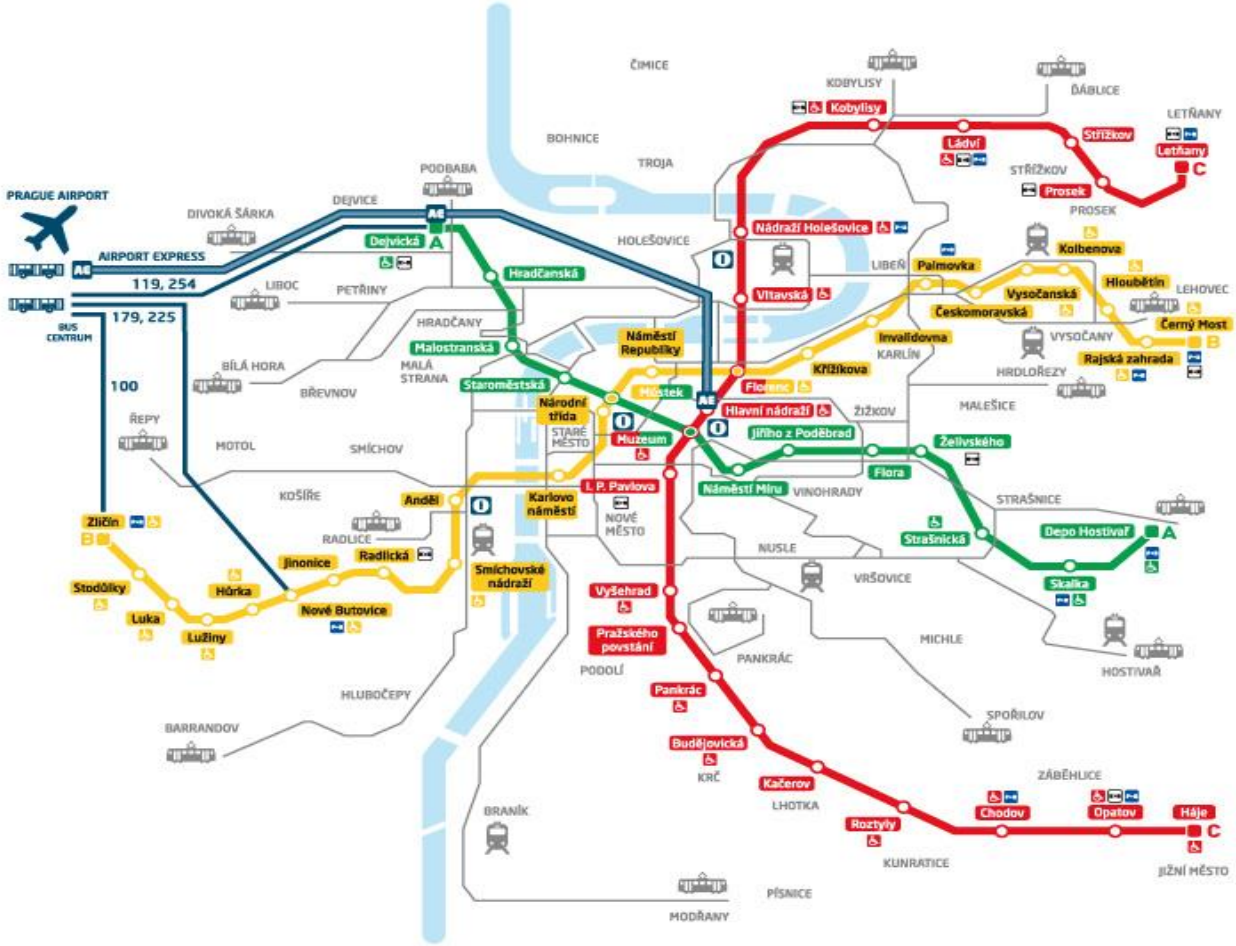
SYNTH CHALLENGE 2017

AMELLE BENSADI-SANIAL

TASK1 : FLIGHT OF THE BUMBLEBEE

TASK2 : TWO OCTAVES WITH CLARINET

TASK3 : PRAGUE METRO SOUND CREATION



I. Introduction

For this challenge, I've realised the three tasks I've been asked to do in Matlab. I've been able to complete them. They are the following ones:

1) Synthesis of the arbitrary instrument for the composition „**Flight of the Bumblebee** "by *Nikolaj Rimskij Korsakov*.

2) Two octaves of the major scale for one chosen instrument from the previous synthesis.

3) Arbitrary realization of audio signals in MATLAB.

I've found that some of the tasks were harder than others This report explain and describe how I managed to complete this synth challenge beginning in the order from task 1 to 3.

II. Task 1: Flight of the Bumblebee with clarinet

For this first task, I've decided to make the synthesis of « Flight of the Bumblebee » by a clarinet.

First of all, I have analysed a downloaded clarinet note to have the amplitude and the frequency of harmonics. I used the function « analysis » to plot the spectrum of my clarinet note. This function is a faster way to compute the Fourier Transform of an audio signal.

```
function analysis(file_name)
%function analysis plots amplitude spectrum
%file_name contains file name in format name.wav

[sig,fs]=audioread(file_name);
N=length(sig);
c=fft(sig)/N;
A=2*abs(c(2:floor(N/2)));
f = (1:floor(N/2)-1)*fs/N;
plot(f,A)

end
```

Fig.1 function « Analysis »

Amplitude

Fig.2 Spectrum of the downloaded clarinet sound (code « analysduson.m »)

Thanks to that I have created a vector of amplitude coefficients normalized by the amplitude of the fundamental frequency.

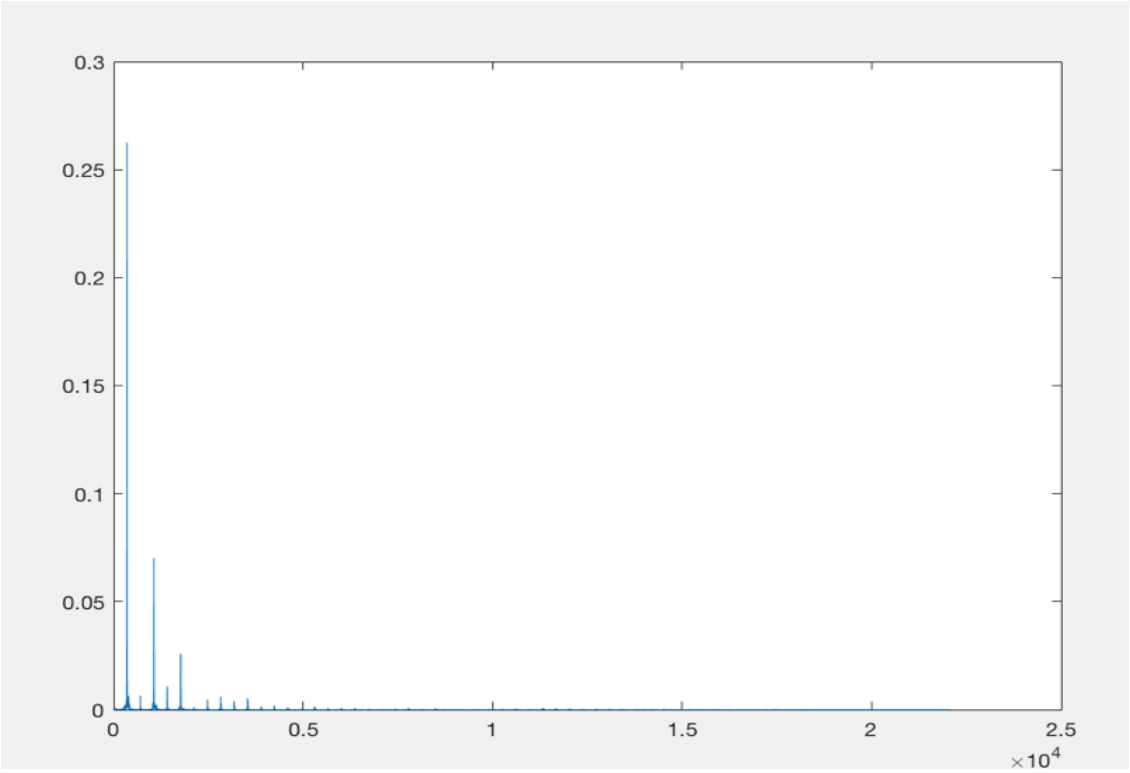
Then, I have created the sound of my clarinet thanks to these coefficients, and to a sinus function inside a « for » circle. This loop's aim is to implement the fundamental frequency by the number of harmonics. This index corresponds to the number of values in the amplitude coefficient vector. I have also added inside the sinus function another sinus which purpose is to create a vibrato. For that, I needed a sinus with a low value of frequency to have small vibrations of the air which correspond to the vibrato.

The file "synthese_clarinette.m" can be compute and we can hear the correct synthesis of the sound. This is the part that I have add to the function "synth.m" to make it work with the clarinet.

Frequency (Hz)

Also, since in the Matlab table of instruments, clarinet is number 72, I had to modified the maximal number of instrument which was 64. I putted 73.

Unfortunately, didn't use channels because I only used one instrument but I



tuned it.

Finally, I created the envelope of the signal. For that I used the function « interp1 » which allows me to choose the position of my envelope by positioning the points where I wanted.

After putting all my code in the « synth » code I obtained the synthesis of the « Flight of the Bumblebee » by my clarinet.

I could after use the code “Bumble01after.m” using the function “synthchallenge” to create an audio file that you can find in the folder “result”.

III. Task 2: Two octaves with Clarinet

For this second task I had to make the synthesis of two octaves of the major scale with the clarinet I used for the previous task.

First of all, I made a research on the internet to find all the frequencies of the major scale with the aim of write them down in my code as a vector.

Then, the goal was to play all the notes one by one and not in the same time. This is why, I've decided to create a « for » circle in which I can call the synth function on every loop and I put the note played in a vector with all the previous notes inside.

```
y=[];
for i=1:length(freq)
    f=freq(i);
    x=synth(f,dur,amp,fs,synthtype);
    y=[y x];
end
```

Fig.3 For circle that enables us to play tone after tone

Thanks to that all the notes are played one after one.

I've finally used the function audiowrite to create an audio file with the major scale.

You can find it under the name « Major-Scale .m4a»

IV. Task 3: Prague metro sound creation

This task inspired me the most. I've chosen to create the sound of the Prague's metro because I think it's the sound/noise I hear the most every day. Each day of the week I'm coming to Dejvickà from Můstek with the metro and then vice-versa. So I recorded the bell ringing, the announcements and also the noise of the metro itself when it's resting, closing doors and running.

First of all, I've separated my records into small records because it was easier to compute the data with smaller samples. So I finally end up with five records to exploit:

1 – Metro Bell (name: Metro-Bell.m4a)

2 – Announcement: « Ukončete prosím výstup a nástup, dveře se zavírají » (name: Prosim.m4a)

3 – Announcement: « Příští stanice – Můstek » (name: Stanice-Mustek.m4a)

4 – Metro running (name: Metro-Running.m4a)

5 – Announcement: « Můstek » (name: Mustek-Arrival.m4a)

Second of all, I've found the frequencies of the ringing bell. It's something I've been able to find with the FFT of the signal (using the "analysis" function). Then I've chosen the coefficient that correspond to a bell. Thanks to this, I succeeded to reproduce this sound and play it on Matlab.

Third, I kept my records number 2, 3 and 5 like they are. It was completely impossible for me to reproduce the voice speaking with Matlab. So I only call them in my code.

Finally, I tried to reproduce the noise of the metro. It was the most difficult part because first, I've tried to do it with only a random white noise. But it didn't sound like the metro. So I decided to look at the spectrum of my record of the metro running. I've been able to find several frequencies that

were the ones I heard the most when I was listening to the record. Then I've made an additive synthesis of those frequencies. It's was kind of nice sound but something was missing. The noise of the metro is « windy ». Since it's running in a closed space you can hear the pressure of the air inside and outside of the wagon. So I've decided to add my frequencies to a « wind sound ». As we saw in class, I've reproduce the sound of the wind and again made an additive synthesis of the frequencies and the wind. It was nice and kind of sounded like the metro. But again, still not satisfying.

I needed a sound that had an increasing amplitude at the beginning and decreasing amplitude at the end. It's logic because the metro has to set off and stop before and after running at full power! So I created an envelope with the function "interp1". I've chosen different amplitudes for different times positions and then I was able to add this envelope to my sound. It's was a long part because I had to try a lot of different amplitudes and times and listen every time how it sounded, but at the end I was more satisfy when I listened to it than before.

Then, I had then all my sounds like I wanted. It was the time to put them together. So I normalized them and try to sum them but it was not enough. Thanks to some help from M. Novotny. I've succeeded to do it. Actually, I've defined a duration of my final sound. I've chosen 50 seconds so it was enough for sure. Then I took each of my five sounds (actually there are 6 because the bell is coming two times) and I filled them with zeros where there was no music playing. Thanks to that, they were all the same size.

Finally, I had to find where to make each sound begin and to know where it ends so I could make another one begin and so on, with the aim of putting them in order correctly. So I created vectors "position" and I tried a lot of combinations until I finally found one that was satisfying! At the end I only had to sum them and put them in one big vector named "finalsound".

```
% Position of the begining of the sounds of the announcements
position3=2.15;
position4=8.25;

[y1,fs] = audioread('Prosim.m4a');
y1=y1./max(abs(y1));
y1=y1';
y1s=zeros(1,samples);
position3=round(position3*fs);
y1s(position3:position3+length(y1)-1)=y1;

[y2,fs] = audioread('Stanice-Mustek.m4a');
y2=y2./max(abs(y2));
y2=y2';
y2s=zeros(1,samples);
position4=round(position4*fs);
y2s(position4:position4+length(y2)-1)=y2;
```

Fig.4 Example of how I've normalised, zero padding and position the sounds

You can hear the result under the name "MetroMustek.m4a" or read and compute it under "MetroMustek.m"

But still, I am not completely satisfied with my creation. There are some things I would have had modified or done if I had the time and skills to do it! Like for instance I would have liked to create the sound of the door closing but unfortunately I was not able to make it or I would have make a better combination of frequencies with more of them to add to my wind noise…

I can say that it was a really interesting task and I had pleasure to make it even if sometimes I spent a lot of time on it!

V. Conclusion

To conclude, I would like to say that I've found this challenge really interesting and it made me understand even more Matlab and what we can do with it. I've tried a lot of things and spent a lot of time sometimes to understand functions and also it made me train the things we saw together in class.

Even if I enjoyed less the first two tasks, it was interesting in a technical point of view. The last one was more for me a way to express something, more in an « artistic way » and to be honest it was my favourite task.

VI. References

- http://sami.fel.cvut.cz/sms_eng/

- http://www.universal-soundbank.com/clarinette.htm

- https://www.pinterest.co.uk/pin/553168766707135561/ (Picture of Prague metro map)