1 handout: slides
- check your Wildcat Pass before coming to campus
- send me email if you need to isolate/quarantine
Control

- COVID
- Problems
- Control
- MPC
- Break
- P Control
- PD Control
- PID Control
- Bisection Search
- See Also
- EOLQs
Observability: complete, partial, hidden
State: discrete, continuous
Actions: deterministic, stochastic, discrete, continuous
Nature: static, deterministic, stochastic
Interaction: one decision, sequential
Time: static/off-line, on-line, discrete, continuous
Percepts: discrete, continuous, uncertain
Others: solo, cooperative, competitive
low-level planning

- stochastic effects: policy, ‘control law’
- continuous state: how to represent?
used with ‘receding horizon’ (≈ real-time search)

simulate a bunch of controls (near nominal), pick best!

or steer to a bunch of states (near nominal), pick best!

flexible, dangerous
Break

- COVID
- Control
- Problems
- Control
- MPC
- Break
- P Control
- PD Control
- PID Control
- Bisection Search
- See Also
- EOLQs

- asst3
- projects
- wildcard class
$$u = K_P(x_r - \hat{x})$$

responsiveness vs smoothness  
= spring model  
allows persistent error!  
unstable with inertia!
\[ u = K_P (x_r - \hat{x}) + K_D \frac{d(x_r - \hat{x})}{dt} \]

dampen correction if error is changing a lot
= dampened spring model

does nothing if persistent error is constant!
PID Control

\[ u = K_P(x_r - \hat{x}) + K_I \int (x_r - \hat{x}) \, dt + K_D \frac{d(x_r - \hat{x})}{dt} \]

removes any persistent error
however, ‘wind-up’

widely used. not optimal or necessarily stable.

tune by hand, or
Thrun says coordinate-wise bisection search
Bisection Search

given $f$ and initial guesses $l$ and $r$

1. bracket a local minimum
   (a) try guess $m$ in middle
   (b) if $m$ smallest, done! (local min between $l$ and $r$)
   (c) if $l$ smallest, $r \leftarrow m$, $m \leftarrow l$ and move $l$ left
       move $l$ by at least original $r - l$ (double interval)
   (d) if $r$ smallest, $m \leftarrow r$ and move $r$ right

2. refine estimate
   (a) try $lm$ between $l$ and $m$.
   (b) if smaller than $m$, $r \leftarrow m$ and $m \leftarrow lm$
   (c) otherwise, try $mr$ between $m$ and $r$.
   (d) if smaller than $m$, $l \leftarrow m$ and $m \leftarrow mr$
   (e) otherwise $m$ is smallest, $l \leftarrow lm$ and $r \leftarrow mr$
   (f) until range small or values close
optimal control: eg, Linear-Quadratic-Gaussian (LQG)
 discrete control: eg, Markov decision processes
 state estimation aka filtering: eg, Kalman filter, particle filter
Please write down the most pressing question you have about the course material covered so far and put it in the box on your way out.

*Thanks!*