Labview Implementation*

James Finnigan
Jeremy Bass
Claiborne McPheeters
Edward Rodriguez

This work is produced by The Connexions Project and licensed under the
Creative Commons Attribution License †

Abstract

In this section the details of the simulated microphone array are discussed

1 Labview VIs Used in Simulation

Top Level Diagram and Overall Organization
The figure below is a top level diagram of a six microphone array simulation VI. This VI is named "Multiple Frequency Simulation" Starting at the left with the signal generation, the code flows to the right. This top VI is made out of smaller VIs that each perform a specific function. Detailed descriptions of each of these can be found in the following paragraphs.

*Version 1.10: Apr 30, 2005 5:23 pm -0500
†http://creativecommons.org/licenses/by/1.0

Top Level Diagram (Multiple Frequency Simulation)

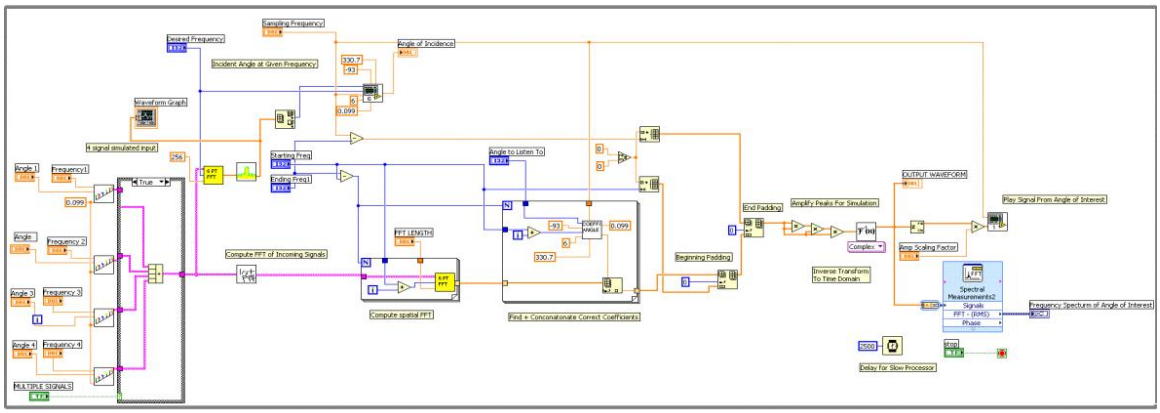

Figure 1: Top Level Diagram

2 Simulated Signal Generation VI

Simulate Signal Icon

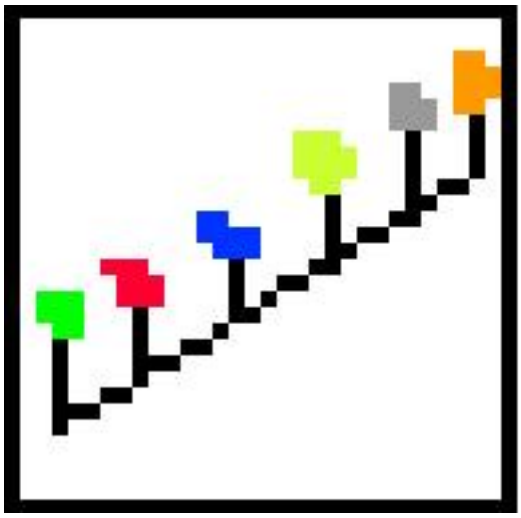

Figure 2: Simulate Signal Icon

Signal Generation VI

The above icon corresponds to the Signal Generation VI. At the far left are four signal generation VIs. Each of these VIs produces six different signals that simulate the delayed signals that each of the microphones would hear in real life. These six signals are then bunched together in a cluster (pink wire) to keep the amount of wires down. The three inputs to this VI are the frequency of the sinusoid desired, the direction of the signal, and the distnace between microphones (9.9 cm in this example). The user sets these paramaters on the front panel and is free to change them at any time.

Our VI uses the formula discussed in previous modules that relates the time delay of signals to the distance between microphones. Using the desired angle that the signal is coming from, the speed of sound (C), the distance between microphones, and a conversion between degrees and radians, the VI first computes the time delay between microphones. Becuase we are interested in the delay between the first microphone and all others, the amount of delay is multiplied by zero for the first mic, one for the second mic, two for the third, and so on. These delays are then used as phase inputs to the six identical Simulate Signal VIs. Adding phase to a periodic signal like the sinusoids being used here has the same effect as delaying the signal in time. Finally the six signals are merged together into a cluster so that they will be easier to deal with as a group in other VIs.

Simulate Signal Block Diagram

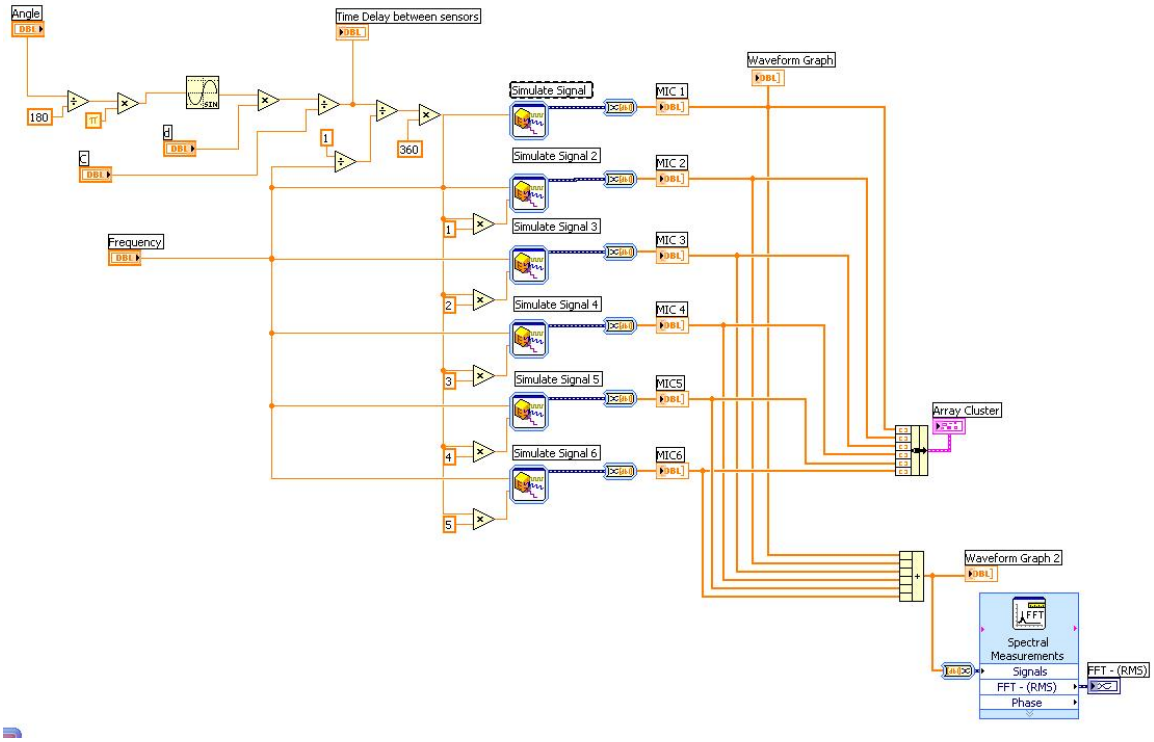

Figure 3: Simulate Signal Block Diagram

In this simulation we simulate four different frequencies and directions. Once we have the data of these four separate signals, we sum the signals on each of the channels to get the final output on each of the microphones. Doing so leaves us solely with what we would hear on the six microphones if this were set up in real life. If more signals are required for future testing or other applications, onc can simply copy and

paste the signal generation VI and then sum it with the other signals.

Once this signal is comple, we move to the right in our VI. From here we branch into two separate areas. Following the pink line up takes us to where we calculate the angle that a given frequency is coming from,and to the right is where we perform the calculations to listen to signals from a given direction.

3 Frequency and Spatial FFT Computation VIs

1rst FFT Icon (Time - Frequency)

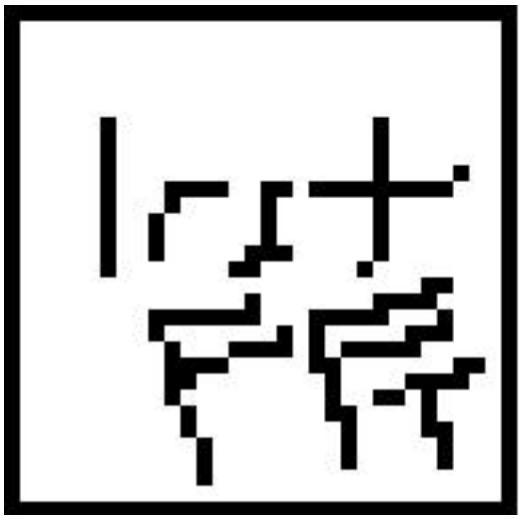

Figure 4: 1rst FFT Icon

1rst FFT VI

In order to listen to a certain direction we first need to transform our signals from the microphone in to the frequency domain. To do this we made the "1rst FFT" Vi (see icon above). This sub-VI is fairly simple so there is no need to show the block diagram. It takes the FFT of each of the six channels channels, transfroming the data from the time domain to the freqeuncy domain For simulation, we looked at 1 second samples at 4000hz. This VI then takes the FFTs of the six 4000 element long signals from the simulated microphones.

Six Point FFT Icon (Frequency - Spatial)

Figure 5: Six Point FFT Icon(Frequency - Spatial)

Six Point FFT VI

Moving to the right after the first FFT, we find the "6 pt FFT" VI. This VI is used to transform our frequency domain data into the spatial domain. This VI needs to run for every frequency of interest, so in the top level diagram this VI is found inside of a for loop. The user can control what range of frequencies are of interest on the front panel of the main VI, and the loop will run for those frequencies.

Inside this VI, the complex frequency coefficient for the given frequency is extracted from the array for each microphone channel. These six values are then concantonated into a lengh six array. Next, the array is zeropadded to a user specified length (we used 256 in our simulation) so more resolution can be observed in the resulting spatial transform. Finally, the FFT is performed on these values transforming them into the spatial domain. With the data in the spatial domain we are easily able to figure out the magnitude of any frequency from any direction in our 180 degrees of interest. How to do this can be found in the magnitude VI.

Six Point FFT block Diagram (Frequency - Spatial)

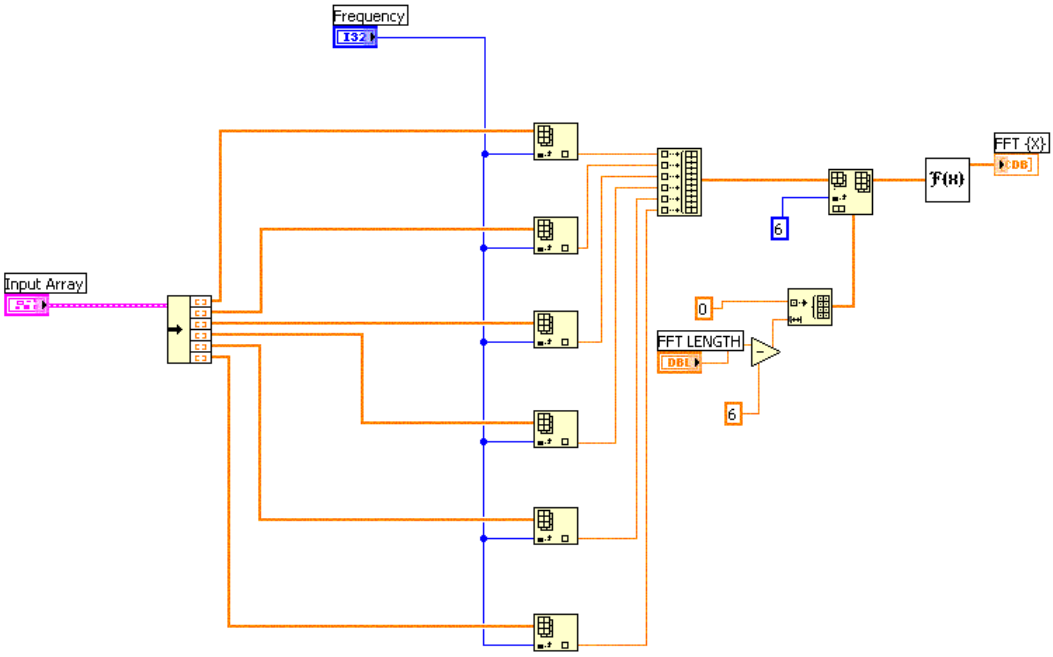

Figure 6: Six Point FFT block Diagram (Frequency - Spatial)

4 Coefficient, Angle, and Magnitude Calculation VIs

Coefficient Calculator

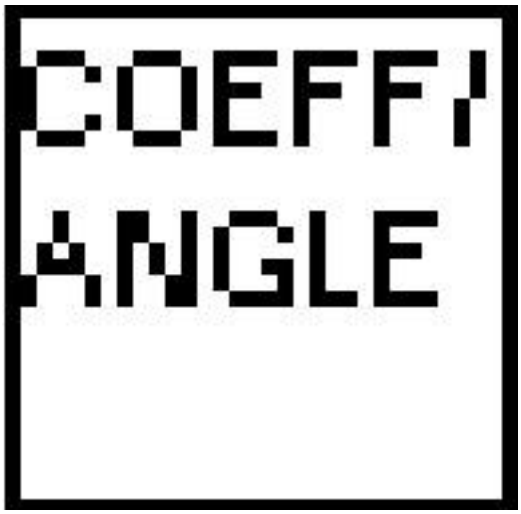

Figure 7: Coefficient Calculator

Coefficient Calculator
Now that the signals have been transformed to their spatial representations, the correct coefficients to construct the signal of interest must be computed. To do this we created the Coefficient Calculator VI.. This VI takes in the angle of interest, a frequency, the number of microphones, a constant factor (determined by the setup of the microphones and the amount of zero padding...in this example it is -93), and our original sampling frequency to compute the correct coefficient to look at for the given angle. Once this coefficient is found, we extract the value at that coefficient and append it to our ouptut array as the value at that given frequency. Below is the block diagram for this VI. It consists of a basic formula node and some logic on the front and back to make sure that the calculations are correct when working with angles from negative directions.

Because this formula is dependent on the frequnecy of interest, we are required to run this VI for every frequency we are intereseted in. In order to do this, we put this VI inside a for loop that is controlled by our frequency range. Any coefficient for frequencies outside of this range are simply given a value of zero. The array modules ouside of this for loop are used to do just that. They append arrays with value zero on the front and back of the output of the for loop to return our vector to its original size. From here, we run this vector through a few multiplies to amplify the differnce between the coefficients with high and low magnitudes, and finally we inverse FFT it to get our output array. This array represents a signal in the time domain and is graphed on the front panel along with a graph of its frequency components. We also included a small VI that will play the output waveform on computer speakers. This VI uses a matlab script and requires the user to have matlab 6.5 or earlier.

Magnitude Graph of Coefficients After Spatial FFT

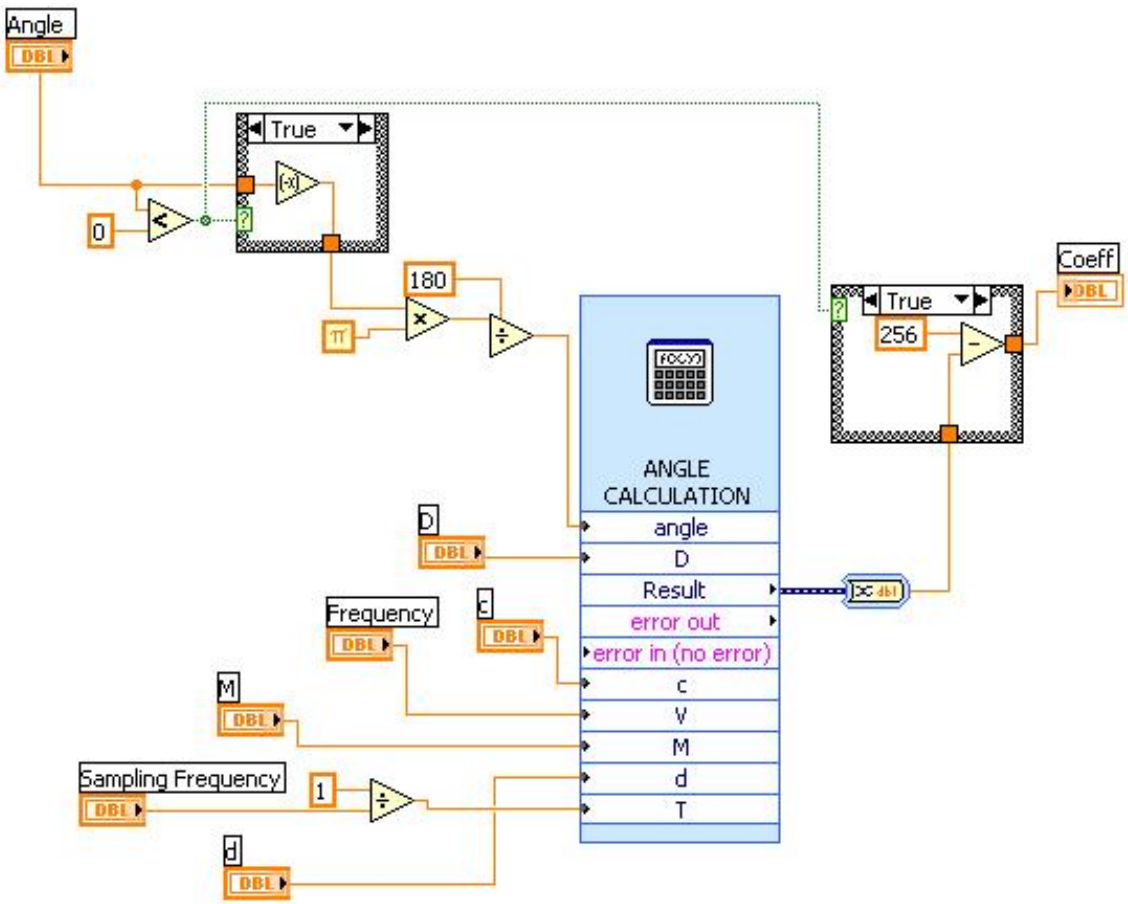

Figure 8: Magnitude Graph of Coefficients After Spatial FFT

Magniutde Calculation VI

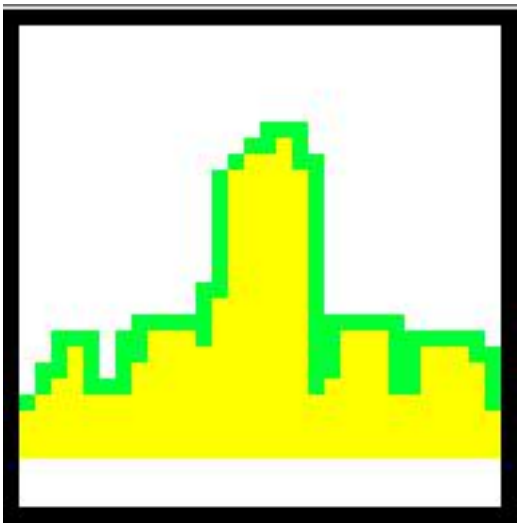

Figure 9: Magnitude Calculation VI

Magnitude Calculation

If we go back to the branch in the pink wire immediatly after the signal generation VIs and move upwards instead of to the right, we come across the code that is used to calculate the angle at which the largest magnitude signal of a given frequency is approaching the array. Another "6 pt fft" VI is used, but this one is slightly modified. It also includes the initial FFTs of all 6 channels. We grouped these two VIs together because ony one spacial FFT is being computed (that at the frequeny of interest).

The resulting vector of the previous six point FFT is immediately used as the input to the Magnitude Calculation VI. The vecor of spatial coefficients from the "six pt FFT" vis are complex, so this VI is used to calculate the magnitudes of the coefficients so the maximum coefficient can be found. The output of this VI is also used to create a visual representation of what direction the specified frequency is coming from. Below is a graph of the magnitude of the coefficients of the spatial FFT. As discussed before, we see the peak correspoinding to the incoming direction and the smaller ripples to each side.

Magnitude Graph of Coefficients After Spatial FFT

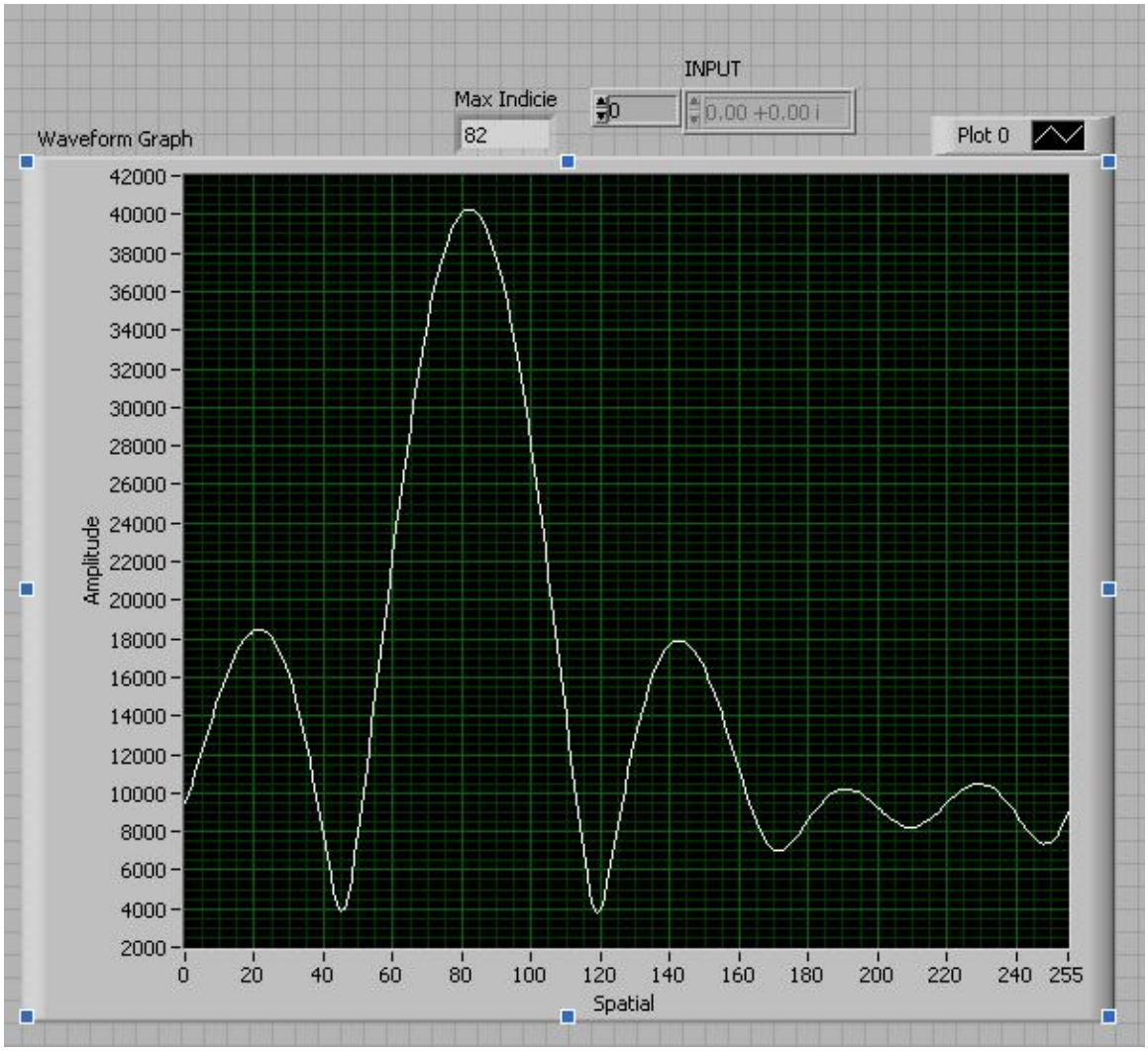

Figure 10: Magniutde of Coefficients after Spatial FFT

Example 1: Magnitude Graph Example

As we can see in the previous figure, the magnitude of the spatial FFT is greatest around coefficient 82. There are also smaller ripples that die off around each end. This graph tells us that the the direction that the given frequency is coming from corresponds to the angle represneted by coefficient 82. To figure out what angle this was, we would use our Coefficient to Angle VI.

Calculation of Angle

Finally, we isolate the index of the maximum angle and use it to compute the angle of incidence. Using the same formula used in the Coefficient Angle Calculator, the Coefficient to Angle VI deduces the angle of incidence. This VI uses the same formula found in the Coefficient Angle Calculator, but is arranged differently so that we can find the angle based on the coefficient instead of the coefficient based on the angle. Once the VI computes this value, the result is output on the front panel.

4.1 Links to Labview Code

- Multiple FrequencySimulation (Top Level VI)[1]
- Coefficient Calculator[2]
- Incident Angle Calculator[3]
- Play Array[4]
- Six Point FFT (with first fft)[5]
- Six Point FFT (without first fft)[6]
- Magnitude Calculation[7]
- First FFT[8]
- Simulate Signal[9]

5 Code Remarks

Overall, the code involved in this method of array signal processing can be broken up into smaller parts that are easy to understand. By combining these smaller parts we are able to create an upper level VI that performs a complicated task that would be difficult to get working using other methods. The major problem with this VI is that it requires a large number of calculations. To increase performance (without upgrading computers) one could decrease the frequency range of interest, or they could lower the amount of zeropadding. They could aslo look at a smaller time period.

[1] http://cnx.org/content/m12565/latest/Multiple_Frequency_Simulation.vi
[2] http://cnx.org/content/m12565/latest/Coefficient_Calculator.vi
[3] http://cnx.org/content/m12565/latest/Incident_Angle_Calculator.vi
[4] http://cnx.org/content/m12565/latest/play_array_anything.vi
[5] http://cnx.org/content/m12565/latest/Six_Point_FFT_with_first_fft.vi
[6] http://cnx.org/content/m12565/latest/Six_Point_FFT_without_first_fft.vi
[7] http://cnx.org/content/m12565/latest/Magnitude_Calculation.vi
[8] http://cnx.org/content/m12565/latest/First_FFT.vi
[9] http://cnx.org/content/m12565/latest/Simulated_Input.vi