The Application of Readily Available Software Packages in Communication and Radar System Simulations

Pearl Yum†, Karim Akel‡, and Tai-Wu Kao, senior member, IEEE
Loyola Marymount University
Los Angeles, CA 90045-8130
Email: tkao@lmumail.lmu.edu
Telephone: (310) 338-2824
Fax (310): 338-7339

Abstract

This paper summarizes the application of software packages such as Mathcad and MATLAB in system simulations of communication and radar systems. With the help of the software packages, little time was spent on computer programming; nearly all the energy was concentrated on the analyses of the fundamental theory behind the simulations.

I. Introduction

Whether in communication systems, terrestrial or in space, or in radar systems, target detection or ground mapping, a signal processing unit is required to handle signals in the presence of additive noise. The design of any system depends, to a large extent, upon the types of signal and noise to be processed. In today’s world, cost is the single most limiting factor of a design. Consequently, competition is extremely fierce to yield an end product with less cost. To this end, accurate and good estimations of signals plus noise in both frequency and time domains, and their effects in signal detection are essential. This, in turn, leads to the computer simulations of the communications system and radar system in their early design stages.

In the early times, the computer simulations were carried out exclusively on large main frame computers. FORTRAN, C, or other types of high level programming languages were used in formulating the simulations. Lately there appear many powerful computer software packages that can be used to accomplish the same tasks on a workstation or a power PC. They have become the premier software packages for interactive numeric computation, data analysis, signal processing and graphics in engineering and scientific applications. A few simple instructions in calling and arranging the software files plus a limited programming knowledge is all that is needed to accomplish most of the simulation functions. Two of the popular software packages are MATLAB and Mathcad. Both packages are packed with a vast array of codes to perform many of the mathematical and signal processing functions.

MATLAB and Mathcad are used to simulate the signal processing unit of a ground moving target radar and to obtain the spectrum estimation of a communications system respectively. This paper summarizes the learning from the projects carried out by two students. Detection of useful information from signals contaminated with additive random noise was the main concern in both cases. The choice of software to be used in each case is not by preference, but by the availability of the software to the student at that time.

II. Simulation of Ground Moving Target Radar Signal Processing Unit

A ground moving target detection radar receives echoes from the ground clutter and targets simultaneously. The means of extricating target from ground clutter is the Doppler resolution from the echoes. Figure 1 represents a block diagram of a typical radar system signal processing unit. All the signal manipulations, such as filtering, down conversion, mixing, and demodulation before this base band signal processing unit are assumed ideal. The inputs to the signal processor are n by m matrix of complex numbers that represent n time samples for m pulses from the received echoes. The outputs of the unit consist of the amplitude and Doppler frequency information from the radar echoes.

---

† Pearl Yum is with Hughes Electronics, El Segundo, CA
‡ Karim Akel is with 46th Test Squadron, Eglin AFB, FL
The first operation performed to the input signal is pulse compression. Pulse compression is commonly used in radar systems in order to increase the average power while keeping the peak power low. With pulse compression, radar transmits pulses of relatively long duration at low power to attain the same range resolution and detection performance that can only be obtained from a high power system using short pulses. Three techniques are used to modulate the transmitted wave form for pulse compression at the receiver; they are:
- Linear Frequency Modulation
- Frequency Stepping
- Binary Phase Modulation.

For the project, binary phase modulation was used. The transmitted radio wave is phase modulated by a predetermined pulse sequence, such as the Barker codes.

The next process performed on the returned echoes is Fourier transform the compressed pulses. This will extract the frequency information from the observed radar returns. The Doppler frequencies or the changes in frequency in the returned signal indicate the size and location of the moving targets the radar intended to detect. The input data are weighted first before Fourier transform is performed. The frequency spectrum of an unweighted pulse resembles the sinx/x function. There are sidelobes to either side of the peak response. The first, and largest sidelobe is only 13.2 dB down from the peak. These sidelobes are often objectionable since a large target return might mask nearby, smaller targets. These sidelobes are due to the pulses at the beginning and end of the received pulse trains. By progressively reducing the amplitude of these pulses, the spectral sidelobes can be substantially reduced. However, the mainlobe width broadens and there will be some power loss relative to the unweighted response. MATLAB has built in functions that can provide the appropriate weighting functions corresponding to a desired sidelobe reduction in dB. Moreover, amplitude weighting and FFT are often found as one combined function in the FFT operations.

The outputs in frequency and phase from the amplitude weighting and FFT block are called range-Doppler cells. They are grouped by another n by m matrix where n represents the range and m represents the Doppler frequency of the target return. Again, a built-in function in MATLAB was used to perform the vector summation to obtain magnitude and phase information in the frequency domain operations.

Noise estimation and hit threshold logic were straightforward. Finally, the outputs of the signal processor unit were used to generate a 3-dimensional plot to indicate the magnitude and location of the ground moving targets. The authors were especially impressed by the simplicity of the plotting command. The mesh function of the MATLAB plots a 3-dimensional power magnitude of the range-Doppler cells in a relatively short time.

III. The Application of Mathcad in Spectrum Estimations

The objective of this project was to devise a strategy that will allow an increase in accuracy in measuring or estimating the power spectral density of a signal which is an ergodic wide sense stationary random process. The power spectrum of the process discloses its weighting distributions in frequencies. It is the Fourier transform of the autocorrelation function of the random process. In communications system design, prior knowledge of the power spectrum of the signal being processed is essential. It determines whether one can design a realistic system that can process the signal properly. It is especially obvious in filtering, and in design of the narrow band transmission systems. To this end, accurate and good estimation of the power spectra is extremely important in the outcome of the final designs. If the power spectrum is not known, it becomes necessary to estimate it from the observations of the noise corrupted received signal.

The methods currently used to estimate and measure the spectra are periodogram, spectra from lag windows, spectra from autocorrelation. Each of the existing methods were simulated using Mathcad. It was demonstrated that the spectra calculated using the above mentioned methods are not as accurate as one desires. It was especially noticeable for a short data record. It also demonstrated that all these methods exhibited a crippling
inaccuracy when noise level was high. The main causes for the inaccuracy are due to aliasing, and leakage as the number of data samples used in the calculations is limited.

A method to improve the measurement of the power spectrum was devised. The basic idea of the second approach is to take the measurements in power directly in the frequency domain. The power spectrum can be obtained directly from the measurements as shown in Figure 2. However, this method will raise the question again in that over what frequency bandwidth should the measurement be taken to achieve a desired accuracy. This consideration, in turn, requires a prior knowledge in the approximate frequency band in which the signal resides.

Figure 2 Power Spectrum Measurement System

Mathcad was used to simulate the block diagram. Each of the operating functions in the bandpass filter, integration, and square law device can be realized by a single code in Mathcad. In addition, the built-in mathematical features in the software were used to generate data sequence with known power spectrum, the white noise with a given variance, perform the fast Fourier transforms, and finally compute and extract the power spectrum. The plotting and graphics feature of the software were also used to produce the final visual presentation of the simulation results.

IV. Examples of Simulation Code

One of many advantages of using MATLAB and Mathcad languages is the compactness of code that resembles the actual mathematical formula. To show the compactness of the software syntax, the following sample codes are presented as examples. These codes were used frequently in simulations of either the communications system or the radar systems as described in the above sections.

1. In phase and quadrature phase signal

In radar and communications system analysis, it is advantageous to resolve a signal into two components having the same frequency and amplitude but differing in phase by 90 degrees. They are called the in phase (I) channel and the quadrature phase (Q) channel signals. This can be accomplished by writing the signal in a complex representation. The real part represents the I-channel signal and the imaginary part represents the Q-channel one. An exponential signal is written as:

$$x(t) = A \exp(j2\pi ft)$$

where $A =$ amplitude  
$f =$ frequency  
$t =$ sampling time

In MATLAB, the following one line code generates 1024 samples for each of the I- and Q- inputs. They were used to represent the inputs that represent the echo signals at the radar signal processing unit as seen in Figure 1.

$$x = \exp \left( j \times 2 \times \pi \times f \times (0:1023) / \text{PRF} \right)$$

Where PRF is the Pulse Repetition Frequency used.

The equivalent code in C or FORTRAN would be much longer and complex to write. It would be cumbersome to debug if something went wrong. In addition, the programmer must also be concerned with keeping track of the data type (e.g., float or integer, complex or real, single or double precision) of the variable, its size and memory allocation, and indexing the elements of the array. It is obvious that the application of a software package in system engineering simulations will save the engineer much time and energy.

2. Pulse Compression and Barker Code

Another example is simulating the pulse compression in a radar system. The transmitted signal is modulated by a pulse train based on the type of compression required. One of the commonly used compression is the Barker code compression. The MATLAB software can generate the 13:1 Barker code with the following command:

$$\text{PCR\_code} = [1 1 1 1 -1 -1 1 1 -1 1 -1 1 1].$$
To compress the incoming pulses at the receiver, one would have to take the convolution of the complex input signal and perform convolution with the same Barker code, as follows:

\[ X = \text{conv} (\text{flip}(PCR\_code), x). \]

3. Fast Fourier Transform and Plotting Routine

The Fast Fourier Transform of a time domain signal reveals the frequency information contained in the signal. The frequency distribution of the received signal will determine how the receiving system be designed to recover the original information in a communications system. The frequency distributions in a received radar echo determine the location and movement of the detected targets. In MATLAB and Mathcad the Fast Fourier transform on the data \( X \) is performed by the following code,

\[ Y = \text{fft}(X). \]

The final step of the simulation involved plotting the log magnitude of the processed signal in 3 dimensions to show the size and location of the targets. MATLAB can convert the \( n \times m \) complex matrix of the output from the FFT block to another \( n \times m \) matrix in magnitude only, take the log, and then graph the result all in one line, such as,

\[ \text{mesh}(20 \times \log_{10}(\text{abs}(Y))). \]

Although the examples given above sample appear to perform many complicated mathematical or functional operations with a single line of code, there were, nonetheless, many pitfalls and difficulties encountered during the simulation processes. To prevent many of the difficulties to occur, one must be careful to represent the system faithfully, to organize the data accurately, and to interpret the output correctly. Most of the all, one must have the fundamental knowledge to reject the faulty results when something went wrong in the simulations.

V. Conclusion

Many off-the-shelf software packages are available for mathematical and engineering applications. They are extremely important tools in computing and validating engineering and scientific ideas. By using the readily available software packages one can concentrates on solving engineering problems without being sidetracked in the computer programming.