EE3032, Dr. Durant, Winter 2019-20, Homework 3

Due Wednesday of week 4 in class

1.33(d) Determine if each of the following signals is a power signal, an energy signal, or neither.

$x_4(t) = e^{-2t}u(t)$

Let's try the energy integral. The unit step zeros the signal before t=0. We can ask MATLAB to do the symbolic integration:

```
syms t
E = int(exp(-2*t).^2,0,inf)
```

E =

$$\frac{1}{4}$$

MATLAB shows us that the energy is 1/4, which is finite, so this is an ***energy signal***.

1.35(b) Determine if each of the following signals is a power signal, an energy signal, or neither.

$x_2(t) = 2\sin(4t)\cos(4t)$

```
x2 = simplify(2*sin(4*t)*cos(4*t))
```

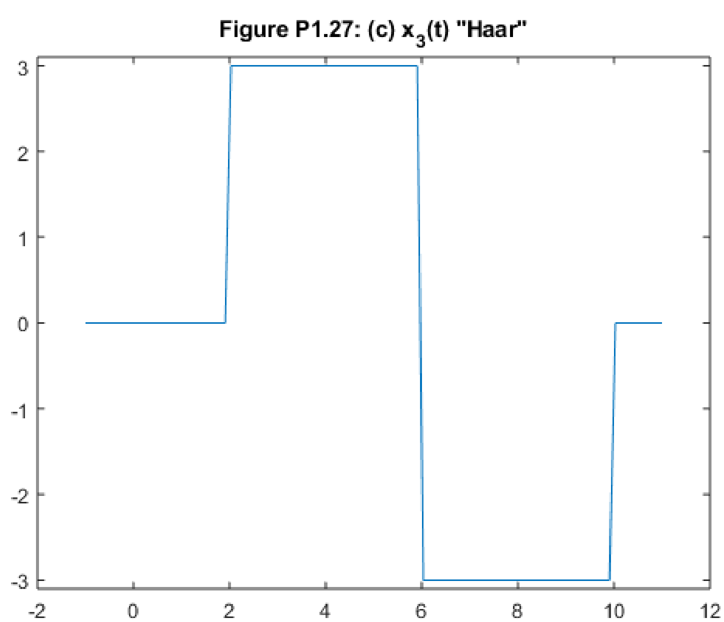
x2 = $\sin(8\,t)$

Trig. identities confirm that this is just a sinusoid, which we know to have infinite energy but finite power (specifically half the square of the amplitude), so this is a ***power signal***.

1.37(c) Use the properties of Problem 1.36 to compute the energy of the three signals in Fig. P1.27.

First, let's replicate the plot.

```
t = linspace(-1,11);
x3 = zeros(size(t));
x3(t>=2 & t<= 6) =  3;
x3(t>=6 & t<=10) = -3;
figure, plot(t,x3), ylim([-3.1 3.1])
title('Figure P1.27: (c) x_3(t) "Haar"')
```



We have 2 rectangular pulses of +/- 3, each with a duration of 4.

1. For energy, only signal magnitude (not sign/phase) matters, we can treat this as a single pulse of amplitude 3 for a duration of 8.
2. Property (a) tells us that the time shift can be ignored.

3. Property (b) tells we that we can factor out amplitude scaling, so the total energy will be 3**2==9 times the energy of a pulse of amplitude 1 for a duration of 8.
4. Property (c) tells us that we can treat the duration of 8 as a duration of 1 followed by a time-saling with a = 1/8 (to slow down the signal). Thus the energy is 9 * 8 (=72) * the ernergy of a pulse of 1 amplitude for duration 1. That pulse has energy 1.
5. Thus, the energy is 72.

Let's check it in MATLAB:

```
dt = diff(t(1:2)); % time step used in t above
E = sum(dt*abs(x3).^2) % energy integral, Riemann sum
```

```
E = 72.0000
```

The result of 72 is consistent with step 5 above.

1.39(a) Compute the average power of the following signals.

$x_1(t) = e^{jat}$ for real-valued a

The conjugate form of power yields a huge amount of simplification...

$$P_1 = \frac{1}{T_0}\int_{T_0} x_1(t)x_1^*(t)dt = \frac{1}{T_0}\int_{T_0} 1dt = 1$$

So, the power is just 1, which is amplitude squared. Notice that this is double the power of a sinusoid. That is because the complex exponential contains both the cosine and j times the sine. That j makes the sine an "orthogonal" signal, which means we can compute its power independently and add it, just as we do for sinusoids of differing frequencies.

2.1(a,g) For each of the following systems, specify whether or not the system is: (i) linear and/or (ii) time-invariant.

(a) $y(t) = 3x(t) + 1$

(i) The system is ***not*** linear. One way to show this is that the scalability property does not apply to this system...

$3cx(t) + 1 \neq c(3x(t) + 1)$

(ii) The system ***is*** time-invariant. If we apply delay first, that is x(t) is delayed by T, we can see the output must be 3x(t-T)+1. If we apply delay second, we take the system definition and replace t with T-T, also yieding 3x(t-T)+1. Since the order of delay and applying the system doesn't matter, the system is TI.

(g) $y(t) = \int_{t-1}^{t+1} x(\tau)d\tau$

(i) This system ***is*** linear. Scalability applies; if the integrand (which is the input) is replaced by a version scaled by c, that constant c can be pulled out of integration. That is the definition of scalability. Similarly, the integration operation has the additivity property.

(ii) The system ***is*** TI.

Delay first, replace x(.) with x(.-tau): $\int_{t-1}^{t+1} x(\tau - T)d\tau = \int_{t-1-T}^{t+1-T} x(\tau)d\tau$

Delay second, replace t with t-T in expression for y(t): $\int_{t-T-1}^{t-T+1} x(\tau)d\tau$

After a trivial rearrangement, we see that the limits of integration are the same.

So, this system is LTI. Integration is a linear operation, so integration with time-invariant limits is LTI.

2.2(b,g) For each of the following systems, specify whether or not the system is: (i) linear and/or (ii) time-invariant.

(b) $y(t) = tx(t)$

(i) The system ***is*** linear. For example, additivity applies. $t(x_1(t) + x_2(t)) = tx_1(t) + tx_2(t)$

That is, applying the system to the sum gives the same expression as summing the result of applying the system to each input component individually.

(ii) The system ***is not*** time-invariant.

Delay first: $tx(t - T)$

Delay second: $y(t - T) = (t - T)x(t - T)$

We can see that the signal multiplier differ by -T, so the order does matter, which means the system is not TI.

(g) $y(t) = \int_{t}^{2t} x(\tau)d\tau$

(i) The system ***is*** linear. The reasoning is exactly the same as in 2.1(g).

(ii) The system ***is not*** TI.

Delay first: $\int_t^{2t} x(\tau - T)d\tau = \int_{t-T}^{2t-T} x(\tau)d\tau$

Delay second: $\int_{t-T}^{2(t-T)} x(\tau)d\tau$

Note that the upper limits of integration differ. In general, whenever the time variable is scaled, or is used in a way besides indexing the signal, the result is a system that is not TI.

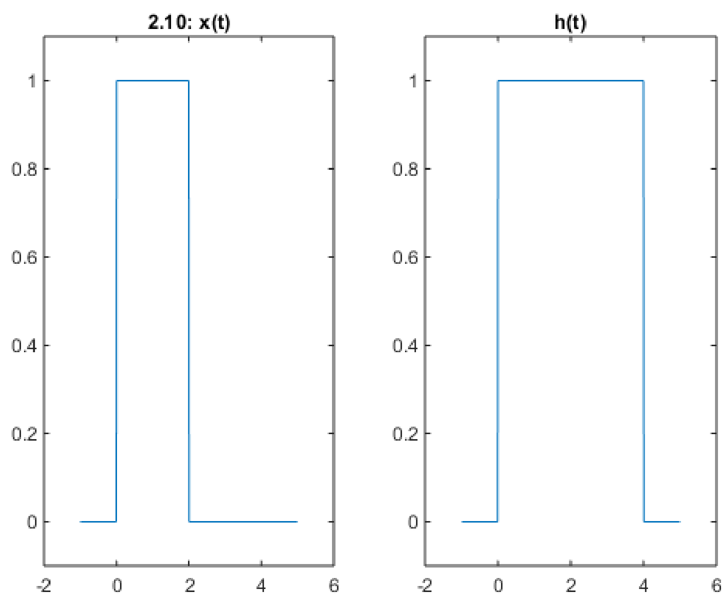
2.10(a) Functions x(t) and h(t) are both rectangular pulses, as shown in Fig. P2.10. Apply graphical convolution to determine y(t) = x(t) ∗ h(t) given the following data.

The problem statement almost directly translates to MATLAB code

```
A = 1; B = 1; T1 = 2; T2 = 4;
```

Graphing the given functions...

```
Tmax = max(T1,T2) + 1; % go a bit past the longest pulse
Tmin = -1; % start a bit before 0
dt = 0.01;
t = Tmin:dt:Tmax;
x = zeros(size(t)); x(0<=t & t<=T1) = A;
h = zeros(size(t)); h(0<=t & t<=T2) = B;
figure
subplot(121),plot(t,x),ylim([-0.1 1.1]),title('2.10: x(t)')
subplot(122),plot(t,h),ylim([-0.1 1.1]),title('h(t)')
```



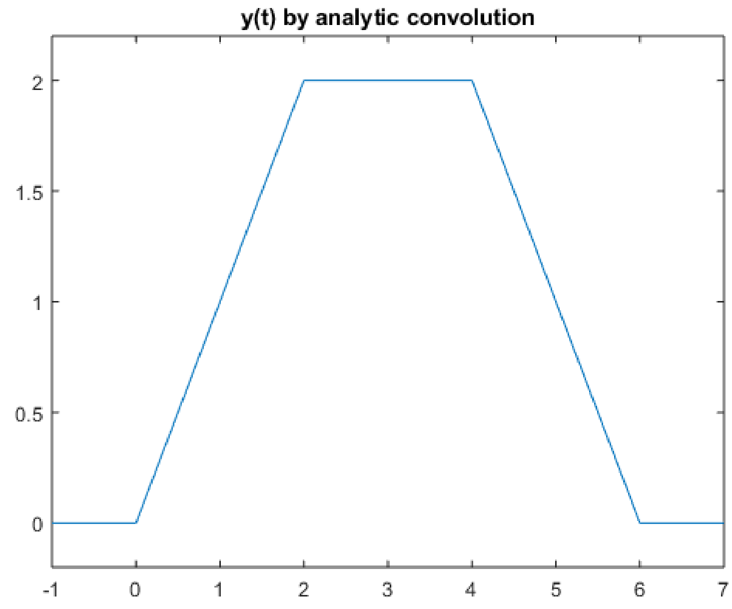
Let us choose to reverse x instead of h. As we delay the reversed x, we encounter 5 regions...

1. t<0, no overlap, result is 0
2. 0<t<T1=2, partial overlap of width t
3. 2<t<4, complete overlap of x of width T1=2
4. 4<t<6, partial overlap from t-2 to 4.
5. 6<t, no overlap, result is 0.

Since there are no impulses, all the inequalities are also satisfied at the edges. That is, the area resulting from integration cannot change instantaneously. Putting this together...

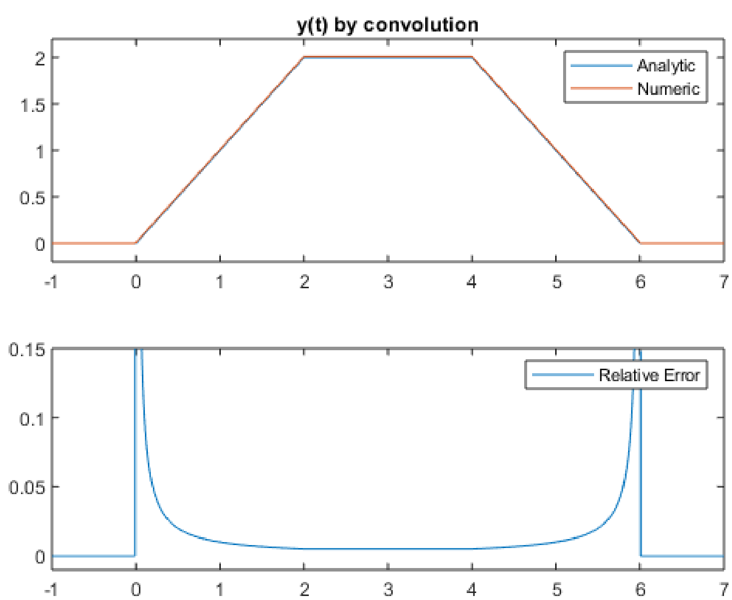
```
t2 = -1:dt:7; % go beyond the result boundaries
y = zeros(size(t2)); % regions 1 and 5
r2 = 0<=t2 & t2<=T1;
y(r2) = t2(r2) * (A * B); % rectangle W*H, where height is product function
r3 = 2 <= t2 & t2 <= 4;
y(r3) = T1 * (A * B);
r4 = 4<=t2 & t2<= 6;
```

```
y(r4) = (6-t2(r4)) * (A * B); % width = 4-(t-2) = 6-t
figure
plot(t2,y),ylim([-0.2 1.1*max(y)])
title('y(t) by analytic convolution')
```



As a check, have MATLAB compute the convolution numerically and compare it with the analytic result.

```
y3 = dt * conv(x,h);
t3 = (2*Tmin):dt:(2*Tmax); % width property applied to MATLAB's domain of x and h
reg = t2(1)<=t3 & t3<=t2(end); % subset where wide result overlaps domain of y above
y3 = y3(reg); % keep only the values in this region; now y3 has corresponding time values t2 (not t3)
figure
subplot(2,1,1),plot(t2,y,t2,y3),ylim([-0.2 1.1*max(y)])
title('y(t) by convolution'),legend('Analytic','Numeric')
relErr = (y3-y) ./ (y+eps); % Error of y3 relative to y, add epsilon to denominator to regularlize near 0: 0/(0+eps)->0
subplot(2,1,2),plot(t2,relErr),legend('Relative Error')
ylim([-0.01 30*median(relErr)]) % don't let outliers at transition from y(t)=0 dominate the scale
```



The error will improve as dt is made smaller. The main soure of error here is that the analytic solution is the true result while the conv() approach approximates the convolution integral using Riemann rectangles.