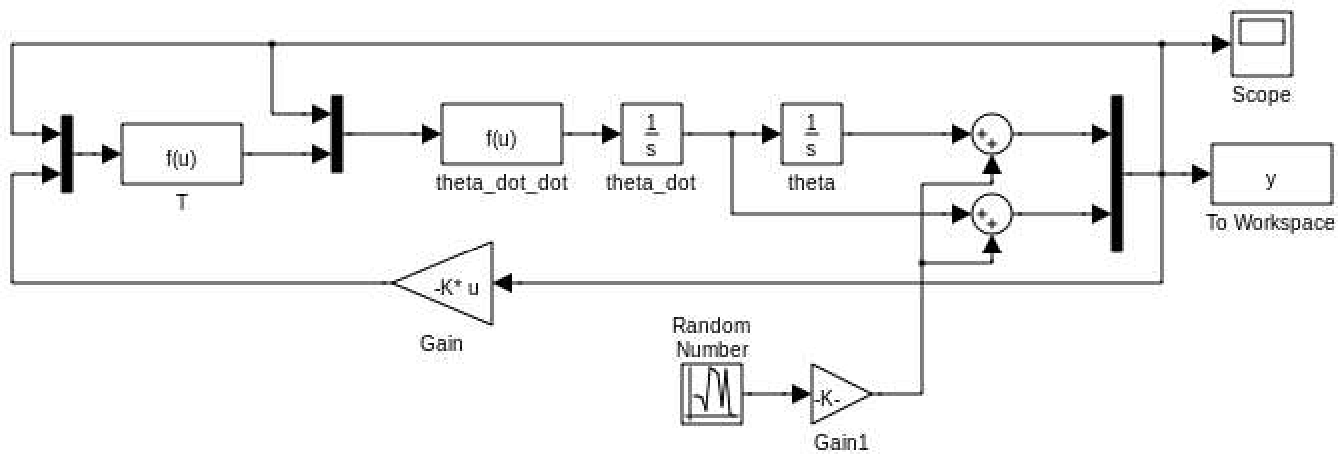
ECE 147C
Solutions, Problem 12.2

Design a feedback linearization controller to drive the inverted pendulum in Example 12.1 to the up-right position. Use the given parameter values. Verify the performance of your system in the presence of measurement noise using Simulink.

```
clc; clear all; close all;

l = 1; m = 1; b = 0.1; g = 9.8;

open('P12_2_model.mdl')
```



The equation of motion for Inverted Pendulum:

$$ml^2 * \ddot{\theta} = mgl * \sin(\theta) - b * \dot{\theta} + T$$

Choose the feedback-linearization control law

$$T = -mgl * sin(\theta) + b * \dot{\theta} + ml^2 * u,$$

where u is the controller's input.

Thus, we have $\ddot{\theta} = u$. That's the state-space system

```
A = [0 1; 0 0]; B = [0; 1]; C = eye(2); D = [0; 0];
```

This system is just a double integrator. It can be stabilized by a simple PD (state feedback) controller.

We choose to penalize only θ and u, but not $\dot{\theta}$:

```
G = [1 0]; H = 0; QQ = G'*G; p = 10; RR = H'*H + p*1; NN = G'*H;
```

The State-Feedback LQR controller (Optimal) is:

```
[K,S,E]=lqr(A,B,QQ,RR,NN);
K
K =

    0.3162    0.7953
```

The state feedback control law is $u := -K * [\theta, \dot{\theta}]$.

Initial conditions.

```
theta0 = 0.5; theta_dot0 = 0.25;
```

Initally turn off noise.

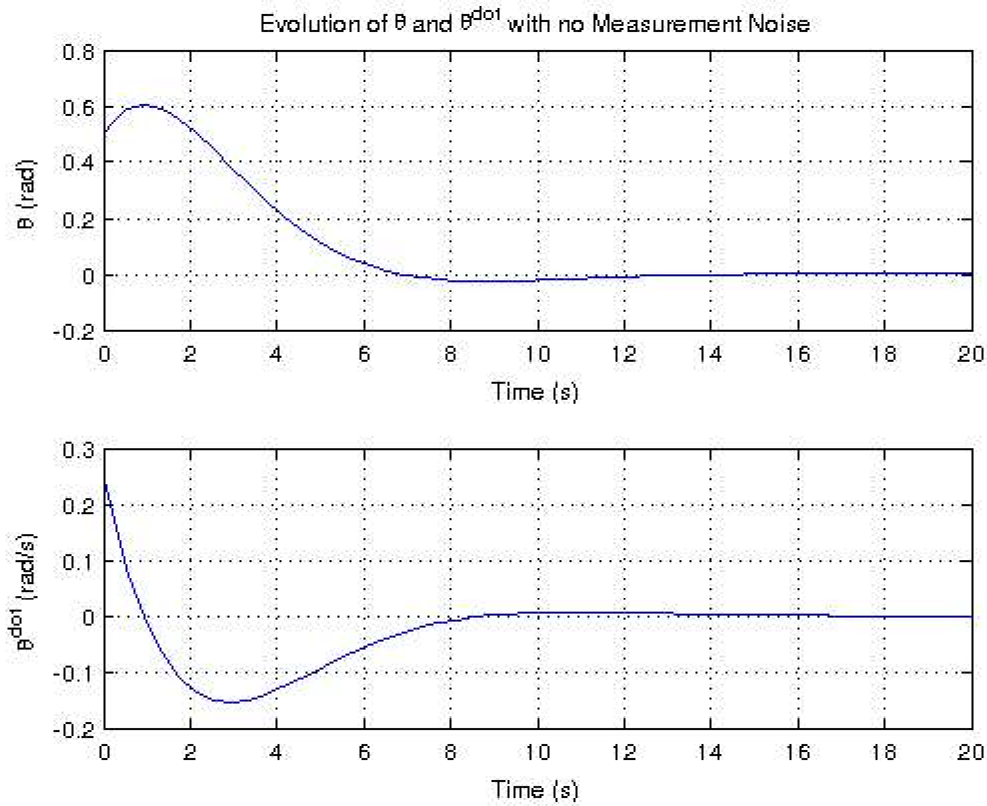
```
Noise = 0.001; Noise_ON = 0;
```

Simulate.

```
warning off; sim('P12_2_model',20);
theta = y.signals.values(:,1);
theta_dot = y.signals.values(:,2);
```

Plot.

```
figure(1); subplot(2,1,1);
plot(tout,theta); xlabel('Time (s)'); ylabel('\theta (rad)'); grid on;
title('Evolution of \theta and \theta^{dot} with no Measurement Noise');
subplot(2,1,2); plot(tout,theta_dot); xlabel('Time (s)');
ylabel('\theta^{dot} (rad/s)'); grid on;
```



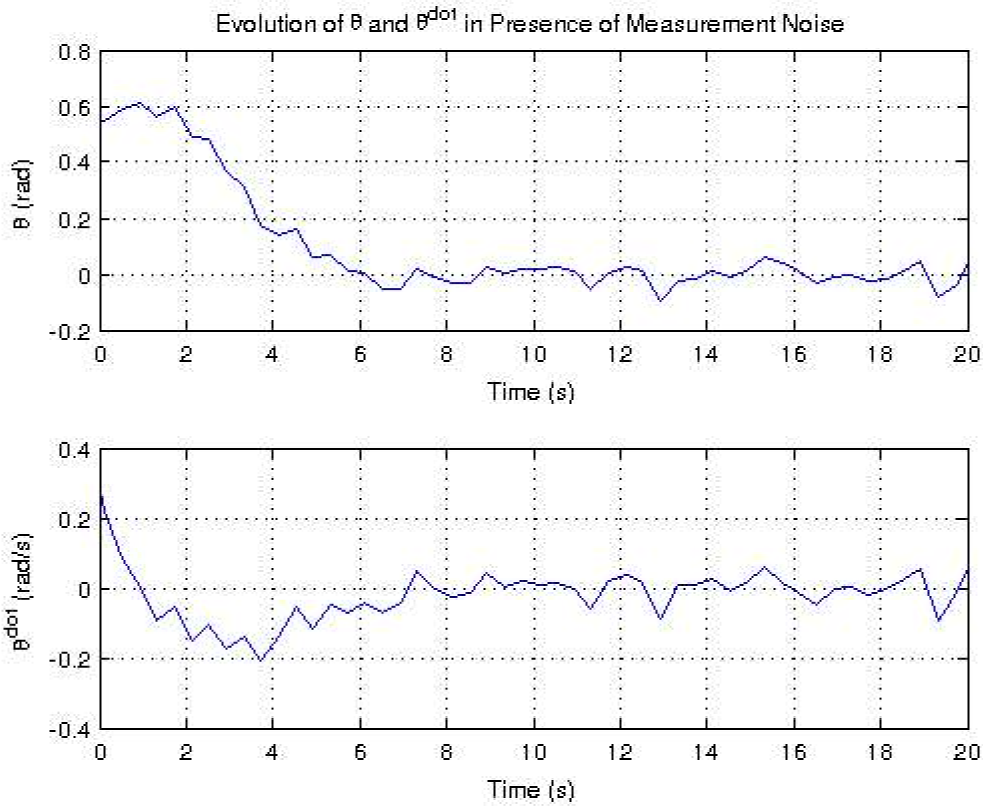
We observe that this controller stabilizes the pendulum from an initial condition (at least in the absence of measurement noise).

Re-simulate with noise.

```
Noise_ON = 1;
sim('P12_2_model',20);
theta = y.signals.values(:,1);
theta_dot = y.signals.values(:,2);
```

Plot.

```
figure(2); subplot(2,1,1);
plot(tout,theta); xlabel('Time (s)'); ylabel('\theta (rad)'); grid on;
title('Evolution of \theta and \theta^{dot} in Presence of Measurement
Noise');
subplot(2,1,2); plot(tout,theta_dot); xlabel('Time (s)');
ylabel('\theta^{dot} (rad/s)'); grid on;
```



We observe that this controller stabilizes the pendulum from an initial condition in the presence of measurement noise. As it can be seen, the controller manages to keep the pendulum close to upward position.