```
function bicycle_multi_input

close all
clear all

% Goal is to design controller u = -Kx for bicycle dynamics using model
% given in Minor Test 1, Problem 1

% Observer analysis for this in Minor Test 2, Problem 1

% Parameters (rounded to one decimal place)
M = [80.8 2.3; 2.3 0.3];
g = 9.8; %m/s^2
v = 2; %m/s in range 2-7 m/s, for example
C1 = [0 33.9; -0.9 1.7];
K0 = [-81 -2.6; -2.6 -0.8];
K2 = [0 76.6; 0 2.7];

% State-space matrices
B = [0 0; 0 0; 1 0; 0 1];

iM = inv(M);
A21 = -iM*(g*K0 + v^2*K2);
A22 = -iM*v*C1;
A = [B';A21 A22];

% Note that the system is stable
eig(A);

% Can the eigenvalues be arbitrarily placed?
rank([B A*B A*A*B A*A*A*B]);
% % or equivalently
% Wc = ctrb(A, B);
% rank(Wc)

% b1 = B(:,1);
% rank(ctrb(A,b1));
%
% b2 = B(:,2);
% rank(ctrb(A,b2));

% Desired eigenvalues can be written in p
p = [-1, -2, -3, -4];

% The Gain matrix is
K = place(A, B, p);

% Note: Typically, the faster the desired response, the larger is the K matrix,
% meaning that even for small deviation, larger control input is required

% Output Matrix
C = [1 -1 0 0];

% a way to check observability
rank(obsv(A, C));
```

```
% Desired eigenvalues of observer relative to controller
q = p;

% The observer Gain matrix is
L = (place(A', C', q))';

% Can check that the eigenvalues are what is desired
%eig(A - L*C)

% For variables [x; e], e = x - x_hat, the state matrix is
A_cl = [A-B*K B*K;
    0*ones(size(A)) A-L*C];

% time (arbitrary units)
t = 0:.1:100;

% initial conditions
x_cl_0 = [0 0 0 0 10 0 0 0]';

for i=1:length(t)
    % Matrix exponential
    x_cl(i,:) = expm(A_cl*t(i))*x_cl_0;
end

figure;
plot(t, x_cl(:,1), 'r', t, x_cl(:,5), 'b');
```