LINMA2725 Stochastic Optimal Control and Reinforcement Learning
Part III

Course 4: Online Learning Methods

Guillaume Berger

December 2, 2024

References: [1]; [2]; [3], Section 7.8 and Appendix C.

Any questions or feedback are welcome.

Stochastic Optimal Control and Reinforcement Learning

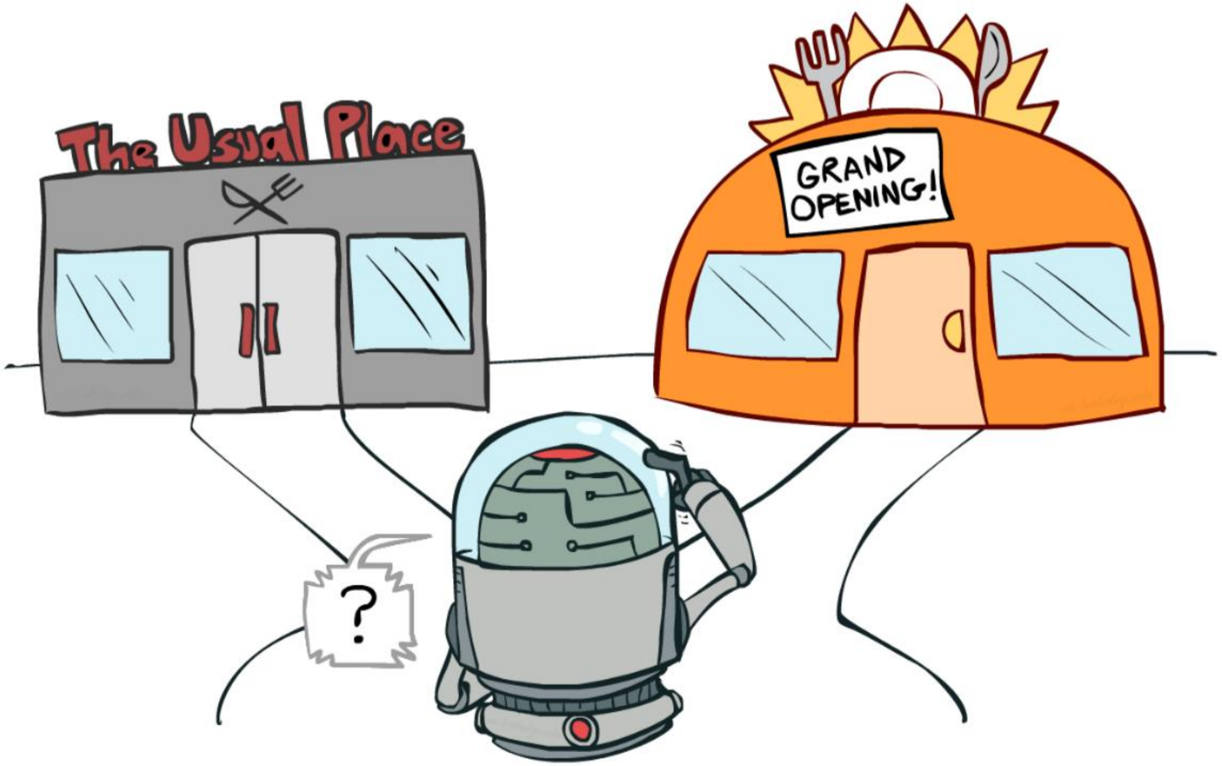
Part III: Stochastic Systems

Guillaume Berger

Table of content

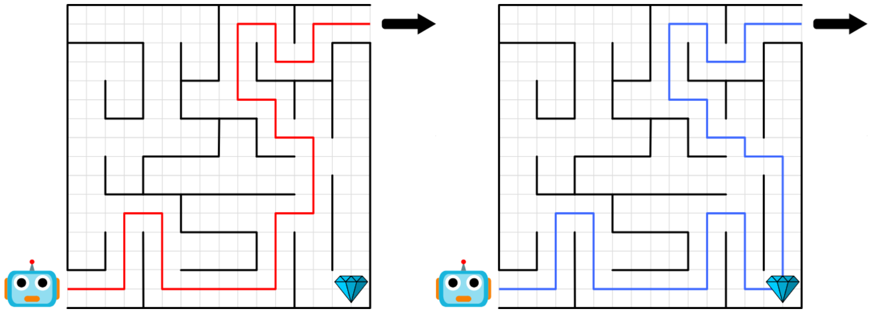
- Stochastic systems and stochastic control (1 course)
- Learning techniques for stochastic control (1-2 courses)
- Online learning techniques for stochastic control (1 courses)
    - Bandit problem: introduction, techniques and analysis
    - Model-based methods in adaptive control: overview
    - Optimal control with partial information: belief state, separation principle

Exploitation and exploration



Exploitation and exploration

- Game Playing
  Exploitation: Play the move you believe is best
  Exploration: Play an experimental move
- Restaurant Selection
  Exploitation: Go to your favorite restaurant
  Exploration: Try a new restaurant
- Oil Drilling
  Exploitation: Drill at the best known location
  Exploration: Drill at a new location
- Online Banner Advertisements
  Exploitation: Show the most successful advertisement
  Exploration: Show a different advertisement



Multi-armed bandit

K slot machines indexed by $k \in \{1, \dots, K\}$

At each step $t \geq 1$, you choose one slot machine: $u_t \in \{1, \dots, K\}$

You receive a reward $X_t \sim D_{u_t}$
where D_k is the reward distribution of the k^{th} machine

Example:
- D_1 is the uniform distribution in $[0,1]$
- D_2 is the exponential distribution $p(x) = e^{-x}\{x \geq 0\}$

Which slot machine should you choose?

Regret in multi-armed bandit

Optimal expected gain: expected gain by playing always the machine with the highest expected reward (μ^*)

Regret: difference between your gain and the optimal expected gain

$$R(t) \overset{\text{def}}{=} \sum_{s=1}^{t} X_s - t\mu^*$$

Goal: Minimizing the growth of the expected regret

Example: $\mathbb{E}[R(t)] \leq C_1\sqrt{t} + C_2$ (sublinear growth)

Estimator of expected reward

For each $k \in \{1, \ldots, K\}$ and $t \geq 1$, let

$$T_k(t) \overset{\text{def}}{=} \{s \in [1, t] : u_s = k\}$$

(times at which you chose machine k) and $N_k(t) = |T_k(t)|$

Define the following estimate of μ_k (the mean of D_k):

$$\bar{\mu}_k(t) \overset{\text{def}}{=} \frac{1}{N_k(t)} \sum_{s \in T_k(t)} X_s$$

(sample average of reward of machine k)

Greedy algorithm

Algorithm: at each step $t \geq 1$, choose k for which $\bar{\mu}_k(t)$ is largest

Example:

- At $t = 1$: choose $u = 1$. You get $X_1 = 0{,}6$
- At $t = 2$: choose $u = 2$. You get $X_2 = 0{,}3$

Now, $\bar{\mu}_1(2) = 0{,}6$ and $\bar{\mu}_2(2) = 0{,}3$

Hence, at $t = 3$, you will choose $u = 1$

Greedy algorithm and exploration

The greedy algorithm prevents exploration:

> “If we were unlucky in our first reward of machine k we will not choose it again”

Two (partial) remedies:

- Baseline
- ϵ-greedy algorithm

Baseline adds optimism

<u>Optimism:</u> "I believe that all choices are good, so that I need "many" observations of a low reward to conclude that a given choice is bad"

<u>In practice:</u> change the definition of $\bar{\mu}_k(t)$ by

$$\bar{\mu}_k(t) \overset{\text{def}}{=} \frac{1}{N_k(t)}\left(b + \sum_{s \in T_k(t)} X_s\right)$$

where b is the <u>baseline</u>

Limitations of baseline greedy algorithm

Choosing the baseline requires assumptions (prior knowledge)

Does not guarantee that the expected reward has sublinear growth

ϵ-greedy algorithm

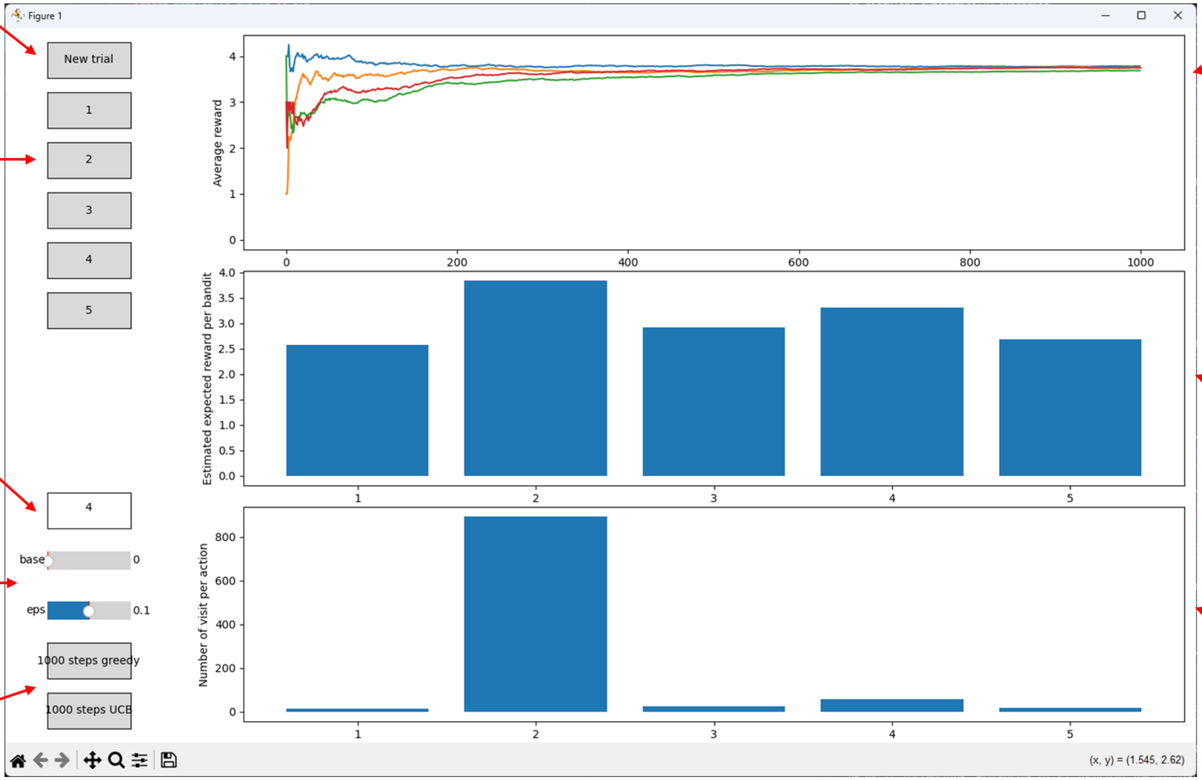
Algorithm: at each step $t \geq 1$,

- with probability $1 - \epsilon$, make the greedy choice;
- with probability ϵ, choose a machine uniformly at random.

Limitations of ϵ-greedy algorithm

Impossible to achieve sublinear growth of the expected regret

Time for illustration



Reset all quantities but keep previous plots of average reward

Activate bandit n° x

Observe reward after activating the banding

Choose parameters baseline and ϵ for greedy strategy

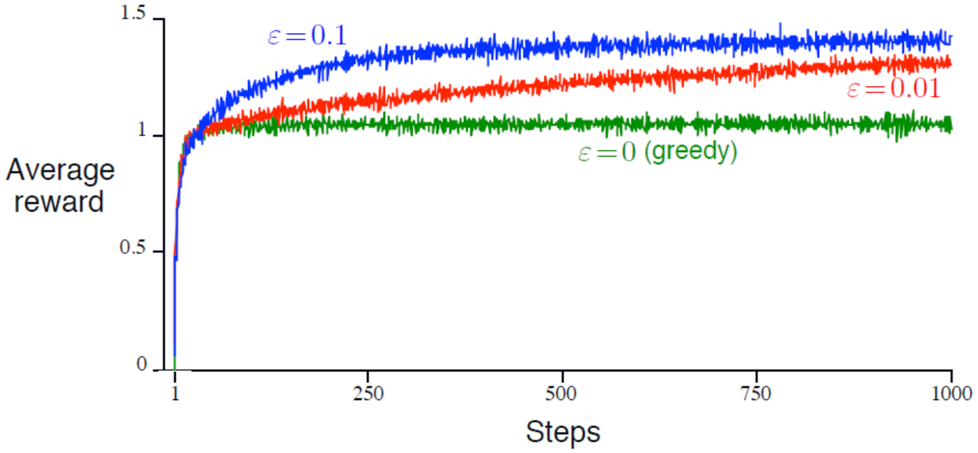
Simulate 1000 steps of greedy strategy or UCB strategy

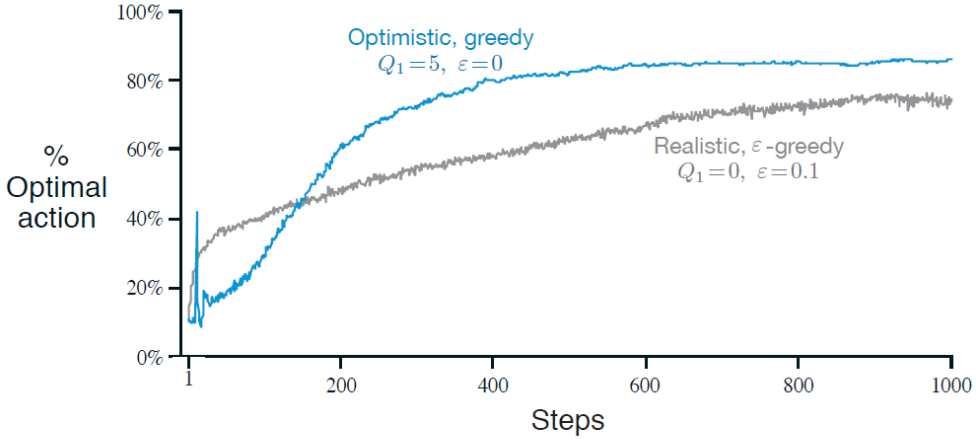
Average reward: different colors for different 'trials'

Estimation of the expected reward of each bandit based on past observed rewards (reset at each new trial)

Number of visit of each (reset at each new trial)

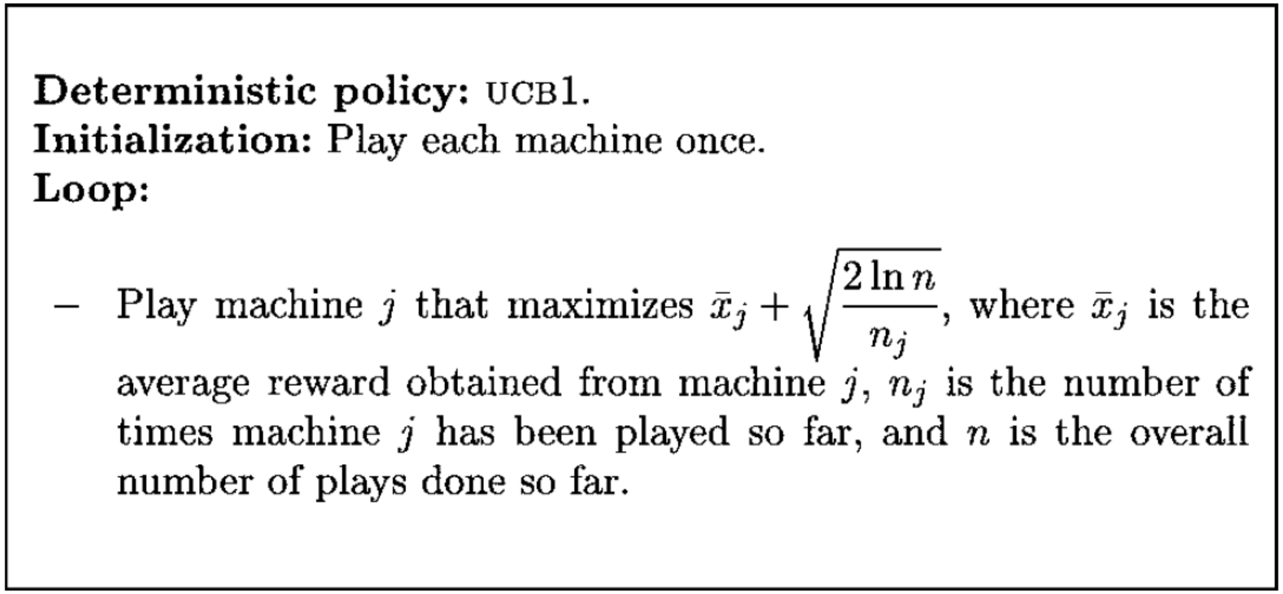
Time for illustration





Upper Confidence Bound (UCB) algorithm

Algorithm:

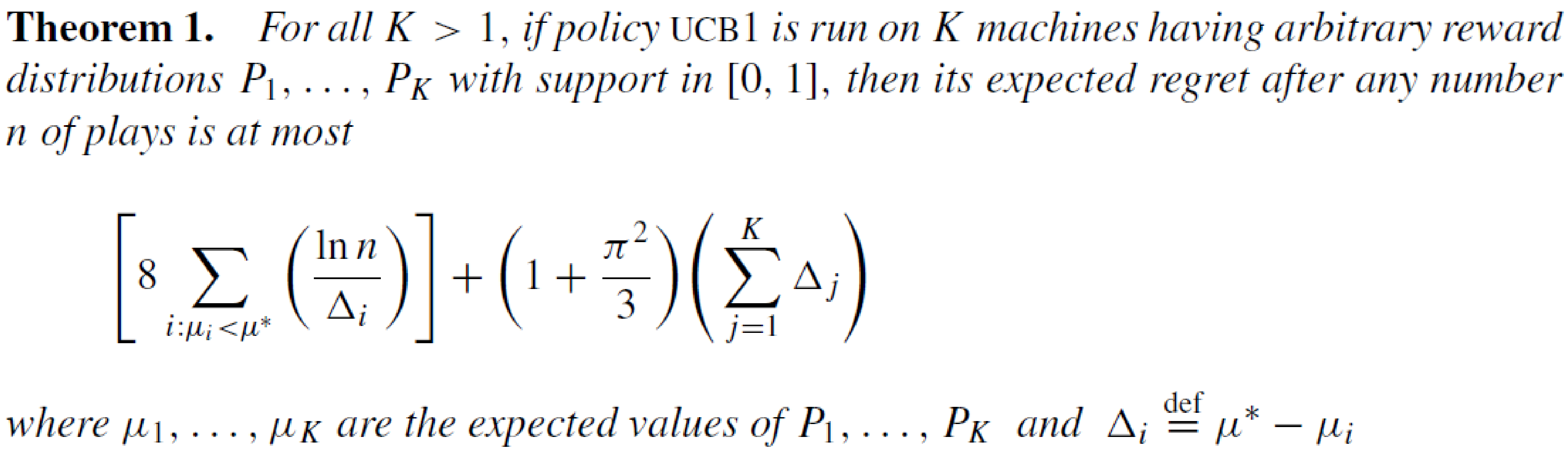


Deterministic policy: UCB1.
Initialization: Play each machine once.
Loop:

- Play machine j that maximizes $\bar{x}_j + \sqrt{\frac{2 \ln n}{n_j}}$, where $\bar{x}_j$ is the average reward obtained from machine j, n_j is the number of times machine j has been played so far, and n is the overall number of plays done so far.

Reference: [1]. Notice the slightly different notation (although explained in the algorithm).

Analysis of UCB algorithm



Theorem 1. *For all* $K > 1$, *if policy* UCB1 *is run on* K *machines having arbitrary reward distributions* $P_1, \ldots, P_K$ *with support in* $[0, 1]$, *then its expected regret after any number* n *of plays is at most*

$$\left[8 \sum_{i:\mu_i<\mu^*} \left(\frac{\ln n}{\Delta_i}\right)\right] + \left(1 + \frac{\pi^2}{3}\right)\left(\sum_{j=1}^{K} \Delta_j\right)$$

where $\mu_1, \ldots, \mu_K$ *are the expected values of* $P_1, \ldots, P_K$ *and* $\Delta_i \overset{\text{def}}{=} \mu^* - \mu_i$

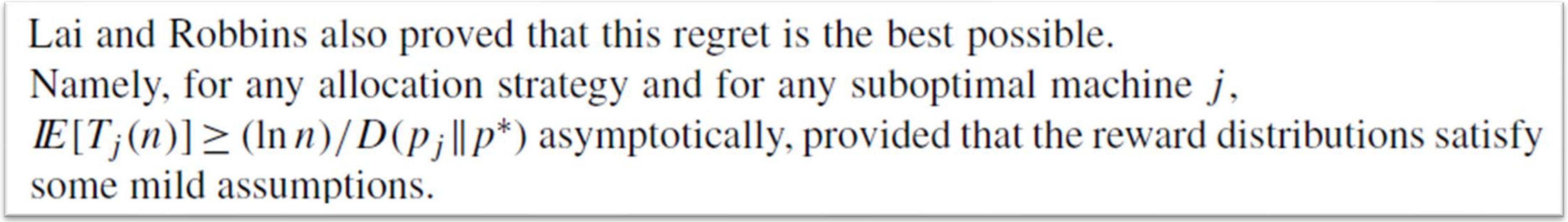
See [1, Theorem 1] for a proof.

Analysis of UCB algorithm

(continued)

Hence, UCB achieves logarithmic growth!

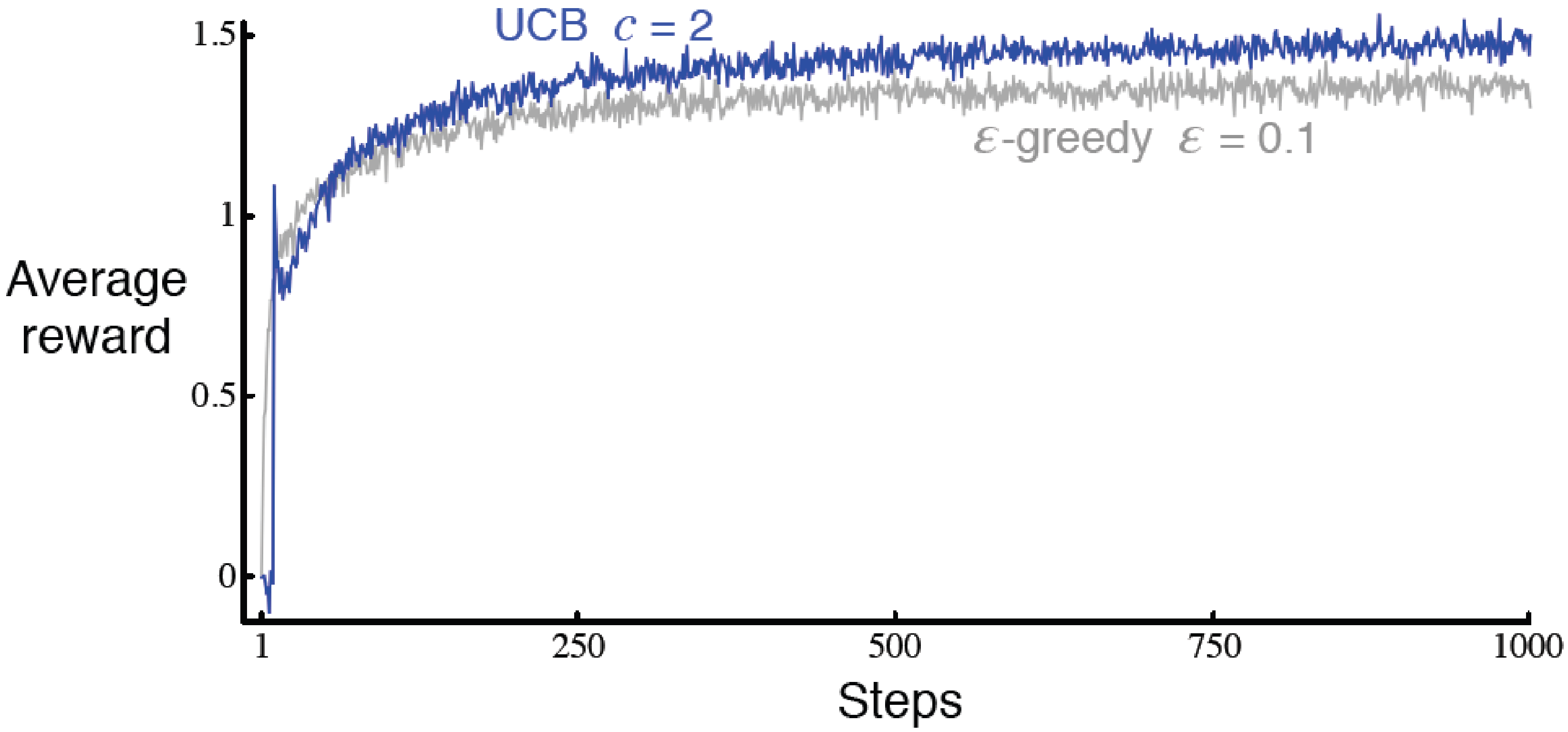
It can be shown that this is the best achievable growth



Lai and Robbins also proved that this regret is the best possible.
Namely, for any allocation strategy and for any suboptimal machine j, $\mathbb{E}[T_j(n)] \geq (\ln n)/D(p_j \| p^*)$ asymptotically, provided that the reward distributions satisfy some mild assumptions.

See [1, Section 7.8.3] for a discussion.

Illustration of UCB algorithm



Gradient bandit algorithms

<u>Preference</u> for each machine: $\theta_k \in \mathbb{R}$

<u>Gibbs or Boltzmann policy:</u> $p^\theta(k) = \frac{\exp \theta_k}{\sum_{\ell=1}^K \exp \theta_\ell}$

Goal: Find θ that maximizes the expected reward of p^θ

$$\Gamma(\theta) \overset{\text{def}}{=} \sum_{k=1}^{K} p^\theta(k)\mu_k$$

using gradient ascent

Gradient of expected reward

Result 1:

$$\nabla_\theta \Gamma(\theta) = \sum_{k=1}^{K} \mu_k p^\theta(k) \nabla_\theta \log\left(p^\theta(k)\right)$$

Result 2:

$$\frac{\partial}{\partial \theta_\ell} \log\left(p^\theta(k)\right) = p^\theta(k)\left(\{\ell = k\} - p^\theta(\ell)\right)$$

A stochastic gradient ascent algorithm

Algorithm: given an initial $\theta^0 \in \mathbb{R}^K$, at each step $t \geq 0$,

- sample

$$u_{t+1} \sim p^{\theta^t}(\cdot) \quad \text{and} \quad X_{t+1} \sim D_{u_{t+1}},$$

- define

$$\theta_k^{t+1} = \theta_k^t + \alpha_{t+1} X_{t+1} \left(\{k = u_{t+1}\} - p^{\theta^t}(k) \right) \quad \forall\, k$$

Improvement: use $X_{t+1} - \bar{X}_t$ instead of X_{t+1} where $\bar{X}_t \overset{\text{def}}{=} \frac{1}{t} \sum_{s=1}^{t} X_s$ (baseline) to reduce variance

The improvement part is reminiscent of the use of the advantage function to reduce the variance for the actor–critic method. It follows from the observation that

$$\nabla_\theta \Gamma(\theta) = \sum_{k=1}^{K} (\mu_k - \nu) p^\theta(k) \nabla_\theta \log(p^\theta(k))$$

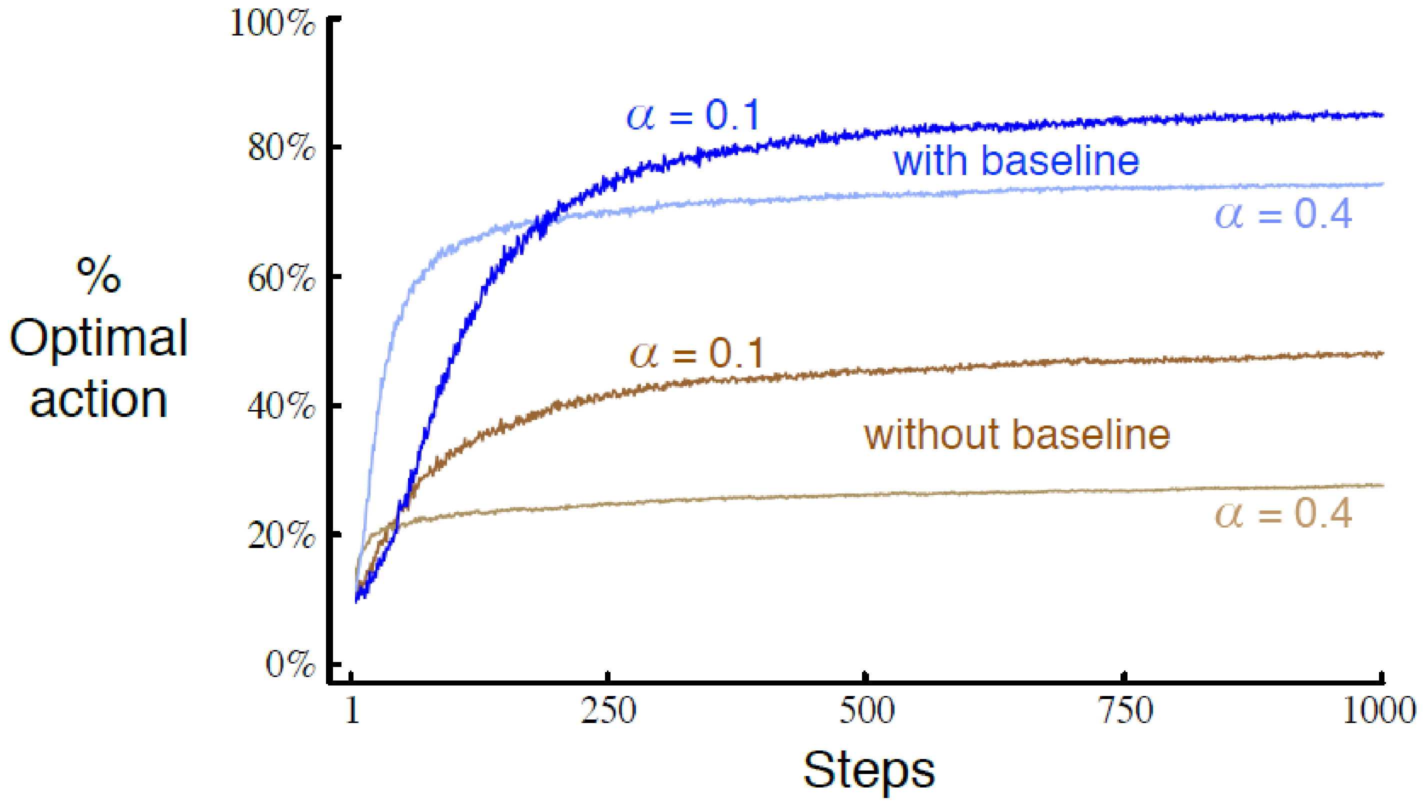
for any ν that is independent of k.

Proof. Observe that

$$\sum_{k=1}^{K} p^\theta(k) \nabla_\theta \log(p^\theta(k)) = \sum_{k=1}^{K} \nabla_\theta p^\theta(k) = \nabla_\theta \sum_{k=1}^{K} p^\theta(k) = \nabla_\theta(1) = 0,$$

concluding the proof. □

Illustration of gradient bandit algorithm



We see that in this experiment, the “improvement” leads to a better regret (“with baseline”) compared to the “unimproved” algorithm (“without baseline”).

Beyond the bandit setting

Main theme: model of the system is unknown (uncertain system)

(Example: in bandit problem, the distributions D_k are unknown)

We can rely only on data to learn the optimal controller

Most of the techniques seen so far in the course are data-based (e.g., LSTD, TD(λ)-learning, Q(λ)-learning, actor-critic methods, etc.)

Hence, they can be applied for the optimal control of uncertain systems

Beyond the bandit setting

(continued)

The novelty brought by bandit problems is the notion of regret: the cost of learning is not the number of samples or the computational power, it is the suboptimal reward that we get

Unfortunately, regret bounds for the aforementioned techniques are mostly elusive

Adaptive LQR control

We consider the setting of controlling an uncertain linear system

$$x_{t+1} = Ax_t + Bu_t + w_t$$

where w_t is noise, and A and B are unknown

We consider a quadratic cost whose associated regret is

$$R(T) = \sum_{t=0}^{T-1} x_t^\top Q x_t + u_t R u_t - TJ^\star$$

where $J^\star$ is the optimal average quadratic cost (Q and R are known)

Adaptive LQR control

(continued)

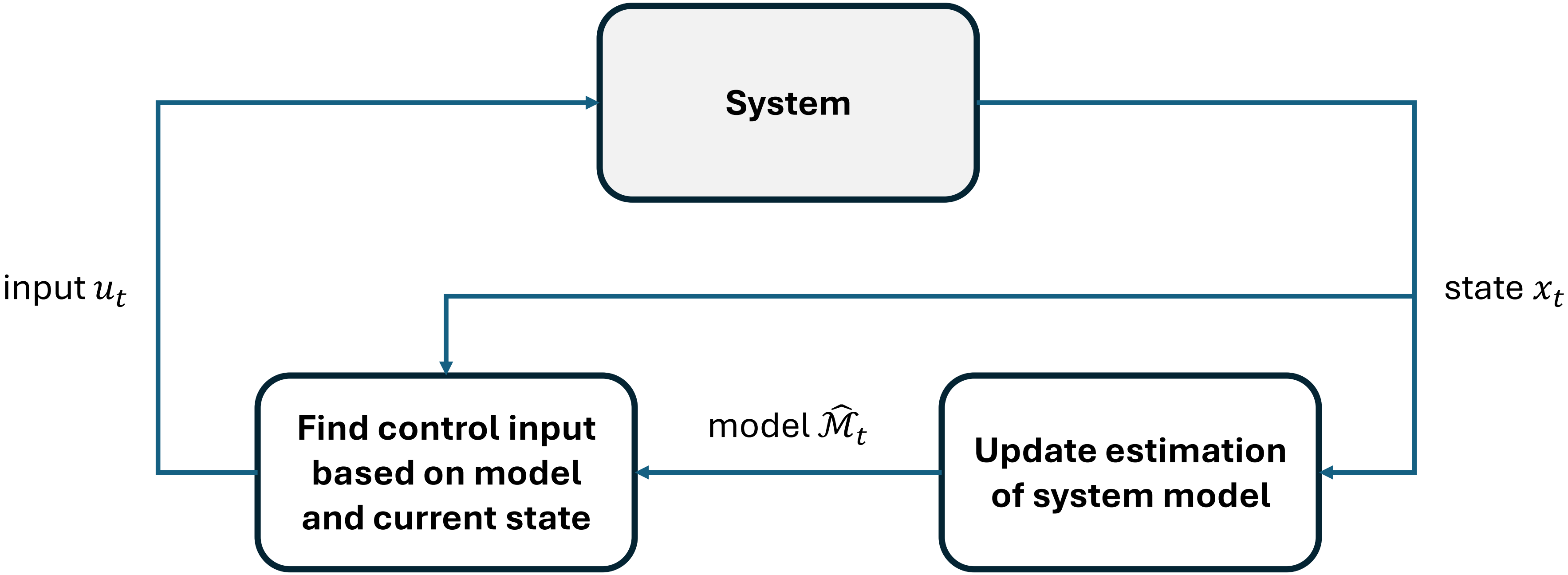
Model-based methods are expected to perform better for adaptive LQR because we have the prior knowledge that the system is linear so that we can learn the matrices A and B from previous data

We discuss two such methods:

- Robust adaptive LQR
- Certainty Equivalence LQR

and their associated regret bounds

Model-based methods



System

input u_t

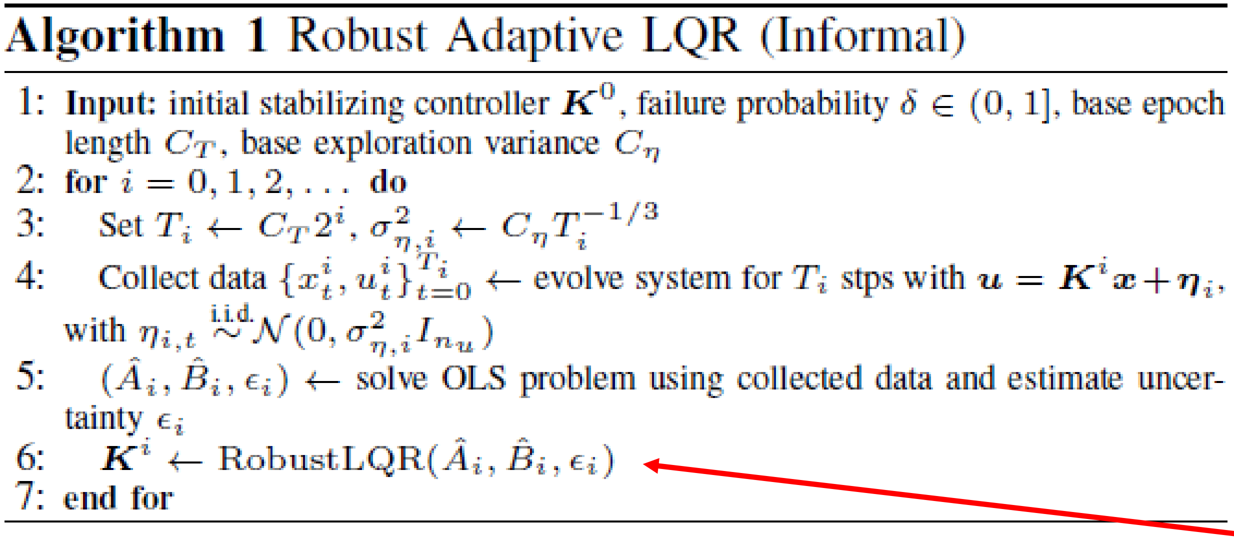
state x_t

model $\widehat{\mathcal{M}}_t$

Find control input based on model and current state

Update estimation of system model

Robust adaptive LQR



Algorithm 1 Robust Adaptive LQR (Informal)

```
Input: initial stabilizing controller $\boldsymbol{K}^0$, failure probability $\delta \in (0,1]$, base epoch length $C_T$, base exploration variance $C_\eta$
for $i = 0, 1, 2, \ldots$ do
    Set $T_i \leftarrow C_T 2^i$, $\sigma^2_{\eta,i} \leftarrow C_\eta T_i^{-1/3}$
    Collect data $\{x_t^i, u_t^i\}_{t=0}^{T_i} \leftarrow$ evolve system for $T_i$ stps with $\boldsymbol{u} = \boldsymbol{K}^i \boldsymbol{x} + \boldsymbol{\eta}_i$, with $\eta_{i,t} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma^2_{\eta,i} I_{n_u})$
    $(\hat{A}_i, \hat{B}_i, \epsilon_i) \leftarrow$ solve OLS problem using collected data and estimate uncertainty $\epsilon_i$
    $\boldsymbol{K}^i \leftarrow \mathrm{RobustLQR}(\hat{A}_i, \hat{B}_i, \epsilon_i)$
end for
```

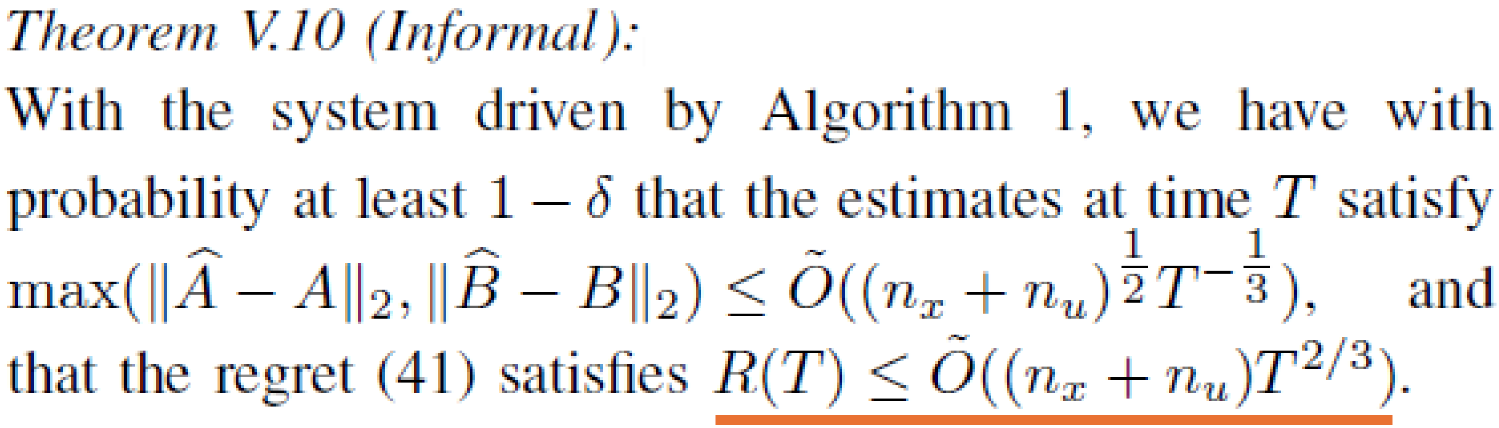
Identify $\hat{A}$ and $\hat{B}$ from data in the least-square sense

$$(\widehat{A}, \widehat{B}) \in \arg\min_{A,B} \frac{1}{2} \sum_{t=0}^{T-1} \|x_{t+1} - Ax_t - Bu_t\|_2^2 \,,$$

and define ϵ the error

Find the best worst-case controller for all systems that are at distance ϵ from $(\hat{A}, \hat{B})$

Analysis of robust adaptive LQR



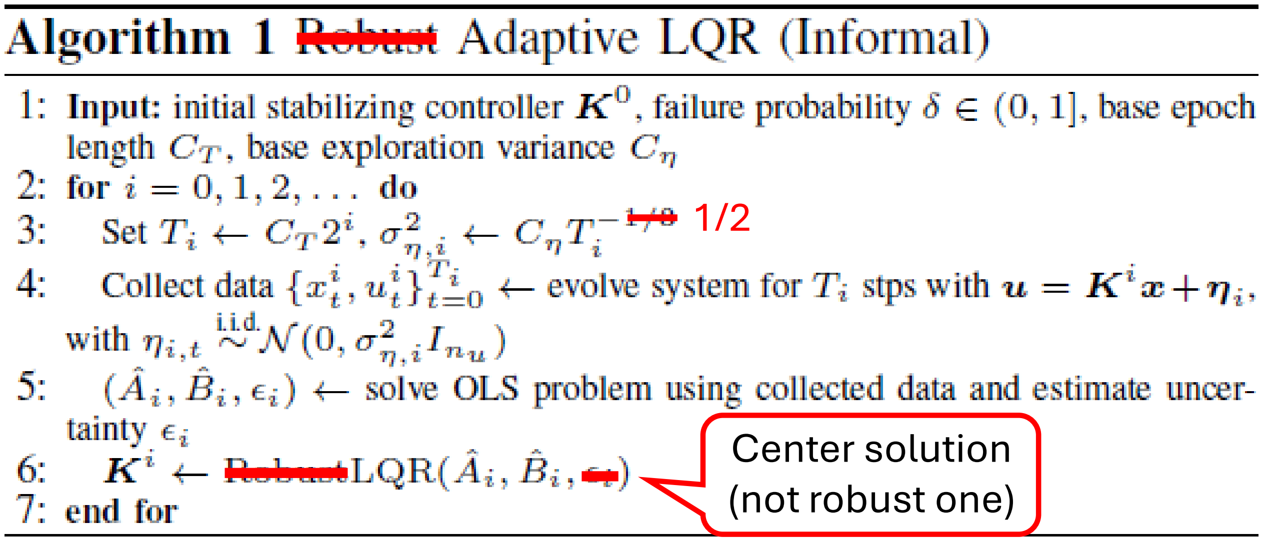
Theorem V.10 (Informal):
With the system driven by Algorithm 1, we have with probability at least $1-\delta$ that the estimates at time T satisfy $\max(\|\widehat{A}-A\|_2, \|\widehat{B}-B\|_2) \leq \tilde{O}((n_x+n_u)^{\frac{1}{2}}T^{-\frac{1}{3}})$, and that the regret (41) satisfies $\underline{R(T) \leq \tilde{O}((n_x+n_u)T^{2/3})}$.

sublinear regret $O\left(T^{2/3}\right)$

Reference: [2, Section V.B].

Certainty Equivalence LQR

CE

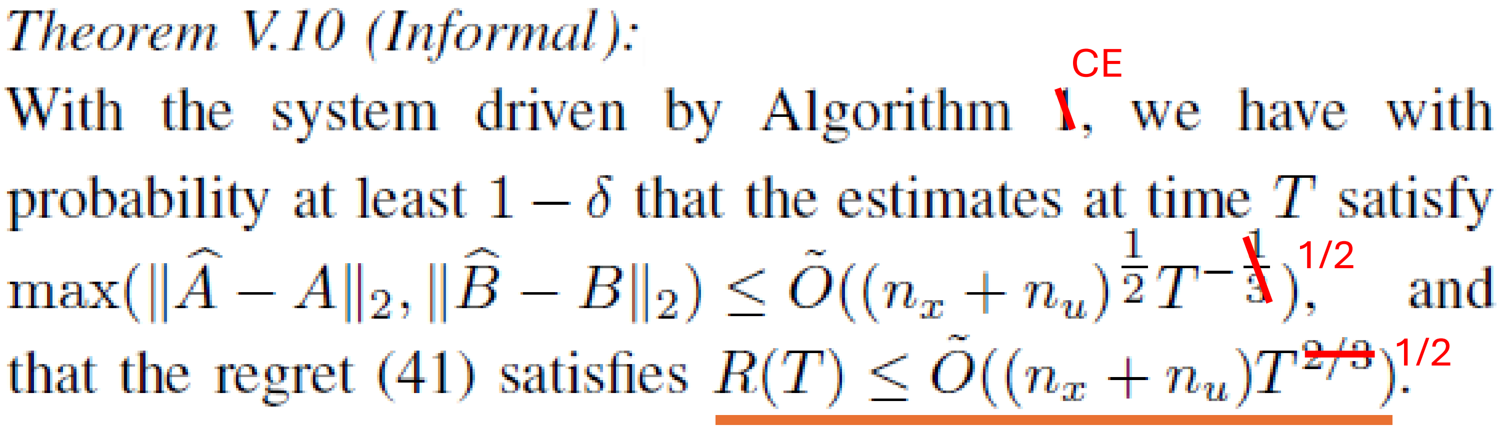


Algorithm 1 ~~Robust~~ Adaptive LQR (Informal)

```
Input: initial stabilizing controller $\boldsymbol{K}^0$, failure probability $\delta \in (0,1]$, base epoch length $C_T$, base exploration variance $C_\eta$
for $i = 0, 1, 2, \ldots$ do
    Set $T_i \leftarrow C_T 2^i$, $\sigma^2_{\eta,i} \leftarrow C_\eta T_i^{-\cancel{1/8}}$ 1/2
    Collect data $\{x_t^i, u_t^i\}_{t=0}^{T_i} \leftarrow$ evolve system for $T_i$ stps with $\boldsymbol{u} = \boldsymbol{K}^i \boldsymbol{x} + \boldsymbol{\eta}_i$, with $\eta_{i,t} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma^2_{\eta,i} I_{n_u})$
    $(\hat{A}_i, \hat{B}_i, \epsilon_i) \leftarrow$ solve OLS problem using collected data and estimate uncertainty $\epsilon_i$
    $\boldsymbol{K}^i \leftarrow$ ~~Robust~~LQR$(\hat{A}_i, \hat{B}_i, \cancel{\epsilon_i})$
end for
```

Center solution
(not robust one)

Analysis of Certainty Equivalence LQR

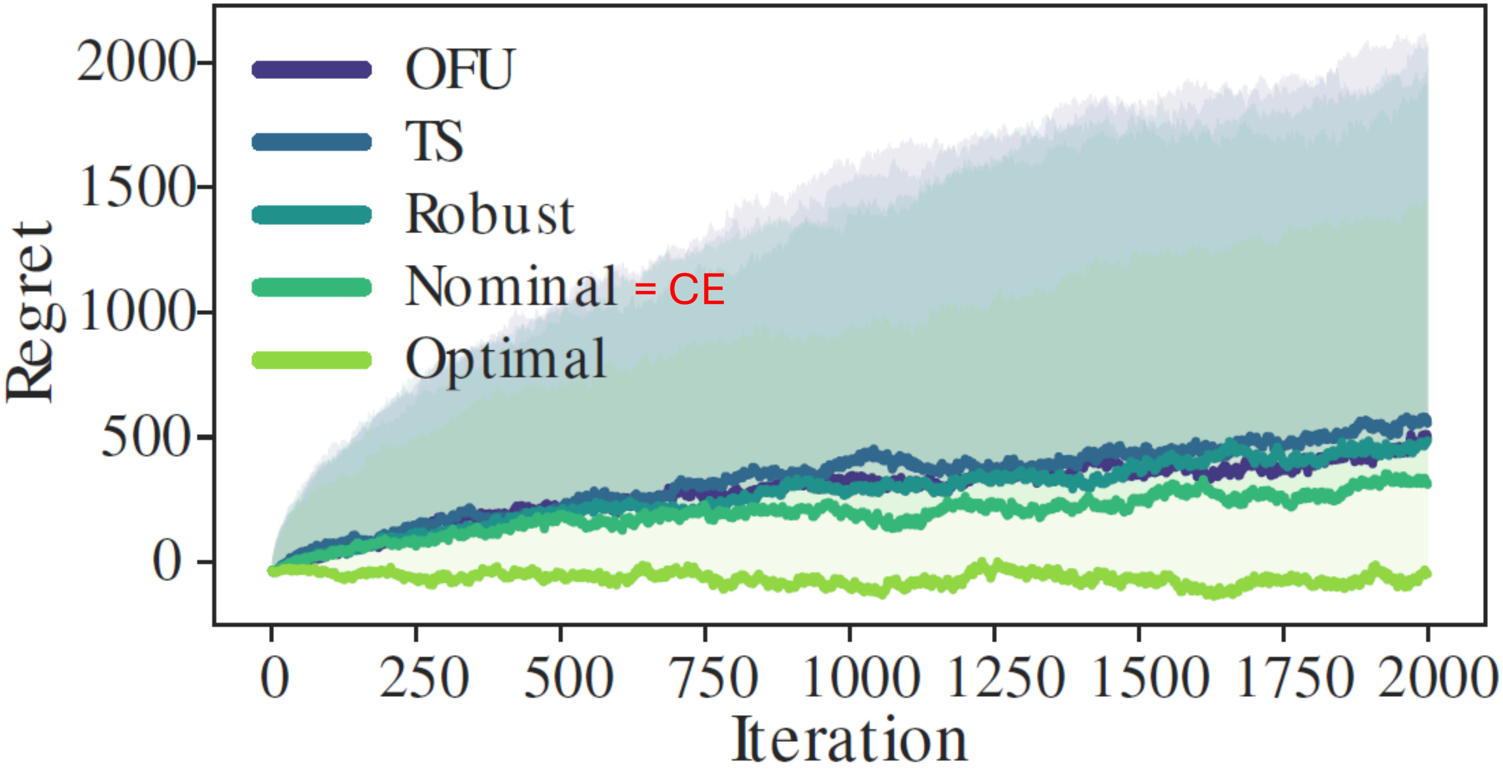


Theorem V.10 (Informal):
With the system driven by Algorithm ~~1~~ CE, we have with probability at least $1-\delta$ that the estimates at time T satisfy $\max(\|\widehat{A}-A\|_2, \|\widehat{B}-B\|_2) \le \tilde{O}((n_x+n_u)^{\frac{1}{2}} T^{-\cancel{\frac{1}{3}}\ 1/2})$, and that the regret (41) satisfies $\underline{R(T) \le \tilde{O}((n_x+n_u)T^{\cancel{2/3}\ 1/2})}$.

sublinear regret $O\left(T^{1/2}\right)$

Reference: [2, Section V.C].

Comparison of robust vs CE LQR



Discussion of model-based adaptive LQR

Regret grows slowerly with Certainty Equivalence LQR than with robust LQR: $O\left(T^{1/2}\right)$ vs $O\left(T^{2/3}\right)$

However, Certainty Equivalence LQR requires a good initial approximation of the model (ϵ small), whereas for robust LQR the initial uncertainty on the model can be larger

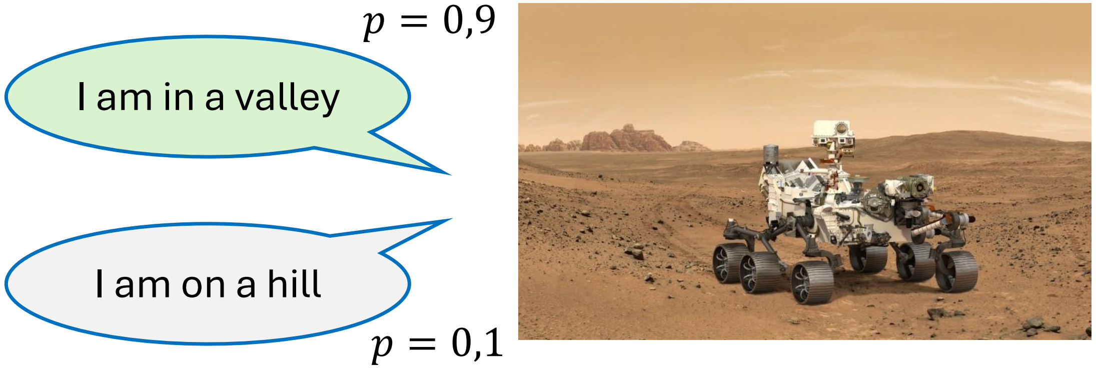
Partially observable stochastic systems

Often called POMDP (Partially Observable MDP)

Example: controlling a rover on Mars, but you get only noisy partial measurements of the position:

$$Y_n = g(X_n, W_n)$$

where $\boldsymbol{W}$ is i.i.d. noise



$p = 0{,}9$

I am in a valley

I am on a hill

$p = 0{,}1$

General POMDP model

State (X) and output (Y) dynamics:

$$\begin{aligned} X_{n+1} &= f(X_n, U_n, N_{n+1}) \\ Y_{n+1} &= g(X_{n+1}, W_{n+1}), \qquad n \geq 0, \end{aligned}$$

where $(\boldsymbol{N}, \boldsymbol{W})$ is i.i.d., and mutually independent

Admissible inputs:

$$U_n = \phi_n(Y_0, \ldots, Y_n)$$

(input depends only on current and past <u>observations</u>)

Belief state

Key (amazing) result: The only information you need to do optimal control of POMDP is the belief state $b_n(\cdot)$ at each step $n \geq 0$

Belief state: for each $x \in \mathsf{X}$,

$$b_n(x) = \mathsf{P}\{X_n = x \mid \mathcal{Y}_n\},$$

in which $\mathcal{Y}_n = \sigma(Y_k : k \leq n)$.

(probability of being in state x given current and past observations)

Dynamics of belief state

Result: The dynamics of the belief state is that of a fully observed deterministic Markov process with inputs $\boldsymbol{U}$ and $\boldsymbol{Y}$: formally,

there is a mapping $\mathcal{M}\colon \mathcal{S} \times \mathsf{Y} \times \mathsf{U} \to \mathcal{S}$ such that for each $n \geq 0$,

$$b_{n+1} = \mathcal{M}(b_n, Y_{n+1}, U_n)$$

Consequence:

for any function $F\colon \mathcal{S} \times \mathsf{Y} \to \mathbb{R}$,

probability of observing y' if state is x'

$$\mathsf{E}[F(b_{n+1}, Y_{n+1}) \mid \underbrace{\mathcal{Y}_n}_{\text{not used! (Markov)}} ;\ b_n = b,\ \ U_n = u\}] = \sum_{y'} \sum_{x,x'} b(x) P_u(x, x') q(y' \mid x') F(\mathcal{M}(b, y', u), y')$$

See [3, Proposition C.1].

Belief state is the only needed information

For each $n \geq 0$, let

$$V_n^\star \overset{\text{def}}{=} \min_{\boldsymbol{U}} \mathbb{E}\left[\, \sum_{k=n}^{\infty} c\big(X(n), U(n)\big) \mid \mathcal{Y}_n \,\right]$$

(optimal future total cost given current and past observations)

Key (amazing) result rephrased: there is a function $\mathcal{V}_n : \mathcal{S} \to \mathbb{R}$ indexed by n such that

$$V_n^\star = \mathcal{V}_n(b_n)$$

where b_n is the belief state at step n

See [3, Proposition C.2].

Belief state is the only needed information

(continued)

Consequence of key result: e.g., to find the value function for the optimal finite-horizon cost, you just need to solve:

$$\mathcal{V}_n(b) = \min_u \Big\{ \mathcal{C}(b,u) + \sum_{y'} \sum_{x,x'} b(x) P_u(x,x') q(y' \mid x') \mathcal{V}_{n-1}(\mathcal{M}(b,y',u)) \Big\},$$

where $\mathcal{C}(b,u) \overset{\text{def}}{=} \sum_x b(x) c(x,u)$

Conclusion of POMDP

Optimal control of POMDPs can thus be solved exactly as MDPs by working with the belief state

Caveat: the belief state is defined on a continuous state space even if the state space of the POMDP is finite.

This is a major challenge to address, and requires techniques for continuous MDP seen for instance in this course

If the state space of the POMPD is a vector space of finite dimension, then the belief state lives in a vector space of infinite dimension, except in some special cases (like LGQ)

LQG and the separation principle

For LQG, the belief state is fully described by its mean and covariance matrix. Hence, it lives in a space of dimension $\sim n^2$

Furthermore, we can show that the optimal total cost satisfies

$$V_n^\star = \mathcal{V}_n(b_n) = \mathcal{V}_n(\hat{x}_n)$$

where $\hat{x}_n \overset{\text{def}}{=} \int x b_n(x) \, \mathrm{d}x$ is the mean of the belief state

This is called the separation principle and justifies the LQG controller where only the estimated state $\hat{x}_n$ (obtained using Kalman filter) is used

The proof is based on the fact that $\hat{x}_{n+1}$ is a linear function of $\hat{x}_n$, Y_{n+1} and U_n, and $b_n(x)$ is symmetric around $\hat{x}_n$, i.e., $b_n(\hat{x}_n + a) = b_n(\hat{x}_n - a)$. Based on this, we can show that $\mathcal{V}_n(b_n) \triangleq \hat{x}_n^\top P_n \hat{x}_n$, where P_n is the value function of the associated (deterministic) LQR problem, solves the dynamic programming equation two slides earlier. Details are omitted.

References

[1] Pete Auer, Nicoló Cesa-Bianchi, and Paul Fischer. "Finite-time analysis of the multiarmed bandit problem". In: *Machine Learning* 47 (2002), pp. 235–256.

[2] Nikolai Matni et al. "From self-tuning regulators to reinforcement learning and back again". In: *2019 IEEE 58th Conference on Decision and Control (CDC)*. IEEE. 2019, pp. 3724–3740.

[3] Sean Meyn. *Control systems and reinforcement learning*. Cambridge University Press, 2022.