

## Exercises:

The following exercises illustrate the execution of collaborative simulations in J-DSP. The exercises namely a Pole-zero cancellation simulation and a Peak-picking analysis and synthesis simulation deal with important signal processing concepts. One of the users is labeled *user-1* and the collaborator *user-2*.

### Pole-zero cancellation simulation

#### Introduction:

This exercise involves the study of the relation between pole-zero locations and the resulting frequency response. It also involves a simple pole-zero cancellation example. *User-1* establishes a filter system with single pole and *user-2* cancels the effect of this pole on the frequency response by placing a zero at the exact same location.

A Linear Time Invariant (LTI) system can be specified by its transfer function  $H(z)$  given by:

$$H(z) = \frac{B(z)}{A(z)} = \frac{\sum_{i=0}^L b^i z^{-i}}{1 + \sum_{k=1}^M a^k z^{-k}}$$

where the poles of the system are given by the roots of the denominator polynomial  $A(z)$  and the zeros are given by the roots of the numerator polynomial  $B(z)$ . Also, the  $b_i$ 's and the  $a_k$ 's are called the filter coefficients of the system, with  $a_0$  always being equal to one. The frequency response of this system is obtained by substituting  $z=e^{j\Omega}$  in the transfer function  $H(z)$ .  $H(z)$  can be obtained by taking the z-transform of the impulse response  $h(n)$  of the system. The two J-DSP blocks used in this exercise can be found in **Filter Blocks** menu of the J-DSP editor and are described below:

**PZ Placement:** The pole-zero placement block, **PZ Placement**, can be used to create a pole-zero plot. The **PZ Placement** block can be connected to either the bottom of a **Filter** block or to a **Freq-Resp** block. Connecting the **PZ Placement** block to the bottom of a **Filter** block will automatically set the filter coefficients of that **Filter** block so that its poles and zeros are at the locations specified in the **PZ Placement** block. Connecting the **PZ Placement** block to a **Freq-Resp** block will display, in the **Freq-Resp** window, the frequency response of the

filter having poles and zeros specified in the **PZ Placement** block. Poles and zeros can be placed on the plot by using either the mouse or the keyboard. To place them using the mouse, select “graphical” from the pop-down menu, press the [Add Pole] or [Add Zero] button and then click on the plot in the desired location of the zero or pole. To place poles and zeros using the keyboard, select one of the manual options from the pop-down menu, choose either the “pole” or “zero” radio button, enter the location using the keyboard and press the [Enter] button on the right edge of the window. Please note that because of the graphical interface the positions of poles and zeros are on a grid and hence the entered values may change according to the grid values. If you want to avoid changes in manually entered roots, then avoid switching from manual to graphical and vice versa. To delete pole(s) or zero(s), select the pole(s) or zero(s) to be deleted on the plot by clicking on it and then press the [delete] button at the bottom of the window. To move a pole or zero, click on the pole(s) or zero(s) in the plot and while holding the mouse button down drag it to a new location. The [Reset] button will erase all the poles and zeros from the plot.

**Freq-Resp:** The frequency response block **Freq-Resp** can be connected to the top of a **Filter** block, to a **PZ Placement** block as explained above, or to a **Filter design** block such as **IIR Filter**, **FIR Filter** or **Kaiser**. The **Freq-Resp** block will show the frequency response of the filter to which it is connected or if connected to the **PZ Placement** block, the frequency response of a filter having poles and zeros specified in the **PZ Placement** block.

#### **Exercise:**

**Step 1 (User-1):** Establish a simulation diagram as shown in Figure 1.1 and open the dialog windows of both the **PZ placement** block and the **Freq-Resp** block. Place a pole at (0.33, 0.78). Observe the frequency response magnitude plot in the dialog window. How is the pole location related to the characteristics of the magnitude response plot?

Inform *user-2* through the chat dialog to click on the [LoadScript] button.

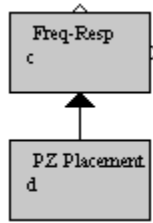


Figure 1.1: J-DSP simulation diagram to study effects of pole and zero locations on frequency response.

**Step 2 (User-2):** Click on the [LoadScript] button in the J-DSP editor frame. Observe the *PZ placement* dialog. Observe the frequency response plot. How is the pole location related to the characteristics of the magnitude response plot? Now place a zero on the existing pole i.e., at (0.33, 0.78). Observe the frequency response plot in the dialog window. What happened to the frequency response and why?  
Inform *user-1* through the chat dialog to click on the [LoadScript] button.

**Step 3 (User-1):** Click on the [LoadScript] button in the J-DSP editor frame window. Observe the frequency response plot in the dialog window. Observe the frequency response plot in the dialog window and also the pole-zero placement dialog. What happened to the frequency response? What changes in the *PZ placement* dialog resulted in this frequency response?

**Step 4 (Both users):** Now remove the pole and its conjugate from the *PZ placement* dialog. The *PZ placement* dialog now consists of only zero(s) at (0.33, 0.78) and (0.33, -0.78). Observe the resulting frequency response plot. How is the zero location related to the characteristics of the magnitude response plot? Does this explain how the placement of the zero in step-2 or step-3 resulted in the corresponding frequency response?

**Step 5 (User-1):** Reset the pole-zero locations in the *PZ placement* dialog by clicking on the [Reset] button. Now place a pole on the vertical axis at (0.0, 0.5). Observe the frequency at which the peak is located in the magnitude frequency response plot. What does this say? Now move the pole up and down on the vertical axis of the *PZ placement* dialog, with the help of a mouse and observe how the peak varies in the *Freq- Resp* dialog.

What happens to the peak in the magnitude response when a pole is moved closer to and away from the unit circle?

Inform *user-2* through the chat dialog to click on the [LoadScript] button.

**Step 6 (User-2):** Click on the [LoadScript] button in the J-DSP editor frame window. Observe the frequency at which the peak is located in the magnitude frequency response plot. What does this say? Now move the pole up and down on the vertical axis of the *PZ placement* dialog, with the help of a mouse and observe how the peak varies in the *Freq- Resp* dialog. What happens to the peak in the magnitude response when a pole is moved closer to and away from the unit circle?

**Step 7 (Both users):** Now move the pole to (0.0, 1.0) by manually typing the pole location in to the *PZ placement* dialog. Observe the frequency response plot (observe the axis value). Is the present system stable? What does this say? How do you make it stable? Discuss your ideas with your collaborator through the chat dialog.

**Step 8 (Both users):** Now, add a zero at (0.0, 1.0) by manually typing the zero location in to the dialog. What happened to the frequency response? Can we use the pole-zero cancellation to transform an unstable system in to a stable one? Discuss your ideas with your collaborator through the chat dialog.

## **Peak picking analysis and synthesis of a speech signal**

### **Introduction:**

This exercise involves the Peak picking analysis and synthesis of a speech signal. The synthesized speech signal obtained through the peak picking method is compared with that of another method which involves selection of the first few components to achieve compression. Both subjective and objective comparisons are done. User-1 establishes the analysis part of the simulation and user-2 appends the synthesis part.

Peak picking is a data compression scheme where the signal to be compressed is transformed in to frequency domain by taking an N-point DFT of it and then selecting only  $n$  highest components out of the

available  $N$  DFT components for reconstructing the signal back (note that  $n \leq N$ ).

**Pk Pking:** The Peak Picking block, **Pk Pking** available in JDSP implements this  $n$  peaks selection scheme. It also has another option, through which the first  $n$  DFT components instead of peaks can be chosen. This block can be located in the **Frequency Blocks** group of the J-DSP editor frame. The number of components selected, up to  $n=N/2$  (i.e., in frequency domain up to  $\Omega=\pi$ ), can be programmed, and conjugate symmetry is preserved around  $\pi$  to reconstruct a real-valued signal in the synthesis stage. This block also has an inbuilt graph panel where the user can view the various signals involved.

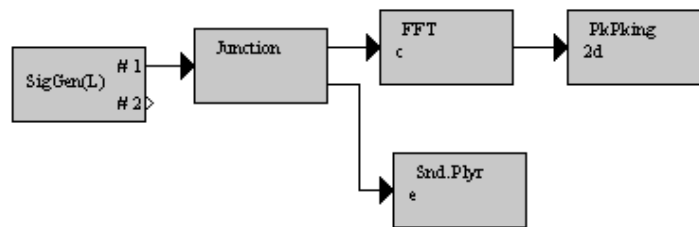


Figure 2.1: Peak Picking Analysis Stage.

### Exercise:

**Step 1 (User-1):** Establish a simulation diagram as shown in Figure 2.1 and open the dialog windows of the Long signal generator, **SigGen(L)** and the Sound Player block, **Snd.Plyr**. Run the entire male speaker signal in the **SigGen(L)** dialog window by pressing the “>>|” button. Click on the [Play] button of the **Snd.Plyr** block to listen to the original male speaker signal generated by **SigGen(L)**.

Inform User-2 through the chat dialog to click on the [LoadScript] button.

**Step 2 (User-2):** Click on the [LoadScript] button in the J-DSP editor frame window. Run the entire male speaker signal in the **SigGen(L)** dialog window by pressing the “>>|” button. Click on the [Play] button of the **Snd.Plyr** block to listen to the original male speaker signal generated by **SigGen(L)**. Append the Inverse FFT block, **IFFT** and another **Snd.Plyr** block as shown in Figure 2.2, to complete the synthesis part. Open the dialog window of the **Pk Pking** block.

Inform User-1 through the chat dialog to click on the [LoadScript] button.

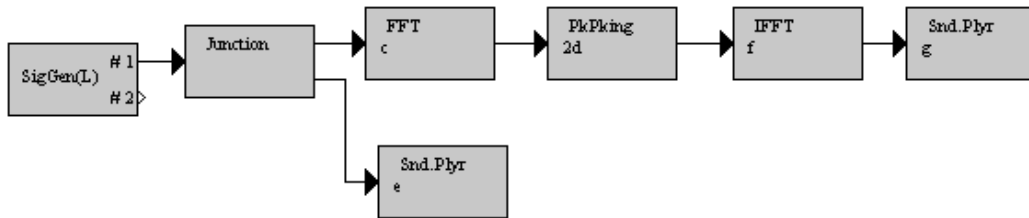


Figure 2.2: Peak Picking Analysis & Synthesis simulation diagram in J-DSP.

**Step 3 (User-1):** Click on the [LoadScript] button in the J-DSP editor frame window. This brings up the entire analysis-synthesis simulation diagram.

**Step 4 (Both users):** In the *Pk Pking* dialog window, change the “Components Selected” option to 8 and press the [update] button. Rerun the entire male speaker signal in the *SigGen(L)* by clicking on the [Rerun] button and listen to the *Snd.Plyr* block connected to the *IFFT* block. This is the speech signal reconstructed by using only the highest eight components or peaks of the frequency domain speech signal. Similarly, listen to the reconstructed speech by changing the option “Components Selected” to 32, 64 and rerunning the entire male speaker signal in the *SigGen(L)*. How does the reconstructed speech quality change as the no. of highest components or peaks selected is increased?

Also, change the option “Highest” to “First” in the *Pk Pking* Dialog window. This would result in the selection of the first few components of the frequency domain speech signal rather than the peaks. Listen to the reconstructed speech for each case by selecting 8, 32, 64 components. Remember to rerun the entire male speaker signal for each case. How does the reconstructed speech quality change as the no. of first components selected is increased? Did the selection of the first 8 components suffice the information required for the male speaker speech signal reconstruction? Which ones among the highest peaks and the first components are important for obtaining better reconstructed speech quality? Discuss your ideas with your collaborator through the chat dialog.

**Step 5 (Both users):** Place an *SNR* block available in **Basic Blocks** on the J-DSP editor frame. Remove the connections to sound player blocks and connect the output of the *Junction* block i.e., the original signal to the upper pin of the *SNR* block and connect the output of the *IFFT* block or reconstructed speech signal to the bottom pin of the *SNR* block. Rerun the entire male speaker signal in *SigGen(L)* for all the 6 cases and observe the “Overall SNR” in each case. Tabulate the “Overall SNR” values obtained for all the 6 cases in Table 2.1 and compare them. How does *SNR* vary with the number of components selected? Also how does it vary when the highest components or the peaks are selected and when the first components are selected? Discuss your ideas with your collaborator through the chat dialog.

Table 2.1: SNR values

N	Overall SNR when N Peaks or N higher energy FFT components are chosen for reconstruction	Overall SNR when first N FFT components are chosen for reconstruction
8		
32		
64		