

# Äänisynteesi ja efektit

Lähteet: -Tolonen, Välimäki, Karjalainen. (1998). "Evaluation of modern sound synthesis methods". Report no. 48, Helsinki University of Technology, Acoustics Lab.  
-Roads. (1996). "Computer music tutorial". MIT Press.  
-Tuomas Virtanen.(2001)."Audio signal modeling with sinusoids and noise".D-työ.  
-Vesa Peltonen, "Spectrum modeling with sinusoids, noise, and transients",  
Audioryhmän seminaari *Konekuulo*, kesä 2000.  
-Juha Tuomi, "Audio effects processing", Matti Vihola, "Sound synthesis methods"  
Audioryhmän seminaari *Äänisynteesi ja efektit*, kesä 2001.



## Äänisynteesi

- Johdanto
- Abstraktit algoritmit
- Näytteistävä synteesi
- Spektrimallit
- Fysikaalinen mallinnus

## Efektit

- Kiinteän viiveen efektit
- Muuttuvan viiveen efektit
- Wah-wah

## Johdanto

- Äänisynteessin tavoitteena on tuottaa ääniä, jotka
  - ovat musiikillisesti kiinnostavia
  - ovat realistisia (=muistuttavat aitoa soitinta, ei aina tavoitteena)
- Algoritminen kompleksisuus
  - äänet on voitava tuottaa reaaliajassa
  - toisaalta laadukas synteesi vaatii komplekseja järjestelmiä; liikaa yksinkertaistettu ääni kuulostaa keinotekoiselta
- Syntetisoidun äänen intuitiivinen kontrolli
  - soittajan interaktio tekee tuotetusta äänestä eloisan
- Synteesimenetelmien luokittelu
  1. abstraktit algoritmit
  2. näytteistämiseen perustuva synteesi (tallennus, prosessointi)
  3. spektrimallit
  4. fysikaaliset mallit

## 1 Abstraktit algoritmit

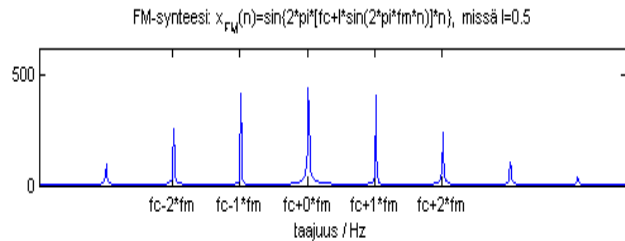
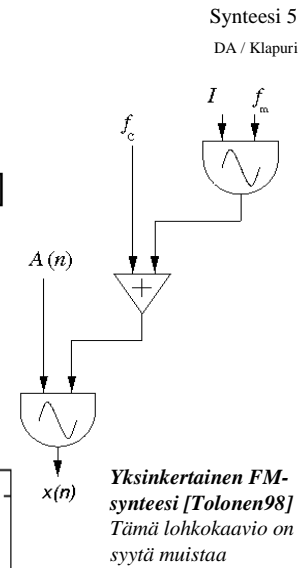
- Abstrakteihin algoritmeihin perustuvalla synteessillä on tyypillistä yksinkertaisuus ja helppo toteutettavuus
  - usein 10 Matlab-koodiriviä riittää
  - dataa ei tarvita muutamaa parametriarvoa enempää

## 1.1 FM-synteesi

- FM (engl. *frequency modulation*)
- Käytetty esim. radiolähetyksissä jo puoli vuosisataa
- 60-luvun lopulla John Chowning keksi soveltaa FM-synteesiä äänen tuottamiseen
  - havainto: varsin monimutkaisia spektrejä voidaan tuottaa vain parilla jänniteohjatulla oskillaattorilla
  - voidaan tuottaa myös aikamuuttuvia ja siten luonnollisilla ääniä
- 1983 Yamaha julkaisi DX7-syntetisaattorin
  - suuri kaupallinen menestys: äänenlaadultaan hyvä ja hinnaltaan tavallisen kuluttajan hankittavissa oleva soitin
  - FM-synteesi pysyi dominoivana synteesimenetelmänä vuosia
  - käytössä edelleen useissa syntikoissa ja "SoundBlaster-compatible" äänikorteissa

# FM-synteesi

- Kuva: yksinkertainen FM-synteesi
 
$$x(n) = A(n) \sin[2\pi(f_c + I \sin(2\pi f_m n))n]$$
  - koostuu kahdesta sini-oskillaattorista
  - kanta-aallon taajuutta  $f_c$  moduloi toinen oskillaattori (modulaatiotaajuus  $f_m$ )
  - modulaatioindeksi  $I$
  - äänen aikariippuva amplitudi  $A(n)$
- Alakuva: syntyvän äänen spektri

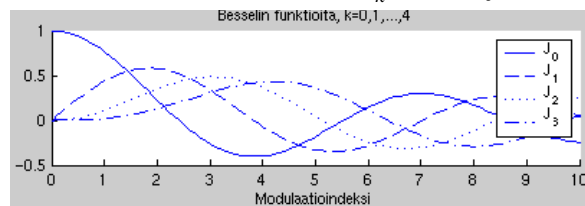


# FM-synteesi

- Tarkasteltaessa FM-synteesiä analyyttisesti kirjoitetaan
 
$$x(n) = A(n) \sin[2\pi f_c n + I \sin(2\pi f_m n)]$$
  - huomaa että tässä kaavassa moduloidaan vaihetta eikä taajuutta  
→ oikeampi nimi olisi vaihemodulaatio (PM)
  - FM-synteesiä voidaan tarkastella käyttäen yo. lauseketta, sillä vaihe on taajuuden integraali ja sinin integraali on sini  
→ vaiheen / taajuuden modulointi tuottaa äänisynteesin kannalta olennaisesti samantyyppisen äänen
  - analogilaitteissa aina FM-toteutus, PM onnistuu käytännössä vain digitaalisesti
- FM-synteesin ja PM-synteesin tuottamien äänten spektrejä on vertailtu pari sivua edempänä

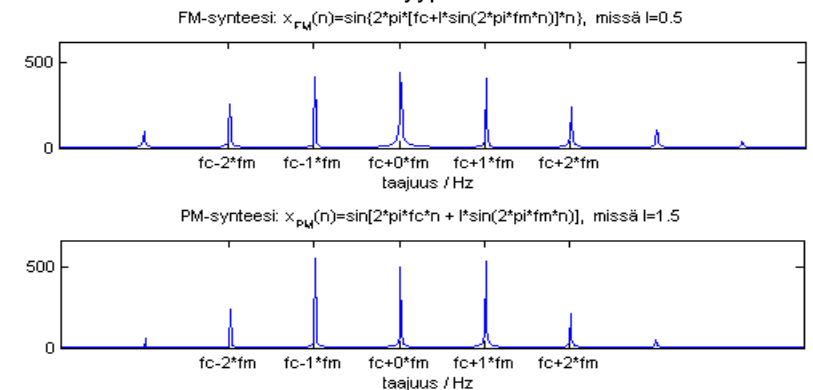
# FM-synteesi

- Edellisen sivun lauseke voidaan kirjoittaa muotoon
 
$$x(n) = A(n) \sum_{k=-\infty}^{\infty} J_k(I) \sin[2\pi(f_c + kf_m)n]$$
 missä  $J_k$  on Besselin funktio astetta  $k$ .
- Yllä olevasta lausekkeesta voi todeta, että PM-synteesi (ja myös FM-synteesi) tuottaa taajuuskomponentit
 
$$f_n = f_c \pm kf_m, k = 1, 2, \dots$$
- Kuva alla: Besselin funktioiden  $J_k(I)$  arvoja



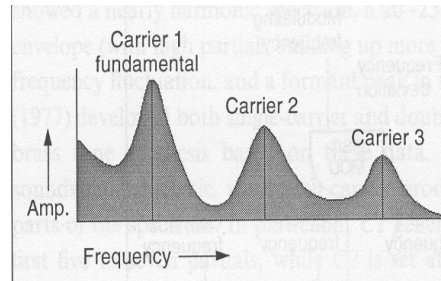
# FM-synteesi

- Kuvat: FM- ja PM-synteesin tuottamien äänten spektrit
  - harmoninen ääni saadaan asettamalla kanta-aallon ja modulaattorin taajuudet kokonaislukuosuhteeseen
  - muut kuin harmoniset äänet tyyppillisesti 'kellomaisia' tai 'metallisia'



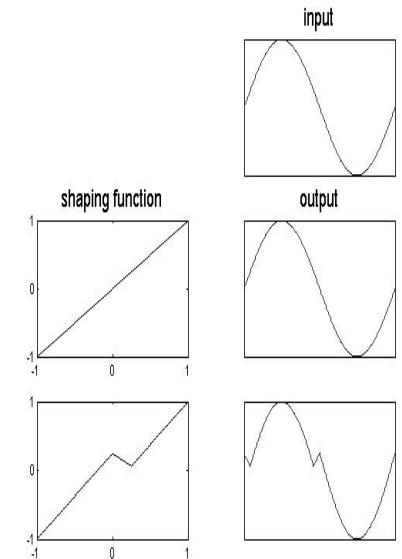
## FM-synteesi – jatkokehitystä

- Yksinkertaisessa FM-synteesissä taajuuskomponenttien amplitudisuhteet ovat melko epäsäännöllisiä, kun modulaatioindeksiä kasvatetaan suureksi
- Rikkaita ja silti säännönmukaisia spektrejä saadaan tietynlaisilla takaisinkytkentärakenteilla
  - selvittää edelleen parilla oskillaattorilla
- Toinen mahdollisuus on koostaa haluttu spektri summaamalla useiden yksinkertaisten FM-synteesien äänet
  - kuva [Roads96]



## 1.2 Waveshaping-synteesi

- Idea: käytetään epälineaarista *funktiota* muotoilemaan sisääntulevan *herätesignaaliin* aaltomuotoa
- Muotoilufunktio kuvaa herätesignaalin tason väliltä  $[-1, 1]$  ulostulon samalle välille
- Kuva: kaksi eri muotoilufunktiota ja niiden tuottama aaltomuoto siniherätteelle
- Alhaalla oikealla olevan aaltomuodon tuottaminen Matlabissa:
  - `t=0:0.1:2*pi;`
  - `x=sin(t);`
  - `y=0.25+1.25*x-min(0.5,max(0,2*x));`



## Waveshaping-synteesi

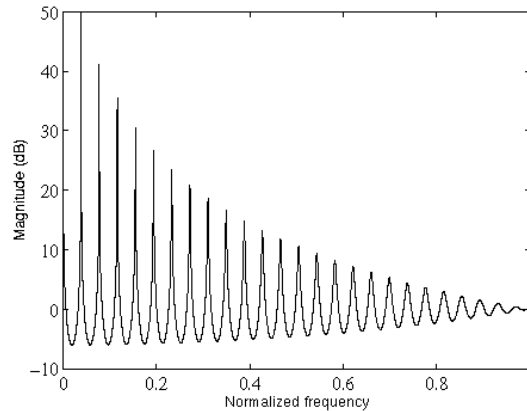
- Käyttämällä muotoilufunktiona Chebyshevin polynomien lineaarikombinaatiota voidaan säätää harmonisten keskinäisiä suhteita
  - $k$ :n asteen Chebyshevin polynomi muotoilufunktiossa tuottaa sinimuotoiselle herätesignaalille ulostulon sinin  $k \cdot f$
  - saadaan hallitusti approksimoitua esim. todellisen instrumentin harmonista rakennetta
- Perustapauksessa herätesignaalina on sini, mutta se voi tietysti olla jokin muukin
- Äänen aikakäyttäytymistä voidaan hallita jälkiprosessoinnilla (esim. amplitudimodulaatio)

## 1.3 Karplus-Strong algoritmi

- Hyvin yksinkertainen ja laskennallisesti tehokas algoritmi
  1. lyhyt puskuri alustetaan satunnaisilla näytearvoilla
  2. puskuri soitetaan läpi
  3. puskurin arvot suodatetaan matala-asteisella alipäästösuodattimella, esim.
 
$$y(n) = 0.5[y(n-P) + y(n-P-1)]$$
 missä  $y(n)$  on puskurin arvo ja  $y(n-P)$  on  $P$ -mittaisen puskurin arvo edellisellä soittokerralla. Yllä olevan suotimen taajuusvaste:
 
$$H(z) = 0.5(1 + z^{-1})$$
  4. palataan kohtaan 2 (soitetaan puskuri yhä uudelleen)
- Sopivilla parametreilla algoritmin tuottamat äänet muistuttavat näpättyä kielisoitinta

## Karplus-Strong algoritmi

- Kuva: Karplus-Strong mallin taajuusvaste [Tolonen98]



## Karplus-Strong algoritmi

- Lisäparametri K-S algoritmiin saadaan muuttamalla alipäästösuodatus seuraavanlaiseksi operaatioksi

$$y(n) = \begin{cases} 0.5[y(n-P) + y(n-P-1)] & \text{jos } r < b \\ -0.5[y(n-P) + y(n-P-1)] & \text{jos } r > b \end{cases}$$

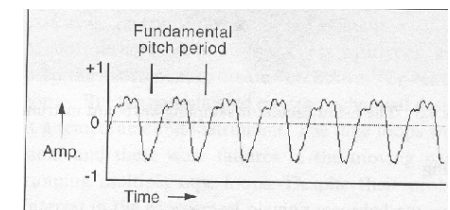
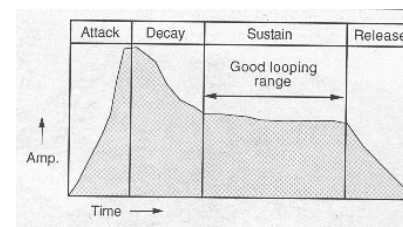
- missä  $r$  on satunnaismuuttuja välillä  $[0, 1]$ , ja  $b$  on "blend factor"
- puskurin kukin näytearvo kerrotaan satunnaisesti joko 1:llä tai -1:llä alipäästösuodatuksen jälkeen
- mikäli  $b=1$ , algoritmi redusoituu edellisellä sivulla esitettyyn
- mikäli  $b=1/2$ , keskimäärin joka toinen näyte kääntyy vastakkaismerkkiseksi, äänen jaksollisuus katoaa, ja saadaan rumpumaisia, perkussiivisiä ääniä
- mikäli  $b=0$ , koko signaali käännetään joka lukukerran jälkeen, äänen aallonpituus kaksinkertaistuu ja äänenkorkeus puolittuu

## 2 Näytteistykseen perustuva synteesi

- Engl. *sampling synthesis*
- Äänitettyjen äänen manipulointia ja yhdistelyä on harrastettu 1920-luvulta asti
  - ns. konkreettinen musiikki
- Laskennallisesti on äärimmäisen yksinkertaista vain soittaa tallennettu ääni
- Muistivaatimukset ovat tämän lähestymistavan ongelma
  - ei ole taloudellista tallentaa kaikkia mahdollisia ääniä eri instrumenteilla, eri korkeuksilta, eri soittotavoin
  - näytteistykseen perustuvan synteessin taito on *muistinkulutuksen minimoimisessa ja samalla äänenlaadun maksimoimisessa*
- Perusideoita äänen mallintamiseksi vähemmällä datalla:
  - silmukkapuskuri, äänenkorkeuden siirtäminen, datan reduktio

## 2.1 Wavetable-synteesi (aaltotaulukkos.)

- Silmukkapuskurin käyttö (engl. *looping*)
  - kuva alla [Roads96]: useimpien instrumenttiäänten steady-state (=sustain) osa on suurinpiirten jaksollinen
    - voidaan ottaa lyhyt näyte ja soittaa sitä silmukkapuskurista
  - äänen lyhyt alkutransientti voidaan haluttaessa mallintaa erikseen
  - silmukkapuskurin pituus täytyy olla aallonpituuden monikerta
  - yleensä puskuriiin täytyy ottaa enemmän kuin yksi aalto, jotta äänen aikamuuttuvia ominaisuuksia saadaan mukaan



## Wavetable-synteesi

- Äänenkorkeuden muuntelu
  - muistia voidaan säästää tallentamalla soittimelta vain esim. joka 3. tai joka 4. nuotti
  - väliin jäävät nuotit saadaan muuntelemalla näytteiden soitt nopeutta, tai uudelleennäytteistämällä ääntä digitaalisesti
- Muita datamäärän vähennyskeinoja
  - häviötön kompressio tai kuuloon perustuva häviöllinen koodaus
  - karkeamman näytteistystaajuuden tai kvantisoinnin käyttö

## 2.2 Multiple wavetable -synteesi

- Useita näytekureita (aaltotaulukoita) soitetaan yhtä aikaa
- Wavetable cross-fading
  - tallennetaan näytteitä useista kohdista mallinnettavaa ääntä
  - näytteitä soitetaan silmukassa ja ristiin-feidataan pehmeästi edellisestä näytteestä seuraavaan
  - jälkikäsitellään kertomalla halutulla amplitudiverhokäyrällä
- Wavetable stacking (pinoaminen)
  - haluttu aaltomuoto koostetaan muodostamalla painotettu summa useista yhtäaikaan soitettavista elementaarisisista aaltomuodoista
  - kaupallisissa laitteissa yleensä 4-8 wavetablea pinossa
  - ongelma: löytää joukko elementaarisia aaltomuotoja ja amplitudiverhokäyriä, joilla pystytään tehokkaasti esittämään erilaisia luonnollisia ääniä

## 2.3 Raesynteesi

- Käytetään myös nimeä granulaarisynteesi (engl. *granular synthesis*)
- Idea on koostaa haluttu ääni "ääniatomeista" tai "rakeista" (grains)
  - ääni saadaan summaamalla näitä elementaarisia osia aikatasossa
  - äänen tyypin määräävät *rakeiden aaltomuoto* ja niiden *ajallinen jakauma* yhdessä
  - yhden rakeen kesto voi olla n. 1-300 ms (ikkunoiu sini / näytteistetty ääni)
- Asynkroninen raesynteesi
  - sirotellaan äänirakeita tilastollisesti (satunnaisesti) yli aika-taajuus tason → "äänipilvi"
  - tehokas uusien äänitapahtumien generoimisessa, todellisten äänten simulointi on vaikeaa
- Äänen korkeuteen synkronoitu raesynteesi
  - jaksollisuus vastaa perustaajuutta
  - parempi suorituskyky todellisten äänten simuloinnissa

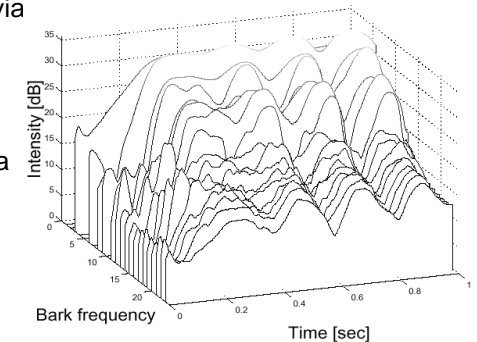
## 3 Spektrimallit

### 3.1 Additiivinen synteesi

- Idea: koostetaan ääni summaamalla sinikomponentteja

$$y(t) = \sum_k A_k(t) \sin[2\pi f_k(t)t]$$

- yksittäisten sinien amplitudien ja taajuuksien kontrollifunktiot  $A_k$  ja  $f_k$  ovat hitaasti muuttuvia
- Saadaan periaatteessa erittäin korkea laatu
- Varjopuolia
  - täytyy tallentaa paljon dataa (kontrollifunktioiden parametrit)
  - suuri määrä oskillaattoreita synteessissä
- Kuva: huilun harmonisten komponenttien ajallinen kehitys [Eronen2000]

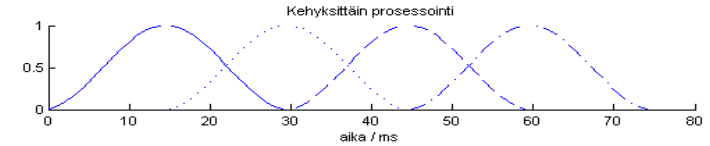


## 3.2 Vaihevokooderi

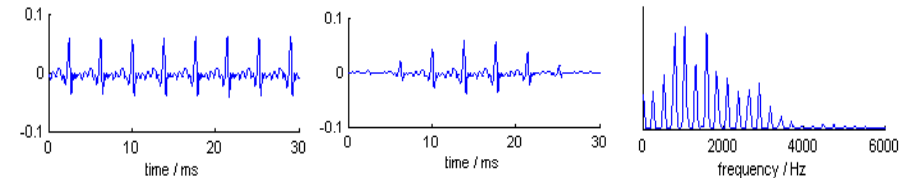
- Engl. *phase vocoder*
  - Keksittiin 60-luvulla puheen kompressiomenetelmiä tutkiessa; vocoder = voice coder
  - *sovelluksia nykyään: äänen aikaskaalaus, äänenkorkeuden muuntelu, äänen morphaus, äänen aika-taajuus tason muokkaus*
- Yleisnimi analyysi-synteesimenetelmille, joissa
  - äänisignaali esitetään sinikomponenttien summana
  - magnitudien ja taajuuksien lisäksi tallennetaan ja *syntetisoidaan myös sinien vaihe*
- Voidaan toteuttaa suodinpankkina tai lyhytaikaisen Fourier-muunnoksen (STFT) avulla
  - STFT yleisempi, esitämme vain sen

## Vaihevokooderi Kehyksittäin prosessointi

- Lyhytaikaista signaalinkäsittelyä (engl. *short-time SP*)
  - signaali käydään läpi kehyksittäin (paloittain)
  - kerrotaan sig. ikkunafunktiolla, joka on nolla kehyksen ulkopuolella



- kussakin kehyksessä signaali (1) painotetaan ikkunafunktiolla ja (2) lasketaan lyhytaikainen diskreetti Fourier muunnos kehyksessä



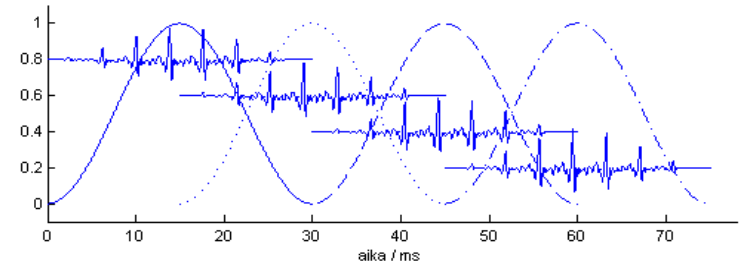
## Vaihevokooderi Kehyksittäin prosessointi

- Edellisen sivun prosessointi tuottaa spektrogrammin
  - aika-taajuus tason esitys: kehyksien kompleksiset spektrin yli ajan
- Miksi yleensäkin ottaen prosessoidaan kehyksittäin?
  - Fourier-m. esittää signaalin *vakiotaajuuksisten* sinien summana
  - todelliset äänisignaalit eivät kuitenkaan ole stationäärisiä, vaan aikamuuttuvia
  - oletetaan signaali stationääriseksi riittävän lyhyessä aikakehyksessä
- Äänisignaaleille kehyksen pituus vaihtelee sovelluksesta riippuen välillä 10 ms – 100 ms
  - puhesignaaleille tyypillisesti n. 20 ms
- Signaalin esittäminen kehyksittäisen spektrin avulla toimii hyvin harmonisille ja hitaasti muuttuville äänille, mutta transienttimaiset äänet leviävät ajallisesti koko kehykseen

## Vaihevokooderi Aikatason signaalin rekonstruointi

- Overlap-add tekniikka:
  1. muunnetaan käänteisellä Fourier muunnoksella ( $DFT^{-1}$ ) kunkin kehyksen spektri takaisin aikatason signaaliksi
  2. ikkunoidaan signaali kussakin kehyksessä Hanning-ikkunalla
  3. peräkkäiset kehykset asetetaan 50 % tai enemmän limittäin, ja summataan pisteittäin

Overlap-add: summataan peräkkäisten kehyksien ikkunoidut signaalit



- Vaihevokooderi mahdollistaa äänen keston ja korkeuden muuttelun
- Aikaskaalaus
  - muutetaan kehysten välistä askelta synteesivaiheessa TAI vaihtoehtoisesti tuplataan / jätetään väliin kehyksiä sopivin välein
  - aikatazon ikkunointi täytyy suunnitella huolella ettei tule artefaktoja
  - spektrinäytteiden vaiheita käsitellään siten, että vaiheen aikaderivaatta säilyy muuttumattomana (kesto • 2 → vaiheet • 2)
- Äänen korkeuden muuttaminen
  - aikaskaalataan ensin, sitten muutetaan näytteistystaajuutta siten että aikaskaala palautuu normaaliksi mutta äänenkorkeus muuttuu
  - suuret äänen korkeuden muutokset saavat erityisesti puheen kuulostamaan erikoiselta, koska äänen formantit (karkea spektri) siirtyvät
- Spektrogamma-esitysmuodossa ääntä voidaan muillakin tavoin kätevästi muokata ennen käänteismuuntamista takaisin aikatasoon
  - esim. suodatus: kerrotaan spektri suotimen vasteella
  - äänten morphaus: magnitudi- ja vaihespektrien interpoloinnilla

### 3.3 Sini+kohina -malli

- Signaalimalli

$$x(t) = \sum_{n=1}^N a_n(t) \cos[2\pi f_n(t)t + \varphi_n(t)] + r(t)$$

→ Mallinnettu signaali  $x(t)$  esitetään  $N:n$  sinin (taajuus, amplitudi, vaihe) ja *kohinaresiduaalin*  $r(t)$  avulla

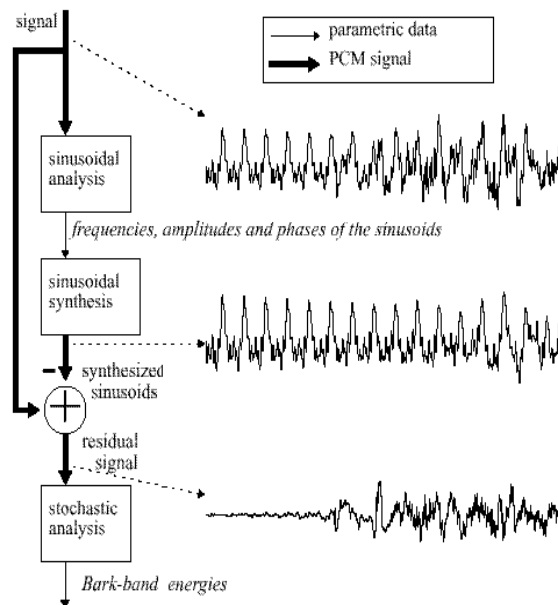
- Additiivinen synteesi

- Fourierin teoreeman mukaan mikä tahansa aaltomuoto voidaan esittää sinien summana
- järkevää vain jaksollisille signaaleille, joille tarvittavien sinien määrä on pieni
- ei-deterministinen osa vaatisi suuren määrän sinikomponentteja → käytetään stokastista mallinnusta

### Sini+kohina -malli

#### Analyysi

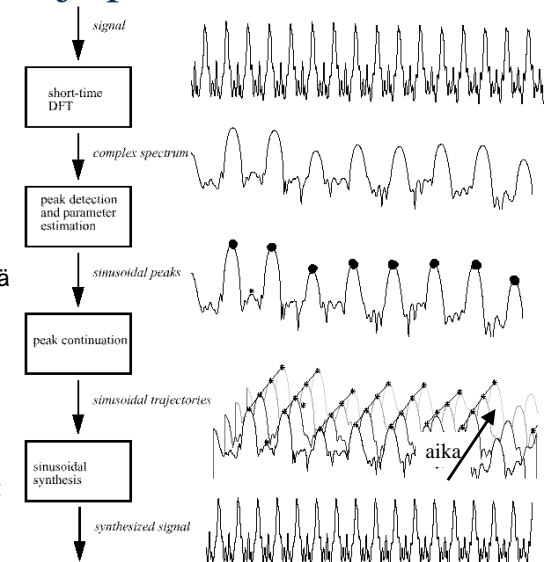
- Vuokaavio [Virtanen 2001]
  1. havaitaan sinit kehyksittäin spektristä
  2. estimoidaan sinien parametrit ja syntetisoidaan ne uudelleen
  3. vähennetään sinit alkup. signaalista
  4. mallinnetaan jäljelle jäävä kohinaresiduaali
- Saadaan
  - sinien parametrit
  - kohinan taso eri taajuuskaistoilla



### Sini+kohina -malli

#### Sinien havaitseminen ja parametrit

- Vuokaavio: [Virtanen01]
- Spektriippiikit tulkitaan siniaalloiksi
  1. "piikki": lokaali maksimi kehysten magnitudi-spektrissä
  2. piikin taajuus, amplitudi ja vaihe voidaan poimia kompleksisesta spektristä
- Peräkkäisissä kehyksissä havaittujen piikkien yhdistely
  - saadaan ajallisesti jatkuvan sinin aikamuuttuvat parametrit
  - "sinusoidal trajectory"

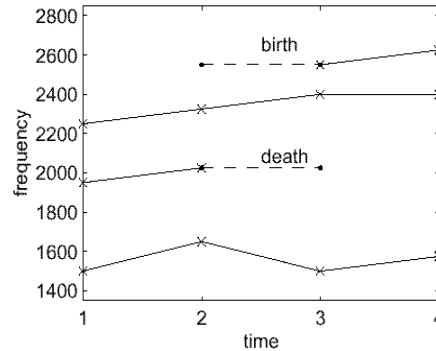


## Sini+kohina –malli Piikkien yhdistely

Synteesi 29  
DA / Klapuri

- **Tarvittaessa** voidaan peräkkäisissä kehyksissä havaitut spektrihiikit assosoida, yhdistellä aikamuuttuviksi sineiksi  
→ taajuudelle, amplitudille ja vaiheelle poimitaan arvo kehys
- Kuva: piikin yhdistelyalgoritmi [Virtanen2001]

- perustuu esim. syntyvän käyrän derivaattoihin; yritetään muodostaa pehmeä käyrä
- kill: jollei jatkoa löydy, lopetetaan sini
- birth: jos spektrihiikki ei ole jatkoa millekään entiselle sinille, luodaan uusi



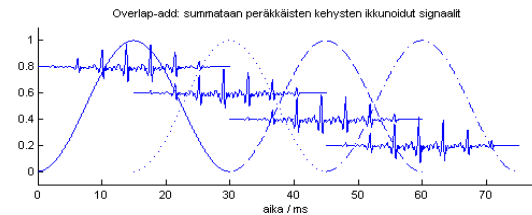
## Sini+kohina –malli Sinien synteesi

Synteesi 30  
DA / Klapuri

- Additiivinen synteesi

$$s(t) = \sum_{n=1}^N a_n(t) \cos[2\pi f_n(t)t + \varphi_n(t)]$$

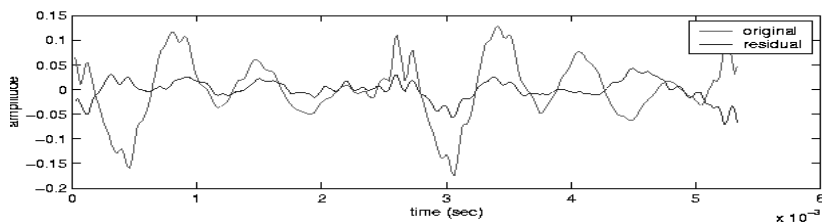
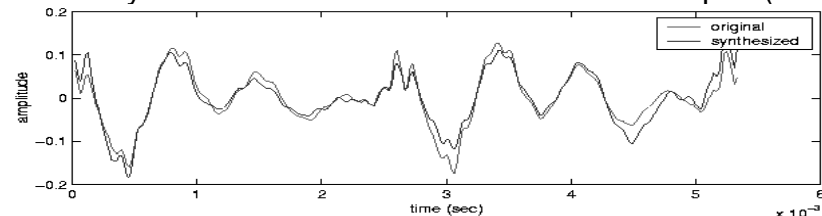
- Usein edellä esitetty piikkien yhdistely ei ole tarpeen, vaan
  - syntetisoidaan kussakin kehyksessä havaitut sinit erikseen, parametrit pysyvät vakioina koko kehyksen ajan
  - ikkunoidaan saatu aikatason signaali Hanning-ikkunalla
  - overlap-add: peräkkäiset kehykset limittäin, summataan pisteittäin



## Sini+kohina –malli Sinien synteesi, vähennys alkuperäisestä

Synteesi 31  
DA / Klapuri

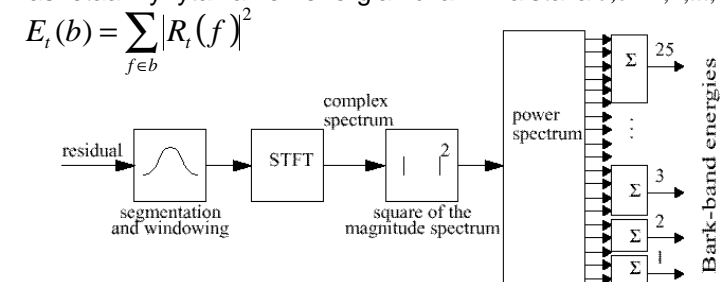
- Syntetisoidut sinit vs. alkuperäinen signaali (yllä)
- Vähennyksen tuloksena saatu residuaali vs. alkup.s. (alla)



## Sini+kohina –malli Kohinaresiduaalin mallinnus

Synteesi 32  
DA / Klapuri

- Residuaalisignaali saadaan vähentämällä syntetisoidut sinit alkuperäisestä signaalista aikatasossa
- Residuaalisignaali analysoidaan kehyksittäin
  - lasketaan residuaalin spektri  $R_t(f)$  kehyksessä  $t$
  - jaetaan spektri kuulon mukaisiin 25:een Barkin taajuuskaistaan
  - lasketaan lyhytaikainen energia kullakin kaistalla  $b, b=1,2,\dots,25$



$$E_t(b) = \sum_{f \in b} |R_t(f)|^2$$

## Kohinan syntetisointi parametreista

- Kohinaresiduaali esitetään parametreilla
  - tallennetaan kustakin aikakehyksestä vain *lyhytaikaiset energiat kullakin Barkin kaistalla*,  $E_i(b)$
  - tämä mallinnus voidaan tehdä, koska kohinan tapauksessa kuulo ei ole herkkä energiamuutoksille yhden Barkin kaistan sisällä
- Synteessissä  $R_i(f) = \sqrt{E_i}$ 
  1. generoidaan magnitudispektri, jossa kunkin Barkin kaistan energia jaetaan tasaisesti ko. kaistalle
  2. generoidaan *satunnaiset vaiheet* yli spektrin
  3. muunnetaan takaisin aikatasoon
  4. ikkunoidaan Hanning-ikkunalla
  5. overlap-add

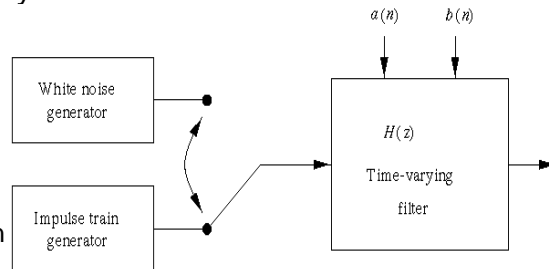
## Yleiskommentti

- Sini+kohina –mallilla on useita hyviä ominaisuuksia
  - esittää äänisignaalin kompaktisti verrattuna aikatason signaaliin
  - mallin avulla syntetisoitu ääni on hyvälaatuinen
  - malli on yleispätevä: mikä tahansa äänisignaali voidaan vetää läpi
  - suoraviivainen laskea (varsinkin ellei sinea j yhdistellä)
- Keston ja äänenkorkeuden muuttaminen helppoa
  - keston muuttaminen kuten vaihevokooderissa
  - äänenkorkeuden muuttaminen: manipuloidaan sinien taajuuksia
- Transienttiäännet ovat mallin ongelma
  - analyysikehyksen pituus määrää aikaresoluution  
→ transientit leviävät
  - on kehitetty myös sini+kohina+transientti –malli, jossa transientit mallinnetaan erikseen

## 3.4 Lähde-suodin synteesi

- Engl. *source-filter synthesis*
- Herätesignaalia suodatetaan aikamuuttuvalla suodattimella
- Käytetään myös nimeä *subtraktiivinen synteesi*:
  - spektriltään rikasta herätesignaalia suodatetaan halutuksi
- Sopii hyvin puheen syntetisointiin

- impulssijono-heräte vokaaleille
- kohinaheräte konsonanteille
- aikamuuttuvalla suodattimella tuotetaan foneemien spektrimuodot



## 4 Fysikaalinen mallinnus

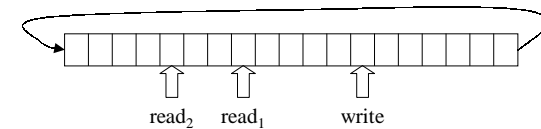
- Uusin synteessimenetelmä
  - 1970-luku: analogi-syntetisaattorien aika
  - 1980-luku: FM-synteesi
  - 1990-luku: fysikaalinen mallinnus
  - elektr. soittimien historiaa: [http://www.obsolete.com/120\\_years/](http://www.obsolete.com/120_years/)
- Simuloi soittimen akustista äänentuottomekanismia
- Käyttö musiikillisissa äänissä
  - jäljitellään olemassa olevia soittimia
  - mahdollistaa myös mielikuvituksellisten "soittimien" äänittämisen
- Joissain instrumenteissa on saavutettu erittäin hyvä äänenlaatu (kieli- ja puhallinsoittimet)
- Auttaa ymmärtämään soittimen toimintaa
  - mitkä piirteet äänissä ovat tärkeitä, mikä saa kuulost. hyvältä
- Vielä paljon tutkittavaa ja kehitettävää
- TTK:n Akustiikan labra (Otaniemi) on kansainvälisesti kovaa tasoa

## 5 Ääniefektit

- Tarkoitus tuottaa äänenväriin vaihtelua ja eloisuutta
  - efektejä kuullaan päivittäin musiikissa radiosta ja CD:ltä
  - myös ääniefektit elokuvissa ja mainoksissa
- Kaiunnan mallintaminen kuuluu efektien piiriin
  - tilaprosessointi yleensäkin
  - laaja aihe, jota ei käsitellä tällä luennolla
- Myös dynamiikan hallinta voidaan nähdä efektien näkökulmasta
- Seuraavassa rajoitutaan tarkastelemaan lähinnä viiveen käyttöön perustuvia ja yleisesti käytössä olevia efektejä

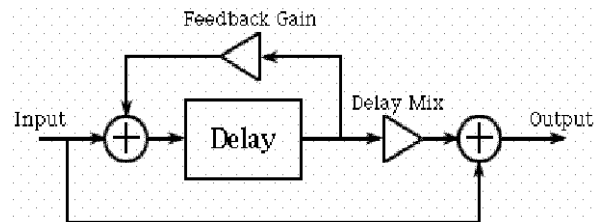
## Digitaalinen viivelinja

- Engl. *digital delay line (DDL)*
- Tallentaa näytteet myöhempää palautusta varten
- Viivelinja on signaaliprosessoreissa yleensä toteutettu rengaspuskurina
  - jono peräkkäisiä muistiosoitteita, joihin näytteet tallennetaan
  - kiinteä viiveen aika, suhteessa jonon pituuteen
- Multitap-viivelinja (ks. kuva)
  - jokaista jonoon tehtävää kirjoitusoperaatiota kohti voidaan tehdä monta lukua – ei vaadi sen enempää kuin tavallinenkaan viivelinja



## 6 Kiinteän viiveen efektit

- Ryhmitellään yleensä kolmeen pituusalueeseen
  - lyhyt (< 10 ms)
  - keskipitkä (10-50 ms)
  - pitkä (> 50 ms)
- Käyttämällä takaisinkytkentää alle 1:n vahvistuksella, viivettä voidaan toistaa kunnes viivästetyn signaalin amplitudi putoaa taustakohinan tason alapuolelle



### 6.1 Lyhyt viive

- Alle 10 ms
- Hyvin lyhyt viive (muutamien näytteiden luokkaa) miksataan alkuperäisen signaalin kanssa
  - ekvivalentti FIR-alipäästösuodatuksen kanssa
- Kampasuodinefekti tulee havaittavaksi kun viive on 0.1 – 10 ms

## Lyhyt viive Kampasuodin

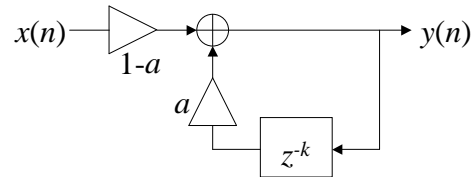
Synteesi 41  
DA / Klapuri

- Kampasuotimen (engl. *comb filter*) siirtofunktio on muotoa

$$H(z) = \frac{1-a}{1-az^{-k}}$$

missä

- $a$  määrää takaisinkytkennän vahvistuksen ( $a < 1$  jotta stabiili)
- $k$  määrää viiveen
- $(1-a)$  osoittajassa normalisoi vasteen maksimikohdan 0 dB:hen

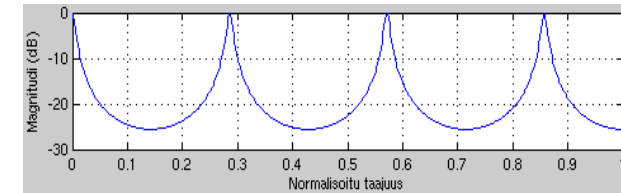


- Kampasuotimille on useita käyttötarkoituksia audio-DSP:ssä: mm. kaiun mallintamisessa, efekteissä, ja digitaalisten resonaattorien teossa

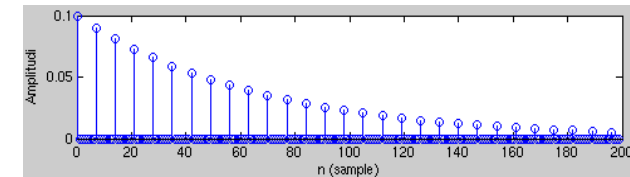
## Lyhyt viive Kampasuodin

Synteesi 42  
DA / Klapuri

- Tehdäänämpä kampasuodin Matlabissa  
`a=0.9; k=7;`  
`B=1-a; A=[1 zeros(1,k-1) -a];`
- Taajuusvaste (`freqz(B,A)`) on "kampamainen"



- Impulssivaste (`impz(B,A)`) on geometrinen sarja jossa on nollia välissä



## 6.2 Keskipitkä viive

Synteesi 43  
DA / Klapuri

- Noin 10 – 50 ms
- Käytetään yleensä "tukevoittamaan" ääntä
  - esim. pop-musiikissa
  - laulu, rummut, syntetisaattorit
  - aiheuttaa *illusion kasvaneesta äänekkydestä*, ilman että signaalin todellista amplitudia kasvatetaan
- Viivästetyt äänet *eivät* kuulu erillisinä kaikuina
  - presedenssiefekti ihmiskuulossa toimii, jos aikaero on alle 35 ms
  - äänen tulosuunta kuullaan ensimmäisen aaltorintaman mukaan
- "Kaksinkertaistumisefekti"
  - alkuperäisen viivästetyn signaalin miksaaminen saattaa aiheuttaa *aistimuksen useista äänilähteistä*

## 6.3 Pitkä viive

Synteesi 44  
DA / Klapuri

- Käytetään tuottamaan diskreettejä kaikuja, alkuperäisen signaalin toistoja, jotka kuulostavat siltä kuin signaali olisi heijastunut jostain pinnasta
- 50 ms viive implikoi noin 8 metrin etäisyyttä lähteen ja heijastavan pinnan välillä
  - ihmiskorva pyrkii välittämään tietoa fyysikaalisesta ympäristöstä

## 7 Muuttuvan viiveen efektit

Synteesi 45  
DA / Klapuri

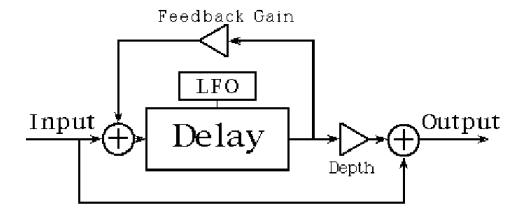
### 7.1 Flanger

- Flangerin yleinen periaate on
  - flanger = signaali + viivästetty signaali, missä viiveen pituus  $T_d$  vaihtelee jatkuvasti
- Ääniefektinä tätä käytti ensimmäisenä Les Paul v. 1945
  - järjestelmä koostui kahdesta nauhurista, joista toisen pyörimisnopeutta voidaan kontrolloida
  - 1960-luvulla flanger-efekti saatiin aikaan kahdella synkronoidulla analogisella nauhurilla ja miksauskonsolilla. Efekti aiheutettiin koskettamalla toisen kelan reunaa (flange), ja näin hidastamalla sitä

Synteesi 46  
DA / Klapuri

### Flanger

- Elektroniset flangerit käyttävät viivelinjaa, jonka pituutta varioi matalataajuuksinen oskillaattori
  - engl. *low-frequency oscillator, LFO*
  - yleensä sini- tai kolmioaaltoa taajuudella 0.1 – 20 Hz
- Jotta vältetään epäjatkuvuuskohdat ja napsahdukset, voidaan käyttää murtoviiveitä
  - viive ei ole näytevälän monikerta
  - Käytetään interpoloivaa FIR-suodinta tai kokopäästö-suodinta



### Flanger

Synteesi 47  
DA / Klapuri

- Flangeria voidaan ajatella pyyhkäisevänä kampsuodinefektinä
  - useita nolliä pyyhkäisee spektrin läpi ja suotimen piikit sijaitsevat taajuuden ( $1/T_d$ ) monikerroilla, missä  $T_d$  on aikaviive
- Flanger-efektin ”syvyys” on suurimmillaan kun alkuperäisen ja viivästetyn signaalin amplitudit ovat samat
- Flanger aiheuttaa myös äänenkorkeuden modulaatiota viivästettyyn signaaliin, saaden sen kuulostamaan ”heleältä”

Synteesi 48  
DA / Klapuri

### 7.2 Phaser

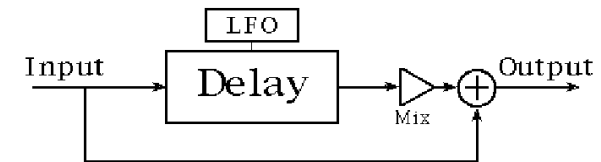
- Tunnetaan myös nimellä *phase shifting*
- Phaser toteutetaan yleensä kytkemällä sarjaan matalataajuuksisia (1. tai 2. asteen) kokopäästö-suotimia
  - kokopäästösuodin: *magnitudivaste on 1* yli koko taajuusalueen
  - kunkin kokopäästösuotimen *vaihesiirtoa moduloi LFO* ja suotimien ulostulot miksataan alkuperäisen signaalin kanssa vahvistuksella  $g$  (yleensä yksikkövahvistus)
- Flangerin ja phaserin pääerot:
  - flanger tuottaa monia harmonisia piikkejä ja nolliä spektriin, kun taas phaser tuottaa vain muutamia (suodintasojen määrä) ei-harmonisia kuoppia joiden syvyys ja leveys voi vaihdella
  - flanger saattaa kumota harmonisen äänen, tästä ei ole vaaraa phaserin tapauksessa
  - käytännössä phaserin kuoppataajuuksia usein siirretään eksponentiaalisesti, eikä LFO:lla kuten flangerissa

## 7.3 Chorus

- Chorus-efektiä käytetään tuottamaan vaikutelma siitä, että soittamassa on useita instrumentteja, vaikka tosiasiasa äänen tuottaa vain yksi instrumentti
- Todellisuudessa kaksi samanlaista instrumenttia ei koskaan kuulosta siltä että ne soittavat täsmälleen samaa nuottia (unisono)
  - pieniä eroja äänenkorkeudessa tarkasta virityksestä huolimatta
  - pieniä eroja synkronoinnissa, aiheuttaen pieniä viiveitä
- Muutokset viiveessä ja äänenkorkeudessa voidaan helposti tuottaa käyttäen vaihtelevan mittaista viivelinjaa
  - chorus-efekti muistuttaa flanger-efektiä

## Chorus

- Kuva 3: chorus-efektin vuokaavio
- Pääerot choruksen ja flangerin välillä:
  - choruksessa ei yleensä käytetä takaisinkytkentää ja
  - viive on yleensä pitempi (20–30 ms) kuin flangerissa
- Viivettä kontrolloi yleensä LFO
  - usein käytetään myös satunnaisviivettä, jotta mallinnetaan paremmin unisonossa soittavia muusikoita
  - myös viivästetyn äänen voimakkuutta voidaan varioida LFO:lla, sillä todellisessa tilanteessa esiintyy myös voimakkuusvaihtelua



## 8 Wah-wah

- Wah-wah efekti voidaan tuottaa joko
  - resonaattorilla (kaistanpäästösuotimella), jonka keskitaajuutta moduloidaan LFO:lla
  - alipäästösuotimella, jonka rajataajuutta moduloidaan LFO:lla
- Auto-wav, toiselta nimeltään "envelope follower"
  - resonaattori, jonka keskitaajuus määräytyy automaattisesti sisääntulevan signaalin amplitudista
- Yleensä wah-wah efektin keski- / rajataajuutta kontrolloidaan (jalka)pedaalilla