



Smart Throttle Control

Control Techniques-1
(3)



Tracking

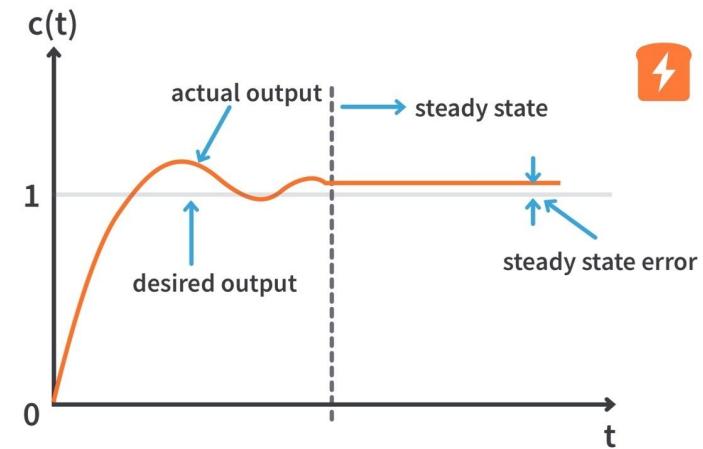
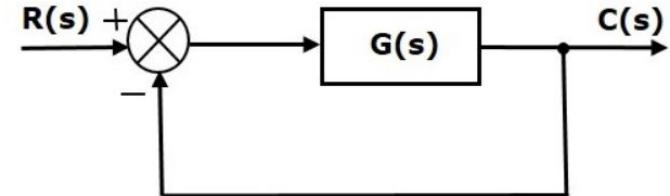
For unity feedback

$$e_{ss} = \lim_{t \rightarrow \infty} e(t) = \lim_{s \rightarrow 0} sE(s)$$

$$E(s) = \frac{R(s)}{1 + G(s)}$$

$$e_{ss} = \lim_{s \rightarrow 0} \frac{sR(s)}{1 + G(s)}$$

Goal is to make $G(s)$ such that e_{ss} is minimum





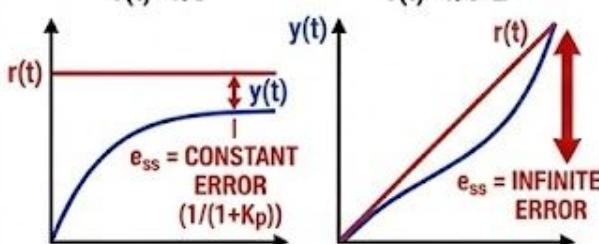
System Type and e_{ss}

TYPE 0 (N=0 Integrators)

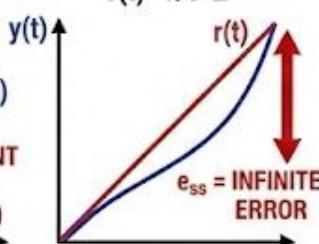
$$\text{Open-Loop TF: } G(s) = \frac{K}{s + a}$$

ERROR RESPONSE:

Step Input
 $r(t)=1/s$



Ramp Input
 $r(t)=1/s^2$



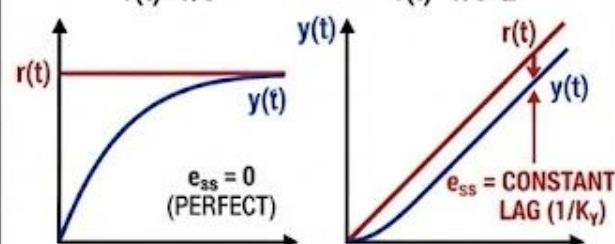
RESULT: Fails to track Step perfectly.
Fails to track Ramp completely.

TYPE 1 (N=1 Integrator)

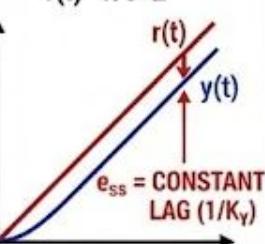
$$\text{Open-Loop TF: } G(s) = \frac{K}{s(s + a)}$$

ERROR RESPONSE:

Step Input
 $r(t)=1/s$



Ramp Input
 $r(t)=1/s^2$



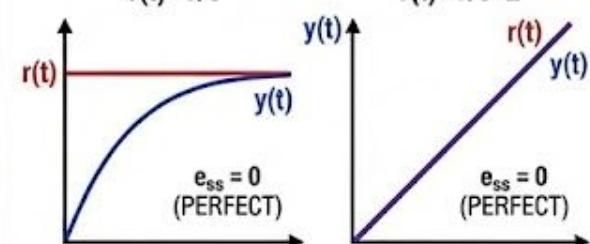
RESULT: Tracks Step perfectly.
Tracks Ramp with constant lag.

TYPE 2 (N=2 Integrators)

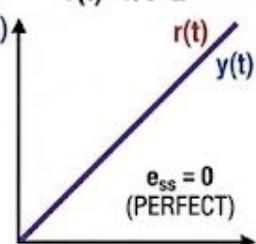
$$\text{Open-Loop TF: } G(s) = \frac{K}{s^2(s + a)}$$

ERROR RESPONSE:

Step Input
 $r(t)=1/s$



Ramp Input
 $r(t)=1/s^2$



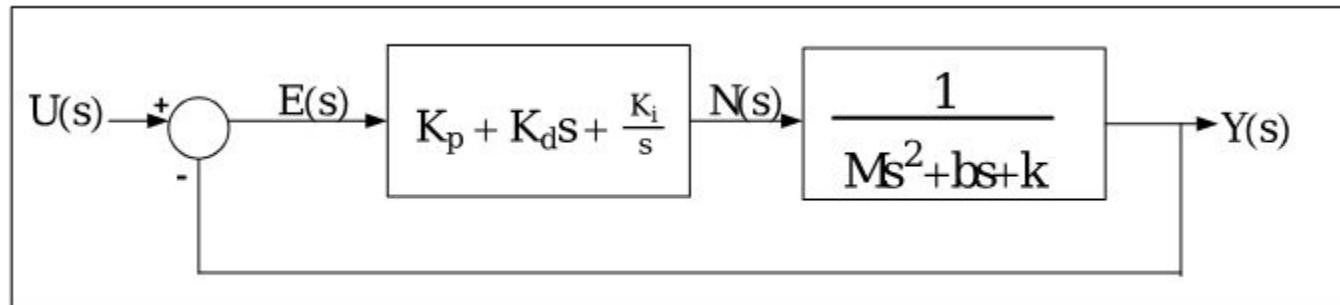
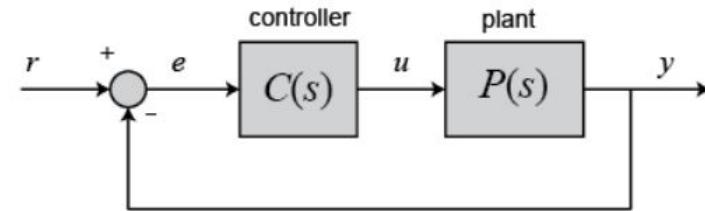
RESULT: Tracks both Step and Ramp perfectly.



What is a PID

It corrects system error by balancing reaction (Proportional), memory (Integral), and anticipation (Derivative).

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{d}{dt} e(t)$$



System Block diagram with a PID controller



Proportional Error Constant (K_p)

How K_p changes response

Fast Response: K_p immediately reacts to error changes. A higher K_p leads to a faster rise time

Stability vs. Oscillation: Increasing K_p improves speed but can introduce oscillations and overshoots, potentially leading to instability if K_p is too high.

Steady-State Error: A P-controller alone often results in a non-zero steady-state error, as the control output becomes zero when the error is zero.



Integral Error Constant (Ki)

How Ki changes response

Eliminates Steady-State Error: Ki ensures that if any error persists over time, ultimately driving the steady-state error to zero.

Stability & Overshoot: While Ki improves accuracy, it can increase overshoot and settling time. A high Ki may lead to oscillations or even system instability.

Integral Windup: In systems with actuator saturation, the integral term can accumulate excessively, causing delayed or poor response. Therefore, **anti-windup** techniques are often required.

The diagram illustrates a feedback control system. A reference input (a step function) enters a summing junction. The output of this junction is fed into a controller block, which contains the integral windup formula: $\frac{1}{s} \left[K_i E(s) + \frac{1}{T_t} (U_{sat}(s) - U_{calc}(s)) \right]$. The controller's output is then compared with a saturation limit (U_{sat}) to produce the actual control signal (U_{calc}). A dashed red line highlights the integral term in the controller, emphasizing its role in accumulating error over time.



Derivative Error Constant (K_d)

How K_d changes response

Damping & Stability: K_d introduces damping into the system, which reduces oscillations and overshoot, making the system more stable—especially for systems with high inertia.

Faster Settling Time: By anticipating changes in error, the derivative term helps the system reach the setpoint faster, thereby reducing settling time.

Noise Sensitivity: The derivative action is highly sensitive to noise in the error signal because it amplifies high-frequency components. This can cause erratic control action, so **filtering** is usually necessary.

The diagram illustrates the effect of the derivative error constant. On the left, a grey arrow represents a step input signal. On the right, a red dashed box encloses a mathematical expression: $\frac{K_d \cdot s}{1 + \alpha \cdot K_d \cdot s}$. This expression represents the derivative component of the system's response, where K_d is the derivative error constant and s is the complex frequency variable. The red dashed box highlights this term, indicating its role in shaping the system's transient response.



PID Controller

| CL RESPONSE | RISE TIME | OVERSHOOT | SETTLING TIME | S-S ERROR |
|-------------|--------------|-----------|---------------|-----------|
| K_p | Decrease | Increase | Small Change | Decrease |
| K_i | Decrease | Increase | Increase | Decrease |
| K_d | Small Change | Decrease | Decrease | No Change |

Various techniques can be followed to get a set characteristics out of a PID by changing parameters (tuning the PID controller)

Ziegler-Nichols !!!

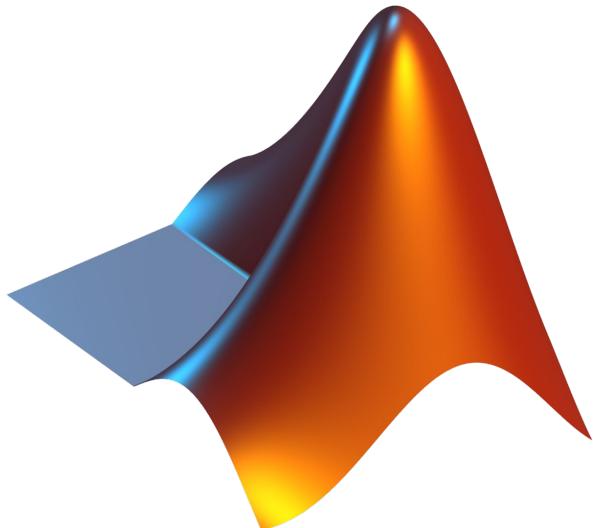
Moving on to an example:

<https://ctms.engin.umich.edu/CTMS/index.php?example=Introduction§ion=ControlPID>



PID Using Matlab

Moving on to



Thank You !!!