Homework 2 - Student Report
CMPE 362 Spring 2016

Batuhan Demir - 2013400159

March 16, 2016

1 Questions & Answers

1.1 Clap Counter

ClapCounter.m : In this question we have .wav files in the same file with script and we read them by *dir* function inside for loop. To analyze the voice i manually determined threshold(0.25) above an amplitude and i determined a radius for area near claps so that i am able to clear area. I looked for claps in a while loop as long as amplitudes are higher than threshold. And at the end in since we have only 2 type of outputs, in case of wrong calculation i displayed 'one clap' if number of claps found is lower or equal to 1 and 'two claps' otherwise.

```
hfile = dir('*.wav');

for id = 1 : numel(hfile);
    d = fullfile(hfile(id).name);
    [stereo1, Fs] = audioread(d);
    mono1 = mean(stereo1,2);

    [max_value,idx] = max(mono1);
    threshold = 0.25; %// amplitude threshold
    radius = 4000 ; %// data around clap
    number_of_claps = 0;
    while max_value > threshold
        min_bound = max(1,idx-radius);
        max_bound = min(idx+radius,length(mono1));
        mono1(min_bound:max_bound) = 0; %// after a clap found, delete it
        [max_value,idx] = max(mono1);
        number_of_claps = number_of_claps + 1;
    end

    if number_of_claps<=1
        disp('one clap')
    else
        disp('two claps')
    end
end
```

1.2 Frequency(Pitch) of a Sound

wavexample.m : In this question I simply changed data by jumping over y vector. If interval goes by number less than 1 this means speeding up frequency and otherwise l slowing it. If i needed to change sound by playing with frequencies i simply did mathematical operations on it.

```
% % % Exercise 1 % % %
sound(y(1:4:end), Fs);
duration = numel(y) / Fs;                    % Calculate the duration
pause(duration + 2)
% % % Exercise 2 % % %
sound(y(1:0.5:end), Fs);
duration = numel(y) / Fs;                    % Calculate the duration
pause(duration + 2)
% % % Exercise 3 % % %
sound(y, 2*Fs);
duration = numel(y) / Fs;                    % Calculate the duration
pause(duration + 2)
% % % Exercise 4 % % %
sound(y, Fs/2);
```

```
duration = numel(y) / Fs;                    % Calculate the duration
pause(duration + 2)
```

1.3 Spline Interpolation

piecequad.m : In this question first i determined an interval where interpolation is(u). And corresponding y values(splinevals). First interval between 1. and 2. data is linear line and its slope is *dy*. I computed the coefficients in a for loop without *linsolve* method and in the end function returns variable 'v' which corresponds y values in figure. Evaluation of interpolant is done by function piecequad.m and actual plotting is done in the quadspline.m script.

Actual figure is different than quadratic spline plotting. We have 9 data points and between these data we draw spline by quadratic equations in quadratic interpolation but in actual drawing it is same polynomial figure.

```
function v = piecequad(x,y,u)
%
%  v = piecequad(x,y,u) finds the quadratic spline q(x)
%  with q(x(j)) = y(j), q'(x(1)) = dy and q'(x) continuous.
%  returns v(k) = q(u(k)). dy = f(x(1)) i.e. the value of the
%  derivative at the left endpoint.

%  Find subinterval indices k so that x(k) <= u < x(k+1)

   n = length(x);
   b = zeros(n);
   c = zeros(n);

   dy=2.77;

   b(1) = dy;
   h = x(2)-x(1);
   c(1) = (y(2)-y(1)-dy*h)/h^2;

   k = ones(size(u));
   for j = 2:n-1

%     compute coefficients in quadratic spline
      b(j) = -b(j-1)+2*(y(j)-y(j-1))/(x(j)-x(j-1));
      h = x(j+1)-x(j);
      c(j) = (y(j+1)-y(j)-b(j)*h)/h^2;

%     find sub-intervals where lie points u where to interpolate
      k(x(j) <= u) = j;
   end

%  Evaluate interpolant

   s = u - x(k);
   v = y(k) + b(k).*s + c(k).*s.^2;
```