LAB 10 Spectral analyis lab

Table of Contents

Run tests

```
test_lab8_2024
```

```
s1: 123456789*0# is O.K.
s2: 123456789*0# is O.K.
s3: 123456789*0# is O.K.
s4: 123456789*0# is O.K.
s5: 123456789*0# is O.K.
s6: 123456789*0# is O.K.
```

Print program

```
disp(' ')
disp('--- dtmfdecode.m --------------------------------')
type('dtmfdecode')
```

```

--- dtmfdecode.m --------------------------------

function str = dtmfdecode(s, fs)
% DTMFDECODE Decode DTMF tones
%            str = decodedtmf(s, fs)
%            Accepts a array, s, which corresponds to the DTMF tones
%            sampled at fs.
%            Produces a string transcript that decodes the tones.
%
% Authors: Saul J. Cervantes-Hernandez, Jose A. Leandro

    % chunck the signal into 10ms windows
    wl = fs * 0.01; % window length
    windows = [];
    for i = 1:length(s)/wl
        l_bound = (i-1)*wl + 1;
        r_bound = i*wl;
        % append the current window as a row
        windows = [windows; s(l_bound:r_bound)'];
    end

    % calculate energy of each window
    energies = floor(sum(windows.^2, 2));
    max_energy = max(energies); % used for reference

    % find the tones by looking for windows with high energy
    tones = []; % holds all tones
    temp = []; % holds the current tone
    for i = 1:length(energies)
        % if the energy is within 50% of the max energy then
        % assume it is a tone
        if energies(i) > max_energy * 0.5
            % append to temp to make a single tone
            temp = [temp, windows(i, :)];
        elseif length(temp) > 0
            % append the temporary tone as a row
            tones = [tones; temp];
            temp = [];
        end
    end
    tones = [tones; temp]; % append the last tone

    % characters for the dtmf tones
    dtmf = ['1' '2' '3';...
            '4' '5' '6';...
            '7' '8' '9';...
            '*' '0' '#'];

    % standard frequencies for the dtmf tones
    row_freqs = [697, 770, 852, 941];
    col_freqs = [1209, 1336, 1477];

    tone_length = length(tones(1, :));
    % frequencies as indices
    row_freqs = round(row_freqs * tone_length / fs) + 1;
    col_freqs = round(col_freqs * tone_length / fs) + 1;

    % build string
    str = '';
    for i = 1:length(tones(:, 1))
        % get magnitudes of the fft
        curr_tone = abs(fft(tones(i,:)));
        % find the row and column frequencies
        [val, row] = max(curr_tone(row_freqs));
        [val, col] = max(curr_tone(col_freqs));

        str = [str, dtmf(row, col)];
    end
end
```

Published with MATLAB® R2023b