Digital Signal Processing

Lab 2

Rozaliya Amirova, BS3-SE

Disclaimer: I made function in separate files, because my Matlab version does not support putting them in one. Sorry for any inconvenience.

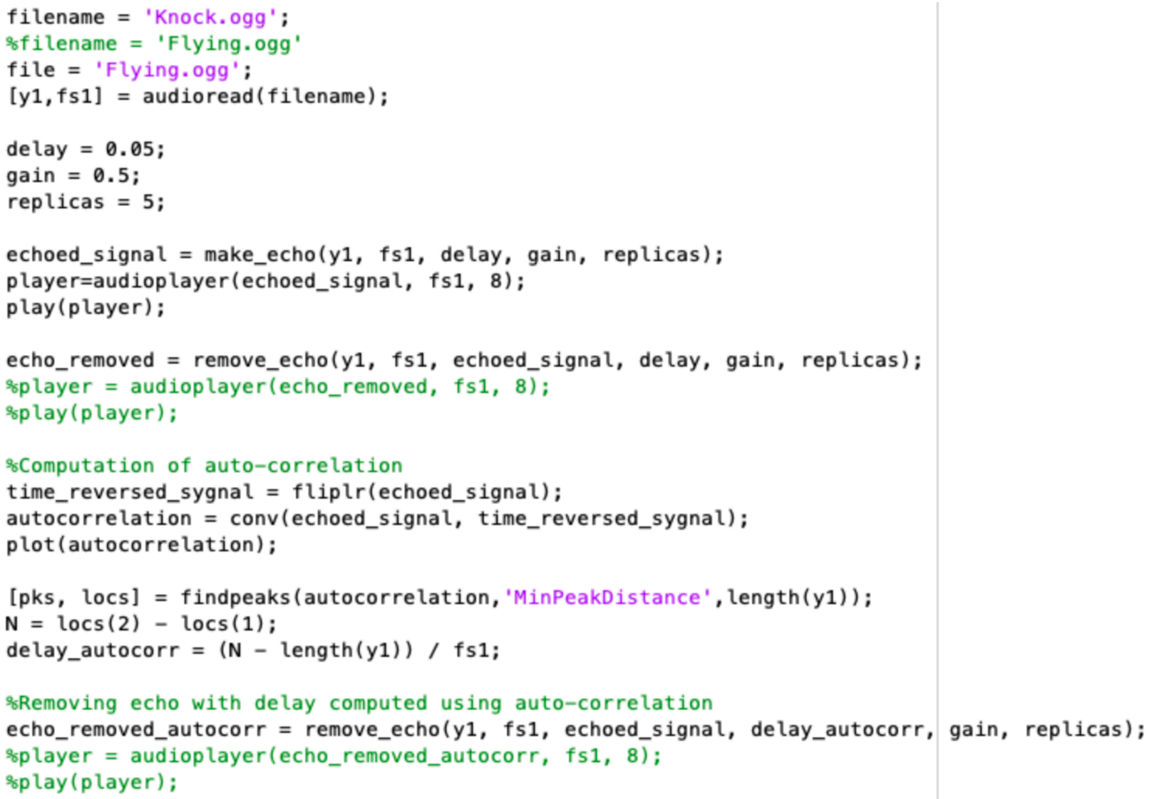
1. Base assignment

Files:

Flying.ogg
Knock.ogg
Lab2.m
make_echo.m
remove_echo.m

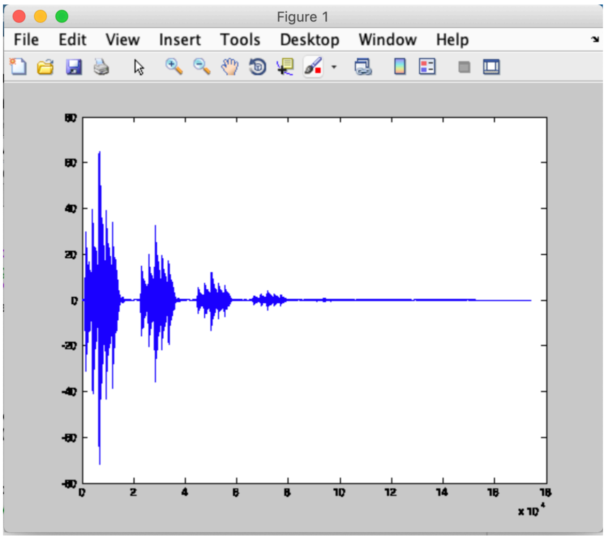
I used 2 audio samples: knocking and artificial notification sound.

There you can see echo generation function call (you can specify number of echo replicas), then removing echo with hard-coded delay value. Then I compute delay using autocorrelation. Autocorrelation can be computed as covariance from signal and its time-reversed version. Then I remove echo using computed delay value.

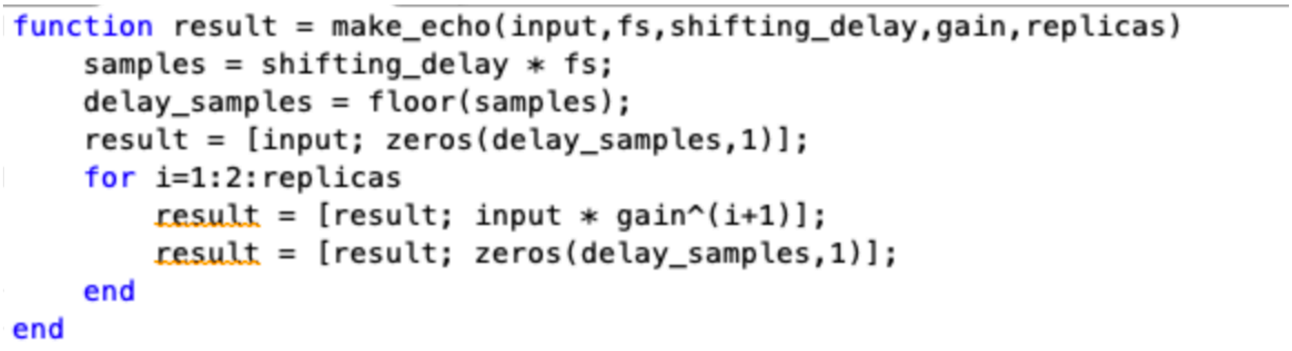


```
filename = 'Knock.ogg';
%filename = 'Flying.ogg'
file = 'Flying.ogg';
[y1,fs1] = audioread(filename);

delay = 0.05;
gain = 0.5;
replicas = 5;

echoed_signal = make_echo(y1, fs1, delay, gain, replicas);
player=audioplayer(echoed_signal, fs1, 8);
play(player);

echo_removed = remove_echo(y1, fs1, echoed_signal, delay, gain, replicas);
%player = audioplayer(echo_removed, fs1, 8);
%play(player);

%Computation of auto-correlation
time_reversed_sygnal = fliplr(echoed_signal);
autocorrelation = conv(echoed_signal, time_reversed_sygnal);
plot(autocorrelation);

[pks, locs] = findpeaks(autocorrelation,'MinPeakDistance',length(y1));
N = locs(2) - locs(1);
delay_autocorr = (N - length(y1)) / fs1;

%Removing echo with delay computed using auto-correlation
echo_removed_autocorr = remove_echo(y1, fs1, echoed_signal, delay_autocorr, gain, replicas);
%player = audioplayer(echo_removed_autocorr, fs1, 8);
%play(player);
```

Autocorrelation:

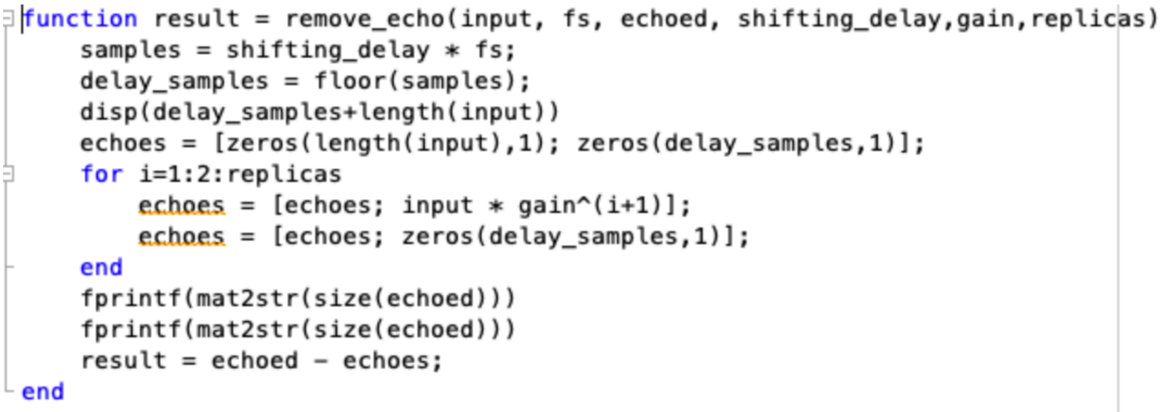


1.1 Echo generation



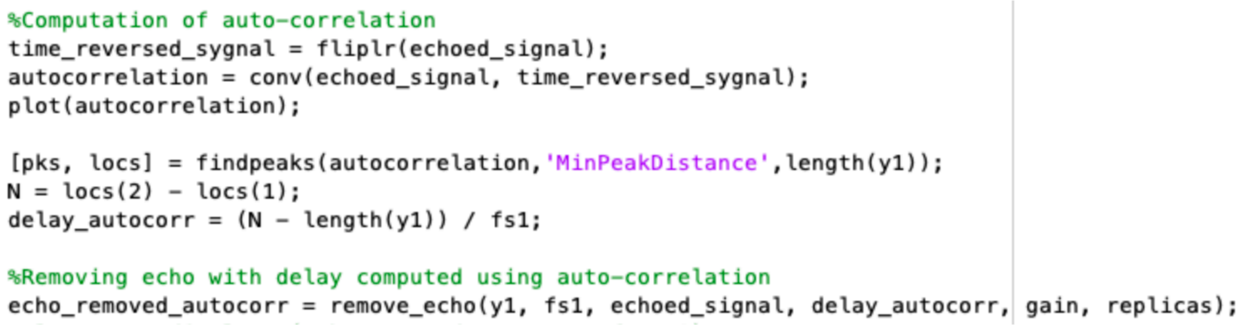
```
function result = make_echo(input,fs,shifting_delay,gain,replicas)
    samples = shifting_delay * fs;
    delay_samples = floor(samples);
    result = [input; zeros(delay_samples,1)];
    for i=1:2:replicas
        result = [result; input * gain^(i+1)];
        result = [result; zeros(delay_samples,1)];
    end
end
```

1.2 Echo cancellation



```
function result = remove_echo(input, fs, echoed, shifting_delay,gain,replicas)
    samples = shifting_delay * fs;
    delay_samples = floor(samples);
    disp(delay_samples+length(input))
    echoes = [zeros(length(input),1); zeros(delay_samples,1)];
    for i=1:2:replicas
        echoes = [echoes; input * gain^(i+1)];
        echoes = [echoes; zeros(delay_samples,1)];
    end
    fprintf(mat2str(size(echoed)))
    fprintf(mat2str(size(echoed)))
    result = echoed - echoes;
end
```

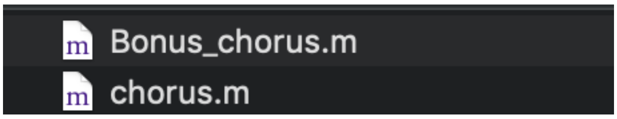
1.3 Echo cancellation using autocorrelation



```
%Computation of auto-correlation
time_reversed_sygnal = fliplr(echoed_signal);
autocorrelation = conv(echoed_signal, time_reversed_sygnal);
plot(autocorrelation);

[pks, locs] = findpeaks(autocorrelation,'MinPeakDistance',length(y1));
N = locs(2) - locs(1);
delay_autocorr = (N - length(y1)) / fs1;

%Removing echo with delay computed using auto-correlation
echo_removed_autocorr = remove_echo(y1, fs1, echoed_signal, delay_autocorr, gain, replicas);
```

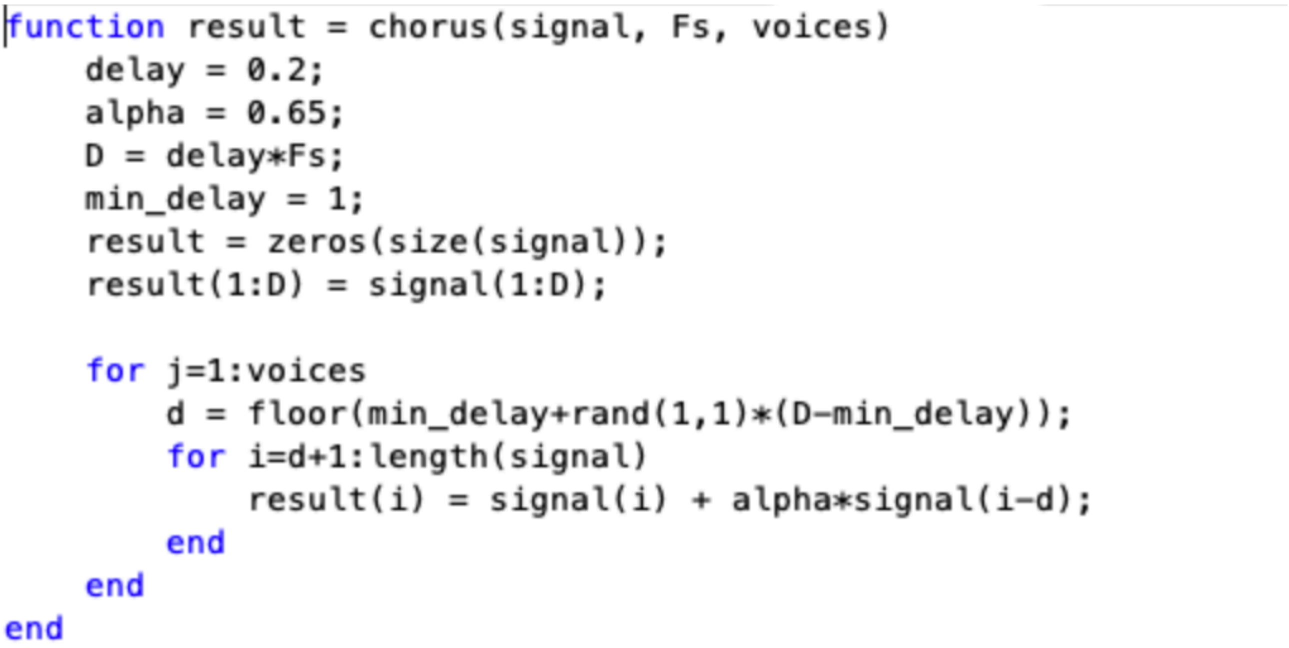
2. Chorus generation

Files:



Bonus_chorus.m

chorus.m

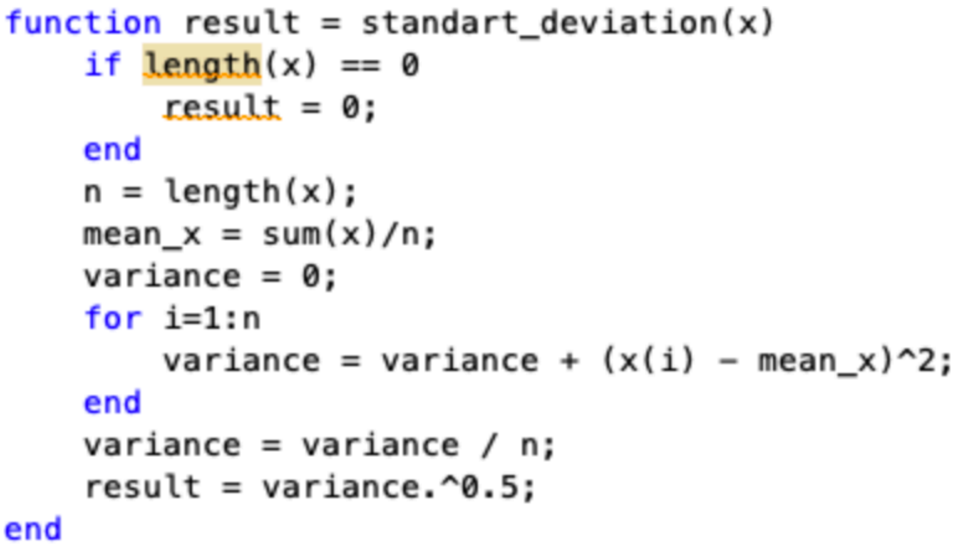


```
function result = chorus(signal, Fs, voices)
    delay = 0.2;
    alpha = 0.65;
    D = delay*Fs;
    min_delay = 1;
    result = zeros(size(signal));
    result(1:D) = signal(1:D);

    for j=1:voices
        d = floor(min_delay+rand(1,1)*(D-min_delay));
        for i=d+1:length(signal)
            result(i) = signal(i) + alpha*signal(i-d);
        end
    end
end
```

3. Functions implementation

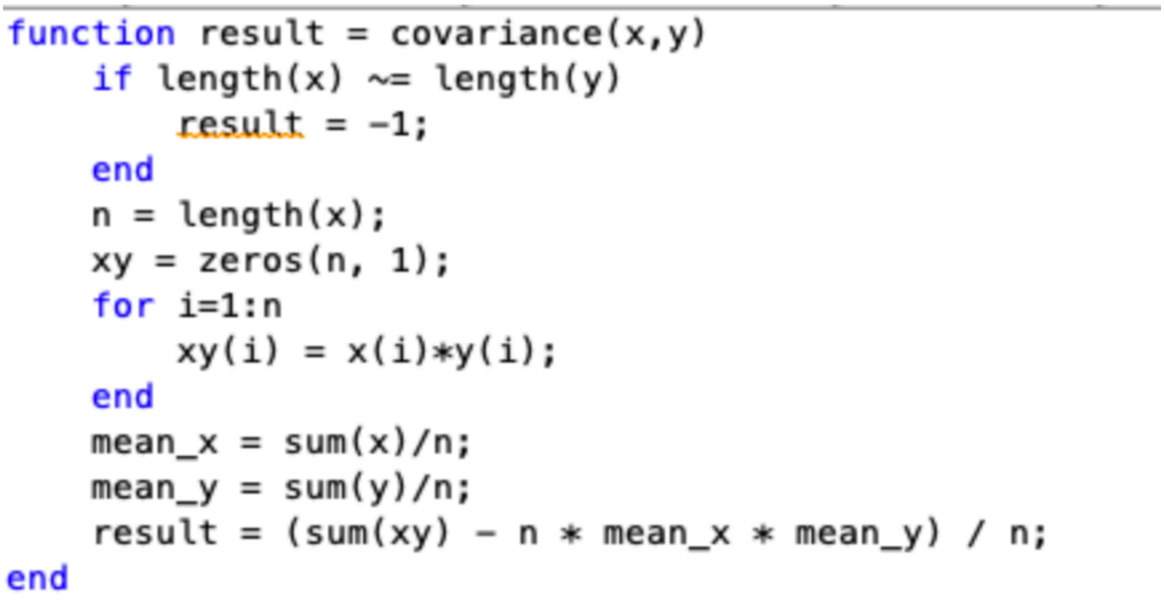
In "correlation" package

3.1 Standard deviation



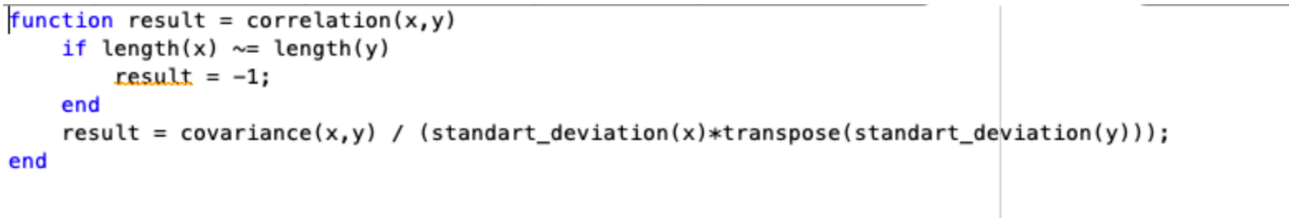
```
function result = standart_deviation(x)
    if length(x) == 0
        result = 0;
    end
    n = length(x);
    mean_x = sum(x)/n;
    variance = 0;
    for i=1:n
        variance = variance + (x(i) - mean_x)^2;
    end
    variance = variance / n;
    result = variance.^0.5;
end
```

3.2 Covariance



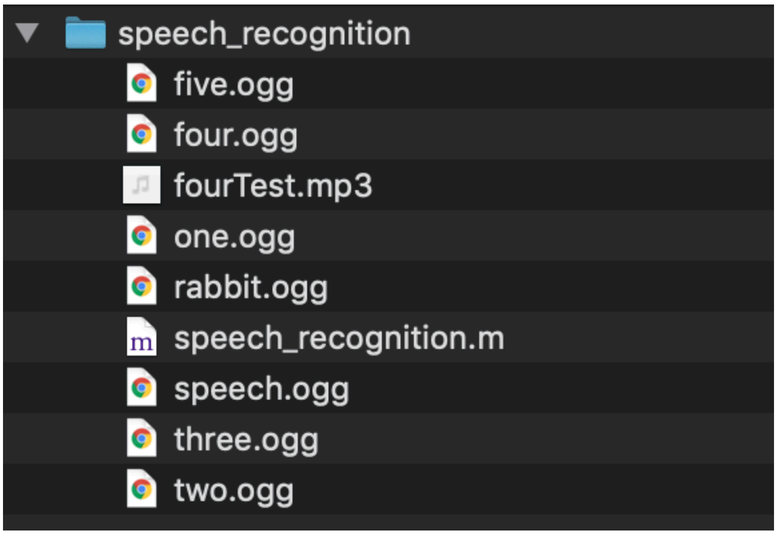
```
function result = covariance(x,y)
    if length(x) ~= length(y)
        result = -1;
    end
    n = length(x);
    xy = zeros(n, 1);
    for i=1:n
        xy(i) = x(i)*y(i);
    end
    mean_x = sum(x)/n;
    mean_y = sum(y)/n;
    result = (sum(xy) - n * mean_x * mean_y) / n;
end
```

3.3 Correlation



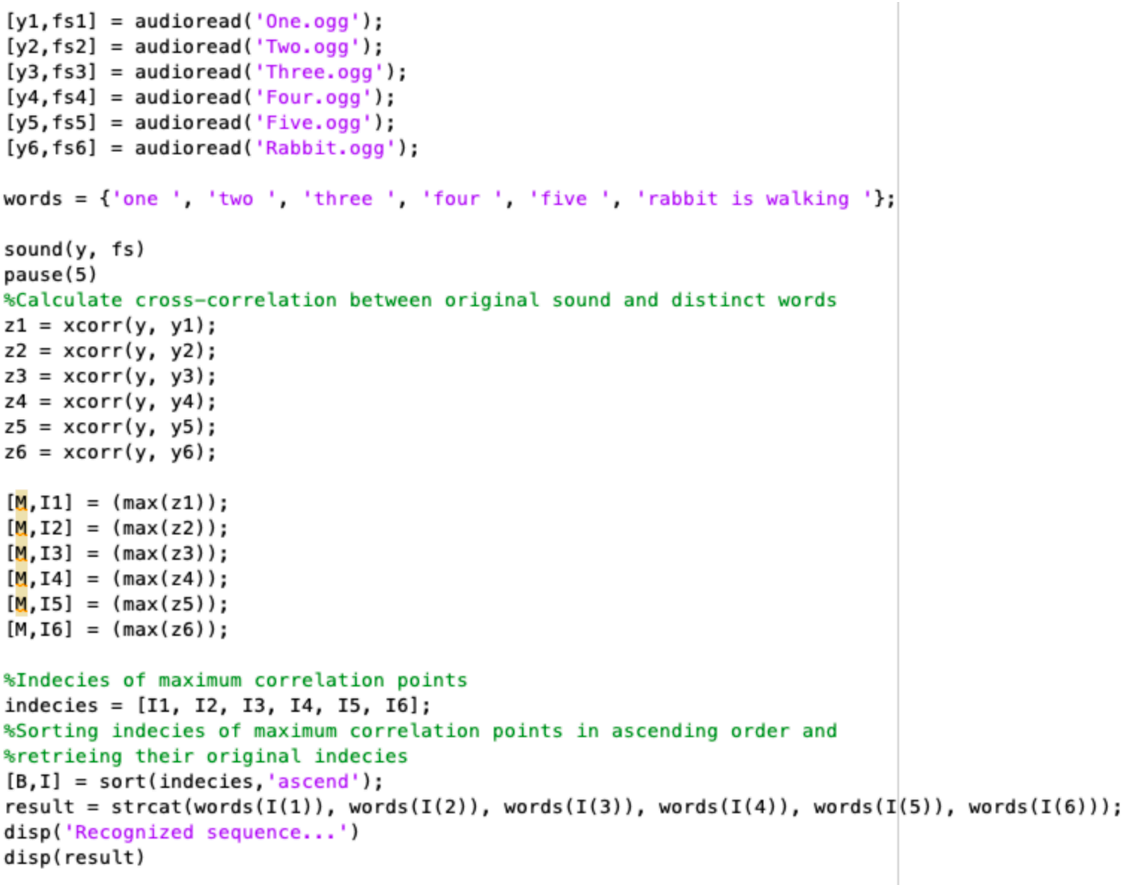
```
function result = correlation(x,y)
    if length(x) ~= length(y)
        result = -1;
    end
    result = covariance(x,y) / (standart_deviation(x)*transpose(standart_deviation(y)));
end
```

4. Speech recognition



```
▼ speech_recognition
      five.ogg
      four.ogg
      fourTest.mp3
      one.ogg
      rabbit.ogg
      speech_recognition.m
      speech.ogg
      three.ogg
      two.ogg
```

4.1 Sequence

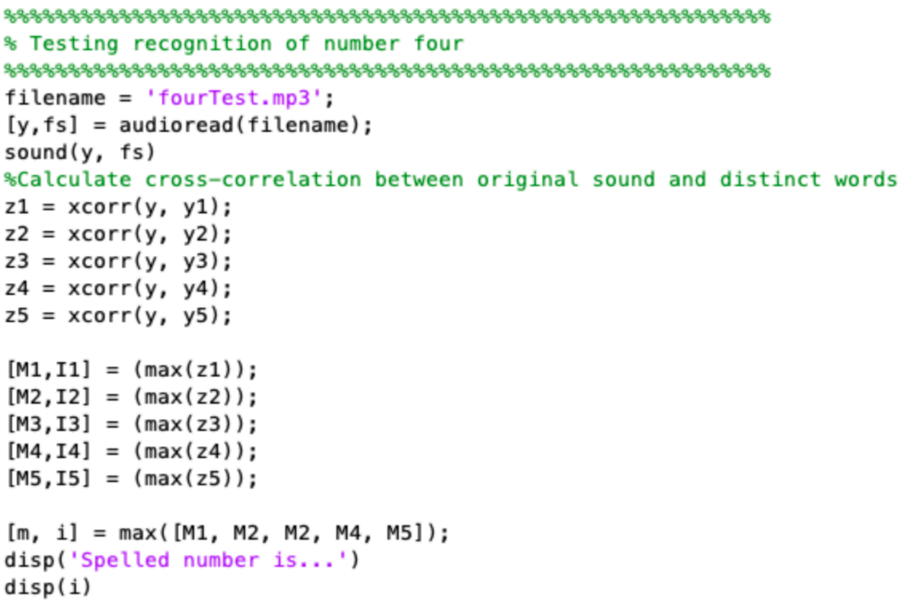


```
[y1,fs1] = audioread('One.ogg');
[y2,fs2] = audioread('Two.ogg');
[y3,fs3] = audioread('Three.ogg');
[y4,fs4] = audioread('Four.ogg');
[y5,fs5] = audioread('Five.ogg');
[y6,fs6] = audioread('Rabbit.ogg');

words = {'one ', 'two ', 'three ', 'four ', 'five ', 'rabbit is walking '};

sound(y, fs)
pause(5)
%Calculate cross-correlation between original sound and distinct words
z1 = xcorr(y, y1);
z2 = xcorr(y, y2);
z3 = xcorr(y, y3);
z4 = xcorr(y, y4);
z5 = xcorr(y, y5);
z6 = xcorr(y, y6);

[M,I1] = (max(z1));
[M,I2] = (max(z2));
[M,I3] = (max(z3));
[M,I4] = (max(z4));
[M,I5] = (max(z5));
[M,I6] = (max(z6));

%Indecies of maximum correlation points
indecies = [I1, I2, I3, I4, I5, I6];
%Sorting indecies of maximum correlation points in ascending order and
%retrieing their original indecies
[B,I] = sort(indecies,'ascend');
result = strcat(words(I(1)), words(I(2)), words(I(3)), words(I(4)), words(I(5)), words(I(6)));
disp('Recognized sequence...')
disp(result)
```

I recorded voice with words "Вышел зайчик погулять 1 5 3 4 2", then calculated xcorr of this recording with distinct words, found indices of highest correlation, sorted them in ascending order and printed corresponded words,

4.2 One number

I made new recoding of number 4 and then compared with samples 1-5. Then took one with maximum correlation and printed corresponding index.



```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Testing recognition of number four
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
filename = 'fourTest.mp3';
[y,fs] = audioread(filename);
sound(y, fs)
%Calculate cross-correlation between original sound and distinct words
z1 = xcorr(y, y1);
z2 = xcorr(y, y2);
z3 = xcorr(y, y3);
z4 = xcorr(y, y4);
z5 = xcorr(y, y5);

[M1,I1] = (max(z1));
[M2,I2] = (max(z2));
[M3,I3] = (max(z3));
[M4,I4] = (max(z4));
[M5,I5] = (max(z5));

[m, i] = max([M1, M2, M2, M4, M5]);
disp('Spelled number is...')
disp(i)
```