

MUMT-307 Report: Using FFT analysis and resynthesis to create infinite sustained sounds and spectral reverb.

Eduardo Aparecido Lopes Meneses
Schulich School of Music – McGill University
Montreal, Canada
(438) 927-8491
Email: eduardo.meneses@mail.mcgill.ca
Site: <http://www.edumeneses.com>

I. INTRODUCTION

One of my research interests is the modification of the sound characteristics of the classical guitar. This instrument features rich timbre but its notes have short sustain.

The use of Fast Fourier Transform (FFT) to analyze the sound of the instrument can allow the synthesis of sounds that have a similar pitch to that of the classic guitar, but which can be synthesized indefinitely.

II. A QUICK SUMMARY OF OBJECTIVES

In summary, the objectives of this final project are:

- The analysis of classical guitar sound in real time;
- Manipulation or processing performed in the frequency domain in order to generate either a sound of infinite sustain and a spectral delay;
- Sound resynthesis using the inverse FFT process (IFFT).

III. THE PROJECT DESIGN

Although my research points to the possibility of achieving the goals of this project using either granular synthesis or delay lines (in a complex and possibly unstable configuration), the frequency domain sound manipulation allows me to further expand the functionality of the programmed patch, including pitch Control and time stretching.

For this project I opted for programming in Pure Data¹ because this language allows me to execute the programmed patch on computers running operating systems based on Linux. This feature will allow me to use the patch on embedded systems based on Raspberry Pi², for example.

The patch contains 2 operation modes: a continuous analysis and resynthesis mode (spectral delay) and a manual triggering mode (freeze). The spectral delay mode will analyze blocks in a given time interval that can be set by the user and the output will respond according to the real time input intensity after N samples. The freeze mode will synthesize a continuous sound

¹<https://puredata.info/>.

²<https://www.raspberrypi.org/>.

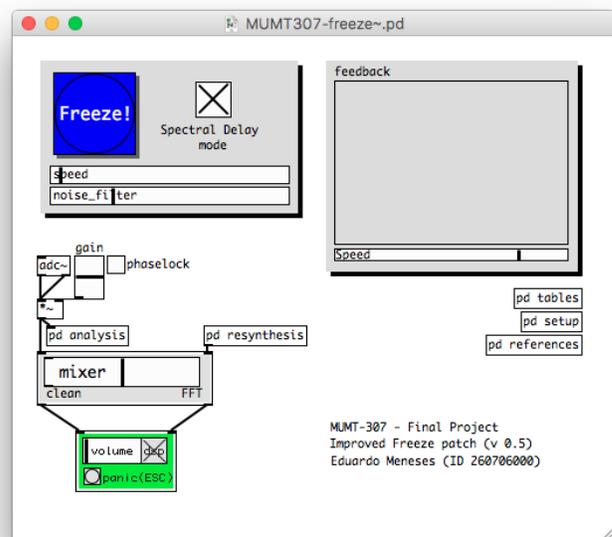


Fig. 1. Main patch window.

from the blocks acquired immediately after the user activates the function, i.e. push the freeze button.

The programming includes several concepts learned during the course MUMT-307 (Music & Audio Computing II): Signal Analysis using STFT and resynthesis using IFFT, digital delay lines, feedback (in filters and frequency domain processing), windowing, FFT "hop" and "Phase", overlapping, convolution, etc.

IV. PARTICULAR ISSUES THAT PROVED CHALLENGING / REWARDING

Programming in Pure Data presents some extra challenges in relation to Max/MSP³: The lack of objects in Pure Data

³<https://cycling74.com/products/max/>.

makes it necessary to use multiple objects to perform functions that are easily solved with only one object in Max/MSP. An interesting example is the counter object. The Vanilla version of the PD does not have any counter object, being necessary to the use of the metro object and the creation of accumulators using mathematical operators with storage objects.

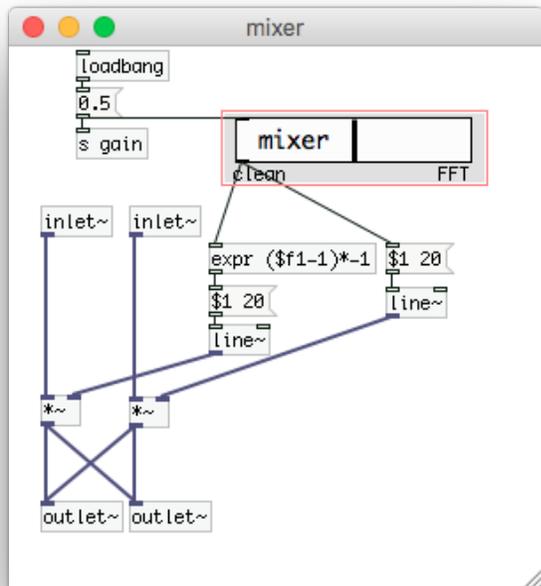


Fig. 2. Subpatch acting as a mixing desk (depth control).

This problem can be solved by using the PD-Extended version or by using other distributions such as Pd-L2Ork⁴ (Purr Data). The use of PD-Extended is not the ideal solution since it is abandoned. Pd-L2Ork is the best option to use external and libraries that leave PD functionality similar to Max/MSP.

One last important comment about choosing which version of PD to use is the issue of compatibility: PD-Vanilla is still a good software choice if the intention is to make the programmed patch available in all operating systems and devices, including mobile phones.

All the references provided great information to build the patch. Some examples of different approaches for spectral freeze and spectral delay were found among Pure Data and Max/MSP community. Some of these patches provided good ideas and were implemented in the resulting patch, including visual feedback and “speed control” (Ralf’s Freeze3.pd) and practical tips for implementing procedures (Jean-francois Charles paper and patches in Max/MSP).

Unfortunately most implementations found in both the PD and Max/MSP communities have conceptual errors or present very simple implementations that do not provide sound quality or different configuration possibilities.

Even using the different procedures studied during the discipline, the resulting sound quality can vary drastically according to the sound source used and the signal-to-noise ratio. The patch proved itself moderately robust for sounds with concentrated energy in the medium-high regions of the spectra, such as trumpet’s high register or a whistle.

A mixing object has been programmed to control the relationship between the original sound and its synthesized version (depth control). This patch can be checked in figure 2.

For the low region of the spectrum (more present in classical guitar spectra) the noise level in the synthesized sound is considerably higher and greater attenuation of the more energetic frequency peaks is required. This attenuation is provided in the form of a noise filter control, which provides destructive interference to the signal to eliminate excess noise.

V. EXAMPLE USAGE / DEMO

Both the patch and a web version of this document can be found in <http://edumeneses.com/en/2017/04/09/mumt-307-report-fall-2017/>.

VI. REFERENCES

Arfib, D. and Keiler, F. and Zölzer U. and Verfaillé, V. and Bonada, J. DAFX: Digital Audio Effects, Second Edition, Chapter 7: Time-Frequency Processing. DOI: 10.1002/9781119991298.ch7.

Charles, Jean-francois. A Tutorial on Spectral Sound Processing Using Max/MSP and Jitter. Computer Music Journal, Fall 2008, Vol. 32, No. 3, Pages: 87-102. DOI:10.1162/comj.2008.32.3.87.

Charles, Jean-francois. Max/MSP patches share: "3-freeze-slide", "4-freeze-tail" and "5-freeze-denoiser". Available in <https://cycling74.com/share.html>.

Dolson, Mark. The Phase Vocoder: A Tutorial. Computer Music Journal, Vol. 10, No. 4 (Winter, 1986), pp. 14-27

Guy John (rumblesan). Patch-A-Day-month-2010-11. Available in <https://github.com/rumblesan/Patch-A-Day-month-2010-11/tree/master/27-FFTPitchShiftandFreeze>.

Kreidler, Johannes. PD Tutorial website. Available in <http://www.pd-tutorial.com>.

McClellan, James and Schafer, Ronald and Yoder, Mark. DSP First: A Multimedia Approach. Prentice Hall; 2nd edition. ISBN-10: 1.32432e+08

Ralf's freeze3.pd (2011). Available in <http://forum.pdpatchrepo.info/>.

Scavone, Gary. MUMT 307: Music & Audio Computing II: Frequency-Domain Processing & Resynthesis. Available in <http://www.music.mcgill.ca/~gary/307/week11/index.html>.

⁴<http://l2ork.music.vt.edu/main/make-your-own-l2ork/software/>.