CS 4/5789 16 Feb 2022 Prof Sarah Dean

Lecture 8: Limitations in Action & Observation

1) PID Control

We spent several lectures discussing continuous state & action spaces, mostly focusing on (near) optimal policies. It's worth introducing a particular type of policy which is not optimal, but widely used in practice, often as a low-level controller.

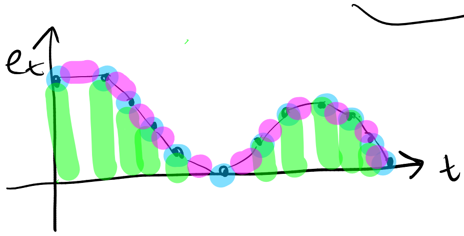
Setting → observation $o_t \in \mathbb{R}$ and set point $o_t^* \in \mathbb{R}$
→ we would like to minimize the error

$$e_t = o_t^* - o_t$$ based on measurement at t (no lookahead)

→ action $a_t \in \mathbb{R}$ is "correlated" with o_t (positive actions increase o_t)

Proportional - Integral - Derivative Control

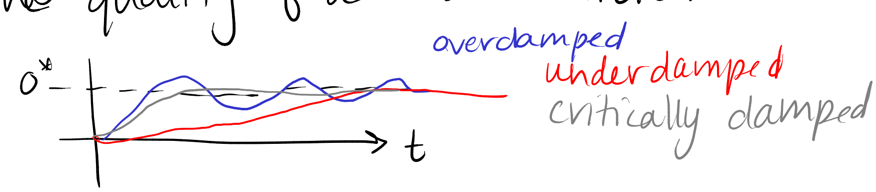
$$a_t = \underbrace{K_P \cdot e_t + K_I \sum_{k=0}^{t} e_k + K_D (e_t - e_{t-1})}_{\pi_{PID}(e_t, e_{t-1}, \dots, e_0)}$$



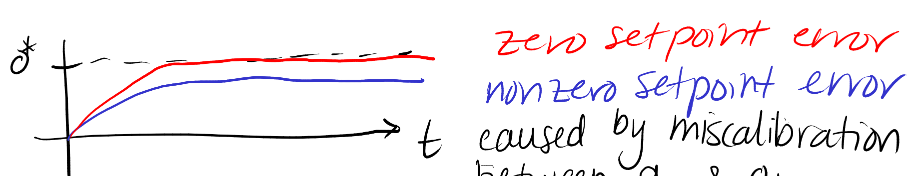
$\pi_{PID}(e_t, e_{t-1}, \dots, e_0)$ depends on history of errors.

There are three parameters to tune - often done by hand using heuristics. Rather than cumulative reward, the quality of a PID controller is judged by:

1) damping: (K_P & K_D)



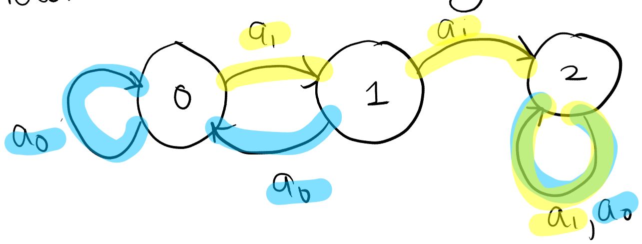
2) set point error: (K_I)



So far we've focused on how to compute a (near) optimal policy given the model of an MDP. Soon, we will turn to <u>learning</u> near optimal policies from data, without a model. But first, today we will step back to ask: How good can the optimal policy even be? Are there inherent properties of systems that limit performance?

2) Reachability

Consider the following motivating example:



Deterministic MDP 3 states
2 actions

$$r(s,a) = \begin{cases} 1 & s=0 \\ 0 & \text{otherwise} \end{cases}$$

$\pi^*(s) = a_0$

$V^*(2) = 0$. Even acting optimally, no reward possible if starting in s_2!

<u>Definition</u> (Reachability in Discrete MDP)

- State s' is <u>reachable</u> from state s if there exists a sequence of actions $a_0, \ldots, a_{T-1}$ for finite T such that $\mathbb{P}(s_T = s' \mid s_0 = s, a_0, \ldots, a_{T-1}) > 0$.
- MDP is <u>reachable</u> if all states are reachable from any state

<u>Theorem</u> (Discrete Reachability):

Given $\mathcal{S}, \mathcal{A}, P$, construct a directed graph with vertices $V = \mathcal{S}$ and edges from s to s' if $P(s'|s,a) > 0$ for some a.

Then MDP Reachable if the graph is <u>strongly connected</u>.

(ie., there is a path from every vertex to every other vertex)

Proof: Since the graph is strongly connected, there exists a directed path from $s \to s'$ for any s, s'. Let T be its length. By construction, each edge along this path corresponds to some action a_i and some nonzero transition probability p_i.
Then

$$P(s_T = s' \mid s_0 = s, a_0, \dots, a_{T-1}) > \prod_i p_i > 0.$$

product

Another Motivating Example:

$$s_{t+1} = \begin{bmatrix} 1/2 & 0 \\ 0 & 1 \end{bmatrix} s_t + \begin{bmatrix} 0 \\ 1 \end{bmatrix} a_t$$

No matter the actions, $s_t^{(1)} = (1/2)^t s_0^{(1)}$

Definition (Deterministic Reachability):
- state s' is reachable from s if there exist a finite sequence of actions $a_0, \dots, a_{T-1}$ such that

$$s' = s_T = f(s_{T-1}, a_{T-1}) \quad \cdots \quad s_0 = s.$$

- system is reachable if all states reachable from any state.

Theorem (Linear Reachability)
A linear system $s_{t+1} = As_t + Ba_t$ is reachable if the controllability Gramian C is full rank

$$\mathrm{rk}([B \quad \underbrace{AB \quad A^2B \cdots \quad A^{n_s-1}B}_{C \in \mathbb{R}^{n_s \times n_s \cdot n_a}}]) = n_s$$

Proof: recall that $s_t = A^t s_0 + \sum_{k=0}^{t-1} ABa$

$$s_{n_s} - A s_0 = \underbrace{[B \quad AB \quad \cdots \quad A^{n_s-1}B]}\begin{bmatrix} a_{n_s-1} \\ \vdots \\ a_0 \end{bmatrix}$$

if full rank, can solve for $a_0, \dots, a_{n_s-1}$.

3) Limitations in Observation

So far (and for most of the rest of this course) we assume that we observe the state directly. But what if this assumption is violated?

Delays:

suppose

$$P(s_{t+1}=s \mid s_0,\dots,s_t, a_0,\dots,a_t) = P(s_{t+1}=s \mid s_t, \underbrace{a_{t-D}}_{D\text{ step delay}})$$

This violates the Markovian assumption. But not fundamentally - if we carefully redefine the state (HW1).

Partial observation: (PO)

What if $o_t = g(s_t)$? if g is invertible, o_t would be a valid equivalent state.

But if o_t is not invertible, or there is noise, it's not.

The correct approach for POMDPs is to consider the distribution of possible states given observations & actions.

$$\mathbb{P}(s_t=s \mid a_0,\dots, a_t, o_0,\dots,o_t)$$

This is easy for Linear-Gaussian systems (Kalman filtering) but in general difficult because it depends on the entire history (a common approximate approach is called particle filtering) and requires using knowledge of the transition model.

Another approximation is to construct a "state" based on some truncated history of observations/actions, which can be exactly valid in some settings (HW1)

$$s_t \approx \begin{bmatrix} o_t \\ \vdots \\ o_{t-H} \\ a_t \\ \vdots \\ a_{t-H} \end{bmatrix}$$

4) Model Mis-specification & robustness

What if we compute an optimal policy for a slightly incorrect system model?

Example: $\min \mathbb{E}_w\left[\sum \|s_t - ba_t\|_2^2 \;\middle|\; s_{t+1} = ba_t + w_t\right]$ $s, a, w \in \mathbb{R}$

The optimal policy is $a_t = s_t/b$ and under this policy, the system:

$$s_{t+1} = s_t + w_t$$

First, note that it is <u>marginally stable</u> (random walk).
Second, if the dynamics <u>are actually</u>

$s_{t+1} = \tilde{b} a_t + w_t$, then

$$s_{t+1} = \frac{\tilde{b}}{b} s_t + w_t$$ is <u>unstable</u> whenever $\tilde{b} > b$

and the cost is <u>actually</u> $\|s_t - \tilde{b} a_t\|_2^2$

$$c_t = \|s_t - \tfrac{\tilde{b}}{b} s_t\|_2^2 = (1 - \tilde{b}/b)^2 \|s_t\|^2 \to \infty$$

if $\tilde{b} > b$ by instability.

<u>Moral</u>: Arbitrarily small specification errors lead to arbitrarily bad performance!

The feild of robust control studies this type of phenomena.

For most of this class, we focus on <u>optimization</u> given an MDP rather than <u>design</u>: building a system and modelling it as an MDP. But in real applications, design is just as (if not more) important. E.g. If reachability is an issue, can we add another actuator? If observation is an issue, should we add another sensor? If robustness is an issue, should we tweak our cost/reward function? Towards the end of this course we will revisit some of these issues.