SGN-16006, IMAGE PROCESSING ASSIGNMENT

Jonas Nikula, Vili Saura

240497, 240264
jonas.nikula@student.tut.fi,vili.saura@student.tut.fi

1. INTRODUCTION

In this assignment we implemented two methods to detect the DTMF frequencies. We used National Instruments myRIO-1900 and LabVIEW coding environment. Both methods calculated the value of certain frequency in the input signal. This value is compared to threshold, and if the value is greater than the threshold, a correspondig LED in myRIO-1900 will light.

2. BACKGROUND THEORY

In telephone networks the dialed number is transmitted with DTMF signals. Every number is a result of two frequencies, as seen in figure 1. We used two algorithms that detected the vertical frequencies and

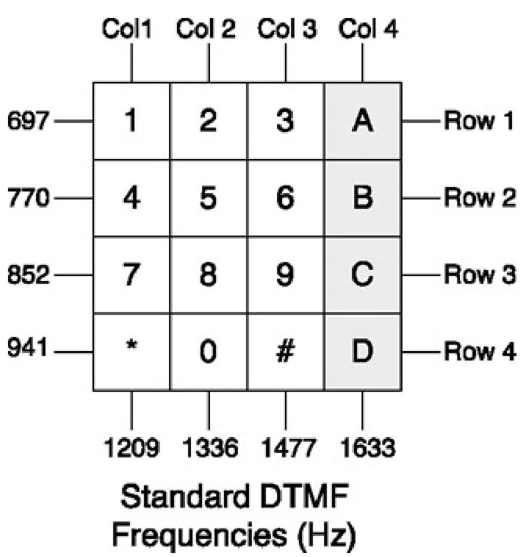

Fig. 1: DTMF frequencies [1]

3. METHODS

Two different methods were used in this assignment, Correlation detection and Goertzel -algorithm. In the first method we calculated the correlation between the input data and and the signals 1 and 2.

$$c_{697}(n) = \cos(2\pi \times 697 \times n/F_n) \tag{1}$$

$$s_{697}(n) = \sin(2\pi \times 697 \times n/F_n) \tag{2}$$

The correlations are defined by equations 3 and 4.

$$c_{\cos}(697) = \sum_{n=0}^{N-1} c_{697}(n) \times x(n) \tag{3}$$

$$c_{\sin}(697) = \sum_{n=0}^{N-1} s_{697}(n) \times x(n) \tag{4}$$

Next we estimated the signal power using equation 5.

$$var = \sum_{n=0}^{N-1} x(n)^2 \tag{5}$$

Finally, we tested the presence of the frequency using equation 6 In this case, we tested the presence of the 697Hz component. By replacing the 697 value in the equations, we can test the presence of the other components.

$$w_{697} = \frac{c_{cos}^2(697) + c_{sin}^2(697)}{var} \tag{6}$$

The resulting LabVIEW can be seen in figure 2

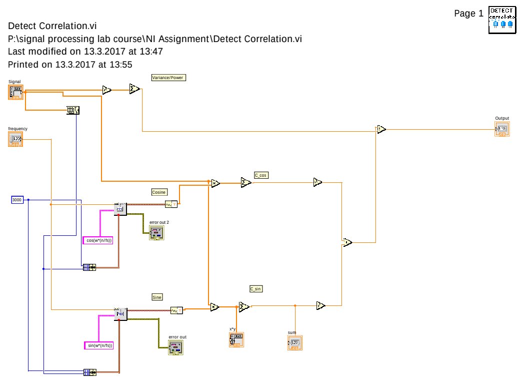

Fig. 2: LabVIEW code for Detect Correlation

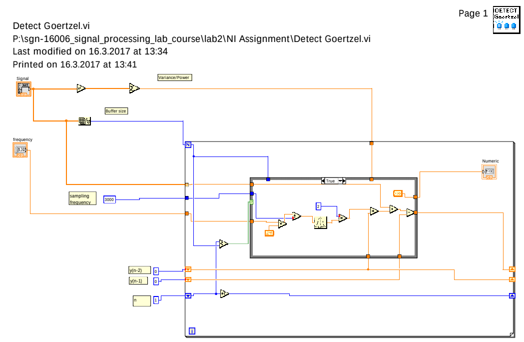

Fig. 3: Goertzel-algortihm in LabView when n < N

The Goertzel-algorithm uses an unstable IIR filter that amplifies the selected frequencies. Detecting the frequency f with sampling frequency F_s can be done using Goetzel filter 8.

$$y(n) = x(n) + 2\cos(2\pi f/F_s)y(n-1) - y(n-2) \quad (7)$$

$$y(n) = x(n) + 2\cos(2\pi f/F_s)y(n-1) - y(n-2) \quad (8)$$

However, we cannot use this equation as a direct indication on which LED should light. Instead, we can use equation 9.

$$|Y(e^{(i\omega)}|^2 = y^2(N-1) + y^2(N-1) - 2\cos(2\pi f/F_s)y(N-2)y(N-1) \quad (9)$$

The Goertzel algorithm itself consists of three steps. We loop through all the steps, increasing the value of n with every step. Step 1. we initialize n , y(n-1) and y(n-2) as 0. Then in step 2, if n ¡ N we calculate y(n) using 8, increment n = n+1 and return to step 2.In step 3, if n is equal to N, we calculate 9. and divide it with variance, using equation 5. This value is compared to the thresholdn and if it is greater than the threshold, LED is turned on, otherwise it's turned off. Again, we increment n=n+1 and return to step 1

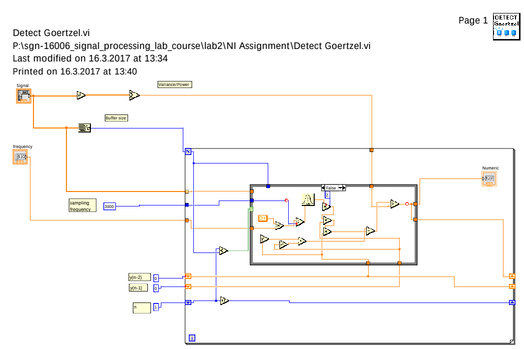

Fig. 4: Goertzel-algortihm in LabView when n = N

4. RESULTS

Using both correlation detection and Goertzel-algorithm we calculated the values of the frequencies in the input signal. Then, we compared these values to a threshold. If the value is greater thn this threshold, a corresponding LED will light. We used the following MATLAB code to test the thresholds. The first part of the code goes through alla the frequencies, lighting one LED at a time. The others test on one or two frequencies.

```
t = 0:1/2000:5;
x = chirp(t, 0, 5, 1000);
soundsc(x, 2000);

%%
fs = 8192;
f = 697; % This is the studied frequency
t = 0:1/fs:5;
x = sin(2*pi*t*f);
soundsc(x, fs);

%%
f = 697; % This is the studied frequency
n = 1:8192;
x = sin(2*pi*n*f / 8192);
soundsc(x);

%%
fs = 8192;
f1 = 770;
f2 = 1336;
t = 0:1/fs:5;
x = sin(2*pi*t*f1)+sin(2*pi*t*f2);
soundsc(x, fs);

%%
fs = 8192;
f1 = 852;
f2 = 1209;
t = 0:1/fs:5;
x = sin(2*pi*t*f1)+sin(2*pi*t*f2);
soundsc(x, fs);
```

Using these tests we were able to pinpoint a threshold which would be able to distinguish the different frequencies. With sampling frequency of 3000HZ and period of period 333s the threshold is 8 in both methods.

5. CONCLUSIONS

With correct parameters the threshold can be found experimentally. At first, we tested correlation detection with sampling frequency of 44100Hz and period of 10s. These parameters worked with the first part of the Matlab code, resulting in threshold of 90. However, this threshold wouldn't work with the other Matlab tests. Then we changed the values to 3000HZ and 333s and all test worked with threshold of 8. Goertzel-algorithm passed the tests with same values, but we had to start indexing n from 1.

REFERENCES

[1] J. R. Smith, "Chapter 17: Dtmf tone decoding and telephone interface," http://www.globalspec.com/reference/24976/203279/chapter-17-dtmf-tone-decoding-and-telephone-interface, accessed: 2017-03-17.