Very simple control, with PID

D.A. Forsyth, UIUC
We assume that everything is linear

- This creates huge mathematical simplifications
- Linear system:
  - accepts a signal $x(t)$
  - produces a signal $y(t) = K x(t)$
  - AND
  - $K (x(t) + y(t)) = K x(t) + K y(t)$
  - $K (a x(t)) = a K x(t)$
  - (notice this means $K 0 = 0$)

$K$ stands for a linear operator, so that (for example) we could have

$K x(t) = a x(t)$

or

$K x(t) = dx/dt$
In fact, study only the response to a step

- You can approximate any function with a lot of steps
- Step is $u(t)$
  - this is 0 for $t \leq 0$, 1 otherwise
  - so $u(t) - u(t + dt)$ is a bar
- Approximate $f(t)$ by

$$\sum_i f(i\Delta t)(u(i\Delta t) - u(i\Delta t + \Delta t))$$

- ex: simplify this expression
- ex: we know $K u(t)$ - what is $K f(t)$?
Ideas: plant/process, control

• Plant/process is the thing we wish to control
  • assume: 1 input, 1 output, linear
  • for simple examples, I’ll write out the form of the plant
    • but very often, it isn’t known exactly
      • System Identification

• Control:
  • supply the plant with the input needed to produce the output you want
  • Q: why is this hard?
    • A1: Plant may not be exactly known
    • A2: Plant may have dynamics
    • A3: Desired output may change
The very simplest control

• Plant: \( K x(t) = c x(t) \)
  • here \( c \) is a known constant

• We’d like the output to be 1
  • feed plant with \( 1/c \)
    • and go home early

• Example of open loop control
  • compute a fixed input and supply to plant
    • whatever the plant

• Advantages:
  • simple, sometimes works

• Disadvantages:
  • what if your model is wrong?
Example: move car at constant velocity

- **How?**
  - supply accelerator input so that car moves at constant velocity
- **Open loop:**
  - Figure out how acceleration causes velocity
    - $a = \frac{dv}{dt} - \text{(some frictional loss)}$
  - Supply acceleration
    - a burst of acceleration to get to speed
    - then constant acceleration to cope with loss
- **Doesn’t seem all that practical**
History of feedback

Watt’s Flyball Governor

These were still in use in late C20!

Watt’s flyball governor, C19
Example: move car at constant velocity

- How?
  - supply accelerator input so that car moves at constant velocity

- Closed loop:
  - Supply acceleration, measure speed
    - adjust acceleration so that car keeps speed
      - Q: how?
Closed loop control

• Derive an input to the plant from
  • setpoint (where you want the output to be)
  • current plant output

• The form we will discuss is:
We have

\[ c(t) = G \left( i(t) - o(t) \right) \]
\[ o(t) = H \ c(t) \]

so

\[ o(t) + H \ G \ o(t) = H \ G \ i(t) \]

which you should remember
Simple, worrying example

- \[ H \, c(t) = a \, c(t) \]
- \[ G \, x(t) = b \, x(t) \]
- \[ o(t) + ab \, o(t) = ab \, i(t) \]

Now imagine that \( i(t) \) is a step function
- for \( t>0 \) we have
- \( o(t) = \frac{ab}{1+ab} \)
  - which isn’t what we wanted
  - (remember, \( i(t) \) is the output value we want)
- steady state error is \( \lim t \to \infty (o(t)-i(t)) \)
Fix with integral term

- Idea:
  - if \((i(t)-o(t))\) is not zero, there should be some control input
  - magnitude increases until it is zero

\[
Gx(t) = bx(t) + c \int_0^t x(s) \, ds
\]
Fixing with integral term

\[ o(t) + abo(t) + ac \int_0^t o(s) \, ds = abi(t) + ac \int_0^t x(s) \, ds \]

Differentiate

\[ (1 + ab) \frac{do(t)}{dt} + aco(t) = ab \frac{di(t)}{dt} + aci(t) \]

BUT we’re interested in \( t > 0 \), and \( i(t) \) is a step at 0

\[ (1 + ab) \frac{do(t)}{dt} + aco(t) = aci(t) \]
Fixing with integral term

\[(1 + ab) \frac{do(t)}{dt} + aco(t) = ac\]

Assume that \(\frac{do}{dt} \rightarrow 0\) as \(t \rightarrow \text{infinity}\)
(we’ll see it does in a moment)

\[o(t) = 1\]

For large \(t\), which is what we wanted
Fixing with integral term

\[
\frac{(1 + ab)}{ac} \frac{do(t)}{dt} + o(t) = 1 \\
\]

\[
o(0) = 0
\]

\[
o(t) = (1 - e^{\frac{-ac}{1+ab}t})
\]
Example

• is it a good idea to get a faster response by making \( c \) bigger?
A more interesting plant

- Apply a force to the car to control its velocity
  - e.g., braking

\[ v(t) = v(0) + \int_0^t \frac{F(s)}{m} \, dt \]

\[ v(t) = \int_0^t \frac{F(s)}{m} \, dt \]
Proportional control

\[ o(t) + H G o(t) = H G i(t) \]

\[ G x(t) = b x(t) \]

\[ o(t) + H [bo(t)] = H [bi(t)] \]

\[ o(t) + \frac{b}{m} \int_0^t o(s) ds = \frac{b}{m} \int_0^t i(s) ds \]

\[ \frac{do}{dt} + \frac{b}{m} o(t) = \frac{b}{m} \quad \text{Recall that } t > 0, \ i(t) = 1 \]
Notice

\[
\frac{do}{dt} + \frac{b}{m}o(t) = \frac{b}{m}
\]

\[
o(t) = (1 - e^{-\frac{bt}{m}})
\]

- steady state error is now zero
- larger b/m \(\rightarrow\) faster response
  - BUT larger forces applied to car
- (obvious) b/m <0 \(\rightarrow\) unstable behavior
- Example
Examples
Examples

Bigger \( b/m \) -> faster rise time

- green: output
- blue: proportional term
- black: demand
Examples

Very big b/m -> fast rise time

output
proportional term
demand
Examples

Gigantic b/m -> integrator panics

output
proportional term
demand
Examples

Gigantic b/m, smarter integrator -> very fast rise time
But...

- Controller has only one parameter - b
  - the weight of the proportional term
- Integral term was useful for the simplest plant
  - what about here?
  - we’d have another parameter to adjust.
Proportional - Integral (PI) control

\[ o(t) + H \times G \times o(t) = H \times G \times i(t) \]

\[ Gx(t) = bx(t) + c \int_{0}^{t} x(s)ds \]
Proportional - Integral (PI) control

\[ o(t) + H \, G \, o(t) = H \, G \, i(t) \]

Plant is:

\[ v(t) = \int_0^t \frac{F(s)}{m} \, dt \]

\[ o(t) + H \left[ bo(t) + c \int_0^t o(s) \, ds \right] = H \left[ bi(t) + c \int_0^t i(s) \, ds \right] \]
Proportional - Integral (PI) control

\[ o(t) + H G o(t) = H G i(t) \]

Plug in plant, controller to get

\[
o(t) + \frac{1}{m} \int_{0}^{t} \left[ bo(u) + c \int_{0}^{u} o(s) \, ds \right] = \frac{1}{m} \int_{0}^{t} \left[ bi(u) + c \int_{0}^{u} i(s) \, ds \right]
\]

Differentiate twice, to get

\[
\frac{d^2 o}{dt^2} + \frac{b}{m} \frac{do}{dt} + \frac{c}{m} o(t) = \frac{c}{m}
\]

(recall \( t > 0 \), \( i(t) = 1 \))
Steady state error is zero:

\[ \frac{d^2 o}{dt^2} + \frac{b}{m} \frac{do}{dt} + \frac{c}{m} o(t) = \frac{c}{m} \]

Assume derivatives \( \to 0 \) as \( t \to \) infinity (we’ll see they do)
then \( o(t) = 1 \) for very large \( t \), which is what we wanted
\[
\frac{d^2 o}{dt^2} + \frac{b}{m} \frac{do}{dt} + \frac{c}{m} o(t) = \frac{c}{m}
\]

Solving:

Assume solution is of form:

\[
A_1 e^{zt} + A_2 t + A_3
\]

To get

\[
A_1 e^{zt} \left( z^2 + \frac{b}{m} z + \frac{c}{m} \right) + A_2 \left( \frac{b}{m} + t \frac{c}{m} \right) + A_3 \frac{c}{m} = \frac{c}{m}
\]
What about $A$ values?

\[ A_1 e^{zt} \left( z^2 + \frac{b}{m} z + \frac{c}{m} \right) + A_2 \left( \frac{b}{m} + t \frac{c}{m} \right) + A_3 \frac{c}{m} = \frac{c}{m} \]

\[ A_2 = 0 \quad \text{because there's no t on the right hand side} \]

\[ z^2 + \frac{b}{m} z + \frac{c}{m} = 0 \quad \text{because there's no t on the right hand side} \]

\[ A_3 = 1 \quad \text{to match } c/m \]

\[ A_1 = -1 \quad \text{because solution at time}=0 \text{ is 0 and solution is} \]

\[ A_1 e^{zt} + A_2 t + A_3 \]
Solution is:

\[(1 - e^{zt})\]

Where

\[z^2 + \frac{b}{m}z + \frac{c}{m} = 0\]

So

\[z = \frac{1}{2} \left[ -\frac{b}{m} \pm \sqrt{\frac{b^2}{m^2} - 4\frac{c}{m}} \right]\]
Solution properties:

\[ z = \frac{1}{2} \left[ -\frac{b}{m} \pm \sqrt{\frac{b^2}{m^2} - 4\frac{c}{m}} \right] \]

(1 - e^{zt})

Cases:

\( b^2 - 4cm > 0 \) (two real roots; sum of exponentials)

\( b^2 - 4cm = 0 \) (two copies of the same root - this is known as critical damping)

\( b^2 - 4cm < 0 \) (sinusoid with exponential amplitude)

Stability:

\(-b/m > 0\) - soln GROWS with time, otherwise OK
Cases:

I: two real roots; sum of exponentials

II: two copies of the same root

III: sinusoid with exponential amplitude
PI control $m=1, b=10, c=300$

Cases:

I: two real roots; sum of exponentials

II: two copies of the same root

III: sinusoid with exponential amplitude
PI control $m=1$, $b=1$, $c=300$
PI control $m=1$, $b=10$, $c=25$

Cases:

I: two real roots; sum of exponentials

II: two copies of the same root

III: sinusoid with exponential amplitude
Cases:

\[ b^2 - 4cm > 0 \] (two real roots; sum of exponentials)

\[ b^2 - 4cm = 0 \] (two copies of the same root - this is known as critical damping)

\[ b^2 - 4cm < 0 \] (sinusoid with exponential amplitude)

Cases:

I: two real roots; sum of exponentials

II: two copies of the same root

III: sinusoid with exponential amplitude
Cases:

I: two real roots; sum of exponentials

II: two copies of the same root

III: sinusoid with exponential amplitude
PI control $m=1$, $b=10$, $c=300$ step waveform
PI control $m=1$, $b=1$, $c=300$ step waveform
More on quadratic equations!

\[ z^2 + \frac{b}{m}z + \frac{c}{m} = 0 \]

\[ z^2 + 2\zeta \omega z + \omega^2 = 0 \]

\[ z = -\omega \left( \zeta \pm i\sqrt{1 - \zeta^2} \right) \]

Critical damping occurs when there is a double root
equivalently when \( \zeta = 1 \)
\( \zeta < 1 \) underdamped (soln. wobbles)
\( \zeta > 1 \) overdamped (slow rise time)
More on quadratic equations!

\[ z^2 + 2\zeta \omega z + \omega^2 = 0 \]

\[ z = -\omega \left( \zeta \pm i\sqrt{1 - \zeta^2} \right) \]

Our equation

\[ z^2 + \frac{b}{m}z + \frac{c}{m} = 0 \]

\[ \omega = \sqrt{\frac{c}{m}} \quad \zeta = \frac{1}{2} \frac{b}{\sqrt{cm}} \]

Critical damping:

\[ b = 2\sqrt{cm} \]
PI control critical damping $m=1$, $b=20$, $c=100$
A derivative term

- **Issue:**
  - may be hard to get fast rise time
  - big m requires big b for critical damping
  - this may be because we are feeding back the current error

- **Idea:**
  - predict future error
  - this is equivalent to feeding back some fraction of the derivative
The most important slide

- A very high fraction of all controllers in the real world are:

\[ Gx(t) = Ki \int_0^t x(u) du + Kp x(t) + Kd \frac{dx}{dt} \]

- PID controller
A more interesting plant

\[ v(t) = v(0) + \int_0^t \frac{F(s)}{m} \, dt \]

- Apply a force to the car to control its velocity
  - eg braking

\[ v(t) = \int_0^t \frac{F(s)}{m} \, dt \]
PID control critical damping $m=1$, $kp=20$, $ki=100$, $kd=0$
Proportional-Integral-Derivative (PID) control

Thrash through math of PI slide, and end up with:

\[
\frac{d^2 o}{dt^2} + \frac{K_p}{m + K_d} \frac{do}{dt} + \frac{K_i}{m + K_d} o = \frac{K_i}{m + K_d}
\]

Compare to:

\[
\frac{d^2 o}{dt^2} + \frac{b}{m} \frac{do}{dt} + \frac{c}{m} o(t) = \frac{c}{m}
\]

Kd makes the mass look smaller!
PID control critical damping $m=100$, $kp=20$, $ki=100$, $kd=0$
PID control critical damping $m=100$, $kp=20$, $ki=100$, $kd=-99$
In case you missed it...

- A very high fraction of all controllers in the real world are:

\[ Gx(t) = K_i \int_0^t x(u)du + K_p x(t) + K_d \frac{dx}{dt} \]

- PID controller
Yet more interesting plant

Apply a force to the mass, want to control its position.

\[ m \frac{d^2 x}{dt^2} + b \frac{dx}{dt} + kx = F \]
Proportional-Integral-Derivative (PID) control

Thrash through math of past slides, and end up with:

\[
\frac{d^2 o}{dt^2} + \frac{K_p + b}{m + K_d} \frac{dx}{dt} + \frac{K_i + k}{m + K_d} x = \frac{K_i + k}{m + K_d}
\]

Compare to:

\[
m \frac{d^2 x}{dt^2} + b \frac{dx}{dt} + kx = F
\]

Kd makes the mass look smaller! Kp changes the damping constant! Ki changes the spring constant!
PID control critical damping $m=1, b=0.01, c=0.01, kp=20, ki=300, kd=-0.9$
PID control  \( m=1, b=0.01, c=0.01, kp=20, ki=300, kd=-0.95 \)
PID control $m=1, b=0.01, c=0.01, kp=20, ki=300, kd=-0.98$
Proportional-Integral-Derivative (PID) control

Thrash through math of past slides, and end up with:

\[
\frac{d^2 o}{dt^2} + \frac{K_p + b}{m + K_d} \frac{dx}{dt} + \frac{K_i + k}{m + K_d} x = \frac{K_i + k}{m + K_d}
\]

Compare to:

\[
m \frac{d^2 x}{dt^2} + b \frac{dx}{dt} + k x = F
\]

Kd makes the mass look smaller! Kp changes the damping constant! Ki changes the spring constant!
Examples

PID control  \( m=10, b=0.01, c=0.01, kp=2001, ki=300001, kd=0 \)

PID control  \( m=10, b=0.01, c=0.01, kp=20, ki=300, kd=-9.9 \)
If the system must remain online, one tuning method is to first set $K_i$ and $K_d$ values to zero. Increase the $K_p$ until the output of the loop oscillates, then the $K_p$ should be set to approximately half of that value for a "quarter amplitude decay" type response. Then increase $K_i$ until any offset is corrected in sufficient time for the process. However, too much $K_i$ will cause instability. Finally, increase $K_d$, if required, until the loop is acceptably quick to reach its reference after a load disturbance. However, too much $K_d$ will cause excessive response and overshoot. A fast PID loop tuning usually overshoots slightly to reach the setpoint more quickly; however, some systems cannot accept overshoot, in which case an overdamped closed-loop system is required, which will require a $K_p$ setting significantly less than half that of the $K_p$ setting that was causing oscillation.
Kd = 0 for about 75% of deployed systems
Question

• Q: Why does the Federal Reserve Bank not just:
  • stick a PID controller on the inflation rate
    • input: desired inflation
    • output: base rate
  • and forget it?

• A: (obvious) it wouldn’t work
  • otherwise they’d be doing it

• Q: Why wouldn’t it work?
Stability and oscillation (rough)

- Linear systems can clearly oscillate
  - generally, too big a Kp or Kd can cause problems

- Nonlinearities can easily cause oscillations

- Delays cause oscillations
PI control around delay of 1e-4s, plant=1, kp=0.1, ki=5000
PI control around delay of 20e-4s, plant=1, kp=0.1, ki=5000
Demand is a step - this should look unpromising…

NOTICE Plant is 1 (really simple)
Unrecoverable

Pushing up Ki speculatively doesn’t help
Ideas

• Plant/process
• control
• Open vs closed loop
• stability
• Linear vs non-linear
• Simplest linear feedback control
  • x constant
  • with derivative term
  • large gains can cause instability
  • steady state error is a problem
• Delay is a problem
• non-linearities can create excitement
Ideas

• **PID control**
  • standard procedure
    • (there are tons in the car software)
  • P controls; I reduces steady state error; D increases response speed
  • Straightforward tuning procedure
    • (see software example)
The next few lectures

• You can stick a PID controller on
  • speed
  • distance
  • steering angle
  • etc.

• So you can cause the vehicle to follow a given path
  • at a given speed, etc.

• Where does the path come from?
  • simple planning (next)

• How do we know where obstacles are?
  • simple localization from point clouds (after that)