Chapter 2: Time-Domain Models of Systems

2.1 $y[n] = \sum_{i=0}^{N-1} ab^i x[n-i]$, where $a = \dfrac{1-b}{1-b^N}$

For $n \geq N$,

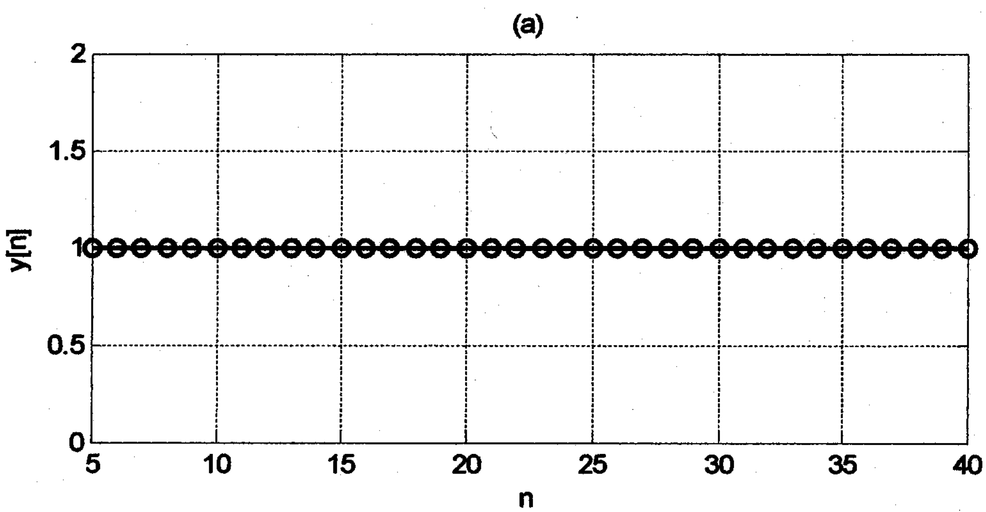
$$y[n] = ac + abc + ab^2c + \cdots + ab^{N-1}c$$

$$y[n] = ac\left[1 + b + b^2 + \cdots b^{N-1}\right]$$

$$y[n] = ac\left[\frac{1-b^N}{1-b}\right] = \left[\frac{1-b}{1-b^N}\right]c\left[\frac{1-b^N}{1-b}\right] = c$$

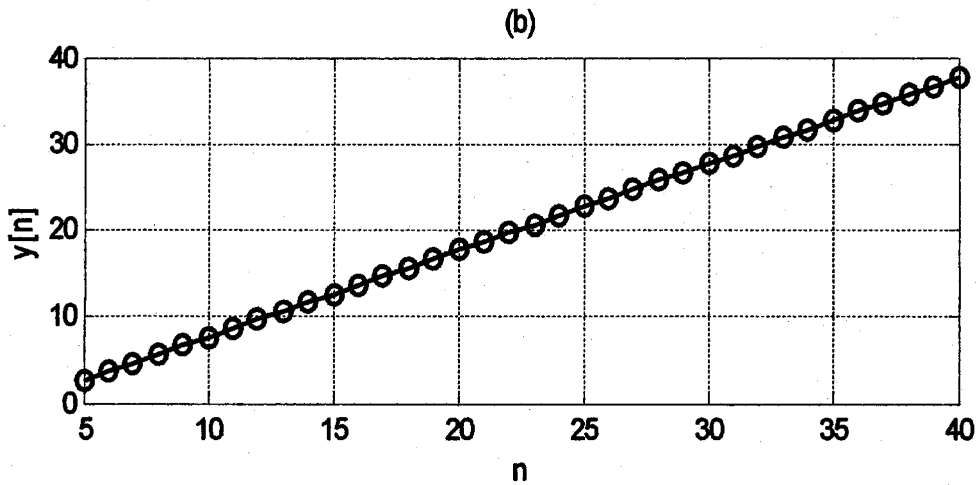
2.2 (a)

```
x=ones(1,41);
b=0.7;
a=(1-b)/(1-b^5);
i=1:5;
w=a*(b.^(5-i));
for n=5:40;
y(n)=w*(x(n-4:n)');
end;
n=5:40;
plot(n,y(n),n,y(n),'o','LineWidth',1.5)
grid
xlabel('n')
ylabel('y[n]')
title('(a)')
```



2.2 (b)

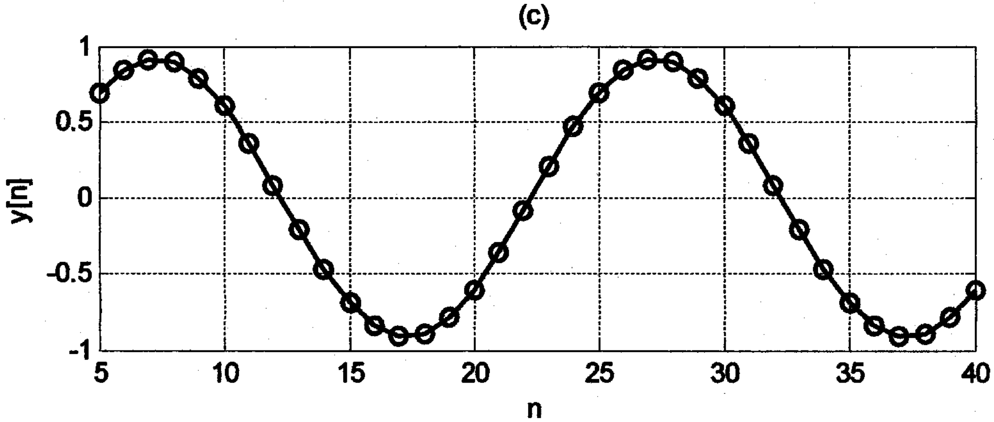
```
n=0:40;
x=n;
b=0.7;
a=(1-b)/(1-b^5);
i=1:5;
w=a*(b.^(5-i));
for n=5:40;
y(n)=w*(x(n-4:n)');
end;
n=5:40;
plot(n,y(n),n,y(n),'o','LineWidth',1.5)
grid
xlabel('n')
ylabel('y[n]')
title('(b)')
```



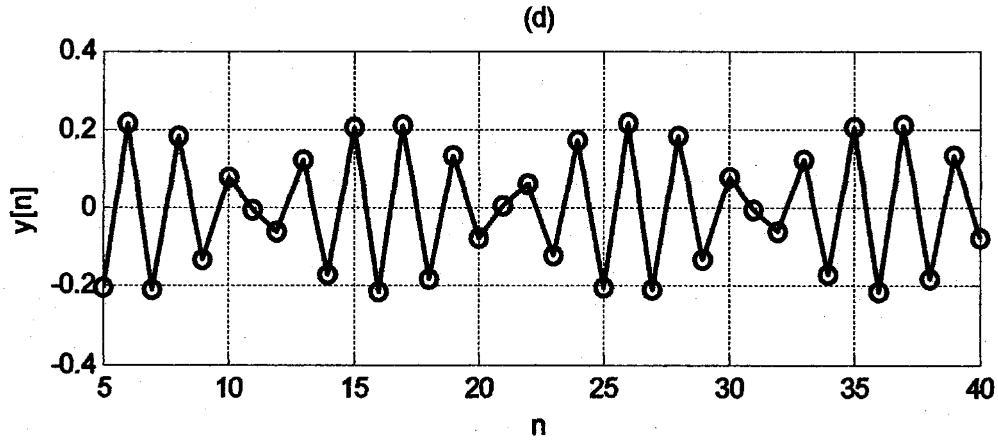
(c)

```
n=0:40;
x=sin(pi*n/10);
b=0.7;
a=(1-b)/(1-b^5);
i=1:5;
w=a*(b.^(5-i));
for n=5:40;
y(n)=w*(x(n-4:n)');
end;
n=5:40;
plot(n,y(n),n,y(n),'o','LineWidth',1.5)
grid
xlabel('n')
ylabel('y[n]')
title('(c)')
```

2.2 (c) continued



(d) Run the MATLAB program in Part (c) except replace `x` by `x=sin(.9*pi*n)`.

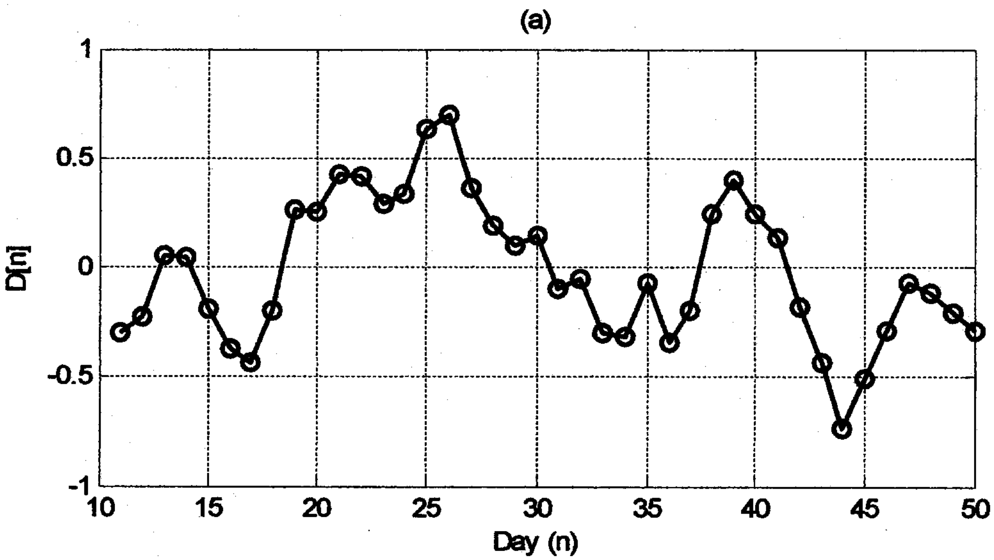


(e) The signal in Part (d) has a much higher frequency than that of the sinusoidal signal in Part (c). Since the 5-point EWMA flter is a "lowpass filter," it attenuates the high- frequency signal in Part (d), while it passes the low-frequency signal in Part (c) with very little attenuation.

2.3 By l'Hospital's Rule, $\lim_{b\to 1}\frac{1-b}{1-b^N}=\lim_{b\to 1}\frac{-1}{-Nb^{N-1}}=\frac{1}{N}$. Also $\lim_{b\to 1} b^i = 1$ for any integer i, and thus

$\lim_{b\to 1} ab^i = \lim_{b\to 1} a = \lim_{b\to 1}\frac{1-b}{1-b^N}$ for any i. Thus, $\lim_{b\to 1} ab^i = \frac{1}{N}$ and $y[n]=\sum_{i=0}^{N-1}\frac{1}{N}x[n-i]$, which is the input/output representation of the N-point MA filter.

2.4 (a)
```
c=csvread('QQQQdata2.csv',1,4,[1 4 50 4]);
b1=0.3;
a1=(1-b1)/(1-b1^11);
b2=0.7;
a2=(1-b2)/(1-b2^11);
i=1:11;
w1=a1*(b1.^(11-i));
w2=a2*(b2.^(11-i));
for n=11:50;
y1(n)=w1*c(n-10:n);
y2(n)=w2*c(n-10:n);
D(n)=y1(n)-y2(n);
end;
n=11:50;
plot(n,D(n),n,D(n),'o','LineWidth',1.5)
grid
xlabel('Day (n)')
ylabel('D[n]')
title('(a)')
```



(b) There is a buy on day 13, a sell on day 15, a buy on day 19, a sell on day 31, a buy on day 38, and a sell on day 42.

(c) c[15]-c[13] = 34.75 – 35.50 = loss of - 0.75
c[31] – c[19] = 36.63 – 35.34 = gain of 1.29
c[42] – c[38] = 36.20 – 36.92 = loss of -0.72

(d) There is a net loss of – 0.18. There are various modifications of this trading scheme that often do much better. One such modification is to wait until $D[n]$ is negative and then buy on day n when $D[n] > \varepsilon$ and then sell on day n when $D[n] < \varepsilon$, where ε is a positive threshold value. The stock is bought again only after $D[n]$ has gone negative. For example, if we take $\varepsilon = 0.15$, there is a buy on day 19, a sell on day 29, a buy on day 38, and a sell on day 41. This gives c[29] – c[19] = 1.60, c[41] – c[38] = -0.05, so there is a net gain of 1.55.

2.5 **(a)** $h[n+1]+h[n]=2\delta[n]$, and thus $h[n+1] = -h[n]+2\delta[n]$.

Setting $n = -1$ gives $h[0] = -h[-1] + 2\delta[-1] = 0 + 0 = 0$
Setting $n = 0$ gives $h[1] = -h[0] + 2\delta[0] = 0 + 2 = 2$
Setting $n = 1$ gives $h[2] = -h[1] + 2\delta[1] = -2 + 0 = -2$
Setting $n = 2$ gives $h[3] = -h[2] + 2\delta[2] = 2 + 0 = 2$

(b) $h[n+1]+(1/2)h[n]=\delta[n]$, and thus $h[n+1] = -(1/2)h[n]+\delta[n]$.

Setting $n = -1$ gives $h[0] = -(1/2)h[-1] + \delta[-1] = 0 + 0 = 0$
Setting $n = 0$ gives $h[1] = -(1/2)h[0] + \delta[0] = 0 + 1 = 1$
Setting $n = 1$ gives $h[2] = -(1/2)h[1] + \delta[1] = -(1/2) + 0 = -1/2$
Setting $n = 2$ gives $h[3] = -(1/2)h[2] + \delta[2] = -(1/2)(-1/2) + 0 = 1/4$

(c) $h[n+2]+1.5h[n+1] + 0.5h[n] =\delta[n]$, and thus $h[n+2] = -1.5h[n+1] - 0.5h[n] + \delta[n]$,

Setting $n = -2$ gives $h[0] = -1.5h[-1] -0.5h[-2] + \delta[-2] = 0 + 0 + 0 = 0$
Setting $n = -1$ gives $h[1] = -1.5h[0] -0.5h[-1] + \delta[-1] = 0 + 0 + 0 = 0$
Setting $n = 0$ gives $h[2] = -1.5\text{h}[1] -0.5h[0] + \delta[0] = 0 + 0 + 1 = 1$
Setting $n = 1$ gives $h[3] = -1.5h[2] -0.5h[1] + \delta[1] = -1.5(1) + 0 + 0 = -1.5$

(d) $h[n+2]+ (1/2)h[n+1] + (1/4)h[n] = \delta[n+1] - \delta[n]$, and thus $h[n+2] = -(1/2)h[n+1] - (1/4)h[n] + \delta[n+1] - \delta[n]$,

Setting $n = -2$ gives $h[0] = -(1/2)h[-1] - (1/4)h[-2] + \delta[-1] - \delta[-2] = 0 + 0 + 0 - 0 = 0$
Setting $n = -1$ gives $h[1] = -(1/2)h[0] - (1/4)h[-1] + \delta[0] - \delta[-1] = 0 + 0 + 1 - 0 = 1$
Setting $n = 0$ gives $h[2] = -(1/2)\text{h}[1] - (1/4)h[0] + \delta[1] - \delta[0] = -(1/2)(1) + 0 + 0 -1= -3/2$
Setting $n = 1$ gives $h[3] = -(1/2)h[2] - (1/4)h[1] + \delta[2] - \delta[1] = -(1/2)(-3/2) - (1/4)(1) + 0 - 0 = 1/2$

(e) $h[n+2]+ (1/4)h[n+1] - (3/8)h[n] = 2\delta[n+2] - 3\delta[n]$, and thus $h[n+2] = -(1/4)h[n+1] + (3/8)h[n] + 2\delta[n+2] - 3\delta[n]$,

Setting $n = -2$ gives $h[0] = -(1/4)h[-1] +(3/8)h[-2] + 2\delta[0] - 3\delta[-2] = 0 + 0 + 2 - 0 = 2$
Setting $n = -1$ gives $h[1] = -(1/4)h[0] +(3/8)h[-1] + 2\delta[1] - 3\delta[-1] = -(1/4)(2) + 0 + 0 + 0 - 0 = -1/2$
Setting $n = 0$ gives $h[2] = -(1/4)\text{h}[1] +(3/8)h[0] + 2\delta[2] - 3\delta[0] = -(1/4)(-1/2) + (3/8)(2) + 0 -3= -17/8$
Setting $n = 1$ gives $h[3] = -(1/4)h[2] + (3/8)h[1] +2\delta[3] - 3\delta[1] = -(1/4)(-17/8) + (3/8)(-1/2) + 0 - 0 = 11/32$

2.6 **(a)** Replacing n by n-1 gives $y[n] + y[n-1] =x[n-1]$, and thus $h[n] = -h[n-1] + \delta[n-1]$. Then $h[0] = 0 + 0 = 0$, $h[1] = 0 + 1 = 1$, $h[2] = -1 + 0 = -1$, $h[3] = -(-1) + 0 = 1$, $h[4] = -1 + 0 = -1$,... The pattern shows that $h[0] = 0$ and $h[n] = (-1)^{n-1}$ for $n \geq 1$. Hence, $h[n] = (-1)^{n-1}u[n-1]$.

(b) $h[n] = -(1/2)h[n-1] + \delta[n]$. Then, $h[0] = 0 + 1 = 1$, $h[1] = -(1/2)(1) + 0 = -1/2$, $h[2] = -(1/2)(-1/2) + 0 = 1/4$, $h[3] = -(1/2)(1/4) + 0 = -1/8$, $h[4] = -(1/2)(-1/8) +) = 1/16$, ...The pattern shows that $h[n] = (-1/2)^n u[n]$.

(c) Since the system is linear, the impulse response $h[n]$ is equal to the sum $h_1[n] + h_2[n]$ of the impulse responses to the systems defined by $y_1[n+1]+2y_1[n]=2x[n+1]$ and $y_2[n+1]+2y_2[n]= -2x[n]$. Now $h_1[n] = -2h_1[n-1] + 2\delta[n]$, and thus $h_1[0] = 0 + 2 = 2$, $h_1[1] = -(2)(2) + 0 = -4$, $h_1[2] = -(2)(-4) = 8$, ... The pattern shows that $h_1[n] = -(-2)^{n+1}u[n]$. Using this result and linearity and time invariance of the system, $h_2[n] = = (-2)^n u[n-1]$.

$$h[0] = h_1[0] + h_2[0] = 2 + 0 = 2$$

$$h[n] = h_1[n] + h_2[n] = -(-2)^{n+1} + (-2)^n = 2(-2)^n + (-2)^n = 3\left(-2\right)^n,\ n \geq 1$$

(d) Using the same method as given in Part (c), we have that

$$h[0] = 1$$
$$h[n] = 2\left(\frac{1}{2}\right)^n, \ n \geq 1$$

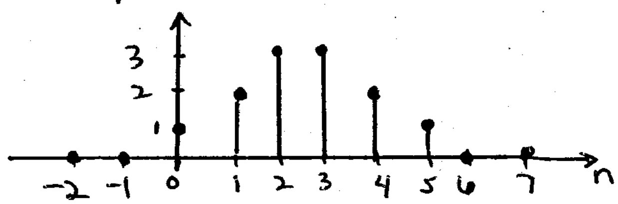
(e) $h[n] = -(1/2)h[n-2] + 2\delta[n] - \delta[n-2]$. Then $h[0] = 0 + 2 - 0 = 2$, $h[1] = 0 + 0 - 0 = 0$, $h[2] = -(1/2)(2) + 0 - 1 = -2$, $h[3] = -(1/2)(0) + 0 + 0 = 0$, $h[4] = -(1/2)(-2) + 0 + 0 = 1$, $h[5] = -(1/2)(0) + 0 + 0 = 0$, $h[6] = -(1/2)(1) + 0 + 0 = -1/2$, $h[7] = -(1/2)(0) + 0 + 0 = 0$, $h[8] = -(1/2)(-1/2) + 0 + 0 = ¼$.,.....Pattern shows that

$$h[0] = 2$$
$$h[n] = 0, \ n = 1,3,5,7,\ldots$$
$$h[n] = 4\left(-\frac{1}{2}\right)^{n/2}, \ n = 2,4,6,\ldots$$

2.7 (a) $x[n] * v[n] = \sum_{i=-\infty}^{\infty} x[i]v[n-i]$

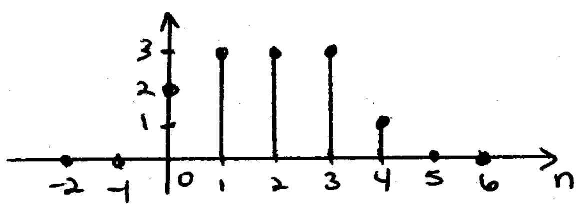
$$= \sum_{i=0}^{2} v[n-i] = v[n] + v[n-1] + v[n-2]$$

n	0	1	2	3	4	5	6	$n>6, n<0$
$x[n]*v[n]$	1	2	3	3	2	1	0	0



(b) $x[n] * v[n] = 2v[n] + v[n-1]$

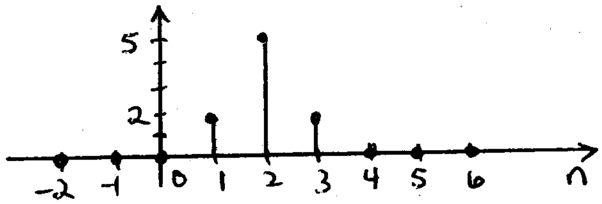
n	0	1	2	3	4	5	$n>5$
$x[n]*v[n]$	2	3	3	3	1	0	0



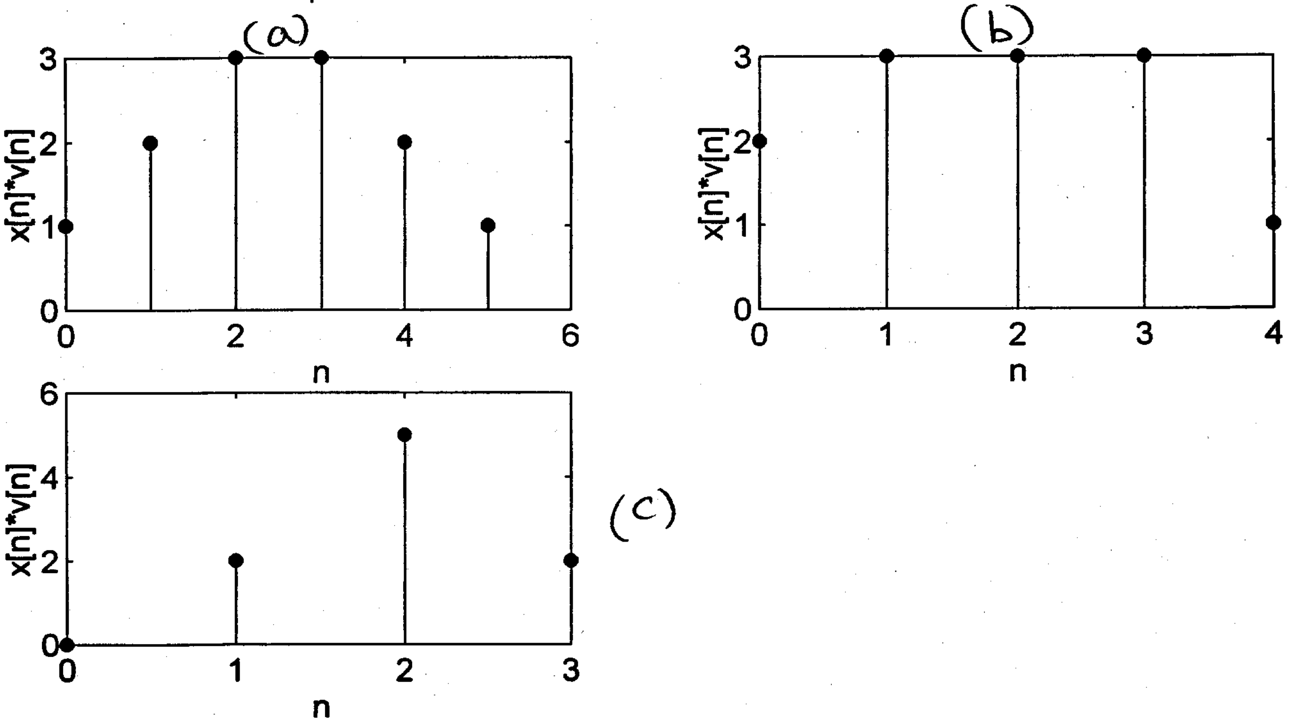
2.7 continued

(c) $x[n] * v[n] = 2v[n] + v[n-1]$

n	0	1	2	3	4	$n >$
$x[n]*v[n]$	0	2	5	2	0	0



MATLAB plots obtained from conv:



2.8 (a) Using the array method in Section 2.2:

$x[0] \searrow$

	4	1	-1	0	0
$v[0] \rightarrow$ 1	4	1	-1	0	0
-2	-8	-2	2	0	0
3	12	3	-3	0	
-4	-16	-4	4		
0	0	0	0		
0	0	0	0		

2.8 (a) continued Let $y[n] = x[n]*v[n]$. Then $y[0] = 4$, $y[1] = 1-8 = -7$, $y[2] = -1-2+12 = 9$, $y[3] = 2+3-16 = -11$, $y[4] = -3-4 = -7$, $y[5] = 4$, $y[n] = 0$, $n \geq 6$.

(b)

	x[-1] 1	1	1	1	0	0	0
v[-1] -1	-1	-1	-1	-1	0	0	0
0	0	0	0	0	0	0	0
8	8	8	8	8	0	0	0
2	2	2	2	2	0	0	
3	3	3	3	3	0		
0	0	0	0	0			
0	0	0	0				
	0	0					

Again let $y[n] = x(n)*v(n)$. Then $y[-2] = -1$, $y[-1] = -1$, $y[0] = -1+8 = 7$, $y[1] = -1+8+2 = 9$, $y[2] = 8+2+3 = 13$, $y[3] = 8+2+3 = 13$, $y(4) = 2+3 = 5$, $y[5] = 3$, $y[n] = 0$, $n \geq 6$

(c) $x[n]*v[n] = \sum_{i=-\infty}^{\infty} v[i]x[n-i]$

$$= v[0]x[n] + v[1]x[n-1] + v[2]x[n-2] + v[3]x[n-3]$$

$$= 2(2)^n - 3(2^{n-1}) + 6(2^{n-3}) \quad \text{for } n \leq 3$$

$$x[n]*v[n] = \tfrac{5}{4}(2^n), \text{ for } n \leq 3$$

$$x[n]*v[n] = -3(2^3) + 6(2^0) = -12 \quad \text{for } n=4$$

$$x[n]*v[n] = 6(2^2) = 24 \quad \text{for } n=5$$

$$x[n]*v[n] = 6(2^3) = 48 \quad \text{for } n=6$$

$$x[n]*v[n] = 0, \quad \text{all } n>6$$

2.8 continued

(d) $x[n] * v[n] = -2\,x[n-2] - 5\,x[n-3]$

where $x[n] = \frac{1}{n}\left(u[n-2] - u[n-6]\right)$

Then

$$x[n] * v[n] = \begin{cases} 0 & n \le 3 \\ -1 & n = 4 \\ \frac{-2}{n-2} - \frac{5}{n-3} & n = 5, 6, 7 \\ -1 & n = 8 \\ 0 & n \ge 9 \end{cases}$$

(e) $x[n] * v[n] = 0$ for $n < 0$

$$x[n] * v[n] = \sum_{i=0}^{n} x[i]\,v[n-i] = \sum_{i=0}^{n} (1) = n+1$$

(f) For $n \ge 1$, $x[n] * v[n] = \sum_{i=1}^{n} \ln(i) = \ln(n!)$

For $n \le 0$ $x[n] * v[n] = 0$

(g) $x[n] * v[n] = \cos\left(\frac{\pi n}{3}\right) u[n] - \cos\left(\frac{\pi(n-2)}{3}\right) u[n-2]$

2.9 (a) $v[n] * x[n] = \sum_{i=0}^{n} 2^i$ for $n \ge 0$

$$= \frac{1 - 2^{n+1}}{1-2} = 2^{n+1} - 1 \quad \text{for } n \ge 0$$

$v[n] * x[n] = 0$ for $n < 0$

(b) $v[n] * x[n] = \sum_{i=0}^{n} (0.25)^i$ for $n \ge 0$

$$= \frac{1 - (0.25)^{n+1}}{1 - 0.25} = \frac{4}{3} - \frac{4}{3}(0.25)^{n+1} = \frac{1}{3}\left(4 - (0.25)^n\right), n \ge 0$$

$v[n] * x[n] = 0$ for $n < 0$

2.9 continued

(c) $v[n] * x[n] = \sum_{i=0}^{n} 2^{i} (0.5)^{n-i}$ for $n \geq 0$

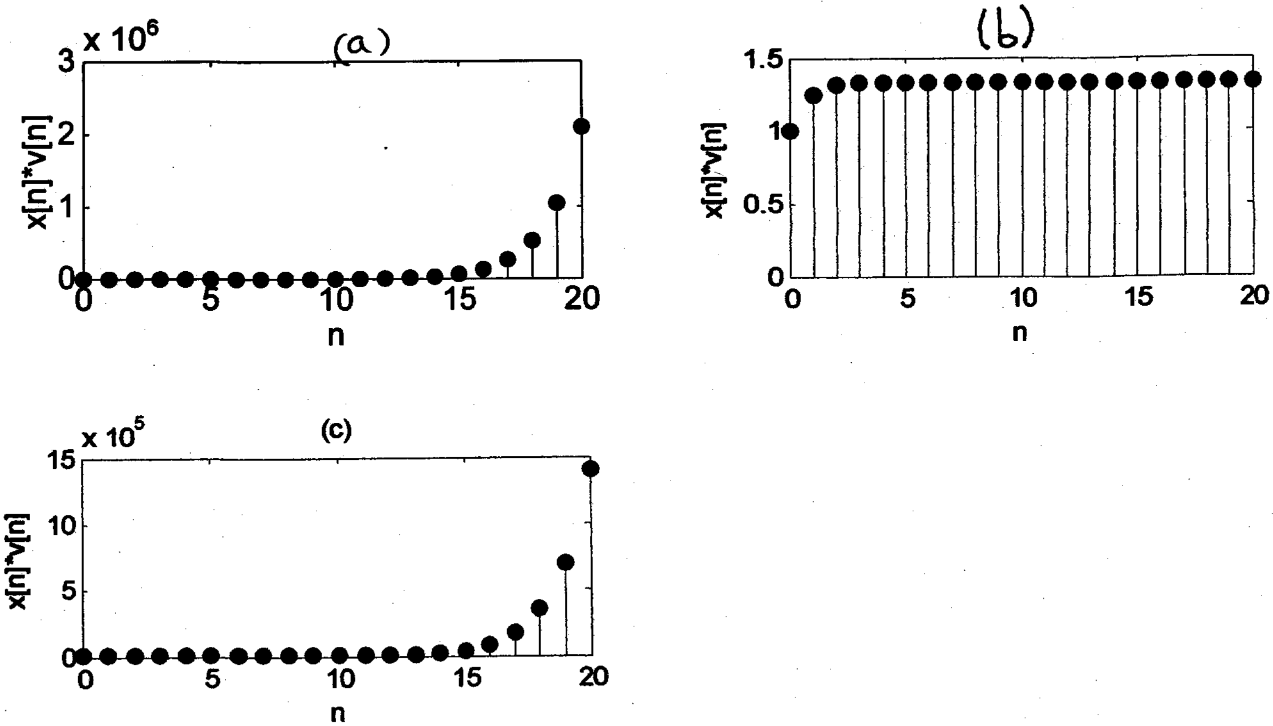
$$= (0.5)^{n} \sum_{i=0}^{n} 4^{i} = (0.5)^{n} \frac{1-4^{n+1}}{1-4}$$

$$= -\frac{1}{3}(0.5)^{n} + \frac{1}{3}(0.5)^{n} 4^{n+1}$$

$$= -\frac{1}{3}(0.5)^{n} + \frac{4}{3} 2^{n}, \quad n \geq 0$$

$v[n] * x[n] = 0, \ n < 0$

(d)



2.10 (a) from Problem 2.8(e), $u[n] * u[n] = n+1$

(b) $x[n] * x[n] = 0$ for $n < 0$

$x[n] * x[n] = \sum_{i=0}^{n} (1) = n+1$ for $0 \leq n \leq q$

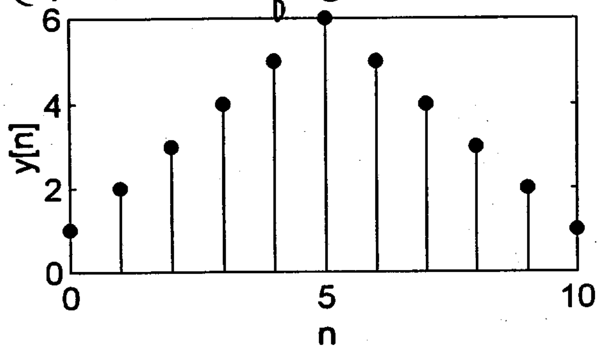
$x[n] * x[n] = \sum_{i=n-q}^{q} (1) = 2q - n + 1$ for $q < n \leq 2q$

2.10 (b) continued

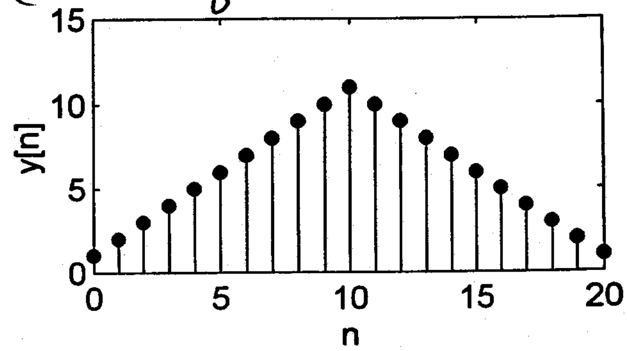
$x[n] * x[n] = 0$ for $n > 2q$

Hence, $x[n] * x[n] = u[n] * u[n]$ only when $n \le q$

(c) $q=5$



(d) $q=10$



The Matlab results match those predicted in part (b). It matches the convolution of two step functions only for $0 \le n \le q$.

2.11 **The MATLAB results match the analytical results.**

(a)
```
v = [1 -2 3 -4];
x = [4 1 -1];
y = conv(x,v)
```

yields

```
y =

     4    -7     9   -11    -7     4
```

↑ $n=0$

2.11 continued

(b)

```
v = [-1 0 8 2 3];
x = [1 1 1 1];
y = conv(x,v)
```

yields y = -1 -1 7 9 13 13 5 3

$n=-2$

(c)

```
n=-5:3;
x=2.^n;
v=[2 -3 0 6];
y=conv(x,v)
```

yields y = 0.0625 0.0313 0.0625 0.3125 0.6250 1.25 2.5 5 10 -12 24 48

$n=-5$

(d)

```
n=2:5;
x=1./n;
v=[-2 -5];
y=conv(x,v)
```

yields y = -1 -3.1667 -2.1667 -1.65 -1

$n=4$

(e)

```
n=0:5;
x=ones(1,6);
v=ones(1,6);
y=conv(x,v);
y=y(1:length(n)) % remove incorrect terms
```

yields y = 1 2 3 4 5 6

$n=0$

(f)

```
x=ones(1,6);
n=1:6;
v=log(n);
y=conv(x,v);
y=y(1:length(n))
```

yields y = 0 0.6931 1.7918 3.1781 4.7875

$n=1$

(g)

```
n=0:5;
x=[1 0 -1];
v=cos(pi*n/3);
y=conv(x,v);
y=y(1:length(n))
```

yields

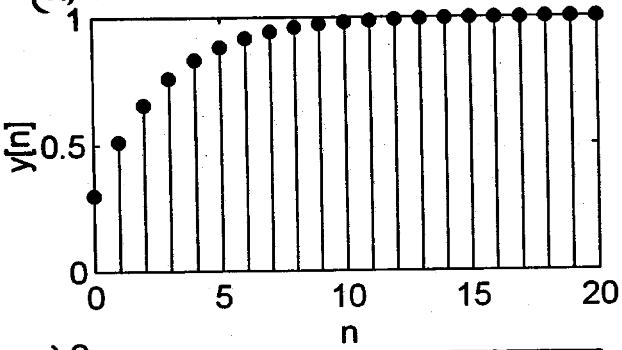
y = 1 0.5 -1.5 -1.5 0 1.5

$n=0$

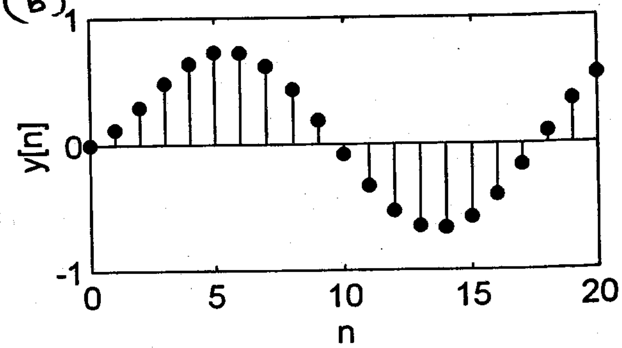
All of the above answers agree with the answers obtained in Problem 2.8.

2.12 See problem 2.10 regarding the numerical convolution of infinite duration signals

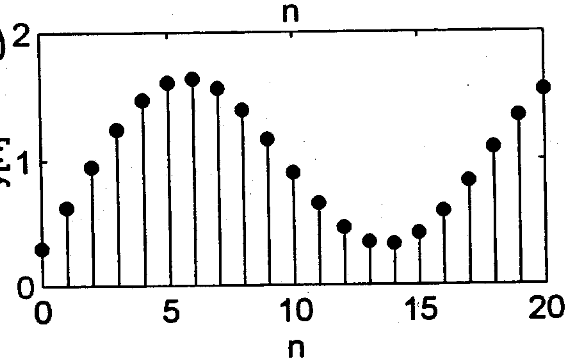
(a)



(b)



(c)



(d) $$y[n] = \sum_{i=-\infty}^{n} x[i]\,h[n-i]$$

$$y[n] = \sum_{i=-\infty} x[i](.3)(.7)^{n-i}$$

$$y[n-1] = \sum_{i=-\infty}^{n-1} x[i](.3)(.7)^{n-1-i}$$

A first oder difference equation has the form

$$y[n] + a y[n-1] = b_0 x[n] + b_1 x[n-1]$$

In this case,

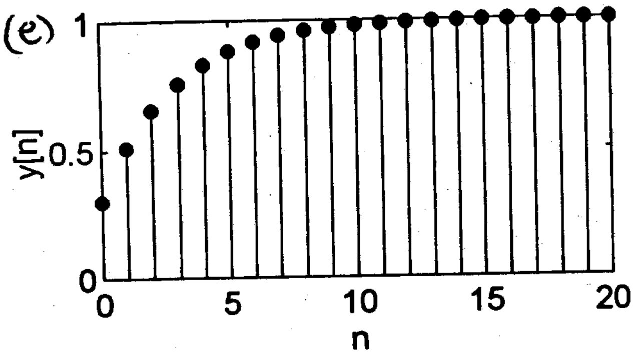
$$y[n] + a y[n-1] = \sum_{i=-\infty}^{n} x[i](.3)(.7)^{n-i} + a \sum_{i=-\infty}^{n-1} x[i](.3)(.7)^{n-1-i}$$

$$= \sum_{i=-\infty}^{n-1} x[i]\,.3(.7)^{n-i-1}(.7+a) + x[n](.3)$$

Let $a = -.7$, then $y[n] - .7\,y[n-1] = .3\,x[n]$

2.12 continued

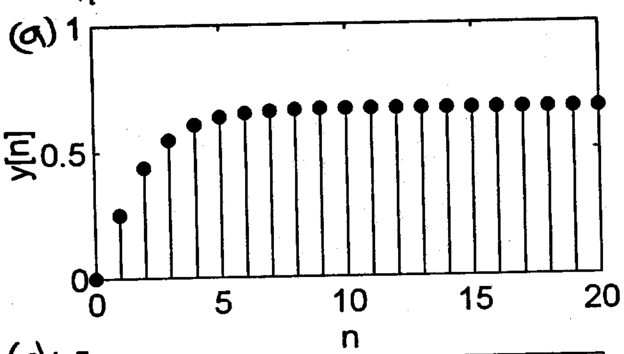
(e)



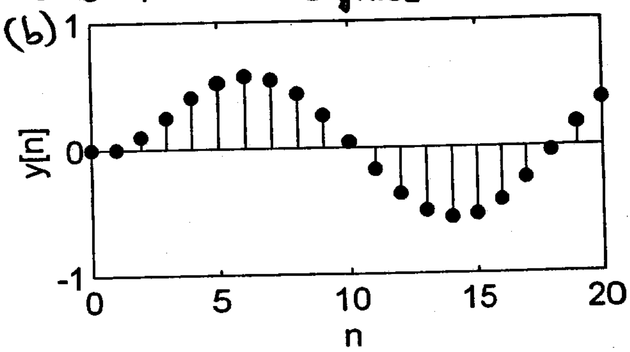
results from recur match those in part (a)

2.13 See the comments in Problem 2.10 regarding the numerical convolution of infinite duration signals

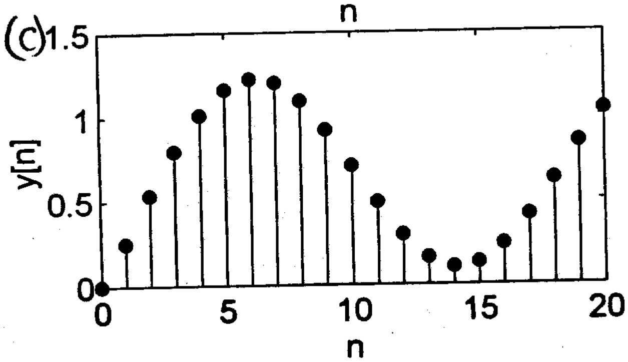
(a)



(b)



(c)



(d) Let $x[n]=\delta[n]$ and $y[n]=h[n]$, then

$$h[n+2]-.75\,h[n+1]+.125\,h[n]$$

$$=(.5^{n+2}-.25^{n+2})u[n+2]-.75(.5^{n+1}-.25^{n+1})u[n+1]$$
$$+.125\,(.5^{n}-.25^{n})\,u[n]$$
$$=.5^{n}(.25-.375+.125)+.25^{n}(-.0625+.1875-.125)$$

for $n\geq 0$

2.13 (d) continued

$$h[n+2] - .75h[n+1] + .125h[n]$$

$$= 0 \quad \text{for } n \geq 0$$

$$= .5^{n+1}(.5-.75) + .25^{n+1}(-.25+.75) \quad \text{for } n=-1$$

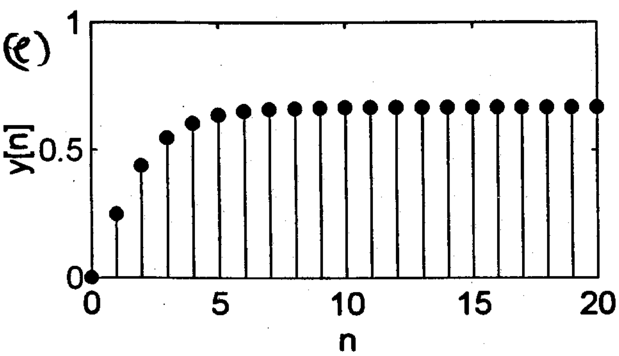
$$= 0.25 \quad \text{for } n=-1$$

$$= 0 \quad \text{for } n \leq -2$$

Therefore,

$h[n+2] - .75\,h[n+1] + .125\,h[n] = 0.25\,\delta[n+1]$
which satisfies the difference equation

(e)



The results found using recur are the same as those found in part (a)

2.14

$$y[n] = \left(1+\frac{I}{12}\right)^n y[0] - \sum_{i=1}^{n}\left(1+\frac{I}{12}\right)^{n-i} \underbrace{x(i)}_{c}$$

Setting $n = N$ gives

$$y(N) = \left(1+\frac{I}{12}\right)^N y(0) - \underbrace{\sum_{i=1}^{N}\left(1+\frac{I}{12}\right)^{N-i} c}_{\frac{12}{I}\left[\left(1+\frac{I}{12}\right)^N - 1\right]c}$$

Setting $y(N) = 0$ and solving for c

$$c = \frac{\left(1+\frac{I}{12}\right)^N y(0)}{\frac{12}{I}\left[\left(1+\frac{I}{12}\right)^N - 1\right]}$$

$$c = \frac{I\,y(0)}{12\left[1-\left(1+\frac{I}{12}\right)^{-N}\right]}$$

2.15 (a) $y[n+1] = 1.025\,y[n] + x(n+1)$

$$y[n+1] = 1.025\,y[n] + 1000, \quad n \geq 0$$

$$y(0) = 1000$$

$$y(1) = (1.025)(1000) + 1000 = 2025$$

$$y(2) = (1.025)(2025) + 1000 = 3075.625$$

$$y(3) = (1.025)(3075.625) + 1000 = 4152.5156$$

$$y(4) = (1.025)(4152.5156) + 1000 = 5256.3286$$

(b) $$y(n) = \sum_{i=1}^{n} \left(1 + \frac{I}{4}\right)^{n-i} x(i), \quad n \geq 1$$

$$y(N) = \sum_{i=1}^{N} \left(1 + \frac{I}{4}\right)^{N-i} c = c\,\frac{1 - \left(1 + \frac{I}{4}\right)^N}{1 - \left(1 + \frac{I}{4}\right)}$$

solving for N

$$\left(1 + \frac{I}{4}\right)^N = 1 + \frac{I}{4c}\,y(N)$$

$$N \ln\left(1 + \frac{I}{4}\right) = \ln\left(1 + \frac{I}{4c}\,y(N)\right)$$

$$N = \frac{\ln\left(1 + \frac{I}{4c}\,y(N)\right)}{\ln\left(1 + \frac{I}{4}\right)}$$

(c) $$y[N] = (1.0125)^N y[0] + \sum_{i=1}^{N} (1.0125)^{N-i} x[i]$$

$$= (1.0125)^N y[0] + c\,\frac{1 - (1.0125)^N}{1 - 1.0125}$$

$$= (1.0125)^N 2000 - \frac{5000}{.0125}\left(1 - (1.0125)^N\right)$$

$$= (1.0125)^N (2000 + 400000) - 400{,}000$$

Hence, $500{,}000 = 402{,}000\,(1.0125)^N - 400{,}000$

2.15 (c) continued

Then $(1.0125)^N = \frac{900,000}{402,000}$

Taking the ln of both sides and solving for N gives

$$N = \frac{\ln(2.2388)}{\ln(1.0125)} = 64.87755 \text{ quarters}$$

or $N = 16.219$ years

(d) The modified program for part (a) is shown below.
The answers match those found in (a)

```
% Investment program
% Program computes investment y[n]
y0 = input('Initial Investment ');
I = input('Yearly Interest rate ');
c = input('Quarterly Investment ');    % x[n] = c
y = [];      % defines y as an empty vector
y(1) = (1 + (I/4))*y0 + c;
for n=2:4,
  y(n) = (1 + (I/4))*y(n-1) + c;
  if y(n) < 0, break, end
end
% The following commands are for displaying the results
format bank
n=1:length(y); i = 1;
fprintf(' \n          n    y[n] in dollars ')
[n' y']
format short e
```

Running the above MATLAB program with the conditions in Part (c) for a longer number of steps yields y(64) = 490,240, y(65) = 501,370. Hence, a little more than 64 quarters (or 16 years) is required to reach \$500,000 in the savings account. Note that if the quarterly deposits of \$5,000 were stuffed under one's mattress instead of investing the money (so there is no interest earned), the amount after 16 years would be 16× \$20,000 = \$320,000. So even though a five percent interest rate is not that great, approximately \$180,000 of interest would have accumulated over the 16-year period.

2.16 (i) (a) $y[0] = -1.5\,y[-1] = -3$

$y[1] = -1.5\,y[0] = 4.5$

$y[2] = -1.5\,y[1] = -6.75$

2.16 continued

(b) $y[0] = 0$

$y[1] = 1$

$y[2] = -1.5\,y[1] + 1 = -0.5$

(c) $y[0] = -1.5\,y[-1] = -3$

$y[1] = -1.5\,y[0] + 1 = 5.5$

$y[2] = -1.5\,y[1] + 1 = -7.25$

(ii) (a) $y[0] = -.8\,y[-1] = -1.6$

$y[1] = -.8\,y[0] = 1.28$

$y[2] = -.8\,y[1] = -1.024$

(b) $y[0] = 0$

$y[1] = 1$

$y[2] = -.8\,y[1] + 1 = 0.2$

(c) $y[0] = -.8\,y[-1] = -1.6$

$y[1] = -.8\,y[0] + 1 = 2.28$

$y[2] = -.8\,y[1] + 1 = -0.824$

(iii) (a) $y[0] = .8\,y[-1] = 1.6$

$y[1] = .8\,y[0] = 1.28$

$y[2] = .8\,y[1] = 1.024$

(b) $y[0] = .8\,y[-1] = 0$

$y[1] = .8\,y[0] + 1 = 1$

$y[2] = .8\,y[1] + 1 = 1.8$

(c) $y[0] = .8\,y[-1] = 1.6$

$y[1] = .8\,y[0] + 1 = 2.28$

$y[2] = .8\,y[1] + 1 = 2.824$

2.17 (i) $$y[n]=(-1.5)^n y[0] + \sum_{i=1}^{n}(-1.5)^{n-i}x[i]$$

(a) $$y[n]=(-1.5)^n 2$$

(b) $$y[n]= \sum_{i=1}^{n}(-1.5)^{n-i}$$

this can be simplified further by using the relationship

$$\sum_{i=1}^{N} a^i = \frac{a-a^{N+1}}{1-a}$$

$$y[n]= (-1.5)^n \sum_{i=1}^{n}\left(\frac{-1}{1.5}\right)^i = \left(\frac{-2/3-(-2/3)^{n+1}}{5/3}\right)(-3/2)^n$$

$$y[n]= .4-.4\,(-3/2)^n \;,\; n\geq 0$$

(c) $$y[n]=(-1.5)^n 2 + \sum_{i=1}^{n}(-1.5)^{n-i}$$

from the answer to Part (b),

$$y[n]= (-1.5)^n 2 + 2 - 2(-3/2)^n \;,\quad n\geq 0$$

(ii) $$y[n] = (-.8)^n y[0] + \sum_{i=1}^{n}(-.8)^{n-i}x[i]$$

(a) $$y[n]=(-.8)^n 2$$

(b) $$y[n]= \sum_{i=1}^{n}(-.8)^{n-i}$$

$$= (-.8)^n\sum_{i=1}^{n}\left(-\frac{5}{4}\right)^i = \frac{-5/4-(-5/4)^{n+1}}{9/4}\left(\frac{-4}{5}\right)^n$$

$$y[n]= \frac{5}{9}-\frac{5}{9}\left(\frac{-4}{5}\right)^n \;,\; n\geq 0$$

2.17 continued

(c) $y[n] = (-.8)^n 2 + \frac{5}{9} - \frac{5}{9}\left(-\frac{4}{5}\right)^n, \; n \geq 0$

(iii) $y[n] = (.8)^n y[0] + \sum_{i=1}^{n} (.8)^{n-i} x[i]$

(a) $y[n] = (.8)^n 2$

(b) $y[n] = \sum_{i=1}^{n} (.8)^{n-i}$

$y[n] = 5 - 5\left(\frac{4}{5}\right)^n, \; n \geq 0$

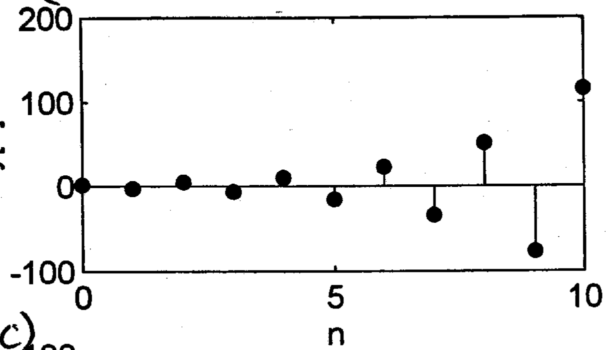
(c) $y[n] = (.8)^n 2 + 5 - 5\left(\frac{4}{5}\right)^n, \; n \geq 0$

(d) The computations required are given below for system (i), parts (a) and (b)

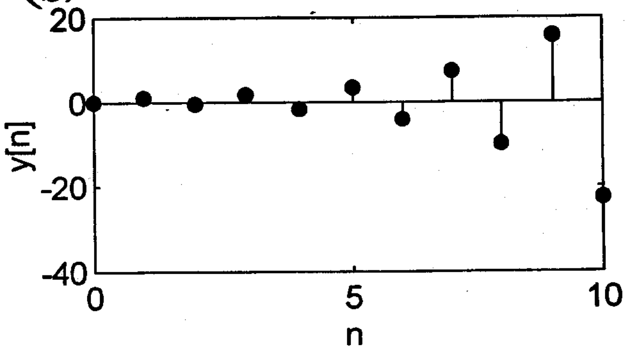
```
% (i), part a)
a =  1.5; b = [0 1];
y0 = 2; x0 = 0;
n = 1:10;
x = zeros(1,length(n));
y = recur(a,b,n,x,x0,y0);  % computes y for n=1 to 10
y = [2 y];  % augments y0 onto y
n = 0:10;
ya = 2*(-1.5).^n;  % analytical solution

% (i), part b)
a =  1.5; b = [0 1];
y0 = 0; x0 = 1;
n = 1:10;
x = ones(1,length(n));
y = recur(a,b,n,x,x0,y0);  % computes y for n=1 to 10
y = [0 y];  % augments y0 onto y
n = 0:10;
ya = .4-.4*(-1.5).^n; % analytical solution
```

2.17 continued

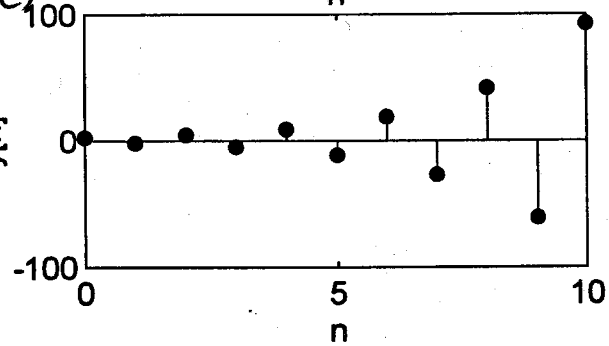
(i) (a)



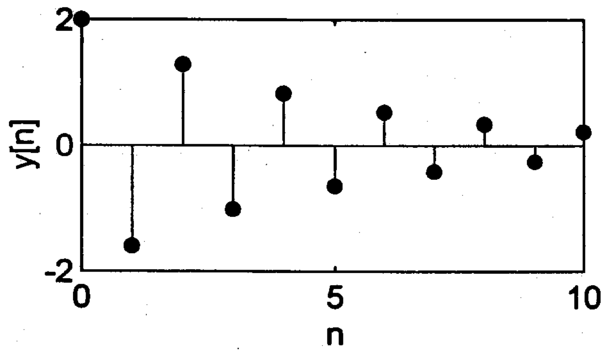
(b)



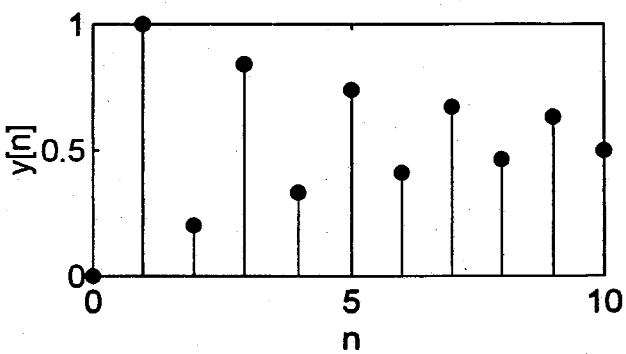
(c)



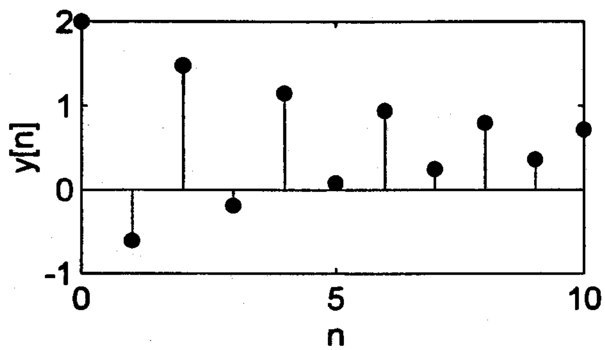
(ii) (a)



(b)

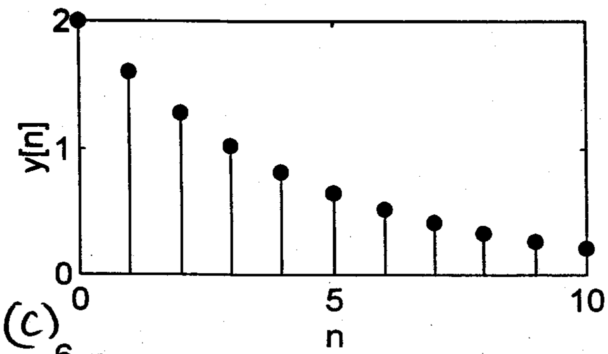


(c)

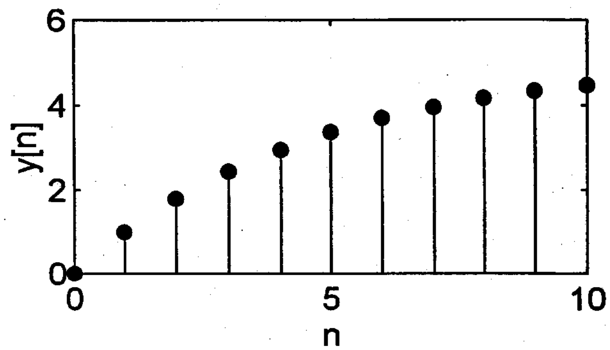


2.17 continued

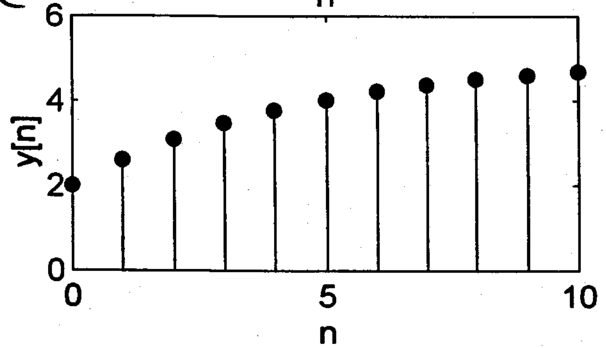
(iii) (a)



(b)



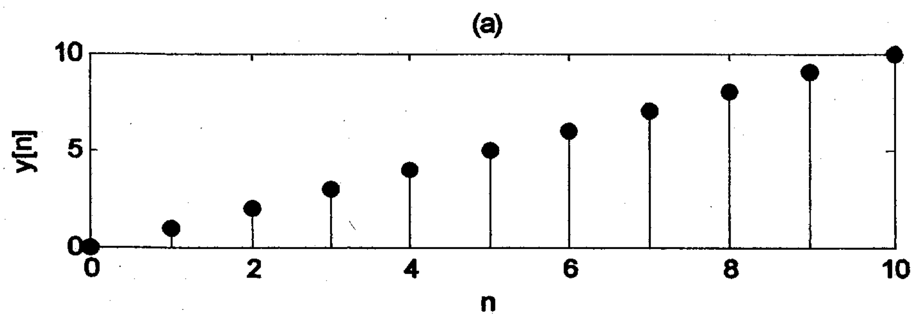
(c)



2.18 (a)
```
a=-1;b=[0 1];
n=0:10;
y0=0;x0=0;
x=ones(1,11);
y=recur(a,b,n,x,x0,y0)
stem(n,y,'filled')
xlabel('n')
ylabel('y[n]')
title('(a)')
```

yields

```
y =

     0     1     2     3     4     5     6     7     8     9    10
```



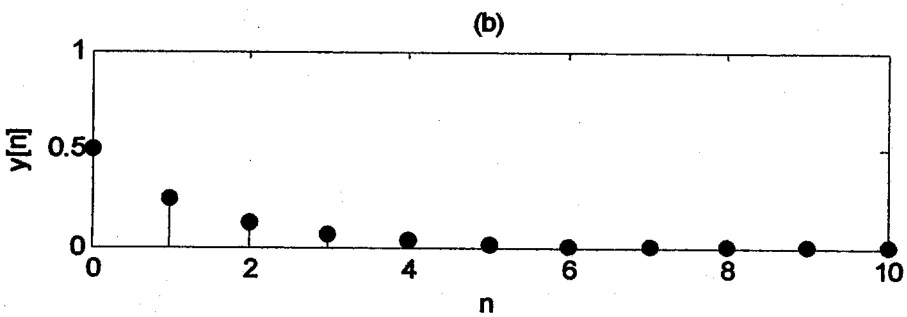
2.18 continued

(b)

```
a=-0.5;b=[0 0];
n=0:10;
y0=1;x0=0;
x=zeros(1,11);
y=recur(a,b,n,x,x0,y0)
stem(n,y,'filled')
xlabel('n')
ylabel('y[n]')
title('(b)')
```

yields

```
y =

    0.5000    0.2500    0.1250    0.0625    0.0313    0.0156    0.0078
    0.0039    0.0020    0.0010    0.0005
```

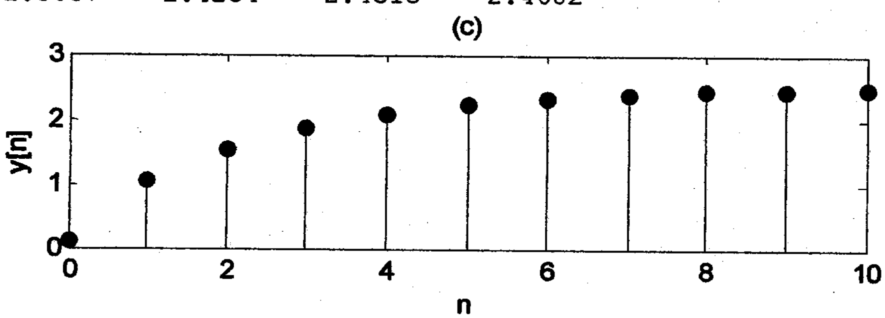


(c)

```
a=[-0.5 -0.1];b=[0 1];
n=0:10;
y0=[1 0];x0=0;
x=ones(1,11);
y=recur(a,b,n,x,x0,y0)
stem(n,y,'filled')
xlabel('n')
ylabel('y[n]')
title('(c)')
```

yields

```
y =

    0.1000    1.0500    1.5350    1.8725    2.0898    2.2321    2.3250
    2.3857    2.4254    2.4513    2.4682
```

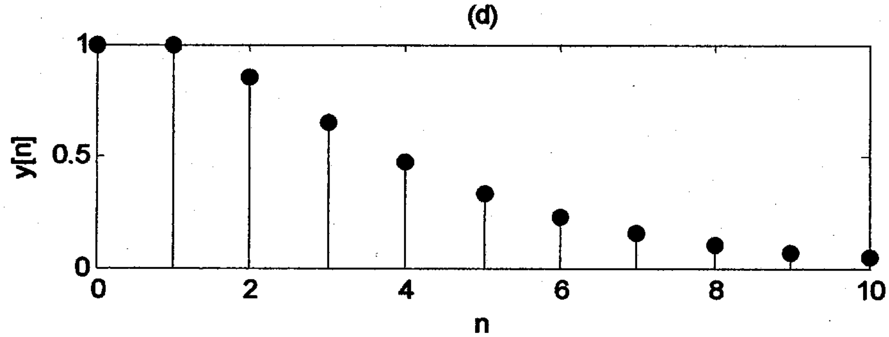


2.18 continued

(d)
```
a=[-0.1 -0.5];b=[1 0];
y0=[0 0];x0=0;
n=0:10;
x=(0.5).^n;
y=recur(a,b,n,x,x0,y0)
stem(n,y,'filled')
xlabel('n')
ylabel('y[n]')
title('(d)')
```

yields

```
y =

   1.0000    0.6000    0.8100    0.5060    0.5181    0.3361    0.3083
   0.2067    0.1787    0.1232    0.1026
```



2.19 $y[n] = -0.75y[n-1] - 0.125y[n-2] + x[n-2]$

(a) $y[0] = -(0.75)(2) - (0.125)(-1) + 0 = -1.375$
$y[1] = -(0.75)(-1.375) - (0.125)(2) + 0 = 0.78125$
$y[2] = -(0.75)(0.78125) - (0.125)(-1.375) + 0 = -0.4140625$
$y[3] = -(0.75)(-0.4140625) - (0.125)(0.78125) + 0 = 0.21289$

(b) $y[0] = -(0.75)(0) - (0.125)(0) + 1 = 1$
$y[1] = -(0.75)(1) - (0.125)(0) + 1 = 0.25$
$y[2] = -(0.75)(0.25) - (0.125)(1) + 1 = 0.6875$
$y[3] = -(0.75)(0.6875) - (0.125)(0.25) + 1 = 0.453125$

(c) $y[0] = -(0.75)(2) - (0.125)(-1) + 1 = -0.375$
$y[1] = -(0.75)(-0.375) - (0.125)(2) + 1 = 1.03125$
$y[2] = -(0.75)(1.03125) - (0.125)(-0.375) + 1 = 0.2734375$
$y[3] = -(0.75)(0.2734375) - (0.125)(1.03125) + 1 = 0.6660156$

(d) $y[0] = -(0.75)(3) - (0.125)(2) + 0 = -2.5$
$y[1] = -(0.75)(-2.5) - (0.125)(3) + 0 = 1.5$
$y[2] = -(0.75)(1.5) - (0.125)(-2.5) + 0 = -0.8125$
$y[3] = -(0.75)(-0.8125) - (0.125)(1.5) + 1 = 1.42185$

2.19 continued

(e) $y[0] = -(0.75)(4) - (0.125)(-2) + 0 = -2.75$
$y[1] = -(0.75)(-2.75) - (0.125)(4) + 0 = 1.5625$
$y[2] = -(0.75)(1.5625) - (0.125)(-2.75) + 0 = -0.82825$
$y[3] = -(0.75)(-0.82825) - (0.125)(1.5625) + 1 = 1.425875$

2.20 $$L\frac{di(t)}{dt} + Ri(t) + \frac{1}{C}\int_0^t i(\lambda)d\lambda = x(t)$$

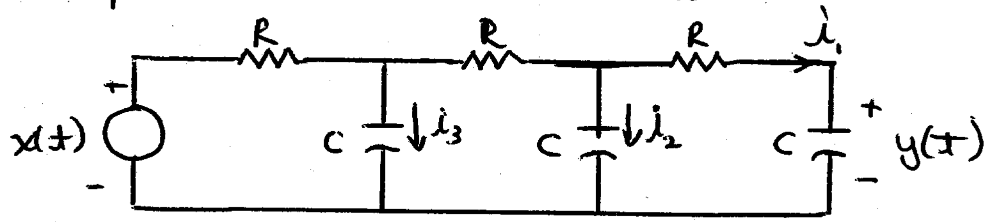
(b) $$L\frac{d^2 i(t)}{dt^2} + R\frac{d\,i(t)}{dt} + \frac{1}{C}i(t) = \frac{dx(t)}{dt}$$

(a) $V_c(t) = \frac{1}{C}\int_{-\infty}^{t} i(\lambda)\,d\lambda$, so

$$L\frac{d^2 v_c(t)}{dt^2} + R\frac{dv_c(t)}{dt} + \frac{1}{C}v_c(t) = \frac{1}{C}x(t).$$

2.21 (a) By the voltage division rule, $y(t) = \frac{1}{2}x(t)$

(b) Assign the currents shown below:



Using Kirchoff's laws, we have

$$y(t) + RC\,\dot{y}(t) = \frac{1}{C}\int i_2(\lambda)d\lambda$$

$$\frac{1}{C}\int i_2(\lambda)d\lambda + R\left(i_1(t) + i_2(t)\right) = \frac{1}{C}\int i_3(\lambda)d\lambda$$

Combining gives

$$y(t) + RC\,\dot{y}(t) + R\left(i_1(t) + i_2(t)\right) = \frac{1}{C}\int i_3(\lambda)d\lambda$$

Now $i_1(t) = C\dot{y}(t)$ and $i_2(t) = C\dot{y}(t) + RC^2\ddot{y}(t)$

Thus $y(t) + 3RC\,\dot{y}(t) + R^2C^2\ddot{y}(t) = \frac{1}{C}\int i_3(\lambda)d\lambda$

2.21 (b) continued

Also, $\frac{1}{C}\int i_3(\lambda)d\lambda + R\left(i_1(t) + i_2(t) + i_3(t)\right) = x(t)$

and $i_3(t) = C\dot{y}(t) + 3RC^2\ddot{y}(t) + R^2C^3\dddot{y}(t)$

Combining equation gives

$$y(t) + 6RC\dot{y}(t) + 5R^2C^2\ddot{y}(t) + R^3C^3\dddot{y}(t) = x(t)$$

or $$R^3C^3\dddot{y}(t) + 5R^2C^2\ddot{y}(t) + 6RC\dot{y}(t) + y(t) = x(t)$$

2.22 $$M\frac{d^2y(t)}{dt^2} + D\frac{dy(t)}{dt} + Ky(t) = x(t) + Mg$$

2.23 Using D'Alembert's principle, we have

$$M_1\frac{d^2q(t)}{dt^2} = -k_1q(t) + k_2\left(y(t) - q(t)\right) + x(t)$$

$$M_2\frac{d^2y(t)}{dt^2} = -k_2\left(y(t) - q(t)\right) - k_3y(t)$$

Differentiating the second equation twice and using the first equation yields

$$M_2\frac{d^4y(t)}{dt^4} = -(k_2+k_3)\frac{d^2y(t)}{dt^2} + \frac{k_2}{M_1}\left[-k_1q(t) + k_2\left(y(t) - q(t)\right) + x(t)\right]$$

But $q(t) = \frac{1}{k_2}\left[M_2\frac{d^2y(t)}{dt^2} + (k_2+k_3)y(t)\right]$

Combining equations gives

$$M_2\frac{d^4y(t)}{dt^4} + \left[k_2 + k_3 + \frac{M_2}{M_1}(k_1+k_2)\right]\frac{d^2y(t)}{dt^2} +$$

$$\frac{1}{M_1}\left[(k_1+k_2)(k_2+k_3) - k_2^2\right]y(t) = \frac{k_2}{M_1}x(t)$$

2.24 (a) Summing currents we have

$$\frac{1}{R}y(t) + \frac{1}{L}\int_{-\infty}^{t} y(\lambda)\,d\lambda = i(t)$$

Differentiating both sides yields

$$\frac{1}{R}\frac{dy(t)}{dt} + \frac{1}{L}y(t) = \frac{di(t)}{dt}$$

so with $R = L = 1$, we have

$$\frac{dy(t)}{dt} + y(t) = \frac{di(t)}{dt}$$

(b) $\frac{di(t)}{dt} = \delta(t) - \delta(t-1)$. **Using the Symbolic Math Toolbox:**

```
y=dsolve('Dy=-y+dirac(t)-dirac(t-1)','y(0)=1')
```

yields

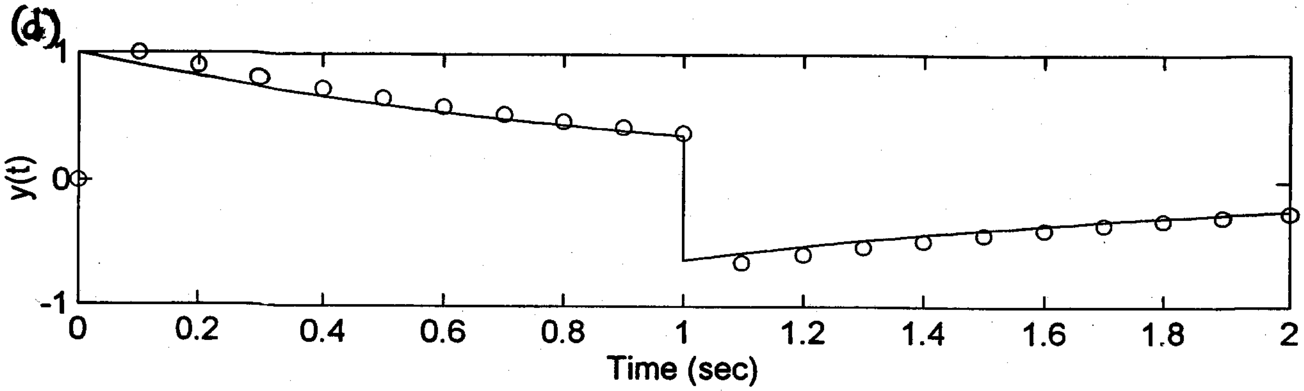
```
y =

(heaviside(t)-exp(1)*heaviside(t-1))*exp(-t)
```

Here `heaviside(t)` is the step function $u(t)$.

(c) From the results in Section 2.5,

$$\frac{y[n+1] - y[n]}{T} + y[n] = \frac{i[n+1] - i[n]}{T}$$

(d)



Approximate solution from part (c) is close to the exact solution (solid line), except at $t = 0$.

2.24 (e) The MATLAB command `ode45` cannot handle functions with impulses, and thus it is first necessary to rewrite the input/output differential equation as follows: Define $v(t)$ so that $\frac{dv(t)}{dt}+v(t)=i(t)$. Then we claim that $y(t)=i(t)-v(t)$. To prove this, first take the derivative of both sides of $y(t)=i(t)-v(t)$, which gives

$$\frac{dy(t)}{dt}=\frac{di(t)}{dt}-\frac{dv(t)}{dt}=\frac{di(t)}{dt}+v(t)-i(t)$$

But $v(t)=i(t)-y(t)$, and inserting this into the above equation gives the input/output differential equation of the circuit. Since the input $i(t)$ is a pulse equal to 1 for $0 < t < 1$ and equal to 0 for $t > 1$, the MATLAB program for using the ODE solver requires the following function file `RL1_func.m` containing the commands:

```
function dv = RL1_func(t,v);
dv = 1 - v;
```

and the function file `RL2_func.m` containing the commands:

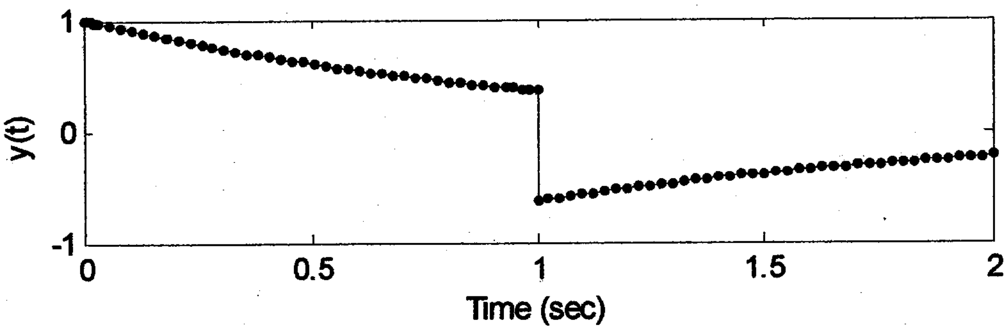
```
function dv = RL2_func(t,v);
dv = - v;
```

In terms of these two files the MATLAB program for computing the approximate and exact responses is as follows:

```
tspan=[0 1];
v0=0;
[t,v]=ode45(@RL1_func,tspan,v0);
y1=1-v; % approximate solution for 0 < t < 1
plot(t,y1,'.')
hold on
y2=exp(-t); % exact solution for 0 < t < 1
plot(t,y2)
a=exp(-1);
tspan=[1 2];
v0=0.6321; % value of v(t) at t = 1
[t,v]=ode45(@RL2_func,tspan,v0);
y2=-v; % approximate solution for t > 1
plot(t,y2,'.')
y2=(1-exp(1))*exp(-t); % exact solution for t > 1
plot(t,y2)
n=[1 1];
y=[a y2(1)];
plot(n,y)
xlabel('Time (sec)')
ylabel('y(t)')
hold off
```

This program results in the following plot:

2.24(e) continued



In the above plot, the dotted curve is the approximate solution and the solid curve is the exact solution, both of which are virtually identical as seen from the plot.

2.25 (a)

$$\frac{dy(t)}{dt} = -.5e^{-.5t}\left(\sin(2t) + 2\cos(2t)\right) + e^{-.5t}\left(2\cos(2t) - 4\sin(2t)\right)$$

$$\frac{d^2y(t)}{dt^2} = .25e^{-.5t}\left(\sin(2t) + 2\cos(2t)\right) - .5e^{-.5t}\left(2\cos(2t) - 4\sin(2t)\right) - .5e^{-.5t}\left(2\cos(2t) - 4\sin(2t)\right) + e^{-.5t}\left(-4\sin(2t) - 8\cos(2t)\right)$$

Substitution of these expressions and the solution $y(t)$ into the differential equation yields

$$e^{-.5t}\sin(2t)\left[.25 + 2 + 2 - 4\right] + e^{-.5t}\cos(2t)\left[.5 - 1 - 1 - 8\right] +$$

$$e^{-.5t}\sin(2t)\left[-.5 - 4\right] + e^{-.5t}\cos(2t)\left[-1 + 2\right] +$$

$$e^{-.5t}\sin(2t)(4.25) + e^{-.5t}\cos(2t)(8.50) = 0$$

So, the differential equation is satisfied. Also, the solution for $y(t)$ satisfies the initial conditions.

2.25 continued

(b) $$\frac{y[n+2]-2y[n+1]+y[n]}{T^2} + \frac{y[n+1]-y[n]}{T} + 4.25\,y[n] = x[n]$$

$$y[n+2] + y[n+1](T-2) + y[n](1-T+T^2 4.25) = T^2 x[n]$$

(c) With $T = .1$

$$y[n+2] - 1.9\,y[n+1] + 0.9425\,y[n] = 0.1\,x[n]$$

$y[0] = 2$ and $y[1] = y[0] + T\dot{y}(0) = 2.1$

(d) With $T = 0.05$

$$y[n+2] - 1.95\,y[n+1] + 0.9606\,y[n] = 0.05\,x[n]$$

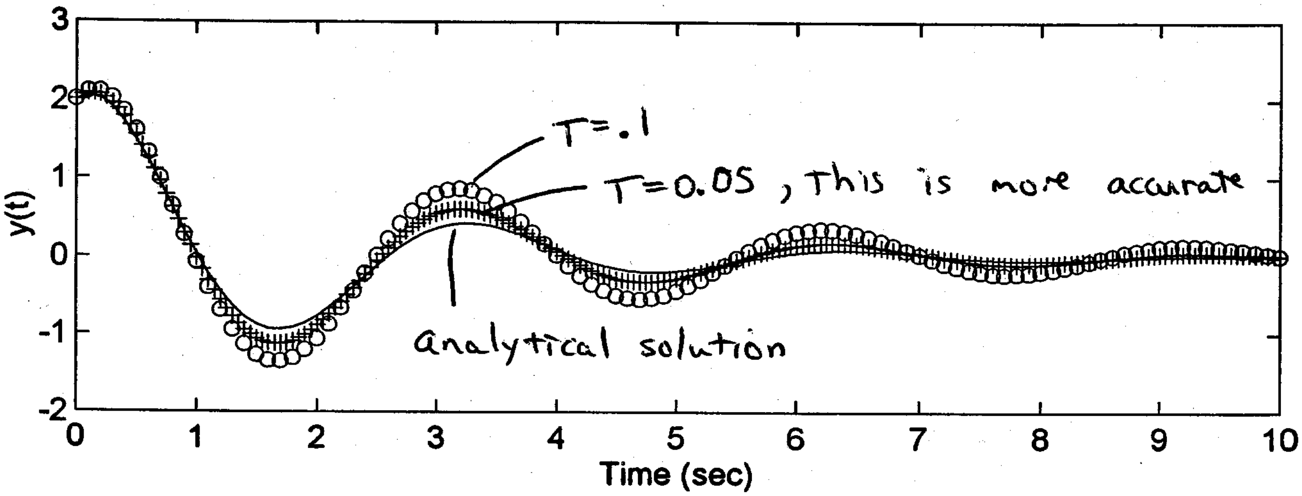
$y[0] = 2$ and $y[1] = 2.05$

The following commands can be used to calculate the approximation when $T = 0.1$

```
T = 0.1;
a = [T-2 1-T+4.25*T*T];
b = [0 0 0];
y0 = [2 2+T]
x0 = [0 0];
n = 2:100;
x = zeros(1,length(n));
y1 = recur(a,b,n,x,x0,y0);
y1 = [y0 y1];
n1 = 0:100;
subplot(211),plot(n1,y1,'o')
```

2.25 continued

(e)



2.26 (a) The MATLAB command

```
y = dsolve('D2y = -3*Dy - 2*y','Dy(0) = 0','y(0) = 1')
```

yields

```
y =

-exp(-2*t)+2*exp(-t)
```

(b) $\dfrac{y[n+2]-2y[n+1]+y[n]}{T^2} + 3\,\dfrac{y[n+1]-y[n]}{T} + 2y[n] = x[n]$

$y[n+2] + y[n+1](3T-2) + y[n](1-3T+2T^2) = x[n]$

With $T=0.4$,

$y[n+2] - 0.8y[n+1] + 0.12y[n] = x[n]$

$y[0]=1$ and $y[1] = y(0) + T\dot{y}(0) = 1$

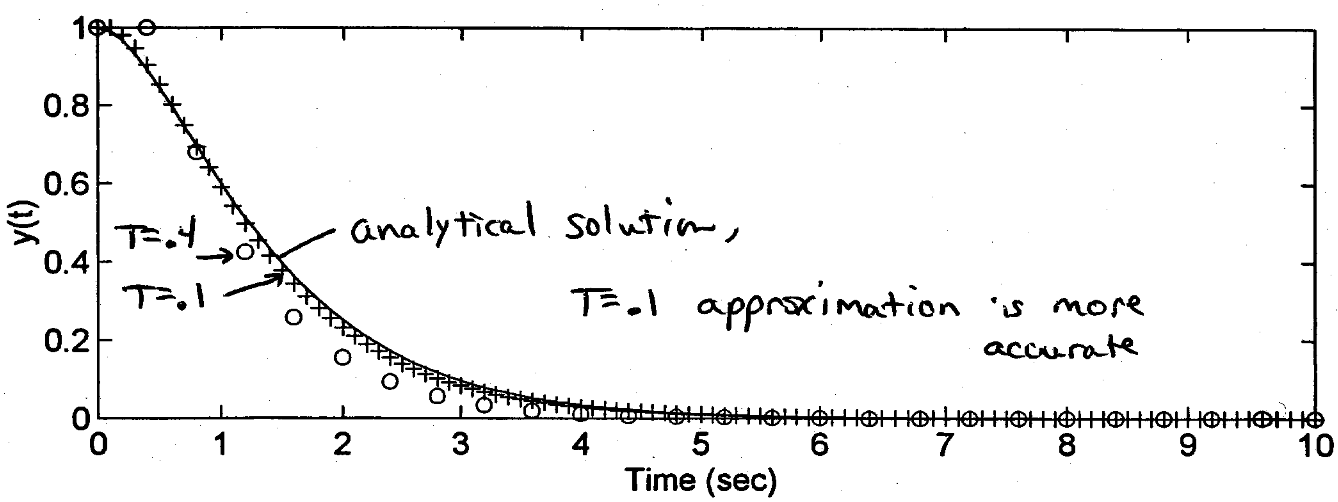
(c) With $T=0.1$,

$y[n+2] - 1.7y[n+1] + 0.72y[n] = x[n]$

$y[0]=1$ and $y[1]=1$

2.26 (c) continued

The Matlab code needed to compute the approximations is similar to that used for Problem 2.25



2.26 (d) Let $Y(t)=\begin{bmatrix} y(t) \\ \dot{y}(t) \end{bmatrix}$, so that $Y_1(t)=y(t)$ and $Y_2(t)=\dot{y}(t)$. Then

$$\frac{dY_1(t)}{dt}=\dot{y}(t)=Y_2(t)$$

$$\frac{dY_2(t)}{dt}=\ddot{y}(t)=-3\frac{dy(t)}{dt}-2y(t)=-3Y_2(t)-2Y_1(t)$$

with the initial conditions $Y_1(0)=1$, $Y_2(0)=0$. Then solving the differential equation using the ODE solver requires the function file de_func containing the commands:

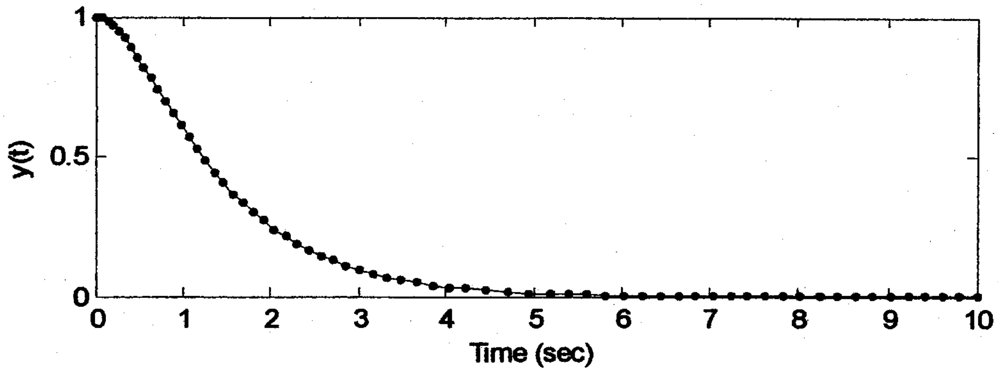
```
function dY = de_func(t,Y)
dY=zeros(2,1); % defines Y to be a two-element column vector
dY(1)=Y(2); % Y(1) and Y(2) are the first and second elements of Y
dY(2)=-3*Y(2)-2*Y(1);
```

In terms of this file the MATLAB program for computing the approximate and exact responses is as follows:

```
tspan = [0 10];
Y0 = [1 0];
[t,Y]=ode45(@de_func,tspan,Y0);
plot(t,Y(:,1),'.') % approximate response
hold on
y=-exp(-2*t)+2*exp(-t); % exact solution
plot(t,y)
xlabel('Time (sec)')
ylabel('y(t)')
hold off
```

2.26 (d) continued

The resulting plot is:



The dotted line is the approximate solution and the solid line is the exact solution. Clearly, they are identical.

(e) See the above plots in the solutions for Parts (c) and (d).

2.27 (a) $$\frac{dy(t)}{dt} = -2e^{-t} + e^{-t} - te^{-t} = -e^{-t} - te^{-t}$$

$$\frac{d^2y(t)}{dt^2} = e^{-t} - e^{-t} + te^{-t} = te^{-t}$$

Substitution of these expressions and $y(t)$ into the differential equation yields

$$te^{-t} - 2e^{-t} - 2te^{-t} + 2e^{-t} + te^{-t} = 0$$

Also, the initial conditions are satisfied.

(b) $$\frac{y[n+2] - 2y[n+1] + y[n]}{T^2} + 2\,\frac{y[n+1] - y[n]}{T} + y[n] = x[n]$$

$$y[n+2] + y[n+1](2T-2) + y[n](T^2 - 2T + 1) = x[n]$$

(c) With $T = .4$,

$$y[n+2] + y[n+1](-1.2) + .36y[n] = x[n]$$

with $y[0] = 2$ and $y[1] = y(0) + T\dot{y}(0) = 1.6$

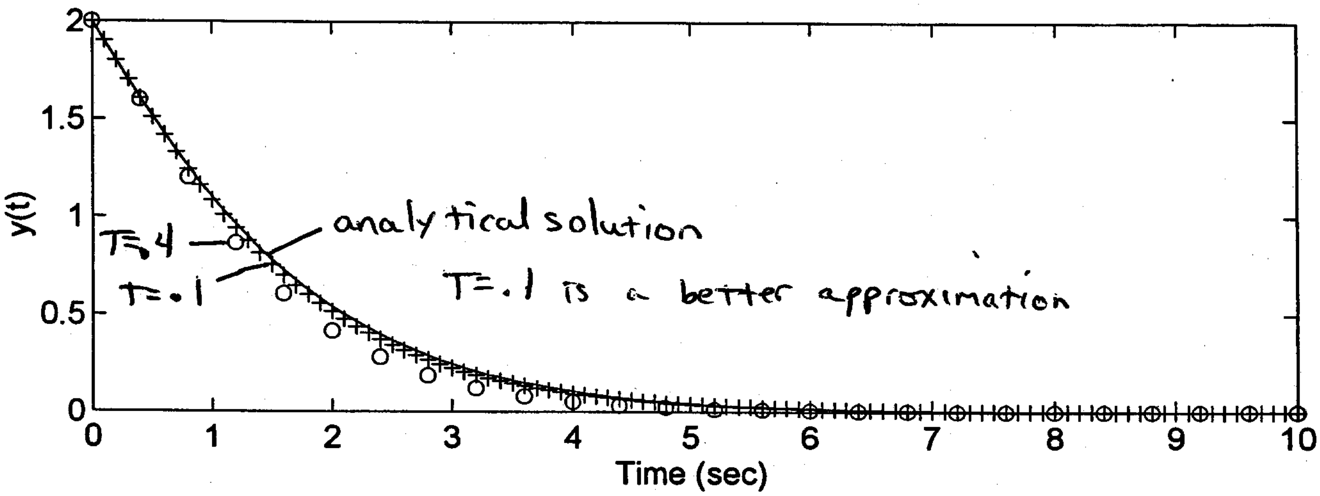
2.27 continued

(d) with $T = .1$,

$$y[n+2] - 1.8\,y[n+1] + .81\,y[n] = x[n]$$

with $y[0] = 2$ and $y[1] = 1.9$

(e)



2.28 As in the solution to Problem 2.26(d), let $Y(t) = \begin{bmatrix} y(t) \\ \dot{y}(t) \end{bmatrix}$, so that $Y_1(t) = y(t)$ and $Y_2(t) = \dot{y}(t)$. Then

$$\frac{dY_1(t)}{dt} = Y_2(t)$$

$$\frac{dY_2(t)}{dt} = -DY_2t - KY_1(t) + x(t)$$

(a) Solving the differential equation using the ODE solver requires the function file `de2_func` containing the commands:

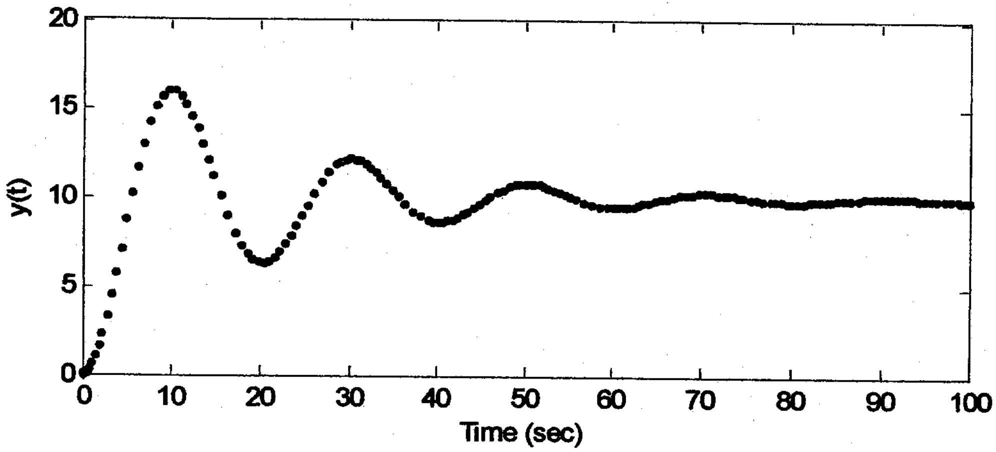
```
function dY = de2_func(t,Y)
dY=zeros(2,1);
dY(1)=Y(2);
dY(2)=-0.1*Y(2)-0.1*Y(1)+1;
```

In terms of this file the MATLAB program for computing the approximation to the output response is:

```
tspan = [0 100];
Y0 = [0 0];
[t,Y]=ode45(@de2_func,tspan,Y0);
plot(t,Y(:,1),'.')
xlabel('Time (sec)')
ylabel('y(t)')
```

2.28 continued

(b) The program in Part (a) yields the following plot:



Note that the response $y(t)$ reaches a steady-state value of 10.

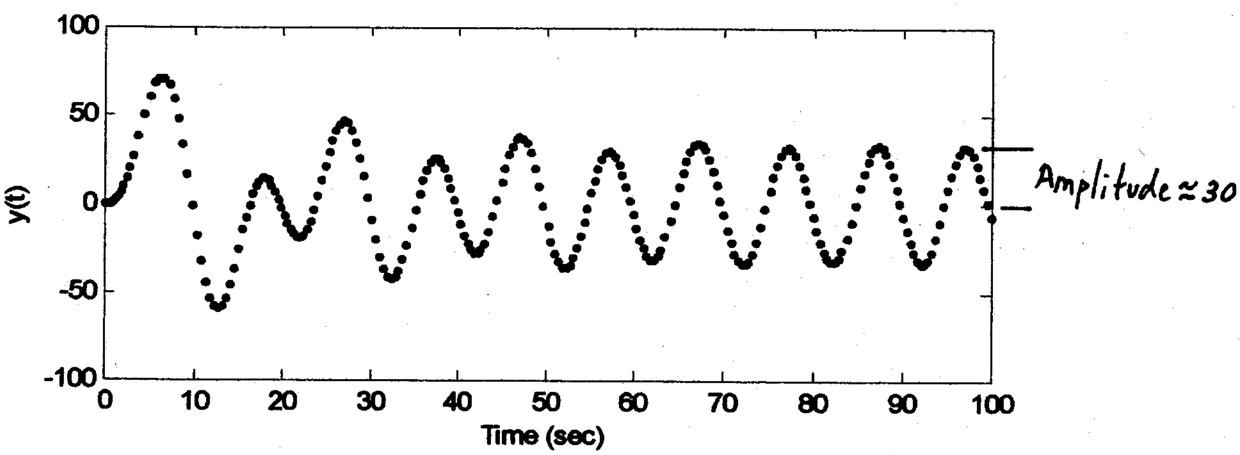
(c) Generate the function file `de3_func` containing the commands:

```
function dY = de3_func(t,Y)
dY=zeros(2,1);
dY(1)=Y(2);
dY(2)=-0.1*Y(2)-0.1*Y(1)+10*sin(0.2*pi*t);
```

In terms of this file the MATLAB program for computing the approximation to the output response is:

```
tspan = [0 100];
Y0 = [0 0];
[t,Y]=ode45(@de3_func,tspan,Y0);
plot(t,Y(:,1),'.')
xlabel('Time (sec)')
ylabel('y(t)')
```

This produces the response



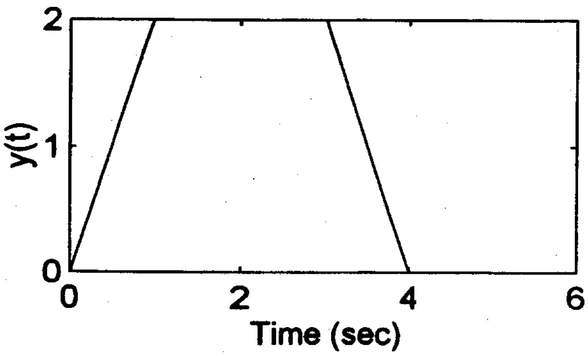
2.28 continued

From the plot, the amplitude of the sinusoid is approximately equal to 30.

(d) The results in Parts (b) and (c) correspond very closely to the results from the online demo.

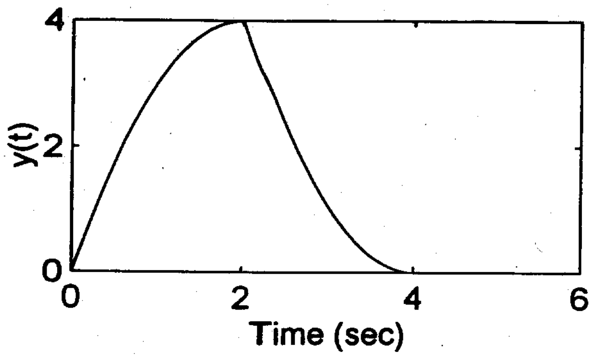
2.29 (a)

$$x(t)*v(t)=\begin{cases} 2t, & 0 \le t \le 1 \\ 2, & 1 \le t \le 3 \\ -2t+8, & 3 \le t \le 4 \\ 0, & t \ge 4 \end{cases}$$



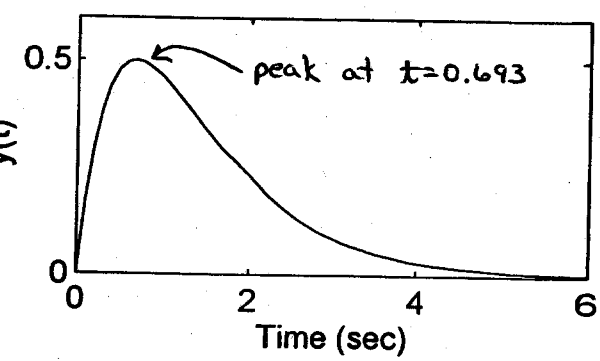
(b)

$$x(t)*v(t)=\begin{cases} -t^2+4t, & 0 \le t \le 2 \\ t^2-8t+16, & 2 \le t \le 4 \\ 0, & t \ge 4 \end{cases}$$



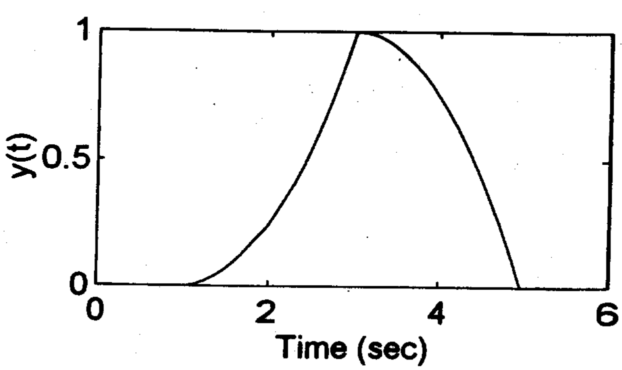
(c)

$$x(t)*v(t)=\begin{cases} 2e^{-t}(1-e^{-t}), & 0 \le t \le 2 \\ 2e^{-t}(1-e^{-2}), & t \ge 2 \end{cases}$$



(d)

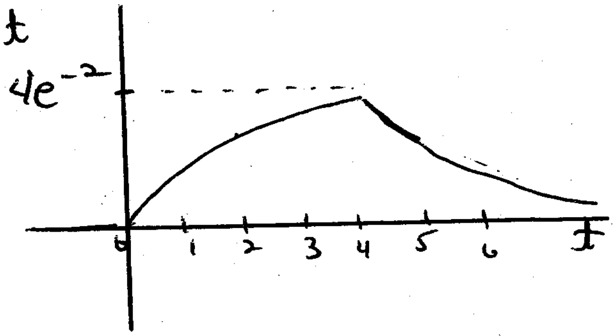
$$x(t)*v(t)=\begin{cases} 0, & 0 \le t \le 1 \\ \frac{1}{4}(t-1)^2, & 1 \le t \le 3 \\ \frac{1}{4}(5-t)(t-1), & 3 \le t \le 5 \\ 0, & t \ge 5 \end{cases}$$



2.29 continued

(e)

$$x(t)*v(t) = \begin{cases} 0 & t \le 0 \\ 4e^{-2}(1-e^{-2t}), & 0 < t \le 4 \\ 4e^{-2t}(e^{6}-e^{-2}), & 4 < t \end{cases}$$



2.30 For $t \le -1$, $x(t)*v(t) = 0$

For $-1 \le t \le 0$, $x(t)*v(t) = \int_0^{t+1} 2\,d\lambda = 2t+2$

For $0 \le t \le 1$, $x(t)*v(t) = \int_0^{t} 1\,d\lambda + \int_t^{1} 2\,d\lambda + \int_1^{t+1} 4\,d\lambda = 3t+2$

For $1 \le t \le 2$, $x(t)*v(t) = \int_{t-1}^{1} 1\,d\lambda + \int_1^{t} 2\,d\lambda + \int_t^{2} 4\,d\lambda = -3t+8$

For $2 \le t \le 3$, $x(t)*v(t) = \int_{t-1}^{2} 2\,d\lambda = -2t+6$

For $t \ge 3$, $x(t)*v(t) = 0$

2.31 (a) $h(t) = \int_{-\infty}^{t} (t-\lambda+2)\,\delta(\lambda)\,d\lambda = \begin{cases} t+2, & t \ge 0 \\ 0, & t < 0 \end{cases}$

(b) for $1 \le t \le 2$

$$y(t) = \int_0^1 (t-\lambda+2)\,d\lambda + \int_1^t -(t-\lambda+2)\,d\lambda$$

$$= t\lambda - \frac{\lambda^2}{2} + 2\lambda \Big|_{\lambda=0}^{\lambda=1} - \left[t\lambda - \frac{\lambda^2}{2} + 2\lambda\right]_{\lambda=1}^{\lambda=t}$$

$$= t + \frac{3}{2} - \left[t^2 - \frac{t^2}{2} + 2t\right] + \left[t + \frac{3}{2}\right] = -\frac{t^2}{2} + 3$$

2.32 (a) $g(t) = \int_0^t h(\lambda)d\lambda = \int_0^t [e^{-\lambda} + \sin\lambda]\,d\lambda$

$$= -e^{-t} - \cos t + 2, \quad t \geq 0$$

(b) Response to $u(t) - u(t-2)$ is $g(t) - g(t-2)$

Hence,

$$y(t) = (-e^{-t} - \cos t + 2)u(t) - (e^{-(t-2)} - \cos(t-2) + 2)u(t-2)$$

2.33 $y(t) = h(t) * x(t)$

For $t \leq 2$, $y(t) = 0$

For $2 \leq t \leq 3$, $y(t) = \int_2^t \sin\lambda\, d\lambda = -\cos t + \cos 2$

For $t \geq 3$,

$$y(t) = \int_2^t \sin\lambda\, d\lambda - \int_2^{t-1} \sin\lambda\, d\lambda$$

$$= -\cos t + \cos(t-1)$$

2.34 (a) Both systems have impulse response e^{-t}, $t \geq 0$.

(b) $h(t) = e^{-t} * e^{-t} = te^{-t}$, $t \geq 0$

(c) The input/output differential equation of the *RLC* circuit is

$$\ddot{y}(t) + \dot{y}(t) = \dot{v}(t) = -v(t) + x(t) = -[\dot{y}(t) + y(t)] + x(t)$$

which can be rewritten as

$$\ddot{y}(t) + 2\dot{y}(t) + y(t) = x(t)$$

Using the function file `de4_func` containing the commands:

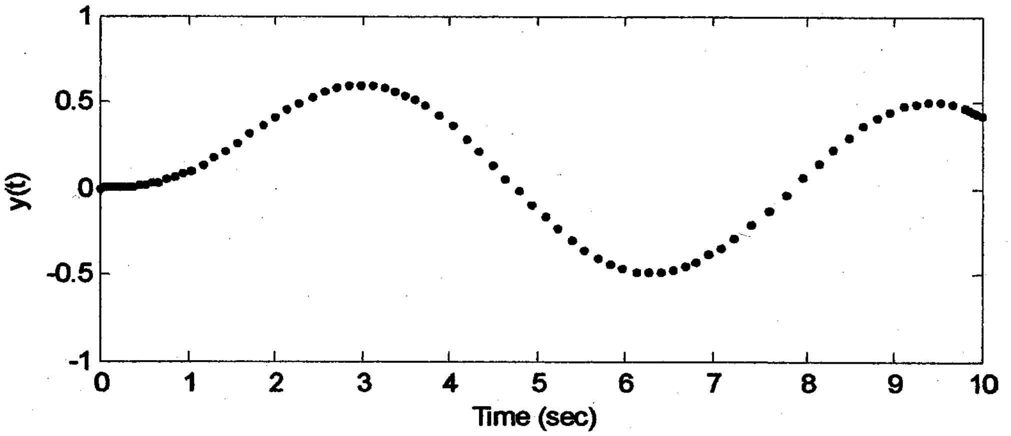
```
function dY = de4_func(t,Y)
dY=zeros(2,1);
dY(1)=Y(2);
dY(2)=-2*Y(2)-Y(1)+sin(t);
```

and the MATLAB program

```
tspan = [0 10];
Y0 = [0 0];
[t,Y]=ode45(@de4_func,tspan,Y0);
plot(t,Y(:,1),'.')
xlabel('Time (sec)')
ylabel('y(t)')
```

2.34 (c) continued

we obtain the following output response:



(d) The discretized input/output equation is

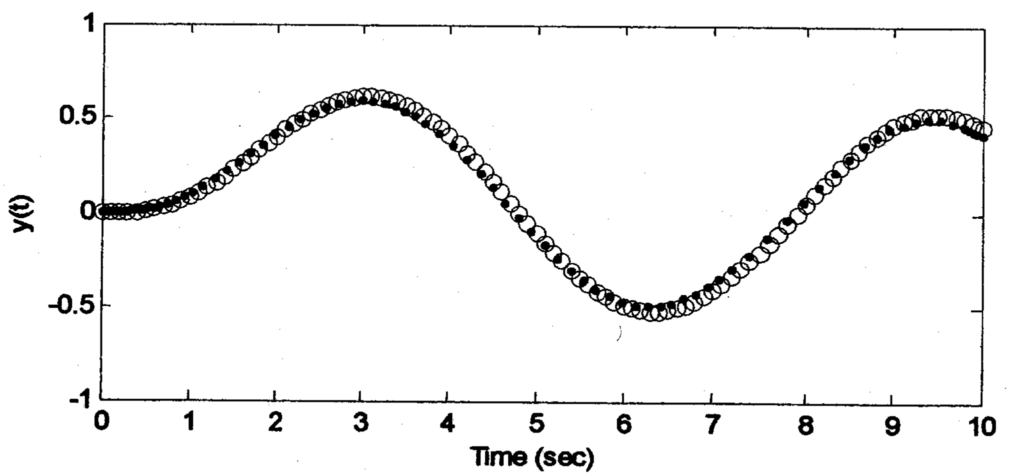
$$\frac{y[n+2]-2y[n+1]+y[n]}{T^2}+2\frac{y[n+1]-y[n]}{T}+y[n]=x[n]$$

$$y[n+2]+(2T-2)y[n+1]+\left(1-2T+T^2\right)y[n]=T^2x[n]$$

When $T=0.1$, the discretized equation becomes $y[n+2]-1.8y[n+1]+0.81y[n]=0.01x[n]$. Replacing n by n-2 gives $y[n]-1.8y[n-1]+0.81y[n-2]=0.01x[n-2]$. The MATLAB program for computing the approximate output response using `recur` is

```
a=[-1.8 0.81];b=[0 0 .01];
y0=[0 0];x0=[0 0];
n=0:100;
x=sin(.1*n);
y=recur(a,b,n,x,x0,y0);
plot(.1*n,y,'o')
```

The response is plotted below using o's, along with the plot obtained in Part (c):



2.34 (d) continued

The plot shows that the approximations are very close.

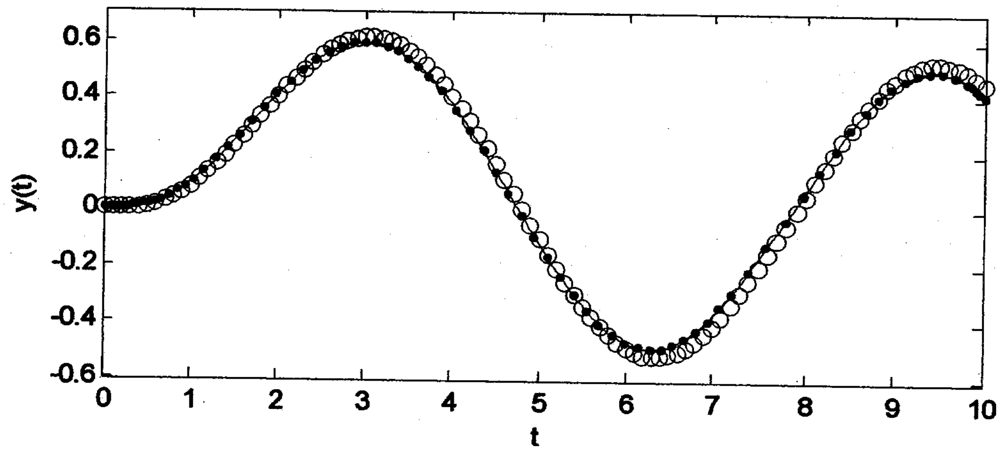
(e) The exact response can be found using the Symbolic Math Toolbox using the command

```
y=dsolve('D2y=-2*Dy-y+sin(t)','Dy(0)=0','y(0)=0')
```

which gives

```
y =

1/2*exp(-t)+1/2*exp(-t)*t-1/2*cos(t)
```

The symbolic solution can be plotted over the interval $0 \le t \le 5$ by using the command `ezplot(y,[0 5])`. The result is plotted below as a solid curve along with the responses found in Parts (c) and (d):



From this plot it is seen that the approximation generated in Part (c) is virtually the same as the exact solution while the approximation using recur generated in Part (d) is off a little.

2.35 (a) Let $y(t) = \theta_E(t)$. Then using the symbolic math command

```
y=dsolve('Dy=(1/Te)*(-y+A*b*heaviside(t-d)-A*b*heaviside(t-d-
c)+A*heaviside(t-d))','y(0)=0')
```

yields the solution

```
y =

exp(-1/Te*t)*(-A*b*heaviside(-d)+A*b*heaviside(-
d)*exp(1/Te*d)+A*b*heaviside(-d-c)-A*b*heaviside(-d-c)*exp(1/Te*(c+d))-
A*heaviside(-d)+A*heaviside(-d)*exp(1/Te*d))+A*b*heaviside(t-d)-
A*b*heaviside(t-d)*exp(-(t-d)/Te)-A*b*heaviside(t-d-c)+A*b*heaviside(t-
d-c)*exp(-(t-d-c)/Te)+A*heaviside(t-d)-A*heaviside(t-d)*exp(-(t-d)/Te)
```

Here `heaviside(t)` is the step function $u(t)$. Now since d and c are positive constants, this expression for the response reduces to

$$y(t) = (b+1)A\left[1-e^{-\frac{t-d}{T_e}}\right]u(t-d) - bA\left[1-e^{-\frac{t-d-c}{T_e}}\right]u(t-d-c)$$

2.35 continued

(b) For $t \geq d+c$,

$$y(t) = (b+1)A\left[1-e^{-\frac{t-d}{T_e}}\right] - bA\left[1-e^{-\frac{t-d-c}{T_e}}\right] = A-(b+1)Ae^{-\frac{t-d}{T_e}} + bAe^{-\frac{t-d-c}{T_e}}$$

Hence, in order to have $y(t) = A$ for $t \geq d+c$, it must be true that

$$-(b+1)Ae^{-\frac{t-d}{T_e}} + bAe^{-\frac{t-d-c}{T_e}} = 0$$

Canceling terms gives

$$-(b+1) + be^{\frac{c}{T_e}} = 0$$

And thus $b\left[e^{\frac{c}{T_e}} - 1\right] = 1$. Solving for b, we have that $b = \dfrac{1}{e^{\frac{c}{T_e}} - 1}$

2.36 As in Problem 2.35, let $y(t) = \theta_E(t)$. Then $y(t) = 0$ for $t \leq 0.2$, and

$$\frac{dy(t)}{dt} = -10y(t) + 10(b+1),\ 0.2 < t < 0.3$$

$$\frac{dy(t)}{dt} = -10y(t) + 10,\ t \geq 0.3$$

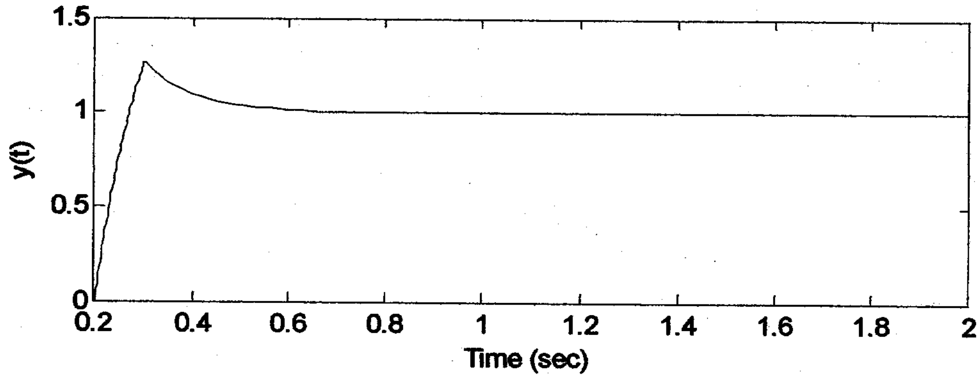
(i) The computation of the response when $b = 1$ requires the function files `de5_func` and `de6_func` which contain the commands:

```
function dy = de5_func(t,y);
b=1;
dy = -10*y+10*(b+1);

function dy = de6_func(t,y);
dy = -10*y+10;
```

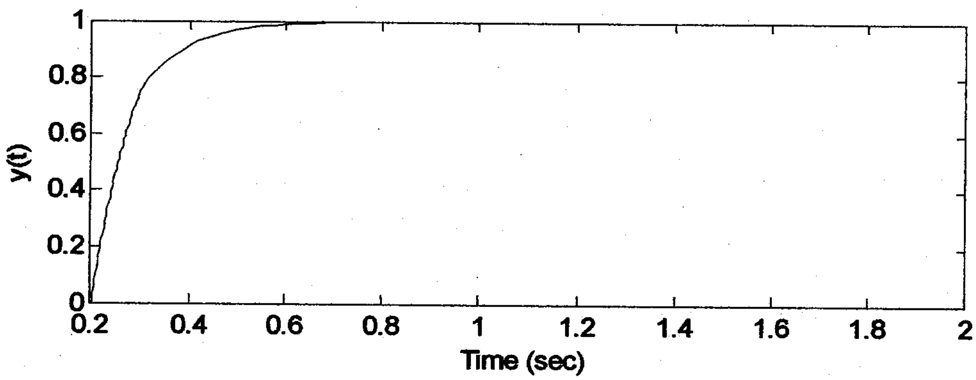
Using these files, the MATLAB program for computing the response when $b = 1$ is:

```
tspan=[.2 .3];
y0=0;
[t,y]=ode45(@de5_func,tspan,y0);
plot(t,y)
xlabel('Time (sec)')
ylabel('y(t)')
hold on
tspan=[.3 2];
y0=1.264;
[t,y]=ode45(@de6_func,tspan,y0);
plot(t,y)
hold off
```

This yields the following plot:

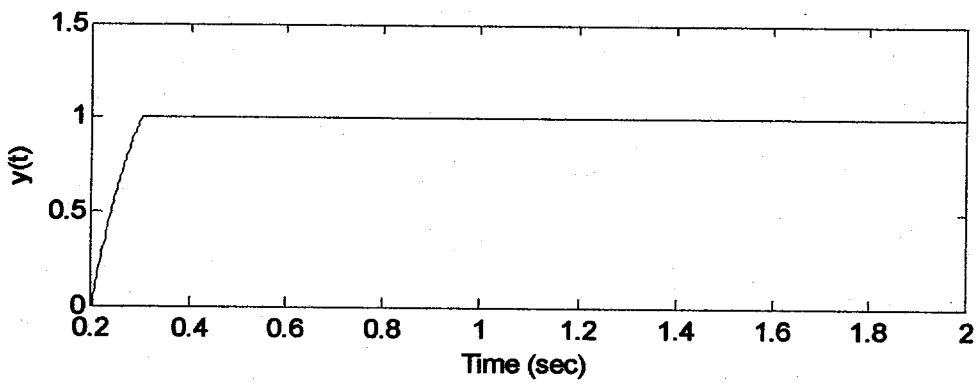


(ii) Via similar procedure, we find that the plot when $b = 0.2$ is:



(iii) $b = \dfrac{1}{e^{\frac{c}{T_e}} - 1} = \dfrac{1}{e-1} = 0.582$. Again carrying out a procedure similar to that given in the solution to Part (a),

we have the following plot when b = 0.582:



From the above plots, we see that the eye does lock onto the target for all three values of b. However, when $b = 1$, there is overshoot, and when $b = 0.2$, the response is somewhat slow in reaching the steady-state value. The best result is obtained when $b = 0.582$, which is the value of b computed analytically in Part (b) of Problem 2.35.