

Laboratoriearbeid

PHYS116

Lab 09: **DTMF**

Høsten 2005

Rapport fra:
Erik Grindheim

På gruppen:

Saroj Sahajpal
Erik Grindheim

DTMF

Rapport fra: **Erik Grindheim**
Gruppe: **Saroj S. & Erik G.**
Dato ferdig: **01. desember 2005**

4.1 a)

Det er to måter å bestemme β på, slik at filterets maksimale frekvensrespons blir 1. Enten kan Matlab gjøre dette numerisk, eller vi kan regne oss frem til en god verdi for β . Vi prøver her å gjøre det siste. Den høyeste amplituden kommer ut hvis vi sender inn samme frekvens som den avkuttete cosinusen i filterets impulsrespons:

$$h[n] = \beta \cdot \cos(2\pi \cdot f_b \cdot n \cdot T_s) \quad 0 \leq n \leq L-1 \quad \text{Impulsresponsen.}$$

$$x[n] = \cos(2\pi \cdot f_b \cdot n \cdot T_s + \varphi) \quad \text{For alle verdier av } n. \text{ Dette er inngangssignalet.}$$

$$y[n] = x[n] * h[n] = \sum_{k=0}^{L-1} \beta \cdot \cos(2\pi \cdot f_b \cdot k \cdot T_s) \cdot \cos(2\pi \cdot f_b \cdot (n-k) \cdot T_s + \varphi) \quad \text{Utgangssignalet}$$

$$y[n] = \beta \cdot \sum_{k=0}^{L-1} \cos(2\pi \cdot f_b \cdot k \cdot T_s) \cdot [\dots]$$

$$[\dots] = \cos(2\pi \cdot f_b \cdot k \cdot T_s) \cdot \cos(2\pi \cdot f_b \cdot n \cdot T_s + \varphi) + \sin(2\pi \cdot f_b \cdot k \cdot T_s) \cdot \sin(2\pi \cdot f_b \cdot n \cdot T_s + \varphi)$$

Deler opp summen i to del-summer, slik: $y[n] = y_a[n] + y_b[n]$, der...

$$y_a[n] = \beta \cdot \sum_{k=0}^{L-1} \cos(2\pi \cdot f_b \cdot k \cdot T_s) \cdot \cos(2\pi \cdot f_b \cdot k \cdot T_s) \cdot \cos(2\pi \cdot f_b \cdot n \cdot T_s + \varphi) \quad \text{og}$$

$$y_b[n] = \beta \cdot \sum_{k=0}^{L-1} \cos(2\pi \cdot f_b \cdot k \cdot T_s) \cdot \sin(2\pi \cdot f_b \cdot k \cdot T_s) \cdot \sin(2\pi \cdot f_b \cdot n \cdot T_s + \varphi)$$

De siste faktorene avhenger ikke av summasjonsvariabelen k , og settes utenfor summen:

$$y_a[n] = \beta \cdot \cos(2\pi \cdot f_b \cdot n \cdot T_s + \varphi) \cdot \sum_{k=0}^{L-1} \cos^2(2\pi \cdot f_b \cdot k \cdot T_s)$$

$$y_b[n] = \beta \cdot \sin(2\pi \cdot f_b \cdot n \cdot T_s + \varphi) \cdot \sum_{k=0}^{L-1} \cos(2\pi \cdot f_b \cdot k \cdot T_s) \cdot \sin(2\pi \cdot f_b \cdot k \cdot T_s)$$

Litt enkel trigonometrisk tricksing, og vi får:

$$y_a[n] = \beta \cdot \cos(2\pi \cdot f_b \cdot n \cdot T_s + \varphi) \cdot \sum_{k=0}^{L-1} \frac{1}{2} + \frac{1}{2} \cos(4\pi \cdot f_b \cdot k \cdot T_s)$$

$$y_b[n] = \beta \cdot \sin(2\pi \cdot f_b \cdot n \cdot T_s + \varphi) \cdot \sum_{k=0}^{L-1} \frac{1}{2} \sin(4\pi \cdot f_b \cdot k \cdot T_s)$$

Når L er stor (går mot uendelig) så ser vi at $y_b \rightarrow 0$, og at y_a blir slik:

$$y_a[n] \approx \beta \cdot \cos(2\pi \cdot f_b \cdot n \cdot T_s + \varphi) \cdot \frac{L}{2}$$

Dermed blir utgangssignalet $y[n]$ slik:

$$y[n] = y_a[n] + y_b[n] \approx \beta \cdot \frac{L}{2} \cdot \cos(2\pi \cdot f_b \cdot n \cdot T_s + \varphi)$$

Ut får vi altså en sinusoid med lik fase som den vi sendte inn. For å få amplituden lik 1 ser vi at konstanten β må få verdien $2/L$.

4.1 b)

Slik ser den ut, filen som vi lagde og som heter `dtmfdesign.m`:

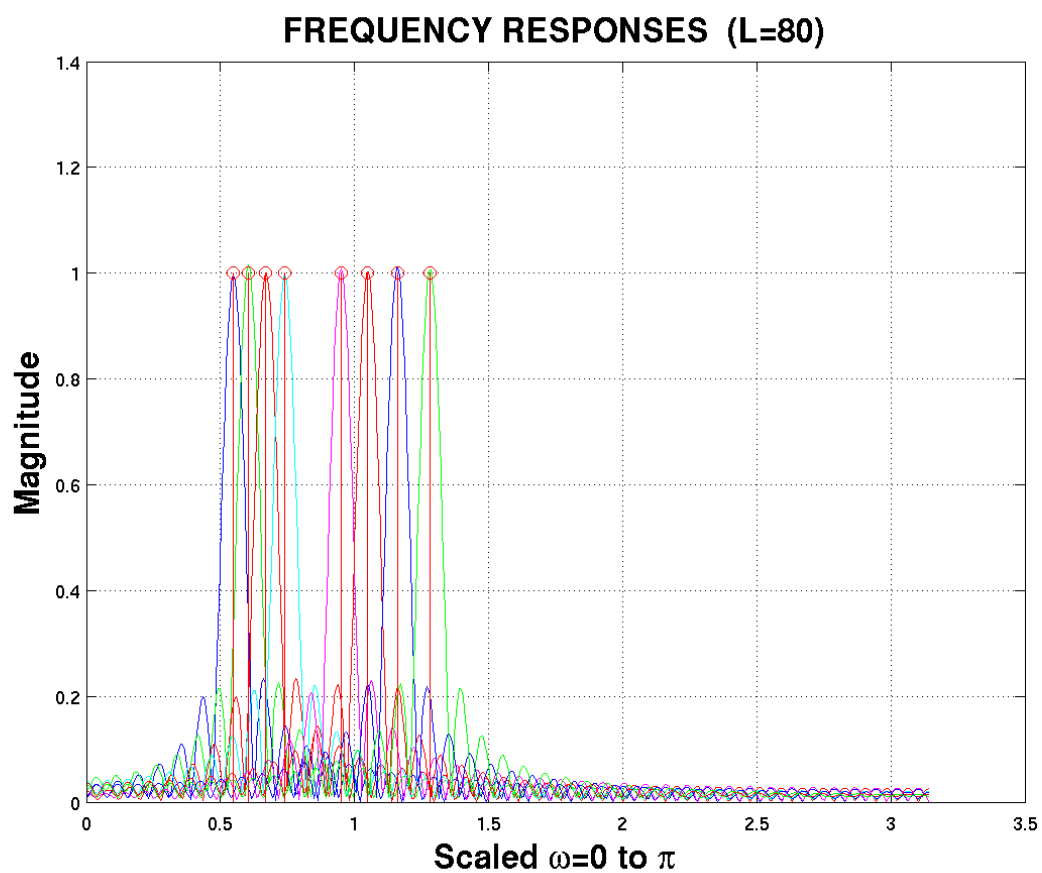
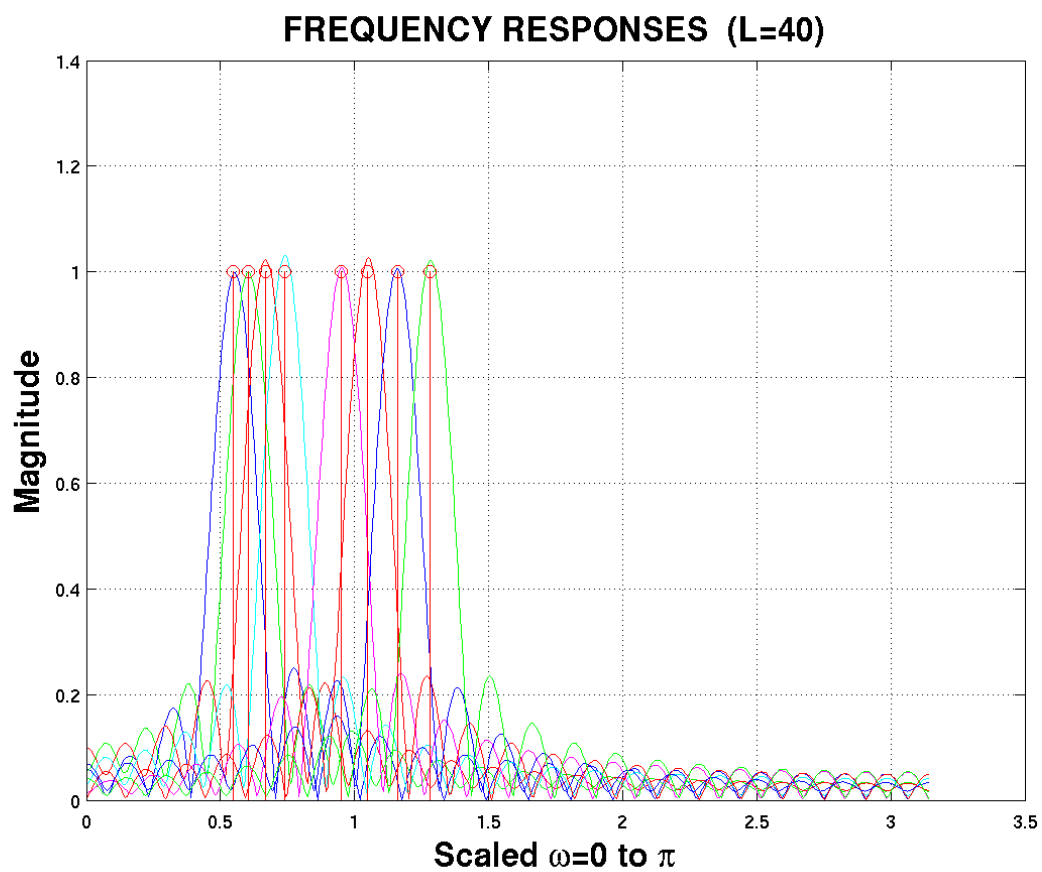
```
1 function hh = dtmfdesign(fb, L, fs)
2
3 for k = 1 : length(fb)
4     hh(k, :) = 2*cos(2*pi*fb(k)*(0:L-1)/fs)/L;
5 end
6
7 hh = hh';
```

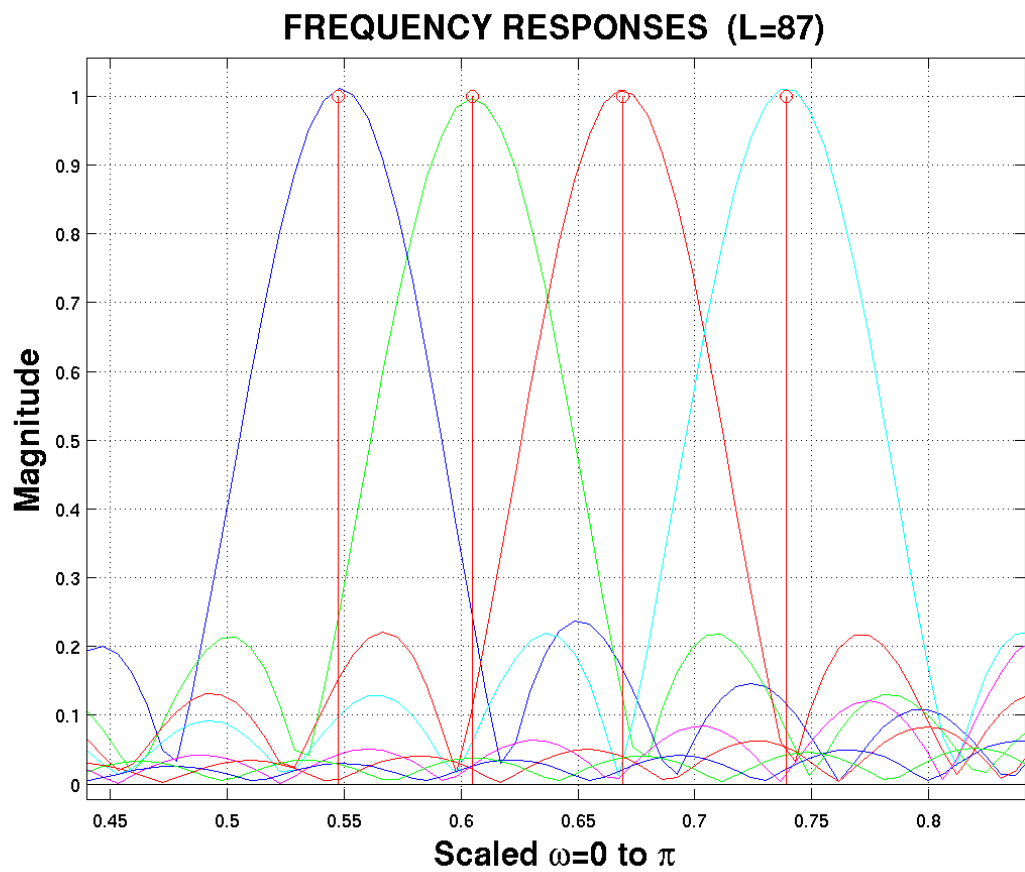
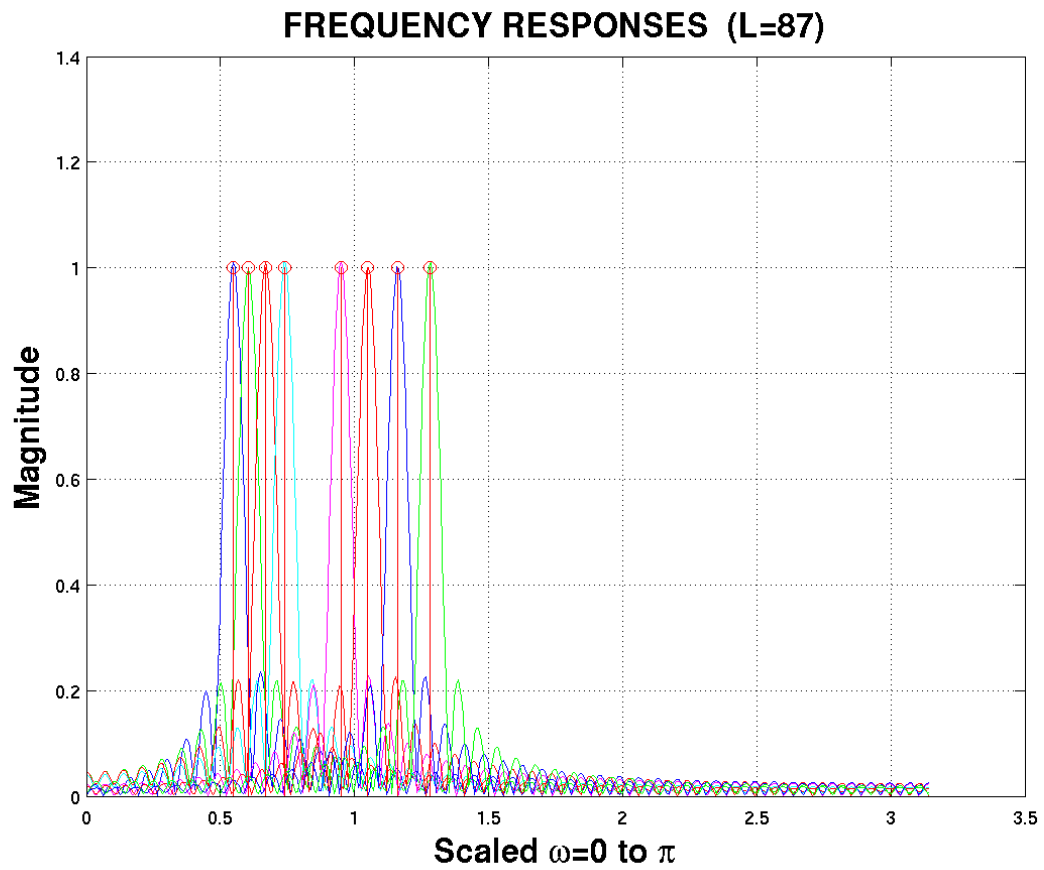
4.1 c,d,e,f)

For å se hvor stor L må være laget vi følgende Matlab script-fil som plotter frekvensresponsen til de 8 ulike filtrene i DTMF-systemet. Her er $L = 40$:

```
1 L = 40;
2 hh = dtmfdesign([697 770 852 941 1209 1336 1477 1633], L, 8000);
3 ww = linspace(0,pi,500);
4 FRESP = freqz(hh(:, 6), 1, ww);
5 plot(ww, abs(FRESP)), grid
6 colors = ['b', 'g', 'r', 'c', 'm', 'r', 'b', 'g'];
7 hold on
8 for j = 1 : 8
9     FRESP = freqz(hh(:, j), 1, ww);
10    plot(ww, abs(FRESP), colors(j)), grid
11    stem(2*pi*[697 770 852 941 1209 1336 1477 1633]/8000, ones(1,8), 'r')
12 xlabel('Scaled \omega=0 to \pi','FontSize',16,'FontWeight','bold')
13 ylabel('Magnitude','FontSize',16,'FontWeight','bold')
14 title(['FREQUENCY RESPONSES (L=' num2str(L) ')'],'FontSize',16,'FontWeight','bold')
15 end
```

Resultatene er vist på de neste sidene. Vi ser at de fire lave frekvenskomponentene kommer inn i passbåndet til nabofrekvensens filter når $L = 40$. Når $L = 80$ er dette problemet løst, men fortsatt ligger ikke de laveste frekvensene i stoppbåndet til alle de 7 andre filtrene. Vi må øke L til 87 for å også tilfredsstille dette kravet, dersom stoppbåndet er definert som området der frekvensresponsen har mindre enn $1/4$ amplitude (i forhold til den maksimale amplituden):





Vi bruker samme verdi av L for alle de 8 filtrene. Vi ser at det er filtrene for de laveste frekvensene som skaper problemer. Årsaken til dette er at hvis filterenes impulsrespons av en gitt lengde f.eks. tilsvarer tre perioder av en sinus på 700 Hz, så vil lengden tilsvare 6 perioder av en 1400 Hz sinus. Altså er impulsresponsene for de høye frekvensenes filtre bedre tilnærminger til ideelle filtre enn impulsresponsene for de lave frekvensenes filtre. Et utsnitt (rektangulært vindu) på 6 perioder av en uendelig sinus er ”mer ideelt” enn et utsnitt på bare 3 perioder.

Når vi sammenligner bredden på passbåndet til ett av filtrene ser vi at denne er omtrent omvendt proporsjonal til L . Ved $L = 40$ og $L = 80$ målte vi breddene (skalerte vinkelfrekvenser) til 0,141 og 0,070, henholdsvis.

4.2 a)

Slik ser vår versjon ut, av filen `dtmfscore.m`:

```
1 function sc = dtmfscore(xx, bb)
2 xx = xx*(2/max(abs(xx)));
3 yy = conv(bb', xx);
4 %plot(yy), grid
5 sc = (max(abs(yy)) >= 0.59);
```

4.2 b)

Når amplituden på det filtrerte signalet er $\geq 0,59$ blir den boolske testen på linje nr. 5 sann, og funksjonen returnerer verdien 1 (*sann*).

4.2 c)

Skalering av amplituden skjer i linje nr. 2. Hvis en DTMF-lyd er tilstede vil denne bestå av sinuser som ikke er harmoniske, dermed skal total amplituden maksimalt bli 2 dersom de to frekvenskomponentene er like sterke, hver med amplitude 1.

4.2 d)

Når inngangssignalet består av frekvenskomponenter som hver for seg har en amplitude på 1 og vi skal detektere om utgangssignalet overstiger 59 % ved å teste om $y \geq 0,59$, så forutsetter vi at filteret har enhets-forsterkning, dvs. at maksimal amplitude i frekvensresponsen er 1. For å ha kontroll med hva som kommer ut (y), så må vi vite hva vi putter inn (x), og hva vi putter det inn i (h).

4.3)

Slik ble vår versjon av filen `dtmfrun.m`, denne gangen også med kommentarer!

```
1 function keys = dtmfrun(xx,L,fs)
2 center_freqs = [697 770 852 941 1209 1336 1477 1633]; % Center frequencies
3 hh = dtmfdesign(center_freqs,L,fs); % Creates coefficients for the bandpass filters
4
5 dtmf.keys = ...
6 ['1','2','3','A';
7  '4','5','6','B';
8  '7','8','9','C';
9  '*', '0', '#', 'D'];
10
11 [nstart,nstop] = dtmfcut(xx,fs); % Detects the individual audio segments of the waveform
12 keys = [];
13 for kk=1:length(nstart) % One run per audio segment (per DTMF-tone)
14     x_seg = xx(nstart(kk):nstop(kk)); % Extract the audio
15     tones_detected = zeros(1,8); % Initialize with zeros...
16     for m=1:8 % Look for eight different tones:
17         tones_detected(m) = dtmfscor(x_seg,hh(:,m)); % Check the audio against each of the 8 filters
18     end % ..and make a '1' if the freq. component is present
19     lo_tones = tones_detected(1:4); % Binary vector representing 697, 770, 852 and 941 hz
20     hi_tones = tones_detected(5:8); % Binary vector representing 1209, 1336, 1477 & 1633 hz
21     if (l=sum(lo_tones) && l==sum(hi_tones)) % Test to see if there is one tone present from each group
22         lo_tones = sum( lo_tones .* [1 2 3 4] ); % Find row no. (1 to 4)
23         hi_tones = sum( hi_tones .* [1 2 3 4] ); % Find col. no. (1 to 4)
24     end %
25     keys = [keys, dtmf.keys(lo_tones,hi_tones)]; % Append DTMF-character from table to the string.
26 end
27
28 F_RESOLUTION = 1000;
29 ww = linspace(0,pi,F_RESOLUTION);
30 FRESP = freqz(xx, 1, ww);
31 ww = ww * 8000/(2*pi);
32 plot(ww(1:F_RESOLUTION/1.7),abs(FRESP(1:F_RESOLUTION/1.7)),grid
```

4.4)

For å teste hele systemet bruker vi følgende kommando i Matlab:

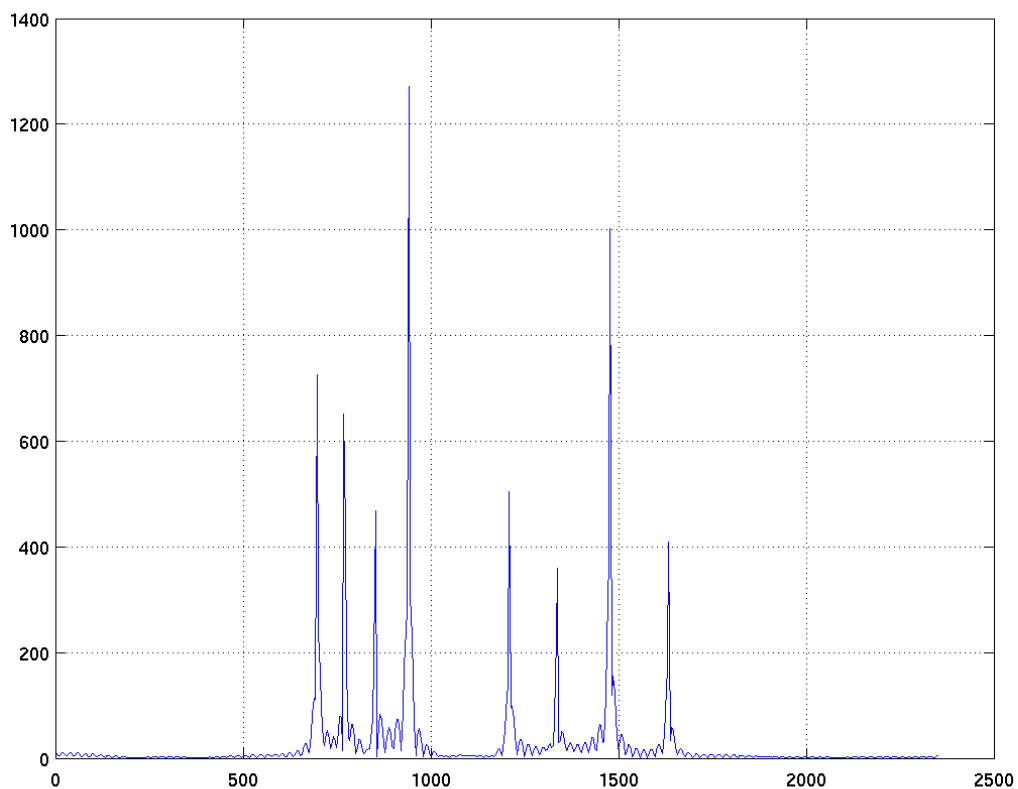
```
>> dtmfrun(dtmfdial('407*89132#BADC',8000),80,8000)

ans =

407*89132#BADC

>> |
```

Systemet fungerer som det skal, og returnerer den dekodete strengen med DTMF-karakterer. Linjene 28 til 32 i `dtmfrun.m` lager et plott som viser at alle de 8 frekvensene i DTMF-systemet er med i de DTMF-tonene som ble brukt i dette eksemplet, se neste side:



Plottet illustrerer tilstedeværelsen av alle de 8 frekvensene som brukes i DTMF-systemet. Vi skal ikke legge så mye vekt på de relative amplitudene her, ettersom fasene til komponentene i de enkelte "tastetrykkene" er noe tilfeldig. Altså får vi en salig blanding av mot- og medfase, som gir svakere eller sterkere amplituder. Uansett vil vi sjelden få total kansellering av frekvenskomponenter som er med, derfor vises alle frekvensene tydelig i plottet.

Erik Grindheim