ECE 486: Control Systems

Lecture 8B: Proportional-Derivative (PD) Control

Key Takeaways

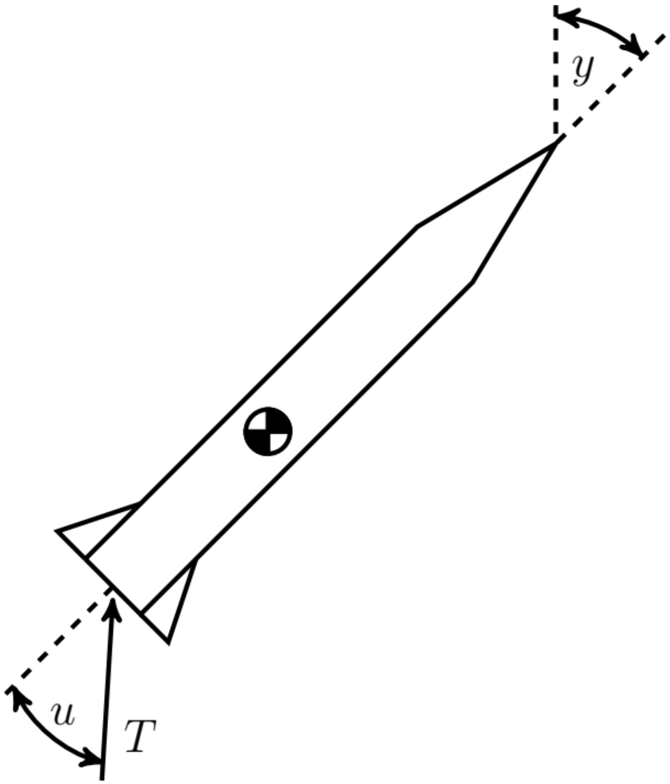
This lecture describes proportional-derivative control. The controller sets the plant input with two terms: (i) proportional to the error and (ii) proportional to the derivative of the error.

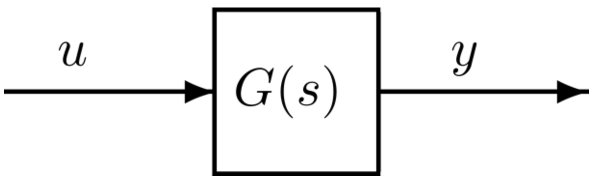
Key properties of PD control:

1. Some plants cannot be stabilized by P or PI control. This motivates the use of PD.
2. A basic implementation of PD control will amplify noise.
3. Common implementations use a “smoothed” derivative or a direct measurement of the derivative of the output.

Rocket Attitude Control

Rockets require precise control of their heading direction (attitude) to reach their desired final destination.



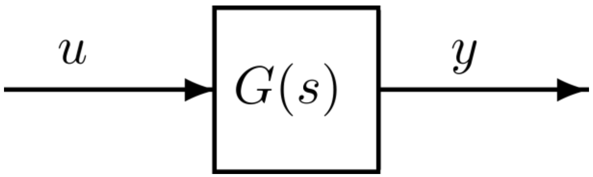


u:= Thrust angle (rad)

y:= Heading angle (rad)

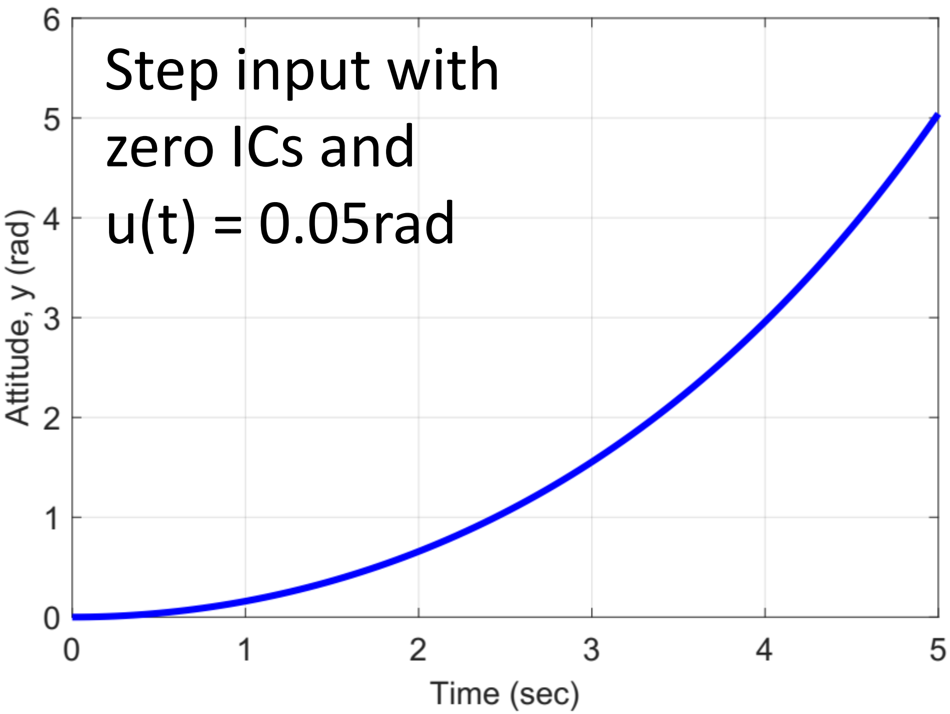
Model of Rocket Attitude Dynamics

If $|u| \ll 1$ and $|y| \ll 1$ then the dynamics are approximated by:



$$\ddot{y}(t) + a_1\dot{y}(t) + a_0 y(t) = b_0\, u(t) + b_0\, d(t)$$

$$\text{where: } a_1 = 0\frac{1}{sec},\ a_0 = -0.12\frac{1}{sec^2},\ \text{and } b_0 = 6.32\frac{1}{sec^2}$$



Transfer Function:

$$G(s) = \frac{6.32}{s^2 - 0.12}$$

Poles:

$$s_{1,2} = \pm 0.346\frac{rad}{sec}$$

System is unstable

Proportional Control

Model of rocket attitude:

$$\ddot{y}(t) + a_1\dot{y}(t) + a_0 y(t) = b_0\, u(t) + b_0\, d(t)$$

$$\text{where:} \quad a_1 = 0\frac{1}{sec}, \;\; a_0 = -0.12\frac{1}{sec^2}, \;\; \text{and} \;\; b_0 = 6.32\frac{1}{sec^2}$$

Sub $u = K_p(r - y)$ into plant model:

$$\ddot{y}(t) + \underbrace{a_1}_{2\zeta\omega_n} \dot{y}(t) + \underbrace{(a_0 + b_0 K_p)}_{\omega_n^2} y(t) = (b_0 K_p)\, r(t) + b_0\, d(t)$$

The coefficient of $\dot{y}$ is = 0 and is unaffected by K_p. The closed-loop will be unstable.

The rocket dynamics cannot be stabilized by P-control. Moreover it cannot be stabilized by PI-control (Routh-Hurwitz criterion can be applied to show this).

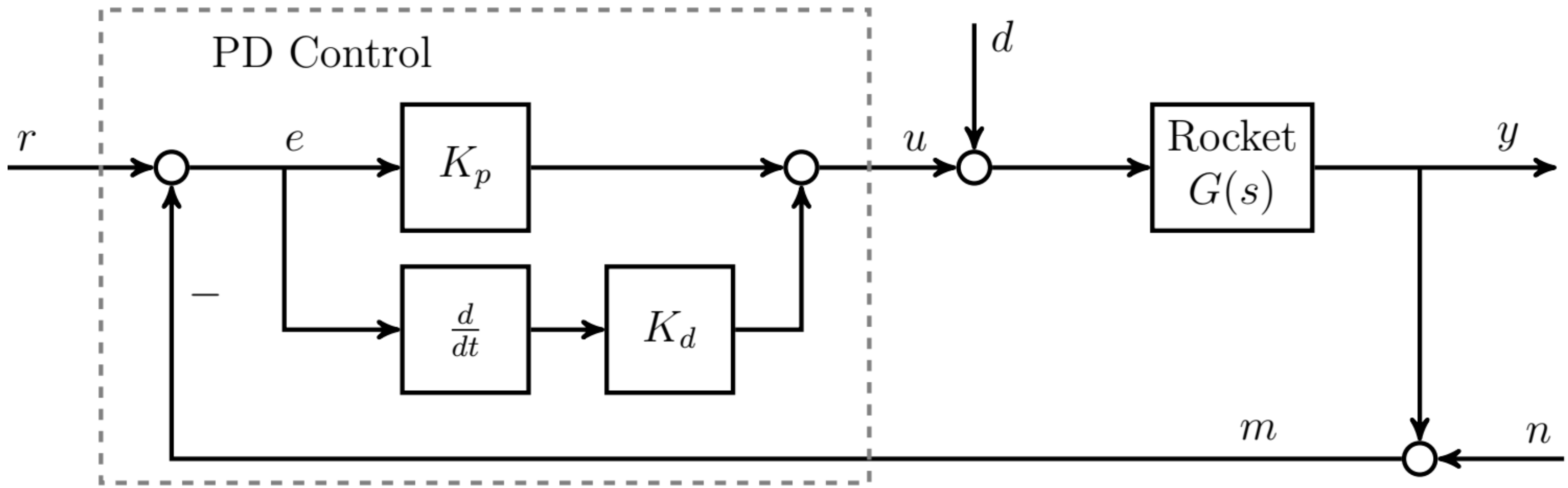
Proportional-Derivative (PD) Control

Closed-loop, proportional-derivative control for rocket:

1. User specifies the desired heading angle, *r(t)*
2. Controller computes the tracking error *e(t) = r(t)-y(t)*
3. Controller sets input thrust angle to:

$$u(t) = K_p e(t) + K_d \dot{e}(t)$$

where K_p and K_d are gains to be selected.



Effect of P and D Terms

P Control: $u(t) = K_p e(t)$

K_p affects settling time, steady-state error, control input

PD Control: $u(t) = \boxed{K_p e(t)} + \boxed{K_d \dot{e}(t)}$

Use two gains to independently modify the transient and steady-state characteristics:

P-Term: Reacts to present (current error).

D-Term: Reacts to future (derivative of error), i.e. $\dot{e}$ indicates the direction the error is headed. Has no effect in steady-state.

Model for Closed-Loop Control

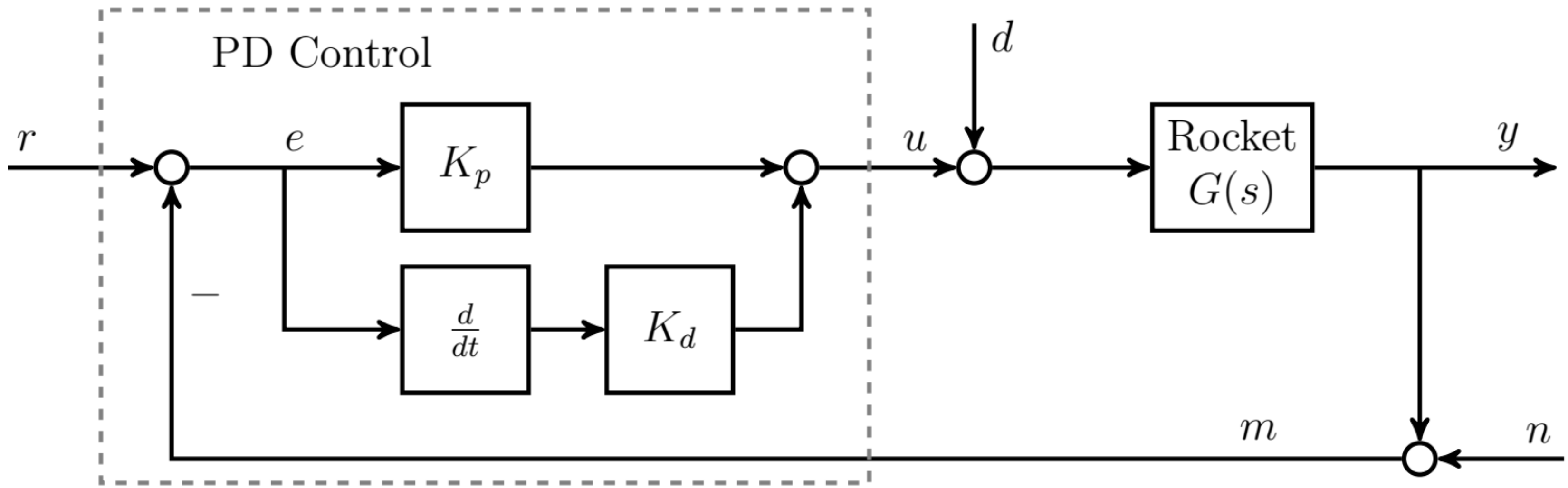
Recall the second-order model for the rocket:

$$\ddot{y}(t) + a_1\dot{y}(t) + a_0 y(t) = b_0\, u(t) + b_0\, d(t)$$

Substitute $u = K_p e + K_d\, \dot{e}$ and combine terms:

$$\ddot{y}(t) + \underbrace{(a_1 + b_0 K_d)}_{:=2\zeta\omega_n}\, \dot{y}(t) + \underbrace{(a_0 + b_0 K_p)}_{:=\omega_n^2}\, y(t) = (b_0 K_d)\, \dot{r}(t) + (b_0 K_p)\, r(t) + b_0 d(t)$$

This is a second-order closed-loop model from *(r,d)* to *y*.



Closed-Loop Response

The dynamics of the closed-loop system are:

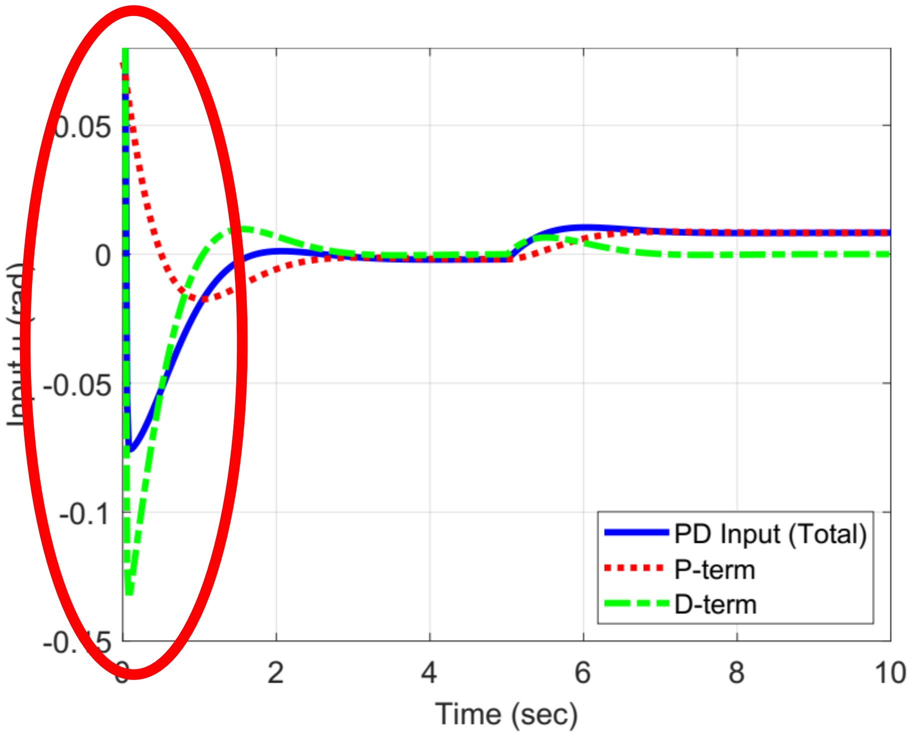
$$\ddot{y}(t) + \underbrace{(a_1 + b_0 K_d)}_{:=2\zeta\omega_n} \dot{y}(t) + \underbrace{(a_0 + b_0 K_p)}_{:=\omega_n^2} y(t) = (b_0 K_d)\, \dot{r}(t) + (b_0 K_p)\, r(t) + b_0 d(t)$$

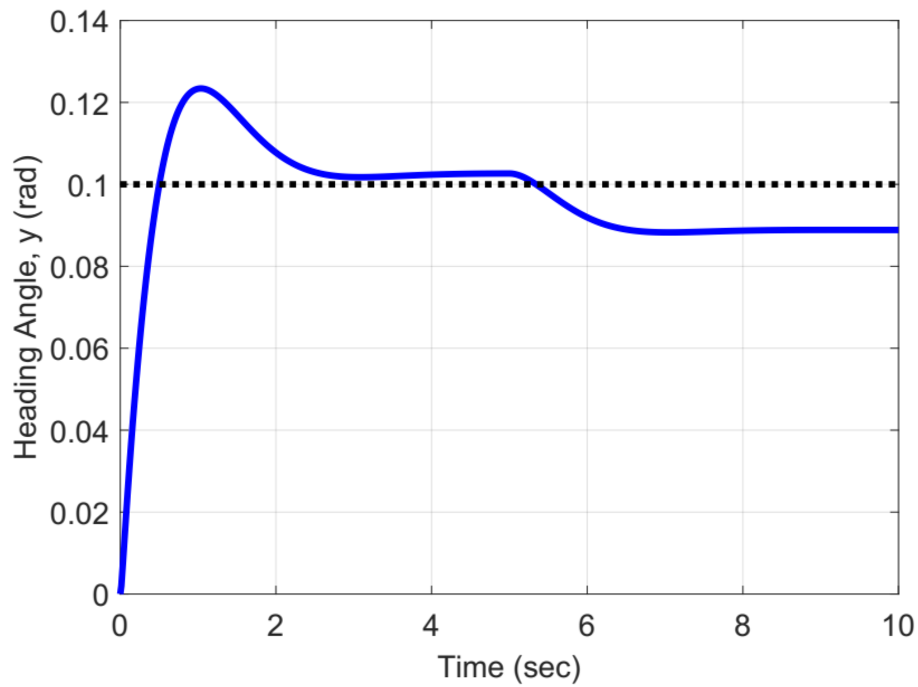
- Closed-loop is stable if and only if $a_0+b_0K_p>0$, $a_1+b_0K_d>0$.
- We can place the two closed-loop poles anywhere by proper choice of (K_p,K_d). [Always true if plant is 2^{nd} order.]
- We are able to use the derivative term to modify the damping and stabilize the rocket attitude dynamics.

Example of PD Control

- Simulate with gains $(K_p, K_d) = (0.75, 0.47)$ and
  - $r(t) = 0.1 rad$,
  - $d(t) = -0.01 rad$ for $t \geq 5 sec$
- Closed-loop is underdamped with:

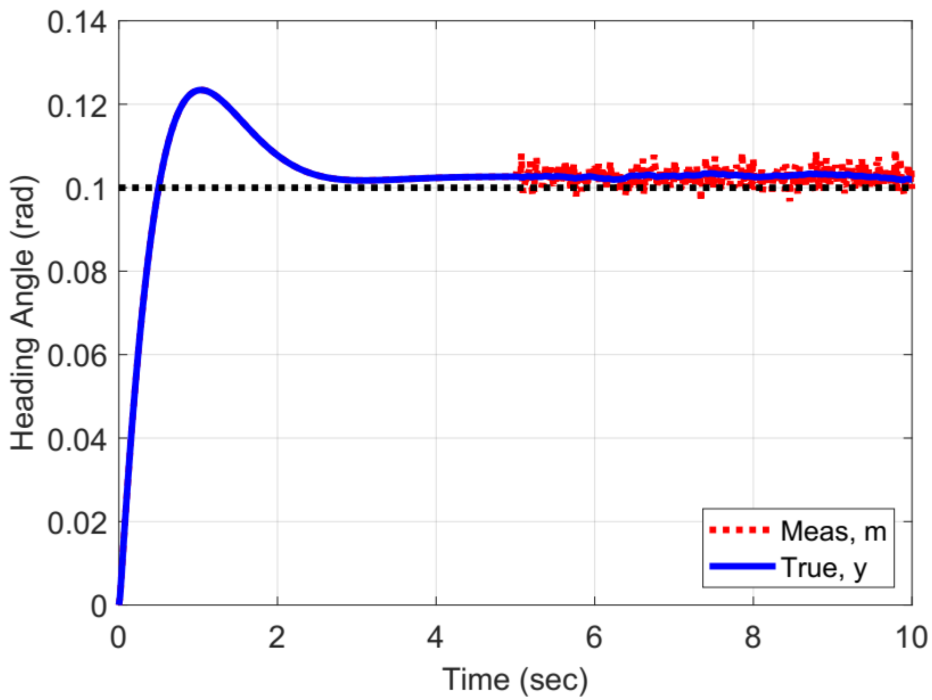
$$\omega_n = \sqrt{a_0 + b_0 K_p} = 2.14 sec \quad \text{and} \quad \zeta = \frac{a_1 + b_0 K_d}{2\sqrt{a_0 + b_0 K_p}} = 0.7$$





Effect of Noise

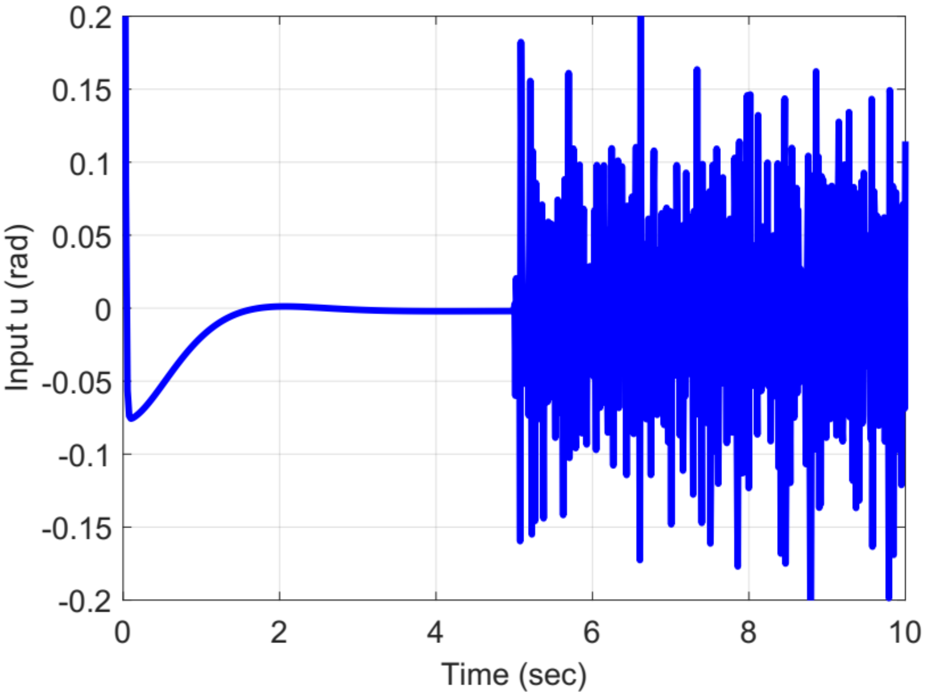
- Simulate with gains $(K_p, K_d) = (0.75, 0.47)$ and
  - $r(t) = 0.1rad,$
  - Sensor noise $n(t)$ for $t \geq 5sec$ [Zero mean, Standard Dev=0.005]

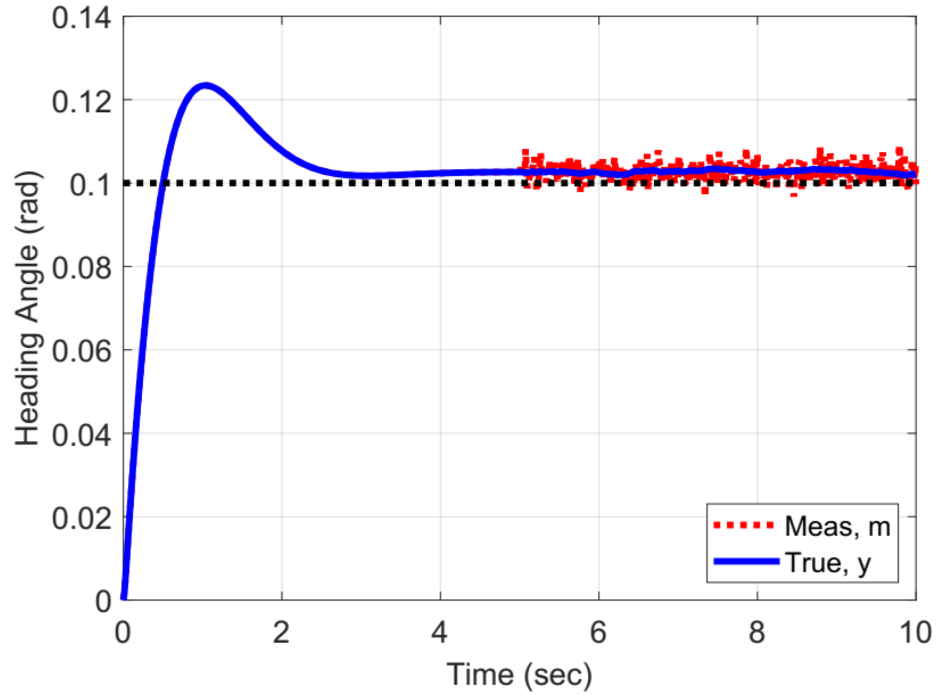


Effect of Noise

- Simulate with gains $(K_p, K_d) = (0.75, 0.47)$ and
  - $r(t) = 0.1 rad$,
  - Sensor noise $n(t)$ for $t \geq 5 sec$ [Zero mean, Standard Dev=0.005]

Derivative control can lead to large control inputs due to fast changes in the reference command or sensor noise.





Implementations for PD Control

1. Use $K_d v$ where v is an approximate (smoothed) derivative:

$$\dot{\hat{e}}(t) + \alpha_0 \hat{e}(t) = \alpha_0 e(t) \text{ with IC: } \hat{e}(0) = 0$$

$$v(t) = \dot{\hat{e}}(t)$$

2. Rate-feedback implementation:

$$u(t) = K_p \left(r(t) - y(t) \right) - K_d \dot{y}(t)$$

This form avoids differentiating the reference. It typically uses a direct measurement of $\dot{y}$.