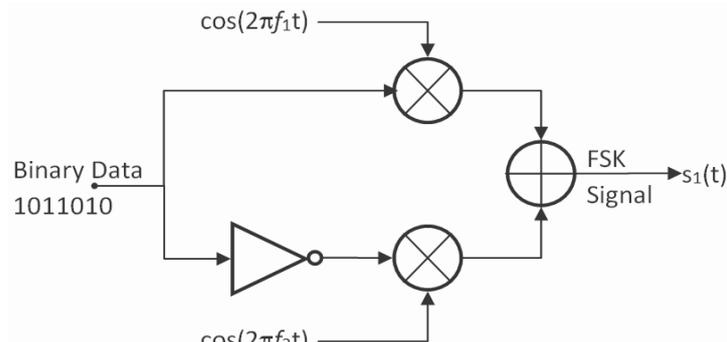


Figure 1. A block diagram of frequency-shift keying (FSK) modulation.

Multiple-access methods enable several users/terminals to access and share the same transmission medium and to transmit over it. Fundamental types of multiple-access methods are space-division multiple access (SDMA), time-division multiple access (TDMA), frequency-division multiple access (FDMA), and code-division multiple access (CDMA). Space-division multiple access uses different point-to-point wires for different channels. Time-division multiple access assigns different time slots for different channels, whereas frequency-division multiple access reserves different frequency bands for different channels. In code-division multiple access, several channels simultaneously share the same frequency spectrum, which is much higher than the data rate of each of the transferred bit streams, and different spreading codes are utilized. The simulation package introduced in Section 3 uses the frequency-division multiple access multiple access method. Simple block diagrams of frequency-shift keying plus a frequency-division multiple access transmitter and receiver are plotted in Figures 2 and 3.

Electronic circuits and receivers are affected by a variety of noise sources. Thermal noise (also known as white noise, Gaussian noise, internal noise) dominates the others for

frequencies above a few hundred MHz's. Thermal noise is directly proportional to the receiver bandwidth, and can be calculated as

$$N_f = kTB [W], \quad (1)$$

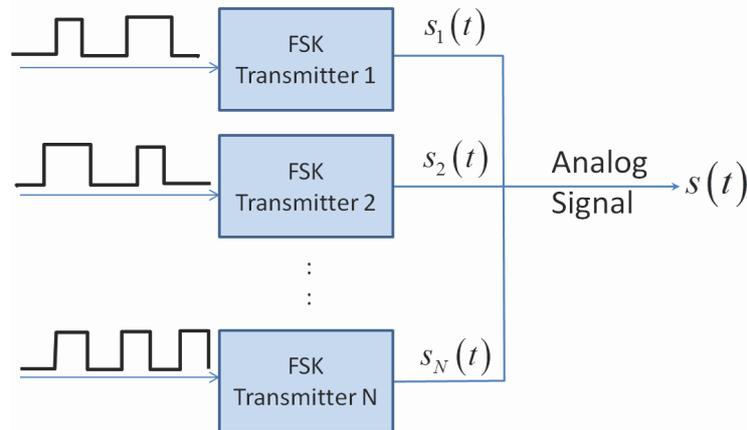


Figure 2. A simple block diagram of a frequency-shift keying (FSK) transmitter.

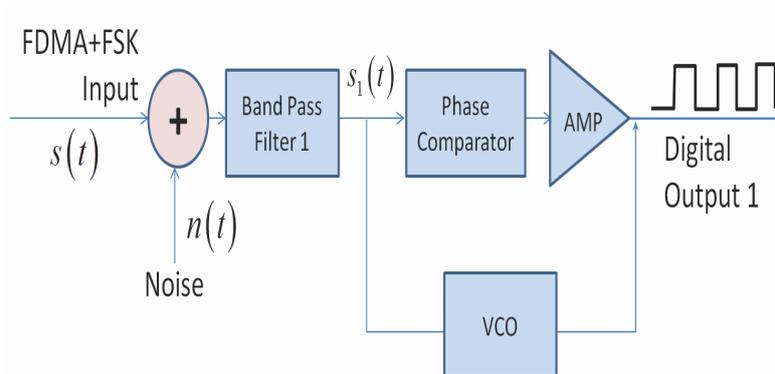


Figure 3. A simple block diagram of a frequency-shift keying plus frequency-division multiple access (FSK+FDMA) receiver.

where B [Hz] is the noise bandwidth, k is Boltzman's constant ($k=1.38 \times 10^{-23}$ J/K) and T is the temperature in Kelvin. Since $kT = -204$ dBW/Hz at 300°K , thermal noise power can also be calculated as [16]

$$N_f = -204 + B [\text{dB}]. \quad (2)$$

Note that noise is random but uncorrelated. Its power spectrum is flat, and the spectral density is $S(f) = kT$. The noise-voltage samples (i.e., noise amplitudes) erratically fluctuate, and have Gaussian probability distribution characteristics [16]. The *DigiComm* virtual tool was designed using *MATLAB*'s *GUIDE* (*GUIDEsign*) command. The synthetic simulation data is generated as follows

Transmitterside: (see Figure 2)

- A user-specified number of random bits is generated for the four channels. A bit is generated using the *rand* command. The *rand* command generates a uniformly distributed real number between zero and one. If the generated number is greater than or equal to 0.5, the generated bit will be ONE, otherwise it is ZERO.
- The bit stream of each channel is fed into the frequency-shift keying modulator and the $s_1(t)$ analog signal is generated. Here, $s_1(t) = \cos(2\pi f_i t)$, where $f_i = f_1$ if the bit is ZERO and $f_i = f_2$ if the bit is ONE.

- The frequency-division multiple access method is used to generate the transmitted signal, $s(t)$.

The transmitted signal will then be $s(t) = s_1(t) + s_2(t) + s_3(t) + s_4(t)$.

Receiver side (see Figure 3)

- Use the rand command and generate Gaussian distributed random noise $n(t)$ for a given SNR. Form the received signal, $r(t) = s(t) + n(t)$.
- Distinguish channels using proper bandpass filters.
- Finally, feed $r(t)$ through a phase comparator (i.e., count the zero crossings for each bit), and decide whether the received bit is a ZERO or ONE.

The DigiComm Package

The MATLAB -based virtual tool DigiComm, with the front panel displayed in Figure 4, was prepared to visualize the signals in a frequency-shift keying plus frequency-division multiple access based communication system. The front panel of the DigiComm package is divided into two parts. The upper part belongs to the four-channel transmitter, and contains information. A graph displays channel signals

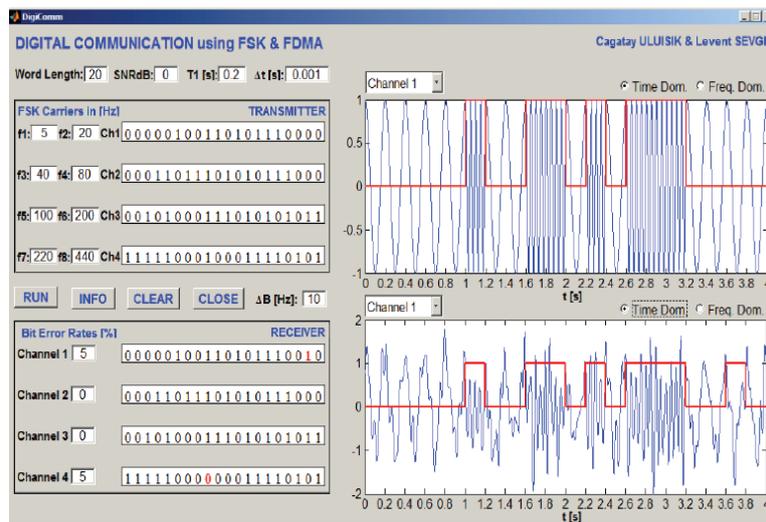


Figure 4. The front panel of DigiComm in the time domain. The transmitted and received signals of Channel 1 are displayed in blue, and the corresponding bit sequences are displayed in red.

In the time or frequency domain. The total (all-channel) signal can also be displayed. The lower part belongs to the receiver, with the same display capabilities. Operational parameters are positioned on top of the transmitter block, while command push buttons are inserted between the transmitter and receiver blocks. The Info push button opens a MATLAB help window, which includes explanations on how to use the package. This can also be done by typing help DigiComm at the MATLAB command line. The Clear Graph button clears the graphs; the Close button exits the package.

Arrays of uniformly distributed random binary numbers are generated for the four transmitter channels, and are displayed in the transmitter block once the Run button is pressed. The length of the array (i.e., the number of ZEROs and ONES) is supplied by the user as the Word Length. In Figure 4, the Word Length was 20, and all channels confined 20 random bits. The generated bits of any of these channels can be selected and plotted in red on the upper graph. The Word Length can be selected to be higher, but only the first 20 bits will be displayed in the text boxes; all of them will be plotted on the graph. The time required to send each bit (bit duration = $1/T$ in seconds) is specified by the user. The Word Length duration (T) will then be $Word\ Length \times T_1$. This is $T=4s$ for the example displayed in Figure 4.

The frequency-shift keying (FSK) technique is used to modulate the frequency of an analog sinusoidal carrier by the binary sequence of each channel. Two discrete frequencies ($1f$ and $2f$) are specified for each channel by the user. ZEROs (ONES) are transmitted over the lower (upper) frequency of each channel. The carrier frequencies should be specified as $f_1 < f_2, f_3 < f_4, f_5 < f_6, f_7 < f_8$. The frequency-modulated analog signal

sequence is plotted on the upper graph in blue, together with the red-plotted binary data, as in Figure 4. A pop-up menu on top of the graph allows the user to select one of the four channels to be displayed on the graph. The user may also select plotting the modulated signal in the time or frequency domain. The radio buttons located on the top are reserved for this purpose. In Figure 5, the modulated signal at the second channel is plotted in the frequency domain.

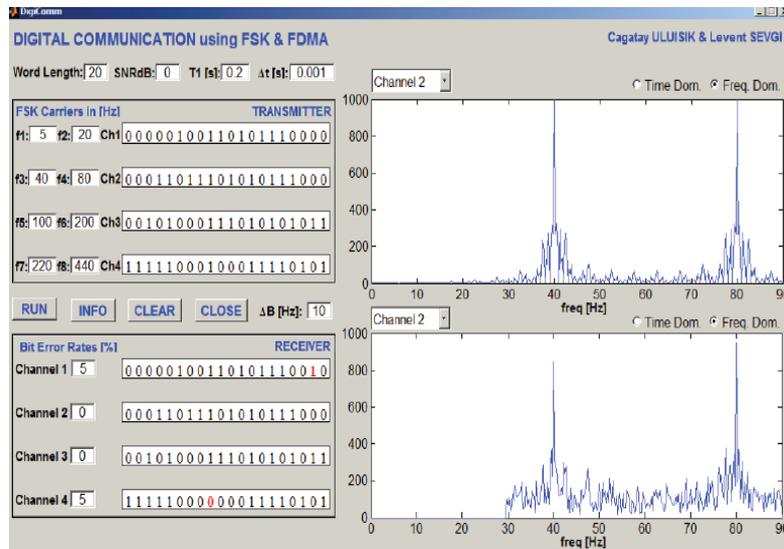


Figure 5. The front panel of DigiComm in the frequency domain. The transmitted and received signals of Channel 2 are displayed in the upper and lower graphs, respectively.

The received signal contains the transmitted signal plus noise. White noise is added to all of the four transmitter channels to pretend a more-realistic communication system. The noise power is specified through the signal-to-noise ratio (SNR). The user supplies the signal-to-noise ratio in dB with the SNRdB parameter, and the signal-to-noise ratio is computed via $SNR=10^{SNRdB/10}$. Since the signal is sinusoidal with amplitude $V_m=1$, the signal power is equal to $P_S=0.5V_m^2=0.5$. The power of the noise is then determined as $P_N=P_S/SNR$. For each channel, the noise is obtained by generating an array of normally distributed random numbers (using MATLAB's randn command) with zero mean and unit variance. Multiplying this array by $\sqrt{P_N}$ yields random noise samples with the specified signal-to-noise ratio. This noise is then added to the signal. The total signal with noise further can be displayed on the upper graph by selecting Channels All from the pop-up menu. The sum of four signals with noise added (for SNRdB=0) is displayed in Figure 6a in the time domain, and its frequency spectrum is given in Figure 6b for the parameters specified in Figure 4. It is difficult to say anything about the signal in the time domain. However, the frequencies of the carriers can be easily observed in the frequency spectrum.

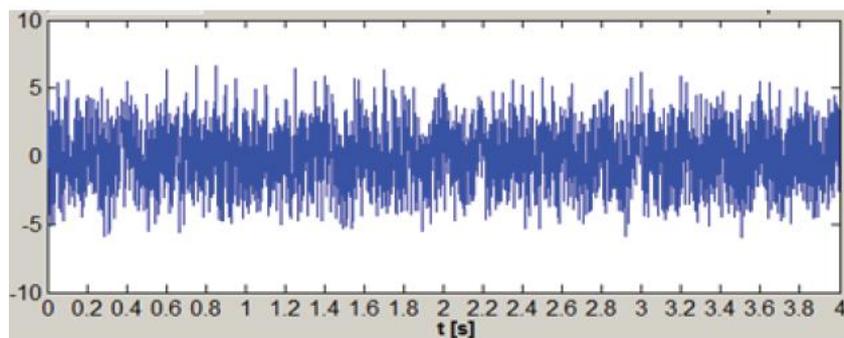


Figure 6a. The total frequency-shift keying plus frequency-division multiple access (FSK+FDMA) signal plus noise in the time domain.

Four channels are combined according to the frequency-division multiple access (FDMA) technique. The total band-width is divided into four discrete bands for four channels. The frequency band $[f_1-\Delta B, f_2+\Delta B]$ is used by Channel 1, $[f_3-\Delta B, f_4+\Delta B]$ is used by Channel 2, $[f_5-\Delta B, f_6+\Delta B]$ is used by Channel 3, and $[f_7-\Delta B, f_8+\Delta B]$ is used by Channel 4. The parameter ΔB is used as the inter-channel guard band, and is specified by the user. If $f_1-\Delta B < 0$, then Channel 1 will use the frequency band $[0, f_2+\Delta B]$. Similarly if $f_8+\Delta B > f_{max}$,

then Channel 4 will use the frequency band $[f_7 - \Delta B, f_{\max}]$. Here, f_{\max} is the maximum noticeable frequency, and can be computed from the user-specified sampling interval, $\Delta t (f_{\max} = 1/(2\Delta t))$. The carrier frequencies $f_1, f_2, f_3, \dots, f_8$ and ΔB should be specified so as not to yield overlap among frequency bands.

Band pass filters (BPF) are used at the input of the receiver to distinguish among four channels. The pass bands of the channels are the same as the transmitter channel band-widths. The filtered signal of every channel is plotted in blue on the lower graph, in the time domain as in Figure 4, or in the frequency domain as in Figure 5. The number of zero crossings for each bit (i.e., in each T_1 duration) is counted, and the binary data is extracted. The decision for a ZERO or ONE bit is made according to this number. For Channel 1, this number should be $2T_1f_1 \times \text{zero}$ crossings for a ZERO bit, and $2T_1f_2 \times \text{zero}$ crossings for a ONE bit. The threshold is therefore set to $T_1(f_1 + f_2)$. The bit streams for the other channels are found in a similar manner, and all of them are displayed in the receiver block. They are also plotted on the lower graph in red. If a bit stream contains errors, the erroneous bits are displayed in red, and the correct bits are displayed in black. The receiver block also shows the bit-error rates (BER) for each channel. In the example displayed in Figure 5, all of the 20 bits of Channels 2 and 3 were received correctly. However, the 19th bit of Channel 1 was ONE where it should actually have been ZERO, and the 9th bit of Channel 4 was ZERO where it should essentially have been ONE. Only one bit out of 20 bits was erroneously received for Channels 1 and 4, which yielded a BER of 5%. For a 0 dB SNR, these error rates should be considered good system performance.

III. Characteristic Examples

The DigiComm package was used to visualize how the carrier frequencies; sampling interval, Δt ; guard-band parameter, ΔB ; SNRdB; and total observation period, T; affected the bit error rates. Examples are assumed in Figures 7 to 10. In all of these examples, the Word Length was 20. The first example illustrated in Figure 7 aimed to emphasize the effect of the sampling rate, Δt . According to the Nyquist sampling criterion, the sampling rate must be greater than twice the highest frequency ($f_{\max} = 1/(2\Delta t)$) of the time record. In this example, the sampling interval was specified as $\Delta t = 0.002$ s, which yielded a maximum frequency of $f_{\max} = 250$ Hz. The carrier frequencies of the first three channels were less than 250 Hz, but since $f_8 = 440$ Hz was greater than 250 Hz, aliasing occurred, and f_8 appeared in the frequency band at $500 - 440 = 60$ Hz, as shown in the upper graph. The band pass filter at the receiver passed signals lying in the band $[210\text{Hz}, 450\text{Hz}]$. Since there was no frequency component at $f_8 = 440$ Hz, all of the bits were assigned to ZERO, which corresponded to $f_7 = 220$ Hz. Even for an SNR value as high as 10 dB, Channel 4 erroneously presumed every transmitted ONE to be ZERO, and a BER of about 50% was observed. Only doubling the sampling interval, Δt , increased the maximum frequency to 500 Hz, which covered all of the carrier frequencies. No aliasing occurred, and the BER drastically decreased.

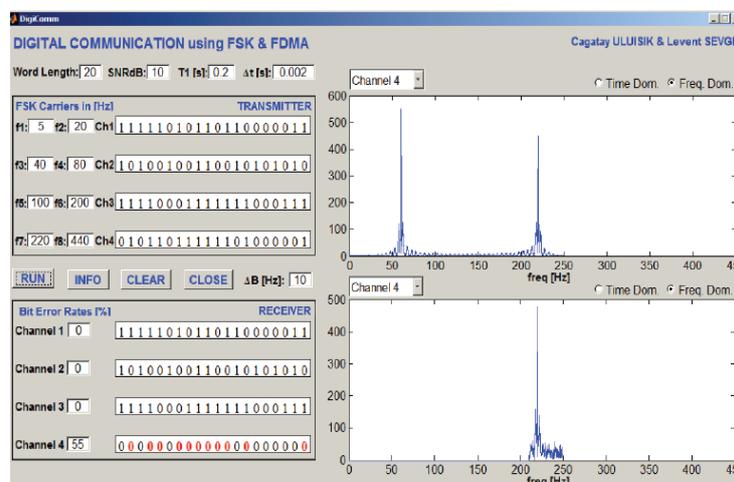


Figure 7. The effects of the sampling interval, Δt (SNRdB = 10 , T1 = 0.2 s, $\Delta t = 0.002$ s, $\Delta B = 10$ Hz, $f_1 = 5$ Hz, $f_2 = 20$ Hz, $f_3 = 40$ Hz, $f_4 = 80$ Hz, $f_5 = 100$ Hz, $f_6 = 200$ Hz, $f_7 = 220$ Hz, $f_8 = 440$ Hz).

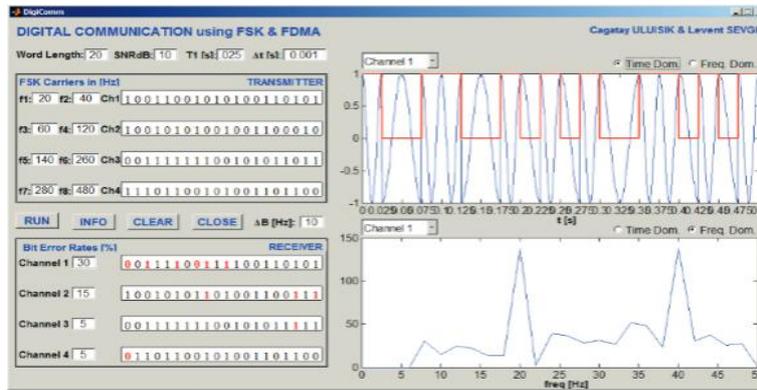


Figure 8. The effects of the observation period, T , ($\text{SNRdB} = 10$, $T_1 = 0.025$ s, $\Delta t = 0.001$ s, $\Delta B = 10$ Hz, $f_1 = 20$ Hz, $f_2 = 40$ Hz, $f_3 = 60$ Hz, $f_4 = 120$ Hz, $f_5 = 140$ Hz, $f_6 = 260$ Hz, $f_7 = 280$ Hz, $f_8 = 480$ Hz).

An example concerning the selection of the total observation time, T , is given in Figure 8. Here, the sampling interval was chosen as $\Delta t = 0.001$ s, corresponding to a maximum frequency of 500 Hz, which enclosed all of the carrier frequencies. T_1 was chosen to be 0.025 s, and the Word Length was 20. The total observation period became $T = 0.5$ s, which yielded a frequency resolution of $\Delta f = 1/T = 2$ Hz. The lower graph in the figure shows Channel 1 at the receiver in the frequency band, and the low-frequency resolution could be clearly observed. The frequency for sending the binary ZERO in the first channel was chosen to be 20 Hz. This corresponded to a period of 0.05 s, but the time T_1 for sending a bit was specified as 0.025 s. This covered only half of the cycle, and occasioned in discontinuities in the time signal. High bit-error rates were obtained, even for $\text{SNR} = 10$ dB. Only increasing T_1 would increase the total time, T , and result in better frequency resolution and a lower BER.

The next example, in Figure 9, illustrated overlapping frequency bands. In this example, the sampling interval was $\Delta t = 0.0005$ s, and corresponded to a maximum frequency of 1000 Hz. This covered all of the carrier frequencies. T_1 was chosen to be 0.2 s, and yielded a total observation period of $T = 4$ s, with a frequency resolution of $\Delta f = 0.25$ Hz. The frequency band between 210 Hz and 540 Hz of Channel 3 overlapped with the bands of the second and fourth channels. The lower graph in the figure shows the output of the filter of Channel 3, which also contained the carrier frequencies $f_4 = 220$ Hz and $f_7 = 480$ Hz of the second and fourth channels, respectively. Due to this overlapping, the acknowledged bit sequences at the second, third, and fourth channels contained a number of incorrect bits. Changing the carrier frequencies of the third channel to $f_5 = 250$ Hz and $f_6 = 450$ Hz would solve this problem.

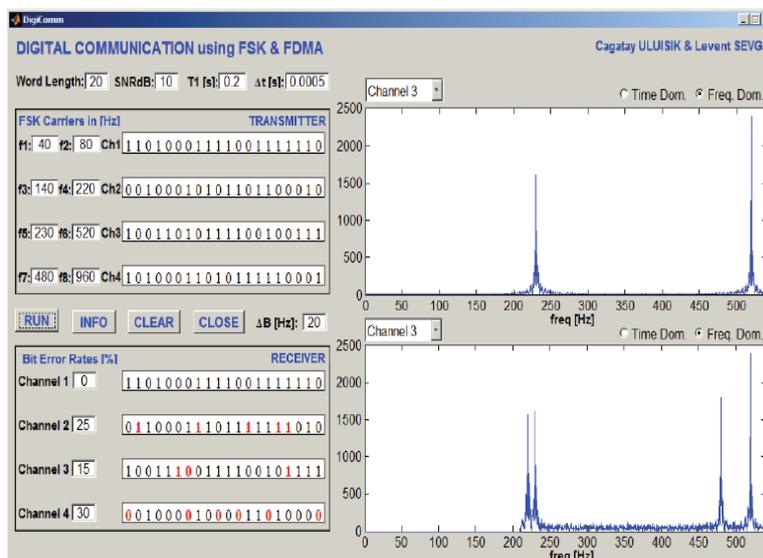


Figure 9. The effects of carrier frequencies ($\text{SNRdB} = 10$, $T_1 = 0.2$ s, $\Delta t = 0.0005$ s, $\Delta B = 20$ Hz, $f_1 = 40$ Hz, $f_2 = 80$ Hz, $f_3 = 140$ Hz, $f_4 = 220$ Hz, $f_5 = 230$ Hz, $f_6 = 520$ Hz, $f_7 = 480$ Hz, $f_8 = 960$ Hz).

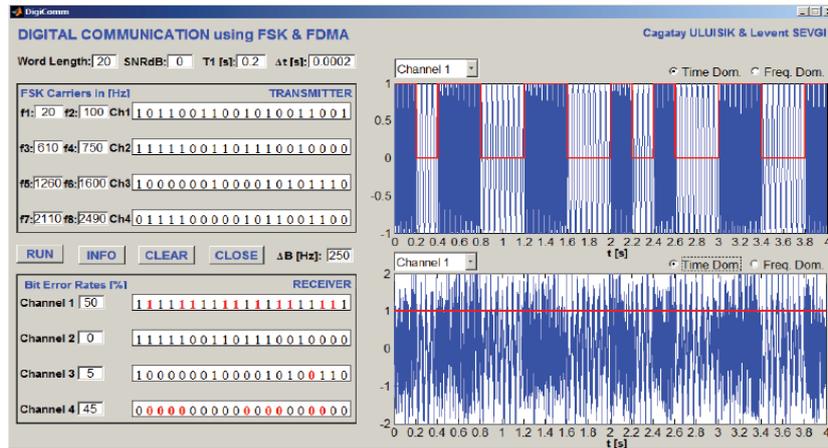


Figure 10. The effects of ΔB (SNRdB = 0, $T_1 = 0.2$ s, $\Delta t = 0.0002$ s, $\Delta B = 250$ Hz, $f_1 = 20$ Hz, $f_2 = 100$ Hz, $f_3 = 610$ Hz, $f_4 = 750$ Hz, $f_5 = 1260$ Hz, $f_6 = 1600$ Hz, $f_7 = 2110$ Hz, $f_8 = 2490$ Hz).

The last example, in Figure 10, was reserved for the effects of the guard band, ΔB . Here, the sampling interval was $\Delta t=0.0002$ s, and corresponded to a maximum frequency of 2500 Hz. It covered all of the carrier frequencies. T_1 was 0.2 s, and yielded a total observation period of $T=4$ s with $\Delta f=0.25$ Hz. For the specified $\Delta B=250$ Hz, none of the frequency bands overlapped. However, almost half of the bits of the first and last channels were incorrectly received, and all of the bits in Channel 1 were received as ONES, and all of the bits in Channel 4 were received as ZEROS. Channel 1 used the frequency band between 20 Hz and 350 Hz. The SNR was 0 dB. The carrier frequencies were at 20 Hz and 100 Hz, but white noise occupied the whole band. The frequency components of noise between 100 Hz and 350 Hz were much more than those between 0 Hz and 20 Hz, which caused every bit to be received as a ONE. Similarly, Channel 4 used the frequency band between 1860 Hz and 2500 Hz. The carrier frequencies were at 2110 Hz and 2490 Hz, but white noise occupied the whole band. Frequency components of noise between 1860 Hz and 2110 Hz were much more than those between 2490 Hz and 2500 Hz, which caused every bit to be received as a ZERO. Declining ΔB will reduce noise; $\Delta B=0$ would correspond to an ideal band pass filter. In practice, it is impossible to achieve $\Delta B=0$, but keeping ΔB as small as possible will result in better receiver performance.

IV. Conclusions

A user-friendly, simple MATLAB-based virtual tool (DigiComm) was introduced. DigiComm simulates the transmission of binary data through four different channels, using frequency-shift keying (FSK), and accessing the shared media via the frequency-division multiple access (FDMA) method. The effects of the sampling interval, carrier frequencies, noise, and filters on the bit-error rate (BER) were investigated. The tool may be used in undergraduate lectures such as “Introduction to Communication Systems,” “Analog Communication,” “Digital Communication,” “Fundamentals of Wireless Communication,” etc.

References

- [1]. L. Sevgi and C. Gökna, “An Intelligent Balance between Numerical and Physical Experimentation,” IEEE Potentials Magazine, 23, 4, October/November 2004, pp. 40-44.
- [2]. L. Sevgi, “Virtual Tools/Labs in Electrical Engineering Education,” ELEKTRIK, Turkish J. of Electrical Engineering and Computer Sciences, 14, 1, March 2006, pp. 113-127.
- [3]. L. Sevgi, “A Ray Shooting Visualization MATLAB Package for Two-Dimensional Ground wave Propagation Simulations,” IEEE Antennas and Propagation Magazine, 46, 4, August 2004, pp. 140-145.
- [4]. L. B. Felsen, F. Akleman, and L. Sevgi, “Wave Propagation Inside a Two-dimensional Perfectly Conducting Parallel Plate Waveguide: Hybrid Ray-Mode Techniques and Their Visualisations,” IEEE Antennas and Propagation Magazine, 46, 6, December 2004, pp. 69-89.
- [5]. L. Sevgi and Ç. Uluşık, “A MATLAB-Based Visualization Package for Planar Arrays of Isotropic Radiators,” IEEE Antennas and Propagation Magazine, 47, 1, February 2005, pp. 156-163.
- [6]. L. Sevgi, Ç. Uluşık, and F. Akleman, “A MATLAB-Based Two-Dimensional Parabolic Equation Radio wave Propagation Package,” IEEE Antennas and Propagation Magazine, 47, 4, August 2005, pp. 164-175.
- [7]. L. Sevgi and Ç. Uluşık, “A MATLAB-Based Transmission Line Virtual Tool: Finite Difference Time Domain Reflectometer,” IEEE Antennas and Propagation Magazine, 48, 1, February 2006, pp. 141-145.
- [8]. L. Sevgi and Ç. Uluşık, “A Tutorial on the Bessel Functions and Numerical Evaluations of Bessel Integrals,” IEEE Antennas and Propagation Magazine, 51, 6, December 2009, pp. 222-233.
- [9]. M. A. Uslu and L. Sevgi, “A MATLAB-Based Filter-Design Program: From Lumped Elements to Micro strip Lines,” IEEE Antennas and Propagation Magazine, 53, 1, February 2011, pp. 213-224.

- [10]. Ç. Uluşık and L. Sevgi, "A MATLAB-Based Visualization Package for Complex Functions, and their Mappings and Integrals," *IEEE Antennas and Propagation Magazine*, 54, 1, February 2012, pp. 243-253.
- [11]. L. Sevgi and Ç. Uluşık, "A Lab VIEW-based Virtual Instrument for Engineering Education: A Numerical Fourier Transform Tool," Special issue of ELEKTRİK, *Turkish Journal of Electrical Engineering and Computer Sciences*, 14, 1, January 2006, pp. 129-152.
- [12]. Ç. Uluşık and L. Sevgi, "A Lab VIEW-Based Analog Modulation Tool for Virtual and Real Experimentation," *IEEE Antennas and Propagation Magazine*, 54, 6, December 2012, pp. 246-254.
- [13]. Ç. Uluşık and L. Sevgi, "Web-Based Virtual Laboratories for Antenna Arrays, Radio wave Propagation, and Filter Design," *IEEE Antennas and Propagation Magazine*, 53, 4, August 2011, pp. 252-260.
- [14]. L. Sevgi, "Innumeracy: The Meaning of the Numbers We Use," *IEEE Antennas and Propagation Magazine*, 49, 2, April 2007, pp. 195-190.
- [15]. L. Sevgi, "Numerical Fourier Transforms DFT and FFT," *IEEE Antennas and Propagation Magazine*, 49, 3, June 2007, pp. 238-243.
- [16]. L. Sevgi, "Synthetic Radar Signal Environment: Computer Generation of Signal, Noise, and Clutter," *IEEE Antennas and Propagation Magazine*, 49, 5, October 2007, pp.192-198.
- [17]. L. Sevgi, "Hypothesis Testing and Decision Making: Constant-False-Alarm Rate," *IEEE Antennas and Propagation Magazine*, 51, 3, June 2009, pp. 218-224.
- [18]. J. G. Proakis and M. Salehi, *Fundamentals of Communication Systems*, Second Edition, Upper Saddle River, NJ, Prentice-Hall Inc., 2005.
- [19]. S. Haykin, *Communication Systems*, Fourth Edition, New York, John Wiley & Sons Inc., 2001.