Lecture 4:
Basic Concepts in Control

CS 344R/393R: Robotics

Benjamin Kuipers

Controlling a Simple System

- Consider a simple system: $\dot{x} = F(x, u)$
  - Scalar variables x and u, not vectors $\mathbf{x}$ and $\mathbf{u}$.
  - Assume x is observable: $y = G(x) = x$
  - 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 $e = 0$
  - error e depends on state x (via sensors y)
- What does pushing back do?
  - Depends on the structure of the system
  - Velocity versus acceleration control
- How much should we push back?
  - What does the magnitude of u depend on?

Velocity or acceleration control?

- If error reflects $\mathbf{x}$, does $\mathbf{u}$ affect $\mathbf{x}'$ or $\mathbf{x}''$?
- Velocity control: $\mathbf{u} \rightarrow \mathbf{x}'$ (valve fills tank)
  - let $\mathbf{x} = (x)$

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

- Acceleration control: $\mathbf{u} \rightarrow \mathbf{x}''$ (rocket)
  - let $\mathbf{x} = (x\ v)^T$

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

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

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.
- Who is right? Why should we care?
  - (We'll come back to this.)

The Bang-Bang Controller

- Push back, against the *direction* of the error
  - with constant action u
- Error is $e = x - x_{set}$

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

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

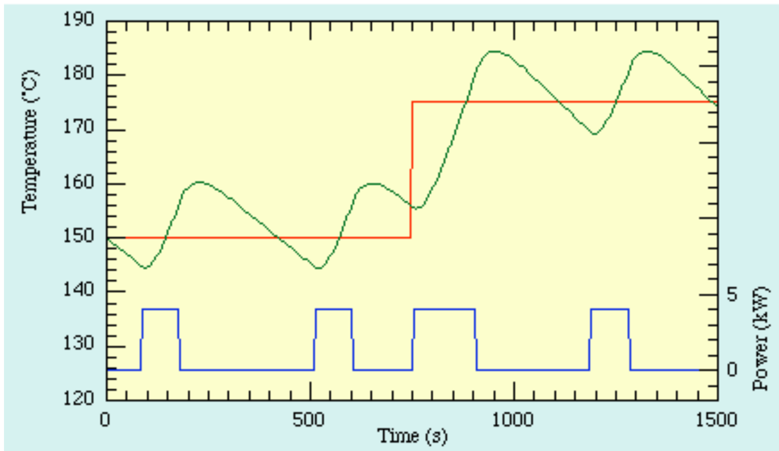
- To prevent chatter around $e = 0$,

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

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

- Household thermostat. Not very subtle.

Bang-Bang Control in Action



– Optimal for reaching the setpoint
– Not very good for staying near it

Proportional Control

- Push back, *proportional* to the error.

$$u = -ke + u_b$$

– set u_b so that $\dot{x} = F(x_{set}, u_b) = 0$

- For a linear system, we get exponential convergence.

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

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

Velocity Control

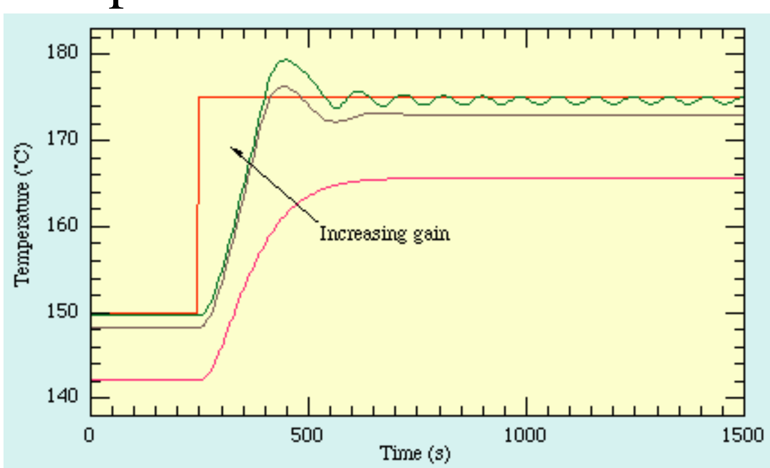
- You want to drive your car at velocity $v_{set.}$
- You issue the motor command $u = pos_{accel}$
- You observe velocity v_{obs}.

- Define a first-order controller:

$$u = -k(v_{obs} - v_{set}) + u_b$$

  – k is the controller gain.

Proportional Control in Action



– Increasing gain approaches setpoint faster

– Can leads to overshoot, and even instability

– Steady-state offset

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 changes
- Must adapt u_b to different disturbances d.

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 $\dot{u}_b = -k_I e$ means

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

- Therefore

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

- The Proportional-Integral (PI) Controller.

Nonlinear P-control

- Generalize proportional control to

$$u = -f(e) + u_b \quad \text{where} \quad f \in M_0^+$$

- 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.

Stopping Controller

- Desired stopping point: x=0.
  - Current position: x
  - Distance to obstacle: $d = |x| + \varepsilon$
- Simple P-controller: $v = \dot{x} = -f(x)$
- Finite stopping time for $f(x) = k\sqrt{|x|}\,\mathrm{sgn}(x)$

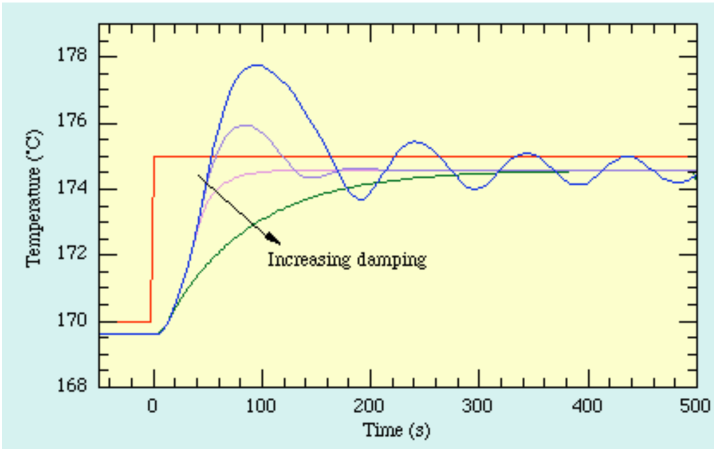
Derivative Control

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

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

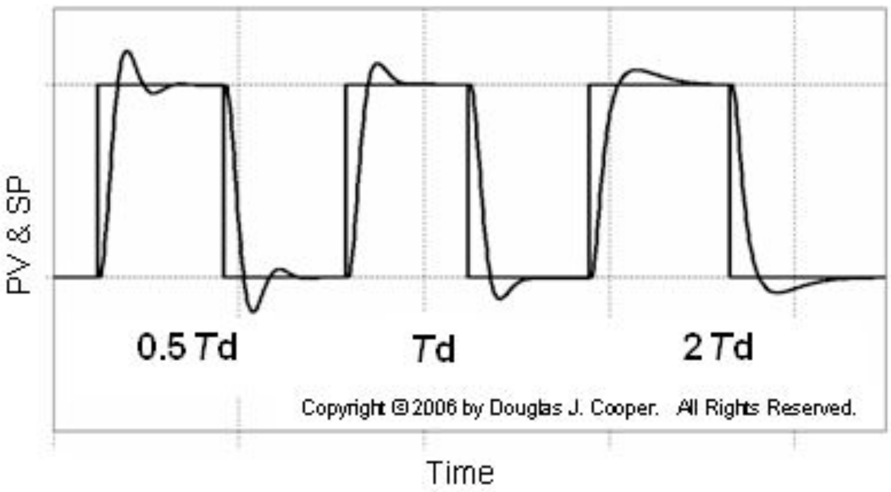
- Estimating a derivative from measurements is fragile, and amplifies noise.

Derivative Control in Action



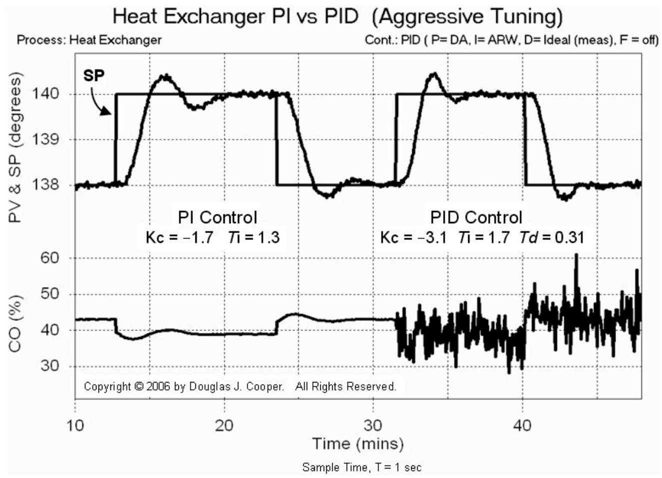
– Damping fights oscillation and overshoot
– But it's vulnerable to noise

Effect of Derivative Control



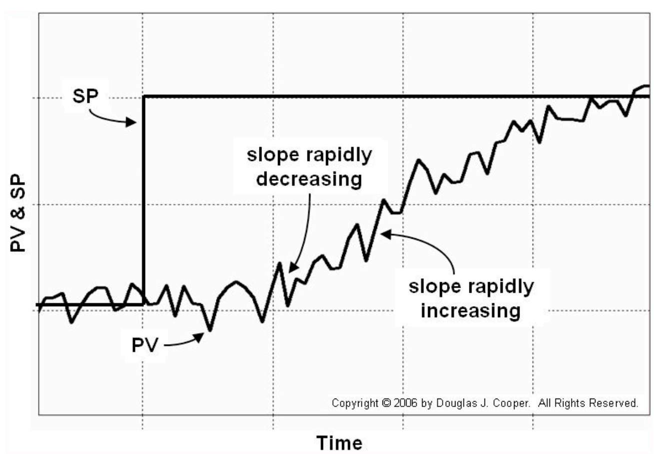
– Different amounts of damping (without noise)

Derivative Control Can Add Noise



– Why?

Derivatives Amplify Noise



– This is a problem if control output (CO) depends on slope (with a high gain).

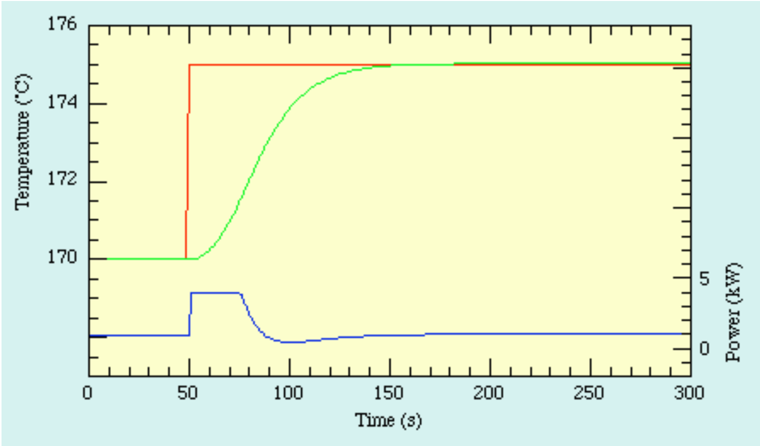
The PID Controller

- 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.
  – Next lecture includes some tuning methods.

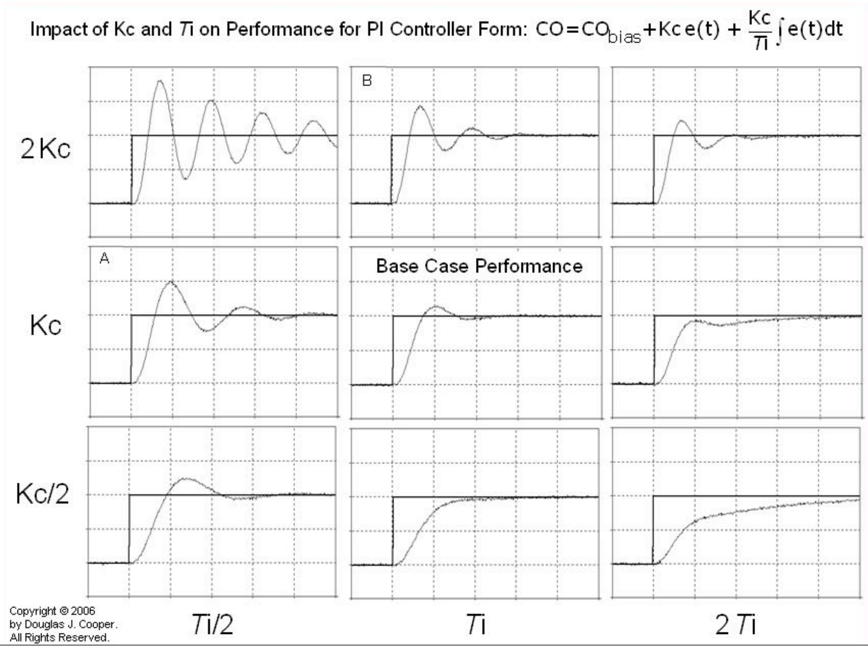
PID Control in Action



– But, good behavior depends on good tuning!

– More on this later.

Exploring PI Control Tuning



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.

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.
- Who is right? Why should we care?

Who is right? Aristotle!

- Try it! It takes constant force to keep an object moving at constant velocity.
  - Ignore brief transients
- Aristotle was a genius to recognize that there could be laws of motion, and to formulate a useful and accurate one.
- This law is true because our everyday world is *friction-dominated*.

Who is right? Newton!

- Newton's genius was to recognize that the true laws of motion may be different from what we usually observe on earth.
- For the planets in orbit, without friction, motion continues without force.
- For Aristotle, "force" means $F_{external}$.
- For Newton, "force" means F_{total}.
  - On Earth, you must include $F_{friction}$.

From Newton back to Aristotle

- $F_{total} = F_{external} + F_{friction}$
- $F_{friction} = -f(v)$ for some monotonic f.
- Thus: $\begin{pmatrix} \dot{x} \\ \dot{v} \end{pmatrix} = \begin{pmatrix} v \\ F/m \end{pmatrix} = \begin{pmatrix} v \\ \frac{1}{m} F_{ext} - \frac{1}{m} f(v) \end{pmatrix}$
- Velocity v moves quickly to equilibrium:

$$\dot{v} = \tfrac{1}{m} F_{ext} - \tfrac{1}{m} f(v)$$

- Terminal velocity v_{final} depends on:
  - F_{ext}, m, and the friction function $f(v)$.
  - *So Aristotle was right!* In a friction-dominated world.

For More Information …

- There are lots of good tutorials on the Web
- Search on:
  - PID control
  - process control
  - etc.
- http://www.controlguru.com
  - a blog about automatic process control