

KITARAMUSIIKIN KORKEALAA TUINEN SYNTEESI

Mikael Laurson¹, Vesa Välimäki^{2,3} ja Mika Kuuskankare¹

¹Sibelius-Akatemia, Musiikkiteknologian laitos, PL 86, 00251 Helsinki

²Porin korkeakouluyksikkö, PL 300, 28101 Pori

³TKK, Akustiikan ja äänenkäsittelytekniikan laboratorio, PL 3000, 02015 TKK, Espoo
laurson@siba.fi, vesa.valimaki@hut.fi, mkuuskan@siba.fi

1 JOHDANTO

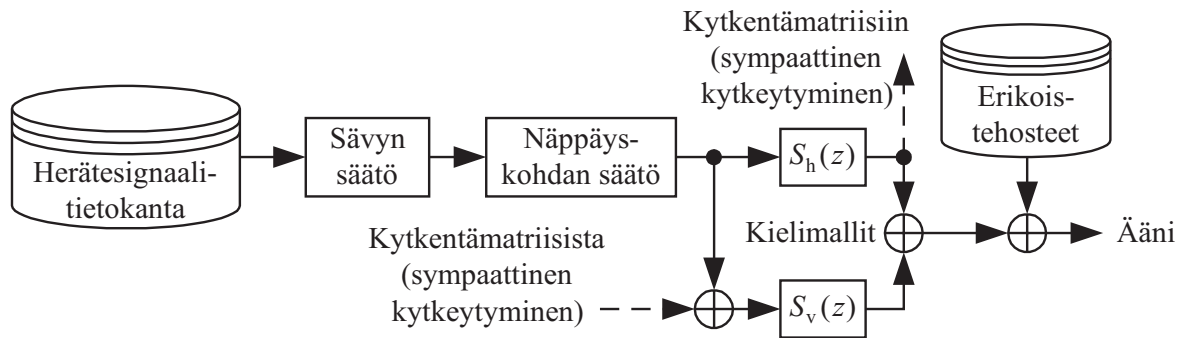
Akustisten soitinten mallinnusta on tutkittu tietokoneen avulla usean vuosikymmenen ajan, Suomessakin jo 10 vuotta [1]. Henkilökohtaisten tietokoneiden kehitys on johtanut siihen, että reaaliaikaisia soitinmalleja voi nyt toteuttaa korkean tason ohjelmointikielillä. Pullonkaula soitinmallien käytännön sovelluksissa on ollut tarvittavien työkalujen tai ohjaimien puute, joilla näitä malleja voisi soittaa. Esimerkiksi kaupalliset MIDI-koskettimistot eivät sovellu kitaramallin ohjaukseen ja MIDI-kitarat ovat osoittautuneet epäluotettaviksi ja hitaiksi, eivätkä näin ollen ole käyttökelpoisia.

Tämä artikkeli on yhteenveto viimeisen kahden vuoden aikana toteutuneen tutkimusprojektin tuloksista. Tutkimushankkeessa on keskitytty näppäilytävien kielisoitinten mallinnukseen. Eri-tyisen kiinnostuksen kohteena on ollut klassinen kitara. Tavoite on ollut mahdollisimman aidolta kuulostavan kitarasynteesin kehittäminen. Hankkeessa on luotu ohjelmistotyökaluja, joilla voidaan generoida ohjaustietoa, joka vuorostaan mahdollistaa synteessimallin soittamisen käyttämällä soittimelle luonteenomaisia soittotekniikoita.

Artikkeli jakautuu viiteen osaan. Ensimmäisessä osassa (kappale 2) kuvataan kitaramalli, jonka pohjana on kommutatiivinen kaksoispolarisaatiokielimalli. Toisessa osassa (kappale 3) kerrotaan kuinka kaiuttomassa huoneessa äänitetyt kitaransoittonäytteet muokataan erityisen analyysi- ja kalibrintiohjelman avulla sopiviksi signaaleiksi, jotka toimivat kielimallin herätteinä. Samalla määritetään myös parametriarvot kielimallin digitaalisuotimia varten. Kolmannessa osassa (kappale 4) esitellään kitarasyntetisaattorin ohjauspuoli. Pääpaino on laajennetussa nuotinnusohjelmassa nimeltä ENP, jonka avulla käyttäjä voi antaa soittotapah- tuman perusmateriaalin (säveltasot ja rytmit) ja soitinkohtaiset lisämääreet, jotka kuvaavat tarkasti kuinka soitinmallin tulee käyttäytyä soiton aikana. Seuraavaksi (kappale 5) esitellään kuinka kitarasyntetisaattori on toteutettu PWSynth-nimisen graafisen synteesiohjelman avulla. Viimeisessä osassa (kappale 6) kerrotaan esimerkkien avulla, kuinka kitaralle ominaisia soit- totekniikoita, kuten vasemman käden legato ja pizzicato, voidaan toteuttaa kitarasyntetisaatto- rin avulla.

2 KITARAMALLI

Käytämme värähtelevän kitaran kielen mallinnukseen menetelmää, jota kutsutaan kommu- toiduksi aaltojohtomalliksi [1], [2], [3], [4], [5]. Siinä kielen poikittaisen värähtelyn etene- mistä ja vaimenemista kielessä jäljitellään takaisinkytkentäsilmukalla, jossa on sävelen jak- sonpituutta vastaava viivelinja ja alipäästösuodin, jolla säädetään äänen vaimenemisomina- suuksia. Kielimalli on yksinkertainen ja laskennallisesti tehokas signaalinkäsittelyjärjestelmä, jonka toteutuksessa tarvitaan vain muutamia kerto- ja yhteenlaskuja yhtä signaalinäytettä



Kuva 1. Kitarasynteesimallin lohkokaavio yhden kielen osalta.

kohti. Siinä käytetty alipäästösuodin on ensimmäisen asteen rekursiivinen suodin, jonka parametrit ovat vahvistus ja alipäästön rajataajuutta säätelevä takaisinkytkentäkerroin.

Kitaran kielen poikittaisella värähtelyllä on 2 polarisaatiota, kaikukopan kannen suuntainen ja siihen nähden kohtisuora. Värähtelyn polarisaatio vaihtelee kaiken aikaa. Tämä ilmiö voidaan mallintaa kahdella rinnakkaisella kielimallilla, jotka kytketään toisiinsa. Käytännössä kannattaa syöttää signaali vain yhteen suuntaan, jotta mallissa ei syntyisi ylimääräistä takaisinkytkentää, joka voi aiheuttaa stabiilisuusongelmia. Hyvä ratkaisu on syöttää kaikkien horisontaalisten kielimallien lähtösignaali takaisinkytkentämatriisiin, jolla tuotetaan samalla kielten välinen sympaattinen kytkeytyminen [4]. Matriisista syötetään signaali kaikkiin pystypolarisaatiota mallintaviin kielimalleihin. Tuloksena on lähes luonnonmukainen kytkeytyminen, mutta stabiilisuus on taattu: kytkeytymiskertoimet voivat olla miten suuria tahansa. Käytännössä valitaan tietenkin luonnolliselta kuulostavat kytkentäkertoimien arvot.

Koko kitaramalli koostuu kuudesta tällaisesta kaksoispolarisaatiokielimallista. Kuvassa 1 näytetään kitaramallin lohkokaavio yhden kielen osalta. Mallin herätesignaali poimitaan tietokannasta. Se suodatetaan kahdella digitaalisuotimella, joista ensimmäinen on ali- tai ylipäästösuodin ja toinen FIR-kampasuodin, jolla voidaan tuottaa näppäyskohdan paikasta johtuva äänenvärimuutos. Näin prosessoitu herätesignaali syötetään kahteen rinnakkaiseen kielimalliin, $S_h(z)$ ja $S_v(z)$, joiden lähtösignaalien summa on lopullinen synteettinen ääni. Siihen voidaan vielä lisätä erikoistehosteita, kuten kopan koputuksia tai muita soittajan aikaansaamia ääniä, joita käytetään paljon modernissa musiikissa [6].

3 HERÄTESIGNAALIEN JA PARAMETRIEN KALIBROINTI

Herätesignaalit ja syntetisaattorin parametrien arvot täytyy mitata oikeasta soitosta, jotta voidaan tuottaa korkealaatuista äänisynteesiä. Kitaransoittoa äänitetään kaiuttomassa huoneessa, ja saadut näytteet analysoidaan perustaaajuustahdistetulla Fourier-analyysillä. Tuloksena saadaan kaikkien ylä-äänesten verhokäyrä ajan funktiona. Verhokäyristä mitataan äänen vaimenemisnopeus taajuuden funktiona, mihin voidaan sovittaa kielimallin alipäästösuotimen parametrit.

Herätesignaalin määrittämisessä tavoitteena on poistaa äänitetystä kitarasignaalista ylä-äänekset, jolloin jäljelle jää kitaran näppäilytahtumalle ominainen lyhyt hälymäinen raapaisu. Kun prosessoitu heräte syötetään kielimalliin, se rekonstruoi ylä-äänekset. Ylä-äänekset voidaan poistaa parilla vaihtoehtoisella tavalla. Nykyisin käytämme menetelmää, jossa ylä-ääneksille laaditaan sinimallit, jotka vähennetään alkuperäisestä äänitteestä [5], [7]. Herä-

tesignaaleihin on vielä lisättävä korjaussignaali, jolla säädetään ylä-äänesten amplitudin alkuarvo.

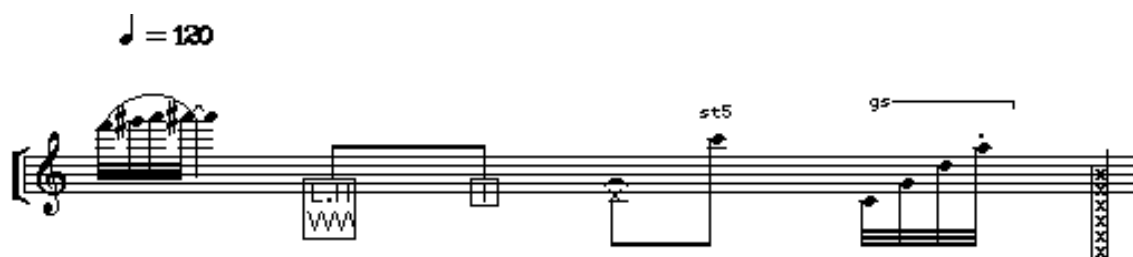
Kitaraäänitteet ovat kohinaisia, mikä häiritsee parametrien estimointia ja voi johtaa virheelliseen harmonisten vaimenemisnopeuteen. Parannusehdotukseksi on kehitetty iteratiivinen menetelmä, jossa estimointirytyksen jälkeen tuotetaan synteettinen signaali saaduilla parametrin arvoilla ja sen vaimenemisominaisuudet analysoidaan [5]. Tämän jälkeen tehdään korjaus, jolla vaimeneminen saatetaan lähemmäksi alkuperäisen äänitteen keskimääräistä vaimenemisnopeutta. Menetelmä lisää automaattisen parametriestimoinnin luotettavuutta.

4 SOITINMALLIN OHJAUS

Soitinmallia ohjataan tietokoneessa toteutetun nuotinkirjoitusohjelman avulla. Ohjelman englanninkielinen nimi on “Expressive Notation Package” tai lyhennettynä ENP [8]. ENP on toteutettu PatchWork-ohjelmointiympäristössä [9]. PatchWork on visuaalinen ohjelmointikieli tietokoneavusteista sävellystä varten ja se on toteutettu Lispillä ja CLOSilla (Common Lisp Object System). Perinteisen nuotinkirjoituksen käyttö mahdollistaa soitinmallin tarkan ohjauksen. Lisäksi musiikkinotaatio on tuttu suurelle joukolle muusikoita, joten nyt soitinmallin ohjausinformaation voi syöttää musiikin ammattilainen, jolla ei tarvitse olla koulutusta tietokoneohjelmoinnista. Tämä mahdollistaa yhteistyön, jossa voidaan yhdistää paras tietämys sekä teknisiltä että musiikillisilta osa-alueilta.

Käyttäjä syöttää halutun musiikillisen materiaalin tietokoneeseen ENP:n avulla käyttämällä normaalia nuotinkirjoitusta. ENP:ssä on graafinen käyttöliittymä eikä siinä edellytetä tekstimuotoista tiedon syöttöä. Käyttäjä voi lisätä partituuriin sekä yleisesti käytettyjä nyanssimerkintöjä että erityismerkintöjä, jotka mahdollistavat soitinkohtaisten soittotapahtumien määrittelyn suurella tarkkuudella. Määrittelyt voidaan kohdistaa yksittäisille nuoteille tai nuottiryhmälle. Ryhmät voivat sijaita limittäin ja ne voivat sisältää olioita, kuten esimerkiksi murtovivafunktioita. Makromäärittelyt tuottavat soiton yhteydessä lisänuotteja. Tällä tavoin voidaan toteuttaa kitaramusiikille tyypillisiä soittotapahtumia, kuten tremolo, trilli, portamento sekä rasgueado (flamenco-musiikin voimallinen soittotapa, jossa kieliä soitetaan nopeasti oikean käden kynnen selällä).

Nuottien ajoitus hienosäädetään ENP:ssä graafisten tempofunktioiden avulla. Tempomuutokset määritellään valitsemalla partituurista ryhmä nuotteja, joihin käyttäjä liittää tempofunktion. Tempokäyrää voidaan editoida hiiren avulla. ENP:ssä voidaan laatia esityssääntöjä, jotka mahdollistavat erilaisten musiikillisten parametrien, kuten esimerkiksi dynamiikan tai nuottien alku- ja loppuhetkien, automaattisen muokkauksen.



Kuva 2. ENP-esimerkki modernin kitaramusiikin alueelta.

Kuvassa 2 on ote Juan Antonio Muron kitarakappaleesta ”Lettera Amorosa” ENP:llä kirjoitettuna. Esimerkki sisältää tavallisten nuottisymbolien lisäksi useita esikoismerkintöjä, jotka kuvaavat kuinka soittajan tulee toteuttaa poikkeuksellisia soittotapahtumia, kuten kielten hankaus ja erilaisia koputuksia.

Kun partituuri on syötetty lisämääreiden kera, se käännetään ohjaustiedoksi. Tämä toteutetaan kahdessa vaiheessa. Aluksi partituurissa annettu nuottitieto muokataan tempofunktioiden ja esityssääntöjen mukaan. Tämän jälkeen kaikki nuotit ajastetaan. Kun ajastin on käynnissä, jokainen nuotti lähettää viestin nuotin instrumenttioliolle, joka vuorostaan käynnistää muita prosesseja, jotka generoivat lopullisen ohjaustiedon. Nämä prosessit luovat joko diskreettejä ohjaustapahtumia, kuten herätetietoja, tai jatkuvaa ohjaustietoa, kuten suodinkertoimia, skaalausarvoja sekä muita matalan tason parametrejä.

5 KITARASYNTEETISAATTORIN TOTEUTUS

Kappaleessa 2 kuvattu kitaramalli on toteutettu PatchWorkissa sijaitsevaan PWSynth-nimiseen visuaaliseen äänisynteesikieleen [10]. Kuten PatchWork, PWSynth on toteutettu Lispillä ja CLOSilla. Nopeutta vaativat alirutiinit on ohjelmoitu C-kielellä.

Yksinkertaisissa tapauksissa PWSynth muistuttaa monia muita äänisynteesiympäristöjä, joissa käyttäjä määrittelee halutun lopputuloksen visuaalisesti laatikoiden ja niiden välisten kytkentöjen avulla. PWSynth-laatikot ovat muuten tavallisia PatchWork-laatikoita, paitsi että ne sisältävät kukin C-struktuurin. C-strukturit toimivat rajapintana Lisp-kielen ja C-kielisten alirutiinien välillä. Lisp kääntää PWSynth-ikkunan puurakenteeksi, joka koostuu C-struktuureista. Kaikki rakenteet alustetaan annetuilla alkuarvoilla Lispin toimesta. Kun puurakenne on valmis, PWSynth käynnistää C-kielisen päärutiinin, joka alkaa tuottaa ääntä tietokoneen DAMuuntimien kautta.

PWSynth tukee sisäkkäisiä C-struktuureja eli mikä tahansa C-struktuuri voi sisältää muita C-struktuureja. Tämä piirre on käytännöllinen monimutkaisia soitinmalleja laadittaessa. Edellä kuvattu kitaramalli on toteutettu C-struktuurin avulla, joka kuvaa koko soitinta. Tämä pää rakenne sisältää seitsemän C-alistruktuuria, jotka vastaavat kuutta kaksoispolarisaatiokieltä ja yhtä takaisinkytkentämatriisia. Kaksoispolarisaatiokieli koostuu yhdeksästä C-struktuurista, jotka sisältävät kielimallissa käytetyt samplesoittajat, digitaalisuotimet ja viiveet.

6 SOITTOTEKNIKOIDEN TOTEUTUS

Esittelemme seuraavaksi joitakin ideoita kuinka klassisen kitaristin käyttämiä soittotekniikoita voidaan realisoida ENP:n avulla. Aloitamme yleisimmistä tekniikoista, kuten näppäily, nopea uudelleennäppäily, soivan kielen kuoleetus sekä vasemman käden legatotekniikka. Lopuksi siirrymme tekniikoihin, joissa soittaja muuttaa säveltäsoa, ja mainitsemme lyhyesti kuinka forte, piano sekä pizzicato voidaan toteuttaa.

6.1 Yksinkertainen näppäily

Tavallista näppäilyä simuloidaan seuraavasti. Ohjelma valitsee partituurissa olevan nuotin perusteella tietokannasta sopivan herätteen, perustaaajuuden ja muita ohjausparametrejä. Sekä näppäilyvoimakkuus että -paikka luetaan nuotista. Kun kaikki uudet parametrit on kirjoitettu asianmukaisesti C-struktuureihin, samplesoittaja liipaisee herätteen, joka saa kielimallin värähtelemään halutulla taajuudella.

6.2 Uudelleennäppäily ja kuoletus

Juuri ennen kuin kieltä näppäillään uudestaan, kielimallin silmukkasuotimen vahvistus pienennetään nopeasti kohti nollaa verhokäyräfunktion avulla. Tämä vaihe kestää kielestä riippuen noin 10 ms. Jos kieli vielä värähtelee uudelleennäppäilyn hetkellä, ohjelma lähettää ylimääräisen hyvin hiljaisen herätteen kielimalliin [6]. Tämän tarkoitus on simuloida oikean käden kynnen osumista kieleen uuden näppäilyn yhteydessä. Samankaltaista tekniikkaa käytetään myös staccatosoiton simuloinnissa.

6.3 Vasemman käden legato ja portamento

Vasemman käden legatotekniikka toteutetaan syöttämällä normaali herätesignaali hyvin hiljaa kielimalliin ja vaihtamalla samalla perustaajuutta [6]. Tällöin silmukkasuotimen vahvistusta ei muuteta. Portamento-efekti voidaan simuloida helposti erittäin nopealla kahden nuotin välisellä kromaattisella asteikolla. Tällöin näppäillään ainoastaan ensimmäinen nuotti ja kaikki muut nuotit tuotetaan käyttämällä vasemman käden legatotekniikkaa.

6.4 Glissando ja vibrato

Ylimääräinen säveltasoinformaatio on toteutettu ENP:ssä skaalausarvon avulla. Jos nuotin yhteydessä ei ole glissandoa eikä vibratoa, sen skaalausarvo on 1.0. Jos taas kahden nuotin väliin on merkitty glissando, skaalausarvoa ohjataan vaiheittain verhokäyrän avulla, jotta ensimmäisen nuotin lopussa saavutetaan seuraavan nuotin säveltasoa.

Vibrato toteutetaan moduloimalla säveltasoskaalausarvoa sinifunktion avulla. Vibraton maksimisyvyys määräytyy toisaalta sormituksen ja toisaalta nuotissa sijaitsevan vibratoparametrin avulla. Kun nuotti soitetaan avoimelta kieleltä, vibratoa ei käytetä lainkaan. Jos taas nuotti soitetaan painamalla vasemman käden sormella kitaran otelautaa, modulaation syvyys on suoraan verrannollinen nauhan järjestyslukuun: mitä korkeampi nauha, sitä suurempi vibratovoimakkuus. Tällä pyritään simuloimaan sitä tosiseikkaa, että soittajan on vaikeampi kontrolloida säveltason vakautta korkeammassa asemassa ja helpompi alemmissa, koska kieli on löysempi kielen keskivaiheilla. Kaikissa näissä tapauksissa vibrato on kuitenkin melko vähäinen. Varsinainen vibrato määräytyy nuotissa annettavien erityismerkintöjen avulla. Tällä tavoin käyttäjä voi ilmoittaa ENP:ssä halutun vibratovoimakkuuden joka nuotille erikseen. Vibraton nopeus on noin 5 Hz ja se on yleensä nuotin aikana vakio.

6.5 Dynamiikka

Dynamiikka toteutetaan säätelemällä herätesignaalin voimakkuutta. Piano-soiton (hiljaa) yhteydessä herätesignaalia alipäästösuodatetaan voimakkaammin. Tästä johtuen kitaramallin ääni on tummempi hiljaisilla nuoteilla. Forte-soitossa lähtösäveltaso voidaan hiukan nostaa nuotin alussa ja heti näppäilyn jälkeen säveltasoa palautuu liukumalla normaalitasoon. Tällä voidaan simuloida kireysmodulaatiota eli kielen venymistä voimakkaan näppäilyn yhteydessä.

6.6 Pizzicato

Pizzicato-tekniikka toteutetaan vähentämällä silmukkasuotimen vahvistuskerrointa sekä alipäästön rajataajuutta. Parempaa simulaatiota varten tarvittaisiin joukko pizzicato-herätesignaaleja. Yksi vaihtoehto on alipäästösuodattaa normaali herätesignaali käyttämällä korkeasteista suodinta.

7 YHTEENVETO

Tässä artikkelissa tarkasteltiin, kuinka korkeatasoista kitaransoittoa voidaan tuottaa tietokoneen avulla. Tulokset on saatu TKK:n ja Sibelius-Akatemian tutkijoiden yhteistyönä. Synteettinen äänentuotto perustuu värähtelevän kielen malliin, jonka parametrit ja herätesignaali saadaan kaiuttomassa huoneessa tehdyistä kitaräänitteistä. Tähän käytetään TKK:n akustiikan ja äänenkäsittelytekniikan laboratoriossa kehitettyä analyysi- ja kalibrointiohjelmaa. Kitaramallin ohjausohjelmat ja musiikkiesimerkit on toteutettu Sibelius-Akatemiassa. Tietokoneella tuotettua kitaramusiikkia voi kuunnella Internet-osoitteessa:

<http://www.siba.fi/soundingscore>

KIITOKSET

Tämä hanke on toteutettu Suomen Akatemian rahoituksen avulla. Kirjoittajat haluavat kiittää Cumhur Erkutia, joka on laatinut analyysi- ja kalibrointiohjelmiston.

LÄHTEET

1. VÄLIMÄKI V & KARJALAINEN M, Soittimien laskennallinen mallintaminen. *Akustiikkapäivä 1993*, 20.10.1993, Helsinki, 53–60.
2. KARJALAINEN M, VÄLIMÄKI V & JÁNOSY Z, Towards high-quality sound synthesis of the guitar and string instruments. *Proc Int Computer Music Conf*, 10.–15.9.1993, Tokio, Japani, 56–63.
3. SMITH J O, Efficient synthesis of stringed musical instruments. *Proc Int Computer Music Conf*, 10.–15.9.1993, Tokio, Japani, 64–71.
4. KARJALAINEN M, VÄLIMÄKI V & TOLONEN T, Plucked-string models: from the Karplus–Strong algorithm to digital waveguides and beyond. *Computer Music J* **22**(1998)3, 17–32.
5. ERKUT C, VÄLIMÄKI V, KARJALAINEN M & LAURSON M, Extraction of physical and expressive parameters for model-based sound synthesis of the classical guitar, *AES 108th Convention*, 19.–22.2.2000, Pariisi, Ranska, preprint 5114.
6. LAURSON M, ERKUT C, VÄLIMÄKI V & KUUSKANKARE M, Methods for modeling realistic playing in acoustic guitar synthesis. *Computer Music J* **25**(2001)3.
7. TOLONEN T & VÄLIMÄKI V, Akustisen kitaran äänisynteesi. *Akustiikkapäivät 1997*, 8.–9.10.1997, Espoo, 107–112.
8. KUUSKANKARE M & LAURSON M, Expressive Notation Package (ENP), a tool for creating complex musical output. *Proc Les Journées d'Informatique Musicale*, 2000, Bordeaux, Ranska, 49–56.
9. LAURSON M, *PATCHWORK: A Visual Programming Language and Some Musical Applications*. Väitöskirja, Sibelius-Akatemia, Helsinki 1996.
10. LAURSON M, HIIPAKKA J, ERKUT C, KARJALAINEN M, VÄLIMÄKI V & KUUSKANKARE M, From expressive notation to model-based sound synthesis: a case study of the acoustic guitar. *Proc Int Computer Music Conf*, 22.–28.10.1999, Peking, Kiina, 1–4.