

# 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.  
-Zölzer (ed.) (2002) DAFX – Digital Audio Effects. John Wiley & Sons.  
-Jukka Rauhala (2007) "Fysikaaliseen mallinnukseen perustuva äänisynteesi",  
Riitta Väänänen (2005) "3-D Audio and Virtual Acoustics" vierailuluento DA –kurssilla  
-T. Virtanen. (2006) Sound source separation in monaural audio signals, PhD thesis.



## Äänisynteesi

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

## Efektit

- Johdanto
- Kaiunnan mallintaminen ja 3D-audio

## Johdanto

- Äänisynteesin tavoitteena on tuottaa ääniä, jotka
  - ovat musiikillisesti kiinnostavia
  - ovat realistisia (=muistuttavat aitoa soitinta, ei aina tavoitteena)
- Algoritminen kompleksisuus
  - äännet 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

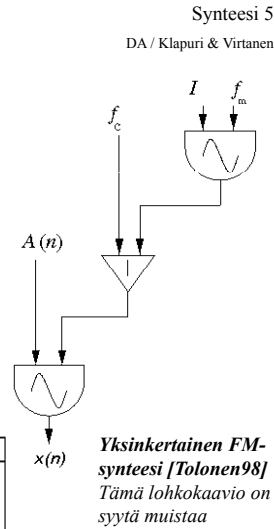
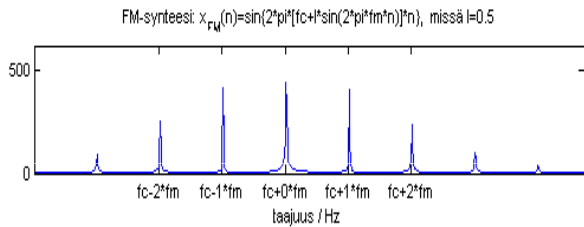
- Abstrakteihin algoritmeihin perustuvalla synteesillä on tyypillistä yksinkertaisuus ja helppo toteutettavuus
  - usein 10 Matlab-koodiriviä riittää
  - dataa ei tarvita muutamaa parametriarvoa enempää

## 1.1 FM–synteesi

- FM (engl. *frequency modulation*) - käytetty tietoliikenteessä esim. radiolähetyksissä jo puoli vuosisataa
- 60-luvun lopulla John Chowning keksi soveltaa FM-synteesiä äänten 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 (erityisesti "Sound Blaster -yhteensopivat" äänikortit)
- FM-synteesin patentti raukesi 1995
  - Menetelmä nykyään käytössä useimmissa syntikoissa ja äänikorteissa muiden synteesimenetelmien ohella

# 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
  - kantoaallon taajuutta  $f_c$  moduloi toinen oskillaattori (modulaatiotaajuus  $f_m$ )
  - modulaatioindeksi  $I$
  - äänen aikariippuva amplitudi  $A(n)$
- Alakuva: syntyvän äänen spektri

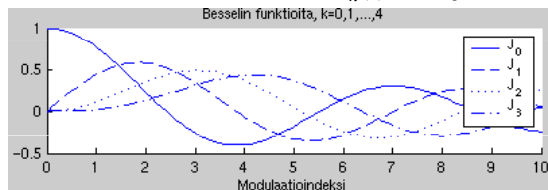


# FM-synteesi

- FM-synteesin kaava voidaan myös kirjoittaa
 
$$x(n) = A(n) \sin[2\pi f_c n + I \sin(2\pi f_m n)]$$
  - 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

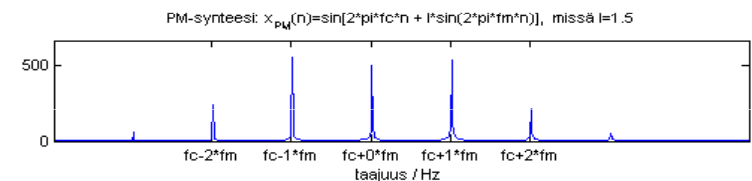
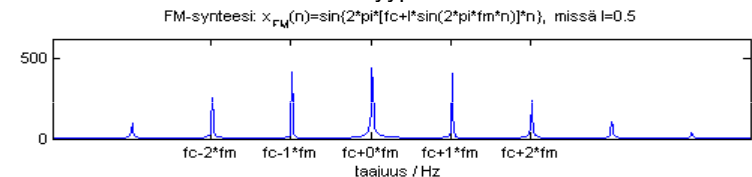
# FM-synteesi

- 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



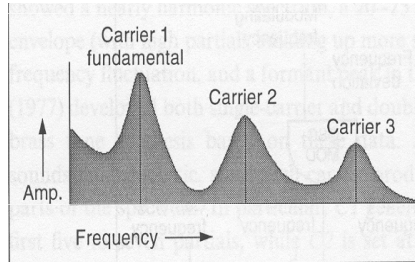
# FM-synteesi

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



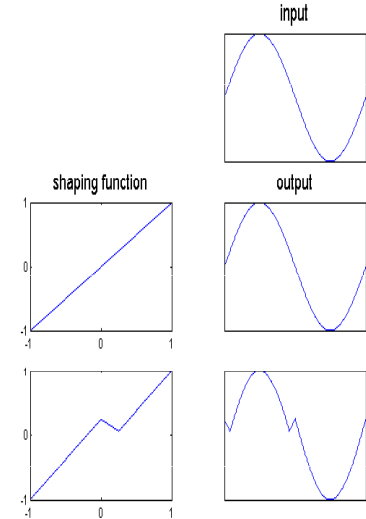
## FM-synteesi – jatkokehitystä

- Yksinkertaisessa FM-synteessissä 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-synteisien äänet
  - kuva [Roads96]



## 1.2 Waveshaping-synteesi

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

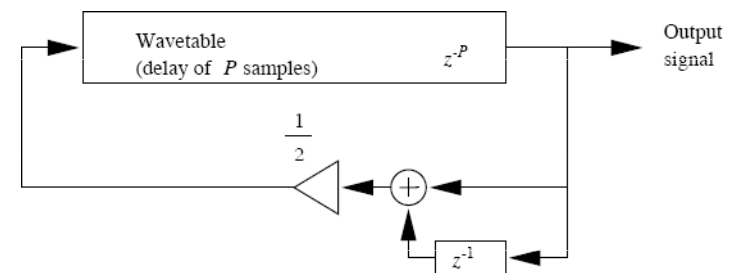


## Waveshaping-synteesi

- Käyttämällä muotoilufunktiona Chebyshevin polynomien lineaarikombinaatiota voidaan säätää harmonisten keskinäisiä suhteita
  - $k$ :n asteen Chebyshevin polynomi muotoilufunktiossa tuottaa sinimuotoiselle herätesignaaliin ulostulon  $\sin(kf)$
  - saadaan hallitusti approksimoitua esim. todellisen instrumentin harmonista rakennetta
- Perustapauksessa herätesignaalina on sini, mutta se voi tietysti olla jokin muukin
  - Menetelmää käytetään yleisesti epälineaarisen särön aikaansaamiseen
- Äänen aikakäyttäytymistä voidaan hallita jälkiprosessoinnilla (esim. amplitudimodulaatio)

## 1.3 Karplus-Strong algoritmi

- Hyvin yksinkertainen ja laskennallisesti tehokas algoritmi
- Sopivilla parametreilla algoritmin tuottamat äänet muistuttavat näpättyä kielisoitinta
- Lohkokaavio:



# Karplus-Strong algoritmi

Synteesi 13  
DA / Klapuri & Virtanen

## ■ Algoritmin toiminta:

1. lyhyt puskuri alustetaan satunnaisilla näytearvoilla
2. puskuri soitetään 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:

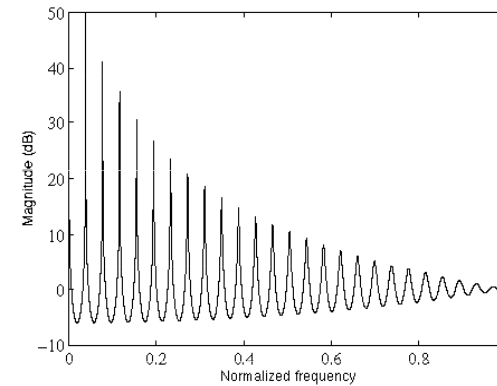
$$H(z) = 0.5(1 + z^{-1})$$

4. palataan kohtaan 2 (soitetaan puskuri yhä uudelleen)

# Karplus-Strong algoritmi

Synteesi 14  
DA / Klapuri & Virtanen

## ■ Kuva: Karplus-Strong mallin taajuusvaste [Tolonen98]



# Karplus-Strong algoritmi

Synteesi 15  
DA / Klapuri & Virtanen

## ■ 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

Synteesi 16  
DA / Klapuri & Virtanen

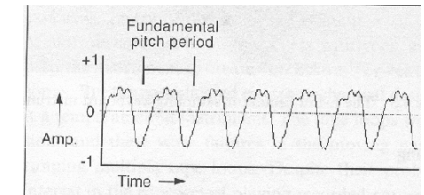
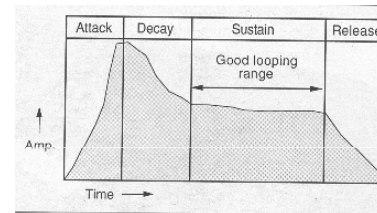
- Engl. *sampling synthesis*
- Laskennallisesti on äärimmäisen yksinkertaista vain soittaa tallennettu ääni
- Erittäin yleinen synteessimenetelmä
- Muistivaatimukset ovat tämän lähestymistavan ongelma
  - ei ole taloudellista tallentaa kaikkia mahdollisia ääniä eri instrumenteilla, eri korkeuksilta, eri soittotavoin
  - näytteistykseen perustuvan synteessin taito on **muistinkulutuksen minimoimisessa ja samalla äänenlaadun maksimoimisessa**
- Perusideoita äänten mallintamiseksi vähemmällä datalla:
  - silmukkapuskuri, äänenkorkeuden siirtäminen, datan reduktio

## Näytteistykseen perustuva synteesi

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

## 2.1 Silmukkapuskuri

- Useimpien instrumenttiäänten steady-state (=sustain) osa on suurinpiirten jaksollinen  
→ voidaan ottaa lyhyt näyte ja soittaa sitä silmukkapuskurista (kuva alla)
- Äänen lyhyt alkutransientti voidaan mallintaa erikseen
- Silmukkapuskurin pituus täytyy olla aallonpituuden monikerta
- Yleensä puskuriin täytyy ottaa enemmän kuin yksi aalto, jotta äänen aikamuuttuvia ominaisuuksia saadaan mukaan



## 2.2 Multiple wavetable -synteesi

- Useita näytempuskureita (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 soitetavista elementaarisisista aaltomuodoista
  - kaupallisissa laitteissa yleensä 4-8 wavetablea pinossa
  - ongelma: löytää joukko elementaarisia aaltomuotoja ja amplitudiverhokäyriä, joilla pystytään tehokkaasti esittämään erilaisia luonnollisia ääniä

## 3 Spektrimallit

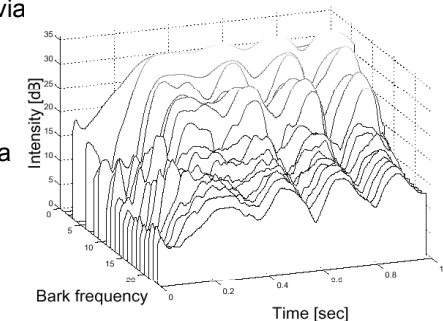
### 3.1 Additiivinen synteesi

- 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 synteeseissä
- Kuva: huilun harmonisten komponenttien ajallinen kehitys [Eronen2000]



## 3.2 Vaihevokooderi

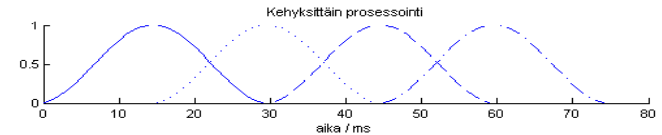
Synteesi 21  
DA / Klapuri & Virtanen

- Engl. *phase vocoder*
  - Keksittiin 60-luvulla puheen kompressiomenetelmiä tutkiessa; vocoder = voice coder
  - *sovelluksia nykyään: äänten aikaskaalaus, äänenkorkeuden muuntelu, äänten 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

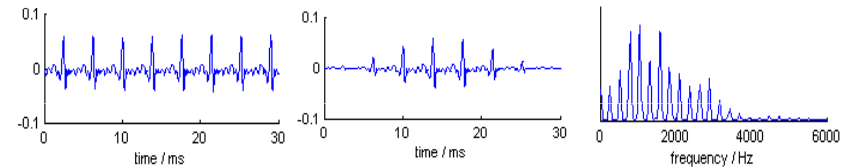
## Vaihevokooderi Kehyksittäin prosessointi

Synteesi 22  
DA / Klapuri & Virtanen

- 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

Synteesi 23  
DA / Klapuri & Virtanen

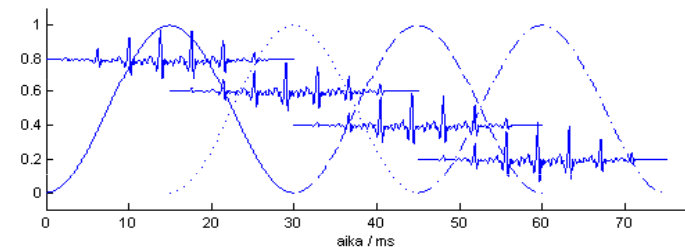
- Edellisen sivun prosessointi tuottaa spektrogrammin
  - aika-taajuus tason esitys: kehyksien kompleksiset spektrin yli ajan
- Miksi yleensäkin ottaen prosessoidaan kehyksittäin?
  - Fourier-m. esittää signaalin *vakiotaajuuksisten* sinien summana
  - todelliset äänisignaalit eivät kuitenkaan ole stationäärisiä, 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

## Aikatason signaalin rekonstruointi: overlap add tekniikka

Synteesi 24  
DA / Klapuri & Virtanen

1. Muunnetaan käänteisellä Fourier muunnoksella ( $DFT^{-1}$ ) kunkin kehyksen spektri takaisin aikataso signaaliksi
2. ikkunoidaan signaali kussakin kehyksessä
3. peräkkäiset kehykset asetetaan 50 % tai enemmän limittäin, ja summataan pisteittäin (sama overlap kuin analyysissä)
4. jos synteesi-ikkuna on valittu oikein (riippuu analyysi-ikkunasta ja overlapista), saadaan täydellinen rekonstruktio alkup. signaalista

Overlap-add: summataan peräkkäisten kehyksien ikkunoidut signaalit



- Vaihevokooderi mahdollistaa äänen keston ja korkeuden muuttelun
- Aikaskaalaus
  - muutetaan kehysten välistä askelta synteessivaiheessa TAI vaihtoehtoisesti tuplataan / jätetään väliin kehyksiä sopivin välein
  - aikatasoinnointi 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
  - äänen 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

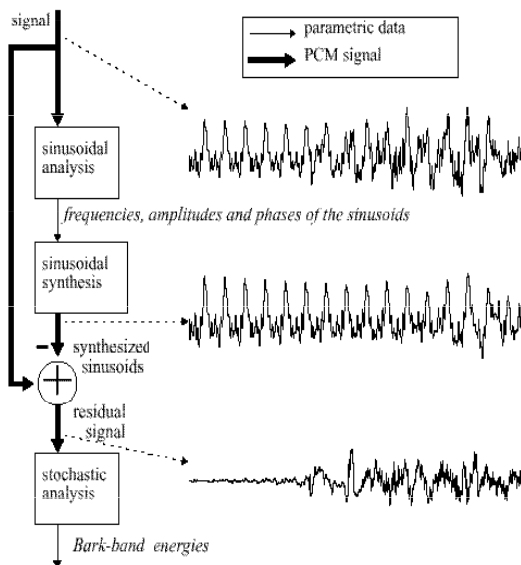
## ■ Additiivinen synteesi

- 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

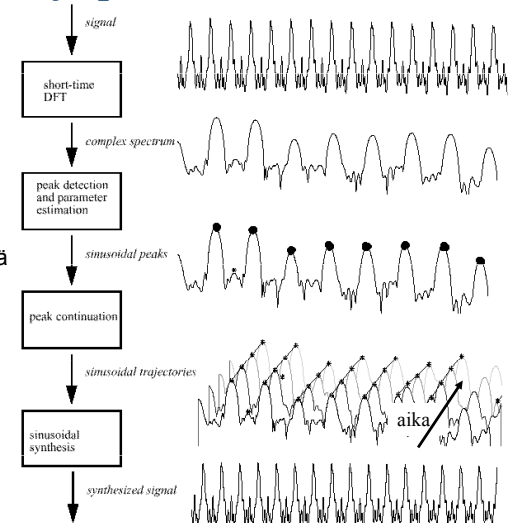
- Vuokaavio [Virtanen 2001]
  1. havaitaan nitit kehyksittäin spektristä
  2. estimoidaan sinien parametrit ja syntetisoidaan ne uudelleen
  3. vähennetään nitit alkup. signaalista
  4. mallinnetaan jäljelle jäävä kohinaresiduaali
- Saadaan
  - sinien parametrit
  - kohinan taso eri taajuuskaistoilla



## Sini+kohina -malli

## Sinien havaitseminen ja parametrit

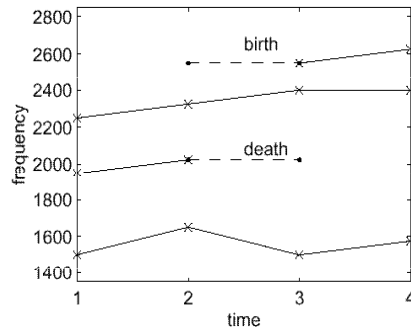
- Vuokaavio: [Virtanen01]
- Spektripiekit tulkitaan siniaalloiksi
  1. "piikki": lokaali maksimi kehyksen magnitudi-spektrissä
  2. piikin taajuus, amplitudi ja vaihe voidaan poimia kompleksisesta spektristä
- Peräkkäisissä kehyksissä havaittujen piikkien yhdistely
  - saadaan ajallisesti jatkuvan sinin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



## Sini+kohina –malli Piikkien yhdistely

Synteesi 29  
DA / Klapuri & Virtanen

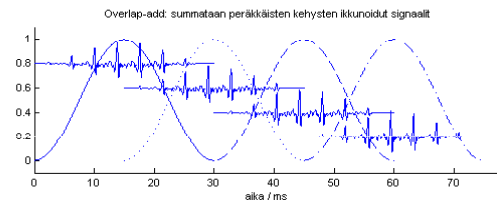
- **Tarvittaessa** voidaan peräkkäisissä kehyksissä havaitut spektriipiikit assosoida, yhdistellä aikamuuttuviksi sineiksi  
→ taajuudelle, amplitudille ja vaiheelle poimitaan arvo kehys
- Kuva: piikin yhdistelyalgoritmi [Virtanen2001]
  - perustuu esim. syntyvän käyrän derivaattoihin; yritetään muodostaa pehmeä käyrä
  - kill: jollei jatkoa löydy, lopetetaan sini
  - birth: jos spektriipiikki ei ole jatkoa millekään entiselle sinille, luodaan uusi



## Sini+kohina –malli Sinien synteesi

Synteesi 30  
DA / Klapuri & Virtanen

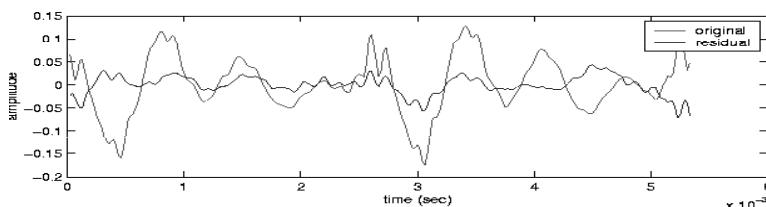
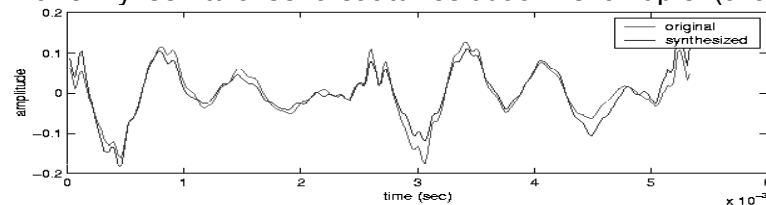
- Additiivinen synteesi
 
$$s(t) = \sum_{n=1}^N a_n(t) \cos[2\pi f_n(t)t + \varphi_n(t)]$$
- Usein edellä esitetty piikkien yhdistely ei ole tarpeen, vaan
  - syntetisoidaan kussakin kehyksessä havaitut sinit erikseen, parametrit pysyvät vakioina koko kehyksen ajan
  - ikkunoidaan saatu aikatason signaali Hanning-ikkunalla
  - overlap-add: peräkkäiset kehykset limittään, summataan pisteittäin



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

Synteesi 31  
DA / Klapuri & Virtanen

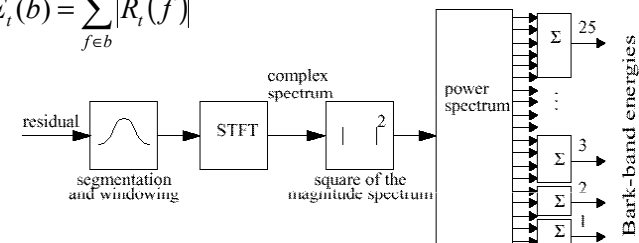
- Syntetisoidut sinit vs. alkuperäinen signaali (yllä)
- Vähennyksen tuloksena saatu residuaali vs. alkup.s. (alla)



## Sini+kohina –malli Kohinaresiduaalin mallinnus

Synteesi 32  
DA / Klapuri & Virtanen

- Residuaalisignaali saadaan vähentämällä syntetisoidut sinit alkuperäisestä signaalista aikatasossa
- Residuaalisignaali analysoidaan kehyksittäin
  - lasketaan residuaalin spektri  $R_t(f)$  kehyksessä  $t$
  - jaetaan spektri kuolon mukaisesti 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

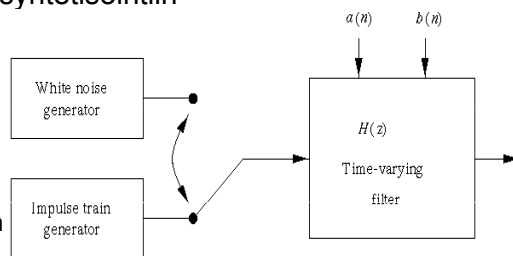
- Kohinaresiduaali esitetään parametreilla
  - tallennetaan kustakin aikakehyksestä vain *lyhytaikaiset energiat kullakin Barkin kaistalla*,  $E_t(b)$
  - tämä mallinnus voidaan tehdä, koska kohinan tapauksessa kuulo ei ole herkkä energiamuutoksille yhden Barkin kaistan sisällä
- Synteessissä  $R_t(f)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



## 3.5 Koodikirjapohjaiset esitysmuodot

- Idea: spektrivektori  $\mathbf{x}_t$  mallinnetaan koodikirjavektorien  $\mathbf{b}_n$  painotettuna summana:
 
$$\mathbf{x}_t \approx \sum_n \mathbf{b}_n a_{nt}$$
- Painot  $a_{nt}$  estimoidaan minimoimalla esitysmuodon rekonstruktio virhe
  - Tyypillisissä sovelluksissa on hyödyllistä rajoittaa painot ei-negatiivisiksi
- Esitystä käytetään yleisimmin teho- tai magnitudispektrille
  - Vaiheiden pois jättäminen mahdollistaa pienet koodikirjat
  - Vaiheet täytyy tallentaa ja syntetisoida erikseen
- Käytössä mm. äänen separoinnissa
  - Kukin koodikirjavektorin on assosioitu tiettyyn äänilähteeseen jolloin kunkin äänilähteen vektorit voidaan syntetisoida erikseen

## 4 Fysikaalinen mallinnus

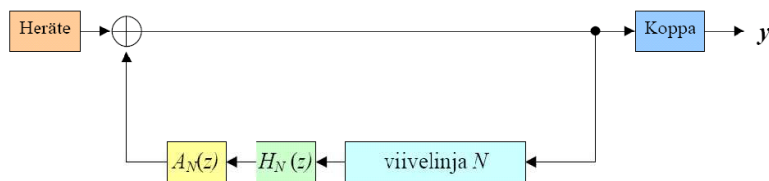
- Uusin synteesimenetelmä
  - 1970-luku: analogi-syntetisaattorien aika
  - 1980-luku: FM-synteesi
  - 1990-luku: fysikaalinen mallinnus
  - elektr. soittimien historiaa: [http://www.obsolete.com/120\\_years/](http://www.obsolete.com/120_years/)
- Simuloi soittimen akustista äänentuottomekanismia
- Käyttö musiikillisissa äänissä
  - jäljitellään olemassa olevia soittimia
  - mahdollistaa myös mielikuvituksellisten ”soittimien” äänittämisen

## Fysikaalinen mallinnus

- Etuja:
  - pieni muistin tarve
  - parametrisyys (ääniä voidaan muokata helposti)
  - Auttaa ymmärtämään soittimen toimintaa: mitkä piirteet äänissä ovat tärkeitä, mikä saa kuulost. hyvältä
- Ongelmia
  - realistisen äänenlaadun saavuttaminen - joissain instrumenteissa on saavutettu erittäin hyvä äänenlaatu (kieli- ja puhallinsoittimet)
  - laskennallinen kompleksisuus
- TKK:n Akustiikan labra (Otaniemi) on kansainvälisesti kovaa tasoa

## Digitaalinen aaltojohtotekniikka

- Käytetyin fysikaalisen mallinnuksen synteesimenetelmä
- Esimerkki: yksinkertainen kitaran kielimalli (vrt. Karplus-Strong algoritmi!)



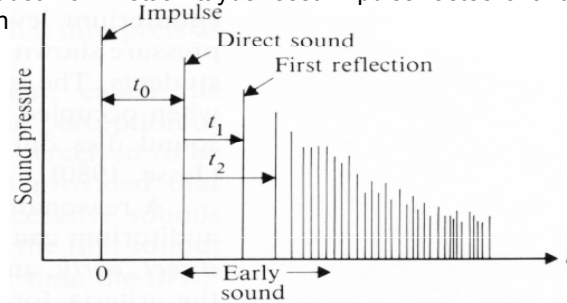
- $H_N(z)$  mallintaa energiahäviöitä kielessä
- $A_N(z)$  mallintaa dispersiota: eri taajuudet kulkevat kielessä eri nopeudella

## 5 Ääniefektit

- Tarkoitus tuottaa äänenväriin vaihtelua ja eloisuutta
  - efektejä kuullaan päivittäin musiikissa radiosta ja CD:itä
  - myös ääniefektit elokuvissa ja mainoksissa
- Esimerkkejä yleisimmistä efekteistä:
  - Dynamiikan hallinta, de-esser
  - Vastekorjaus (jopa pelkkä ali- ja ylipäästösuodatus), wah-wah
  - Flanger, chorus, phaser
  - Vahvistin- ja kaiutinsimulointi, särö
  - Kaiunnan mallintaminen ja muut tilaefektit
  - Äänenkorkeuden- ja keston muuttaminen
- Osa ylläolevista käsitellään omilla luentokerroillaan, seuraavassa esimerkkinä kaiunnan mallintaminen

## 5.1 Kaiunnon mallintaminen

- Hyvin yleisesti käytetty audiotuotannossa
  - Usein äänet nauhoitetaan "kuivana", vähäkaikuisassa tilassa
  - Haluttu tilaefekti voidaan lisä jälkepäin
- Konserttisalin (tai muu huone) voidaan mallintaa lineaarisena aikainvarianttina systeeminä
- Ominaisuudet mallinnetaan täydellisesti impulssivasteella lähteestä mikrofoniin



## 5.2 Konvoluutio impulssivasteella

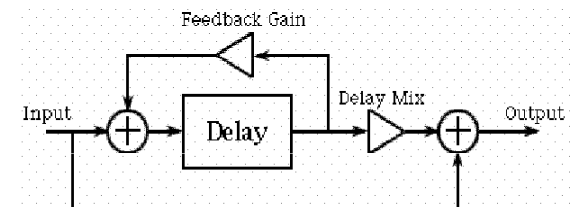
- Realistinen impulssivaste voidaan mitata "oikeasta" tilasta tai mallintaa virtuaalisesta ympäristöstä
- Impulssivaste voi olla useita sekunteja pitkä
  - Suora toteutus konvoluutiolla laskennallisesti erittäin raskas
  - FFT:n avulla toteutettavissa, mutta aiheuttaa viivettä
- Impulssivasteen reaaliaikainen muokkaaminen hankalaa
  - Ei sovellu interaktiivisiin sovelluksiin

## 5.3 Parametrinen kaiunnon mallintaminen

- Suoraa ääntä, varhaisia heijastuksia ja jälkikaiuntaa voidaan mallintaa parametrisesti
- Varhaiset heijastukset: 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 fyysisestä ympäristöstä
- Jälkikaiunta IIR-suodattimilla

## Parametrinen kaiunnon mallintaminen

- Kuva alla: yksinkertainen tasavälisesti viivästyneitä kaikuja tuottava IIR-suoradin
- Takaisinkytkentä alle 1:n vahvistuksella: kaiku toistuu kunnes viivästetyn signaalin amplitudi putoaa taustakohinan tason alapuolelle
- Yhdistelemässä useita allaolevan kaltaisista lohkoja joiden välillä ristiinkytkentää saadaan monipuolinen kaiku



## 5.4 3D-audio

- Halutaan luoda vaikutelma äänistä tietyssä suunnassa
- Kuulon tulosuunnan aistimiseen käyttämät signaalin ominaisuudet: aikaero, intensiteettiero, HRTF-vaste
- Ylläolevia ominaisuuksia voidaan syntetisoida suodattamalla eri korviin menevät signaalit
- Voidaan toteuttaa kuulokkeilla tai kahdella kaiuttimella
  - Kuulokkeissa pään asento pitää ottaa huomioon
  - Kaiutinkuuntelussa vaste kaiuttimista molempiin korviin (crosstalk) pitää huomioida – myös huonevaste vaikuttaa laatuun
- Useammalla kaiuttimella saadaan parempi illuusio kuulijan takana olevista äänilähteistä