## 3 Introduction

At this point in the series we should be comfortable building modules that generate sounds without the sequencer or any user input, furthermore we should be able to create some sonically interesting arrangements too. However, we need to fill in a few gaps here and there, expand on some previously introduced techniques and explore some ideas about how to get various sounds to interact.

The main topics for this part in the series are: generating sequences, audio to CV conversion, using the sound spectrum and random chord generation.

For those of you who have just arrived please read [part 1](https://medium.com/generative-music-using-reason/9063f7a03f3e) & [part 2](https://medium.com/generative-music-using-reason/e6466262ed40) before delving into this one. As with the previous parts I have provided source files throughout, some of which are Combinator patches.

## 3.1 Random sequence trigger

As we have learned, there are lots of ways of triggering sounds via CV and one particular configuration used Thor’s sequencer set to Random, Kong’s sequencer inputs and then the Kong gate outs as triggers for various instrument devices. We can augment this by triggering whole sequences, which can help us build up a more rich and complex piece of music.

To do this grab the file [**random-sampler-notes.cmb**](http://point-b.co.uk/blog/autonomous-generative/part2/random-sampler-notes.cmb) from [Part 2](https://medium.com/generative-music-using-reason/e6466262ed40) and swap out the NN-19 devices for 6 Thors, only this time take the gate out signals(on the fist 6 pads) of the Kong and connect them to the Step Sequencer’s Trigger in socket in each of the Thors as show in Figure 1.1 .

**Fig. 1.1**

Gate Trigger In on the back of Thor’s sequencer

To avoid a potential sonic mess ensure the Thor devices we’re triggering are set to One-shot mode — this means that the sequence will only play once each time it is triggered.

**DL:** [**thorSeqRandomTriggers**](http://point-b.co.uk/blog/autonomous-generative/part3/thorSeqRandomTriggers.cmb)

You’ll notice that I also substituted the Line Mixer for a Mixer so that we have more available channels. As there are 16 pads on Kong we may want to trigger 16 separate instruments. Remember that we have limited the number of pads that the random sequencer triggers by reducing the step count of the master (Thor) sequencer. To increase the range of Kong pads (and therefore instruments) simply increase the number of steps on the Thor sequencer or preferably via the Combinator Rotary dial labelled ‘Steps’ that we configured earlier as illustrated below in Figure 1.2.

**Fig. 1.2**

The ‘Steps’ rotary, which has been configured to control the number of steps the Thor sequencer uses

### 3.1.2 Chaining

If we wanted to chain sequences together we could take a Step Sequencer Trigger Out and plug it in to a Step Sequencer Trigger In on another Thor. This is an effective way of extending the length of your Thor sequences and can be done over and over until your processor starts to complain. However, if overused it may bring a little too much order into our system and start to make our generative music sound a little too regular.

### ****3.1.3 Opening up****

I find that this arrangement can suffer from a bit too much clutter, but there are obvious ways to mitigate this problem.

We may want to open up the space between each sequence in time which can be easily achieved by lowering the tempo of the master sequencer or better still, by automating the rate at which the master random sequencer plays. To do that simply pick a source such as an LFO from any instrument and plug it into the Rate In of the Thor sequencer or use the ‘Sequencer’ Thor itself by assigning one of the internal LFOs to the Sequencer Rate as shown in Figure 1.3 below.

**Fig. 1.3**

LFO 1 to Sequencer Rate on the ‘Sequencer’ Thor

Remember that use very dynamic signals as trigger sources is a great way to add even more variety and unpredictability into our system. Typically dynamic sources could be merged CV signals, something we touched on in part 1, or audio sources.

## 3.2 Dynamic variety

We can use the velocity value from any sequence to trigger another, which is particularly interesting if we use audio as our source. Essentially this configuration uses the peak values of a signal to trigger something else.

### 3.2.1 Audio to CV

Converting from audio to CV in Reason can be done a number of ways, but one of the most effective ways is by using the Thor’s programmer. To do this simply take the left (or mono) audio out from your preferred device and connect it to the Audio Input 1 socket on the back of the Thor (see Figure 1.4 below). Then, flipping to the front of the device, route the Audio Input 1 to the CV output 1 inside the programmer and turn the Amount up to 100 as illustrated in Figure 1.5. Note that in Figure 1.4 I have split the audio signal using a Spider Audio Merger & Splitter so that we can hear the audio as well as convert it to CV.

**Fig. 1.4**

Audio to CV connections

**Fig. 1.5**

Audio to CV programmer

Provided your audio has some extreme peaks and troughs this should work. I recommend something that is dynamic, in terms of amplitude, so that it shapes the parameter — something without enough contrast will not give us enough variety and so we won’t hear much.

The CV signal we get from our interesting audio sample can be used to do anything we want but one technique I want to share turns the audio into a step sequencer trigger.

### ****3.2.2 Audio peaks as a trigger****

In the example Reason file provided below I have used a field recording which contains a sequence of random(ish) sounds and they in turn are the peak values used to trigger our Thor sequencer, only I don’t want it to play the whole sequence each time it’s triggered, just the next note. This gives us the rhythmic unpredictability we can use in our generative music.

So, with our set up as it is flip around to the back of the Audio to CV Thor which is doing our conversion and take the CV 1 out and plug it in to the Sequencer Trigger In as shown in Figure 1.6.

**Fig. 1.6**

Thor CV1 to Sequencer Trigger In

Return to the front of the device and set the Sequencer play mode to ‘Step’. Now when our audio peaks the sequencer will move on one step. The sequencer trigger is (like a lot of Gate inputs) is quite sensitive, so you may well need to adjust Audio to CV amount on Thor programmer. Both these changes are shown in Figure 1.7 below.

**Fig. 1.7**

Thor Sequencer mode set to Step and the Audio In 1 to CV 1 Out amount adjusted

As you may notice in the Figure above I have set the notes on the sequencer to a simple chromatic scale to demonstrate more clearly the effect of the audio source’s peaks on the notes output by this Thor. It isn’t an interesting piece of music more a just as a proof of concept.

Mute channel one on the Line Mixer to hear the scale being played without the original audio clip playing. If you haven’t found a suitable audio clip or want to see the complete arrangement download the Reason file provided below.

DL: [**Audio-to-CV-Trigger.reason**](http://www.point-b.co.uk/blog/autonomous-generative/part3/Audio-to-CV-Trigger.reason)

To trigger the audio clip I have used a Pulsar LFO with two signals (one of which is inverted). Of course, you an use any external MIDI device or the sequencer to trigger the audio clip, but in the spirit of this series we want to avoid that.

The audio clip may take a second or two to start playing depending on where the LFO is when the document opens. Not being able to detect the end point of an audio clip is a bit of a problem in Reason, but with a bit of trila and error you should be able to generate a signal, as I have, that repeats at a similar inverval to the length of the clip.

### 3.2.3 Audio as multi-device trigger

Moving on from the example above let’s use our audio clip to trigger multiple devices. You may have guessed already that the best way for us to do this is to use our Audio/CV signal to trigger our Thor Step Sequencer, which in turn uses the Kong channels to trigger our target devices. It might appear a little convoluted but once we have this set up working we can use it as a template for future modules or projects.

In my source file below I have assigned Rotary one to a master Filter Cutoff effect and Rotary 2 to the fine tune of the array of Saw Synths. Download the file and have a play around with the possibilities.

DL: **Audio-to-CV-Trig-Seq.reason**

Some further reading about Audio to CV can be found here <http://www.reasonexperts.com/Audio-to-CV-in-Reason.html>

## 3.3 Shaping

We can use signals to shut off sounds which can create a very powerful effects and importantly it can be used to help prevent any sonic bloat. A common technique producers use to create a “Ducking” or “Dipping” effect utilises side-chained compression. This is often used so that the kick drum knocks back the rest of the track. We can do this purely in CV and gain a lot more control from doing so.

Let’s create a new Combinator module based on our [randomdrums.cmb](http://point-b.co.uk/blog/autonomous-generative/part2/random-drums.cmb) file from Part 2. In this example we want to use the CV from one of our drum sounds to shut off the sound of another instrument. So take a CV Gate Out from one of the Kong drum channels and hook it up to the CV 1 in of a new Thor that we name “CV Envelope” as illustrated below in Figure 1.8.

**Fig. 1.8**

A Gate Out from the Kong to CV1 in on the Envelope Thor

The next step is to take a cable from Thor’s Global Envelope and plug it in to one of our Combinator’s Rotaries. This way our Global Envelope can shape anything we want within the Combinator module.In my example Rotary 1 is being used for the drum Sequencer rate so I have used Rotary 2.

**Fig. 1.9**

1.9 Thor Global Env Out to Combinator Rotary 2 In

Flip back to the front of the rack and let’s get the Global Envelope working as we want it. In the Thor programmer select CV1 as the source (this is our drum Gate Out signal from Kong) and set the Destination to the Global Envelope Gate Trigger and turn the amount up to 100 as shown in Figure 2.0 below.

**Fig 2.0**

Global Envelope triggered by CV1 Source

Our Global Envelope is now controlling the Rotary whenever it is triggered so all that remains is to assign our Rotary to something. In order to create our “Dipping” effect we can assign Rotaty 2 to Level 1 on the Mixer and ensure that the Min value is higher than the max. Now, any adjustments we make to the Global Envelope itself will affect how long or short the “Dipping” effect is. Download the Combinator patch below to see my example.

DL: [**shut-off.cmb**](http://www.point-b.co.uk/blog/autonomous-generative/part3/shut-off.cmb)

As you can hear it is very powerful because the envelope has so many parameters. In the patch below I have added another pad sound and used the Envelope signal (from Rotary 2) to boost the level. As you can hear it acts in a similar way to the crossfade we made earlier except that we have more control over the shape of the fade.

DL: [**shut-off-fill.cmb**](http://www.point-b.co.uk/blog/autonomous-generative/part3/shut-off-fill.cmb)

Try using this arrangement to shape other aspects of this module or one of your own and hopefully you will see that using sources to affect the shape of other sounds can help structure a track…or make it more wild.

## ****3.4 The Specrum****

Taking what we have just explored in 3.2.3 we can set up interesting ways of modulating the behaviour of our modular synths. So, what are our options? We could have patterns trigger each other or we could use the features of one sound to affect another. I will cover each of these areas in turn

There is one device in Reason which permits us to use different frequencies from a sound source and that is the BV512 Vocoder. So far we have only used the volume of a signal to trigger or shape sounds, but with this handy device we can use the values from different frequency bands to control other sounds.

To get started, create a new Combinator with a Mixer and BV512 inside. Then choose an instrument and load a favourite patch of yours. Now we need to split the audio signal from this device as we want to be able to hear it as well as process it, so create a Spider Audio Merger/Splitter and split the signal so that one copy of it goes to the Mixer and the other into the (Mono) Modulation input on the BV512. Figure 2.1 below shows the basic routing for this set up.

Ensure the BV512 is set to Vocoder mode. Now when you play you should be able to hear your chosen instrument and see which frequencies it uses on BV512's “Modulation Levels” display.

**Fig 2.1**

Basic cabling configuration for using the BV512 as a spectral display unit

In order to use the levels from these bands to generate other sounds we can take a CV signal from any of the 16 “Individual Band Levels” Out sockets on the back of the Vocoder device and plug them into CV inputs on other devices. In my example file below I have used a Kong Drum Designer with some simple synth drums assigned to the first few channels and routed some of the Band Level outputs from the back of the BV512 to the Gate In on the aforementioned drum channels.

DL: **spectral-composer.cmb**

Again, this is just a proof of concept to demonstrate how the spectral range of one sound can be mapped to a series of individual sounds. With a little imagination and some perseverance I'm sure you’ll find some interesting ways in which to use this arrangement.

Before moving on notice how the different Oscillators in the Thor create different results. Each is sweeping through a range of frequencies fuelled by LFO1, but each different wave type produces different harmonics which play the drum machine differently. This arrangement is very sensitive so be careful.

You can just spend the whole night putting cables in and seeing what happens when you turn it on in the morning. And it’s quite unpredictable…it reacts to itself

James Holden

## ****3.4 Chords****

One conspicuous aspect of music creation I have omitted so far is how to generate chords. So lets look at a couple of ways of tackling this.

### ****3.4.1 The simple way****

The most basic way to achieve this is to use a Thor with three Oscillators that all use roughly the same sound and feed in three separate random CV note signals. You could use any LFO source to do this though I have opted for a Pulsar Duel LFO for brevity.

Simply take three cables from Pulsar’s LFOs (one regular signal from LFO1, one inverted signal and another from LFO2) and plug them into the CV in sockets on the back of out Thor synth as shown in Figure 2.2 below.

**Fig 2.2**

Pulsar LFO CV outs to Thor CV Ins

I have named the Pulsar as LFO Note so it is clear that it is the source of our note data.

Ensure both LFOs are set to random square wave and that the three Thor Oscillators are routed through one of the Filters or bypass. On the front of the Thor use the programmer to assign the CV In signals to the Pitch of each Oscillator as showin in Figure 2.3.

**Fig 2.3**

Thor CV Ins to Osc Pitch

Now all we need to do is trigger Thor’s sequencer to play the notes, so I will create a separate LFO source (another Pulsar), set it to the same rate as the other one and plug it in to the Gate In on Thor’s Sequencer Control. So the complete configuration looks like Figure 2.4 below.

**Fig 2.4**

Pulsar to Thor Seq Gate In

I have named the source device of our Gate signal as LFO Gate and used LFO 2 as our source as it has built in On/Off switch which we can use to start or stop the patch.

In order to give us control over the rate at which the chords are played I have assigned both Pulsar’s LFO Rate parameters to Rotary 2. Rotary 1 is being used to control the range of random notes via the LFO level parameters on the LFO Note Pulsar. Feel free to download the Combinator patch below to hear it in action.

DL: [**random-chords.cmb**](http://www.point-b.co.uk/blog/autonomous-generative/part3/random-chords.cmb)

Remember, if for any reason it goes out of synch and you start to hear notes leaking from one step to the next press play (and then stop) on the main Reason sequencer and all the LFOs should go back in phase.

Of course this set up limits us to combinations of just three notes and our random LFO shape may well produce non-chromatic notes. So let’s look at a way of using more notes and gaining a little more control.

### 3.4.2 Multiple random notes

Rather than using one Thor and 3 internal Oscillators let’s use an array of Thors whose sequencers we’ll use to play our random notes.

In the supplied example patch ([random-chords-seq.cmb](http://www.point-b.co.uk/blog/autonomous-generative/part3/random-chords-seq.cmb)) I have configured 4 Thors with 12 notes as the maximum range, each note a semitone apart and set the Oscillator on each Thor set to a different octave.

DL: [**random-chords-seq.cmb**](http://www.point-b.co.uk/blog/autonomous-generative/part3/random-chords-seq.cmb)

We can control the rate of all them them simultaneously via one of the rotaries and assign each Sequencer Step Count to another rotary to give us our note range. So, more specifically, in the Combinator programmer we assign the Sequencer Step Count range from 1-12 on Rotary 1 and the Sequencer Synched Rate on Rotary 2 for each Thor device we have created. Figure 2.5 below demonstrates what the Combinator programmer looks like

**Fig. 2.5**

Combinator programmer for random chords

As you can hear, this technique ensures we get clean, distinct chords. However, the chords generated in both the previous demos are quite challenging, so I’ll leave you with something that has been fine tuned to be a little more easy on the ear:

DL: [**random-chords-seq-alt.cmb**](http://www.point-b.co.uk/blog/autonomous-generative/part3/random-chords-seq-alt.cmb)

Each Thor can play a range of notes that you decide, so we can manually create an arrangement that includes or excludes certain notes, making for a less discordant result.

## 3.5 Conclusion

The three main areas we have covered in Part 3 should arm you with new ideas that can help make your arrangements more dynamic and more structured. Next time I will show how we can get various modules to interact and introduce some nice Rack Extensions that can help us navigate tricky situations and problem solve common issues.