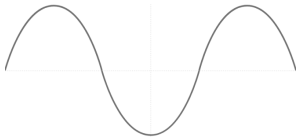


1 of 2 oscillators is oké

Synthesis is niet in de eerste plaats een waaier van klanken.

Het is een manier om geluid te begrijpen als een geheel van op elkaar inwerkende processen.



Vragen

Les 1 stelt de vraag: Waaruit bestaat een geluid? We gaan van één sine wave naar waveform, spectrum, beating, additive synthesis en envelopes.

Les 2 stelt de vraag: Hoe kunnen eenvoudige signalen complex worden? We verkennen filtering, modulation, noise, randomness en een klein generative system.

Aan het einde kunnen we een eenvoudige MSP signal chain lezen en doelbewust een eenvoudige synthesiser opbouwen, in plaats van alleen presets te selecteren.

Voorbereiding

Gebruik de bijbehorende patches 01-10. De spectroscope dient om te bevestigen wat het oor al aan het ontdekken is.

Max/MSP gebruikt een signal network: MSP objects worden met elkaar verbonden om digital audio te produceren en te bewerken. De huidige tutorial-reeks van Cycling '74 groepeerde additive/modulation synthesis, filtering/subtractive synthesis, dynamics, analysis en verwante technieken op deze manier. De onderstaande lessenreeks volgt bewust dezelfde conceptuele opbouw, zonder dat we tijdens de les de tutorials hoeven te raadplegen.

LES 1 - Geluid opbouwen uit bijna niets

Centrale vraag: Hoe kan een vibrerend getal iets worden dat we als geluid herkennen?

1. Opening: synthesis als constructie

Wat is het eenvoudigste mogelijke elektronische geluid? Een sine wave. Anders dan bij een opname zit er geen vastgelegde akoestische gebeurtenis achter: de computer berekent voortdurend opnieuw een waveform. Synthesis begint wanneer we beslissen wat moet variëren - frequency, amplitude, spectrum, time, of de relatie tussen verschillende signals.

Het eenvoudigste instrument: GENERATOR -> AMPLITUDE -> SPEAKER. In MSP kan dit

cycle~ ->

*~ ->

ezdac~ zijn.

De tilde is belangrijk: die duidt objecten aan die werken met continu berekende audio signalen.

2. Frequency en amplitude

Open patch 01. Een sine oscillator is pedagogisch nuttig omdat de oscillator energie concentreert op één frequency. Stel hem in op 440 Hz, daarna op 220 Hz en 880 Hz. Frequency beschrijft het aantal cycles per second en wordt voor- namelijk waargenomen als pitch, hoewel pitch *perception* niet identiek is aan *frequency*. Een verdubbeling van de frequency levert een octaaf op.

Laat nu de frequency ongewijzigd en verander de amplitude. In MSP schaal je het vermenigvuldigen van een signal de amplitude ervan. Dit levert een cruciaal synthesis principle op: een signal kan numeriek worden behandeld. Een geluid is geen onaantastbaar object; het is data die kan worden multiplieerd, added, filtered en gebruikt om een ander geluid te control.

3. Waveform en spectrum

Open patch 02. Vergelijk sine, triangle, sawtooth en rectangular/pulse waves bij dezelfde fundamental frequency. Hun repetition rate kan identiek zijn terwijl hun timbre verschilt, omdat hun spectra verschillen. Een sine is het grensgeval: in essentie één spectral component. Meer hoekige periodic shapes vereisen extra harmonics.

Gebruik de spectroscope. Een sawtooth bevat een dichte harmonic series; een triangle bevoordeelt sterk de lagere odd harmonics; een square wave wordt gedomineerd door odd harmonics. De belangrijke relatie is: SHAPE IN TIME <-> DISTRIBUTION OF ENERGY IN FREQUENCY.

phasor~ is bijzonder bruikbaar als een 0-1 ramp/control signal, terwijl Cycling '74 de band-limited saw~ aanbeveelt wanneer het doel een vloeiender hoorbare sawtooth is. Dit onderscheid introduceert het idee dat mathematically simple digital waveforms ongewenste aliasing kunnen veroorzaken.

4. Beating: complexiteit zonder een complexe oscillator

Open patch 03 en stel beide oscillators in op 440 Hz. Verander er vervolgens één naar 441 Hz. De gecombineerde amplitude stijgt en daalt nu ongeveer één keer per seconde. Probeer 443, 445 en 450 Hz. Er is niets expliciet geprogrammeerd om een rhythm te creëren; het rhythm ontstaat uit **interference** tussen twee continuous oscillations.

Dit vormt een nuttige conceptuele brug naar mediakunst:

complexiteit kan ontstaan in de relatie tussen eenvoudige componenten, in plaats van binnen de componenten zelf.

5. Additive synthesis

Open patch 04. Voeg verschillende sine waves samen waarvan de frequenties gehele veelvouden zijn van een fundamental. Een complexe periodic tone kan uit partials worden opgebouwd. Cycling '74 beschrijft additive synthesis precies als het combineren van sine oscillators waarbij de frequency en amplitude van elke partial afzonderlijk worden gecontroleerd.

Begin alleen met de fundamental. Voeg partial 2 toe, daarna 3, 4 en 5. Vraag na elke toevoeging: veranderde de pitch, of vooral de kleur van het geluid? Verander vervolgens de partial amplitudes. Zo wordt hoorbaar dat timbre kan worden behandeld als een distribution of energy, in plaats van als een benoemde preset.

6. Envelopes: geluid als gebeurtenis

Een voortdurend draaiende oscillator gedraagt zich nog niet echt als een muzikale of omgevingsgebeurtenis. Open patch 05. De ADSR envelope verdeelt een gesture in attack, decay, sustain en release. In Max genereert adsr~ dit control signal, dat met de oscillator wordt multiplieerd.

Vergelijk een attack van 5 ms met 1000 ms. Vergelijk een korte en een lange release. Houd pitch en waveform ongewijzigd. Dit toont dat identiteit niet alleen spectral is: temporal shape heeft een grote invloed op hoe een geluid wordt waargenomen.

??

Wat hadden we eigenlijk nodig om al deze geluiden te maken?

Heel weinig: oscillation, addition, multiplication en change over time.

Luistervoorbeelden / artistieke verbanden

Eliane Radigue - sustained electronic tones en uiterst trage spectral change. Nuttig om te bespreken hoe 'minimal' oscillator material zeer gedetailleerd luisteren kan ondersteunen.

Alvin Lucier - beating, resonance en acoustic phenomena vormen uitstekende bruggen tussen synthesis en physical sound.

Ryoji Ikeda - sine tones, pulses, thresholds en sterk gereduceerde electronic materials plaatsen frequency zelf op de voorgrond.

Pauline Oliveros - nuttig als tegengewicht: synthesis wordt betekenisvoller wanneer we leren aandachtig te luisteren naar kleine veranderingen, in plaats van alleen maar complexere patches te produceren.

LES 2 - Eenvoudige systemen complex maken

Centrale vraag: Wat gebeurt er wanneer eenvoudige processen elkaar beginnen te wijzigen, in plaats van rechtstreeks een ingewikkeld geluid te tekenen?

1. Subtractive synthesis

Open patch 06. Begin met een spectrally rich sawtooth en stuur die door een low-pass filter. Dit keert de logica van additive synthesis om. In plaats van componenten samen te stellen, begin je met een overvloed aan spectral energy en verwijder je een deel ervan. Cycling '74 beschrijft subtractive synthesis als het vormgeven van complexe signals met filters door energy weg te nemen.

Sweep de cutoff langzaam. Verander daarna de resonance. Leg cutoff uit als het gebied waar de attenuation begint en resonance als een accentuering rond de cutoff. Bekijk hoe het spectrum krimpt terwijl je luistert.

SOURCE -> FILTER -> AMPLIFIER. Leg uit dat deze architecture aan de basis ligt van een enorme hoeveelheid classic analogue synthesis. Voeg een envelope of LFO toe aan de filter en een static timbre wordt een bewegend timbre.

2. Modulation: een signal kan een signal controleren

Dit is het conceptuele middelpunt van les 2. Een oscillator hoeft niet rechtstreeks hoorbaar te zijn. Hij kan een control source worden. Wanneer hij traag loopt, noemen we hem een LFO; bij audio rate wordt het onderscheid tussen 'control' en 'sound' minder stabiel.

Open patch 07. Een trage sine verandert de amplitude van een carrier: tremolo. Verhoog de modulator frequency. Zodra de modulation het hoorbare bereik binnengaat, verschijnen nieuwe spectral components. Bij amplitude modulation wordt de amplitude van de carrier gevarieerd door een ander signal; bij audio rates ontstaan hierdoor sidebands die samenhangen met sommen en verschillen van frequencies.

Experiment

Houd de carrier op 220 Hz. Probeer modulation op 1, 4, 20, 110 en 220 Hz. Wanneer horen we niet langer 'een geluid dat luider en zachter wordt' horen, maar 'een ander timbre'.

3. Frequency modulation

Open patch 08. In plaats van amplitude te vermenigvuldigen, gebruik je één oscillator om voortdurend de frequency van een andere te veranderen. Bij een trage rate is het resultaat vibrato. Verhoog de modulation frequency tot in het audio range en het resultaat wordt spectral: extra components verschijnen rond de carrier.

Maak een duidelijk onderscheid tussen twee parameters: modulation frequency bepaalt hoe snel de carrier frequency wordt verplaatst; modulation depth bepaalt hoe ver die wordt verplaatst. De FM-introductie van Cycling '74 gebruikt precies dit onderscheid.

Demonstreer een carrier van 440 Hz met een modulator van 2 Hz en een bescheiden depth. Gebruik daarna modulation van 110 of 220 Hz en verhoog de depth. Vraag waarom een systeem dat alleen sine oscillators bevat plots metallic, bell-like of harsh kan klinken. Het antwoord is niet dat er een nieuwe waveform werd geselecteerd; complexiteit wordt gegenereerd door interaction.

4. Noise: signal zonder periodic repetition

Open patch 09. noise~ vormt een nuttig contrast met periodic oscillation. Filter broadband noise totdat die een gevoel van pitch of kleur begint te krijgen. Noise is bijzonder geschikt om subtractive synthesis te demonstreren, omdat er veel frequency content beschikbaar is om weg te nemen.

Schakel daarna de random control section in. Hier is randomness niet de sound source; ze verandert de filter cutoff. Dit onderscheid is belangrijk:

NOISE kan audio material zijn, terwijl **RANDOMNESS** een methode kan zijn om parameters te kiezen.

6. Van synthesiser naar systeem

Open patch 10. Een metro genereert events. Elk event kiest een pitch en filter cutoff, triggered een envelope en activeert een sawtooth/filter chain. De patch is bewust primitief. Het doel is de overgang zichtbaar te maken van een instrument met vaste controls naar een system met gedrag, behaviour.

Verander telkens slechts één rule: metro speed, pitch range, envelope length, filter range. Op welk moment begint de patch aan te voelen alsof hij een eigen character heeft? Dit vormt een bruikbare ingang naar generative music, cybernetic systems en later werk met sensor/electrode inputs.

Belangrijkste onderscheidingen

frequency != amplitude

waveform != envelope

fundamental != partial

sound source != control source

additive = construct spectrum; subtractive = remove from spectrum

slow modulation wordt vaak gehoord als movement; audio-rate modulation kan timbre worden

noise != randomness

complex gedrag vereist geen complexe onderdelen

Patch index

1. 01_sine_frequency_amplitude.maxpat - oscillator, Hz, amplitude, scope/spectrum
2. 02_waveforms_spectrum.maxpat - vergelijking van sine/saw/triangle/rectangle
3. 03_beating.maxpat - interference tussen nabijgelegen frequencies
4. 04_additive_partials.maxpat - vijf sine partials
5. 05_adsr_envelope.maxpat - temporal amplitude shaping
6. 06_subtractive_filter.maxpat - sawtooth door resonant low-pass filter
7. 07_amplitude_modulation.maxpat - tremolo naar audio-rate AM
8. 08_frequency_modulation.maxpat - vibrato naar audio-rate FM
9. 09_noise_random_control.maxpat - filtered noise plus random cutoff
10. 10_mini_generative_synth.maxpat - pitch, envelope, filtering en randomness

2 folders met meer uitgebreide voorbeelden om geluidssynthese in een muzikale context te gebruiken zoals techno of jungle.

Verdere Max referenties

Cycling '74, MSP Tutorials: <https://docs.cycling74.com/learn/series/msp-tutorials/>

Relevante sections: Basics; Additive and Modulation Synthesis; Filters and Subtractive Synthesis; Dynamics; Analysis.
Object references die hier worden gebruikt zijn onder meer cycle~, saw~, phasor~, adsr~, noise~, lores~, spectroscope~, line~, *~ en +~.