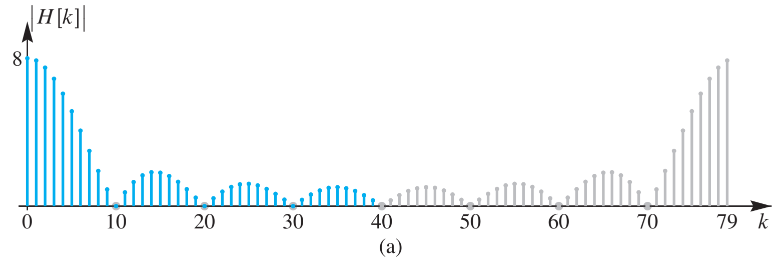

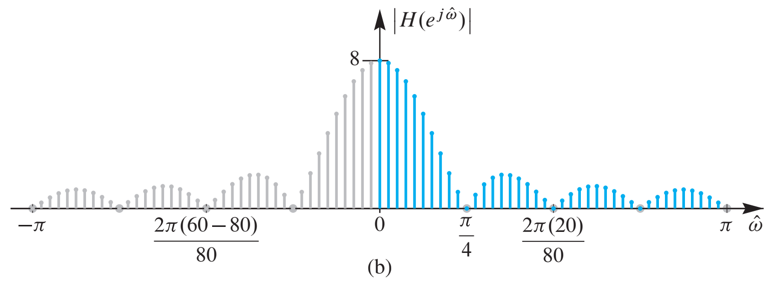

Figure 8-1: Frequency response of 8-point running sum filter. (a) Magnitude response obtained from length-80 FFT plotted versus DFT index (k). (b) Magnitude response plotted versus $\hat{\omega}$ after moving the negative frequency components shown in orange. The indices $0 \le k < 40$ map to the frequencies $\hat{\omega} = 2\pi k/N$; while the indices $40 \le k < 80$ map to the negative frequencies $\hat{\omega} = 2\pi(k - N)/N = 2\pi k/N - 2\pi$.

Example 8-4: Suppose that we want to plot the frequency response of an 8-point running sum FIR filter. Although MATLAB can do the job with its frequency response function called `freqz`, we can also do the evaluation directly with the DFT by calling `fft` in MATLAB. This provides insight into the internal structure of `freqz` which calls the FFT to do the actual work. To evaluate the frequency response at many closely spaced frequencies we need to use zero padding, so we pick $N = 80$ and do the following in MATLAB:

```
N = 80;
hpadded = [ ones(1,8), zeros(1,N-8) ];
Hk = fft(hpadded);
stem( 0:N-1, abs(Hk) )   % Plot H[k] vs. k
```

Zero-padding is a very common operation, so the `fft` function has a second argument that sets the FFT length and enables zero-padding. Thus, `fft(hn,N)` computes an N-point DFT of `hn`, with zero-padding if `N>length(hn)`. Figure 8-1(a) shows the resulting plot. If we compare the plot from this code to the magnitude plot of other running sum filters in Chapter **??**, we see that the result is similar, but there is a big issue: the axis labels must be converted from "frequency index" k to $\hat{\omega}$. In addition, the frequency axis needs to have $\hat{\omega}$ running from $-\pi$ to π, so the negative frequency components must be moved as explained in Section **??**. This is accomplished by treating the second half of the `Hk` vector differently from the first half as in the following code (which assumes `N` is even):

```
HkCentered = [ Hk(N/2+1:N), Hk(1:N/2) ]; %- fftshift.m does this
kCentered =  [-N+(N/2+1:N),   (1:N/2) ] - 1;
    %- make negative frequency indices
    % kCentered is [-N/2,-N/2+1,...,-2,-1,0,1,2,3,...,N/2-1]
stem( (2*pi/N)*kCentered, abs(HkCentered) )
```

The flipping operation above occurs so often when using the FFT in MATLAB that a special function called `fftshift` has been created to perform the reordering. The frequencies have to be reordered separately. When the frequency response has conjugate symmetry, it is customary to plot only the first half of `Hk` which corresponds to the positive frequency region.