EE 456 - Digital Communications
Laboratory #5
Quadrature Amplitude Modulation (QAM)
and
Modulation Error Ratio (MER)

Fall 2018

Learning Outcomes

Upon completion of this lab, students should:

1. Be capable of determining the coefficients for square root raised cosine (SRRC) pulse-shaping filters based upon given filter parameters.

2. Understand how to build a Gray-coded 16-QAM mapper.

3. Be capable of using a Vector Signal Analyzer to measure the MER of a 16-QAM signal.

Lab Equipment

1. Altera University Program DE2-115 board and Terasic ADC/DAC daughter board.

2. Computer running the Quartus software.

3. CXA N9000A Vector Signal analyzer with VSA option.

Introduction

This lab explores the implementation of a number of functional blocks required in a digital modulator. It is anticipated that this lab will serve as an introduction to the system which will be designed in EE465 in Term 2. The modulator being designed will generate a 16-Quadrature Amplitude Modulation (QAM) signal and will operate with an FPGA system clock rate of 25 MHz. The symbol rate of the modulated waveform will be 1.5625 Megasymbols per second (Msps).

The block diagram in Figure 1 shows the structure of the QAM transmitter explored in this lab. The initial block in the QAM transmitter is a Linear Feedback Shift Register (LFSR). The purpose of the LFSR is to generate pseudo-random binary data that simulates the actual information to be sent by the transmitter. The generated binary data is then passed into a mapper.

As its name implies, the function of the mapper is to map the incoming binary data (string of 0s and 1s) into signal points of a QAM constellation. Once the data has been mapped, it is then upsampled and passed through a pulse-shaping filter. The main purpose of the pulse-shaping filter is to reduce the bandwidth occupied by the QAM signal, allowing for more efficient usage of the frequency spectrum. This reduction in bandwidth increases the time duration of each QAM symbol, potentially resulting in inter-symbol interference (ISI) which impairs detection of the transmitted signal. Theoretically, the effect of ISI on the detection (i.e., demodulation) performance could be completely eliminated if the pulse-shaping filter is designed to meet the so-called Nyquist zero-ISI criterion (discussed in detail in the lectures). Unfortunately, an ideal Nyquist filter has infinite length and hence impractical. Practical pulse-shaping filters are finite length and the resulting ISI will impact the quality of the receiver.

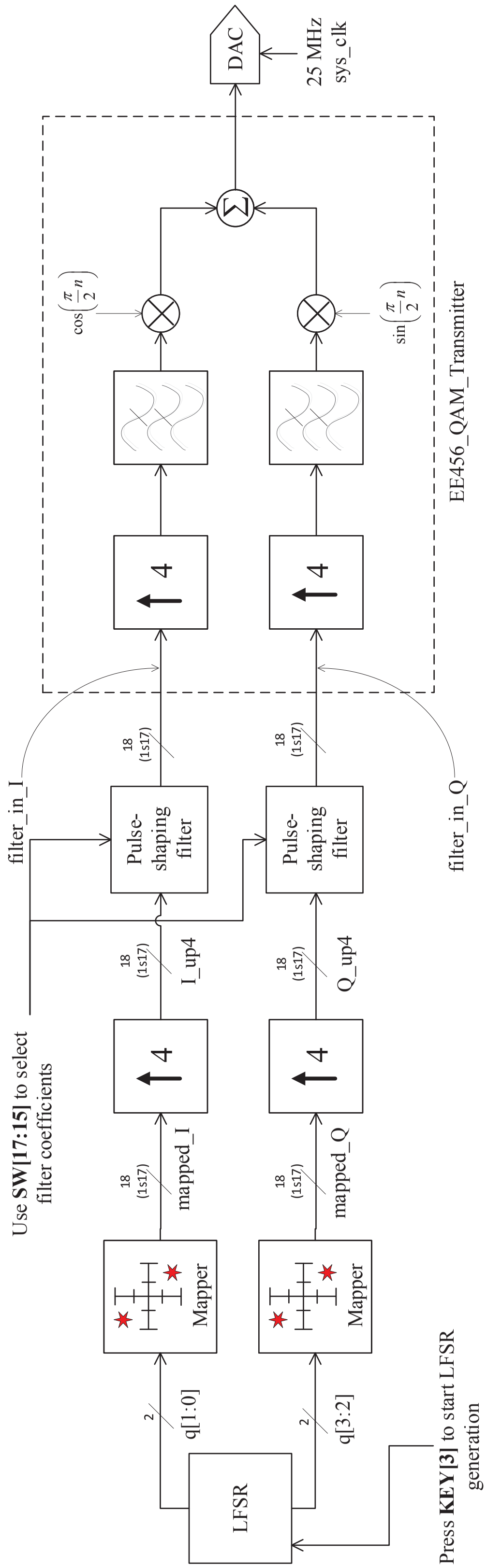

Figure 1: QAM transmitter circuit.

After passing through the pulse shaping filter, the signal is further upsampled to a frequency that is suitable for RF transmission. The signal must be filtered once again to remove the spectral images created by the upsampling process. Upon passing through the image rejection filter the in-phase and quadrature portions of the QAM signal are used to modulate a pair of quadrature carriers (cosine and sine of the carrier, respectively), summed, and then sent to a Digital-to-Analog Converter (DAC) for transmission towards a receiver.

The receiver demodulates the QAM signal and attempts to reconstruct the originally transmitted binary data sequence. During transmission, however, the signal encounters a number of impairments that reduce the fidelity of the received signal. In the system considered in this lab, the impairments are primarily related to the quality of the pulse-shaping filter and image rejection filter being used.

This lab will examine the effects of the length and coefficient scaling of the pulse-shaping filter on the quality of the received signal. To evaluate the reception quality, a measurement known as the modulation error ratio (MER) is used.

MER is a measurement that examines the ratio of the power in the expected ideal QAM constellation points to the power in the error in the received constellation points. The error in this context is defined as the difference between the expected constellation points and the actual received constellation points. Mathematically the MER of a QAM signal can be computed as:

$$\text{MER (dB)} = \lim_{N\to\infty} 10\log_{10}\left(\frac{\sum_{n=1}^{N}\left(V_I^2[n]+V_Q^2[n]\right)}{\sum_{n=1}^{N}\left[\left(V_I[n]-\widehat{V}_I[n]\right)^2+\left(V_Q[n]-\widehat{V}_Q[n]\right)^2\right]}\right),$$

where $V_I[n]$ and $V_Q[n]$ are the ideal values and $\widehat{V}_I[n]$ and $\widehat{V}_Q[n]$ are the actual received values for the in-phase (I) and quadrature (Q) components of the nth QAM symbol. Here, N is the total number of QAM symbols used for calculation (which should be a large enough number for the result to be accurate).

Pseudo-Random Data Generation

Overview

In a real modulator, the data to be transmitted would normally come from some external data source supplied by a user. The system under consideration simplifies this process by internally generating hypothetical data for transmission. Therefore, the first step in creating a design that can be used to measure the MER of a QAM transmitter is to a generate a data sequence which can act as the input data for the modulator. In order to facilitate accurate measurement of the transmitter's performance, it is desired that the power spectrum of the input data be approximately uniform. This implies that a pseudo-random sequence generated from an LFSR could be a suitable data source for the modulator, as the power spectrum of such a sequence is approximately uniform and becomes progressively more uniform as the repetition period of the sequence is increased.

In your CME341 "hotel card lock" lab you have created a pseudo-random number generator based on an LFSR structure. This circuit can be used to generate a pseudo-random sequence that should ideally produce a uniform power spectrum if the LFSR is made sufficiently long.

Review CME341 lab notes on the LFSR in the "Electronic Card Lock for Hotel Room Doors" lab to recall how the LFSR works.

In order to speed up your development time a 16-bit long LFSR module has been provided for you in the "LFSR.v" file. This LFSR has been designed so that its feedback taps are chosen in such a way that the LFSR produces a maximum length sequence. This will ensure that the power spectrum of the input data is sufficiently uniform to evaluate the performance of the transmitter.

In this lab we wish to create a modulator that will produce a 16-QAM signal. This means that there are 16 possible constellation points (waveforms) that could be transmitted during each symbol period. In order to represent these 16 possibilities, 4 binary data bits are required per input symbol ($2^4 = 16$). The LFSR provided is designed to output four bits to provide a pseudo-random symbol on each clock edge. *The LFSR must be initialized with a seed value prior to generating data, this seed value is loaded by pressing **KEY[3]**.*

QAM Mapper

Overview

Once the pseudo-random binary data has been generated the binary data must be mapped into signal points of a QAM constellation. In this lab you will be asked to map your binary data into a 16-QAM signal space. The mapper essentially operates as a look-up table, with the input being a set of data bits which chooses one of the possible constellation points and the output being specific I and Q signal levels corresponding to the chosen constellation point. Figure 2 depicts the 16-QAM constellation mapping to be used in this lab. This type of mapping is known as Gray coding and has the desirable property that any pair of adjacent constellation points differs by only a single data bit.

Careful examination of the constellation mapping shown in Figure 2 reveals that the four input data bits $b_3b_2b_1b_0$ map to I and Q values in a very convenient manner. Specifically, bits b_3 and b_2 uniquely define the I value, whereas bits b_1 and b_0 uniquely define the Q value. Furthermore, the mapping rule defining the relationship between a pair of relevant data bits and the output signal values is identical for the I and Q components. This means that the overall mapping operation can be achieved through two instantiations of a single "smaller" mapping module which accepts two input data bits and generates a single output constellation component.

Summary of Actions

1. Write a Verilog HDL file for a 16-QAM mapper using the template provided in the supplied "mapper.v" file. The input to the mapper should be a 2-bit value representing

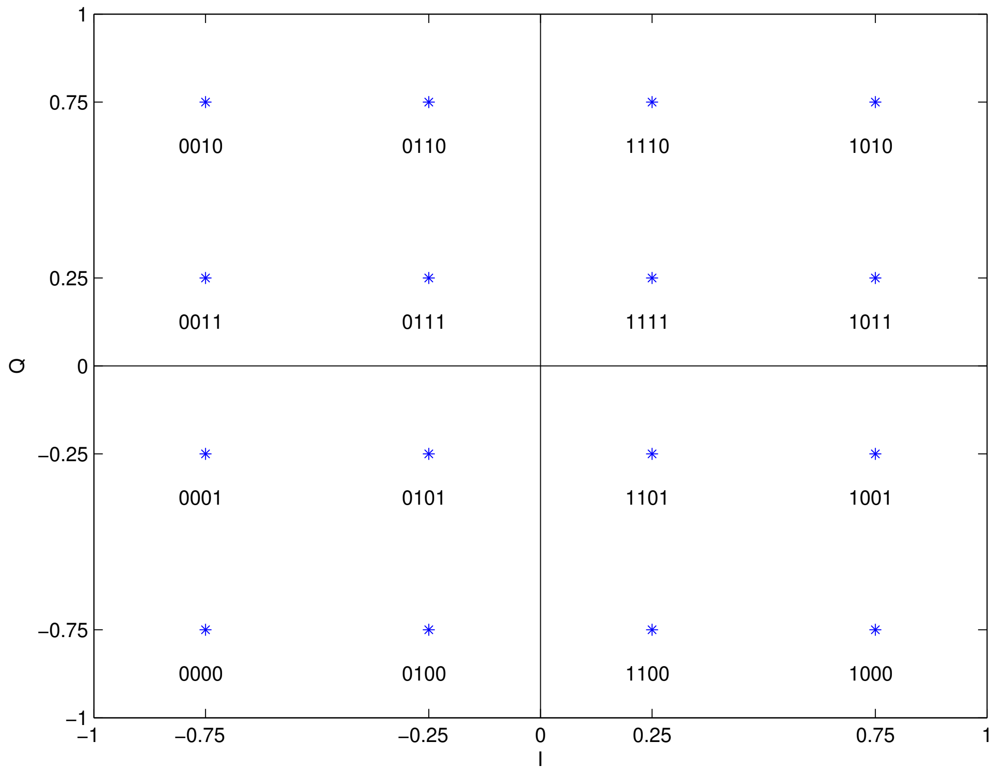

Figure 2: 16-QAM constellation with a Gray mapping.

the bits used for a single phase (i.e., I or Q). The output of the mapper should be a 1s17 output that can take on one of the following values: $\{3/4, 1/4, -1/4, 3/4\}$.

2. Ensure that your mapper has been instantiated into the top level project file and that two instances of the mapper are used, one for the I component and one for the Q component.

Pulse-Shaping Filters

Overview

The pulse-shaping filter to be used in this lab is a square root raised cosine (SRRC) filter. This lab investigates the effects of two implementation issues upon the performance of an SRRC filter, namely the filter length (number of coefficients used in the filter) and the scaling of the coefficients. As will be seen, these factors have an important effect upon the amount of ISI that is present in a communication system as well as upon the distribution of the transmitted energy across the RF spectrum.

To examine how these factors influence the MER of a QAM system you will create several SRRC filters of different lengths and scalings. A general template has been provided that allows for up to 29 coefficients to be entered. The structure of the file is the same as that used in the EE461 lab on FIR filters. The file has room for 8 separate sets of coefficients, with the active set of coefficients at any given time being chosen based on the positions of three of the switches on the DE2-115 board. The structure of the supplied pulse-shaping filter modules is the same as that used in EE461 lab 2 manual which you are expected to be familiar with at this point. Refer to EE461 lab 2 manual to review the structure of the filter if you wish.

Before designing the filter coefficients some specifications of the pulse shaping filters must be known. The specifications needed for this design are given in the table below.

Table 1: SRRC filter specifications.

Parameter	**Value**
Symbol rate	1.5625 Msym/s
Samples per symbol	4
Roll-off factor	0.25

Summary of Actions

1. Use MATLAB to generate the coefficients for four different lengths of SRRC filters. The filters should be 9, 17, 25 and 29 coefficients long.

   To generate the coefficients you may use the built-in MATLAB function `firrcos(N, Fc, r, Fs, 'rolloff', 'sqrt')`, where N, Fc, r and Fs are the order, cutoff frequency, roll-off factor and sampling rate of the filter, respectively. *Recall that the cutoff frequency of a SRRC pulse-shaping filter is equal to half of the symbol rate.*

2. For each set of coefficients computed above, normalize the coefficients so that the center coefficient is equal to 1. This can be done by dividing all of the coefficients in each set by the value of the center coefficient (i.e., the largest coefficient in each set).

3. For each set of normalized coefficients computed above, compute two sets of scaled coefficients as follows:

   - Scaling 1: Divide all coefficients by 1.25 and format the coefficients as 1s17 numbers.
   - Scaling 2: Divide all coefficients by 0.8 and format the coefficients as 1s17 numbers.

4. The provided template file is preloaded with the first of the 8 required sets of coefficients (corresponding to the length-9 filter scaled according to Scaling 1 above). Please verify that the coefficients you computed in the previous step match the provided values. Then add the coefficient values corresponding to the remaining 7 filters into the provided filter template.

Connecting the Entire System

Once your pseudo-random data generator, mapper and filters have been designed and debugged, connect the output of your filters to the **filter_in_I** and **filter_in_Q** inputs of the **EE456_QAM_Transmitter** module that has been supplied to you. Figure 5 shows what the output should look like on the spectrum analyzer when the length-9 SRRC filter using the Scaling 1 factor is selected. *Note: Your signal magnitude may be different than what shown in the figure.*

The **EE456_QAM_Transmitter** module upsamples the outputs of the SRRC filters by a factor of 4 and applies image rejection filters to these upsampled outputs. The filtered outputs are used to modulate orthogonal sinusoids at the carrier frequency and then summed. The carrier frequency used in this system is 6.25 MHz, which is one quarter of the system clock frequency.

1. Once your modules have been connected to the supplied transmitter, compile the project, program the DE2-115 board and view the output of the DAC on the signal analyzer. You should be able to see a QAM signal, which will resemble the shape of the selected SRRC filter, centered at 6.25 MHz.

Measuring MER with the Signal Analyzer

The CXA N9000A-series signal analyzer in the lab has been equipped with special software that allows the analyzer to compute the MER of a QAM constellation. This section goes through the steps required to have the analyzer measure the MER of the QAM transmitter designed in this lab.

1. Start the Vector Signal Analyzer software of the CXA signal analyzer by pressing the **Mode** button on the face plate and selecting **89601 VSA** then select **start 89601B**. *Note: the revision number of the software on the VSA is not important.*

2. Connect the keyboard and mouse to the CXA signal analyzer by pressing the button on the KVM (keyboard, video and mouse) switch located at your station (see Fig. 3). *Note: If the keyboard and mouse are directly connected to the computer at your station, you need to reconnect the keyboard and mouse to the KVM first.*

3. Once the VSA software has loaded you should see the screen shown in Figure 4. Maximize the VSA window by quickly double clicking the title bar of the window.

4. From the menus in the VSA software, use the mouse to select **MeasSetup** → **New Measurement** → **General Purpose** → **Digital Demod**.

5. Select **MeasSetup** → **Digital Demod Properties** from the menus at the top of the screen.

   Once the digital demodulation properties window has opened, set the demodulation type to be 16-QAM from the format drop down menu and set the symbol rate to be 1.5625 MHz (or mega-symbols per second), as shown in Figure 7.

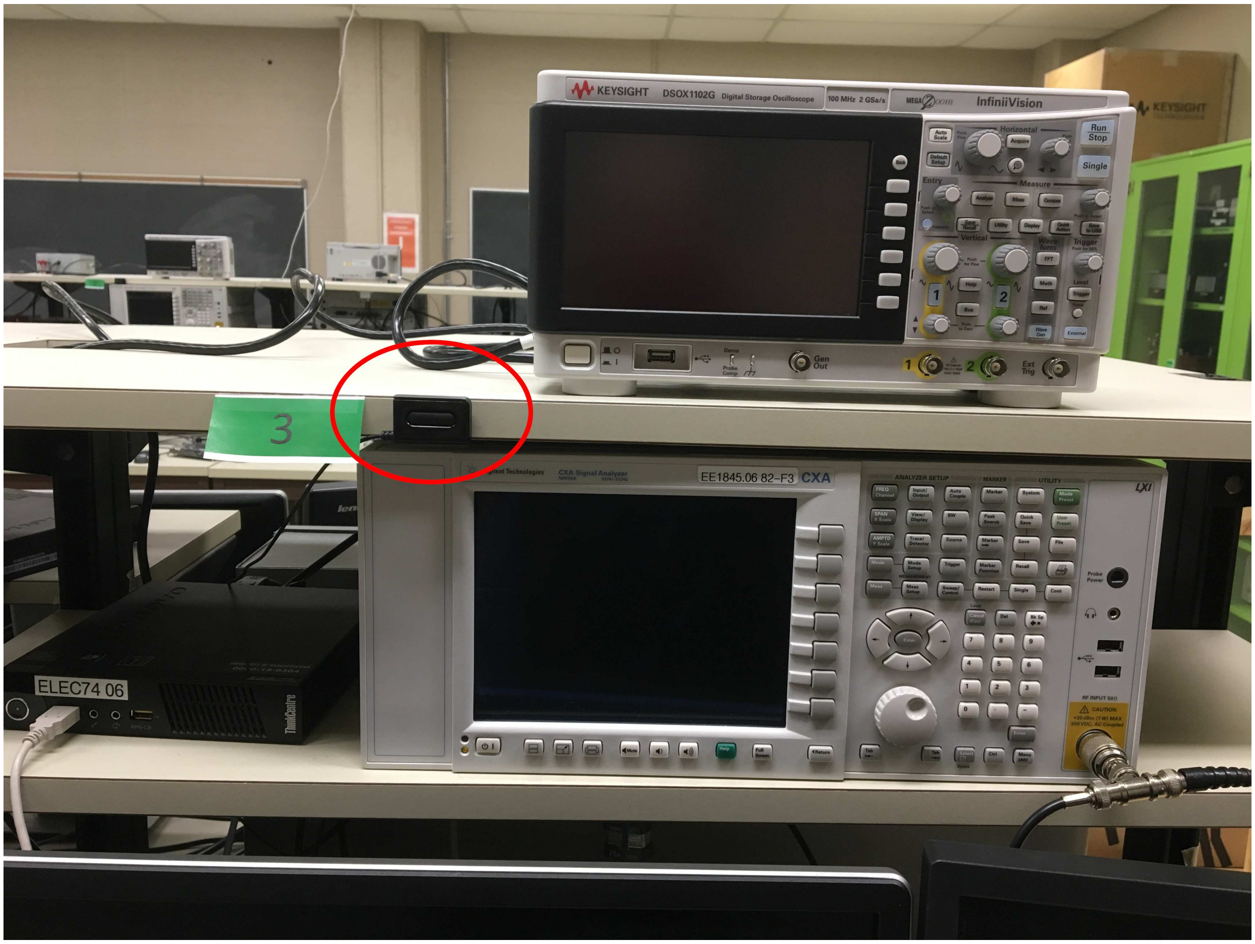

Figure 3: KVM button.

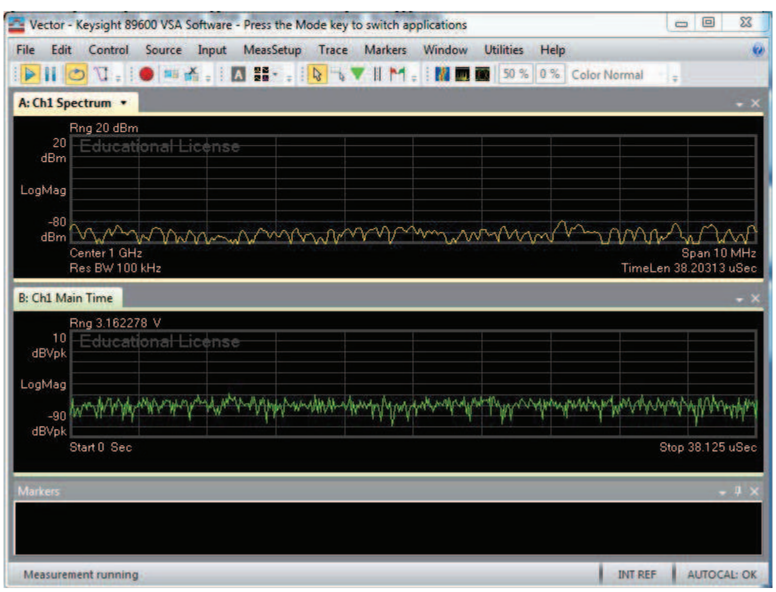

Figure 4: VSA software main screen.

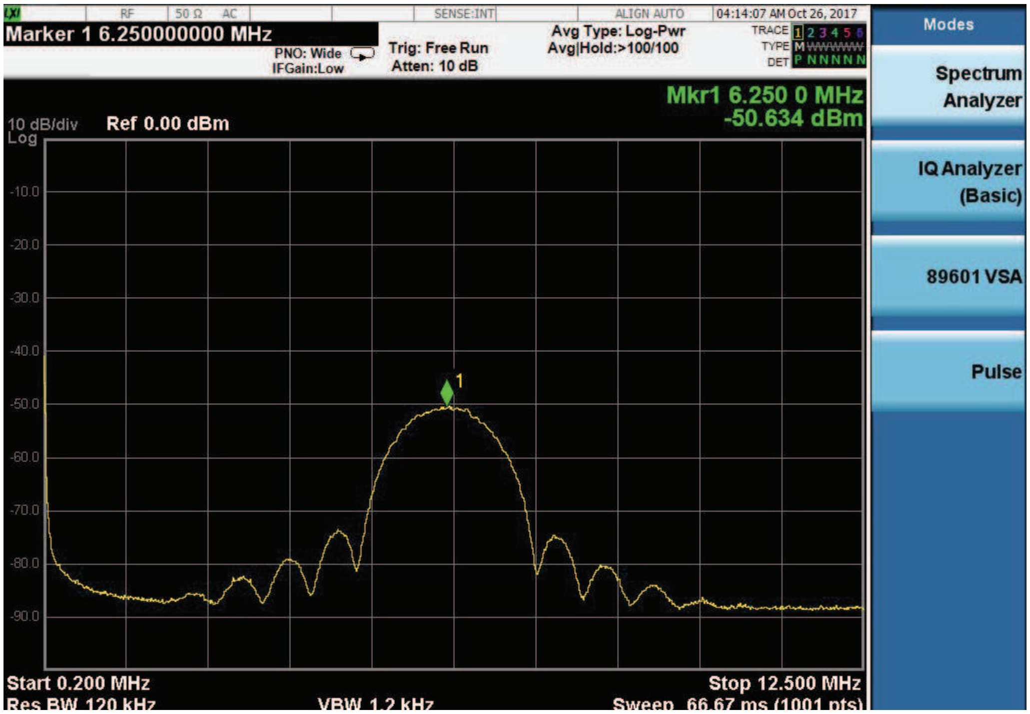

Figure 5: Select VSA mode.

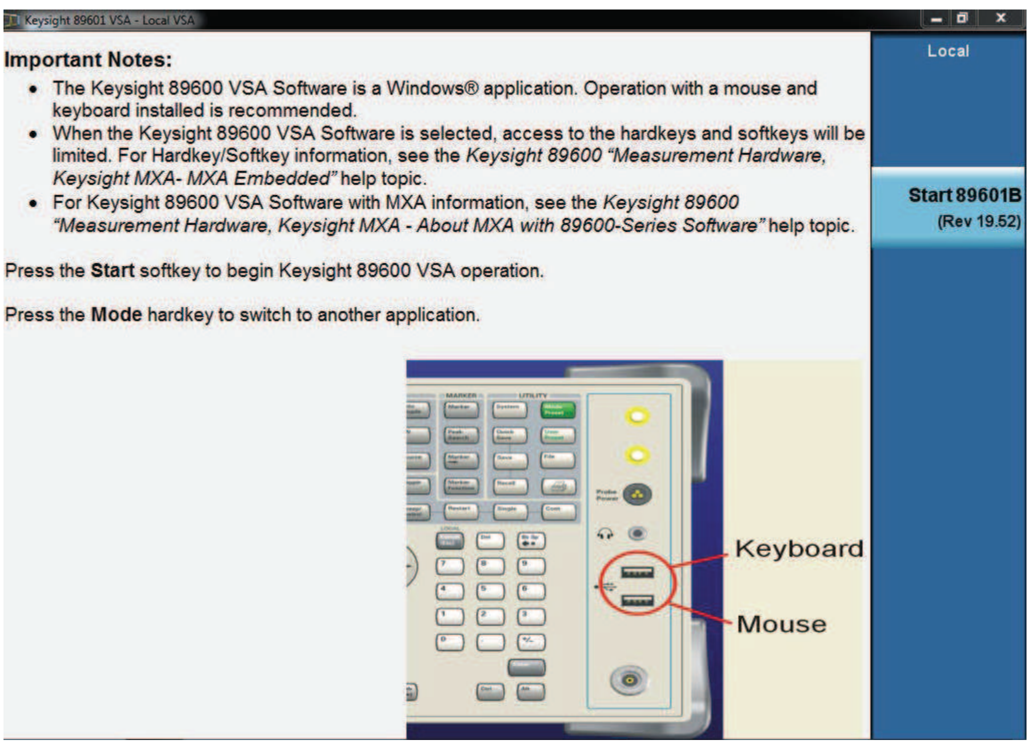

Figure 6: Start VSA software.

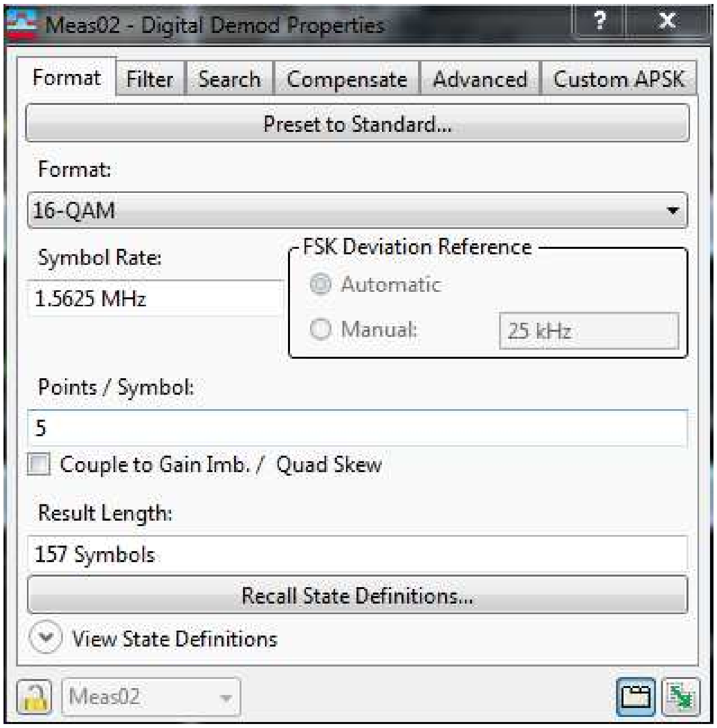

Figure 7: Digital demodulation properties window.

6. Select the **Filter** tab and set the roll-off factor (**alpha**) of the matched filter to be 0.25 as shown in Figure 8.

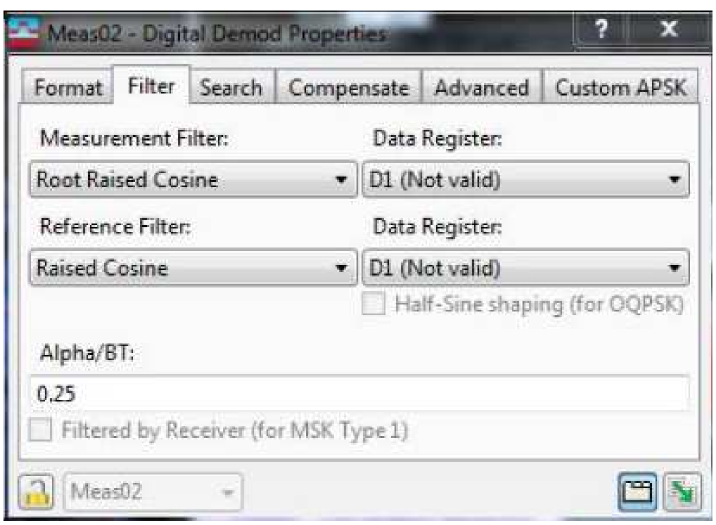

Figure 8: Set the roll-off factor of the matched filter.

7. Select the **Search** tab and ensure that the "pulse search" box is **not** checked.

8. Under the **compensate** tab ensure that the **equalization filter** is **not** checked. Close the digital demod properties window by clicking the X in the top right corner of the

window.

9. Click the run button, shaped like a triangle, in the top left corner of the VSA window.

Figure 9: Run button.

10. Close the **Meas01 spectrum** and **Meas01 main time** sub-windows if they are open. You should now have a screen that looks like the one shown in Figure 10.

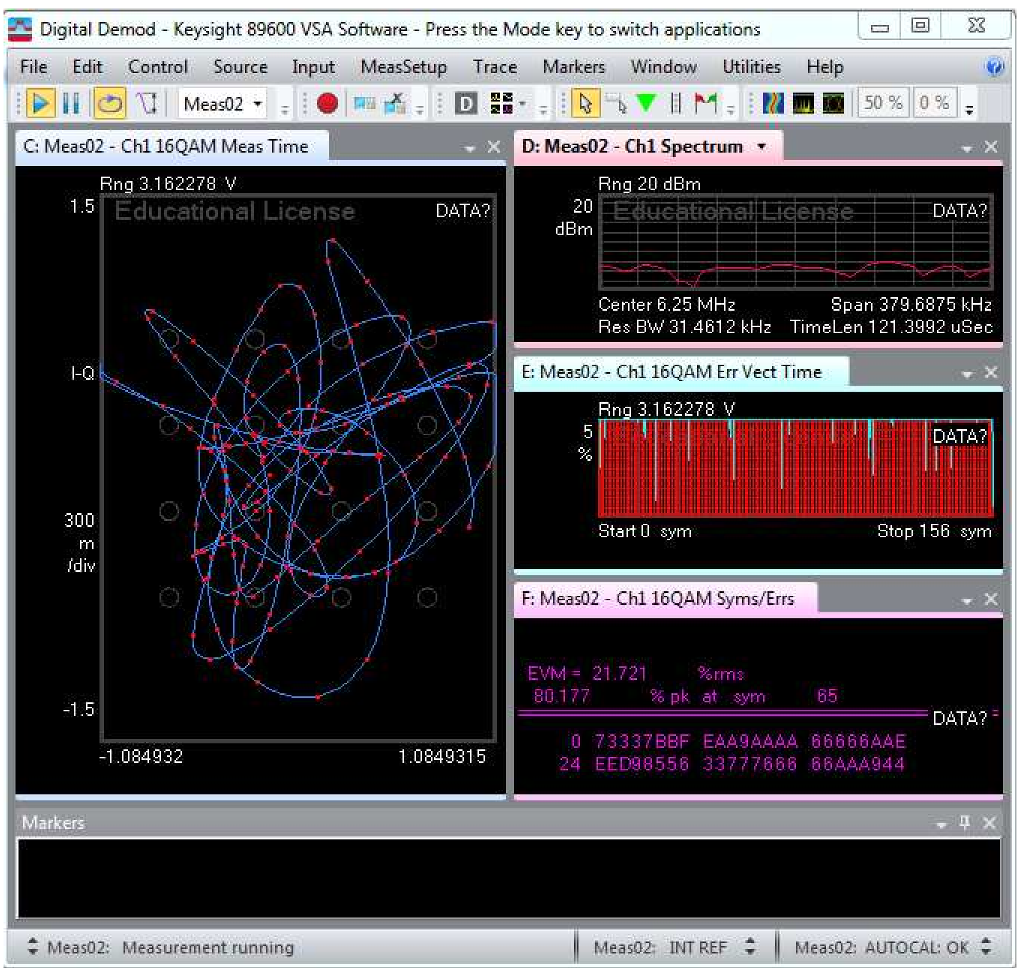

Figure 10: VSA software window.

11. In the **Meas02** spectrum window, double click the value next to the word **center** and set it to 6.25 MHz as shown in Figure 11. This value is the center frequency of the QAM signal that is being generated.

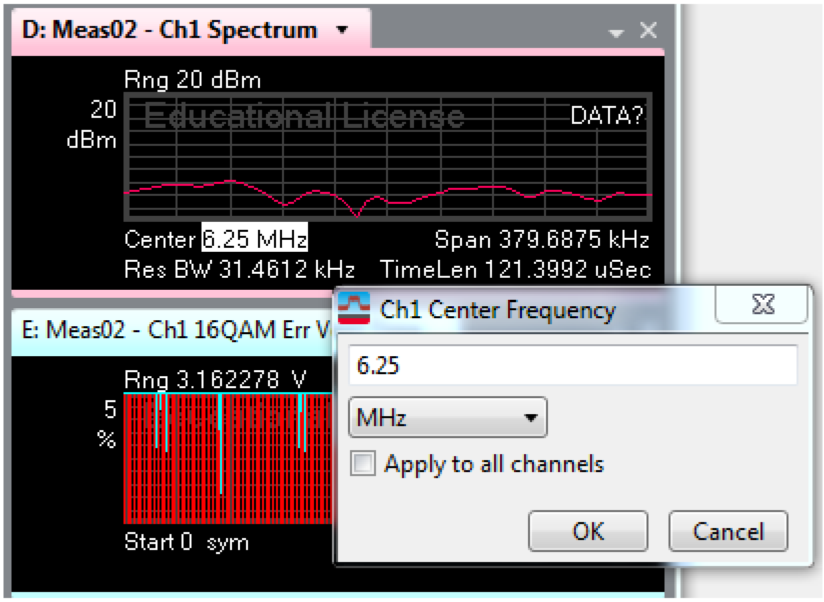

Figure 11: Window to input center frequency

12. Double click the value next to the word **span** and set it to 2 MHz. This value should be close to the symbol rate of the system multiplied by the roll-off factor of the SRRC filter plus the symbol rate (i.e., span $\geq f_{\text{sym}}(1 + \alpha_{\text{roll-off}})$).

13. Click once on the number to the left of the word **rng** in the **Meas02 spectrum** window and adjust it by using the scroll wheel on the mouse to make the spectrum magnitude as large as possible without the signal exceeding the top of the spectrum plot. At this point you should begin to see the symbols that are being transmitted on the constellation plot (**16QAM Meas Time** window) on the left side of the VSA window. These received symbols are indicated by red dots on the plot shown in Figure 12.

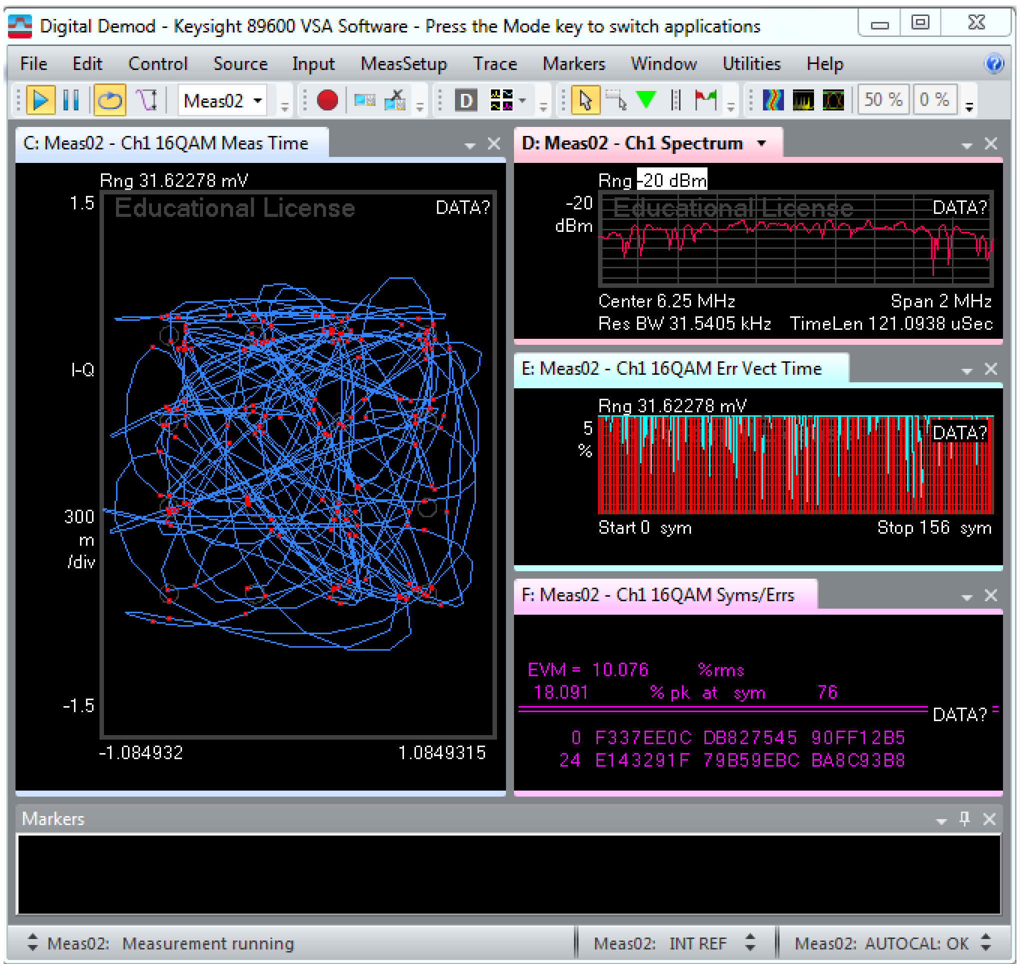

Figure 12: Adjustment of range setting

14. Right click inside the **16QAM Syms/Errs** window and select **average**. Select **RMS (video)**. Check the **repeat average** check box and set **Count** to 100, to continuously take the RMS average of 100 symbols to compute the MER. You may then close the average window once the settings have been made. *Note: You may have to click the run button in the top left corner of the VSA software window again after changing these settings.*

15. The VSA software should now be computing the SNR (MER) value for you. The MER value that is computed should appear in the middle right of the 16 QAM Syms/Errs window.

16. In the constellation plot window, adjust the colors to more easily view the ideal symbol locations. To adjust the colors follow these steps:

    - Select the **Utilities** menu, then **Display Preferences**, then the **color** tab.
    - Select **Grid** and set the color to yellow.

    The constellation plot should now look similar to the one in Figure 13.

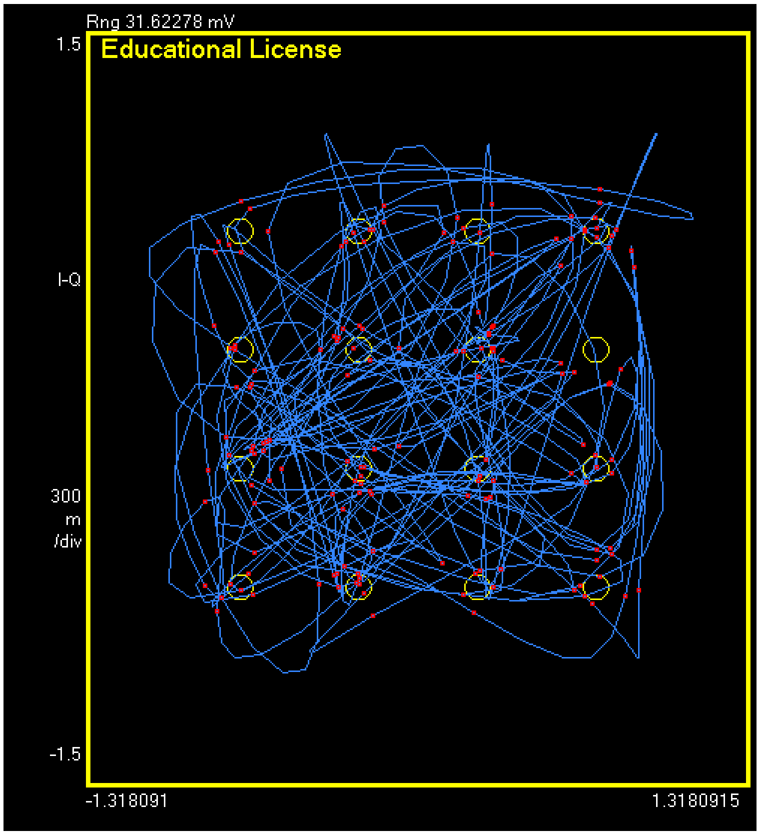

Figure 13: Constellation plot after color adjustment.

17. Using the filters with coefficients scaled according to Scaling 1, examine the constellation diagram and frequency spectrum for each filter length by adjusting the switches on the DE2-115 board. The red dots show the received symbols that the VSA detects.

Record the MER value measured by the VSA for each filter length.

18. As the filter length is increased, what is the effect upon the MER of the received signal by the VSA? As the filter length increases, what happens to the received symbols on the constellation plot? Do they more closely agree with the ideal symbol locations?

19. Switch to filter Scaling 2 and examine the constellation diagram for each filter length by adjusting the switches on the DE2-115 board. Record the MER value measured by the VSA for each filter length.

    Return to the regular spectrum analyzer mode by pressing the mode button on the CXA faceplate and selecting **Spectrum Analyzer**. Examine the frequency spectrum for each filter length using the Scaling 2 factor. You should notice that the spectrum is distorted and no longer resembles a SRRC filter frequency response.

    What is the effect of changing the filter scaling from Scaling 1 to Scaling 2 upon the overall system performance? You should observe the system performance becomes significantly worse. Why might this be happening? *Hint:* Think about binary number formats. This topic will be discussed in detail in EE465.