Kurs i Max/MSP - uppgifter

Henrik Frisk
Doctorate student
Malmö Academy of Music - Lund University
mail@henrikfrisk.com http://www.henrikfrisk.com/

9 november 2004

1 Uppgift 1 - synthesizer

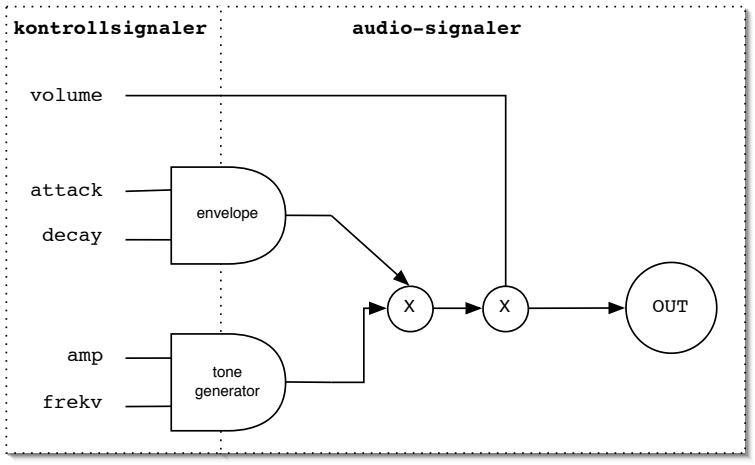
Bygg en synthesizer i Max/MSP. Du kan välja vilken syntesmetod som helst och den kan vara monofonisk. Förslagsvis börjar du med att bygga synten med ett `cycle~` objekt (en ren sinustongenerator).

Du *ska* ha följande kontrollsignaler:

- Volym
- Envelope attack och decay
- Amplitud
- Frekvens

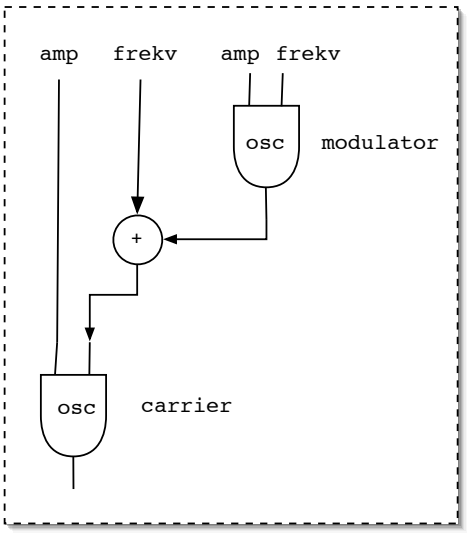
Hur dessa kontrolleras är upp till dig själv men man ska kunna spela olika toner/ljud på synten i Max. I figur 1 finns en schematisk uppställning som du kan utgå ifrån. För själva syntesen

Figur 1: Schema över enkel synthesizer



kan du sedan gå vidare med att titta på FM-syntes. Enklast möjliga FM-syntes kan beskrivas som i figur 2. Denna finns implementerad i en Max/MSP patch som du kan ladda ner på min hemsida (FM synthesis.pat). Så länge förhållandet mellan modulator frekvensen och carrier frekvensen bibehåller en heltals ratio (1:1, 1:2, 1:3, 1:4, ...) så blir resultatet ett harmoniskt ljud.

Figur 2: Schema över enkel FM syntes



Andra objekt av intresse:

- `line~` (för envelope)
- `phasor~`
- `triangle~`
- `wave~`
- `mtof~`

2 Uppgift 2 - sampler

Bygg en liten sampler (monofon) som kan ladda in en sampling och spela upp den i olika transponeringar. Använd samma envelope som i uppgift 1 och lägg till möjligheten att ändra envelopen på loopade samplingar. Den generella strukturen är samma som i figur 1, men ljudkällan är istället en sampling.

Titta på dessa objekt:

- `buffer~`
- `play~`
- `groove~`
- `index~`

3 Uppgift 3 - audioeffekt

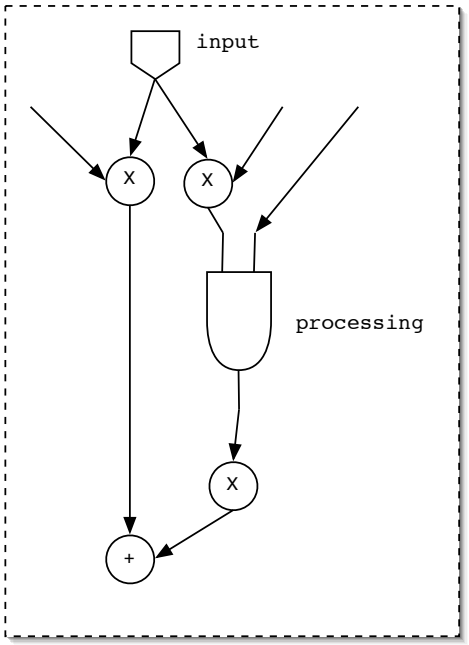
Bygg en enkel audioeffekt som tar ljud från en ljudfil eller från en `adc~`. Figur 3 visar en enkel schematisk uppställning över signalvägarna. Man ska kunna kontrollera balansen mellan effekten och det rena ljudet.

Börja med att titta igenom dokumentationen på dessa objekt:

- `tapin~, tapout~`

- `comb~`
- `lores~`
- `onepole~`
- `stutter~`
- `teeth~`

Figur 3: Schema över en enkel effektenhet



4 Inkapsling och återanvändning

En av de stora fördelarna med Max/MSP är att man kan skapa små patcher med en specifik uppgift och sedan återanvända dessa när man behöver dem i ett annat sammanhang. Detta kräver dock att man tänker igenom sitt arbete ordentligt och försöker dela in patchen i olika delar. Var och en av dessa delar måste ha ett tydligt och helst enhetligt sätt att ta emot meddelanden och skicka vidare meddelanden. Detta kallas på programmeringsspråk för *objektorientering*.

Rent praktiskt fungerar det så i Max/MSP att du sparar en patch och ger den ett namn. Öppna en ny patch och spara denna i samma mapp. Nu kan du skapa en instans av den första patchen genom att skriva in namnet på den i ett objekt i den nya patchen. De kan kommunicera med varandra genom `inlet, outlet` eller genom `send, receive, send~, receive~`.

Objektet `poly~` kan man använda för att dynamiskt skapa flera instanser (röster) av en patch. Om du har tänkt igenom designen på dessa tre övningarna så blir det med hjälp av `poly~` ganska enkelt att skapa multitimbrala versioner av alla tre övningarna.