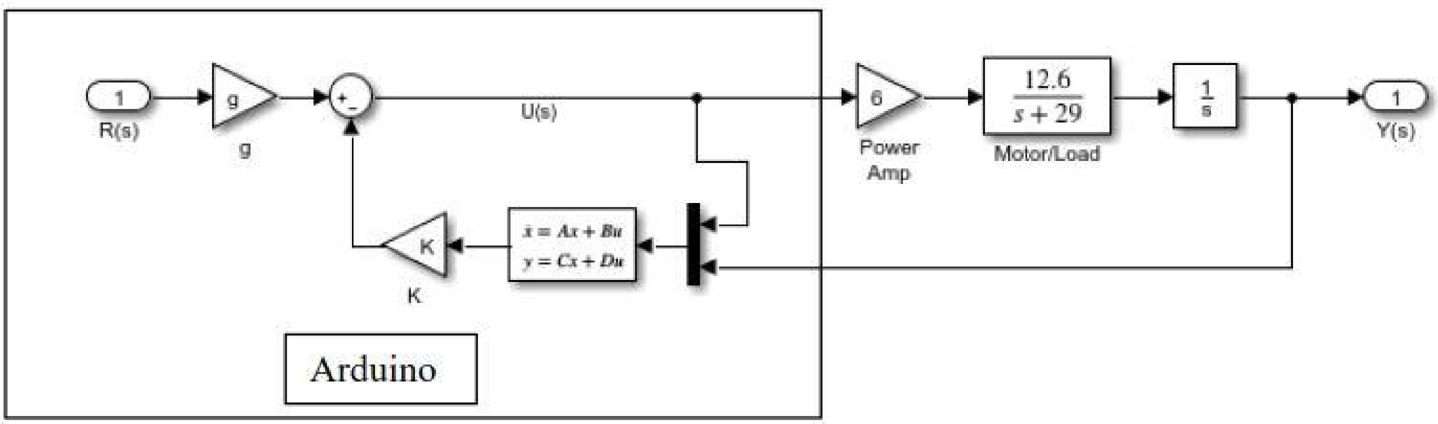
Observer Feedback Control System with Arduino

Develop the Controller

The implementation of observer feedback control with the Arduino. The analog implementation of an observer feedback is fairly complex. Therefore, these systems are typically implemented digitally.

Considering the following system, we will design an observer feedback system for controlling the position of the system. Assume the closed loop state feedback poles at:

-10+/-j20 and the observer poles at -40+/-j40.



Our goal is to find the state-space representation of the controllerwhich we will implement with Arduino microcontroller.

```
s = tf('s');

Plant = 6*12.6/(s^2+29*s)
```

```
Plant =

      75.6
  ----------
  s^2 + 29 s

Continuous-time transfer function.
```

```
state_poles = [-10+1i*20 -10-1i*20]
```

```
state_poles =
 -10.0000 +20.0000i -10.0000 -20.0000i
```

```
obs_poles = [-40+1i*40 -40-1i*40]
```

```
obs_poles =
 -40.0000 +40.0000i -40.0000 -40.0000i
```

```
% SS Model of the Plant
[A,B,C,D] = tf2ss([0 0 75.6],[1 29 0])
```

```
A =

   -29     0
     1     0
B =

     1
     0
C =

         0   75.6000
D = 0
```

```
% Control Gains
K = place(A,B,state_poles)
```

```
K =

   -9.0000  500.0000
```

```
eig(A-B*K)
```

```
ans =
 -10.0000 +20.0000i
 -10.0000 -20.0000i
```

```
% Feedforward Gain
g = -1/(C*inv(A-B*K)*B)
```

```
g = 6.6138
```

Develop the Observer

```
syms s g1 g2;
G = [g1; g2];

eqn = collect(det(s*eye(2)-A+G*C))
```

eqn =

$$s^2 + \left(\frac{378\,g_2}{5} + 29\right) s + \frac{378\,g_1}{5} + \frac{10962\,g_2}{5}$$

```
desired_char = simplify((s-obs_poles(1))*(s-obs_poles(2)))
```

desired_char = $s^2 + 80\,s + 3200$

```
g2 = double(solve(378*g2/5+29==80,g2))
```

```
g2 = 0.6746
```

```
g1 = double(solve(378*g1/5+10962*g2/5==3200,g1))
```

```
g1 = 22.7646
```

```
G = [g1; g2]
```

```
G =

   22.7646
    0.6746
```

```
F = A-G*C
```

```
F =

           -29       -1721
             1         -51
```

```
H = B
```

```
H =

     1
     0
```

```
% sanity check
eig(A-G*C)
```

```
ans =
 -40.0000 +40.0000i
 -40.0000 -40.0000i
```

```
G = place(A',C',obs_poles)'
```

```
G =

   22.7646
    0.6746
```

```
damp(-10+20j)
```

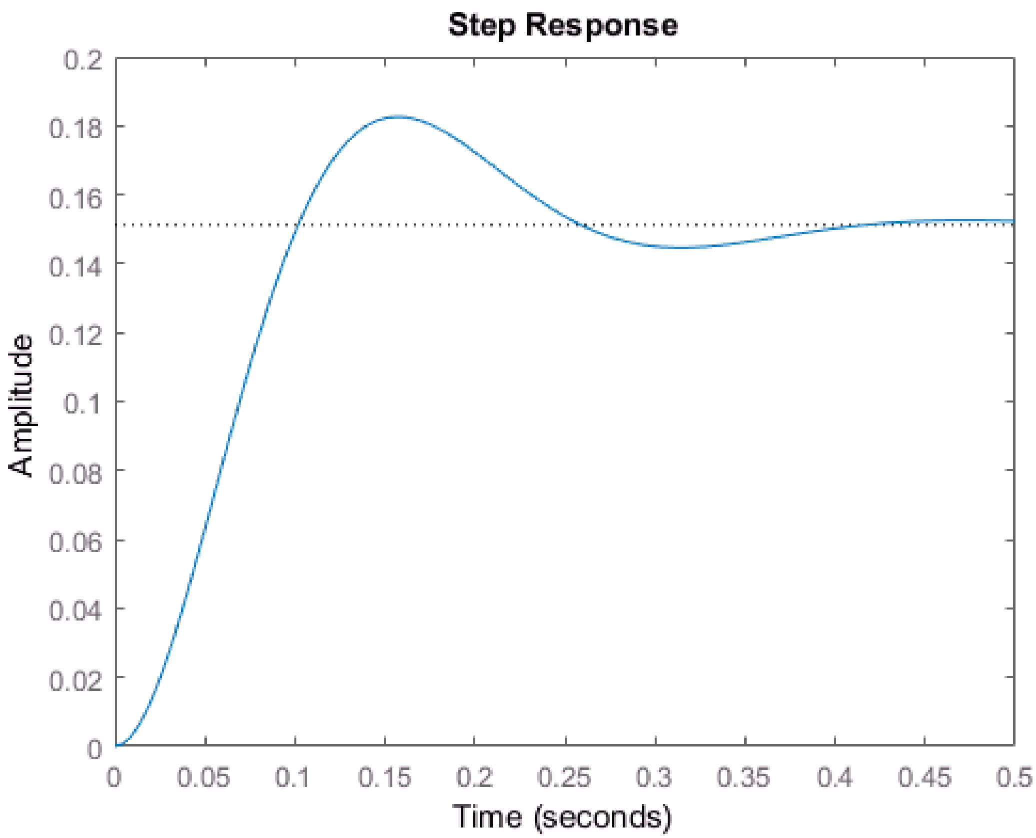
```
         Pole                 Damping       Frequency       Time Constant
                                          (rad/TimeUnit)     (TimeUnit)
```

```
  -1.00e+01 + 2.00e+01i     4.47e-01       2.24e+01          1.00e-01
```

```
% dampig ratio for the poles = 0.447
percent_overshoot = 100*exp(-pi*.447/sqrt(1-.447^2))
```

```
percent_overshoot = 20.8075
```

```
% step response
step(ss(A-B*K,B,C,D))
```




```
stepinfo(ss(A-B*K,B,C,D))
```

```
ans =
        RiseTime: 0.0690
    SettlingTime: 0.3735
     SettlingMin: 0.1383
     SettlingMax: 0.1826
       Overshoot: 20.7866
      Undershoot: 0
            Peak: 0.1826
        PeakTime: 0.1566
```

For the Arduino controller part

```
Ac = A-G*C
```

```
Ac =

   1.0e+03 *
   -0.0290   -1.7210
    0.0010   -0.0510
```

```
Bc = [B G]
```

```
Bc =

    1.0000   22.7646
         0    0.6746
```