Prob 1

\[ H(s) = \frac{1+2s}{1+4s+3s^2} = \frac{1+2s}{(1+3s)(1+s)} = \frac{1}{1+3s} + \frac{1}{1+s} = \frac{1}{s+1} + \frac{1}{s+3} \]

Pole @ \( s = -\frac{1}{3}, s = -1 \).

Response 1: ROC: \( \text{Re}\{s\} < -1 \)

\[ h(t) = -\frac{1}{6} e^{-\frac{1}{3}t} u(-t) - \frac{1}{2} e^{-t} u(-t) \]

Response 2: ROC: \( -1 < \text{Re}\{s\} < -\frac{1}{3} \)

\[ h(t) = -\frac{1}{6} e^{-\frac{1}{3}t} u(-t) + \frac{1}{2} e^{-t} u(t) \]

Response 3: ROC: \( \text{Re}\{s\} > -\frac{1}{3} \)

\[ h(t) = \frac{1}{6} e^{-\frac{1}{3}t} u(t) + \frac{1}{2} e^{-t} u(t) \]

Response 3 is causal \( h(t) \)

Ans: # of \( h(t) = 3 \)

causal \( h(t) = \left[ \frac{1}{6} e^{-\frac{1}{3}t} u(t) + \frac{1}{2} e^{-t} u(t) \right] \).
\[
X(s) = \int_{-\infty}^{\infty} te^{-t} e^{-st} u(t) \, dt
\]
\[
= \int_{0}^{\infty} te^{-t} e^{-st} \, dt.
\]
\[
u = t, \quad du = e^{-(s+1)t} \, dt \quad \Rightarrow \quad v = \frac{e^{-(s+1)t}}{-(s+1)}
\]
\[
t = \frac{e^{-(s+1)t}}{-(s+1)} \quad \Rightarrow \quad e^{-(s+1)t} = \frac{1}{s+1}
\]
\[
\frac{1}{s+1} \left[ \frac{-1}{s+1} \right] \]
\[
\frac{1}{(s+1)^2}
\]

Alternative Method

\[
\frac{d}{ds} X(s) \quad \text{to} \quad \frac{d}{dt} x(t)
\]
\[
m = x(t) \quad \text{to} \quad X(s)
\]
\[
t = x(t) \quad \text{to} \quad \frac{d}{ds} X(s)
\]
\[
\frac{d}{dt} x(t) = e^{-t}, \quad X(s) = \frac{1}{s+1}
\]
\[
\Rightarrow \quad \frac{d}{ds} X(s) = \frac{1}{(s+1)^2}
\]
Problem 3

\( h[n] = n^2 \cdot u[n] \)

\( H(z) = \frac{1}{1 - z^{-1}} \)

Then:

\( n \cdot h[n] \rightarrow \frac{1}{1 - z^{-1}} \cdot z^n = \frac{z^n}{z-1} \)

Thus:

\( u[n] \rightarrow \frac{1}{1 - z^{-1}} = \frac{z}{z-1} \)

\( n \cdot u[n] \rightarrow -z \frac{d}{dz} \left( \frac{z}{z-1} \right) = -z \left( \frac{z-1-z}{(z-1)^2} \right) = \frac{z}{(z-1)^2} \)

\( n^2 \cdot u[n] \rightarrow -z \frac{d}{dz} \left( \frac{z}{(z-1)^2} \right) = -z \left( \frac{(z-1)^2 - z(z-2)(z-1)}{(z-1)^4} \right) \)

\( = -z \left( \frac{z-1-z}{(z-1)^3} \right) = \frac{z(z+1)}{(z-1)^3} \)

(Page 775 of Book)
Problem 4
\[ a_k = \begin{cases} \frac{1}{2}i_k, & k \neq 0 \\ 0, & k = 0 \end{cases} \]

Problem 5
Given \( a_k = 0 \) for \( k = 0 \).

Priseval's Theorem
\[ \sum_{k=-\infty}^{\infty} \left| \frac{1}{2i_k} \right|^2 = \frac{1}{2\pi} \int_{-\infty}^{\infty} \left( \frac{\pi}{2} - \frac{1}{2}t^2 \right) dt. \]

\[ \Rightarrow \sum_{k=-\infty}^{\infty} \frac{1}{4k^2} = \frac{1}{2\pi} \int_{-\infty}^{\infty} \left( \frac{\pi^2}{4} - \frac{\pi}{2}t + \frac{1}{4}t^2 \right) dt. \]

\[ \Rightarrow \sum_{k=-\infty}^{\infty} \frac{1}{k^2} = \frac{2}{\pi} \left[ \frac{\pi^2}{4} \left( \frac{1}{2} - \frac{1}{3} \right) \right]. \]

\[ \Rightarrow \sum_{k=-\infty}^{\infty} \frac{1}{k^2} = \frac{2}{\pi} \left( \frac{\pi^3}{2} - \frac{\pi^3}{3} + \frac{2}{3} \pi^3 \right) = \frac{\pi^2}{6}. \]
Problem 5

a) \( f(t) = \sum_{n=1}^{\infty} \frac{\sin(nt)}{n} \), \( f'(t) = \sum_{n=1}^{\infty} \frac{\cos(nt)}{n} = \frac{1}{4} \left( \sum_{n=-\infty}^{\infty} e^{int} + e^{-int} \right) - \frac{1}{2} \)

given identity: \( \sum_{n=-\infty}^{\infty} e^{int} = 2\pi \delta(t) \)

\( \Rightarrow f'(t) = \frac{1}{4} (4\pi \delta(t)) - \frac{1}{2} = \pi \delta(t) - \frac{1}{2} \).

b) \( a_k = \frac{1}{2\pi} \int_{0}^{2\pi} \left( \frac{\pi}{2} - \frac{1}{2}t \right) e^{-ikt} \, dt \)

\( \Rightarrow a_k = \frac{1}{4} \int_{0}^{2\pi} e^{-ikt} \, dt - \frac{1}{4\pi} \int_{0}^{2\pi} te^{-ikt} \, dt \)

\( \Rightarrow a_k = \frac{1}{2 \pi} \left[ \frac{e^{-ikt}}{-ik} \right]_0^{2\pi} - \frac{1}{4\pi} \left[ \frac{te^{-ikt}}{-ik} \right]_0^{2\pi} + \int_{0}^{2\pi} \frac{1}{ik^2} e^{-ikt} \, dt \)

\( \Rightarrow a_k = 0 - \frac{1}{4\pi} \left[ \frac{e^{-ikt}}{-ik} \right]_0^{2\pi} + \int_{0}^{2\pi} \frac{1}{ik^2} e^{-ikt} \, dt \)

\( \Rightarrow a_k = \left[ \frac{1}{2} \right] \frac{1}{ik} \)

\( a_k = \frac{1}{2 \cdot i} \)
Problem 5c)

Matlab Code:

t=[0:2*pi/1000:2*pi];
y=zeros(1, size(t));
N=10;

for n=1:N
    y=y+sin(n.*t)./n;
end

y_target=pi/2-.5.*t;
plot (t,y,t,y_target);

xlabel('time')
ylabel('F_N(t)')
title('N=10')
\[ t_c = n \cdot \frac{1}{2} \cdot (\frac{1}{x} - 0.5) \]

\[ t_c = n \cdot \frac{1}{2} \cdot (\frac{1}{x} - 1) \]

\[ t_c = n \cdot \frac{1}{2} \cdot (\frac{1}{x} - 2) \]
Error seems to tend roughly towards 0.15.
Problem 5d)

Matlab Code:

```matlab
t=[0:2*pi/1000:2*pi];
y=zeros(1, size(t));
z=zeros(1, size(t));
N=200;

for n=1:N
    for m=1:n
        y=y+sin(m.*t)./m;
    end
    z=z+y./N;
    y=zeros(1, size(t));
end

y_target=pi/2-.5.*t;
plot(t,z,t,y_target);
xlabel('time')
ylabel('F_N(t)')
title('N=200')
```

![Graph showing time vs. \( F_N(t) \) for different values of \( N \).]
The curves in part d is a lot smoother due to the fact that the lower frequency harmonics contribute a higher portion into the computation of $F_N(t)$ than into the computation of $f_N(t)$.
a. The desired "matched filter" is $p[n]$ shown in the following plot.

$$
\begin{align*}
\alpha^*(t) &= 3 \delta_{-1} \alpha(t) + 2 \delta_{0} \alpha(t) \\
\beta^*(t) &= 3 \delta_{-1} \beta(t) - 3 \delta_{0} \beta(t)
\end{align*}
$$
If we flip this signal about $n = 0$ and then shift it to the right by two, we get $p[n]$ to line up with the desired pattern in $x[n]$.

b. The following code was used to verify that the matched filter worked:

```
z = sign(randn(1,50));
y = conv(z,[1 -1 -1]); % note: the matched filter is flipped
figure(1); stem(1:50,z);
figure(2); stem(1:52,y);
find(y==max(y))
```

Figure 1 shows the random sequence of 1's and -1's used for the test.

Figure 2 shows the output of the matched filter.

The `find` command located maxima at $n = 3, 8, 11, 19, 27, 32, 37, 40, \text{ and } 40$. These points are displaced to the right by one because the `conv` command does not allow the matched filter to start at $n = -1$. 

8
c. The following code was used to find smiley.

```matlab
findsmiley = (double(imread('findsmiley.bmp'))-126.)/128.;
zz = conv2([+1 +1 +1 +1 +1 +1 +1;+1 -1 -1 -1 -1 -1 +1;+1 -1 +1 +1 +1 -1 +1;+1 -1 -1 -1 -1 -1 +1;+1 -1 +1 -1 +1 -1 +1;+1 -1 -1 -1 -1 -1 +1;+1 +1 +1 +1 +1 +1 +11,findsmiley);
[r,c]=find(zz==max(max(zz)));
```

The code performs a two-dimensional convolution and then finds the peak using the `find` command. The reported peak is at row 758, column 847. Of course, this is expected to be shifted down by 4 and to the right by 3 from the actual location of the nose (because Matlab doesn't allow functions to start at n < 1). Thus the nose should be at row 754, column 844. The following code and resulting image verifies this prediction.

```matlab
imagesc(findsmiley(754-3+754+3,844-4:844+4)); colormap gray;
```

![Image of the code results](image_url)

d. By repeating trials with different levels of noise, it was found that when `amplitude=1`, the matched filter found smiley in more than half of the random noise patterns that were tested. This is pretty good when you consider how noisy the resulting smiley looked. The following figure shows a test case with `amplitude=1` where smiley was still found!