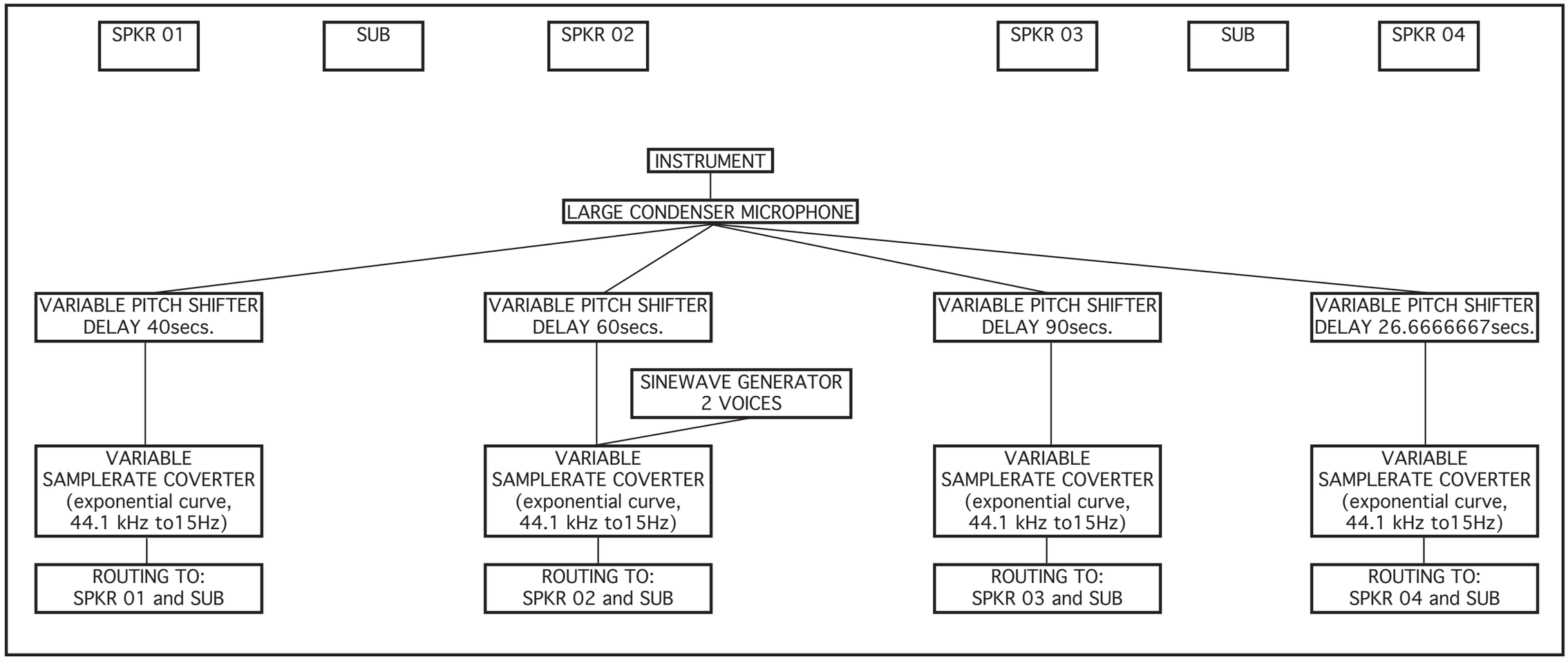
MARTIN LORENZ
OSCILLATIONS (2012) for Low Pitched Instrument and Electronics

www.martinlorenz.ch

Electronics Setup:



The amplitude levels of the instrument, the sinewave generators and the (transposed) feedback should be the same.
The performers should strive for a consistently loud volume throughout. The mix engineer can provide assistance, including compression, as necessary and appropriate.

OSCILLATIONS for Low Pitched Instrument and Electronics

Martin Lorenz 2012

PITCHSHIFTER 01 / OUTPUT DELAY 40secs.

P:SHIFTER: +700 cents | ±0 cents | +700 cents | +1400 cents | +700 cents | ±0 cents | +700 cents | +1400 cents | +1400 cents | +700 cents | ±0 cents | +700 cents | ±0 cents | +700 cents | +1400 cents | +700 cents | ±0 cents

S.RATE: 44100Hz | 44100Hz | 2160Hz | 1440Hz | 640Hz | 640Hz | 1440Hz | 1440Hz | 05Hz

PITCHSHIFTER 02 / OUTPUT DELAY 60secs.

P.SHIFTER: +700 cents | ±0 cents | +700 cents | +1400 cents | +700 cents | ±0 cents | +700 cents | +1400 cents | +1400 cents | +700 cents | ±0 cents

S.RATE: 44100Hz | 44100Hz | 10Hz

PITCHSHIFTER 03 / OUTPUT DELAY 90secs.

P.SHIFTER: +700 cents | ±0 cents | +700 cents | +1400 cents | +700 cents | ±0 cents | +700 cents | +1400 cents

S.RATE: 22050Hz | 44100Hz | 07Hz | 07Hz

PITCHSHIFTER 04 / OUTPUT DELAY 26.66667secs.

P.SHIFTER: +700 cents | ±0 cents | +700 cents | +1400 cents | +700 cents | ±0 cents | +700 cents | +1400 cents | +700 cents | ±0 cents | +700 cents | ±0 cents | +700 cents | +1400 cents | +1400 cents | +700 cents | ±0 cents | +700 cents | +1400 cents | +700 cents | ±0 cents | +700 cents | ±0 cents | +700 cents

S.RATE: 3000Hz | 3000Hz | 106.6Hz | 08Hz

0:00 | 1:00 | 2:00 | 3:00 | 4:00 | 5:00 | 6:00 | 7:00 | 8:00 | 9:00 | 10:00

360Hz
240Hz
160Hz
106.6Hz
71.1Hz
47.4Hz
31.6Hz
21Hz

INST.: 106.6Hz | 71.1Hz | 106.6Hz | 71.1Hz | 106.6Hz | 160Hz | 106.6Hz | 71.1Hz

FOR THE FIRST PERFORMANCE OF THE COMPOSITION, LUC DÖBEREINER PROVIDED THE FOLLOWING SUPERCOLLIDER CODE:

```
(
// Init Variables and Synth Defintion for ”Sidedelays”
~inputIdx = 0; // Input
~unit = 60; // unit duration in seconds

~mainRS1 = Bus.control(s,1);
~mainRS2 = Bus.control(s,1);
~mainRS3 = Bus.control(s,1);

SynthDef(”sidedelay”, {arg in = 4, out = 2, unit = 30, unitFac = 0.75, rs = 44100;
      var input,unitThis,ps,del,res,sin,sin2,seiten,rsfr;
      unitThis = unit * unitFac;
      input = SoundIn.ar(in);
      ps = PitchShift.ar(input,pitchRatio: EnvGen.kr(Env.new([1.5,1,1.5,2.25,1.5,1,1.5,2.25,2.25,1.5,1],
            [1,1,1,1,1,1,1,1,1,1]),
            gate: Impulse.kr(1/(10*unitThis)),timeScale: unitThis));
      del = DelayL.ar(ps,unitThis,unitThis);
      rsfr = K2A.ar(In.kr(rs));
      res = LPF.ar(Resampler.ar(del,rsfr),(rsfr/2).lag(1).clip(25,22050));
      Out.ar(out, res);
}).send(s)
)

(
{ Out.kr(~mainRS1,EnvGen.kr(Env.new([3000,3000,106.6,8],[7,3,1],[1,-2,-2]), timeScale: ~unit)) }.play;
{ Out.kr(~mainRS2,EnvGen.kr(Env.new([44100,44100,540*4,360*4,160*4,160*4,360*4,360*4,5,5],
      [3,1,1,1,1,1,2,1,1], [1,1,-2,-2,1,2,1,-2,1]), timeScale: ~unit)) }.play;
{ Out.kr(~mainRS3,EnvGen.kr(Env.new([22050,44100,7,7],[7,0,4],[2,1,1]), timeScale: ~unit)) }.play;

~osc1 = Synth(”sidedelay”, [\in, ~inputIdx, \out, 0, \unit, ~unit, \rs, ~mainRS2, \unitFac, 0.66666667]);
~osc2 = Synth(”sidedelay”, [\in, ~inputIdx, \out, 3, \unit, ~unit, \rs, ~mainRS1, \unitFac, 0.444444445]);
~osc3 = Synth(”sidedelay”, [\in, ~inputIdx, \out, 2, \unit, ~unit, \rs, ~mainRS3, \unitFac, 1.5]);

{
      var input,unit,ps,del,res,envres,sin,sin2,seiten;
      unit = ~unit;
      input = SoundIn.ar(~inputIdx);
      ps = PitchShift.ar(input,
            pitchRatio: EnvGen.kr(
                  Env.new([1.5,1,1.5,2.25,1.5,1,1.5,2.25,2.25,1.5,1],[1,1,1,1,1,1,1,1,1,1]),
                  timeScale: unit));
      del = DelayL.ar(ps,unit,unit);
      sin = SinOsc.ar(EnvGen.kr(
            Env.new([71.1, 106.6, 160, 240, 240, 160, 106.6, 160, 160, 240, 160],[1,1,1,1,1,1,1,1,1,1]),
            timeScale: unit)) * 0.1;
      sin2 = SinOsc.ar(EnvGen.kr(Env.new([0,0,21,31.6,47.4,31.6],[7,0,1,1,1]), timeScale: unit)) * 0.1;
      envres = EnvGen.ar(Env.new([44100,44100,10],[5,6],[1,-2]), timeScale: unit);
      res = LPF.ar(Resampler.ar(del+sin+sin2,envres),(envres/2).lag(1).clip(25,22050));
      Out.ar(1, res);
}.play;

Task({ var t=0; inf.do({t.postln; t=t+~unit; ~unit.wait;}) }).start;
)
```