Obiwannabe

Use the source...

Sponsored by the number 3,720:1

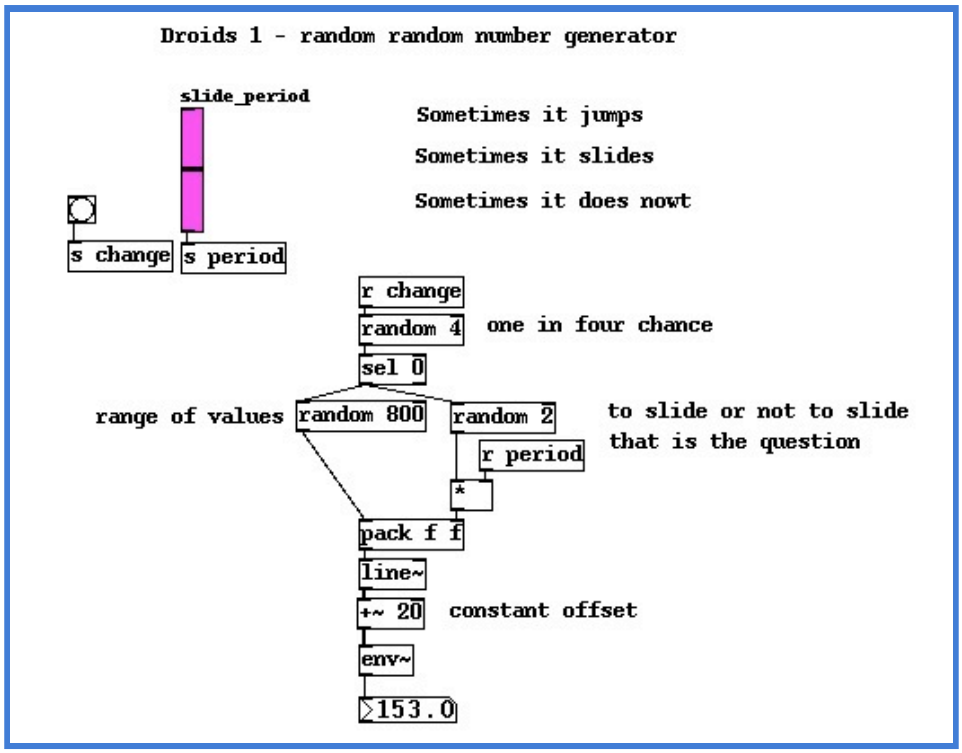
Astromech Droid

I'm sure plenty has been written about Star Wars sounds. These noises and algorithms are my own take from first principles. If you want an exact R2D2 sound you're going to have to tweak this patch, but it's not my intention to exactly replicate that precise sound. From observation R2D2 in the Star Wars films sounds like a few familiar things. He has something in common with bird call. A blackbird exhibits quite a few spectral and amplitude gestures not entirely unlike the lovable tin can, there are modulation sweeps, pitch and modulation sweeps, just pitch sweeps, constant pitches and breaks of continuous or very broken calling. For a while I wondered whether some of the R2D2 sounds were treated bird call, possibly with ring modulation. Another sound that comes to mind is the 303 acid sound, especially regarding the pitch slides, but there is the suggestion that a sawtooth phasor is the waveform underlying the whole sound. Many of R2D2s sounds were realised on an ARP2600 by sound designer Ben Burtt, but not all the sounds in the films are synthesised,

"50% of the droid's voice is generated electronically; the rest is a combination and blending of water pipes, whistles, and vocalizations by (Ben) Burtt."**

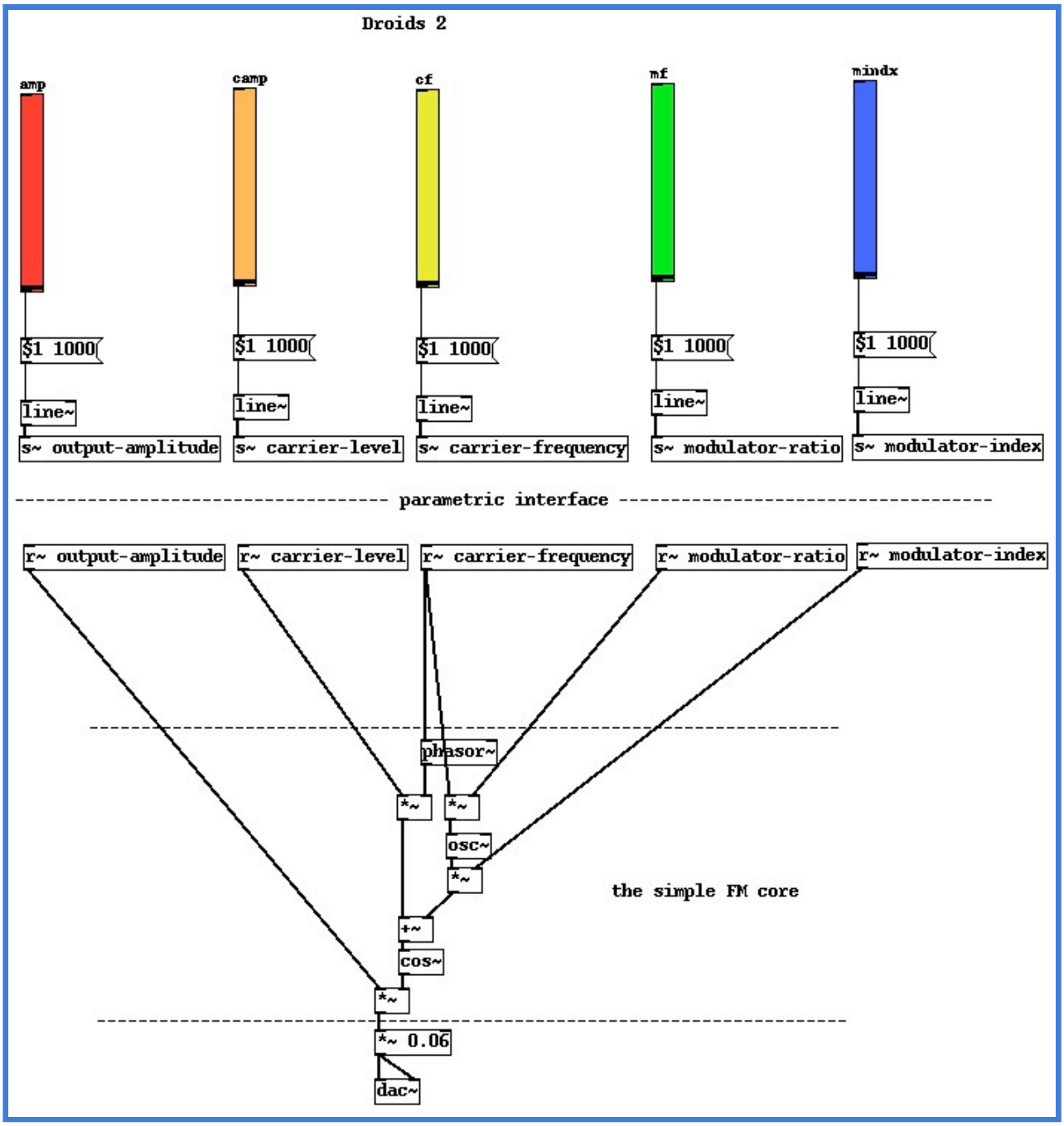
The Puredata diagram at the end of this section looks rather complex. It is not. It just looks frightening because of the large number of units needed to make it, but both the operational theory and the synthesis method at the heart of it are very easy indeed, if you've come this far through the tutorials you are going to laugh once you realise how simple it really is.

Here is one of the most frequently used patterns in the patch, it's a random value generator. When it receives a bang message it doesn't always generate a new random number. You could say it's a random random number generator, each time it generates one random number to decide whether to generate the next. There are a few variations you will see which either output just a random number, stepping from one value to another, or output a line segment that slides between the previous and new value. The slide time is always derived from the tempo that the metronome runs at, hence the input labled [r period] which makes the slide shorter or longer as the tempo changes. The trick is that it doesn't always slide or step, it just randomly selects which one to do each time.



Puredata file .pd

Here's the synthesis block at the heart of the patch. It's just a good old fashioned FM algorithm. The only things worth noting are that we are going to drive the modulation and index values well outside the normal range to deliberately introduce foldover aliasing. Why? Because it just sounds cooler and more technological that's all. If you look in the final patch there are a few [lop~] and [hip~] units to kill the really low and high frequencies that might damage speakers or ears, but apart from that we make no attempt to control the spectrum or bandlimit any signals the way we would for a musical instrument, we actually want it to sound a bit messed up.

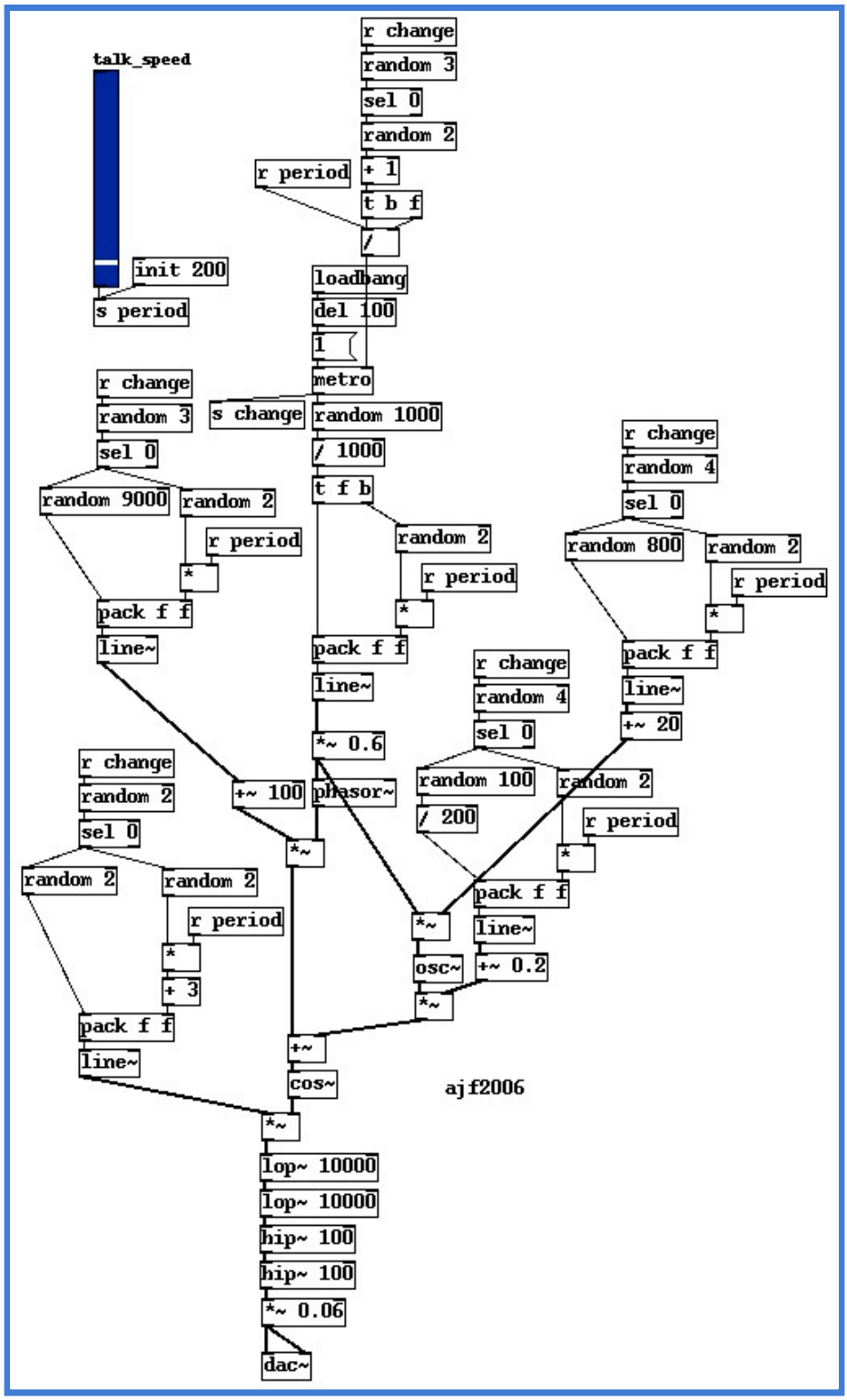


Puredata file .pd

All that I've done in the final patch is to connect random value generators that either slide or step (at random) between random values for the carrier frequency, the modulation frequency and the modulation index, and the patch output amplitude. There are a few constants to offset one or two values and make sure they aren't too high or low, but that's about it. Some of the constants are picked to keep him sounding quite cute and unthreatening. If you want to extend the ranges and get an Evil™ R2D2 try changing the carrier scaling from [*~ 0.6] to a bigger value. Imperial probe droids use a different algorithm though, which we might look at later, but you can easily convert R2 to the dark side by adding a little AM (ring modulation to his output).

It's operation is random. Switch it on and you don't know what might happen, it

might not even make any good noises for a while. Most of the time it sounds something a bit like a maintainance droid, other times it wigs out and makes the strangest noises, once or twice I swear I've heard it talk. The story behind R2D2, and his name, is that the soundreel on which his effects were recorded was labled R2-d2 (probaby reel 2 dub 2). The Lucas sound dudes probably did very much like myself and other producers, when you get a good sound coming out of a synth just record loads of it down to listen through later. In that sense R2D2s sounds are not so much "designed" as *selected*. If you notice in the films there's a few really good ones they use over and over, in fact if you could speak droid protocol as well as C3PO you would probably get just as annoyed at R2D2 because he's always repeating himself. You could use a patch like this to generate hours of synthetic babble and then audition it for juicy nuggets of goodness. Personally I would prefer to have it synthesised in a game in an entirely organic way (but see the caveat in the section on advanced parameterisation, regarding replication of shared experiences for multiplayer games if you are a games designer).



Audio .mp3

Puredata file .pd

Aside (technical philosophy):

The astonishing range of sounds that emerge from this patch shows us a few things. Firstly all those sonds are "implicit" in the extremely simple synthesis algorithm in the middle. All we have to do to create them deterimistically is to supply just *five*

parameters. But that's the trick see, due to dimentional explosion the number of possible values that make meaningful timbres is gargantuan. You could run this patch for years and *never* get the same exact sound twice. On the other hand every single sound shares a kind of familiar form. This is almost paradoxical. There are a near infinite number of sounds this circuit can produce, and yet they all sound somehow alike. We don't need Russell or set theory to understand why. Pure information analysis tells us that the number of distinct one second sounds is greater than the number of atoms in the universe. Sometime in synthesis I talk about a parametric timbre (or spectral) space, which is a subspace of a much larger universe of sounds. We could never enumerate these theroetical spaces, it would take more than a human lifetime to hear all the possible one second sounds made by even a 3 dimentional sound! And yet after listening to this patch babble away for about 20 minutes we feel as though we know and could recognise the patch and all the sounds it could ever produce. How can that be? It's because the way we recognise sounds has as much to do with absolute waveforms as the way we recognise a painting by an artist has to do with photons of light. We are amazing at extrapolating the timbre space from a few examples. This is how a bunch of random sqeals and beeps can make us say "Aha! that's R2D2" based only on the generative method, not the actual noises. Bear this theory in mind when considering whether any two sounds can truly be identical, how obvious it is when you sample someone elses work (because the chances of identical waveforms occuring is negligable) and consequently why sampling sucks because the brain grows tired quickly of repetitive instances drawn from the same space.

Links:

http://www.filmsound.org/starwars/

http://en.wikipedia.org/wiki/R2D2

next tutorial

tutorials list

* The odds of successfully navigating an asteroid field.

** Filmsound.org