

Mini-Project 05: Note Detection

Background

For this mini project you will write a simple LabVIEW VI that listens to a tone and identifies what note it is. The files **Note0N.wav** (where N is a number from 2 to 9) were zipped and are right below this file on the DSP First web site. Your job is to decide what note is in each file. Use the spectrogram on the **tripleDisplay** to find the frequency of the lowest note. Call it f_0 . The highest note should be one octave above ($2f_0$).

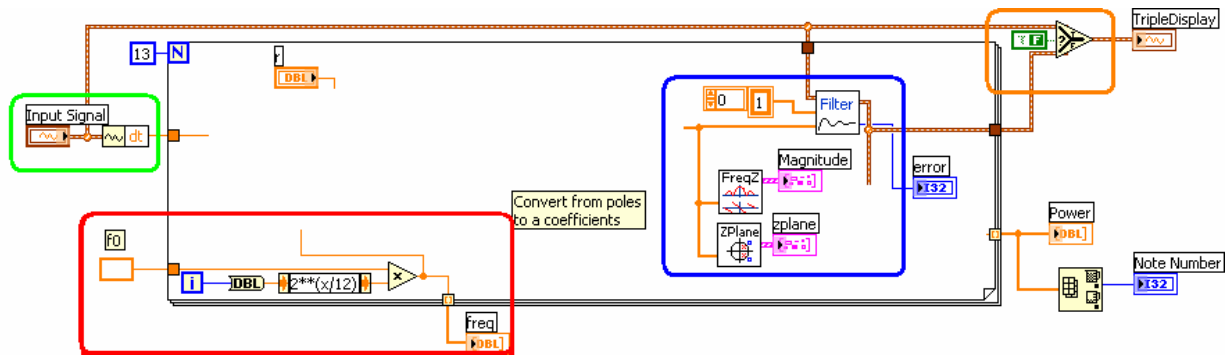
Approach

You are to design 13 filters. Each filter will pass just one note. The first filter will be tuned to pass f_0 ; the next filter will pass $f_0 * 2^{1/12}$ Hz, and so on. The 13th filter will pass $2f_0$. See Mini Project 1 if you don't understand the $2^{1/12}$. You will then pass the unknown signal through each of the filters, i.e., the input signal is the input to each filter. You then measure the power coming out of each filter and the filter with the largest power corresponds to the note that was played.

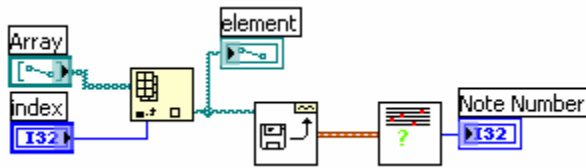
Sample Code

The following LabVIEW code is a good starting point. Things to note:

- A *for* loop surrounds most of the blocks. The upper left of the loop shows 13 is wired to N . Each iteration of the for loop applies one filter. 13 iterations, 13 filters.
- The blocks in the lower left **red** rectangle compute the passband frequency of each filter. $f0$ and i are used. You will have to fill in your own value for $f0$. What values does i take on? Given those values, and your value for $f0$, draw a table that shows which values of i result in value of $freq$.
- The upper left **green** rectangle shows how the **Get Waveform Components** block is used to get the time between samples, dt , for the input signal. Note: $dt=1/fs$.
- Use dt and $freq$ to compute $\hat{\omega}$.
- Use the value of $\hat{\omega}$ to compute the a coefficients. You b_0 coefficient will be 1 and all the others zero.
- In the **blue** rectangle you see **freqz** and **zplane** being used to check to see if the filter makes sense. You also see the **filter** VI filtering the signal.
- You need to add some blocks to compute the power in the signal coming from the filter. Note the use of the **Array Min & Max** block to decide which filter output was the loudest. We are not using the max value from this block, but rather the max value **index**. Why are we doing this?
- Note the use of the **Select** block in the **orange** rectangle to select the input or the output of the **filter** for displaying.
- Build this as a sub VI. That is, attach **Input Signal** to an input and **Note Number** to an output.
- You can always right-click on a block and select **Select Breakpoint**. This will cause LabVIEW to stop every time it executes the selected block. You can then use **Probes** to see the values being carried by the wires at that time.



Test your code with something like:



Here is how it works:

- Array** is an array of **path constants**, one entry in the array for each test wav file.
- The **Index Array** block selects which file you want to test.
- The **wavread** VI reads the wav file and sends it to the **NoteDetect** sub VI that you are writing.
- NoteDetect** passes the input signal through all 13 filters and returns the number of the filter with the most power out.

Filter the Frequencies

You should know enough to pick the a coefficients for your filter. Be sure the a coefficients are real so your output will be real. Also be sure your poles aren't on the unit circle; it could become unstable. Think in terms of poles and zeros and where you would like to place them. Don't use any more poles/zeros than needed. Hint: LabVIEW's **Create Polynomial From Roots** VI may come in handy. In your memo, note the pole/zero locations you are using.

What's Due

For this mini-project you are to work on your own. You may discuss your ideas with others, but you may not share LabVIEW files. The policy stated on the Homework Procedures handout under "Responsibility" applies to this and other mini-projects.

What is due:

- One page memo describing what you did. Be sure to have a sentence or two intro and conclusion. Have a table listing the wav file name (e.g. Note02.wav), and the note you think is in the file for each wav file. Highlight any **extras** you did.
- Include a pole/zero plot (use `zplane`) and a frequency response plot (use `freqz`) of one of your filters. You can get these plots by right-clicking on the desired display and selecting **Data Operations»Export Simplified Image...** and then pasting them you're report. Place these inline in your memo, not as separate pages. Be sure to give them captions and refer to them in your memo.

3. Print out your LabVIEW code. Selecting File→Print will start a print wizard. When you reach the “Print Contents” window, select “Icon, description, panel, and diagram.” This will print out the whole works in a compact format.

A hard copy of the memo and LabVIEW printout are to be handed in at the start of class on the due date.