ECE 341 - Homework #7

Uniform and Exponential Distributions. Summer 2024

Uniform Distributions

Let

- **a** be a sample from A, a uniform distribution over the range of (1, 3)
- **b** be a sample from B, a uniform distribution over the range of (1, 4)

1) Determine the pdf for **y = a + b** using moment generating funcitons (i.e. LaPlace transforms)

$$A = \frac{1}{2s}(e^{-s} - e^{-3s})$$

$$B = \frac{1}{3s}(e^{-s} - e^{-4s})$$

$$Y = AB = \left(\frac{1}{2s}(e^{-s} - e^{-3s})\right)\left(\frac{1}{3s}(e^{-s} - e^{-4s})\right)$$

$$Y = \left(\frac{1}{6s^2}\right)(e^{-2s} - e^{-4s} - e^{-5s} + e^{-7s})$$

Taking the inverse LaPlace transform

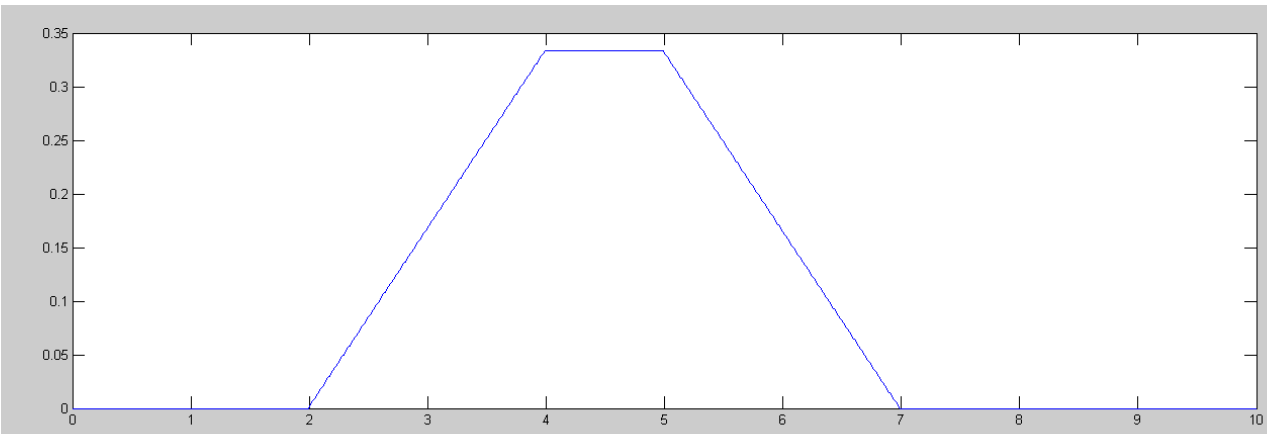
$$y(x) = \left(\frac{1}{6}\right)((x-2)u(x-2) - (x-4)u(x-4) - (x-5)u(x-5) + (x-7)u(x-7))$$

or, putting it another way

$$y(x) = \begin{cases} 0 & x < 2 \\ \left(\frac{1}{6}\right)(x-2) & 2 < x < 4 \\ \left(\frac{2}{6}\right) & 4 < x < 5 \\ \left(\frac{1}{6}\right)(7-x) & 5 < x < 7 \\ 0 & x > 7 \end{cases}$$

2) Determine the pdf for **a** + **b** using convolution (by hand or Matlab)

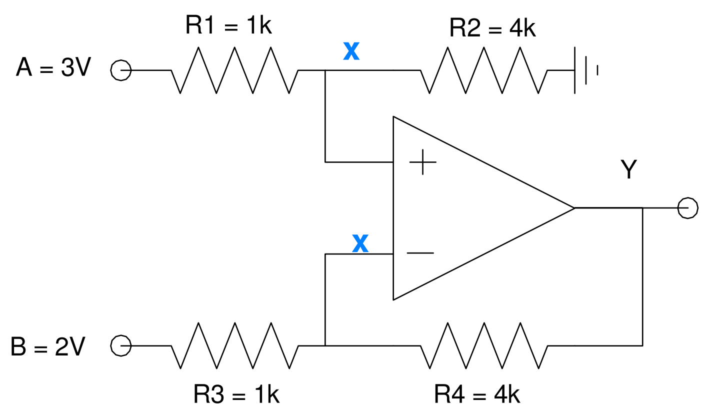
```
>> dx = 0.01;
>> x = [0:dx:10]' + 1e-9;
>> A = 1/2 * (x>1).*(x<3);
>> B = 1/3 * (x>1).*(x<4);
>> y = conv(A,B) * dx;
>> y = y(1:length(x));
>> plot(x,y)
>> xlim([0,10])
>>
```



Same answer as problem #1 but in a graph

3) Assume each resistor has a tolerance of 5% (i.e. a uniform distribution over the range of (0.95, 1.05) of the nominal value. For the following circuit, determine

- The voltage at Y as a funciton of {R1, R2, R3, and R4}, and
- The mean and standard deviation for the voltage at Y using a Monte Carlo simulation.



Using voltage nodes:

Vp = Vm

$$\left(\frac{x-A}{R_1}\right)+\left(\frac{x}{R_2}\right)=0$$

$$\left(\frac{x-B}{R_3}\right)+\left(\frac{x-y}{R_4}\right)=0$$

Solve for y:

$$R_2(x-A)+R_1x=0$$

$$R_4(x-B)+R_3(x-y)=0$$

$$(R_1+R_2)x=R_2A \qquad * \text{ (R3 + R4)}$$

$$(R_3+R_4)x-R_3y=R_4B \qquad * \text{ (R1 + R2)}$$

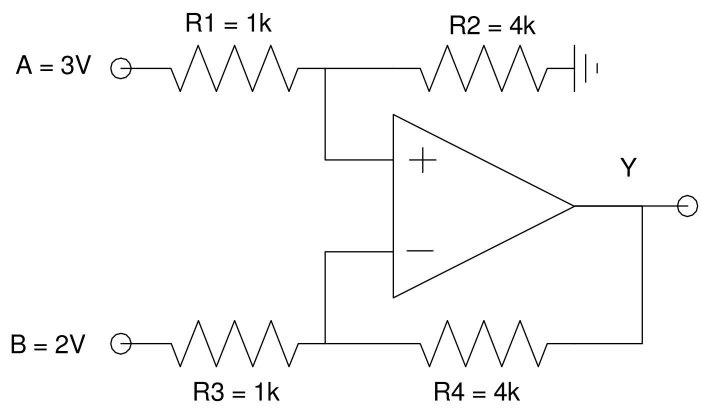
Get x to drop out (gauss elimination)

$$R_3(R_1+R_2)y=R_2(R_3+R_4)A-R_4(R_1+R_2)B$$

$$Y=\left(\frac{R_2(R_3+R_4)}{R_3(R_1+R_2)}\right)A-\left(\frac{R_4(R_1+R_2)}{R_3(R_1+R_2)}\right)B$$

$$Y=\left(\frac{R_2(R_3+R_4)}{R_3(R_1+R_2)}\right)A-\left(\frac{R_4}{R_2}\right)B$$

Using superposition



A = 0:

$$Y = -\left(\frac{R_4}{R_3}\right)B$$

B = 0:

$$Y = \left(\frac{R_3+R_4}{R_3}\right)\left(\frac{R_2}{R_1+R_2}\right)A$$

Total Answer:

$$Y = \left(\frac{R_3+R_4}{R_3}\right)\left(\frac{R_2}{R_1+R_2}\right)A - \left(\frac{R_4}{R_3}\right)B$$

(same as before)

Finding Y using a Monte Carlo simulation

Result

```
x =    4.0003

s =    0.1730
```

Matlab Code:

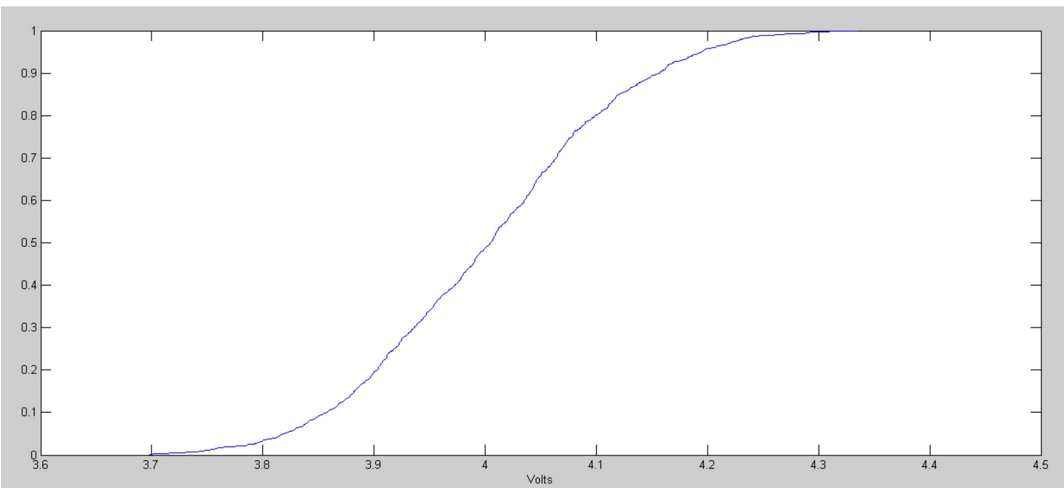
```
n = 1e3;
Y = zeros(n,1)
for i = 1:n
   R1 = (1 + 0.05*(rand*2-1)) * 1000;
   R2 = (1 + 0.05*(rand*2-1)) * 4000;
   R3 = (1 + 0.05*(rand*2-1)) * 1000;
   R4 = (1 + 0.05*(rand*2-1)) * 4000;

   A = 3;
   B = 2;

   Y(i) = (R3+R4)/R3 * R2/(R1+R2) * A - (R4/R3)*B;
end

x = mean(Y)
s = std(Y)

Y = sort(Y);
p = [1:n]' / n;
plot(Y,p)
xlabel('Volts');
```



cdf for y
(will be used when covering Weibull distributions)

Queueing Theory

Assume you are running a fast-food restraunt.

- The time between customers arriving at a restaraunt is an exponential distribution with a mean of 60 seconds.
- The time it takes to serve each customer is an exponential distribution with a mean of 30 seconds.

4) Run a single Monte-Carlo simulation for this restaraunt over the span of one hour.

- Give the formula for each column in you simulation
- What is the longest waiting time for a customer in your simulation?
- What is the largest queue over the span of one hour?

	Cust customer number	t(arr) customer arrival time	t(order) time customer places order	t(serve) time to complete order	Tdone customer receives food	Twait time customer waits in line	Queue lendth of line when customer arrives
1	a	b	c	d	e	f	g
2	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0
10	1	49.9	49.9	10.38	60.29	0	0
11	2	69.32	69.32	1.91	71.23	0	0
12	3	151.47	151.47	6.2	157.67	0	0
13	4	174.66	174.66	26.44	201.1	0	0
14	5	220.35	220.35	8.42	228.77	0	0
15	6	222.36	228.77	6.05	234.82	6.41	1
16	7	244.1	244.1	2.83	246.93	0	0
17	8	274.91	274.91	37.17	312.08	0	0
18	9	286.68	312.08	0.41	312.49	25.4	1
19	10	328.44	328.44	42.03	370.47	0	0
20	11	334.24	370.47	3.3	373.77	36.24	1
21	12	347.08	373.77	1.27	375.05	26.69	2
22	13	391.34	391.34	19.15	410.49	0	0
:	:	:	:	:	:	:	:

```
b(10) = b(9) - 60*log(1-rand())
c(10) = max(b10,e9)
d(10) = -60*log(1-rand())
e(10) = c10 + d10
f(10) = c10 - b10
h(10) = =1*(b10<e9)+1*(b10<e8)+1*(b10<e7)+1*(b10<e6)+1*(b10<e5)+1*(b10<e4)
```

Longest Wait = 36.24 seconds, Longest Queue = 2