Extra Credit part

Design

The method is a little bit similar to the regular code. We still need to find the index of the maximum of magnitude value. The different points in extra credit part is that this time we get three note input, referring that there are three peaks of each note, (the total is six but they are symmetric.)

The magnitude of these three different peaks are kind of close but they are not exact the same value. Therefore, prediction and tolerance should be considered.

In the while loop, I create another parallel for loop that similar to out loop, except the condition is more complex. The general idea is that the peak value would bigger than its next value and its last value. Use for loop to run through this condition again and again until find the peak and record the index. Another different part in the condition is that where to end counting. All thing are in the while loop so the data we get will finally updated, so I decided always start from the beginning and end at different position. In this way, I first get the biggest value, and then second biggest value and finally get the smallest value. The second peak will find between the start and the end that some distance before the largest index value.

if(   sqrt(  pow(out[k-1][0],2) + pow(out[k-1][1],2) )<  sqrt(  pow(out[k][0],2) + pow(out[k][1],2) )  && 63                         (       sqrt(  pow(out[k][0],2) + pow(out[k][1],2) ) > sqrt(  pow(out[k+1][0],2) + pow(out[k+1][1],2) )    ) )

for(k=0;k<indexPeak1-30;k++){         //find the second largest  index value 74 75                         if( sqrt(  pow(out[k-1][0],2) + pow(out[k-1][1],2) )<  sqrt(  pow(out[k][0],2) + pow(out[k][1],2) )  && 76                         (       sqrt(  pow(out[k][0],2) + pow(out[k][1],2) ) > sqrt(  pow(out[k+1][0],2) + pow(out[k+1][1],2) )    ) && 77                         (               k!=indexPeak1           ) ){ 78                                 indexPeak2 = k;

Testing

The first time I count index of peak from start to the end, but notice in the for loop that in the while loop, the last value would be finally updated and replace the initial one. Counting from start will result that we only get one max peak instead of three.

The second time I count from the start, but each time I meet a peak value, I record that value and then define at that index, the magnitude = 0, and then recount again, and use the same method to find next peak index and so on. In this case, the question is that all things, all the codes are in the while loop, so even if I define the mag = 0, it would do it over and over again. And at the end, the other index value become 0 for all the time.