

# Laboration: Digital ljudsyntes

## Uppgift 1

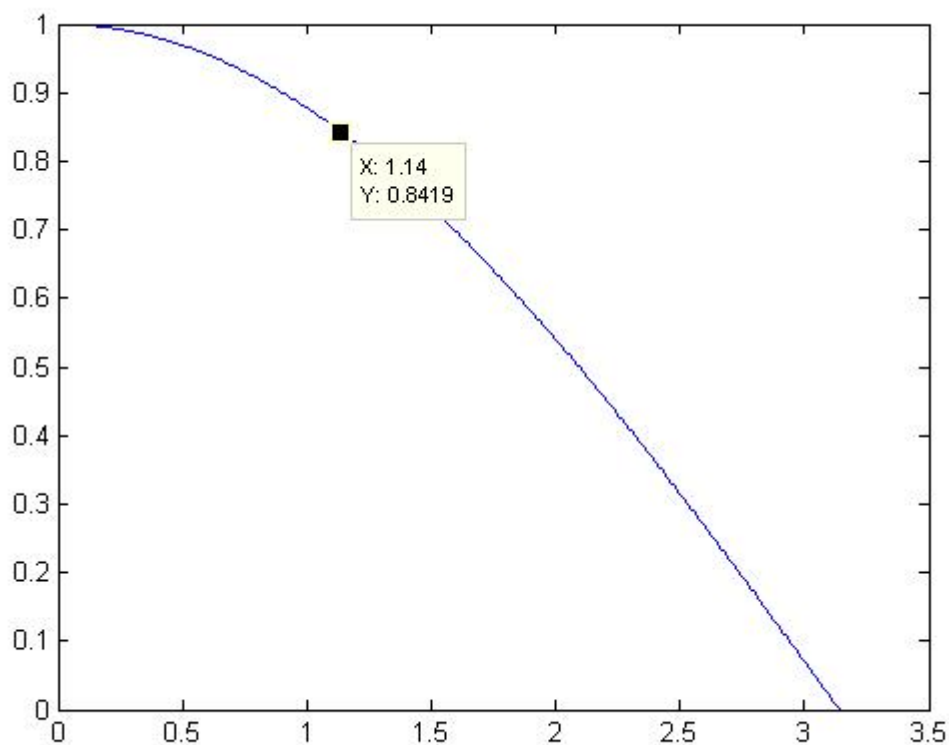
*Konstruera ett kamfilter med med grundfrekvens ungefär lika med C (523.25 Hz) .*

*Lyssna på resultatet och kommentera. Hur nära kan du komma ?*

Mitt gehör hör ett C i alla fall.

*Redovisa grafen för lågpassfiltrets amplitudegenskaper.*

*Hur mycket dämpar filtret vid 4000 Hz ?*



$\pi =$  Halva nyqvist, för 22050 Hz  $\Rightarrow$  11025 Hz

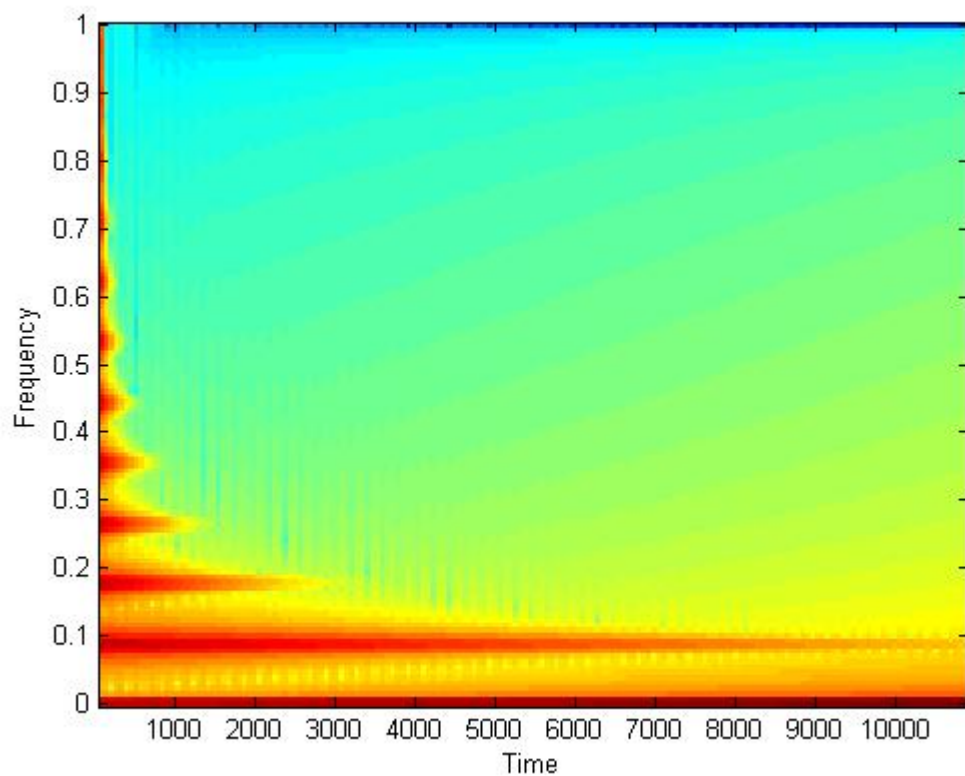
$x = \pi / 11025 * 4000 \approx 1.14$

$y[1.14] \approx 0.84 = 84\%$  släpper den igenom.

Svar: Den dämpar med ca 16%.

## Uppgift 2

*Modifiera ditt kamfilter med lågpasskompensering enligt ovan. Justera parametrarna så att utsignalen låter "behaglig". Rita spektrogram och jämför med nedan.*



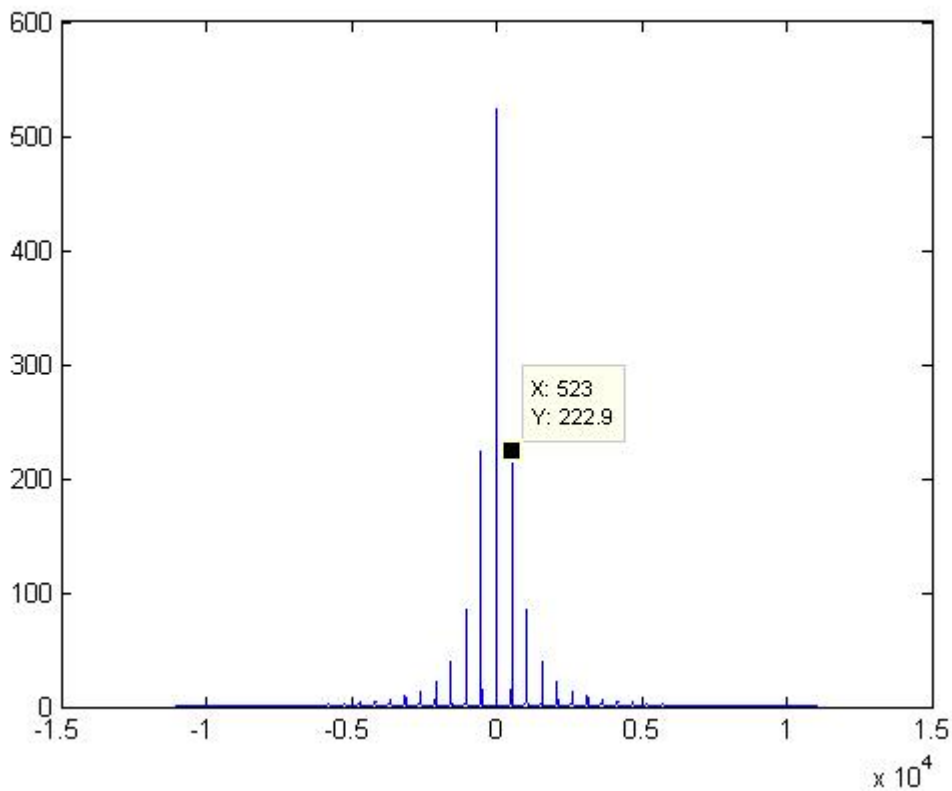
## Uppgift 3

Komplettera ditt plucked-string-filter med en all-pass-länk. Härled och redovisa den totala överföringsfunktionen.

Den totala överföringsfunktionen blir  $\frac{\text{Lagpassfiltret} \cdot \text{AllPassfiltret}}{1 - \text{Lagpass} \cdot \text{AllPass} \cdot \text{Kamfiltret}}$  se nedan:

$$\frac{0.5 \cdot (1 + z^{-1}) \cdot \frac{z^{-1} + a}{1 + a \cdot z^{-1}}}{1 - (R^L \cdot z^{-L}) \cdot (0.5 \cdot (1 + z^{-1})) \cdot \left(\frac{z^{-1} + a}{1 + a \cdot z^{-1}}\right)}$$

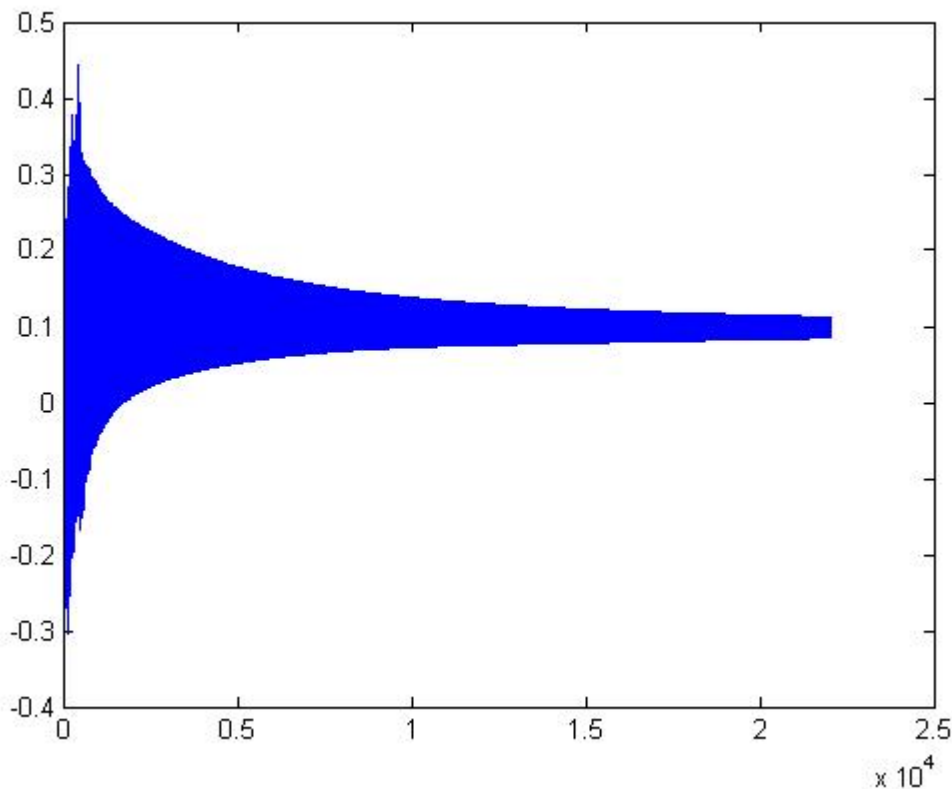
Amplitudegenskaper för totala filtret: (Vald ton 523.25 Hz / C5)



## Uppgift 4

*Beskriv den insignal du använt. Plotta den intressantaste delen av den. Vilken karaktär på ljudet har du strävat efter? Spela in en enkel melodi med hjälp av din synthesizer. Redovisa inspelningen. Kommentera sluresultatet.*

Som insignal har jag använt ca 500 slumpstal följt av resterande nollor (frek-500), sedan har jag tagit signalen gånger en negativt exponentiell funktion (minus roten ur x) där x går från 0 till 1. Sedan har jag lagt på kam-, lågpass- och All-Pass-filter på signalen. Jag har strävat efter att få en plucked-string ton som klingar ut. Jag tycker tonerna blev mycket fina.



För att skapa slutlåten (Tetris) har jag tagit ut pianotoner från en youtube film där filmskaparen går igenom tonerna för vänster och höger hand. Jag skapade först hela låten med höger hand och sedan vänster vilket gör att de tyvärr kanske inte ligger helt rätt tillsammans. Jag matchade storleken på första delen av låten med vänster och höger o samma sak med andra delen vilket gör att i alla fall andra delen blev ganska bra matchad. Det är dock svårt att höra om det verkligen är rätt eller fel eftersom noterna är tänkt att spelas med ett riktigt piano som har en helt annan klang. I låten jag skickar in spelas därför först vänster sedan höger och tillsist vänster

och höger hand tillsammans (det är en paus mellan varje).

Jag har även sänkt volymen på vänster hand så den inte tar över för mycket.

## Källkod

```
%Uppgift 1
% fs = 22000;
% z = tf('z');
% R = 1.0001;
% L = 42;
% H = 1/(1-R^L*z^(-L));
% y = impulse(H,1*fs);
% sound(y, fs);

% Uppgift 1b
% O = 0:0.01:pi;
% H1 = 0.5.*(1+exp(1i.*O));
% plot(O,abs(H1));

% Uppgift 2
% H1 = R^L*z^(-L);
% H2 = 0.5*(1+z^(-1));
% Htot = H1/(1-H1*H2);
% y2 = impulse(Htot, 1*fs);
% sound(y2, fs);
% specgram(y2);

% Uppgift 3 / 4
fs = 22050;
z = tf('z');
R = 0.9999999999999999;
fc = 523.25 ; %sökta frek
L = floor(fs/fc-0.5);
LF = fs/fc;
f0 = fs/L; %2200
O = -0.7:0.1:2*pi*f0/fs;
apa = fs/f0; % 22
```

## Mediesignaler, TFE-10

2010-03-09

```
d = LF-L-0.5;
a = sin((1-d)*O/2)/sin((1+d)*O/2);
%filtrerade signalen
Hkam = R^L*z^(-L);
Hlp = 0.5*(1+z^(-1));
Hall = (z^(-1)+a)/(1+a*z^(-1));
Htot = (Hlp*Hall)/(1-Hkam*Hlp*Hall);

% -- Uppgift 3 filteregenskaper --
% p = impulse(Htot, 1*fs);
%
% N=length(p);
% df=fs/N;
% f=- (N/2):1:(N/2)-1;
% f = f*df;
% yy=abs(fftshift(fft(p)));
% plot(f,yy);

%
% %signalen
% i1 = -0.2;
% i2 = 0.2;
% r = i1 + (i2-i1).*rand(525,1);
% g = zeros(fs-525,1);
% p = vertcat(r, g); %r;%[r,g];
% % [Num, Den] = tfdata(Htot, 'v');
% %y = filter(Num, Den, p);
%
% x = 0:1/(fs-1):1;
% q = -sqrt(x)+1;
% %plot(q);
% p = p(:).*q(:);
%
% [Num, Den] = tfdata(Htot, 'v');
% y = filter(Num, Den, p);
%
% plot(y);
% sound(y, fs);
```

## Mediesignaler, TFE-10

2010-03-09

```
%  
% WAVWRITE(y, fs, 'alb.wav');  
  
% -- UPPSPELNING --  
% a = wavread('a.wav');  
% a1 = wavread('a1.wav');  
% a2 = wavread('a2.wav');  
% alb = wavread('alb.wav');  
% a2b = wavread('a2b.wav');  
% la = wavread('la.wav');  
% b = wavread('b.wav');  
% b1 = wavread('b1.wav');  
% b2 = wavread('b2.wav');  
% lb = wavread('lb.wav');  
% lbh = wavread('lbH.wav');  
% h = b;  
% c = wavread('c.wav');  
% c2 = wavread('c2.wav');  
% c3 = wavread('c3.wav');  
% d = wavread('d.wav');  
% d2 = wavread('d2.wav');  
% d3 = wavread('d3.wav');  
% e = wavread('e.wav');  
% e2 = wavread('e2.wav');  
% e3 = wavread('e3.wav');  
% f = wavread('f.wav');  
% g = wavread('g.wav');  
% giss = wavread('giss.wav');  
% hgiss = wavread('hgiss.wav');  
%  
% fs = 22050;  
% p = zeros(1*fs,1);  
% % vänster hand del 1  
% tunelp1 = [e(1:0.4*fs);lb(1:0.4*fs);c(1:0.3*fs);d(1:0.3*fs);  
%     e(1:0.3*fs);d(1:0.3*fs);c(1:0.3*fs);lb(1:0.3*fs);la(1:0.4*fs);  
%     la(1:0.4*fs);c(1:0.4*fs);e(1:0.4*fs);  
%     d(1:0.4*fs);c(1:0.4*fs);lb(1:0.4*fs);lb(1:0.4*fs);  
%     c(1:0.4*fs);d(1:0.4*fs);e(1:0.4*fs);c(1:0.4*fs);la(1:0.4*fs);la(1:0.4*fs);
```

## Mediesignaler, TFE-10

2010-03-09

```

%      d(1:0.4*fs);f(1:0.4*fs);a(1:0.4*fs);
%      g(1:0.3*fs);f(1:0.3*fs);e(1:0.3*fs);
%      c(1:0.4*fs);e(1:0.4*fs);d(1:0.4*fs);c(1:0.3*fs);lb(1:0.4*fs);
%
lb(1:0.3*fs);c(1:0.3*fs);d(1:0.3*fs);e(1:0.4*fs);c(1:0.4*fs);la(1:0.4*fs);la(1:0.4*fs)];
%part 2 ->
%
% % vänster hand del 2
% tune1p2 = [
e(1:0.6*fs);c(1:0.6*fs);d(1:0.6*fs);lb(1:0.6*fs);la(1:0.6*fs);c(1:0.6*fs);a(1:0.4*fs);
a(1:0.4*fs);hgiss(1:0.4*fs);
%
e(1:0.6*fs);c(1:0.6*fs);d(1:0.6*fs);lb(1:0.6*fs);la(1:0.6*fs);c(1:0.6*fs);a(1:0.4*fs);
a(1:0.4*fs);hgiss(1:0.4*fs)];
%
% % höger hand del 1
% tlen = 0.4083;
% tune2p1 = [e2(1:tlen*fs); e3(1:tlen*fs);e2(1:tlen*fs); e3(1:tlen*fs);
%      a1(1:tlen*fs); a2(1:tlen*fs);a1(1:tlen*fs); a2(1:tlen*fs);
%      alb(1:tlen*fs); a2b(1:tlen*fs);alb(1:tlen*fs); a2b(1:tlen*fs);
%      a1(1:tlen*fs); a2(1:tlen*fs);a1(1:tlen*fs); a2(1:tlen*fs);
%      a1(1:tlen*fs); a1(1:tlen*fs); b1(1:tlen*fs); c2(1:tlen*fs);
%      d2(1:tlen*fs); d3(1:tlen*fs);d2(1:tlen*fs); d3(1:tlen*fs);
%      c2(1:tlen*fs); c3(1:tlen*fs);c2(1:tlen*fs); c3(1:tlen*fs);
%      b1(1:tlen*fs); b2(1:tlen*fs);b1(1:tlen*fs); b2(1:tlen*fs);
%      a1(1:tlen*fs); a2(1:tlen*fs);a1(1:tlen*fs); a2(1:tlen*fs);p(1:27)];
%
% % höger hand del 2
% tlen = 0.3;
% tune2p2 = [a1(1:tlen*fs); e2(1:tlen*fs);a1(1:tlen*fs); e2(1:tlen*fs);
%      alb(1:tlen*fs); e2(1:tlen*fs);alb(1:tlen*fs); e2(1:tlen*fs);
%      a1(1:tlen*fs); e2(1:tlen*fs);a1(1:tlen*fs); e2(1:tlen*fs);
%      alb(1:tlen*fs); e2(1:tlen*fs);alb(1:tlen*fs); e2(1:tlen*fs); %repeat ->
%      a1(1:tlen*fs); e2(1:tlen*fs);a1(1:tlen*fs); e2(1:tlen*fs);
%      alb(1:tlen*fs); e2(1:tlen*fs);alb(1:tlen*fs); e2(1:tlen*fs);
%      a1(1:tlen*fs); e2(1:tlen*fs);a1(1:tlen*fs); e2(1:tlen*fs);
%      alb(1:tlen*fs); e2(1:tlen*fs);alb(1:tlen*fs); e2(1:tlen*fs)];
%

```



Mediesignaler, TFE-10

2010-03-09

```
% % tuneT1 = tune1p1+(tune2p1.*0.7);
% % tuneT2 = tune1p2+(tune2p2.*0.7);
% % tuneT = vertcat(tuneT1, tuneT2);
% % tuneV = vertcat(tune1p1, tune1p2);
% % tuneH = vertcat(tune2p1, tune2p2);
% % tuneAll = vertcat(tuneV, p(1:fs), tuneH, p(1:fs), tuneT);
% % soundsc(, fs);
% % soundsc(p(1:fs), fs);
% % soundsc(vertcat(tune2p1, tune2p2), fs);
% % soundsc(p(1:fs), fs);
% soundsc(tuneT, fs);

%WAVWRITE(tuneAll, fs, 'låt2.wav');
```