Matlab Exercise 9:
Frequency Analysis of Signals.

IT University of Copenhagen
Multimedia Technology
Signal Processing Course

Objective: The objective of this exercise is to obtain fundamental understanding of the frequency-domain representation of signals, classified as periodic, aperiodic, and stochastic signals. The discrete Fourier transform is introduced as the computational tool used in frequency analysis, which in practical applications is based on the fast Fourier transform algorithm.

Literature:

– P&M, section 4.2, 5.1, pp. 247–256, 264–265, 399–403

9.1 Periodic Signals: A discrete-time periodic signal $x(n)$ with fundamental period N, that is $x(n) = x(n+N)$, can only consist of frequency components separated by $2\pi/N$ radians or $f = 1/N$ cycles. The basic frequency representation of periodic signals is the discrete-time Fourier series (DTFS), which is a linear weighted sum of these N harmonically related frequency components as defined in the synthesis/analysis equations

$$x(n) = \sum_{k=0}^{N-1} c_k e^{j2\pi kn/N} \quad \leftrightarrow \quad c_k = \frac{1}{N}\sum_{n=0}^{N-1} x(n) e^{-j2\pi kn/N}$$

The Fourier coefficients c_k for $0 \le k \le N-1$ provide the description of $x(n)$ in the frequency domain, in the sense that $c_k = |c_k|e^{j\theta_k}$ represents the amplitude and phase associated with the frequency component $e^{j\omega_k n}$ where $\omega_k = 2\pi k/N$. An example of a periodic signal is shown in Figure 1, and the corresponding amplitude and phase spectra are shown in Figure 2.

Note, that the Fourier coefficients c_k form a periodic sequence when extended outside of the fundamental range $0 \le \omega_k < 2\pi$ for $0 \le k \le N-1$, i.e., $c_{k+N} = c_k$, which is consistent with the fact that the highest relative frequency that can be represented by a discrete-time signal is equal to π. Furthermore, if the signal $x(n)$ is real, then the magnitude spectrum is even ($|c_{-k}| = |c_k|$), and the phase spectrum is odd ($\theta_{-k} = -\theta_k$).

Thus, the power in a periodic signal exist only at discrete values of frequencies as given by the power density spectrum $|c_k|^2$, and the signal is said to have a line spectrum. The average power is given by Parseval's relation for discrete-time periodic signals, i.e.,

$$P_x = \frac{1}{N}\sum_{n=0}^{N-1} |x(n)|^2 = \sum_{k=0}^{N-1} |c_k|^2$$

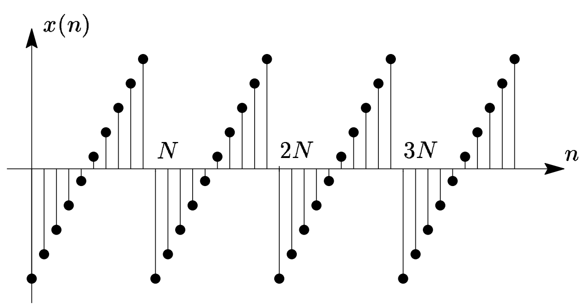


Figure 1 Discrete-time periodic sawtooth-wave signal ($N = 10$).

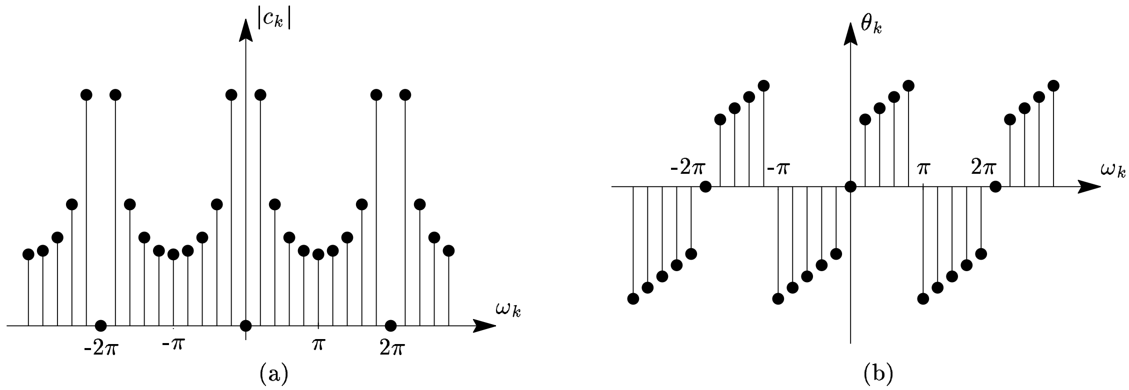


Figure 2 Magnitude and phase spectra of the periodic signal shown in Figure 1.

Consider the sinusoidal signal $x(n) = \cos(2\pi f_0 n + \phi)$ for the case $f_0 = \frac{1}{10}$ and $\phi = \frac{\pi}{6}$ as illustrated in Figure 3(a). Thus, $x(n)$ is periodic with fundamental period $N = 10$, and may also be expressed as

$$x(n) = \frac{1}{2}e^{j(2\pi n/10+\pi/6)} + \frac{1}{2}e^{-j(2\pi n/10+\pi/6)}$$

Hence, we expect two non-zero Fourier coefficients c_k.

9.1.1 Determine the spectra for the sinusoidal signal $x(n)$ in the range $-\pi \le \omega_k < \pi$ by using the commands (the use of the Matlab functions `fft` and `fftshift` to determine the discrete-time Fourier series is discussed in the next section)

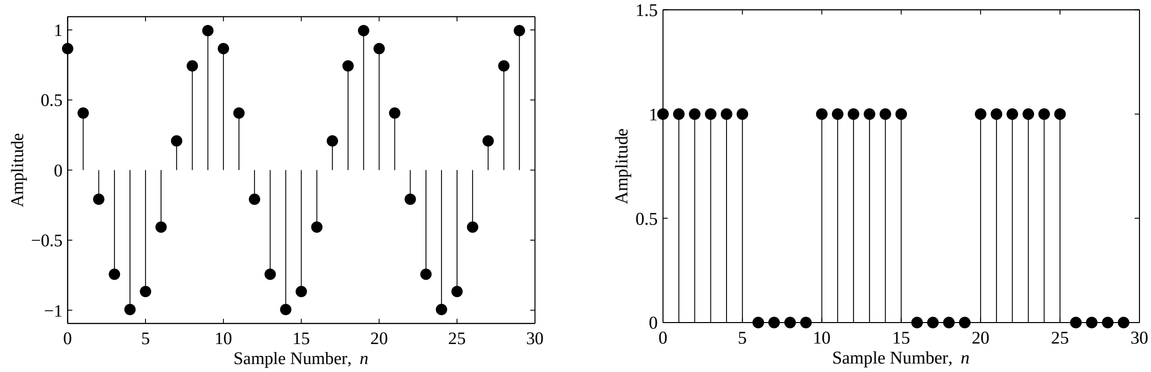
```
N = 10;
x = sinsig(3*N,1/N,1,pi/2+pi/6); n = 0:3*N-1;
stem(n,x)
ck = fftshift(fft(x(1:N)))/N; k = -N/2:N/2-1;
idx = find(abs(ck)<1e-12); ck(idx) = zeros(1,length(idx));
stem(k,abs(ck))
stem(k,angle(ck))
```

Verify that the magnitude and phase values are correct, and notice the symmetry properties for the spectra.

9.1.2 Try to use all three periods of the signal to determine the spectra, i.e.,

```
N = 30;
ck = fftshift(fft(x(1:N)))/N; k = -N/2:N/2-1;
idx = find(abs(ck)<1e-12); ck(idx) = zeros(1,length(idx));
stem(k,abs(ck))
stem(k,angle(ck))
```

Can you explain the difference?



(a) (b)

Figure 3 (a) Sinusoidal signal with $f_0 = \frac{1}{10}$ and $\phi = \frac{\pi}{6}$. (b) Periodic square-wave signal with $A = 1$, $L = 6$ and $N = 10$.

Now, let $x(n)$ be a discrete-time periodic square-wave signal, i.e.,

$$x(n) = \begin{cases} A, & n = 0, 1, \ldots, L-1 \\ 0, & n = L, \ldots, N-1 \end{cases}$$

as shown in Figure 3(b) for $A = 1$, $L = 6$ and $N = 10$. The Fourier series coefficients for $x(n)$ is given by

$$c_k = \frac{1}{N}\sum_{n=0}^{L-1} A e^{-j2\pi kn/N} = \begin{cases} \frac{AL}{N}, & k = 0 \\ \frac{A}{N} e^{-j\pi k(L-1)/N} \frac{\sin(\pi kL/N)}{\sin(\pi k/N)}, & k = 1, \ldots, N-1 \end{cases}$$

9.1.3 Determine the magnitude spectrum for the signal $x(n)$ when $N = 100$ and $L = 10$ by using the commands

```
N = 100; L = 10;
x = 0.5*rectsig(N,1/N,1,0,L/N*100) + 0.5;
ck = fftshift(fft(x(1:N)))/N; k = -N/2:N/2-1;
stem(k,abs(ck))
```

Verify that the DC value c_0 and the zeros ($c_k = 0$) agree with the theoretical formula.

9.1.4 Compute the average power of $x(n)$ using Parseval's relation

```
Px = ck*ck'
```

and verify that a direct time-domain approach `Px = x*x'/N` produces the same result.

9.1.5 Determine the magnitude spectrum for the periodic square-wave signal with $N = 20$ and $L = 10$, and compare it with the previous plot. Which harmonics ω_k (even and odd) are needed to represent a square-wave signal with duty-cycle 50 percent?

9.1.6 Plot the magnitude spectrum for the square-wave signal with $N = 20$ when $L = 2$ and $L = 1$ (pulse-train). Can you provide a theoretical explanation for the last result?

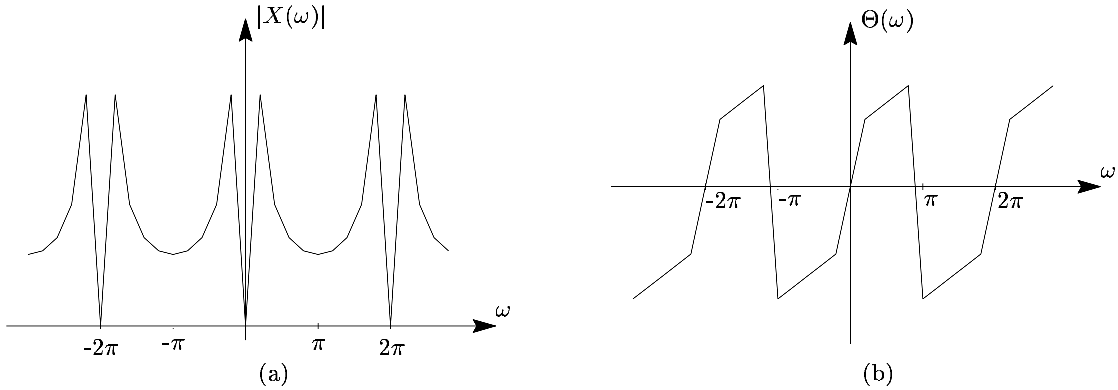


Figure 4 Magnitude and phase spectra of the aperiodic signal defined as one period of the sawtooth-wave signal in Figure 1.

9.2 Aperiodic Signals: The frequency analysis of discrete-time aperiodic finite-energy signals $x(n)$ involves a Fourier transform as defined in the synthesis/analysis equations

$$x(n) = \frac{1}{2\pi}\int_{2\pi} X(\omega)e^{j\omega n}d\omega \qquad \leftrightarrow \qquad X(\omega) = \sum_{n=-\infty}^{\infty} x(n)e^{-j\omega n}$$

where $X(\omega) = |X(\omega)|e^{j\Theta(\omega)}$ represents the frequency content of the signal $x(n)$. For example, the first N samples of the sawtooth-wave signal in Figure 1 is an aperiodic signal, resulting in the amplitude and phase spectra shown in Figure 4.

Note, that the Fourier transform $X(\omega)$ is periodic with period 2π, which is a consequence of the fact that the frequency range for any discrete-time signal is limited to $(-\pi, \pi)$. If the signal $x(n)$ is real, then the magnitude spectrum is even ($|X(-\omega)| = |X(\omega)|$), and the phase spectrum is odd ($\Theta(-\omega) = -\Theta(\omega)$). The distribution of energy as a function of frequency is given by the energy density spectrum $|X(\omega)|^2$, and the energy is given by Parseval's relation for discrete-time aperiodic signals, i.e.,

$$E_x = \sum_{n=-\infty}^{\infty} |x(n)|^2 = \frac{1}{2\pi}\int_{-\pi}^{\pi} |X(\omega)|^2 d\omega$$

The Fourier transform $X(\omega)$ is a continuous function of frequency, so in the digital domain, the sampled spectrum is used to represent an aperiodic signal of length L, which leads to the discrete Fourier transform (DFT) pair

$$x(n) = \frac{1}{N}\sum_{k=0}^{N-1} X(k)e^{j2\pi kn/N} \qquad \leftrightarrow \qquad X(k) = \sum_{n=0}^{N-1} x(n)e^{-j2\pi kn/N}$$

Consequently, the frequency samples $X(2\pi k/N)$ represents a periodic repetition of the finite duration sequence $x(n)$ (padded by $N - L$ zeros), and if $L \leq N$, it is possible to recover $x(n)$ from its periodic extension (no time-domain aliasing).

In Matlab, the DFT implementation is based on the fast Fourier transform algorithm `fft`. This function has already been used to find the spectra of periodic signals due to the fact that $c_k = X(k)/N$. Note, that the output of the `fft` function can be centered around $f = 0$ by using the function `fftshift`.

Now, let $x(n)$ be a discrete-time aperiodic square-wave signal, i.e.,

$$x(n) = \begin{cases} A, & n = 0, 1, \ldots, L-1 \\ 0, & \text{otherwise} \end{cases}$$

which has the Fourier transform

$$X(\omega) = \sum_{n=0}^{L-1} Ae^{-j\omega n} = Ae^{-j\omega(L-1)/2} \frac{\sin(\omega L/2)}{\sin(\omega/2)}$$

9.2.1 Determine the magnitude spectrum for $x(n)$ when $A = 1$, $L = 10$ and $N = 20$ by using the commands

```
N = 20; L = 10;
x = 0.5*rectsig(N,1/N,1,0,L/N*100) + 0.5;
X = fftshift(fft(x)); k = -N/2:N/2-1;
plot(k,abs(X))
```

Is this a good display of the continuous magnitude spectrum $X(\omega)$?

9.2.2 Illustrate the effect of zero padding using the square-wave and the FFT, e.g.,

```
N = 128;
X = fftshift(fft(x,N)); k = -N/2:N/2-1;
plot(k,abs(X))

N = 1024;
X = fftshift(fft(x,N)); k = -N/2:N/2-1;
plot(k,abs(X))
```

Is the frequency resolution enhanced by padding zeros?

Limiting the duration of a sequence $x(n)$ to L samples is equivalent to multiplying $x(n)$ by a rectangular window $w(n)$ of length L. That is

$$\hat{x}(n) = x(n)w(n) \quad \leftrightarrow \quad \hat{X}(\omega) = \frac{1}{2\pi}\int_{-\pi}^{\pi} X(\theta)W(\omega - \theta)d\theta$$

Thus, the above mentioned square-wave signal can be considered as a constant signal $x(n) = A$ multiplied by a rectangular window of length L. In the frequency domain, this is the delta function $X(\omega) = A\delta(\omega)$ convolved with the Fourier transform of the window sequence

$$W(\omega) = e^{-j\omega(L-1)/2} \frac{\sin(\omega L/2)}{\sin(\omega/2)}$$

so when we pad zeros to the square-wave signal, we obtain a better display (more spectral samples) of the rectangular window $W(\omega)$, however, a closer resemblance to the spectrum of the constant signal is not obtained. The magnitude spectrum $|\hat{X}(\omega)|$ is shown in Figure 5 for $L = 10$ and $N = 1024$, and we note that the power of the original signal $X(\omega) = A\delta(\omega)$ that was concentrated at a single frequency $f = 0$ has leaked out into the entire frequency range due to the windowing.

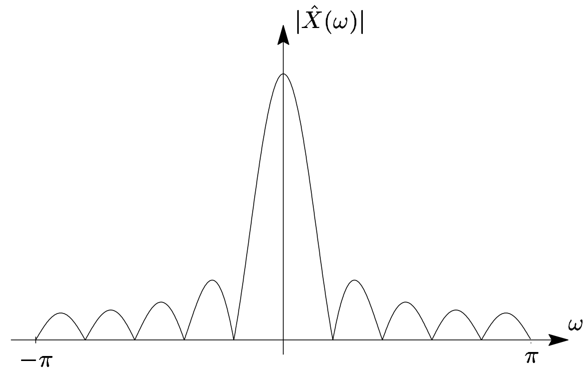


Figure 5 Magnitude spectrum for the square-wave signal with $L = 10$ and $N = 1024$, illustrating the occurrence of leakage.

Another effect of windowing is reduced spectral resolution, since the ability to resolve spectral lines of different frequencies is limited by the window main lobe width. This can be illustrated for the following signal

$$x(n) = \cos(2\pi f_0 n) + \cos(2\pi f_1 n) + \cos(2\pi f_2 n)$$

where $f_0 = 0.1$, $f_1 = 0.11$, and $f_2 = 0.3$.

9.2.3 Determine the magnitude spectrum for $x(n)$ when the window length selected are $L = 25$, 50 and 100, i.e., by using the commands

```
N = 1024; k = -N/2:N/2-1;
x = sinsig(N,[0.1 0.11 0.3],[1 1 1],[0 0 0]);
X1 = fftshift(fft(x(1:25),N)); plot(k,abs(X1))
X2 = fftshift(fft(x(1:50),N)); plot(k,abs(X2))
X3 = fftshift(fft(x(1:100),N)); plot(k,abs(X3))
```

In which cases can the spectral lines be resolved? Which relationship exists between time resolution and frequency resolution?

9.2.4 Other window functions can be used which has lower sidelobes (reduced leakage) at the expense of an increase in the width of the main lobe (reduced resolution). Determine the magnitude spectrum for $x(n)$, using a Hamming window with length $L = 25$, 50 and 100, i.e.,

```
X4 = fftshift(fft(x(1:25).*hamming(25)',N)); plot(k,abs(X4))
X5 = fftshift(fft(x(1:50).*hamming(50)',N)); plot(k,abs(X5))
X6 = fftshift(fft(x(1:100).*hamming(100)',N)); plot(k,abs(X6))
```

Compare the sidelobe level and the main lobe width with those of the rectangular window.

Clearly, when the number of samples L is small, the window function has a wide spectrum and will therefore have a smoothing effect on the spectrum $X(\omega)$, i.e., the DFT of the data reflects the spectral characteristics of the window function. Of course, this situation should be avoided.

Window functions are for example used to analyze dynamic (time-varying) signals like speech. Figure 6 shows a simplified block diagram of the speech production process, where the frequency response of the vocal tract filter $H(z)$ determines the short-time spectral envelope of the speech signal. $H(z)$ can be described by an 8–12 order all-pole filter, and the excitation signal depends on whether the speech sound is voiced or unvoiced. Figure 7(a) shows a voiced part of the speech signal `ma1_1` (sample numbers 4161–4460), and Figure 7(b) shows the corresponding magnitude response $|H(\omega)|$.

9.2.5 Determine the DFT based magnitude spectrum of the voiced sound by using the commands

```
load ma1_1;
x = ma1_1(4161:4460); plot(x)
N = 1024; k = -N/2:N/2-1;
X = fftshift(fft(x.*hann(length(x)),N));
plot(k,20*log10(abs(X))), axis([0 512 -40 40])
```

Can you identify the resonant frequencies (formants)? How does the pitch period influence the spectrum?

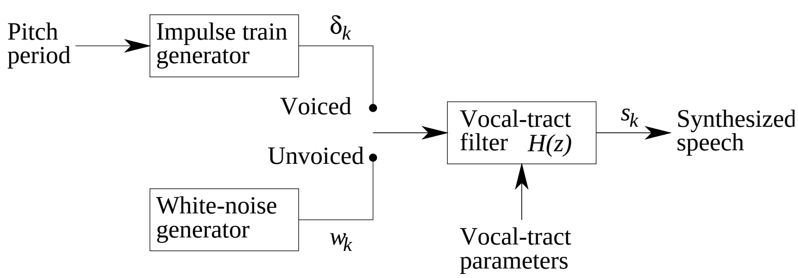


Figure 6 Model for the speech production process.

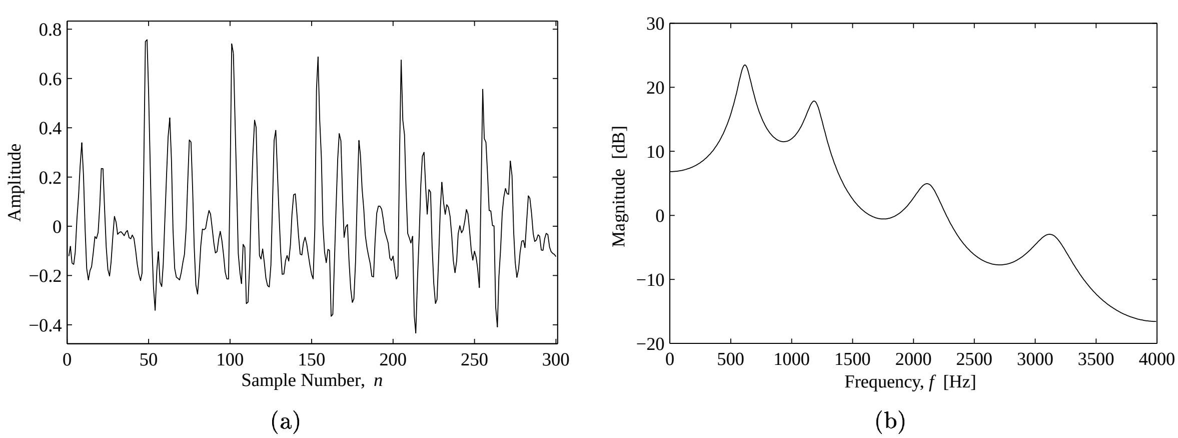


Figure 7 (a) Amplitude waveform of voiced speech frame corresponding to the /Y/ sound in "prices". (b) 10th order model-based magnitude spectrum.

9.3 Stochastic Signals: The spectral characteristic, i.e, the power density spectrum $\Gamma_{xx}(f)$, of a discrete-time stochastic signal is obtained by computing the Fourier transform of the autocorrelation function $\gamma_{xx}(m)$, that is

$$\gamma_{xx}(m) = \int_{-1/2}^{1/2} \Gamma_{xx}(f) e^{j2\pi fm} df \qquad \leftrightarrow \qquad \Gamma_{xx}(f) = \sum_{n=-\infty}^{\infty} \gamma_{xx}(m) e^{-j2\pi fm}$$

It follows that $\Gamma_{xx}(f)$ is the distribution of power as a function of frequency. In practice, we are dealing with finite-duration sequences $x(n) = (x(0), x(1), \ldots, x(N-1))$, and the autocorrelation values may be estimated as

$$\hat{\gamma}_{xx}(m) = \frac{1}{N-m} \sum_{n=m}^{N-1} x(n)x(n-m), \qquad 0 \leq m \leq N-1$$

Autocorrelation function estimates as given by the above mentioned formula can be obtained by using the Matlab function

```
[gamma,m] = xcorr(x,mmax,'unbiased')
```

9.3.1 The autocorrelation function for white noise (with variance one) as generated by the Matlab function `randn` is given by:

$$\gamma_{ww}(m) = \delta(m) \qquad \leftrightarrow \qquad \Gamma_{ww}(f) = 1$$

Verify this result, e.g., by using the following Matlab commands

```
w = randn(5000,1);
[gamma,m] = xcorr(w,100,'unbiased');
plot(m,gamma);
N = 1024; k = -N/2:N/2-1;
Gamma = fftshift(fft(gamma.*hann(length(gamma)),N));
plot(k,20*log10(abs(Gamma)))
```

9.3.2 Let $x(n)$ be a lowpass filtered white noise signal with bandwidth $f_g = 0.1$. Then estimate the power density spectrum $\Gamma_{xx}(f)$ using the following Matlab commands

```
b = fir1(50,0.2);
x = filter(b,1,w);
[gamma,m] = xcorr(x,100,'unbiased');
plot(m,gamma);
Gamma = fftshift(fft(gamma.*hann(length(gamma)),N));
plot(k,20*log10(abs(Gamma)))
```

9.3.3 In practice, power density spectra can be obtained by using the Matlab function `spectrum` which is based on averaging in the frequency domain instead of averaging in the time domain (the correlation sum). Try for example the command

```
spectrum(x)
```

9.4 Home Work: Consider the two sinusoidal signals $x_1(n)$ and $x_2(n)$ with frequencies $f_1 = 0.01$ and $f_2 = 0.2$, respectively.

9.4.1 Generate the two signals, and plot the product signal $x(n) = x_1(n)x_2(n)$, i.e.,

```
N  = 1000;
x1 = sinsig(N,0.01,1,0)';
x2 = sinsig(N,0.2,1,0)';
x  = x1.*x2;
plot(x), axis([0 100 -1 1])
```

9.4.2 Plot the magnitude spectra for each of the three signals $x_1(n)$, $x_2(n)$, and $x(n)$. Which of the frequencies f_1, f_2, $f_1 - f_2$, and $f_1 + f_2$ are represented in the product signal $x(n)$?

9.4.3 Obviously, the low-frequency signal $x_1(n)$ has been translated to higher frequencies by multiplication with the high-frequency signal $x_2(n)$. The signal $x_1(n)$ can be translated back again by multiplying $x_2(n)$ once more, i.e.,

```
y = x.*x2;
plot(y), axis([0 100 -1 1])
```

Is the signal $y(n)$ identical to $x_1(n)$?

9.4.4 Plot the magnitude spectrum for the signal $y(n) = x(n)x_2(n)$, and consider how the signal $x_1(n)$ can be obtained from $y(n)$.