# Department of

# Electrical & Electronics Engineering

## Abdullah Gül University

---

**Progress Report (DSP Part)**

**EE3001 TELECOMMUNICATION SYSTEM DESIGN USING DSP CAPSULE**

---

**Submitted on:** (14.01.2022)

**Submitted by:** (Muhammet İlal 110110212)

(Atıf Koçak 11010244)

**Grade: / 100**

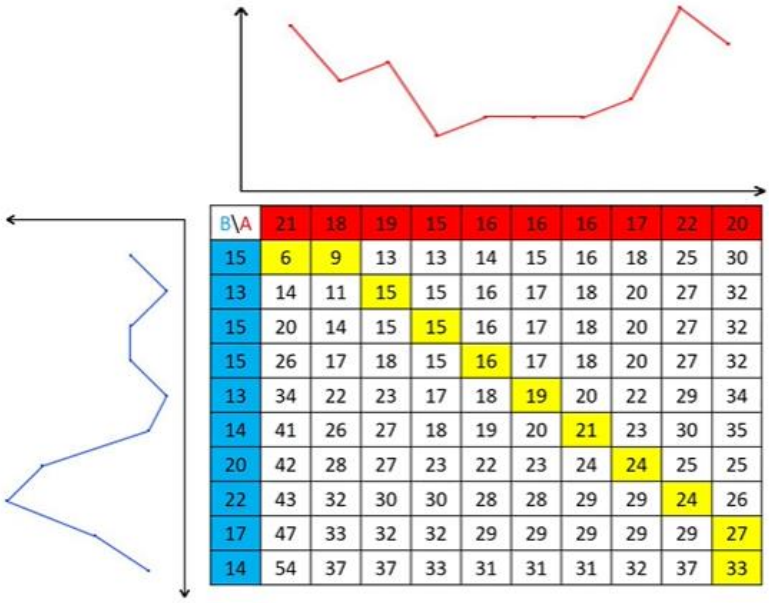
**OBJECTIVE**

In the DSP progress report, which is the third phase of our project, we will introduce the type of DSP we will implement and present the progress we have achieved in our report. Our project aim is to design a communication system that will move an electric chair back and forth and left and right with voice command.

We will use dynamic time warping in our project, and using this type of DSP will allow us to compare the previously recorded audio data with the audio real-time data while processing the audio signal.

Dynamic time warping will compare the previously recorded "forward", "backward", "right" and "left" audio data with the audio signal given to the system by the user in real-time, distinguish the correct voice command and provide the command data needed by the motor of the electric chair.

**BACKGROUND**



| B\A | 21 | 18 | 19 | 15 | 16 | 16 | 16 | 17 | 22 | 20 |
|---|---|---|---|---|---|---|---|---|---|---|
| 15 | 6 | 9 | 13 | 13 | 14 | 15 | 16 | 18 | 25 | 30 |
| 13 | 14 | 11 | 15 | 15 | 16 | 17 | 18 | 20 | 27 | 32 |
| 15 | 20 | 14 | 15 | 15 | 16 | 17 | 18 | 20 | 27 | 32 |
| 15 | 26 | 17 | 18 | 15 | 16 | 17 | 18 | 20 | 27 | 32 |
| 13 | 34 | 22 | 23 | 17 | 18 | 19 | 20 | 22 | 29 | 34 |
| 14 | 41 | 26 | 27 | 18 | 19 | 20 | 21 | 23 | 30 | 35 |
| 20 | 42 | 28 | 27 | 23 | 22 | 23 | 24 | 24 | 25 | 25 |
| 22 | 43 | 32 | 30 | 30 | 28 | 28 | 29 | 29 | 24 | 26 |
| 17 | 47 | 33 | 32 | 32 | 29 | 29 | 29 | 29 | 29 | 27 |
| 14 | 54 | 37 | 37 | 33 | 31 | 31 | 31 | 32 | 37 | 33 |

*Figure: Example of DTW*

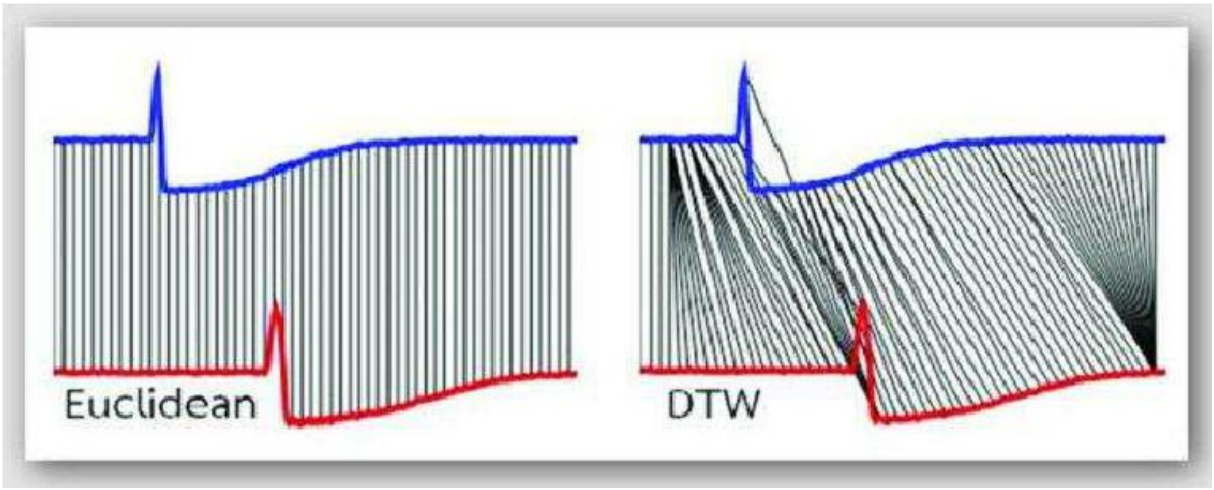


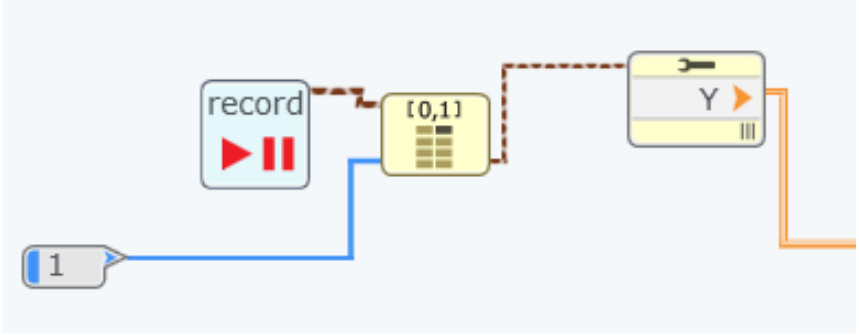

*Figure: Example of the difference between Euclidean and DTW matching*

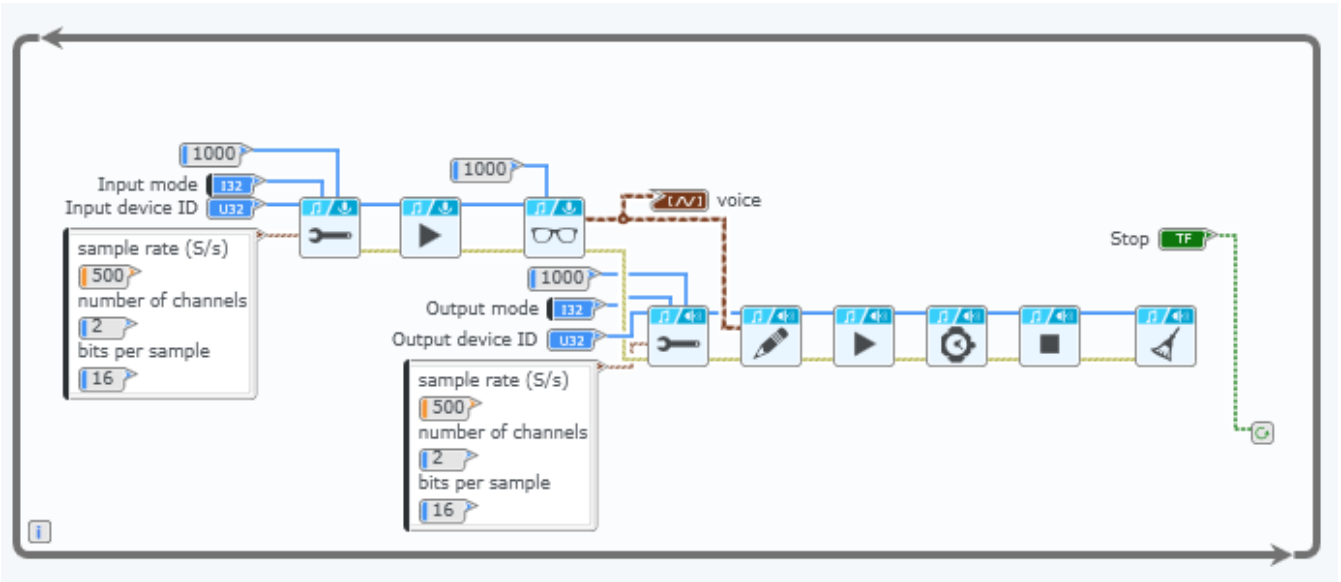
It is a digital signal processing algorithm used to check the similarity of Dynamic time warping (DTW) series to each other. It can be calculated by using Euclidean distances when comparing two data with each other, but as you can see in the example given above, if a comparison is made with Euclidean connection depending on time, Euclidean will not be able to detect the shift in the signal and will give the data that the two signals are different from each other. However, if this comparison is made independent of time with Dynamic time warping, all samples will match with the closest sample and output the similarity of the signals to each other. In this way, the command received from the user using dynamic time warping is compared with the registered commands and can be matched with the most similar one.

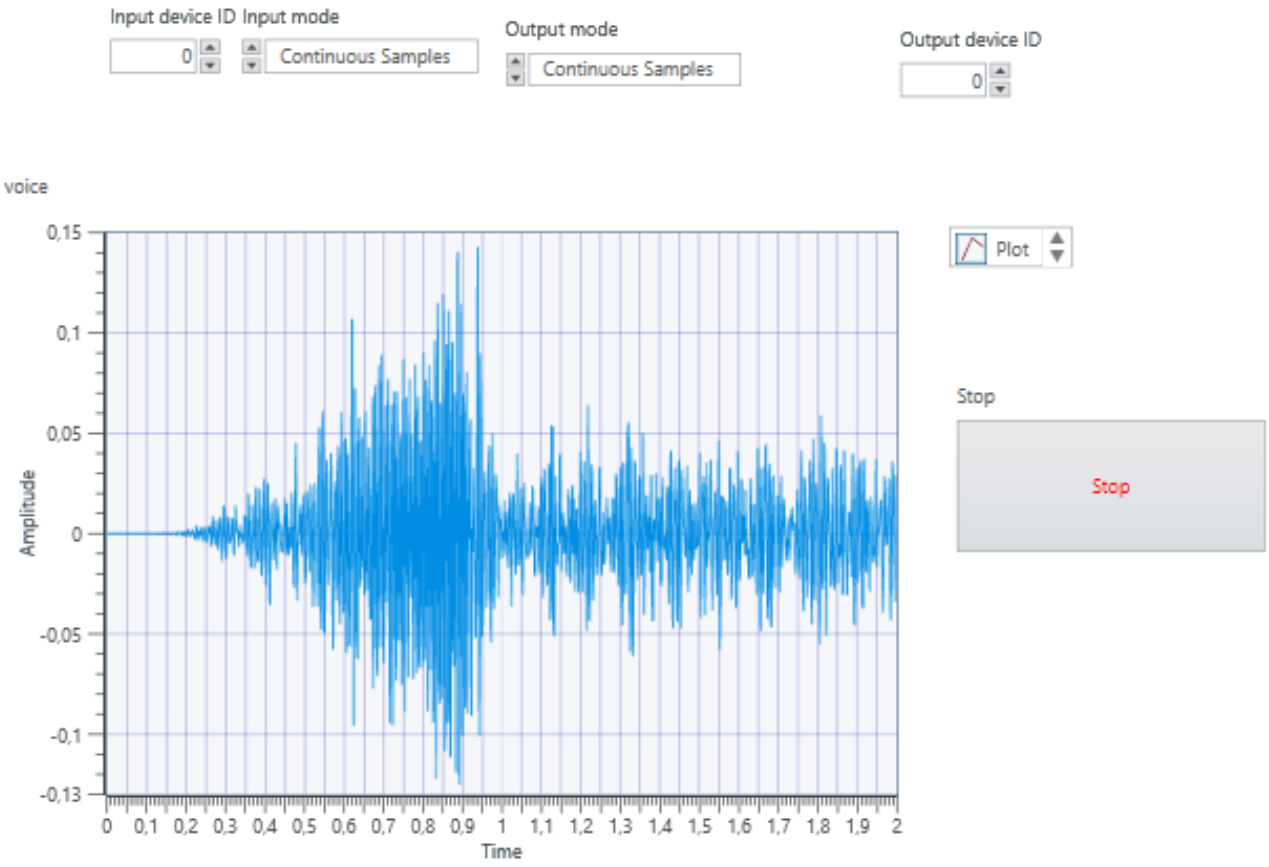
**DESIGN AND PROCEDURE**

After that, DTW is done with using LabVIEW. Steps of the DTW in LabVIEW are below.

- Firstly the sound is recorded in real time. For this purpose, subVI is created.



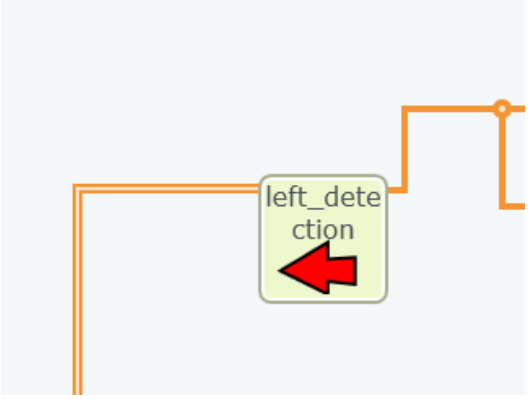



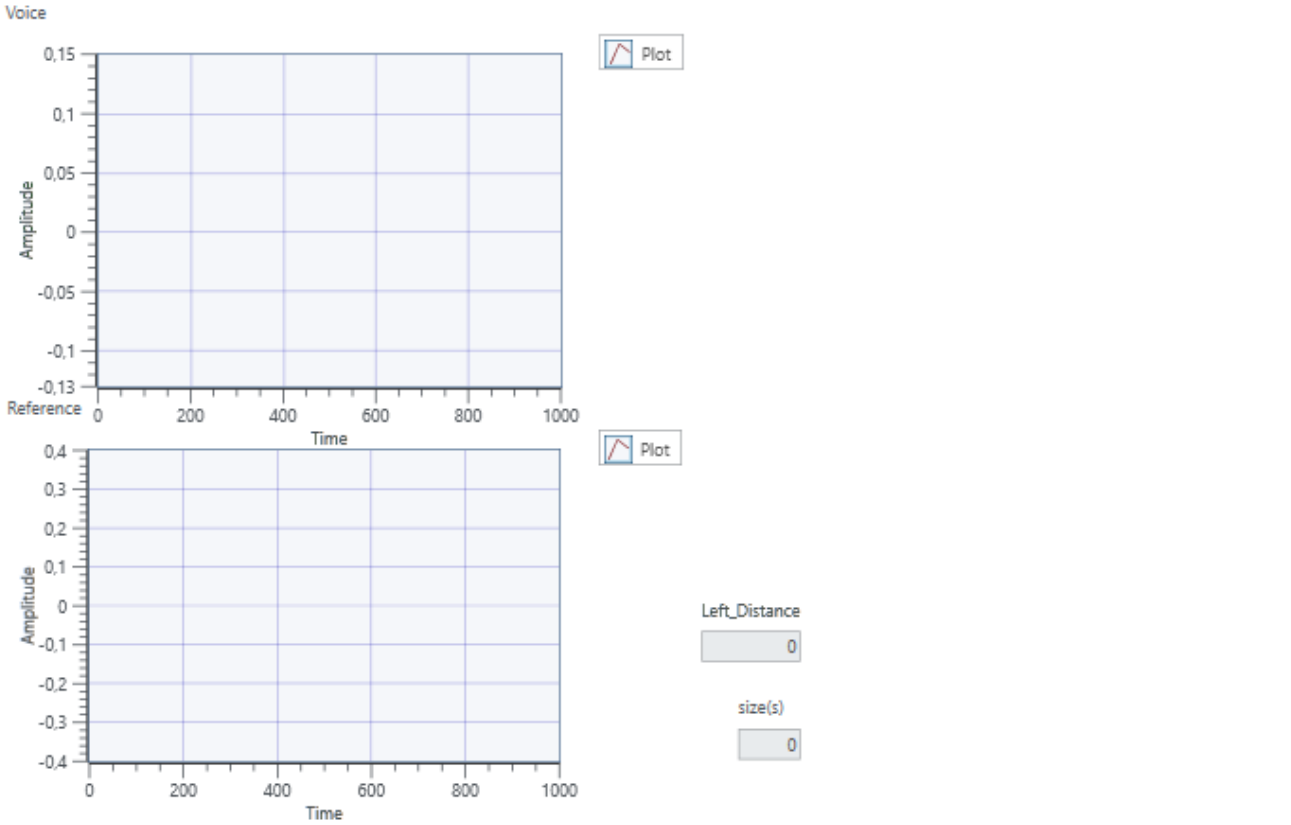


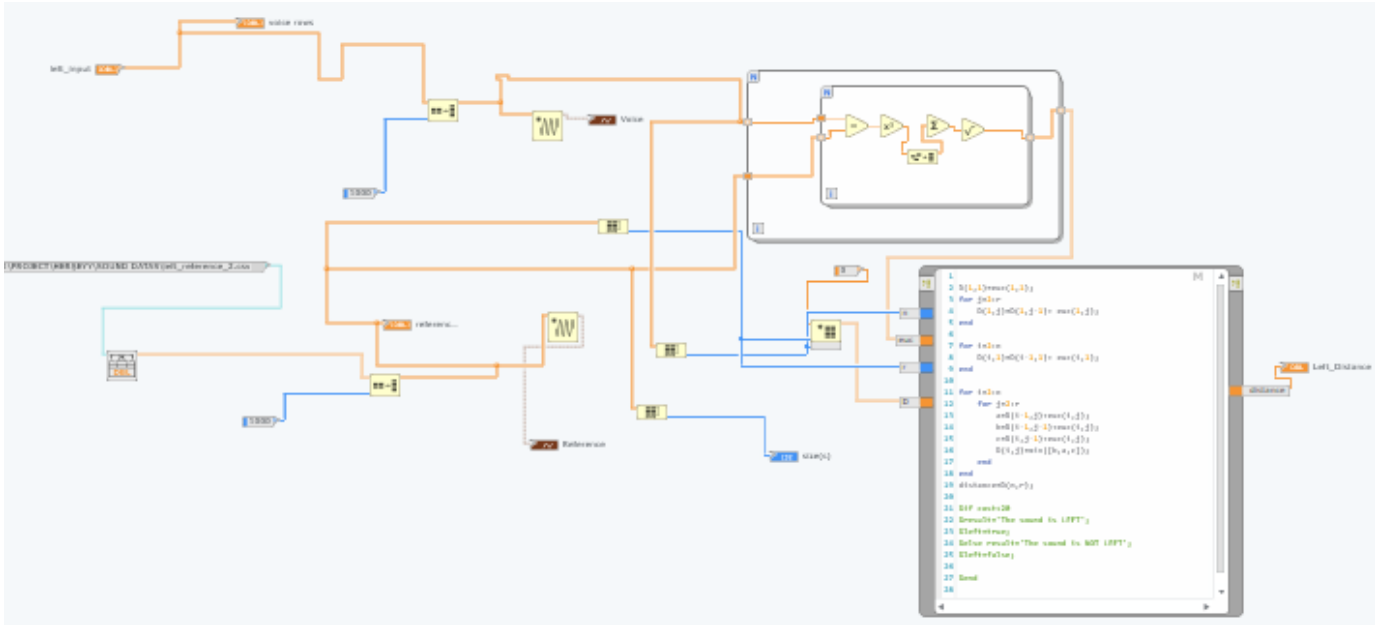

- After that for voice recognition, DTW algorithm is done.

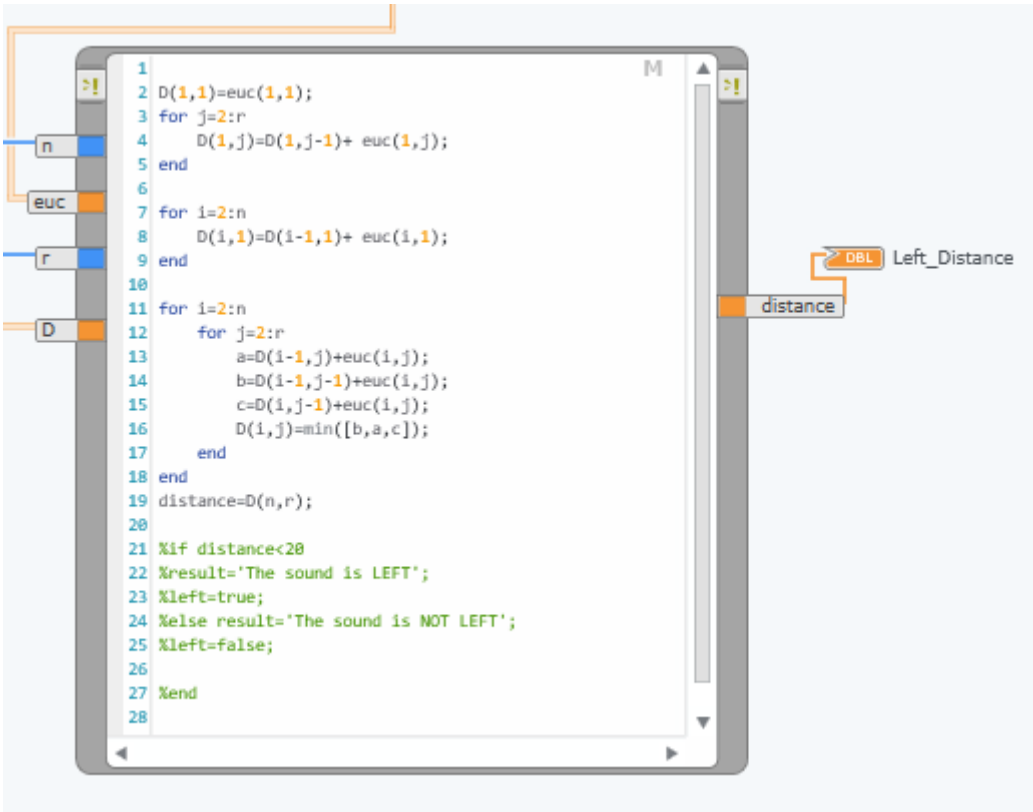
Sound detection detects "LEFT", "right", "forward" and "backward" commands. For this, separate subVIs were made to detect each command.

The block diagram of the subVIs are the same. Only reference voices are different. That’s why, subVI for the Left detection is given.



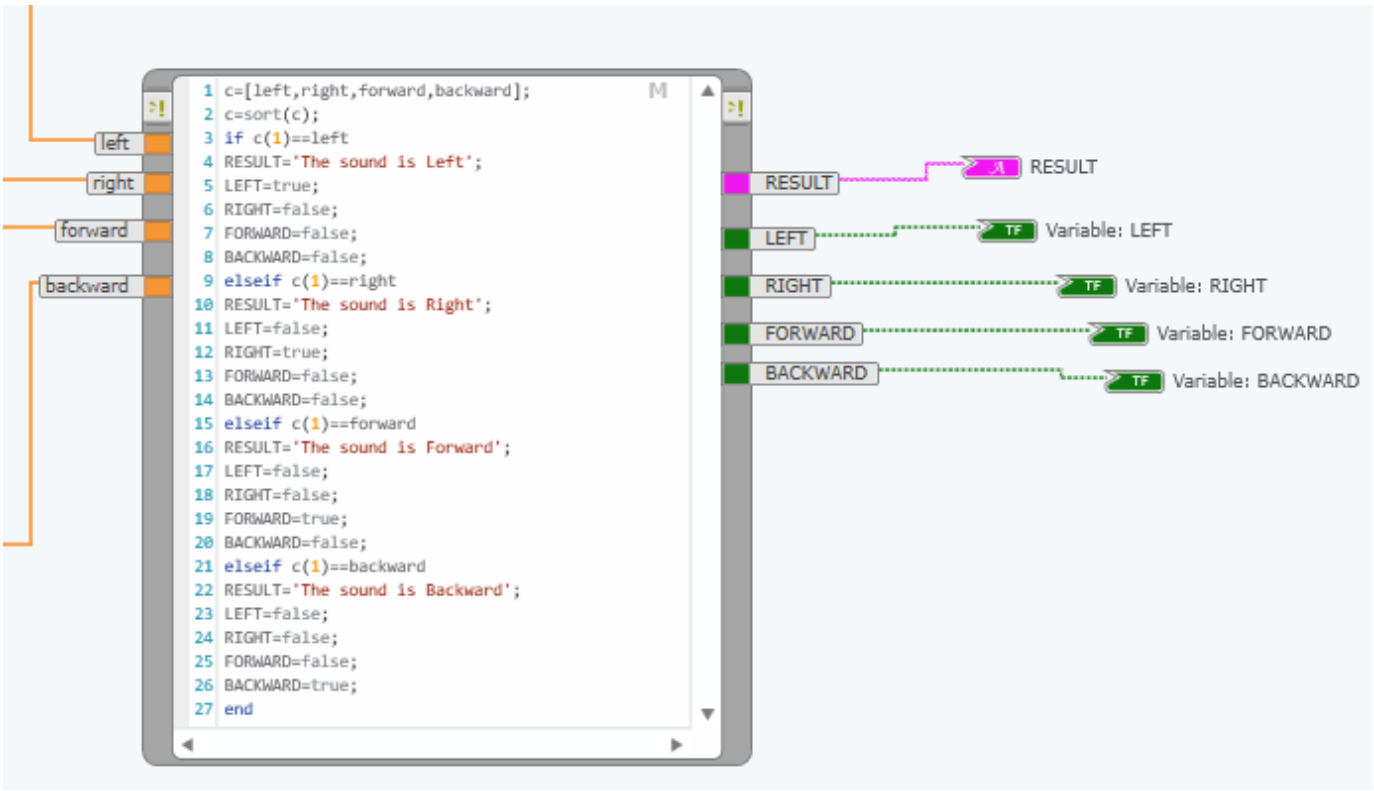

Voice
Plot
0,15
0,1
0,05
0
-0,05
-0,1
-0,13
Amplitude
Reference
0
200
400
600
800
1000
Time
Plot
0,4
0,3
0,2
0,1
0
-0,1
-0,2
-0,3
-0,4
Amplitude
0
200
400
600
800
1000
Time
Left_Distance
0
size(s)
0
voice rows
0
reference_rows
0



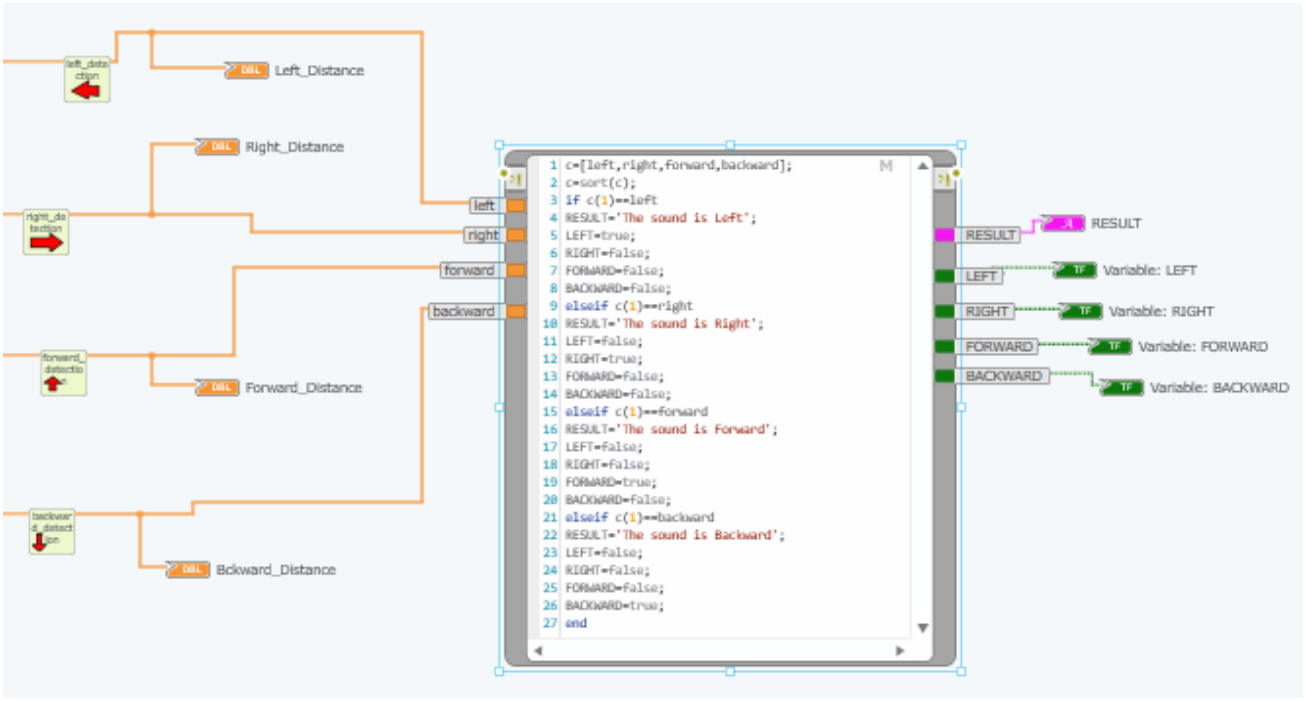



```
D(1,1)=euc(1,1);
for j=2:r
    D(1,j)=D(1,j-1)+ euc(1,j);
end

for i=2:n
    D(i,1)=D(i-1,1)+ euc(i,1);
end

for i=2:n
    for j=2:r
        a=D(i-1,j)+euc(i,j);
        b=D(i-1,j-1)+euc(i,j);
        c=D(i,j-1)+euc(i,j);
        D(i,j)=min([b,a,c]);
    end
end
distance=D(n,r);

%if distance<20
%result='The sound is LEFT';
%left=true;
%else result='The sound is NOT LEFT';
%left=false;

%end
```

- For command recognition subVIs give distances. So the algorithm that is written in math script finds the minimum distance which means which command that our sound represents.

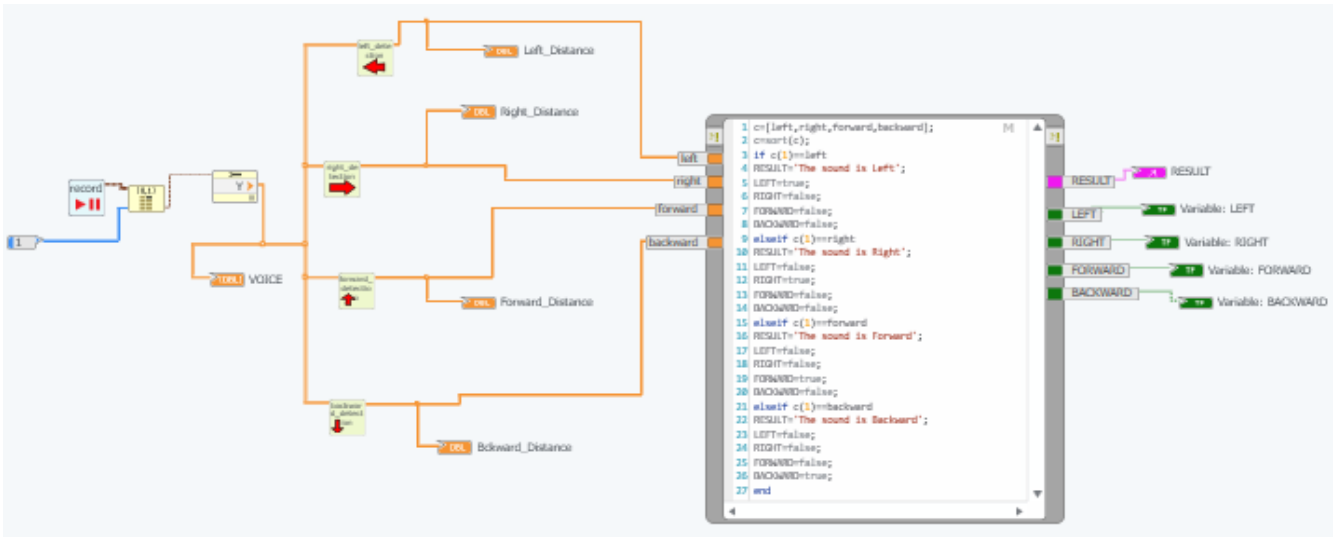



```
c=[left,right,forward,backward];
c=sort(c);
if c(1)==left
RESULT='The sound is Left';
LEFT=true;
RIGHT=false;
FORWARD=false;
BACKWARD=false;
elseif c(1)==right
RESULT='The sound is Right';
LEFT=false;
RIGHT=true;
FORWARD=false;
BACKWARD=false;
elseif c(1)==forward
RESULT='The sound is Forward';
LEFT=false;
RIGHT=false;
FORWARD=true;
BACKWARD=false;
elseif c(1)==backward
RESULT='The sound is Backward';
LEFT=false;
RIGHT=false;
FORWARD=false;
BACKWARD=true;
end
```

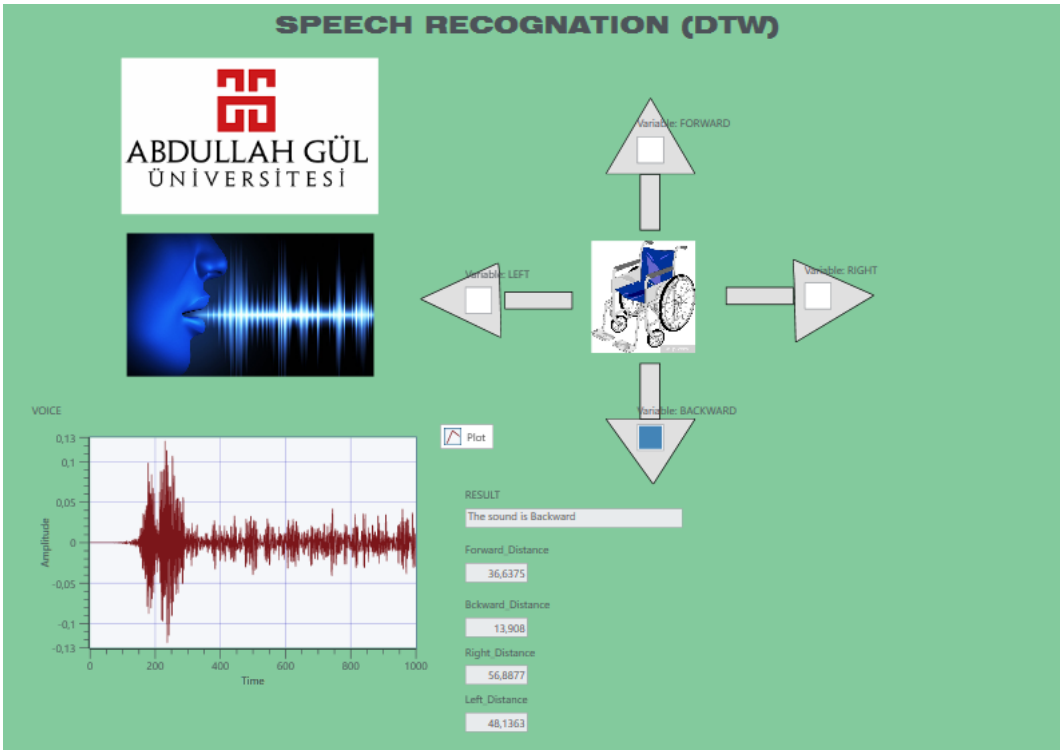



As can be seen, Boolean outputs are obtained to turn led and distances are given.

The whole block diagram of the DTW part are below.



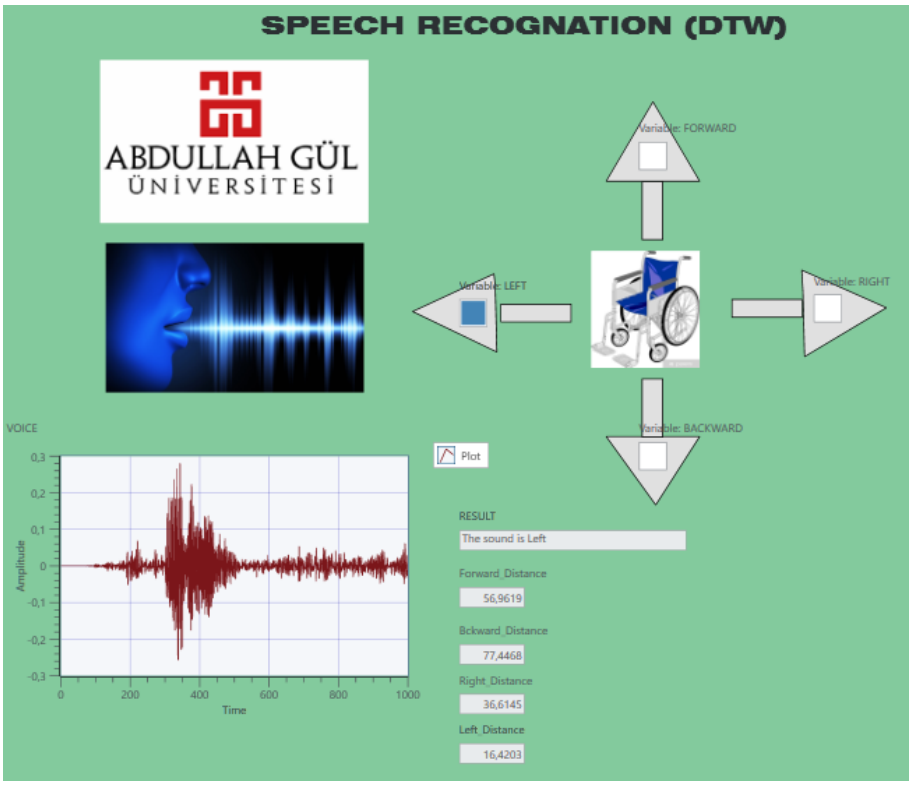
The front panel of the DTW part is below.



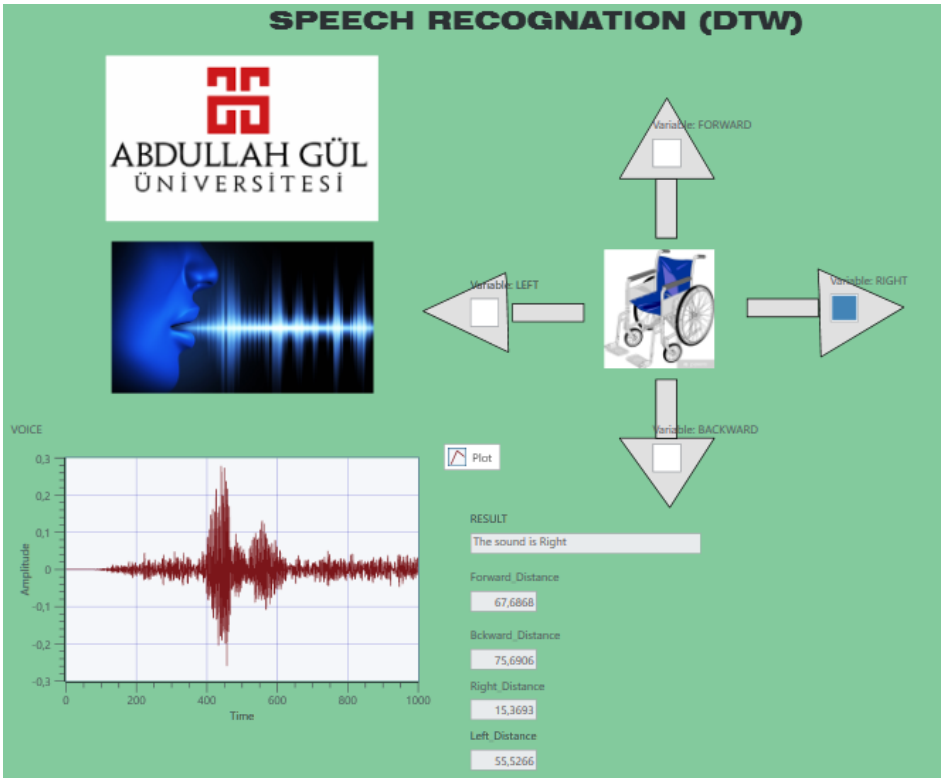

**RESULTS**

Tests were made to see if the DTW system was working properly. The results are below.

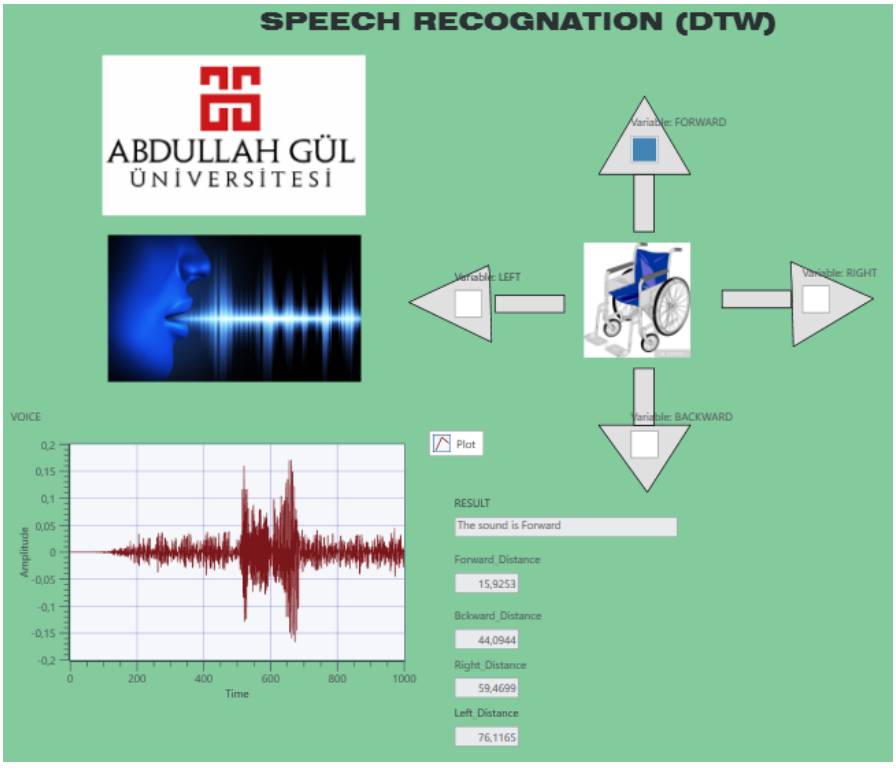
Firstly, the users say “LEFT”



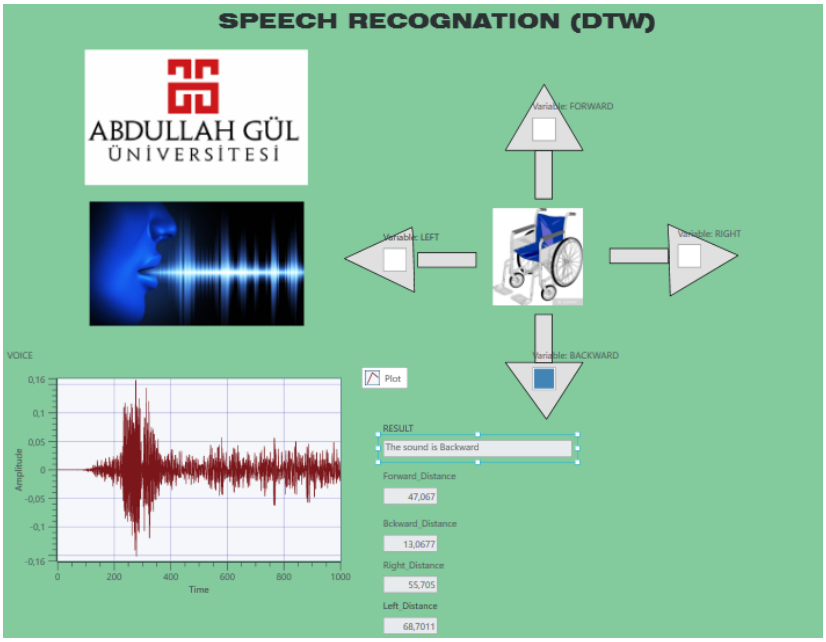
For “RIGHT’




For “FORWARD”



For “BACKWARD”




**CONCLUSION**

As a conclusion, the DTW system we have developed for sound recognition is working properly in line with the mission we have chosen for the DSP part of the Project. In the next stages, we plan to get sharper results and make our system work much faster.