HOMEWORK 4 SOLUTIONS

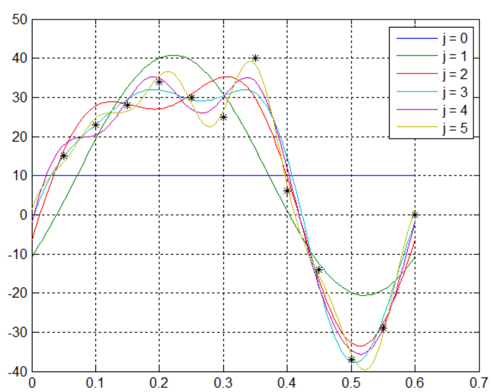
1. The following code calculates the Fourier series coefficients a_j, b_j of the input force and calculates the response of the system:

```
clc;
clear all;
close all;

% data
t_data = 0.05:0.05:0.6;
f_data = [15 23 28 34 30 25 40 6 -14 -37 -29 0];

% compute Fourier coefficients
N = length(t_data);
a0 = (2/N)*sum(f_data)
n_Four = 5;
a = zeros(n_Four,1);
b = zeros(n_Four,1);
for j = 1:n_Four
    a(j) = (2/N)*sum(f_data.*cos((2*j*pi/t_data(end))*t_data));
    b(j) = (2/N)*sum(f_data.*sin((2*j*pi/t_data(end))*t_data));
end
a
b

% plot Fourier series approximations
t_approx = linspace(0,t_data(end),101);
f_approx = a0/2*ones(n_Four+1,length(t_approx));
str{1} = 'j = 0';
for j = 1:n_Four
    f_approx(j+1,:) = f_approx(j,:) + a(j)*cos((2*pi*j/t_data(end))*t_approx) + b(j)*sin((2*pi*j/t_data(end))*t_approx);
    str{j+1} = sprintf('j = %d',j);
end
figure;
plot(t_approx,f_approx,t_data,f_data,'k*');
grid on;
legend(str);

% plot system response to input
m = 1;
k = 15000;
zeta = 0.1;
wn = sqrt(k/m);
w = 2*pi/t_data(end);
r = w/wn;
x_approx = a0/(2*k);
for j = 1:n_Four
    phi = atan2(2*zeta*j*r,1-j^2*r^2)
    cos_amp = a(j)/k/sqrt((1-j^2*r^2)^2+(2*zeta*j*r)^2)
    sin_amp = b(j)/k/sqrt((1-j^2*r^2)^2+(2*zeta*j*r)^2)
    x_approx = x_approx + cos_amp*cos(j*w*t_approx-phi) + sin_amp*sin(j*w*t_approx-phi);
end
figure;
plot(t_approx,x_approx);
grid on;
```

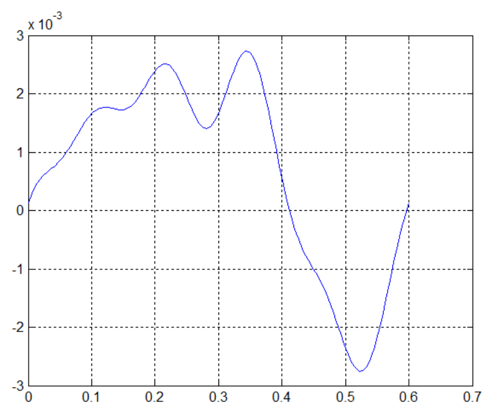
The output is the Fourier series coefficients, and the amplitudes and phase angles of the terms of the response $x(t)$:

```
a0 =

  20.166666666666664

a =

 -20.791022319648810

   4.333333333333342

   4.833333333333341

  -0.333333333333321

   3.457688986315494

b =

  22.535039255505101

  12.413030787576952

  -1.333333333333339

   3.175426480542941

  -2.868372588838444
```

```
phi =
   0.017224901248208
cos_amp =
  -0.001396068944514
sin_amp =
   0.001513175638232
phi =
   0.035217050129206
cos_amp =
     2.974069128730587e-04
sin_amp =
     8.519356536764968e-04
phi =
   0.054860170603792
cos_amp =
     3.443979333738073e-04
sin_amp =
    -9.500632644794713e-05
phi =
   0.077309453666741
cos_amp =
    -2.509079470706951e-05
sin_amp =
     2.390219217920947e-04
phi =
   0.104246556215106
cos_amp =
     2.805346473304514e-04
sin_amp =
    -2.327213048388109e-04
```



Fourier series approximations



Response x(t)

2. In this problem, the input is a constant 1000 N-m for 15/16th of the period, $\tau = \frac{60\text{ s/min}}{1000\text{ rpm}} = 0.06\text{ s}$, and 0 N-m for the last 1/16th of the period. The Fourier series coefficients can thus be found explicitly:

$$a_0 = \frac{2}{\tau}\int_0^{15\tau/16} 1000\,dt = 1875$$

$$a_j = \frac{2}{\tau}\int_0^{15\tau/16} 1000\cos\left(\frac{2\pi jt}{\tau}\right)dt = \frac{1000}{\pi j}\sin(1.875\pi j)$$

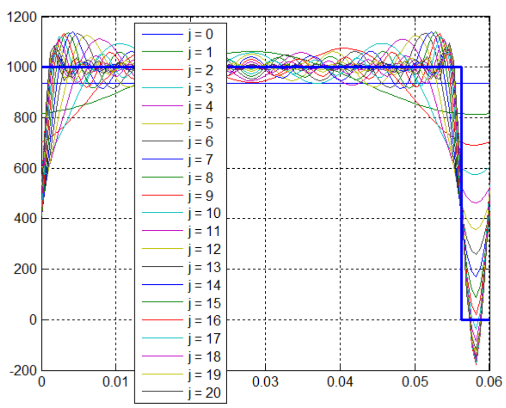
$$b_j = \frac{2}{\tau}\int_0^{15\tau/16} 1000\sin\left(\frac{2\pi jt}{\tau}\right)dt = \frac{1000}{\pi j}\left[1-\cos(1.875\pi j)\right]$$

In this problem, $\omega = \frac{2\pi}{\tau} = \frac{100\pi}{3} = 105\text{ rad/s}$, $\omega_n = \sqrt{\frac{k}{J_0}} = 1982\text{ rad/s}$, $r = \frac{\omega}{\omega_n} = 0.053$, and $\zeta = 0$.

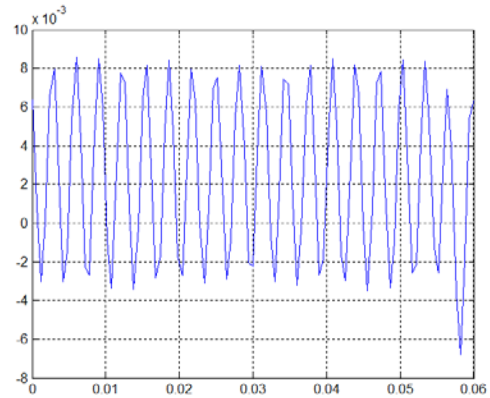
The following code calculates the Fourier series coefficients a_j, b_j of the input force and calculates the response of the system:

```
clc;
clear all;
close all;

% data
w = 2*pi*1000/60
tau = 2*pi/w
t_data = [0 15/16*tau 15/16*tau tau];
m_data = [1000 1000 0 0];

% compute Fourier coefficients
a0 = 1875
n_Four = 20;
a = zeros(n_Four,1);
b = zeros(n_Four,1);
for j = 1:n_Four
    a(j) = 1000/(pi*j)*sin(1.875*pi*j);
    b(j) = 1000/(pi*j)*(1-cos(1.875*pi*j));
end
a
b

% plot Fourier series approximations
t_approx = linspace(0,tau,101);
f_approx = a0/2*ones(n_Four+1,length(t_approx));
str{1} = 'j = 0';
for j = 1:n_Four
    f_approx(j+1,:) = f_approx(j,:) + a(j)*cos((2*pi*j/tau)*t_approx) +
b(j)*sin((2*pi*j/tau)*t_approx);
    str{j+1} = sprintf('j = %d',j);
end
figure;
plot(t_approx,f_approx);
line(t_data,m_data,'LineWidth',2);
grid on;
legend(str);
```

```
% plot system response to input
J0 = 0.1;
k = 392700;
zeta = 0;
wn = sqrt(k/J0);
w = 2*pi/tau;
r = w/wn;
tht_approx = a0/(2*k);
for j = 1:n_Four
    phi = atan2(2*zeta*j*r,1-j^2*r^2)
    cos_amp = a(j)/k/sqrt((1-j^2*r^2)^2+(2*zeta*j*r)^2)
    sin_amp = b(j)/k/sqrt((1-j^2*r^2)^2+(2*zeta*j*r)^2)
    tht_approx = tht_approx + cos_amp*cos(j*w*t_approx-phi) + sin_amp*sin(j*w*t_approx-phi);
end
figure;
plot(t_approx,tht_approx);
grid on;
```



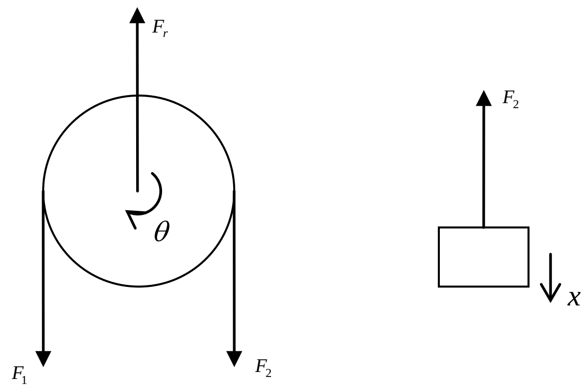
Fourier series approximations



Response θ(t)

Note that more terms are required to get a good approximation of the input signal due to the sharp discontinuities.

3. Free body diagrams of the pulley and mass are, respectively:



Since the block only translates, we can write the sum of the forces in the vertical direction:

$$m\ddot{x} = \sum F_x = -F_2 = -k_2\left(x - r\theta\right)$$
$$m\ddot{x} + k_2 x - k_2 r\theta = 0$$

Since the pulley only rotates, we can write the sum of the moments about its point of rotation:

$$J_0\ddot{\theta} = \sum M_0 = -F_1 r + F_2 r = -k_1\left(r\theta\right)r + k_2\left(x - r\theta\right)r$$
$$\frac{1}{2}m_0 r^2\ddot{\theta} - k_2 r x + \left(k_1 + k_2\right)r^2\theta = 0$$

Writing these equations in matrix-vector form yields

$$\underbrace{\begin{bmatrix} m & 0 \\ 0 & \frac{1}{2}m_0 r^2 \end{bmatrix}}_{\mathbf{M}}\underbrace{\begin{bmatrix} \ddot{x} \\ \ddot{\theta} \end{bmatrix}}_{\ddot{\mathbf{q}}} + \underbrace{\begin{bmatrix} k_2 & -k_2 r \\ -k_2 r & \left(k_1 + k_2\right)r^2 \end{bmatrix}}_{\mathbf{K}}\underbrace{\begin{bmatrix} x \\ \theta \end{bmatrix}}_{\mathbf{q}} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$