Chapter 4 HW Solution

Problem 1. Here we want to verify the Fourier series representing the impulse train:

$$\sum_{k=-\infty}^{\infty} \delta(t-kT) = \frac{1}{T} \sum_{n=-\infty}^{\infty} e^{jn\omega_s t} \tag{1}$$

I wrote a MATLAB function called `train.m` (after all, we're trying to represent a *train* of impulses):

```
function [t,imp] = train(T,N)

% Function [t,imp] = TRAIN(T,N) creates a Fourier series approximation to a
% train of impulses with period T using N harmonic terms each side of zero.
% Three periods of the function are generated.  Returned variable "imp" is
% the impulse train, while "t" is a corresponding time vector, useful in
% plotting.

ws = (2*pi)/T;  % Sampling frequency in rad/s

t = [0:0.001*T:3*T]';  % Create a column time vector over 3 periods with 1000 points/period

imp = zeros(length(t),1);  % Dummy array to hold impulse train

for n = -N:N    % FOR loop over the number of coefficients
    imp = imp + real(exp(j*n*ws*t));  % Ignore imag part (should be zero anyway)
end;
```

Using `train.m` with `N = 6, 20, and 100` terms, I obtained the plot in Figure 1 below. Looks pretty good!

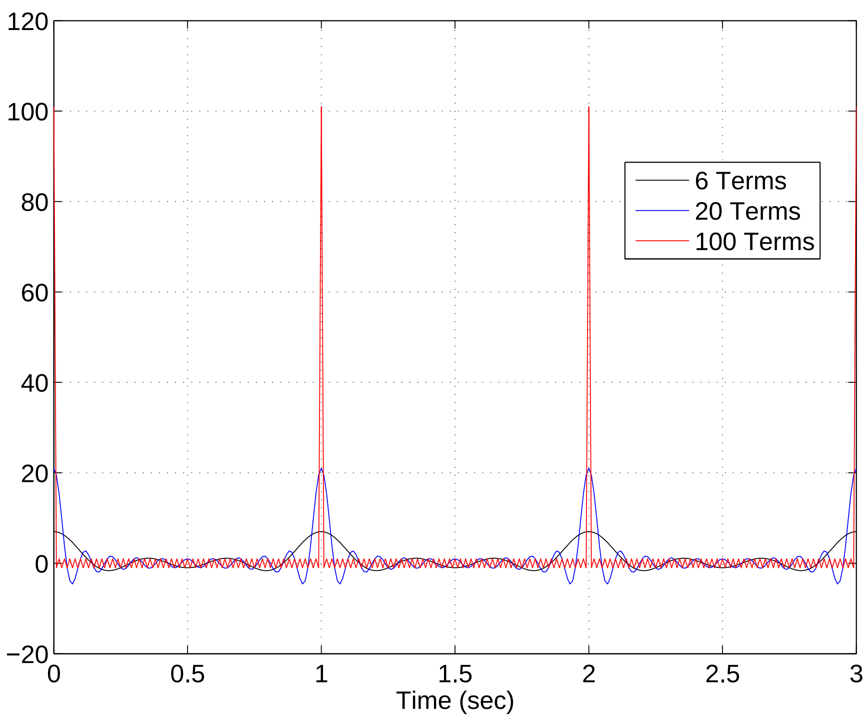


Figure 1: Fourier series approximation to impulse train.

Problem 2. The "ZOH" discretization of $G(s)$ yields

$$G(z) = \frac{z-1}{z}\mathcal{Z}\left\{\frac{G(s)}{s}\right\} = \frac{z-1}{z}\mathcal{Z}\left\{\frac{1}{s(s+1)}\right\} = \frac{z-1}{z}\left\{\frac{z(1-e^{-T})}{(z-1)(z-e^{-T})}\right\} = \frac{0.6321}{z-0.3679}. \tag{2}$$

Using MATLAB, we can accomplish the same thing:

```
>> T = 1;
>> numc = [0 1];  % G(s) numerator is 1
>> denc = [1 1];  % G(s) denominator is (s+1)
>> Gc = tf(numc,denc);  % Form G(s) as LTI system
>> Gd = c2d(Gc,T,'zoh')  % Discretize using 'zoh' method

Transfer function:
  0.6321
----------
z - 0.3679

Sampling time: 1
```

So we get the same result. Using this $G(z)$, the closed-loop transfer function is

$$W(z) = \frac{Y(z)}{R(z)} = \frac{G(z)}{1+G(z)} = \frac{0.6321}{z+0.2642} \tag{3}$$

You can also find the closed-loop transfer function using MATLAB:

```
>> Wd = feedback(Gd,1)  % Close unity feedback loop around Gd (the '1' is the feedback TF)

Transfer function:
  0.6321
----------
z + 0.2642
```

Note that the closed-loop pole location is $z = -0.2642$, which corresponds to "alternating" behavior (*not* oscillating—refer to Section 2.4.2).

Anyway, I only asked you to find the first three samples (0,1,2) of the unit step response, and those samples can be found using MATLAB as

```
>> y = step(Wd,2)  % Find output samples 0,1,2

y =      0
    0.6321
    0.4651
```

Those are the *samples* (plotted in Figure 2(a)), but we also want the response *between* samples. As discussed in Section 4.5.3, the response between samples is the (scaled and biased) plant step response. So if the step response of $G(s)$ is of the form $g(t)$, between samples you just fit the curve

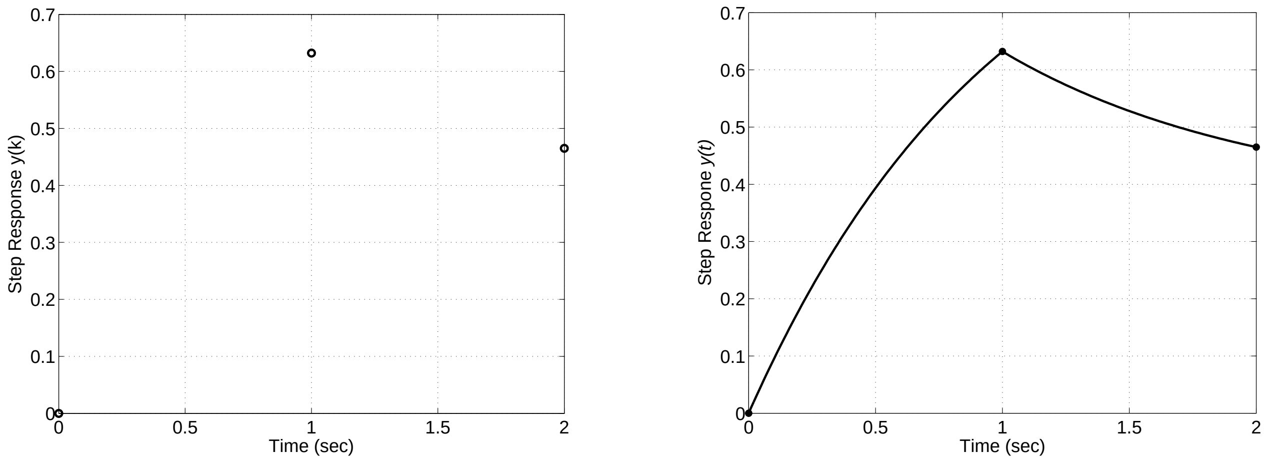
$$y(t) = a_0 + a_1 g(t)$$

where A and B are constants to be determined. The unit step response of this system is

$$1 - e^{-t} \tag{4}$$

so between samples we fit the equation

$$y(t) = a_0 + a_1 e^{-t} \tag{5}$$



(a) Samples of unit step response.

(b) Response between samples.

Figure 2: Unit step response: samples and response *between* samples.

The result is shown in Figure 2(b). So now you know what would happen between samples.

Problem 3. Using the "zero-order hold" method of obtaining an equivalent $G(z)$, we get
(a) $G(s)$ is a double integrator $1/s^2$ (like an inertia with force/torque input); we can use #5 in the (online) transform table:

$$G(z) = \frac{z-1}{z}\mathcal{Z}\left\{\frac{1}{s^3}\right\} = \frac{z-1}{z}\left\{\frac{T^2}{2}\frac{z(z+1)}{(z-1)^3}\right\} = \frac{T^2}{2}\frac{z+1}{(z-1)^2} = \frac{0.5(z+1)}{(z-1)^2} \tag{6}$$

(b) $G(s)$ is $1/[s(s+1)]$ (a DC motor with voltage input or an inertia with force/torque input and viscous friction); we can use lucky #13 in the transform table:

$$\begin{aligned} G(z) &= \frac{z-1}{z}\mathcal{Z}\left\{\frac{1}{s^2(s+1)}\right\} = \frac{z-1}{z}\left\{\frac{z[(T-1+e^{-T})z+(1-e^{-T}-Te^{-T})]}{(z-1)^2(z-e^{-T})}\right\} \\ &= \frac{(T-1+e^{-T})z+(1-e^{-T}-Te^{-T})}{(z-1)(z-e^{-T})} = \frac{0.3679z+0.2642}{(z-1)(z-0.3679)} = \frac{0.3679(z+0.7183)}{(z-1)(z-0.3679)} \end{aligned} \tag{7}$$

(c) $G(s)$ is $1/(s^2-1)$, an unstable system (like a magnetic levitation system); this must be expanded in partial fractions (either before or after multiplying by $1/s$...I'll do it before, since the algebra is a little simpler):

$$G(s) = \frac{1}{s^2-1} = 0.5\left(\frac{1}{s-1} - \frac{1}{s+1}\right) \tag{8}$$

hence the discrete equivalent $G(z)$ is

$$G(z) = 0.5\left[\frac{z-1}{z}\right]\mathcal{Z}\left\{\frac{1}{s(s-1)} - \frac{1}{s(s+1)}\right\} \tag{9}$$

which, after some algebraic reduction simplifies to

$$G(z) = 0.5\left[\frac{(e^T+e^{-T}-2)z+(e^T+e^{-T}-2)}{(z-e^T)(z-e^{-T}}\right]_{T=1} = \frac{0.5431(z+1)}{(z-2.7183)(z-0.3679)} \tag{10}$$

If you use MATLAB `c2d(Gc,T,'zoh')` with `T = 1` you will get exactly the same results as if you substitute $T = 1$ into (6), (7), and (10). Note that (c) is unstable in both continuous and discrete domains (which must be true).

Problem 4. Here you're going to recontruct a time function using only its samples. The ideal reconstruction formula is equation (4.28):

$$r(t) = \sum_{k=-\infty}^{\infty} r(kT)\,\mathrm{sinc}\frac{\pi(t-kT)}{T} \tag{11}$$

You will only be summing from $k = 0...5$, and the samples $r(kT)$ come from the 1 Hz sine wave. I wrote a MATLAB script file to compute the reconstruction using two nested loops: the outer loop over time (0 to 1 seconds with 0.01 time step) and the inner loop over k to compute the summation in (2).

NOTE: The MATLAB `sinc` function *will not work* for this assignment. I think you have to use the `sinc.m` function I have on my website, since I don't think you can scale the MATLAB function to work properly.

```
% This script file verifies the "ideal reconstruction" formula of equation
% (4.28), using 6 samples (0..5) of a 1 Hz sine wave sampled at T=0.2 sec.

% First create the sample to be used in reconstruction

k = [0:5]';
T = 0.2;
tk = k*T;
rk = sin(2*pi*k*T);  % Calculate samples

% Next calculate the reconstruction using a finer time step

t = 0:0.01:1;   % Time step for generating reconstruction
r = sin(2*pi*t); % Calculate continuous time function to compare
N = length(t);  % Number of sample in time vector
rr = zeros(N,1); % Create zero vector to hold reconstructed r(t)

for i = 1:N % Loop over time
    for k = 0:5  % Loop to compute reconstruction
        rr(i) = rr(i) + rk(k+1)*sinc(pi*(t(i)-k*T)/T);
    end;
end;
```

The results are shown in Figure 3.

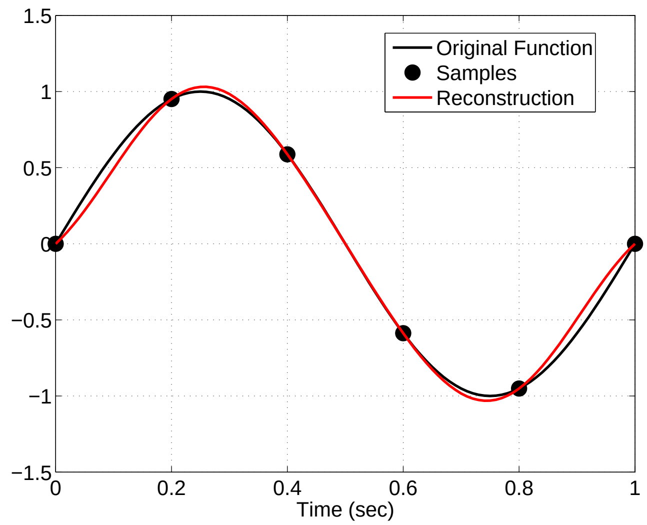


Figure 3: Original, sampled, and reconstructed sine wave.