Sampling-based approximate optimal control under temporal logic constraints

Jie Fu[1] Ivan Papusha[2] Ufuk Topcu[2]

[1]Worcester Polytechnic Institute

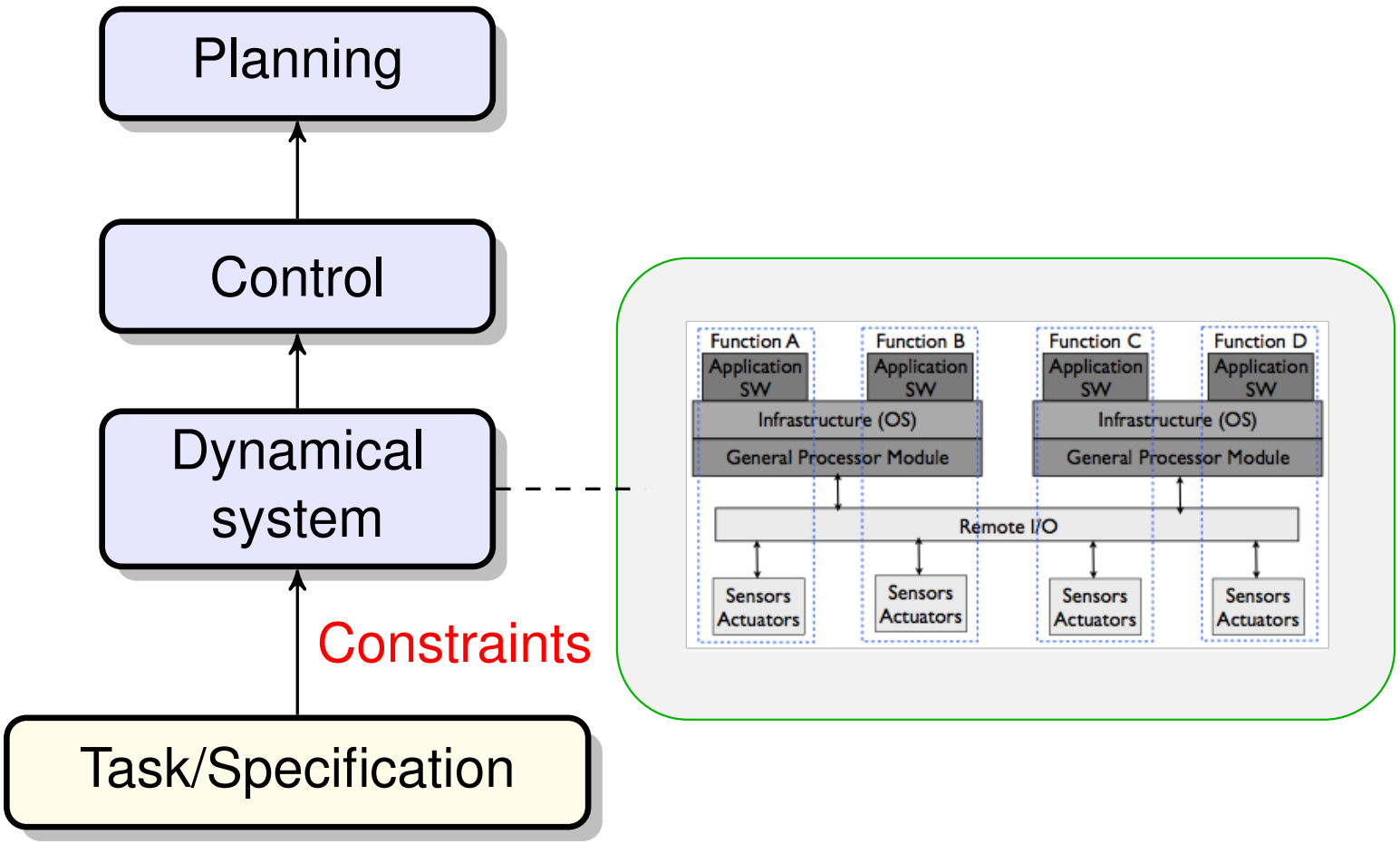
[2]University of Texas at Austin

HSCC, 2017

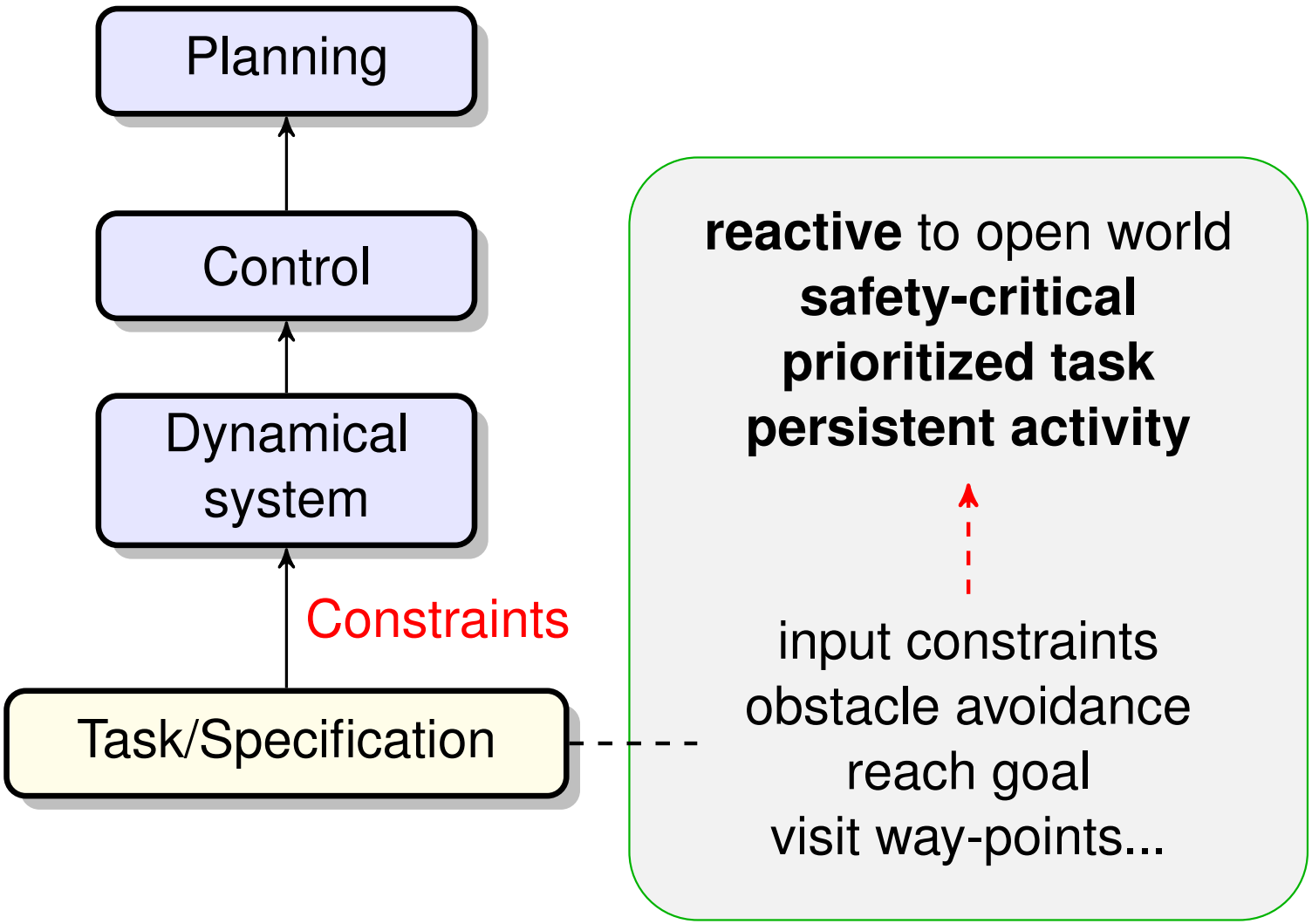
Outline

1. Introduction
2. Hybrid system formulation
3. Approximate-optimal temporal logic planning
4. Integrate importance sampling

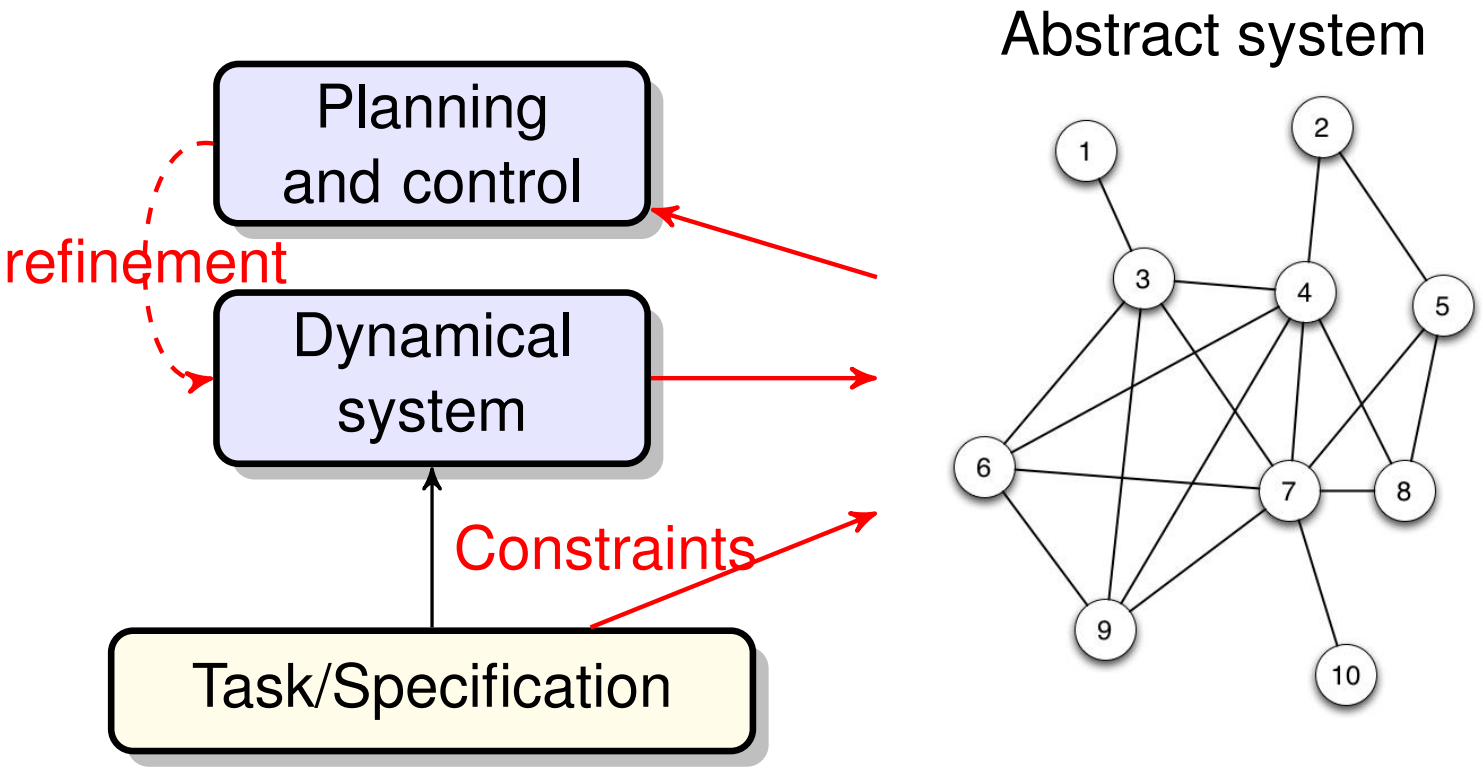
Challenges: Large-scale systems



Challenges: Stringent and complex constraints



A widely explored approach



- Abstraction results in high-dimensional systems — Scalability.
- Seek an alternative: **No explicit abstraction!**
- no free lunch: must resort to approximations

Problem formulation

- A dynamical system: $\dot{x} = f(x, u)$, x: state, u: input.
- Performance measured by cost function

$$J(x, u) = \int_0^T \ell(x, u)dt + G(x(T), u(T))$$

- **Labeling**: Atomic propositions $\mathcal{AP}$
- **Specifications in temporal logic**: φ over $\mathcal{AP}$.
- **Goal**: Design a controller $u : [0, T] \to U$ such that the system minimizes the cost while **strictly satisfying** φ.

From LTL to Automaton

For a given formula φ (subset of LTL), the automaton is

$$A_\varphi = \langle Q, 2^{AP}, \delta, q_0, F \rangle.$$

- $Q = \{q_0, q_1, q_2, q_3, q_4\}$
- $\delta : Q \times 2^{AP} \to Q$ (labeled directed edges).
- q_0: initial state
- F: accepting states.

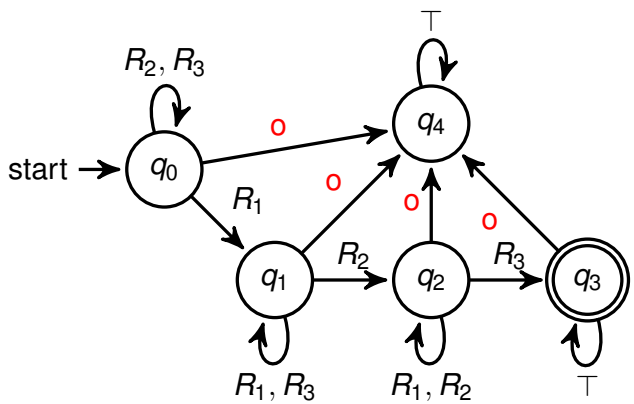


Figure: $\varphi = \Diamond(R_1 \wedge \Diamond(R_2 \wedge \Diamond R_3)) \wedge \Box\neg\text{Obs}$.

Co-safe LTL fragment

An **LTL co-safety formula** can be represented by a deterministic finite state automaton,

$$\underbrace{\rho = q_0 q_1 \ldots q_n}_{\text{a finite good prefix}} \text{ is accepting if } q_n \in F,$$

where F is the set of accepting states in A_φ.

- can contain the X (next), $\Diamond$ (eventually), U (strong until).
- negation only occurs in the front of atomic propositions.

Limited expressiveness:

- Reachability: $\Diamond$goal.
- Safety: $\neg$obs U goal.
- Sequencing: $\Diamond(\text{goal}_1 \wedge \Diamond(\text{goal}_2 \wedge \Diamond\text{goal}_3))$.
- cannot specify fairness and recurrence properties.

Approximate optimal control for co-safe LTL

Given a dynamical system

$$\dot{x} = f(x, u)$$

Find u^* that minimizes

$$J(x_0, u) = \int_{t=0}^{T} \ell(x, u) dt$$

subject to $\exists t_0, t_1, \ldots, t_k$

$$\delta(q_i, x(t_i)) = q(t_{i+1}),$$
$$i = 0, 1, ..., k$$
$$q_f \in F, x(t_k) = x_f.$$

Approximate optimal control for co-safe LTL

Given a dynamical system

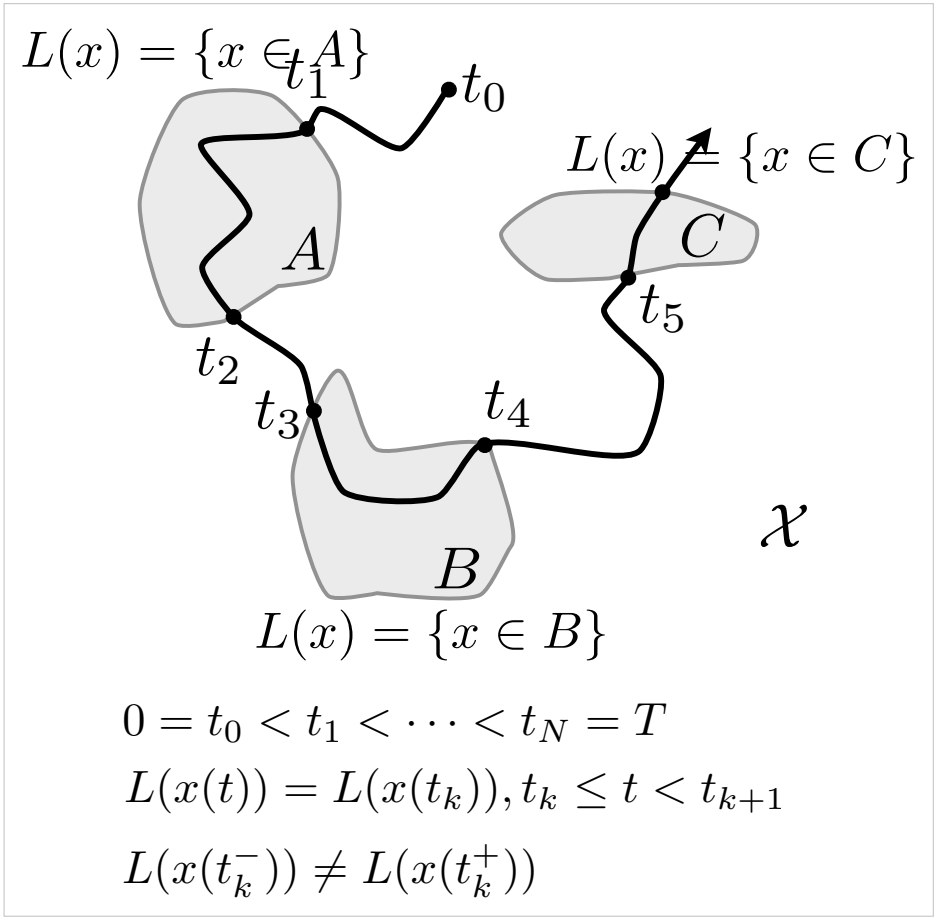
$$\dot{x} = f(x, u)$$

Find u^* that minimizes

$$J(x_0, u) = \int_{t=0}^{T} \ell(x, u) dt$$

subject to $\exists t_0, t_1, \ldots, t_k$

$$\delta(q_i, x(t_i)) = q(t_{i+1}),$$
$$i = 0, 1, ..., k$$
$$q_f \in F, x(t_k) = x_f.$$



How do we develop a scalable control method that avoids discretizing the state space?

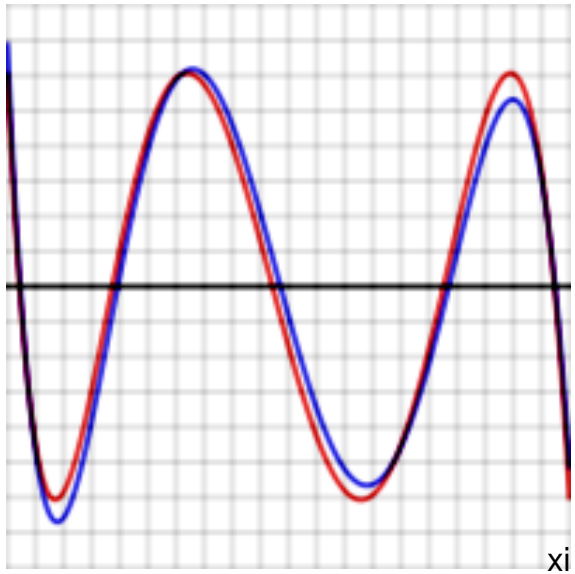
Approximate value function

Idea: The **unknown** value function can be **arbitrarily closely** approximated by a linear combination of bases.

$$V_q(x) \simeq \hat{V}_q(x) = \sum_{i=1}^{n} w_{i,q}\phi_{i,q}(x)$$

e.g. polynomial $\phi_q(x) : 1, x^2, x^3, \ldots$.

unknown value function—**unknown** (finite many) parameters w_i.



Approximate optimal control

$$\min_u \left(\frac{\partial V_q^\star}{\partial x} f(x,u) + \ell(x,u) \right) = 0, \ \forall q \in Q, \ \forall x \in \mathbf{Inv}(q)$$

$$V_q^\star(x) - V_{q'}^\star(x) = s(q,q'), \ q' = \delta(q, L(x)).$$

and boundary condition $V_q(x_f) = 0, \quad \forall q \in F.$

Approximate optimal control

$$\frac{\partial \hat{V}_q}{\partial x} f(x,u) + \ell(x,u) \geq 0, \quad \forall q \in Q, \quad \forall x \in \mathrm{Inv}(q), \quad \forall u \in \mathcal{U},$$

$$\hat{V}_q(x) - \hat{V}_{q'}(x) = s(q,q'), \quad q' = \delta(q, L(x)).$$

and boundary condition

$$\hat{V}_q(x_f) = 0, \forall q \in F.$$

Approximation $\hat{V}$ is a **lower bound** [CDC'16] on the true value function, i.e., $\hat{V}_{q_0}(x_0) \leq V^\star_{q_0}(x_0)$

Approximate optimal control

$$\frac{\partial \hat{V}_q}{\partial x} f(x,u) + \ell(x,u) \geq 0, \quad \forall q \in Q, \quad \forall x \in \mathrm{Inv}(q), \quad \forall u \in \mathcal{U},$$

$$\hat{V}_q(x) - \hat{V}_{q'}(x) = s(q,q'), \quad q' = \delta(q, L(x)).$$

and boundary condition

$$\hat{V}_q(x_f) = 0, \forall q \in F.$$

Approximation $\hat{V}$ is a **lower bound** [CDC'16] on the true value function, i.e., $\hat{V}_{q_0}(x_0) \leq V^\star_{q_0}(x_0)$

Approximate optimal control

$$\max_{w} \hat{V}_{q_0}(x_0)$$

$$\text{subject to: } \frac{\partial \hat{V}_q}{\partial x} f(x,u) + \ell(x,u) \geq 0, \quad \forall q \in Q, \quad \forall x \in \text{Inv}(q),$$

$$q_0 = \delta(q_{init}, L(x_0)),$$

$$\hat{V}_q(x) - \hat{V}_{q'}(x) = s(q,q'), \quad q' = \delta(q, L(x)).$$

$$\hat{V}_q(x_f) = 0, \forall q \in F.$$

Approximate optimal control

Given w (weight parameter), and ϕ_q (vector of basis functions),

$$\max_{w} w_{q_0}^T \phi_{q_0}(x_0)$$

$$\text{subject to: } \frac{\partial w_q^T \phi_q}{\partial x} f(x,u) + \ell(x,u) \geq 0, \quad \forall q \in Q, \quad \forall x \in \text{Inv}(q),$$

$$q_0 = \delta(q_{init}, L(x_0)),$$

$$w_q^T \phi_q(x) - w_{q'}^T \phi_{q'}(x) = s(q,q'), \quad q' = \delta(q, L(x)).$$

$$w_q^T \phi_q(x_f) = 0, \forall q \in F.$$

where $\hat{V}_q(x) = \sum_i w_{i,q} \phi_{i,q}(x)$ as the value function approximation.

Approximate optimal control

- LTI system with **quadratically representable** invariance regions and guards $\Rightarrow$ SDP [CDC'16] and SYDAR tool

```
pip install sydar
```

- Nonlinear system: Semi-infinite program.
- Existing algorithms [R Hettich, 1993] are inefficient with **non-differentiable** objective functions.

Propose: An importance-sampling based search!

Model Reference Adaptive Search

Our method is based on MRAS [Hu, Fu, & Marcus, 2007], a general sampling-based method for global optimization

$$x^\star \in \text{argmax}_{x \in \mathcal{X}} H(x), \quad \mathcal{X} \subseteq \mathbb{R}^n.$$

Model Reference Adaptive Search

Our method is based on MRAS [Hu, Fu, & Marcus, 2007], a general sampling-based method for global optimization

$$x^\star \in \text{argmax}_{x \in \mathcal{X}} H(x), \quad \mathcal{X} \subseteq \mathbb{R}^n.$$

1. Define a sequence of reference distributions $\{g_k(\cdot)\}$ that converges to a "Dirac" around $x^\star$, e.g.,

$$g_k(x) = \frac{H(x) g_{k-1}(x)}{\int_Z H(x') g_{k-1}(x') \nu(dx')}, \quad k = 1, 2, \ldots$$

Model Reference Adaptive Search

Our method is based on MRAS [Hu, Fu, & Marcus, 2007], a general sampling-based method for global optimization

$$x^\star \in \text{argmax}_{x\in\mathcal{X}} H(x), \quad \mathcal{X} \subseteq \mathbb{R}^n.$$

1. Define a sequence of reference distributions $\{g_k(\cdot)\}$ that converges to a "Dirac" around $x^\star$, e.g.,

$$g_k(x) = \frac{H(x)g_{k-1}(x)}{\int_Z H(x')g_{k-1}(x')\nu(dx')}, \quad k = 1, 2, \ldots$$

2. Approximate the exact reference distributions $\{g_k(\cdot)\}$ by a parameterized family of distributions $\{p(\cdot, \theta) \mid \theta \in \Theta\}$.

Model Reference Adaptive Search

Our method is based on MRAS [Hu, Fu, & Marcus, 2007], a general sampling-based method for global optimization

$$x^\star \in \text{argmax}_{x\in\mathcal{X}} H(x), \quad \mathcal{X} \subseteq \mathbb{R}^n.$$

1. Define a sequence of reference distributions $\{g_k(\cdot)\}$ that converges to a "Dirac" around $x^\star$, e.g.,

$$g_k(x) = \frac{H(x)g_{k-1}(x)}{\int_Z H(x')g_{k-1}(x')\nu(dx')}, \quad k = 1, 2, \ldots$$

2. Approximate the exact reference distributions $\{g_k(\cdot)\}$ by a parameterized family of distributions $\{p(\cdot,\theta) \mid \theta \in \Theta\}$.
3. Generate a sequence of parameters $\{\theta_k\}$ by minimizing the KL divergence between $g_k(\cdot)$ and $p(\cdot,\theta)$,

$$D_{KL}(g_k, p(\cdot,\theta)) := \int_Z \ln \frac{g_k(x)}{p(x,\theta)} g_k(x)\nu(dx).$$

A simple/naive formulation

Add a **terminal cost** to penalize value/policy that does not satisfy the temporal logic constraints.

$$\text{minimize}_W J(x_0, u) + \bar{J} \times (1 - 1_F(q_f)).$$

- Given $\hat{V}$, derive u from the HJB—assuming that $\hat{V}$ **is** optimal.

$$u(x, q) = \arg\min_{u \in \mathcal{U}} \{ \frac{\partial \hat{V}_q(x)}{\partial x} \cdot f(x, u) + \ell(x, u) \}$$

$$\hat{V}(x, q) = \min_{q'} \{ \hat{V}(x, q') + s(x, q, q') \},$$

$$\forall x \in G_e, \ \forall e = (q, \sigma, q') \in E, q(t_k^+) = \delta(q(t_k^-), L(x(t_k^+))).$$

- Approximate actor-critic NN fails with local optimality and discontinuity in the value function!

Problem: randomly sampling weights that satisfy all the constraints that are also optimal is a **rare** event.

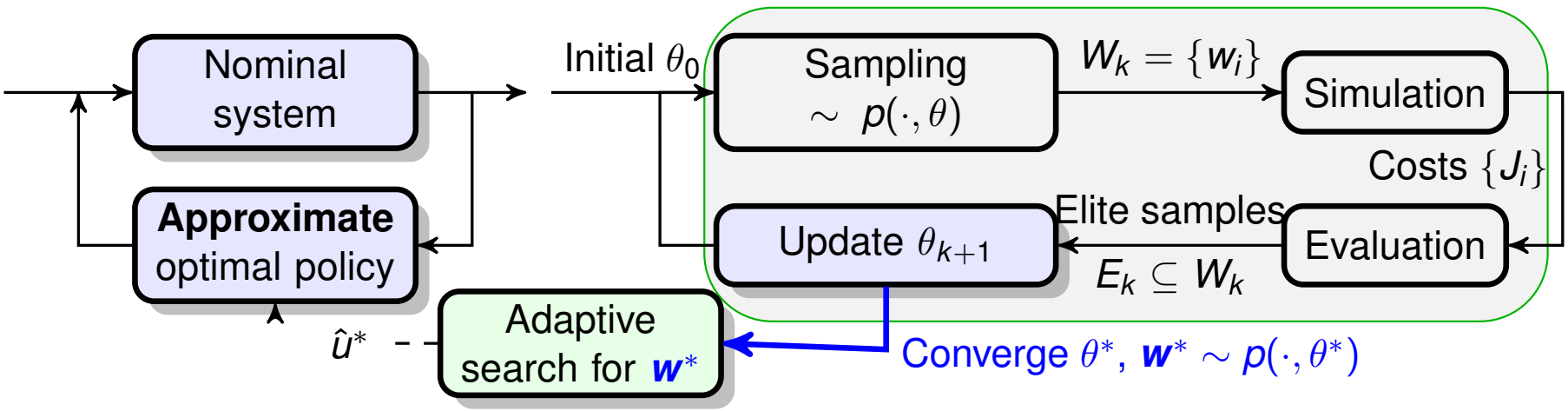


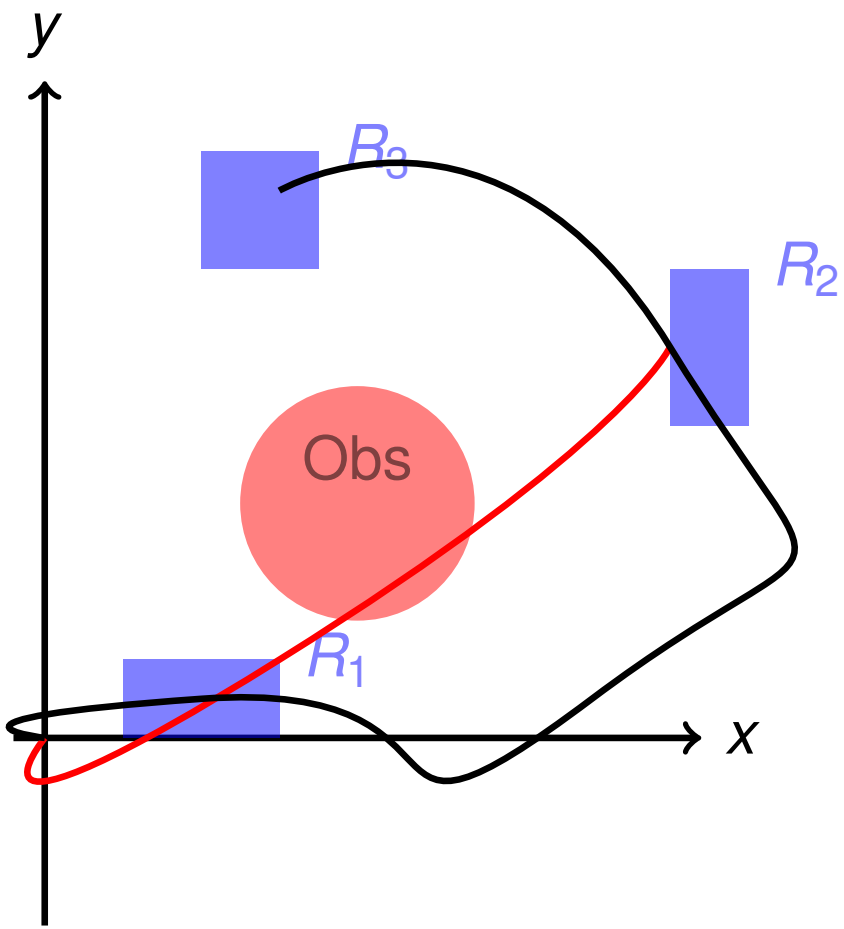
Figure: The overview of the proposed sampling-based optimal control for LTL constraints.

Learning from “bad” weights

- A “bad” weight results in a policy that violates the constraints.
- Difficulty with a good start.

Solution: Learning from the “bad” weights using guided search.

$\Diamond(R_1 \wedge \Diamond(R_2 \wedge \Diamond R_3)) \wedge \Box\neg\text{Obs}$.



- Red: satisfies partial spec.
- Black: satisfies the spec but not optimal.

Both provide information about optimal policy.

Learning from “bad” weights

Idea:

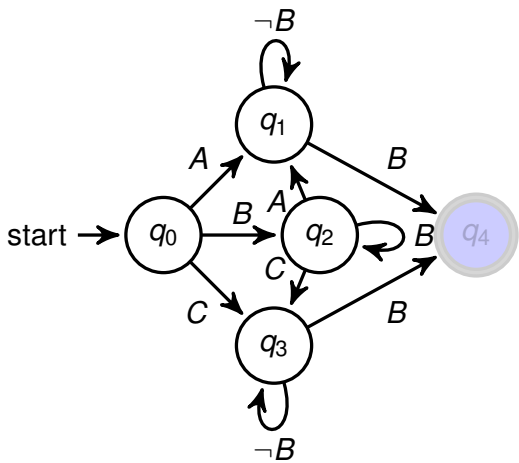
- Instead of throwing away unsatisfying weights,
- ... penalizes with a cost.

Rank-guided weighting of samples.

1. $\text{rank}(q) = 0$ for all final q.
2. $\text{rank}(q) = \min_{q' \in Q}\{\text{rank}(q') + 1 \mid \exists A \in 2^{AP}, \delta(q, A) = q'\}$.

A state-dependent terminal cost:

$$h(q) = \begin{cases} 0 & \text{for all } q \in F. \\ \text{rank}(q) \times c & \text{otherwise.} \end{cases}$$



Learning from “bad” weights

Idea:

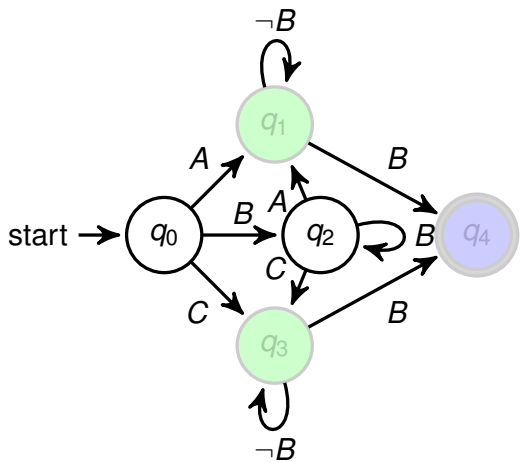
- Instead of throwing away unsatisfying weights,
- ... penalizes with a cost.

Rank-guided weighting of samples.

1. $\text{rank}(q) = 0$ for all final q.
2. $\text{rank}(q) = \min_{q' \in Q}\{\text{rank}(q') + 1 \mid \exists A \in 2^{AP}, \delta(q, A) = q'\}$.

A state-dependent terminal cost:

$$h(q) = \begin{cases} 0 & \text{for all } q \in F. \\ \text{rank}(q) \times c & \text{otherwise.} \end{cases}$$



Learning from “bad” weights

Idea:

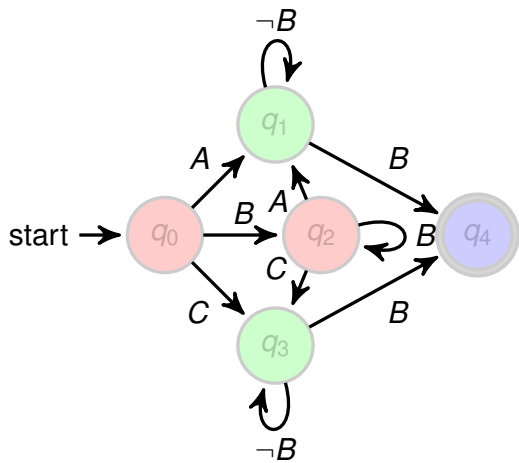
- Instead of throwing away unsatisfying weights,
- ... penalizes with a cost.

Rank-guided weighting of samples.

1. $\text{rank}(q) = 0$ for all final q.
2. $\text{rank}(q) = \min_{q' \in Q}\{\text{rank}(q') + 1 \mid \exists A \in 2^{AP}, \delta(q, A) = q'\}$.

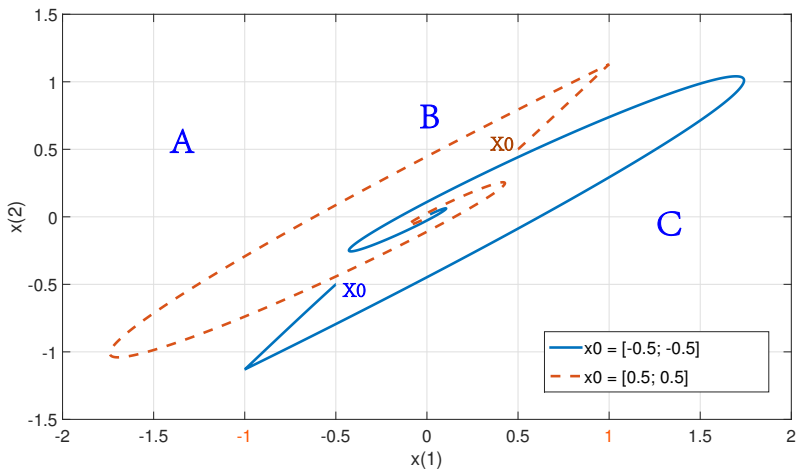
A state-dependent terminal cost:

$$h(q) = \begin{cases} 0 & \text{for all } q \in F. \\ \text{rank}(q) \times c & \text{otherwise.} \end{cases}$$



Example: Linear Quadratic control with LTL

For linear quadratic system, the optimal value function is polynomial.
Red polynomial basis $\phi(x) = [x_1 x_2, x_1^2, x_2^2]$, for each $q \in Q$.
The number of total weight vectors 15.



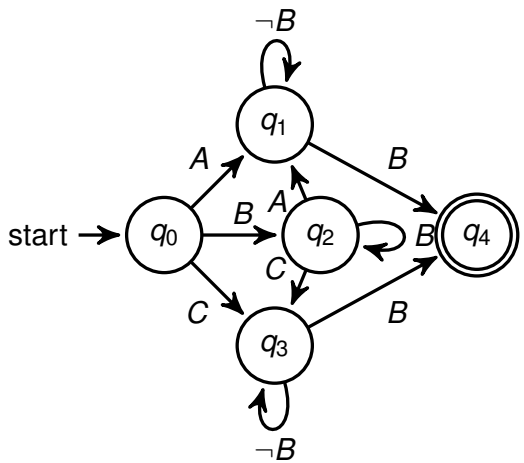
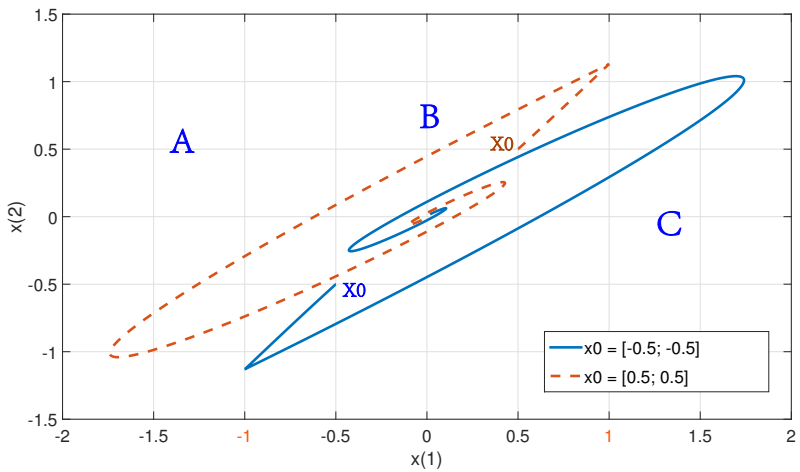
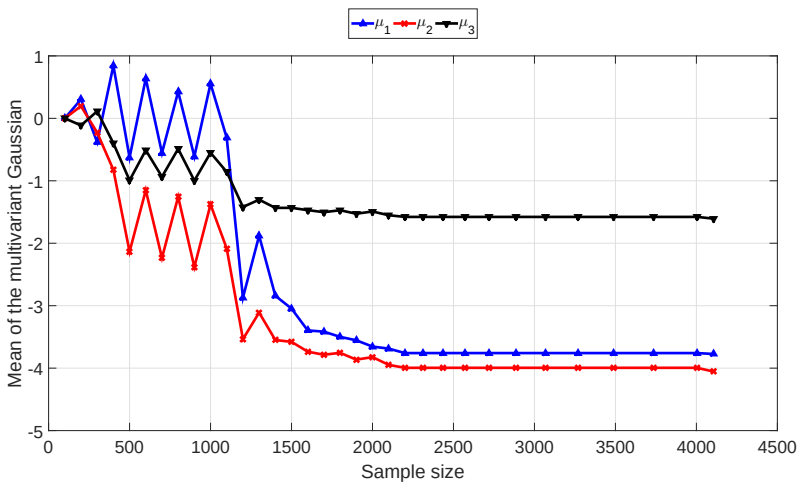


Figure: Automaton $\mathcal{A}_{\varphi_1}$ for $\varphi_1 = (A \rightarrow \Diamond B) \wedge (C \rightarrow \Diamond B) \wedge (\Diamond A \vee \Diamond C)$.

Example: Linear Quadratic control with LTL

The convergence of weight vectors (converge after 10-12 iterations. Each iteration uses 100 samples.)





Example: Dubins car optimal control with LTL

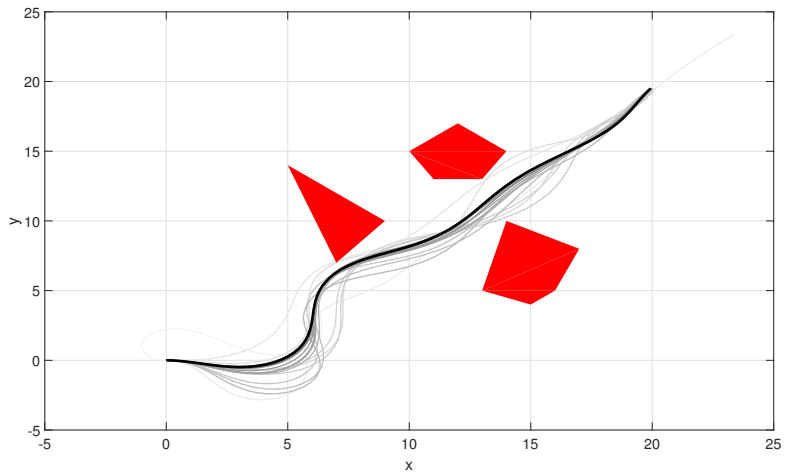
Specification: Visit the target while avoiding the obstacles.
A mixture of bases:

- Localized Radial Gaussian basis:

$$\phi(x) = \exp\left(\frac{-\|x - x_i\|^2}{2\sigma^2}\right)$$

  for pre-selected discrete centers (in x-y coordinate) x_i and σ.

- Trigonometric functions: $\cos(\theta)$, $\sin(\theta)$.



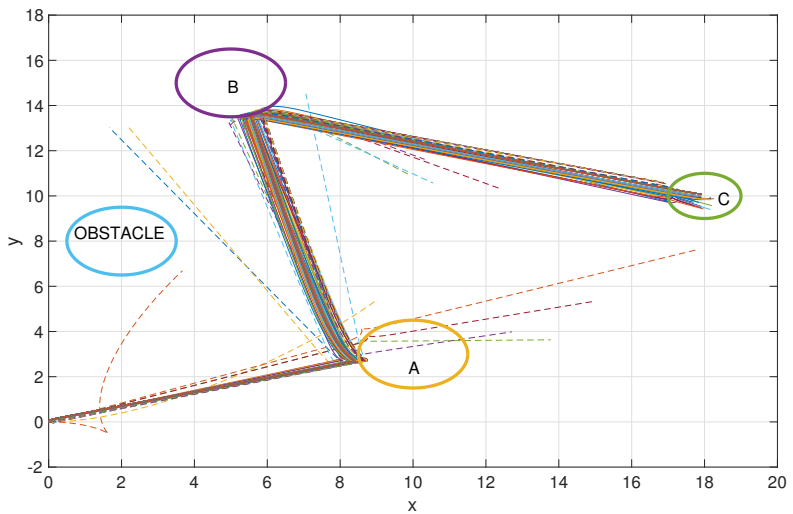
Example: Dubins car optimal control: LTL co-safe spec

Specification: visit regions A, B, C in any order and avoid the obstacles.
A mixture of bases:

- Localized Radial Gaussian basis:

$$\phi(x) = \exp\left(\frac{-\|x - x_i\|^2}{2\sigma^2}\right)$$

  for pre-selected discrete centers (in x-y coordinate) x_i and σ.
- Trigonometric functions: $\cos(\theta)$, $\sin(\theta)$.



Conclusion and future work

- We proposed an importance sampling based approximate optimal control algorithm under temporal logic constraints.
- Introducing rank-guided policy search — similar to reward shaping — to enable learning from bad samples.

Current and future work:

- Feature selection: How to select a sparse set of basis function.
- Good starting point matters: Value function relates to control Lyapunov function for switched linear systems.