A differential equation view of closed loop control systems

Nasser M. Abbasi

July 2, 2015 Compiled on October 27, 2025 at 12:12am

Contents

1 Proportional controller

Let us consider a closed loop control system with proportional controller k_p and with plant transfer function given by standard second order mass-spring-damper system $G(s) = \frac{1}{ms^2+cs+k}$ where m is the mass and c is the damping coefficient and k is the stiffness coefficient of the spring. In block diagram, the system is

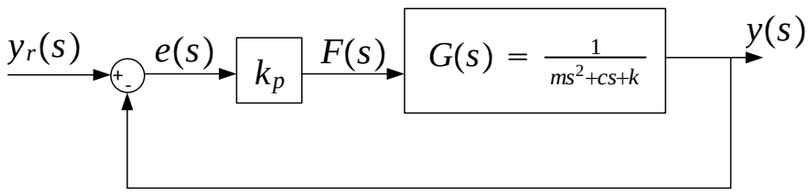

Figure 1: system in block diagram

The error is $e(s) = y_r(s) - y(s)$ where $y_r(s)$ is the reference input that we want the output $y(s)$ to track. We are now interested in finding how adding the controller and closing the loop changes the dynamics of the plant by viewing the changes in the differential equation of the new system. Without the controller and the closed loop, the system was

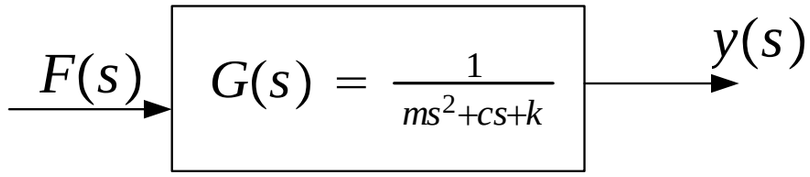

Figure 2: system with no controller and closed loop

In time domain this is represented by the differential equation $my'' + cy' + ky = f(t)$ where $F(s)$ is the Laplace transform of $f(t)$. We now ask, how does this differential equation changes by adding the controller k_p and closing the loop? From the first diagram, $F(s) = e(s)\, k_p$ or $F(s) = (y_r(s) - y(s))\, k_p$, hence the plant now appears as

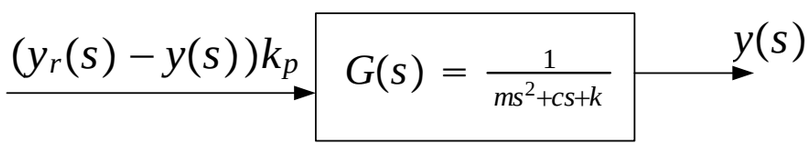

Figure 3: Plant

Hence

$$y(s)\left(ms^2 + cs + k + k_p\right) = y_r(s)\, k_p$$

And in time domain, the differential equation now becomes

$$my'' + cy' + (k + k_p)\, y = \mathcal{L}^{-1}(y_r(s)\, k_p)$$

Considering the case where the reference signal is a unit step (with amplitude that has units of distance or length), then $\mathcal{L}(y_r(s)) = \frac{1}{s}$ and the above reduces to

$$my'' + cy' + (k + k_p)\, y = k_p \tag{1}$$

We see now that the effect of adding a proportional controller k_p is to increase the stiffness of the plant by an amount k_p and also forcing the plant with constant force k_p (actuating force). Increasing the effective stiffness also means the natural frequency ω of the plant will increase, since $\omega = \sqrt{\frac{k_{effective}}{m}}$ and the mass remained the same. A note on the units: The controller k_p has units of Newton per unit length in this case (stiffness coefficient units) and the force k_p will have units of Newton only (since we multiplied it by the unit step reference single which had units of length). Even though the term k_p appears as stiffness on the left side and as force in the right side, it represents only the magnitude, and the units will depend on the context.

It is easy to verify the above. We can connect the system, view the output $y(t)$ of the closed loop for a unit step input $y_r(t)$, and compare it the solution of the differential equation as given by (1). The differential equation is solved using zero initial conditions. We see below that the same result shows up.

```
m = 1; c = 1; k = 20; kp = 400;

plant       = TransferFunctionModel[1/(m s^2 + c s + k), s];
controller  = TransferFunctionModel[kp, s];
sys         = SystemsModelSeriesConnect[plant, controller];
sys         = SystemsModelFeedbackConnect[sys];
o           = OutputResponse[sys, UnitStep[t], {t, 0, 6}];
p1          = Plot[o, {t, 0, 6}, Frame -> True, PlotRange -> All,
                FrameLabel -> {{"y(t)", None}, {"t (sec)",
                "response of closed loop system"}}, BaseStyle -> 14, ImageSize -> 350,
                Epilog -> {Red, Line[{{0, 1}, {10, 1}}]}];

sol = First@DSolve[{m y''[t] + c y'[t] + (k + kp) y[t] == kp,y[0] == 0, y'[0] == 0}, y, t];

p2 = Plot[Evaluate[y[t] /. sol], {t, 0, 6}, Frame -> True,
         PlotRange -> All, FrameLabel -> {{"y(t)", None},
         {"t (sec)", "solution of closed loop differential equation"}},
         BaseStyle -> 14, ImageSize -> 350];

Grid[{{p1, p2}}]
```

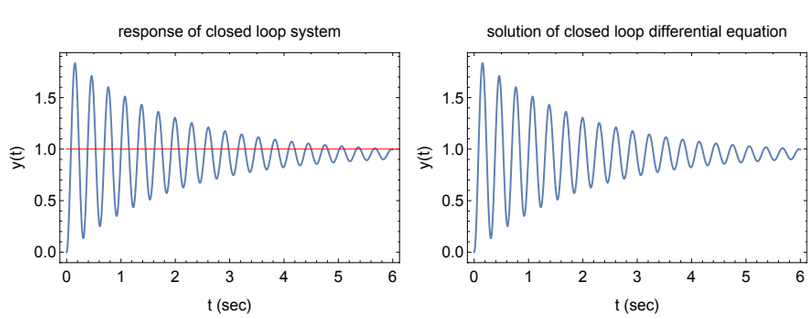

Figure 4: Showing result

The differential equation given by (1) has an analytical solution when the force is constant k_p and assuming the damping ratio is $\xi < 1$ (under-damped), where $\xi = \frac{c}{c_r} = \frac{c}{2\sqrt{km}}$. The analytical solution is given by

$$y(t) = e^{-\xi\omega t}\left(\left(-\frac{k_p}{k+k_p}\right)\cos\omega_d t - \frac{k_p}{k+k_p}\xi\frac{\omega}{\omega_d}\sin\omega_d t\right) + \frac{k_p}{k+k_p}$$

Where ω_d is the damped natural frequency $\omega_d = \omega\sqrt{1-\xi^2}$ and ω is the undamped natural frequency $\omega = \sqrt{\frac{k+k_p}{m}}$. We see now that as $\lim_{t\to\infty} y(t) = \frac{k_p}{k+k_p}$ and hence the steady state solution will never reach unity (which is the reference signal in this example) but will get closer to it as k_p is made larger and larger. This is the reason why response to a unit step using a proportional controller will always have a steady state error given by $1 - \frac{k_p}{k+k_p}$ where k_p is the controller gain and k is the plant original stiffness coefficient. The term $\frac{k_p}{k+k_p}$ is called the static deflection. The final steady state can also be found in Laplace domain using final value theorem as follows.

$$\begin{aligned}
F(s)\frac{1}{ms^2+cs+k} &= y(s)\\
\frac{(y_r(s)-y(s))\,k_p}{ms^2+cs+k} &= y(s)\\
y(s)\left(ms^2+cs+k\right) &= y_r(s)\,k_p - y(s)\,k_p\\
y(s)\left(ms^2+cs+k+k_p\right) &= y_r(s)\,k_p\\
y(s) &= \frac{k_p}{ms^2+cs+k+k_p}y_r(s)
\end{aligned}$$

Assuming $y_r(s)$ is unit step, then

$$y(s) = \frac{\frac{k_p}{s}}{ms^2+cs+k+k_p}$$

And using final value theorem

$$\begin{aligned}
y(\infty) &= \lim_{s\to 0} sy(s)\\
&= \lim_{s\to 0}\frac{k_p}{ms^2+cs+k+k_p}\\
&= \frac{k_p}{k+k_p}
\end{aligned}$$

Which is the static deflection from the analytical solution of the differential equation found above.

2 PID controller

Let us now look at the dynamics of the plant when the controller is a PID given by $k_p + k_D s + k_i\frac{1}{s}$. The block diagram of the system is

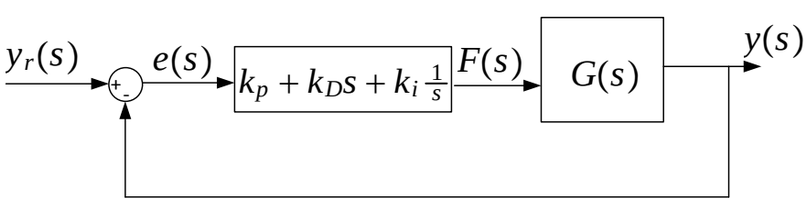

Figure 5: Block diagram with PID

The error is $e(s) = y_r(s) - y(s)$ where $y_r(s)$ is the reference input that we want the output $y(s)$ to track. As was done for the case of the proportional controller, the plant now appears as

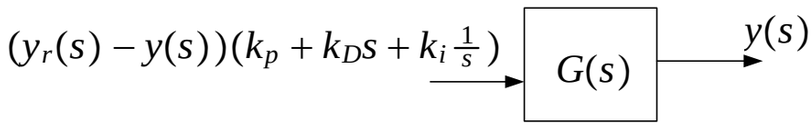

Figure 6: Plant diagram

Hence

$$y(s)\left(ms^2 + cs + k\right) = \left(y_r(s) - y(s)\right)\left(k_p + k_D s + k_i \frac{1}{s}\right)$$

$$y(s)\left(ms^2 + cs + k + k_p + k_D s + k_i \frac{1}{s}\right) = y_r(s)\left(k_p + k_D s + k_i \frac{1}{s}\right)$$

And in time domain, the differential equation now becomes

$$my'' + (c + k_D)\, y' + (k + k_p)\, y + k_i \int_0^t y(\tau)\, d\tau = k_p y_r + k_D y_r' + k_i \int_0^t y_r(\tau)\, d\tau$$

For the special case when the reference signal is a unit step

$$my'' + (c + k_D)\, y' + (k + k_p)\, y + k_i \int_0^t y(\tau)\, d\tau = k_p + k_i \int_0^t y_r(\tau)\, d\tau$$

$$my'' + (c + k_D)\, y' + (k + k_p)\, y = k_p + k_i \int_0^t \left(y_r(\tau) - y(\tau)\right) d\tau$$

The term $\int_0^t \left(y_r(\tau) - y(\tau)\right) d\tau$ is the integral of the error $e(t)$. When $y_r(t)$ is unit step the above becomes

$$\begin{aligned} my'' + (c + k_D)\, y' + (k + k_p)\, y &= k_p + k_i \int_0^t \left(1 - y(\tau)\right) d\tau \\ &= k_p + k_i t - k_i \int_0^t y(\tau)\, d\tau \end{aligned}$$

Therefore, using a PID controller has the effect of making the system more damped, since the effective damping coefficient has now become $c + k_D$ compared to just c before, and making the system more stiff by adding k_p to the stiffness coefficient. The actuating force has two components (for the special case of unit step reference), which is constant force of k_p and a force which is proportional to the error: $k_i \int_0^t \left(1 - y(\tau)\right) d\tau$.

PID controller allows the steady state to become zero. This can be seen by using final value theorem in Laplace domain as follows. The transfer function can be found in Laplace domain using final value theorem as follows.

$$F(s)\frac{1}{ms^2 + cs + k} = y(s)$$

$$\frac{\left(y_r(s) - y(s)\right)\left(k_p + k_D s + k_i \frac{1}{s}\right)}{ms^2 + cs + k} = y(s)$$

$$y(s)\left(ms^2 + cs + k\right) = \left(y_r(s) - y(s)\right)\left(k_p + k_D s + k_i \frac{1}{s}\right)$$

$$y(s)\left(ms^2 + cs + k + k_p + k_D s + k_i \frac{1}{s}\right) = y_r(s)\left(k_p + k_D s + k_i \frac{1}{s}\right)$$

$$y(s) = \frac{\left(k_p + k_D s + k_i \frac{1}{s}\right)}{\left(ms^2 + cs + k + k_p + k_D s + k_i \frac{1}{s}\right)} y_r(s)$$

Assuming $y_r(s)$ is unit step, then

$$y(s) = \frac{\left(k_p + k_D s + k_i \frac{1}{s}\right)}{\left(ms^2 + cs + k + k_p + k_D s + k_i \frac{1}{s}\right)} \frac{1}{s}$$

And using final value theorem

$$\begin{aligned} y(\infty) &= \lim_{s \to 0} s y(s) \\ &= \lim_{s \to 0} \frac{\left(k_p + k_D s + k_i \frac{1}{s}\right)}{\left(ms^2 + cs + k + k_p + k_D s + k_i \frac{1}{s}\right)} \\ &= \lim_{s \to 0} \frac{sk_p + k_D s^2 + k_i}{\left(ms^3 + cs^2 + ks + k_p s + k_D s^2 + k_i\right)} \\ &= \frac{k_i}{k_i} \\ &= 1 \end{aligned}$$

Hence $y(\infty)$ is a unit step. So using PID controller it was possible to achieve the same value as the desired tracking signal. Hence the steady state error is zero.

Plot of the step response of the above is given below. k_i was made large to force the steady state error to go to zero in about 6 seconds.

```
m = 1; c = 1; k = 20; kp = 400; kd = 1; ki = 200;

plant      = TransferFunctionModel[1/(m s^2 + c s + k), s];
controller = TransferFunctionModel[kp + kd s + ki 1/s, s];
sys        = SystemsModelSeriesConnect[plant, controller];
sys        = SystemsModelFeedbackConnect[sys];
o          = OutputResponse[sys, UnitStep[t], {t, 0, 6}];
p1         = Plot[o, {t, 0, 6}, Frame -> True, PlotRange -> All,
               FrameLabel -> {{"y(t)", None}, {"t (sec)",
               "response of closed loop system, PID controller"}},
               BaseStyle -> 14, ImageSize -> 400,
               Epilog -> {Red, Line[{{0, 1}, {10, 1}}]}
               ]
```

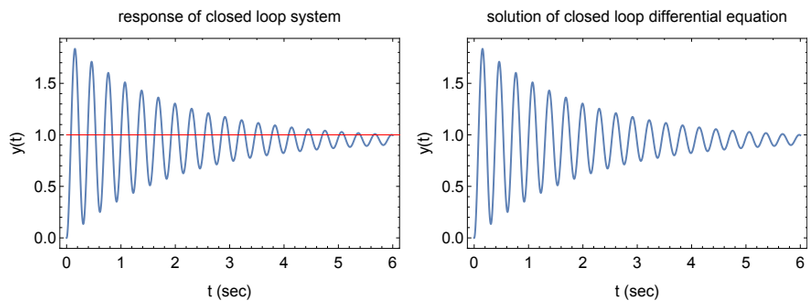

Figure 7: Result from above