AN EXAMPLE OF DESIGN OF CHAINS OF DELAY LINES IN FAUST LANGUAGE FOR PEDAGOGICAL AND CREATION PURPOSES IN MIXED MUSIC

Alain Bonardi

CICM – Laboratoire MUSIDANSE
University of Paris 8, Saint-Denis, France

alain.bonardi@univ-paris8.fr

ABSTRACT

This paper presents several applications of chains of delay lines designed in Faust language for mixed music, either in pedagogical or in creation contexts. The experiments we led in research and creation show the expressive potential of this idea, when articulated in an object-oriented approach with the constitution of classes enabling various musical behaviours to be explored.

1. INTRODUCTION

For several years we have been experimenting with the design of the electronic part of our mixed music pieces according to an object-oriented approach. This means building templates of sound processes, considered as classes that are instantiated in different ways to provide different musical behaviours and interesting features on a compositional point of view. The idea is to musically design the DSP part of the live electronics using such object-oriented paradigms as classes, encapsulation, and polymorphism.

The object approach has been intensively used in sound and music computing for many years, starting from languages themselves for instance with *SuperCollider*, developed by James Mc Cartney (first version in 1996 [1]). Its syntax is globally close to C structures. The object paradigm has also contributed to new models of music representation [2]: among numerous proposals, let us mention the MODE/SMOKE environment in SmallTalk for composition, performance and analysis by Stephen Pope [3]; FORMES environment for composition and the temporal programming of processes by Xavier Rodet and Pierre Cointe [4]; MusES environment dedicated to the representation of knowledge in tonal music [5]. More recently, OpenMusic [6] also proposes a modelling approach and enables to take into account the classes created by the users. The field of real time sound transformation has also explored the object paradigm for instance for the control of synthesizers in Lounette M. Dyer's Ph. D. [7]. Object-oriented environments have been proposed for Max or PureData like *Odot* developed at CNMAT [8].

On the other side, object-oriented approaches in electronic music can be driven by compositional purposes with or without any computerized implementation. Such composers as Horacio Vaggione in the field of electroacoustic music have been inspired by object-oriented approaches both theoretically and practically to implement their compositions by building object networks [10][11].

In a previous paper published in 2016 [9], we explored a simple but fruitful approach based on one simple sound process consisting in a delay line followed by a harmonizer. The control data of this structure were considered as the member data of a simple class including the delay duration, the value of feedback, the dispatching between delay and harmonizer, the transposition, the input and output gains. This simple class was instantiated 16 times. We then created classes above this elementary simple class, that we described as instruments. For instance, the *timbals* class required 4 elementary lines (delay + harmonizer) and proposed several methods of musical behaviour. This approach was applied to *Pianotronics 3*, a piece for piano and real time electronics commissioned by Scrime in Bordeaux.

Contrary to approaches of mixed music as for instance in Philippe Manoury's works [12], we do not start from an orchestra of different modules (harmonizers, frequency-shifters, samplers, etc.), but from a common class to be specialized to get different sound features. Inheritance has also been used to group instruments in general classes having close features. In our case, object-oriented principles drive both the compositional basis and the implementation of the electronics.

In this paper, we would like to present a variation of this principle using delay line chains of variable length implemented in Faust language as a basic class of organization of the sound processes. Though Faust is not an obvious choice for object-oriented implementation, we have chosen this language both for sound quality (more accurate and better defined than in Max or PureData) and pedagogical reasons: we put the emphasis on Faust in our computer music courses for undergraduated students at the Music Department of Paris 8 University. We also wanted to take advantage of the Kiwi[1] environment developed by our team during the MUSICOLL [13] project (2016-2019): this software enables to collaboratively and remotely create and play patches based on the same principles as in PureData or Max softwares; it embeds a Faust compiler [14], thanks to the *faust~* object that enables a semi-collaborative edition of Faust code in the framework of the collaborative environment.

We first show the interest of delay chains in mixed music composition. We then demonstrate a first simple realization used in the framework of pedagogical workshops with young musicians, based on Kiwi patches implementing Faust code. We finally explain how we used this concept in the composition of *Lire de Soi*, a new piece for octave mandolin and live electronics based on the exploration of tremolo as a model. This shows how a generic sound process can be at the basis of very different musical and social uses.

2. DELAY CHAINS FOR MIXED MUSIC

Delay lines are interesting objects for composition in mixed music. Driven by two control values, the delay duration and the feedback value, they can provide three different behaviours: echoes of the input when the delay duration remains macroscopic

[1] https://kiwi.univ-paris8.fr/

(above 50 milliseconds); reverberation around 40 milliseconds with a high level of reinjection; comb filter around 20 milliseconds also with a lot or reinjection.

Having chains of delay lines used to generate rhythmic motives is not a new compositional idea, since Pierre Boulez used it in his famous work *Répons* (first version in 1981) for 6 soloists, ensemble and live electronics. For instance, in the second section of the work, he uses a canon between the second piano part and the series of delays that both play the same rhythm.

In our case, the idea is to combine elementary delay lines in a series to create rhythmic patterns with separate outputs and individual gains so that the rhythm sounds livelier because of variations of intensity. Figure 1 shows the general idea of these delay chains, in this case with 4 elementary delay lines and 4 amplitudes. In this case, the original note is first played and then three echoes of it with various durations, and a possibility of reinjection if necessary.

Another constraint we set is to express the duration of the delay lines not in milliseconds but in terms of musical durations. It means we use a general convention stating that a quarter-note is represented by 1, an eighth-note by 0.5, a sixteenth-note by 0.25, and so on (as in Antescofo language for instance).

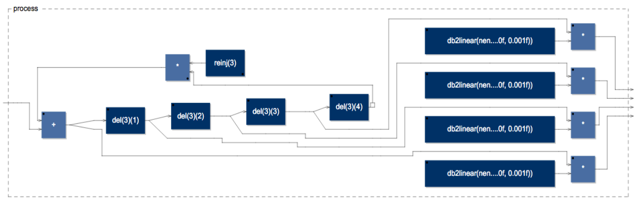
process
reinj(3)
del(3)(1)
del(3)(2)
del(3)(3)
del(3)(4)
db2linear(nen....0f, 0.001f))
db2linear(nen....0f, 0.001f))
db2linear(nen....0f, 0.001f))
db2linear(nen....0f, 0.001f))

Figure 1: *An example of a chain of 4 delay lines with 4 different amplitudes.*

There is a general tempo and the durations of the delay lines in number of samples are computed from the tempo and the sampling rate. To start with the details of the Faust implementation, we create a tempo controller:

```
tempo = nentry("tempo", 60, 1, 10000, 1);
```

and then we have a list of musical durations named *rhythm*, for instance:

```
rhythm = (0.5, 1, 0.25, 0.75);
```

This means that three echoes of the original sound will be heard: after the original sound being 'held' during an eighth-note, the first echo is played, then the second one after a quarter-note, the third one after a sixteenth-note. The cell can be repeated if the reinjection is different from 0, in this case after a dotted eighth-note.

The amplitudes of the delay line outputs can be set individually and relatively to the amplitude of the original sound. We considered mF (mezzo forte) as the reference, at 0 dB. We created 6 levels of attenuation that were given 6 labels:

- +5 dB (*fo*, *forte*, bit higher),
- 0 dB (*mf*, *mezzo forte*, identical to the original),
- -5 dB (*mp*, *mezzo piano*, a bit lower),
- -10 dB (*pi*, *piano*, lower),
- -15 dB (*pp*, *pianissimo*, much lower),
- -20 dB (*vl*, very low).

In our example, the list of amplitudes *dynamics* is coded this way:

```
dynamics = (mf, pp, pi, mp);
```

When combining this rhythm and dynamics patterns, we get the following result on an original note as shown on figure 2:

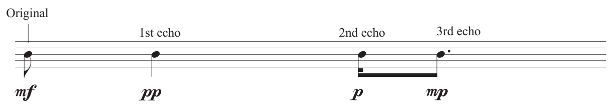

Figure 2: *The musical result produced by a pattern of 4 delay lines on an original note (B in this case).*

Delay lines are defined using the *de.sdelay* function proposed in Faust libraries, which is a smooth delay that doesn't click when the delay time is changed. They need to get durations in number of samples that are computed from the musical duration and the tempo according to the following formula:

```
dur(i) = ba.take(i, rythm) * 60 / tempo *
ma.SR;
```

Each elementary delay line is therefore defined by:

```
del(i) = de.sdelay(262144, 16, dur(i));
```

Chains of delays are defined recursively:

```
delaychain(1) = del(1);
delaychain(2) = del(1) <: (del(2), _);
delaychain(n)  =  delaychain(n-1)  :  ((_  <:
(del(n), _)), si.bus(n-2));
```

This enables to easily modify the number of delay lines in a chain, just by updating the number of values in the *rhythm* and *dynamics* lists, and recompiling the code.

3. PEDAGOGICAL USE OF DELAY CHAINS

The first use of these chains of delays was pedagogical. At the Music Department of the University of Paris 8, we have been collaborating for seven years with the Saint-Denis Conservatory for mixed music creation. During the last years, this collaboration was based on the composition of short pieces of mixed music by composers studying at Paris 8 to be created by young musicians studying in the Conservatory, with the help of their instrument professors. Every year, 5 to 7 new works of mixed music were composed by the students within 4 months and created at the end of March (figure 3, that shows the process of collaboration).

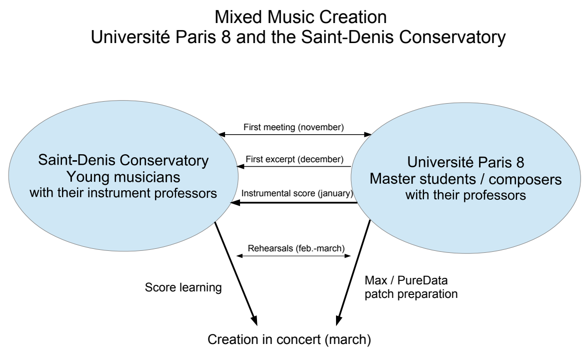

Figure 3: *The previous Mixed Music Project between the Université Paris 8 and the Saint-Denis Conservatory.*

For the academic year 2019-2020, we decided to put the emphasis on the initiation to mixed music for the professors and young musicians of the Conservatory. From 2016 to 2018, we experimented with various collaborative ways of initiation to patching for beginners, starting with workshops for high school pupils up to complete courses for the undergraduate students in the Music Department of our university.

The purpose this year (see figure 4) is to enable professors and young musicians to directly use Kiwi whereas during the previous years they did not handle the patches composed by our students in Paris 8. Therefore a first training session was organized for the instrument professors in November 2019 to get familiar with the main concepts of patching in Kiwi. In parallel, we exceptionally started our teaching of Faust language at the university at the first semester to enable undergraduate students to get the essential of it and be able to write small pieces of code. One of the most important exercises we dealt with during this Faust course was precisely to write code with chains of delays as described in the previous section of this article. Then 5 students of the University of Paris 8 proposed Kiwi / Faust patches (most of them based on the chain of delay lines) to the professors of the Saint-Denis Conservatory, who then showed them to their pupils. These patches can be run either on the professors' computers or even at home by the young musicians when they can access a computer. A first workshop gathered all people taking part in the project at the end of January 2020.

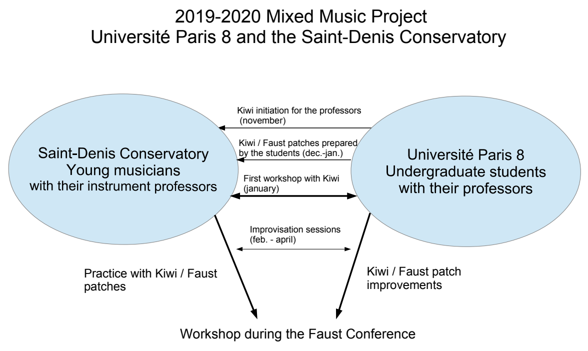
2019-2020 Mixed Music Project
Université Paris 8 and the Saint-Denis Conservatory
Saint-Denis Conservatory
Young musicians
with their instrument professors
Université Paris 8
Undergraduate students
with their professors
Kiwi initiation for the professors (november)
Kiwi / Faust patches prepared by the students (dec.-jan.)
First workshop with Kiwi (january)
Improvisation sessions (feb. - april)
Practice with Kiwi / Faust patches
Kiwi / Faust patch improvements
Workshop during the Faust Conference

Figure 4: *The ongoing Mixed Music Project for 2019-2020.*

The pattern of chains of delay lines was very easily understood by the various musicians who used the patches since it can be easily interpreted as a rhythmic motive than can be easily modified. With the *faust~* object in Kiwi, it is easy to edit the Faust code, and to modify the durations and dynamics of the pattern. Also by changing the tempo, the various aspects of delay lines appear: from a usual tempo that enables to play rhythmic sentences to a very high value of tempo that transforms the module into a comb filter.

4. COMPOSING MIXED MUSIC WITH SEVERAL INSTANCES OF DELAY CHAINS

This idea of chains of delay lines is the keystone of our new piece for octave mandolin and live electronics, *Lire de Soi*. From 2016 to 2018, we composed two previous pieces for guitar and live electronics written in Faust language, *Fil de Soi 1* and *Fil de Soi 2*. We now create this new piece in the framework of a research and creation process about plucked instruments and their live timbre transformation.

In *Lire de Soi*, we wanted to investigate the idea of modelling the tremolo of the octave mandolin (typical of that instrument) in the electronic part and be able to create various kinds of tremolos. Each chain of delay lines is a specific variation on the idea of tremolo, with particular durations and dynamics.

4.1. Composing an array of delay chains

We transformed the chains of delay lines first by adding a harmonizer at the entrance, to enable the transposition of the incoming note (see figure 5).

We created several patterns of this kind with various duration and dynamics profiles. We give the examples of patterns #3 and #4:

```
rythm(3)  =  (1,  0.5,  1,  0.5,  1,  0.5,  2);
rythm(4) = (1.5, 1.25, 1., 0.75, 0.75, 0.5,
1.25);
dynamics(3) = (mp, mp, mf, fo, mp, pi, mf);
dynamics(4) = (pp, pi, mp, mp, mf, mf, fo);
```

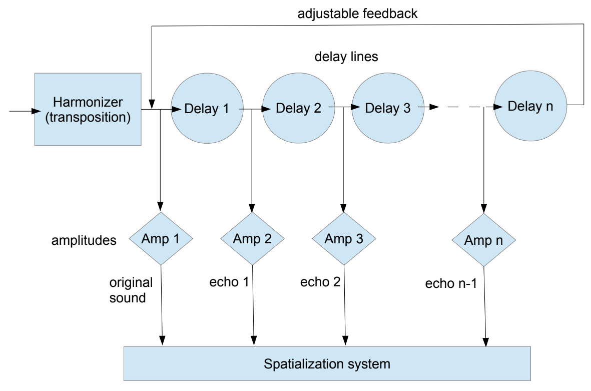

Figure 5: *The structure of a chain of delays starting with a harmonizer.*

The patterns #3 and #4 listed above correspond to the following musical cells on figure 6.

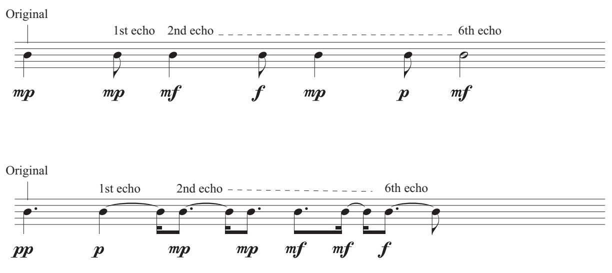

Figure 6: *Patterns #3 and #4.*

In *Lire de Soi*, the set of patterns is used to create polyrhythmic moments that are slowly modulated, changing the volume of diffusion of each or its temporal stretching.

4.2. Managing a flexible polyrhythmic structure

This set of rhythmic patterns enables us to create a polyrhythmic structure, a kind of poly-tremolo from the sounds played by the octave mandolin. But these patterns are fixed. To enable the composition of this structure in time, we implemented parameters at both global and local levels.

The first compositional parameter is a global one; it is the tempo that can be modified to shorten or stretch the duration of this global structure.

We also have two local compositional parameters for each rhythmic pattern. The first one is the temporal offset expressed as a music duration that enables to delay the first echo and the whole chain. For instance if the temporal offset is 1, it means that the series of echoes will be delayed by one quarter-note. The second parameter is the time stretching. It enables to contract or stretch the local duration of the pattern, without altering the speed of the other patterns. Combining these two parameters leads to various behaviours as for any delay line, but also to compose the temporal density and overlapping of the rhythmic events of the structure.

The time stretching parameter can also be modified by a LFO periodic signal (either a sinusoid or a phasor), provoking a kind of flanger on all the delay lines, and therefore various effects depending on the speed of the delay variation.

4.3. Object-oriented composition

Object-oriented composition is here conceived in a modelling and musical approach, not in terms of implementation (realized for instance in Javascript as in [9]). This principle of delay chains has led us to design four structures that are related. The first one, called *struct* is the set of the necessary information for the musical behaviour of the pattern:

- the transposition of the incoming sound expressed in midicents;
- the temporal offset of the structure expressed as a musical duration;
- the description of the time stretching of the structure:
    - fixed or variable
    - amplitude and offset (the latter being used only if the time stretching is variable)

The second class is called *enveloped_struct* and inherits from the previous class *struct*, including a sound detector and envelop triggering, adding threshold parameters to the member data. It is no longer the direct input but the enveloped octave mandolin sound that is sequentially delayed.

We created a third class that is close to *struct*, called *infinite_struct*. It musically represents a temporally 'perfect' tremolo but also the infinite limit of a never ending structure of delay lines having all the same duration. It is actually based on a single variable delay line with a reinjection equal to 1, as shown on figure 7. The rest of the structure is the same as in *struct*: a harmonizer, a temporal offset and the time stretching. Last but not least, *infinite_struct* also led to an *enveloped_infinite_struct* class that includes a sound detector and envelop generator as *enveloped_struct* proposed it compared to *struct*.

The musical combination of *struct* and *infinite_struct* patterns is very interesting to create some musical ambiguity and evolving behaviour.

We must add a word about the spatialization of these structures. We used the HOA library developed by our team at CICM / Musidanse Laboratory [15]. The ambisonic model enables to get various sound fields between two positions: on one hand, punctual sources; on the other hand, diffuse field. As in *Fil de Soi 1* and *Fil de Soi 2* pieces for acoustic guitar and live electronics, we use a matrix enabling us to vary the connections between the chains of delay lines and the spatial ambisonic components. With short values of delays in a chain, we can imagine a diffuse field based on decorrelation; on the contrary for the *infinite_struct* we connected the module to an encoder to establish a relationship between the duration of repetition and the speed of rotation.

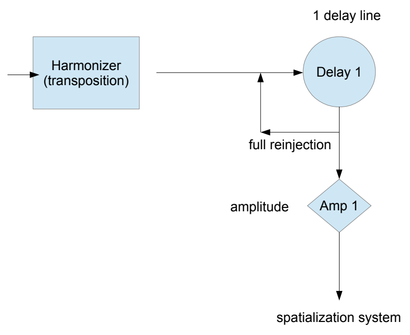

Figure 7: *The musical principle for the* infinite_struct *class.*

5. CONCLUSION

In this article we presented some research and creation work about the use of chains of delay lines in mixed music designed with Faust language. We have shown various applications of this principle, starting from pedagogy with the use of Kiwi collaborative environment embedding a Faust compiler. We also presented our new piece *Lire de Soi*, for octave mandolin and live electronics that uses several chains of delay lines including other functions as a harmonizer. In our object-oriented approach of mixed music, the control data of this kind of structure were grouped into a basic class with other classes that were derived from it. We expect this idea of chain of delay lines to be fruitful at various musical levels, also in relationship with ambisonic spatialization.

6. REFERENCES

[1] Scott Wilson, David Cottle, Nick Collins, *The SuperCollider Book.* The MIT Press, 2011.

[2] Roger Dannenberg, "Music Representation Issues, Techniques, and Systems", *Computer Music Journal*, 17(3), 1993, pages 20-30.

[3] Stephen Travis Pope, Introduction to MODE: The Musical Object Development Environment. *The Well- Tempered Object: Musical Applications of Object- Oriented Software Technology*. In S. T. Pope, MIT Press, Boston, 1991, pages 83-106.

[4] Xavier Rodet, Pierre Cointe. FORMES: Composition and Scheduling of Processes. *Computer Music Journal* 8(3):32-50, Fall, 1984.

[5] François Pachet, Geber Ramalho, Jean Carrive, Guillaume Cornic, Representing temporal musical objects and reasoning in the MusES system. *Journal of New Music Research*, 1996, (25) 3, pp. 252-275.

[6] Carlos Agon, Gérard Assayag, Olivier Delerue, Camillo Rueda, "Objects, Time and Constraints", *Proceedings of the ICMC 1998 Conference*, Ann Arbor, USA, 1998.

[7] Lounette M. Dyer, An object-oriented real-time simulation of music performance using interactive control, Ph.D. dissertation, Stanford University, 1991.

[8] Adrian Freed, John Maccallum, Andrew Schmeder, A., "Dynamic, instance-based, object-oriented programming in Max/MSP using open sound control message delegation", *Proceedings of the ICMC 2011 Conference*, Huddersfield, UK, 2011.

[9] Alain Bonardi, « Composition mixte à base de traitements et contrôles orientés objet – Exemple de mise en œuvre dans Pianotronics 3 », *Proceedings of the Journées d'Informatique Musicale 2016*, Albi, France, 2016.

[10] Horacio Vaggione, « Représentations musicales numériques : temporalités, objets, contextes », *Manières de faire des sons*, L'Harmattan, Paris, 2010.

[11] João Svidzinski, Alain Bonardi, « Vers une théorie de la composition musicale numérique fondée sur des réseaux d'objets », *Proceedings of the Journées d'Informatique Musicale 2015*, Montréal, 2015.

[12] Alain Bonardi, « Analyser l'orchestre numérique interactif dans les œuvres de Manoury », *Analyser la musique mixte,* Sampzon : Éditions Delatour, 2017.

[13] Anne Sèdes, Alain Bonardi, Eliott Paris, Jean Millot, Pierre Guillot, « Teaching, researching, creating: MUSICOLL ». *Innovative Tools and Methods to Teach Music and Signal, Processing*, (Laurent Pottier ed.), Paris : Presses des Mines - Transvalor, 2017.

[14] Pierre Guillot, Eliott Paris, Alain Bonardi, Intégration du compilateur Faust dans l'application de patching collaboratif Kiwi, *Actes des Journées d'Informatique Musicale 2019*, Bayonne, 2019.

[15] Alain Bonardi, Pierre Guillot, Concevoir des traitements ambisoniques en 2D et en 3D : l'exemple de Pianotronics 2. *Actes des Journées d'Informatique Musicale 2015*, Université de Montréal.