

**MAX-OHJELMALLA TOTEUTETTU
MUSIIKKITEKNOLOGIAN
PERUSTEIDEN OPPIMATERIAALIKOKONAISUUS**

Tapio Lappalainen

Otm Suurprojekti (Opinnäyte)

Musiikkiteknologian aineryhmä

Musiikkikasvatuksen, jazzin ja kansanmusiikin osasto

Taideyliopiston Sibelius-Akatemia

syksy 2014

TIIVISTELMÄ

Musiikkiteknologian merkitys musiikin tuottamisessa on korostunut viime vuosikymmenien aikana. Tämän on mahdollistanut tarvittavan kaluston ja ohjelmistojen huomattava halpeneminen. Näin ollen sitä voidaan opettaa jo ihan perustasolla.

Tässä opinnäytetyössä kehitetään oppimateriaalikokonaisuus musiikkiteknologian opettamista varten erityisesti musiikkiteknologisen kehityksen varhaisvaiheissa oleville henkilöille Max-ohjelmaa käyttäen. Oppimateriaalikokonaisuus sisältää haastetta myös pidemmälle ehtineille. Toinen vaihtoehto käytettäväksi ohjelmaksi oli Pure Data -ohjelma. Molemmat ohjelmat täyttävät adaptiivisuus vaatimuksen, joka oli tärkeä valintaperuste. Pure Data -ohjelma kuitenkin tarjoaa käyttäjälle heikompaa tukea, joten valinta kohdistui Max-ohjelmaan. Lisäksi Max -ohjelma on yhteensopiva Ableton Live -ohjelman kanssa, mitä Pure Data ei ole. Ohjelman käyttämiseen riittää tavallisen kotitietokoneen suorituskyky.

Työssä esitellään musiikkiteknologian osa-alueita yksityiskohtaisesti sekä pohditaan alan opetusfilosofiaa. Sen jälkeen kuvataan Max-ohjelman ominaisuuksia ja sen käyttöä opetuksen apuvälineenä. Tehtäväpatchit ovat työn keskeinen sisältö, joissa opiskelijalle esitellään ääniopin käsitteitä, äänisynteesimetodien perusteita sekä tarkastellaan efekti- ja prosessointilaitteiden toimintaperiaatteita. Työhön liitetty oppimateriaalikokonaisuus muodostuu 40 patchistä, joita voidaan sellaisenaan käyttää hyväksi opetustilanteissa.

Musiikkiteknologia-käsite määritetään erilaisten musiikkiteknologisten työvaiheiden näkökulmasta. Musiikkiteknologian opetustapoja vertaillaan soitinopetuksen opetustapoihin. Opinnäytetyössä pohditaan instrumenttikäsitteen määritelmää. Instrumentti-käsite on pilkottu sen komponentteihin, jonka seurauksena sähköiset soittimet ja musiikkiteknologiset laitteet on helpompi nähdä instrumentteina (= työkaluina) musiikin rakenteiden luomiseen, äänilähteiden rakentamiseen, musiikin esittämiseen ja musiikin/äänien taltioimiseen erilaisille medioille.

Kuvatun ohjelman toimivuutta on tutkittu lukuvuoden 2013-2014 aikana Jokilaaksojen musiikkiopiston oppilaiden kyselytutkimuksena, johon he osallistuivat aktiivisesti. Heidän reaktionsa olivat voittopuolisesti myönteisiä.

ASIASANAT: Efektilaite, graafinen ohjelmointi, instrumentti, ohjelmointi, oppimateriaali, musiikkiteknologia, musiikkiteknologian perusteet, musiikkiteknologian opettaminen, äänioppi, äänisynteesi

ABSTRACT

The importance of music technology to the making of music has increased during recent decades. The cost of the necessary equipment and computer software has reduced dramatically, and this trend has brought teaching music technology to beginners within reach.

This dissertation reports on a project to develop, using the Max program, study material for a music technology program specially designed for individuals in their early stages of music technological development. The study material provides a challenge also to more advanced students. The Pure Data -program (PD) was considered as an alternative to Max. Both programs are adaptive, which is an important feature. However, Pure Data provides weaker user support, leading me to choose Max. Moreover, Max, unlike PD, has the additional advantage of being compatible with the Ableton Live -program. Regular home computers provide plenty enough power to execute these programs.

In the dissertation, different sectors of music technology are introduced in detail and teaching philosophy is pondered in relation to them. Properties of the Max program are described, along with its application to the role in assisting in the teaching process. At the heart of this dissertation are the Max patches which form assignments for the students, introducing them to concepts of acoustics, basic sound synthesis methods and the working principles of effects units and sound processors. The study material includes 40 patches which can be used as class assignments.

The concept of music technology is defined in terms of the different work phases of music technology. Methods of teaching music technology are compared to methods of teaching traditional instruments. The concept of instrument is pondered and defined in detail. A novel definition of instrument makes it easier to consider electric instruments and electronic devices as instruments/tools for creating musical structures, for building sound sources, for musical performance and for recording music and sound for different types of media.

The study material was tested during the academic year 2013-14 and feedback was solicited from the students of the Jokilaaksot Music Institute using a questionnaire. Students participated in this study actively and their response was in the main positive.

KEY WORDS: Acoustics, basics of music technology, effect processors, graphical programming, instrument, instruction of music technology, programming, music technology, study material, sound synthesis

SISÄLLYSLUETTELO

TIIVISTELMÄ

ABSTRACT

SISÄLLYSLUETTELO

1 ALKULAUSE	1
2 SUOMENKIELINEN TERMINOLOGIA	2
3 JOHDANTO	3
4 OPPIMATERIAALIKOKONAISUUDEN KOHDERYHMÄ	5
5 MUSIIKKITEKNOLOGIA	6
5.1 Musiikkiteknologian määrittelevä	6
5.2 Musiikkiteknologian osa-alueita	8
5.2.1 Säveltämiseen liittyvä teknologinen toiminta	8
5.2.1.1 Sävellysalgoritmit	8
5.2.1.2 Sekvensseriohjelmat	8
5.2.2 Nuotintaminen teknologian avulla	8
5.2.3 Esityksen johtaminen teknologialla	9
5.2.4 Sähköisten soittimien ja äänilähteiden rakentaminen	9
5.2.4.1 Analogiaelektroniikka	9
5.2.4.2 Äänisynteesiohjelmat	10
5.2.4.3 Mikrokontrollerialustat	10
5.2.5 Esityksen toteuttaminen teknologialla	10
5.2.5.1 Sähköiset soittimet	10
5.2.5.2 Improvisointi elektronisella laitteistolla	10
5.2.5.3 Sekvensseriohjelmat	11
5.2.5.4 Elektroakustinen äänidiffuusio	11
5.2.5.5 Ääni-installaatiot	11
5.2.5.6 Mikrokontrollerialustat	12
5.2.6 Tallenteen toteutus	12
5.2.6.1 Äänittäminen	12

5.2.6.2 Miksaaminen	12
5.2.6.3 Masterointi	13
5.2.7 Äänentoisto	13
5.2.7.1 Äänentoisto	13
5.2.7.2 Elektroakustinen äänidiffuusio	14
5.2.8 Musiikkisovellusten ohjelmointi	14
5.3 Musiikkiteknologian spesialisti tai moniosaaja	15
6 MUSIIKKITEKNOLOGIAN HOLISTINEN OPETTAMINEN	17
7 INSTRUMENTTI JA MUUSIKKOUS	19
7.1 Instrumentin määrittely	19
7.2 Instrumentin komponentit	20
7.2.1 Instrumenttien erilaiset käyttöliittymät	20
7.2.2 Mekanismi käyttöliittymän ja värähtelyn sytyttäjän välillä	21
7.2.3 Erilaisia värähtelyn sytyttäjiä	22
7.2.4 Erilaisia värähtelijöitä	22
7.3 Automaatti-instrumentin komponentit	22
7.4 Muusikon toiminnasta	23
7.4.1 Vuorovaikutus ihmisen ja instrumentin käyttöliittymän välisessä rajapinnassa	24
7.4.2 Muusikon fyysinen toiminta	24
7.4.3 Muusikon auditiivinen toiminta	24
7.5 Musiikkoperheen uusi jäsen	24
8 MAX-OHJELMA	26
8.1 Max-ohjelman taustaa	26
8.2 Ohjelmointi Max-ohjelmalla	26
8.3 Max-ohjelman edut opetuskäytössä	26

9 OPPIMATERIAALIKOKONAISUUDEN SISÄLTÖKATEGORIAT	28
9.1 Ääniopin ja musiikkiteknologian peruskäsitteitä esittelevät patchit ja niiden oppimistavoitteet	29
9.2 Äänisynteesimetodien perusteita esittelevät patchit ja niiden oppimistavoitteet	29
9.3 Efekti- / prosessointilaitteiden toimintaperiaatteita esittelevät patchit ja niiden oppimistavoitteet	30
9.4 Muut patchit	30
10 TEHTÄVÄPATCHIT	31
10.1 Tehtäväpatchien käyttöajatus	31
10.2 Toistuvat objektirakenteet	31
10.2.1 Audio on/off -rakenne	31
10.2.2 Spektrianalysaattori -rakenne	32
10.2.3 Audiosignaalin monitorointi	33
10.2.4 Sämplerin käyttö -rakenne	33
10.2.5 Verhokäyrän toteutus -rakenne	33
10.2.6 Äänilähteiden spektrin suodatus -rakenne	35
10.3 Johdatus Max-patchien rakentamiseen	35
10.4 Kirjalliset kuvaukset tehtäväpatchien toiminnasta	37
10.4.1 Ääniopin ja musiikkiteknologian peruskäsitteitä esittelevät patchit	37
1. Siniääni	37
2. Ääniopin ja musiikkiteknologian peruskäsitteitä	38
3. Erilaisia aaltomuotoja	39
4. Kohinatyyppejä	41
5. Vibrato	42
6. Verhokäyrä	43
7. Vaihe	44
8. Ylä-äännessarja	46
9. Äänennopeus	48
10. Näytteenottotaajuus-amplitudiresoluutio	49
10.4.2 Äänisynteesimetodien perusteita esittelevät patchit	51
11. Aaltotaulukkosynteesi (<i>wavetable synthesis</i>)	51
12. Lisäävä synteesi 1 (<i>additive synthesis</i>)	53
13. Lisäävä synteesi 2	54

14. Vähentävä synteesi 1 (<i>subtractive synthesis</i>)	55
15. Vähentävä synteesi 2	56
16. Suotimet	57
17. Taajuuskorjaimet	58
18. Pientaajuusoskillaattori (<i>low frequency oscillator</i>)	60
19. Unipolaari- ja bipolaarisignaali	62
20. Amplitudimodulaatio (<i>amplitude modulation</i>)	63
21. Rengasmodulaatio (<i>ring modulation</i>)	64
22. Rengas- ja amplitudimodulaation vertailu	65
23. Tremolo	67
24. FM-synteesi (<i>FM-synthesis</i>)	67
25. Raesynteesi (<i>granular synthesis</i>)	69
26. Aaltomuotoilusynteesi (<i>waveshaping</i>)	74
27. Kahden aaltotaulukon ristiinfeidaus (<i>Wavetable crossfading</i>)	76
 10.4.3 Efekti- / prosessointilaitteiden toimintaperiaatteita esittelevät patchit	 78
28. Viive 1 (<i>delay</i>)	78
29. Viive 2	79
30. Läpäsähdysviive (<i>slapback delay</i>)	80
31. Viiveillä toteutettu pieni tila	81
32. Kaiku (<i>reverb</i>)	83
33. Koorus (<i>chorus</i>)	84
34. Flangeri (<i>flanger</i>)	86
35. Feiseri (<i>phaser</i>)	87
36. Wah-wah	89
37. Särö (<i>distortion</i>)	90
 10.4.4 Muut patchit	 91
38. Erilaiset johtotyytit	91
39. Monitorointirakenteita	93
40. Ulkoisen kontrollerin kytkeminen	94
 11 LOPPUPÄÄTELMÄT	 96
 LÄHTEET	 101
 LIITTEET	 103
Liite 1 Max-ohjelmalla toteutettu musiikkitekniikan perusteiden oppimateriaalikonaisuus	

1 ALKULAUSE

Haluan esittää lämpimät kiitokset seuraaville henkilöille erittäin asiantuntevista kommentteista liittyen asiasisältöön: professori Andrew Bentley (*ohjaaja*) ja professori Pentti Lappalainen. Seuraavien henkilöiden korjaukset suomenkielen sisällöstä olivat myös hyvin tärkeitä: professori Andrew Bentley, professori Pentti Lappalainen, apulaisrehtori/viulunsoiton opettaja Kristiina Penttinen sekä säveltäjä/musiikin perusteiden lehtori Sanna Ahvenjärvi. Lisäksi professori Vesa Välimäki ja yliassistentti Kalev Tiits antoivat sähköpostitse edeltävien henkilöiden lisäksi arvokkaita musiikkiteknologiatermien suomentamiseen liittyviä kommentteja. Kiitokset myös Jokilaaksojen musiikkiopiston musiikkiteknologiaryhmien oppilailleni: Anttu, Dicle, Elena, Elli-Maarit, Emilia, Hanna, Helka, Hilda, Julia, Jutta, Miska, Miro, Riina, Tarvo, Tuukka ja Ville, joilta sain tärkeää palautetta patcheistä, kun niitä koekäytettiin lukuvuoden 2013-2014 aikana musiikkiteknologian tunneilla.

Haluan myös kiittää Taideyliopiston Sibelius-Akatemiaa ja sen musiikkiteknologian aineryhmää antamastaan tuesta.

2 SUOMENKIELINEN TERMINOLOGIA

Opinnäytetyön suomenkielinen terminologia tukeutuu Aalto-yliopiston teknillisen korkeakoulun audiosignaalinkäsittelyn professori Vesa Välimäen ylläpitämään internetsivustoon audiosignaalinkäsittelyn sanastosta. Sivuston linkki:

<http://users.spa.aalto.fi/vpv/ask-sanasto.htm>

Kävin myös lukuisia mielenkiintoisia keskusteluja ja sähköpostikeskusteluja suomenkielisistä termeistä, mitkä myös vaikuttivat merkittävästi käyttämäni suomenkielisten termien muotoutumiseen. Paine käyttämäni termien suomentamiselle tulee siitä, että omat musiikkiteknologian oppilaani Jokilaaksojen musiikkiopistossa voivat olla niin nuoria, että heidän englanninkielen taitonsa ovat vielä hyvin vaatimattomat. Koska musiikkiteknologian opettamisen historia lapsille ja nuorille Suomessa on vielä hyvin lyhyt, on aika mielestäni suomentaa näitä alan perustermejä.

Suomen kieleen on lainattu runsaasti sanoja muista kielistä siten, että vieraskielisiin sanoihin on laitettu i-kirjain perään. Esimerkkeinä tästä ovat: seminar - seminaari, hertz - hertsi, manual - manuaali ja president - presidentti. Tällä samalla logiikalla istuu suomenkielisiksi sanoiksi käyttämäni termit, flanger - flangeri sekä phaser - feiseri. Chorus - efektin suomentaminen kuoro-efektiksi toimii erinomaisesti, mutta *“kentällä”* tilanne on kuitenkin se, että *“kaikki”* puhuvat kooruksesta. Koen, että choruksen kirjoittaminen kooruksena on yhtä luonteva kuin controller-sanan kirjoittaminen kontrollerina. Sampleri-sanan kirjoittamisen ä-pisteiden kera sämplerinä perustelen samansukuisen brändi-sanan yleistyneellä kirjoitusasulla. Slapback delayn suomennos läpsähdysviive kuvaa hyvin läpsähdysviiveen äänellisen olemuksen.

3 JOHDANTO

Tässä opinnäytetyössä kehitetään oppimateriaalikonaisuus musiikkiteknologian opettamista varten erityisesti musiikkiteknologisen kehityksen varhaisvaiheissa oleville henkilöille Max-ohjelmaa käyttäen *hands-on* -periaatteella. Tehtävien päämääränä on toimia matalankynnyksen väylänä musiikkiteknologian maailmaan. Opetusmateriaali ei ole tarkoitettu ensisijaisesti oppikirjaksi, vaikkakin patch¹-kuvausteksteistä asiatietoa löytyykin. Kohderyhmästä johtuen on teksti pyritty pitämään riittävän selkeänä ja helppolukuisena.

Minulla oli useampia syitä oppimateriaalikonaisuuden tekemiseen. Tarvitsin musiikkiteknologian oppilailleni sopivaa opetusmateriaalia, tarvitsin aiheen opinnäytetyöhöni, halusin kehittyä ohjelmoinnissa Max-ohjelmalla sekä olin kiinnostunut etsimään vastausta kysymykseen *“Mitkä ovat kaikki äänisynteesimetodit?”*. Äänisynteesimetodien kartoituksen koen jatkeeksi aiempien sävellysosointojeni yhteydessä tapahtuneeseen akustisten instrumenttien *“kaikkien eri sointivärien”* kartoituspyrkimykseen.

Olen opettanut musiikkiteknologiaa Jokilaaksojen musiikkiopistossa säännöllisesti vuodesta 2008 alkaen. Musiikkiteknologiatuntien käytännön tarve tehtäville kannusti minua tekemään tämän oppimateriaalikonaisuuden. Tehtävien myötä oppilaasta tulee aktiivinen toimija, joka alkaa miettimään, lukemaan, ajattelemaan ja hankkimaan tietoa käsillä olevasta asiasta. Lapsille ja nuorille suunnatussa musiikkiteknologian opetuksessa luentopohjaisen opetuksen vaarana on se, että asiat, runsaine vierasperäisine ammattitermeineen, voivat helposti olla liian vaikeita ymmärrettäviksi.

Lapsille ja nuorille suunnatussa musiikkiteknologian opetuksessa tehtävät ovat opetuksen alkupiste. Tehtävänannon jälkeen selvitykset asiasta tulevat välittömänä seurauksena. Tehtäviä ei saa tehtyä ilman, että asioita selitetään. Tehtävien tekemisen yhteydessä tapahtuva asioiden selittäminen toimii oppimisen tukena.

Suuri osa musiikkiteknologiakirjallisuudesta, jota kävin läpi tätä oppimateriaalikonaisuutta suunnitellessa, oli oppikirjoja. Asiatietoa on tarjolla nykyään runsaasti. Tehtäviä näissä kirjoissa oli yllättävän vähän. Eräässä kirjassa oli runsaasti tehtäviä, mutta suuri osa kysymyksistä oli sopivia vain korkeakoulutasoisille opiskelijoille. Miksi musiikkiteknologian oppimateriaalia vasta-alkajille on tarjolla varsin vähän johtune mm.

¹ Patch-käsite on englanninkielinen Max-ohjelman termi, jolla tarkoitetaan objekteista rakennettavaa kytkentäkokonaisuutta.

siitä, että laitteisto on vielä muutama vuosikymmen aiemmin ollut erittäin arvokasta verrattuna tämän päivän tietokoneisiin ja niihin saataviin musiikkiohjelmiin. Tänä päivänä kotitietokone muutamine lisälaitteineen ja musiikkiohjelmineen mahdollistaa kenelle tahansa ominaisuuksia, joista n. 30 vuotta sitten pienen kotistudion omistaja ei voinut edes haaveilla. Toinen syy lienee se, että aika ei ollut vielä kypsä ajatukselle, että esim. lapset tekisivät äänityksiä. Levyttäminen saatettiin ajatella olevan vain huippumuusikoiden oikeus.

Tehtäväpatchit on tehty Max-ohjelmalla, koska sitä käytettäessä on mahdollista pitää näkyvissä vain opetettavan asian ydin. Max adaptoituu erinomaisesti kulloinkin opetettaviin asioihin. Liitteessä olevien tehtäväpatchien alaosiin on kirjattu nettilinkkejä, joiden avulla oppilaan on tarkoitus päästä helposti aiheeseen liittyvän tiedon äärelle.

Musiikkiteknologian opettamiseen olen ottanut esikuvaksi instrumenttiopetuksen holistisen opetustavan, jossa yhtä aikaa tehdään, kuunnellaan ja ajatellaan. Ollakseni tietoinen siitä mitä olen opettamassa koin että minun pitää jäsentää, mistä musiikkiteknologian kokonaisuudessa on kyse. Tämän jäsennyksen musiikkiteknologian osa-alueista tein työvaiheiden näkökulmasta.

Jotta olen voinut tehdä vertailua musikkiteknologisten laitteiden ja akustisten instrumenttien välillä, olen tehnyt yksityiskohtaisen jäsennyksen ja määrittelyn siitä, mitä instrumentti-käsitteellä tarkoitetaan.

Tehtävämonisteet jäsentyvät neljään eri kategoriaan: Ääniopin ja musiikkiteknologian peruskäsitteitä esittelevät patchit, äänisynteesimetodien perusteita esittelevät patchit, efekti- / prosessointilaitteiden toimintaperiaatteita esittelevät patchit sekä muut patchit.

Midi-protokollaa ei ole käsitelty tässä opinnäytetyössä juuri lainkaan, koska midistä on todennäköisesti kirjoitettu jo lähes kaikki mahdollinen Midin reilun kolmenkymmenen olemassaolovuoden aikana. Lisäksi midi on vanhenevaa teknologiaa.

4 OPPIMATERIAALIKOKONAISUUDEN KOHDERYHMÄ

Kohderyhmään kuuluvat kaikki musiikkiteknologisen kehityksensä varhaisvaiheissa olevat henkilöt. Tehtäviä voi käyttää musiikkiteknologian peruskursseilla eri tason musiikkioppilaitoksissa. Lisäksi materiaali soveltuu myös itsenäiseen opiskeluun ilman opettajan ohjausta. Kokonaisuus saattaa kiinnostaa muitakin musiikkiteknologian oppimateriaaleista kiinnostuneita, esimerkiksi niitä, jotka tekevät päätöksiä opetussisällöstä sekä opetuksen toteutuksesta.

5 MUSIIKKITEKNOLOGIA

5.1 Musiikkiteknologian määrittäminen

Musiikkiteknologia-käsite voidaan määritellä erilaisina työvaiheina, joita teknologisilla laitteilla tehdään. Kun laitteistolla tehdään jotain mikä liittyy suoraan tai välillisesti musiikkiin, ääneen tai nuottikuvaan on toiminta musiikkiteknologista toimintaa. Jäsennystapa on lähtökohtaisesti musiikkityyleistä riippumatonta, vaikkakin tapauskohtaiset esimerkit erilaisista työtavoista ovatkin kytköksissä tiettyihin musiikkityyleihin, usein sen takia, että ne ovat syntyneet tietyn musiikkityylin toteutuksen tarpeista.

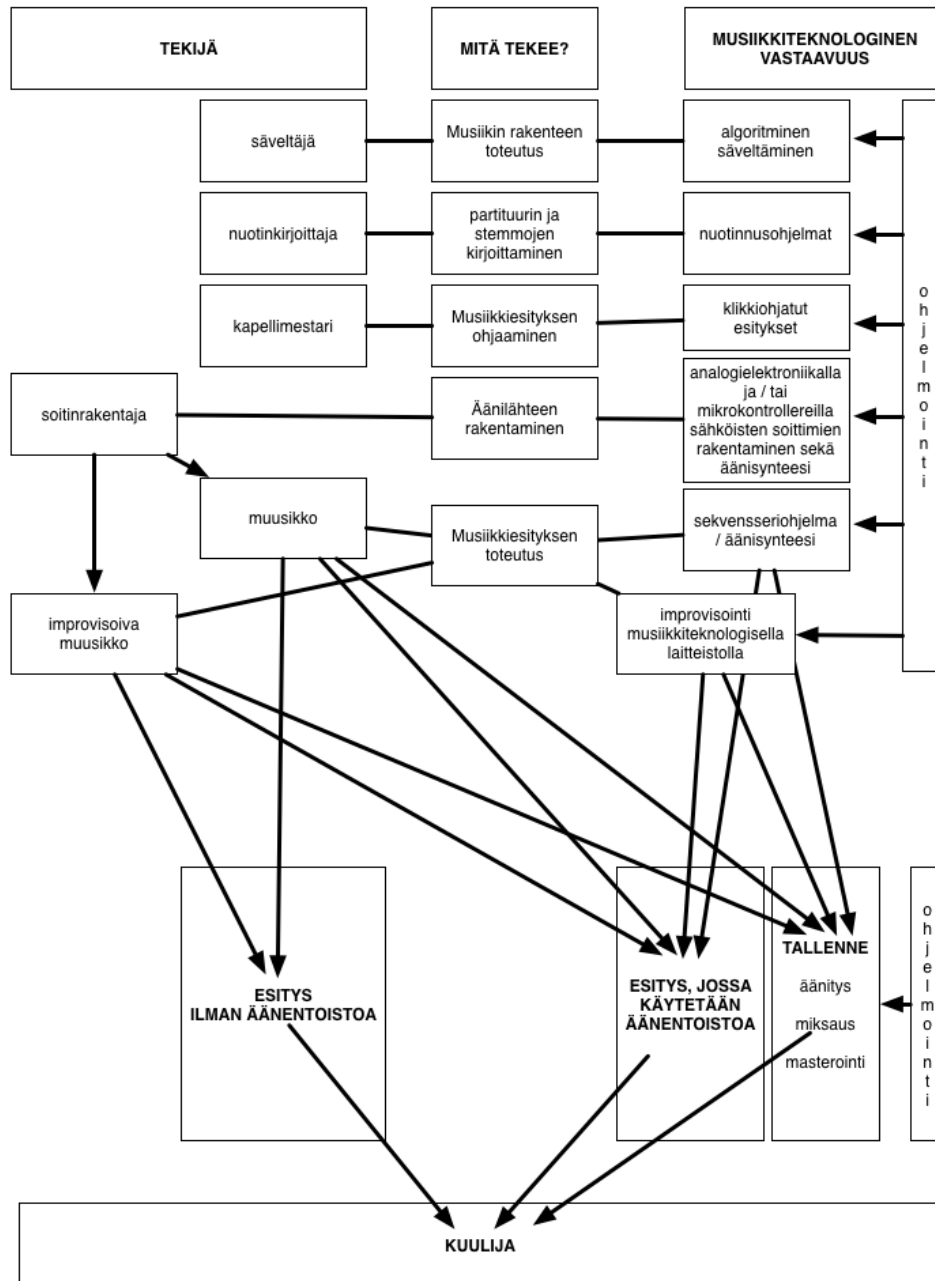
Yksi luonteva tapa jäsentää musiikkiteknologiaa on käydä läpi perinteiset musiikin teon vaiheet ja niiden tekijät ja tehdä niistä rinnastukset musiikkiteknologiaan. Yleisellä tasolla musiikin teosta voidaan todeta seuraavaa. Joku keksii, miettii ja suunnittelee musiikillisen kokonaisuuden rakenteen, *säveltäjä*. Jotta idea musiikista saadaan esitettyä jonkun pitää mahdollisesti kirjoittaa kirjalliset ohjeet (=nuotit) esitystä varten siitä mitä soitetaan ja miten, *nuotinkirjoittaja*. Joku tai jotkut toteuttavat esityksen näistä musiikillisista ideoista, *muusikot*. Esittäjiä saattaa olla läsnä niin paljon, että tarvitaan erillinen henkilö ohjaamaan esitystä ja harjoituksia, *kapellimestari*. Lisäksi jonkun pitää vielä rakentaa työkalut joilla esittäjät toteuttavat esityksen, *soitinrakentaja*.

Kuvan 1 kaavio alkaa näillä perinteisillä musiikin ja musiikillisten työkalujen teon vaiheilla (*säveltäminen, nuotin kirjoittaminen, esityksen ohjaaminen, soitinrakennus ja esittäminen*) ja niiden musiikkiteknologisilla rinnastuksilla, joiden perässä on vain sähköisinä olemassa olevat työvaiheet (*äänentoisto, äänitys, miksaus ja masterointi*). Kaavion oikeassa reunassa mainittu ohjelmointi voi liittyä periaatteessa mihin tahansa musiikilliseen työvaiheeseen. Tietyissä tilanteissa ohjelmointi voi olla ainoa mahdollisuus toteuttaa haluttu asia, ei kuitenkaan aina nopein ja kätevin tapa.

Vaihtoehdot musiikin kuuntelulle ovat live-esitys ilman äänentoistoa, live-esitys, jossa käytetään äänentoistoa ja musiikin kuuntelu tallenteelta kaiuttimien tai kuulokkeiden kautta. Äänentoiston käyttö esitystilanteessa voi tarkoittaa minimissään yksittäisen hiljaisen instrumentin vahvistamista muiden soittimien rinnalle, kaikkien soittimien vahvistamista tai jotain edeltävien ääripäiden väliltä. Esityksessä voidaan myös toistaa tallennetta yhtenä tai ainoana esityksen äänilähteistä, esim. sämplerit, nauhateokset, elektroakustinen äänidiffuusio

- tai playback-esitykset. Näitä tekstissä hahmoteltuja erilaisia välimuotoja ei kaaviossa havainnollisteta.

Moneen työvaiheeseen on listattu kyseisen työvaiheen keskeiset työkalut.



KUVA 1. Musiikin tekemiseen liittyvät perinteiset työvaiheet ja niiden musiikkiteknologiset vastaavuudet, sekä vain sähköisinä esiintyvät vaiheet.

5.2 Musiikkiteknologian osa-alueita

5.2.1 Säveltämiseen liittyvä teknologinen toiminta

5.2.1.1 Sävellysalgoritmit

Algoritmisella säveltämisellä tarkoitetaan sävellystekniikkaa, jossa algoritmeja käytetään musiikin luomiseen. Sävellysalgoritmi tekee valintoja tekijän määrittelemien musiikillisten sääntöjen ja lainalaisuuksien puitteissa generoidessaan musiikin rakenteita. Kun sävellysalgoritmi generoi rakennetta vain yhteen tai muutamaa musiikilliseen parametriin, puhutaan tietokoneavusteisesta säveltämisestä. Sävellysalgoritmit automatisoivat siis säveltäjän työtä.

Esimerkkejä ohjelmista ja ohjelmointikielistä sävellysalgoritmien tekoon ovat mm. Calmus, Common Lisp Music, Csound, Max, Open Music, PWGL, Slippery Chicken ja Super Collider.

5.2.1.2 Sekvensseriohjelmat

Sekvensseriohjelmat eivät generoi musiikillista rakennetta, mutta niiden avulla sävelletään musiikkia. Sekvensseriohjelma mahdollistaa säveltäjälle hänen musiikillisista ideoistaan enemmän tai vähemmän realistisen kuulohavainnon. Sekvensseriohjelmat mahdollistavat musiikillisen idean toteuttamisen soivaan muotoon ohjelmistosyntetisaattoreiden ja sämplereiden avulla. Nuotinnusohjelmat sisältävät usein myös mahdollisuuden musiikin kuunteluun, joten niitä voidaan käyttää sekvensseriohjelman kaltaisesti säveltämisen apuvälineinä.

Esimerkkejä sekvensseriohjelmista ovat mm. Ableton Live, Cubase, Digital Performer, Live, Logic Pro, Protools ja Sonar.

5.2.2 Nuotintaminen teknologian avulla

Kynällä nuottien kirjoittaminen sekä tietokonenuotintaminen ovat molemmat edelleen käsityötä. Tietokoneet tosin mahdollistavat muusikon soiton nuotintamisen reaaliaikaisesti midi-instrumenttien avulla. Yleensä tässä vaihtoehdossa ohjelmat nuotintavat vain

sävelkorkeudet ja aika-arvot, jolloin dynamiikkamerkkien, soitossa tapahtuneiden epätarkkuuksien korjaaminen sekä muiden esitysmerkkien sisällyttäminen nuottiin jatkuu käsityönä.

Esimerkkejä nuotinnusohjelmista ovat mm. Finale, Forte, Igor Engraver, LilyPond, MagicScore, Muse Score ja Sibelius.

5.2.3 Esityksen johtaminen teknologialla

Klikkiohjatut esitykset automatisoivat kapellimestarin työtä. Klikkiohjatuissa esityksissä muusikoille laitetaan yleensä vain toiseen korvaan kuuloke, josta muusikko kuulee esiohjelmoidun klikin. Moni rytmimusiikin levy äänitetään käyttäen klikkiä. Elokuvamusiikkiäänityksissä, joissa kokoonpanona on sinfoniaorkesteri, on tyypillistä tehdä orkesteriäänitykset klikin kanssa, jotta musiikin jaksojen kestot toteutuvat ennalta määrätyn pituisina. Äänen synkronoiminen kuvan kanssa voi olla erityistä tarkkuutta vaativaa työtä. Klassisesta musiikista löytyy teoksia, joissa yksittäisillä muusikoilla voi olla henkilökohtaisia klikkiraitoja. Tällöin klikkiraidat ohjaavat teoksen ajallista synkronointia, ikään kuin läsnä olisi useampi kapellimestari.

Klikkiraidat voidaan toteuttaa esimerkiksi seuraavanlaisella laitteistolla: tietokone, sekvensseriohjelma, äänikortti, kuunteluvahvistin ja kuulokkeita.

5.2.4 Sähköisten soittimien ja äänilähteiden rakentaminen

5.2.4.1 Analogiaelektroniikka

Esimerkkejä sähköisistä soittimista, jotka on rakennettu analogiaelektroniikalla ovat mm. Dynamophon/Telharmonium, Hammond-urku, Moog-syntetisaattori, Ondes Martenot, Sphärophon, Theremin ja Trautonium (Lindeman 1980: 12-18).

Keskeiset komponentit sähköisten soittimien rakentamisessa analogiaelektroniikalla ovat vastus, kondensaattori, kelat, diodit, transistorit (FET, BJT), operaatiovahvistimet, putket, regulaattorit, erilaiset lineaaripiirit ja kiertokytkin.

5.2.4.2 Äänisynteesiohjelmat

Virtuaalisia soittimia/äänilähteitä voidaan rakentaa tietokoneohjelmoinnilla. Tuloksena syntyvät ohjelmat pohjautuvat jollekin äänisynteesimetodille. Ks. nro 10.4.2 Äänisynteesimetodien perusteita esittelevät patchit ja niiden kuvaustekstit.

5.2.4.3 Mikrokontrollerialustat

Esimerkkejä mikrokontrollerialustoista joita käytetään musiikin ja ääni-installaatioiden esityksissä ovat mm. Arduino, Raspberry pi, BeagleBone Black ja UDOO Dual.

Esimerkiksi Arduino on avoimeen laitteistoon perustuva mikrokontrollerialusta ja ohjelmointiympäristö. Arduinon pinneihin voidaan kytkeä erilaisia sensoreita, moottoreita, LED-valoja ja muita komponentteja. Laitteistoa ohjelmoidaan C++:aan perustuvalla Arduino-ohjelmointikielellä (Arduino 2014).

5.2.5 Esityksen toteuttaminen teknologialla

5.2.5.1 Sähköiset soittimet

Musiikin esittämistä varten suunniteltuja sähköisiä soittimia ovat mm. Dynamophon/Telharmonium, Hammond-urku, Moog-syntetisaattori, Ondes Martenot, Sphärophon, sähköbasso, sähkökitara, Theremin ja Trautonium.

Sähköisistä soittimista löytyy samat komponentit kuin perinteisistäkin soittimista eli: käyttöliittymä, mekanismi käyttöliittymän ja värähtelyn sytyttäjän välillä, värähtelyn sytyttäjä ja värähtelijä. Mekanismia käyttöliittymän ja värähtelyn sytyttäjän väliltä ei sähköisistä soittimista juurikaan löydy. Käyttöliittymät vaihtelevat lähes soittimittain. Värähtelijänä on useimmiten kaiutin. Värähtelyn sytyttäjänä toimii vaihtojännite, jota ohjataan käyttöliittymästä käsin. Ks. myös nro 6.1 instrumentin määrittely ja nro 6.2 instrumentin komponentit.

5.2.5.2 Improvisointi elektronisella laitteistolla

Elektronisen improvisointilaitteiston keskeiset osat ovat: käyttöliittymä(t), äänigeneraattori(t) ja värähtelijä(t). Käyttöliittymät ovat erilaisia kontrollereita, kuten esimerkiksi erilaiset

keyboardit ja pädit, Monome, Miburi, Radio Baton, Reactable, Space Palette, SLABS ja Tenori-On (Linn 2011). Äänigeneraattoreita ovat mm. soundi-modulit, ohjelmistosyntetisaattorit ja sämplerit. Värehtelevä äänilähteenä on yksi tai useampi kaiutin.

5.2.5.3 Sekvensseriohjelmat

Sekvensseriohjelmat automatisoivat muusikon työtä. Sekvensseriohjelmat mahdollistavat musiikillisen idean toteuttamisen soivaan muotoon ohjelmistosyntetisaattoreiden ja sämplereiden avulla. Sekvensseriohjelmia käytetään mm. elokuvamusiikkien demo-versioiden toteuttamiseen. Demo-versiot mahdollistavat sen, että elokuvan ohjaaja ja tuottaja voivat saada jonkinlaisen kuvan elokuvaan tulevista musiikeista ennen kuin tehdään päätös musiikkien äänittämisestä todellisella orkesterilla.

Esimerkkejä sekvensseriohjelmissa ovat mm. Ableton Live, Cubase, Digital Performer, Logic Pro, Protools ja Sonar.

5.2.5.4 Elektroakustinen äänidiffuusio

Elektroakustinen äänidiffuusio -käsitteellä tarkoitetaan yleensä kaksikanavateoksen esittämistä kaiutinorkesterin välityksellä. Kaiutinorkesteri koostuu kymmenistä kaiuttimista. Näitä ovat mm. Gmebaphone/Cybernophone, Acousmonium ja BEAST -järjestelmä. Acousmoniumissa on 80 eri kokoista ja muotoista kaiutinta, BEAST -järjestelmä (*Birmingham ElectroAcoustic Sound Theatre*) voi sisältää yli 100 kaiutinta.

5.2.5.5 Ääni-installaatiot

Ääni-installaatio voidaan määrittää olemaan musiikillinen vastine veistokselle, auralinen kokemus yleensä kuvataiteen näyttelytilaan tai tiettyyn arkkitehtoniseen tilaan suunnitellusta teoksesta. Monet ääni-installaatiot ovat hyödyntäneet elektroakustisen musiikin tekniikoita ja teknologioita niiden luomisessa ja esityksissä. Ääni-installaation kuunteleminen voidaan yleensä aloittaa mistä tahansa satunnaisesta kohdasta ja lopettaa minä tahansa satunnaisena ajankohtana (Sound Installation 2014).

5.2.5.6 Mikrokontrollerialustat

Mikrokontrollerialustoilla työskentelevä henkilö usein paitsi rakentaa itselleen soittimen, toimii myös esittävänä taiteilijana. Ks. myös nro 5.2.4.3 Mikrokontrollerialustat.

5.2.6 Tallenteen toteutus

5.2.6.1 Äänittäminen

Äänityksessä taltioidaan äänilähteiden tuottamat äänet tallentimelle. Äänityksessä tarvittavia laitteita ovat mikrofoni, etuvahvistimet, AD-muuntimet mikäli kyseessä on digitaalitallennin, analogiset tai digitaaliset tallentimet sekä tarpeelliset johdot laitteiden väliin. Pienikokoisessa käsitalentimessa on edellämainitut osat sisällytetty samaan laitteeseen. Äänitysvaiheessa voidaan käyttää myös seuraavia laitteita: ekvalisaattorit, dynamiikkaprosessorit, kaikulaitteet sekä muita efektilaitteita.

Laitteita tärkeämpi asia äänityksen onnistumisen kannalta on äänittäjän ammattitaito, joka koostuu mm. seuraavista asioista: äänittäjän tuntemus äänitettävästä musiikkityylistä, soittimista, mikrofoneista sekä erilaisista mono-, stereo- ja surround-mikrofoniteknikoista, mikrofoni sijoittelusta, musiikille sopivasta äänitystilasta valinnasta sekä äänittäjän psykologisista taidoista toimia ryhmässä.

5.2.6.2 Miksaaminen

Yleisiä miksausvaiheen päämääriä on luoda äänitteeseen tilantuntu (*leveys, ja syvyys*), selkeys, erottelu ja kirkkaus. Keskeiset miksausvaiheen työkalut ovat: suotimet, taajuuskorjaimet, dynamiikkaprosessorit (*kompressorit, limiterit, ekspanderit ja kohinasalpa*), kaikulaitteet, viiveet sekä modulaatioefektit (*koorus, flangerit ja feiserit*).

Miksaaja toteuttaa äänitettyjen ääniraitojen miksaamisen formaattiin, jossa taltiota voidaan kuunnella äänitys- ja miksaustudion ulkopuolella. Näitä formaatteja ovat mm. stereomiksaus, jolloin äänitettä voidaan kuunnella esim. CD-levyltä tai mp3-tiedostosta, monikanavamiksaus, jolloin äänitettä voidaan kuunnella esim. DVD-audiolevyltä tai SACD-levyltä.

Moni miksaaja kuulee jonkinlaisen version valmiista miksauksesta mielessään ennen kuin he aloittavat miksaamisen (Owsinski 2006: 7).

5.2.6.3 Masterointi

Masterointi on viimeinen luova vaihe audiotuotannon prosessissa, silta miksauksen ja monistuksen tai jakamisen välissä. Masterointi on viimeinen mahdollisuus parantaa soundia tai korjata ongelmia audio-mikroskoopilla tilassa jossa on tarkoitusta varten suunniteltu akustiikka. Masteroija tuo mukanaan projektille objektiivisen ja kokeneen korvan. Masteroijat ovat hyvin perillä siitä mikä voi mennä vikaan niin teknisesti kuin myös taiteellisesti. Joskus masteroijat eivät tee äänitteelle masteroinnissa mitään! Yksinkertainen hyväksyntä tarkoittaa, että miksaus on valmis monistukseen. Toisinaan masteroijat auttavat tuottajaa ratkaisemaan ongelmat, joita miksausvaiheessa ei kyetty ratkaisemaan, tai voivat lisätä viimeisen silauksen äänitteelle, joka tekee levystä valmiin ja monenlaisissa äänilaitteistoissa toistettavan tallenteen (Katz 2007: 12).

Tärkeimmät masteroijan työkalut ovat: kaiuttimet, ekvalisaattorit, erilaiset dynamiikkaprosessorit (*kompressorit, monikanavakompressorit, limiterit*), stereoprosessorit ja signaalianalysointilaitteet. Laitteiden kytkentäjärjestys signaalireitissä voi olla hyvin merkittävä laatuun vaikuttava tekijä (Music Radar 2013).

5.2.7 Äänentoisto

5.2.7.1 Äänentoisto

Äänentoisto on äänen sähköistä vahvistamista ja muokkaamista ammatti- ja kotikäytössä (Äänentoisto 2014). Konserttitilanteessa äänentoiston päämääränä on tarjota jokaiselle kuulijalle yhtä laadukas kuuntelukokemus. Hyvin toimivan saliaäänijärjestelmän ominaisuuksia ovat mm. hyvä ymmärrettävyys, tasainen äänenpaine ja riittävän tasainen taajuusvaste koko salissa (Lepoluoto 2007).

Äänentoiston toteutukseen vaikuttavat mm. tilan koko, tilan muoto, tilan pintamateriaalit, onko kysymys puheen vai musiikin äänentoistosta, musiikkityyli, yhtyeen kokoonpano ja soittimien/äänilähteiden lukumäärä.

Keskeisiä työkaluja äänentoistoa tehtäessä ovat mikrofonit, DI-boxit, äänipöytä, ristikytkentä, kauko-ohjaimet, taajuuskorjaimet, kaikulaitteet, viivelaitteet, dynamiikkaprosessorit, kohinan rajoittimet, matriisit, päätevahvistimet, erilaiset kaiuttimet (*mm. linja-, pallo- ja pistesäteilijä*), jakosuotimet, tarkkailukuulokkeet, mittarit, analysaattorit, mikrofoni- ja kaiutinkaapelit, verkkojohdot, moninapaiset kaukokaapelit lavarasioineen ja liitinviuhkoineen, sovittimet eli adapterit, mikrofoni- ja kaiutintelineet (Blomberg & Lepoluoto 1993/2005: 200) ja (Lepoluoto 2007).

Äänentoistossa käytetään monia samoja työkaluja kuin äänitettäessä. Äänentoisto kuitenkin edellyttää erityisesti äänentoistoa varten suunniteltujen työkalujen käyttöä. Esimerkiksi paras mikrofoni äänitystä varten ei ole todennäköisesti paras mikrofoni äänentoistoa varten. Tämä johtuu siitä, että äänitettäessä ei tarvitse miettiä mahdollista kierto-ongelmaa, joka äänentoistoa tehtäessä on asia, joka pitää ottaa huomioon kaikissa laitevalinnoissa.

5.2.7.2 Elektroakustinen äänidiffuusio

Elektroakustisen musiikin äänidiffuusio menee äänentoiston erityistapauksiin. Siinä toteutetaan yleensä stereotallenteesta esitys kaiutinatorkesterille, jossa voi olla jopa 100 kaiutinta. Ks. myös musiikin esittämisen kohta nro 5.2.5.4 Elektroakustinen äänidiffuusio.

5.2.8 Musiikkisovellusten ohjelmointi

Musiikkisovellusten ohjelmointi on merkitty kuvan 1 kaavion siten, että siitä on vedetty nuolet lähes kaikkiin kohtiin. Periaatteessa ohjelmointi voi liittyä mihin tahansa musiikin teon vaiheeseen tai musiikillisen työkalun tekoon. Näitä ovat mm. sävellysalgoritmin, nuotinnusohjelman, klikkiraidan, ohjelmistosyntetisaattorin, äänitysohjelman, miksauksessa käytettävän liitännäisohjelman (plug-in) ja masterointiohjelman ohjelmointi.

Ohjelmointi liittyy kohtaan “*improvisointi teknologisella laitteistolla*” myös siten, että improvisoija omatessaan ohjelmointitaitoja saattaa ohjelmoida itselleen improvisoinnissa tarpeellisia ohjelmia. Esimerkiksi käyttöliittymän konfigurointi, mappaus ja sen hienosäätö voivat olla jotain mitä improvisoija voi haluta ohjelmoida juuri itselleen sopiviksi.

Ohjelmointitaidoilla ei ole musiikkiteknologista merkitystä mikäli ohjelmointitaidot eivät ole palvelemissa musiikillista sisältöä, äänen laatua tai siistiä nuottisivua.

5.3 Musiikkiteknologian spesialistit ja moniosaajat

Ihminen voi olla joko yhden musiikkiteknologisen sektorin spesialisti tai useamman sektorin moniosaaja. Musiikkiteknologia palauttaa takaisin renessanssin ihanteen *l'uomo universale* (*Homo Universalis*).² Äärimmäisessä tapauksessa musiikkiteknologinen laitteisto mahdollistaa tilanteen, jossa yksi ihminen voi tehdä kaikkia musiikin tekemiseen liittyviä vaiheita. Laitteistojen avulla hän voi toimia niin säveltäjänä, nuottien puhtaaksi kirjoittajana, "kapellimestarina"³, soitinrakentajana, muusikkona, äänittäjänä, miksaajana sekä masteroijana ja jopa käyttämiensä ohjelmien ohjelmoijana. Tyypillisempää kuitenkin on, että henkilöllä on mielenkiintoa ja taipumuksia toteuttaa yksin vain joitakin musiikkiteknologian osa-alueita. Lisäksi projektien nopeamman valmiiksi saamisen kannalta ei ole kovin tehokasta yrittää tehdä kaikkea yksin.

Musiikkiteknologian kentällä esiintyy niin erikoistujia kuin moniosaajiakin. Joku voi olla erikoistunut esimerkiksi ohjelmoimaan sävellysalgoritmeja, joku toinen taas tietokonenuotinnukseen, kolmas äänisynteesiin, neljäs äänittämiseen ja miksaamiseen ja viides masterointiin. Tietokonetyöasemalla tapahtuvassa musiikkityöskentelyssä moniosaajuus on myös hyvin luonteva tapa työskennellä. Soitinlähtöisen musiikin tekemisen rinnalle on kasvanut teknologialähtöinen musiikin tekeminen, jossa yksittäisen perinteisen soittimen soittotaito ei ole enää niin merkittävässä roolissa. Esimerkkejä tästä ovat mm. elektroakustinen musiikki, jossa kaikki äänet voidaan generoida, prosessoida, leikata ja miksata kokonaan elektronisilla laitteilla tai rytmi- ja elokuvamusiikki, joissa elektronisia soittimia ja äänilähteitä käytetään akustisten instrumenttien rinnalla.

Xavier Serra luennossaan Iso-Britanniassa digitaalisen musiikin tutkimusverkostolle (*Digital Music Research Network*) vuonna 2003 esitti vetoituksen sen puolesta, että tieteen,

² "Ihminen, joka oli monen alan osaaja" (Renaissance Man 2009).

³ Kapellimestari on kirjoitettu lainausmerkkeihin sen takia, että kapellimestarin asemasta muusikoita ohjataan klikkiraitojen avulla. Klikkiraidat soitetaan muusikoiden korviin kuulokkeiden välityksellä, jolloin yleisö ei kuule niitä lainkaan.

teknologian ja taiteiden spesialistit työskentelisivät yhdessä tai vaihtoehtoisesti kannusti spesialisteja erikoistumaan useampaan kuin vain yhteen alueeseen (Landy 2007: 225).

6 MUSIIKKITEKNOLOGIAN HOLISTINEN OPETTAMINEN

Soitonopetuksessa fyysinen suoritus (=itse tekeminen), kuulohavainto ja kuulohavaintoon pohjautuva ajattelu ja analyysi ovat soitinopinnoissa läsnä tasapuolisesti ja kokonaisvaltaisesti opintojen alkuvaiheesta lähtien. Tekemistä seuraa kuulohavainto, kuulohavainnon perusteella tehdään reaaliajassa analyysiä, meneekö hyvin vai pitääkö jotain muuttaa tai korjata. Mahdollista korjausajatusta seuraa teko, jota seuraa kuulohavainto, jota seuraa uusi analyysi. Tämä takaisinkytkentä: *”toiminta - kuuntelu - ajattelu - toiminta - kuuntelu - ajattelu...”* ohjaa muusikon toimintaa koko ajan ja reaaliajassa. Kuuntelun, analysoinnin ja muutosohjeiden määrittämisessä tosin opettajan rooli on varsinkin alussa hyvin merkittävä. Tätä holistista näkökulmaa olen halunnut soveltaa myös musiikkiteknologian opetuksessani. Opinnäytetyön Max-patchit ovat syntyneet tarpeesta opettaa musiikkitekologiaa soitonopetuksen kaltaisessa hengessä. Oppimateriaalin Max-patchejä käytetään siten, että oppilaalle tulostetaan paperille tehtäväpatch, jonka oppilas paperilta kopioiden rakentaa Max-ohjelmaan tietokoneen äärellä. Patchien rakentaminen paperille printatusta mallista edellyttää kahta taitoa, taitoa luoda uusia objekteja sekä taitoa tehdä kytkentöjä näiden objektien välille. Näiden taitojen opettamiseen menee noin 10 minuuttia aikaa. Ks. tarkemmin kohdasta 10.3 Johdatus Max-patchien rakentamiseen. Tämä mahdollistaa sen, että oppilaat pääsevät jo 10 minuutin kuluttua itse työskentelemään Max-ohjelmalla. Patchin tekemiseen kuluva aika vaihtelee sen mukaan, miten monta objektia tehtävän patchissä on. Sen jälkeen kun patch on tehty valmiiksi oikein, alkaa ääni kuulua kaikissa muissa oppimateriaalin patcheistä paitsi patchissä nro 19 bipolaari- ja unipolaarisignaali.

Tämä opetusmalli mahdollistaa sen, että oppitunnit eivät ala luennolla aiheesta, vaan aktiivisella työskentelyllä. Tällaisessa holistisessa opetusmetodissa luentoa tai luennonkaltaista osiota käytetään oppilaiden ymmärtämyksen syventämiseen sen jälkeen kun asioihin on perehdytty käytännössä.

Syitä tähän toimintamalliin on se, että musiikkitekologia pitää sisällään lainasanoihin perustuvaa terminologiaa sekä haastavia teoreettisia asioita. Vasta-alkajien ollessa oppilaina musiikkiteknologian vierasperäisiin sanoihin pohjautuva terminologia sekä teoreettiset asiat voivat olla vaikeita ymmärtää ilman käsitteitä ja teorioita konkretisoivia kuultavia ääniä. Äänen avulla terminologia ja teoriala havainnollistuvat sekä ymmärtäminen helpottuu. Pyrin tällä tavoin herättämään mielenkiinnon ja uteliaisuuden musiikkiteknologian asiantietoon. Kun

oppilaalle on tullut patchin kautta vastaan ongelma, jota hän ei ole kyennyt ratkaisemaan, on oppilaassa heräävä tiedon nälkä ja uteliaisuus erinomaista maaperää, jolle tarjota vierasperäiset musiikkiteknologiset termit sekä kertoa kyseiseen aiheeseen liittyviä teoreettisia asioita.

Pyrkimys soitonopetuksen kaltaiseen opetukseen musiikkiteknologian opetuksessa herätti minussa tarpeen jäsentää miten instrumentti-käsite määritetään yksityiskohtaisemmin, jotta voin vertailla mitä yhtäläisyyksiä musiikkiteknologisella laitteistolla työskentelyn ja soittimien soiton väliltä löytyy.

7 INSTRUMENTTI JA MUUSIKKOUS

7.1 Instrumentin määrittely

Instrumentilla tarkoitetaan mitä tahansa välinettä tai työkalua (Instrumentti 2013). Grove Music Onlinessä instrumentti-käsite määritellään erilaisten akustisten instrumenttien värähtelijöiden mukaan. Nämä kategoriat ovat: 1) *aerofonit* (esim. huilu) 2) *kordofonit* (esim. viulu) 3) *membranofonit* (esim. bassorumpu) ja 4) *idiofonit* (esim. kellopeli). Aerofoneissa äänilähteen ensisijainen värähtelijä on ilmapatsas, kordofoneissa kieli, membranofoneissa kalvo ja idiofoneissa idiofonin oma massa tai suurin osa siitä (Brown, Howard Mayer & Palmer, Frances 2014).

Pierre Schaefferin/Michel Chionin määritelmä instrumentille kirjassa *Guide to Sound Objects* on seuraava: *“Jokainen laite, jolla voidaan tuottaa monipuolinen kokoelma ääniä, on musiikillinen instrumentti perinteisessä merkityksessä kaikille sivilisaatioille.”* (Chion 1983: 53).

Instrumentilla on eri funktio eri tekijöille. Esittävän muusikon näkökulmasta instrumentti on muusikon kehon ja mielen jatke, kun taas säveltäjälle instrumentti on äänilähde (*ulkoinen objekti*), jolla tuotetaan musiikkia. Soitinrakentajalle valmis instrumentti on työprosessin päätepiste.

Instrumentti käsitettä pohdittaessa, esittävän muusikon näkökulmasta, olennaista ei ole se minkä materiaalin ihminen kohtaa ihmisen ja instrumentin välisessä vuorovaikutusrajapinnassa. Olennaista ei myöskään ole se pitääkö tässä rajapinnassa puhalttaa, painaa, hieroa, raapia tai heiluttaa tuottaakseen ääniä. Olennaista ei ole myöskään se otetaanko tässä rajapinnassa kontakti koskettimeen, kieleen, kalvoon, läppään, venttiiliin, potentiometriin tai kontrolleriin. Olennaista ei ole myöskään se onko instrumentti sähkötön tai sähköinen.

Olennaista instrumentin käytön ja sillä tuotetun esteettisen elämyksen kannalta on se, miten käyttöliittymän käyttäjä osaa muokata haluamiaan ja valitsemiaan musiikillisia parametreja valitsemallaan työkalulla ja sen käyttöliittymällä. Instrumentin tulee luoda edellytykset ilmaista niitä musiikillisia pyrkimyksiä, jotka instrumenttia soittavalla muusikolla on. Kaikilla soittimilla ei pystytä kontrolloimaan kaikkia musiikillisia parametreja hyvin, eikä ole tarpeenkaan. Erilaisilla soittimilla on erilaiset roolit. Esim. triangelilla

pystytään soittamaan hyvin ilmeikkäästi eri äänenvoimakkuuksilla rytmikasta puhumattakaan. Triangelin soinnin spektriä voidaan myös muokata eri tavoin esim. vaimentamalla triangelia kädellä tai valitsemalla tikun materiaali. Triangelin tuottaman äänen spektri erottuu orkesterin seasta selkeästi. Yksittäisellä triangelilta ei kuitenkaan kannata odottaa kykyä toimia esim. melodiainstrumenttina. Triangeli ei soittimena ole olemassa melodiainstrumentin roolia varten.

Äänittäjä-miksaaja Bruce Swedien toteaa seuraavaa: *“Katson olevani artisti äänitysstudioissa.”* (Swedien 2003: takakansi). Äänittäjä-tuottaja Silja Suntola toteaa taas kirjassaan seuraavaa: *“Tekniikan kehitys on antanut mahdollisuuden pelkän äänentoiston sijasta luoda täysin uudenlaisia ja kuvitteellisiakin äänimaailmoja. Äänitysstudiosta on tullut uusi soitin, joka antaa meille välineet maalaila näitä äänikuvia, oma mielikuvituksemme vain rajana...”* (Suntola 2000).

7.2 Instrumentin komponentit

Seuraavassa käyn läpi yleisen tason määritelmäni instrumentin, ihmisen toiminnan kohteena olevan objektin, eri komponenteista. Näistä komponenteista välttämätön osa on värähtelijä. Kaikkia näitä komponentteja ei kaikissa soittimissa kuitenkaan ole.

- 1) käyttöliittymä
- 2) mekanismi käyttöliittymän ja värähtelyn sytyttäjän välillä
- 3) värähtelyn sytyttäjä
- 4) värähtelijä(t)

7.2.1 Instrumenttien erilaiset käyttöliittymät

Perinteisten soittimien soittaminen tapahtuu käsillä, jaloilla ja/tai huulilla tapahtuvan fyysisen kosketuksen välityksellä. Näitä pintoja, joissa fyysinen kosketus tapahtuu, nimitän käyttöliittymiksi.⁴ Seuraavassa muutamia esimerkkejä soittimien erilaisista käyttöliittymistä.

Kosketinsoittimissa (esim. *piano, urut, cembalo*) sormilla painetaan koskettimia ja jaloilla mahdollisia pedaaleja tai jalkiota. Puhallinsoittimissa huulilla kosketetaan

⁴ Poikkeuksena theremin, jota soitetaan liikuttamalla käsiä antennien läheisyydessä ilman fyysistä kosketusta.

suukappaletta (*esim. ruokolehdykkää klarinetissa ja saksofonissa, kaksoisruokolehdykkää oboessa, metallista suukappaletta kaikissa vaskipuhaltimissa*) sekä käsillä/sormilla tukitaan reikiä tai painellaan läppiä tai venttiilejä. Jousisoittimissa toisessa kädessä pidetään joustia ja toisen käden sormilla painetaan kieliä otelautaa vasten. Lyömäsoittimia soitetaan esim. lyömällä kalvoja, kieliä tai levyjä sormenpäillä, kapuloilla/malleteilla tai pedaalin nuijalla. Toisia lyömäsoittimia soitetaan taas ravistamalla, raaputtamalla tai hankaamalla.

Seuraavassa esimerkkejä elektronisista käyttöliittymistä. Syntetisaattoreissa on pianon koskettimiston näköinen koskettimisto, tosin kosketus on hyvin erilainen kuin pianossa. Midi-kontrollereissa painetaan sormilla kytkimiä tai liu'utetaan liukuja. Monomessa painellaan kytkimiä, Reactablea soittaessa Reactablen päälle aseteltavia tangible-objekteja käännellään ja siirrellään. Space Palettessa ei ole kosketettavaa käyttöliittymä, vaan sitä soitetaan heiluttamalla käsiä. Space Palettessa on Kinect-kamera, joka tunnistaa käsien sijainnin sivu-, pysty- ja syvyyssuunnassa. SLABS:ssä sormilla painetaan pädejä, jotka reagoivat sormien kosketuksen sijaintiin x- ja y-akselilla sekä paineeseen (*z-akseli*), joka syntyy sormilla painettaessa. SLABS:n pädit eivät siis ole vain triggereitä. Tenori-On:ssa sormilla liipaistaan samplejä tai luuppeja.

Älypuhelimia voidaan käyttää myös kontrollereina. Esimerkiksi Applen iPhone -puhelimista löytyy erilaisia sensoreita: gyroskooppi, kiihtyvyysanturi, kompassi, läheisyysanturi ja hämärävalon anturi. Hexler yhtiön tekemällä TouchOSC -applikaatiolla saadaan iPhonen kiihtyvyysanturista kolme erillistä datavirtaa, jotka on mahdollista ohjata Max-ohjelmaan ja käyttää niitä ohjaussignaaleina äänen tuottamisessa ja/tai muokkaamisessa.

7.2.2 Mekanismi käyttöliittymän ja värähtelyn sytyttäjän välillä

Piano, urut ja harmonikka ovat esimerkkejä instrumenteista, joista löytyy kompleksi mekanismi koskettimiston (*käyttöliittymä*) ja värähtelyn sytyttäjän väliltä. Esimerkiksi pianossa sormen energia, jolla kosketin painetaan pohjaan, välitetään pianon koneiston välityksellä vasaralle. Puupuhaltimien läppämekanismeilla ja vaskipuhaltimien venttiilimekanismeilla vaikutetaan värähtelevän ilmapatsaan pituuteen ja sitä kautta sävelkorkeuteen. Näppäilysoittimet esim. kitara ja harppu ovat esimerkkejä instrumenteista, joista ei löydy lainkaan mekanismeita käyttöliittymän ja värähtelyn sytyttäjän väliltä, koska sormen pää itsessään on värähtelyn sytyttäjä.

Mahdollinen mekanismi käyttöliittymän ja värähtelyn sytyttäjän välillä on akustisissa soittimissa mekaaninen, jonka johdosta tätä komponenttia ei elektronisista soittimista löydy.

7.2.3 Erilaisia värähtelyn sytyttäjiä

Erilaisia värähtelyn sytyttäjiä ovat mm. jousen jouhet jousisoittimissa, ilmavirta tai huulet puhallinsoittimissa, sormenpää, kynsi tai plektra näppäilysoittimissa, kapula tai malletti monissa lyömäsoittimissa, vasara pianossa ja celestassa sekä vaihtojännite sähköisissä soittimissa/laitteistoissa, joissa kaiutin on värähtelijänä.

Huulilla voidaan todeta olevan dualistinen rooli. Toisaalta ne toimivat värähtelyn sytyttäjinä ja toisaalta ne ovat jo itse primääri värähtelijä. Esimerkiksi vaskipuhaltimissa ne voidaan katsoa olevan värähtelyn sytyttäjiä, koska huulten värähtely siirtyy suokappaleen kautta soittimen putken värähtelyksi ja sitä kautta putken sisällä olevan ilmapatsaan värähtelyksi.

7.2.4 Erilaisia värähtelijöitä

Soittimien primäärisinä värähtelijöinä voivat toimia kieli (*viulu*), huulet (*trumpetti, pasuuna etc.*), kalvo (*rumpu*), ruokolehdykkä (*klarineti, oboe, fagotti etc.*), ilmapatsas (*huilu*), palkki/kieli (*kellopeli, ksylofoni, marimba*), levy (*syмбаали, tamtam*) sekä kaiutin (*telharmonium, theremin, ondes martenot*). Usein kuulijan kuulema ääni kuitenkin muodostuu useamman värähtelijän summasta. Esimerkkeinä voidaan mainita klarinetissa ruokolehdykkä ja ilmapatsas, pianossa kielet ja äänipohja sekä kitarassa kielet ja koppa (Rossing, Moore & Wheeler 2002: 31).

7.3 Automaatti-instrumentin komponentit

Automaatti-instrumentit eroavat perinteisistä instrumenteista siten, että automaatti-instrumenttien rakenteessa on esittävän muusikon työn automatisoiva komponentti. Esimerkkejä automaatti-instrumenteista ovat mm. automaattipiano, -urut, soittorasia ja posetiivi.

Yleisellä tasolla automatisoinnista voidaan puhua silloin, kun äänilähde generoi yhdellä ihmisen antamalla ärsykkeellä tai eleellä enemmän kuin yhden eriaikaan tapahtuvan äänitapahtuman.

Automaattipianoissa ja -uruissa automatisointi on toteutettu pyörivällä rullamekanismilla, johon on yleensä koodattu musiikillisista parametreista lähinnä sävelkorkeudet, nuottien alkuajat ja nuottien kestot. Muita musiikillisia parametreja, tempoa, äänenvoimakkuutta, fraseerausta pianolisti voi säätää esityksen aikana musiikillisen näkemyksensä mukaisesti (Lawson 2004). Osa automaatti-instrumenteista toimii käynnistyksen jälkeen ilman ihmisen osallistumista, toiset taas vaativat/mahdollistavat jonkun asteista manuaalista kontrollointia.

Ihmisellä on luontainen mielenkiinto automatisoida asioita. 1800-luvun loppupuolella, kun ensimmäiset automaatti-instrumentit kehitettiin, tavoitteena oli mahdollistaa musiikin kuunteleminen ilman, että muusikoiden täytyy olla aina esittämässä musiikkia. Äänilevyteollisuuden nousu toteutti sittemmin tämän tarpeen monipuolisemmin ja edullisemmin syöden automaatti-instrumenttien silloiset markkinat.

Musiikkiteknologia mahdollistaa musiikillisten parametrien automatisoinnin. Tietokoneiden laskentateho ja älykkäät algoritmit antavat huomattavasti monipuolisemmat mahdollisuudet automatisointiin verrattuna automaatti-instrumenttien rullamekanismeihin.

7.4 Muusikon toiminnasta

Esittävän muusikon tehtävä on tuottaa eri määrä säveltasollisia tai säveltasottomia ääniä haluttuina ajankohtina muusikon esteettisen näkemyksen mukaisesti. Soivasta äänestä muusikon kontrolloimia parametreja ovat sävelkorkeus, rytmiiikka, sävelkesto, äänenvoimakkuus, sointiväri, tempo ja artikulaatio.

Musiikkiteknologisella laitteistolla voidaan myös tuottaa tulkitsijan esteettisen näkemyksen mukaisesti eri määrä säveltasollisia tai säveltasottomia ääniä tulkitsijan haluamina ajankohtina. Sillä voidaan myös muokata generoidun musiikin rakenteen sävelkorkeuksia, rytmiikkaa, kestoja, äänenvoimakkuuksia, spektriä / sävyjä, tempoa ja äänien artikulaatiota hyvin monin keinoin.

7.4.1 Vuorovaikutus ihmisen ja instrumentin käyttöliittymän välisessä rajapinnassa

Ihminen voi instrumentin/äänilähteen äärellä tehdä seuraavia asioita (Cook 2002: xiii):

- 1) puhalttaa
- 2) lyödä, painaa tai näpätä sekä
- 3) hangata, raapia tai silittää.

Edeltävistä kategorioista kohta 2 voidaan yleistää *ajallisesti lyhytkestoinen fyysinen kosketus* -kategoriaan ja kohta 3 *ajallisesti pitkäkestoinen fyysinen kosketus* -kategoriaan.

7.4.2 Muusikon fyysinen toiminta

Soittaessaan muusikot koskettavat joko käyttöliittymää (*esim. pianon koskettimisto*), värähtelyn sytyttäjää (*esim. marimban malletti*) tai värähtelijää (*esim. kitaran kieli*). Muusikon fyysistä toimintaa ohjaa hänen kuulohavaintonsa.

7.4.3 Muusikon auditiivinen toiminta

Muusikko säätää kuulohavaintoonsa pohjautuen fyysistä toimintaansa siten, että kuulohavainto täyttää hänen esteettisen näkemyksensä esitettävästä musiikista. Esittäessään muusikko muokkaa musiikilliset parametrit (*sävelkorkeus, sävelkesto, rytmikka, tempo, äänenvoimakkuus, sointiväri ja artikulaatio*) näkemyksensä mukaisiksi.

Soittimien soittamisessa on siis viime kädessä kyse instrumentin värähtelijän/värähtelijöiden hallitusta kontrolloimisesta käyttöliittymästä käsin esittäjän esteettisen näkemyksen mukaisesti riippumatta siitä onko kosketettavan käyttöliittymän tai värähtelijän materiaali puuta, metallia, lasia tai onko käyttöliittymässä, värähtelijässä tai niiden välillä sähköisiä komponentteja.

7.5 Muusikkoperheen uusi jäsen

Musiikkiteknologista on tullut "*muusikkoperheen*" uusi jäsen. Miksi säveltäjää, kapellimestaria ja esittävän säveltaiteen edustajaa kutsutaan muusikoiksi, vaikka työtehtävät heillä ovat hyvin erilaisia? Mikä on yhdistävä tekijä näiden erilaisten työtehtävien välillä? Säveltäjä Vesa Valkaman mukaan yhdistävä tekijä on näkemys esitettävästä musiikista. Säveltäjä suunnittelee teoksen rakenteen ja määrittelee kaikkien musiikillisten parametrien

yksityiskohdat. Kapellimestari luo näkemyksensä säveltäjän määrittelemän teoksen kokonaisuudesta ja sen yksityiskohdista ja pyytää muusikoita toteuttamaan yksityiskohdat kapellimestarin näkemyksen mukaisesti. Esittävät muusikot hoitavat käytännön toteutuksen kokonaisuudesta yksityiskohdista käsin nuotti/ääni kerrallaan.

“Muusikkoperheen jäsenyys” ei siis perustu siihen mitä säveltäjä, kapellimestari tai muusikko tekee käytännön työssään vaan *“Muusikkoperheen jäsenyyden”* ytimessä on teos, johon kullakin *“muusikkoperheen”* jäsenellä tulee olla yhteys. Säveltäjä luo teoksen, esittävät muusikot esittävät sen, kapellimestari johtaa kokoonpanoa, musiikkiteknologi voi olla esim. esiintymässä elektronisella laitteistolla tai toteuttamassa taltiointia esityksestä. Detaljitason ratkaisuihin jokaisella tekijän teolla on yhteys teokseen ja valinnat tehdään teoksen ehdoilla. Tekijän ammattitaito punnitaan ja arvioidaan tehtyjen valintojen perusteella.

8 MAX-OHJELMA

Patch-käsitteellä tarkoitetaan Max-ohjelman objekteilla rakennettavia kytkentäkaavioita. Opinnäytetyön patcheissä käytetyt objektit ovat kaikki ohjelman mukana tulevia objekteja. Max 6:n mukana tulee n. 1000 valmista objektia, joilla voidaan rakentaa mitä halutaan ja osataan. Kolmansien osapuolien tekemiä lisäobjekteja löytyy mm. seuraavalta verkkosivustolta: <http://www.maxobjects.com>.

8.1 Max-ohjelman taustaa

Max -ohjelma on Miller Pucketten 1980-luvun puolivälissä ohjelmoima ohjelma, jonka hän teki työskennellessään IRCAM:ssa, Pariisissa, Ranskassa. Vuonna 1997 Max:iin lisättiin MSP -osio, joka tuo mukanaan audiosignaali-prosessoinnin. MSP -akronyymi tulee sanoista Max Signal Processing. MSP akronyymia väitetään tulevan myös Miller Smith Pucketten nimien ensimmäisistä kirjaimista. Max on nimetty yhdysvaltalaisen tietokonemusiikin pioneerin Max Mathews (1926-2011) mukaan. Max Mathews ohjelmoi maailman ensimmäisen tietokonemusiikkiesityksen vuonna 1957 IBM 704 -tietokoneella Music I -ohjelmalla. Sävellys oli 17 sekuntia pitkä.

8.2 Ohjelmointi Max-ohjelmalla

Max -ohjelmointi on visuaalinen ohjelmointikieli äänen ja kuvan generointia varten. Max-ohjelmointi kuuluu *data flow* -ohjelmointikategoriaan.

Ohjelmointi (programming) on suppeassa mielessä toimintaohjeiden kirjoittamista jollain ohjelmointikielellä. Laajassa mielessä se on ohjelmistotyötä, jolloin siihen kuuluvat seuraavat työvaiheet: määrittely, suunnittelu, toteutus, testaus, dokumentointi ja ylläpito (Silander 2000: 20).

8.3 Max-ohjelman edut opetuskäytössä

Alla on lueteltu Maxin ominaisuuksia, jotka tekevät siitä myös lasten ja nuorten opetuskäyttöön hyvin soveltuvan ohjelman.

1) Patchien rakentaminen Maxillä on helppoa. Rakentaakseen paperille printatun mallin mukaisen patchin, oppilaan tarvitsee osata vain luoda objekteja ja tehdä kytkentöjä niiden välille.

- 2) Max-ohjelma mahdollistaa opettajalle sen, että käsillä olevassa patchissä on näkyvissä vain opetettavan asian kannalta olennaiset asiat. Tämä mahdollistaa oppilaan huomion fokusoimisen käsiteltävään asiaan.
- 3) Max-patchien visuaaliset kytkentäkaaviot mahdollistavat patchien toimintaperiaatteiden yksityiskohtaisen opiskelun. Max-patchien rakenne on käyttäjän näkyvissä toisin kuin monissa ohjelmistosyntetisaattoreissa, joissa käyttäjälle näytetään ainoastaan käyttöliittymä.
- 4) Kokemus on osoittanut, että pitämällä objektien lukumäärä pienenä, n. 20, pysyy patchin toiminta käyttäjälle ymmärrettävänä.

9 OPPIMATERIAALIKOKONAISUUDEN SISÄLTÖKATEGORIAT

Oppimateriaalin tehtävät voidaan jakaa neljään kategoriaan:

- 1) Ääniopin ja musiikkiteknologian peruskäsitteitä esittelevät patchit
- 2) Äänisynteesimetodien perusteita esittelevät patchit
- 3) Efekti- ja prosessointilaitteiden toimintaperiaatteita esittelevät patchit
- 4) Muut patchit

Oppimateriaalikokonaisuuden ensimmäisen, toisen ja kolmannen kategorian patcheissa ilmiöitä demonstroidaan ääninäytteiden sekä osassa myös visuaalisten näytteiden avulla. Toisen kategorian patchissä nro 19 *Unipolaari- ja bipolaarisignaali*, käsitteitä demonstroidaan vain visuaalisesti. Neljännen kategorian patchit liittyvät patchien rakentamiseen ja käyttöön, joten niillä ei ole tarkoitus demonstroida ääni-ilmiöitä ääninäyttein eikä visuaalisten näytteiden avulla.

Esimerkiksi ensimmäisen kategorian patchin nro 2 *Ääniopin ja musiikkiteknologian peruskäsitteitä* avulla on mahdollista kuulla aaltomuoto äänenä, nähdä aaltomuodon kuvaaja sekä nähdä luvulla määritetty taajuus nuottina nuottiviivastolla. Lisäksi näkyvän aaltomuodon yhteydestä kaiuttimen kartioon voidaan todeta, että signaalin arvojen ollessa positiivisia kaiuttimen kartio liikkuu lepokohdan etupuolella, kun taas signaalin arvojen ollessa negatiivisia kaiuttimen kartio liikkuu lepokohdan takapuolella.

Ensimmäisen kategorian patcheilla käydään läpi ääneen liittyviä peruskäsitteitä ja ominaisuuksia. Ensimmäisen kategorian patchit ovat pohjatietoa toisen kategorian patchien rakentamiselle, toteutettaessa erilaisia ääniä moninaisten äänisynteesimetodien avulla.

Kolmannen kategorian patcheilla efektoidaan ja prosessoidaan olemassa olevia ääniä. Kolmannen kategorian patchien äänilähteiden tilalle käyttäjä voi vaihtaa rakentamiaan toisen kategorian patcheja.

Neljännen kategorian patcheissa demonstroidaan Maxin erilaiset johtotyytit, datan erilaiset monitorointitavat sekä opetetaan, miten ulkoinen midi-kontrolleri voidaan kytkeä Max-ohjelmaan.

9.1 Ääniopin ja musiikkiteknologian peruskäsitteitä esittelevät patchit ja niiden oppimistavoitteet

Ensimmäisen kategorian patcheihin kuuluvat seuraavat patchit: 1. *Siniääni*, 2. *Ääniopin ja musiikkiteknologian peruskäsitteitä*, 3. *Erilaisia aaltomuotoja*, 4. *Kohinatyyppejä*, 5. *Vibrato*, 6. *Verhokäyrä*, 7. *Vaihe*, 8. *Ylä-äänessarja*, 9. *Äänennopeus ja aallonpituus sekä* 10. *Näytteenottotaajuus ja amplitudiresoluutio*.

Käsitteitä ja ilmiöitä, jotka näiden patchien puitteissa havainnollistetaan ja tehdään tutuksi ovat mm. aallonpituus, aaltoliike, ADSR-verhokäyrä, amplitudi, amplitudiresoluutio, erilaisia aaltomuotoja (sini-, kosini-, kantti- ja sahalaita-aalto) kokonaislukukerrannaiset, harmoninen osaaäneistö, häly, ihmisen kuuloalue, jaksollinen ja jaksoton ääni, laskostuminen, Nyqvistin taajuus, näytteenottotaajuus, osaaänes, osaaäneistö, siniääni, taajuus, taajuuskaista, vaaleanpunainen kohina, valkoinen kohina, vaihe, verhokäyrä, vibrato, vibraton laajuus, vibraton nopeus, värähtely, äänenvoimakkuus ja äänen nopeus.

Oppimistavoitteena ensimmäisen kategorian patcheilla on auttaa oppilasta oppimaan ääneen liittyviä musiikkiteknologian perustermejä sekä tutustuttaa oppilas äänen fysikaalisiin parametreihin.

9.2 Äänisynteesimetodien perusteita esittelevät patchit ja niiden oppimistavoitteet

Toisen kategorian patchit esittelevät erilaisia äänisynteesimetodeja. Näitä äänisynteesimetodeja ovat: 11. *Aaltotaulukkosynteesi*, 12. *Lisäävä synteesi 1*, 13. *Lisäävä synteesi 2*, 14. *Vähentävä synteesi 1*, 15. *Vähentävä synteesi 2*, 18. *Pientaajuusoskillaattori*, 20. *Amplitudimodulaatio*, 21. *Rengasmodulaatio*, 22. *Amplitudi- ja rengasmodulaatioiden vertailu*, 24. *FM-synteesi*, 25. *Raesynteesi*, 26. *Aaltomuotoilusynteesi* ja 27. *Kahden aaltomuodon ristiinfeidaus*.

Lisäksi tässä osiossa käydään läpi vähentävän synteesin keskeiset työkalut: patch nro 16. *Suotimet* ja 17. *Taajuuskorjaimet*. Patchillä nro 18. *Pientaajuusoskillaattori* voidaan kokeilla pientaajuusoskillaattorin erilaisia käytännön sovelluksia. Patch nro 19. *Unipolaari- ja bipolaarisignaali* toimii johdantona amplitudi- ja rengasmodulaatioille. Amplitudi- ja rengasmodulaatioiden vertailua varten on oma patchinsä nro 22.

Patch 23. Tremolo kuuluu kolmanteen kategoriaan: efekti- ja prosessointilaitteiden toimintaperiaatteita esitteleviin patcheihin, mutta koska se on amplitudimodulaation eräs alalaji, on tremolo sijoitettu amplitudi- ja rengasmodulaatioiden yhteyteen.

Oppimistavoitteina on oppia periaatteita näiden äänisynteesimetodien toiminnasta sekä tulla tietoiseksi minkälaisia ohjattavia parametrejä kussakin äänisynteesimetodissa on.

9.3 Efekti- ja prosessointilaitteiden toimintaperiaatteita esittelevät patchit ja niiden oppimistavoitteet

Kolmannen kategorian patcheilla toteutetaan audiosignaaliin efektointia ja prosessointia. Näitä efektilaite-patcheja ovat: 28. *Viive 1*, 29. *Viive 2*, 30. *Läpsähdysviive*, 31. *Viiveillä toteutettu pieni tila*, 32. *Kaiku*, 33. *Koorus*, 34. *Flangeri*, 35. *Feiseri*, 36. *Wah-wah* sekä 37. *Särö*.

Efektipatcheista viive 1, viive 2, läpsähdysviive, viiveillä toteutettu pieni tila, kaiku, koorus ja flangeri pohjautuvat viivästämisen erilaisille toteutustavoille.

Oppimistavoitteina on oppia periaatteita näiden efektilaitteiden toiminnasta sekä tulla tietoiseksi minkälaisia ohjattavia parametrejä kussakin efektilaitteessa on.

Patch 23. Tremolo kuuluisi olla myös tämän kategorian *efekti- ja prosessointilaitteiden toimintaperiaatteita esittelevissä* patcheissä, mutta koska se on amplitudimodulaation eräs alalaji, on tremolo sijoitettu amplitudi- ja rengasmodulaatioiden yhteyteen *Äänisynteesimetodien perusteita esittelevät patchit* -otsikon alle.

9.4 Muut Max-patchit ja niiden oppimistavoitteet

Tähän kategoriaan kuuluvat muut patchit, jotka eivät mahdu edeltäviin kolmeen kategoriaan. Nämä patchit ovat: 38. *Erilaiset johtotyypit*, 39. *Monitorointi-objektien esittely* ja 40. *Ulkoisen kontrollerin kytkeminen*.

Oppimistavoitteina on oppia tunnistamaan erilaiset johtotyypit, oppia monitoroimaan datan kulkua, jotta ongelmatilanteissa voi järkeillä ratkaisua ongelmaan sekä oppia liittämään ulkoinen kontrolleri Maxiin, jotta patchejä pääsee soittamaan omin käsin.

10 TEHTÄVÄPATCHIT

10.1 Tehtäväpatchien käyttöajatus

Oppilaat saavat tehtäväpatchin paperille printattuna, minkä jälkeen heidän tehtävänä on rakentaa patch tietokoneelle Max -ohjelmaan. Tämä mahdollistaa oppilaiden välittömän osallistumisen käytännön työhön. Patchin rakennusprosessin aikana oppilaille tulee yleensä vastaan ongelmia sekä herää kysymyksiä patchin rakentamisesta ja sen toiminnasta. Kun patch on saatu toimimaan, opettaja voi antaa ryhmälle tehtäväksi selittää patchin kaikkien objektien funktiot. Näin ryhmä päätyy opettamaan itse itseään. Jäljelle jääneiden objektien funktiot jäävät tällöin opettajan selitettäväksi. Toinen vaihtoehto opettajalle, sen jälkeen kun patch on saatu toimimaan, on pitää luentotyypinen yksityiskohtainen selvitys patchin rakenteesta ja sen objektien funktioista. Yhdistelmä edellämainituista menetelmistä tuottaa yleensä parhaan tuloksen.

Jokaisen valmiiksi tehdyn patchin jälkeen on toivottavaa, että oppilaat itse miettivät, miten käsillä olevaa patchia voidaan varioida ja kehittää eteenpäin.

10.2 Toistuvat objektirakenteet

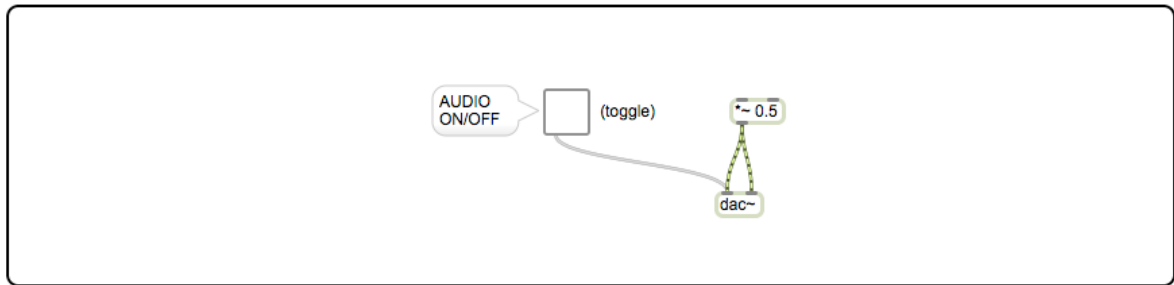
Seuraavia objektirakenteita on käytetty monien patchien yhteydessä. Näitä objekteja ja niistä muodostettujen objektikokonaisuuksien toimintaa ei selitetä jatkossa enää yksittäisten patch-kuvauksien yhteydessä.

10.2.1 Audio on/off-rakenne

Toggle - ja *~ -objektit on kytketty *dac~* -objektiin. *Toggle* -objekti toimii kytkimenä, jolla audio (= MSP) kytketään päälle ja pois. *~ -objektilla säädetään *~ -objektiin tulevan audiosignaalin amplitudi tasolle, jolla signaali ei useimmissa tapauksissa säröydy. *~ -objektin argumentilla, tässä tapauksessa 0,5, säädetään tämä taso. Signaalin amplitudin ei tulisi ylittää tasoa 1,0 tai alittaa tasoa -1,0 muulloin kuin tarkoitushakuisissa säröyttämisyhteyksissä.

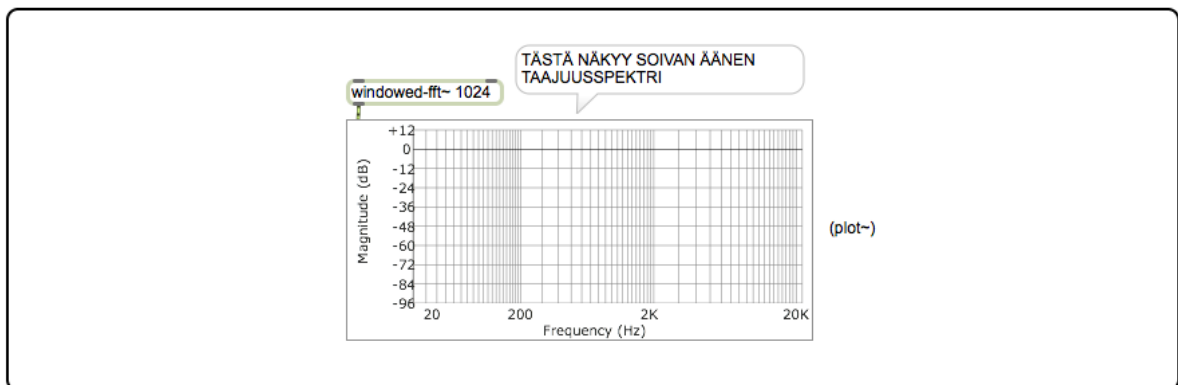
Mikäli audiosignaali säröytyy, kytketään *~ -objektin oikeanpuoleiseen sisäänmenoon *flonum* -objekti, jonka avulla voidaan *~ -objektille syöttää lukua 0,5 pienempi arvo säröytymisen estämiseksi. *Dac~* -objekti luo yhteyden Max-ohjelmasta tietokoneen

kaiuttimeen, jotta patchin generoima ääni saadaan kuulumaan. *Dac* -lyhennys tulee sanoista *digital-to-analog converter*.



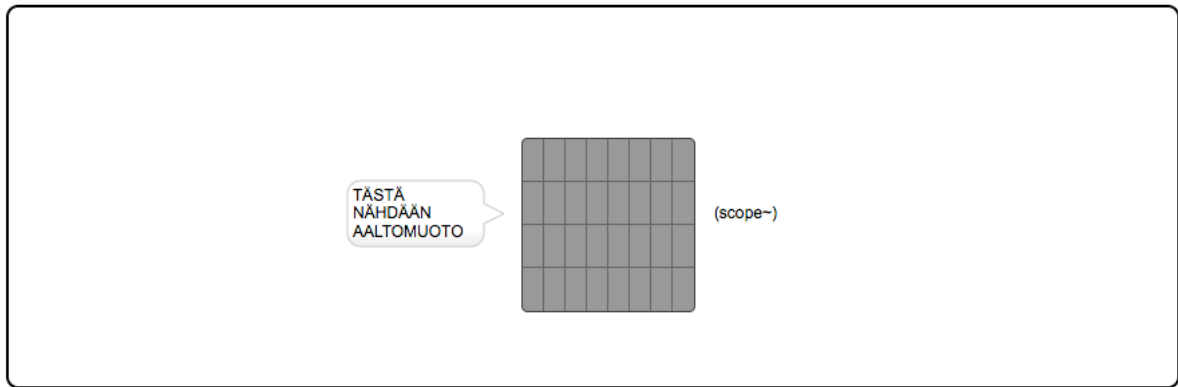
10.2.2 Spektrianalysaattori -rakenne

Windowed fft~ - ja *plot~* -objektit yhdessä muodostavat spektrianalysaattorin, joka tekee graafisen visualisoinnin audiosignaalin spektristä. Helpoin tapa toteuttaa *plot~* -objekti spektrianalysaattorimoodissa on se, että ensin luodaan n -kirjaimella tyhjä objekti, jonka sisään kirjoitetaan objektin nimeksi: *plot~*. Tämän jälkeen se aktivoidaan klikkaamalla hiirellä objektia. Sitten Help -valikosta valitaan “Open *plot~* -help” -kohta. Auenneesta ikkunasta klikataan yläreunan spectral -välilehteä. Tällä spectral -välilehdellä *plot~* -objekti on spektrianalysaattorimoodissa. Tämän jälkeen View -valikosta valitaan Edit -kohta, jolla patchin lukitus avataan ja päästään editoimaan tätä patchiä. Tässä editointi -moodissa klikataan *plot~* -objektiin, jonka jälkeen voidaan Edit -valikosta valita Copy. Nyt *plot~* -objekti (spektrianalysaattorimoodissa) on talletettu tietokoneen leikepöytään, josta se on mahdollista sijoittaa mihin tahansa käyttäjän rakentamaan Max-patchiin.



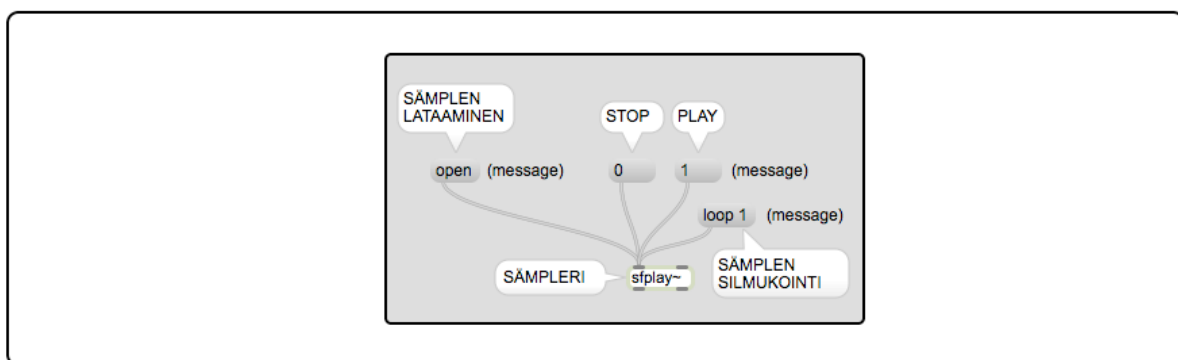
10.2.3 Audiosignaalin monitorointi

Scope~ -objekti piirtää kuvaajan siihen kytketystä audiosignaalista. X-akseli on aika-akseli ja y-akseli on audiosignaalin amplitudi (*äänenvoimakkuus*).



10.2.4 Sämplerin käyttö -rakenne

Sfplay~ -objektiin on kytketty *message* -objekteja. *Sfplay~* -objekti on sämpleri, joka pystyy soittamaan äänitiedostoja tietokoneen kovalevyiltä. *Message* -objektien viesteillä ohjataan *sfplay~* -objektin käyttöä. *Open*-viestillä ladataan sämpleriin uusi sample, *1*-viestillä käynnistetään äänitiedoston toistaminen, *0*-viestillä pysäytetään äänitiedoston toistaminen, "*loop 1*" -viestillä laitetaan äänitiedoston silmukointi päälle.



10.2.5 Verhokäyrän toteutus -rakenne

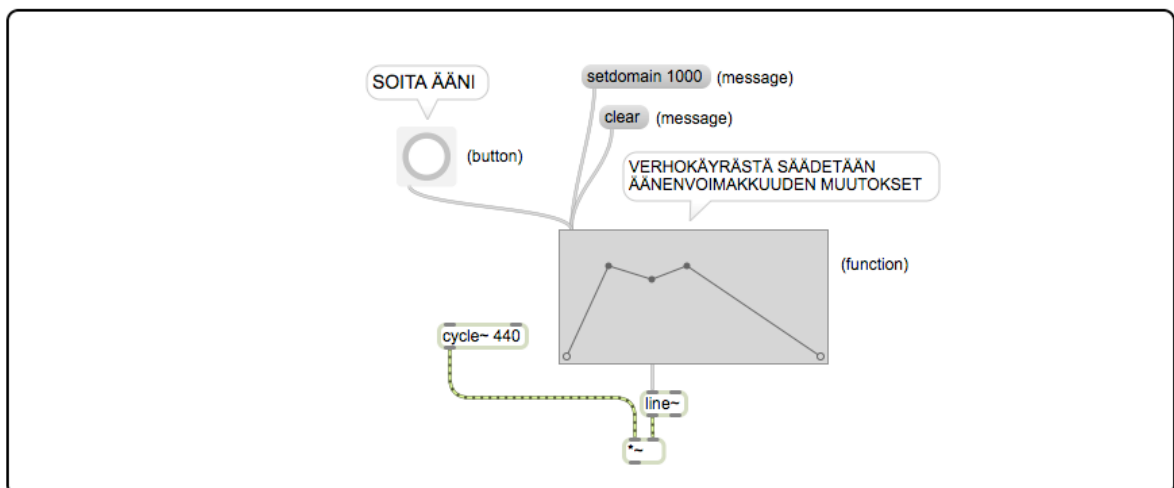
Verhokäyrä toteutetaan seuraavilla objekteilla: *function*, *line~*, *cycle~*, **~*, *button*, *setdomain-* ja *clear* -viestit. Verhokäyrän muoto määritetään klikkailemalla *function* -objektiin pisteitä

haluttuihin kohtiin. Verhokäyrän kokonaiskesto määritetään *setdomain* -viesteillä *message*-objekteissa. Luku *setdomain* -viestin perässä määrittää verhokäyrän keston millisekunneissa. Yläreunan *button* -objektilla toistetaan itse ääni. Silloin kun *function* -objektin viimeinen piste on oikeassa alareunassa oleva valkoinen piste, ääni päättyy hiljaisuuteen. Mikäli viimeinen piste ei ole alareunassa valkoisena pisteenä, vaan on ylempänä mustana pisteenä, ääni jatkaa soimista. *Clear* -viestillä *function* -objekti tyhjennetään, minkä jälkeen voidaan aloittaa alusta verhokäyrän muotoileminen. *Number* -objektilla määritetään taajuus *cycle~* -objektille. Osassa patcheista tämä *number*-objekti puuttuu. *Function* -objektin alareunan toinen ulostulo vasemmasta reunasta on kytketty *line~* objektiin. Tästä *function* -objektin ulostulosta data tulee ulos seuraavan tyyppisessä muodossa:

0.0000

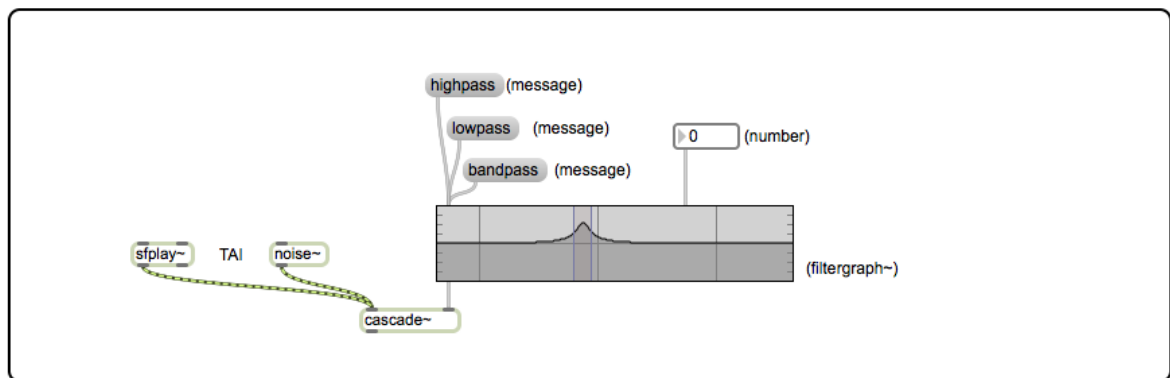
0.693182 1941.148938 0. 3058.510742.

Ensimmäinen luku (0.0000) edustaa ensimmäisen pisteen amplitudiarvoa. Tämän jälkeiset luvut listassa tulevat pareittain siten, että ensimmäinen luku on seuraavan pisteen amplitudiarvo ja toinen luku ilmaisee keston millisekunneissa edeltävästä pisteestä mitattuna. *Function* -objektin perässä oleva *line~* objekti toteuttaa lineaarisen interpolaation amplitudiarvojen välille. *Line~* -objektin jälkeinen **~* -objekti tekee kertolaskun *line~* -objektista tulevien verhokäyrän amplitudiarvojen ja *cycle~* -objektista tulevan signaaliin välille.



10.2.6 Äänilähteiden spektrin suodatus -rakenne

Erilaisia suotimia ja taajuuskorjaimia toteutetaan *cascade~*, *filtergraph~*, *number* ja *message* -objektien kokonaisuudella. Näiden objektien lisäksi mukana on myös äänilähteenä joko *noise~* tai *sfplay~* -objekti. *Filtergraph~* -objekti toimii graafisena käyttöliittymänä rajataajuuden määrittämiseen. *Filtergraph~* -objektilla ohjataan *cascade~* -objektia, joka muodostuu sarjaankytketyistä biquad -suotimista. Biquad -suodin on toisen kertaluvun rekursiivinen, lineaarinen suodatin, jonka siirtofunktiossa on kaksi nollaa ja kaksi napaa. *Number* -objektilla säädetään suotimen rajataajuus. *Message*-objektien viesteillä (*highpass*, *lowpass*, *bandpass* etc.) määritetään suodintyyppi (*ylipäästösuodin*, *alipäästösuodin*, *kaistanpäästösuodin* etc.). Äänilähteinä näiden kokonaisuuksien yhteydessä esiintyy joko *noise~* -objekti, joka on satunnaiskohinageneraattori, tai *sfplay~* -objekti, joka on sämpleri.



10.3 Johdatus Max-patchien rakentamiseen

Max-patchien rakentaminen aloitetaan tyhjälle työpohjalle (patcher). Patchien rakentamiseen tarvitaan kaksi taitoa. Pitää osata luoda erilaisia objekteja ja tehdä kytkentöjä niiden välille. Helppo ja yksinkertainen tapa luoda objekteja on käyttää *n-kirjain* -pikanäppäinkomentoa tyhjän objektin luomista varten ja tämän jälkeen kirjoittaa tyhjään objektiin sen nimi. Sen jälkeen se muuttuu aktiiviseksi ja käytettäväksi objektiksi. Osaan objekteista sen nimi jää näkyviin, mutta osa objekteista, esim. *toggle* tai *scope~*, muuttuvat graafisiksi kuvakkeiksi,

joissa nimi ei ole enää näkyvissä. Opinnäytetyön tehtäväpatcheissä objektien nimet on kirjoitettu systemaattisesti niiden oikealle puolelle sulkumerkkien sisään, mikäli sen nimi tai symboli ei ole näkyvissä objektissa itsessään. Objektin luomisen jälkeen siihen ilmestyvät mahdolliset sisäänmenot ja ulostulot. Mahdolliset sisäänmenot ovat objektin yläreunassa olevia pieniä mustia vaakaviivoja. Mahdolliset ulostulot ovat objektin alareunassa olevia pieniä mustia vaakaviivoja. Sisäänmenoja ja ulostuloja voi objekteissa olla nolla, yksi tai useampia.

Max -ohjelmasta löytyy kolmenlaisia objekteja: Max-, MSP- sekä Jitter-objektit. Tämän opinnäytetyön patcheissä ei ole käytetty lainkaan Jitter -objekteja, jotka on suunniteltu ensi sijaisesta kuvan generoinnin ja kuvankäsittelyn tarpeita varten. Jitter -objektit tunnistaa siitä, että niiden nimet alkavat jit -kirjainlyhenteellä, esim. jit.window tai jit.matrix. MSP -objektit tunnistaa siitä, että niiden nimien lopussa on *tilde*-merkki: ~ . Muut objektit ilman jit-aluketta tai tilde-merkkiä nimen lopussa ovat Max-objekteja.

Max-ohjelman virtuaaliset kytkentäjohtot ovat joko yksivärisiä johtoja tai kaksivärisiä katkoviivajohtoja. Ks. patchkuvausteksti patchistä nro 38 *Erilaiset johtotyypit*, jossa on yksityiskohtainen selvitys erilaisista johtotyypeistä ja niiden käytöstä.

Kytkenät objektien välille tehdään siten, että hiiren kursori viedään niin lähelle yläpuolella olevan objektin alareunassa olevaa ulostuloa (outlet), että näkyviin ilmestyy pyöreä ympyrä. Sen jälkeen painetaan hiiri pohjaan. Sitten hiiren kursori viedään näppäin pohjaan painettuna niin lähelle alapuolella olevan objektin yläreunassa olevaa sisäänmenoa (inlet), että näkyviin ilmestyy jälleen pyöreä ympyrä. Sitten vapautetaan hiiren näppäin. Kytkenä objektien välille on luotu ja johto ilmestyy näkyviin. Osa näistä ympyröistä on sinisiä ja osa punaisia. Ympyrän sininen väri ilmaisee "*kylmän*" sisäänmenon ja punainen "*kuuman*" sisäänmenon. Kylmällä sisäänmenolla tarkoitetaan sitä, että otettuaan vastaan luvun tai muun datayksikön, objekti ei tee sen jälkeen enää mitään. Kuumalla sisäänmenolla tarkoitetaan sitä, että otettuaan vastaan luvun tai muun datayksikön, objekti tekee sen toiminnon, jonka se on ohjelmoitu tekemään ja toimittaa ulostulosta lopputuloksen. Objekteille voidaan määrittää argumentteja. Nämä argumentit kirjoitetaan objektin nimen viereen oikealle puolelle välilyönnillä objektin nimestä erotettuna. Mikäli argumentteja on useampi, kukin argumentti kirjoitetaan objektin nimen oikealle puolelle kukin välilyönnillä toisistaan erotettuina. Argumentit voivat olla mm. lukuja, kirjaimia, symboleja ja viestejä.

10.4 Kirjalliset kuvaukset tehtäväpatchien toiminnasta

Tästä alkaa jakso, jossa jokainen patch käydään läpi yksityiskohtaisesti selittäen, mitä patch tekee ja mikä on kunkin objektin funktio patchissä. Lisäksi selvitetään, mitkä ovat kunkin patchin oppimistavoitteet.

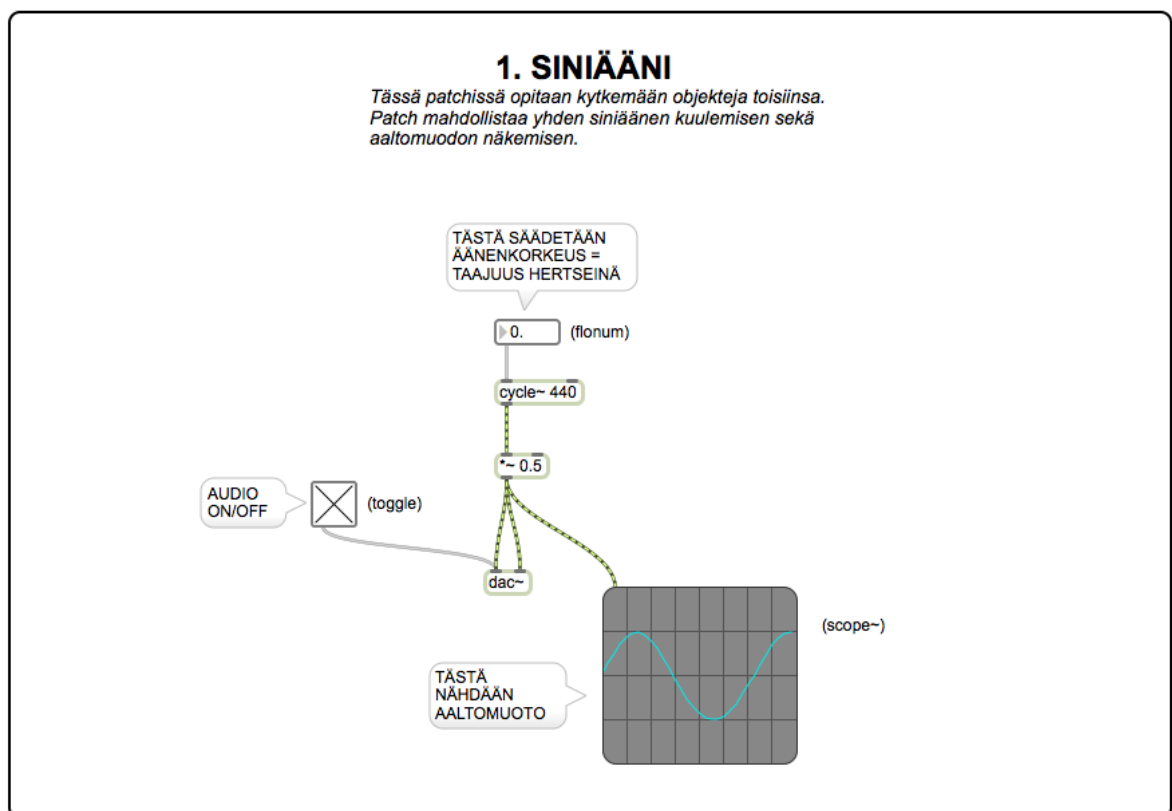
10.4.1 Ääniopin ja musiikkiteknologian peruskäsitteitä esittelevät patchit

PATCH 1: SINIÄÄNI

Patchin toiminta ja objektien funktiot

Tässä ensimmäisessä Max-patchissä tutustutaan Max-ohjelman käyttöön. Siniääni-patch mahdollistaa yhden sini-/kosiniäänen kuulemisen sekä aaltomuodon näkemisen. Sini- ja kosiniäänen aaltomuodot ovat muuten identtiset, mutta ne ovat 90 asteen vaihesiirrossa suhteessa toisiinsa.

Flonum -objekti on desimaalilukuobjekti, jolla määritetään kuultavan siniäänen taajuus. *Cycle~* -objekti generoi kosinimuotoisen aaltotaulukon, jota ohjelma toistaa sekunnin aikana niin monta kertaa, kuin *cycle~* -objektiin liitetyllä *flonum* -objektilla on määritetty.



*~ -objekti toimii "*vahvistimena*". *~ -objektin vasempaan sisäänmenoon tulevaa audiosignaalia kerrotaan *~ -objektin argumentilla, joka voi olla joko kokonaisluku tai desimaaliluku. Tällä kertolaskulla säädetään audiosignaalin äänenvoimakkuutta. *~ -objektin ulostulo on kytketty *dac~* -objektin molempiin sisäänmenoihin, jotta siniääni kuuluu molemmista kaiuttimista. *Dac~* -objekti luo yhteyden Max-ohjelmasta tietokoneen kaiuttimiin tai kuulokkeisiin tai tietokoneeseen kytkettyyn äänikorttiin liitettyihin kaiuttimiin tai kuulokkeisiin.

PATCH 2: ÄÄNIOPIN JA MUSIIKKITEKNOLOGIAN PERUSKÄSITTEITÄ

Patchin toiminta ja objektien funktiot

Patchin avulla voidaan opiskella ääneen liittyviä peruskäsitteitä ja ilmiöitä, kuten taajuus, amplitudi, aallonpituus, äänennopeus, ihmisen kuuloalue, infra- ja ultraäänet. Patchillä voi kokeilla ihmisen kuuloalueen rajoja; kuinka korkeita tai matalia ääniä itse kuulee. Aiheeseen liittyvä jatkokysymys on: Miksi mitään ei välttämättä kuulekaan, kun taajuus on 30-50 Hz:n välillä, vaikka ihmisen kuuloalueen on määritelty menevän noin 20 Hz:iin asti? Syynä voi olla esimerkiksi tietokoneen pienet heikkolaatuiset kaiuttimet, joiden taajuusvaste (= kyky toistaa taajuuksia), ei mene niin matalalle kuin mitä ihmisen korva kykenee kuulemaan.

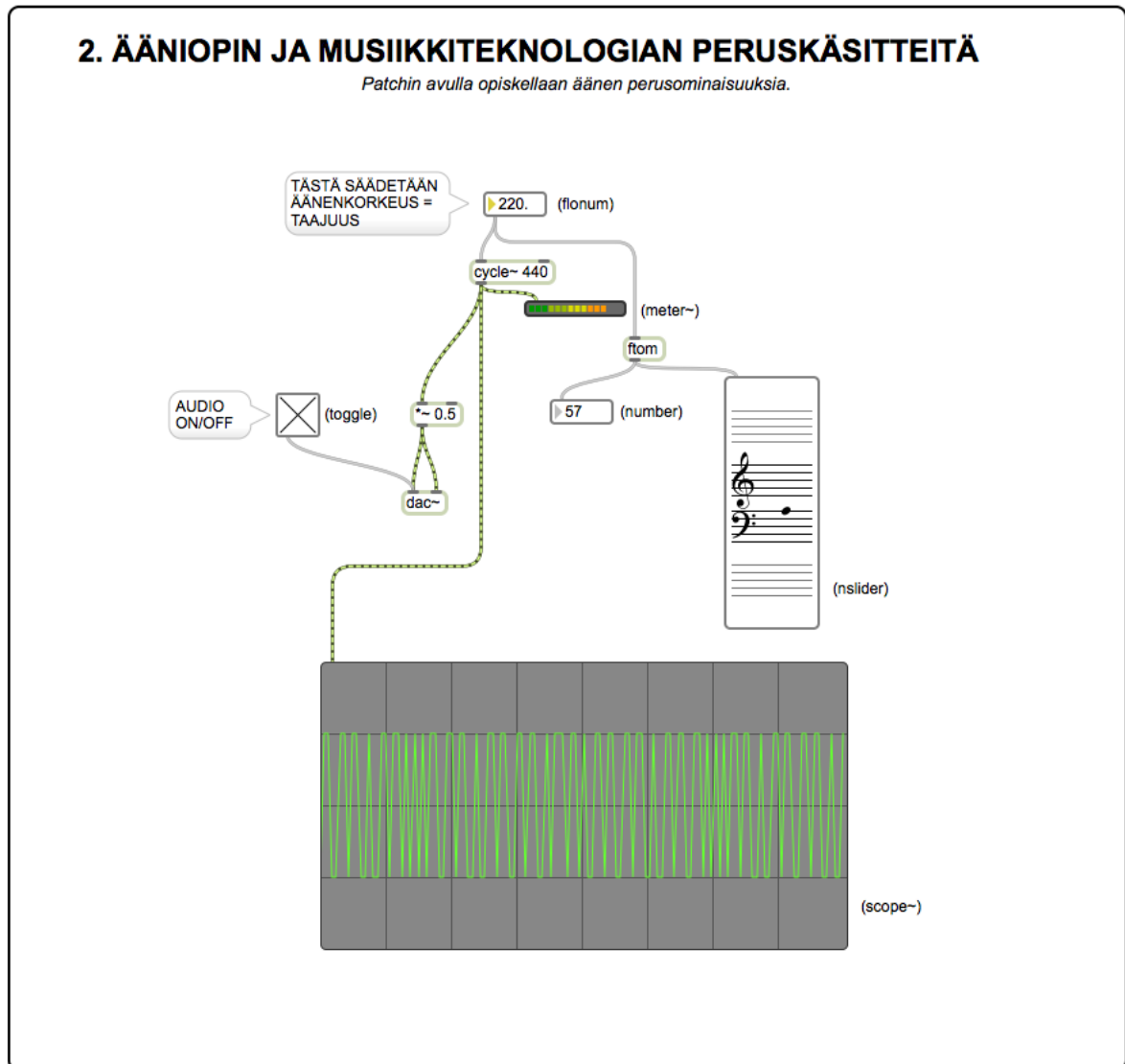
Tällä patchillä voidaan käyttäjän määrittämän taajuuden synnyttämästä aaltomuodosta nähdä samanaikaisesti reaaliaikainen graafinen kuva, kuulla ääni sekä nähdä nuotti nuottiviivastolla.

Patchillä on tarkoitus havainnollistaa ja luoda yhteys käyttäjän määrittämän taajuutta edustavan luvun, kuulohavainnon ja näköhavainnon välille, ja sitä kautta pyrkiä helpottamaan asioiden ymmärtämistä.

Flonum -objekti on desimaalilukuobjekti, jolla määritetään kuultavan siniäänien taajuus. *Cycle~* -objekti generoi tämän sini-/kosinimuotoisen äänen. *Meter~* -objekti monitoroi *cycle~* objektin amplitudin (= äänenvoimakkuus). *Ftom* -objektin kirjaimet ovat lyhenys sanoista frequency to midi. *Ftom* -objekti muuttaa sisään tulevan taajuutta edustavan luvun luvuksi, joka edustaa midinuottia. *Nslider*-objektin vasempaan sisäänmenoon syötetyt kokonaisluvut *nslider* tulkitsee sävelkorkeuden midi-arvoina ja nuotintaa ne viivastolle.

Muita vaihtoehtoja

*~ -objektin argumentiksi voidaan vaihtaa 0.5:n tilalle ensin 0.2 ja sen jälkeen 0.8. *Scope~* -objektista voidaan seurata miten kertoimen muutos vaikuttaa piirtyvään aaltomuotoon. Kertoimen muuttamiseksi luodaan uusi *flonum* -objekti, joka on desimaaliluku-objekti. Tämä *flonum* -objekti liitetään *~ -objektin oikean puoleiseen sisäänmenoon, jonka jälkeen voidaan syöttää kerroin tästä *flonum* -objektista.

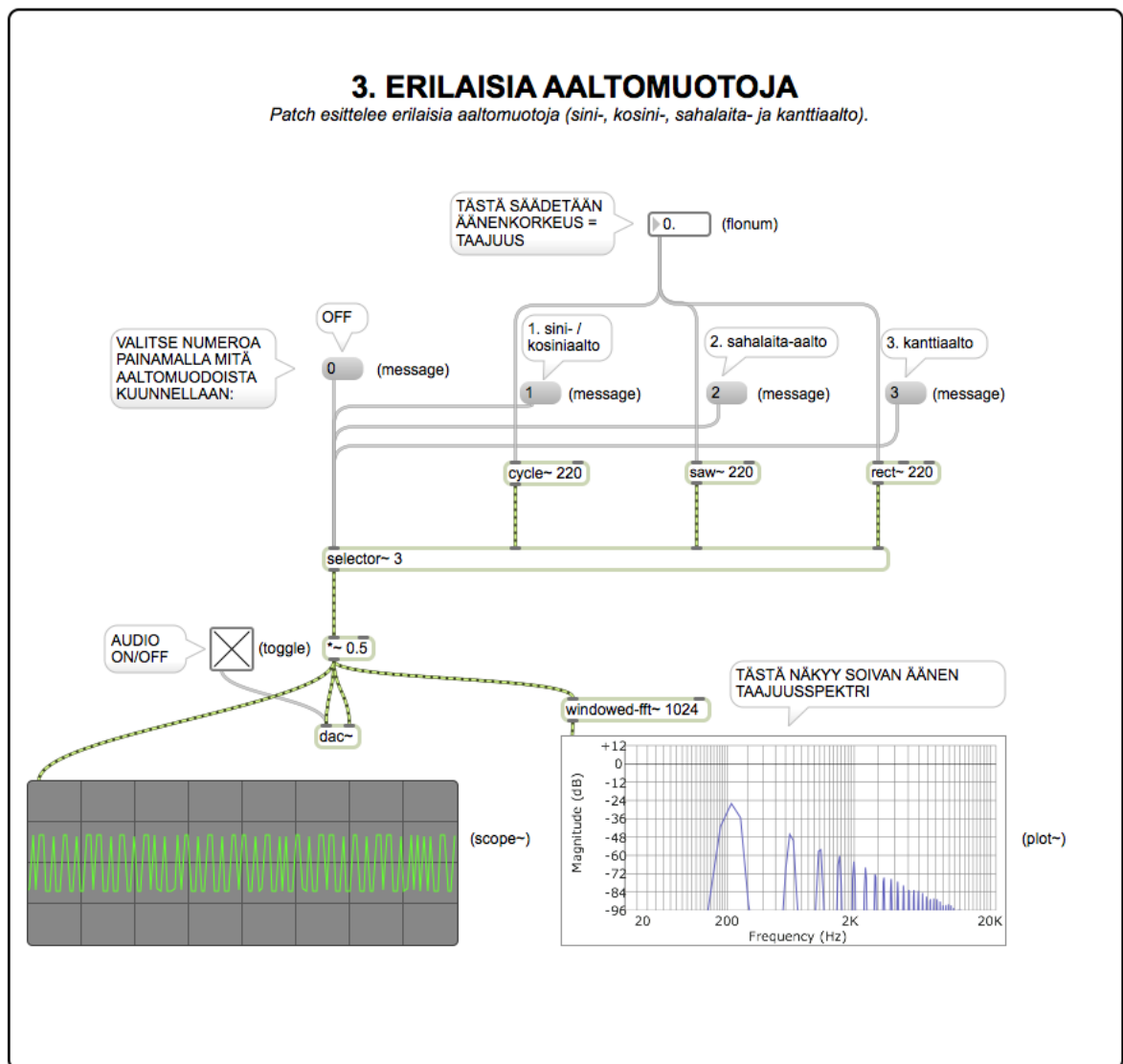


PATCH 3: ERILAISIA AALTOMUOTOJA

Patchin toiminta ja objektien funktiot

Patchillä voi kuunnella erilaisia aaltomuotoja, kuten sini-/kosini-, sahalaita- ja kanttiaaltoja ja nähdä äänien spektrisisällön *plot~* -objektista.

Flonum -objektilla, joka on desimaalilukuobjekti, määritetään taajuus (äänenkorkeus). Patchissä valittu taajuus on mahdollista kuunnella kolmella erilaisella aaltomuodolla. Nämä aaltomuodot ovat: 1) sini-/kosiniaalto 2) sahalaita-aalto ja 3) kanttiaalto. Sini-/kosiniaalto on toteutettu *cycle~* objektilla, sahalaita-aalto *saw~* objektilla ja kanttiaalto *rect~* objektilla. *Selector~* objekti toimii valitsimena, joka mahdollistaa lukuisista äänilähteistä yhden äänilähteen kuuntelemisen kerrallaan. *Message* -objekteissa (viestit) oleviin numeroihin klikkaamalla tehdään valinta kuunneltavasta äänilähteestä.

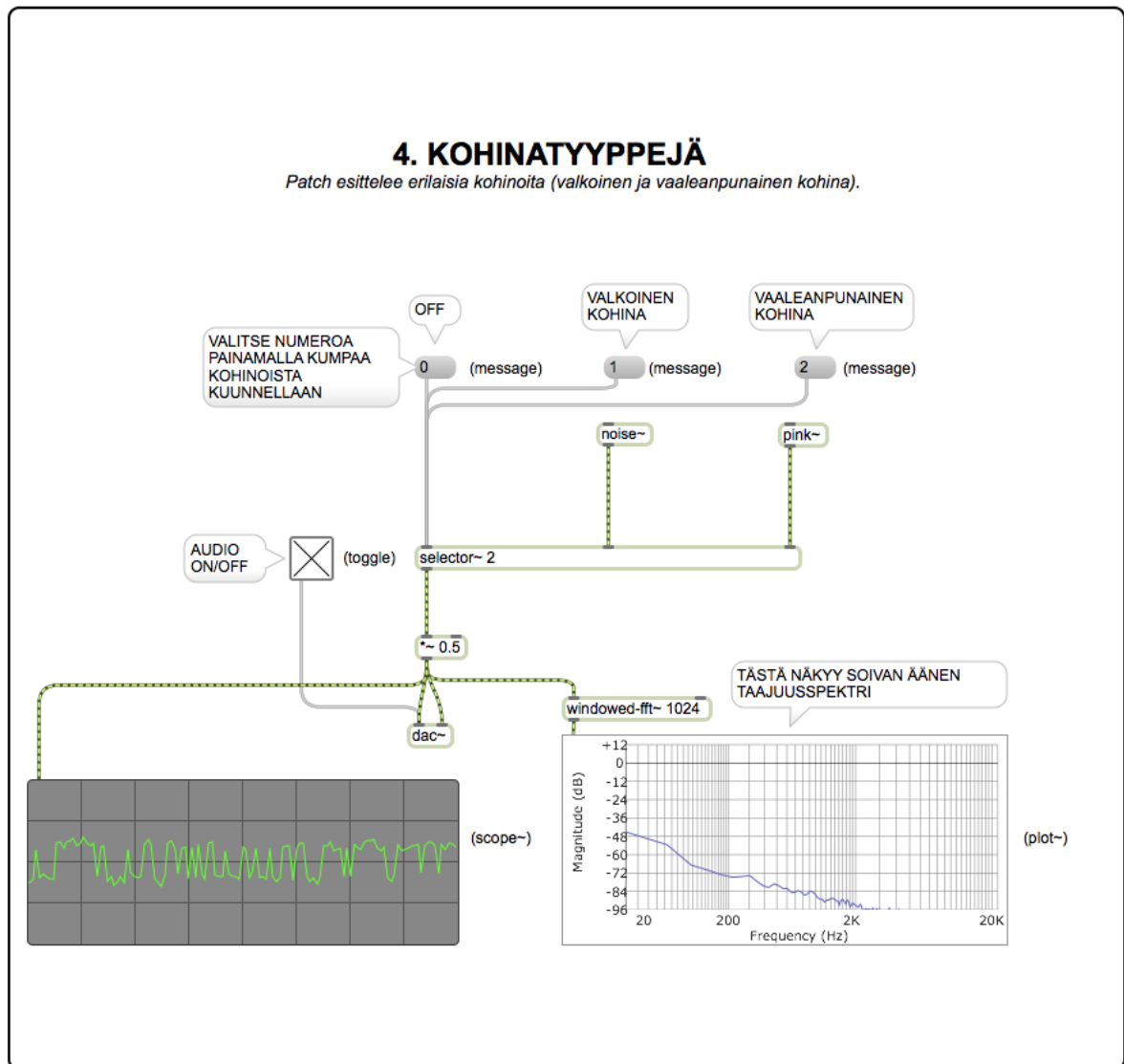


PATCH 4: KOHINATYYPPEJÄ

Patchin toiminta ja objektien funktiot

Patchillä voi kuunnella miltä valkoinen ja vaaleanpunainen kohina kuulostavat. Valkoisen kohinan tehospektri on vakio tarkastellulla taajuusalueella (White Noise 2014). Vaaleanpunaista kohinaa kutsutaan myös 1/f-kohinaksi, jossa tehospektritiheys on kääntäen verrannollinen signaalin taajuuteen. Siinä oktaavit sisältävät yhtä suuret kohinatehot (Pink Noise 2014).

Neljännän patchin toiminta on muuten samanlainen kuin patchin nro 3 *Erilaisia aaltomuotoja* sillä erotuksella, että kolmannen patchin aaltomuodot ovat säveltasollisia kun taas neljännessä patchissä äänilähteet ovat säveltasottomia äänilähteitä, kohinoita.

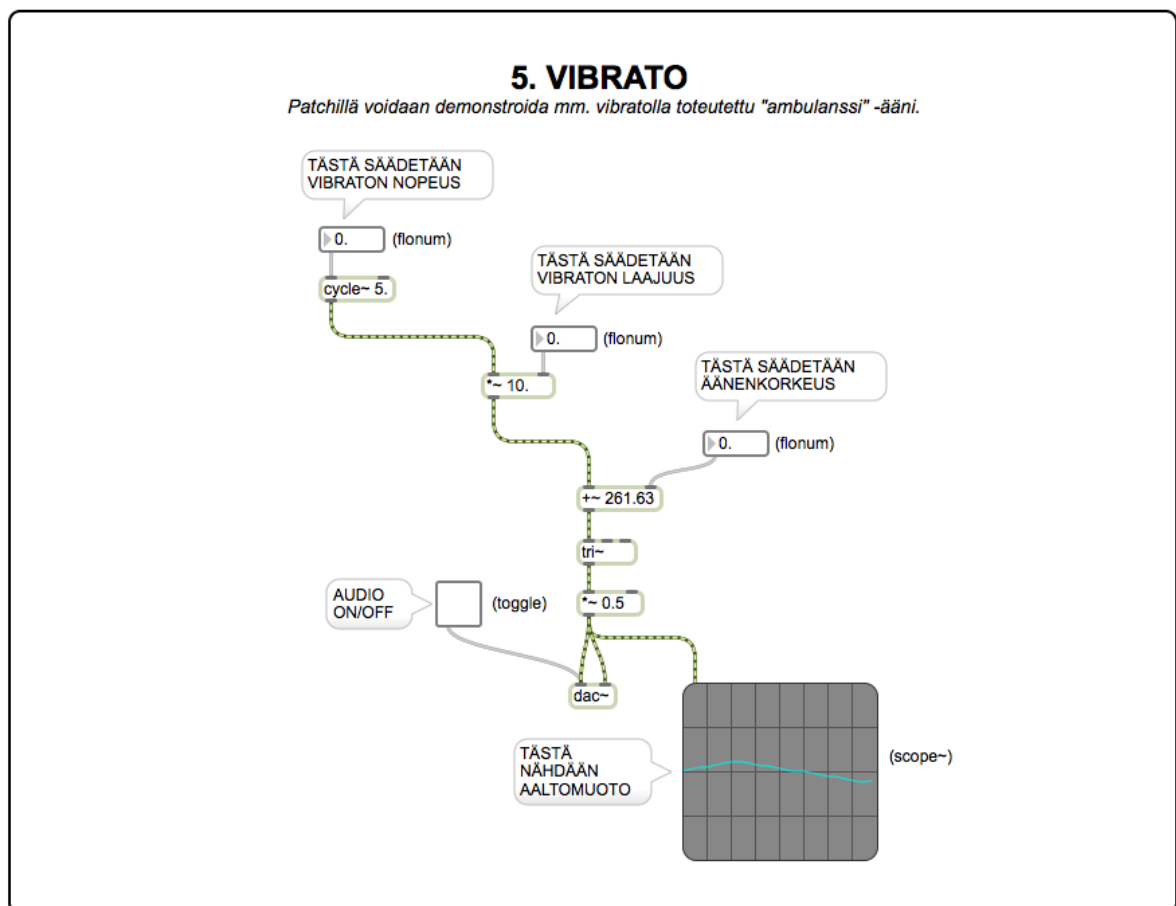


PATCH 5: VIBRATO

Patchin toiminta ja objektien funktiot

Vibrato on taajuuden hidasta variointia (Roads 1996: 215). Tällä patchillä toteutetaan kolmioaaltoon vibrato. Vibraton muuttujat ovat vibraton nopeus ja vibraton laajuus. Soiva oskillaattori tässä patchissä on *tri~* -objekti. Aloitusarvona patchissä soivan oskillaattorin taajuudelle on 261,63 hertsiä (*I-viivainen c-sävel*). Tämä luku välittyy *tri~* -objektille *+~* -objektilta. Taajuus määritetään “*tästä säädetään äänenkorkeus*” -puhekuplan yhteydessä olevalla *flonum* -objektilla (*desimaalilukuobjekti*).

Vibraton laajuus määritetään “*Tästä säädetään vibraton laajuus*” -puhekuplan yhteydessä olevalla *flonum* -objektilla. Vibraton nopeus määritetään “*Tästä säädetään vibraton nopeus*” -puhekuplan yhteydessä olevalla *flonum* -objektilla. Vibraton laajuus toteutetaan kertomalla *cycle~* -objektista tuleva audiosignaali halutulla **~* -objektiin syötetyllä desimaaliluvulla. *+~* objekti summaa perustaajuuteen *cycle~* -objektin ja **~* -



objektin muodostaman kokonaisuuden synnyttämän vibraton. *Tri~* -objekti on patchin oskillaattori (äänilähde).

Patchillä voi demonstroida ambulanssin sireenin seuraavalla tavalla: säädä vibraton nopeus hitaaksi, esim. 1 Hz ja vibraton laajuus laajaksi, esim. 100, ja soivan oskillaattorin taajuudeksi n. 800 Hz.

PATCH 6: VERHOKÄYRÄ

Patchin toiminta ja objektien funktiot

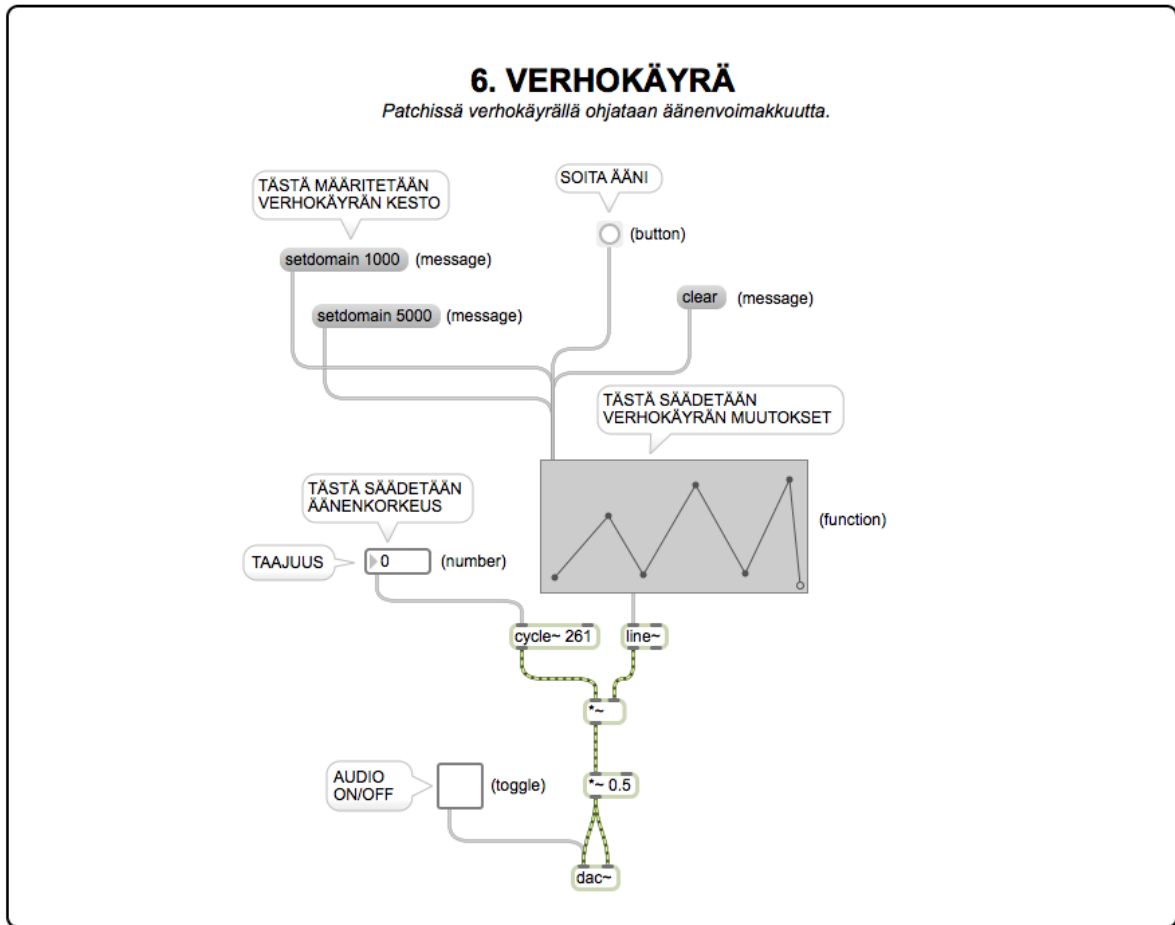
Patchissä ohjataan verhokäyrällä äänenvoimakkuutta. Verhokäyrän muita nimityksiä ovat mm. vaippakäyrä ja envelope. Silloin kun verhokäyrässä on neljä eri vaihetta puhutaan ADSR -verhokäyrästä. Kirjainlyhennykset tulevat sanoista: Attack, Decay, Sustain ja Release. Näistä neljästä kolme, Attack, Decay ja Release määritetään kestoina, kun taas sustain määritetään tasona.

Verhokäyrän muutokset on toteutettu *function-* ja *line~* -objektien avulla. Verhokäyrän muoto määritetään klikkailemalla *function* -objektiin pisteitä haluttuihin kohtiin. Verhokäyrän kokonaiskesto määritetään *setdomain* -viesteillä *message*-objekteissa. Luku *setdomain* -viestin perässä määrittää verhokäyrän keston millisekunneissa. Yläreunan *button* -objektilla soitetään itse ääni. Silloin kun *function* -objektin viimeinen piste on oikeassa alareunassa oleva valkoinen piste, ääni päättyy hiljaisuuteen. Mikäli viimeinen piste ei ole alareunassa valkoisena pisteenä, vaan on ylempänä mustana pisteenä, ääni jatkaa soimista. *Clear* -viestillä *function* -objektin saa tyhjennettyä, minkä jälkeen voidaan aloittaa alusta verhokäyrän muotoileminen. *Number* -objektilla määritetään taajuus *cycle~* -objektille, jonka lähtöarvona on 261 Hz (*yksiviivainen c-sävel*). *Function* -objektin alareunan toinen ulostulo vasemmasta reunasta on kytketty *line~* -objektiin. Tästä *function* -objektin ulostulosta data tulee ulos seuraavan tyyppisessä muodossa:

0.0000

0.693182 1941.148938 0.3058.510742.

Ensimmäinen luku (0.000) edustaa ensimmäisen pisteen amplitudiarvoa. Tämän jälkeiset luvut listassa tulevat pareittain siten, että ensimmäinen luku on seuraavan pisteen amplitudiarvo ja toinen luku kertoo keston millisekunneissa edeltävästä pisteestä mitattuna.

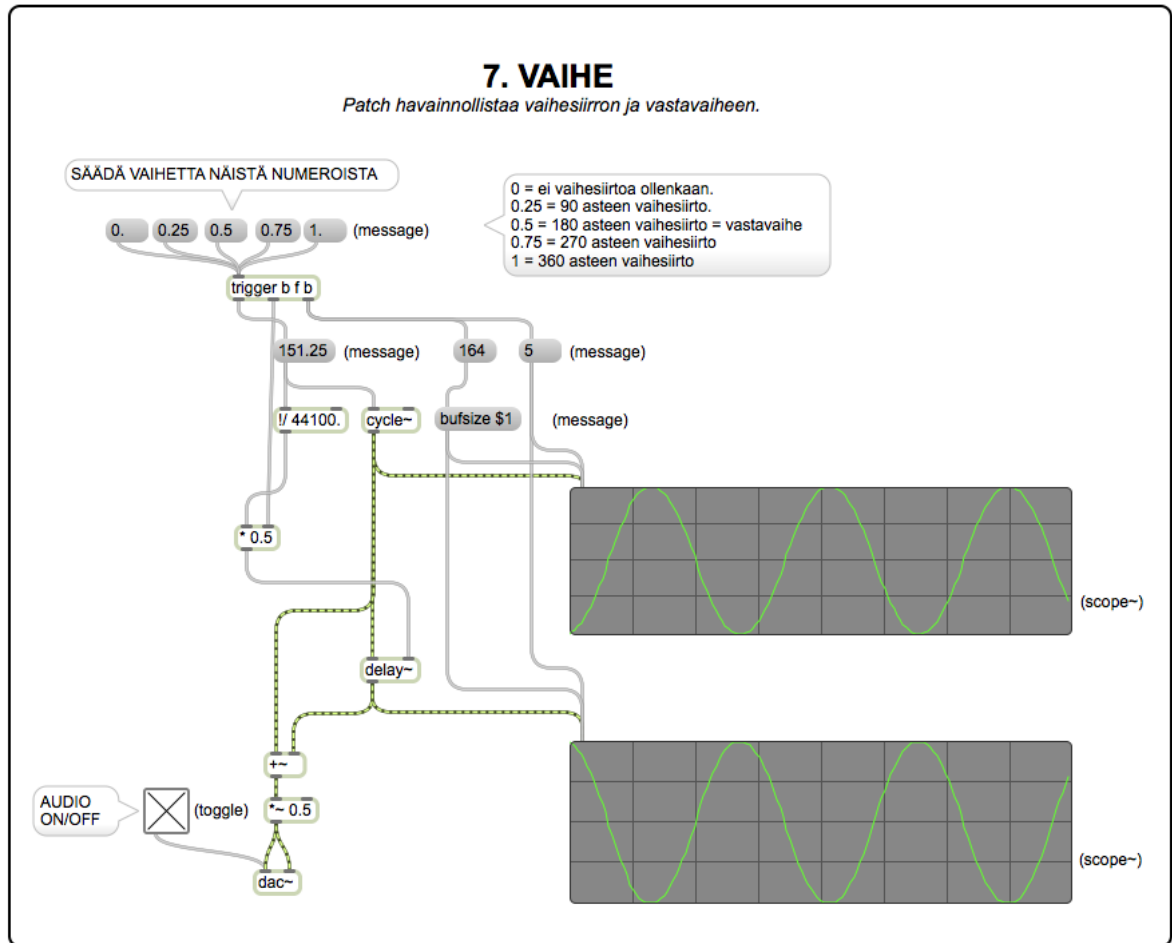


Function -objektin perässä oleva *line~* objekti toteuttaa lineaarisen interpolaation amplitudiarvojen välille. *Line~* -objektin jälkeinen **~* -objekti tekee kertolaskun *line~* -objektista tulevien verhokäyrän amplitudiarvojen ja *cycle~* -objektista tulevan audiosignaaliin välille.

PATCH 7: VAIHE

Patchin toiminta ja objektien funktiot

Tässä patchissä tutustutaan vaihe-käsitteeseen. Patch havainnollistaa 90 asteen, 180 asteen, 270 asteen sekä 360 asteen vaihesiirron. Kun signaali on 180 asteen vaihesiirroksessa, puhutaan vastavaiheesta. Kun signaali summataan vastavaiheessa olevan signaalin kanssa, kumoutuvat nämä signaalit pois, koska vastavaiheessa olevien signaalien summa on nolla. Patchissä käyttäjä määrittää vaihesiirtymän klikkaamalla yläreunan desimaaliluku -viesteihin. *Message* -objektilla, jossa on luku nolla, ei saada aikaan lainkaan vaihesiirtymää, koska tällöin vaihesiirtymä on nolla astetta. *Message* -objektilla, jossa on lukuarvo 0.25, toteutetaan 90



asteen vaihesiirtymä. *Message* -objektilla, jossa on lukuarvo 0.5, toteutetaan 180 asteen (=vastavaihe) vaihesiirtymä. *Message* -objektilla, jossa on lukuarvo 0.75, toteutetaan 270 asteen vaihesiirtymä. *Message* -objektilla, jossa on lukuarvo yksi, toteutetaan 360 asteen vaihesiirtymä. *Trigger*-objektin sisällä olevat kirjaimet symbolisoivat seuraavia asioita: b = bang ja f = float. *Trigger* -objektin funktio on saada *trigger* -objektin jälkeiset tapahtumat toteutumaan käyttäjän haluamassa järjestyksessä. Ensimmäiseksi *trigger* -objekti välittää *bang* -käsken oikeanreunimmaisesta ulostulosta. Toiseksi *trigger* -objekti välittää sisääntulevan desimaaliluvun keskimmaisesta ulostulosta. Kolmanneksi ja viimeiseksi *trigger* -objekti välittää *bang* -käsken vasemmanreunimmaisesta ulostulosta. *Scope~* -objektin vasempaan sisäänmenoon välitettävien viestien, "5" ja "*bufsize \$1*", avulla saadaan *scope~* -objektin piirtämä kuva aaltomuodosta pysäytettyä paikalleen. "*Bufsize \$1*"-viestillä määritetään, kuinka monta sämpliä puskuriin voidaan tallettaa. "*Bufsize \$1*" -viesti sijoittaa \$1-arvoksi *bufsize* -viestin vasempaan sisäänmenoon syötetyn luvun, tässä tapauksessa luvun 164, jonka jälkeen se välittää *scope~* -objektille viestin "*bufsize 164*". *Cycle~* -objekti

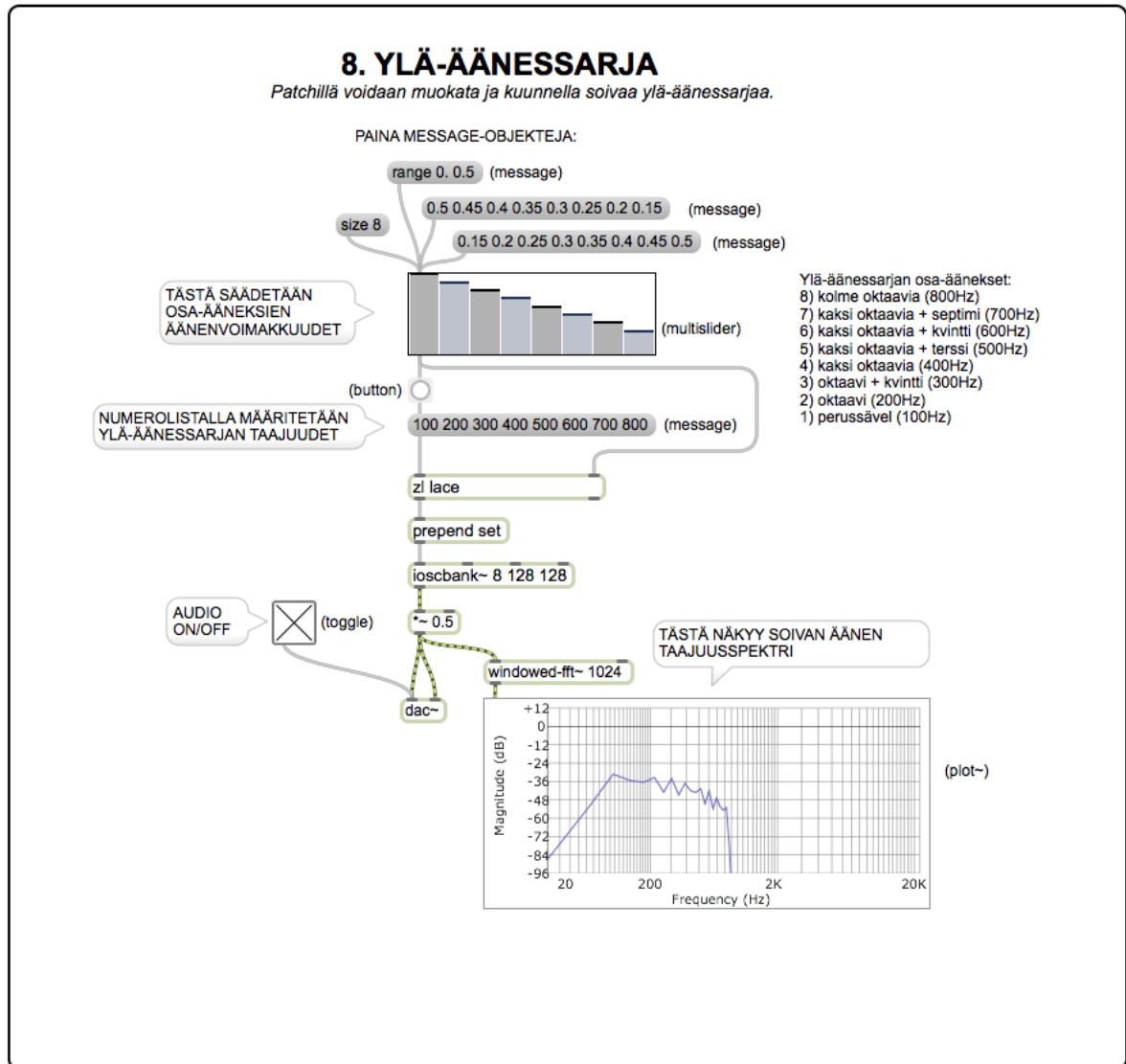
generoi 151,25 Hz:n taajuisen kosiniaallon. 44100 . -objektilla toteutetaan jakolasku: $44100/151.25 = 291,57$. Tämä luku 291,57 kertoo, kuinka monta sampleä käytetään yhteen kosiniaallon jaksoon, kun näytteenottotaajuutena on 44100 näytettä sekunnissa. Mikäli näytteenottotaajuutena käytetään jotain muuta näytteenottotaajuutta kuin 44100 Hz, tällöin tähän jakolaskuun tulee vaihtaa luvun 44100 tilalle käytetyn näytteenottotaajuuden luku. Tämä luku, 291,57, kerrotaan seuraavassa * - objektissa (*kertolaskuobjekti*) * -objektin oikeaan sisäänmenoon tulevalla käyttäjän valitsemalla desimaaliluvulla. Tämän kertolaskun tulo on luku, joka edustaa joko 90 asteen, 180 asteen, 270 asteen tai 360 asteen viivettä sampleinä ilmaistuna. *Delay~* -objektilla toteutetaan signaalin viivästäminen. Alempi *scope~* -objekti piirtää kuvan tästä viivästäytystä signaalista. Ylempi *scope~* -objekti piirtää kuvan viivästäättömästä signaalista. *+~* -objekti summaa alkuperäisen signaalin ja vaihesiirretyn signaalin toisiinsa.

Arkielämän esimerkki vastavaiheesta on kuultavissa symbaalia soitettaessa. Symbaalin lyömisen jälkeen toinen korva laitetaan noin kahden senttimetrin etäisyydelle symbaalin reunasta. Tämän jälkeen päätä liikutetaan hitaasti ylös-alas suunnassa siten, että korva on välillä hieman symbaalin reunan yläpuolella, välillä symbaalin reunan kohdalla ja välillä hieman symbaalin reunan alapuolella. Liikutettaessa korvaa edellä mainitulla tavalla on mahdollista havaita, että tietyssä kohdassa symbaalin ääni heikkenee voimakkaasti. Tässä kohdassa korvaan välittyy samanaikaisesti symbaalista sekä ylös- että alaspäin lähtevää värähtelyä. Nämä värähtelyt ovat tässä kuuntelupisteessä keskenään vastavaiheessa.

PATCH 8: YLÄ-ÄÄNESSARJA

Patchin toiminta ja objektien funktiot

Patchillä voi tuottaa, muokata ja kuunnella ylä-äänessarjaa, jossa on kahdeksan osäänestä. *Multislider* -objektilla määritetään osa-ääneksien äänenvoimakkuudet. "Size 8" -viestillä määritetään *multislider* -objektiin kahdeksan erillistä liukua, joista kukin ohjaa yhden osäänoksen äänenvoimakkuutta. "Range 0. 0.5" -viestillä määritetään multisliderin numeeriseksi toiminta-alueeksi 0:sta 0.5:een. Tämä alue on valittu sen vuoksi, että kun summataan kahdeksan osäänestä, joiden voimakkuus on maksimissaan 0.5, osäänesten summa-aaltomuoto ei säröytyisi patchin ulostulossa, *dac~* -objektissa. "Range 0. 0.5" -viestin



alapuolella on kaksi lukulistaa, jotka ovat preset-listoja. Niitä klikkaamalla välittyvät listoissa olevat lukuarvot multisliderin yksittäisten liukujen arvoiksi. *Multislider* -objektin alapuolella olevasta *button* -objektista lähtevä käsky välittää *button* -objektin alapuolella olevan lukulistan *zl lace* -objektille. Tämän listan luvut edustavat ylä-äänessarjan taajuuksia. *Zl lace* -objekti liittää yhteen sen kahteen yläreunan sisäänmenoon tulevat *message* -viestilistat siten, että *zl lace* -objektin alareunan ulostulosta tulee ulos yksi lista, johon *zl lace* -objekti syöttää kaikki luvut yksitellen vuoron perään sisääntulevista listoista. *Prepend* -objekti liittää *zl lace* -objektista tulevan lukulistan alkuun *set* -sanana. *Ioscbank~* -objekti on oskillaattoripankki, joka pitää sisällään lukuisia oskillaattoreita. *Ioscbank~* -objektin ensimmäisellä argumentilla määritetään oskillaattorien lukumäärä. Tässä patchissä oskillaattoreista kahdeksan on käytössä. *Ioscbank~* -objektin toinen argumentti on nimeltään: *frequency-smoothing-factor*.

Tämä argumentti on luku, jolla määritetään, kuinka monen nämpnen aikana tasoittava interpolaatio yhdestä taajuudesta toiseen taajuuteen toteutetaan.

Ioscbank~ -objektin kolmas argumentti on nimeltään: *amplitude-smoothing-factor*. Tämä argumentti on luku, jolla määritetään, kuinka monen nämpnen aikana tasoittava interpolaatio yhdestä amplituditasosta toiseen amplituditasoon toteutetaan.

Ylä-äännessarjaan läheisesti kytköksissä olevia käsitteitä ovat mm. osäänessarja, osääneistö, harmoninen osääneistö, luonnonsävelsarja, jousisoittimilla flageoletit, puhaltimilla ylipuhaltaminen sekä kokonaislukukerrannaiset.

PATCH 9: ÄÄNENNOPEUS JA AALLONPITUUS

Patchin toiminta ja objektien funktiot

Patchissä on neljä eri laskuria, joita voi käyttää erilaisissa aallonpituuteen, aikaan, taajuuteen ja äänennopeuteen liittyvissä laskuissa.

Laskurilla 1 voidaan laskea aika, jona ääni etenee tietyn matkan. Laskurilla 2 voidaan laskea matka, jonka ääni kulkee tietyssä ajassa. Laskurit 1 ja 2 pohjautuvat samaan fysikaaliseen kaavaan: $t = s / v$, missä t on aika, s on etäisyys ja v on äänennopeus.

Laskurilla 3 voidaan laskea tietyn taajuuden aallonpituus. Laskurilla 4 voidaan laskea taajuus, kun äänen aallonpituus tunnetaan. Laskurit 3 ja 4 pohjautuvat samaan fysikaaliseen kaavaan: $\lambda = v / f$, missä λ on aallonpituus, v on äänennopeus ja f on taajuus.

Äänennopeutena voidaan käyttää 344m/s, joka on äänennopeus n. 20°C lämpöisessä ilmassa (Moore 1990: 18).

Seuraavassa on muutamia esimerkkikysymyksiä.

Kuinka pitkän matkan ääni kulkee 0,4 sekunnin aikana ilmassa?

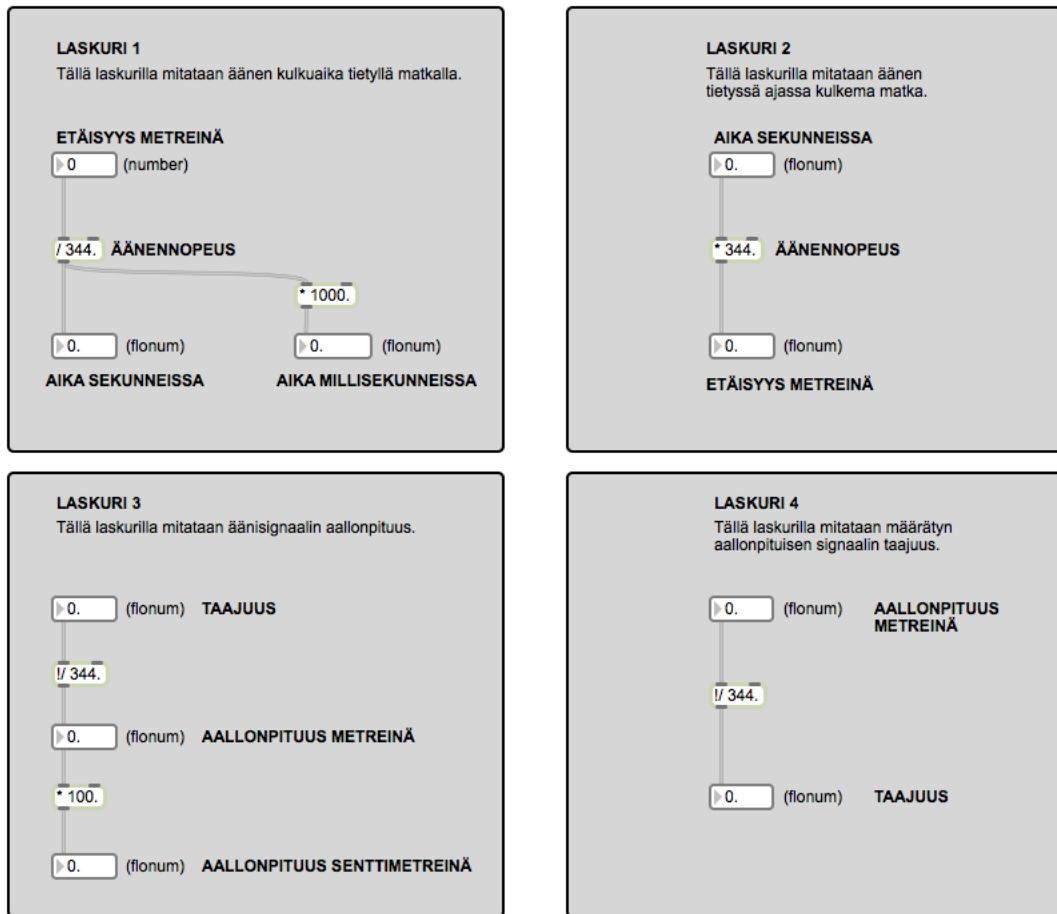
Kuinka kauan äänellä kestää kulkea 688 metrin matka?

Mikä on 10000 Hz:n taajuuden aallonpituus?

Mikä on taajuus kun äänen aallonpituus on 5m?

9. ÄÄNENNOPEUS ja AALLONPITUUS

Patchissä on neljä eri laskuria, joilla voidaan laskea erilaisia aallonpituuteen, aikaan, taajuuteen ja äänennopeuteen liittyviä asioita.



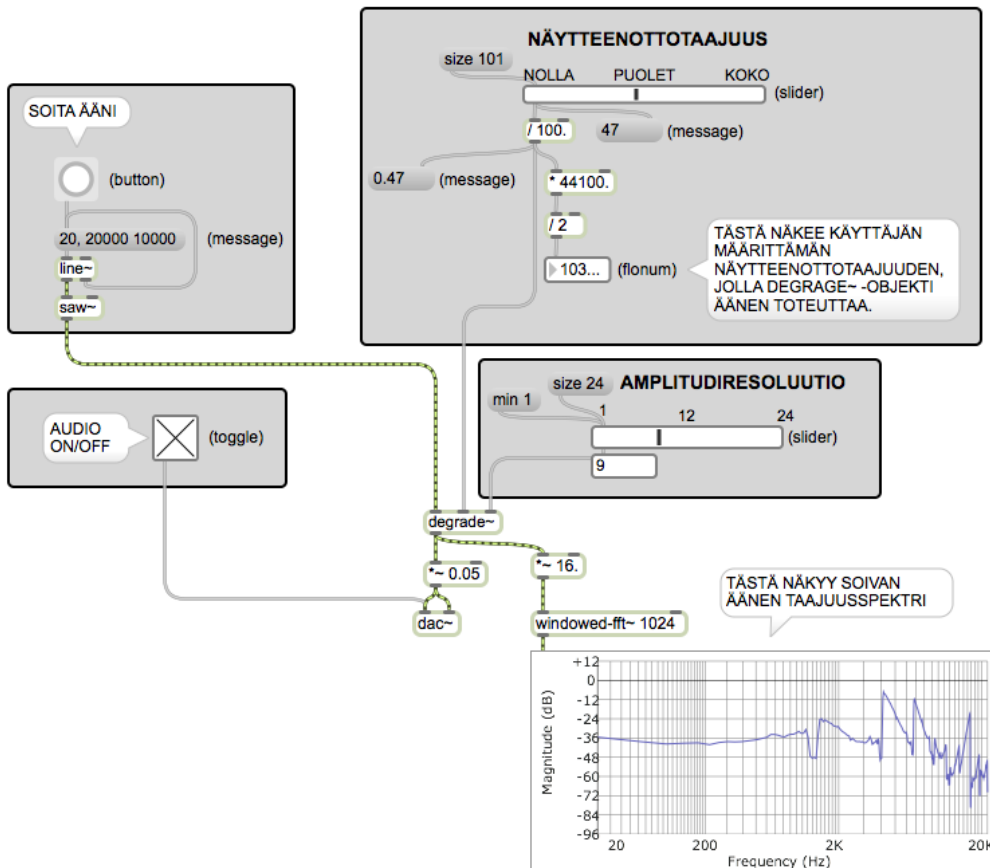
PATCH 10: NÄYTTEENOTTOTAAJUUS JA AMPLITUDIRESOLUUTIO

Patchin toiminta ja objektien funktiot

Patchillä voi kokeilla näytteenottotaajuuden ja amplitudiresoluution muutoksien vaikutusta äänenlaatuun. Patchissä äänilähteenä toimii *saw~* -objekti. *Saw~* -objekti ottaa vastaan lukuarvoja *line~* -objektista. *Button* -objektia klikkaamalla *line~* -objektille välittyy lukulista "20, 20000 10000". *Line~* -objekti tekee näistä luvuista asteittaisen siirtymän 10000 millisekunnin aikana luvusta 20 lukuun 20000. Kun *line~* -objekti pääsee päämääräarvoonsa 20000, *line~* -objektin oikeanpuoleisesta ulostulosta lähtee *bang* -käsky joka on takaisinkytketty liipaisemaan uudestaan saman viestin "20, 20000 10000". *Saw~* -objektin

10. NÄYTTEENOTTOTAAJUUS JA AMPLITUDIRESOLUUTIO

Patchillä voidaan kokeilla näytteenottotaajuuden ja amplitudiresoluution muutoksien vaikutusta äänenlaatuun.



ulostulosta audiosignaali on kytketty *degrade~* -objektiin, jonka funktio on heikentää äänenlaatua. Ylhäällä keskellä olevalla liikusäätimellä, *slider* -objektilla, säädetään näytteenottotaajuus halutuksi. *Slider* -objektin koko määritetään "size 101" -viestillä. "Size 101" -viestin klikkaamisen jälkeen *slider* -objektin ulostulosta tulee lukuja asteikolla 0-100, liikusäädintä säädettyä. *Slider*- ja *degrade* -objektien välissä on jakolaskuobjekti (/100.), jolla skaalataan *slider*-objektin ulostulosta asteikolla 1-100 tulevat kokonaisluvut *degrade*-objektin edellyttämälle asteikolle 0.0-1.0.

Patchin oikeassa reunassa olevilla * 44100. -objektilla ja / 2 -objektilla toteutetaan laskutoimitus, jonka lopputuloksesta nähdään kulloinkin käytössä oleva näytteenottotaajuus. *Degrade~* -objektin oikeanpuoleiseen sisäänmenoon kytketyllä *slider* -objektilla määritetään haluttu bittisyvyys. Tähän slideriin liitettyllä "size 24" -viestillä määritetään sliderin kooksi 24

-askelmaa. “Size 24” -viestin klikkaamisen jälkeen *slider* -objektin ulostulosta tulee lukuja 0:n ja 23:n väliltä, kun slideria säädetään. “Min 1” -viestiä klikattaessa *slider* -objektin minimiarvo muutetaan nolasta yhdeksi, jonka jälkeen slider -objektin ulostulosta tulee lukuja 1:n ja 24:n väliltä säädettyä slideria.

10.4.2 Äänisynteesimetodien perusteita esittelevät patchit

PATCH 11: AALTOTAULUKKOSYNTESI

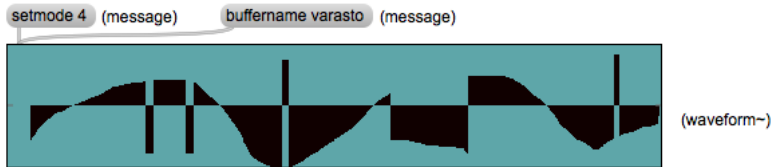
Patchin toiminta ja objektien funktiot

Patch esittelee aaltotaulukkosynteesin (*wavetable synthesis*) toimintaperiaatteen. Patchissä *waveform~* -objektiin piirretty kuvaaja muutetaan kuultavaksi ääneksi. Vaiheessa yksi “setmode 4” -viestiin klikkaamalla *waveform~* -objekti muutetaan piirto-moodiin. “Buffername varasto” -viestiä klikkaamalla *waveform~* -objekti liitetään varasto -nimiseen *buffer~* -objektiin, jonne piirretty aaltomuoto talletetaan. Vaiheessa kaksi kaksoisklikkaamalla *buffer~* -objektia voidaan tarkistaa, onko piirretty aaltomuoto tallentunut *varasto* -nimiseen puskuriin. Vaiheessa kolme “set varasto” -viestiä klikkaamalla, *cycle~* -objekti päivittää omaksi uudeksi aaltotaulukokseen viimeksi piirretyn ja *varasto* -puskuriin talletetun aaltomuodon. Tämän jälkeen tästä uudesta aaltotaulukosta tulee uusi kuunneltava aaltomuoto. Vaiheessa neljä taulukon (*wavetable*) lukuarvot on mahdollista tulostaa Max-ikkunaan klikkaamalla *button*-objektia. Tässä esimerkissä taulukko jakaantuu 512 indeksiin aika-akselilla. Max-ikkuna saadaan näkyviin pikanäppäinkomennolla: cmd-m. *Button*-objektia klikattaessa *uzi*-objektin kolmesta ulostulosta oikeanpuoleisimmasta ulostulosta tulee kokonaisluvut 1-512 kronologisessa järjestyksessä alkaen numerosta yksi. *Peek~* -objekti lukee arvot *varasto* -nimisestä puskurista kunkin indeksin kohdalta ja välittää arvot *pack* -objektin oikeaan sisäänmenoon. Samanaikaisesti indeksinumerot menevät suoraan *pack* -objektin vasempaan sisäänmenoon. *Pack* -objekti pakkaa indeksinumerot ja niitä vastaavat arvot listaksi, jonka jälkeen *pack* -objektin jälkeinen *message*-objekti sijoittaa viestin “indeksi \$1 arvo \$2” \$1- ja \$2- merkkien paikoille *pack* -objektin välittämien listojen arvot. *Print* -objekti tulostaa yksitellen nämä 512 listaa Max-ikkunaan. *Uzi*- ja *peek~* -objektin välissä oleva - -objekti (*huomaa miinusmerkki objektin nimenä!*) vähentää *uzi* -objektista

11. AALTOTAULUKKOSYNTESI (wavetable synthesis)

Patch esittelee aaltotaulukkosynteesin toimintaperiaatteen. Patchissä piirretään kuultava aaltomuoto.

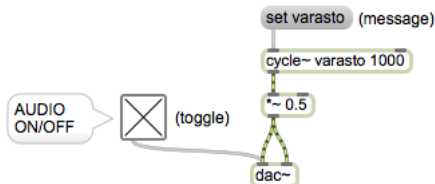
VAIHE 1)
Piirrä aaltomuoto.



VAIHE 2)
Kaksoisklikkaa buffer-objektiin, jotta voidaan tarkistaa, että piirretty aaltomuoto on tallentunut varasto -nimiseen puskuriin.

buffer~ varasto 11.6099

VAIHE 3)
Klikkaa "set varasto", jotta viimeksi piirretty aaltomuoto päivittyy kuunneltavaksi aaltomuodoksi. Laita audio päälle "AUDIO ON", jotta ääni alkaa kuulumaan.



VAIHE 4)
Aaltotaulukon (wavetable) lukuarvot on mahdollista tulostaa Max-ikkunaan klikkaamalla button-objektia. Tässä esimerkissä taulukko jakaantuu 512 indeksiin x-akselilla. Max-ikkuna saadaan näkyviin pikanäppäinkomennolla: cmd - m.



tulevista indeksinumeroista yhden. Tämä vähennyslasku toteutetaan sen takia, että lukuarvot puskurissa on indeksoitu alkamaan indeksistä nolla.

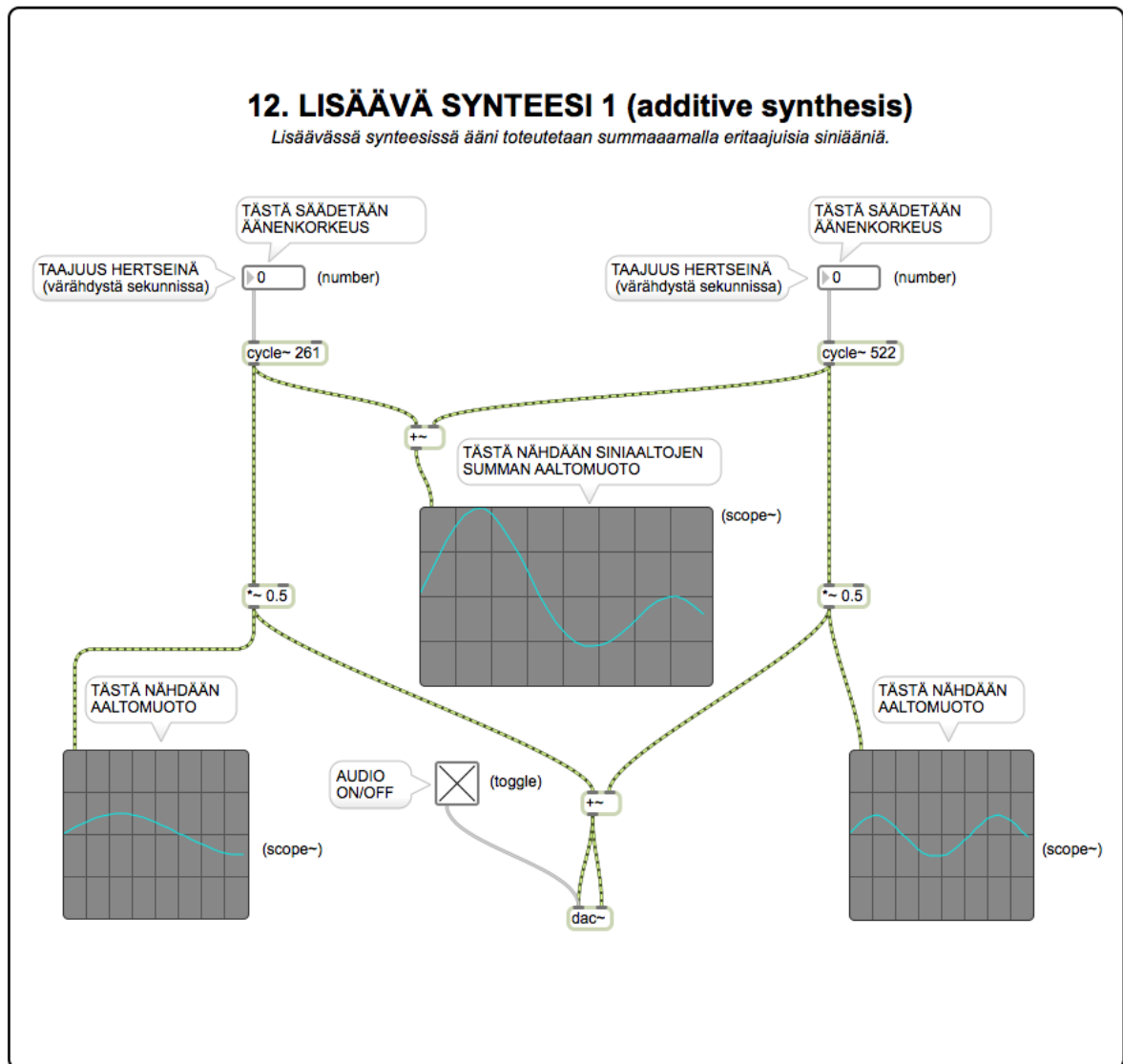
Buffer~ -objektin kaksi argumenttia ovat puskurin nimi ja puskurin koko. Tässä patchissä puskurin nimeksi on määritetty "varasto" ja puskurin kooksi on määritetty: 11.6099 millisekuntia. Laskutoimituksella $(1/44100 \text{ Hz}) * 512$ saadaan selville kuinka monta sekuntia 512 näytettä sisältävä aaltotaulukko kestä. Vastaus on 11.6099 millisekuntia.

PATCH 12: LISÄÄVÄ SYNTEESI 1

Patchin toiminta ja objektien funktiot

Lisäävässä synteesissä (*additive synthesis*) ääni toteutetaan summaamalla siniääniä yhteen. Patchin *cycle~* -objektit generoivat kaksi eri sinitaajuutta. Siniäänien taajuudet määritetään *number* -objekteilla. Sivun vasemmassa ja oikeassa alareunassa olevat *scope~* -objektit mahdollistavat siniaaltojen aaltomuotojen näkemisen graafisina kuvina. Sivun keskellä oleva *scope~* -objekti näyttää graafisena kuvana siniaaltojen summan. *+~* -objektit toteuttavat siniaaltojen summaamisen sekä näkö- että kuulohavaintoa varten.

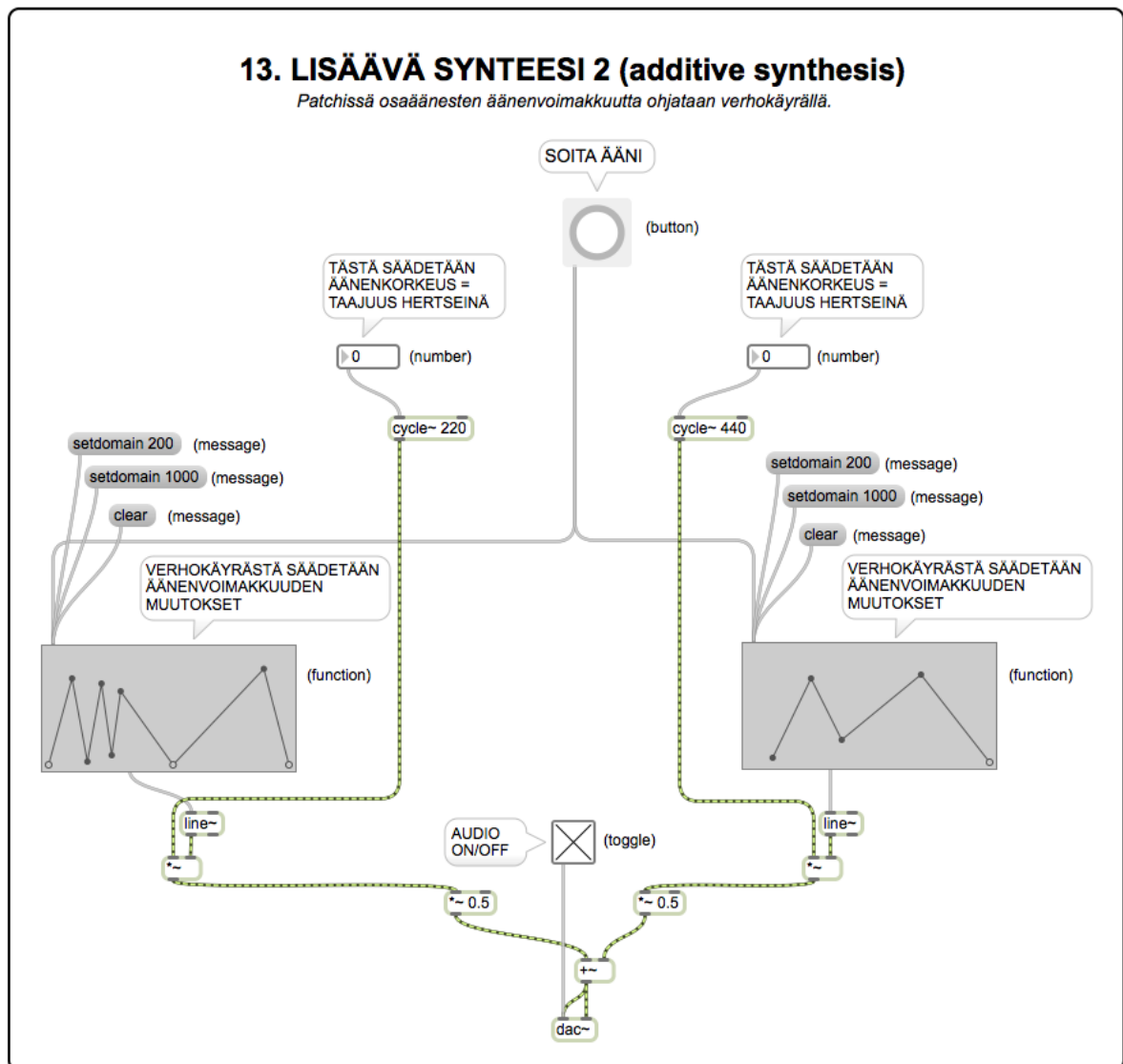
Tämä patch demonstroi lisäävän synteesin yksinkertaisimmillaan. Käyttäjä voi laajentaa patchiä lisäämällä patchiin haluamansa määrän *cycle~* -objekteja.



PATCH 13: LISÄÄVÄ SYNTEESI 2

Patchin toiminta ja objektien funktiot

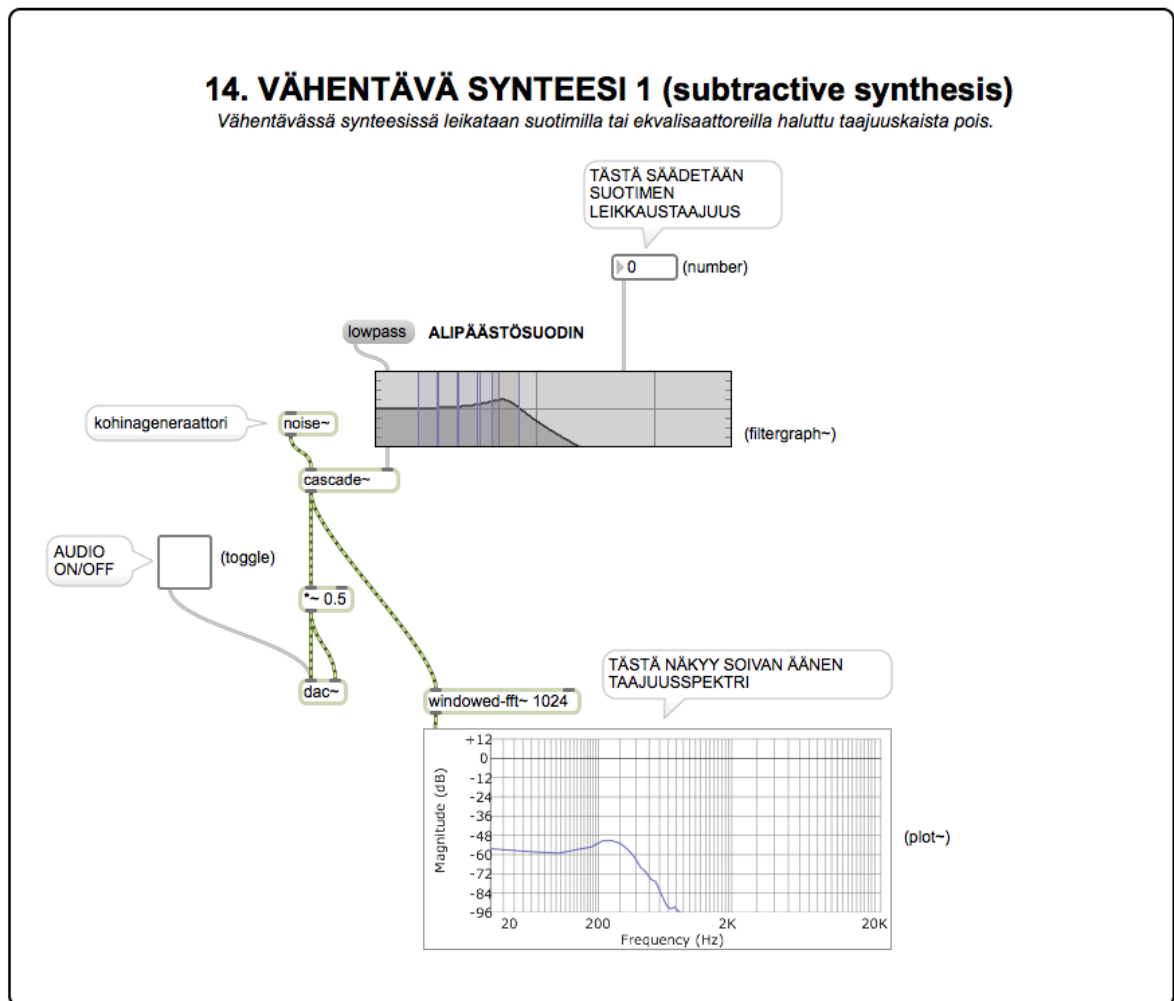
Lisäävä synteesi 2 -patch eroaa lisäävä synteesi 1 -patchistä siten, että lisäävä synteesi 2 -patchiin on lisätty verhokäyrät ohjaamaan osäänesten amplitudin muutoksia. Patchissä nro 6, *Verhokäyrä* -patch, löytyy yksityiskohtainen selvitys, miten verhokäyrät toteutetaan *function*- ja *line~* -objekteilla. “*Setdomain 200*”- ja “*setdomain 1000*” -viesteillä määritetään verhokäyrän kesto millisekunteina. *Clear* -viesteillä *function* -objektit saa tyhjennettyä. Ääni saadaan kuulumaan painamalla yläreunan *button* -objektia.



PATCH 14: VÄHENTÄVÄ SYNTEESI 1

Patchin toiminta ja objektien funktiot

Vähentävän synteesin (*subtractive synthesis*) lähtökohtana on ääni, jonka spektri on rikas. Tästä äänestä leikataan taajuuksia pois suotimilla ja/tai ekvalisaattoreilla. Vähentävän synteesin tärkein työkalu on suodin (Roads 1996: 185). Tässä patchissä äänilähteenä on valkoista kohinaa generoiva *noise~* -objekti. Suotimeksi on valittu alipäästösuodin, jolla valkoisesta kohinasta leikataan pois taajuuksia leikkaustaajuuden yläpuolelta. Kun klikataan *filtergraph~* -objektiin liitettyyn *lowpass* -viestiin, suodin muuttuu alipäästösuotimeksi. *Cascade~* -objekti on patchin varsinainen suodin. *Filtergraph~* -objekti toimii graafisena käyttöliittymänä leikkaustaajuuden määrittämiseen. *Filtergraph~* -objektilla ohjataan *cascade~* -objektia. Ks. myös kohta 10.2 Toistuvat objektirakenteet.

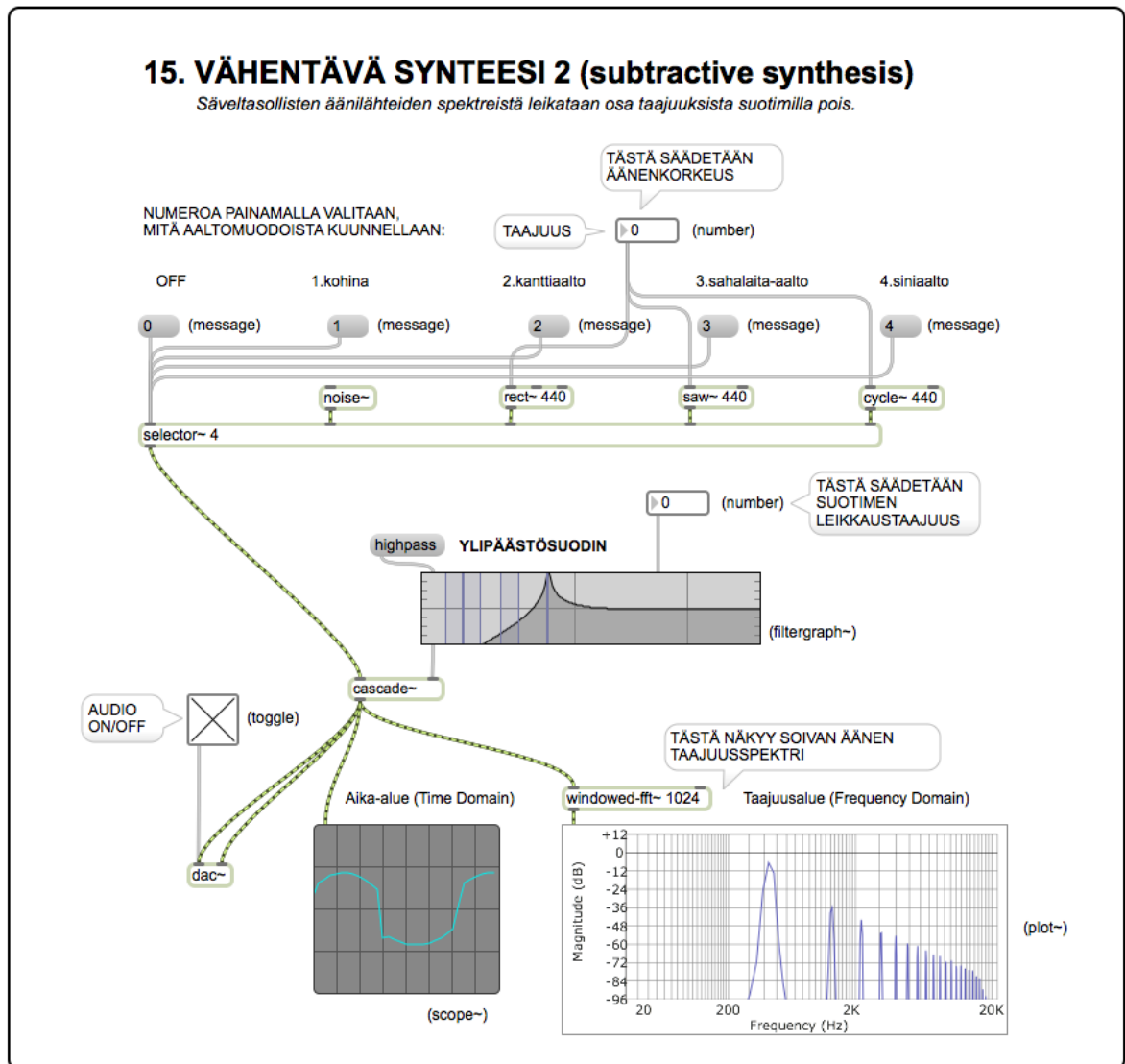


Suotimen leikkaustaajuuden voi määrittää tarttumalla hiirellä *filtergraph~* -objektin kuvaajaan ja liikuttelemalla hiirtä. Toinen vaihtoehto määrittää leikkaustaajuus on syöttää haluttu leikkaustaajuus lukuna *number* -objektiin. Ks. myös patch nro 16 Suotimet.

PATCH 15: VÄHENTÄVÄ SYNTEESI 2

Patchin toiminta ja objektien funktiot

Vähentävä synteesi 2 -patch on rakennettu *vähentävä synteesi 1* -patchin päälle siten, että tähän *vähentävä synteesi 2* -patchiin on lisätty kolme säveltasollista äänilähdettä. Toinen muutos *vähentävä synteesi 1* -patchiin on, että alipäästösuodin on vaihdettu



ylipäästösuotimeen. Kolmas eroavaisuus on, että tähän patchiin on lisätty *scope~* -objekti, joka piirtää graafisen kuvan kulloinkin kuuluvasta äänestä.

Patchin kolme erilaista säveltasollista äänilähdettä ovat: *rect~* -objekti, joka generoi kanttiaaltoa, *saw~* -objekti, joka generoi sahalaita-aaltoa sekä *cycle~* -objekti, joka generoi sini-/kosiniaaltoa. Näistä aaltomuodoista kanttiaallon ja sahalaita-aallon taajuusspektri on laaja, ne siis pitävät sisällään runsaasti ylä-ääneksiä käyttäjän määrittämän perustaajuuden yläpuolelta. Kolmas äänilähdevaihtoehto, siniaalto ei ole spektriltään rikas. Siniaaltogeneraattorilla generoidaan vain yksi perustaajuus. Riippuen tosin käyttäjän laitteista ja muista olosuhteista, muita taajuuksia saatetaan myös kuulla. Nämä ylimääräiset taajuudet voivat johtua siitä, että siniaalto alkaa resonoimaan käytetyn tietokoneen rakenteissa tai siinä (akustisessa) tilassa, jossa käyttäjä työskentelee. *Number* -objektilla määritetään taajuus. *Number* -objekti on kytketty kaikkiin säveltasollista ja jaksollista signaalia generoiviin objekteihin, *rect~*, *saw~* ja *cycle~* -objektit. *Selector~* -objektiin liitetyillä numero -viesteillä 0, 1, 2, 3 ja 4, valitaan äänilähde, jota halutaan kuunnella. *Selector~* -objektista tulee läpi kerrallaan vain yksi siihen liitetyistä audiosignaaleista. Numeroviestillä "0" *selector~* -objekti sulkeutuu, jonka jälkeen siitä ei tule audiosignaalia lainkaan läpi. Kun klikataan *filtergraph~* -objektiin liitettyyn *highpass* -viestiin, suodin muuttuu ylipäästösuotimeksi.

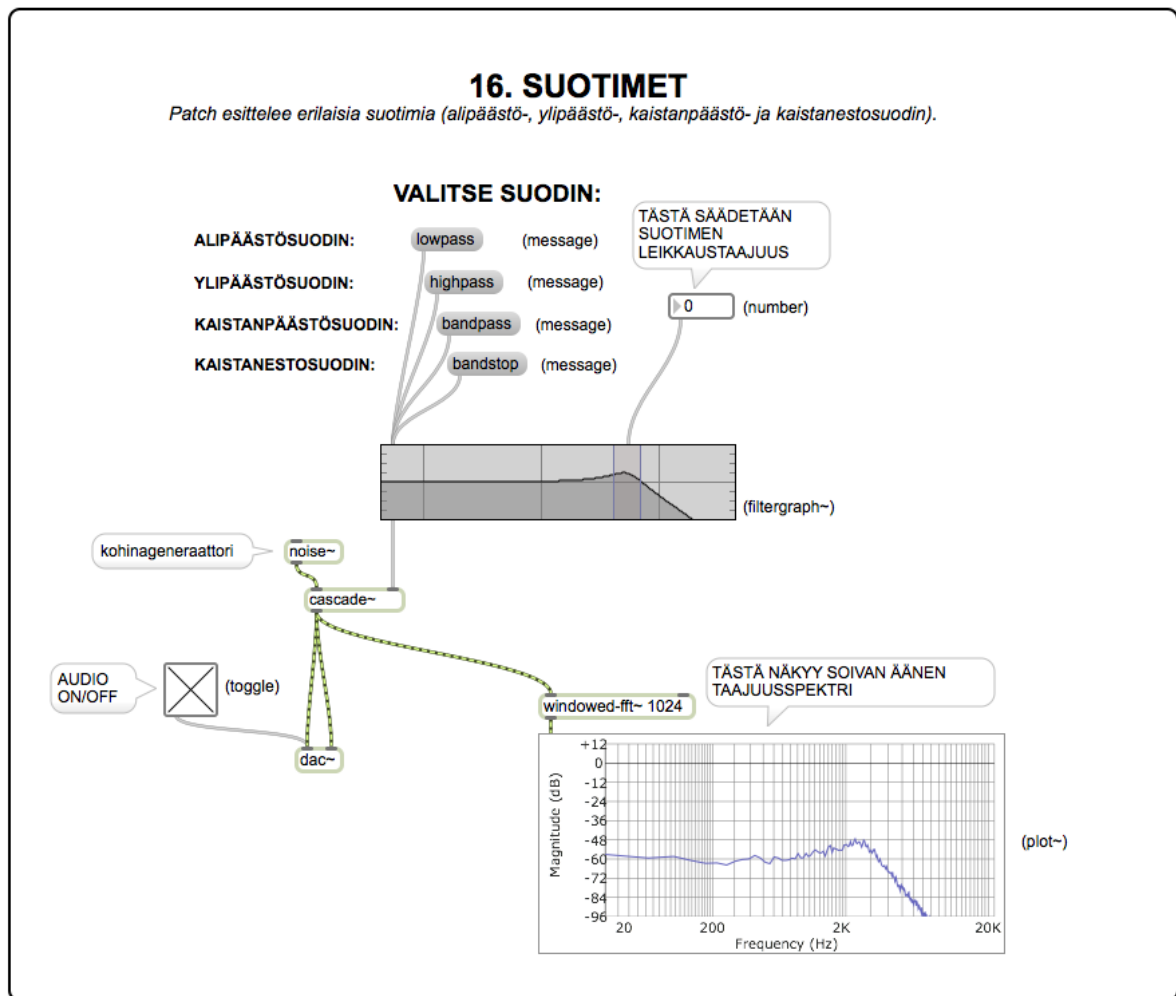
Ensimmäisessä vähentävän synteessin patchissä suodin (*alipäästösuodin*) leikkasi leikkaustaajuuden yläpuoliset taajuudet. Tässä toisessa vähentävän synteessin patchissä suodin (*ylipäästösuodin*) leikkaa leikkaustaajuuden alapuoliset taajuudet.

PATCH 16: SUOTIMET

Patchin toiminta ja objektien funktiot

Patch esittelee erilaisia suotimia (*alipäästö-*, *ylipäästö-*, *kaistanpäästö-* ja *kaistanestosuodin*). *Suotimet* -patch on rakenteeltaan samantyyppinen patch, kuin *vähentävä synteesi 1* -patch on. Tässä patchissä haluttu suodin valitaan klikkaamalla johonkin seuraavista viesteistä: *lowpass* = *alipäästösuodin*, *highpass* = *ylipäästösuodin*, *bandpass* = *kaistanpäästösuodin* tai *bandstop* = *kaistanestosuodin*. *Noise~* -objekti generoi kohinaa, jota valitulla suotimella suodatetaan. *Filtergraph~* -objektilla ohjataan *cascade~* -objektia. Ks. myös kohta 10.2 Toistuvat objektirakenteet.

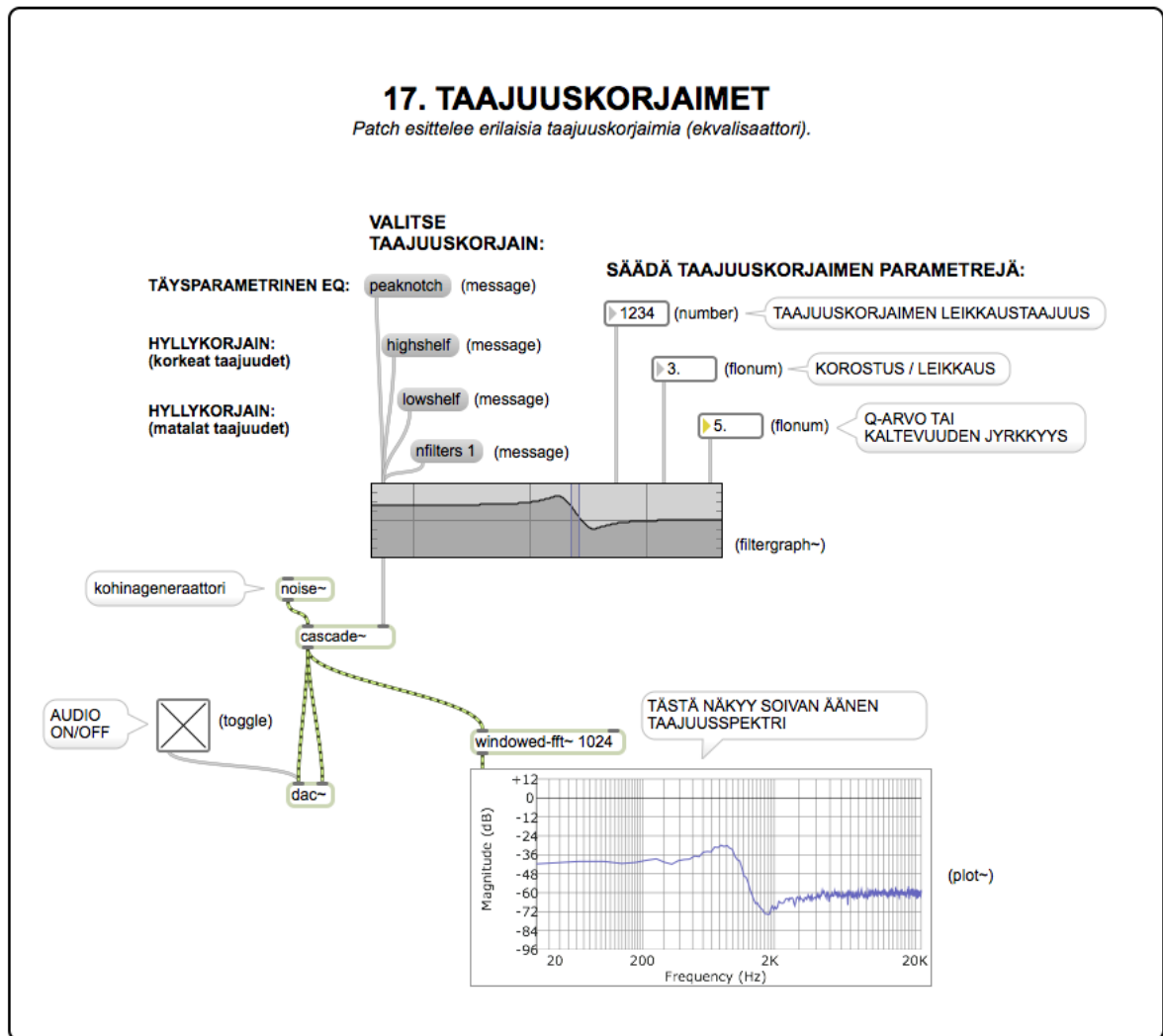
Alipäästösuodin suodattaa signaalista korkeat taajuudet pois, ylipäästösuodin suodattaa signaalista matalat taajuudet pois, kaistanpäästösuodin suodattaa signaalista taajuudet tietyn taajuuskaistan sekä ylä-, että alapuolelta pois ja kaistanestosuodin suodattaa signaalista tietyn rajatun taajuuskaistan taajuudet pois. Rajataajuuden voi määrittää joko syöttämällä haluttu taajuus lukuna *number* -objektiin tai tarttumalla hiirellä *filtergraph~* -objektissa kuvaajaan ja liikuttamalla hiirtä.



PATCH 17: TAAJUUSKORJAIMET / EKVALISAATTORIT

Patchin toiminta ja objektien funktiot

Tällä patchillä voidaan kokeilla erilaisten taajuuskorjaimien käyttöä. Patchin äänilähteenä on *noise~* -objekti, joka generoi valkoista kohinaa. Käyttäjä määrittää taajuusalueet, jotka tästä kohinasta taajuuskorjaimilla joko korostetaan tai vaimennetaan. *Filtergraph~* -objektilla ohjataan *cascade~* -objektia. *Nfilters 1* -viestillä *cascade~* -objekti määritetään sisältämään



vain yhden suotimen. Signaalin varsinainen ekvalisointi tapahtuu siis `cascade~` -objektissa. Ks. myös kohta 10.2 Toistuvat objektitrakenteet.

Tässä patchissä olevat taajuuskorjaimet ovat: täysparametrinen ekvalisaattori sekä korkeiden ja matalien taajuuksien hyllykorjaimet. Patchissä valitaan haluttu ekvalisaattori klikkaamalla johonkin seuraavista viesteistä: `peaknotch` = *täysparametrinen ekvalisaattori*, `highshelf` = *korkeiden taajuuksien hyllykorjain*, `lowshelf` = *matalien taajuuksien hyllykorjain*. Kokopäästösuodinta (*allpass-filter*) varten on oma patchinsä. Feiseri-patch nro 35 on toteutettu kokopäästösuoitimella.

Täysparametrisessä ekvalisaattorissa on kolme parametriä joita voidaan säätää. Nämä parametrit ovat 1) taajuus 2) valitun taajuuskaistan korostuksen tai vaimennuksen määrä ja 3) Q-arvo (*Quality factor*). Taajuus -potentiometrillä määritetään keskitaajuus, jonka ympäriltä joko korostetaan tai vaimennetaan. Q-arvolla määritetään kaistanleveys, alue jolla korostus tai

vaimennus vaikuttaa. Q -arvo määritellään kaistan keskitaajuuden ja sen -3 dB päästökaistan suhteena $Q = f_k/B$, jossa f_k on keskitaajuus ja B on kaistanleveys, -3dB keskitaajuuden tasosta kaistanpäästösuoittimella ja +3dB keskitaajuudesta kaistanestosuoittimella (Honkanen 2014).

Korkeiden taajuuksien hyllykorjaimissa korostus/vaimennus -parametri korostaa tai leikkaa kaikkia taajuuksia valitusta taajuudesta ylöspäin. Matalien taajuuksien hyllykorjaimissa korostus/vaimennus -parametri korostaa tai leikkaa kaikkia taajuuksia valitusta taajuudesta alaspäin. Hyllykorjaimen kolmannella parametrilla säädetään kaltevuutta, kuinka jyrkästi/loivasti korostus/vaimennus -parametri alkaa vaikuttamaan taajuuksiin rajataajuuden kohdalta.

PATCH 18: PIENTAAJUUSOSKILLAATTORI

Patchin toiminta ja objektien funktiot

Pientaajuusoskillaattorilla tarkoitetaan oskillaattoria, jonka taajuus on ihmisen kuuloalueen alapuolella, pienempi kuin 20 Hz. Pientaajuusoskillaattorista puhuttaessa käytetään usein lyhennystä LFO, joka tulee englanninkielisistä sanoista *Low Frequency Oscillator*.

Tässä patchissä LFO:lla toteutetaan vibrato, tremolo ja wah-wah. Vibrato toteutetaan moduloimalla LFO:lla taajuutta, tremolo moduloimalla LFO:lla amplitudia ja wahwah moduloimalla LFO:lla kaistanpäästösuoittimen korostetun alueen keskitaajuutta. Moduloinnilla tarkoitetaan signaalin jonkin parametrin muokkaamista toisen signaalin jollakin parametrilla (Roads 1996: 215).

LFO:lla voidaan moduloida mitä tahansa yksittäistä säädettävää parametriä, joita ovat edeltävien lisäksi mm. vaihe, panorointi, vibraton nopeus, vibraton laajuus ja suoittimen leikkaustaajuus.

Äänilähteenä tässä patchissä on *saw~* -objekti. Oskillaattorin keskitaajuus määritetään *saw~* -objektin yläpuolella olevalla *flonum*-objektilla, joka on desimaaliluku objekti. *Flonum*-objekti on liitetty *saw~* -objektiin *+~* -objektin välityksellä. *Flonum*-objektiin määritetty keskitaajuus välitetään myös **~* -objektille. **~* -objekti, joka on kertolaskuobjekti, toteuttaa kertolaskun keskitaajuuden ja vibrato-rakenteesta tulevan luvun välillä. Tässä **~* -objektissa vibratosta tuleva vaihtelu liitetään keskitaajudeksi määritettyyn lukuun.

Vibraton laajuus valitaan klikkaamalla johonkin patchin vibrato-osiossa olevaan *flonum*-objektiin liitetyistä *message*-viesteistä. Vaihtoehtoina ovat luvut: 0, 0,08, 1 tai 5. Luvulla 0 ei tule lainkaan vibratoa, luvulla 0,08 tulee suppea vibrato, luvulla 1 tulee oktaavin laajuinen vibrato ja luvulla 5 tulee erittäin laaja vibrato.

Vibraton nopeus valitaan klikkaamalla johonkin patchin vibraton-osion vasemmassa reunassa olevaan *flonum*-objektiin liitetyistä *message*-viesteistä. Vaihtoehtoina ovat luvut: 0,25, 1 tai 7. Nämä luvut edustavat vibraton nopeutta. Valittu luku välitetään *flonum*-objektin kautta *cycle~* -objektille. Ennen kuin mitään edellä mainituista luvuista on valittu, *cycle~* -

objekti värähtelee yhden hertsin taajuudella, koska *cycle~* -objektin ensimmäisen argumentin aloitusarvoksi on määritelty luku yksi.

Cycle~ -objekti on kytketty **~* -objektiin, joka toteuttaa kertolaskun *cycle~* -objektilta tulevan signaalin ja *flonum*-objektilta (*vibraton laajuus*) tulevan luvun välillä. *Flonum*-objektilta tuleva luku toimii kertoimena, jolla määritetään *cycle~* -objektilta tulevan signaalin amplitudi. **~* -objektista signaali välitetään *pow~* -objektille. *Pow~*-objektin argumentti on eksponenttifunktion kantaluku, joka tässä patchissä on luku kaksi. *Pow~* -objekti nostaa kantaluvun potenssiin x. X:n arvo on *pow~* -objektin vasemmasta sisäänmenosta menevän signaalin kullakin ajanhetkellä oleva arvo. *Pow~* -objekti toteuttaa vibraaton symmetrisyyden keskitaajuuden suhteen, i.e. sama intervalli keskitaajuuden ylä- ja alapuolella. Ks. myös patch nro 5 Vibrato, jossa vibrato on toteutettu kokonaisluvulla, jolloin vibraaton laajuus on epäsymmetrinen.

Tremolon toteutus

Tremolo kytketään päälle klikkaamalla *flonum*-objektiin liitettyyn *message*-viestiin, jonka luku on 0,5. Viereisellä *message*-viestillä, jonka luku on 0, tremolo kytketään pois päältä. Message-viestien perässä oleva *flonum*-objektin funktio on monitoroida käytössä oleva luku sekä välittää se eteenpäin **~* -objektin oikeaan sisäänmenoon. **~* -objektin vasemman puoleiseen sisäänmenoon on kytketty vibrato-osion *cycle~* -objektin ulostulo. Tremolon nopeus on sama, kuin mikä on valittuna vibraaton nopeudeksi. Ks. myös patch nro 23 Tremolo.

Wah-wahin toteutus

Ks. patch nro 36 wah-wah, jossa on selitetty yksityiskohtaisesti wah-wah -patchin toiminta ja toteutus. Pientaajuusoskillaattori-patchin wah-wah -osio on toteutettu samalla tavalla.

PATCH 19: UNIPOLAARI- JA BIPOLAARISIGNAALI

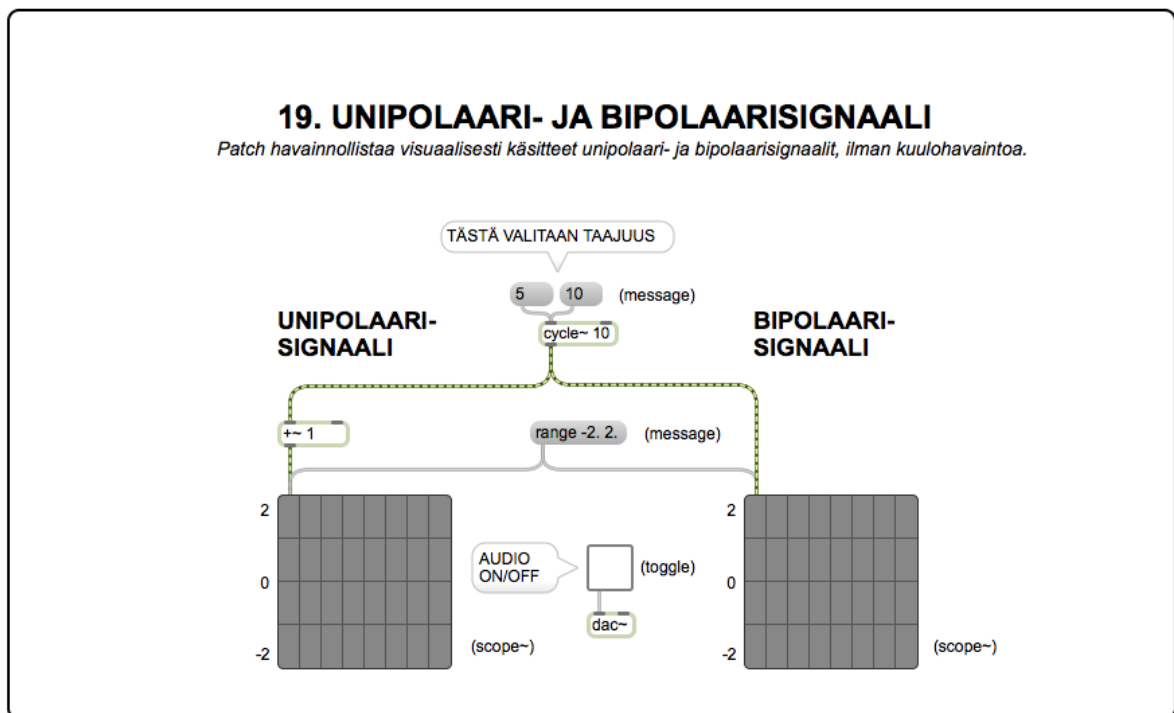
Patchin toiminta ja objektien funktiot

Patch havainnollistaa visuaalisesti käsitteet unipolaari ja bipolaari, piirtämällä uni- ja bipolaareista signaaleista graafiset kuvaajat *scope~* -objekteihin. *Uni-* prepositio tarkoittaa yhtä ja *bi-* prepositio tarkoittaa kahta.

Cycle~ -objektiin liitetyillä numeroviesteillä valitaan joko viiden tai kymmenen hertsin taajuus piirrettäväksi *scope~* -objekteihin. Unipolaarisignaalin hetkellisarvo on aina joko positiivinen tai negatiivinen. Bipolaarisignaalin hetkellisarvo on sekä positiivinen että negatiivinen. Toisin sanoin ilmaistuna: *unipolaarisignaali* -värähtelevää vain toisella puolella x-akselia, kun taas *bipolaarisignaali* värähtelevää x-akselin molemmin puolin.

“*Range -2. 2.*” -viesti määrittää *scope~* -objektien amplitudialueiksi -2:sta 2:een. *+~* -objekti muuntaa *cycle~* -objektista tulevan bipolaarin signaalin unipolaariksi signaaliksi siirtämällä alueella -1 - +1 liikkuvan signaalin liikkumaan alueelle 0 - +2.

Tämä patch ei generoi lainkaan kuultavaa ääntä.



PATCH 20: AMPLITUDIMODULAATIO

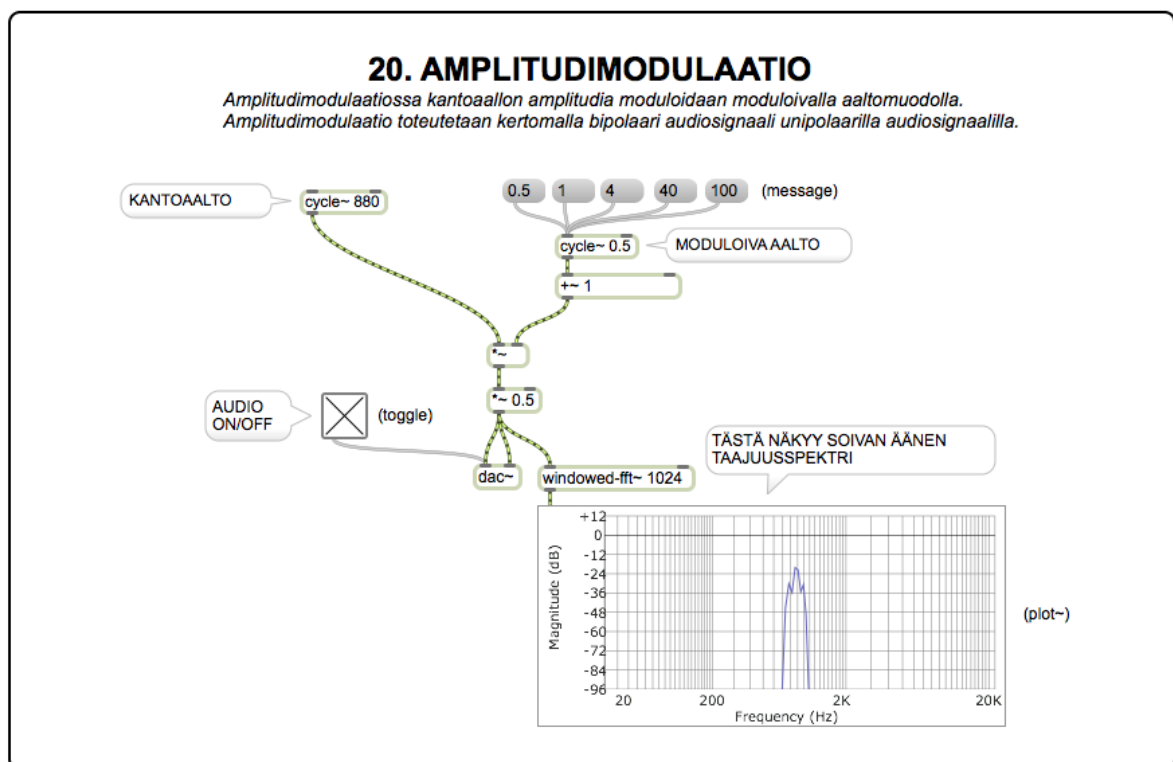
Patchin toiminta ja objektien funktiot

Amplitudimodulaatiosynteesi on äänisynteesimetodi, jossa yhden signaalin amplitudilla kontrolloidaan tai moduloidaan toisen signaalin amplitudia (Amplitude Modulation 2014). Amplitudimodulaatio toteutetaan kertomalla bipolaari audiosignaali unipolaarilla audiosignaalilla. Kun amplitudimodulaatiossa sekä kantoaalto (f_c) että moduloiva aaltomuoto

(f_m) ovat sinimuotoisia, ulostulevan signaalin spektri sisältää kolme taajuuskomponenttia: f_c , (f_c+f_m) ja (f_c-f_m) (Dodge & Jerse 1997: 90).

Amplitudimodulaatio on sukua tremolo-efektille, sillä erolla, että tremolossa moduloiva oskillaattori operoi ihmisen kuuloalueen alapuolisilla taajuuksilla, kun taas amplitudimodulaatiossa moduloiva oskilaattori operoi myös taajuuksilla, jotka ovat ihmisen kuuloalueella.

Patchissä vasemmanpuoleinen *cycle~* -objekti generoi kanta-aallon, joka on bipolaarisignaali. Moduloiva aalto generoidaan oikeanpuoleisella *cycle~* -objektilla, joka on myös bipolaarisignaali, mutta *cycle~* -objektin perässä olevalla *+~* -objektilla signaali muutetaan unipolaariksi signaaliksi. Viesteillä, *0.5*, *1*, *4*, *40* tai *100*, määritetään moduloivan *cycle~* -objektin taajuus klikkaamalla haluttuun *message*-objektiin. Varsinainen amplitudin modulointi tapahtuu **~* -objektissa.



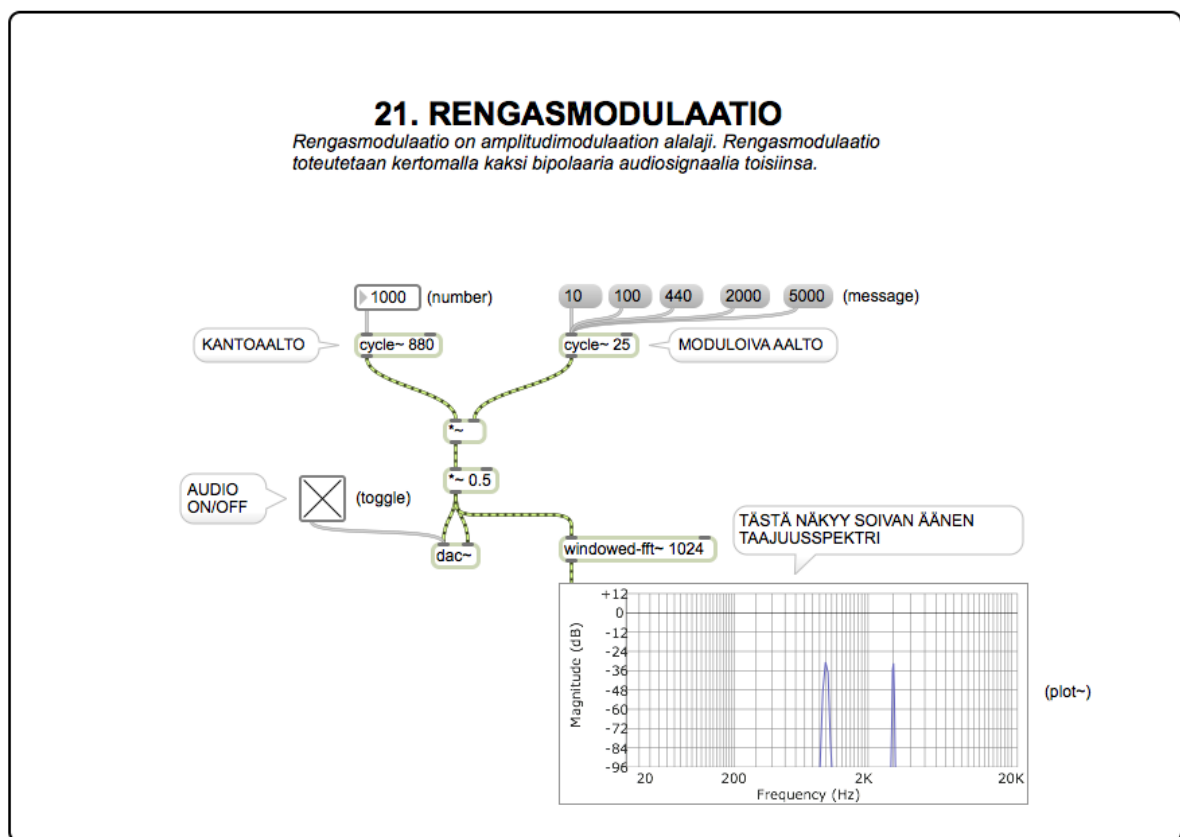
PATCH 21: RENGASMODULAATIO

Patchin toiminta ja objektien funktiot

Rengasmodulaatio (*Ring Modulation*) on amplitudimodulaation alalaji. Ks. myös patchkuvausteksti amplitudimodulaatiosta patch nro 20. Rengasmodulaatio toteutetaan

kertomalla kaksi bipolaaria audiosignaalia keskenään. Kun rengasmodulaatiossa sekä kantaalto (f_c) että moduloiva aaltomuoto (f_m) ovat sinimuotoisia, ulostulevan signaalin spektri sisältää kaksi taajuuskomponenttia: (f_c+f_m) ja (f_c-f_m) (Dodge & Jerse 1985, 1997: 92).

Patchissä on kaksi *cycle~* -objektia, joiden generoimat audiosignaalit kerrotaan **~* -objektilla keskenään. Vasemmanpuoleinen *cycle~* -objekti generoi kantaallon, jossa taajuuden aloitusarvona on 880 hertsiä. Kantaallon taajuus määritetään *cycle* -objektiin liitettyllä *number* -objektilla. Oikeanpuoleinen *cycle~* -objekti generoi moduloivan aaltomuodon, jonka taajuus määritetään viesteillä 10, 100, 200, 2000 tai 5000 klikkaamalla haluttuun *message*-objektiin.



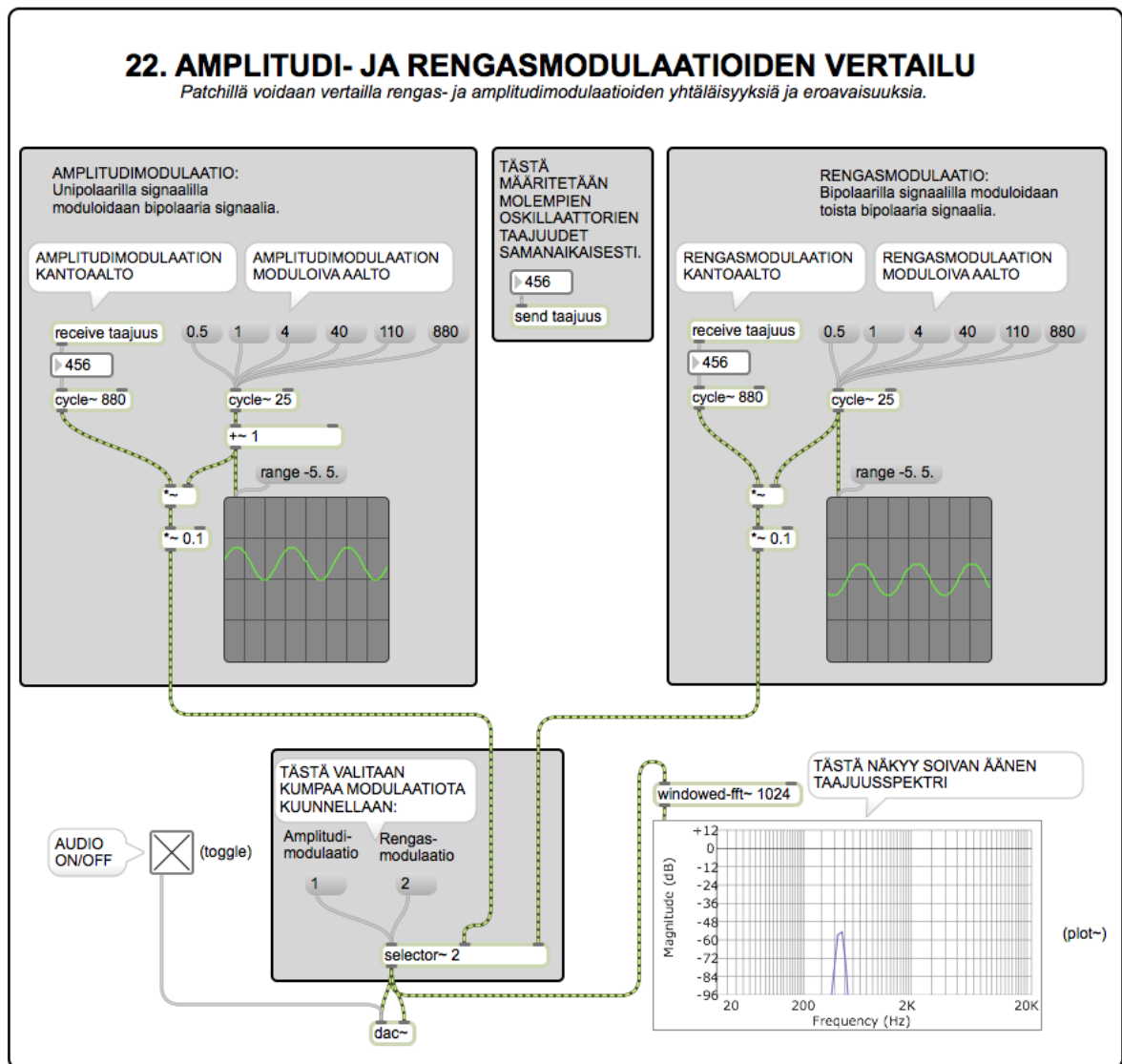
PATCH 22: AMPLITUDI- JA RENGASMODULAATIOIDEN VERTAILU

Patchin toiminta ja objektien funktiot

Tämä patch sisältää sekä amplitudi-, että rengasmodulaatiotoiminnot. Sen avulla voidaan tehdä vertailua modulaatiotyyppien välillä. Keskellä on *number* -objekti, jolla määritetään

käytettyjen kanta-aaltojen taajuudet yhtäaikaan. *Number* -objektista luku välitetään *send* -objektille. *Send*- ja *receive* -objekteilla voidaan toteuttaa datan siirtoa ilman kytkentäjohtoja. *Send*- ja *receive* -objekteilla pitää olla argumentteinaan sama "nimi", jotta niiden välille syntyy langaton yhteys. *Send*-objekti, jonka argumenttina on *taajuus* lähettää vastaanottamansa datan *receive*-objektille, jonka argumenttina on myös *taajuus*. *Send* voidaan myös lyhentää s-kirjaimella ja *receive* voidaan lyhentää r-kirjaimella.

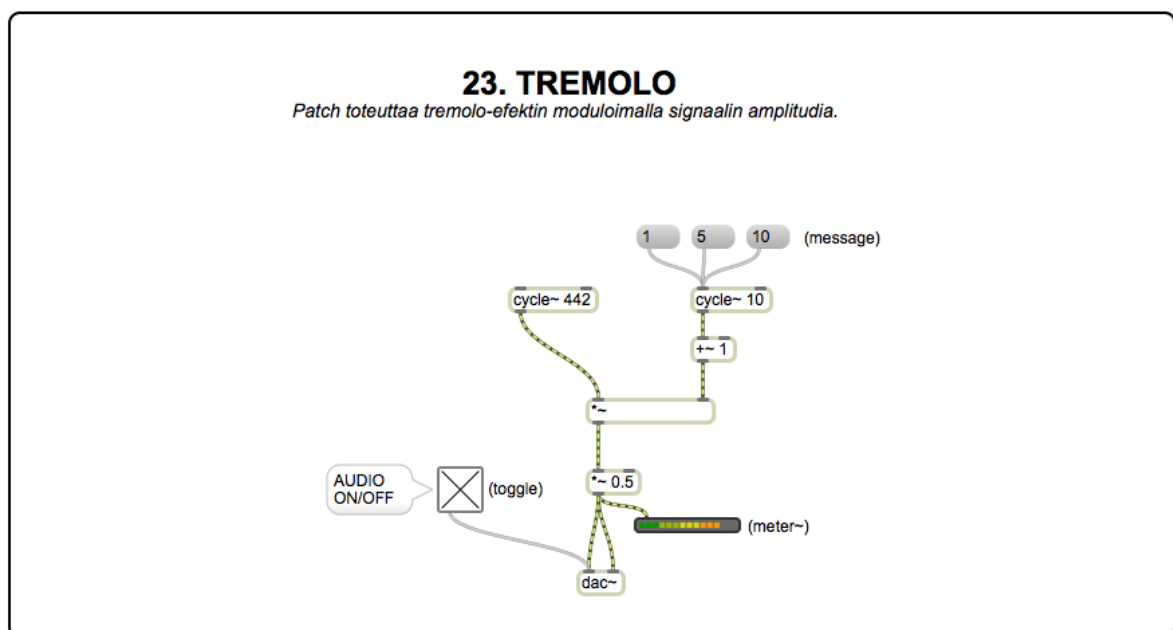
Ks. myös patchkuvaustekstit patch nro 20 *Amplitudimodulaatio* sekä patch nro 21 *Rengasmodulaatio*.



PATCH 23: TREMOLO

Patchin toiminta ja objektien funktiot

Tremolo on amplitudin hidasta variointia (Roads 1996: 215). Tässä patchissä äänilähteeseen toteutetaan tremolo-efekti moduloimalla signaalin amplitudia. Tremolo on amplitudimodulaation alalaji. Tremolo-efektissä moduloivan signaalin taajuus on ihmisen kuuloalueen alapuolella, pienempi kuin 20 Hz. Amplitudimodulaatiossa moduloivan signaalin taajuus voi olla ihmisen kuuloalueella. Ks. myös patchkuvausteksti patchistä nro 20 amplitudimodulaatio.



PATCH 24: FM-SYNTEESI

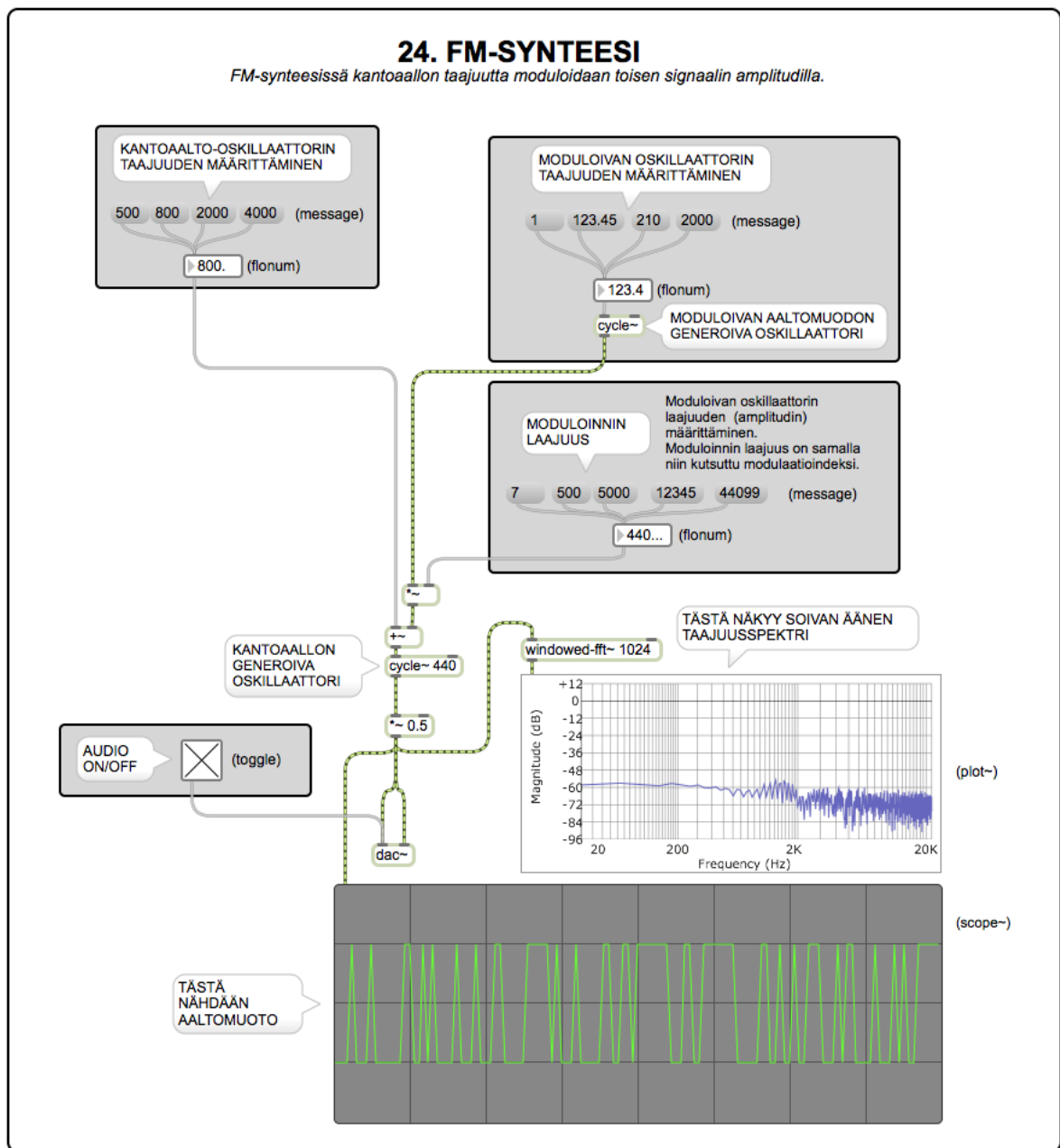
Patchin toiminta ja objektien funktiot

FM-synteesissä moduloivan aaltomuodon amplitudilla moduloidaan kanta-aallon taajuutta (Dodge & Jerse 1997: 115). Kun FM-synteesissä sekä kanta-aalto (f_c) että moduloiva aaltomuoto (f_m) ovat sinimuotoisia, ulostulevan signaalin spektri voidaan ilmaista matemaattisella kaavalla: $f_c \pm k f_m$, missä k on kokonaisluku, joka on suurempi tai yhtäsuuri kuin nolla (Dodge & Jerse 1997: 117). Sivukaistojen keskenäisiä voimakkuuksia määrittelevät 1. asteen Bessel funktiot.

FM-synteesi on äänisynteesimetodi, jolla saadaan toteutettua komplekseja ääniä pienellä määrällä Max-objekteja.

Patchin värähtelevät signaaligeneraattorit ovat kaksi *cycle~* -objektia, joista toinen generoi kantoaallon ja toinen moduloivan signaalin. Kantoaallon generoivan *cycle~* -objektin argumenttina on luku 440, joka on oskilaattorin taajuus. Moduloivan signaalin generoivalla *cycle~* -objektilla ei ole argumenttia, joka määrittäisi sen taajuden.

Kantoaallon taajudeksi valitaan jokin *message*-viesteissä olevista luvuista 500, 800, 2000 tai 4000 klikkaamalla hiirellä valittua lukua. Moduloivan signaalin taajudeksi valitaan



jokin *message*-viesteissä olevista luvuista 1, 123.45, 210 tai 2000 klikkaamalla hiirellä valittua lukua.

Moduloinnin laajuus valitaan klikkaamalla hiirellä johonkin *message*-viesteissä olevista luvuista 7, 500, 5000, 123456 tai 1000000. Näillä lukuarvoilla määritetään moduloivan oskillaattorin amplitudi. Moduloinnin laajuus on samalla niin kutsuttu modulaatioindeksi. Mitä suurempi modulaatioindeksi on, sitä enemmän sivutaajuuksia syntyy ja sitä rikkaampi ulostulevan signaalin spektri on.

*~ -objekti skaalaa *cycle~* -objektilta tulevan sinimuotoisen aaltomuodon amplitudin *moduloinnin laajuus* -osion *flonum* -objektilta tulevalla luvulla. *~ -objektin alapuolella olevalla +~ -objektilla summataan kantoaallontaajuuteen *~ -objektilta tuleva moduloiva aaltomuoto. +~ -objektilta tuleva taajuus välitetään kantoaallon generoivalle *cycle~* -objektille.

PATCH 25: RAESYNTEESI

Patchin toiminta ja objektien funktiot

Tällä patchillä voi kokeilla raesynteesin (*Granular synthesis*) käyttöä. Raesynteesi on äänisynteesimetodi, jossa olemassa olevasta pidemmästä ääninäytteestä (*host-sample*) poimitaan erittäin lyhyitä ääninäytteitä. Erona näytteilyyn, raesynteesissä nämä näytteet ovat pituudeltaan korkeintaan 100 ms (Roads 2001: viii).

Näitä erittäin lyhyitä näytteitä kutsutaan äänirakeiksi (*sonic grains*). Näitä äänirakeita (*mikroään*) "ripotellaan" ajassa siten, että niistä muodostuu eri tiheyksisiä ja eri paksuisia äänikenttiä (*makroään*). Äänikentän tiheydellä tarkoitetaan sitä, kuinka monta ääniraetta liipaistaan tietyssä aikayksikössä. Äänikentän paksuudella tarkoitetaan sitä, kuinka monta eri sävelkorkeuksista ääniraetta soi yhtäaikaan päällekkäin. Vertauskuvana voidaan käyttää sadetta. Siinä missä sade muodostuu lukuisista vesipisaroita, niin raesynteesi muodostuu lukuisista äänirakeista.

Raesynteesi-patch muodostuu kahdesta erillisestä patchistä. Ensimmäinen on tämä raesynteesi-patch kokonaisuudessaan, ilman oikean reunan harmaalla alueella olevaa *rae~* patch osiota. Toinen tehtävä patch on tämä harmaalla alueella oleva *rae~* -patch osio, joka on *poly~* -objektin alipatch. Tämä *rae~* -patch tulee olla talletettuna samaan kansioon

raesynteesi-patchin kanssa, jotta raesynteesi-patchin *poly~* -objekti osaa käynnistettäessä ladata käyttöönsä tämän *rae~* -patchin. Lisäksi, jotta yhteys raesynteesi-patchin ja *rae~* -patchin välillä toimii, *rae~* -patchissä tulee käyttää *thispoly~* -objektia.

Poly~ -objektin ensimmäinen argumentti on ladattavan alipatchin nimi. *Poly~* -objektin toisella argumentilla määritetään, kuinka monta alipatchin kloonia siihen sisällytetään. Raesynteesi-patchissä *poly~* -objekti monistaa *rae~* -patchin kuudeksitoista erilliseksi tapaukseksi. Näitä tapauksia *poly~* -objektissa kutsutaan voiceiksi.

Kun patchillä tehdään raesynteesiä, pitää ensin *buffer~* -objektin *sample*-nimiseen puskuriin ladata *host-sample*. Se tapahtuu siten, että klikataan *poly~* -objektin oikeanpuoleiseen sisäänmenoon kytkettyyn *read*-viestiin. Klikkauksen jälkeen aukeaa valikko, josta valitaan äänitiedosto, jota halutaan käyttää *host-sampler*ina.

Toggle -objektilla käynnistetään *metro*-objekti, jolla määritetään äänirakeiden generointitaajuus. *Metro* -objektiin liitetyllä *number*-objektilla määritetään sen liipaisutiheys millisekunneissa. Sillä siis määritetään äänimassan tiheys.

“Äänirakeen alkukohta” -puhekuplan yhteydessä olevalla *number*-objektilla määritetään äänirakeen alkukohta. Luvulla määritetään äänirakeen toiston aloituskohta millisekunneina *host-sampler*in alusta. Tämä luku tulee talletetuksi *int*-objektiin, josta luku liipaistaan käyttöön seuraavan *metro*-objektista tulevan bang-käskyn myötä.

“Äänirakeen loppukohta” -puhekuplan yhteydessä olevalla *number*-objektilla määritetään äänirakeen loppukohta. Tämä luku välittyy *trigger*-objektille. *Trigger* -objektin funktio on saada *trigger* -objektin jälkeiset tapahtumat toteutumaan oikeassa järjestyksessä. Patchissä olevien *trigger*-objektien sisällä ovat kirjaimet symbolisoivat seuraavia asioita: b = bang, i = integer (kokonaisluku) ja f = float (desimaaliluku). Ensimmäiseksi *trigger*-objekti välittää saamansa kokonaisluvun oikeanreunimmaisesta ulostulosta + -objektin oikeanpuoleiseen kylmään sisäänmenoon. Tämän jälkeen *trigger*-objektin vasemmanpuoleisesta ulostulosta lähtee bang-käsky, joka välittyy + -objektin vasemmanpuoleiseen kuumaan sisäänmenoon. Ks. selvitys kuumista ja kylmistä sisäänmenoista kohdasta 10.3 *Johdatus Max-patchien rakentamiseen*. Kun + -objektin vasempaan kuumaan sisäänmenoon tulee luku, objekti aktivoituu toteuttamaan yhteenlaskun vasempaan sisäänmenoon syötetyn luvun ja oikeaan sisäänmenoon syötetyn luvun välillä ja välittää summan eteenpäin *pack* -objektin keskimmaiseen sisäänmenoon.

“Äänirakeen kestokerroin” -puhekuplan yhteydessä olevalla *flonum*-objektilla määritetään äänirakeen kestokerroin. Tällä desimaaliluvulla voidaan vaikuttaa sävelkorkeuteen. *Flonum* -objektin luku välittyy *trigger* -objektille, joka ensin välittää desimaaliluvun *trigger*-objektin oikeanpuoleisesta ulostulosta *-objektin oikeanpuoleiseen sisäänmenoon. Seuraavaksi *trigger*-objekti lähettää *bang*-käskyn *-objektin vasempaan sisäänmenoon, joka liipaisee kertolaskun juuri *-objektin oikeaan sisäänmenoon saapuneen desimaaliluvun ja *-objektin vasempaan sisäänmenoon syötetyn luvun välille. *-objektin vasempaan sisäänmenoon lukuja saapuu “äänirakeen loppukohta” -puhekuplan yhteydessä olevasta *number*-objektista. *-objekti välittää kertolaskun tulon *pack*-objektin oikeanpuoleisimpaan sisäänmenoon.

Pack -objekti ottaa vastaan kolme kokonaislukua ja tekee niistä listan. Sen sisällä olevat i-kirjaimet tulevat sanasta integer (*kokonaisluku*). *Pack* -objektia seuraava “*note \$1 \$2 \$3*” -viesti ottaa vastaan *pack*in lähettämän kolmen luvun listan ja tallettaa nämä luvut \$1, \$2 ja \$3 merkkien paikalle. Lisäksi “*note \$1 \$2 \$3*” -viesti lisää lukulistan eteen *note* -sanan. *Poly~* -objekti ottaa vastaan tämän listan. *Poly~* -objekti mahdollistaa polyfonisen äänikudoksen toteuttamisen. *Poly~* -objektia varten käyttäjän tulee tehdä tehtävä sivun oikeassa reunassa oleva alipatch ja tallettaa se *rae~* -nimellä. *Poly~* -objektin argumentteina ovat 1) *poly~* objektiin ladattavan alipatchin nimi sekä 2) lukuna se, kuinka monta kloonaa alipatchista *poly~* -objektin on tarkoitus generoida.

Rae~ alipatchin toiminta ja objektien funktiot

Tehdään erillinen *rae~* niminen patch ja talletetaan se samaan kansioon missä raesynteesi -niminen patch sijaitsee. Tällöin raesynteesi-patchin *poly~* -objekti osaa luoda yhteyden *rae~* -patchiin *rae~* -patchissä olevan *thispoly~* -objektin avulla ja käyttää *rae~* -patchiä alipatchinään.

Rae~ -patchin “*in 1*” - ja “*in 2*” -objektit synnyttävät automaattisesti *poly~* -objektiin kaksi sisäänmenoon. Kun *poly~* -objektin vasempaan sisäänmenoon tulee viesti “*note \$1 \$2 \$3*” -objektista, välittää *poly~* -objekti tämän viestin lukuarvot ensimmäiselle vapaalle voicelle, koska viestin alussa oleva *note*-sana ohjaa *poly~* -objektin niin tekemään. *Rae~* -patchin “*in 1*” -objekti ottaa vastaan tämän kolmen luvun listan ja välittää sen “*\$1, \$2 \$3*” -message-objektille. Tämä message-objekti lisää pilkun kahden ensimmäisen luvun väliin. Nyt

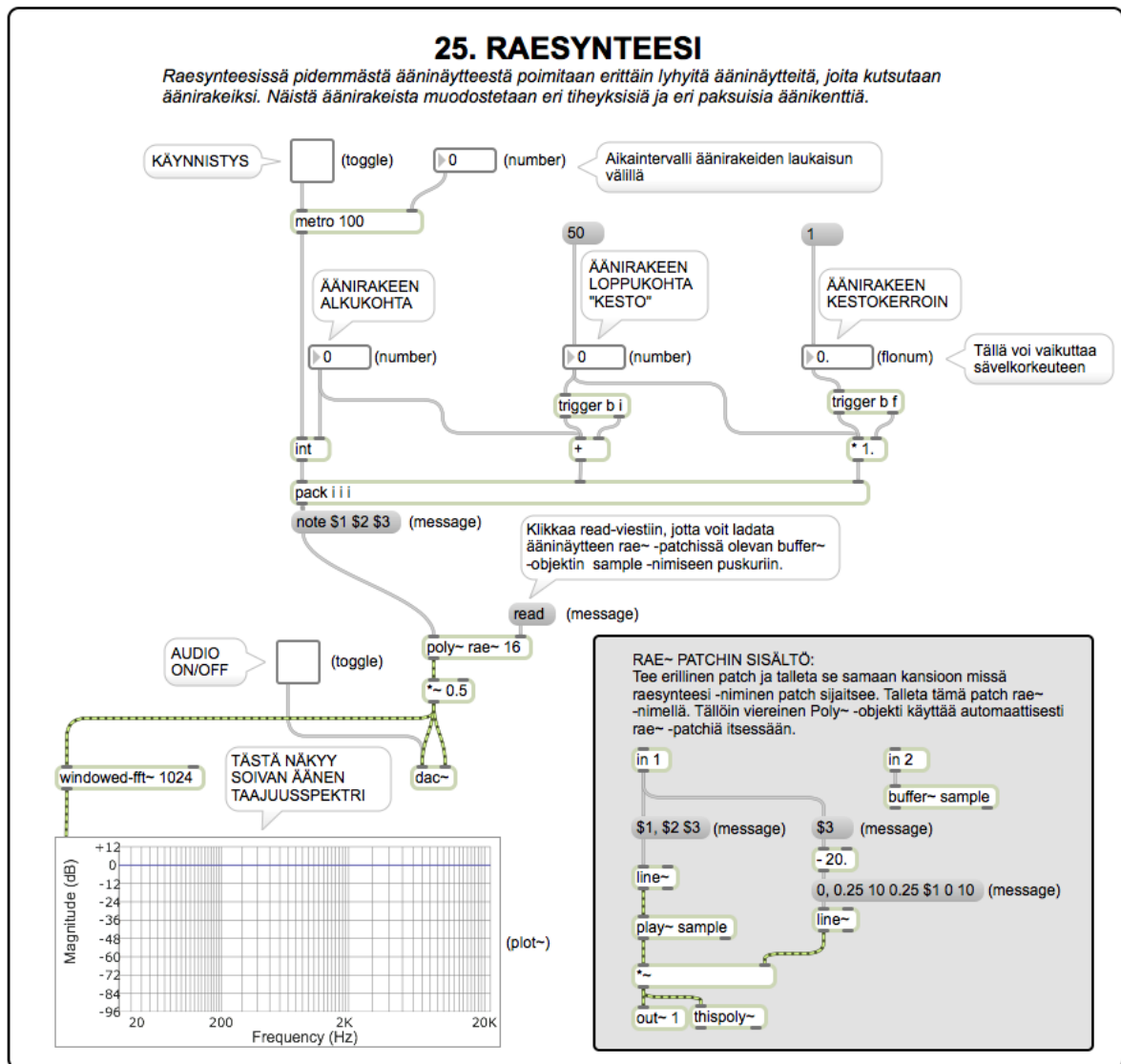
viesti on *line~* -objektin edellyttämässä formaatissa. *Line~* -objekti tulkitsee vastaanottamansa listan seuraavalla tavalla. Ensimmäinen luku on lähtöarvo, toinen luku on päämääräarvo ja kolmas luku on kesto millisekunteina, jonka aikana *line~* -objekti tekee asteittaisen muutoksen lähtöarvosta päämääräarvoon. *Line~* -objekti siis generoi kaikki lähtöarvon ja päämääräarvon väliin tulevat arvot Max-ohjelmasta käsin määritellyllä näytteenottotaajuudella. CD-tasoisessa äänessä näytteenottotaajuus on 44100 näytettä sekunnissa. Kun Max-ohjelmassa näytteenottotaajuudeksi on valittu 44100 Hz, *line~* -objekti generoi lukuja lähtöarvon ja päämääräarvon väliin 44100 lukua sekunnissa. *Play~* objekti lukee sample nimiseen puskuriin talletetun ääninäytteen taulukosta arvoja *line~* -objektista saamiensa lukujen edellyttämässä järjestyksessä ja muodostaa niistä toistettavan aaltomuodon, jonka se välittää *~ -objektille.

“*In 1*” -objektista tuleva toinen johto on kytketty \$3 -message-objektiin. Tämä message-objekti poimii siihen välitetyn kolmen luvun viestistä ainoastaan kolmannen luvun ja välittää sen eteenpäin - -objektille (*huomaa miinusmerkki objektin nimenä!*). Tämä - -objekti tekee vähennyslaskun, jossa se vähentää sisääntulevasta luvusta, joka edustaa kestoa, luvun 20. Syy tämän vähennyslaskun tekemiseen selviää hieman myöhemmin. Tästä vähennyslaskusta syntyvän erotuksen - -objekti välittää “0, 0.25 10 0.25 \$1 0 10” -message-objektille, joka sijoittaa saamansa luvun \$1 -symbolin paikalle. Tällä “0, 0.25 10 0.25 \$1 0 10” -message-viestillä määritetään yksittäisen äänirakeen verhoikäyrä. Tätä “0, 0.25 10 0.25 \$1 0 10” -message-viestiä seuraava *line~* -objekti tulkitsee luvut seuraavalla tavalla: Ensimmäinen nolla ennen pilkkua on aloitusarvo. Äänirakeiden verhoikäyrien aloitusarvoksi halutaan yleensä nolla, jotta äänirae alkaa hiljaisuudesta, ilman mahdollista räsähdystä. Pilkun jälkeiset numerot muodostavat kahden numeron pareja, joista ensimmäinen on seuraava kohdearvo ja jälkimmäinen pari edustaa aikaa millisekunneissa, jonka aikana asteittainen muutos edeltävästä arvosta seuraavaan kohdearvoon tulee toteuttaa. Pilkun jälkeinen 0.25 - luku on siis seuraava kohdearvo, johon *line~* -objekti toteuttaa asteittaisen muutoksen 10 ms:n aikana. Seuraavat luvut “0.25 \$1” *line~* -objekti tulkitsee siten, että siirtymä seuraavaan kohdearvoon 0.25 toteutetaan \$1:ksi välitetyn luvun aikana. Listan viimeiset kaksi lukua aikaansaavat asteittaisen muutoksen \$1 -arvosta nollaan 10 ms:n aikana. Tämä verhoikäyrä siis toteuttaa jokaisen äänirakeen alkuun 10 ms:n mittainen hyvin nopean fade in:in ja loppuun 10 ms:n mittaisen fade out:n. Fade in -alukkeen - ja fade out -lopukkeen avulla estetään

räsähdykset, joita voi syntyä mikäli host-sampleä aletaan toistamaan ilman niitä. Aiempi vähennyslasku, jossa \$3:n välittämästä arvosta vähennettiin luku 20 tehtiin sen takia, että fade in:n toteutukseen tarvitaan 10 ms ja fade out:n toteutukseen tarvitaan toiset 10 ms aikaa. Vähennyslaskun avulla kesto ei pitene 20 ms pidemmäksi.

*~ -objekti ottaa vastaan vasempaan sisäänmenoon *play~* -objektista tulevan audiosignaalin ja oikeaan sisäänmenoon *line~* -objektista tulevat luvut, jotka edustavat verhokäyrää. *~ -objekti toteuttaa *play~* -objektista tulevaan audiosignaaliin tämän verhokäyrän. *~ -objekti välittää audiosignaalin *out~ 1* -objektille.

Out~ 1 -objekti välittää siihen saapuvan signaalin *poly~* -objektille takaisin raesynteesi-patchiin, josta signaali menee *poly~* -objektin ensimmäisestä ulostulosta ulos (*Max thispoly~ - object help file*).



*~ -objektin ulostulosta on kytkentä myös *thispoly~* -objektin sisäänmenoon. *Thispoly~* -objekti luo yhteyden raesynteesi-patchin *poly~* objektin ja *rae~* -patchin välille. *Thispoly~* -objekti lähettää ja ottaa vastaan viestit *poly~* -objektilta. *Thispoly~* -objekti myös kontrolloi *poly~* -objektin voicejen käyttöönottoa, allokointia, sekä voicejen määrittämistä vapaiksi voiceiksi. Voicejen lukumäärä määritetään *poly~* -objektin toisella argumentilla. *Thispoly~* -objekti toimii ainoastaan silloin, kun se on sijoitettu *poly~* -objektin alipatchiin (*Max thispoly~ -object help file*).

“In 2” -objektista tuleva johto on kytketty *buffer~* -objektin sisäänmenoon. *Buffer~* -objektin ensimmäinen argumentti on puskurin nimi. Kun käyttäjä klikkaa raesynteesi-patchissä olevan *poly~* -objektin oikeanpuoleiseen sisäänmenoon kytkettyyn *read* -message-viestiin, aukeaa näytölle valikko, josta määritetään äänitiedosto, joka ladataan *Buffer~* -objektin sample-nimiseen puskuriin.

Tämä patch on yksinkertaistettu esitys raesynteesin toimintaperiaatteista. Tästä johtuen patchissä ei esiinny lainkaan raesynteesille tyypillistä satunnaisvaihtelua eri parametreissa.

PATCH 26: AALTOMUOTOILUSYNTEESI

Patchin toiminta ja objektien funktiot

Patchillä voi kokeilla aaltomuotoilusynteesin (*waveshaping*) käyttöä. *Buffer~* -objekti, jonka puskurin nimi on *wavetable*, lataa muistiinsa *X.triangle.aif* -nimisen audiotiedoston. *X.triangle.aif* -sample tulee Max-ohjelman asennuksen mukana. Ääni soitetaan klikkaamalla yläreunan *button* -objektia. Ks. 10.2 *Toistuvat objektirakenteet*, missä patchin objektien *Button*, *function-*, *line~*, *cycle~*, **~*, *clear-* ja *setdomain*-viestien muodostaman kokonaisuuden toiminta on selitetty yksityiskohtaisesti. Tämä kokonaisuus toteuttaa käyttäjän *function*-objektilla muotoileman verhokäyrän *cycle~* -objektista tulevaan sinimuotoiseen signaaliin. Tässä patchissä tähän rakenteeseen on lisätty message-objektista lähetettävä *range 0 40* -viesti. Viesti skaalaa *function*-objektista tulevat arvot asteikolle 0-40.

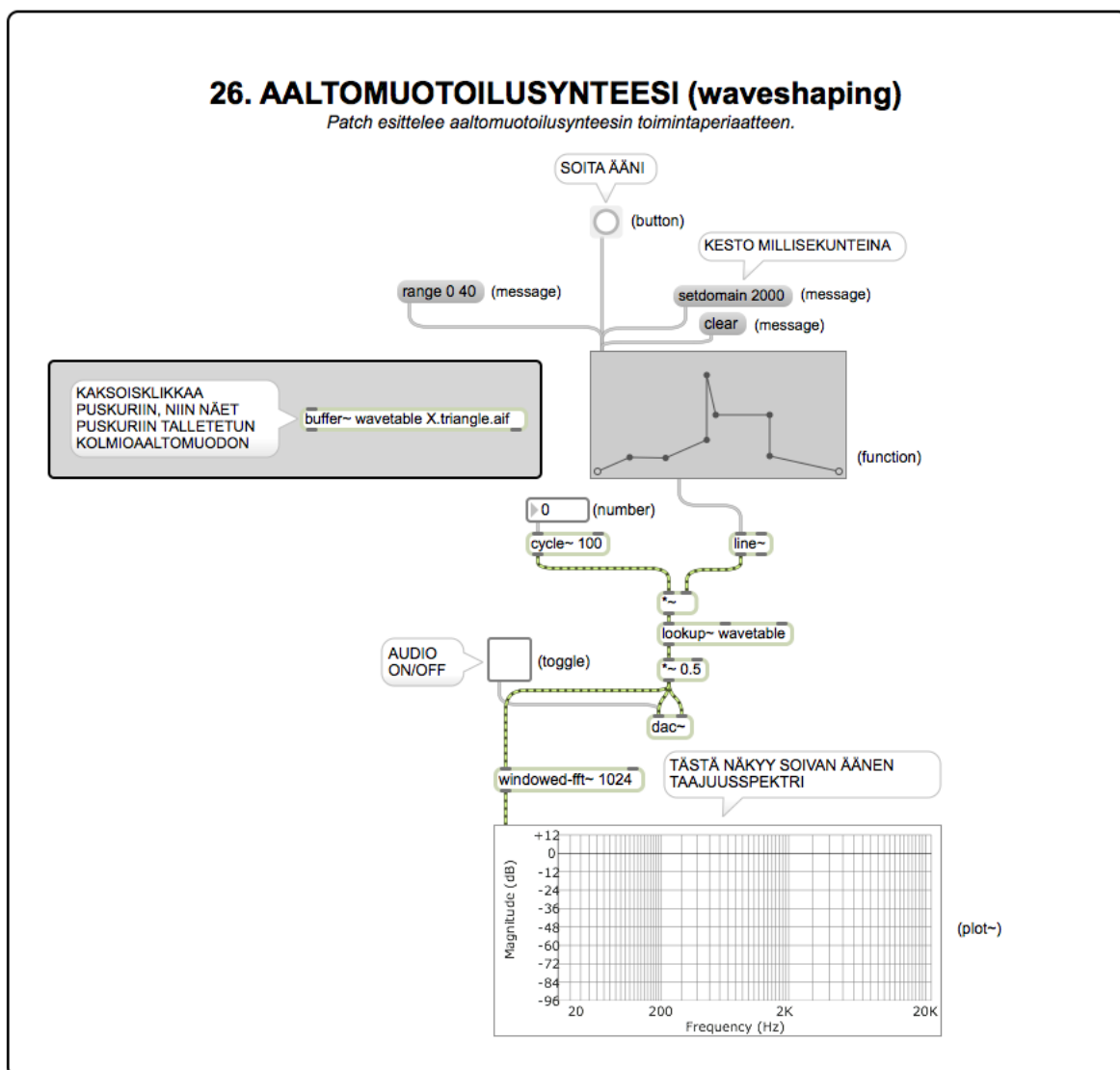
Lookup~ -objekti toteuttaa varsinaisen aaltomuotoilusyntesoinnin. *Lookup~* -objektin avulla puskurin taulukkoon talletetuilla sampleilla tehdään signaalille aaltomuotoilua siten, että signaalin y-arvoja käytetään x-arvoina uuden signaalin y-arvoja määritettäessä.

Sisääntulevat arvot $-1:n$ ja $1:n$ välillä skaalataan taulukon arvoihin nollan (tai määritetyn sample offset-arvon) ja taulukon maksimiarvon välille (*Max lookup~ -object help file*). Edeltävässä mainittu signaali on *cycle~* -objektista tuleva sinimuotoinen signaali, jonka taajuus on 100 Hz.

Lookup~ -objektin ensimmäinen argumentti on puskurin nimi. *Lookup~* -objekti käyttää *buffer~* -objektiin talletettua ääninäytettä.

26. AALTOMUOTOILUSYNTESI (waveshaping)

Patch esittelee aaltomuotoilusynteesin toimintaperiaatteen.



PATCH 27: KAHDEN AALTOTAULUKON RISTIINFEIDAUS

Patchin toiminta ja objektien funktiot

Aaltotaulukoiden ristiinfeidaus (*Wavetable crossfading*) luo ääniä joiden alun ja lopun välillä tapahtuu siirros yhdestä äänestä toiseen (Roads 1996: 160).

“*Kahden aaltotaulukon ristiinfeidaus*” -patchillä havainnollistetaan ääni, joka on toteutettu kahden eri ääneksen, *kanttiaalto* ja *siniaalto*, metamorfoosilla. Patch-kuvan tapauksessa ääni alkaa ensimmäisellä ääneksellä, jonka jälkeen äänen soidessa tapahtuu asteittainen vaihdos ensimmäisestä ääneksestä toiseen äänekseen. Äänen päättyessä kuuluu enää vain jälkimmäinen äänes. Tämä on aaltotaulukkojen ristiinfeidauksen yksinkertaisin toteutus. Kuvaajat *function* -objekteissa edustavat äänesten verhoikäyriä, joilla havainnollistetaan tätä ajatusta graafisesti.

Ääni soitetaan klikkaamalla yläreunan *button* -objektia. Ks. 10.2 *Toistuvat objektirakenteet*, missä patchin objektien *Button*, *function*-, *line*~, *cycle*~, ***~, *clear*- ja *setdomain*-viestien muodostaman kokonaisuuden toiminta on selitetty yksityiskohtaisesti. Tämä kokonaisuus toteuttaa käyttäjän *function*-objekteilla määrittämät verhoikäyrät *rect*~ - ja *cycle*~ -objektien generoimille äänille.

Yksinkertaisimmillaan kuunneltavan äänen alun ja lopun välissä tapahtuu yksi vaihdos yhdestä aaltomuodosta toiseen aaltomuotoon. Luonnollisesti aaltomuotojen vuorottelua voi tapahtua äänen aikana myös useammin kuin vain kerran. Tämä määritetään verhoikäyrillä.

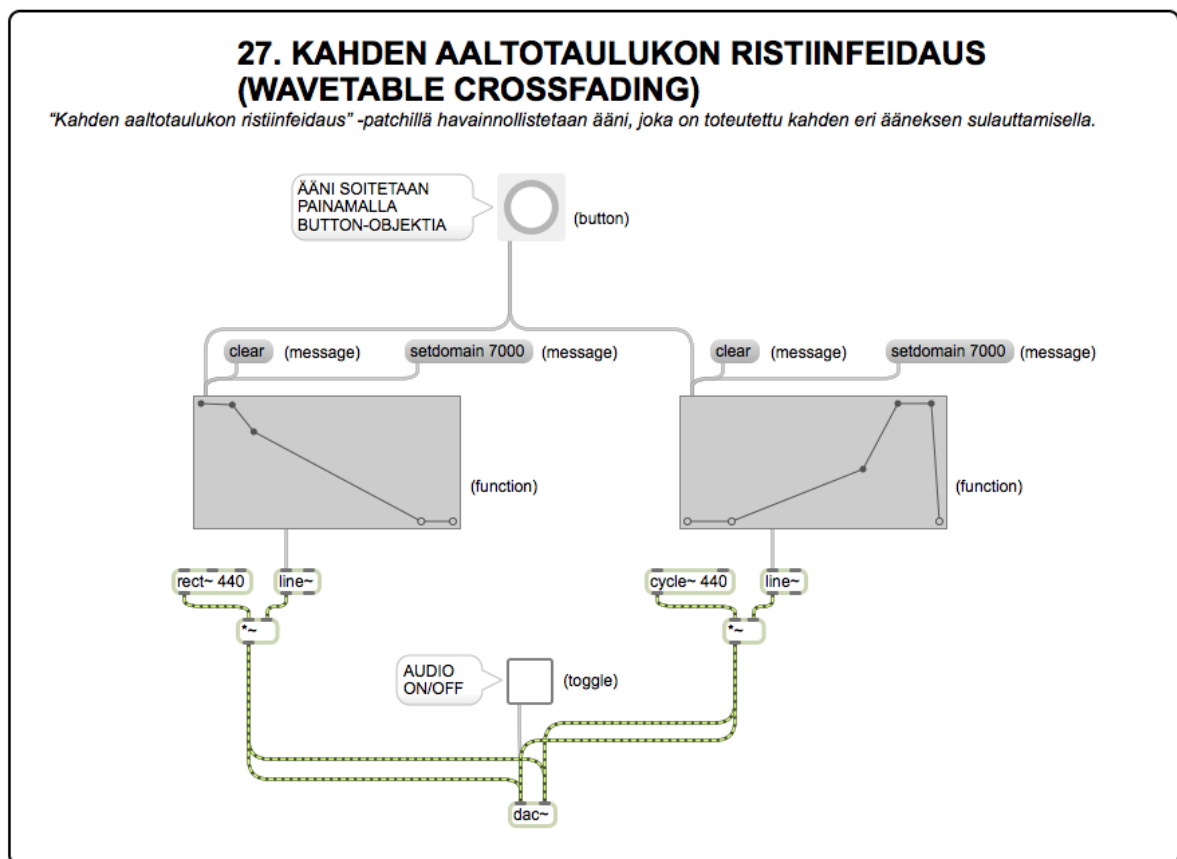
Tehtävä sivun esimerkissä kuultava ääni alkaa 440 hertsin *kanttiaalto*na ja päättyy 440 hertsin *siniaalto*na. Äänen kokonaiskesto on 7000 millisekuntia. Noin 1000 millisekunnin jälkeen *kanttiaalto* alkaa vaimentua ja samanaikaisesti *siniaalto* alkaa voimistua. Noin 1000 millisekuntia ennen äänen loppumista *kanttiaalto* on häipynyt kuulumattomiin, jolloin kuulija kuulee ainoastaan *siniaallon*.

“*Kahden aaltotaulukon ristiinfeidaus*” -patch ja “*lisäävä synteesi 2*” -patch ovat rakenteeltaan lähes samanlaisia. Yksi keskeinen ero niiden välillä liittyy käytettyihin äänilähteisiin. “*Lisäävä synteesi 2*” -patchissä molemmat soivat äänilähteet ovat *cycle*~ -objekteilla toteutettuja *sini*-/*kosini*äänigeneraattoreita, kun taas “*Kahden aaltotaulukon ristiinfeidaus*” -patchissä toinen soivista äänilähteistä on *rect*~ -objektilla toteutettu *kanttiaaltogeneraattori* ja toinen on *cycle*~ -objektilla toteutettu *sini*-/*kosini*äänigeneraattori.

Toinen keskeinen ero patchien välillä on, että *“lisäävä synteesi 2”* -patchissä siniaaltojen taajuudet ovat erit, kun taas *“Kahden aaltotaulukon ristiinfeidaus”* -patchissä sini- ja kantiaallon taajuus on sama (440 Hz).

“Lisäävä synteesi 2” -patchin perusajatus on rakentaa yksi kompleksimpi ääni summaamalla kaksi tai useampia siniääneksiä päällekkäin. Klassisessa lisäävässä synteesissä käytetään vain siniääniä, kun taas *wavestacking*-synteesissä, voivat kaikki pinottavat aaltomuodot olla komplekseja ääniä, kuten esimerkiksi sämplejä (Roads 1996: 160).

“Kahden aaltotaulukon ristiinfeidaus” -patch taas perustuu aaltotaulukoiden ristiinfeidaus -tekniikalle, jossa on käytössä yhtäaikaan vähintään kaksi eri aaltotaulukkoa. Aaltotaulukkojen ristiinfeidaus on yhdistävä ydinasia eri nimisille äänisynteesimetoille kuten compound synthesis (Roads), vector synthesis (Sequential Circuits, Korg, ja Yamaha) ja L/A tai Linear Arithmetic synthesis (Roland) (Roads 1996: 159). Näissä äänisynteesimetoissa yksi ääni toteutetaan kahdesta tai useammasta sointiväriiltään



erilaisesta ääneksestä siten, että kulloinkin soivien äänesten välillä tapahtuu vuorottelua äänen kokonaiskeston aikana.

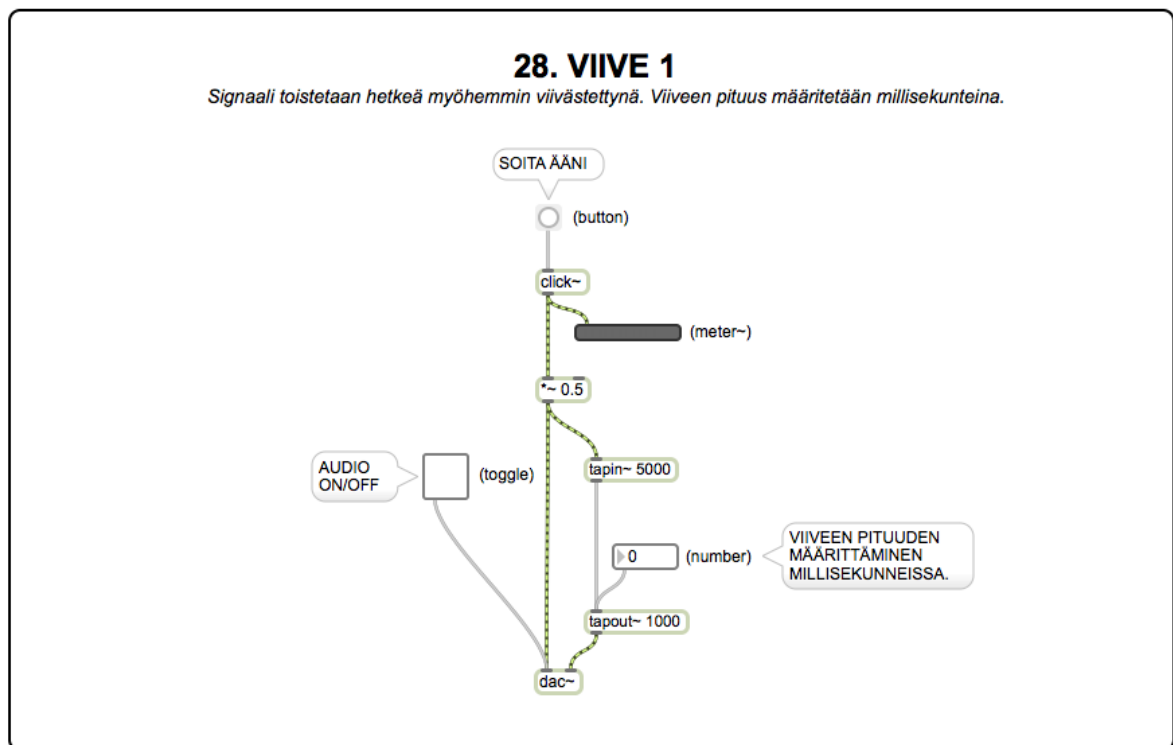
10.4.3 Efekti- / prosessointilaitteiden toimintaperiaatteita esittelevät patchit

PATCH 28: VIIVE 1

Patchin toiminta ja objektien funktiot

Tässä patchissä audiosignaali toistetaan viivästettynä. Viiveen pituus määritetään millisekunteina.

Button~ -objektilla soitetään ääni. Syntyvä ääni on “klik” -ääni, joka generoidaan *click~* -objektilla. *Meter~* -objekti toimii monitorina, jolla voidaan tarkistaa, että *click~* -objektista lähtee audiosignaali ulos. *Meter~* -objekti näyttää *click~* -objektista lähtevän audiosignaalin amplitudin. *Tapin~* - ja *tapout* -objektit muodostavat kokonaisuuden jolla viive toteutetaan. *Dac~* -objektin vasempaan sisäänmenoon audiosignaali tulee viivästämättömänä, oikeaan sisäänmenoon viivästettynä. *Tapin~* -objektin argumentti on pisin mahdollinen

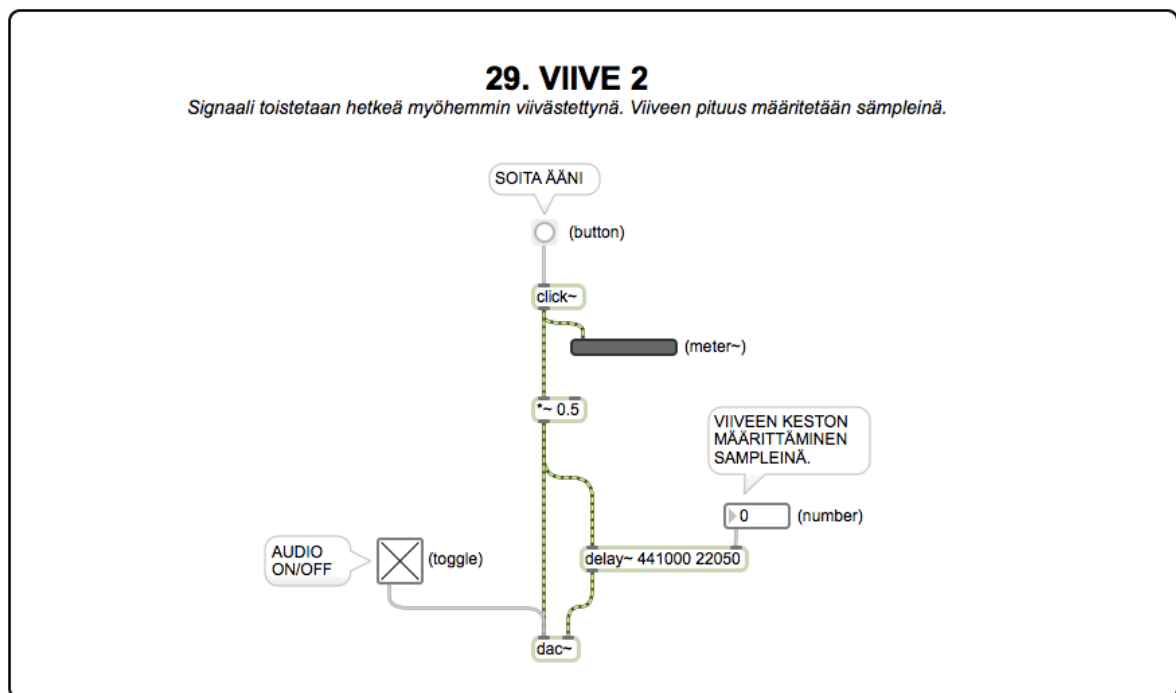


käytettävissä oleva viive. *Tapout~* -objektin argumentti on käytössä olevan viiveen pituus, jonka pituus määritetään *tapout~* -objektiin liitettyllä *number*-objektilla.

PATCH 29: VIIVE 2

Patchin toiminta ja objektien funktiot

Tässäkin patchissä audiosignaali toistetaan viivästettynä. Viiveen pituus määritetään sampleinä. Tämän patchin objektirakenne poikkeaa *Viive 1* -patchin rakenteesta siten, että viivästäminen toteutetaan *tapin~* - ja *tapout~* -objektien asemasta *delay~* -objektilla. *Delay~* -objektin ensimmäinen argumentti on pisin mahdollinen käytettävissä oleva viive ja toinen argumentti on käyttäjän määrittämä viive. Käytössä olevan viiveen pituus määritetään *delay~* -objektin oikeanpuoleiseen sisäänmenoon liitettyllä *number*-objektilla.

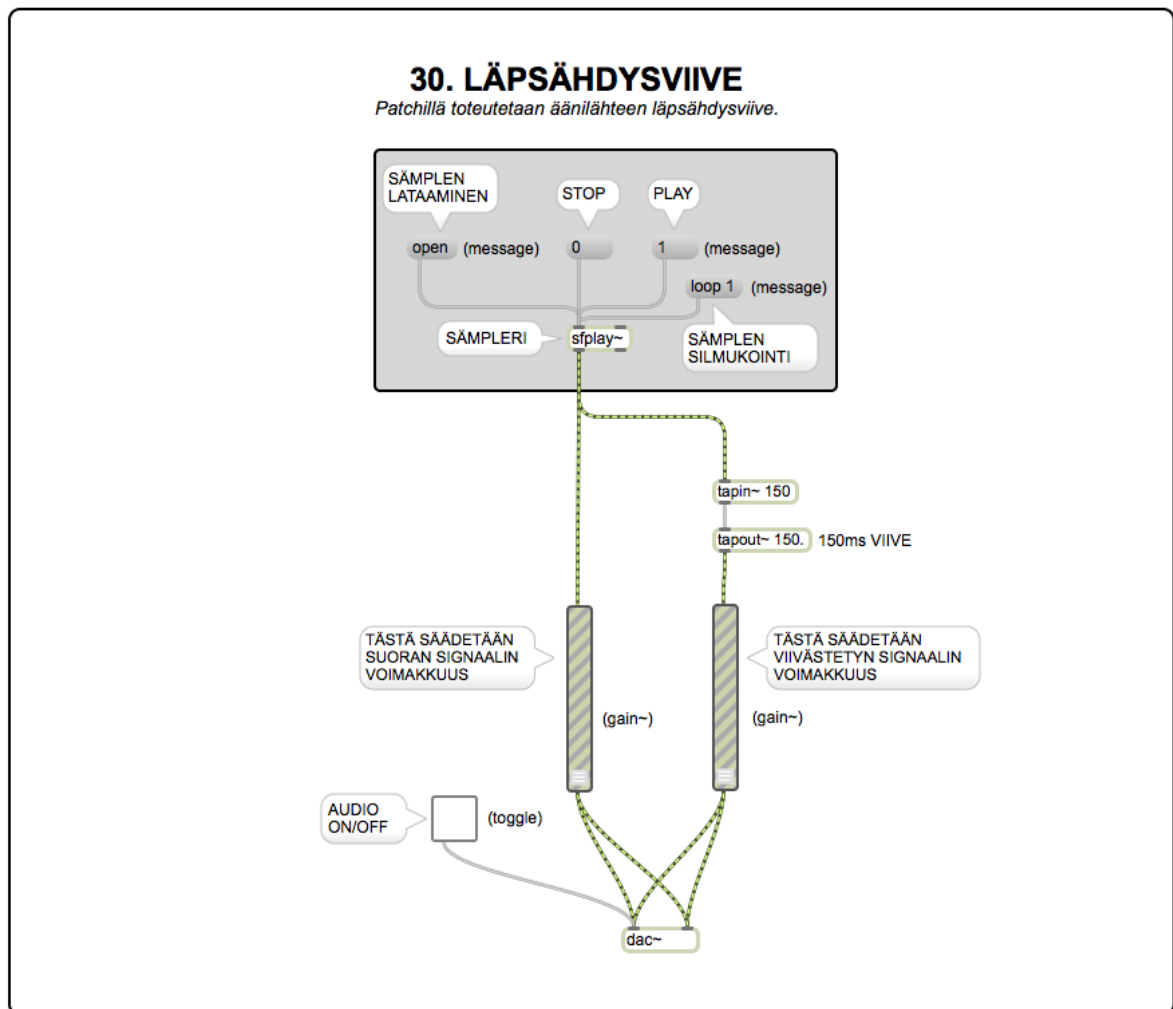


PATCH 30: LÄPSÄHDYSVIIVE

Patchin toiminta ja objektien funktiot

Patchillä toteutetaan äänilähteeseen läpsähdysviive (*slapbackdelay*). Jotta viive kuullostaa läpsähdykseltä, tulee viiveen pituus olla 60-180 ms (Smith 2012).

Patchin äänilähde on *sfplay~* -objekti, joka on sämpleri. *Sfplay~* -objektin ulostulo on kytketty *gain~* -objektiin sekä *tapin~* -objektiin. Vasemmanpuoleisella *gain~* -objektilla säädetään kuivan signaalin voimakkuus. *Tapin~* -objektin argumentilla määritetään viiveen maksimipituus. *Tapin~* -objekti on kytketty *tapout~* -objektiin, jonka argumentilla määritetään käytössä olevan viiveen pituus. *Tapout~* -objekti on kytketty oikeanpuoleiseen *gain~* -objektiin, jolla säädetään viivästetyn signaalin voimakkuus.



PATCH 31: VIIVEILLÄ TOTEUTETTU PIENI TILA

Patchin toiminta ja objektien funktiot

Tässä patchissä keskelle panoroidun monoäänilähteen ympärille synnytetään kahdella eripituisella lyhyellä viiveellä ääni-ilmiö, jonka tarkoitus on synnyttää kuuloaistimus, joka koetaan pienenä tilana monoäänilähteen ympärillä. Kun viive on alle 40 ms ei ihminen kuule ilmiötä kahtena erillisenä äänenä, vaan yhtenä äänenä. Tätä kutsutaan Haas-efektiksi tai presedenssi -efektiksi (Ward 2011).

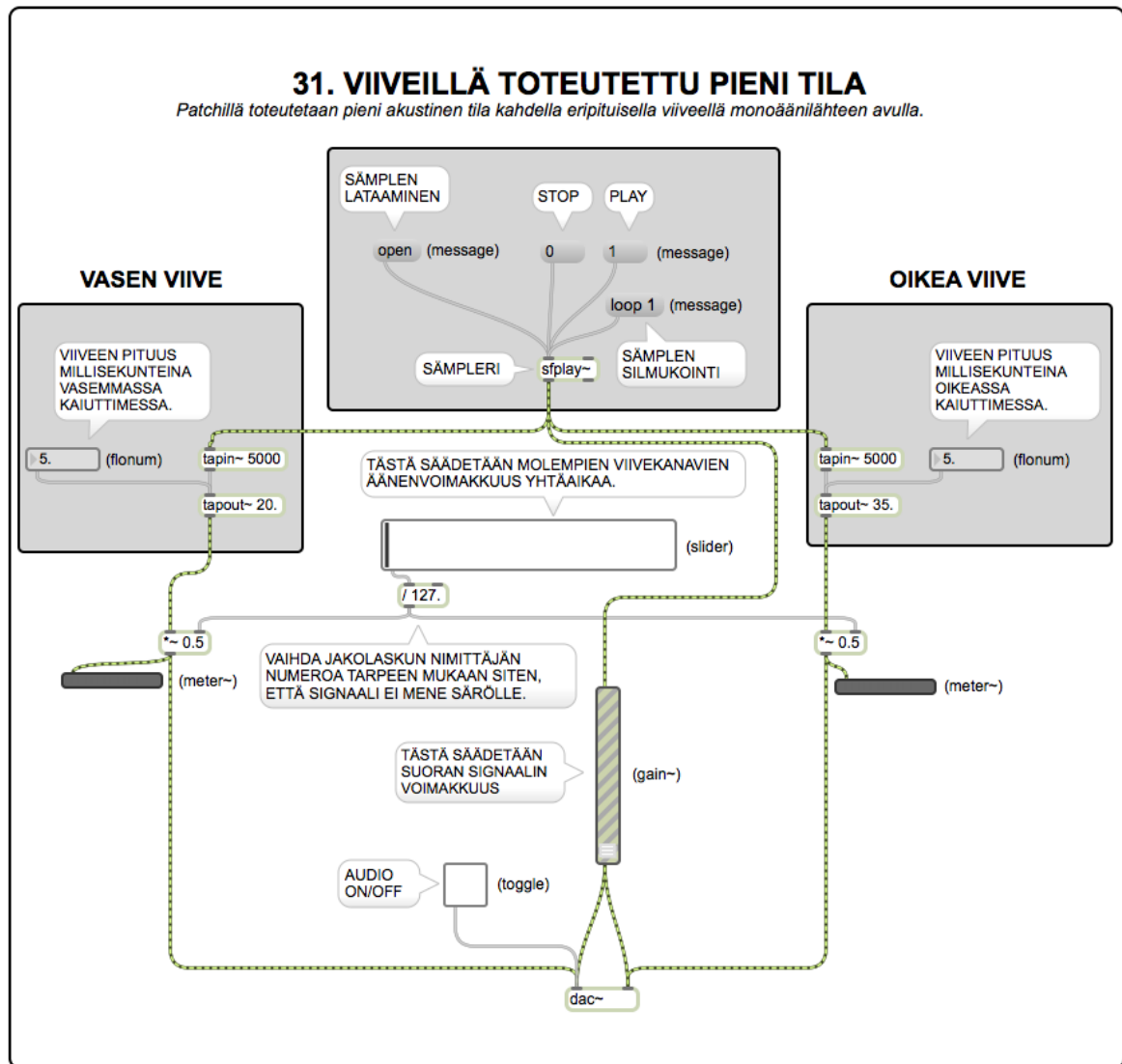
Tilailluusioita voidaan rakentaa mm. siten, että panoroidaan monoäänilähde keskelle, johon summataan yksi alle 40 millisekunnin viive vasemmalle panoroituna ja toinen alle 40 millisekunnin viive oikealle panoroituna. Keskeistä tilailluusion syntymiselle on se, että vasemmanpuoleisen ja oikeanpuoleisen viiveen pitää olla eripituiset. Jos vasemmanpuoleinen viive on kestoaltaan sama kuin oikeanpuoleinen viive, tällöin aistihavaintona on kokemus viiveiden panoroitumisesta keskelle, jolloin leveyssuunnassa tilailluusio jää syntymättä.

Patchin keskeiset objektit ovat samat kuin edeltävässä läpsähdysviive -patchissä. Patchit eroavat siten, että läpsähdysviive -patchissä on kyse monoviiveestä, kun taas tässä “viiveillä toteutettu pieni tila” -patchissä on kyse stereoviiveestä. Toinen keskeinen ero näiden kahden patchin välillä on se, että tässä “viiveillä toteutettu pieni tila” -patchissä viiveiden pituuksien pitää olla riittävän lyhyitä, korkeintaan 40 millisekuntia, kun taas läpsähdysviive -patchissä viiveen pituus tulee olla vähintään 40 millisekuntia.

Soiva äänilähde tässä patchissä on *sfplay~* -objekti, joka on sämpleri. *Sfplay~* -objektista lähtee kolme kytkentäjohtoa, joista yksi menee vasempaan viiveeseen, toinen oikeaan viiveeseen ja kolmas välittää viivästämättömän signaalin suoraan *gain~* -objektiin, jolla säädetään suoran signaalin äänenvoimakkuus. Jotta *sfplay~* -objektista tuleva audiosignaali kuuluu monona keskelle panoroituna, se kytketään *dac~* -objektin molempiin sisäänmenoihin, jolloin se kuuluu molemmista kaiuttimista yhtä voimakkaasti.

Viiveet toteutetaan *tapin~* ja *tapout~* -objekteilla. *Tapout~* -objekteihin liitetyillä *flonum* -objekteilla määritetään viiveiden pituudet. Viiveiden aloitusarvoina ovat vasemman puolen viiveessä 20 ms ja oikean puolen viiveessä 35 ms.

Patchin keskellä on *slider* -objekti, jolla säädetään molempien viivekanavien äänenvoimakkuus yhtäaikaan. Slider-objektin lähtösignaalin skaala on 0-127. Se skaalataan



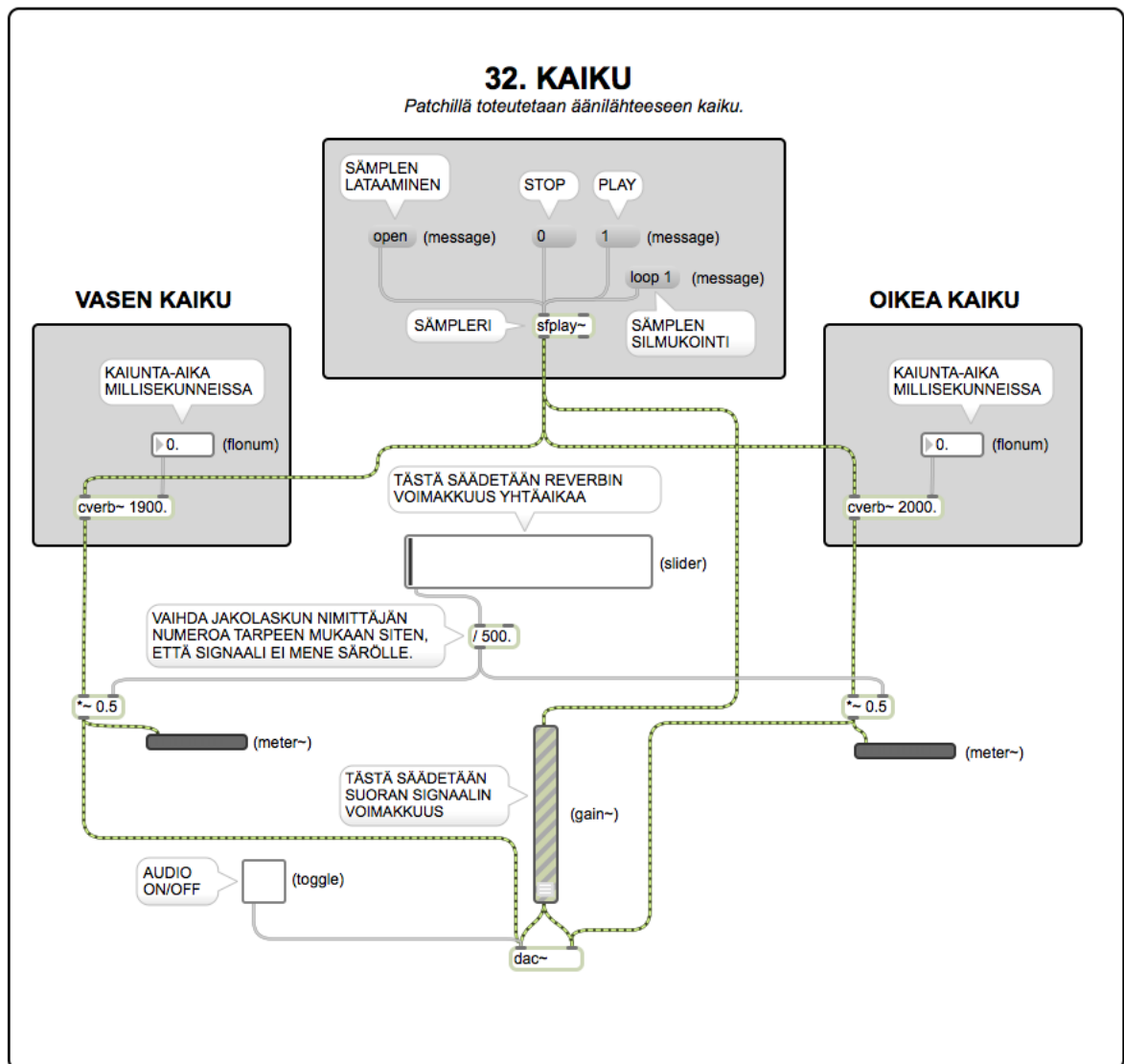
välille 0-1 jakamalla se / -objektilla. Piste jakajan 127 perässä ilmaisee tuloksen olevan desimaaliluku. Skaalaus tehdään siksi, että *~ -objekti vaatii tuloargumentin arvon skaalassa 0-1.

Mikäli viivästetty signaali säröytyy, suurennetaan jakolaskuobjektin jakajaa tarpeen mukaan. *~ -objektit, joiden argumentteina on luku 0,5, toteuttavat viivästettyjen signaalien äänenvoimakkuuksien skaalauksen käyttäjän säätämän sliderin mukaisesti. Meter~ -objektit toimivat monitoreina, joista näkee signaalivoimakkuuksien tasot.

PATCH 32: KAIKU

Patchin toiminta ja objektien funktiot

Tällä patchillä toteutetaan monoäänilähteeseen kaiku. Se on toteutettu muuten täysin samalla tavalla kuin edeltävä *“Viiveillä toteutettu pieni tila”* -patch, mutta *tapin~* ja *tapout~* -objektien tilalla tässä patchissä on *cverb~* -objekti, *kaikulaite* -objekti. *Cverb~* -objektin ainoa argumentti on jälkikaiun pituus millisekunteina määriteltynä. Jälkikaiun pituus määritetään *cverb~* -objektin oikeaan sisäänmenoon kytketyllä *flonum~* -objektilla.



PATCH 33: KOORUS

Patchin toiminta ja objektien funktiot

Koorus-efektillä sisään tulevaa audiosignaalia viivästetään siten, että viiveaikaa moduloidaan jatkuvasti n. 20-30 millisekunnin välillä. Yleisin tapa on valita viiveen muuttumistavaksi jokin säännöllinen aaltomuoto (sini- tai kolmioaalto), jonka taajuus on luokkaa 1-20 Hz (Laaksonen 2006: 367-368).

Tämä viivästyksen jatkuva moduloiminen on kooruksen identifioiva ominaisuus. Viivästetty signaali summataan alkuperäisen audiosignaalin rinnalle käyttäjän määrittämässä suhteessa.

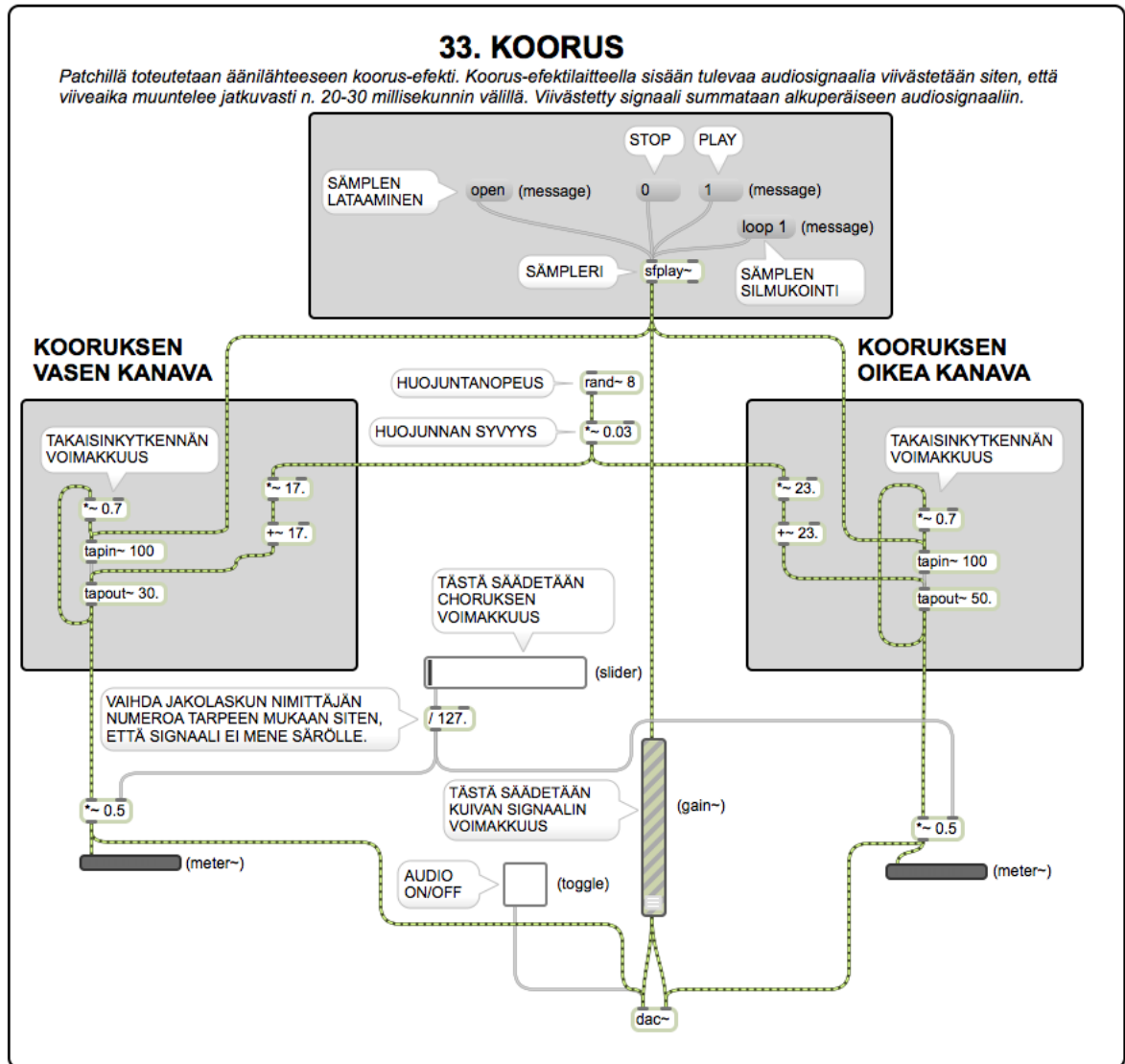
Signaalin viivästys ja takaisinkytkentä

Signaalin viivästäminen toteutetaan *tapin~* - ja *tapout~* -objekteilla. *Tapin~* -objektin argumentilla määritetään viiveen maksimipituus. *Tapin~* -objekti on kytketty *tapout~* -objektiin, jonka argumentilla määritetään käytössä olevan viiveen pituus. *Tapout~* -objektin ulostulo on kytketty **~* -objektin sisäänmenoon. Tämä kytkentä toteuttaa takaisinkytkennän. **~* -objektin argumenttina olevalla desimaaliluvulla, 0,7, määritetään takaisinkytkennän voimakkuus.

Viiveaikojen modulointi

Vasemman kanavan *Tapin~* ja *tapout~* -objekteilla toteutettua viivettä moduloidaan jatkuvasti "*rand~ 8*" -, "**~ 0.03*" -, "**~ 17.*" -, "*+~17.*" -objektien rakenteella. Oikean kanavan *Tapin~* ja *tapout~* -objekteilla toteutettua viivettä moduloidaan jatkuvasti "*rand~ 8*" -, "**~ 0.03*" -, "**~ 23.*" -, "*+~23.*" -objektien rakenteella.

Edeltävistä rakenteista "*rand~ 8*" -, "**~ 0.03*" - objektit ovat yhteiset sekä vasemmalle että oikealle kanavalle. *Rand~* -objekti generoi satunnaislukuja käyttäjän määrittämällä taajuudella. Taajuus määritetään *rand~* -objektin argumentilla, joka tässä patchissä on kahdeksan kertaa sekunnissa. *Rand~* -objekti määrittää kooruksen huojuuntanopeuden (*chorus rate*). *Rand~* -objekti on kytketty **~* -objektiin, jonka argumenttina on luku 0.03. Tällä kertolaskuobjektilla vahvistetaan *rand~* -objektista tulevan signaalin amplitudia ja sillä määritetään kooruksen syvyys (*chorus depth*). Näiden objektien



jälkeen kooruksen vasemman ja oikean kanavan toteutukset eriytyvät. Vasemmassa kanavassa “*~ 0.03” -objektin perään on kytketty sarjaan “*~ 17.” -objekti sekä “+~ 17.” -objekti. Näistä edeltävällä “*~ 17.” -objektilla vahvistetaan signaalin amplitudia. “+~ 17.” -objektilla satunnainen vaihteluväli siirretään luvun 17 välittömään läheisyyteen. +~ -objektista tulee ulos desimaalilukuja noin lukujen 16 ja 18 väliltä. Oikeassa kanavassa “*~ 0.03” -objektin perään on kytketty sarjaan “*~ 23.” -objekti ja “+~ 23.” -objekti. Näistä edeltävällä “*~ 23.” -objektilla vahvistetaan signaalin amplitudia. “+~ 23.” -objektilla satunnainen vaihteluväli siirretään luvun 23 välittömään läheisyyteen. +~ -objektista tulee ulos desimaalilukuja noin lukujen 22 ja 24 väliltä.

Tason säätö

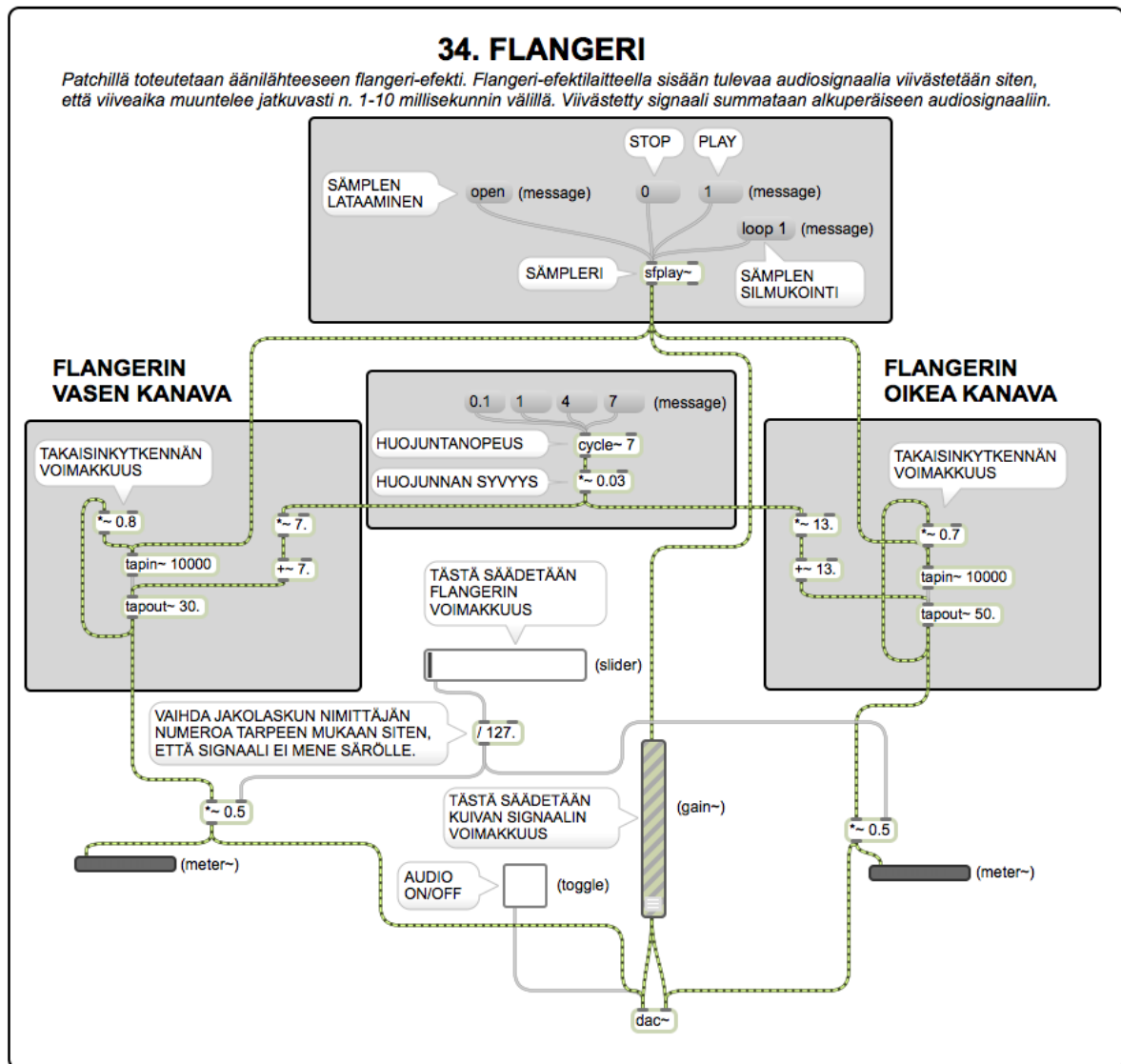
Tapout~ -objektit on kytketty *~ -objekteihin, joiden argumentteina on 0,5. Näillä *~ -objekteilla määritetään koorus-efektin voimakkuus kuuntelussa. *Slider*-objekti on kytketty näihin *~ -objekteihin. *Slider*-objektilla säädetään sekä vasemman, että oikean kanavan koorus-efektin voimakkuutta yhtäaikaan.

PATCH 34: FLANGERI

Patchin toiminta ja objektien funktiot

Patchillä toteutetaan äänilähteeseen flangeri-efekti. Flangeri-efektissä sisään tulevaa audiosignaalia viivästetään siten, että viiveaikaa moduloidaan jatkuvasti n. 1-10 millisekunnin välillä (Laaksonen 2006: 368). Viivästetty signaali summataan alkuperäisen audiosignaalin rinnalle käyttäjän määrittämässä suhteessa.

Keskeinen ero kooruksen ja flangerin välillä on se, että siinä missä kooruksen viiveet vaihtelevat n. 20-30 millisekunnin välillä, niin flangerin viiveet vaihtelevat n. 1-10 ms välillä. Tämä saa aikaan selkeän sävyeron soinnissa, jonka vuoksi koorusta ja flangeria kutsutaan omilla nimillä. Flangeri-patch poikkeaa koorus-patchistä nro 33 siten, että koorus-patchin vasemman kanavan viive vaihtelee 17 ms ja oikea kanava 23 ms ympärillä, kun taas flangeri-patchin vasemman kanavan viive vaihtelee 7 ms ja oikea kanava 13 ms ympärillä. Toinen ero näiden patchien välillä on se, miten huojuunta on toteutettu. Koorus-patchissä huojuunta toteutetaan *rand~* -objektilla, joka on satunnaissignaali-generaattori. Flangeri-patchissä huojuunta toteutetaan *cycle~* -objektilla, joka generoi sinimuotoista signaalia seitsemän hertsin taajuudella. Tätä taajuutta voidaan muuttaa message-viesteillä, joiden vaihtoehdot ovat 0,1, 1, 4 ja 7 hertsiä. Lisäksi koorus-patchissä takaisinkytkennän voimakkuus on molemmissa kanavissa 0,7. Flangeri-patchissä takaisinkytkennän voimakkuus vasemmassa kanavassa on 0,8 ja oikeassa kanavassa 0,7.



PATCH 35: FEISERI

Patchin toiminta ja objektien funktiot

Patchillä toteutetaan äänilähteeseen feiseri -efekti. Feiserissä äänisignaali menee yhden tai useamman kokopäästösuotimen (*all-pass filter*) läpi. Suotimilla muokataan äänisignaalin vaihetta.

Soiva äänilähde tässä patchissä on *sfplay~* -objekti, joka on sämpleri. *Sfplay~* -objektin ulostulo on kytketty **~* -objektiin sekä *cascade~* -objektiin. *Filtergraph~* -objektilla ohjataan *cascade~* -objektia. Ks. myös kohta 10.2 Toistuvat objektirakenteet. *Filtergraph~* -objektiin välitetty *allpass* -viesti tekee suotimesta kokopäästösuotimen. Klikattaessa *Filtergraph~* -objektiin liitettyyn "Phasespect 1" -viestiin, *filtergraph~* -objekti menee

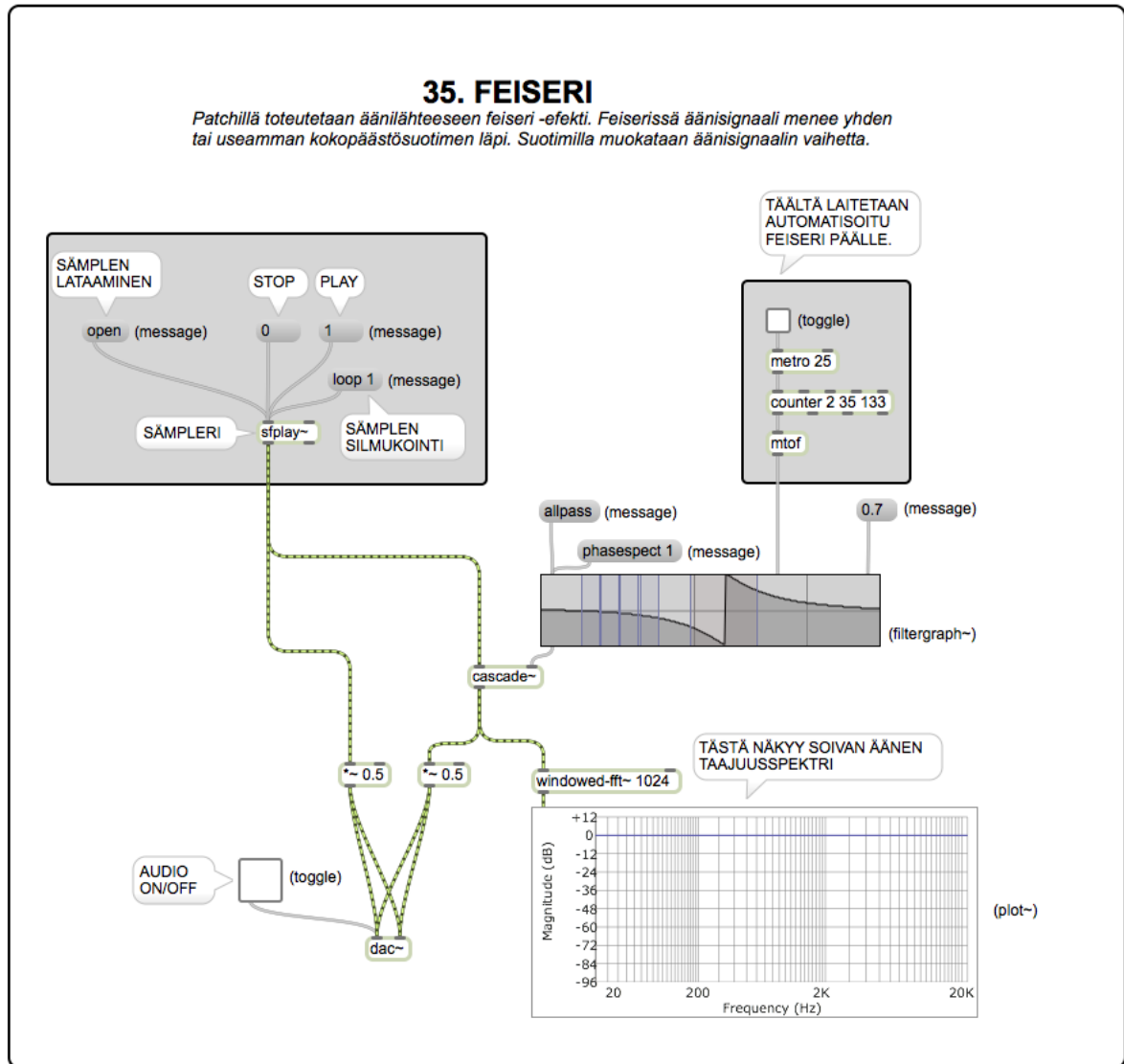
Display Phase Spectrum -moodiin. Tässä *Display Phase Spectrum* -moodissa *filtergraph* -objekti näyttää noin 90 asteen vaihesiirtymän keskitaajuuden molemmiin puolin.

Patchissä on rakenne, jolla voidaan kytkeä automatisoitu feiseri päälle. Tämä rakenne on toteutettu seuraavilla objekteilla: *toggle*, *metro*, *counter* ja *mtof*. *Toggle*-objekti kytkee automaation päälle ja pois. *Metro*-objekti määrittää automatisoidun feiserin nopeuden, uusi bang viesti liipaistaa 25 millisekunnin välein. *Counter*-objekti on laskuri, josta tulee ulos lukuja 35:n ja 133:n välillä alkaen luvusta 35 ja päättyen lukuun 133. *Counter*-objektin ensimmäinen argumentti määrittää moodin, jossa counter-objekti on. Kun argumenttina on numero kaksi, on counter-objekti "*heiluri*"-moodissa (*up-down counter*). Heiluri-moodissa laskuri menee ylöspäin asteittain 1, 2, 3,... 131,132, 133:een asti, jonka jälkeen laskuri lähtee tulemaan asteittain alaspäin: 132,131,130, etc.

Mtof -objektin kirjainlyhennelmä tulee sanoista *midi to frequency*. *Mtof* -objekti tulkitsee sisäänsyötetyt luvut midi-nuotteina ja tekee niistä muunnoksen luvuiksi, jotka edustavat taajuuksia. Kun *counter*-objektiin syötetään luku 35, muuntaa *mtof* -objekti tämän luvuksi 61 (Hz). Kun *counter*-objektiin syötetään luku 133, muuntaa *mtof* -objekti tämän luvuksi 17739 (Hz). Luvut 35 ja 133 on valittu *counter*-objektille sen mukaan, että ne edustavat suurinpiirtein ihmisen kuuloalueen ääripäitä. Alarajalla luvun 35 tilalla olisi voitu mennä jopa lukuun 18 asti joka edustaisi 23 Hz:n taajuutta ja ylärajalla luvun 133 tilalla olisi voitu mennä jopa lukuun 135 asti joka edustaisi 19912 Hz:n taajuutta. Ala- ja ylärajoja ei haluttu viedä niin lähelle ihmisen kuuloalueen äärirajoja, koska jos audiosignaalia kuunnellaan tietokoneen omilla kaiuttimilla tai muilla heikkolaatuisilla kaiuttimilla, on kaiuttimien matalimpien ja korkeimpien taajuuksien toistokyky usein heikohko.

Filtergraph~ -objektin kuvaajaan voi myös tarttua hiirelle kiinni ja kuunnella miltä feiseri kuullostaa käsin säädettynä. Kun feiseriä säädetään käsin, tulee automaatio kytkeä pois päältä automatisoidun feiseri -osion *toggle*-objektilla.

Filtergraph~ -objektin oikeanpuolinmaiseen sisäänmenoon kytketyllä *message*-viestillä, jonka arvona on 0.7, saadaan feiseri-efekti erottumaan hyvin. Tätä lukua voidaan muuttaa ja kokeilla miten se vaikuttaa feiserin äänensävyyn.

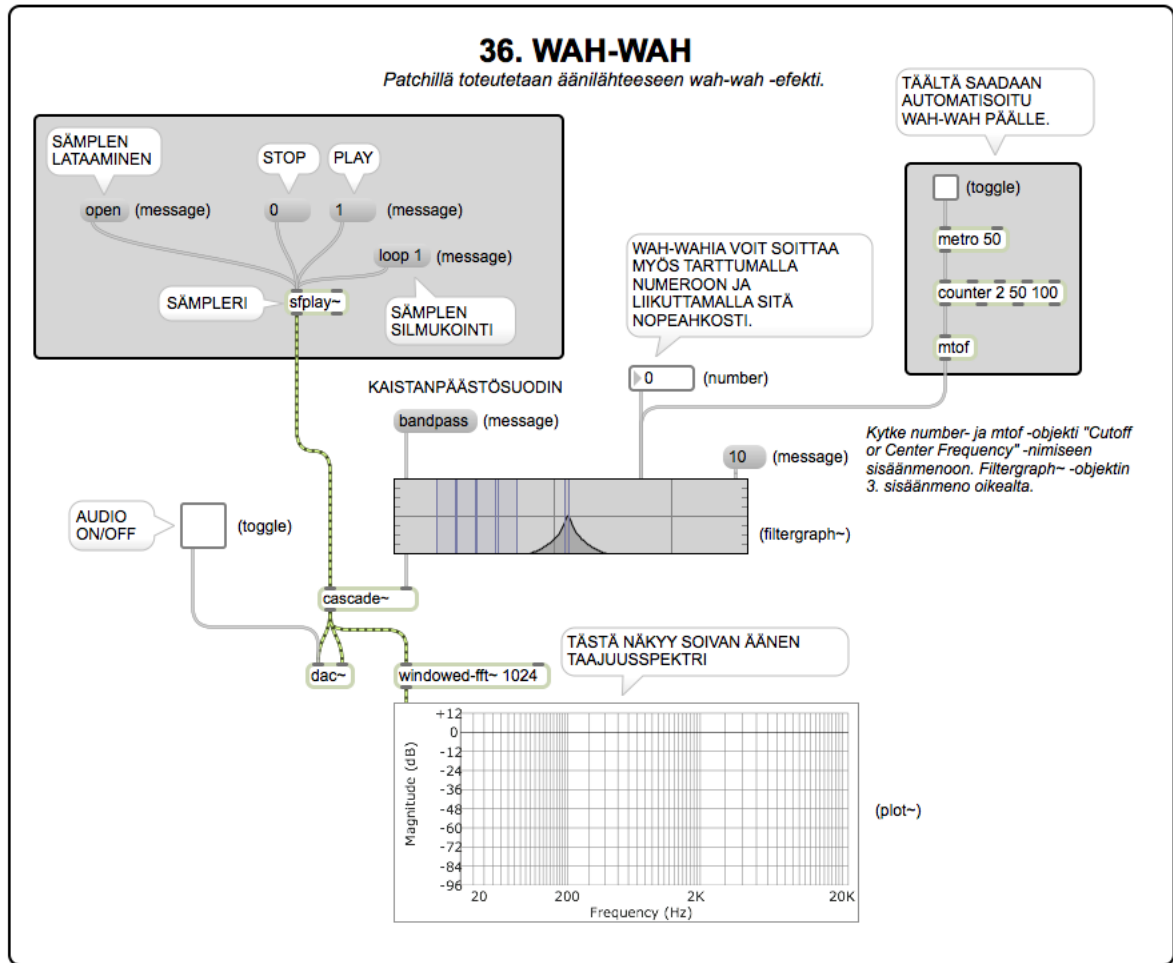


PATCH 36: WAH-WAH

Patchin toiminta ja objektien funktiot

Patchillä toteutetaan äänilähteeseen wah-wah -efekti. Wah-wah-patchillä audiosignaalia muokataan suotimella siten, että suotimen korostetun alueen keskitaajuutta liu'utetaan ylös- alas suunnassa (*spektrinen glissando*) (Wah-wah pedal 2014).

Soiva äänilähde tässä patchissä on *sfplay~* -objekti, joka on sämpleri. *Sfplay~* -objektin ulostulo on kytketty *cascade~* -objektiin. *Filtergraph~* -objektilla ohjataan *cascade~* -objektia. *Filtergraph~* -objektiin liitetty *bandpass* -viesti tekee suotimesta kaistanpäästösuotimen. Ks. myös kohta 10.2 Toistuvat objektirakenteet.

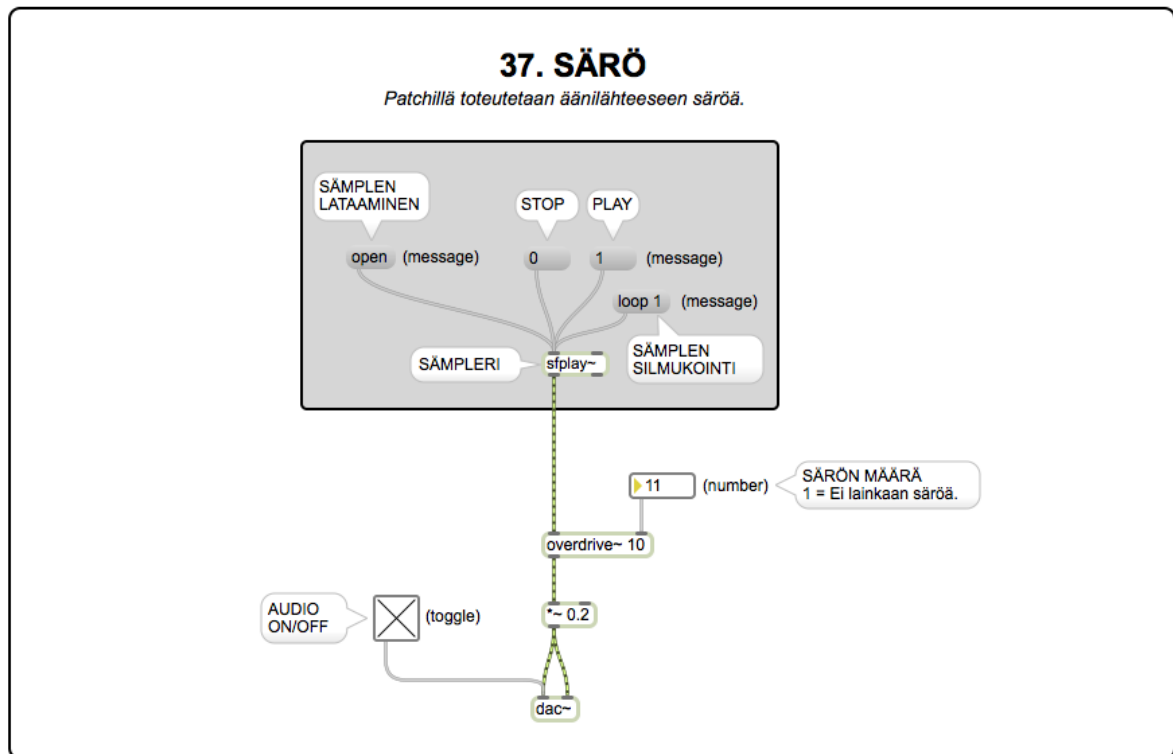


Filtergraph~ -objektiin kytketyllä *number*-objektilla voidaan “soittaa” wah-wahia manuaalisesti. Tähän samaan sisäänmenoon on kytketty myös *toggle*-, *metro*-, *counter*- ja *mtof* -objektien muodostama rakenne, joka säättää wah-wahin suotimen keskitaajuutta automaattisesti. Tämän automaattiorakenteen toimintaperiaate on selitetty feiseri-patchin tekstiosiossa. Suotimen keskitaajuutta voi määrittää myös klikkaamalla hiirellä *filtergraph~* -objektiin ja liikuttelemalla hiirtä.

PATCH 37: SÄRÖ

Patchin toiminta ja objektien funktiot

Tällä patchillä toteutetaan äänilähteeseen säröä. Soiva äänilähde tässä patchissä on *sfplay~* -objekti, joka on sämpleri. *Sfplay~* -objekti on kytketty *overdrive~* -objektiin. *Overdrive~* -objekti säröyttää *sfplay~* -objektista tulevan audiosignaalin. *Overdrive~* -objektin ainoa argumentti määrittää särön määrän. Kun argumentin arvo on 1, tällöin ei ole lainkaan säröä.



Mitä suurempi luku on sitä enemmän *overdrive~* -objekti säröyttää signaalia. Särön määrää voidaan säätää *overdrive~* -objektiin kytketyllä *number* -objektilla.

10.4.4 Muut patchit

PATCH 38: ERILAISET JOHTOTYYPIT

Patchin toiminta ja objektien funktiot

Patch havainnollistaa erilaiset Max-ohjelmassa käytettävät virtuaaliset johtotyypit. Max-ohjelman virtuaaliset kytkentäjohdot ovat joko yksivärisiä johtoja tai kaksivärisiä katkoviivajohtoja.

Max-objektien sisäänmenoissa ja ulostuloissa käytetään aina yksivärisiä johtoja. Yksivärisissä johdoissa data liikkuu hitaammin kuin kaksivärisissä katkoviivajohdoissa. Yksivärisissä johdoissa liikkuvat yksittäiset lukuarvot voivat edustaa mm. sävelkorkeutta, midi-kontrolleridataa, argumentteja matemaattisille operaattoreille tai tuloksia matemaattisilta operaattoreilta.

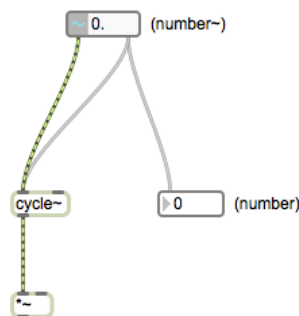
MSP-objektien sisäänmenoihin voidaan liittää sekä yksivärisiä johtoja että kaksivärisiä katkoviivajohtoja. MSP-objektien ulostuloista lähtee kaksivärisiä katkoviivajohtoja, joita ei ole mahdollista kytkeä Max-objektien sisäänmenoihin. MSP-objektien ulostuloista signaali lähtee eteenpäin Max-ohjelman *Audio Status* -ikkunassa määritellyllä näytteenottotaajuudella, esimerkiksi 44100 lukua sekunnissa.

Jitter -objekteissa käytetään myös kaksivärisiä katkoviivajohtoja. Kaksivärisissä katkoviivajohdoissa liikkuva data on joko audio- tai kuvasignaalia.

Lisäksi on hybridi-objekteja, esim. *number~* -objekti, joissa on kaksi erillistä ulostuloa. Toisesta ulostulosta lähtee yksivärinen johto ja toisesta kaksivärinen johto.

38. ERILAISET JOHTOTYYPIT

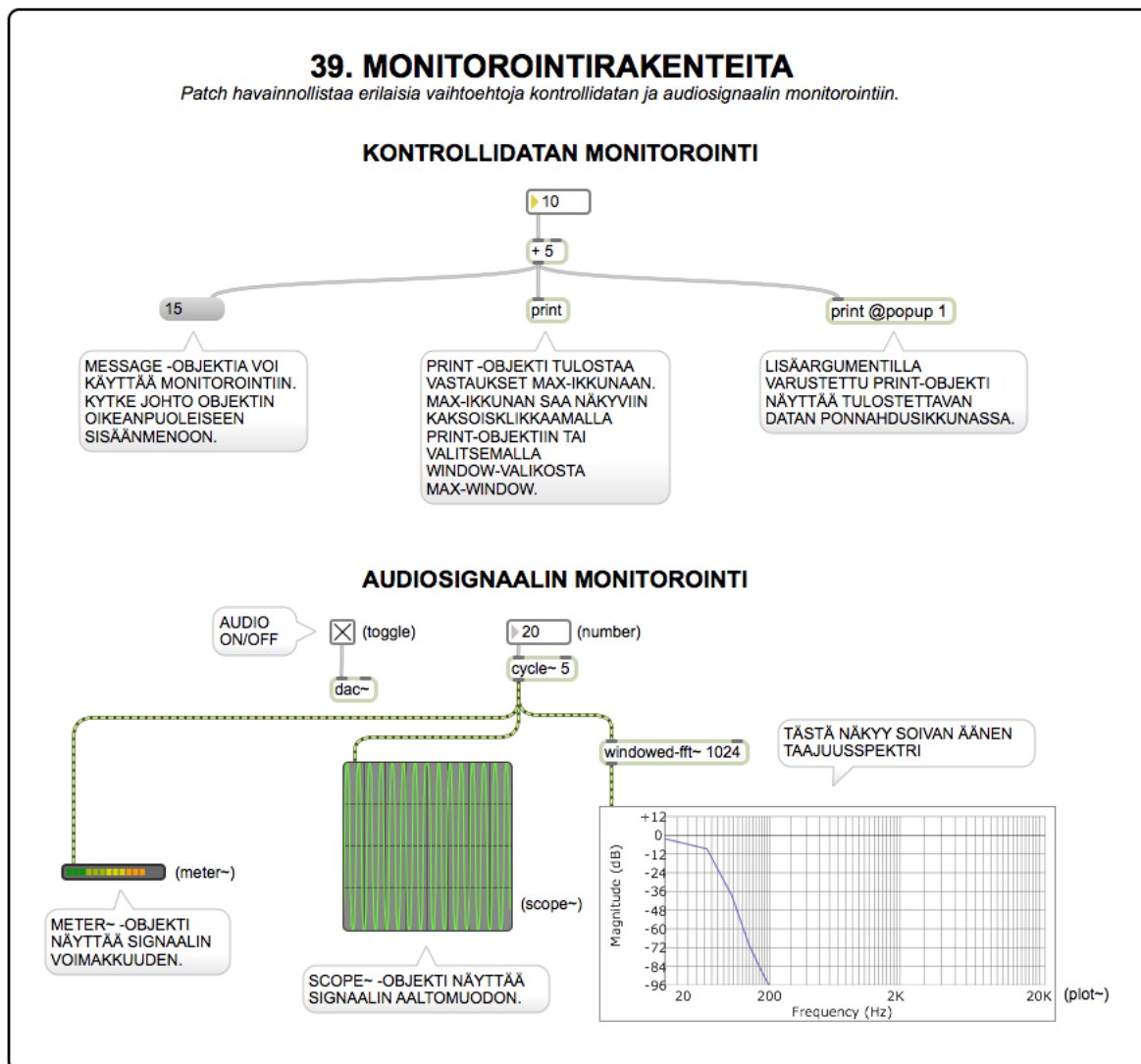
Patch havainnollistaa erilaiset johtotyytit, joita Max:ssä käytetään.



PATCH 39: MONITOROINTIRAKENTEITA

Patchin toiminta ja objektien funktiot

Patch havainnollistaa erilaisia vaihtoehtoja kontrollidatan ja audiosignaalin monitorointiin. Selvitykset tämän patchin objekteista on sisällytetty patchin kuvassa oleviin puhekupliin.



PATCH 40: ULKOISEN KONTROLLERIN KYTKEMINEN

Patchin toiminta ja objektien funktiot

Tämä patch opettaa miten ulkoinen midi-kontrolleri kytketään *ctlin*-objektin avulla Max-ohjelmaan. *Ctlin* -objekti/-objekteja on mahdollista kytkeä kaikkiin tämän oppimateriaalikokonaisuuden patcheihin ja ohjata niissä haluttuja parametrejä.

Parametrejä, joita voidaan ulkoisesta kontrollerista säätää ovat mm. sävelkorkeus, äänenvoimakkuus, suotimen leikkaustaajuus, panorointi, vibraton nopeus, vibraton laajuus etc. Ulkoisen midi-kontrollerin kytkeminen Maxiin mahdollistaa soittamisen kaltaisen käyttöliittymän. Ulkoisessa midi-kontrollerissa voi olla lukuisia potentiometrejä ja liukuja, jotka kaikki voidaan kytkeä kontrolloimaan eri parametrejä.

Ctlin-objektissa on kolme ulostuloa. Oikeanpuoleisesta ulostulosta tuleva luku kertoo midi-kanavan, jonka kautta dataa tulee. Keskimmäisestä ulostulosta tuleva luku kertoo käytössä olevan kontrollerin id-luvun. Vasemmanpuoleisesta ulostulosta tulevat luvut kertovat missä "asennossa" kontrolleri on. Jotta dataa saadaan tulemaan ulos näistä ulostuloista, pitää ensin kytkeä midi-kontrolleri kiinni tietokoneen midi-liityntään tai usb-sisäänmenoon ja säätää kyseistä kontrolleria.

Kontrollerin id-luku on mahdollista syöttää myös *ctlin*-objektin ensimmäisenä argumenttina. Näin tehtäessä, *ctlin*-objektin ulostulojen määrä pienenee kahteen. Tällöin oikeanpuoleisesta ulostulosta tuleva luku kertoo midi-kanavan ja vasemmanpuoleisesta ulostulosta tulevat luvut kertovat missä "asennossa" kontrolleri on.

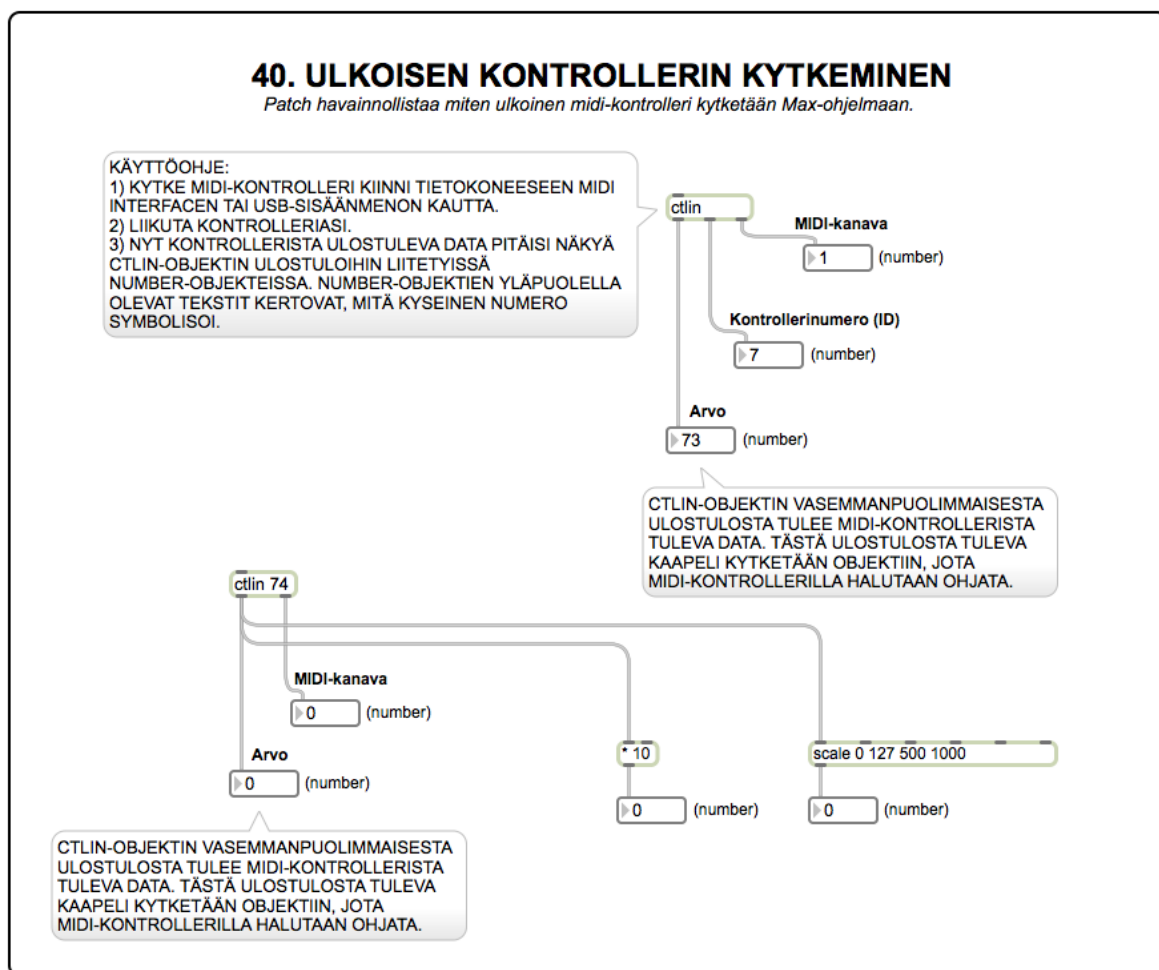
Kun *ctlin*-objektiin on kirjoitettu sisään argumentiksi tietty kontrolleri-id -luku esim.74, ottaa tämä *ctlin*-objekti vastaan dataa enää tältä kontrollerilta, jonka identiteetti on tässä tapauksessa 74.

Scale-objektilla skaalataan *ctlin*-objektista tuleva data. Kun sisään menee lukuja asteikolla 0-127, ulos tulevat luvut skaalataan lineaarisesti samassa suhteessa asteikolla 500-1000.

Patchin toinen vaihtoehto skaalaukselle on toteutettu *-objektilla. *-objekti, joka on kertolaskuobjekti, kertoo jokaisen sisääntulevan luvun 10:llä. Kun sisään menee lukuja

asteikolla 0-127, ulos tulevat luvut skaalataan lineaarisesti samassa suhteessa asteikolla 0-1270.

Scale -objektille ja *** -objektille määritetyt luvut olivat esimerkkejä, jotka käyttäjä voi vaihtaa mieleisikseen tarpeidensa mukaan. Esimerkeissä on käytetty asteikkoa 0-127, koska midi-kontrollereista data tulee usein ulos tällä asteikolla.



11 LOPPUPÄÄTELMÄT

Kaikki tehtävämonisteet on koekäytetty Jokilaaksojen musiikkiopiston musiikkiteknologiaryhmissä lukuvuoden 2013-2014 aikana. Loppukeväästä 2014 tein oppilaille kuuden kysymyksen kyselyn, jossa tiedustelin oppilaiden kokemusten pohjalta syntyneitä näkemyksiä Maxistä.

Seuraavassa on vastauksia kysymykseen: *“Miltä ohjelmoiminen Max-ohjelmalla on tuntunut?”*

“Nopealta ja miellyttävältä. Kun saa aikaan toimivan patchin se tuntuu hyvältä.”

“ Ohjelmointi on selventänyt eri äänenkäytön tekniikoita tietokoneissa hyvin. Se on myös suhteellisen sukkela.”

“ Mielenkiintoiselta ja “palkitsevalta” kun on saanut patchin toimimaan.”

“ Todella kivalta ”

“Mukavalta, varsinkin lopussa kun patchejä yhdistämällä ja soveltamalla on voinut tehdä jonkinasteisia “soittimia”. ”

“ Se oli aluksi hankalaa, mutta se onneksi helpotti, kun alkoi ymmärtää ns. “koodikieltä”. ”

“ Max-ohjelmalla ohjelmoiminen on tuntunut mielenkiintoiselta ja se on tarjonnut sopivasti haastetta.”

Seuraavassa on vastauksia kysymykseen: *“Miten Max-ohjelma mielestäsi soveltuu musiikkiteknologian opettamiseen n. 10-18v. ikäisille ihmiselle? Missä asioissa se on hyvä? Missä asioissa se on huono?”*

”Ohjelmassa hyvää on yksinkertaisuus. Sillä on myös helppo ymmärtää käsitteitä ja toimintoja. Sen takia se on hyvä myös aloittelijoille.”

“Se purkaa tietyt käsitteet, kuten taajuuksien vaiheen visuaalisesti ymmärrettäväksi. Kun patchin jokaisen toiminnon käsittää alusta loppuun asti, tulee useita “ahaa” -elämyksiä.”

Hyvin, koska se havainnollistaa outoja ja myöskin tuttuja käsitteitä.

Hyvä juuri havainnollistamisessa, huono jos ei ymmärrä patchin tarkoitusta käytännön tasolla.”

Max-ohjelma soveltuu mus.teknologian opetukseen hyvin, koska siinä näkyy kuinka eri mus.teknologian “osat” toimii ja millainen rakenne niissä on teoreettisesti. Soveltuu n. 15+ - ikäisille, koska se vaatii hieman ymmärrystä ja tietämystä.”

Mielestäni Max-ohjelma soveltuu musiikkiteknologian opettamiseen hyvin koska se on selkeä ja helppo käyttää. Se havainnollistaa hyvin mistä esim. eri efektit syntyvät.”

Oppilailleni esittämä kysymys: *“Miltä Max-patcheillä tuotetut äänet ovat kuulostaneet?”* toi esille kriittisen näkökulman Max:llä toteutetuista äänistä. Seuraavassa muutamia vastauksia:

“Todella raaoilta. Ei välttämättä tajuntaa räjäyttäviltä. Enimmäkseen ääni oli merkki, että jokin meni oikein reitityksessä.”

“Idea patchin tarkoituksesta on tullut selväksi, mutta äänen laatu ei ole hyvä.”

“Äänet on kuulostaneet tietokoneella tuotetuilta ääniltä.”

“tylsiltä, mutta kiva kuulla ääni että on onnistunut.”

“erittäin tekniseltä”

Seuraavassa vastauksia kysymykseen: Uskotko, että käytät Max-ohjelmaa äänien / musiikin teossa?

“Kyllä, olen käyttänyt sitä muutamiin erikoisempiin melodioihin.”

“Ehkä, mikäli käytäntöön soveltamista oppisi lisää.”

“En osaa nähdä maxia (vielä) hyötynä musiikin tekemiseen muuten kuin teoreettisena apuvälineenä.”

“Voisin ehkä joskus käyttää luodessa ohjelmia/pelejä.”

Seuraavassa vastauksia kysymykseen: Onko jotain muita ajatuksia, joita haluaisit jakaa Max-ohjelmasta?

“Kuinka patchejä voi soveltaa käytäntöön? Mihin patchejä tarvitaan, voiko luoda esimerkiksi musiikin tekoon laadultaan hyviä efektejä tai soundeja?”

“Edellä mainittu soveltaminen, jossain muodossa, voisi auttaa hahmottamaan aiheet paremmin (tehtävät, tms.)”

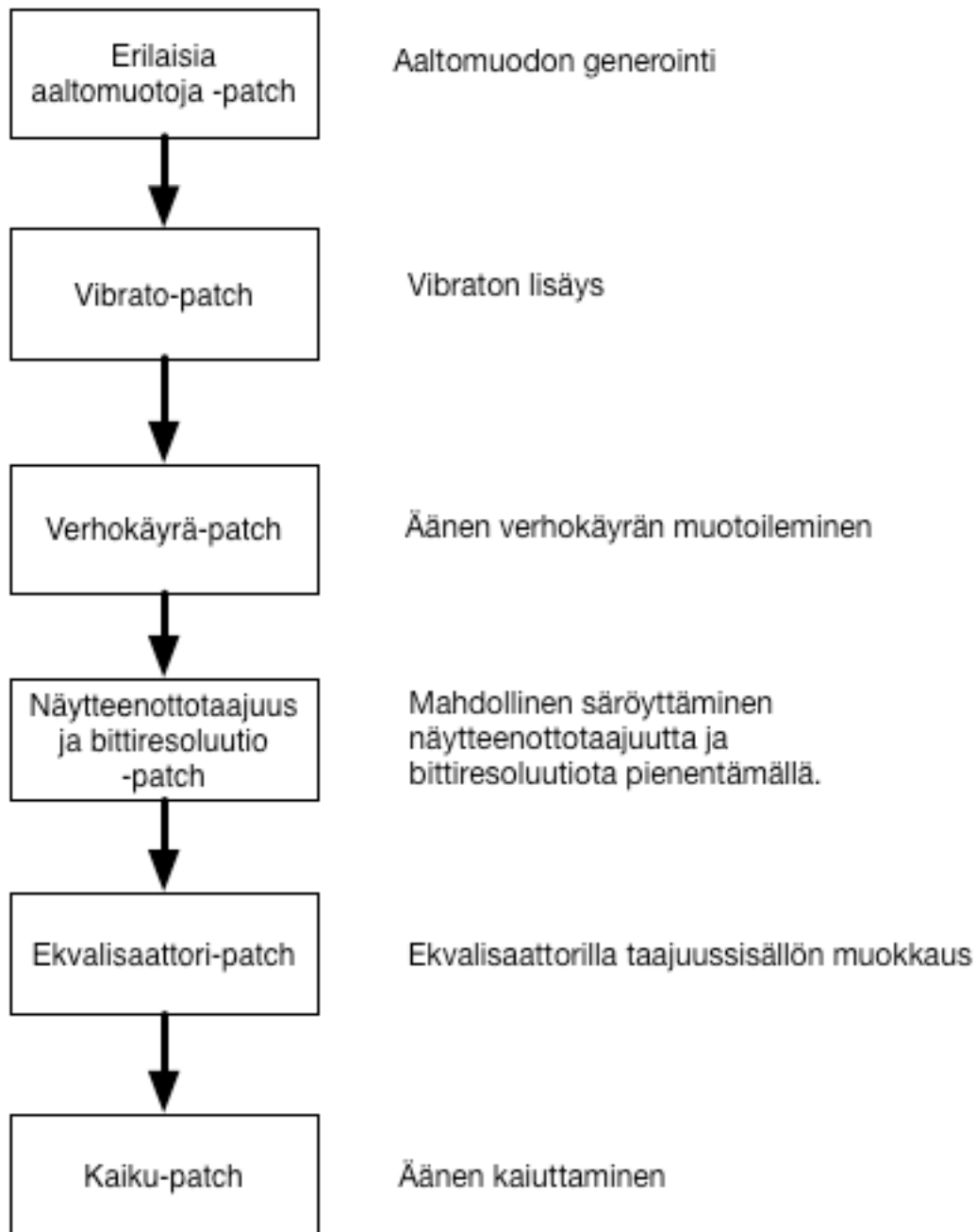
Näiden kysymyksien palautteissa kaivattiin parempaa, elämyksellisempää ja musiikillisempaa ääntä. Kaivattiin konkreettisempaa havainnollistamista siitä, mitä ja mihin näitä Max-patchejä voi käyttää. Ableton Live -ohjelma on yksi vastaus näihin toiveisiin. Ableton Live on live-esityksiin suunnattu audiosekvensseriohjelma, joka käyttää Maxiä *“moottorinaan”*. Ableton -yhtiön tekemä Ableton Live -ohjelma ja Cycling74-yhtiön tekemä Max-ohjelma ovat yhteistyössä luoneet Ableton Liveen sisäänrakennetun mahdollisuuden, Max4Liven, jolla Maxiä voidaan käyttää Ableton Livessä audion ja midin prosessointiin ja generointiin.

Lähes kaikki tämän oppimateriaalikonaisuuden patchit on mahdollista siirtää Ableton Liveen ja käyttää patchejä Ableton Live-ohjelmalla.

Sekvensseriohjelmat mahdollistavat nopean ja intuitiivisen käyttöliittymän kappaleen kokonaisuuden ja rakenteen hahmottamiseen, muokkaamiseen ja työstämiseen. Itse tulen jatkossa integroimaan Max-ohjelman opetukseen hyvin varhaisessa vaiheessa myös Ableton Live -ohjelman käytön, jotta Maxin ”*lähes rajattomien*” mahdollisuuksien merkitys havainnollistuu.

Eräs kysymys oppilailleni, ”*Miltä ohjelmoiminen Max-ohjelmalla on tuntunut?*”, toi seuraavanlaisen vastauksen: ”*Nopealta ja miellyttävältä. Kun saa aikaan toimivan patchin se tuntuu hyvältä. Se silti tuntuu liian paljon kopioimiselta, eli oma-aloitteista reititystä tai objektien luontia ei tapahtunut kovinkaan paljon.*” Minulle oli itsestään selvää, että patchejä voi ”*virittää*” itse niin paljon kuin huvittaa sen jälkeen, kun patchin on kerran saanut toimimaan. Selvästikään en ole sanonut oppilailleni tätä asiaa ääneen riittävän usein ja selkeästi, enkä ole varannut riittävästi aikaa omatoimiseen patchien ”*rassaamiseen*”. Tämän johdosta lisäsin kaikkien liitteessä olevien patchien alareunan teksteihin lauseet: ”*Saatuasi patchin valmiiksi, mieti miten voisit varioida ja kehittää patchiä eteenpäin. Toteuta ideasi!*”

Edeltävän lisäksi kannustan ihmisiä käyttämään näitä patchejä materiaalina joillekin isommille kombinaatiopatcheille. Seuraavassa on yksi esimerkki kokonaisuudesta, joka rakentuu oppimateriaalin yksittäisille patcheille.



Etukäteen pohdin myös voinko käyttää Max -ohjelmaa opetusohjelmana musiikkiteknologian opetuksessa lapsille ja nuorille? Onko ohjelma mahdollisesti liian vaikea? Oman opetuskokemukseni kautta olen tullut siihen lopputulokseen, että voi. Lisäksi oppilaille teettämäni kyselyn vastaukset tukevat tätä näkökulmaa.

Maxin kanssa samankaltainen ohjelma PD, Pure Data, oli mietinnässä, että pitäisikö tehtävät tehdä myös sille, koska se on ilmainen. Syitä miksi tein tehtävät vain Max:llä oli, että totesin, että Max:ssä on parempi sisäänrakennettu tuki objektien help-tiedostojen muodossa, jonka takia se soveltuu paremmin myös itseopiskeluun. Toinen syy oli aiemmin mainitsemani Maxin mahdollisuus yhteistoimintaan Ableton Live -ohjelman kanssa, joka PD:ltä puuttuu.

Kokonaisuutena voin todeta, että prosessi tämän opinnäytetyön läpiviemiseen on ollut erittäin opettavainen projekti niin minulle kuin myös oppilailleni. Näiden tehtäväpatchien olemassaolo monipuolistaa musiikkiteknologian perusteiden opetukseni. Tehtäväpatchien tekeminen syventää kenen tahansa aloittelijan tietämyksen ja taitotason ja antaa mukana tulevien linkkien avulla mahdollisuuksia itsenäisesti paneutua ja syventyä mihin tahansa tehtävissä esiteltyihin aiheisiin tai asiakokonaisuuksiin.

LÄHTEET

Kirjallisuus

Cook, Perry R. 2002. Real Sound Synthesis for Interactive Applications. Natick, Massachusetts: A K Peters.

Dodge, Charles & Jerse Thomas A. 1997 Computer Music, Second edition. U.S.A.: Schirmer.

Katz, Bob 2007. Mastering Audio - the art and the science. New York: Elsevier/Focal Press.

Landy, Leigh 2007. Understanding the Art of Sound Organization. London: The MIT Press.

Lepoluoto, Ari 2007. Luentomateriaalit.

Lindeman, Osmo 1980. Elektroninen musiikki. Helsinki: Otava.

Laaksonen, Jukka 2006. Äänityön kivijalka. Helsinki: Idemco Oy.

Moore, F. Richard 1990. Elements of Computer Music. Englewood Cliffs, New Jersey: P T R Prentice Hall.

Roads, Curtis with John Strawn, Curtis Abbott, John Gordon, and Philip Greenspun 1996. The Computer Music Tutorial, second printing. Cambridge, Massachusetts: The MIT Press.

Roads, Curtis 2001. Microsound. Cambridge, Massachusetts: The MIT Press.

Rossing, Thomas D., Moore, F. Richard & Wheeler, Paul A. 2002. The Science of Sound, third edition. Boston: Addison Wesley.

Silander, Simo 2000. Ohjelmointi, 3. korjattu painos. Jyväskylä: Satku - Kauppakaari.

Suntola, Silja 2000. Luova studiotyö. Helsinki: Idemco Oy.

Swedien, Bruce 2003. Make Mine Music. Norway: MIA Musikk.

Verkkolähteet

Amplitude Modulation 2014. [Viitattu 7.6.2014]. Saatavissa: http://en.flossmanuals.net/pure-data/ch021_amplitude-modulation/

Arduino 2014. [Viitattu 4.10.2014]. Saatavissa: <http://fi.wikipedia.org/wiki/Arduino>

Blomberg, Esa & Lepoluoto, Ari 1993/2005. Audiokirja. Toinen tarkistettu painos. Espoo: Tapiolan Viestintäsuunnittelu Oy. [Viitattu 3.3.2014]. Saatavissa: <http://ari.lepoluo.to/audiokirja/>

Brown, Howard Mayer & Palmer, Frances 2014. [Viitattu 1.10.2014] Grove Music Online. Saatavissa: <http://www.oxfordmusiconline.com.ezproxy.uniarts.fi/subscriber/article/grove/music/13818>

Chion, Michel 1983. Guide to Sound Objects - Pierre Schaeffer and musical research, english translation by John Dack and Christine North 2009. [Viitattu 14.2.2014]. Saatavissa: <http://www.ears.dmu.ac.uk/IMG/pdf/Chion-guide/>

Honkanen 2014. [Viitattu 12.6.2014]. Saatavissa: <http://gallia.kajak.fi/opmateriaalit/yleinen/honHar/ma/SUODATTIMET.pdf>

Instrumentti 2013. [Viitattu 9.8.2014]. Saatavissa: <http://fi.wikipedia.org/wiki/Instrumentti>

Lawson, Rex 2004. Concert Pianolist. [Viitattu 17.5.2014]. Saatavissa: www.rexlawson.com/index.html?contents.html&0

Linn, Roger 2011. New Controller Instruments. [Viitattu 29.3.2014]. Saatavissa: <http://www.rogerlinndesign.com/new-controller-instruments.html>

Music Radar 2013. 8 essential tools for mastering. [Viitattu 26.4.2014]. Saatavissa: <http://www.musicradar.com/tuition/tech/8-essential-tools-for-mastering-586768/8>

Pink Noise 2014. [Viitattu 29.9.2014]. Saatavissa: http://en.wikipedia.org/wiki/Pink_noise

Renaissance Man 2009. [Viitattu 23.8.2014]. Saatavissa: <http://www.thefreedictionary.com/Homo+universalis>

Smith, Geoff 2012. Creating & Using Custom Delay Effects. [Viitattu 7.6.2014]. Saatavissa: <http://www.soundonsound.com/sos/may12/articles/designer-delay.htm>

Sound Installation 2014. [Viitattu 20.9.2014]. Saatavissa: <http://www.ears.dmu.ac.uk/spip.php?rubrique153>

Wah-wah pedal 2014. [Viitattu 9.8.2014]. Saatavissa: http://en.wikipedia.org/wiki/Wah-wah_pedal

Ward, Kevin 2011. Creating Depth: The Haas effect. [Viitattu 22.3.2014]. Saatavissa: <http://mixcoach.com/creating-depth-the-haas-effect-2/>

White Noise 2014. [Viitattu 29.9.2014]. Saatavissa: http://en.wikipedia.org/wiki/White_noise#Music

Äänentoisto 2014. [Viitattu 6.9.2014]. Saatavissa: <http://fi.wikipedia.org/wiki/Äänentoisto>

Ohjelmamanuaalit

Max thispoly~ -object help file: MAX 6 Version 6.1.8 (9fafea6)(32-bit). 2014. San Francisco: Cycling74.

Max lookup~ -object help file: MAX 6 Version 6.1.8 (9fafea6)(32-bit). 2014. San Francisco: Cycling74.

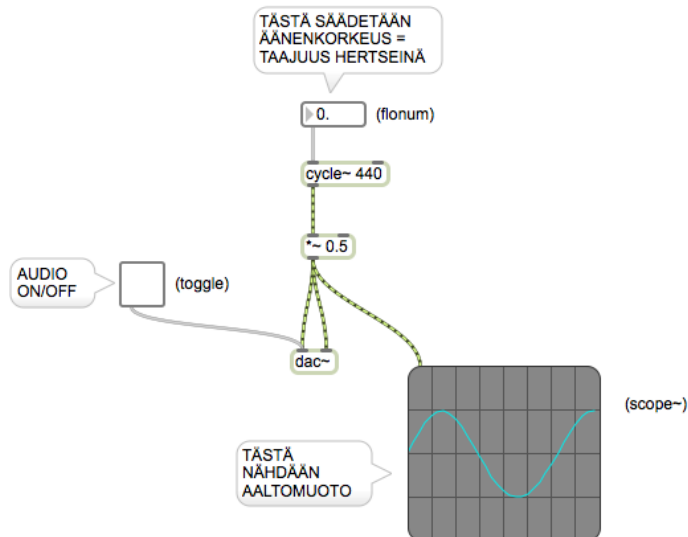
LIITTEET

Liite 1

Max-ohjelmalla toteutettu musiikkiteknologian perusteiden oppimateriaalikokonaisuus.

1. SINIÄÄNI

Tässä patchissä opitaan kytkemään objekteja toisiinsa. Patch mahdollistaa yhden siniäänen kuulemisen sekä aaltomuodon näkemisen.



HELP-TIEDOSTOT:

Objektin Help-tiedoston saa näkyviin klikkaamalla objektiin alt-näppäin pohjassa. Patchin tulee olla avattuna "unlocked"-moodissa.

INSPECTOR-IKKUNA:

Objektin Inspector-ikkunan saa näkyviin seuraavalla tavalla:

- 1) Aktivoi objekti, jonka Inspector-ikkuna halutaan avata.
- 2) Object-valikosta valitaan Inspector (Pikanäppäinkomento: cmd - i).

Saatuasi patchin valmiiksi, mieti miten voisit varioida ja kehittää patchiä eteenpäin. Toteuta ideasi!

KÄSITTEITÄ:

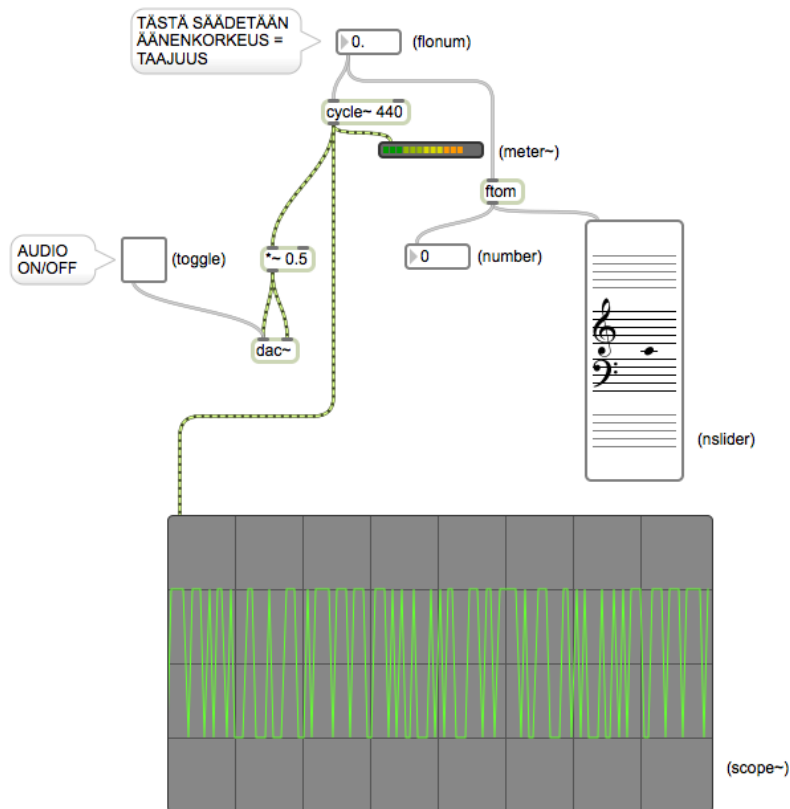
- värähtely
- äänirauta
- taajuus
- sävelkorkeus / äänenkorkeus
- amplitudi
- äänenvoimakkuus
- aallonpituus
- siniaalto
- kosiniaalto
- aaltoliike

LISÄÄ AIHEESTA:

http://opiskele.com/kurssit/musiikkifysiikka/html/01b_aanen_synty.shtml
<http://fi.wikipedia.org/wiki/Siniaalto>
<http://fi.wikipedia.org/wiki/Värähtely>

2. ÄÄNIOPIN JA MUSIIKKITEKNOLOGIAN PERUSKÄSITTEITÄ

Patchin avulla opiskellaan äänen perusominaisuuksia.



HELP-TIEDOSTOT:

Objektin Help-tiedoston saa näkyviin klikkaamalla objektiin alt-näppäin pohjassa. Patchin tulee olla avattuna "unlocked"-moodissa.

INSPECTOR-IKKUNA:

Objektin Inspector-ikkunan saa näkyviin seuraavalla tavalla:

- 1) Aktivoi objekti, jonka Inspector-ikkuna halutaan avata.
- 2) Object-valikosta valitaan Inspector (Pikanäppäinkomento: cmd - i).

Saatuasi patchin valmiiksi, mieti miten voisit varioida ja kehittää patchiä eteenpäin. Toteuta ideasi!

KÄSITTEITÄ:

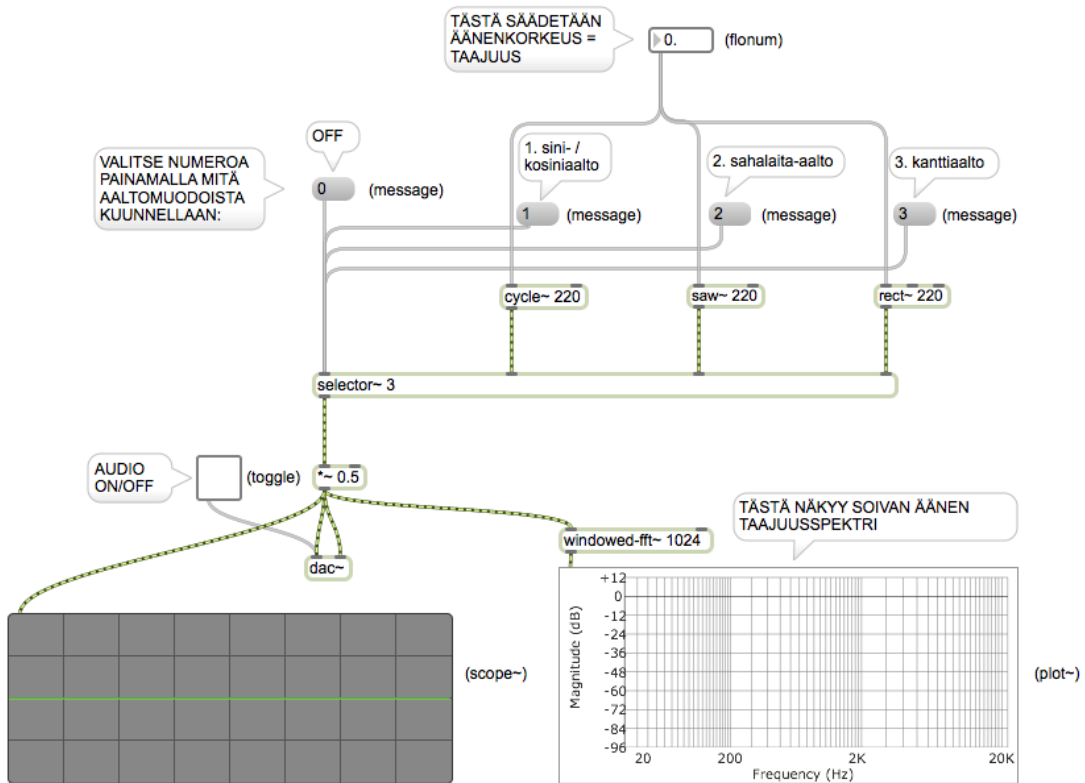
- ihmisen kuuloalue
- äänennopeus
- siniääni
- taajuus = äänenkorkeus
- amplitudi = äänenvoimakkuus
- jakso
- aallonpituus
- infra- ja ultraäänit

LISÄÄ AIHEESTA:

http://music.columbia.edu/cmc/musicandcomputers/chapter1/01_02.php
http://opiskele.com/kurssit/musiikkifysiikka/html/01e_nopeus_aallonpituus.shtml
http://opiskele.com/kurssit/musiikkifysiikka/html/01d_taaajuus.shtml
http://opiskele.com/kurssit/musiikkifysiikka/html/01c_voimakkuus.shtml
<http://fi.wikipedia.org/wiki/Kuulo>
http://fi.wikipedia.org/wiki/Äänen_nopeus
<http://fi.wikipedia.org/wiki/Taajuus>
<http://fi.wikipedia.org/wiki/Amplitudi>
<http://fi.wikipedia.org/wiki/Aallonpituus>
<http://fi.wikipedia.org/wiki/Vaihe>

3. ERILAISIA AALTOMUOTOJA

Patch esittelee erilaisia aaltomuotoja (sini-, kosini-, sahalaita- ja kanttiaalto).



HELP-TIEDOSTOT:

Objektin Help-tiedoston saa näkyviin klikkaamalla objektiin alt-näppäin pohjassa. Patchin tulee olla avattuna "unlocked" -moodissa.

INSPECTOR-IKKUNA:

Objektin Inspector -ikkunan saa näkyviin seuraavalla tavalla:

- 1) Aktivoi objekti, jonka Inspector -ikkuna halutaan avata.
- 2) Object -valikosta valitaan Inspector (Pikanäppäinkomento: cmd - i).

Saatuasi patchin valmiiksi, mieti miten voisit varioida ja kehittää patchiä eteenpäin. Toteuta ideasi!

KÄSITTEITÄ:

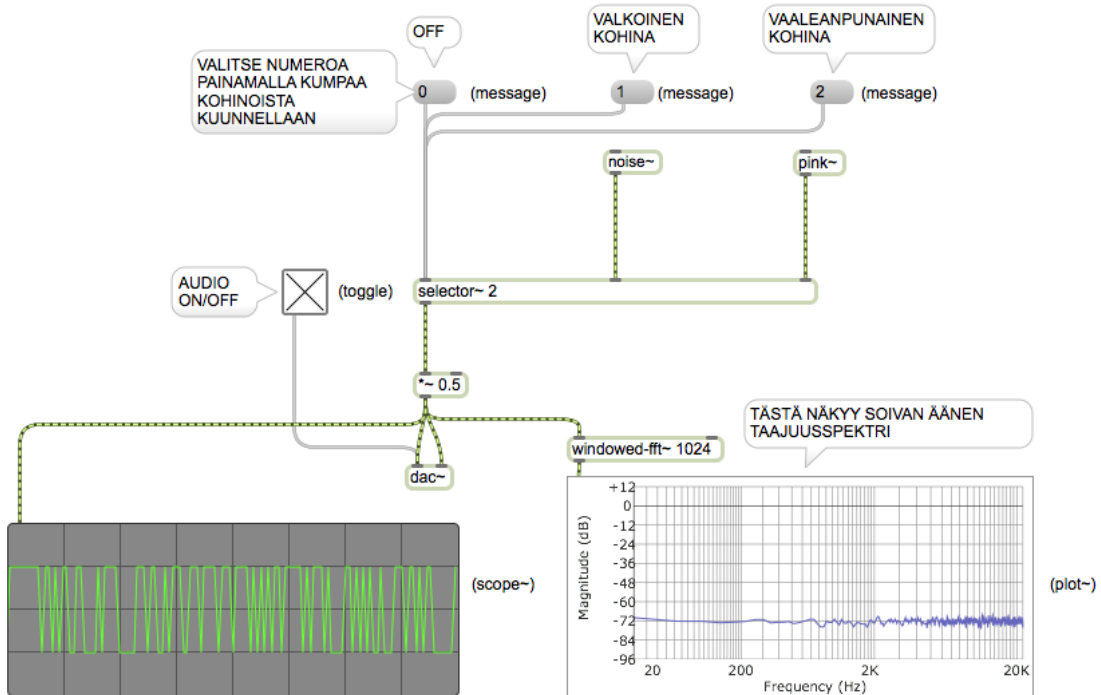
- siniaalto
- kosiniaalto
- sahalaita-aalto
- kanttiaalto

LISÄÄ AIHEESTA:

<http://fi.wikipedia.org/wiki/Kanttiaalto>
<http://fi.wikipedia.org/wiki/Kolmioaalto>
http://en.wikipedia.org/wiki/Sawtooth_wave

4. KOHINATYYPPEJÄ

Patch esittelee erilaisia kohinoita (valkoinen ja vaaleanpunainen kohina).



HELP-TIEDOSTOT:

Objektin Help-tiedoston saa näkyviin klikkaamalla objektiin alt-näppäin pohjassa. Patchin tulee olla avattuna "unlocked"-moodissa.

INSPECTOR-IKKUNA:

Objektin Inspector -ikkunan saa näkyviin seuraavalla tavalla:
1) Aktivoi objekti, jonka Inspector -ikkuna halutaan avata.
2) Object -valikosta valitaan Inspector (Pikanäppäinkomento: cmd - i).

Saatuasi patchin valmiiksi, mieti miten voisit varioida ja kehittää patchiä eteenpäin. Toteuta ideasi!

KÄSITTEITÄ:

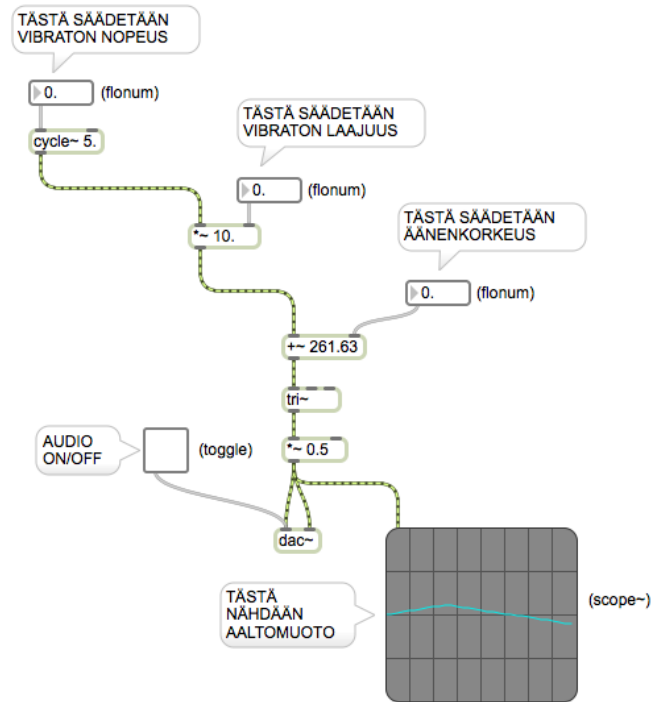
- jaksollinen ääni / jaksollinen värähtely
- jaksoton ääni / jaksoton värähtely
- valkoinen kohina
- vaaleanpunainen kohina

LISÄÄ AIHEESTA:

<http://fi.wikipedia.org/wiki/Kohina>
http://en.wikipedia.org/wiki/Colors_of_noise
[http://en.wikipedia.org/wiki/Noise_\(signal_processing\)](http://en.wikipedia.org/wiki/Noise_(signal_processing))

5. VIBRATO

Patchillä voidaan demonstroida mm. vibratolla toteutettu "ambulanssi" -ääni.



HELP-TIEDOSTOT:

Objektin Help-tiedoston saa näkyviin klikkaamalla objektiin alt-näppäin pohjassa. Patchin tulee olla avattuna "unlocked" -moodissa.

INSPECTOR-IKKUNA:

Objektin Inspector -ikkunan saa näkyviin seuraavalla tavalla:

- 1) Aktivoi objekti, jonka Inspector -ikkuna halutaan avata.
- 2) Object -valikosta valitaan Inspector (Pikanäppäinkomento: cmd - i).

Saatuasi patchin valmiiksi, mieti miten voisit varioida ja kehittää patchiä eteenpäin. Toteuta ideasi!

KÄSITTEITÄ:

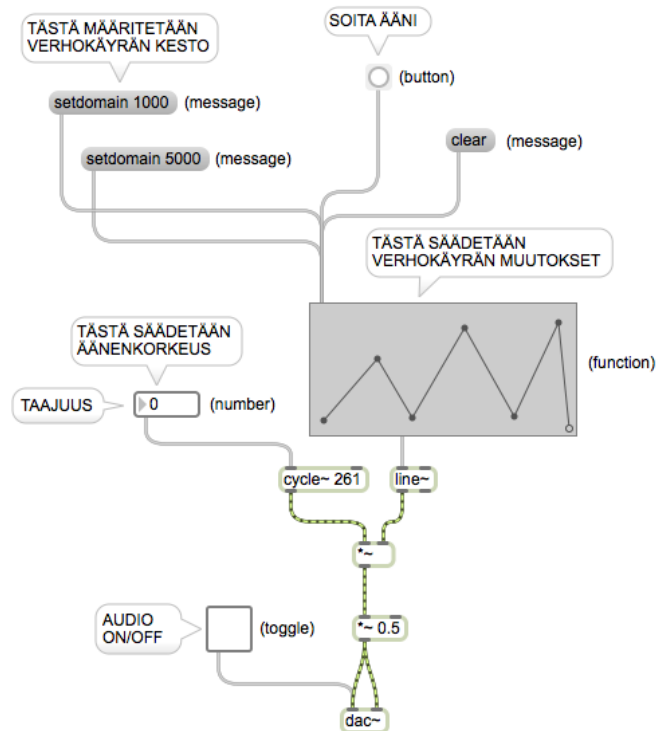
- vibrato
- vibraton laajuuden säätäminen
- vibraton nopeuden säätäminen

LISÄÄ AIHEESTA:

<http://fi.wikipedia.org/wiki/Vibrato>

6. VERHOKÄYRÄ

Patchissä verhokäyrällä ohjataan äänenvoimakkuutta.



HELP-TIEDOSTOT:

Objektin Help-tiedoston saa näkyviin klikkaamalla objektiin alt-näppäin pohjassa. Patchin tulee olla avattuna "unlocked" -moodissa.

INSPECTOR-IKKUNA:

Objektin Inspector -ikkunan saa näkyviin seuraavalla tavalla:
1) Aktivoi objekti, jonka Inspector -ikkuna halutaan avata.
2) Object -valikosta valitaan Inspector (Pikanäppäinkomento: cmd - i).

Saatuasi patchin valmiiksi, mieti miten voisit varioida ja kehittää patchiä eteenpäin. Toteuta ideasi!

KÄSITTEITÄ:

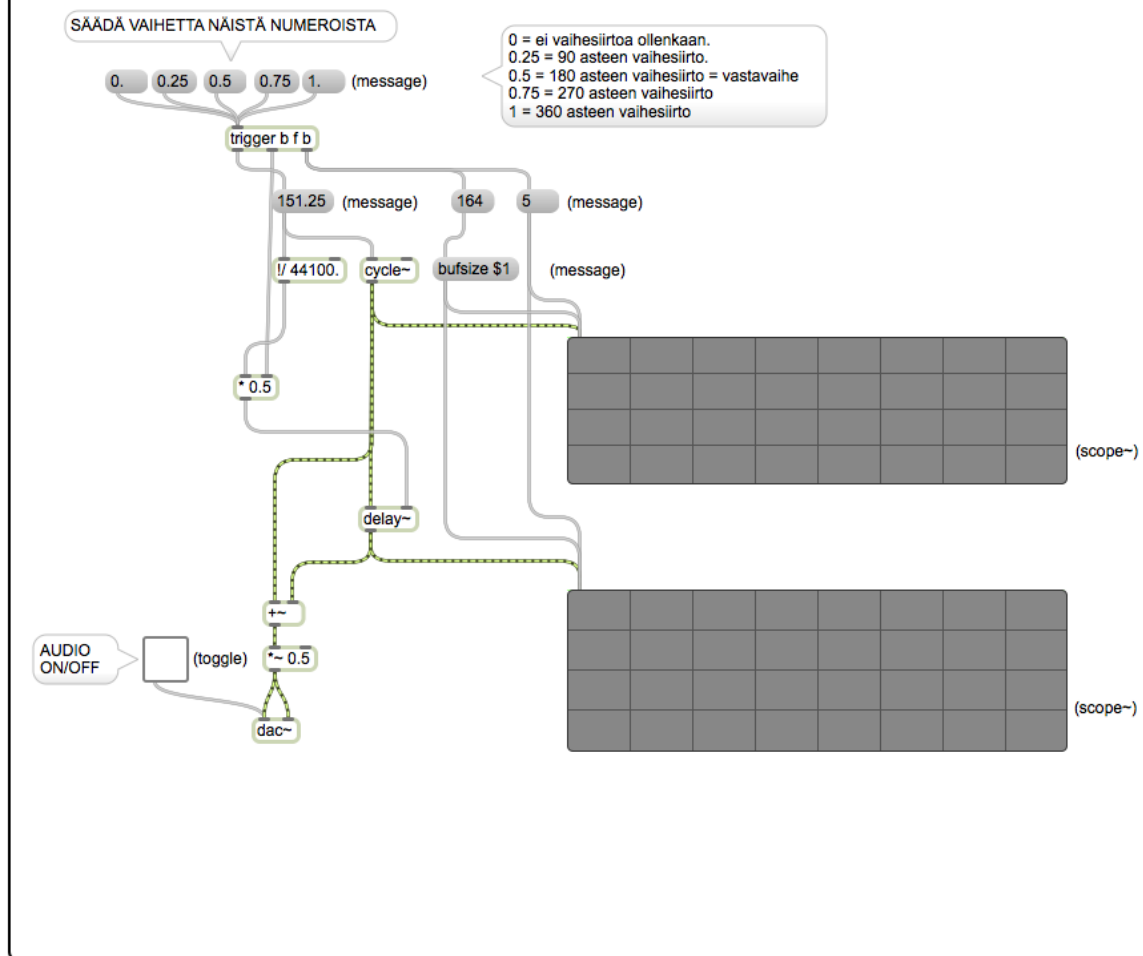
- verhokäyrä = envelope
- aluke
- ADSR = Attack - Decay - Sustain - Release

LISÄÄ AIHEESTA:

<http://www2.siba.fi/akustiikka/index.php?id=16&la=fi>
<http://www.digiwiki.fi/fi/index.php?title=Verhokäyrä>
<http://fi.wikipedia.org/wiki/ADSR>

7. VAIHE

Patch havainnollistaa vaihesiirron ja vastavaiheen.



HELP-TIEDOSTOT:

Objektin Help-tiedoston saa näkyviin klikkaamalla objektiin alt-näppäin pohjassa. Patchin tulee olla avattuna "unlocked" -moodissa.

INSPECTOR-IKKUNA:

Objektin Inspector -ikkunan saa näkyviin seuraavalla tavalla:

- 1) Aktivoi objekti, jonka Inspector -ikkuna halutaan avata.
- 2) Object -valikosta valitaan Inspector (Pikanäppäinkomento: cmd - i).

Saatuasi patchin valmiiksi, mieti miten voisit varioida ja kehittää patchiä eteenpäin. Toteuta ideasi!

KÄSITTEITÄ:

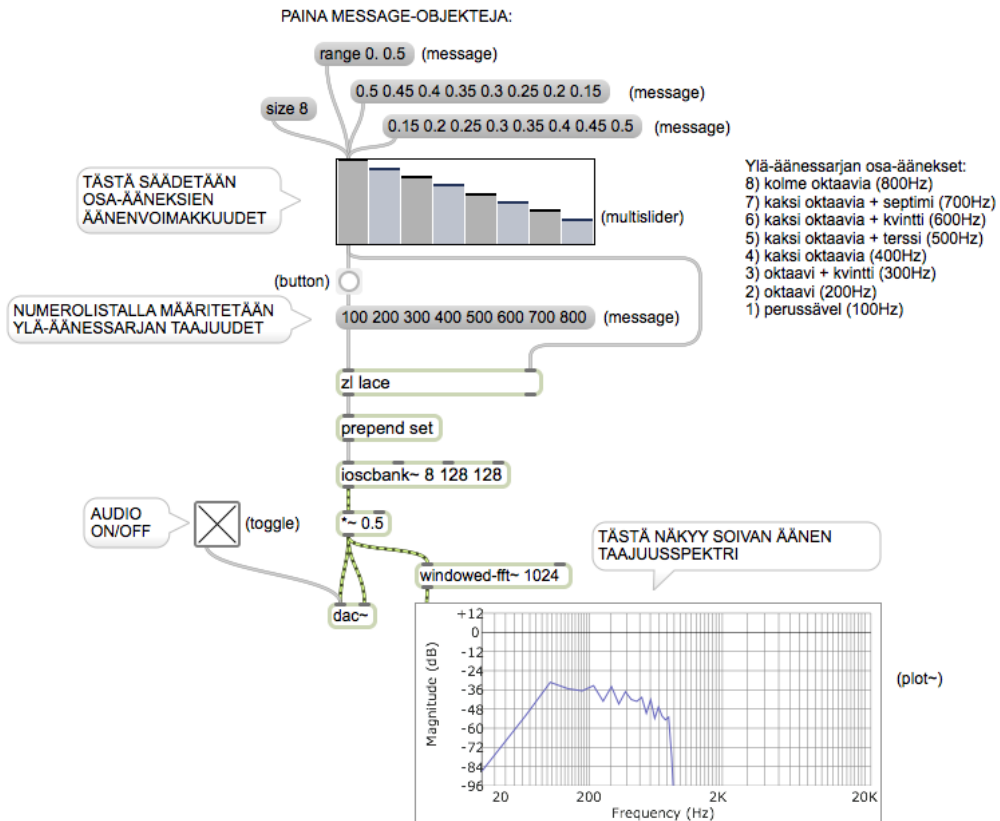
- vaihe
- vaihesiirto
- vastavaihe

LISÄÄ AIHEESTA:

<http://www.soundonsound.com/sos/apr08/articles/phasedemystified.htm>
<http://www.uaudio.com/blog/understanding-audio-phase/>
<http://www.themusicespionage.co.uk/phase-cancellation-explained/>
<http://fi.wikipedia.org/wiki/Vaihe>

8. YLÄ-ÄÄNESSARJA

Patchillä voidaan muokata ja kuunnella soivaa ylä-äänessarjaa.



HELP-TIEDOSTOT:

Objektin Help-tiedoston saa näkyviin klikkaamalla objektiin alt-näppäin pohjassa. Patchin tulee olla avattuna "unlocked" -moodissa.

INSPECTOR-IKKUNA:

Objektin Inspector -ikkunan saa näkyviin seuraavalla tavalla:

- 1) Aktivoi objekti, jonka Inspector -ikkuna halutaan avata.
- 2) Object -valikosta valitaan Inspector (Pikanäppäinkomento: cmd - i).

Saatuasi patchin valmiiksi, mieti miten voisit varioida ja kehittää patchiä eteenpäin. Toteuta ideasi!

KÄSITTEITÄ:

- osääänes
- osäääneistö
- harmoninen osäääneistö
- luonnonsävelsarja
- flageoletit (jousisoittimet)
- ylipuhaltaminen (puhaltimet)
- kokonaislukukerrannaiset

LISÄÄ AIHEESTA:

<http://fi.wikipedia.org/wiki/Yläsävelsarja>
<http://www2.siba.fi/intonaatio/index.php?id=51&la=fi>
<http://www2.siba.fi/akustiikka/index.php?id=14&la=fi>
http://opiskele.com/kurssit/musiikkifysiikka/html/01f_ylasavelsarja.shtml
<http://www2.siba.fi/akustiikka/index.php?id=15&la=fi>
<http://www15.uta.fi/arkisto/mustut/mute/aai04.htm>

9. ÄÄNENNOPEUS ja AALLONPITUUS

Patchissä on neljä eri laskuria, joilla voidaan laskea erilaisia aallonpituuteen, aikaan, taajuuteen ja äänennopeuteen liittyviä asioita.

LASKURI 1

Tällä laskurilla mitataan äänen kulkuaika tietyllä matkalla.

ETÄISYYS METREINÄ

0 (number)

/ 344. ÄÄNENNOPEUS

0. (flonum)

AIKA SEKUNNEISSA

* 1000.

0. (flonum)

AIKA MILLISEKUNNEISSA

LASKURI 2

Tällä laskurilla mitataan äänen tietyssä ajassa kulkema matka.

AIKA SEKUNNEISSA

0. (flonum)

* 344. ÄÄNENNOPEUS

0. (flonum)

ETÄISYYS METREINÄ

LASKURI 3

Tällä laskurilla mitataan äänisignaalin aallonpituus.

0. (flonum) TAAJUUS

/ 344.

0. (flonum) AALLONPITUUS METREINÄ

* 100.

0. (flonum) AALLONPITUUS SENTTIMETREINÄ

LASKURI 4

Tällä laskurilla mitataan määrätyn aallonpituisen signaalin taajuus.

0. (flonum) AALLONPITUUS METREINÄ

/ 344.

0. (flonum) TAAJUUS

HELP-TIEDOSTOT:

Objektin Help-tiedoston saa näkyviin klikkaamalla objektiin alt-näppäin pohjassa. Patchin tulee olla avattuna "unlocked" -moodissa.

INSPECTOR-IKKUNA:

Objektin Inspector-ikkunan saa näkyviin seuraavalla tavalla:

- 1) Aktivoi objekti, jonka Inspector-ikkuna halutaan avata.
- 2) Object-valikosta valitaan Inspector (Pikanäppäinkomento: cmd - i).

KÄSITTEITÄ:

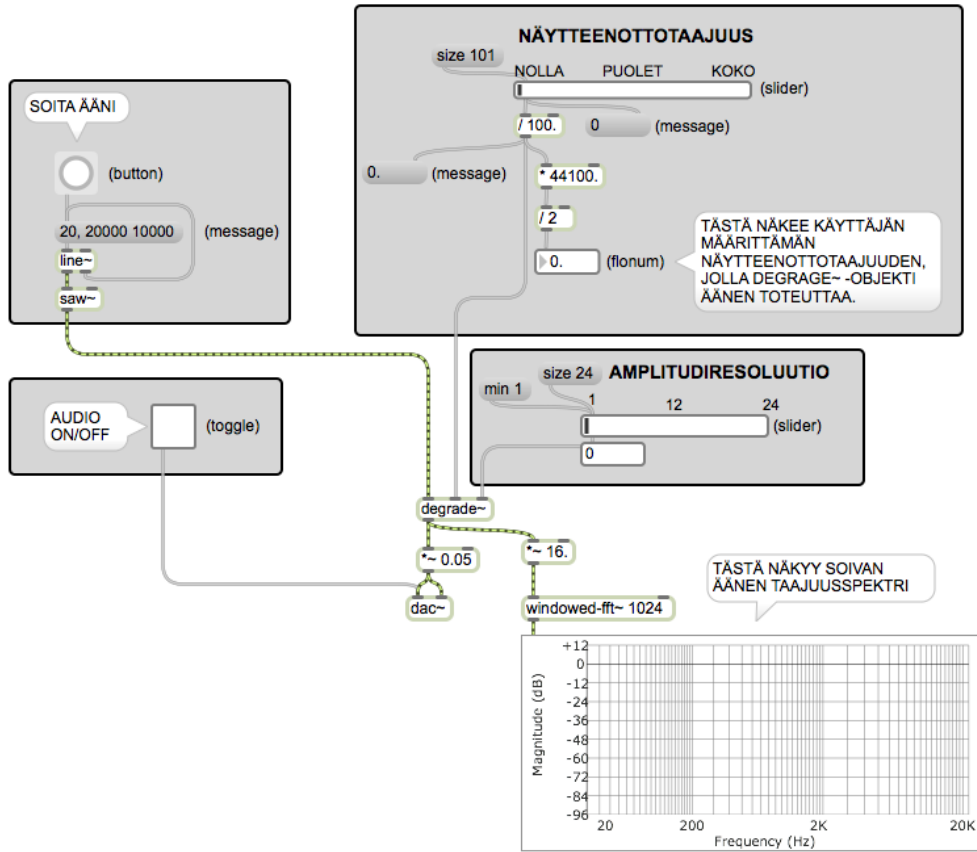
- äänennopeus
- äänennopeuden riippuvuus ilman lämpötilasta
- aallonpituus
- ultraääni
- infraääni
- kuuloalue

LISÄÄ AIHEESTA:

http://fi.wikipedia.org/wiki/Äänen_nopeus
<http://fi.wikipedia.org/wiki/Ultraääni>
<http://fi.wikipedia.org/wiki/Infraääni>
<http://fi.wikipedia.org/wiki/Kuuloaisti>

10. NÄYTTEENOTTOTAAJUUS JA AMPLITUDIRESOLUUTIO

Patchillä voidaan kokeilla näytteenottotaajuuden ja amplitudiresoluution muutoksien vaikutusta äänenlaatuun.



HELP-TIEDOSTOT:

Objektin Help-tiedoston saa näkyviin klikkaamalla objektiin alt-näppäin pohjassa. Patchin tulee olla avattuna "unlocked" -moodissa.

INSPECTOR-IKKUNA:

Objektin Inspector -ikkunan saa näkyviin seuraavalla tavalla:

- 1) Aktivoi objekti, jonka Inspector -ikkuna halutaan avata.
- 2) Object -valikosta valitaan Inspector (Pikanäppäinkomento: cmd - i).

Saatuasi patchin valmiiksi, mieti miten voisit varioida ja kehittää patchiä eteenpäin. Toteuta ideasi!

KÄSITTEITÄ:

- näytteenottotaajuus (sampling frequency)
- amplitudiresoluutio = bittiresoluutio = bittisyvyys
- Nyquistin teoreema
- laskostuminen (aliasing)
- laskostumisen poisto (anti-aliasing)
- taajuuskaista
- taajuusvaste (frequency response)
- sample (näyte)

LISÄÄ AIHEESTA:

<http://fi.wikipedia.org/wiki/Näytteenottotaajuus>
http://en.wikipedia.org/wiki/Audio_bit_depth

11. AALTOTAULUKKOSYNTESI (wavetable synthesis)

Patch esittelee aaltotaulukkosynteesin toimintaperiaatteen. Patchissä piirretään kuultava aaltomuoto.

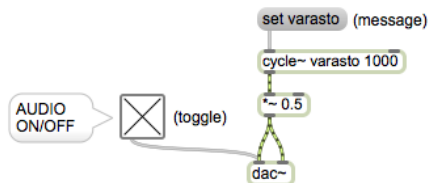
VAIHE 1)
Piirrä aaltomuoto.



VAIHE 2)
Kaksoisklikkaa buffer-objektiin, jotta voidaan tarkistaa, että piirretty aaltomuoto on tallentunut varasto -nimiseen puskuriin.

buffer~ varasto 11.6099

VAIHE 3)
Klikkaa "set varasto", jotta viimeksi piirretty aaltomuoto päivittyy kuunneltavaksi aaltomuodoksi. Laita audio päälle "AUDIO ON", jotta ääni alkaa kuulumaan.



VAIHE 4)
Aaltotaulukon (wavetable) lukuarvot on mahdollista tulostaa Max-ikkunaan klikkaamalla button-objektia. Tässä esimerkissä taulukko jakaantuu 512 indeksiin x-akselilla. Max-ikkuna saadaan näkyviin pikanäppäinkomennolla: cmd - m.



HELP-TIEDOSTOT:

Objektin Help-tiedoston saa näkyviin klikkaamalla objektiin alt-näppäin pohjassa. Patchin tulee olla avattuna "unlocked" -moodissa.

INSPECTOR-IKKUNA:

Objektin Inspector -ikkunan saa näkyviin seuraavalla tavalla:

- 1) Aktivoi objekti, jonka Inspector -ikkuna halutaan avata.
- 2) Object -valikosta valitaan Inspector (Pikanäppäinkomento: cmd - i).

Saatuasi patchin valmiiksi, mieti miten voisit varioida ja kehittää patchiä eteenpäin. Toteuta ideasi!

KÄSITTEITÄ:

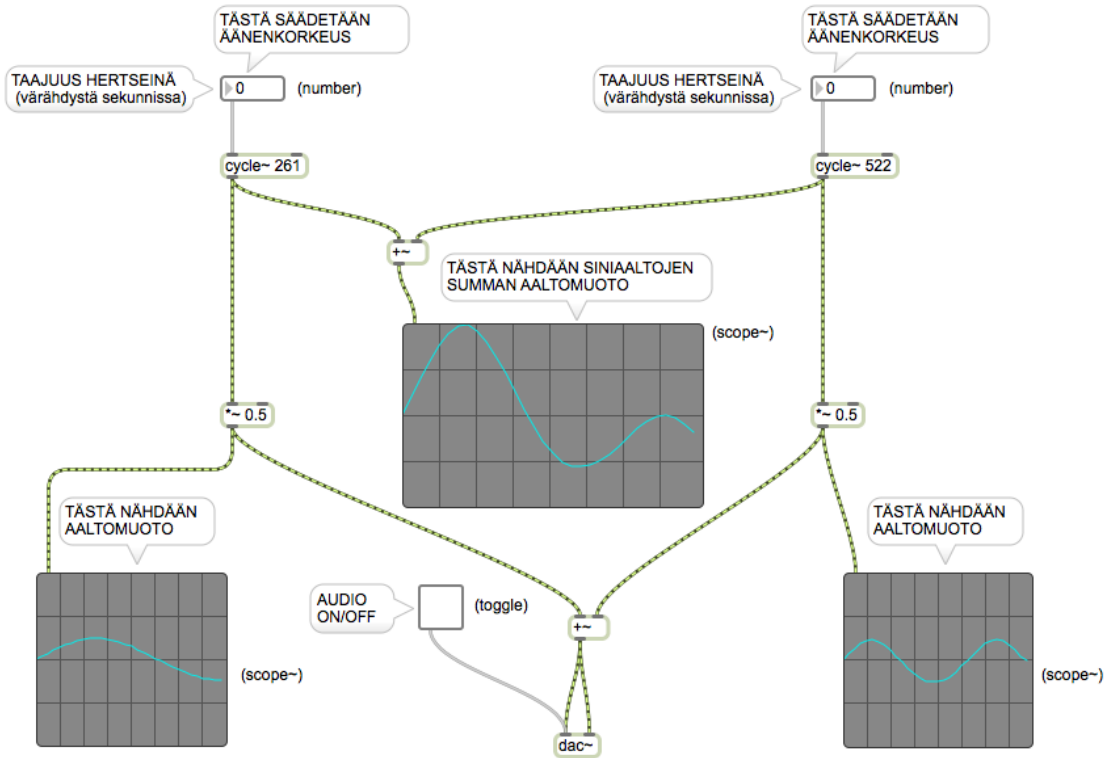
- aaltotaulukko (wavetable)
- näytteenottotaajuus (sampling frequency)
- sample (näyte)

LISÄÄ AIHEESTA:

- <http://computermusicresource.com/Definitions/wavetable.synthesis.html>
- <http://musicdsp.org/files/Wavetable-101.pdf>
- http://en.wikipedia.org/wiki/Wavetable_synthesis

12. LISÄÄVÄ SYNTEESI 1 (additive synthesis)

Lisäävässä synteesissä ääni toteutetaan summaamalla eritaajuisia siniääniä.



HELP-TIEDOSTOT:

Objektin Help-tiedoston saa näkyviin klikkaamalla objektiin alt-näppäin pohjassa. Patchin tulee olla avattuna "unlocked" -moodissa.

INSPECTOR-IKKUNA:

Objektin Inspector-ikkunan saa näkyviin seuraavalla tavalla:

- 1) Aktivoi objekti, jonka Inspector-ikkuna halutaan avata.
- 2) Object-valikosta valitaan Inspector (Pikanäppäinkomento: cmd - i).

Saatuasi patchin valmiiksi, mieti miten voisit varioida ja kehittää patchiä eteenpäin. Toteuta ideasi!

KÄSITTEITÄ:

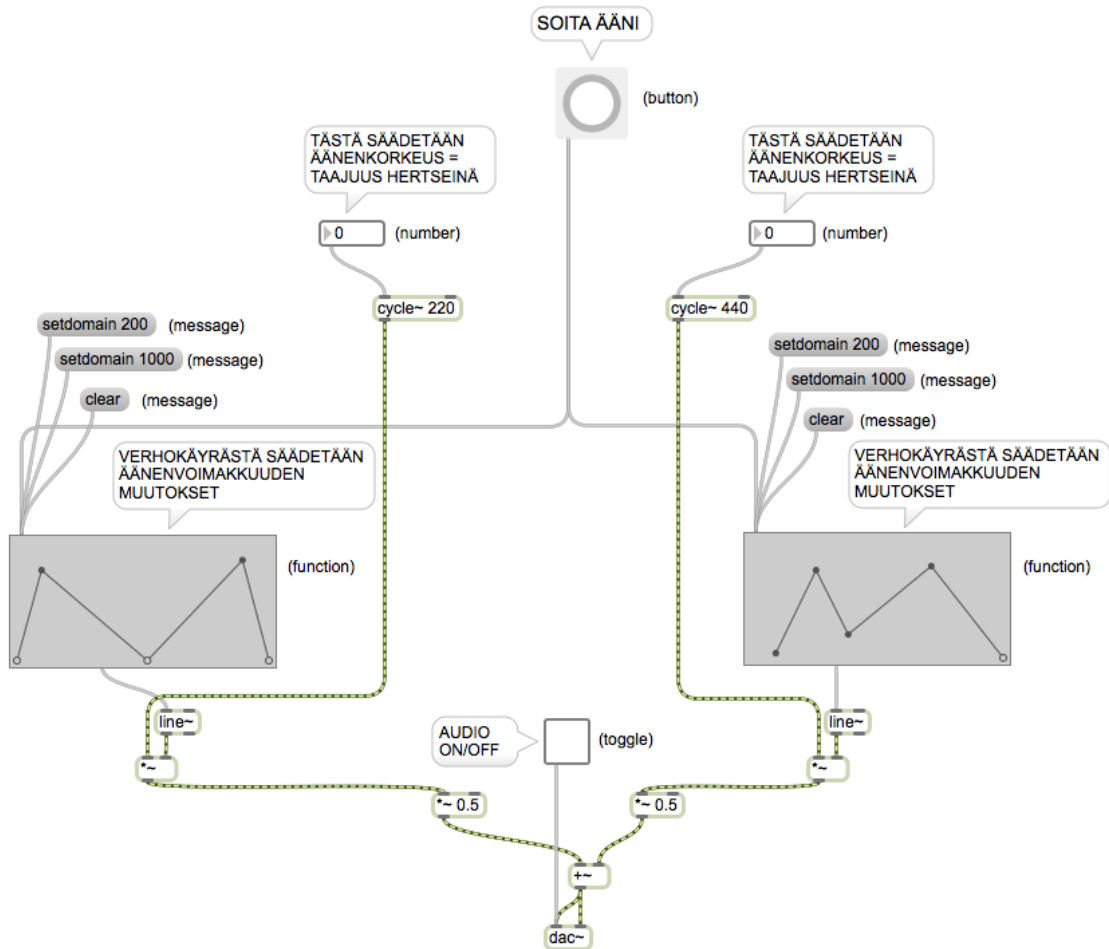
- osaaänekset
- taajuuksien summautuminen
- äänisynteesi
- oskillaattori
- VCO, Voltage Controlled Oscillator

LISÄÄ AIHEESTA:

http://en.wikibooks.org/wiki/Sound_Synthesis_Theory/Additive_Synthesis
http://en.wikipedia.org/wiki/Additive_synthesis
<http://www2.siba.fi/akustiikka/index.php?id=17&la=fi>

13. LISÄÄVÄ SYNTEESI 2 (additive synthesis)

Patchissä osaaänesten äänenvoimakkuutta ohjataan verhokäyrällä.



HELP-TIEDOSTOT:

Objektin Help-tiedoston saa näkyviin klikkaamalla objektiin alt-näppäin pohjassa. Patchin tulee olla avattuna "unlocked" -moodissa.

INSPECTOR-IKKUNA:

Objektin Inspector -ikkunan saa näkyviin seuraavalla tavalla:

- 1) Aktivoi objekti, jonka Inspector -ikkuna halutaan avata.
- 2) Object -valikosta valitaan Inspector (Pikanäppäinkomento: cmd - i).

Saatuasi patchin valmiiksi, mieti miten voisit varioida ja kehittää patchiä eteenpäin. Toteuta ideasi!

KÄSITTEITÄ:

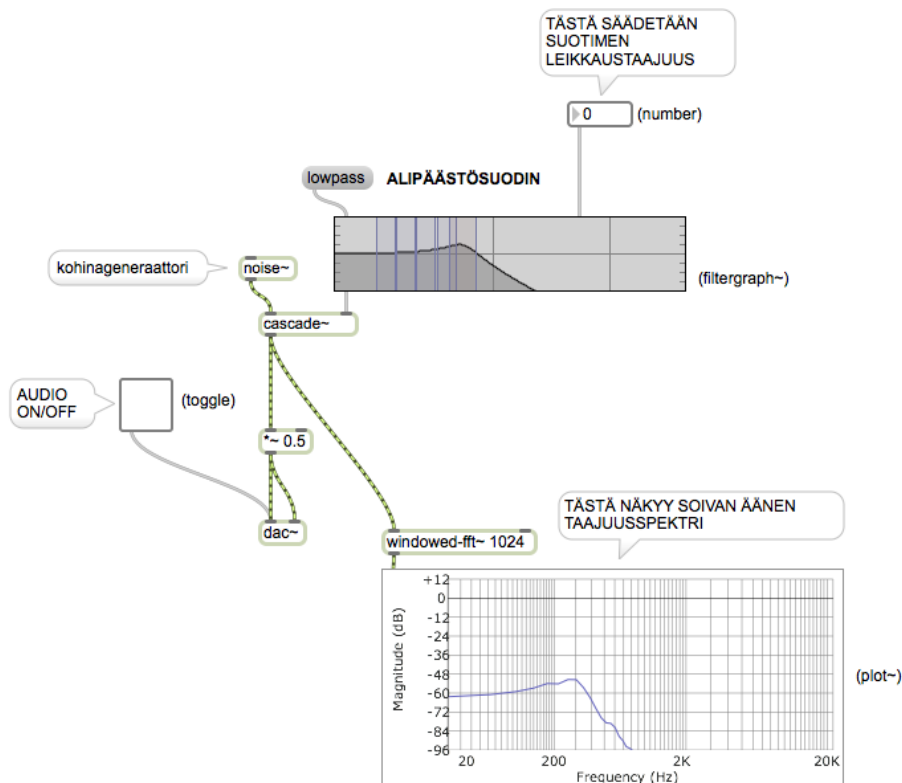
- parittomat osaaänekset
- parilliset osaaänekset
- Fourier -analyysi
- taajuusspektri

LISÄÄ AIHEESTA:

http://en.wikibooks.org/wiki/Sound_Synthesis_Theory/Additive_Synthesis
http://en.wikipedia.org/wiki/Additive_synthesis
<http://www2.siba.fi/akustiikka/index.php?id=17&la=fi>

14. VÄHENTÄVÄ SYNTEESI 1 (subtractive synthesis)

Vähentävässä synteesissä leikataan suotimilla tai ekvalisaattoreilla haluttu taajuuskaista pois.



HELP-TIEDOSTOT:

Objektin Help-tiedoston saa näkyviin klikkaamalla objektiin alt-näppäin pohjassa. Patchin tulee olla avattuna "unlocked" -moodissa.

INSPECTOR-IKKUNA:

Objektin Inspector-ikkunan saa näkyviin seuraavalla tavalla:
 1) Aktivoi objekti, jonka Inspector-ikkuna halutaan avata.
 2) Object-valikosta valitaan Inspector (Pikanäppäinkomento: cmd - i).

Saatuasi patchin valmiiksi, mieti miten voisit varioida ja kehittää patchiä eteenpäin. Toteuta ideasi!

KÄSITTEITÄ:

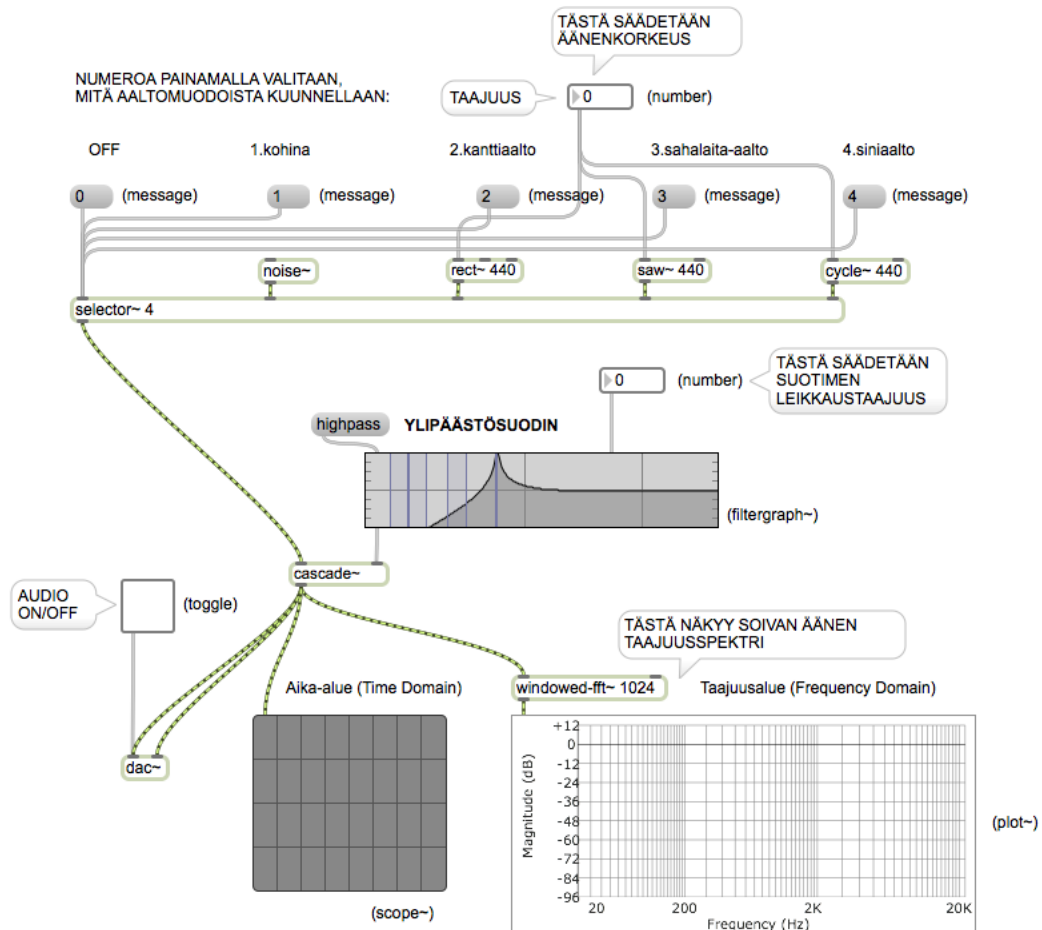
- suodin / filteri
- VCF, Voltage Controlled Filter
- leikkaustaajuus
- kompleksi ääni
- alipäästösuodatin
- ylipäästösuodatin
- kaistanpäästösuodatin
- kaistanestosuodatin
- taajuusspektri

LISÄÄ AIHEESTA:

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

15. VÄHENTÄVÄ SYNTEESI 2 (subtractive synthesis)

Säveltasollisten äänilähteiden spektreistä leikataan osa taajuuksista suotimilla pois.



HELP-TIEDOSTOT:

Objektin Help-tiedoston saa näkyviin klikkaamalla objektiin alt-näppäin pohjassa. Patchin tulee olla avattuna "unlocked" -moodissa.

INSPECTOR-IKKUNA:

Objektin Inspector-ikkunan saa näkyviin seuraavalla tavalla:

- 1) Aktivoi objekti, jonka Inspector-ikkuna halutaan avata.
- 2) Object-valikosta valitaan Inspector (Pikanäppäinkomento: cmd - i).

Saatuasi patchin valmiiksi, mieti miten voisit varioida ja kehittää patchiä eteenpäin. Toteuta ideasi!

KÄSITTEITÄ:

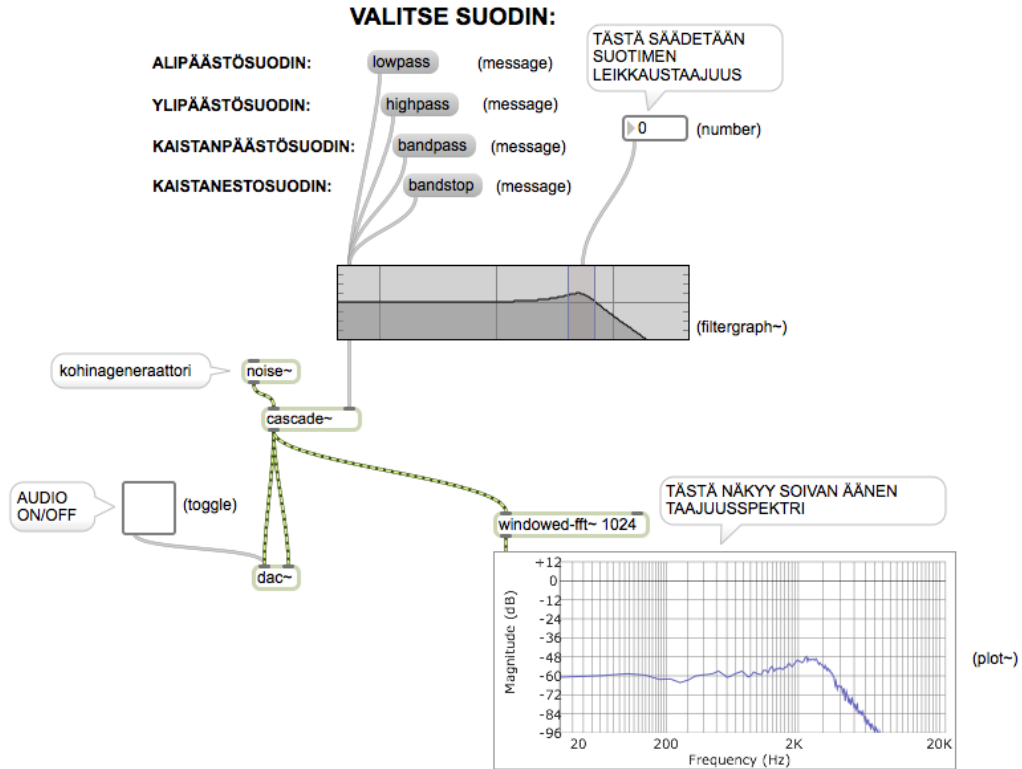
- suodin / filteri
- VCF, Voltage Controlled Filter
- leikkaustaajuus
- kompleksi ääni
- alipäästösuodatin
- ylipäästösuodatin
- kaistanpäästösuodatin
- kaistanestosuodatin
- spektri
- taajuuskaista

LISÄÄ AIHEESTA:

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

16. SUOTIMET

Patch esittelee erilaisia suotimia (alipäästö-, ylipäästö-, kaistanpäästö- ja kaistanestosuodin).



HELP-TIEDOSTOT:

Objektin Help-tiedoston saa näkyviin klikkaamalla objektiin alt-näppäin pohjassa. Patchin tulee olla avattuna "unlocked"-moodissa.

INSPECTOR-IKKUNA:

Objektin Inspector-ikkunan saa näkyviin seuraavalla tavalla:

- 1) Aktivoi objekti, jonka Inspector-ikkuna halutaan avata.
- 2) Object-valikosta valitaan Inspector (Pikanäppäinkomento: cmd - i).

Saatuasi patchin valmiiksi, mieti miten voisit varioida ja kehittää patchiä eteenpäin. Toteuta ideasi!

KÄSITTEITÄ:

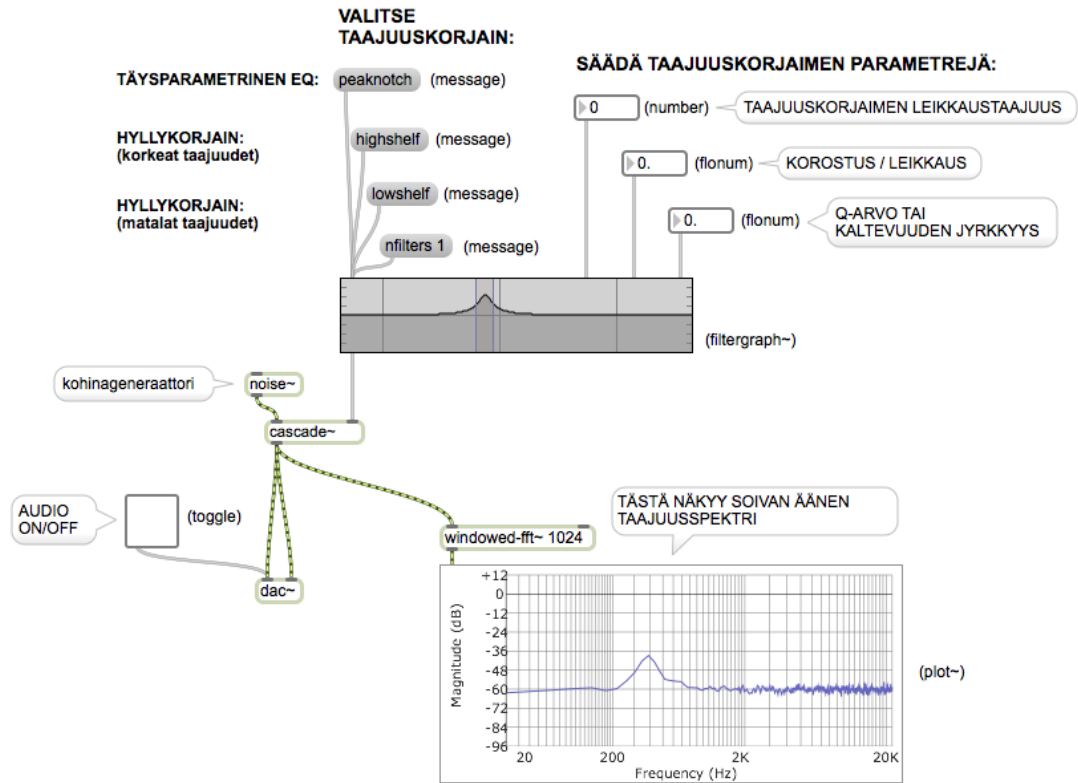
- alipäästösuodin (low pass filter, LPF)
- ylipäästösuodin (high pass filter, HPF)
- kaistanpäästösuodin (bandpass filter, BPF)
- kaistanestosuodin (bandstop filter, BSF)
- taajuusspektri
- Q-arvo

LISÄÄ AIHEESTA:

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

17. TAAJUUSKORJAIMET

Patch esittelee erilaisia taajuuskorjaimia (ekvalisaattori).



HELP-TIEDOSTOT:

Objektin Help-tiedoston saa näkyviin klikkaamalla objektiin alt-näppäin pohjassa. Patchin tulee olla avattuna "unlocked" -moodissa.

INSPECTOR-IKKUNA:

Objektin Inspector -ikkunan saa näkyviin seuraavalla tavalla:

- 1) Aktivoi objekti, jonka Inspector -ikkuna halutaan avata.
- 2) Object -valikosta valitaan Inspector (Pikanäppäinkomento: cmd - i).

Saatuasi patchin valmiiksi, mieti miten voisit varioida ja kehittää patchiä eteenpäin. Toteuta ideasi!

KÄSITTEITÄ:

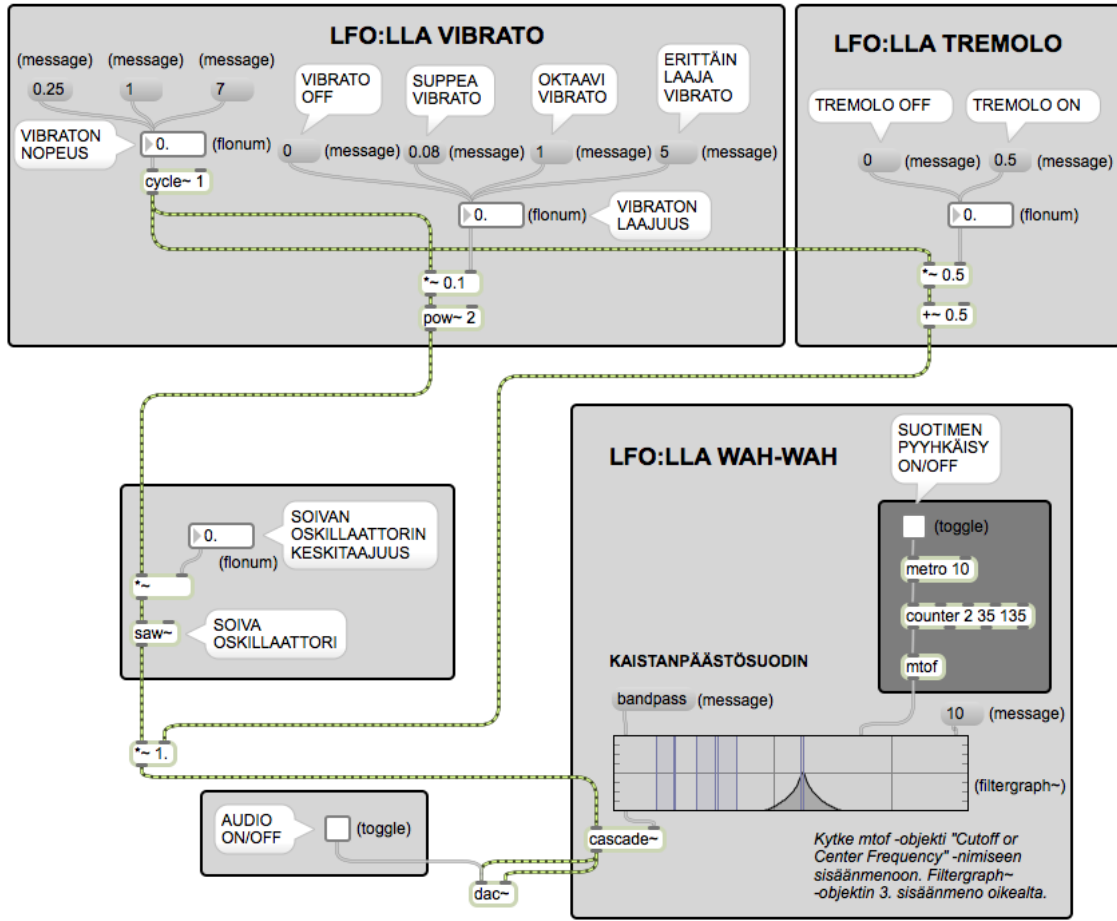
- ekvalisaattori, EQ
- täysparametrinen ekvalisaattori
- puoliparametrinen ekvalisaattori
- hyllykorjain
- korostus / vaimennus (gain)
- Q-arvo

LISÄÄ AIHEESTA:

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

18. PIENTAAJUUSOSKILLAATTORI, LFO = LOW FREQUENCY OSCILLATOR

Patch esittelee erilaisia asioita, joita LFO:lla voidaan toteuttaa.
Näitä ovat mm. vibrato, tremolo sekä wah-wah.



HELP-TIEDOSTOT:
Objektin Help-tiedoston saa näkyviin klikkaamalla objektiin alt-näppäin pohjassa. Patchin tulee olla avattuna "unlocked" -moodissa.

INSPECTOR-IKKUNA:
Objektin Inspector -ikkunan saa näkyviin seuraavalla tavalla:
1) Aktivoi objekti, jonka Inspector -ikkuna halutaan avata.
2) Object -valikosta valitaan Inspector (Pikanäppäinkomento: cmd - i).

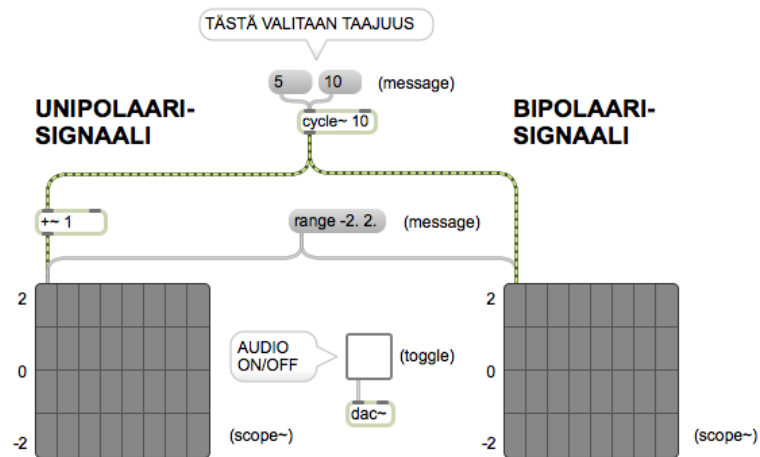
Saatuasi patchin valmiiksi, mieti miten voisit varioida ja kehittää patchiä eteenpäin. Toteuta ideasi!

KÄSITTEITÄ:
- taajuusalueet
- radiotaajuuudet
- audiotaaajuudet
- ultraääni

LINKKEJÄ:
http://electronicmusic.wikia.com/wiki/Low_frequency_oscillator
<http://news.beatport.com/essential-lfo-facts/>
<http://www.sweetwater.com/insync/low-frequency-oscillator-lfo/>

19. UNIPOLAARI- JA BIPOLAARISIGNAALI

Patch havainnollistaa visuaalisesti käsitteet unipolaari- ja bipolaarisignaali, ilman kuulohavaintoa.



HELP-TIEDOSTOT:

Objektin Help-tiedoston saa näkyviin klikkaamalla objektiin alt-näppäin pohjassa. Patchin tulee olla avattuna "unlocked"-moodissa.

INSPECTOR-IKKUNA:

Objektin Inspector -ikkunan saa näkyviin seuraavalla tavalla:

- 1) Aktivoi objekti, jonka Inspector -ikkuna halutaan avata.
- 2) Object -valikosta valitaan Inspector (Pikanäppäinkomento: cmd - i).

KÄSITTEITÄ:

- unipolaari
- bipolaari
- DC offset

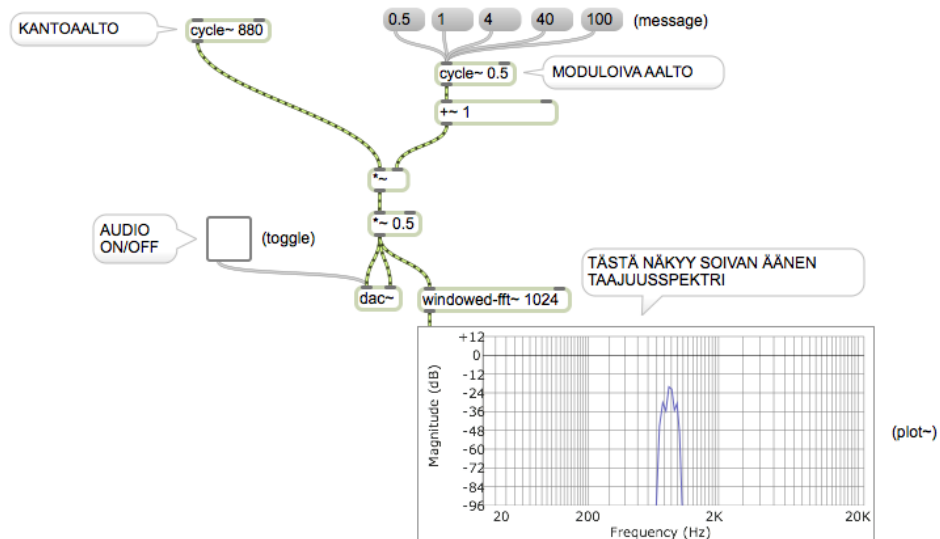
LISÄÄ AIHEESTA:

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

http://en.wikibooks.org/wiki/Sound_Synthesis_Theory/Modulation_Synthesis

20. AMPLITUDIMODULAATIO

Amplitudimodulaatiossa kanta-aallon amplitudia moduloidaan moduloivalla aaltomuodolla.
Amplitudimodulaatio toteutetaan kertomalla bipolaari audiosignaali unipolaarilla audiosignaalilla.



HELP-TIEDOSTOT:

Objektin Help-tiedoston saa näkyviin klikkaamalla objektiin alt-näppäin pohjassa. Patchin tulee olla avattuna "unlocked"-moodissa.

INSPECTOR-IKKUNA:

Objektin Inspector-ikkunan saa näkyviin seuraavalla tavalla:

- 1) Aktivoi objekti, jonka Inspector-ikkuna halutaan avata.
- 2) Object-valikosta valitaan Inspector (Pikanäppäinkomento: cmd - i).

Saatuasi patchin valmiiksi, mieti miten voisit varioida ja kehittää patchiä eteenpäin. Toteuta ideasi!

KÄSITTEITÄ:

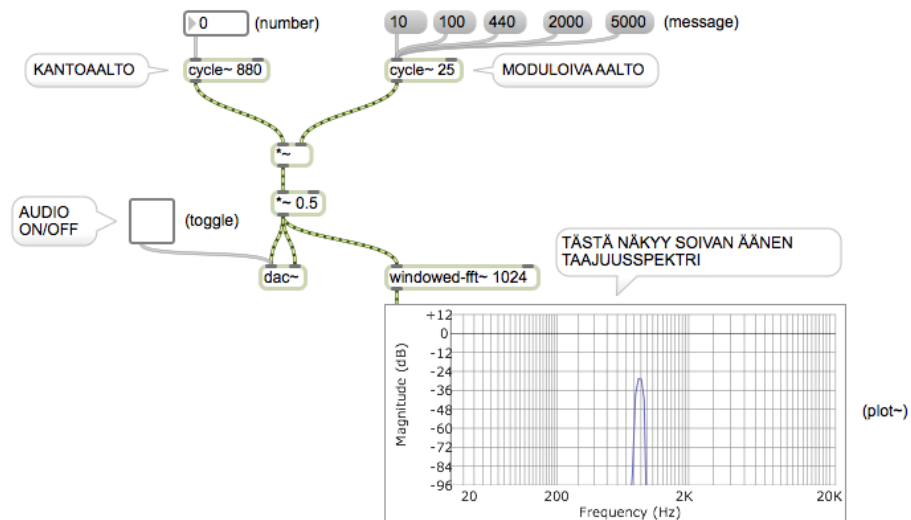
- kanta-aalto
- verhoikäyrä
- taajuusmodulaatio
- modulaatioindeksi

LISÄÄ AIHEESTA:

http://en.wikibooks.org/wiki/Sound_Synthesis_Theory/Modulation_Synthesis
<http://fi.wikipedia.org/wiki/Amplitudimodulaatio>

21. RENGASMODULAATIO

Rengasmodulaatio on amplitudimodulaation alalaji. Rengasmodulaatio toteutetaan kertomalla kaksi bipolaaria audiosignaalia toisiinsa.



HELP-TIEDOSTOT:

Objektin Help-tiedoston saa näkyviin klikkaamalla objektin alt-näppäin pohjassa. Patchin tulee olla avattuna "unlocked" -moodissa.

INSPECTOR-IKKUNA:

Objektin Inspector-ikkunan saa näkyviin seuraavalla tavalla:

- 1) Aktivoi objekti, jonka Inspector-ikkuna halutaan avata.
- 2) Object-valikosta valitaan Inspector (Pikanäppäinkomento: cmd - i).

Saatuasi patchin valmiiksi, mieti miten voisit varioida ja kehittää patchiä eteenpäin. Toteuta ideasi!

KÄSITTEITÄ:

- kantaalto
- moduloiva aalto
- modulointi
- taajuusmodulaatio
- amplitudimodulaatio

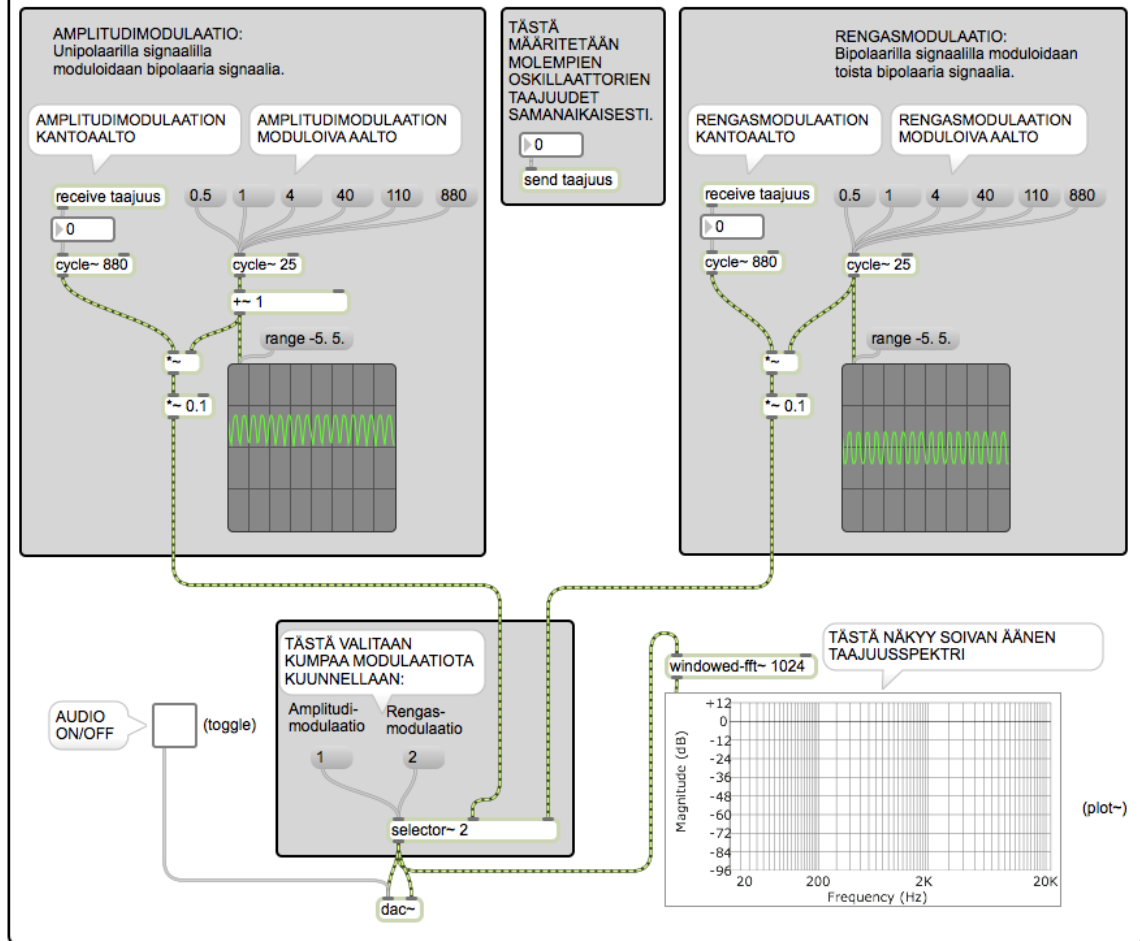
LISÄÄ AIHEESTA:

http://en.wikibooks.org/wiki/Sound_Synthesis_Theory/Modulation_Synthesis

<http://fi.wikipedia.org/wiki/Amplitudimodulaatio>

22. AMPLITUDI- JA RENGASMODULAATIOIDEN VERTAILU

Patchillä voidaan vertailla rengas- ja amplitudimodulaatioiden yhtäläisyyksiä ja eroavaisuuksia.



HELP-TIEDOSTOT:

Objektin Help-tiedoston saa näkyviin klikkaamalla objektiin alt-näppäin pohjassa. Patchin tulee olla avattuna "unlocked"-moodissa.

INSPECTOR-IKKUNA:

Objektin Inspector-ikkunan saa näkyviin seuraavalla tavalla:
1) Aktivoi objekti, jonka Inspector-ikkuna halutaan avata.
2) Object-valikosta valitaan Inspector (Pikanäppäinkomento: cmd - i).

Saatuasi patchin valmiiksi, mieti miten voisit varioida ja kehittää patchiä eteenpäin. Toteuta ideasi!

KÄSITTEITÄ:

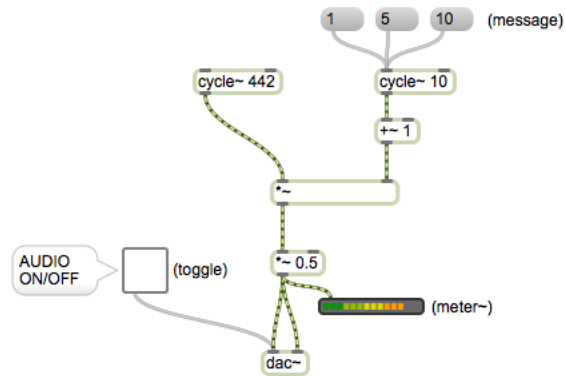
- taajuusmodulaatio
- amplitudimodulaatio
- amplitudimodulaatio radiotekniikassa

LISÄÄ AIHEESTA:

http://en.wikibooks.org/wiki/Sound_Synthesis_Theory/Modulation_Synthesis
<http://fi.wikipedia.org/wiki/Amplitudimodulaatio>

23. TREMOLO

Patch toteuttaa tremolo-efektin moduloimalla signaalin amplitudia.



HELP-TIEDOSTOT:

Objektin Help-tiedoston saa näkyviin klikkaamalla objektiin alt-näppäin pohjassa. Patchin tulee olla avattuna "unlocked" -moodissa.

INSPECTOR-IKKUNA:

Objektin Inspector -ikkunan saa näkyviin seuraavalla tavalla:

- 1) Aktivoi objekti, jonka Inspector -ikkuna halutaan avata.
- 2) Object -valikosta valitaan Inspector (Pikanäppäinkomento: cmd - i).

Saatuasi patchin valmiiksi, mieti miten voisit varioida ja kehittää patchiä eteenpäin. Toteuta ideasi!

KÄSITTEITÄ:

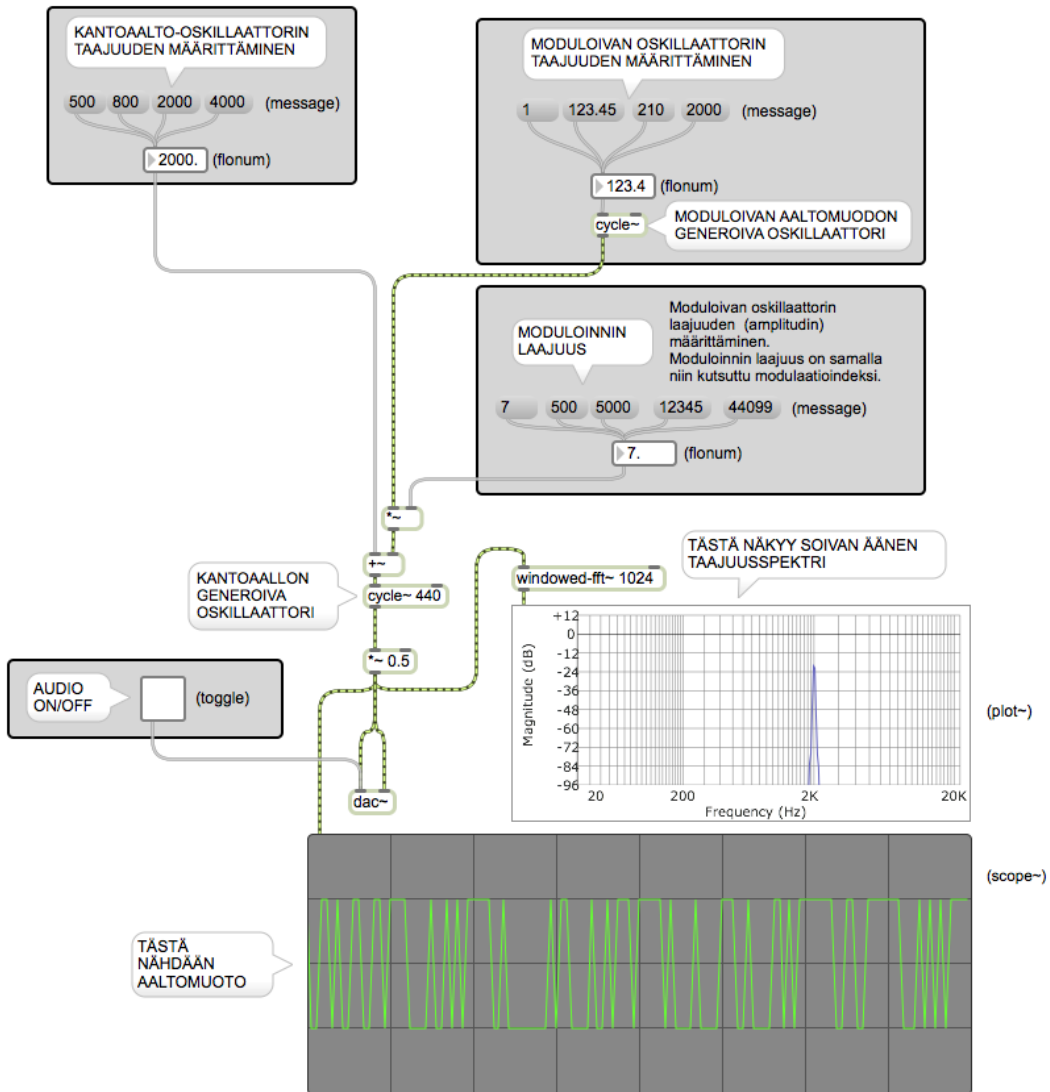
- amplitudimodulaatio
- amplitudimodulaatio radiotekniikassa
- taajuusmodulaatio

LISÄÄ AIHEESTA:

<http://www.premiarguitar.com/articles/19777-a-brief-history-of-tremolo>
<http://fi.wikipedia.org/wiki/Tremolo>
http://www.sweetwater.com/c973-Tremolo_Pedals

24. FM-SYNTESI

FM-synteesissä kantaaallon taajuutta moduloidaan toisen signaalin amplitudilla.



HELP-TIEDOSTOT:
Objektin Help-tiedoston saa näkyviin klikkaamalla objektiin alt-näppäin pohjassa. Patchin tulee olla avattuna "unlocked" -moodissa.

INSPECTOR-IKKUNA:
Objektin Inspector -ikkunan saa näkyviin seuraavalla tavalla:
1) Aktivoi objekti, jonka Inspector -ikkuna halutaan avata.
2) Object -valikosta valitaan Inspector (Pikanäppäinkomento: cmd - i).

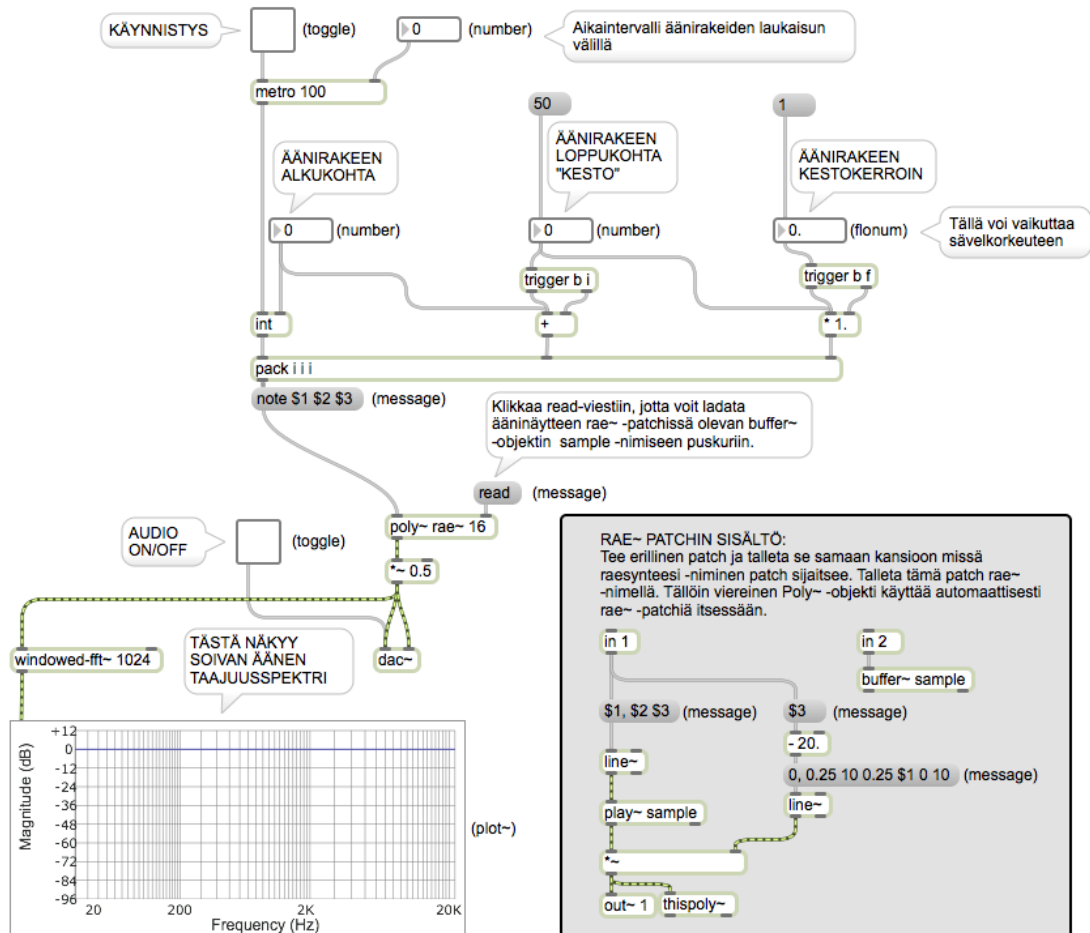
Saatuasi patchin valmiiksi, mieti miten voisit varioida ja kehittää patchiä eteenpäin. Toteuta ideasi!

KÄSITTEITÄ:
- vaihemodulaatio
- amplitudimodulaatio
- taajuusspektri

LISÄÄ AIHEESTA:
<http://www.sfu.ca/~truax/fmtut.html>
<http://www.chipple.net/dx7/english/fm.tone.generation.html>
http://music.columbia.edu/cmc/musicandcomputers/chapter4/04_07.php
http://en.wikipedia.org/wiki/Frequency_modulation_synthesis

25. RAESYNTESI

Raesynteessissä pidemmästä ääninäytteestä poimitaan erittäin lyhyitä ääninäytteitä, joita kutsutaan äänirakeiksi. Näistä äänirakeista muodostetaan eri tiheysisiä ja eri paksuisia äänikenttiä.



HELP-TIEDOSTOT:

Objektin Help-tiedoston saa näkyviin klikkaamalla objektiin alt-näppäin pohjassa. Patchin tulee olla avattuna "unlocked"-moodissa.

INSPECTOR-IKKUNA:

Objektin Inspector -ikkunan saa näkyviin seuraavalla tavalla:
 1) Aktivoi objekti, jonka Inspector -ikkuna halutaan avata.
 2) Object -valikosta valitaan Inspector (Pikanäppäinkomento: cmd - i).

Saatua patchin valmiiksi, mieti miten voisit varioida ja kehittää patchiä eteenpäin. Toteuta ideasi!

KÄSITTEITÄ:

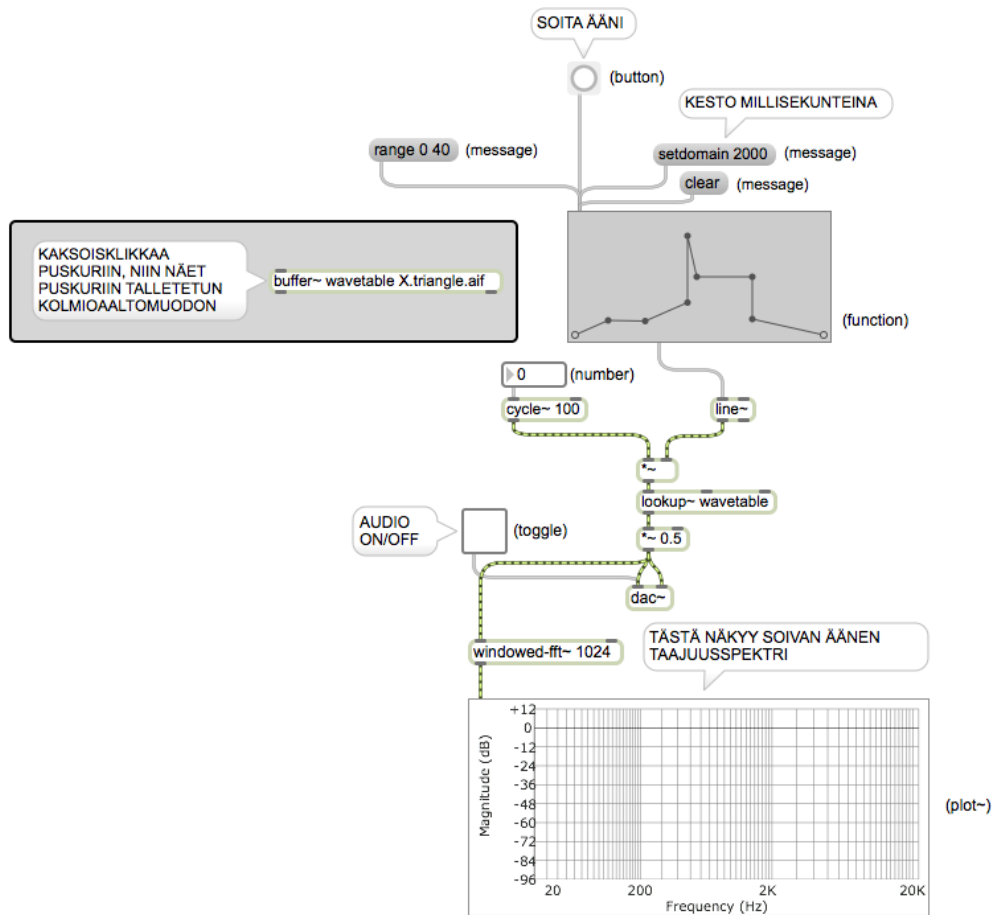
- äänirae (sonic grain)
 - äänirakeen muotoilu verhoikäyrällä

LISÄÄ AIHEESTA:

<http://www.granularsynthesis.com/guide.php>
<http://www.sfu.ca/~truax/gran.html>
http://en.wikipedia.org/wiki/Granular_synthesis
<http://www.sfu.ca/sonic-studio/handbook/Grain.html#gs>

26. AALTOMUOTOILUSYNTEESI (waveshaping)

Patch esittelee aaltomuotoilusynteesin toimintaperiaatteen.



HELP-TIEDOSTOT:

Objektin Help-tiedoston saa näkyviin klikkaamalla objektiin alt-näppäin pohjassa. Patchin tulee olla avattuna "unlocked" -moodissa.

INSPECTOR-IKKUNA:

Objektin Inspector -ikkunan saa näkyviin seuraavalla tavalla:

- 1) Aktivoi objekti, jonka Inspector -ikkuna halutaan avata.
- 2) Object -valikosta valitaan Inspector (Pikanäppäinkomento: cmd - i).

Saatuasi patchin valmiiksi, mieti miten voisit varioida ja kehittää patchiä eteenpäin. Toteuta ideasi!

KÄSITTEITÄ:

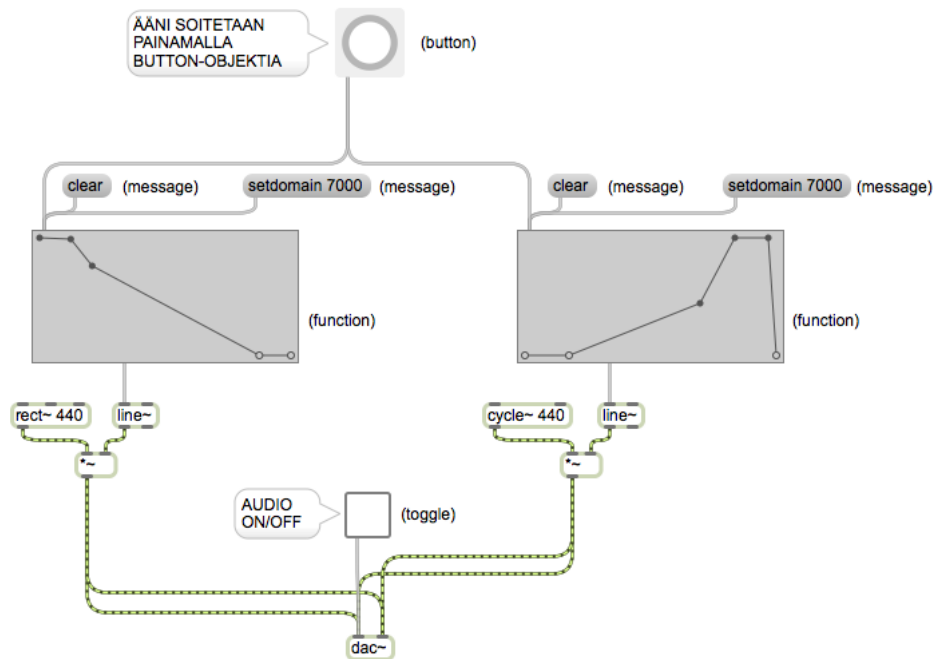
- siirtofunktio = transfer function
- hakutaulukko = look up table
- aaltotaulukko

LISÄÄ AIHEESTA:

http://www.cs.sfu.ca/~tamaras/waveshapeSynth/Waveshaping_Synthesis.html
http://en.wikipedia.org/wiki/Distortion_synthesis
<http://computermusicresource.com/Definitions/waveshaping.synthesis.html>

27. KAHDEN AALTOTAULUKON RISTIINFEIDAUS (WAVETABLE CROSSFADING)

"Kahden aaltotaulukon ristiinfeidaus" -patchillä havainnollistetaan ääni, joka on toteutettu kahden eri ääneksen sulauttamisella.



HELP-TIEDOSTOT:

Objektin Help-tiedoston saa näkyviin klikkaamalla objektiin alt-näppäin pohjassa. Patchin tulee olla avattuna "unlocked" -moodissa.

INSPECTOR-IKKUNA:

Objektin Inspector -ikkunan saa näkyviin seuraavalla tavalla:

- 1) Aktivoi objekti, jonka Inspector -ikkuna halutaan avata.
- 2) Object -valikosta valitaan Inspector (Pikanäppäinkomento: cmd - i).

Saatuasi patchin valmiiksi, mieti miten voisit varioida ja kehittää patchiä eteenpäin. Toteuta ideasi!

KÄSITTEITÄ:

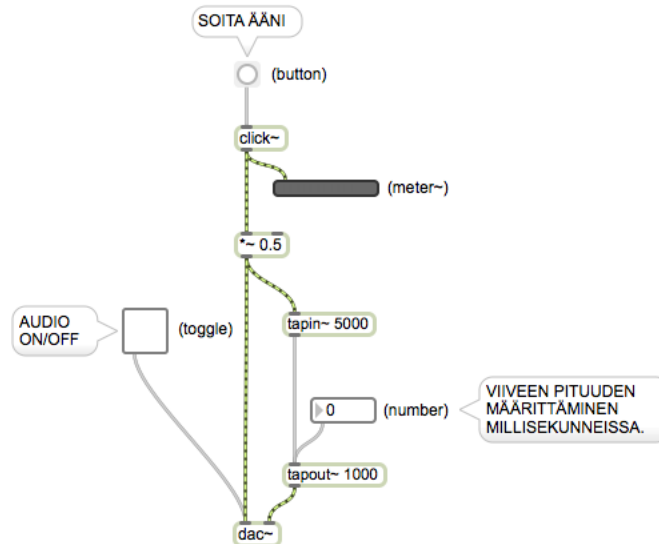
- aaltotaulukko = wavetable
- hakutaulukko
- ristiinvaihto = crossfade

LISÄÄ AIHEESTA:

http://dspwiki.com/index.php?title=Other_Synthesis_Types
<http://musicdsp.org/files/Wavetable-101.pdf>
<http://www.experts123.com/q/what-is-wavetable-crossfading.html>

28. VIIVE 1

Signaali toistetaan hetkeä myöhemmin viivästettynä. Viiveen pituus määritetään millisekunteina.



HELP-TIEDOSTOT:

Objektin Help-tiedoston saa näkyviin klikkaamalla objektiin alt-näppäin pohjassa. Patchin tulee olla avattuna "unlocked" -moodissa.

INSPECTOR-IKKUNA:

Objektin Inspector -ikkunan saa näkyviin seuraavalla tavalla:

- 1) Aktivoi objekti, jonka Inspector -ikkuna halutaan avata.
- 2) Object -valikosta valitaan Inspector (Pikanäppäinkomento: cmd - i).

Saatuasi patchin valmiiksi, mieti miten voisit varioida ja kehittää patchiä eteenpäin. Toteuta ideasi!

KÄSITTEITÄ:

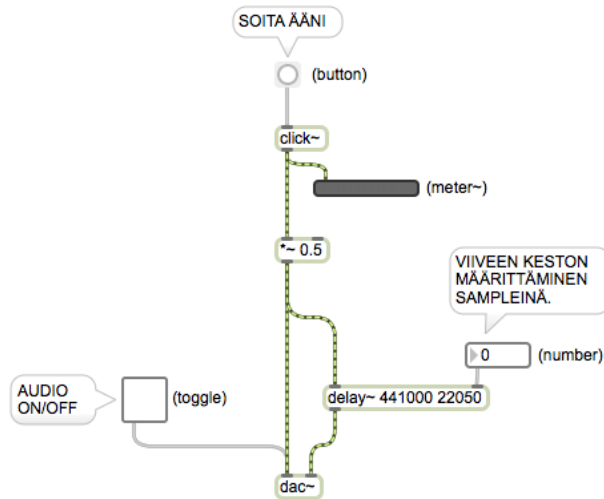
- viive = delay
- viive audioelektronikassa
- TIM-särö

LISÄÄ AIHEESTA:

http://www.soundonsound.com/sos/jul08/articles/qa0708_3.htm
<http://www.soundonsound.com/sos/feb09/articles/deepspace.htm>
http://www.sweetwater.com/cs--Delay_Pedals
[http://en.wikipedia.org/wiki/Delay_\(audio_effect\)](http://en.wikipedia.org/wiki/Delay_(audio_effect))

29. VIIVE 2

Signaali toistetaan hetkeä myöhemmin viivästettynä. Viiveen pituus määritetään sampleinä.



HELP-TIEDOSTOT:

Objektin Help-tiedoston saa näkyviin klikkaamalla objektiin alt-näppäin pohjassa. Patchin tulee olla avattuina "unlocked" -moodissa.

INSPECTOR-IKKUNA:

Objektin Inspector -ikkunan saa näkyviin seuraavalla tavalla:

- 1) Aktivoi objekti, jonka Inspector -ikkuna halutaan avata.
- 2) Object -valikosta valitaan Inspector (Pikanäppäinkomento: cmd - i).

Saatuasi patchin valmiiksi, mieti miten voisit varioida ja kehittää patchiä eteenpäin. Toteuta ideasi!

KÄSITTEITÄ:

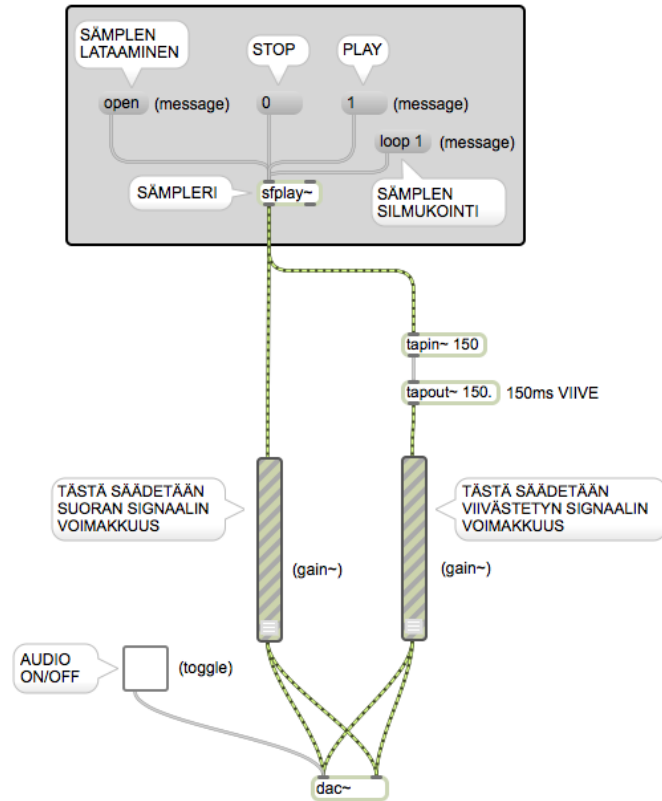
- viive = delay
- sample
- analogisia viivelaitteita: (nauha- ja levyviiveet)
- viive audioelektronikassa
- TIM-särö

LISÄÄ AIHEESTA:

http://www.soundonsound.com/sos/jul08/articles/qa0708_3.htm
<http://www.soundonsound.com/sos/feb09/articles/deepspace.htm>
http://www.sweetwater.com/cs--Delay_Pedals
[http://en.wikipedia.org/wiki/Delay_\(audio_effect\)](http://en.wikipedia.org/wiki/Delay_(audio_effect))

30. LÄPSÄHDYSVIIVE

Patchillä toteutetaan äänilähteen läpsähdysviive.



HELP-TIEDOSTOT:

Objektin Help-tiedoston saa näkyviin klikkaamalla objektiin alt-näppäin pohjassa. Patchin tulee olla avattuna "unlocked" -moodissa.

INSPECTOR-IKKUNA:

Objektin Inspector -ikkunan saa näkyviin seuraavalla tavalla:

- 1) Aktivoi objekti, jonka Inspector -ikkuna halutaan avata.
- 2) Object -valikosta valitaan Inspector (Pikanäppäinkomento: cmd - i).

Saatuasi patchin valmiiksi, mieti miten voisit varioida ja kehittää patchiä eteenpäin. Toteuta ideasi!

KÄSITTEITÄ:

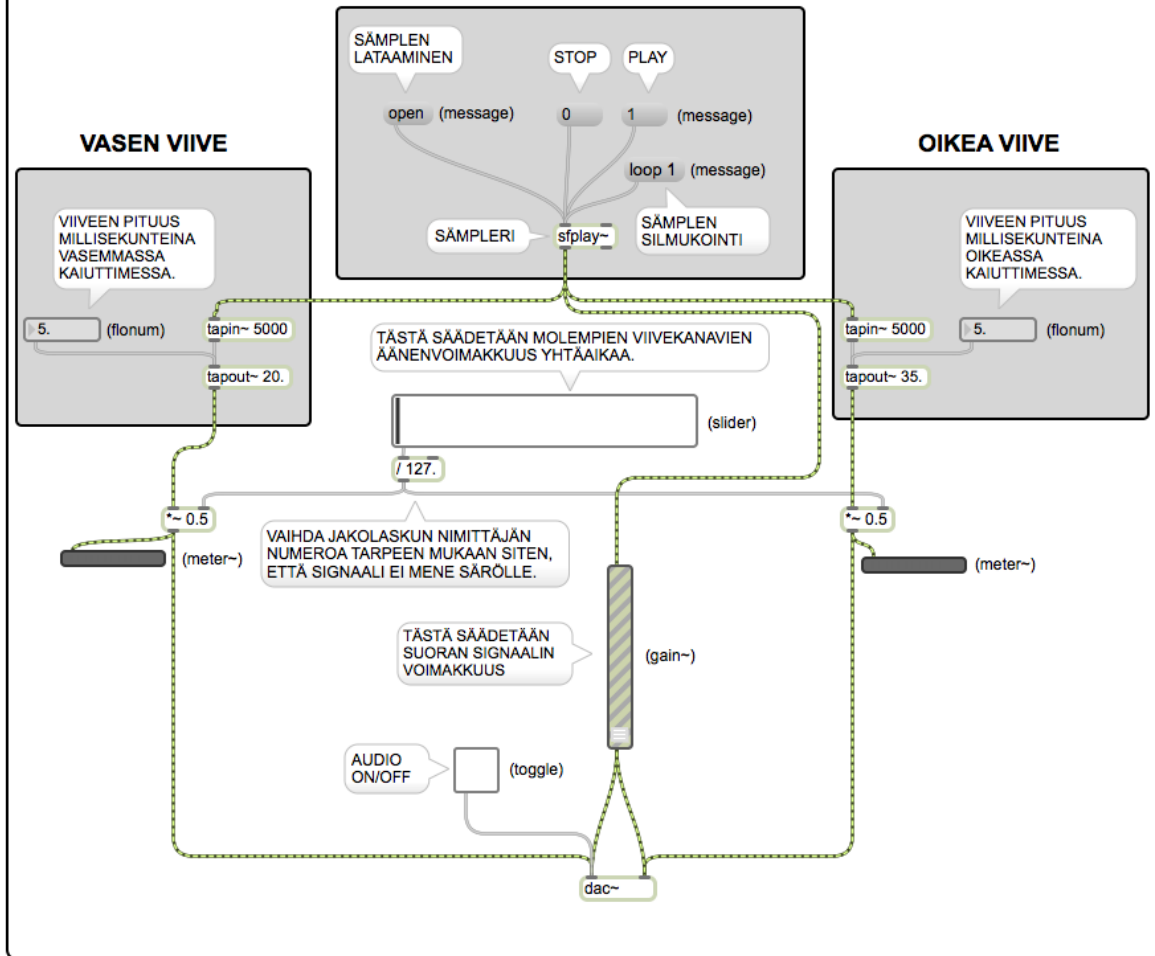
- läpsähdysviive (slapback delay)
- läpsähdysviiveen pituus: 60-180 millisekuntia
- analogisia viivelaiteita: (nauha- ja levyviiveet)

LISÄÄ AIHEESTA:

- <http://www.soundonsound.com/sos/may12/articles/designer-delay.htm>
- <http://www.sweetwater.com/insync/slapback-delay/>
- [http://en.wikipedia.org/wiki/Delay_\(audio_effect\)](http://en.wikipedia.org/wiki/Delay_(audio_effect))

31. VIIVEILLÄ TOTEUTETTU PIENI TILA

Patchillä toteutetaan pieni akustinen tila kahdella eripituisella viiveellä monoäänilähteen avulla.



HELP-TIEDOSTOT:

Objektin Help-tiedoston saa näkyviin klikkaamalla objektiin alt-näppäin pohjassa. Patchin tulee olla avattuna "unlocked" -moodissa.

INSPECTOR-IKKUNA:

Objektin Inspector -ikkunan saa näkyviin seuraavalla tavalla:

- 1) Aktivoi objekti, jonka Inspector -ikkuna halutaan avata.
- 2) Object -valikosta valitaan Inspector (Pikanäppäinkomento: cmd - i).

Saatuasi patchin valmiiksi, mieti miten voisit varioida ja kehittää patchiä eteenpäin. Toteuta ideasi!

KÄSITTEITÄ:

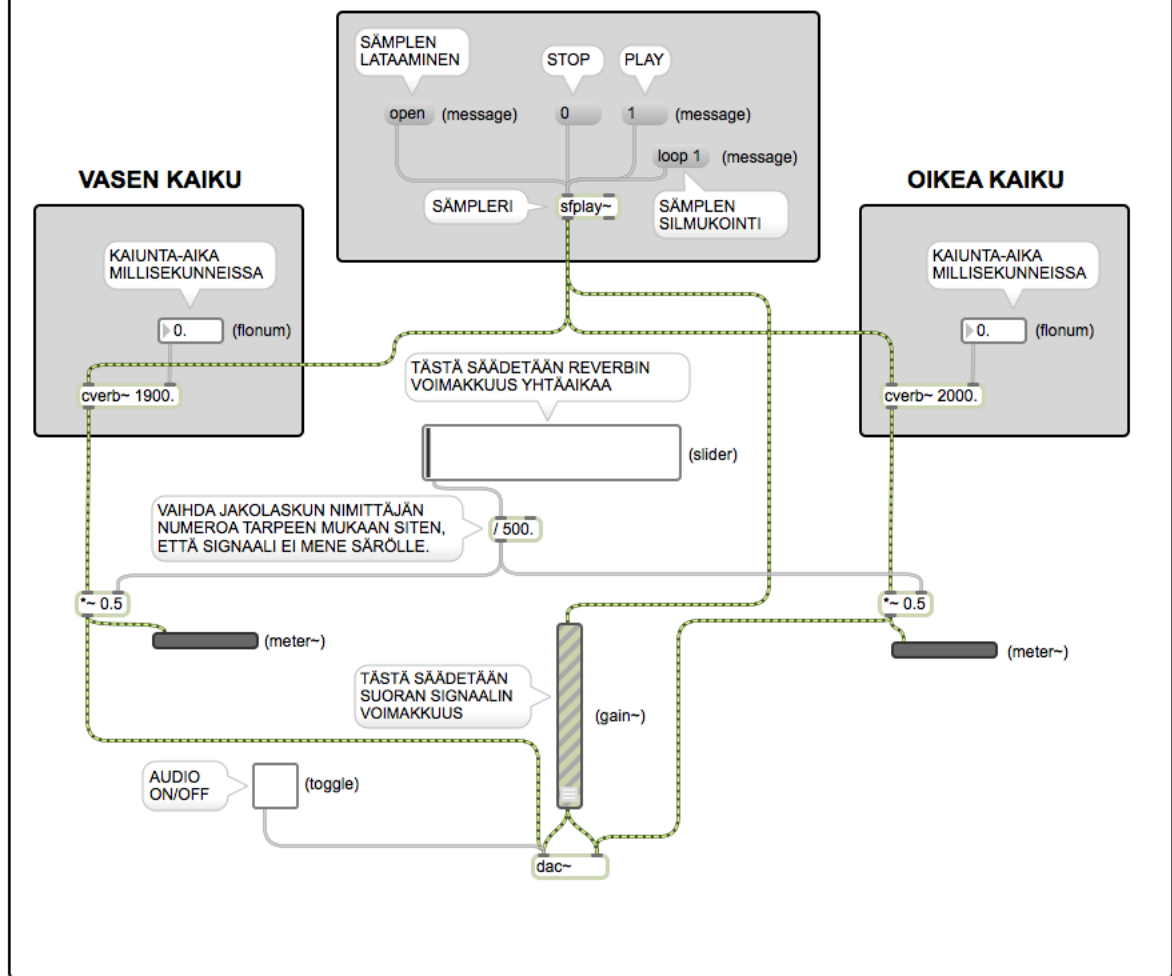
- Haas-/presedenssiefekti
- akustiikka
- kaiku = reverb

LISÄÄ AIHEESTA:

<http://www.soundonsound.com/sos/may12/articles/designer-delay.htm>
http://www.soundonsound.com/sos/oct07/articles/cubasetechnique_1007.htm
[http://en.wikipedia.org/wiki/Delay_\(audio_effect\)](http://en.wikipedia.org/wiki/Delay_(audio_effect))
<http://www.waves.com/reverb-and-delay-tutorial>

32. KAIKU

Patchillä toteutetaan äänilähteeseen kaiku.



HELP-TIEDOSTOT:

Objektin Help-tiedoston saa näkyviin klikkaamalla objektiin alt-näppäin pohjassa. Patchin tulee olla avattuna "unlocked" -moodissa.

INSPECTOR-IKKUNA:

Objektin Inspector-ikkunan saa näkyviin seuraavalla tavalla:

- 1) Aktivoi objekti, jonka Inspector-ikkuna halutaan avata.
- 2) Object-valikosta valitaan Inspector (Pikanäppäinkomento: cmd - i).

Saatuasi patchin valmiiksi, mieti miten voisit varioida ja kehittää patchiä eteenpäin. Toteuta ideasi!

KÄSITTEITÄ:

- kaiku (reverb)
- analogiset kaikulaitteet: kaikuhuone, kaikulevy (plate) ja jousikaiku (spring)
- digitaaliset kaikulaitteet
- kaikutyyppiä: Hall, Chamber, Room, Plate, Spring

- kaikulaitteiden parametrejä:

- kaikuaika (reverberation time), ensimmäiset heijasteet (early reflections), ennakkoviive (pre-delay), kaiun tiheys (diffusion), kaiunnan diskanttivaimennus (roll-off), suoran äänen ja kuitetun äänen välinen suhde (dry/wet)

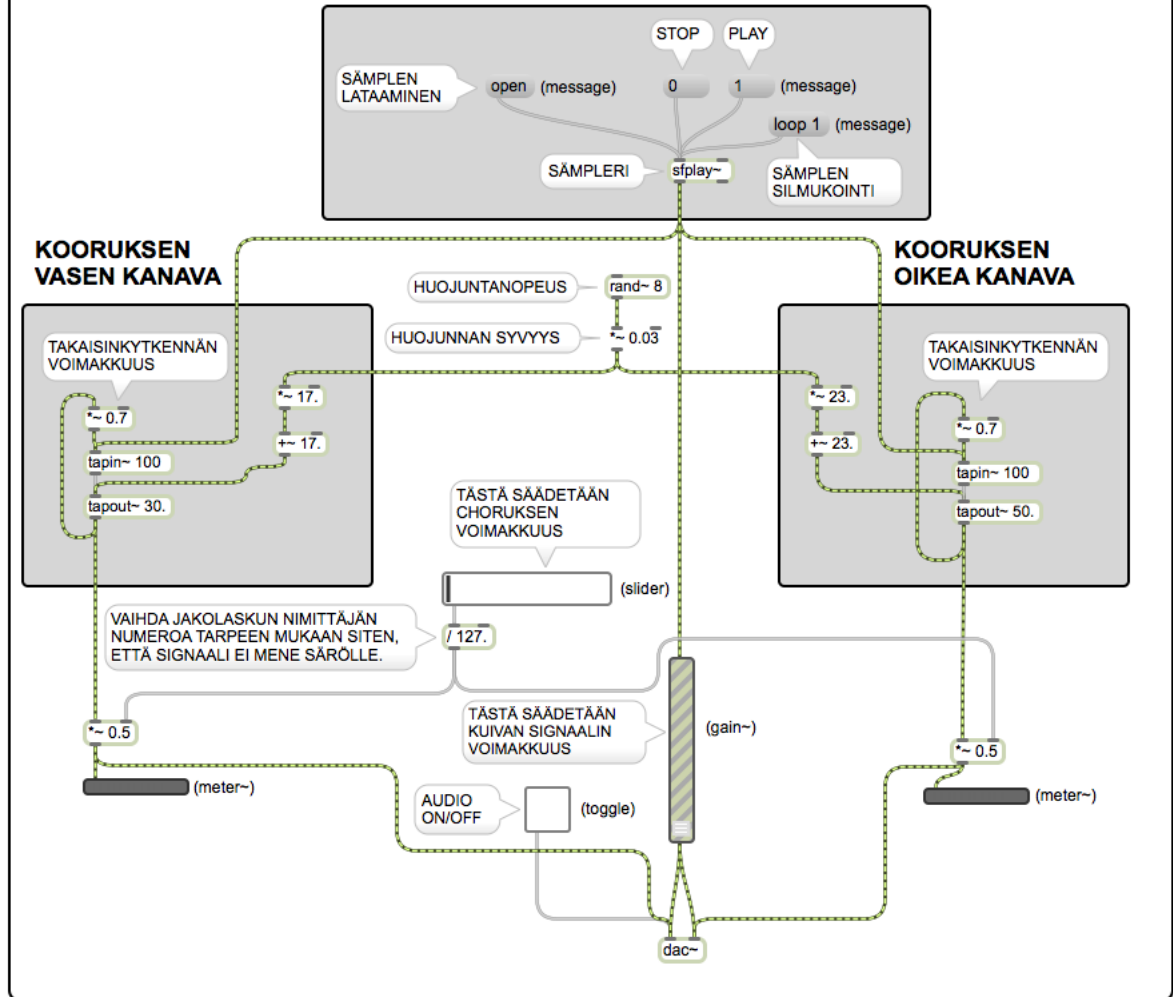
- ei-realistisia kaikutyyppiä: (diffuse, gated, non-linear, reverse)

LISÄÄ AIHEESTA:

<http://en.wikipedia.org/wiki/Reverberation>
http://en.wikipedia.org/wiki/Convolution_reverb
http://en.wikipedia.org/wiki/Gated_reverb
<http://www.harmonycentral.com/15/Effects-Signal-Processors/Convolution-Reverb-Basics/ba-p/35701199>

33. KOORUS

Patchillä toteutetaan äänilähteeseen koorus-efekti. Koorus-efektilaitteella sisään tulevaa audiosignaalia viivästetään siten, että viiveaika muuntelee jatkuvasti n. 20-30 millisekunnin välillä. Viivästetty signaali summataan alkuperäiseen audiosignaaliin.



HELP-TIEDOSTOT:

Objektin Help-tiedoston saa näkyviin klikkaamalla objektiin alt-näppäin pohjassa. Patchin tulee olla avattuna "unlocked" -moodissa.

INSPECTOR-IKKUNA:

Objektin Inspector -ikkunan saa näkyviin seuraavalla tavalla:

- 1) Aktivoi objekti, jonka Inspector -ikkuna halutaan avata.
- 2) Object -valikosta valitaan Inspector (Pikanäppäinkomento: cmd - i).

Saatuasi patchin valmiiksi, mieti miten voisit varioida ja kehittää patchiä eteenpäin. Toteuta ideasi!

KÄSITTEITÄ:

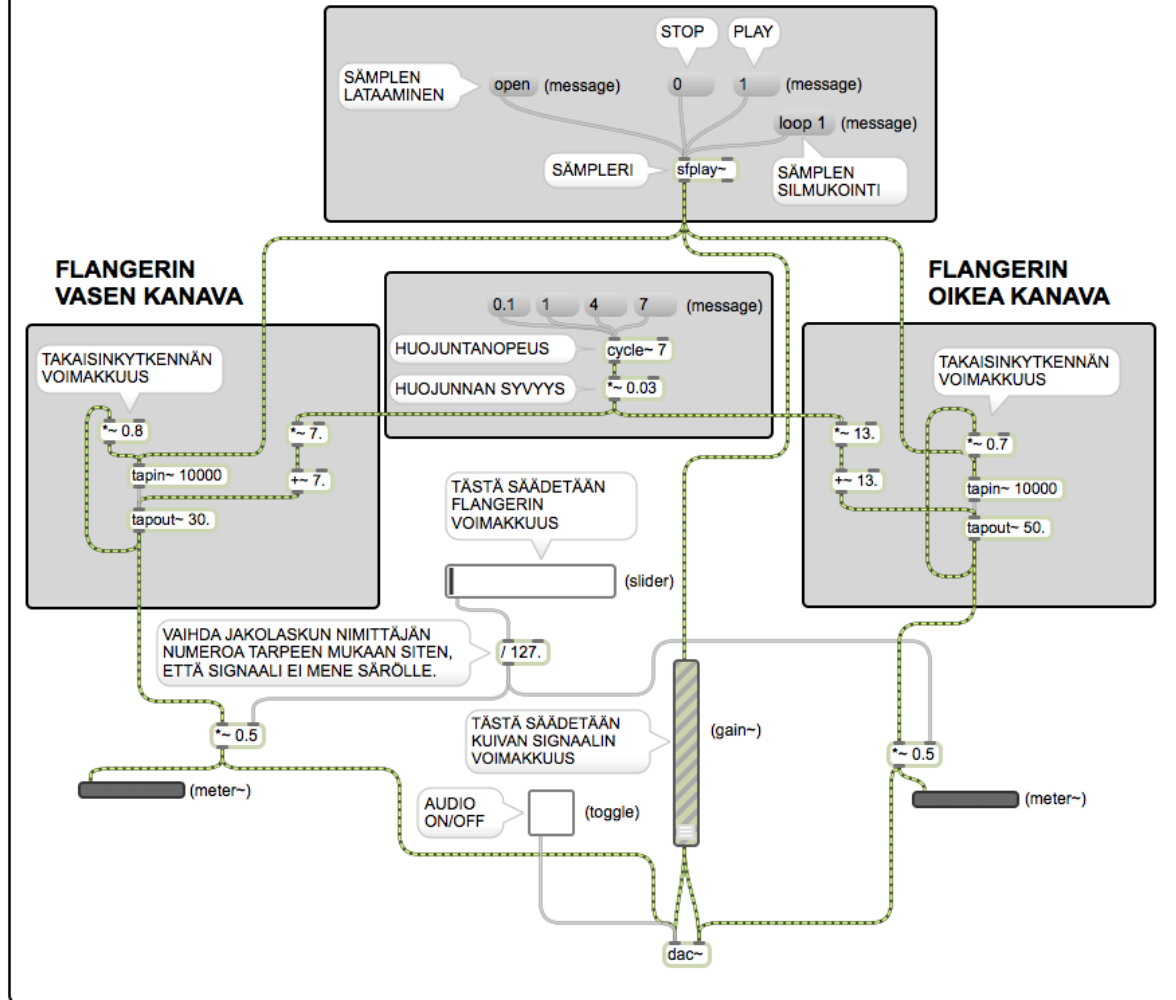
- koorus = chorus = kuoro
- takaisinkytkentä = feedback
- Dopplerin-ilmiö

LISÄÄ AIHEESTA:

<http://testtone.com/fundamentals/what-chorus-effect>
<http://www.musiciansfriend.com/resources/article/What-is-a-Chorus-effect/m710688>
<http://www.mediacollege.com/audio/effects/chorus/>
<http://fi.wikipedia.org/wiki/Chorus>
http://en.wikipedia.org/wiki/Chorus_effect
<http://www2.siba.fi/akustiikka/index.php?id=19&la=fi>

34. FLANGERI

Patchillä toteutetaan äänilähteeseen flangeri-efekti. Flangeri-efekttilaitteella sisään tulevaa audiosignaalia viivästetään siten, että viiveaika muuntelee jatkuvasti n. 1-10 millisekunnin välillä. Viivästetty signaali summataan alkuperäiseen audiosignaaliin.



HELP-TIEDOSTOT:

Objektin Help-tiedoston saa näkyviin klikkaamalla objektiin alt-näppäin pohjassa. Patchin tulee olla avattuna "unlocked"-moodissa.

INSPECTOR-IKKUNA:

Objektin Inspector-ikkunan saa näkyviin seuraavalla tavalla:
1) Aktivoi objekti, jonka Inspector-ikkuna halutaan avata.
2) Object-valikosta valitaan Inspector (Pikanäppäinkomento: cmd - i).

Saatuasi patchin valmiiksi, mieti miten voisit varioida ja kehittää patchiä eteenpäin. Toteuta ideasi!

KÄSITTEITÄ:

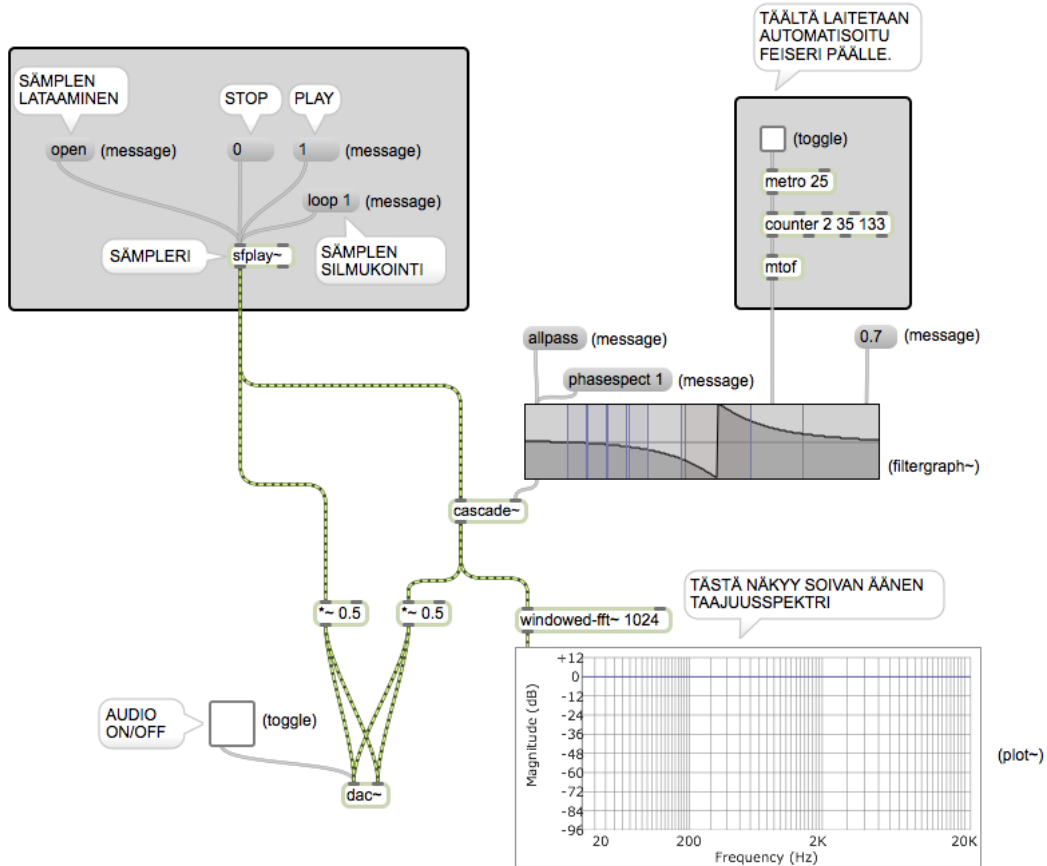
- takaisinkytkentä (feedback)
- Dopplerin-ilmiö

LISÄÄ AIHEESTA:

<http://testtone.com/fundamentals/what-chorus-effect>
<http://www.musiciansfriend.com/resources/article/What-is-a-Chorus-effect/m710688>
<http://www.mediacollege.com/audio/effects/chorus/>
<http://fi.wikipedia.org/wiki/Chorus>
http://en.wikipedia.org/wiki/Chorus_effect
<http://www2.siba.fi/akustiikka/index.php?id=19&la=fi>

35. FEISERI

Patchillä toteutetaan äänilähteeseen feiseri -efekti. Feiserissä äänisignaali menee yhden tai useamman kokopäästösuotimen läpi. Suotimilla muokataan äänisignaalin vaihetta.



HELP-TIEDOSTOT:

Objektin Help-tiedoston saa näkyviin klikkaamalla objektiin alt-näppäin pohjassa. Patchin tulee olla avattuna "unlocked" -moodissa.

INSPECTOR-IKKUNA:

Objektin Inspector -ikkunan saa näkyviin seuraavalla tavalla:

- 1) Aktivoi objekti, jonka Inspector -ikkuna halutaan avata.
- 2) Object -valikosta valitaan Inspector (Pikanäppäinkomento: cmd - i).

Saatuaasi patchin valmiiksi, mieti miten voisit varioida ja kehittää patchiä eteenpäin. Toteuta ideasi!

KÄSITTEITÄ:

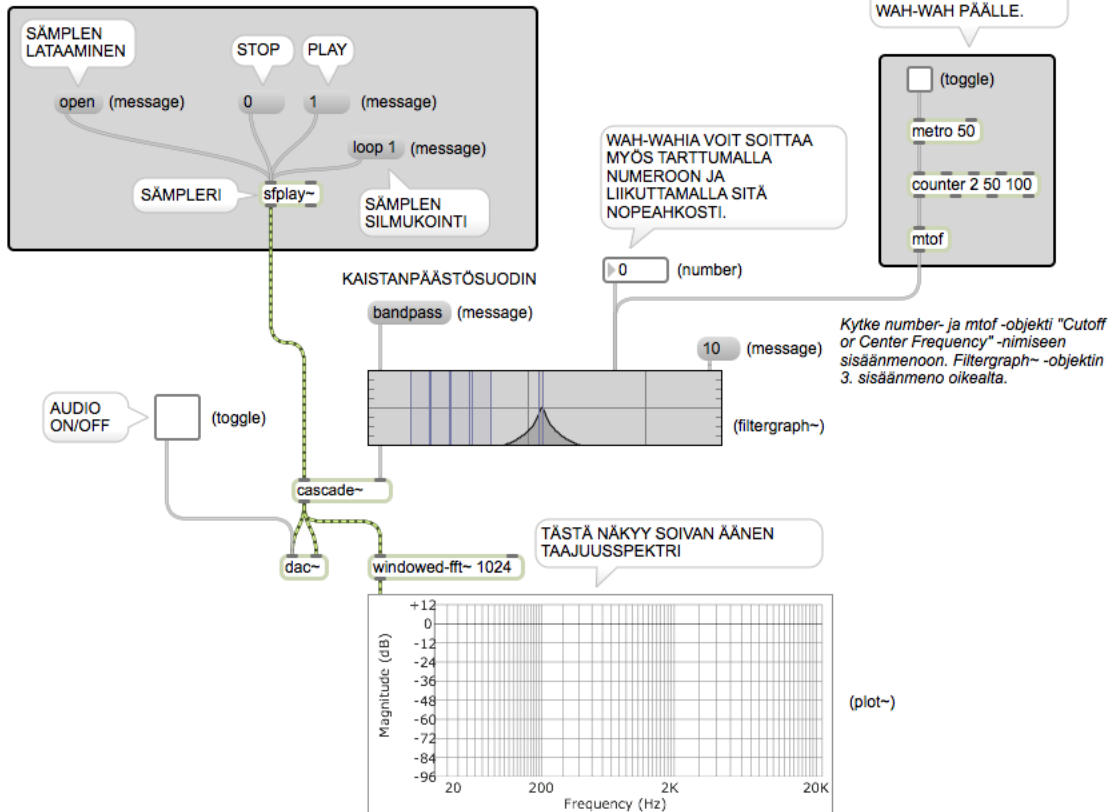
- kokopäästösuodin = all-pass filter
- kampsuodin-efekti

LISÄÄ AIHEESTA:

- <http://www.uaudio.com/blog/modulation-effects-basics/>
- <http://whatis.techtarget.com/definition/phaser>
- <http://testtone.com/fundamentals/what-phaser-effect>
- http://www.soundonsound.com/sos/mar06/articles/qa0306_1.htm
- http://en.wikipedia.org/wiki/Phaser_effect

36. WAH-WAH

Patchillä toteutetaan äänilähteeseen wah-wah -efekti.



HELP-TIEDOSTOT:

Objektin Help-tiedoston saa näkyviin klikkaamalla objektiin alt-näppäin pohjassa. Patchin tulee olla avattuna "unlocked" -moodissa.

INSPECTOR-IKKUNA:

Objektin Inspector -ikkunan saa näkyviin seuraavalla tavalla:

- 1) Aktivoi objekti, jonka Inspector -ikkuna halutaan avata.
- 2) Object -valikosta valitaan Inspector (Pikanäppäinkomento: cmd - i).

Saatuasi patchin valmiiksi, mieti miten voisit varioida ja kehittää patchiä eteenpäin. Toteuta ideasi!

KÄSITTEITÄ:

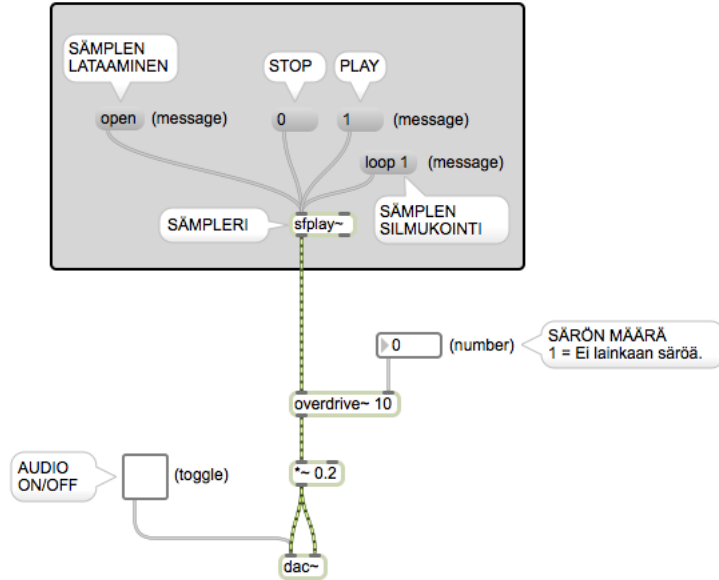
- kaistanpäästösuodatin
- liukuva suodattimen keskitäajuus
- spektrinen glissando

LUE LISÄÄ:

http://www.sweetwater.com/cs-Wah_Pedals
<http://www.jimdunlop.com/product/Cry-Baby>
[http://en.wikipedia.org/wiki/Wah-wah_\(music\)](http://en.wikipedia.org/wiki/Wah-wah_(music))
http://en.wikipedia.org/wiki/Wah-wah_pedal

37. SÄRÖ

Patchillä toteutetaan äänilähteeseen säröä.



HELP-TIEDOSTOT:

Objektin Help-tiedoston saa näkyviin klikkaamalla objektiin alt-näppäin pohjassa. Patchin tulee olla avattuna "unlocked" -moodissa.

INSPECTOR-IKKUNA:

Objektin Inspector -ikkunan saa näkyviin seuraavalla tavalla:
1) Aktivoi objekti, jonka Inspector -ikkuna halutaan avata.
2) Object -valikosta valitaan Inspector (Pikanäppäinkomento: cmd - i).

Saatuasi patchin valmiiksi, mieti miten voisit varioida ja kehittää patchiä eteenpäin. Toteuta ideasi!

KÄSITTEITÄ:

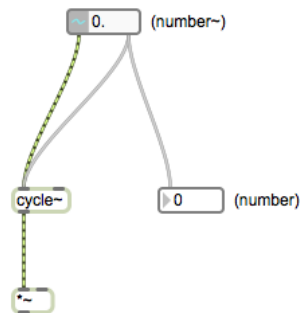
- särö
- vahvistuksen kyllästyminen
- takaisinkytkentä
- harmoninen kokonaissärö
- keskeismodulaatiosärö
- TIM-särö
- distortion
- overdrive
- gain

LISÄÄ AIHEESTA:

<http://www.soundonsound.com/sos/apr10/articles/distortion.htm>
<http://www.soundonsound.com/sos/may11/articles/live-tech-0511.htm>
<http://en.wikipedia.org/wiki/Distortion>
[http://en.wikipedia.org/wiki/Distortion_\(music\)](http://en.wikipedia.org/wiki/Distortion_(music))

38. ERILAISET JOHTOTYYBIT

Patch havainnollistaa erilaiset johtotyytit, joita Max:ssä käytetään.



HELP-TIEDOSTOT:

Objektin Help-tiedoston saa näkyviin klikkaamalla objektiin alt-näppäin pohjassa. Patchin tulee olla avattuna "unlocked" -moodissa.

INSPECTOR-IKKUNA:

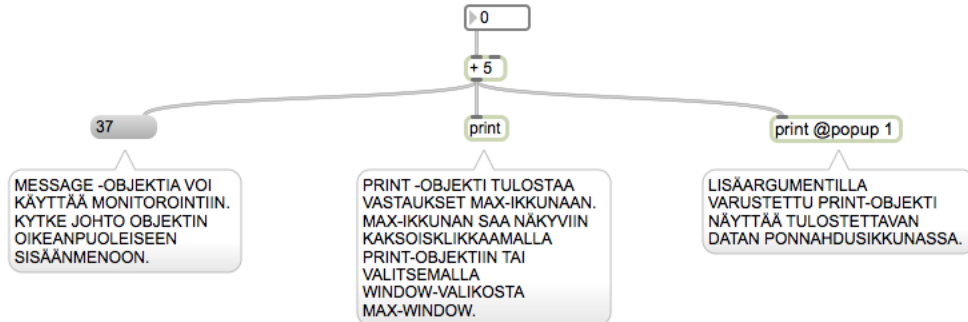
Objektin Inspector -ikkunan saa näkyviin seuraavalla tavalla:

- 1) Aktivoi objekti, jonka Inspector -ikkuna halutaan avata.
- 2) Object -valikosta valitaan Inspector (Pikanäppäinkomento: cmd - i).

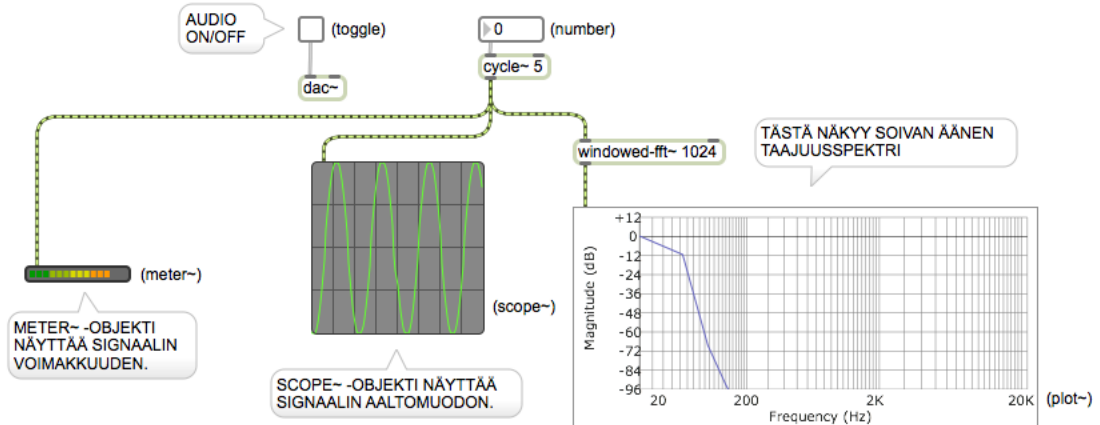
39. MONITOROINTIRAKENTEITA

Patch havainnollistaa erilaisia vaihtoehtoja kontrollidatan ja audiosignaalin monitorointiin.

KONTROLLIDATAN MONITOROINTI



AUDIOSIGNAALIN MONITOROINTI



HELP-TIEDOSTOT:

Objektin Help-tiedoston saa näkyviin klikkaamalla objektiin alt-näppäin pohjassa. Patchin tulee olla avattuna "unlocked" -moodissa.

INSPECTOR-IKKUNA:

Objektin Inspector -ikkunan saa näkyviin seuraavalla tavalla:

- 1) Aktivoi objekti, jonka Inspector -ikkuna halutaan avata.
- 2) Object -valikosta valitaan Inspector (Pikanäppäinkomento: cmd - i).

KÄSITTEITÄ:

- message-objekti
- print-objekti
- Max-ikkuna
- meter~ -objekti
- scope~ -objekti
- plot~ -objekti

LINKKEJÄ:

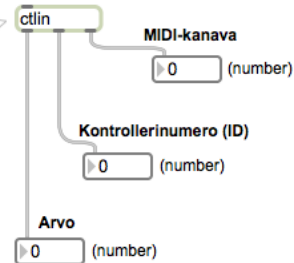
<http://cycling74.com>

40. ULKOISEN KONTROLLERIN KYTKEMINEN

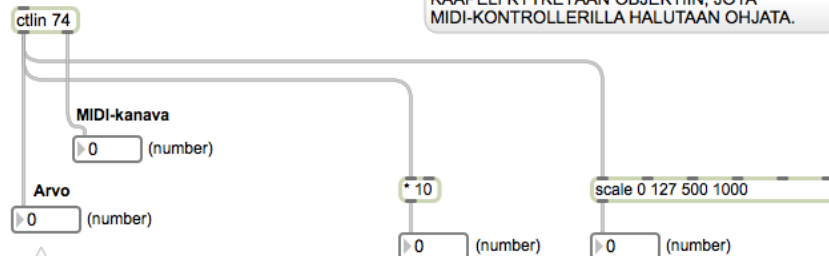
Patch havainnollistaa miten ulkoinen midi-kontrolleri kytketään Max-ohjelmaan.

KÄYTTÖOHJE:

- 1) KYTKE MIDI-KONTROLLERI KIINNI TIETOKONEESEEN MIDI INTERFACEN TAI USB-SISÄÄNMENON KAUTTA.
- 2) LIIKUTA KONTROLLERIASI.
- 3) NYT KONTROLLERISTA ULOSTULEVA DATA PITÄISI NÄKYÄ CTLIN-OBJEKTIN ULOSTULOIHIN LIITETYISSÄ NUMBER-OBJEKTEISSA. NUMBER-OBJEKTIEIN YLÄPUOLELLA OLEVAT TEKSTIT KERTOVAT, MITÄ KYSEINEN NUMERO SYMBOLISOI.



CTLIN-OBJEKTIN VASEMMANPUOLIMMAISESTA ULOSTULOSTA TULEE MIDI-KONTROLLERISTA TULEVA DATA. TÄSTÄ ULOSTULOSTA TULEVA KAAPELI KYTKETÄÄN OBJEKTIIN, JOTA MIDI-KONTROLLERILLA HALUTAAN OHJATA.



CTLIN-OBJEKTIN VASEMMANPUOLIMMAISESTA ULOSTULOSTA TULEE MIDI-KONTROLLERISTA TULEVA DATA. TÄSTÄ ULOSTULOSTA TULEVA KAAPELI KYTKETÄÄN OBJEKTIIN, JOTA MIDI-KONTROLLERILLA HALUTAAN OHJATA.

HELP-TIEDOSTOT:

Objektin Help-tiedoston saa näkyviin klikkaamalla objektiin alt-näppäin pohjassa. Patchin tulee olla avattuna "unlocked" -moodissa.

INSPECTOR-IKKUNA:

Objektin Inspector -ikkunan saa näkyviin seuraavalla tavalla:

- 1) Aktivoi objekti, jonka Inspector -ikkuna halutaan avata.
- 2) Object -valikosta valitaan Inspector (Pikanäppäinkomento: cmd - i).

KÄSITTEITÄ:

- ctlin-objekti
- midi-kanava
- kontrollerinumero

LINKKEJÄ:

<http://www.sweetwater.com/insync/midi-controller-buying-guide/>
<http://www.musicradar.com/news/tech/the-12-best-budget-midi-controller-keyboards-565848>
<http://www.synthtopia.com/content/2012/04/15/midi-keyboard-controller-comparison-what-to-look-for/>
http://en.wikipedia.org/wiki/MIDI_controller