16 Road Vehicle on Random Terrain

An autonomous, four-wheeled road vehicle travels over an uncertain terrain. You are asked to characterize the vertical motion of the center of gravity, and the pitching (nose-up vs. nose-down) response of the vehicle, as it crosses the terrain, using a *linear model.* We know that the vehicle chassis has a mass of $m = 42kg$ and a pitching moment of inertia of $J = 25kg\ m^2$. Consider the tires and suspension to have zero mass, but the front and back suspension systems are each modeled as having a spring of stiffness $k = 1000N/m$, in parallel with a linear damper having coefficient $b = 100N/(m/s)$. The distance between the front and rear wheels is $2l = 0.9m$. The vehicle travels at a speed U, which we will vary below. Because we consider U to be constant, you can use it directly to map between the horizontal terrain coordinate x and time t in this problem.

The terrain roughness has been described only statistically (e.g., using an orbiting camera or some other remote sensing system). Its content is given by the following harmonic components:

n	wave number, 1/m ($k_n = 2\pi/\lambda_n$)	amplitude, m (a_n)
1	0.250	0.01
2	0.275	0.02
3	0.300	0.02
4	0.325	0.04
5	0.350	0.03
6	0.375	0.02
7	0.400	0.01
8	0.425	0.03
9	0.450	0.05
10	0.475	0.01
11	0.500	0.02

Here, k_n is the n'th wave number (spatial frequency) and λ_n is the n'th wavelength (spatial period). We model the ground elevation as a function of the horizontal coordinate:

$$h(x) = \sum_n a_n \sin(k_n x + \phi_n).$$

Notice that $2l << \min(\lambda)$; the vehicle is much shorter than the smallest wavelength, so you can approximate

$$\begin{aligned} h(x) &\approx [h(x+l) + h(x-l)]/2, \\ h'(x) = dh(x)/dx &\approx [h(x+l) - h(x-l)]/2l, \\ h''(x) = d^2h(x)/dx^2 &\approx [h'(x+l) - h'(x-l)]/2l, \quad \text{and so on.} \end{aligned}$$

1. Develop a model that describes the response of the vehicle to terrain variations $h(x)$. It should comprise two uncoupled second-order differential equations, one for vertical motion and the other for pitch. Make an annotated figure to go with the equations. You may like to use the fact that the vertical velocity of the ground *seen from a reference frame moving horizontally at velocity U*, is $Uh'(x)$.

   Newton's laws and some geometry give us

$$\begin{aligned} m\ddot{z} &= -k[(z + l\theta - h(x+l)) + (z - l\theta - h(x-l))] \\ &\quad -b[(\dot{z} + l\dot{\theta} - Uh'(x+l)) + (\dot{z} - l\dot{\theta} - Uh'(x-l))] \\ &\approx -k[2z - 2h(x)] - b[2\dot{z} - 2Uh'(x)] \end{aligned}$$

$$\begin{aligned} J\ddot{\theta} &= -kl[(z + l\theta - h(x+l)) - (z - l\theta - h(x-l))] \\ &\quad -bl[(\dot{z} + l\dot{\theta} - Uh'(x+l)) - (\dot{z} - l\dot{\theta} - Uh'(x-l))] \\ &\approx -kl[2l\theta - 2lh'(x)] - bl[2l\dot{\theta} - 2lUh''(x)]. \end{aligned}$$

   As advertised, these equations are uncoupled, except through the input $h(x)$ and its derivatives.

2. What are the damped and undamped natural frequencies of the two systems? What are the damping ratios? List all six numbers, with units.

   For the vertical motion, the undamped natural frequency is $\omega_n = \sqrt{2k/m} \approx 6.9rad/s$, and the damping ratio is $\zeta = 2b/2m\omega_n \approx 0.35$, giving a damped natural frequency of $\omega_d = \omega_n\sqrt{1-\zeta^2} \approx 6.5rad/s$. For the pitch motion, the undamped natural frequency is $\omega_n = \sqrt{2l^2k/J} \approx 4.0rad/s$, and the damping ratio is $\zeta = 2l^2b/2J\omega_n \approx 0.20$, giving a damped natural frequency of $\omega_d = \omega_n\sqrt{1-\zeta^2} \approx 3.9rad/s$. Both of these modes are rather lightly damped and so may exhibit resonance behavior.

3. Create a set of elevation data $h(x)$ based on the above (spatial) frequency content, and plot it. Consider the range of x from zero to twenty times the longest wavelength; your spacing in x should be no more than one-tenth of the smallest wavelength, to get a good picture. Use a random phase angle ϕ_n for each of your components.

   See the attached figures and code. Note that I used fifty cycles of the longest wavelength (not twenty).

4. What is the maximum amplitude of elevation $h(x)$ in the terrain data you created? What is the maximum amplitude of the *slope*?

   The maximum amplitude shown is about $0.21m$, with a maximum slope of about $4.8°$. These values come out differently each time the program is run because the phases are determined randomly.

5. Now run your vehicle model over this terrain at $U = 5m/s$. What is the maximum amplitude of the vertical motion? What is the maximum amplitude of the pitch motion?

The maximum vertical motion amplitude is about $0.23m$ *and the pitch is about* $6.4°$ *- both of these are larger than the terrain itself, indicating that some dynamic effect is occurring.*

6. Repeat this calculation for speeds of 10, 15, 20, and $25m/s$. Make a tabular listing for the vehicle's maximum amplitudes of vertical motion and pitch, as a function of U. Are the motions consistent with the systems' resonant frequencies and damping properties?

   The vertical motion amplitudes for $U = [5, 10, 15, 20, 25]m/s$ *are approximately* $[0.23, 0.29, 0.34, 0.29, 0.23]m$*, respectively. The peak amplification is* $0.34/0.23$ *or almost fifty percent, consistent with the damping ratio we found. The driving frequencies from the terrain are given by* $\omega_n = Uk_n$*, so that* $15m/s \times 0.45rad/m = 6.75rad/s$ *- a high-amplitude harmonic input (0.05m) is occurring very close to the damped resonant frequency. The pitch motion amplitudes are approximately* $[6.4, 9.9, 4.9, 2.2, 1.3]°$*. The peak amplification is over two, consistent with the lower damping ratio in pitch. The slope of any particular terrain harmonic goes as* $k_n a_n$*, so again the input at* $k = 0.45rad/m$ *is the dominant one.* $10m/s \times 0.45rad/m = 4.5rad/s$*, again pretty close to our damped natural frequency.*

7. What different effects could we expect if the wheelbase became long compared to some of the terrain harmonic components? (No calculations.)

   We would start to see spatial aliasing. This would grotesquely deform our neat explanations above, and invalidate the approximations we used for the slopes and average values of $h(x)$*. This latter point could, however, be easily fixed in the code simply by carrying out the full expressions, that is, using* $h(x + l)$*,* $h(x - l)$*, and so on.*

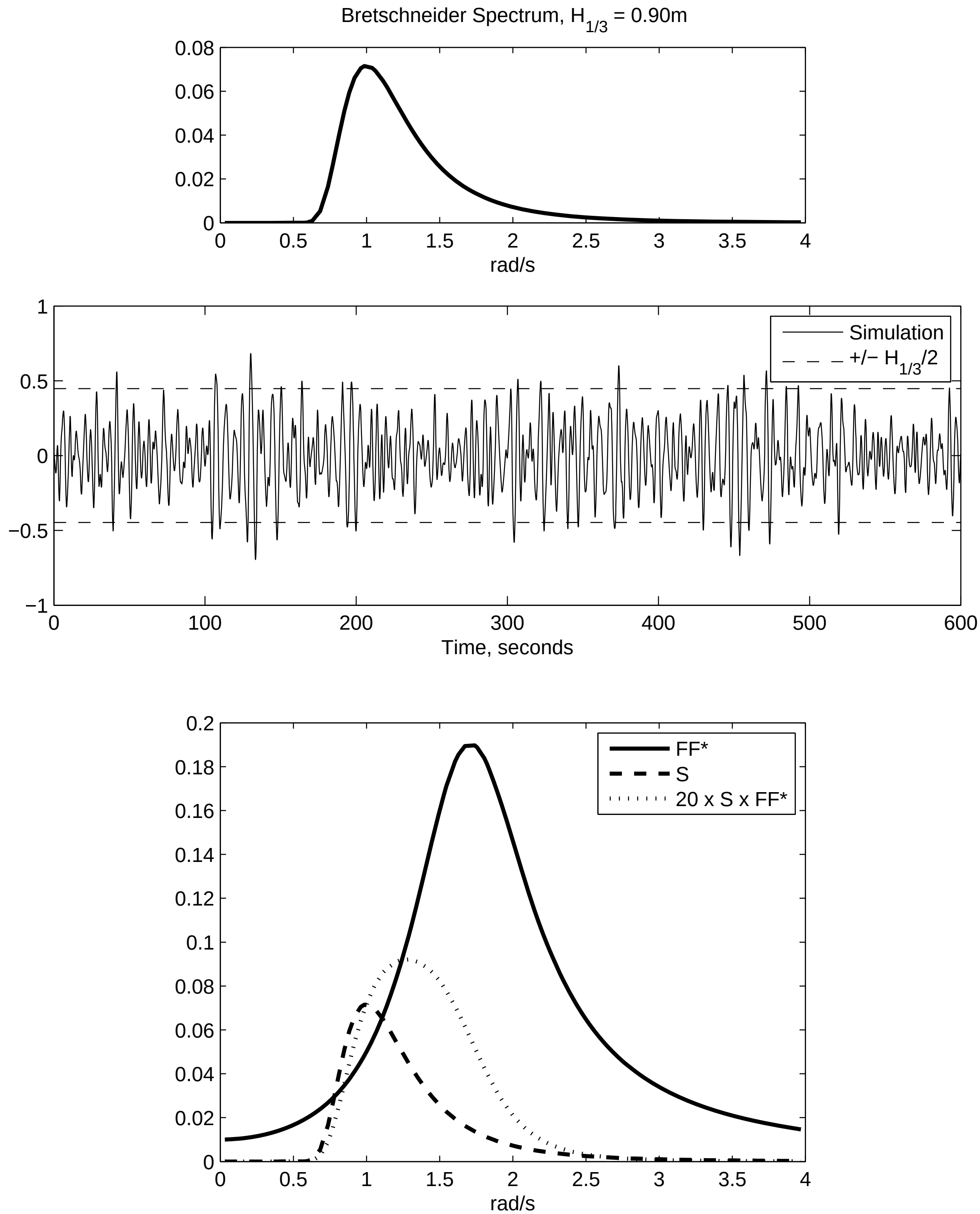
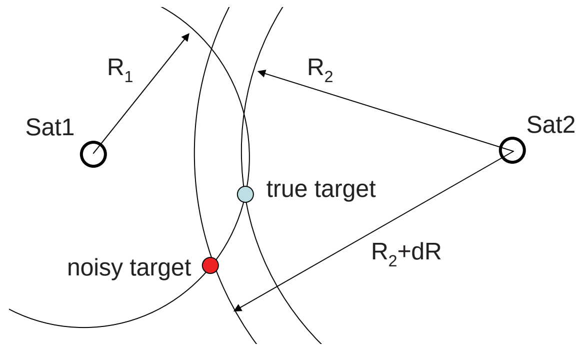
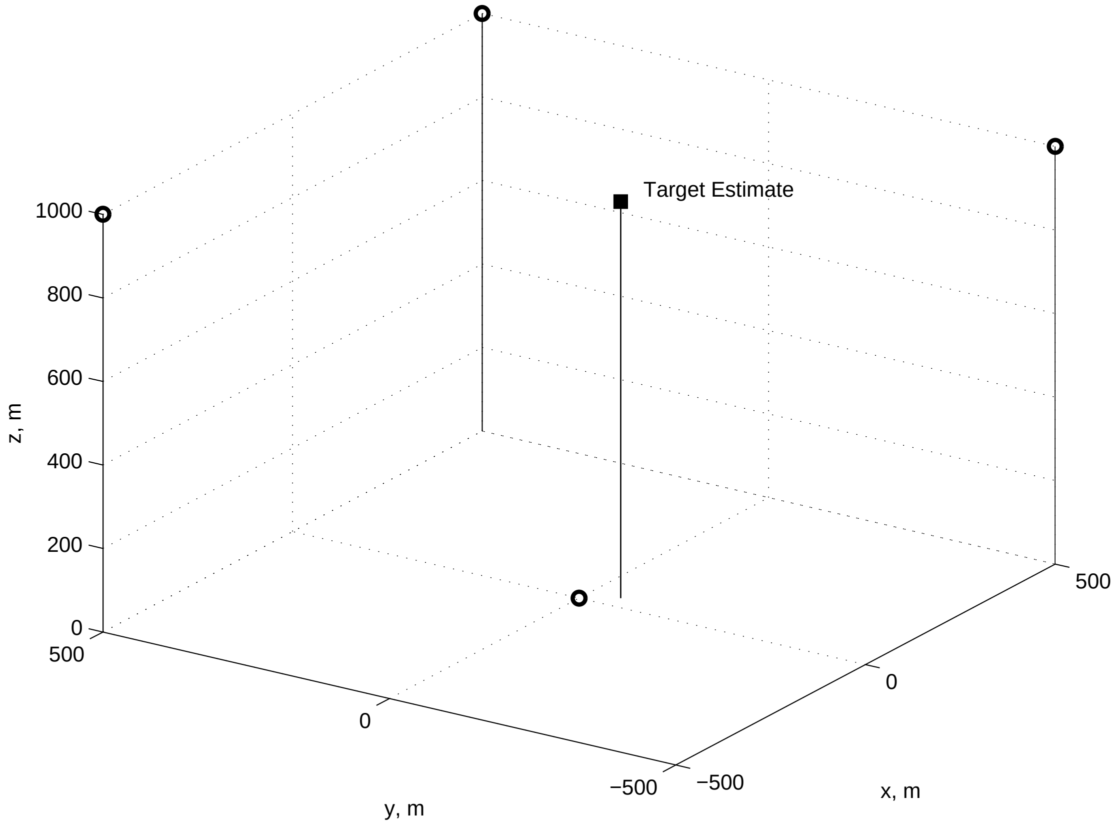
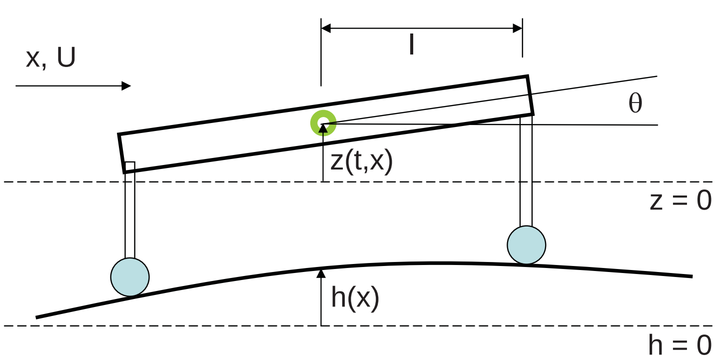
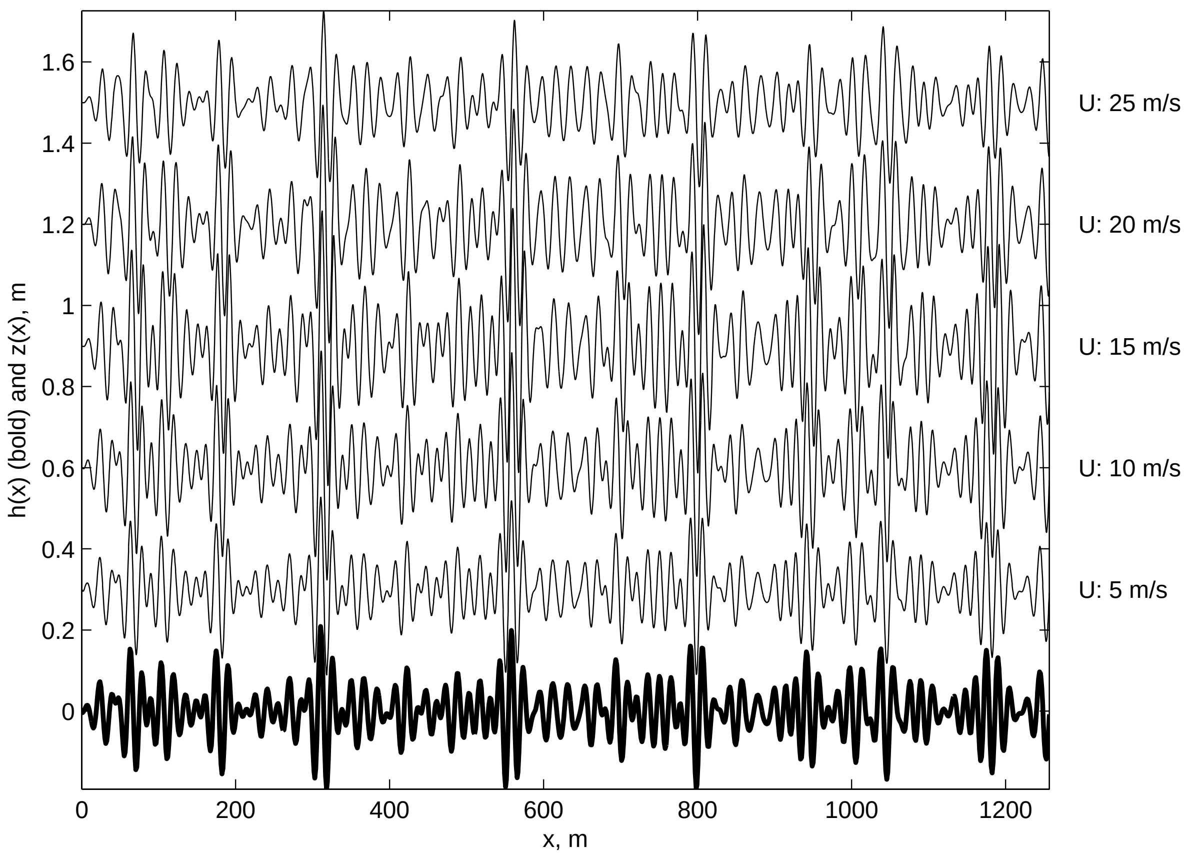


Figure 2: Bretschneider spectrum and vehicle pitch response spectra.

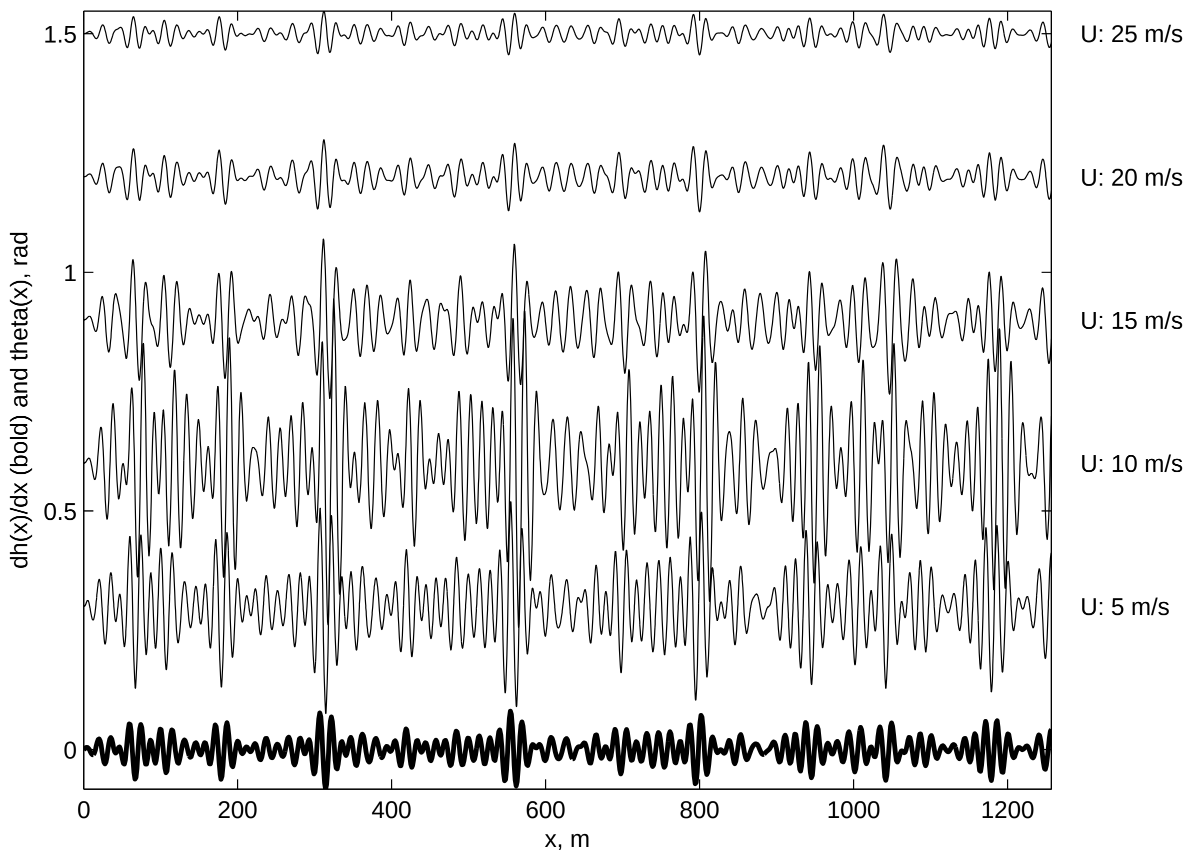

R1
R2
Sat1
Sat2
true target
noisy target
R2+dR


Target Estimate
1000
800
600
400
200
0
z, m
500
0
−500
y, m
−500
0
500
x, m


x, U
l
θ
z(t,x)
z = 0
h(x)
h = 0



1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
h(x) (bold) and z(x), m
0
200
400
600
800
1000
1200
x, m
U: 25 m/s
U: 20 m/s
U: 15 m/s
U: 10 m/s
U: 5 m/s



1.5
1
0.5
0
0
200
400
600
800
1000
1200
x, m
dh(x)/dx (bold) and theta(x), rad
U: 25 m/s
U: 20 m/s
U: 15 m/s
U: 10 m/s
U: 5 m/s

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Land Vehicle Moving Over Rough Terrain
% MIT 2.017 FSH Sept 2009
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear all;
global U m J l k b phi a kk;

k = .25:.025:.5 ; % terrain wavenumbers
a = [.01 .02 .02 .04 .03 .02 .01 .03 .05 .01 .02] ; % amplitudes

m = 42 ; % mass of body
J = 25 ; % rotary moment of inertia about centroid
l = .45 ; % half-width between wheels, m
kk = 1000 ; % strut stiffness, N/m
b = 100 ; % strut damping coefficient

phi = 2*pi*rand(length(k),1) ; % set the random phase angles for this run

% set up and calculate a few derived items

vertDrawingScale = .3 ; % drawing parameters
pitchDrawingScale = .3 ;

if length(a) ~= length(k),
    disp('Inconsistent k and a!');
    break;
end;

lam = 2*pi./k ; % wavelengths

dk = mean(diff(k));
k = k + rand(size(k))*dk/2 ; % randomize the wavenumbers a little bit

dx = min(lam)/10; % pick x-spacing to get ten points in the fastest cycle
xvec = 0:dx:max(lam)*50; % a vector of x-values we'll look at; this
                         % gets 50 cycles of the slowest harmonic

% Characterize the resonance and damping properties of the two LTI systems

disp('-------------------------');
wnVert = sqrt(2*kk/m) ;
disp(sprintf('Undamped natural freq. of vertical system:  %g rad/s', wnVert));
```

```
zetaVert = 2*b / 2 /sqrt(2*kk*m);
disp(sprintf('Damping ratio of vertical system:  %g', zetaVert)) ;

if zetaVert < 1,
    wdVert = wnVert*sqrt(1-zetaVert^2) ;
    disp(sprintf('Damped natural freq. of vertical system:  %g rad/s', wdVert));
else,
    disp('Vertical system is overdamped.');
    wdVert = NaN ;
end;

disp(' ');

wnPitch = sqrt(2*l^2*kk/J) ;
disp(sprintf('Undamped natural freq. of pitch system:  %g rad/s', wnPitch));

zetaPitch = 2*l^2*b / 2 / sqrt(2*l^2*kk*J) ;
disp(sprintf('Damping ratio of pitch system:  %g', zetaPitch)) ;

if zetaPitch < 1,
    wdPitch = wnPitch*sqrt(1-zetaPitch^2) ;
    disp(sprintf('Damped natural freq. of pitch system:  %g rad/s', wdPitch));
else,
    disp('Pitch system is overdamped.');
    wdPitch = NaN ;
end;

% build up the elevation profile for this set of phase angles
for i = 1:length(xvec),
    x = xvec(i) ;
    h(i) = 0 ;
    dhdx(i) = 0 ;
    for j = 1:length(k),
        h(i) = h(i) + a(j)*sin(k(j)*x + phi(j)) ;
        dhdx(i) = dhdx(i) + a(j)*k(j)*cos(k(j)*x + phi(j)) ;
    end;
end;

figure(1);clf;hold off;
subplot('Position',[.15 .15 .7 .75]);
plot(xvec,h,'LineWidth',2) ;
xlabel('x, m');
ylabel('h(x) (bold) and z(x), m');
```

```
figure(2);clf;hold off;
subplot('Position',[.15 .15 .7 .75]);
plot(xvec,dhdx,'LineWidth',2);
xlabel('x, m');
ylabel('dh(x)/dx (bold) and theta(x), rad');

disp(' ');
disp(sprintf(...
   'Terrain:   max|h| %5.3f m   max|dh/dx| %4.2f rad / %4.2f deg',...
    max(abs(h)), max(abs(dhdx)), max(abs(dhdx))*180/pi));

i = 1 ;
for U = [5 10 15 20 25],
    % simulate the system  behavior over this terrain
    clear t s ;
    [t,s] = ode45('vehicleOnTerrainDeriv',[0 max(xvec)/U], [0 0 0 0]');

    figure(1);
    hold on;
    plot(t*U,s(:,2) + vertDrawingScale*i);
    axis('tight');
    text(max(t*U)*1.03,mean(s(:,2)) + vertDrawingScale*i,...
        sprintf('U: %d m/s',U));
    figure(2);
    hold on;
    plot(t*U,2*s(:,4) + pitchDrawingScale*i) ;
    axis('tight');
    text(max(t*U)*1.03,mean(s(:,4)) + vertDrawingScale*i,...
        sprintf('U: %d m/s',U));

    disp(sprintf(...
        'U %2d m/s:  max|z| %5.3f m   max|theta| %4.2f rad / %4.2f deg',...
        U, max(abs(s(:,2))), max(abs(s(:,4))), max(abs(s(:,4)))*180/pi));

    pause(.1);
    i = i + 1 ;
end;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [dsdt] = vehicleOnTerrainDeriv(t,s) ;
global U m J l k b phi a kk;

x = U*t ; % horizontal location of the vehicle
```

```
% build up the elevation and its derivatives at the current time and x
h = 0 ;
dhdx = 0 ;
ddhddx = 0 ;
for j = 1:length(k),
    h = h + a(j)*sin(k(j)*x + phi(j)) ;
    dhdx = dhdx + a(j)*k(j)*cos(k(j)*x + phi(j)) ;
    ddhddx = ddhddx - a(j)*k(j)^2*sin(k(j)*x + phi(j)) ;
end;

% parse out the state vector
zdotIn = s(1);
zIn = s(2) ;
thetadotIn = s(3) ;
thetaIn = s(4) ;

% here are the dynamic equations
zddot = -kk/m*(2*zIn - 2*h) - b/m*(2*zdotIn - 2*U*dhdx);
zdot = zdotIn ;
thetaddot = -kk*l/J*(2*l*thetaIn - 2*l*dhdx) - ...
    b*l/J*(2*l*thetadotIn - 2*l*U*ddhddx) ;
thetadot = thetadotIn ;

% build the state vector derivative
dsdt = [zddot zdot thetaddot thetadot]' ;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

2.017J Design of Electromechanical Robotic Systems
Fall 2009