Bluetooth™ System Simulation with SystemView™

Objectives

- Introduction to a radio system part of short range wireless data system called Bluetooth
- To familiarize with the SystemView software simulation tool

Introduction

Bluetooth is a low cost, low power, short-range radio technology [1]. Originally designed for eliminating all cable connections around electronic appliances such as hand-free wireless headset, Bluetooth system is also expected to offer small network capability called Personal Area Network (PAN). Bluetooth specification defines both radio system, which we will study in this lab demonstration, and protocol stack. Figure 1 shows the comparison between OSI reference model and Bluetooth model.

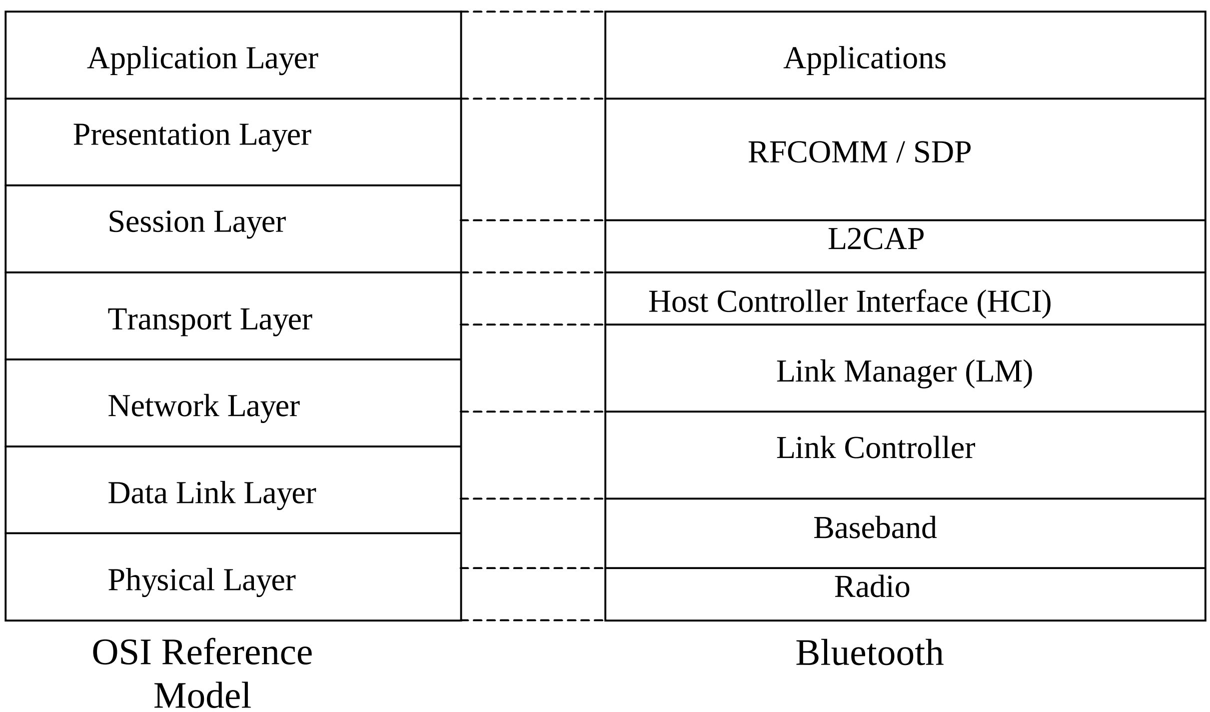

Figure 1. OSI Reference Model and Bluetooth Protocol Stack [1]

Since our experiment with SystemView will focus only on radio part of the Bluetooth system, here we will discuss briefly about the functions of Bluetooth layer up to the Link Controller part. (For further description of higher layer please consult reference [1].) Radio part called BlueRF is responsible for radio interface such as modulation, while Baseband is responsible for channel coding/decoding and low level timing control and management of the link of one data packet. Link Controller responses to the higher level commands from the Link Manager and carries out link level operation over several data packet durations such as packet-by-packet process of connection establishment.

Operating at 2.4GHz, license-free ISM (Industrial, Scientific, and Medical) spectrum, which is expected to be very crowded, Bluetooth must compete with other technologies using the same band such as Wireless LAN. Therefore, a spread spectrum technique called frequency hopping (FHSS) is used for Bluetooth devices to communicate with each other. Frequency hopping technique can be achieved by transmitting the data signal in short duration of time or timeslot in one carrier frequency and tuning to transmit at different carrier frequencies in the next timeslot as show in Figure 2. The level of hopping can be at the bit level which means for each data bit will be transmitted on different carrier. For each transmitted packet, the Bluetooth device will retune to the next radio channel in the hopping sequence; therefore, Bluetooth does hopping at the packet level. To be able to succeed in receiving the data, the receiver needs to know the exact hopping sequence of the transmitter. Thus, the synchronization between the transmitter and receiver is crucial in frequency hopping communications.

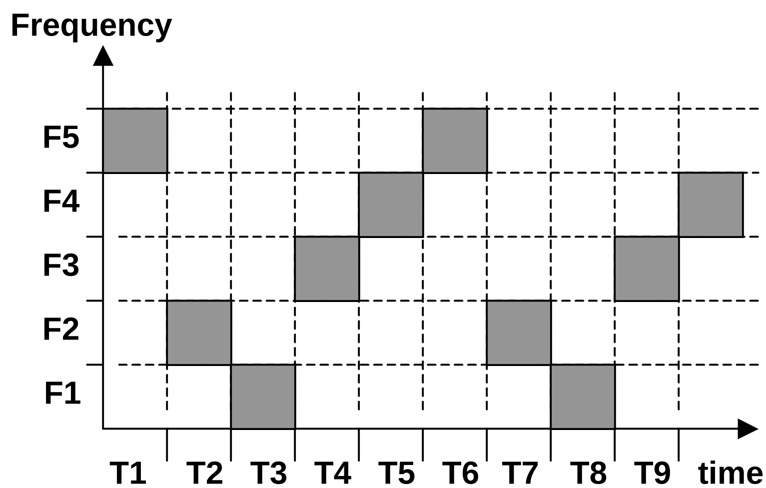

Figure 2. Frequency Hopping Spread Spectrum Technique

Bluetooth's radio channel is a hopping channel with a hop dwell time of 625μsec [3]. A Bluetooth device called the master controls the hopping channels and determines the hopping sequence for all other devices called slaves. Full duplex communications between devices is achieved by time division duplex (TDD). The master device uses a polling technique for each slave-to-master slot and decides which slave is allowed to transmit.

Bluetooth uses Gaussian Frequency Shift Keying (GFSK) modulation scheme which generates positive frequency deviation when a binary 1 is transmitted and generates negative frequency deviation when a binary 0 is transmitted. Finally, the operation ranges are approximately 10m up to 100m depending on the transmitted power from 0dBm up to 20dBm.

Radio part of Bluetooth can be implemented with alternative radio system architectures such as zero IF (Intermediate Frequency) or direct conversion, heterodyne or single bit modulation, multi-bit IQ sample modulation using lookup tables, and direct transmit modulation onto the transmit synthesizer. Figure 3 and Figure 4 illustrate the transmit path and the receive path respectively of a simple single bit modulation heterodyne system. The input of the transmitter and the output from the receiver are connected to the Baseband layers.

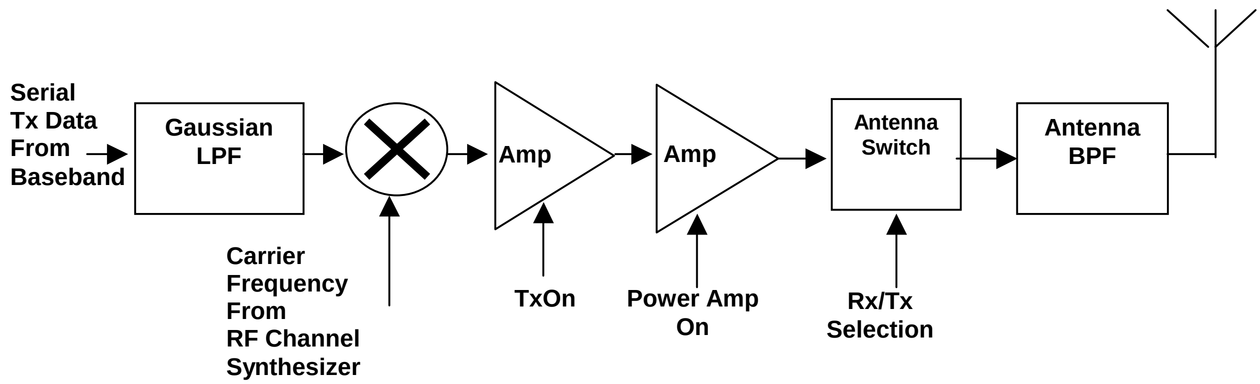

Figure 3. Transmit Path of Simple Single Bit Modulation Heterodyne Radio [1]

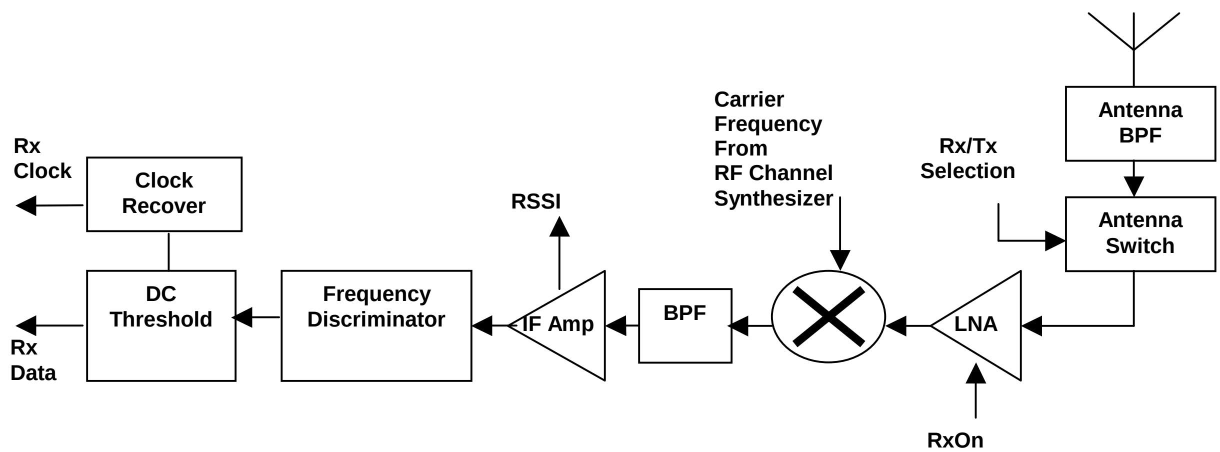

Figure 4. Receive Path of Simple Single Bit Modulation Heterodyne Radio [1]

<u>Experiment Procedures</u>

In this experiment, we demonstrate how to use the SystemView Software. It can be used to design and simulate varieties of applications ranging from analog or digital signal processing, filter design, control system and communication systems to general mathematical system modeling [2]. Although there are several sophisticated tools in this software, we will study only some basic functions of SystemView.

SystemView consists of the System window and the Analysis window. Various symbolic tokens from the SystemView's libraries are placed and connected together on the design area of the System window in the design process. Each token has its corresponding parameters which can be configured from its dialog box. To be able to see output signals or result of analysis, the user has to place sink tokens in the design model. The most important thing in performing the simulation is to set the simulation time and sample rate. The simulation time defines the start and stop time of the simulation. The sample rate defines the maximum frequency that will be allowed in the model. It is recommended that the sample rate should be at least three or four times greater than the highest frequency in the system, e.g. if the maximum frequency is 1MHz, the sample time should be 4MHz. After completing the design, user can analysis the output waveforms in Analysis window. In this window, there is a set of block-processing operations under the Sink Calculator button that can be used to do spectral analysis, filtering, signal comparison, etc.

Dynamic system probe is another important tool provided by the SystemView to perform simple real-time signal analysis in both time and frequency domain. The user can view the output waveform of any functional block in the design area. These analysis capabilities are equivalent to both standard laboratory oscilloscope and spectrum analyzer.

Note that although the frequency ranges of Bluetooth system are in the 2.4GHz band, it is much more convenient and simple to simulate the system in baseband frequency ranges. Our example does not modulate the transmitted signal to the 2.4GHz.

Step 1: Download SystemView Example File

1) Using any Pentium PC in the computer lab room 828.
2) Start a Web browser and go to the following URL:
   http://www2.sis.pitt.edu/~prashant/tel2721/lab1.html
   or
   http://www.elanix.com/support/example_files/bluetooth/
3) Click on "bluetooth.svu" to download the example file.
4) Save the example file into any temporary directory on the PC machine and take note the directory so that you can open it from the SystemView program in the next section.
5) There is an application note which describe our example at the following URL:
   http://www.elanix.com/support/app_notes/an129/an129.pdf

Step 2: Start the SystemView

1) Using the same PC in Step 1, click **Start → Programs → Programming Tools → SystemView by Elanix → SystemView by Elanix**.
2) Select all libraries options in the **Select Libraries for this Session** box by clicking all check boxes and then click **OK**.
3) The program may show a SystemView Tips dialog box, which you can read and then close it.
4) Next a window showed **Recent SystemView Files** lists all previously opened **".svu"** files. If your file is not in the list, you can click **Existing** button to open a file from the directory that you saved in the previous section.
5) In the **Open SystemView File** window, click on drive C: and go to the directory that you saved it and open the file "**bluetooth.svu**".
6) **(Optional)** A "**SystemView Professional Maintenance**" dialog window may ask you to check for upgrades. You can click "**Ask Me Again Later**" to skip checking and continue to use the program.

Step 3: System Window

When you start the program, you will be in System Window. The System Window basically consists of a menu bar, a toolbar, a design area, and a token reservoir (on the left side). The toolbar consists of the buttons that can take actions on tokens or group of tokens in the System window. We can start and stop a simulation from the toolbar and access to other utility functions. A tooltip is available for each button on the toolbar. By positioning your mouse on any button, you will receive a brief description of that button. Details about the functions of the selected button will appear in the information panel in the lower left-hand corner of the System window.

The design area is the area in the middle where the simulation design is performed. Our Bluetooth system example is shown in the design area. The system consists of several tokens connected together and forming a block diagram of system design. On the left side of the screen where the token reservoir is located, it contains generic tokens representing the token libraries available for user to drag and drop onto the design area.

Step 4: Inspect Token's Parameters

You can simply check the parameters of each token in the design area by moving your mouse over the token. Note that each token will has its identification number displayed on the top left corner. You can right-click and then select **Edit Parameters** on each token. Let's inspect a **Uniform Noise** source at token 2 which locates at the bottom of the design diagram.

Note that you can see a description of any token by right-clicking and then selecting Help. This will open a help file and locate the section that includes this token.

Step 5: Run the Simulation

Our example is ready to simulate. Click on the **Run System** icon  or use short-cut keys F5 or Shift+F5 to run the simulation. Wait to see the waveforms. The modulation input signal from Sink Token number 16 will be shown on the top right of the screen while the modulation out signal from Sink Token number 14 will be shown on the lower right of the screen.

Step 6: Configure Simulation Time

The simulation time can be configured by clicking on the stop clock icon  on the toolbar at the top of design area. Figure 5 shows the System Time window. In our example the start time is 0 second and stop time is 81.915 microsecond. The number of samples is 16384 and the sample rate is 200MHz. Bluetooth is a Time Division Duplex (TDD) system and each slot-duration is 625µsec. Therefore, our simulation time is shorter than one timeslot of Bluetooth. The TDD Control token at the top of the diagram limits the time slot for this transmitter. You can extend the simulation time to see the result of TDD control by changing the stop time to 1200µsec. When you change the stop time, the number of samples is changed automatically. Run the simulation and see that output signal is changed after the slot duration has passed. Note that the simulation will take longer time. Now change the stop time back to 81.915µsec.

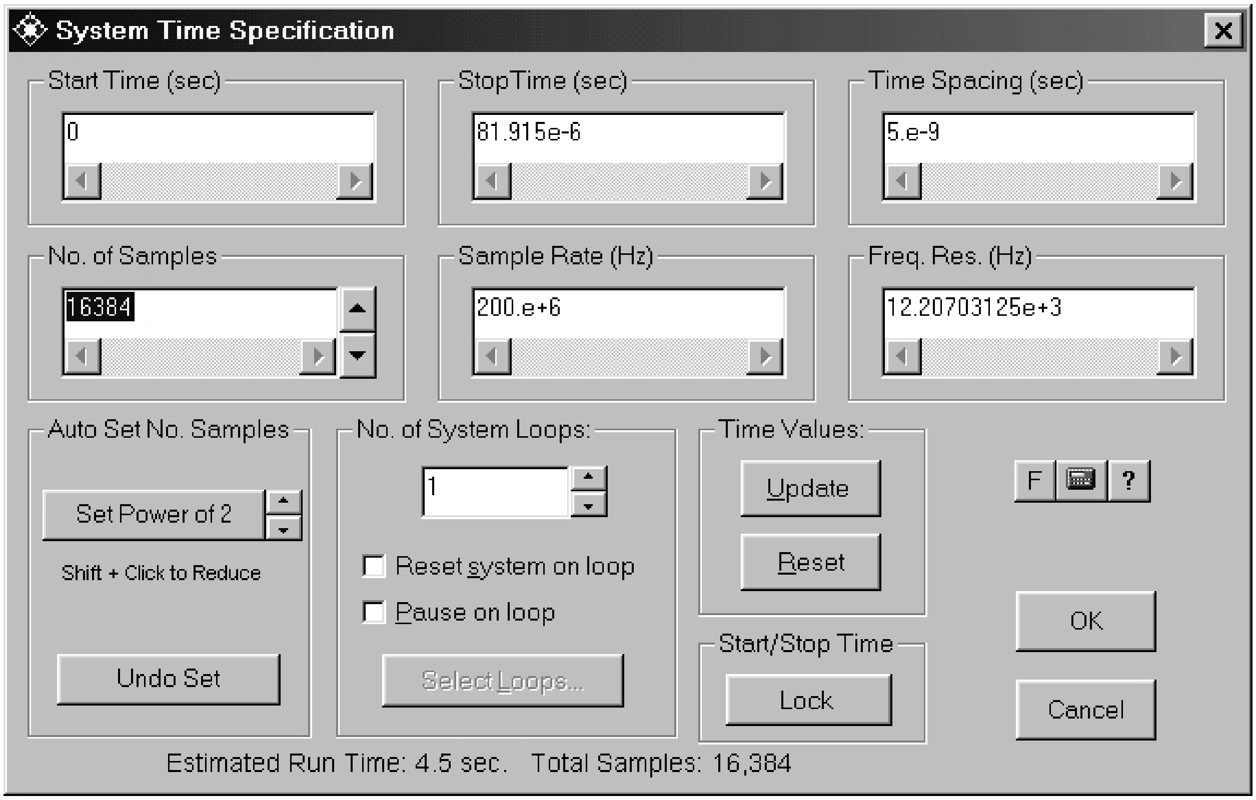

Figure 5. System Time Specification Window

Step 7: Using Dynamic System Probe

Dynamic system probe button is located at the lower left corner of your System window. Drag and drop it on any token. This will allow you to view the output signal of that token in your design when running the simulation. Drop the probe on the Source PN Sequence or token number 6. Now run the simulation again or Click F5 to run. Toggling Time button in Dynamic System probe window will result in Frequency button. This will change the function of our system probe into spectrum analyzer. Note that the maximum frequency range is 100MHz. How is it related to sample rate in the previous section? Play with the system probe by using channel A and channel B probes which should give you two signals at the same time. On the right side of probe window there is a button for probe B Input, which you can drag and drop on any other token. To be able to see both signal from channel A and B, toggle AB display modes button on probe toolbar until you see A and B as a separate plots (an icon with two parallel lines under AB). Close the probe after you finish.

Step 8: Using Analysis Window

To change to Analysis window, click on **Analysis Window** button located on the toolbar. Click '**Yes**' when the program asks if you want it to perform automated Sink Calculator sequence. The windows will be arranged so that you can see the analysis view of modulation input signal, modulation out signal, and the overlay of the two signals. **Please note that after you change your design, your analysis window will not update automatically. You will have**

to click on flashing New Data button on the toolbar to allow program do analysis on the new simulated data.

Click on any window, and drag a rectangular block on some portion of signal to zoom into your signal. This will be helpful to take a closer look at your signal. To make the signal wave form return to the original scale use the short-cut key "**Ctrl+R**".

In this window, you can analyze the output signal using the **Sink Calculator** button at the bottom of the screen. We will use square operator on the input signal. Click on **Modulation input** window and then click on the **Sink Calculator** button. Select **Algebraic** button, and then select **Square** button. Click **OK** to perform the operation. Compare the result of the Modulation input and the new signal in a new window.

Next we will modify and run simulation on our simple Bluetooth system. Click on the **System Window** button or the last button on the toolbar to switch back to the design area of System Window.

<u>Step 9: Modify the Design System</u>

One of the most popular radio channel model is an additive white Gaussian noise. In this section, we will simulate this ideal noise and add it to our Bluetooth transmitted signal. Please follow the following procedure.

1) Disconnect token 10 (multiplier) and token 13 (multiplier): From top toolbar, click on **Disconnect Tokens** and then click on token 10 and then click on token 13.
2) Put an adder: Go to the token reservoir on your left side, drag the **Adder Token** and drop it in between the token 10 and token 13 on your design area. Now Adder token is marked as token 24.
3) Connect token 10 to the Adder (token 24) and Connect the Adder to token 13: Again from the top toolbar, click on **Connect Tokens** and then click on token 10 first and click on the **Adder Token**. Repeat the procedure for the Adder token and token 13.
4) Put a Gaussian noise generator: From the token reservoir on left side of the screen, drag the **Generic Source** and drop it on an empty space above the **Adder Token**. Right click on the **Generic Source**, now token 25, and select **Libraries** menu. A **SystemView Source Library** window will pop up. Click on the second button named **Noise / PN** and then click on the **Gaussian Noise** Token. Next we will configure the parameters for our noise source by clicking on the "**Parameters…**" button. Another popup window named **Gaussian Noise Source (Token 25)** will come up. Change the number under **Std Deviation (v)** box to **0.1 volt,** and make sure that the mean is **0 volt,** then click **OK**. Click **OK** again at the **SystemView Source Library** window. Now you will have a Gaussian noise generator on your design. Finally connect the Gaussian Noise Source to the Adder Token using the same procedure in (3).
5) Save the design: Click on **File** menu and select **Save System As** or use short-cut keys "**F12**" to save your design. Save your design in a different file name. We will need the original file in your assignment.
6) Run the simulation by clicking on the green triangle button, **Run System**, to see the result. Compare the input signal and output signal of this experiment.
7) Using System Probe and observe the spectrum of noise signal generate from the Gaussian noise token: Drag the **Dynamic System Probe** icon and drop it onto the **Gaussian Noise** Token, then run the simulation again by clicking on **Run** button or **F5** inside the

Dynamic System Probe window to see the signal generated by the noise source. Switch between Time and Frequency domain. What is the ideal characteristic of Gaussian noise source?

8) Increase noise level: Click on the **Gaussion Noise Source** Token and select **Edit Parameters**. Change the **Std Deviation (v)** value to **0.5 volt**. Observe the change at output signal.

<u>Assignments</u>

1) Modify the de-hopping sequence

   <u>Directions:</u> Using **Duplicate Tokens** command (either from the top toolbar or from right click menu), duplicate token 2, 3, 1, and 8 and connect new tokens in the same manner. Disconnect token 8 and 17. Now connect the new hopping sequence (from your new **Gain** token) to token 17. Observe the output result.

2) Attach your print out for new simulation design in (1) including the Gaussian noise with **Std. Deviation (v)** setting to **0.1 volt** in Step 9.

   <u>Directions:</u> Click on **File** menu and then select **PrintSystem: Symbolic Tokens**.

3) Attach printouts of the impulse response and frequency response of Gaussian low pass filter in your report.

   <u>Directions:</u> Right click on **Gaussian low pass filter** or token 7 and then select **Edit Parameters**. Make sure that **Time** button is selected. Right click on the diagram of **Impulse Response** and select copy and paste it in your MS-Word lab write-up. Do the same thing for **Frequency Response** by clicking on **Gain** button on the right side of diagram.

4) Using the original system file (no new de-hopping module), instead of additive noise as demonstrated in Step 9, put a FIR band-stop filter (described next) into your design at the same position as the Adder token. Run the simulation several time until there is a **completely different** between your input signal and output signal. Attach a printout of this different output in your report and also the Frequency Response of your FIR band-stop filter.

   <u>Directions:</u> FIR band reject/band stop filter design can be done by using the **Generic Operator** Token from the left side token reservoir. Use same procedure as described in Step 9 to put the token in your design and connect it to the rest of the design. Here we will explain how to select and create the FIR band stop filter. Right click on the token to select the **Library** menu. Make sure the **Filters/Systems** button is selected and then select **Linear Sys Filters** Token. Click on **Parameter** button.

   A new window called **SystemView Linear System** will pop up. Now select the **FIR Filters** under the **Design** box. Another window called **FIR Filter Library** is now opened. Select **Bandstop** button and then click **Design** button. Fill in the data of relative frequencies as shown in Figure 6. Next click **Update Est** button (No. FIR Taps will be updated.) and click **Finish** button. Click **Yes** to accept the design. Finally end your design and exit to the design area by click **OK** in the SystemView Linear System (Band Reject FIR) window.

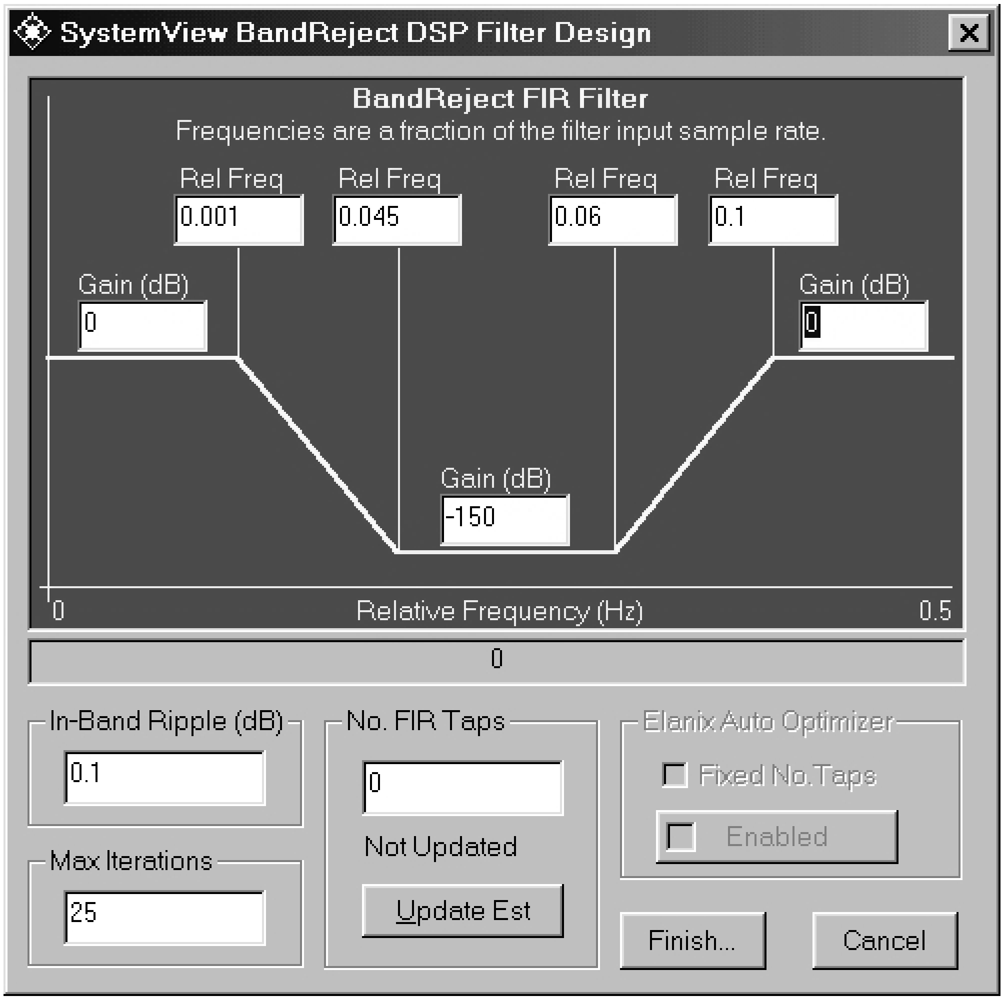

Figure 6. Band Reject FIR Filter Design.

Questions

1) Explain the results of creating new de-hopping sequence at the receiver. What happens after we changed the frequency de-hop sequence? What would you expect to see at the Modulation output after adding new de-hop sequence? What is the reason that our new output is different from the original de-hopping?
2) Explain how the band reject or band stop filter affects your output signal. Why sometimes the outputs are not affected by our band stop filter? (Hint: It has to do with frequency hopping.)
3) What are the cutoff frequencies corresponding to each relative frequency in the FIR band reject design? (Hint: There is a description in the Filter Design window.)
4) What happens to your output signal when you increase the standard deviation of Gaussian noise from 0.1 volt to 1 volt?
5) What is the channel bandwidth or frequency separation of a Bluetooth's channel? Compare with the input data rate of the token in the example Bluetooth system. Is it different?

6) Find out from any the textbook or reference paper about Bluetooth, how many radio channels are defined by the Bluetooth specification? Is it the same all around the world? If it is not, how is it different?
7) From the example system, what is the frequency hopping rate of the Bluetooth system?
8) What is the frequency cut off of the Gaussian low pass filter next to the input signal?
9) What is the lower and upper frequency cut off of the Butterworth band pass filter?
10) From the discussion in our experiment procedure, why is the sample rate of the simulation recommended to be at least three or four times greater than the highest frequency in the system?
11) Based on this Bluetooth system simulation, what is the purpose of that Uniform Noise source token 2 inspected in Step 4? What is the Uniform Noise signal look like?
12) Give a brief explanation for the different between the Uniform Noise and Gaussian Noise. (Hint: Use the dynamic system probe to compare the signals.)

Reference

[1] Jennifer Bray, and Charles F. Sturman, "Bluetooth: Connect Without Cables," Prentice Hall PTR, New Jersey, 2001.
[2] --, "SystemView User's Guide", Elanix Incoporated, 1999.
[3] Jaap C. Haarthsen, "The Bluetooth Radio System," IEEE Personal Communications, February 2000, pp.28 – 36.

Trouble Shooting with SystemView

- Sometime you may encounter a error indicating that your design file is a read-only file. To avoid this error, you should save your design in another file name.
- If you encounter any fatal error, please try to close the program and restart it. If the problem still persists, reboot your PC machine and ask for PC lab GSA to login for you.