

Initialization

```
clear all
N=8000; %N=4000; can be adjusted if the speaker has relevant frequencies in the high register
```

Acquisition

```
seconds=3; % Three seconds for the acquisition time
disp('Press a key and say the password') %Trigger for the acquisition
pause
AI = analoginput('winsound');
addchannel(AI,1:2); %Configure the sampling rate to 11.025 kHz and define a 2 second acquisition.
set(AI,'SampleRate',11025) %set(AI,'SamplesPerTrigger',22050)
start(AI)
wait(AI,seconds);
data = getdata(AI); %Extract the acquired data from the engine and reduce to one channel
leftsig=data(1:N,1);
rightsig=data(1:N,2);
signal=leftsig-rightsig;
```

Plot the data and label the figure axes. (used for visual analysis)

```
%subplot(211), plot(Feml); xlabel('Samples'); ylabel('Signal (Volts)')
%subplot(212), plot(-N/2:N/2-1,abs(fftshift(fft(Feml))), 'b');
%xlabel('frequency')
```

Cepstrum analysis

```
recepts = rceps(signal); %Computes the real and complex cepstrum
concepts = cceps(signal);
in=70; %Analysis example over the cepstrum
su=90;
for a = in:1:su;
A(a-(in-1),1)=recepts(a,1);
end
meanval = mean2(A)
maximval ue = max(A);
absol uteval ue = abs(A);
meanval ue = mean(absol uteval ue);
devl ati on = std(A)
```

DECISION MAKER

```
if meanval ue==meanval ueA
disp('welcome A');
else
if meanval ue==meanval ueB
disp('welcome B')
else
if meanval ue==meanval ueC
disp('welcome C')
else
disp('You are not authorized')
end
end
```