Basics of Control

Based on slides by Benjamin Kuipers

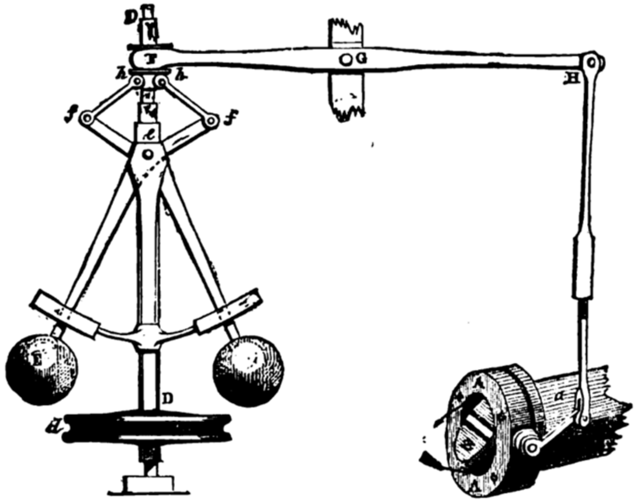
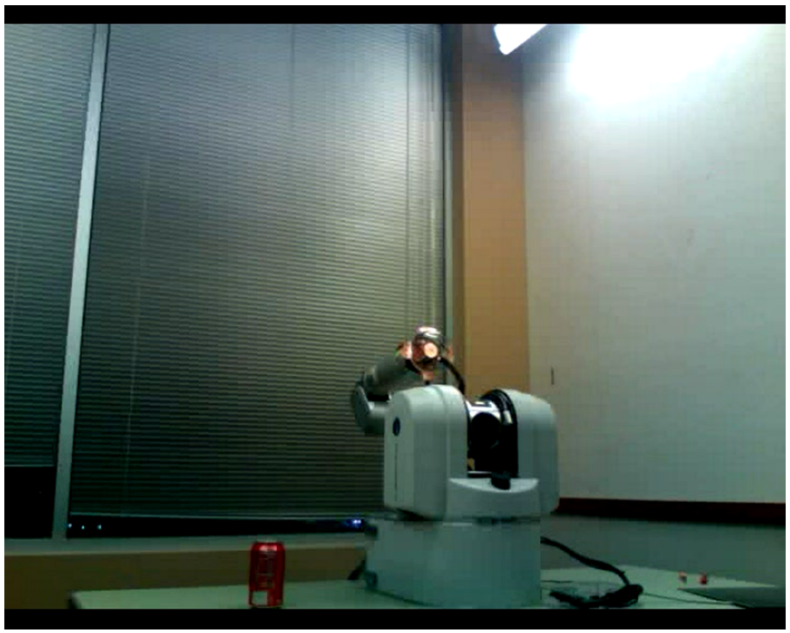


FIG. 4.—*Governor and Throttle-Valve.*

- How can an information system (like a micro-CONTROLLER, a fly-ball governor, or your brain) control the physical world?
- Examples:
  - Thermostat
  - You, walking down the street without falling over
  - A robot trying to keep a joint at a particular angle
  - A blimp trying to maintain a particular heading despite air movement in the room
  - A robot finger trying to maintain a particular distance from an object



Controlling a Simple System

- Consider a simple system: $\dot{x} = F(x, u)$

  Change in state → $\dot{x}$; "is a fn of" → F; State → x; Action → u

  - Scalar variables x and u, not vectors $\mathbf{x}$ and $\mathbf{u}$.
  - Assume effect of motor command u: $\frac{\partial F}{\partial u} > 0$
- The setpoint x_{set} is the desired value.
  - The controller responds to error: $e = x - x_{set}$
- The goal is to set u to reach $e = 0$.

The intuition behind control

- Use action *u* to push back toward error *e* = 0
- What does pushing back do?
  - Position vs velocity versus acceleration control
- How much should we push back?
  - What does the magnitude of *u* depend on?

Velocity or acceleration control?

- Velocity:

$$\mathbf{x} = (x)$$

$$\dot{\mathbf{x}} = (\dot{x}) = F(\mathbf{x}, \mathbf{u}) = (u)$$

- Acceleration:

$$\mathbf{x} = \begin{pmatrix} x \\ v \end{pmatrix}$$

$$\dot{\mathbf{x}} = \begin{pmatrix} \dot{x} \\ \dot{v} \end{pmatrix} = F(\mathbf{x}, \mathbf{u}) = \begin{pmatrix} v \\ u \end{pmatrix}$$

$$\dot{v} = \ddot{x} = u$$

Laws of Motion in Physics

- Newton's Law: *F=ma* or *a=F/m.*

$$\dot{\mathbf{x}} = \begin{pmatrix} \dot{x} \\ \dot{v} \end{pmatrix} = \begin{pmatrix} v \\ F/m \end{pmatrix}$$

- But Aristotle said:
  - *Velocity,* not acceleration, is proportional to the force on a body.
  - True in a friction-dominated setting

The Bang-Bang Controller

- Push back, against the *direction* of the error

$$e = x - x_{set}$$

- Error:

$$e < 0 \quad \Rightarrow \quad u := on \quad \Rightarrow \quad \dot{x} = F(x, on) > 0$$

$$e > 0 \quad \Rightarrow \quad u := off \quad \Rightarrow \quad \dot{x} = F(x, off) < 0$$

- To prevent chatter around $e = 0$

$$e < -\varepsilon \quad \Rightarrow \quad u := on$$

$$e > +\varepsilon \quad \Rightarrow \quad u := off$$

i.e., use small hysteresis ε, instead of 0 as threshold

- Household thermostat. Simple but effective.
- PWM!

Bang-Bang Control

Here, error is

$$e = x_{human} - x_{robot}$$

in some region close to the robot

Proportional Control

- Push back, *proportional* to the error.

$$u = -ke + u_b$$

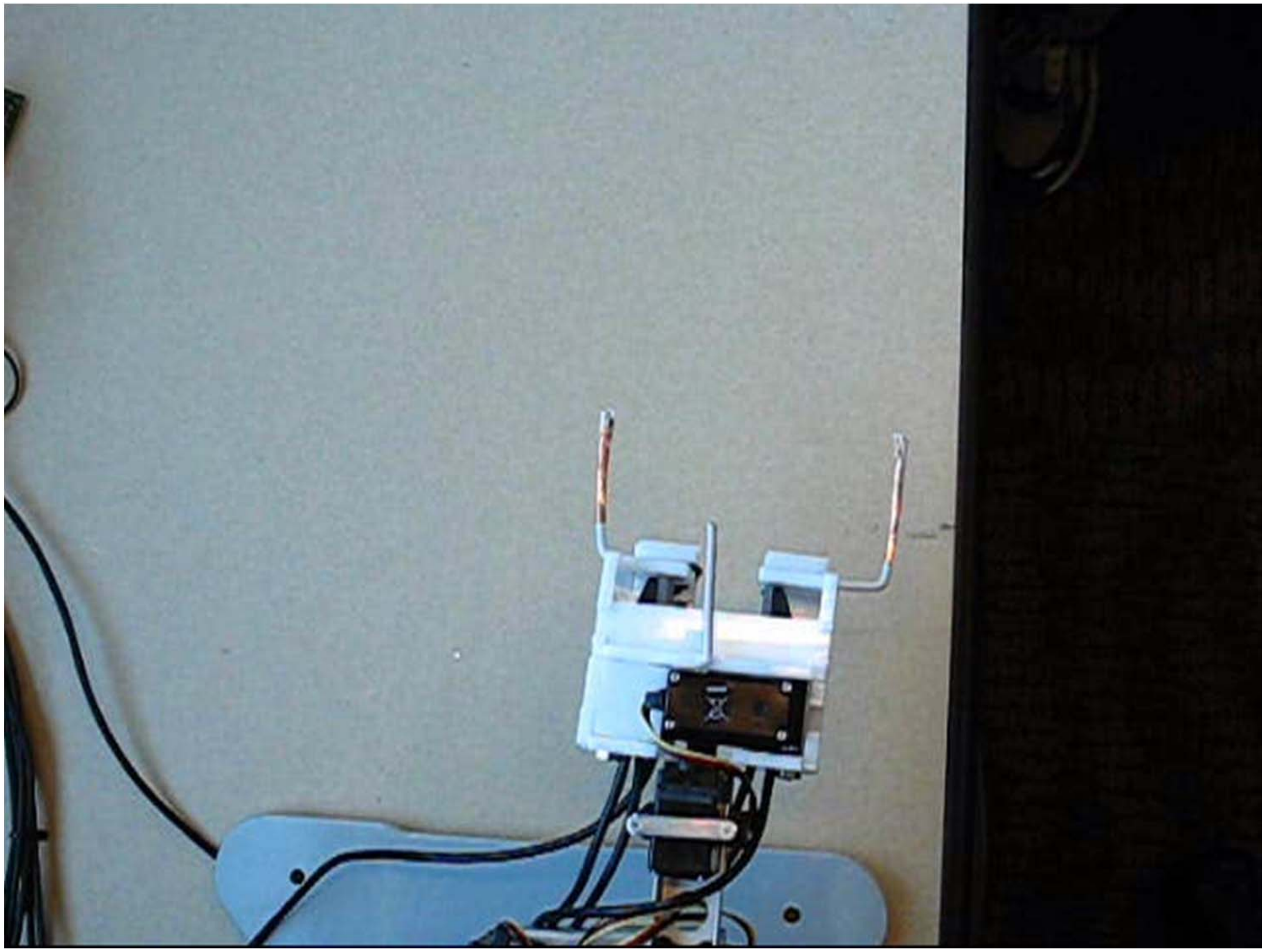
u_b: ***bias*** action

  - Set u_b so that $\dot{x} = F(x_{set}, u_b) = 0$
- For a linear system, exponential convergence.

$$x(t) = Ce^{-\alpha t} + x_{set}$$

- The controller gain *k* determines how quickly the system responds to error.

Proportional control (discrete time)



Velocity Control

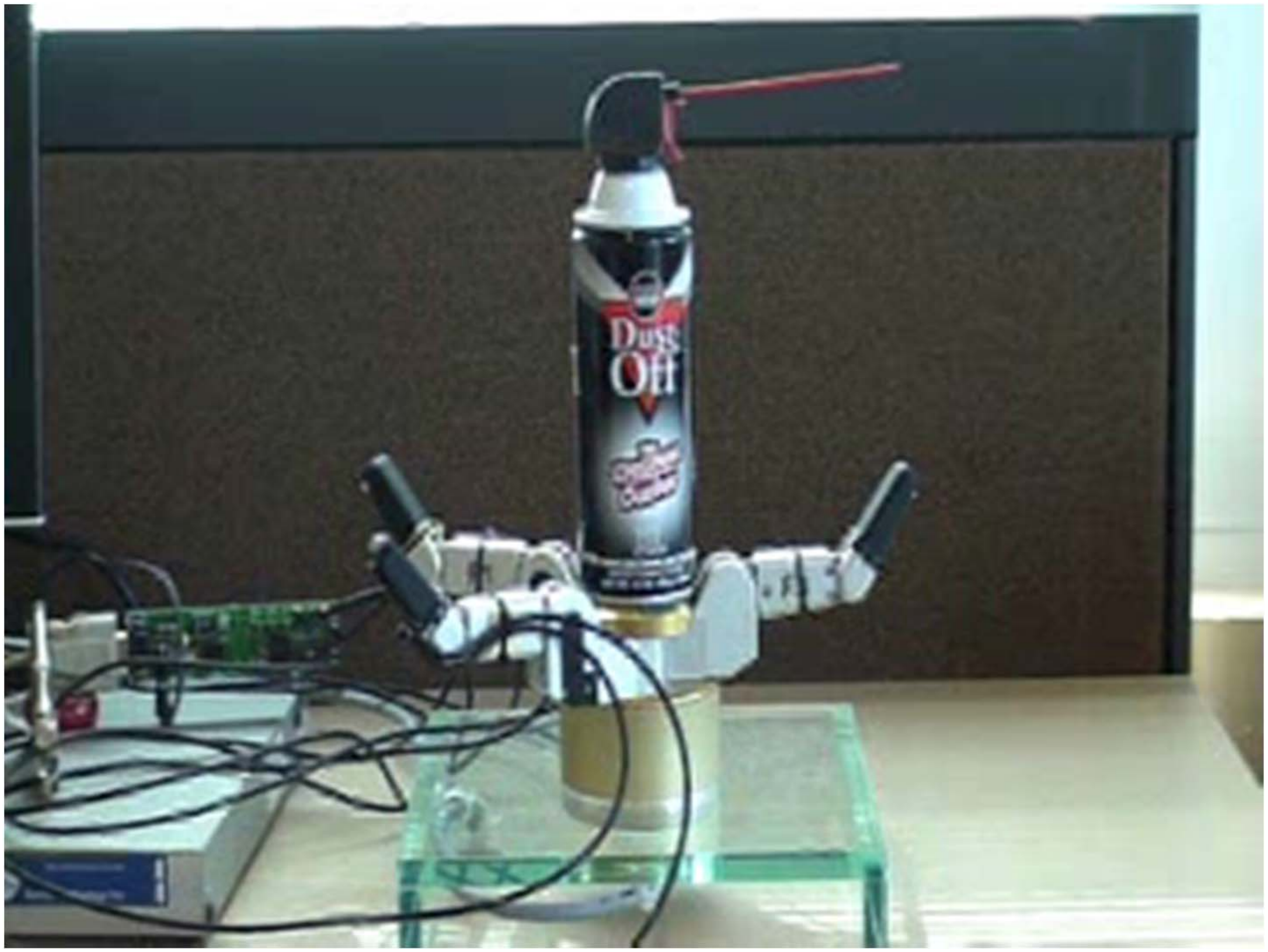
- You want the robot to move at velocity v_{set}.
- You command velocity v_{cmd}.
- You observe velocity v_{obs}.

- Define a first-order controller:

$$\dot{v}_{cmd} = -k\left(v_{obs} - v_{set}\right)$$

  - k is the controller gain.

Velocity control



Steady-State Offset

- Suppose we have continuing disturbances:

$$\dot{x} = F(x, u) + d$$

- The P-controller cannot stabilize at $e = 0$.
  - Why not?

Steady-State Offset

- Suppose we have continuing disturbances:

$$\dot{x} = F(x,u) + d$$

- The P-controller cannot stabilize at $e = 0$.
  - If u_b is defined so $F(x_{set}, u_b) = 0$
  - then $F(x_{set}, u_b) + d \neq 0$, so the system is unstable
- Must adapt u_b to different disturbances d.

Nonlinear P-control

- Generalize proportional control to

$$u = -f(e) + u_b$$

- Nonlinear control laws have advantages
  - *f* has vertical asymptote: bounded error *e*
  - *f* has horizontal asymptote: bounded effort *u*
  - Possible to converge in finite time.
  - Nonlinearity allows more kinds of composition.

Derivative Control

- Damping friction is a force opposing motion, proportional to velocity.
- Try to prevent overshoot by damping controller response.

$$u = -k_P e - k_D \dot{e}$$

- Estimating a derivative from measurements is fragile, and amplifies noise.

Adaptive Control

- Sometimes one controller isn't enough.
- We need controllers at different time scales.

$$u = -k_P e + u_b$$

$$\dot{u}_b = -k_I e \quad \text{where} \quad k_I << k_P$$

- This can eliminate steady-state offset.
  - Why?

Adaptive Control

- Sometimes one controller isn't enough.
- We need controllers at different time scales.

$$u = -k_P e + u_b$$

$$\dot{u}_b = -k_I e \quad \text{where} \quad k_I << k_P$$

- This can eliminate steady-state offset.
  - Because the slower controller adapts u_b.

Integral Control

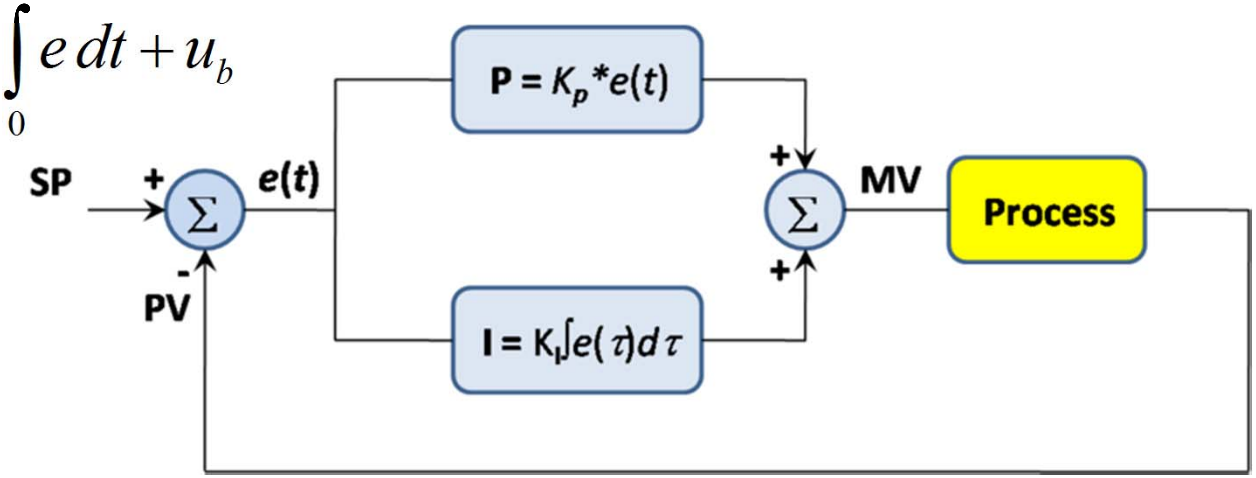
- The adaptive controller $\dot{u}_b = -k_I e$

Integrate both sides wrt time

- Therefore $u_b(t) = -k_I \int_0^t e\,dt + u_b$

- The Proportional-Integral (PI) Controller.

$$u(t) = -k_P\, e(t) - k_I \int_0^t e\,dt + u_b$$



Proportional – Integral (PI) control

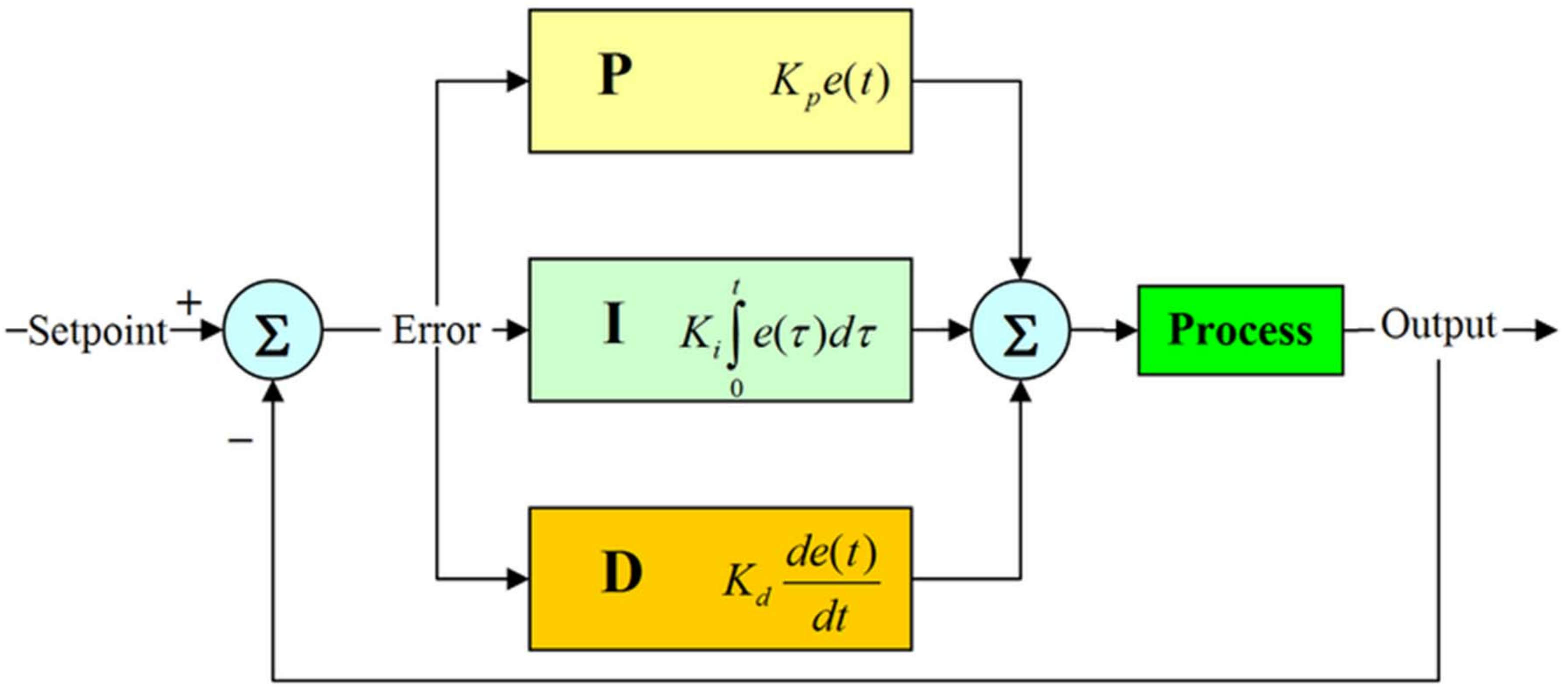
The PID Controller

- A weighted combination of Proportional, Integral, and Derivative terms.

$$u(t) = -k_P\, e(t) - k_I \int_0^t e\, dt - k_D \dot{e}(t)$$

- The PID controller is the workhorse of the control industry. Tuning is non-trivial.

PID controller

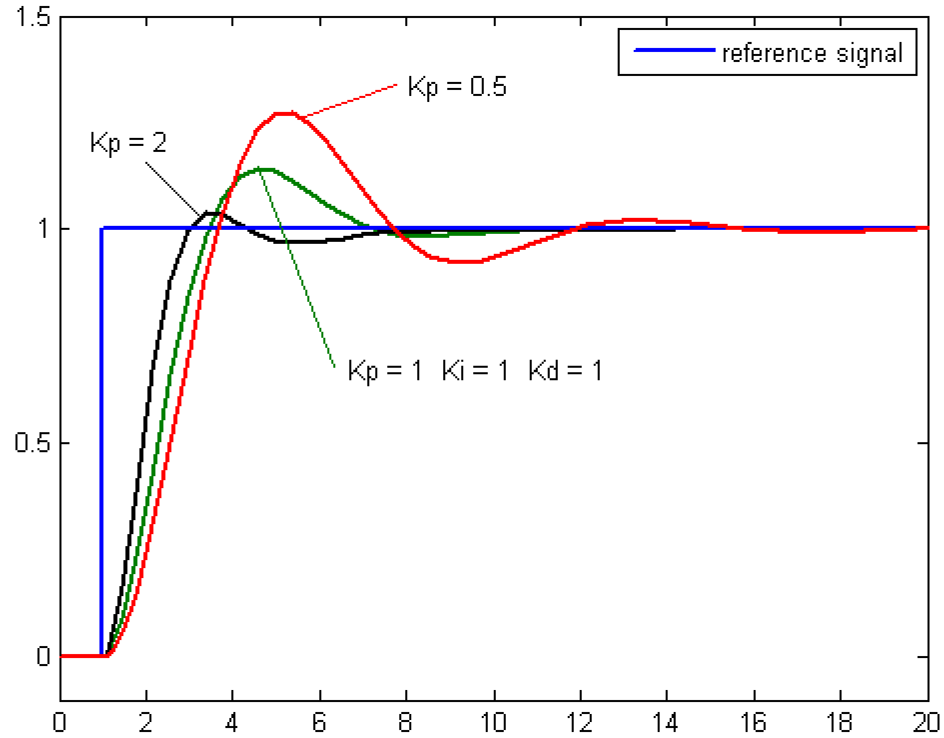


Tuning

PID properties to consider

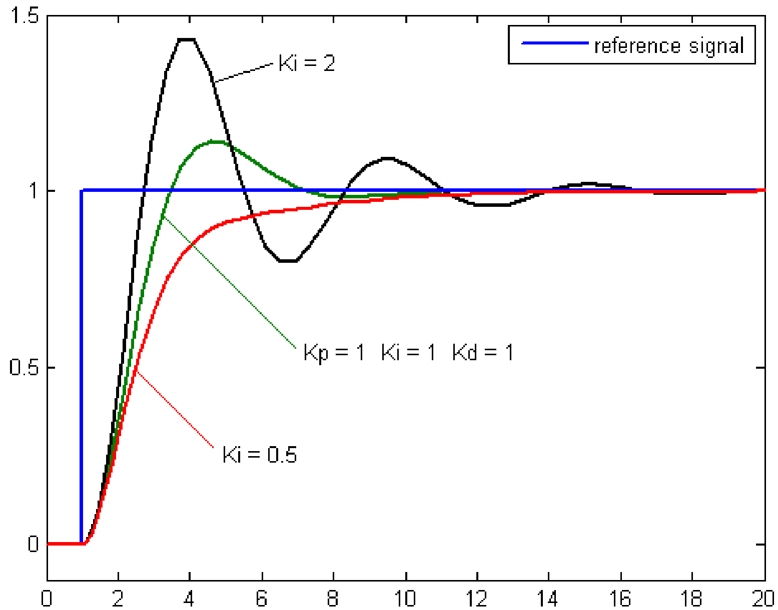
- Rise time
- Overshoot
- Settling time
- Steady-state error
- Stability

Many PID tuning methods exist

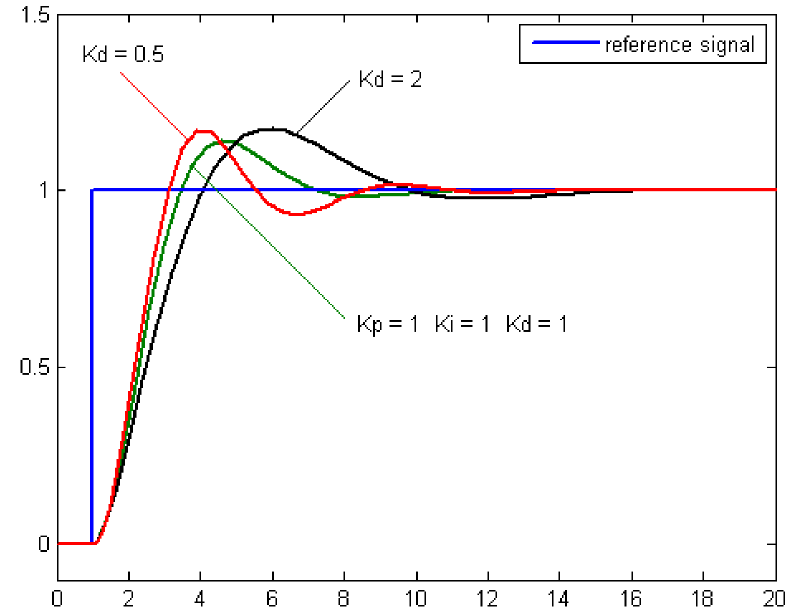


Proportional gain term

Integral and Derivative terms



Integral gain term



Derivative gain term

- Question from class: the integral of the red trace on the left is non-zero. Yet the controller is converging to the setpoint. Is there a decay mechanisms for the I term?
- A: not necessarily. In this example system, a non-zero bias command may be necessary to keep the system at its setpoint. Think of an application like gravity compensation: to keep the joint still, you may have to apply a constant force to counteract gravity. The integral term can find that bias point.

Habituation

- Integral control adapts the bias term u_b.
- Habituation adapts the setpoint x_{set}.
  - It prevents situations where too much control action would be dangerous.
- Both adaptations reduce steady-state error.

$$u = -k_P e + u_b$$

$$\dot{x}_{set} = +k_h e \quad \text{where} \quad k_h << k_P$$

Types of Controllers

- Feedback control
  - Sense error, determine control response.
- Feedforward control
  - Sense disturbance, predict resulting error, respond to predicted error before it happens.
- Model-predictive control
  - Plan trajectory to reach goal.
  - Take first step.
  - Repeat.
  - Combines benefits of planning & control
  - See Emo Todorov's ping pong ball juggling robot

End