

The SysSon Platform

Technical Report TR-2016-11-1
Institute of Electronic Music and Acoustics, Graz
(Status: in progress)

Hanns Holger Rutz

November 2016

1 QBO - Blob Sonification

After some initial experiments with frequency modulation to indicate blob slice height, it was decided to try out other forms of timbre modification such as wave-shaping. SuperCollider provides wave-shaping by means of the **Shaper** UGen, typically with a buffer prepared with Chebychev polynomial functions. However, it seems not possible to continuously fade in a particular timbre by altering the input signal's amplitude, as different lower partials will transitorily be attenuated. Another possibility is through the **VOsc** variable table oscillator. In order to generate the appropriate wave-tables, a graph element **BufferGen** has been added to *Sound Processes*. As **VOsc** depends on a trick of allocating multiple buffers with consecutive identifiers, and this consecutiveness is currently not possible to guarantee in *Sound Processes*, one can simply mix and blend multiple **OSC** instances manually, which has the same effect. The code is shown in Fig. 1, where the **amp** parameter is expected to be in the range from zero to one, and it scans through the different spectra.

Fig. 2 shows the sonogram of a bounce of this sonification model with the QBO blob data. The bounce can be heard at <https://soundcloud.com/sysssonproject/blob-shaper161102>. The parameters are:

- time = 2002-01-16 12:00:00Z to 2016-02-15 12:00:00Z
- lon = 75.00 °W; lat = 2.50 °S
- speed = 6 months/sec, mag-max = 3, min-freq = 300 Hz, max-freq = 800 Hz
- spread-mod-depth = 1.5, spread-mod-offset = 0.3

TODO: continue here

```

def mkOsc(freq: GE, amt: GE): GE = {
  val oddBase = 1f
  val evenBase = 0f
  val oddDamp = 0.7f
  val evenDamp = 0.8f
  val numHarm = 9
  val numBufs = 5
  val tableSz = 1024

  val oscs = (0 until numBufs).map { i =>
    val amps0 = Seq.tabulate(numHarm) { j =>
      val isEven = (j + 1).isEven
      val base = if (isEven) evenBase else oddBase
      val damp = if (isEven) evenDamp else oddDamp
      val exp = (j / 2) * (numBufs - i)
      base * damp.pow(exp)
    }
    // first is forced to be fundamental only
    val amps = if (i == 0) Seq(1f) else amps0
    val buf = BufferGen.sine1(amps, numFrames = tableSz)
    Osc.ar(buf, freq)
  }

  val idx = amt.linlin(0, 1, 0, numBufs - 1)
  val idxF = idx.floor
  val idxC = idx.ceil
  val wC = idx % 1.0
  val wF = 1.0 - wC

  val osc = Select.ar(idxF, oscs) * wF + Select.ar(idxC, oscs) * wC
  osc
}

```

Figure 1: Generation of oscillator mix implementing blending of partial frequencies.

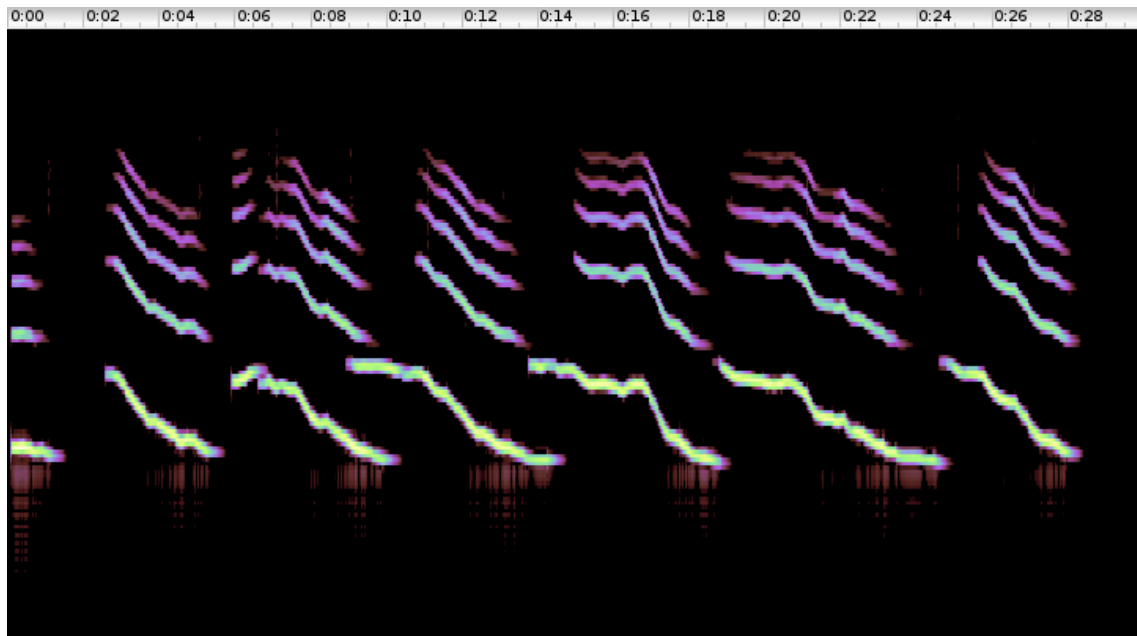


Figure 2: Sonogram of QBO sonification with blob slice height mapped to overtone spectrum