```
//bs1750.cof IIR bandstop coefficient file, centered at 1,750 Hz

#define stages 5              //number of 2nd-order stages

int a[stages][3]=       {     //numerator coefficients
{27940, -10910, 27940},       //a10, a11, a12 for 1st stage
{32768, -11841, 32768},       //a20, a21, a22 for 2nd stage
{32768, -13744, 32768},       //a30, a31, a32 for 3rd stage
{32768, -11338, 32768},       //a40, a41, a42 for 4th stage
{32768, -14239, 32768}  };

int b[stages][2]=       {     //*denominator coefficients
{-11417, 25710},              //b11, b12 for 1st stage
{-9204, 31581},               //b21, b22 for 2nd stage
{-15860, 31605},              //b31, b32 for 3rd stage
{-10221, 32581},              //b41, b42 for 4th stage
{-15258, 32584}         };    //b51, b52 for 5th stage
```

FIGURE D.4. Coefficient file for an IIR bandstop filter centered at 1750Hz, designed using MATLAB's filter designer SPTOOL (`bs1750.cof`).

D.3 MATLAB FOR FIR FILTER DESIGN USING THE STUDENT VERSION

FIR filters can be designed using the Student Version [2] of the MATLAB software package [1]. See also Section D.1 for the design of FIR filters using MATLAB's GUI filter designer SPTOOL.

Example D.3: FIR Filter Design Using MATLAB's Student Version

Figure D.5 shows a listing of a MATLAB program `mat33.m` to design a 33-coefficient FIR bandpass filter. The function `remez` uses the Parks–McClellan algorithm based on the Remez exchange algorithm and Chebyshev's approximation theory. The desired filter has a center frequency of 1kHz with a sampling frequency of 10kHz. The frequency v represents the normalized frequency variable, defined as $v = f/F_N$, where F_N is the Nyquist frequency. The bandpass filter is represented by three bands:

1. The first band (stopband) has normalized frequencies between 0 and 0.1 (0 to 500Hz), with a corresponding magnitude of 0.
2. The second band (passband) has normalized frequencies between 0.15 and 0.25 (750 to 1250Hz), with a corresponding magnitude of 1.
3. The third band (stopband) has normalized frequencies between 0.3 and the Nyquist frequency of 1 (1500 to 5000Hz), with a corresponding magnitude of 0.

```
%Mat33.m MATLAB program for FIR Bandpass with 33 coefficients Fs=10 kHz

nu= [0 0.1 0.15 0.25 0.3 1];    %normalized frequencies
mag= [0 0 1 1 0 0];             %magnitude at normalized frequencies
c=remez (32,nu,mag);            %invoke remez algorithm for 33 coeff
bp33=c';                        % coeff values transposed
save matpb33.cof bp33 -ascii;   %save in ASCII file with coefficients
[h,w] =freqz (c,1,256);         %frequency response with 256 points
plot(5000*nu,mag,w/pi,abs(h))   %plot ideal magnitude response
```

FIGURE D.5. MATLAB program for FIR filter design (`mat33.m`).

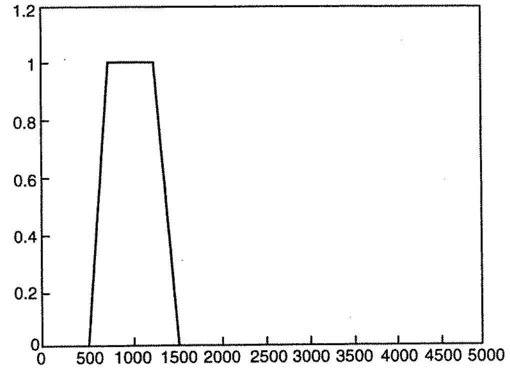

FIGURE D.6. Frequency response of the FIR bandpass filter desired, obtained with MATLAB.

Run this program from MATLAB and verify the magnitude response of the ideal desired filter plotted within MATLAB in Figure D.6. Note that the frequencies 750 and 1250Hz represent passband frequencies with normalized frequencies of 0.15 and 0.25, respectively, and associated magnitudes of 1. The frequencies 500 and 1500Hz represent stopband frequencies with normalized frequencies of 0.1 and 0.3, respectively, and associated magnitudes of 0. The last normalized frequency value of 1 corresponds to the Nyquist frequency of 5000Hz and has a magnitude of zero. The program generates a set of 33 coefficients saved in the coefficient file `matbp33 .cof` in ASCII format.

Example D.4: Multiband FIR Filter Design Using MATLAB

This example extends the preceding three-band example to a five-band design in order to obtain two passbands. The program `mat63.m` (Figure D.7) is similar to the preceding MATLAB program, `mat33.m`. This filter with two passbands is

```
%Mat63.m  MATLAB program for two passbands, 63 coefficients Fs=10 kHz

nu= [0 0.1 0.12 0.18 0.2 0.3 0.32 0.38 0.4 1]; %normalized frequencies
mag= [0 0 1 1 0 0 1 1 0 0];    %magnitude at normalized frequencies
c=remez (62,nu,mg);            %invoke remez algorithm for 63 coeff
bp63=c';                       % coeff values transposed
save mat2bp.cof bp63 -ascii;   %save in ASCII file with coefficients
[h,w] =freqz (c,1,256);        %frequency response with 256 points
plot (500*nu,mag,w/pi,abs(h))  %plot ideal magnitude response
```

FIGURE D.7. MATLAB program for a two-passband FIR filter design (`mat63.m`).

represented by a total of five bands: the first band (stopband) has normalized frequencies between 0 and 0.1 (0 to 500 Hz), with corresponding magnitude of 0; the second band (passband) has normalized frequencies between 0.12 and 0.18 (600 to 900 Hz), with a corresponding magnitude of 1, and so on. This is summarized as follows:

Band	Frequency (Hz)	Normalized f/F_N	Magnitude
1	0–500	0–0.1	0
2	600–900	0.12–0.18	1
3	1000–1500	0.2–0.3	0
4	1600–1900	0.32–0.38	1
5	2000–5000	0.4–1	0

Run this program from MATLAB and verify the magnitude response of the ideal two-passband filter in Figure D.8. This program generates a set of 63 coefficients saved into the coefficient file `mat2bp.cof` in ASCII format.

D.4 MATLAB FOR IIR FILTER DESIGN USING THE STUDENT VERSION

MATLAB can also be used for the design of IIR filters using the Student Edition of MATLAB. See also Section D.2 for the design of IIR filters using MATLAB's GUI filter designer SPTOOL.

Example D.5: IIR Filter Design Using MATLAB's Student Version

The function *`yulewalk`*, available in MATLAB, allows for the design of recursive filters based on a best least squares fit [1,2]. Consider again the MATLAB program `mat33.m` in Figure D.5 to obtain a 33-coefficient FIR bandpass filter centered at 1000 Hz. In lieu of the *`remez`* function for an FIR design, the MATLAB command

```
>>[a,b] = yulewalk(n,nu,mag)
```

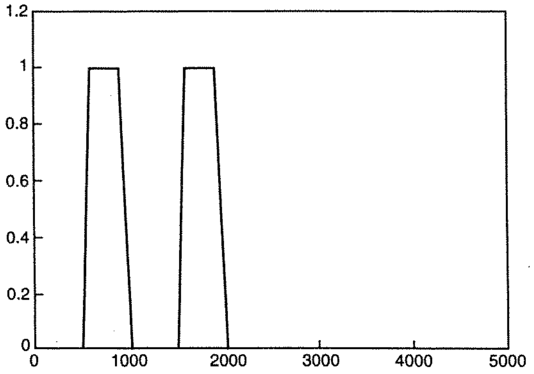

FIGURE D.8. Frequency response of a two-passband FIR filter using MATLAB.

returns the *a* and *b* coefficients in the general input–output equation in Chapter 5, associated with an IIR filter. The filter's order n represents the number of second-order sections. The C program in Example 5.1 implements an IIR filter with cascaded second-order sections, as is most commonly done. For example, if n = 6 in the *`yulewalk`* function, the general transfer function in Chapter 5 in terms of the resulting *a* and *b* coefficients from MATLAB needs to be reduced to one in terms of three cascaded sections.

D.5 BILINEAR TRANSFORMATION USING MATLAB AND SUPPORT PROGRAMS ON DISK

This section expands on the bilinear transformation discussion in Section 5.3.

Exercise D.1: First-Order IIR Lowpass Filter

Given a first-order lowpass analog transfer function $H(s)$, a corresponding discrete-time filter with transfer function $H(z)$ can be obtained. Let the bandwidth or cutoff frequency $B = 1$ *r/s* and the sampling frequency $F_s = 10$ Hz.

1. Choose an appropriate transfer function

$$H(s) = \frac{1}{s+1}$$

which represents a lowpass filter with a bandwidth of 1 *r/s*.