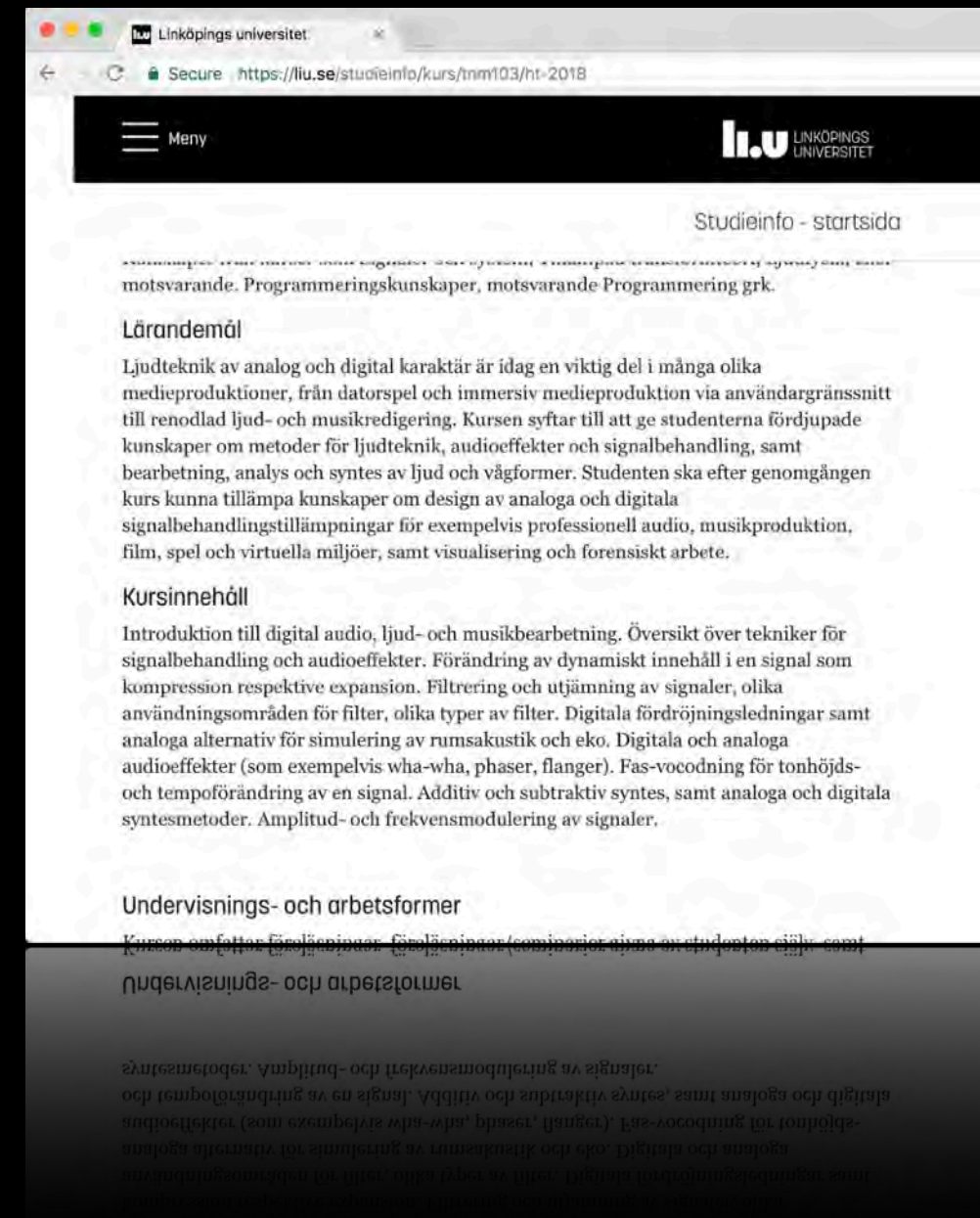


TNM103 - LJUDTEKNIK

INTRODUKTION

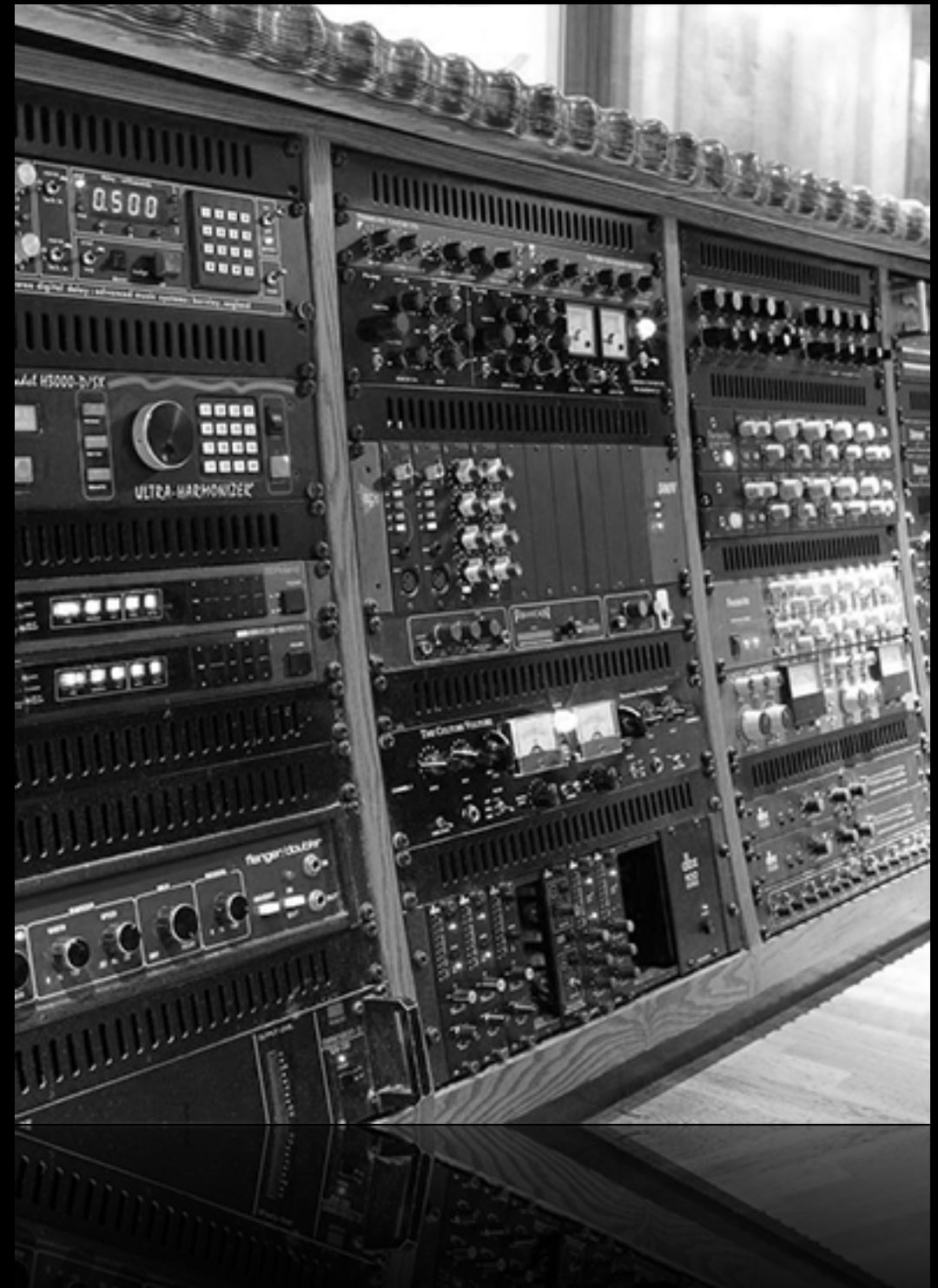
TNM103 - LJUDTEKNIK

- Introduktion till digital audio, ljudbearbetning och datormusik
- Översikt över tekniker för signalbehandling
- Filtrering och utjämning
- Digitala fördröjningsledningar
- Digitala audioeffekter
- Additiv och subtraktiv syntes
- Amplitud- och frekvensmodulering



TNM103 - LJUDTEKNIK

- Grunderna för att tänka ljudeffekter
- Se kopplingar till tidigare kurser
 - Ljudfysik
 - Signaler & system
 - Digitala medier
 - Programmering
- ...



TNM103 - LJUDTEKNIK

- Introduktion till ljudteknik
- Inspelning, mikrofoner etc
- Mix, filter, dynamik
- Rum, reverb, delay, etc
- Ljudsyntes
- Programmering/Matlab
- Arduino



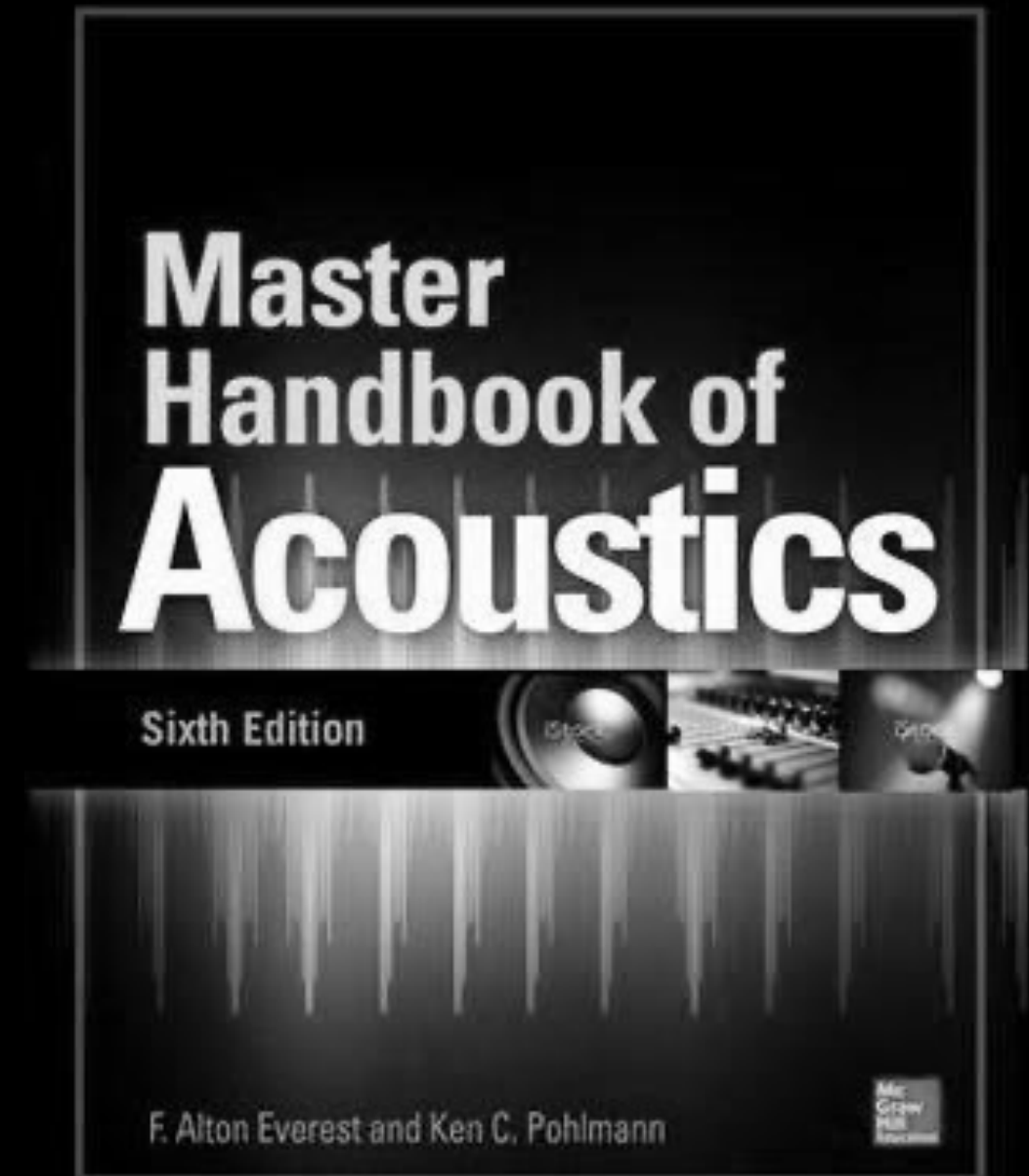
TNM103 - LJUDTEKNIK

- Kurswebb

<http://weber.itn.liu.se/~nikro27/tnm103-2022/>

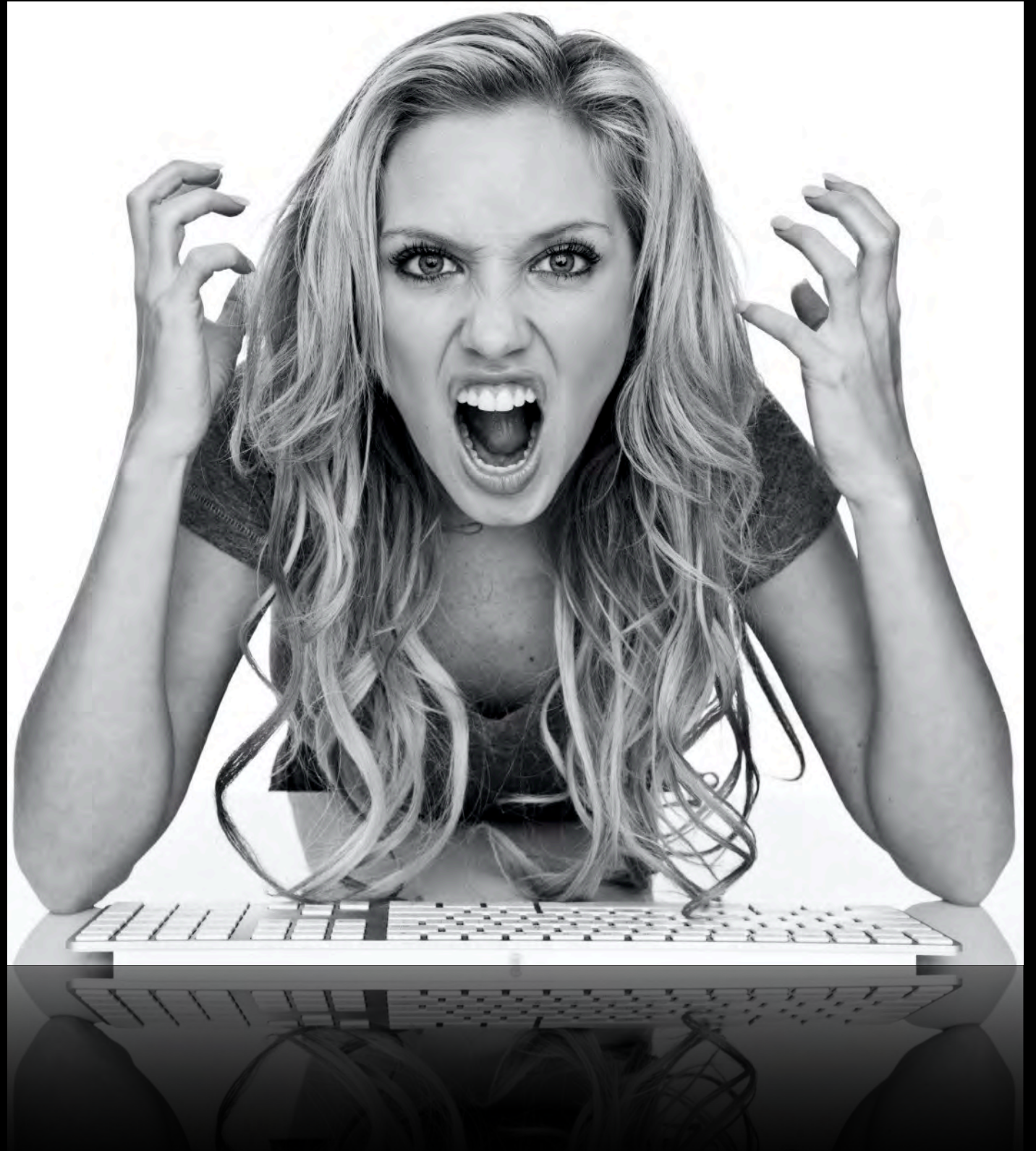
TNM103 - LJUDTEKNIK

- Kurslitteratur
- Det finns böcker men...
- Kurswebben har länkar, föreläsningsslides laborationshandledningar
- Det kommer att komma litteraturkrav till de föreläsningar ni håller.



TNM103 - KURSUTVÄRDERING

- Kursen har fått bra kursutvärderingar...
- Laborationerna och kodmiljö
- Laboration 9 utökas
- Föreläsningsslides före föreläsningar
- ...



TNM103 - LJUDTEKNIK

- Matlab
- Arduino
- SuperCollider
- Mjukvara
- Hårdvara
- Lyssna
- Ljudexempel



TRE GÄSTFÖRELÄSNINGAR

- Rikard Lindell från MDH -
programmering loopapp
- Jacopo från Softube -
programmering/signalbehandling
- Dan Nyberg från NFC -
forensisk ljudanalys
- ...



NI HÅLLER TVÅ "FÖRELÄSNINGAR"

- Måndag 19 september
- Tisdag 20 september
- Alla förväntas närvara...



ERA "FÖRELÄSNINGAR"

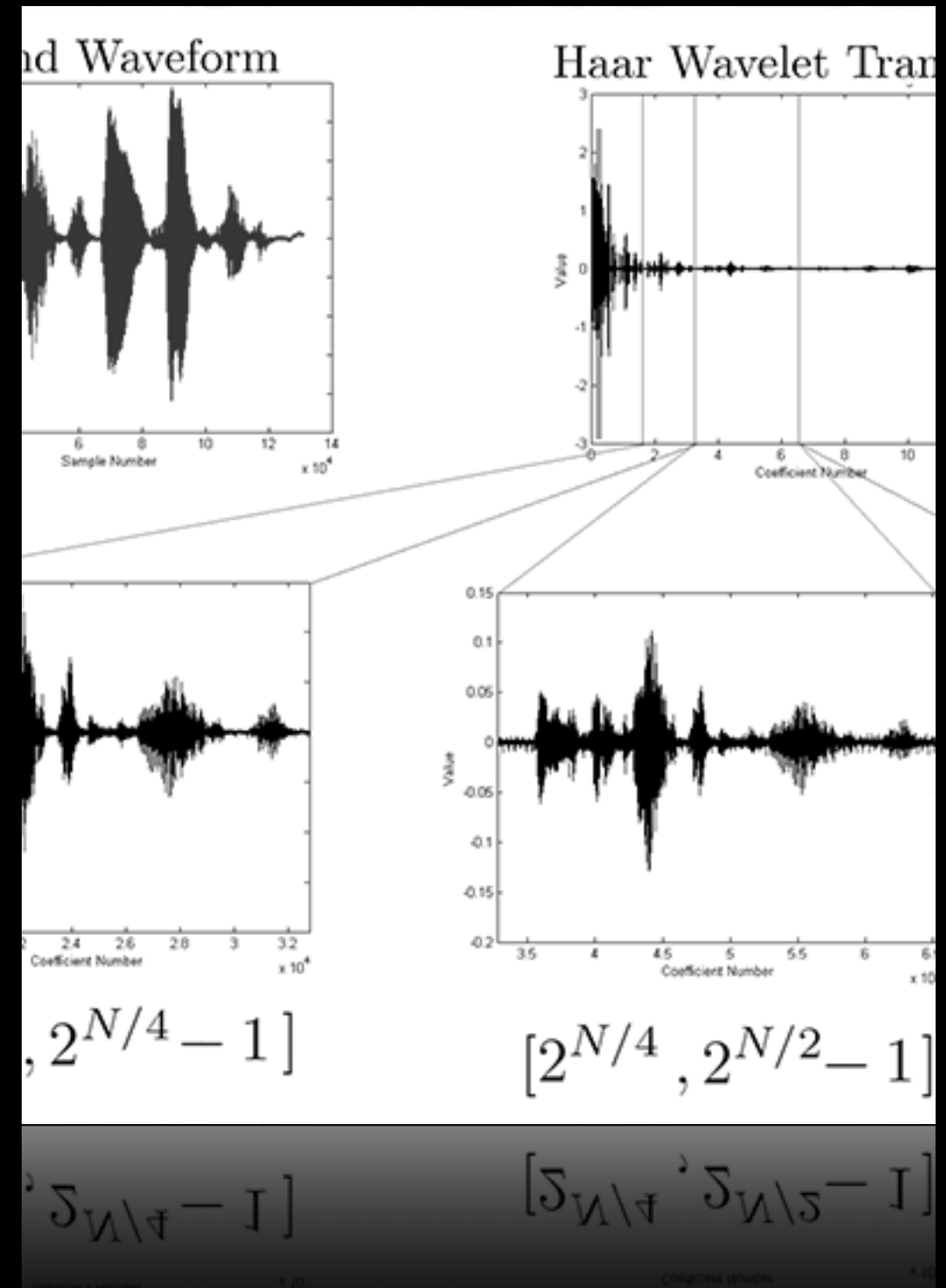
- Tänk mer seminarieform än klassisk föreläsningsform
- 15 minuter per grupp, 2-3 studenter per grupp
- Dvs 10-15 slides
- Googla, biblioteket, och ev kursbok/länkar för info
- Bolla tankar och frågor med mig, men utnyttja alla lärare som finns! (skyll på mig)
- Leta gärna upp ljudexempel!

ERA "FÖRELÄSNINGAR" (FORTS.)

- Presentera artikeln
 - Vad är det artikeln beskriver
 - Vad är problemet och syftet med artikeln/forskningen
 - Hur den tekniska lösningen är gjord/hur är ett problem undersökt/
 - Spela gärna upp ett eller ett par ljudexempel som kopplar till det artikeln handlar om

NIO LABORATIONER

- SuperCollider, vågformer, ljudsyntes
- Arduino, wavetable
- Vågformer, talsyntes, filter i Matlab
- Arduino, filter
- SuperCollider, filter
- Delay, eko i Matlab
- Arduino effekter
- Matlab, forensisk ljudanalys
- Matlab, timestretch & pitchshift
- Extra tillfälle



LABBARN A...

- Tillgång till dator
- Enskilt eller i par, men inte mer än två...
- Hörlurar/externa högtalare
- Matlab
- Arduino IDE
(Arduino Online IDE)
- SuperCollider



TILLFÄLLENA

- 1 halvklass
- 2h per klass...
- 4h-laborationer
- Eget arbete med laborationerna
- Schemalagd tid är för frågor, diskussioner och redovisning



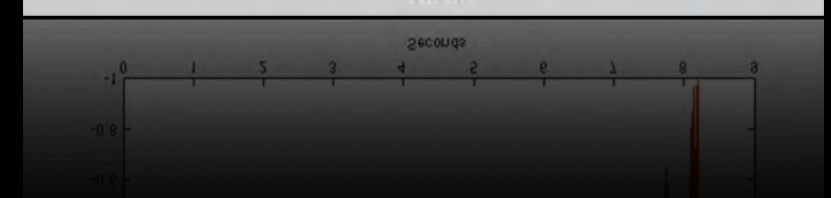
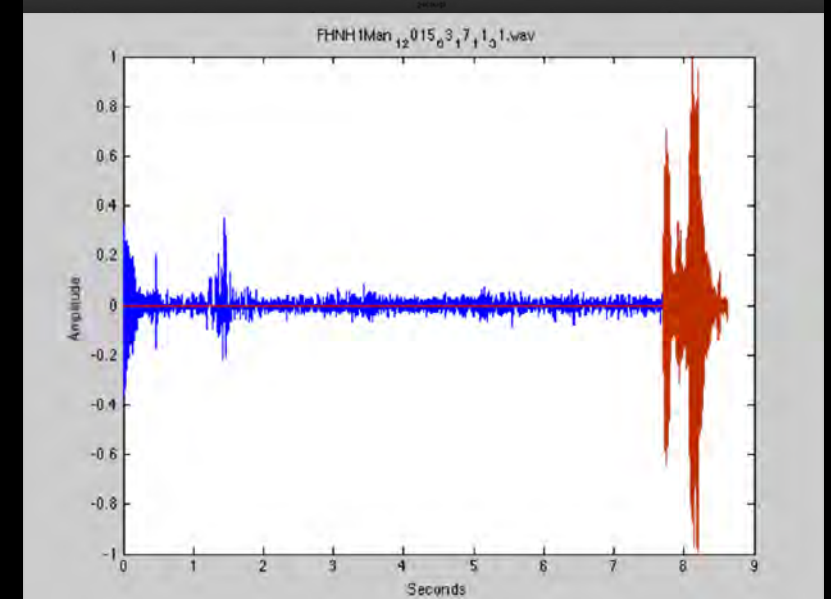
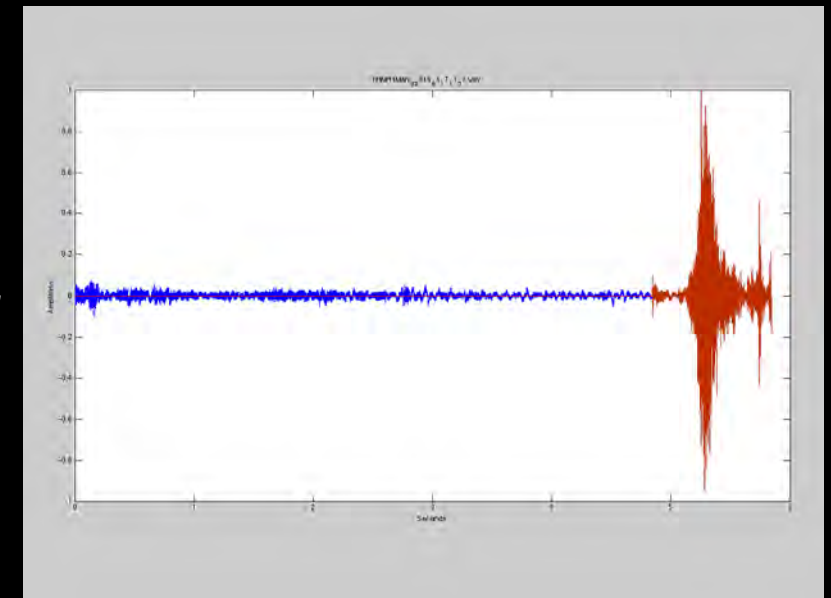
TILLFÄLLENA

- I sal, idén är att kunna diskutera i grupp
- Laborationerna har frågor att fundera kring, besvara, och att jobba vidare från
- Arduino...



HEMTENTAMEN

- Ni kommer att få välja en uppgift, en "ljudgrej" som ska lösas. Ex:
- Ett antal ljudinspelningar av svar
- Automatiskt räkna responstid
- Problem: vad är tyst, harklingar, pauser i responen, bakgrundsbrus, hur räkna ut responstiden



HEMTENTAMEN

- Ni kommer att få välja en uppgift, en "ljudgrej" som ska lösas.
- För att lösa denna uppgift krävs teori, signalbehandling, ljudteknik, programmering.
- Ni ska koppla teori, pseudokod, samt förklara vad, hur och varför ni väljer en viss lösning.
- Ert svar, typ 4 a4-sidor, ska mailas till mig inom utsatt tid.

HEMTENTAMEN

- Uppgiften hämtas vid "ett" kontor i Kopparhammaren
2

26 OKTOBER KLOCKAN 13-13:30

- Svaret mailas till mig, i pdf-format

27 OKTOBER KLOCKAN 9:00

HEMTENTAMEN

Till exjobbet behöver ni repetera en föreläsning om datorgrafik som Stefan Gustavson höll, men som ni själva inte närvarade vid. Ni har dock fått tag på en inspelning av föreläsningen. Men, inspelningen är dålig och ni har svårt att höra vad Stefan säger, då Stefan (av någon outgrundlig anledning) höll föreläsningen i ett trapphus med alldeles för mycket efterklang. Inspe­lningen är också ganska brusig, med lågfrekvent brum från ventilationen vid ytterdörren. Uppgiften är att förbättra ljudets kvalitet så att det går att höra vad Stefan föreläser om, beskriv hur ni skulle gå till väga.

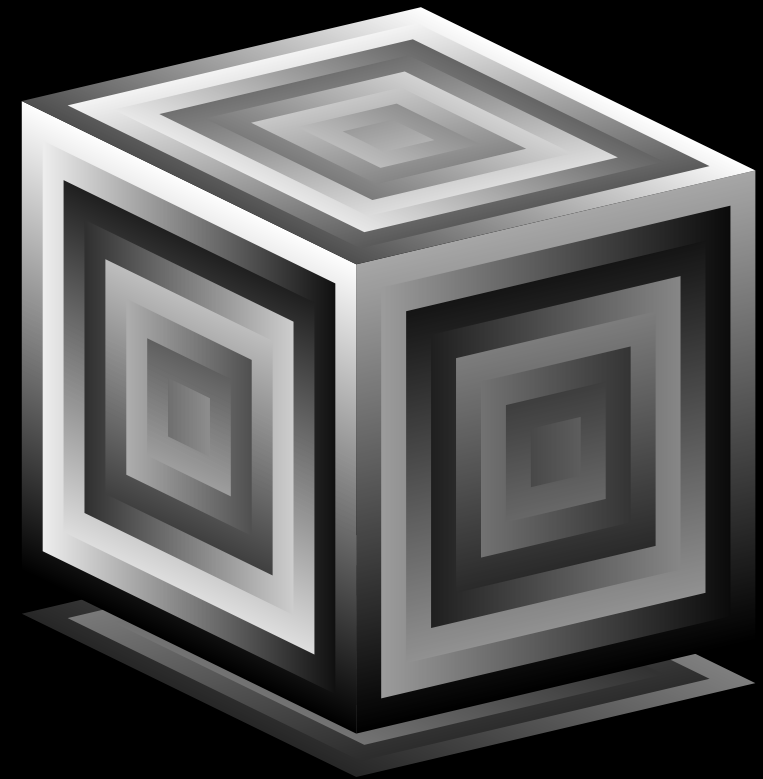
- Efterklangen behöver fixas. Hur? För- och nackdelar med olika metoder? Vilka konsekvenser får de olika metoderna? Vilken metod väljer ni och varför?
- Bruset behöver sänkas. Hur? För- och nackdelar, och vilken metod använder ni?
- Brummet behöver tas bort. Hur? För- och nackdelar...
- Hörbarhetsförbättring, frekvensinnehåll, amplitud, dynamik...

KODNING AV REALTIDSLJUDSYNTES

SUPERCOLLIDER

SUPERCOLLIDER - 1996--

- SuperCollider är ganska gammalt...
- ...med ful syntax...
- ...skumma enheter...
- ...och med fantastiska möjligheter för ljudsyntes...
- ...och alldeles alldeles underbart!



Klient

```
(  
code  
code  
...  
)
```

Server



KOM IGÅNG

- SuperCollider
<https://supercollider.github.io/>
- Eli Fieldsteel
https://youtu.be/yRzsOOiJ_p4
- SuperCollider Code
<https://sccode.org/>

Starta servern...

(
Sedan kod...
)

(
)

Ctrl och Enter
eller
Cmd och Enter

```
(  
fork{
```

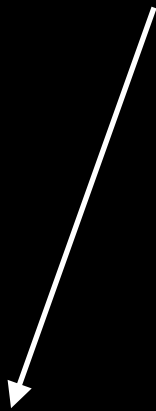
```
};
```

```
)
```

```
(  
fork{  
    loop{  
  
    }  
};  
)
```

```
(  
fork{  
    loop{  
        1.wait;  
    }  
};  
)
```

```
(  
fork{  
  loop{  
    h=([33,38,40].choose);  
  
    1.wait;  
  }  
};  
)
```




```
(  
fork{  
    loop{  
        h=([33,38,40].choose);  
        h.postln;  
  
        1.wait;  
    }  
};  
)
```

```
(  
fork{  
  loop{  
    h=([33,38,40].choose.midicps);  
  
    1.wait;  
  }  
};  
)
```

midi = musical instrument digital interface

midinotnummer = 0 till 127

0 = 8.18Hz och 127 = 12543.85Hz

cps = cycles per seconds

cycles per seconds = Hz

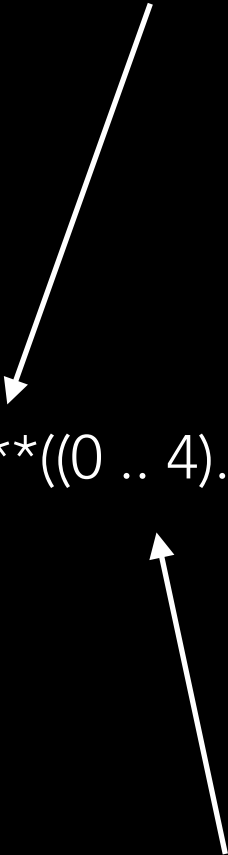
```
(  
fork{  
  loop{  
    h=([33,38,40].choose.midicps);  
    play{  
      Splay.ar({SinOsc.ar(h)});  
    };  
    1.wait;  
  }  
};  
)
```

ar = audio rate

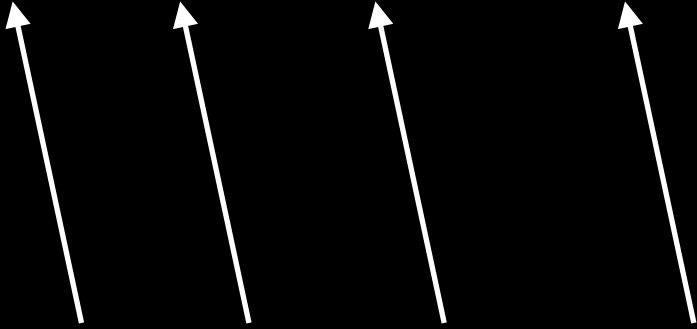
kr = control rate

Ctrl och .
eller
Cmd och .

```
(  
fork{  
  loop{  
    h=([33,38,40].choose.midicps)*(2**((0 .. 4).choose));  
    play{  
      Splay.ar({SinOsc.ar(h)});  
    };  
    1.wait;  
  }  
};  
)
```



```
(  
fork{  
  loop{  
    h=([33,38,40].choose.midicps)*(2**((0 .. 4).choose));  
    play{  
      Splay.ar({SinOsc.ar(h)})*LFGauss.ar(10,1/4,0,0,2);  
    };  
    1.wait;  
  }  
};  
)
```



The image shows a snippet of Pure Data code. The code is enclosed in a fork block and a loop block. Inside the loop, a variable 'h' is calculated based on a random choice of values and a power of 2. Then, a play block is used to play a signal. The signal is the product of a SinOsc.ar function and an LFGauss.ar function. The LFGauss.ar function has five arguments: 10, 1/4, 0, 0, and 2. Four white arrows point from the bottom of the image to these five arguments, specifically highlighting the last four: 1/4, 0, 0, and 2.

```
(  
fork{  
  loop{  
    h=([33,38,40].choose.midicps)*(2**((0 .. 4).choose));  
    play{  
      Splay.ar({SinOsc.ar(exprand(h-(h/64),h+(h/64)),0,0.1)})*LFGauss.ar(10,1/4,0,0,2);  
    };  
    1.wait;  
  }  
};  
)
```

s.scope;

Server.local.plotTree;

```
(  
fork{  
  loop{  
    h=([33,38,40].choose.midicps)*(2**((0 .. 4).choose));  
    play{  
      Splay.ar({SinOsc.ar(exprand(h-(h/64),h+(h/64)),0,0.1)}!8)*LFGauss.ar(10,1/4,0,0,2);  
    };  
    0.1.wait;  
  }  
};  
)
```




```
(  
fork{  
  loop{  
    h=([33,38,40].choose.midicps)*(2**((0 .. 4).choose));  
    play{  
      Splay.ar({LFTri.ar(exprand(h-(h/64),h+(h/64)),0,0.1)}!8)*LFGauss.ar(10,1/4,0,0,2);  
    };  
    0.1.wait;  
  }  
};  
)
```



```
(  
fork{  
  loop{  
    h=([33,38,40].choose.midicps)*(2**((0 .. 4).choose));  
    play{  
      Splay.ar({LFSaw.ar(exprand(h-(h/64),h+(h/64)),0,0.1)}!8)*LFGauss.ar(10,1/4,0,0,2);  
    };  
    0.1.wait;  
  }  
};  
)
```

