6.003 Homework #1

Due at the beginning of recitation on **September 14, 2011.**

Problems

1. **Solving differential equations**

   Solve the following differential equation

   $$y(t) + 3\frac{dy(t)}{dt} + 2\frac{d^2y(t)}{dt^2} = 1$$

   for $t \geq 0$ assuming the initial conditions $y(0) = 1$ and $\left.\frac{dy(t)}{dt}\right|_{t=0} = 2$. Express the solution in closed form. Enter your closed form expression the the box below.

   [Hint: assume the homogeneous solution has the form $Ae^{s_1 t} + Be^{s_2 t}$.]

   $y(t) = $ [   ]

2. Solving difference equations

Solve the following difference equation

$$8y[n] - 6y[n-1] + y[n-2] = 1$$

for $n \geq 0$ assuming the initial conditions $y[0] = 1$ and $y[-1] = 2$. Express the solution in closed form. Enter your closed form expression the the box below.

[Hint: assume the homogeneous solution has the form $Az_1^n + Bz_2^n$.]

$y[n] =$

3. Geometric sums

a. Expand $\dfrac{1}{1-a}$ in a power series.

power series:

For what range of a does your answer converge?

range:

b. Express $\sum_{n=0}^{N-1} a^n$ in closed form.

closed form:

For what range of a does your answer converge?

range:

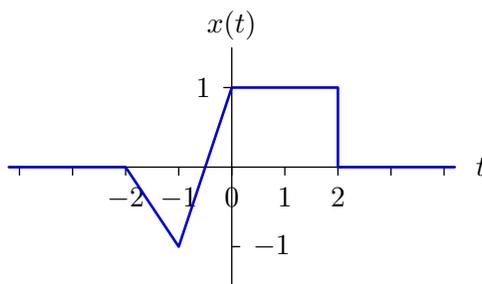**c.** Expand $\dfrac{1}{(1-a)^2}$ in a power series.

power series:

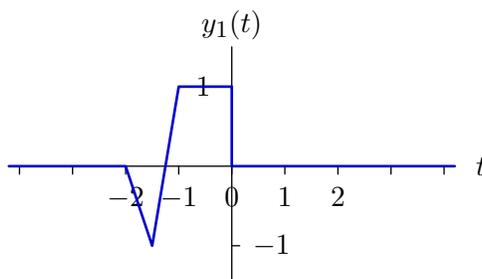For what range of a does your answer converge?

range:

4. CT transformations

Let $x(t)$ represent the signal shown in the following plot.

$x(t)$

The signal is zero outside the range $-2 < t < 2$.

a. The following plot shows $y_1(t)$, which is a signal that is derived from $x(t)$.

$y_1(t)$

Determine an expression for $y_1(t)$ in terms of $x(\cdot)$.

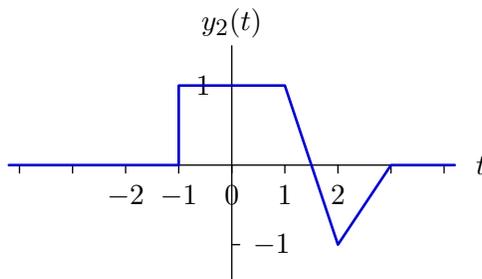$y_1(t) =$ _____

b. The following plot shows $y_2(t)$, which is a signal that is derived from $x(t)$.

$y_2(t)$

Determine an expression for $y_2(t)$ in terms of $x(\cdot)$.

$y_2(t) =$ _____

c. Let $y_3(t) = x(2t + 3)$. Determine all values of t for which $y_3(t) = 1$.

range of t :

---

d. Assume that $x(t)$ can be written as the sum of an even part

$$x_e(t) = x_e(-t)$$

and an odd part

$$x_o(t) = -x_o(-t) \,.$$

For what values of t is $x_e(t) = 0$?

values of t:

Engineering Design Problems

5. Decomposing Signals

The even and odd parts of a signal $x[n]$ are defined by the following:

- $x_e[-n] = x_e[n]$ (i.e., x_e is an even function of n)

- $x_o[-n] = -x_o[n]$ (i.e., x_o is an odd function of n)

- $x[n] = x_e[n] + x_o[n]$

Let $x_r[n]$ represent the part of $x[n]$ that occurs for $n \geq 0$,

$$x_r[n] = \begin{cases} x[n] & n \geq 0 \\ 0 & \text{otherwise} \end{cases}.$$

Let $x_l[n]$ represent the part of $x[n]$ that occurs for $n < 0$),

$$x_l[n] = \begin{cases} x[n] & n < 0 \\ 0 & \text{otherwise} \end{cases}.$$

Notice that $x_r[0] = x[0]$ while $x_l[0] = 0$.

a. Is it possible to determine $x[n]$ (for all n) from $x_e[n]$ and $x_r[n]$?

Yes or **No**:

If yes, explain a procedure for doing so. If no, explain why not.

b. Is it possible to determine $x[n]$ (for all n) from $x_o[n]$ and $x_l[n]$?
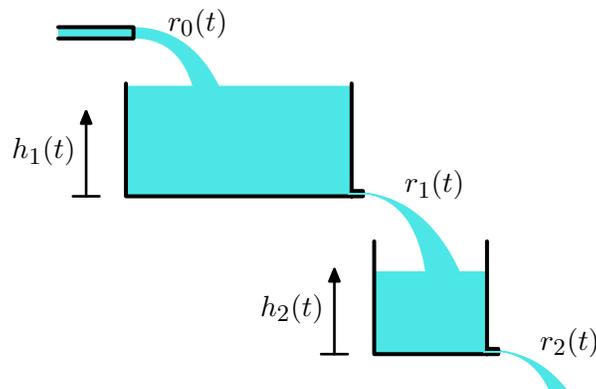
Yes or **No**:

If yes, explain a procedure for doing so. If no, explain why not.

6. Leaky tanks

The following figure illustrates a cascaded system of two water tanks. Water flows

- into the first tank at a rate $r_0(t)$,
- out of the first tank and into the second at a rate $r_1(t)$, and
- out of the second tank at a rate $r_2(t)$.

The rate of flow out of each tank is proportional to the height of the water in that tank: $r_1(t) = k_1 h_1(t)$ and $r_2(t) = k_2 h_2(t)$, where k_1 and k_2 are each $0.2\,\mathrm{m^2}$/second. Both tanks have heights of $1\,\mathrm{m}$. The cross-sectional area of tank 1 is $A_1 = 4\,\mathrm{m^2}$ and that of the second tank is $A_2 = 2\,\mathrm{m^2}$. At time $t = 0$, both tanks are empty.

Part a. Let $x(t) = r_0(t)$ represent the input of the tank system and $y(t) = r_2(t)$ represent the output. Determine the relation between the input and the output. Express this relation as a differential equation of the form

$$a_0 y(t) + a_1 \frac{dy(t)}{dt} + a_2 \frac{d^2 y(t)}{dt^2} + \cdots = x(t) + b_1 \frac{dx(t)}{dt} + b_2 \frac{d^2 x(t)}{dt^2} + \cdots$$

where the coefficient of $x(t)$ is 1.

$a_0,\ a_1,\ a_2,\ \cdots$:

$b_1,\ b_2,\ b_3,\ \cdots$:

Part b. Assume that $r_0(t)$ is held constant at rate r_0. What is the maximum value of r_0 such that neither tank will ever overflow if both tanks start out empty.

r_0:

Part c. Because $r_1(t)$ is both the output of the first tank and the input of the second tank, we can equivalently think of the two-tank system as a cascade of two one-tank systems, as shown in the following figure.

$$r_0(t) \longrightarrow \boxed{\text{tank 1}} \xrightarrow{\;r_1(t)\;} \boxed{\text{tank 2}} \longrightarrow r_2(t)$$
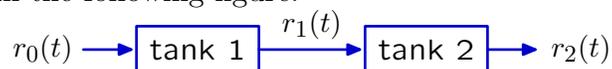
Determine a differential equation that relates $r_1(t)$ to $r_0(t)$. Determine the solution to this differential equation when $r_0(t)$ is held constant at $0.1\,\text{m}^3/\text{s}$. Assume that tank #1 is initially empty.

$r_1(t) =$

Part d. We could similarly determine a differential equation that relates $r_2(t)$ to $r_1(t)$ and solve it for $r_2(t)$ given the solution for $r_1(t)$ given in Part c. As an alternative, we can use a numerical method.

Use the forward Euler approximation to generate a discrete approximation to the differential relation between $r_1(t)$ and $r_0(t)$, as follows. Let $r_0(t)$ and $r_1(t)$ be approximated by discrete sequences $r_0[n] = r_0(nT)$ and $r_1[n] = r_1(nT)$, where T represents the step size. Then approximate the continuous-time derivative at time nT by a first difference:

$$\left.\frac{dr_1(t)}{dt}\right|_{t=nT} \approx \frac{r_1[n+1] - r_1[n]}{T}.$$

Solve this difference equation for $r_1[n+1]$ in terms of values of $r_1[k]$ and $r_0[k]$ where $k < n+1$ and enter the result below.

$r_1[n+1] =$

Part e. Use your favorite computer language to solve this recursion for the special case when the input $r_0[n]$ is held constant at $0.1\,\text{m}^3/\text{s}$, tank #1 is initially empty, and $T = 1$ second (see example code in box below). Make a plot of your solution for $0 < t < 60$. Also plot the analytic result from part c on the same axes. Determine the maximum difference between the analytic and numerical results.

maximum difference:

Part f. Modify your code to calculate numerical approximations to both $r_1(t)$ and $r_2(t)$. Plot results for both on the same axes. Explain similarities and differences of these two results for both small times and large times.

similarities:

differences:

7. **Drug dosing**

When drugs are used to treat a medical condition, doctors often recommend starting with a higher dose on the first day than on subsequent days. In this problem, we consider a simple model to understand why. Assume that the human body is a tank of blood and that drugs instantly dissolve in the blood when ingested. Further assume that drug vanishes from the blood (either because it is broken down or because it is flushed by the kidneys) at a rate that is proportional to drug concentration.

Let $x[n]$ represent the amount of drug taken on day n, and let $y[n]$ represent the total amount of drug in the blood on day n, just after the dose $x[n]$ has dissolved in the blood, so that

$$y[n] = x[n] + \alpha y[n-1] \,.$$

a. Assume that no drug is in the blood before day 0, and that one unit of drug is taken each day, starting with day 0.

1. Determine an expression for the amount of drug in the blood immediately after the dose on day n has dissolved.

amount:

2. Plot the amount of drug in the blood as a function of day number for $\alpha = \frac{1}{2}$, $\frac{3}{4}$, and $\frac{7}{8}$.

3. Determine an expression for the steady-state amount of drug in the blood, i.e., $\lim_{n \to \infty} y[n]$.

$\lim\limits_{n \to \infty} y[n]$:

b. In part a, the amount of drug in the blood ramps up over the first few days, before reaching a steady-state value. Suggest a different initial dose $x[0]$ that will result in a more constant amount of drug in the blood (with $x[n]$ remaining at 1 for all $n \geq 1$).

initial dose:

Appendix: Fibonacci code

You may use Python and/or Matlab/Octave to solve problems in this homework assignment. Octave is a free-software linear-algebra solver, with a syntax that is similar to that of Matlab. Octave is available for most platforms. See **www.octave.org**.

The following code calculates, prints, and plots the first 20 Fibonacci numbers (i.e., $f[0]$ through $f[19]$).

Example Matlab/Octave code

```
y(1) = 1;                       % initial conditions
y(2) = 1;                       % indices start at 1 (not 0)
for i = 3:20
    y(i) = y(i-1)+y(i-2)
end
y                               % print y
stem(0:19,y)
```

Example Python code

```
from pylab import *
y = [1,1]                       # initial conditions
for i in range(2,20):
    y.append(y[i-1]+y[i-2])
print y
stem(range(20),y)
show()
```

6.003 Signals and Systems
Fall 2011