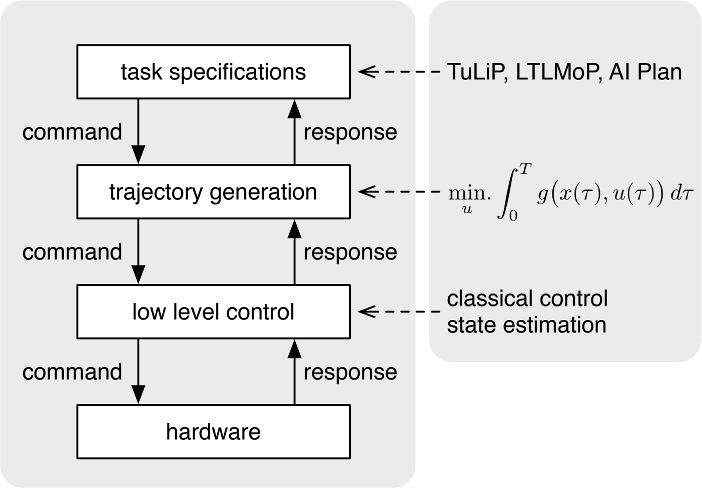
Toward Learning and Adaptation in Optimization Based Control

Ivan Papusha

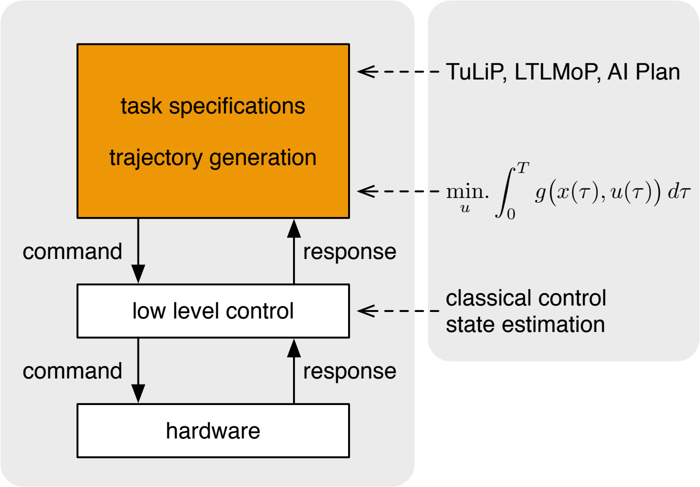
California Institute of Technology
Control and Dynamical Systems

Google, Mountain View, CA
January 12, 2016

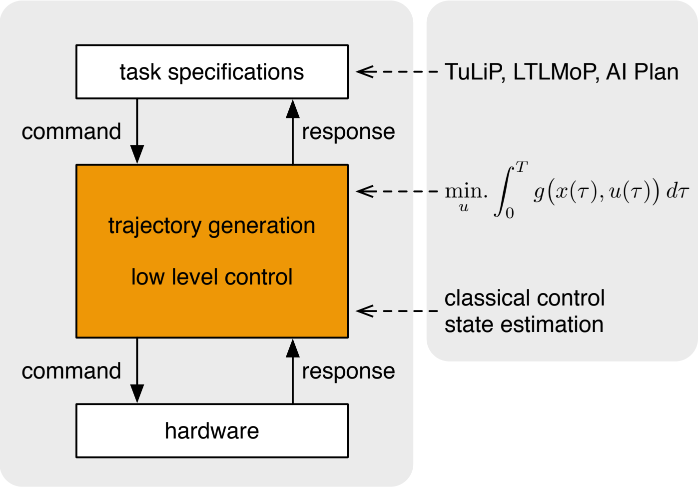
“Post”-modern control



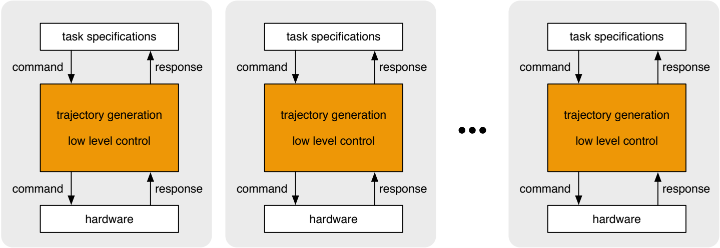
Optimal control + supervisory temporal logic



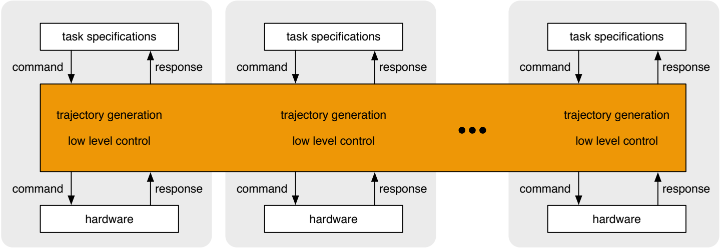
Optimal control + adaptation



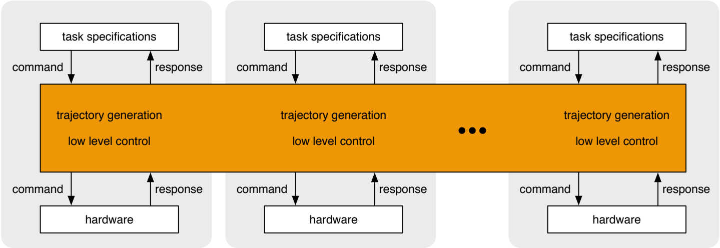
Optimal control + adaptation + multiagent



Optimal control + adaptation + multiagent + networking



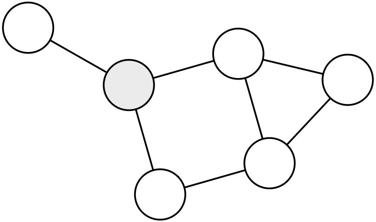
Optimal control + adaptation + multiagent + networking



networked adaptive systems

Applications of networked adaptive systems

- smartgrid: bootstrapping, disturbance rejection
- circuits: high performance phase locked loops
- robotics: distributed bootstrapping with consensus constraints
- adaptive systems: collaborative system identification



Learning safely: why?

Consider a (discrete time) linear dynamical system with state $x_t \in \mathbf{R}^n$ and control input $u_t \in \mathbf{R}^m$, for all $t = 0, 1, \ldots$,

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

We wish to stabilize the system, $x_t \to 0$ as $t \to \infty$. For simplicity, assume $B^T B$ is invertible.

A "reasonable" control scheme

At each time t, choose a control input u_t to make $\|x_{t+1}\|_2^2$ small,

$$u_t \in \underset{u_t \in \mathbf{R}^m}{\operatorname{argmin}} \|Ax_t + Bu_t\|_2^2$$

- in this case $u_t = u_t(x_t)$ only depends on the current state at time t
- optimal input is a constant state feedback

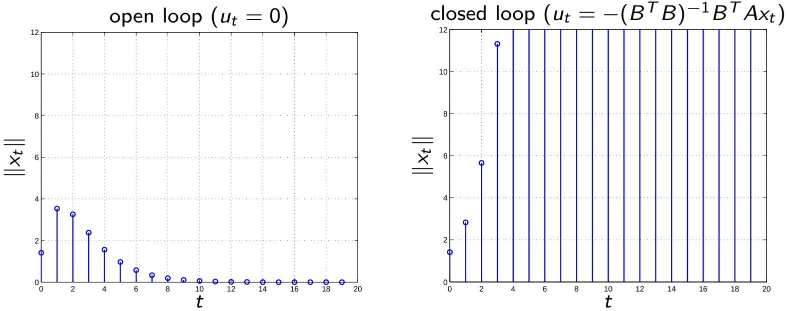
$$u_t = -(B^T B)^{-1} B^T A x_t$$

- closed loop system

$$x_{t+1} = \underbrace{(A - B(B^T B)^{-1} B^T A)}_{A+BK} x_t, \quad t = 0, 1, \ldots$$

Example instance

$$A = \begin{bmatrix} 0.5 & -3 \\ 0 & 0.5 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad x_0 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

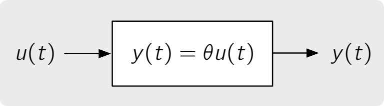


$$\rho(A) = 0.5 < 1$$

$$\rho(A - B(B^T B)^{-1} B^T A) = \rho\left(\begin{bmatrix} 0.25 & -1.75 \\ -0.25 & 1.75 \end{bmatrix}\right) = 2 \not< 1$$

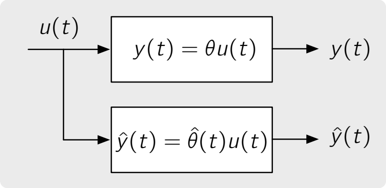
Identification model

- input-output model
$$y(t) = \theta u(t)$$
- at each time $t \geq 0$:
  - select input $u(t) \in \mathbf{R}$
  - measure $y(t) \in \mathbf{R}$
- **goal**: determine θ



Identification approach

- time-varying estimate $\hat{\theta}(t) \in \mathbf{R}$
- simulated output
$$\hat{y}(t) = \hat{\theta}(t)u(t)$$
- **our task**: make simulator match true model
$$(\hat{y}(t) - y(t))^2 \to 0 \quad \text{as} \quad t \to \infty$$



Unconstrained minimization

minimize the instantaneous cost

$$J(\hat{\theta}(t)) = \frac{1}{2}(\hat{y}(t) - y(t))^2 = \frac{1}{2}(\underbrace{\hat{\theta}(t) - \theta}_{\Delta\theta(t)})^2 u(t)^2$$

by gradient descent on $\hat{\theta}(t)$

$$\frac{d}{dt}\hat{\theta}(t) := -\gamma \frac{\partial J}{\partial \hat{\theta}(t)} = -\gamma \Delta\theta(t) u(t)^2,$$

where $\gamma > 0$ is the learning rate

Gradient learning rule

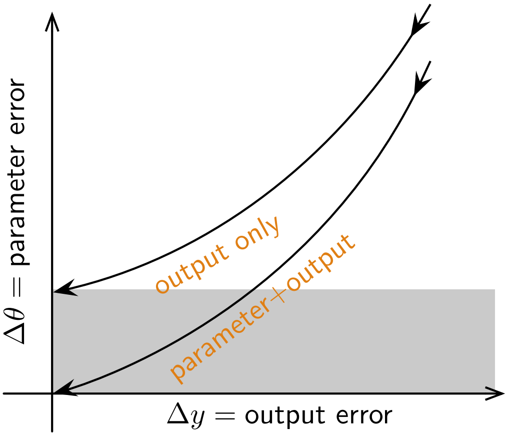
- gradient rule can be implemented online

$$\frac{d}{dt}\hat{\theta}(t) = -\gamma\Delta\theta(t)u(t)^2$$
$$= -\gamma\underbrace{(\hat{y}(t) - y(t))}_{\Delta y(t)}u(t)$$

- output error: $\Delta y(t)$
- parameter error: $\Delta\theta(t)$
- **fact**: output error (usually) converges, $\Delta y(t) \to 0$ as $t \to \infty$ (proof: Lyapunov argument $V(\Delta\theta) = \Delta\theta^2$)
- **question**: when does parameter error converge?

$$\Delta\theta(t) \stackrel{?}{\to} 0 \quad \text{as} \quad t \to \infty$$

Typical error curves



Simple condition on parameter convergence

- parameter error dynamics

$$\begin{aligned}\frac{d}{dt}\Delta\theta(t) &= \frac{d}{dt}\left(\hat{\theta}(t) - \theta\right) \\ &= -\gamma\Delta\theta(t)u(t)^2 \\ &\Downarrow\end{aligned}$$

$$\Delta\theta(t) = \exp\left\{-\gamma\int_0^t u(\tau)^2\,d\tau\right\}\Delta\theta(0)$$

- parameter error converges if $u(t)$ is **persistently exciting**:

$$\lim_{t\to\infty}\int_0^t u(\tau)^2\,d\tau = +\infty$$

Checking the memoryless system

- choose input $u(t) = c$, where $c \neq 0$ is a real constant

$$\lim_{t\to\infty} \int_0^t u(\tau)^2 \, d\tau = \lim_{t\to\infty} \int_0^t c^2 \, d\tau$$
$$= \lim_{t\to\infty} c^2 t$$
$$= +\infty \quad \checkmark$$

- excitation condition:

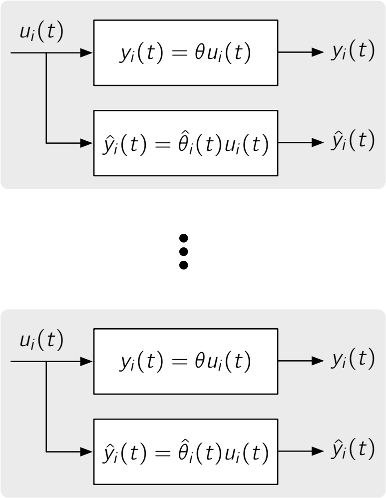
$$u(t) = c \text{ is persistently exciting} \quad \Leftrightarrow \quad c \neq 0$$

- persistence of excitation guarantees parameter convergence

Multiple agent identification model

- n agents labeled $i = 1, \ldots, n$
- at time $t \geq 0$, agent i can measure $x_i(t) \in \mathbf{R}^q$ and $y_i(t) \in \mathbf{R}$
- regressor: $\phi : \mathbf{R}^q \to \mathbf{R}^p$
- parameters: $\theta \in \mathbf{R}^p$
- true output:
$$y_i(t) = \theta^T \phi(x_i(t)), \quad i = 1, \ldots, n$$
- simulated output:
$$\hat{y}_i(t) = \hat{\theta}_i(t)^T \phi(x_i(t)), \quad i = 1, \ldots, n$$
- **goal**: parameter convergence $\|\theta_i(t) - \theta\| \to 0$ for all $i = 1, \ldots, n$.

Multiple agent identification model



Multiple agent consensus scheme

- each agent's parameter estimate is a sum of two terms

$$\frac{d}{dt}\hat{\theta}_i = \underbrace{-\gamma\phi(x_i)(\hat{y}_i - y_i)}_{\text{local information}} + \underbrace{\sum_{j\in\mathcal{N}_i} a_{ij}(\hat{\theta}_j - \hat{\theta}_i)}_{\text{neighboring information}}$$

- can be implemented **online**
- **respects** network communication structure

Interpretations of consensus scheme

- gradient descent on instantaneous cost

$$J(\hat{\theta}_1, \ldots, \hat{\theta}_n) = \underbrace{\sum_{i=1}^{n} (\hat{y}_i(t) - y_i(t))^2}_{\text{identification objective}} + \underbrace{\sum_{\{v_i, v_j\} \in \mathcal{E}} \frac{1}{2} a_{ij} \|\hat{\theta}_j(t) - \hat{\theta}_i(t)\|_2^2}_{\text{disagreement objective}}$$

- distributed PD control
- dynamical model fusion (*cf.* sensor fusion)
- augmented Lagrangian flow

$$\begin{array}{ll} \text{minimize} & \displaystyle\sum_{i=1}^{n} (\hat{y}_i(t) - y_i(t))^2 \\ \text{subject to} & \hat{\theta}_j(t) - \hat{\theta}_i(t) = 0, \quad i, j = 1, \ldots, n \end{array}$$

Convergence

candidate Lyapunov function:

$$V(\Delta\theta) = \sum_{i=1}^{n} \Delta\theta_i^T \Delta\theta_i$$

require:

- connected communication graph $\mathcal{G}$
- bounded (uniformly cts) regressors
- **collective** persistence of excitation

rate determined by:

- algebraic connectivity of $\mathcal{G}$
- minimum level of collective persistence of excitation

Collective persistence of excitation

proof idea:

- error dynamics are (for $\theta, \theta_i \in \mathbf{R}^1$)

$$\frac{d}{dt}\Delta\theta(t) = -(\underbrace{L}_{\text{rank } n-1} + \gamma\Phi(t))\Delta\theta(t)$$

- for $\Delta\theta \to 0$, bound in every direction $w \in \mathbf{R}^n$

$$w^T \left(\frac{1}{t - t_0} \int_{t_0}^{t} L + \gamma\Phi(\tau)\, d\tau \right) w > 0$$

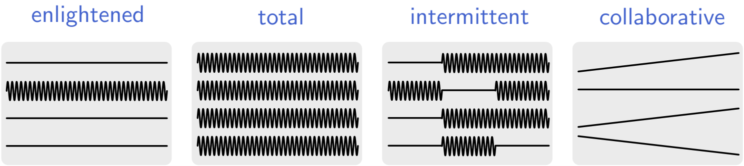
- **collective PE**: there exist positive real numbers $m_1, m_2 > 0$ such that for all $t_0 \geq 0$ and $t > t_0$ the matrix inequality

$$m_2 I \succeq \frac{1}{t - t_0} \int_{t_0}^{t} \sum_{i=1}^{n} \phi_i(\tau)\phi_i(\tau)^T\, d\tau \succeq m_1 I$$

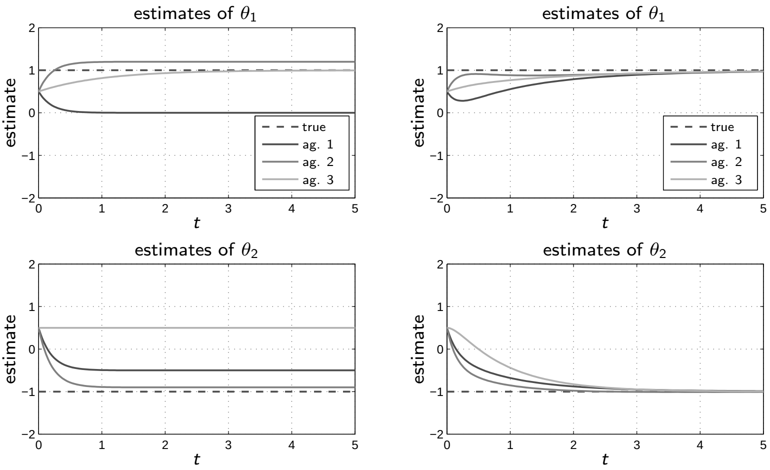
Excitation can be moved around

the following all imply parameter convergence:

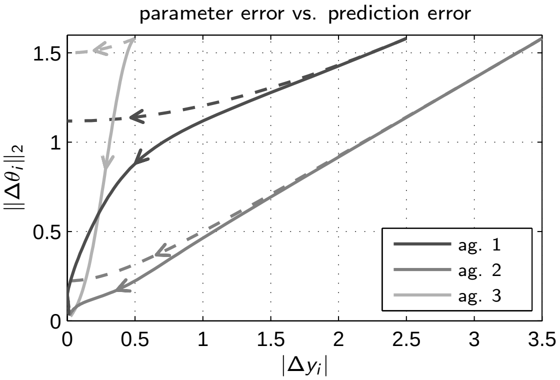
- enlightened: a few ϕ_i are persistently exciting,
- total: every ϕ_i is persistently exciting,
- intermittent: there exists an unbounded sequence of times $t_1, t_2, \ldots$ such that some ϕ_i obeys the collective PE condition in each interval $[t_k, t_{k+1}]$,
- collaborative: none of the ϕ_i is persistently exciting, but the collective PE condition still holds.



Example: collaborative PE (w/o and w/ consensus)



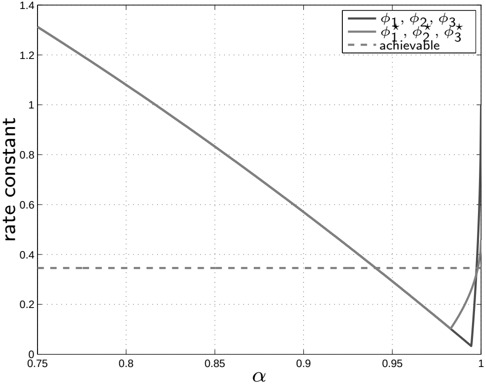
Example: collaborative PE error curves



Rate bound

take direction $w = \underbrace{\alpha \mathbf{1}/\sqrt{n}}_{\text{consensus subspace}} + \sum_{j=2}^{n} \beta_j v_j$

rate $\geq \inf_{|\alpha|\leq 1} \max \left\{ \lambda_2(1-\alpha^2),\ \gamma \frac{\alpha^2}{n} m_1 - 2\gamma m_2 \sqrt{\frac{\alpha^2}{n}(1-\alpha^2)} \right\}$



Model reference adaptive control

- Van der Pol (nonlinear) oscillators (n of them)

$$\ddot{x}_i = -x_i + \mu(1 - x_i^2)\dot{x}_i + u_i, \quad i = 1, \ldots, n$$

- reference model for each oscillator (place poles at $-1 \pm j$)

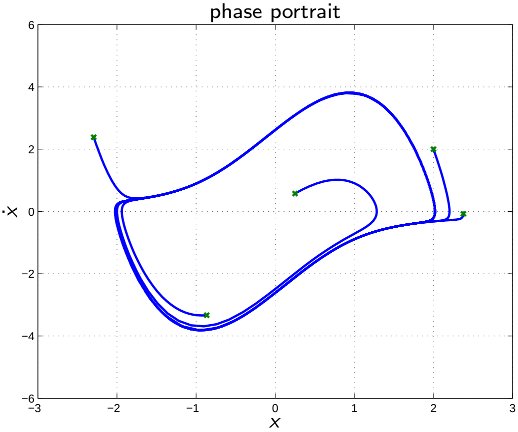
$$\ddot{x}_i^{\text{ref}} = -2(x_i^{\text{ref}} + \dot{x}_i^{\text{ref}}), \quad i = 1, \ldots, n$$

- regressors

$$\phi(x_i) = (1 - x_i^2)\dot{x}_i, \quad i = 1, \ldots, n$$

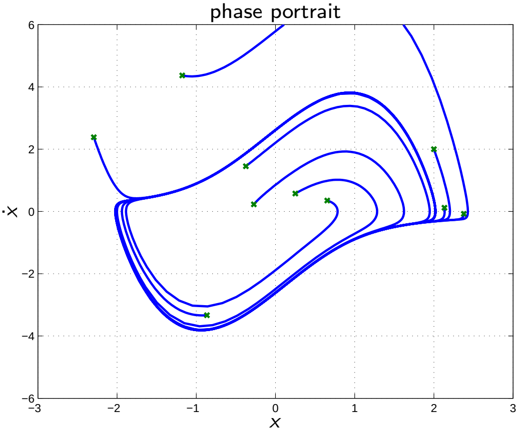
- adaptation: two control gains per agent & $\mu > 0$
- consensus on μ only

Model reference adaptive control



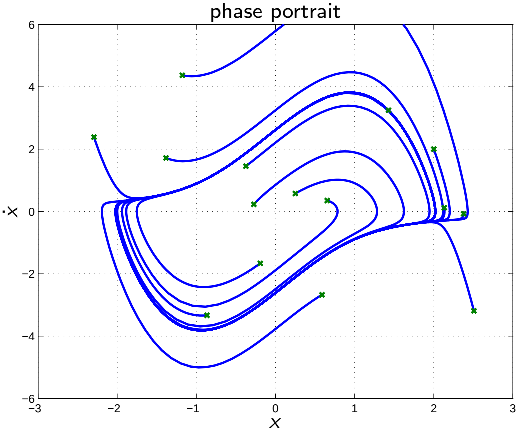
random initial conditions, $n = 5$ agents, open loop

Model reference adaptive control



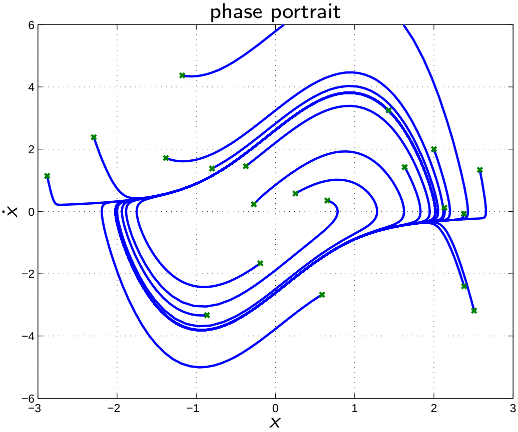
random initial conditions, $n = 10$ agents, open loop

Model reference adaptive control



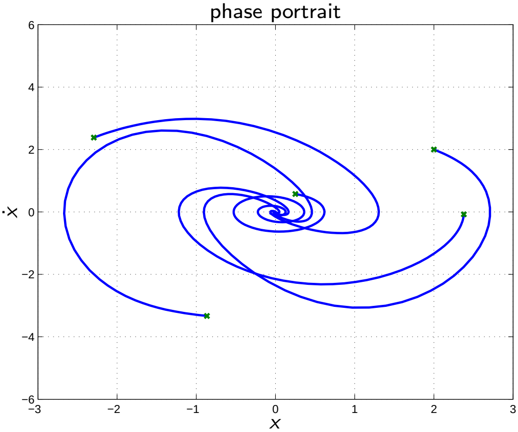
random initial conditions, $n = 15$ agents, open loop

Model reference adaptive control



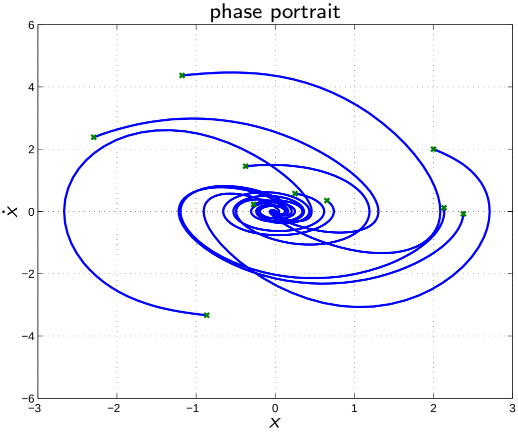
random initial conditions, $n = 20$ agents, open loop

Model reference adaptive control



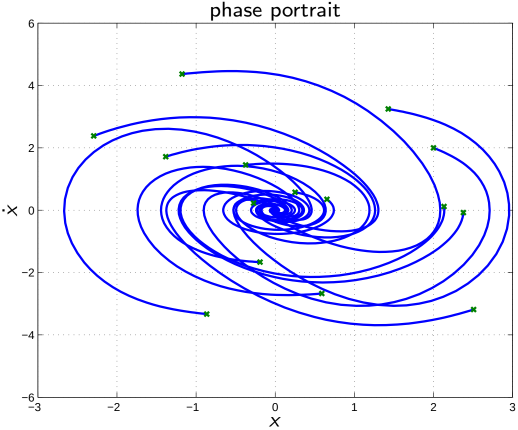
random initial conditions, $n = 5$ agents, MRAC

Model reference adaptive control



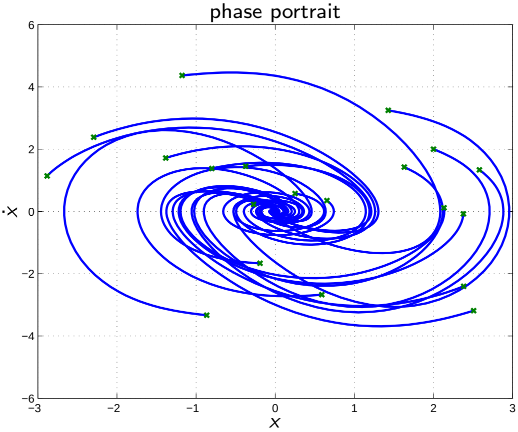
random initial conditions, $n = 10$ agents, MRAC

Model reference adaptive control



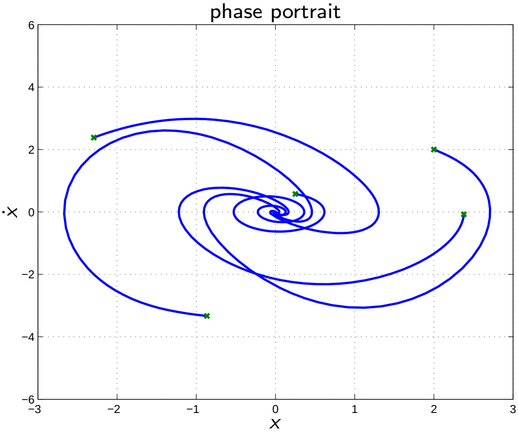
random initial conditions, $n = 15$ agents, MRAC

Model reference adaptive control



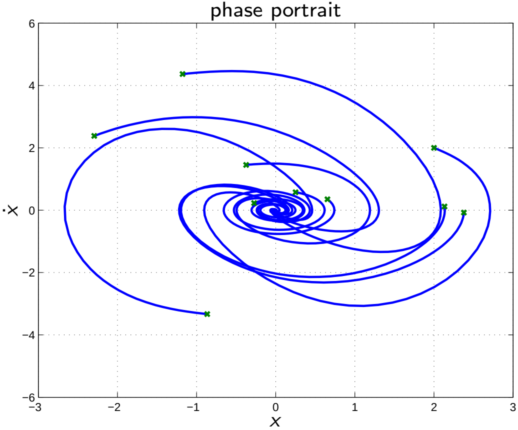
random initial conditions, $n = 20$ agents, MRAC

Model reference adaptive control



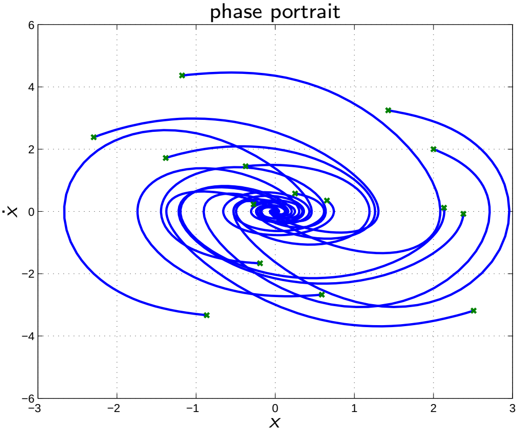
random initial conditions, $n = 5$ agents, MRAC + μ-consensus

Model reference adaptive control



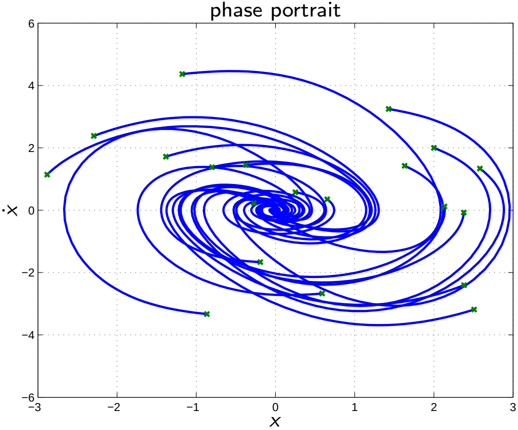
random initial conditions, $n = 10$ agents, MRAC + μ-consensus

Model reference adaptive control



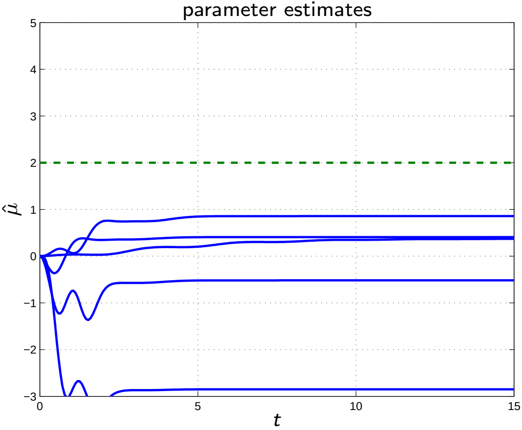
random initial conditions, $n = 15$ agents, MRAC + μ-consensus

Model reference adaptive control



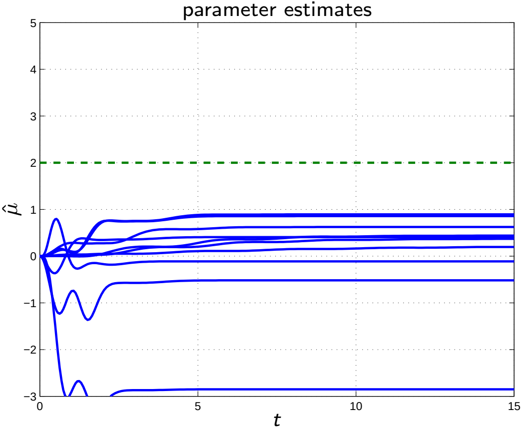
random initial conditions, $n = 20$ agents, MRAC + μ-consensus

Model reference adaptive control



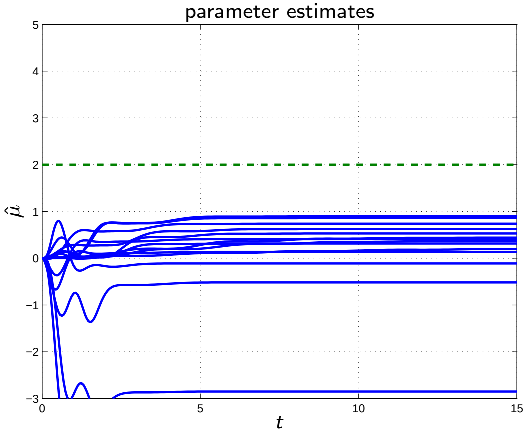
random initial conditions, $n = 5$ agents, MRAC

Model reference adaptive control



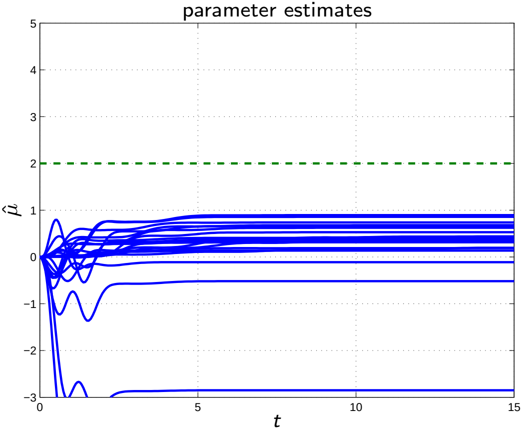
random initial conditions, $n = 10$ agents, MRAC

Model reference adaptive control



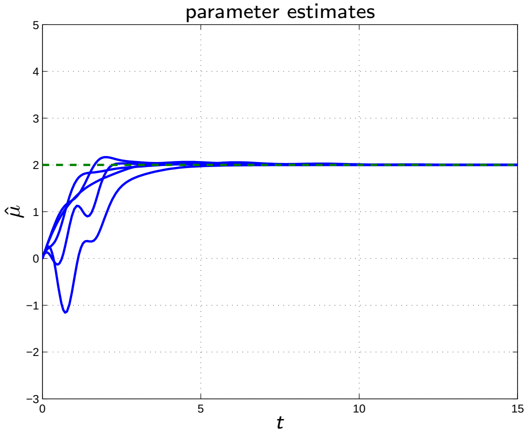
random initial conditions, $n = 15$ agents, MRAC

Model reference adaptive control



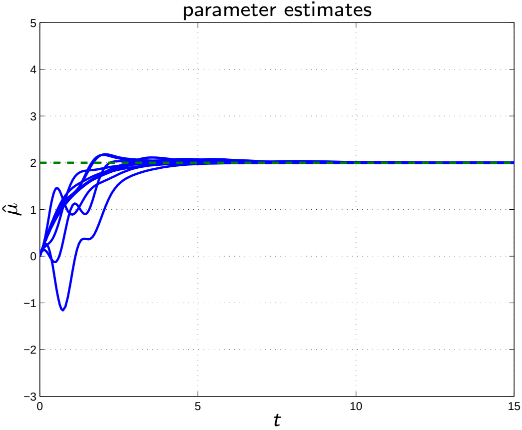
random initial conditions, $n = 20$ agents, MRAC

Model reference adaptive control



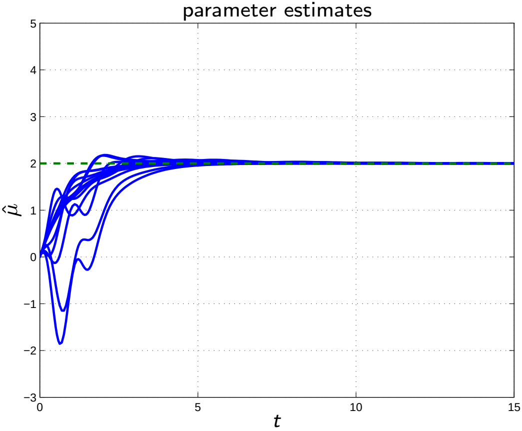
random initial conditions, $n = 5$ agents, MRAC + μ-consensus

Model reference adaptive control



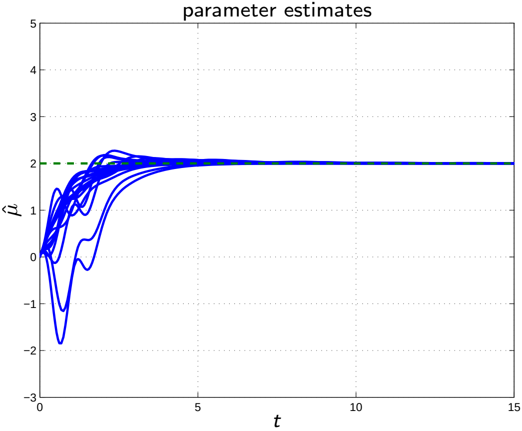
random initial conditions, $n = 10$ agents, MRAC + μ-consensus

Model reference adaptive control



random initial conditions, $n = 15$ agents, MRAC + μ-consensus

Model reference adaptive control



random initial conditions, $n = 20$ agents, MRAC + μ-consensus

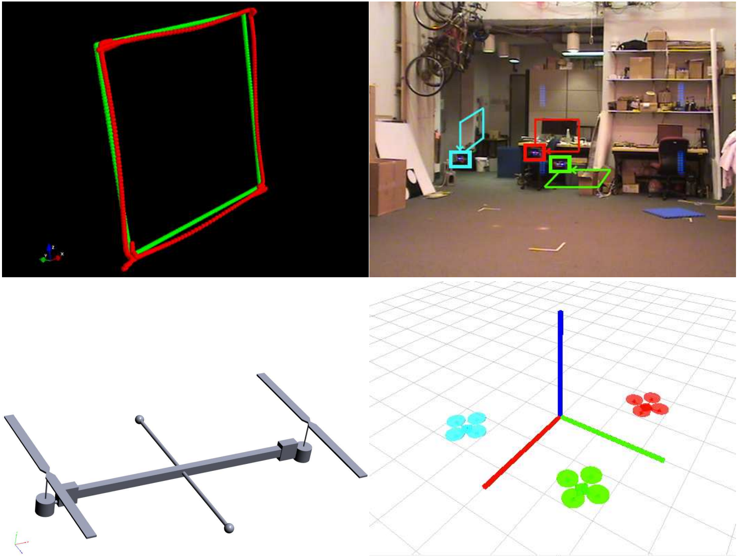
Summary

- simple idea: defined by

$$\hat{\theta}^{(t+1)} := \text{classical update rule} + \text{consensus}$$

- fundamentally nonlinear analysis and tools (mature theory)
- **future directions**:
  - quantitative analysis of noise effects (often) unchanged
  - engineer systems where the network does not fight adaptation
  - adaptation: graceful degradation when network fails
  - network: source of extra performance and robustness

Experiments with flying machines

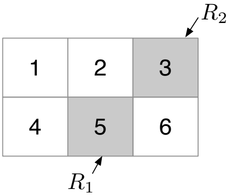


Approximate Dynamic Programming with Guarantees

Finite state Markov Decision Processes

- finite state space $\mathcal{X} = \{1, \ldots, n\}$
- finite action space $\mathcal{U}(i) \subseteq \mathcal{U} = \{1, \ldots, m\}$ available at each state i
- probability of transition $p_{ij}(u)$ from state i to state j under control action $u \in \mathcal{U}(i)$
- incurred stage cost $g(i, u, j)$

example. gridworld



$$\mathcal{X} = \{1, \ldots, 6\}, \quad \mathcal{U} = \{N, S, E, W\}, \quad p_{ij}(u) \in \{0.8, 0.1, 0.1\}$$

Deterministic policies

A policy is a sequence $\pi = \{\mu_0, \mu_1, \ldots\}$ where each $\mu_t : \mathcal{X} \to \mathcal{U}$ is a function that maps a state i to an available action in $\mathcal{U}(i)$.

- Given a policy π, the sequence of states $\{i_0, i_1, \ldots\}$ is a Markov chain with transition probabilities

$$\mathbf{P}(i_{t+1} = j \mid i_t = i) = p_{ij}(\mu_t(i)).$$

- for a given policy $\pi = \{\mu_0, \mu_1, \ldots\}$, we should have

$$\sum_{j=1}^{n} p_{ij}(\mu_t(i)) = 1, \quad \text{for all } i = 1, \ldots, n.$$

example. feasible gridworld policy that gets to $R2$

$$\begin{aligned} &\mu_t(1) = 2, \quad \mu_t(2) = 3, \quad \mu_t(3) = 3, \\ &\mu_t(4) = 5, \quad \mu_t(5) = 6, \quad \mu_t(6) = 3, \quad \text{for all } t = 0, 1, \ldots \end{aligned}$$

Policy cost and stationary policies

- The expected cost of a policy when starting from an initial state i is

$$V^{\pi}(i) = \mathbf{E}\left[\sum_{t=0}^{\infty} \gamma^t g(i_t, \mu_t(i_t), i_{t+1}) \,\middle|\, i_0 = i\right],$$

  where $\gamma \in (0, 1]$ is a discount factor.
- for the infinite horizon case, it is often convenient to consider stationary policies $\pi = \{\mu, \mu, \ldots\}$ and $\gamma < 1$.

example. the policy $\mu_t = \mu$ from the last slide is stationary since it is the same for all $t = 0, 1, \ldots$

Value function

The value function is defined as

$$V^\pi(i) = \mathbf{E}\left[\sum_{t=0}^{\infty} \gamma^t g(i_t, \mu_t(i_t), i_{t+1}) \,\middle|\, i_0 = i\right],$$
$$= \sum_{t=0}^{\infty} \sum_{j=1}^{n} p_{i_t j}(\mu_t(i_t)) \gamma^t g(i_t, \mu_t(i_t), j)$$

- we can think of V^π as a vector in $\mathbf{R}^n$, where each component $V^\pi(i)$ corresponds to the expected cost-to-go starting at state i
- The goal is to find a policy that minimizes the expected cost-to-go,

$$V^*(i) = \min_\pi V^\pi(i).$$

Bellman operator

The optimal cost-to-go satisfies the Bellman equation

$$V^*(i) = \min_{u \in \mathcal{U}(i)} \mathbf{E}[g(i,u,j) + \gamma V^*(j) \mid i, u]$$

$$= \min_{u \in \mathcal{U}(i)} \sum_{j=1}^{n} p_{ij}(u)(g(i,u,j) + \gamma V^*(j)), \quad \text{for all } i = 1, \ldots, n,$$

with the corresponding optimal policy at step t given by

$$\mu_t^*(i) = \operatorname*{argmin}_{u \in \mathcal{U}(i)} \mathbf{E}[g(i,u,j) + \gamma V^*(j) \mid i, u], \quad \text{for all } i = 1, \ldots, n.$$

Value iteration

For any value function vector $(V(1), \ldots, V(n))$ define the vector $\mathcal{T}V$ by the Bellman operator,

$$(\mathcal{T}V)(i) = \min_{u \in \mathcal{U}(i)} \mathbf{E}[g(i,u,j) + \gamma V(j) \mid i, u].$$

Thus the Bellman equation reads $V = \mathcal{T}V$.

- value iteration

$$V^{(k+1)} = \mathcal{T}V^{(k)}, \quad k = 0, 1, \ldots.$$

- for any starting guess $V^{(0)}$, the sequence $\{V^{(0)}, V^{(1)}, \ldots\}$ converges to V^*.
- Under some regularity assumptions and an infinite horizon, this equation has a unique solution V^* with a corresponding stationary policy π^*.

Approximating from below

Any function V that satisfies the Bellman inequality

$$V \leq \mathcal{T} V$$

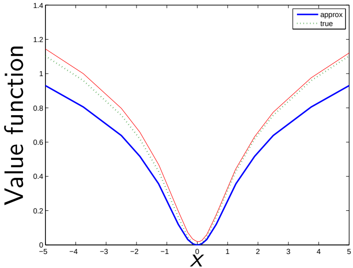
automatically satisfies $V \leq V^*$

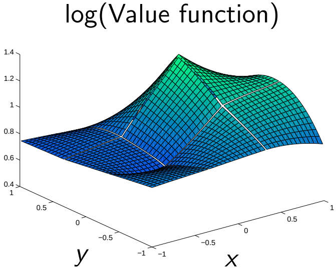
- V is a componentwise lower bound on V^*
- recursively apply $\mathcal{T}$ to both sides and use the monotonicity property,

$$V \leq \mathcal{T} V \leq \mathcal{T}^2 V \leq \cdots = V^*.$$

- **monotonicity**. if $V_1 \leq V_2$, then $\mathcal{T} V_1 \leq \mathcal{T} V_2$ (componentwise)
- the Bellman inequality defines a class of underestimators of V^*, one of which is V^* itself
- underestimators capture a class capture a *performance bound* on the original decision problem
- trivial performance bound: $V = 0$.

Bounds on the value function





Approximating from above

Similarly, any function that satisfies the reverse Bellman inequality

$$\mathcal{T}V \leq V$$

automatically satisfies $V^* \leq V$.

- componentwise upper bound on V^*
- recursively apply $\mathcal{T}$ to both sides of and use the monotonicity property,
$$V^* = \cdots \leq \mathcal{T}^2 V \leq \mathcal{T}V \leq V.$$
- overestimators correspond to suboptimal policies, because their value is greater than or equal to the optimal value

Bound optimization by linear programming

We can attempt to recover V^* by optimizing over the class of value function underestimators,

$$\begin{array}{ll} \text{maximize} & V \\ \text{subject to} & V \leq \mathcal{T}V, \end{array}$$

If the transition probabilities and stage costs are known, then we can rewrite as a linear program (LP),

$$\begin{array}{ll} \text{maximize} & \displaystyle\sum_{i=1}^{n} w(i)V(i) \\ \text{subject to} & V(i) \leq \displaystyle\sum_{j=1}^{n} p_{ij}(u)(g(i,u,j) + V(j)), \\ & \qquad \forall i = 1, \ldots, n, \forall u \in \mathcal{U}(i), \end{array}$$

- variables $V(1), \ldots, V(n)$
- weights $w(1), \ldots, w(n)$ are arbitrary (as long as they are positive)
- number of linear constraints is $O(nm)$, number of variables $O(n)$

Optimization with known transition probabilities

Related underapproximation LP

$$\begin{aligned}
\text{maximize} \quad & \sum_{i=1}^{n} w(i) \sum_{k=1}^{N} \alpha_k \phi_k(i) \\
\text{subject to} \quad & \sum_{k=1}^{N} \alpha_k \phi_k(i) \leq \sum_{j=1}^{n} p_{ij}(u) \left(g(i,u,j) + \sum_{k=1}^{N} \alpha_k \phi_k(j) \right), \\
& \forall i = 1, \dots, n, \forall u \in \mathcal{U}(i),
\end{aligned}$$

- restrict the class of underestimators by further specifying an approximating basis,

$$\widetilde{V}(i) = \sum_{k=1}^{N} \alpha_k \phi_k(i), \quad \phi_k : \mathcal{X} \to \mathbf{R}$$

- number of linear constraints $O(nm)$, number of variables $O(N)$
- ideally, $N \ll n$
- true value V^* is recovered if it is in the span of the basis functions

Uniform approximation guarantees

To get guarantees on approximation accuracy, simultaneously find functions V^+ and V^- in an approximating class (*e.g.*, relative to a fixed basis) such that

$$V^- \leq V^* \leq V^+,$$

and the difference between V^+ and V^- is as small as possible:

$$\begin{array}{ll} \text{minimize} & \max_i \{V^+(i) - V^-(i)\} \\ \text{subject to} & V^- \leq \mathcal{T} V^- \\ & \mathcal{T} V^+ \leq V^+ \\ & V^-, V^+ \in \mathcal{C} \end{array}$$

- variables V^+ and V^-
- $\mathcal{C} \subseteq \mathbf{R}^n$ represents (*e.g.*, basis) restrictions on the approximating class
- optimal value ϵ^* is measure of approximation error over all states
- **extension**. operate at specified level of suboptimality $\leq \epsilon$

Aside: robust LP

Consider a linear program in inequality form,

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & a_i^T x \leq b_i, \quad i = 1, \ldots, m \end{array}$$

over the variable $x \in \mathbf{R}^n$, where c, b_i are fixed, and a_i are known to lie in ellipsoids,

$$a_i \in \mathcal{E}_i = \{\bar{a}_i + P_i u \mid \|u\|_2 \leq 1\}.$$

robust linear programming

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & a_i^T x \leq b_i, \text{ for all } a_i \in \mathcal{E}_i, i = 1, \ldots, m \end{array}$$

Aside: robust LP

We can rewrite the robust LP,

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & a_i^T x \leq b_i, \text{ for all } a_i \in \mathcal{E}_i, i = 1, \ldots, m \end{array}$$

as an SOCP,

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & \bar{a}_i^T x + \|P_i^T x\|_2 \leq b_i, \quad i = 1, \ldots, m \end{array}$$

- notably, the problem is convex
- additional norm terms act as regularization constraints
- efficient solution techniques for medium to large m, n.

Optimization with unknown transition probabilities

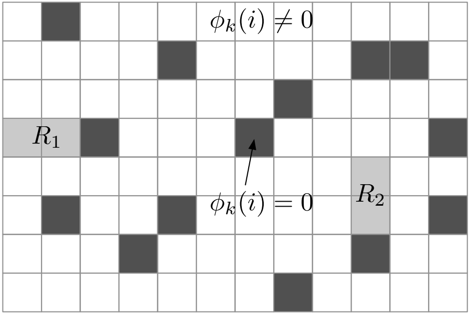
If the transition probabilities are known to lie in an ellipsoid, then we can rewrite the underapproximation LP

$$
\begin{array}{ll}
\text{maximize} & \sum_{i=1}^{n} w(i)V(i) \\
\text{subject to} & V(i) \le \sum_{j=1}^{n} p_{ij}(u)(g(i,u,j) + V(j)), \\
& \qquad \forall i = 1,\ldots,n, \forall u \in \mathcal{U}(i),
\end{array}
$$

as a robust LP (viz., SOCP)

- ellipsoidal outbound probabilities: $p_{i:}(u) \in \mathcal{E}_i(u)$, $\forall i$, $\forall u$
- special case: lower and upper bounds on transition probabilities $p_{ij}(u) \in [\underline{p}_{ij}(u), \overline{p}_{ij}(u)]$
- double-sided LP has *guaranteed* approximation error via objective

Example



- basis vectors ϕ_k encode state membership constraints
- pooling over free regions decreases basis complexity
- policy is robust wrt perturbations in $p_{ij}(u)$
- quantitative measure of suboptimality

Extensions

- Specified basis functions for state constraints
- automaton product MPDs for logic specifications (slightly generalized version of [Wolff et al.'12]). The engineering challenge is to pick appropriate basis vectors.
- enforce the LP constraints only at certain specified states—more tractable with loss of bound guarantees.
- attempt to discover $p_{ij}(u)$ similarly to [Fu et al., '15] PAC-MDP learning, either by simulation or repeated probing.
- It is also possible to talk about the probability of satisfaction by incorporating it, directly or by proxy, into the additive stage costs.
- Similarly, a proxy for exploration can also be part of the objective.

Thanks!

more information: Ivan Papusha, Richard M. Murray

www.ivanpapusha.com

funding: NDSEG, Powell Foundation, STARnet/TerraSwarm, Boeing