Ch. 2 Time-Domain Models of Systems

Kamen and Heck

And

Harman

2.1 Input/Output Representation of Discrete-Time Systems

- N-Point Moving Average

  y[n] = (1/N) {x[n] + x[n-1] + …+x[n-N +1] }   (2.1)

- Generalization (linear, time-invariant, causal)

$$y[n] = \sum_{i=0}^{N-1} w_i x[n-i]$$

This is a weighted Moving average Filter – page 45 K&H

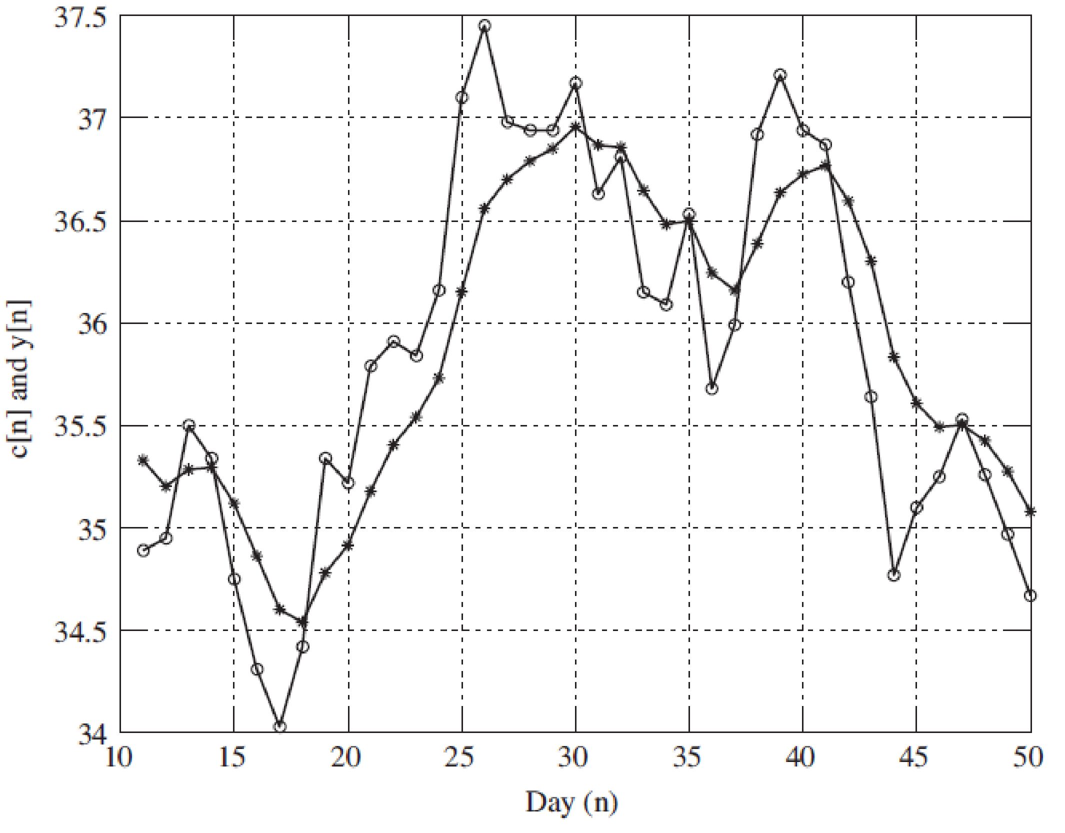
2.1.1 Exponentially Weighted Moving Average

- Let a = (1-b)/(1-b^n)

$$y[n] = \sum_{i=0}^{N-1} a(b^i x[n-i])$$

SPECIAL CASE $a = \frac{1}{N}, \quad b = 1$ Moving Average

The moving average smooths and delays the data P47 K&H



2.1.2 General Class of Systems

- Upper index (N-1) can be replaced by n.
- Unit impulse response of the system can be obtained by letting x[n] = δ[n]. P48 K&H

$$h[n] = \sum_{i=0}^{n} w_i \delta[n-i], n \geq 0$$

- $h[n] = w_n, n \geq 0$ Unit Impulse response

Is equal to the weights (FIR FILTER)

UNIT PULSE (p 15) $\delta = 1 \ if \ n = i$

Convolution Equation

- The input/output representation can be rewritten with the weighting function replaced with the input response function values, h[i]. The result is called convolution.

$$y[n] = h[n] * x[n] = \sum_{i=0}^{n} h[i]x[n-i], n \geq 0$$

THIS IS THE BIG RESULT ! FROM IMPULSE RESPONSE WE FIND THE GENERAL RESPONSE! PAGE 49

WE DEAL WITH SUMS FOR THE DISCRETE CASES.

Geometric Series A *geometric series* is a series with each term after the first being a fixed multiple of the preceding term. The multiplier is a real number r, called the *ratio*, so that $a_{n+1} = ra_n$. If the sum is taken from $n = 0$, the geometric series is represented as

$$\sum_{n=0}^{\infty} ar^n = a + ar + \cdots \quad (a \neq 0). \tag{6.9}$$

From Harman et. al
Advanced Engineering Mathematics With MATLAB

We will show that the series converges to the sum

$$\sum_{n=0}^{\infty} ar^n = \frac{a}{1-r} \tag{6.10}$$

if $-1 < r < 1$ but diverges if $|r| > 1$. Furthermore, if the infinite series with nth partial sum S_n converges and has sum S, then for every number $\epsilon > 0$, there exists a number N such that

$$|S - S_n| < \epsilon$$

for every $n > N$. Thus, we can approximate the sum as closely as desired by taking more terms in the series if necessary.

The nth partial sum for the geometric series is found by subtracting the terms

$$\begin{aligned} S_n - rS_n &= a + ar + \cdots + ar^n - (ar + ar^2 + \cdots + ar^{n+1}) \\ &= a - ar^{n+1}, \end{aligned}$$

so that $S_n - rS_n = a(1 - r^{n+1})$. Thus, solving for S_n leads to the result

$$S_n = \frac{a(1 - r^{n+1})}{1-r}$$

for the sum of the first $n+1$ terms. Taking the limit as n goes to infinity with $|r| < 1$ shows that the sum of the series is $a/(1-r)$, as shown in Equation 6.10.

Consider the fraction $1/3$ represented as the series

$$\frac{1}{3} = \frac{3}{10} + \frac{3}{100} + \frac{3}{1000} + \cdots.$$

Substituting $a = 3/10$ and $r = 1/10$ in Equation 6.9 leads to the result

$$\sum_{n=0}^{\infty} \frac{3}{10}\left(\frac{1}{10}\right)^n = \frac{3}{10}\frac{1}{1-1/10} = \frac{1}{3}.$$

Considering the partial sums of this series, we are confident that taking more terms in a truncated series leads to a better approximation for $1/3$.

MATLAB

- The matlab function conv can be used to compute discrete convolution.
- Examples on page 52 and 53 illustrate the results.
- NEXT SLIDE IS EXAMPLE 2.4 PAGE 52

EXAMPLE 2.4 p52

	n=-2	n=-1	n=0	n=1	n=2	n=3			
V[n] =	-1	5	3	-2	1	0			
x[n] =	0	+1	2	3	4	5			
	-1	5	3	-2	1				
		-2	10	6	-4	2			
			-3	15	9	-6	3		
				-4	20	12	-8	4	
					-5	25	15	-10	5
y[n]	-1	3	10	15	21	33	10	-6	5
n=	-3	-2	-1	0	1	2	3	4	5

Check Length y = 5+5-1=9 ✓ Start=-3 End=5 ✓

```
% Convolution example 2.4 K&H P52
x=[ 1 2 3 4 5]  % Lx=5 Start n=-1, End n=3
v=[-1 5 3 -2 1] % Lv=5 Start n=-2, End n=2
% Expect Lconv= Lx+Lv-1=9,
     Start n=-1-2=-3, End n= 2+3=5


y=conv(x,v)
%y =-1   3   10     15     21    33    10   -6     5
%n = -3  -2  -1      0     1      2     3    4     5
```

Difference Equation Models

- In some applications, a causal linear time-invariant discrete-time system is given by an input/output difference equation instead of an input/output convolution model.
- First order linear difference equation for loan payment K&H p56

$$y[n] - ay[n-1] = -x[n]$$

See K&H P56-57 and MATLAB Figure 2.7 Page 57

2.3.1 Nth-Order Input/Output Difference Equations

- Nth Order Equation (2.25 P57)

$$y[n]+\sum_{i=1}^{N} a_i y[n-i]=\sum_{i=0}^{M} b_i x[n-i]$$

HERE THERE ARE N COEFFICIENTS FOR THE EQUATION

AND M+1 COEFFICIENTS FOR THE INPUT FUNCTION

$$y[n]+\sum_{i=1}^{N}a_i y[n-i]=\sum_{i=0}^{M}b_i x[n-i]$$

- RECURSIVE DIGITAL FILTER
- NEXT VALUE IS COMBINATION OF THE PAST N VALUES
- THUS, OUTPUT NOW DEPENDS ON PAST OUTPUTS
- THE EQUATION REQUIRES N INITIAL VALUES

Compound Interest Problem

Suppose money is deposited in a savings account that pays interest at the rate p percent, paid at regular intervals of time. For example, let \$1000.00 be deposited with the interest rate 6% a year and the interest compounded every year. The value when the first interest payment is made after a year will be \$1000 + 0.06 × \$1000 = \$1060.

In a more general case, consider the compound interest equation

$$y(nT) = y(nT-T) + \frac{p}{100}\, y(nT-T) + x(nT) = \left(1 + \frac{p}{100}\right) y(nT-T) + x(nT), \quad (1)$$

where $y(nT)$ represents the amount of money in an account at time $t = nT$, $y(nT-T)$ is the money in the account at the time of the previous computation, p is the percent interest paid in the interval of time T, and $x(nT)$ is the amount of money deposited or withdrawn at $t = nT$.

Designating the discrete values as $y(n)$, we define

$$y(n) = y(nT), \quad n = 0, 1, 2, \ldots$$

to form a sequence of discrete values of the bank balance. If there are no extra deposits or withdrawals, $x(nT) = 0$ and Equation 1 can be written as

$$y(n) = ay(n-1), \tag{2}$$

where $a = (1 + p/100)$.

As an numerical example, assume that $x(nT)$ is zero and $p = 6\%$ a year with an initial deposit of $y(0)$ dollars. Applying Equation 2 repeatedly yields the equations

$$\begin{aligned} y(1) &= (1.06)\ y(0) \\ y(2) &= (1.06)\ y(1) = (1.06)^2\ y(0) \\ &\vdots \\ y(n) &= (1.06)\ y(n-1) = (1.06)^n\ y(0). \end{aligned}$$

It appears that the solution to the equation $y(n) = a\ y(n-1)$ is

$$y(n) = a^n\ y(0). \tag{3}$$

```
% EX10_1.M MATLAB solution of the
%   compound interest equation
%    y(nT)=y(nT-T) + (p/100)*y(nT-T)
%     for p = 6 percent and initial deposit y0 = $ 1000
%   y(n) represents the balance after the nth year
clear
format bank    % Show results as currency
a=1.06;        % Calculation for 5 years at 6% interest
y0 = 1000      % Initial deposit
for n=1:5
 y(n) = a^(n)*y0
end

% y = 1060.00   1123.60  1191.02  1262.48  1338.23
% year  1          2        3        4        5
```

Examples

- Example 2.6   Second Order System p60
  - System can be solved recursively.
  - Look at this example carefully and trace the steps in the MATLAB program.
  - Remember the MATLAB index y(1), y(2), …

    and the math index   y[-m], y[-m+1] …..

Ex. 2.6 Pg60 He solves the 2nd order difference equation $y[n]-1.5y[n-1]+y[n-2]=2u[n-2]$, $y(-2)=2$, $y(-1)=1$ using Routine recur. N=length a= 2, M=length b -1 =3 -1=2

```
function y = recur(a,b,n,x,x0,y0);
N = length(a);     % Number of Coefficients in y
M = length(b)-1;   % Number of Coefficients -1 in x
if length(y0) ~= N,
  error('Lengths of a and y0 must match')
end
if length(x0) ~= M,
  error('Length of x0 must match length of b-1')
end
y = [y0 zeros(1,length(n))];  % Initial Values
x = [x0 x]
a1 = a(length(a):-1:1)     % reverses the elements in a
b1 = b(length(b):-1:1)
for i=N+1:N+length(n),
  y(i) = -a1*y(i-N:i-1)' + b1*x(i-N:i-N+M)'; % Transpose
end
y = y(N+1:N+length(n))
```

% Output y0, y1, yn (MATH) but

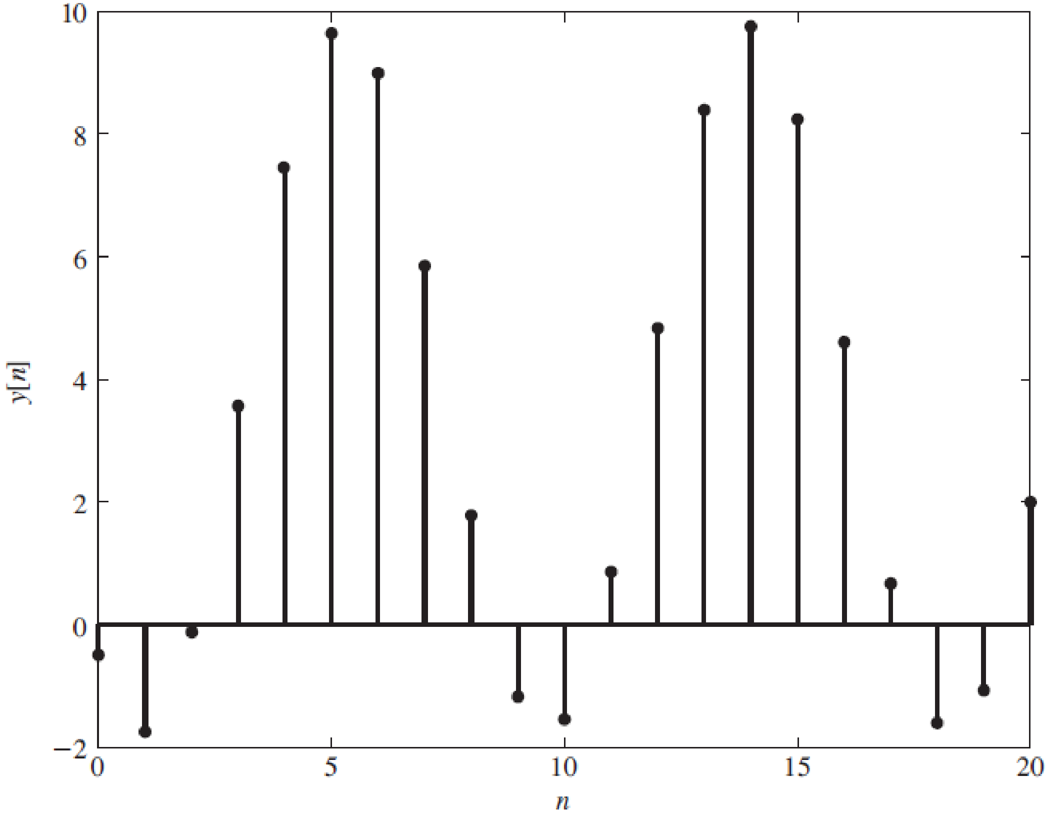
% y(N+1) y(N+2) MATLAB, so y(-2) = y(1)mat

Note Index in loop starts at N=3 for MATLAB

```
% Figure 2.9  Page 62
a = [-1.5 1]; b = [0 0 2]; %[a1 a2]; [b0 b1 b2]
y0 = [2 1]; x0 = [0 0];    % Initial values
n = 0:20;                  % 21 points
x = ones(1,length(n));     % Unit step Input
y = recur(a,b,n,x,x0,y0);
stem(n,y,'filled')         % Plot it
xlabel('n')
ylabel('y[n]')
```

Figure 2.9 Pg62 y[n]-1.5y[n-1]+y[n-2]=2u[n-2]

Figure 2.9 Pg62 y[n]-1.5y[n-1]+y[n-2]=2u[n-2]



Note the two necessary Initial conditions y(-2) and y(-1)

2.4 Differential Equation Models

- Example 2.8 Series RC Circuit P65
- We will do this in detail in later slides

- Other Models:
- Example 2.9 Mass-Spring-Damper System P67
- Example 2.10 Motor with Load P69

First Order Differential Equations Consider the first-order, linear differential equation

$$\frac{dy(t)}{dt} + p(t)y(t) = f(t), \tag{1}$$

which we write as $\dot{y} + p(t)y = f(t)$. Assuming that $p(t)$ and $f(t)$ are continuous in some common interval, the equation can be solved by multiplying each term by an *integrating factor* in the form $e^{\int p(t)dt}$ to yield

$$\dot{y}e^{\int p(t)\,dt} + p(t)ye^{\int p(t)\,dt} = f(t)e^{\int p(t)\,dt}.$$

Notice that the left side of this equation is the derivative of the product $ye^{\int p(t)\,dt}$. Thus,

$$\frac{d}{dt}\left[ye^{\int p(t)\,dt}\right] = f(t)e^{\int p(t)\,dt}$$

can be integrated and solved for $y(t)$, with the result

$$y(t) = e^{-\int p(t)\,dt}\int f(t)e^{\int p(t)\,dt}dt + ce^{\int -p(t)\,dt}, \tag{2}$$

where c is the constant of integration. This expression is the general solution to Equation 1.

The general solution of Equation 2 contains two terms. The first term describes the effect of the function $f(t)$, which is often called the *forcing function* when used in problems that model physical systems. In the study of linear systems in engineering, the function $f(t)$ is also called the *input* or *input function*, and the solution $y(t)$ is termed the *output*. The differential equation describes how the system reacts to the effects of the input function.

The second term in Equation 2 contains an arbitrary constant c, which is determined by demanding that the solution meet an *initial condition*. This specifies the value of y at some specific value of t, say, t_0. Mathematically, we write the initial condition as $y(t_0) = y_0$. The differential equation and the initial condition taken together is called an *initial value problem*.

Constant Coefficients In case the function $p(t) = a$ where a is a scalar, Equation 1 becomes

$$\frac{dy(t)}{dt} + ay(t) = f(t). \tag{3}$$

Assuming that the equation is defined on the interval $t \geq 0$, the integrating factor for the equation is

$$e^{\int_0^t a d\tau} = e^{at},$$

where the variable of integration has been changed to τ to emphasize that the integral is a function of t, the upper limit of integration, not the "dummy" variable τ.

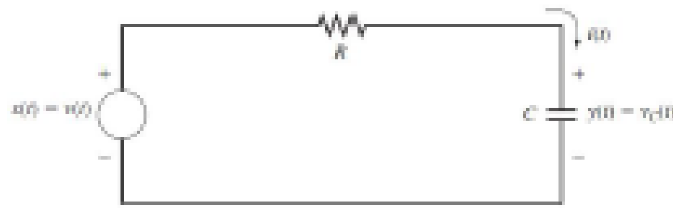
The complete solution then takes the form

$$y(t) = \int_0^t f(\tau)e^{-a(t-\tau)}d\tau + ce^{-at} \tag{4}$$

where the integral term represents *convolution*. See Kamen and Heck result for the RC circuit with a pulse input Example 2.14 Pg 77. This can also be written as

$$y(t) = e^{-at}\int_0^t f(\tau)e^{a\tau}d\tau + ce^{-at}. \tag{5}$$

For the RC circuit



the differential equation is written using Kirchhoff's laws and the basic physics of a capacitor

$$i(t) = C\frac{dv(t)}{dt}.$$

In the figure

$$Ri(t) + y(t) - x(t) = 0$$

so that

$$\frac{dv(t)}{dt} + \frac{1}{RC}y(t) = \frac{1}{RC}x(t).$$

Using the results above with $a = 1/RC$ and $x(t) = U(t)$ the unit step, yields the step response as

$$y(t) = 1 - e^{\frac{-t}{RC}}$$

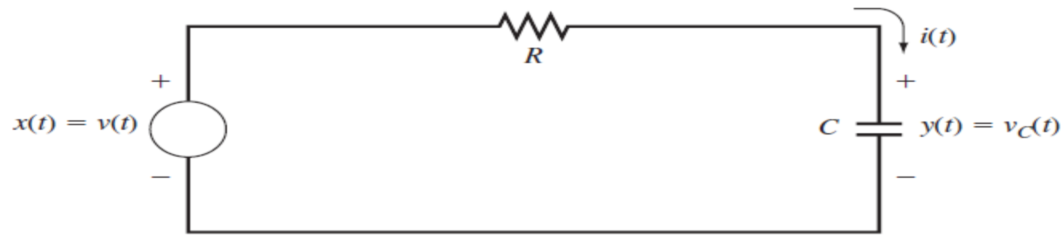
Taking the derivative of the step response yields the impulse response

$$\frac{dv(t)}{dt} = h(t) = \frac{1}{RC}e^{\frac{-t}{RC}}$$

K&H Example 2.11 P72 and Example 2.14 states these results.

2.5 Solution Of Differential Equations

- Example 2.11 Series RC Circuit



- See My slides "First Order Differential Equations" and examples from

    K&H pg 72 (Solution of RC) and pg77 superposition for convolution solution

K&H solve it by recursion, Euler method, and ODE

Example 2.11,2.12 Using Recursion, Euler and MATLAB ODE Solver

Let's cover K&H pages 70 to 75 using the RC circuit

The key to recursion solution is to keep the time between samples T small. What does small mean ? Small compared to the time constant of the system- which is a measure of how fast the system responds to a step input.

Let Ƭ=RC be the time constant of the system, so the step response is

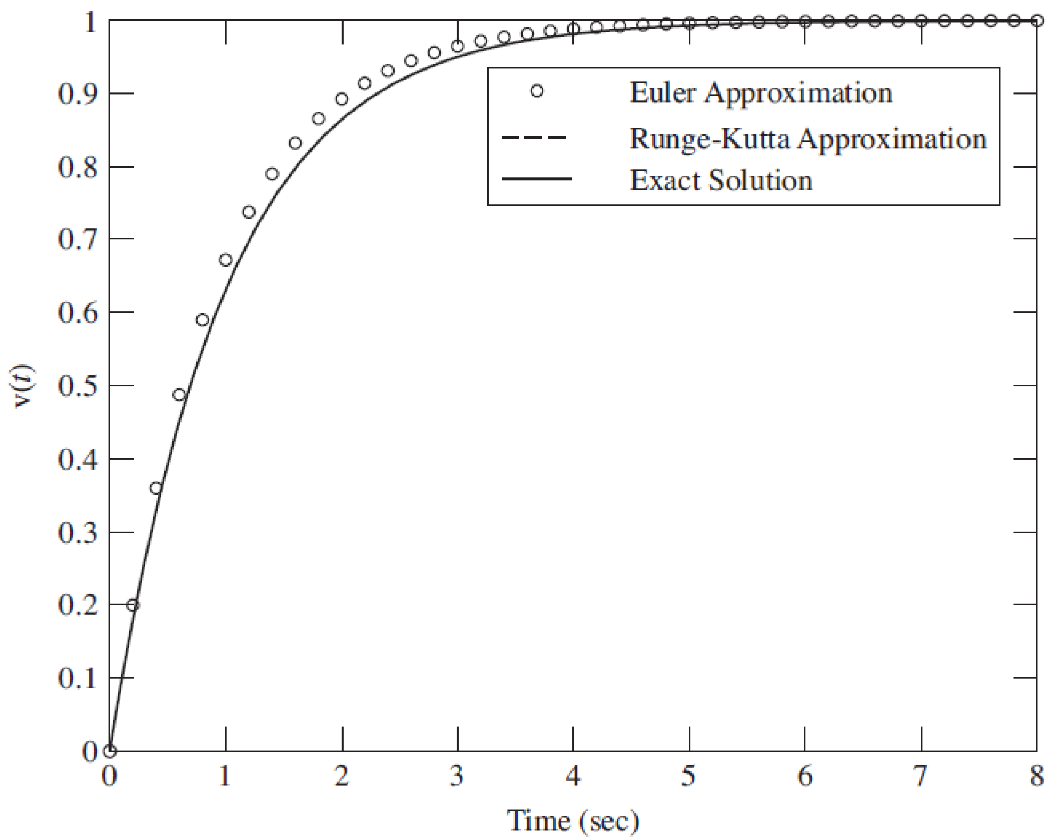
$y(t) = \ 1 - e^{-t/\tau}$ and let a=1/τ in Equation 2.61 (Euler) or in the

Taylor Series solution Equation 2.66.

For all of these to be useful, the sampling time T is selected so that

T << 1/a = $\boldsymbol{\tau \ \ seconds}$.

ODE is a sophisticated routine that chooses T by the Runge-Kutta algorithm.



Euler Approximation
Runge-Kutta Approximation
Exact Solution
v(t)
Time (sec)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
1
2
3
4
5
6
7
8

Example 2.13 MATLAB Symbolic MATH Solver dsolve

```
% Example 2.13 K&H Page75
% Symbolically solve the RC circuit
% He uses R=C=1 (silly)  Let RC=0.5 ms
=tau
tau=0.5*10^(-3)
y=dsolve('Dy=(1/tau)*(1-y)', 'y(0)=0')
%
% y =1 - exp(-t/tau)
```

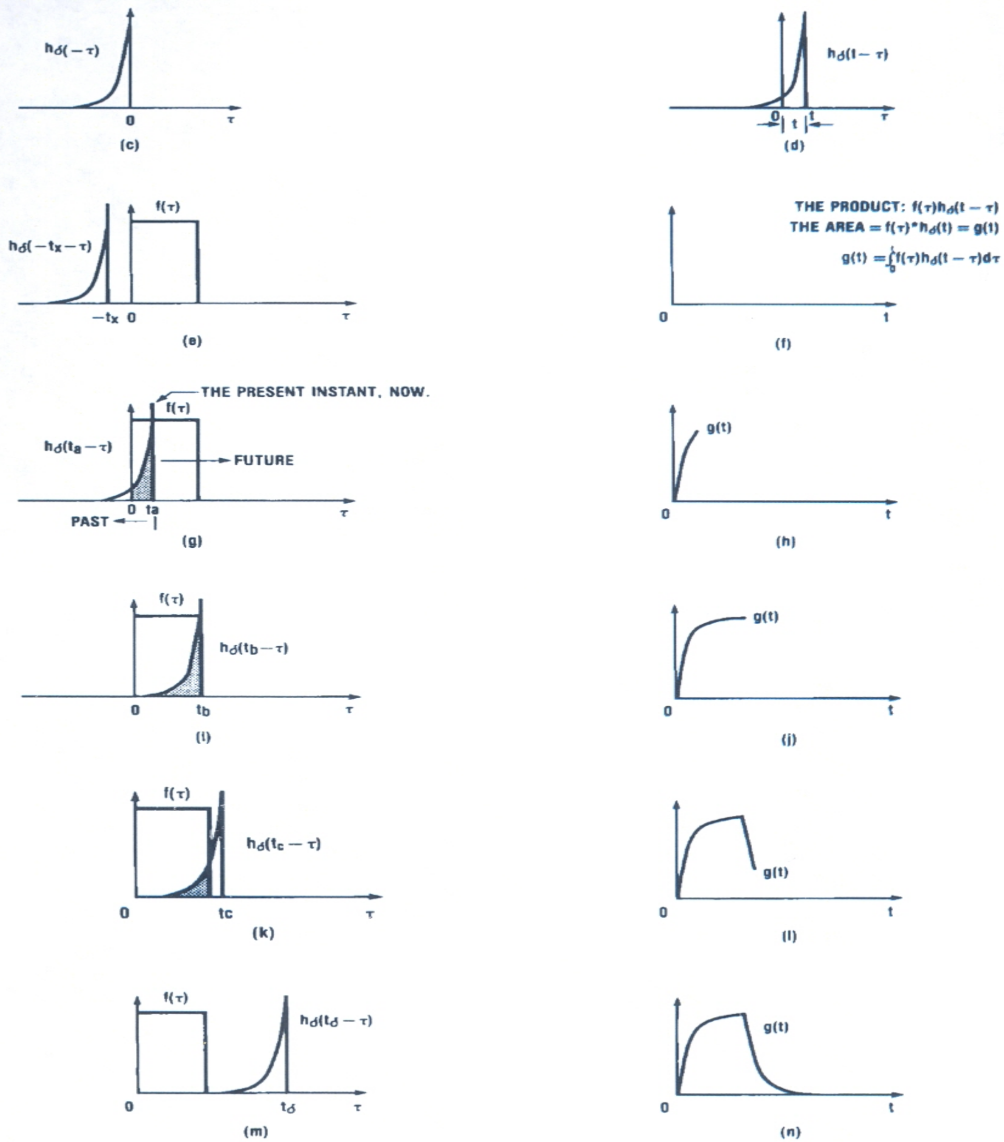
```
Do >>help dsolve for more information
```

See also K&H Example of 2nd order solution – Pg 75

2.6 Convolution Representation of Continuous-Time Systems

- Example 2.14 RC Circuit Page 77
- See Following Slides

- 2.6.1 Graphical Approach to Convolution



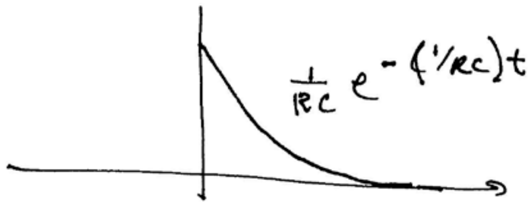
TL/H/5621–7

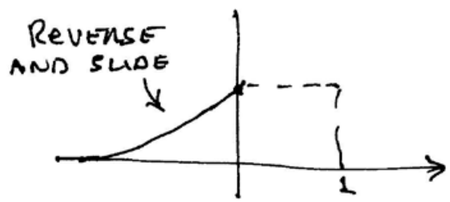
FIGURE 4. cont'd c) $h_\delta(-\tau)$: $h_\delta(\tau)$ folded about the ordinate
d) $h_\delta(t-\tau)$: $h_\delta(r)$ folded and shifted
e) through n) the output response g(t) of the network whose impulse response $h_\delta(\tau)$ is excited by a function $f(\tau)$.
Or the convolution, $f(\tau)*h_\delta(t)$, of f(t) with $h_\delta(t)$.

AN-237
NAT. SEMI

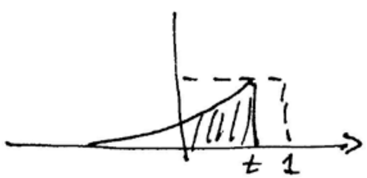
3

K+H p 77-78
Ex 2.14



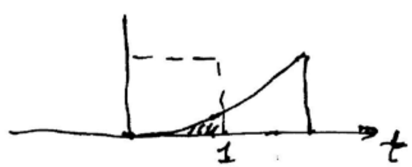


AREA=0 (RC=1)



$$\int_0^t 1 \cdot h(t-\lambda)\, d\lambda = \int e^{-(t-\lambda)} d\lambda$$

$$= e^{-t} \int_0^t e^{\lambda} d\lambda$$

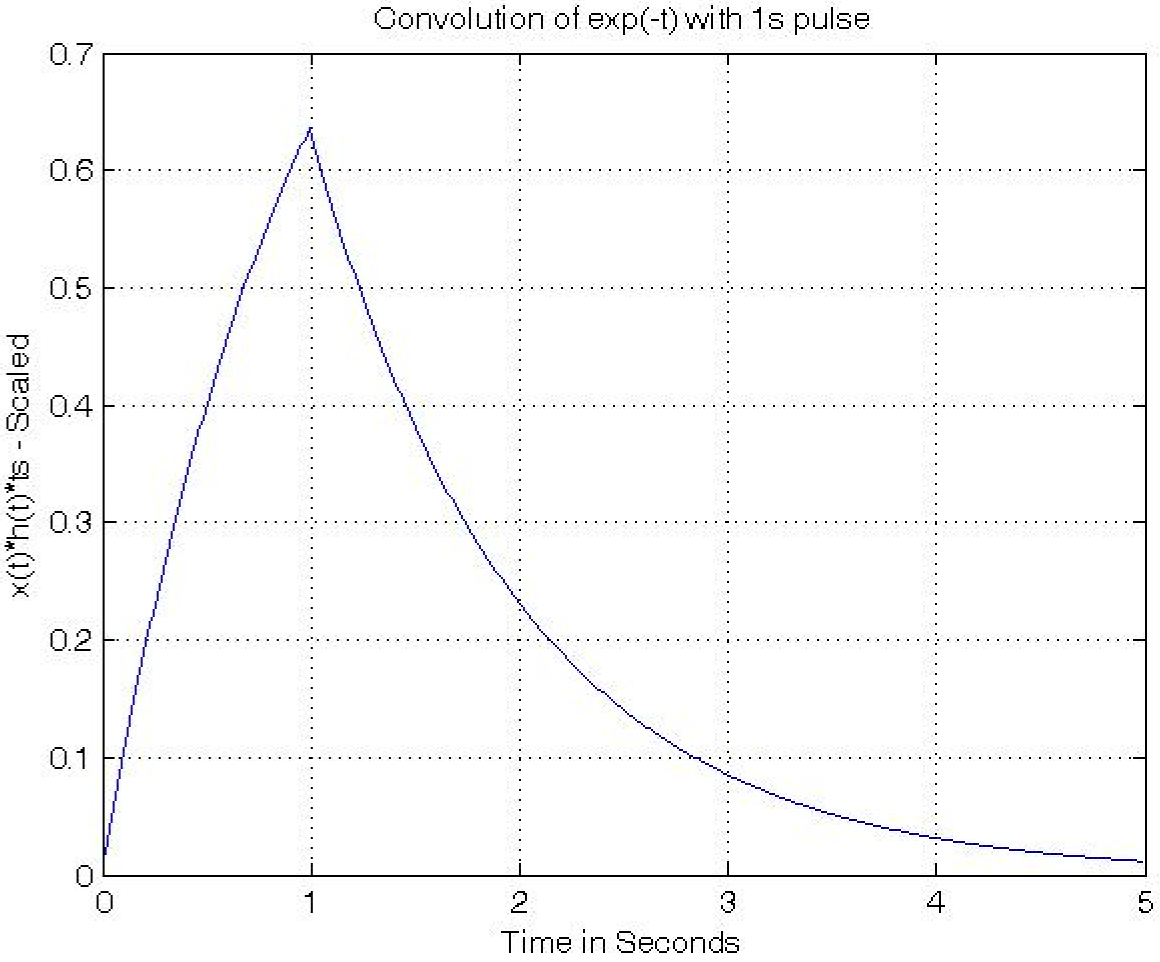


$$\int_0^1 1 \cdot e^{-(t-\lambda)} d\lambda$$

$$= e^{-t}(e^1 - 1) \quad t \geq 1$$

```
% Convolution h(t)= 1/RC)exp(-t/RC) and X(t)=p(t)0 <=t<=1.sec
% Let RC = 1s and assume h(t)=0 after 5s. K&H P77
% Note scaling of the convolution to approximate the integral
n=(0:499);
ts=0.01                     % Sample every 0.01 ms
h=1*exp(-1*n*ts);           % Impulse response
time=(0:499)*ts;
figure(1), plot(time,h)     % Plot impulse response
title('Impulse response 1*exp(-t)')
xlabel('Time in Seconds'),ylabel('h(t)')
%  Pulse One second long  100 x ts
x=zeros(size(n));
x(1:100)=ones(1,100);  % Create the step function
figure(2),plot(time,x)
title('1s pulse input')
xlabel('Time in Seconds'),ylabel('x(t)')
y=conv(x,h)*ts;  % Scale by ts to approximate continuous case
% Length is Lx+Lh -1
figure(3),plot(time,y(1:500))
title('Convolution of exp(-t) with 1s pulse')
xlabel('Time in Seconds'),ylabel('x(t)*h(t)*ts-Scaled'),grid
```

Figure 2.18



The time constant of the circuit is 1 second. The circuit responds 98% in about 5 τ.

Table 7-4: Comparison of convolution properties for continuous-time and discrete-time signals.

Property	Continuous Time	Discrete Time
Definition	$y(t) = h(t) * x(t) = \int_{-\infty}^{\infty} h(\tau)\, x(t-\tau)\, d\tau$	$y[n] = h[n] * x[n] = \sum_{i=-\infty}^{\infty} h[i]\, x[n-i]$
1. Commutative	$x(t) * h(t) = h(t) * x(t)$	$x[n] * h[n] = h[n] * x[n]$
2. Associative	$[g(t) * h(t)] * x(t) = g(t) * [h(t) * x(t)]$	$[g[n] * h[n]] * x[n] = g[n] * [h[n] * x[n]]$
3. Distributive	$x(t) * [h_1(t) + \cdots + h_N(t)] = x(t) * h_1(t) + \cdots + x(t) * h_N(t)$	$x[n] * [h_1[n] + \cdots + h_N[n]] = x[n] * h_1[n] + \cdots + x[n] * h_N[n]$
4. Causal $*$ Causal = Causal	$y(t) = u(t) \int_0^t h(\tau)\, x(t-\tau)\, d\tau$	$y[n] = u[n] \sum_{i=0}^{n} h[i]\, x[n-i]$
5. Time-shift	$h(t-T_1) * x(t-T_2) = y(t-T_1-T_2)$	$h[n-a] * x[n-b] = y[n-a-b]$
6. Convolution with Impulse	$x(t) * \delta(t-T) = x(t-T)$	$x[n] * \delta[n-a] = x[n-a]$
7. Width	width $y(t)$ = width $x(t)$ + width $h(t)$	width $y[n]$ = width $x[n]$ + width $h[n]$ − 1
8. Area	area of $y(t)$ = area of $x(t)$ × area of $h(t)$	$\sum_{n=-\infty}^{\infty} y[n] = \left(\sum_{n=-\infty}^{\infty} h[n]\right)\left(\sum_{n=-\infty}^{\infty} x[n]\right)$
9. Convolution with step (INTEGRATION OR SUM)	$y(t) = x(t) * u(t) = \int_{-\infty}^{t} x(\tau)\, d\tau$	$x[n] * u[n] = \sum_{i=-\infty}^{n} x[i]$

$x[n] = [2,3,4] * h[n] = [5,6,7]$

Since $h[i] = 0$ for all values of i except $i = 0$, 1, and 2, it follows that $h[n-i] = 0$ for all values of i except for $i = n$, $n-1$, and $n-2$. With this constraint in mind, we can apply Eq. (7.52) at discrete values of n, starting at $n = 0$:

$$y[0] = \sum_{i=0}^{0} x[i]\, h[0-i] = x[0]\, h[0] = 2 \times 5 = 10,$$

$$y[1] = \sum_{i=0}^{1} x[i]\, h[1-i] = x[0]\, h[1] + x[1]\, h[0] = 2 \times 6 + 3 \times 5 = 27,$$

$$y[2] = \sum_{i=0}^{2} x[i]\, h[2-i] = x[0]\, h[2] + x[1]\, h[1] + x[2]\, h[0] = 2 \times 7 + 3 \times 6 + 4 \times 5 = 52,$$

$$y[3] = \sum_{i=1}^{2} x[i]\, h[3-i] = x[1]\, h[2] + x[2]\, h[1] = 3 \times 7 + 4 \times 6 = 45,$$

$$y[4] = \sum_{i=2}^{2} x[i]\, h[4-i] = x[2]\, h[2] = 4 \times 7 = 28,$$

$y[n] = 0$, otherwise.

$\sum_n y(n) = 162 = 9 \times 18$

Hence,

$$y[n] = \{\underline{10}, 27, 52, 45, 28\}.$$

$L = 3 + 3 - 1 = 5$ ✓

7-5.2 Discrete-Time Convolution Properties

With one notable difference, the properties of the discrete-time convolution are the same as those for continuous time. If (t) is replaced with $[n]$ and integrals are replaced with sums, the convolution properties derived in Chapter 2 lead to those listed in Table 7-4.