PID Example: DC Motor Speed Control

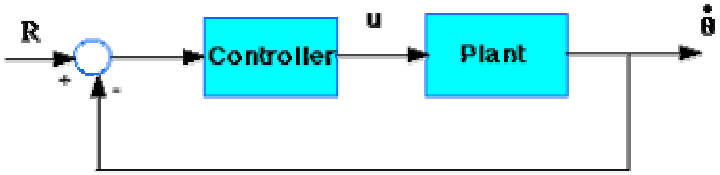
From the main problem, the dynamic equations and the open-loop transfer function of the DC Motor are:

$$s(Js + b)\Theta(s) = KI(s)$$

$$(Ls + R)I(s) = V - Ks\Theta(s)$$

$$\frac{\Theta}{V} = \frac{K}{(Js + b)(Ls + R) + K^2}$$

and the system schematic looks like:



With a 1 rad/sec step input, the design criteria are:

- Settling time less than 2 seconds
- Overshoot less than 5%
- Steady-stage error less than 1%

Now let's design a PID controller and add it into the system. First create a new m-file and type in the following commands

```
J=0.01;
b=0.1;
K=0.01;
R=1;
L=0.5;
num=K;
den=[(J*L) ((J*R)+(L*b)) ((b*R)+K^2)];
```

Recall that the transfer function for a PID controller is:

$$K_P + \frac{K_I}{s} + K_D s = \frac{K_D s^2 + K_P s + K_I}{s}$$

Proportional control

Let's first try using a proportional controller with a gain of 100. Add the following code to the end of your m-file:

```
Kp=100;
numa=Kp*num;
dena=den;
```

To determine the closed-loop transfer function, we use the cloop command. Add the following line to your m-file:

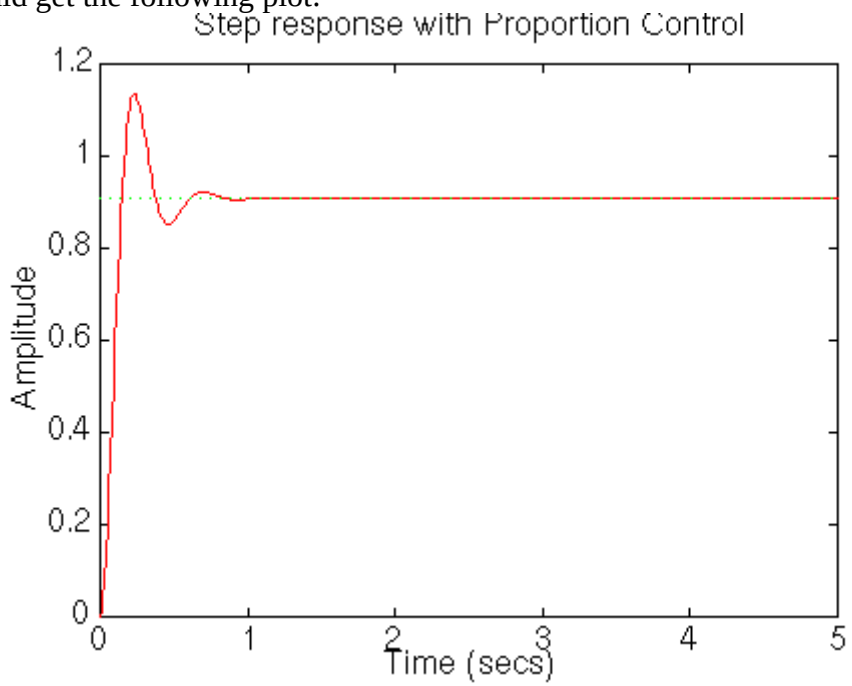
```
[numac,denac]=cloop(numa,dena);
```

Note that `numac` and `denac` are the numerator and the denominator of the overall closed-loop transfer function.

Now let's see how the step response looks, add the following to the end of your m-file, and run it in the command window:

```
t=0:0.01:5;
step (numac,denac,t)
title('Step response with Proportion Control')
```

You should get the following plot:

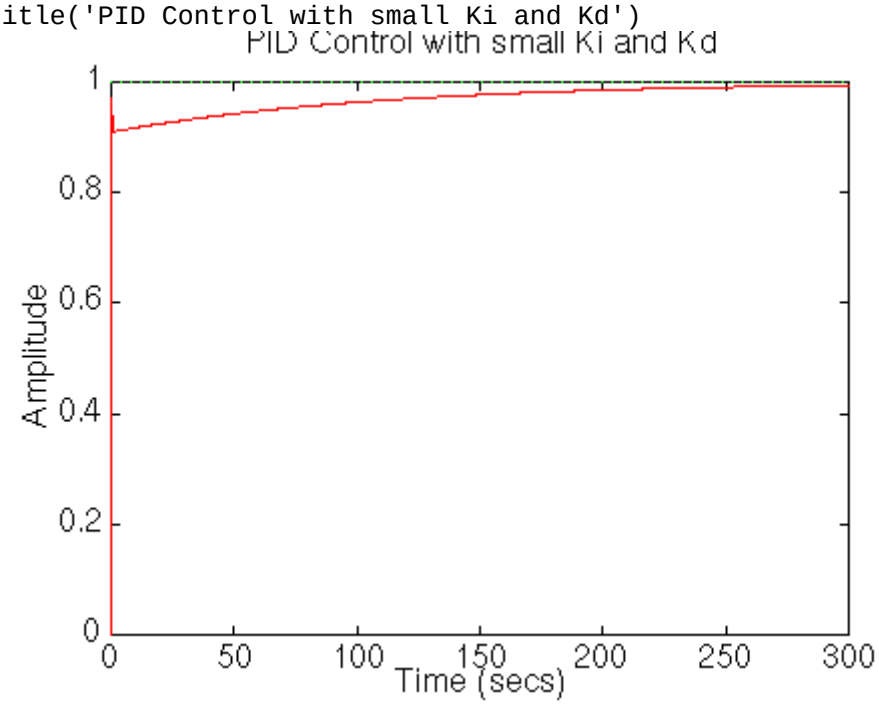


PID control

From the plot above we see that both the steady-state error and the overshoot are too large. Let's try a PID controller with small Ki and Kd. Change your m-file so it looks like the following. Running this new m-file gives you the following plot.

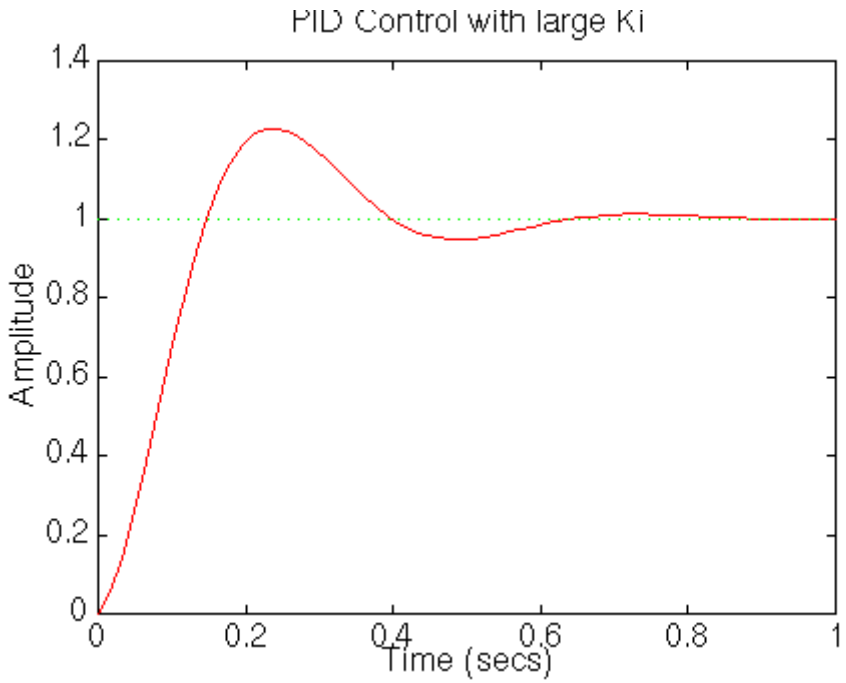
```
J=0.01;
b=0.1;
K=0.01;
R=1;
L=0.5;
num=K;
den=[(J*L) ((J*R)+(L*b)) ((b*R)+K^2)];

Kp=100;
Ki=1;
Kd=1;
numc=[Kd, Kp, Ki];
denc=[1 0];
numa=conv(num,numc);
dena=conv(den,denc);
[numac,denac]=cloop(numa,dena);
step(numac,denac)
title('PID Control with small Ki and Kd')
```

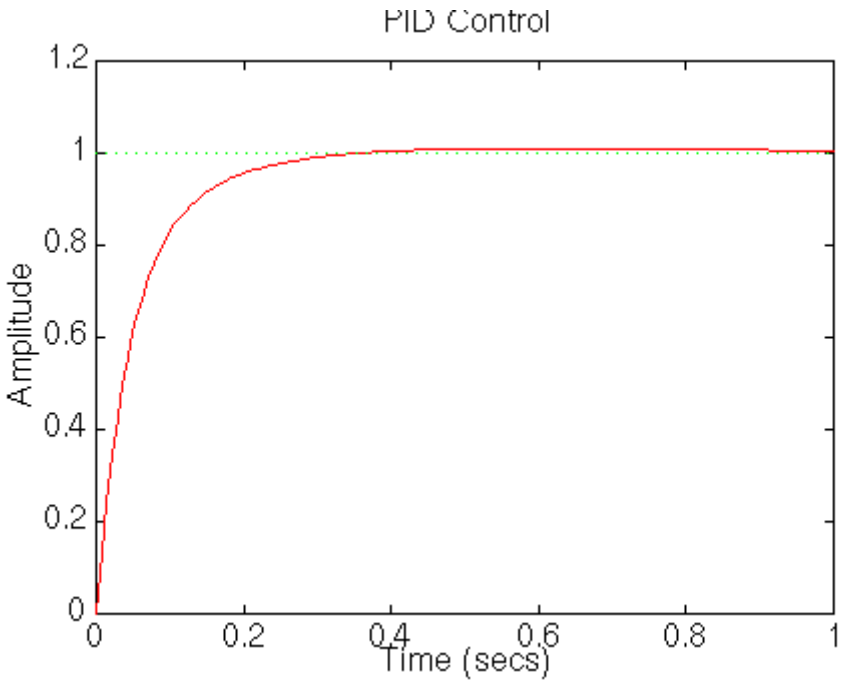


Tuning the gains

Now the settling time is too long. Let's increase K_i to reduce the settling time. Go back to your m-file and change K_i to 200. Rerun the file and you should get the plot like this:



Now we see that the response is much faster than before, but the large Ki has worsened the transient response (big overshoot). Let's increase Kd to reduce the overshoot. Go back to the m-file and change Kd to 10. Rerun it and you should get this plot:



So now we know that if we use a PID controller with

Kp=100,
Ki=200,
Kd=10,

all of our design requirements will be satisfied.