EE3209 - Communications Laboratory

Lab Manual

California State University, Los Angeles
College of Engineering, Computer Science, and Technology
Department of Electrical and Computer Engineering

Instructor: Khashayar Olia

Lab 6: Frequency Modulation

Objective:

This laboratory exercise introduces Frequency Modulation (FM). This lab exercise is a nice illustration of the utility of the software-defined radio approach, since the algorithms for creating and demodulating FM in software are much simpler than those used in the traditional hardware approach.

Background:

Frequency Modulation

Frequency modulation (FM) is one of the method of **Angle Modulation** in which the angle of the carrier signal is varied according to the message signal. FM was introduced by E.A. Armstrong in the 1930's as an alternative to the AM commonly in use at the time for broadcasting. The advantage to frequency modulation is that, for a given transmitted power, the **Signal-to-Noise Ratio (SNR)** is much higher at the receiver output than it is for AM, however this improvement is achieved at the expense of increased transmission bandwidth. The digital version of FM, **Frequency-Shift Keying (FSK)**, has been in use since an even earlier date.

Phase Modulation (PM) is another form of Angle Modulation in which the **instantaneous angle** $\varphi_i(\mathrm{t})$ is varied with the message signal. This lab will be focusing much of discussion on FM modulation only, because both PM and FM are uniquely related to each other and it's possible to derive one from another [1]. In FM, the **instantaneous frequency** of the carrier $f_i(t)$ is modulated to follow the amplitude of the message signal. If we assume $\varphi_i(\mathrm{t})$ as the angle of modulated carrier signal, the general form of an angle modulation signal $s(t)$ is:

$$s(t) = A_c\, cos\big(\varphi_i(\mathrm{t})\big)$$

Eq. 1

where A_c is the amplitude of the carrier signal. The instantaneous frequency of the signal $f_i(t)$ is defined as the following equation:

$$f_i(t) = \lim_{\Delta t \to 0} \left[\frac{\varphi_i(t+\Delta t) - \varphi_i(t)}{2\pi \Delta t} \right]$$

$$= \frac{1}{2\pi} \frac{d\varphi_i(t)}{dt}$$

Eq. 2

In which $\varphi_i(t)$ is instantaneous angle which increases monotonically with time. Figure 1 compares AM and FM modulation with the same message signal:

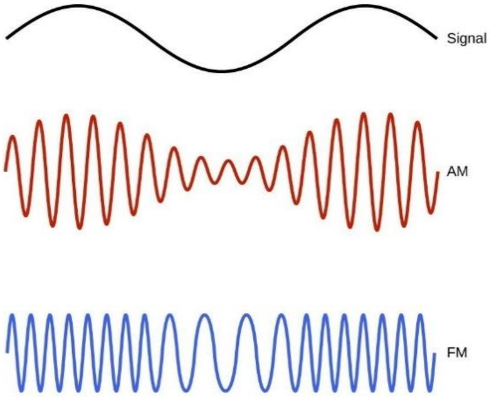

Figure 1: FM and AM Modulation

As it's clear from Figure 1, the instantaneous frequency, $f_i(t)$, of the carrier signal varies linearly with the message signal. We also realize that it's not possible to visualize the message waveform from the modulated waveform (while this is not the case for AM) because of the nonlinearity character of the FM signal [1]. The nonlinearity is one of the property of the FM which results in a superior noise performance compared to AM. The FM signal is actually less sensitive to the presence of additive noise than AM is.

In order to be more specific in constructing the FM signal, if $m(t)$ is a message signal with peak value m_p, then the instantaneous frequency $f_i(t)$ of the carrier is given by:

$$f_i(t) = f_c + k_f m(t)$$

Eq. 3

where f_c is the carrier frequency and k_f is a proportionality constant called the **frequency sensitivity**. The term $k_f m(t)$ is called the **frequency deviation** of the instantaneous frequency from the carrier frequency.

The maximum instantaneous frequency change from the carrier frequency produced by the modulating signal is given by:

$$\Delta f_{i_{max}} = max[f_i(t) - f_c] \Longrightarrow$$
$$= max\left[k_f m(t)\right] = k_f m_p(t)$$

Eq. 4

The Eq. 4 is an important FM system parameter and is called **peak frequency deviation**. Given the instantaneous frequency and considering the Eq. 2, we can find the total instantaneous angle $\varphi_i(t)$ of the carrier by integrating the instantaneous frequency. That is:

$$\varphi_i(t) = 2\pi \int_0^t f_i(\tau)d\tau = 2\pi f_c t + 2\pi k_f \int_0^t m(\tau)d\tau + \varphi(0)$$

Eq. 5

Since the initial angle $\varphi(0)$ is of no consequence, we can simplify the equations by taking $\varphi(0) = 0$. Using Eq. 5, the transmitted FM signal can be expressed in terms of its instantaneous frequency and is then given by:

$$s(t) = A_c\, cos(\varphi) \Longrightarrow$$
$$= A_c\, cos\left[2\pi f_c t + 2\pi k_f \int_0^t m(\tau)d\tau\right] \Longrightarrow$$
$$s(t) = A_c\, cos\left[2\pi f_c t + \underbrace{2\pi\Delta f_{max} \int_0^t \left(\frac{m(\tau)}{m_p}\right) d\tau}_{\theta(t)}\right]$$

Eq. 6

Note that the second term in Eq. 6 represents $\theta(t)$ of FM signal. Thus, the instantaneous phase is the integral of a normalized message multiplied by a peak frequency deviation (Δf_{max}). Figure 2 shows the FM block diagram which illustrates how FM signal is generated using phase modulator. As it’s clear, the integral of the message signal $m(t)$ generates the phase of the carrier signal at frequency f_c.

To create an FM signal using the USRP, the message signal is normalized to a peak value of one $\left(\frac{m(\tau)}{m_p}\right)$,

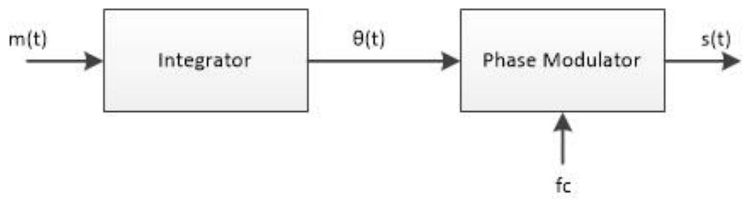

Figure 2: Generating FM Signal Using Phase Modulator

then multiplied by $2\pi\Delta f_{max}$ and integrated to give $2\pi\Delta f_{max}\int_0^t\left(\frac{m(\tau)}{m_p}\right)d\tau$. Next, the complex-valued signal $\tilde{s}(t)$ is formed, where:

$$\tilde{s}(t) = A_c e^{\overbrace{j2\pi\Delta f_{max}\int_0^t\left(\frac{m(\tau)}{m_p}\right)d\tau}^{\theta(t)}}$$

Eq. 7

The complex-valued signal $\tilde{s}(t)$ is sent to the **Write Tx Data** VI, and the USRP produces the FM signal.

Demodulation

An FM demodulator recovers the message signal from the received FM waveform. This requires a circuit that produces an output that is linearly proportional to the instantaneous frequency of the input FM signal. FM demodulation is much easier to carry out using the USRP than it is using conventional hardware. The FM demodulation can be divided into three broad categories: **Frequency discrimination**, **Phase-shift discrimination**, and **Phase-locked loop (PLL)** [1]. This lab focuses solely on frequency discrimination.

The signal provided by the **Fetch Rx Data** VI, $\tilde{r}(t)$ is given by:

$$\tilde{r}(t) = A_r e^{j2\pi\Delta f_{max}\int_0^t\left(\frac{m(\tau)}{m_p}\right)d\tau}$$

Eq. 8

This is identical to the expression given by Eq. 7, except for the magnitude A_r and the presence of noise (not shown in the expression). The idea of FM demodulator using frequency discrimination is to extract the angle of received signal and pass it through a differentiator to extract the message. Figure 3 illustrates the simple architecture of FM demodulator in USRP:

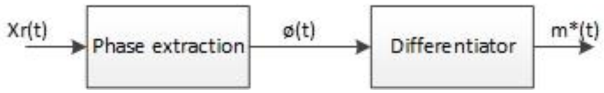

Figure 3: FM Demodulator Simple Architecture

Prelab

Transmitter

A template for the transmitter has been provided in the file **“Lab6TxTemplate.vit”**. This template contains the four interface VI’s along with a **message generator** that is set to produce a message signal consisting of three tones. The three tones are initially set to 1, 2, and 3 kHz, but these frequencies can be changed using front-panel controls. Your task is to add blocks as needed to produce the frequency modulated signal $\tilde{s}(t)$ of Eq. 7 and then to pass this signal to the Write Tx Data VI. In order to make the transmitter ready to run, follow the following steps. ***Note that steps 1 through 3 is being constructed outside of the “While Loop”.***

1. Use the outputs of **Get Waveform Components** block to generate your frequency modulation. In order to make the message signal normalized to a peak value of one $\left(\frac{m(\tau)}{m_p}\right)$, use **Quick Scale** VI:

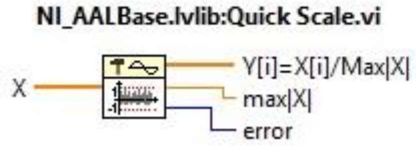

Figure 4: Quick Scale VI

2. In order to compare the modulated and demodulated signals at the Tx and Rx, convert the data values of the scaled message signal into a waveform using **Build waveform** VI, then:
    I. Plot the scaled message waveform (output of the Build waveform block) using the **Waveform Graph**. Now you can compare the results at Tx and Rx.
    II. Don’t forget to set the sampling time (dt) same as that of the original message signal.
3. Now, you have the term $\left(\frac{m(\tau)}{m_p}\right)$ and based on the Eq. 7 you need to do the integral. As we said, the complex-valued signal $\tilde{s}(t)$ is sent to the Write Tx Data, and the USRP produces the FM signal. The only tricky step in generating an FM signal is integrating the message. First of all, you have to convert the normalized analog signal (output of the Quick Scale block) to discrete time. In discrete

time, we can write:

$$\int_{0}^{t} x(\alpha)d\alpha \cong \sum_{k=0}^{n} x(nT) \times T$$

Eq. 9

where T is the sampling time or reciprocal of the IQ sample rate in our case.

I. If we consider the following equation:

$$y[n] = \sum_{k=0}^{n} x(nT) \times T$$

Eq. 10

Then we can write:

$$y[n] - y[n-1] = \sum_{k=0}^{n} x(nT) \times T - \sum_{k=0}^{n-1} x(nT) \times T = x(nT)$$

Eq. 11

Eq. 11 is a recurrence relation, or the difference equation of an **Infinite Impulse Response (IIR)** filter. The IIR filter is a filter whose impulse response does not become zero after a certain point. You can find more information about the IIR filters in [2]. This filter can be implemented using the IIR Filter VI found in the Functions > Signal Processing > Filters > Advanced IIR Filtering palette. ***Note that from this step on, you need to construct the blocks inside the "While Loop".***

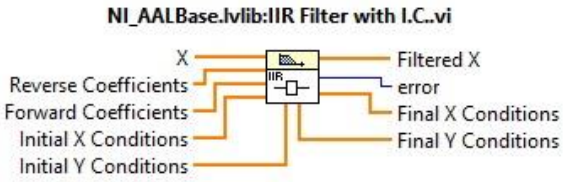

Figure 5: IIR Filter VI

The IIR filter with initial condition (Figure 5) filters the input sequence by using the direct form of IIR filter specified by reverse and forward coefficients.

II. Place this filter inside the While Loop and feed the input “X” with the output of Quick Scale Block.

III. Use a **Forward Coefficients** array of $[1]$ and a **Reverse Coefficients** array of $[1, -1]$. In order to use array for inputs of the filter see Functions > Programming > Array > Array Constant.

IV. Fill the array using **Numeric Constant** with data type double:

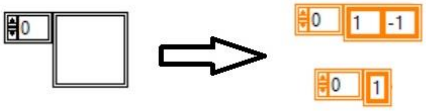

Figure 6: Array Constant Block

4. You need to define the initial and final condition to run the IIR filter appropriately. The best way to define those conditions is to use the register. Using register helps you to initialize your IIR filter with a predefine value, save the final condition after each loop iteration and use the final condition of last loop iteration as the initial condition for the next loop iteration. In order to use the register:

I. Right click on While Loop and select **Add Shift Register**. Use this option and add two shift registers to your While Loop.

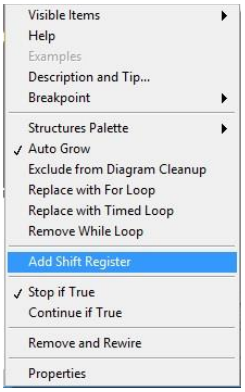

Figure 7: Adding Shift Register to a While Loop

II. Wire the **Final Y Condition** to the right shift register and **Initial Y Condition** to the left one.

III. Initialize shift registers to a zero constant using array constant. These registers allow you to use the final conditions of current loop iteration as the initial conditions of the next loop iteration for the IIR filter.

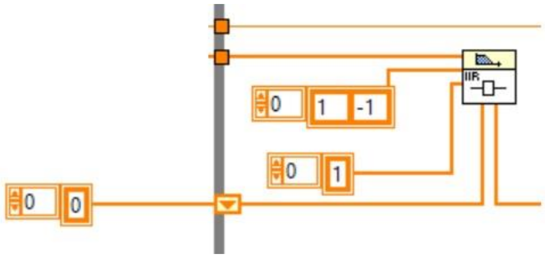

Figure 8: Initializing the IIR Filter

5. The IIR filter provides the discrete filtered $m(t)$ as $m(nT)$. Now you need to multiply $m(nT)$ with $2\pi\Delta f_{max}T$ to produce $\theta(t)$:

$$\theta(t) = 2\pi\Delta f_{max}\sum_{k=1}^{n} m(nT) \times T$$

Eq. 12

   I. The peak frequency deviation $2\pi\Delta f_{max}$ has been provided for you with the following blocks:

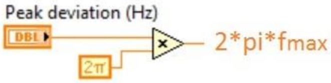

Figure 9: Peak Frequency Deviation

   II. Use the sample rate of **Get Waveform Component** VI as T and multiply it with the Peak Frequency Deviation.

   III. Multiply the output of the IIR filter with $2\pi\Delta f_{max} \times T$.

6. Now you have the $\theta(t)$ as Eq. 12 and you need to construct the Eq. 7. In order to form the complex value frequency modulated signal, use the **Polar to Complex** VI (Functions > Mathematics > Numeric > Complex > Polar to Complex). This block reads the inputs of A_c and $\theta(t)$ and generate the complex value. For A_c use constant value equals to 1.

7. Remember that the output signal generated by the IIR filter is a sequence, so you need to convert it back into an analog waveform before sending it to the buffer, Write Tx Data VI by using the

Build Waveform VI.

8. Save your transmitter in a file whose name includes the letters **FMTx** and your initials. Be sure to save your transmitter as a **vi** and not a **vit** (template). ***Note: You need to upload your program file with your lab report.***

Receiver

A template for the receiver has been provided in the file **"Lab6RxTemplate.vit"**. This template contains the six interface VI's along with a waveform graph to display your demodulated output signal. Your task is to add blocks as needed to demodulate the received signal $\tilde{r}(t)$ of Eq. 7 returned by the Fetch Rx Data VI and display the result by doing the following steps:

1. Use the output of Get Waveform Component to read the $\tilde{r}(t)$.
2. Extract the angle of $\tilde{r}(t)$ using the Complex to Polar VI.
3. Unwrap the angle with the **Unwrap Phase** VI. This block removes jumps of $\pm 2\pi$ on the phase (Functions > Signal Processing > Signal Operations palette).

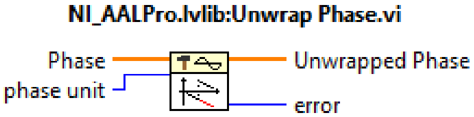

Figure 10: Unwrap Phase VI

4. The Unwrap Phase block gives you the phase of the received signal as Eq. 13:

$$\theta(t) = 2\pi\Delta f_{max} \int_{0}^{t} \left(\frac{m(\tau)}{m_p}\right) d\tau$$

Eq. 13

Now you need to differentiate the phase angle to extract the message. To implement the differentiator, we recognize that in discrete time, we have:

$$\frac{dx(t)}{dt} \cong \frac{x[n] - x[n-1]}{T}$$

Eq. 14

Considering the Eq. 14, the differentiator can be implemented using a **Finite Impulse Response (FIR)** filter. (Functions > Signal Processing > Filters > Advanced FIR Filtering palette). In signal processing, FIR filter is a filter whose impulse response duration is finite. The impulse response of an nth order FIR filter last exactly $n + 1$ samples before it goes to zero. For more information about the FIR filter, see here [2].

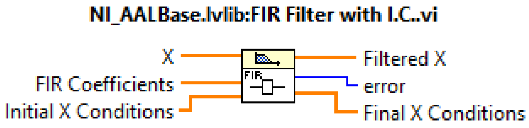

Figure 11: FIR Filter VI

For the **FIR coefficients** array, you need to use $\left[{}^{1}/_{T}, {}^{-1}/_{T}\right]$ to be able to implement the differentiator. T is the sampling interval coming from Get Waveform Component VI. In order to build this array follow below steps:

I. Read sample time dt from Get Waveform Component VI which gives you T.

II. Use **Reciprocal** VI to generate ${}^{1}/_{T}$ (Functions > Programming > Numeric > Reciprocal).

III. Use **Negate** VI to generate ${}^{-1}/_{T}$ (Functions > Programming > Numeric > Negate).

IV. Use **Build Array** VI to generate $\left[{}^{1}/_{T}\quad {}^{-1}/_{T}\right]$ by wiring the output of steps I and II into element inputs of Build Array VI (Functions > Programming > Array > Build Array).

V. Use the output of the Build Array for the FIR Coefficient input of the FIR filter.

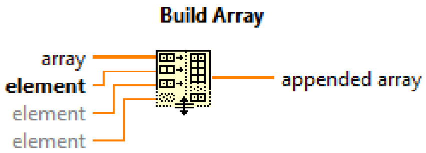

Figure 12: Build Array VI

5. **Initial X Condition** and **Final X Condition** need to be wired correctly with shift registers. Right click on While Loop and select Add Shift Register:

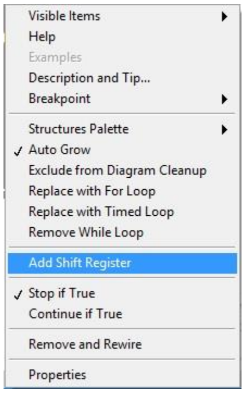

Figure 13: Adding Shift Register to a While Loop

I. This option adds two shift registers to your While Loop. Wire the Final X Condition to the right shift register and Initial X Condition to the left one.

II. Initialize the left shift register to zero constant using array constant. These registers allow you to use the final conditions of the current loop iteration as the initial conditions of the next loop iteration for the FIR filter.

6. The output of the differentiator (FIR filter) is the unwrapped differentiated angle:

$$2\pi\Delta f_{max}\left(\frac{m(t)}{m_p}\right) = 2\pi k_f m(t)$$

Eq. 15

This output is then passed into an envelope detector; a low-pass filter would be best since differentiation tends to enhance high-frequency noise. In this example, we use the **Butterworth Filter** VI, but any low-pass filter will produce usable results.

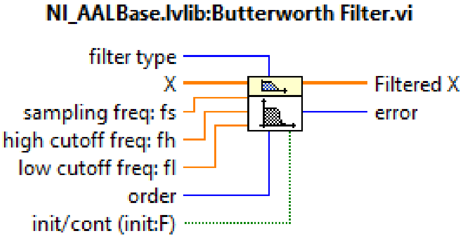

Figure 14: Butterworth Filter VI

Set the filter type as low- pass with below settings:

Table 1: Butterworth Filter Setting

Settings	Values
Filter Order	5
Low Cut-Off Frequency	5000 Hz
Sampling Rate	reciprocal of sampling time

Note that you have already created the reciprocal of sampling time in step 4, so you can easily use that output.

7. Connect the output of the LPF to the Build Waveform VI to construct the waveform. Set the sampling time interval to be the same as that of the FM signal fetched by the buffer.
8. Plot the waveform and compare it with the original message signal you modulated and transmitted.
9. Plot the spectrum of the demodulated signal using the **FFT Power Spectrum and PSD** VI as well.
10. Save your receiver in a file whose name includes the letters **FMRx** and your initials. Be sure to save your transmitter as a **vi** and not a **vit** (template). ***Note: You need to upload your program file with your lab report.***

Lab Procedure

1. Connect a loopback cable and attenuator between the TX 1 and RX 2 antenna connectors. Connect the USRP to your computer with an Ethernet cable and plug in the power to the radio. Run LabVIEW and open the transmitter and receiver VI’s that you created in the prelab.
2. Ensure the transmitter and receiver VI’s are set up based on Table 2, Table 3 and Table 4.
3. Run the transmitter VI, then run the receiver VI. LED “C” will illuminate on the USRP if the radio is receiving data. After a few seconds, stop the receiver using the STOP button, then stop the transmitter (using the STOP button). Use the horizontal zoom feature on the graph palette to expand the message waveform in the transmitter VI and the demodulated output waveform in the receiver. Both waveforms should be identical, except for scaling. (0.5 point)

Table 2: Transmitter Settings

Field	Settings
Device Name	192.168.10.2
IQ Rate	1 MS/sec
Carrier Frequency	915 MHz
Gain	0 dB
Active Antenna	TX1
Peak Deviation	30000 Hz

Table 3: Transmitter Message Settings

Field	Message Settings
Start Frequency	1 kHz
Delta Frequency	1 kHz
Number of Tones	1
Modulation Index	1
Message Length	200,000 samples

Table 4: Receiver Settings

Field	Settings
Device Name	192.168.10.2
IQ Rate	1 MS/sec
Carrier Frequency	915 MHz
Gain	0 dB
Active Antenna	RX2
Number of Samples	200,000

4. Are you able to detect the frequency of your message signal $m(t)$ at the receiver? Plot the result of

the spectrum analyzer in your report. (0.5 point)

5. The spectrum of the FM signal contains a carrier frequency and an infinite set of side frequencies. These side frequencies are located symmetrically on either side of the carrier term and separated by f_m, $2f_m$, $3f_m$, … where f_m is the frequency of the message signal $m(t)$. In order to observe this, modify your transmitter to plot the spectrum of the modulated signal using the FFT Power Spectrum and PSD VI and plot the results in your lab manual. (0.5 point)

6. The transmission bandwidth of a frequency modulated monotone message signal is defined by Carson's rule [1] as follow:

$$BW_{Tx} = 2\Delta f_{max} + 2f_m$$

Eq. 16

where f_m is the frequency of the message signal. Calculate the bandwidth of modulated signal and verify it with your software. Then plot the result of the spectrum analyzer in your report. (0.5 point)

7. Increase the **Peak Deviation** frequency to 40,000 Hz and do the step 6 again. What do you observe? (0.5 point)

8. Decrease the Peak Deviation frequency to 20 kHz and do the step 6 again. What do you observe? Describe how FM bandwidth varies with Δf_{max}. Does your observation verify the Carson's rule? (0.5 point)

9. Increase the tones to 2,3,4 and 5 while the Peak Deviation frequency is 30 kHz. Are you able to reconstruct the message at the receiver? Are you able to detect the different frequencies of demodulated message signal? Plot the results in your report. (1 point)

10. Increase the tone to 6 and 7 while Peak Deviation frequency is 30 kHz. Are you able to reconstruct the message at the receiver? Are you able to detect the different frequencies of demodulated message signal? Plot the results in your report. (0.5 point)

11. Increase the Low Cut-off Frequency of the LPF to 10kHz and repeat the step 10. Are you able to detect the different frequencies of demodulated message signal? Plot the results in your report and

describe your observations. (0.5 point)

12. Calculate the bandwidth in step 9 for 2,3,4 and 5 tones. Are the upper and lower sidebands symmetrical? Plot the results for different tones in your report and describe how FM bandwidth varies with maximum frequency component in message $m(t)$. (1 point)

Report (4 points):

The reports must be prepared in PDF or Microsoft Word format and should follow the standard of laboratory report. The report shall consist of:

1. A title page, including the name and number of the lab
2. The object, purpose and goal of the experiment shall be on the second page
3. A quick description on how the experiment is being done in 1-2 paragraphs
4. Calculations
5. Graphs
6. Pictures
7. Results
8. Conclusion
9. Tx and Rx VI files

All reports are due one week after the receiver experiment is completed and each team must submit only one report along with all the results and necessary files in a zip format file to the Moodle page.

References:

[1] S. Haykin and M. Moher, Introduction to Analog and Digital Communications, 2nd ed., John Wiley & Sons, Inc., 2007.

[2] A. V. Oppenheim, R. W. Schafer and J. R. Buck, Discrete-Time Signal Processing, 2nd ed, New Jersey: Prentice Hall Inc., 1998.