

SSY080

Inlämningsuppgift

Linus Oleander - 880613 - 4873
Jesper Josefsson - 860409 - 5276

2011-10-10

1 Bakgrund

Uppgiften går ut på att genomföra ett antal experiment gällande generering och behandling av signaler med Matlab.

2 Generering av fyrkantsvåg med hjälp av Fourierserie - (3.1)

2.1 Fourierkoefficienter

Den första uppgiften var att ta fram ett slutet uttryck för Fourierseriekoefficienterna A_k och B_k . Vi använde följande samband:

$$\begin{aligned} C_k &= \frac{1}{T} \int_0^T x(t) e^{-jk\omega_0 t} dt = \\ &= \frac{1}{T} \left(\int_0^{\frac{T}{2}} e^{-jk\omega_0 t} dt - \int_{\frac{T}{2}}^T e^{-jk\omega_0 t} dt \right) = \\ &= \frac{1}{-jk\omega_0 T} \left([e^{-jk\omega_0 t}]_0^{\frac{T}{2}} - [e^{-jk\omega_0 t}]_{\frac{T}{2}}^T \right) = \\ &= \left[\omega = \frac{2\pi}{T} \Rightarrow \frac{\omega_0 T}{2} = \pi, \omega_0 T = 2\pi \right] = \\ &= \frac{1}{-jk\omega_0 T} (2e^{-jk\pi} - e^{-jk2\pi} - 1) = \\ &= \begin{bmatrix} 2e^{-jk\pi} & = \begin{cases} -2 & , k \text{ udda} \\ 2 & , k \text{ jämn} \end{cases} \\ e^{-jk2\pi} & = 1 \end{bmatrix} = \\ &= \begin{cases} \frac{-4}{-jk2\pi} = \frac{2}{jk\pi} = -\frac{2j}{k\pi} & , k \text{ udda} \\ 0 & , k \text{ jämn} \end{cases} \\ A_k &= C_k - C_{-k} = -\frac{2i}{k\pi} + \frac{2i}{k\pi} = 0 \\ C_k &= \frac{1}{2}(A_k - iB_k) \Rightarrow B_k = 2iC_k = \frac{4}{k\pi} \end{aligned}$$

2.2 Generering av fyrkantsvåg

Vi använder Fourierkoefficienterna för att generera en fyrkantsvåg i matlab med hjälp av definitionen av Fourierserien på trigonometrisk form:

$$x(t) = A_0 + \sum_{k=1}^{\infty} [A_k \cos(k\omega_0 t) + B_k \sin(k\omega_0 t)]$$

Koden blir som nedan:

```
T = 2;
w=2*pi/T;
M=200;
t=T*(0:M-1)/M;
y = @(t) 0;
bs = [];
for k=1:100
    ck = -(mod(k, 2))*((1i*2)/(pi*k));
    cminusk = -(mod(k, 2))*((1i*2)/(pi*(-k)));
    ak = ck + cminusk;
    bk = 2i*ck;
    bs = [bs bk];
    y = @(t) y(t) + ak*cos(k*w*t) + bk*sin(k*w*t);
end
```

För fyrkantsvåg, se figur 1.

3 Linjära system och sinusar (3.2)

3.1 Konstruering av givet system

Vi konstruerar först ett LTI-system med ett frekvenssvar $G(j\omega)$ som är givet i uppgiften. Därefter plottar vi det i frekvensdomänen samt studerar dess poler och nollställen - figur 2 och 3.

```
% num = (s + 0.1)(s + 10) = s^2 + 10.1s + 1
% den = (s + 1)(s^2 + s + 9) = s^2 + 2s^2 + 10s + 9
num = {[1 10.1 1]};
den = {[1 2 10 9]};
H = tf(num, den);
bode(H)
title('Bode for H');
% pzmap(H);
% sgrid
% grid on
```

I Bode-diagrammet framkommer systemets påverkan på olika frekvenser i insignalen. I det övre diagrammet kan vi se systemets amplitudpåverkan. Till exempel förstärks komponenter med vinkel-frekvens $\omega = 3$ med ungefär $10dB$.

I det undre diagrammet ser vi systemets faspåverkan. Till exempel får komponenter med vinkelfrekvensen $\omega = 3$ en förskjutning på ungefär -100° .

3.2 Insignaler skapas och körs genom systemet

Tre insignaler skapas. Dessa är sinusformade signaler med vinkelfrekvenserna $\omega = [1, 2, 3]$. Därefter används matlab för att simulera systemets påverkan på signalerna. In- och utsignal plottas, se figur 4-6.

```
N = 8192;
f = 100;
Ts = 1/f;
Tmax = (N-1)*Ts;
t= 0:Ts:Tmax;
Xs = ones(3,N);
Y = ones(3,N);
index = 1;
for w=[1 3 5],
    x = sin(w*t);
    Xs(index,:) = x;
    [y, t] = lsim(H,x,t);
    Y(index,:) = y;
    index = index +1;
end
plot(t,Y(1,:), 'b',t, Xs(1,:), 'r')
title('In and out signals for w=1');

% plot(t,Y(2,:), 'b',t, Xs(2,:), 'r')
% title('In and out signals for w=3');

% plot(t,Y(3,:), 'b',t, Xs(3,:), 'r')
% title('In and out signals for w=5');
```

Man kan se att utsignalerna har samma periodtid som insignaler. Man kan också se att det har skett amplitudpåverkan samt faspåverkan. Med hjälp av evalfr kan vi bestämma systemets teoretiska amplitud- och faspåverkan:

```
>> abs(evalfr(H,1i))
ans =
    0.8858
>> abs(evalfr(H,3i))
ans =
    3.3033
>> abs(evalfr(H,5i))
ans =
    0.6541
>> angle(evalfr(H,1i))*180/pi
ans =
    37.8750
>> angle(evalfr(H,3i))*180/pi
ans =
   -56.7750
>> angle(evalfr(H,5i))*180/pi
ans =
  -125.9168
```

Ur detta kan man läsa att förhållandet $y(t) = g(\omega)\sin(\omega t + \phi(\omega))$ stämmer.

4 Periodiska insignaler och DFT

4.1 DFT för matlabs fyrkantsvåg

Först plottades matlabs fyrkantsvåg - se figur 7. Vi konstaterar att funktionen är udda eftersom $f(-x) = -f(x)$. Detta syns dåligt i diagrammet, eftersom vågen genererades med utgångspunkt $t = 0$.

Därefter plottades frekvenssvaret för matlabs fyrkantsvåg, se figur 8.

```
x = square(t);  
% plot(t,x);  
  
k = 0:(N-1);  
wk = 2*pi*fs*k/N;  
X = fft(x,N);  
  
plot(wk, abs(X))
```

Sedan användes ett antal uttryck för att räkna ut de teoretiska Fourierkoefficienterna. Det empiriskt utkomna koefficienterna räknades ut med matlabrutinen evalfr.

Fourierkoefficienter för fyrkantsvåg			
Teoretiska	1.2732	0.4244	0.2546
Empiriska	1.2702	0.4154	0.2398

4.2 Systemet appliceras på fyrkantsvågen

Det tidigare angedda systemets påverkan på fyrkantsvågen beräknas med hjälp av rutinen lsim.

Fourierkoefficienterna för utsignalen y beräknas med hjälp av givna ekvationer och tas även fram empiriskt med hjälp av matlabrutinen evalfr.

Till sist plottas y i frekvensplanet - se figur 9.

Där framgår det att $\omega = 1$ är lite förstärkt, $\omega = 3$ är kraftigt förstärkt, och $\omega = 5$ är kraftigt sänkt.

Fourierkoefficienter för fyrkantsvåg efter system G			
Teoretiska	1.1279	1.4020	0.1666
Empiriska	1.1023	1.3378	0.1513

5 Notch-filter

5.1 Nollställen

Ett polynom för ett notch-filter ska konstrueras.

Nollställen väljs för $\omega = [1, 5, 7, 9]$ samt $\omega = 0$:

$$\text{täljarpolynom} = s(s-1)(s+1)(s-5)(s+5)(s-7)(s+7)(s-9)(s+9)$$

Därefter plottas frekvenssvaret - se figur 10.

```
roots = [0 -1i 1i -5i 5i -7i 7i -9i 9i];  
num = poly(roots);  
sys = tf(num, 1);  
bode(sys)
```

5.2 Poler

Polerna väljs enligt $p = -4$. Antalet bestämmer dämpningen av högre frekvenser. Empiriska test visar att det behövs i storleksordningen 21 poler för att uppnå $60dB$ dämpning för $\omega > 9$.

```
proots = -4*ones(1,21);  
den = poly(proots);
```

5.3 Förstärkning

Förstärkningen ska normalisera till $0dB$ för $\omega = 3$. Det görs genom att man lägger till ett nollställe vid $\omega = 3$ som bildar den förstärkande faktor som behövs:

$$(3i - x) = 1/(\text{skalfaktor före korrigering})$$

Skalfaktorn före korrigering räknas med med evalfr:

```
evalfr(sys, 3i)
```

Den slutgiltiga koden blir:

```
roots = [0 -1i 1i -5i 5i -7i 7i -9i 9i 4.31165e8];  
num = poly(roots);  
proots = -4*ones(1,21);  
den = poly(proots);  
sys = tf(num,den);
```

5.4 Slutgiltiga filterresultat

Se figur 11.

6 Figurer

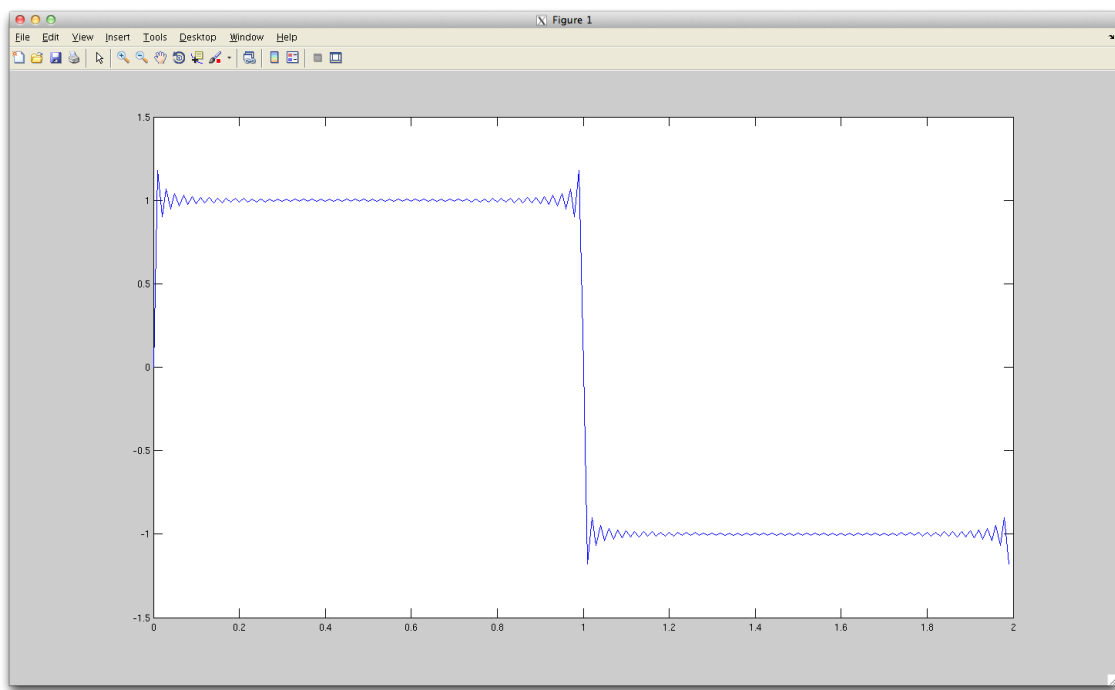


Figure 1: Vår fyrkantsvåg

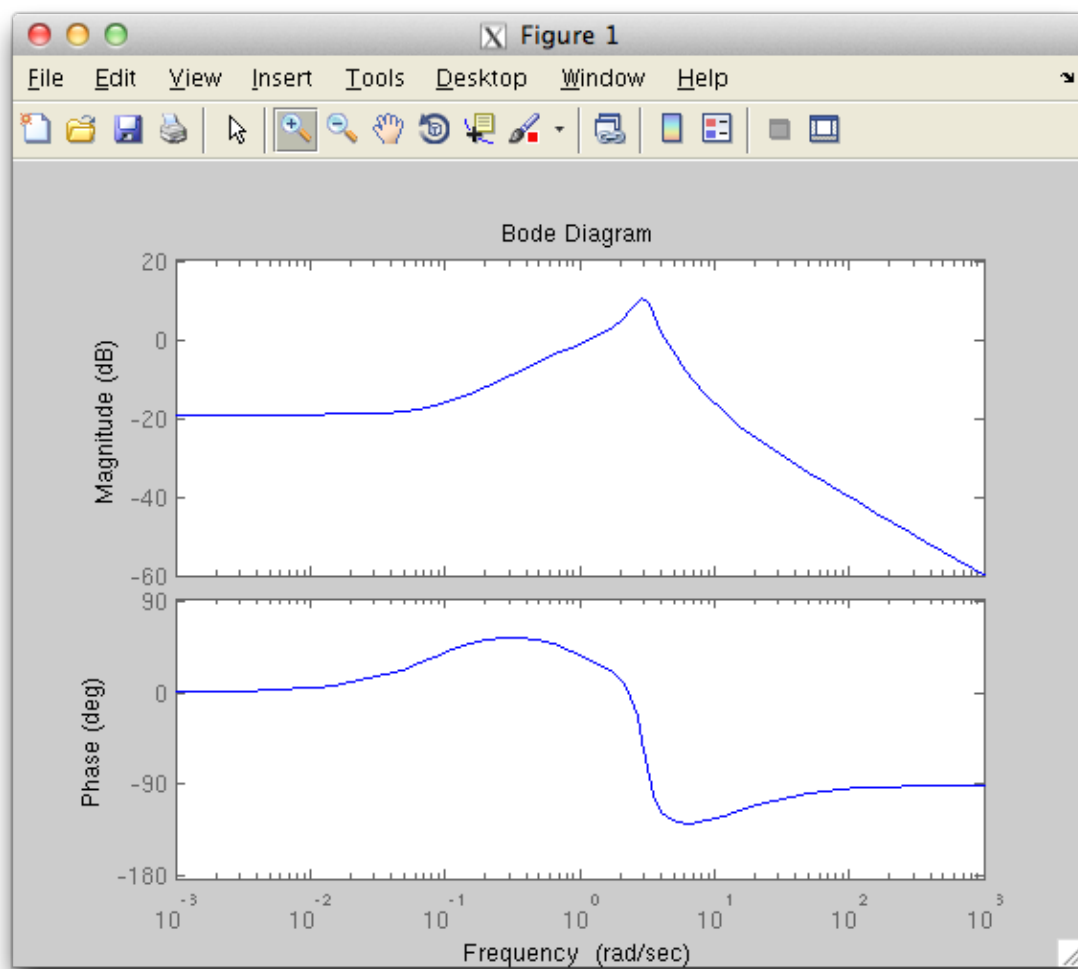


Figure 2: Bode-diagram för $G(j\omega)$

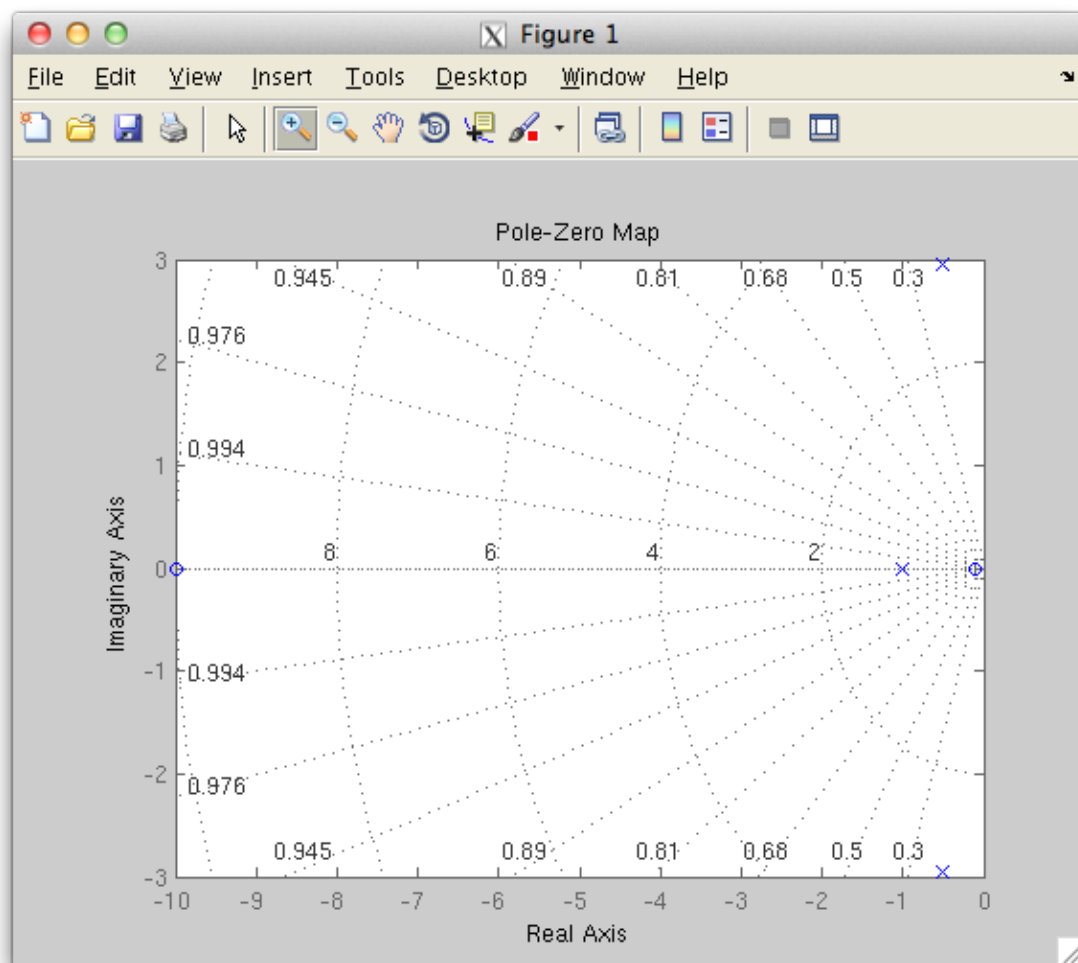
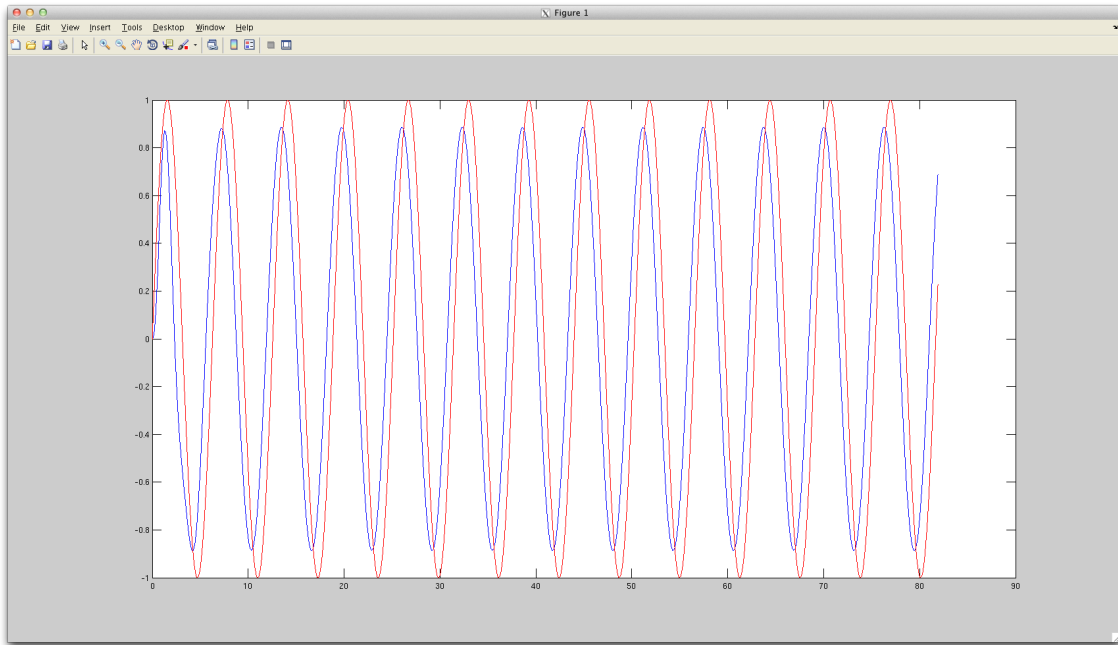
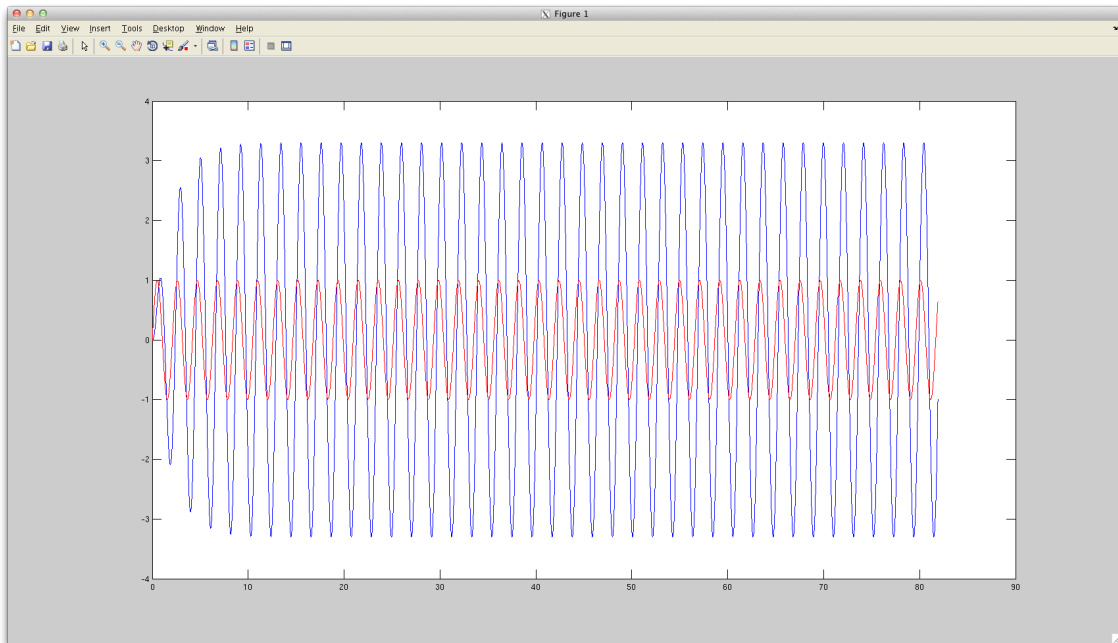


Figure 3: pzmap



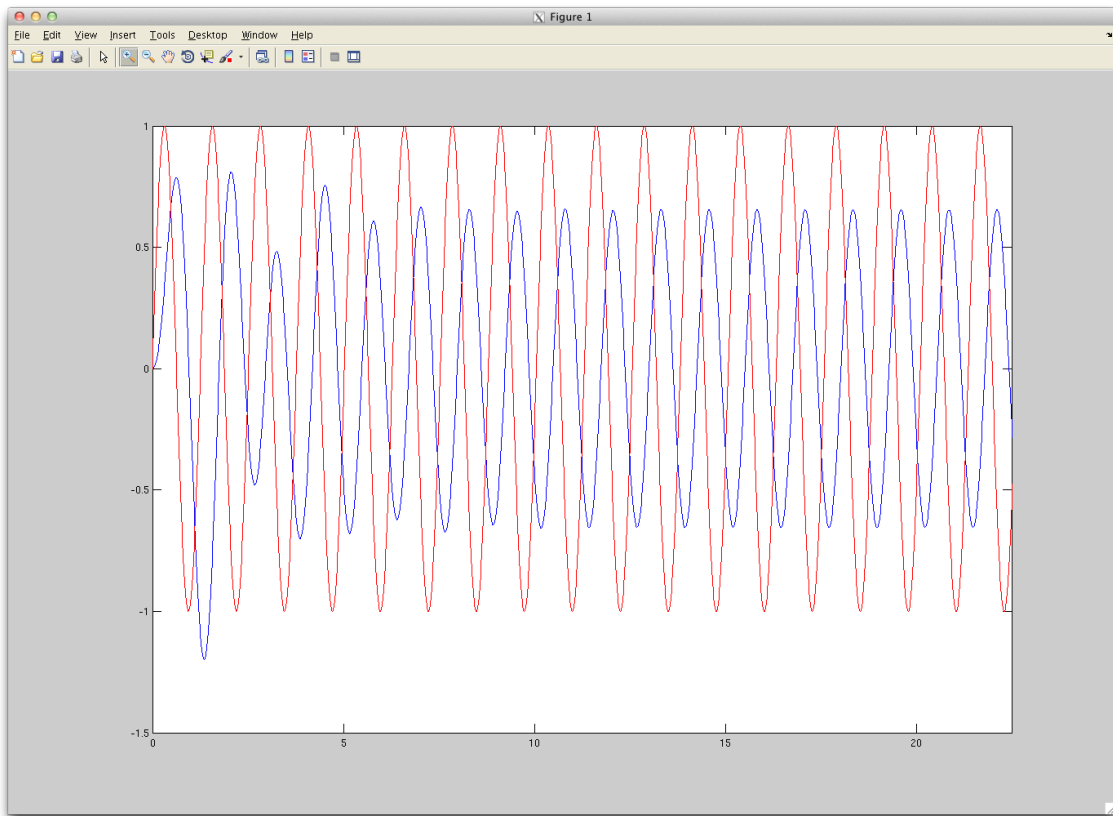
1.png

Figure 4: $\omega = 1$



3.png

Figure 5: $\omega = 3$



5.png

Figure 6: $\omega = 5$

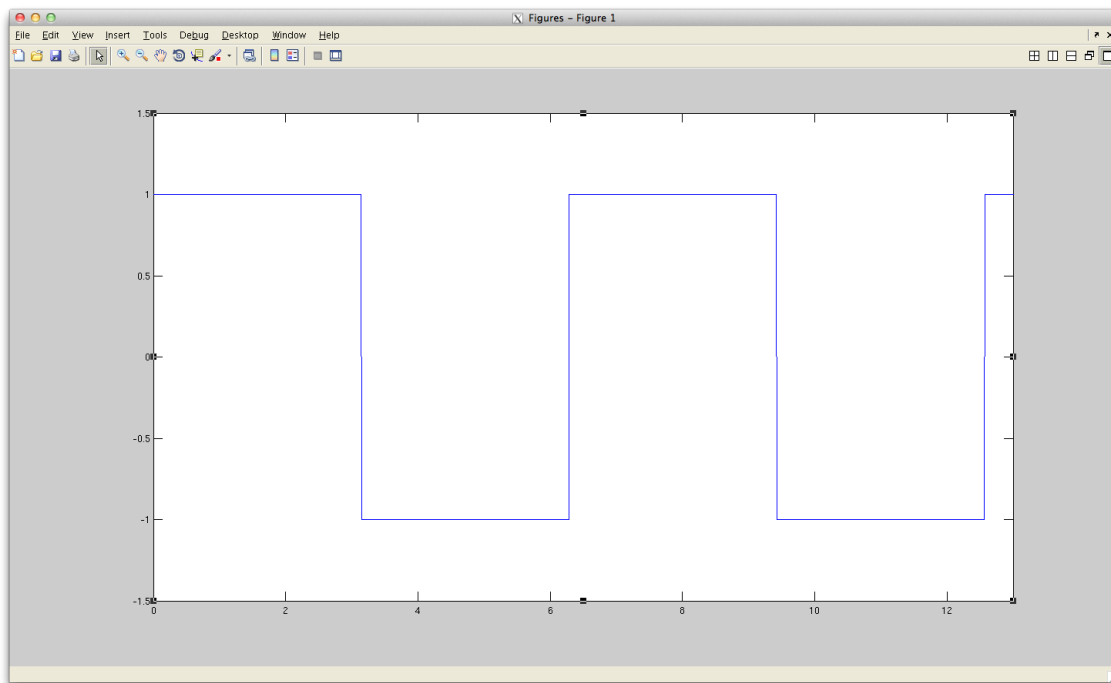


Figure 7: Matlabs fyrkantsvåg

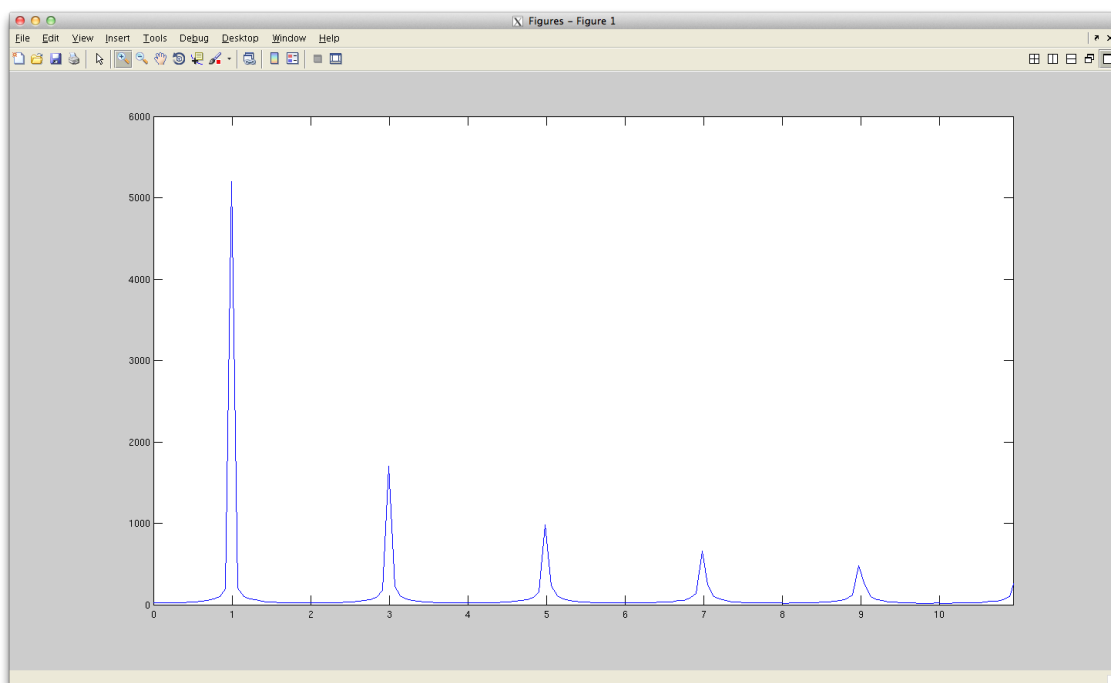


Figure 8: DFT-transform av matlabs fyrkantsvåg

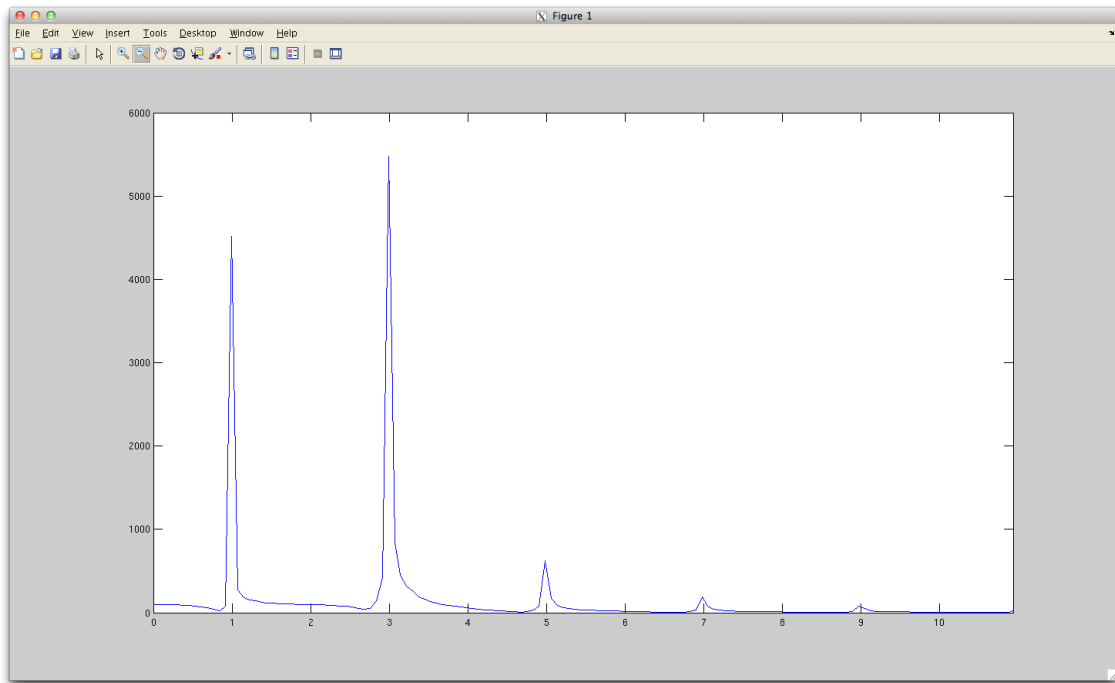


Figure 9: DFT-transform av fyrkantsvåg efter påverkan från system G

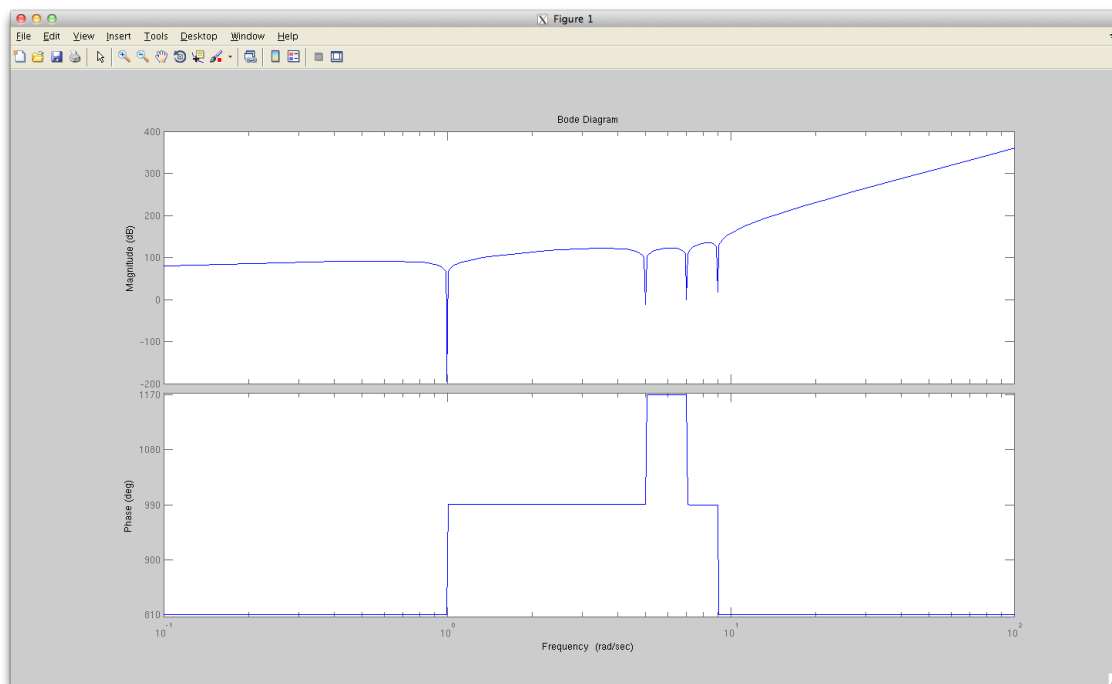


Figure 10: Notchfiler med endast nollställena

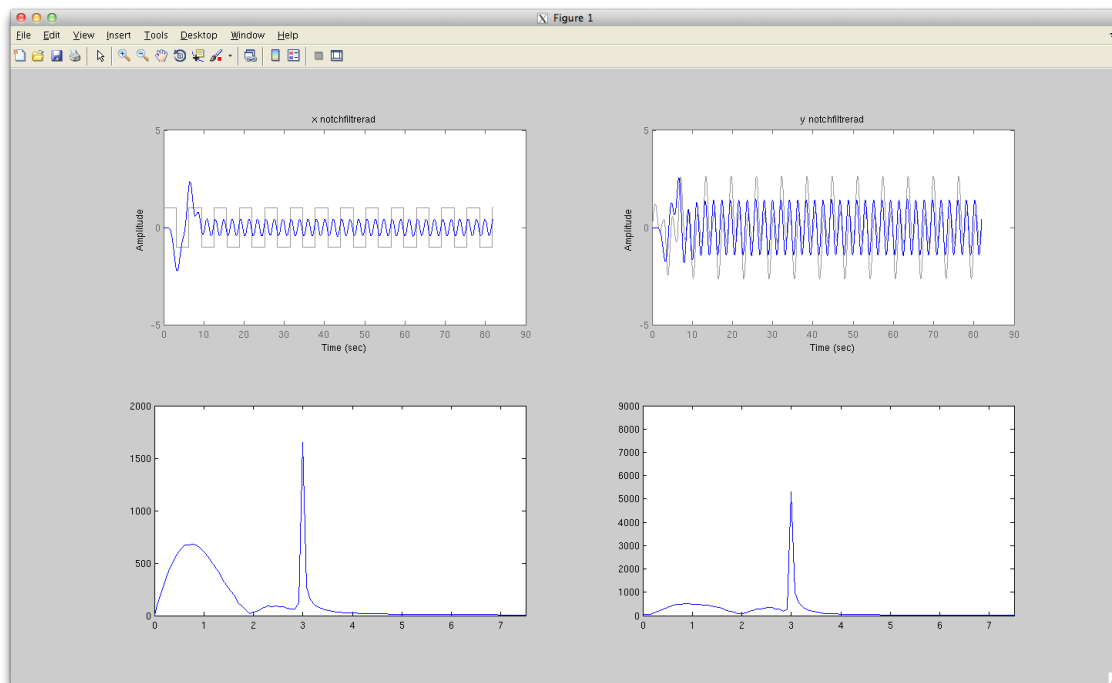


Figure 11: Resultat efter notchfiltrering - x är fyrkantsvåg, y är fyrkantsvåg med påverkan av det givna systemet