6. **Sample summation**: This operation differs from signal addition operation. It adds all sample values of $x(n)$ between n_1 and n_2.

$$\sum_{n=n_1}^{n_2} x(n) = x(n_1) + \cdots + x(n_2)$$

It is implemented by the `sum(x(n1:n2))` function.

7. **Sample products**: This operation also differs from signal multiplication operation. It multiplies all sample values of $x(n)$ between n_1 and n_2.

$$\prod_{n_1}^{n_2} x(n) = x(n_1) \times \cdots \times x(n_2)$$

It is implemented by the `prod(x(n1:n2))` function.

8. **Signal energy**: The energy of a sequence $x(n)$ is given by

$$\mathcal{E}_x = \sum_{-\infty}^{\infty} x(n)x^*(n) = \sum_{-\infty}^{\infty} |x(n)|^2$$

where superscript * denotes the operation of complex conjugation.[1] The energy of a finite-duration sequence $x(n)$ can be computed in MATLAB using

```
>> Ex = sum(x .* conj(x)); % one approach
>> Ex = sum(abs(x) .^ 2); % another approach
```

9. **Signal power**: The average power of a periodic sequence $\tilde{x}(n)$ with fundamental period N is given by

$$\mathcal{P}_x = \frac{1}{N} \sum_{0}^{N-1} |\tilde{x}(n)|^2$$

□ **EXAMPLE 2.1** Generate and plot each of the following sequences over the indicated interval.

a. $x(n) = 2\delta(n+2) - \delta(n-4), \quad -5 \le n \le 5.$
b. $x(n) = n[u(n)-u(n-10)] + 10e^{-0.3(n-10)}[u(n-10)-u(n-20)],\ 0 \le n \le 20.$
c. $x(n) = \cos(0.04\pi n) + 0.2w(n), \quad 0 \le n \le 50$, where $w(n)$ is a Gaussian random sequence with zero mean and unit variance.
d. $\tilde{x}(n) = \{..., 5, 4, 3, 2, 1, \underset{\uparrow}{5}, 4, 3, 2, 1, 5, 4, 3, 2, 1, ...\}; \ -10 \le n \le 9.$

[1]The symbol * denotes many operations in digital signal processing. Its font (roman or computer) and its position (normal or superscript) will distinguish each operation.

Solution

a. $x(n) = 2\delta(n+2) - \delta(n-4), \quad -5 \le n \le 5.$

```
>> n = [-5:5];
>> x = 2*impseq(-2,-5,5) - impseq(4,-5,5);
>> stem(n,x); title('Sequence in Problem 2.1a')
>> xlabel('n'); ylabel('x(n)');
```

The plot of the sequence is shown in Figure 2.1a.

b. $x(n) = n\,[u(n) - u(n-10)] + 10e^{-0.3(n-10)}\,[u(n-10) - u(n-20)],\ 0 \le n \le 20.$

```
>> n = [0:20]; x1 = n.*(stepseq(0,0,20)-stepseq(10,0,20));
>> x2 = 10*exp(-0.3*(n-10)).*(stepseq(10,0,20)-stepseq(20,0,20));
>> x = x1+x2;
>> subplot(2,2,3); stem(n,x); title('Sequence in Problem 2.1b')
>> xlabel('n'); ylabel('x(n)');
```

The plot of the sequence is shown in Figure 2.1b.

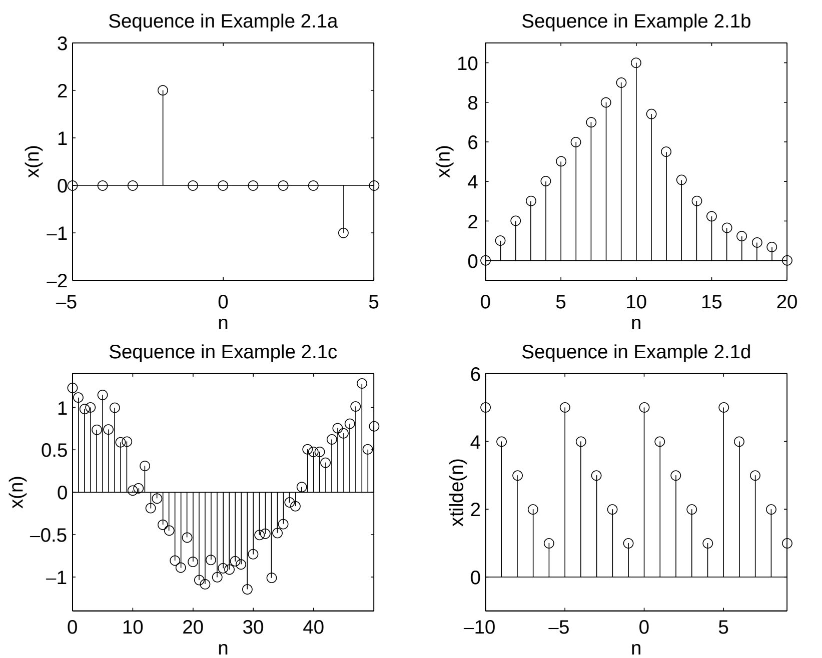

FIGURE 2.1 *Sequences in Example 2.1*

c. $x(n) = \cos(0.04\pi n) + 0.2w(n), \quad 0 \le n \le 50.$

```
>> n = [0:50]; x = cos(0.04*pi*n)+0.2*randn(size(n));
>> subplot(2,2,2); stem(n,x); title('Sequence in Problem 2.1c')
>> xlabel('n'); ylabel('x(n)');
```

The plot of the sequence is shown in Figure 2.1c.

d. $\tilde{x}(n) = \{..., 5, 4, 3, 2, 1, \underset{\uparrow}{5}, 4, 3, 2, 1, 5, 4, 3, 2, 1, ...\}; \ -10 \le n \le 9.$

Note that over the given interval, the sequence $\tilde{x}(n)$ has four periods.

```
>> n = [-10:9]; x = [5,4,3,2,1];
>> xtilde = x' * ones(1,4);   xtilde = (xtilde(:))';
>> subplot(2,2,4); stem(n,xtilde); title('Sequence in Problem 2.1d')
>> xlabel('n'); ylabel('xtilde(n)');
```

The plot of the sequence is shown in Figure 2.1d. □

□ **EXAMPLE 2.2** Let $x(n) = \{1, 2, \underset{\uparrow}{3}, 4, 5, 6, 7, 6, 5, 4, 3, 2, 1\}$. Determine and plot the following sequences.

a. $x_1(n) = 2x(n-5) - 3x(n+4)$
b. $x_2(n) = x(3-n) + x(n)\,x(n-2)$

Solution The sequence $x(n)$ is nonzero over $-2 \le n \le 10$. Hence

```
>> n = -2:10; x = [1:7,6:-1:1];
```

will generate $x(n)$.

a. $x_1(n) = 2x(n-5) - 3x(n+4)$.
The first part is obtained by shifting $x(n)$ by 5 and the second part by shifting $x(n)$ by -4. This shifting and the addition can be easily done using the `sigshift` and the `sigadd` functions.

```
>> [x11,n11] = sigshift(x,n,5); [x12,n12] = sigshift(x,n,-4);
>> [x1,n1] = sigadd(2*x11,n11,-3*x12,n12);
>> subplot(2,1,1); stem(n1,x1); title('Sequence in Example 2.2a')
>> xlabel('n'); ylabel('x1(n)');
```

The plot of $x_1(n)$ is shown in Figure 2.2a.

b. $x_2(n) = x(3-n) + x(n)\,x(n-2)$.
The first term can be written as $x(-(n-3))$. Hence it is obtained by first folding $x(n)$ and then shifting the result by 3. The second part is a multiplication of $x(n)$ and $x(n-2)$, both of which have the same length but different

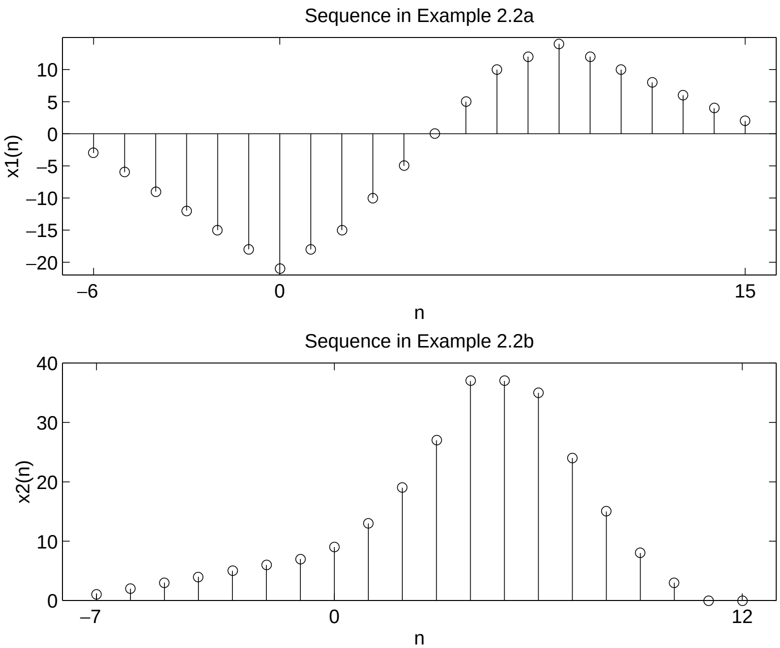

FIGURE 2.2 *Sequences in Example 2.2*

support (or sample positions). These operations can be easily done using the `sigfold` and the `sigmult` functions.

```
>> [x21,n21] = sigfold(x,n); [x21,n21] = sigshift(x21,n21,3);
>> [x22,n22] = sigshift(x,n,2); [x22,n22] = sigmult(x,n,x22,n22);
>> [x2,n2] = sigadd(x21,n21,x22,n22);
>> subplot(2,1,2); stem(n2,x2); title('Sequence in Example 2.2b')
>> xlabel('n'); ylabel('x2(n)');
```

The plot of $x_2(n)$ is shown in Figure 2.2b. □

Example 2.2 shows that the four `sig*` functions developed in this section provide a convenient approach for sequence manipulations.

□ **EXAMPLE 2.3** Generate the complex-valued signal

$$x(n) = e^{(-0.1+j0.3)n}, \quad -10 \le n \le 10$$

and plot its magnitude, phase, the real part, and the imaginary part in four separate subplots.

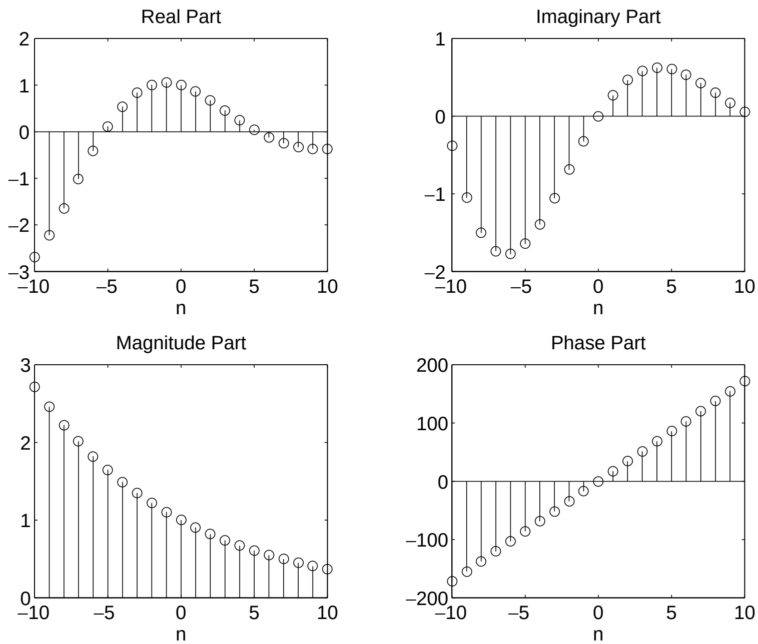

FIGURE 2.3 *Complex-valued sequence plots in Example 2.3*

Solution MATLAB script:

```
>> n = [-10:1:10]; alpha = -0.1+0.3j;
>> x = exp(alpha*n);
>> subplot(2,2,1); stem(n,real(x));title('real part');xlabel('n')
>> subplot(2,2,2); stem(n,imag(x));title('imaginary part');xlabel('n')
>> subplot(2,2,3); stem(n,abs(x));title('magnitude part');xlabel('n')
>> subplot(2,2,4); stem(n,(180/pi)*angle(x));title('phase part');xlabel('n')
```

The plot of the sequence is shown in Figure 2.3. □

2.1.3 DISCRETE-TIME SINUSOIDS

In the last section we introduced the discrete-time sinusoidal sequence $x(n) = A\cos(\omega_0 n + \theta_0)$, for all n as one of the basic signals. This signal is very important in signal theory as a basis for Fourier transform and in system theory as a basis for steady-state analysis. It can be conveniently related to the continuous-time sinusoid $x_a(t) = A\cos(\Omega_0 t + \theta_0)$ using an operation called *sampling* (Chapter 3), in which continuous-time sinusoidal values at equally spaced points $t = nT_s$ are assigned to $x(n)$.