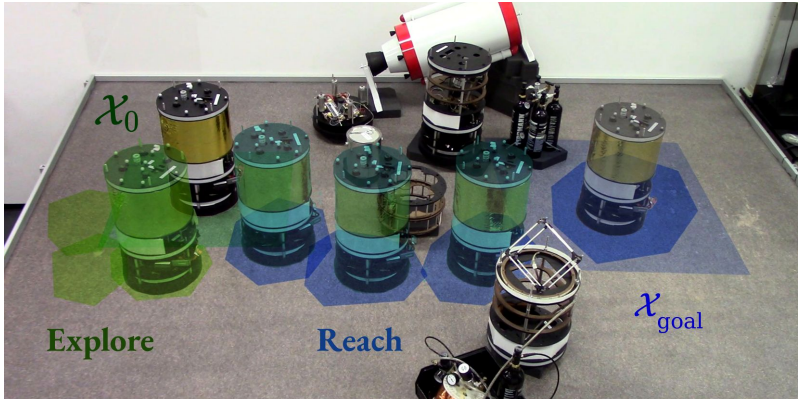
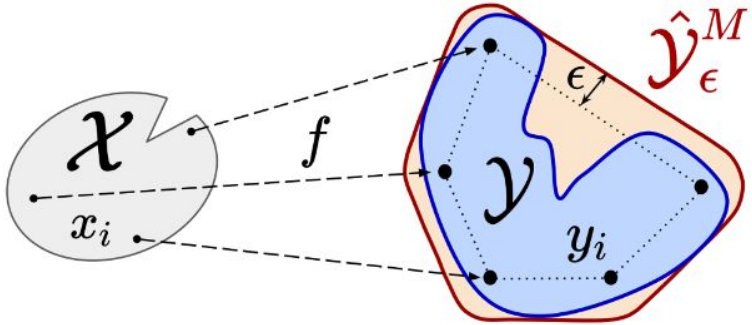


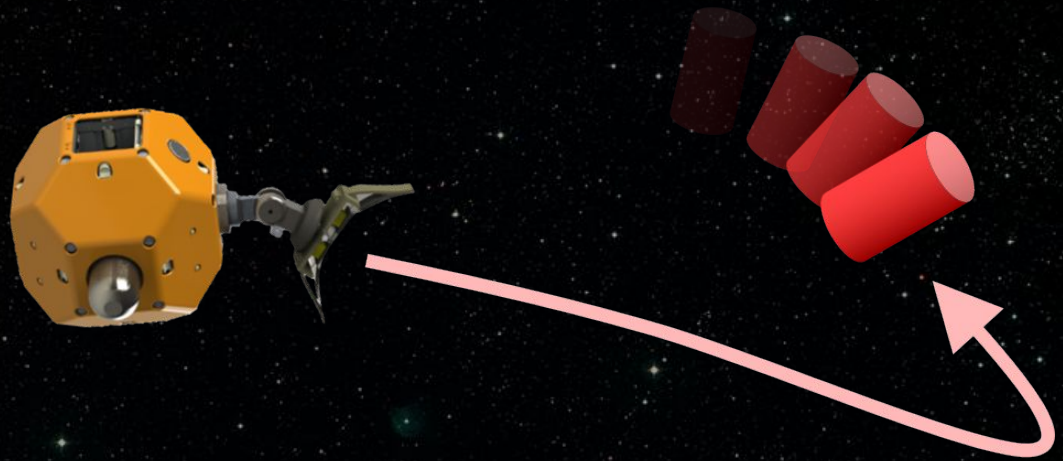


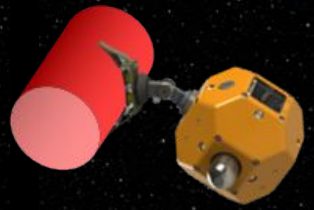
Uncertainty-aware control strategies for safe learning-based robotic autonomy

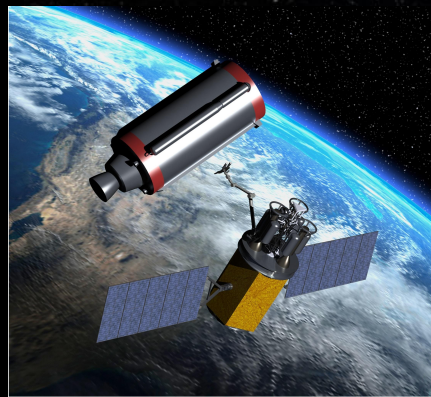
Thomas Lew

Autonomous Systems Lab

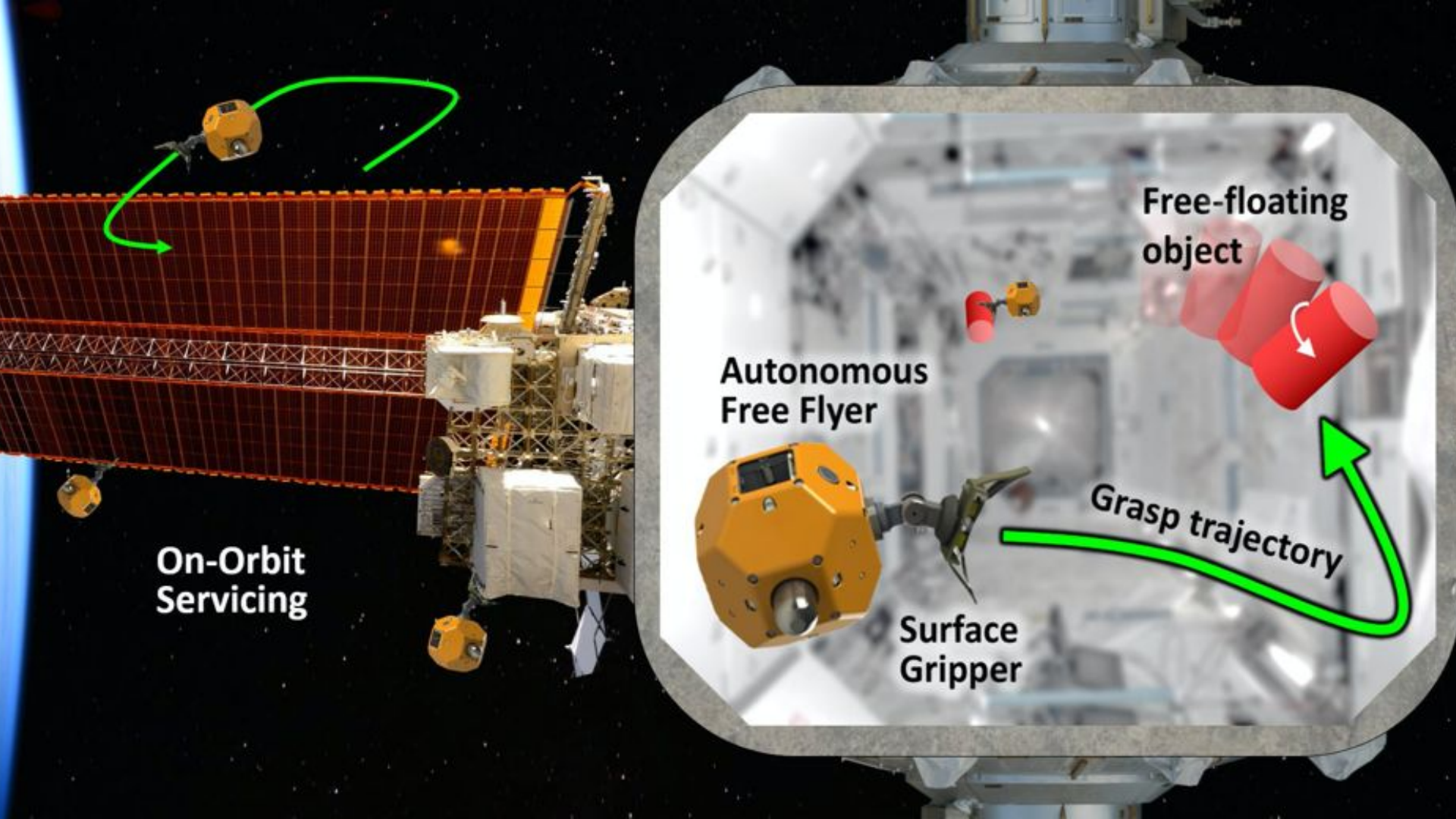
Robotic autonomy in space



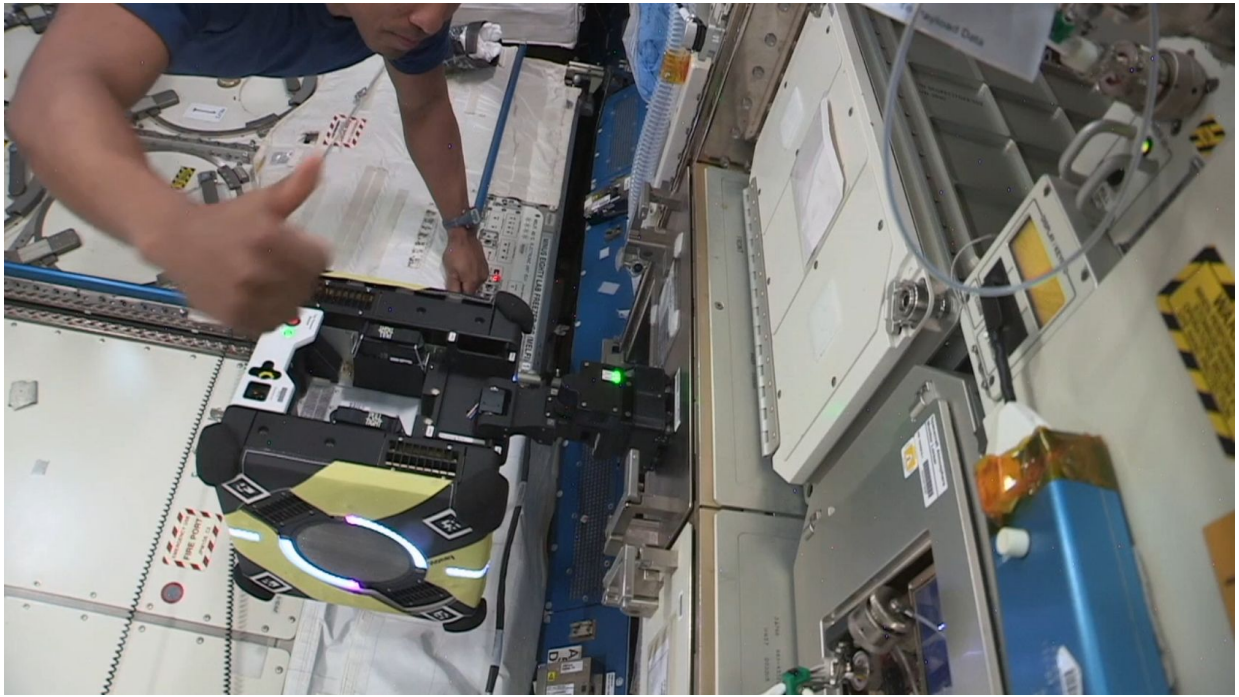


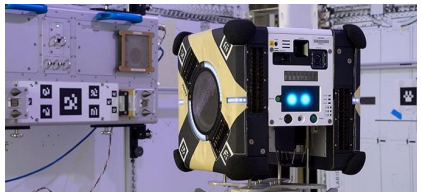


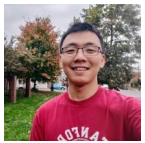
ASL


On-Orbit
Servicing
Free-floating
object
Autonomous
Free Flyer
Grasp trajectory
Surface
Gripper

NASA Astrobee free-flying robots on-board the ISS







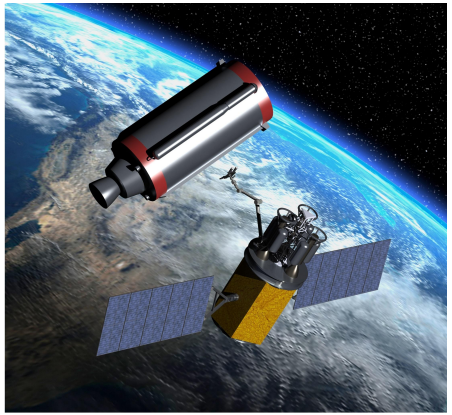
Biomimetics & Dexterous Manipulation Laboratory

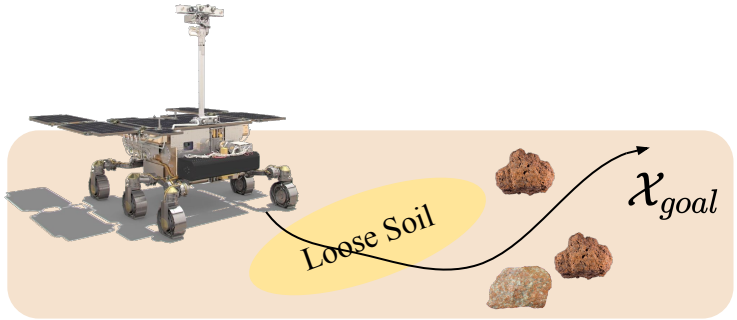
ASU

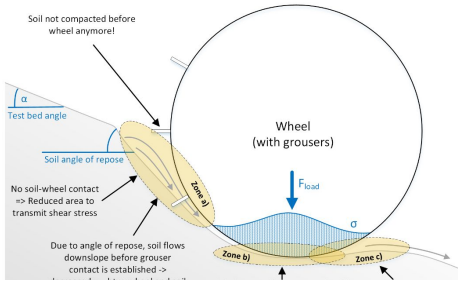
Astronaut McClain with Astrobee.

Robotic autonomy: challenges

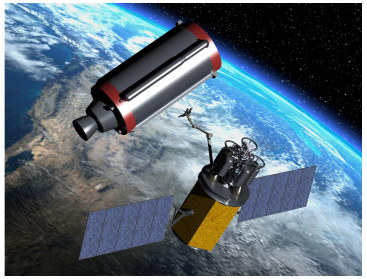
- Uncertain dynamics (unknown payload properties and flexible grasp)
- Safe use of learning-based components?
- Accounting for uncertainty?

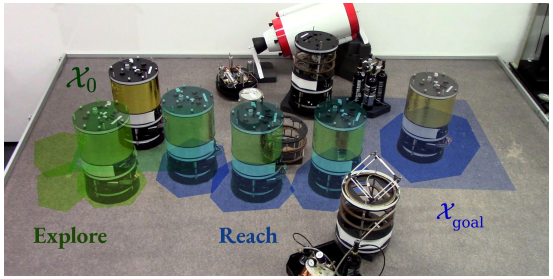


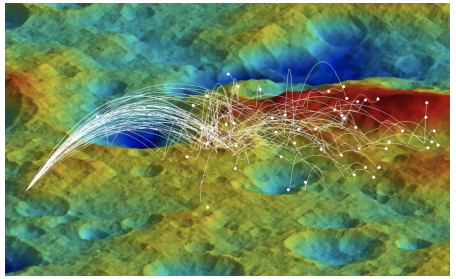




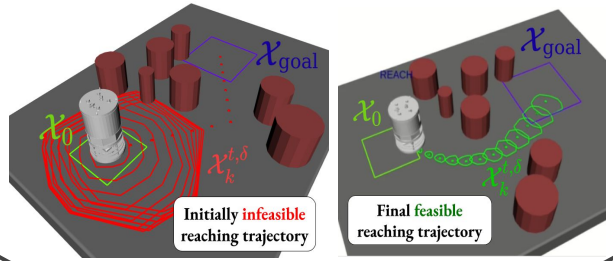
Towards safe learning-based robotic autonomy



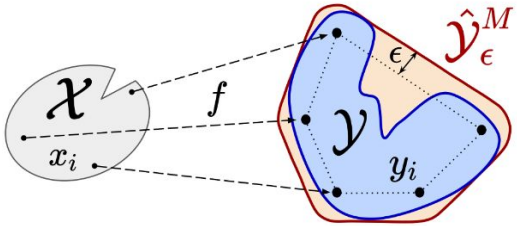




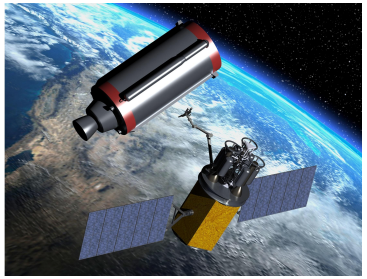
1 safe active dynamics learning and control

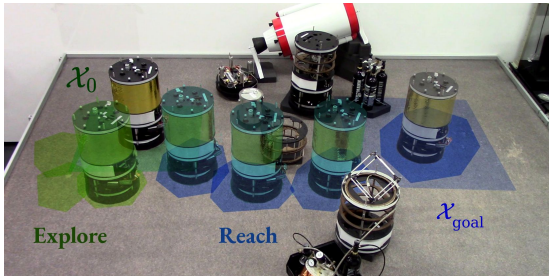


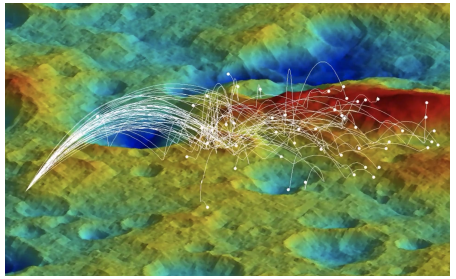
2 sampling-based reachability analysis



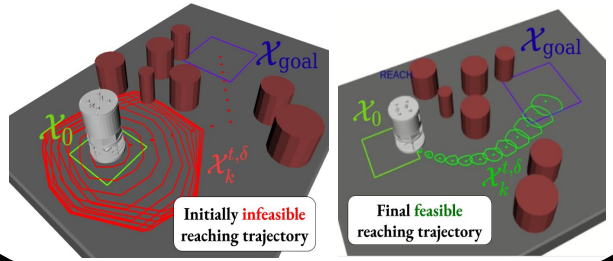
Towards safe learning-based robotic autonomy



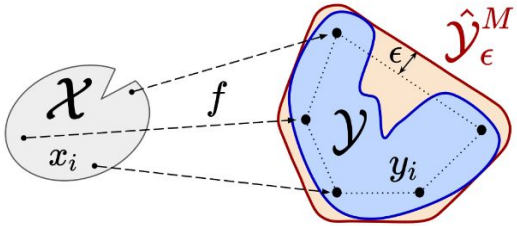




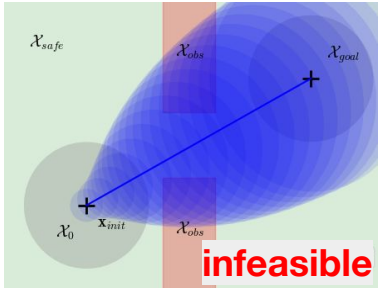
① safe active dynamics learning and control

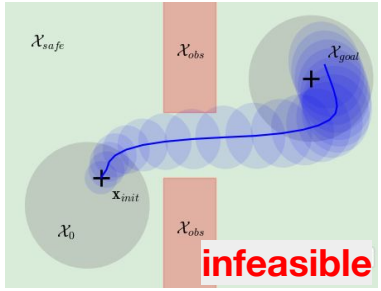


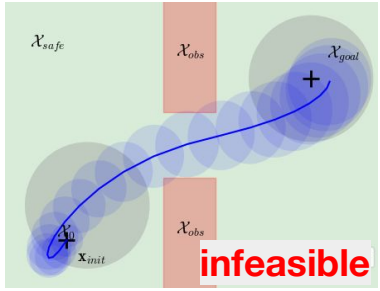
② sampling-based reachability analysis



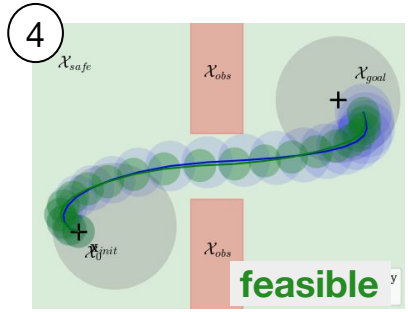
Uncertainty may be too high and cause infeasibility



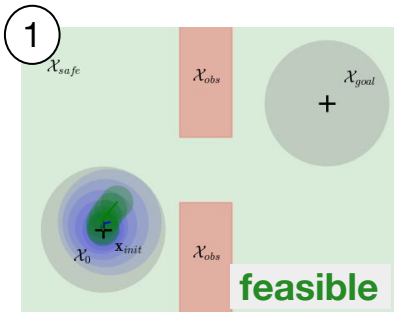


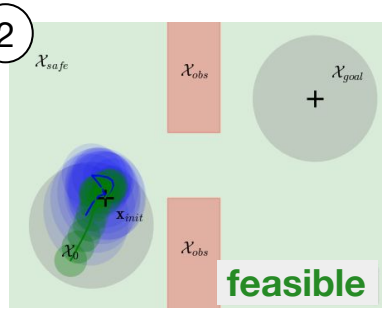


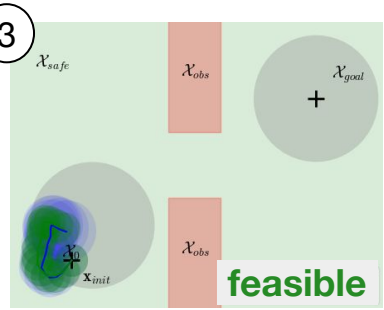
...



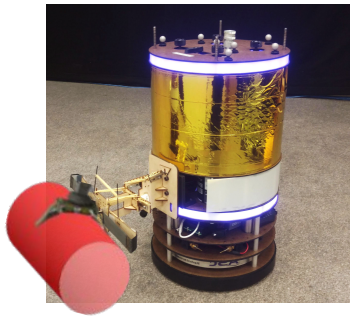
=> first, gather data to learn and reduce uncertainty







free-flyer space robot
with uncertain payload



Lew, Sharma, Harrison, Bylard, Pavone. Safe Active Dynamics Learning and Control: A Sequential Exploration-Exploitation Framework. IEEE T-RO, 2022

Problem formulation

- Uncertain dynamics

$$x_{t+1} = h(x_t, u_t) + g(x_t, u_t, \xi) + \epsilon_t$$

- Constraints $(x, u) \in \mathcal{X} \times \mathcal{U}$

x State
u Control
ξ Parameters
ϵ Disturbances (bounded)

$$\min_{x,u} \quad \mathbb{E}\left[\sum_{t=0}^{T-1} \ell(x_t, u_t)\right] \qquad \text{s.t.} \quad x_0 = x(0)$$

$$x_{t+1} = \text{dynamics}(x_t, u_t), \quad t = 0, \ldots, T-1,$$

$$\mathbb{P}\left(\bigwedge_{t=1}^{T} x_t \in \mathcal{X} \wedge \bigwedge_{t=0}^{T-1} u_t \in \mathcal{U} \wedge x_T \in \mathcal{X}_{\text{goal}}\right) \geq 1 - \delta.$$

Problem formulation

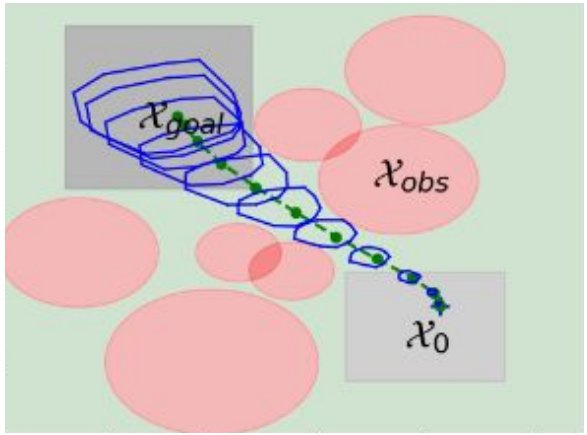
$$\min_{x,u} \quad \mathbb{E}\left[\sum_{t=0}^{T-1} \ell(x_t, u_t)\right] \qquad \text{s.t.} \quad x_0 = x(0)$$

$$x_{t+1} = h(x_t, u_t) + g(x_t, u_t, \xi) + \epsilon_t, \quad t = 0, \ldots, T-1,$$

$$\mathbb{P}\left(\bigwedge_{t=1}^{T} x_t \in \mathcal{X} \wedge \bigwedge_{t=0}^{T-1} u_t \in \mathcal{U} \wedge x_T \in \mathcal{X}_{\text{goal}}\right) \geq 1 - \delta.$$

Additional knowledge:

- Control-invariant set $\mathcal{X}_0$
- Prior data $\mathcal{D}_t = \{(x_k^i, u_k^i, x_{k+1}^i)\}_{k=0\ldots t}^{i=0\ldots B}$



Related work

- Approaches to learn a model / controller to improve performance
  - Reinforcement learning (e.g., Hwangbo et al, 2019)
  - Learning-based MPC (e.g, Deisenroth et al, 2015. Koller et al, 2018)
  - Learning-based control barrier functions (e.g., Khojasteh et al, 2020)
- Dual control typically does not consider constraints (e.g., Mania et al, 2020)
- We enforce a **joint** chance constraint

$$\mathbb{P}\left((x_t \in \mathcal{X} \wedge u_t \in \mathcal{U}\ \forall t),\ (x_T \in \mathcal{X}_{\text{goal}})\right) \geq (1-\delta)$$

which gives stronger guarantees than *pointwise* chance constraints

$$\mathbb{P}(x_t \in \mathcal{X}) \geq 1-\delta \quad \text{for all} \quad t \geq 0$$

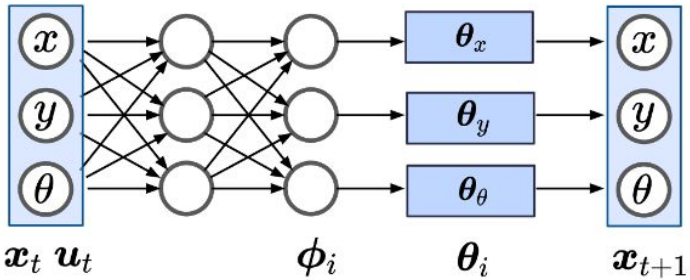
Modeling: linear meta-learning

- Uncertain dynamics

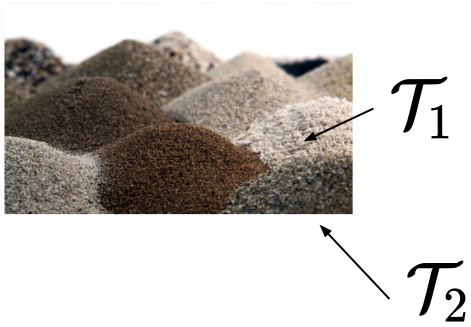
$$x_{t+1} = h(x_t, u_t) + g(x_t, u_t, \xi) + \epsilon_t$$

- Approximate model

$$\hat{g}_i(x_t, u_t, \theta_i) = \theta_i^\top \phi_i(x_t, u_t)$$



train a model capable of *rapid adaptation*



Harrison, Sharma, Pavone. “Meta-Learning Priors for Efficient Online Bayesian Regression,” WAFR, 2018

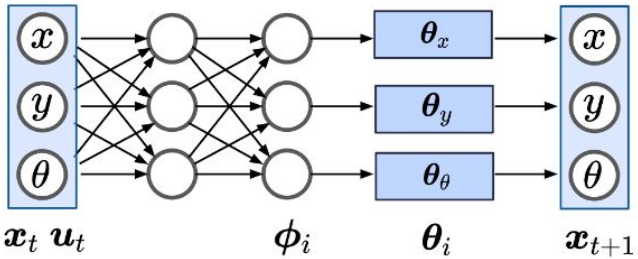
Modeling: linear meta-learning

- Uncertain dynamics

$$x_{t+1} = h(x_t, u_t) + g(x_t, u_t, \xi) + \epsilon_t$$

- Approximate model

$$\hat{g}_i(x_t, u_t, \theta_i) = \theta_i^\top \phi_i(x_t, u_t)$$



- Train a prior $(\bar{\theta}_{i,0}, \Lambda_{i,0})$
- Observe trajectory data

$$(x_0, u_0, x_1, \ldots, x_t, u_t, x_{t+1})$$

- Update estimates $(\bar{\theta}_{i,t}, \Lambda_{i,t})$:

$$\Lambda_{i,t} = \Phi_{i,t-1}^\top \Phi_{i,t-1} + \Lambda_{i,0}$$

$$\bar{\theta}_{i,t} = \Lambda_{i,t}^{-1}(\Phi_{i,t-1}^\top G_{i,t} + \Lambda_{i,0}\bar{\theta}_{i,0})$$

Harrison, Sharma, Pavone. “Meta-Learning Priors for Efficient Online Bayesian Regression,” WAFR, 2018

Self-normalizing martingales

- Uncertain dynamics $x_{t+1} = h(x_t, u_t) + g(x_t, u_t, \xi) + \epsilon_t$
- Approximate model $\hat{g}_i(x_t, u_t, \theta_i) = \theta_i^\top \phi_i(x_t, u_t)$
- Define the confidence sets

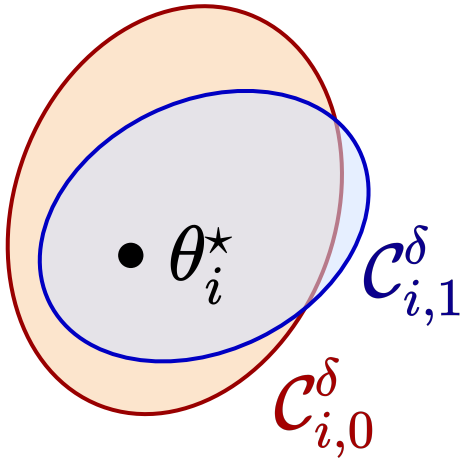
$$\mathcal{C}_{i,t}^{\delta} = \{\theta_i \in \mathbb{R}^d : \|\theta_i - \bar{\theta}_{i,t}\|_{\Lambda_{i,t}} \leq \beta_{i,t}^{\delta}\} \quad \text{with } \beta_{i,t}^{\delta} = \sigma_i \left(\sqrt{2 \log\left(\frac{1}{\delta_i} \frac{\det(\Lambda_{i,t})^{1/2}}{\det(\Lambda_{i,0})^{1/2}} \right)} + \sqrt{\frac{\bar{\lambda}(\Lambda_{i,0})}{\underline{\lambda}(\Lambda_{i,t})}} B \right)$$

Theorem*: If the ϵ_t are σ_i-subGaussian and

$$g \in \mathcal{H}_{\phi_i} = \left\{ \hat{g}_{i,\theta_i} = \theta_i^\top \phi_i \ : \ \|\theta_i - \bar{\theta}_{i,0}\|_{\Lambda_{i,0}}^2 \leq \sigma_i^2 B \right\}.$$

then

$$\mathbb{P}\left(\theta_i^\star \in \mathcal{C}_{i,t}^{\delta} \ \forall t \geq 0 \right) \geq 1 - 2\delta_i$$

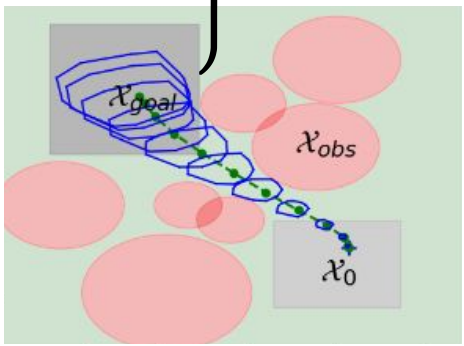


*Adapted from Abbasi-Yadkori, Pal, Szepesvari, "*Improved algorithms for linear stochastic bandits*", NeurIPS 2011

Robust planning via reachability analysis

- Define the reachable sets

$$\mathcal{X}_k^{t,\delta}(u) = \left\{ \begin{array}{c} x_k = f(\cdot, u_{k-1}, \theta, \epsilon_{k-1}) \circ \cdots \circ f(x_0, u_0, \theta, \epsilon_0) \\ \text{with} \quad x_0 = x(t),\ \theta_i \in \mathcal{C}_{i,t}^{\delta},\ \epsilon_{i,s} \in \mathcal{E}_i\ , \\ s=1,\ldots,k-1,\ i=1,\ldots,n \end{array} \right\}$$



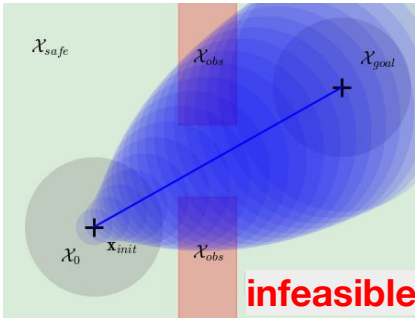
- Define **Reach-OCP(N)**:

$$\min_{\mu,u} \sum_{k=0}^{N-1} \ell(\mu_k, u_k) \text{ s.t. } \begin{array}{l} \bigwedge_{k=1}^{N} \mathcal{X}_k^{t,\delta}(u) \subseteq \mathcal{X}_{safe},\ \mathcal{X}_N^{t,\delta}(u) \subseteq \mathcal{X}_{goal}, \\ \bigwedge_{k=0}^{N-1} u_k \in \mathcal{U},\ \mathcal{X}_0^{t,\delta} = \{x(t)\}, \end{array}$$

If uncertainty is too high: robust exploration

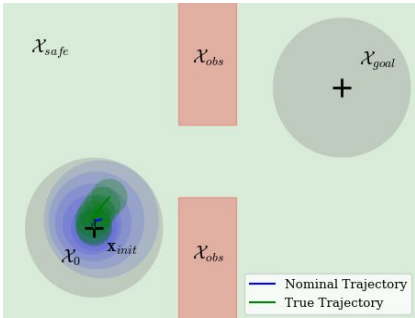
- **Reach-OCP(N)**:

$$\min_{\mu,u} \sum_{k=0}^{N-1} \ell(\mu_k, u_k) \text{ s.t. } \bigwedge_{k=1}^{N} \mathcal{X}_k^{t,\delta}(u) \subseteq \mathcal{X}_{safe},\ \mathcal{X}_N^{t,\delta}(u) \subseteq \mathcal{X}_{goal},\ \bigwedge_{k=0}^{N-1} u_k \in \mathcal{U},\ \mathcal{X}_0^{t,\delta} = \{x(t)\},$$



- **Explore-OCP(N)**:

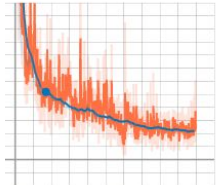
$$\min_{\mu,u} \sum_{k=0}^{N-1} (\ell + \ell_{info})(\mu_k, u_k) \text{ s.t. } \bigwedge_{k=1}^{N} \mathcal{X}_k^{t,\delta}(u) \subseteq \mathcal{X}_{safe},\ \mathcal{X}_N^{t,\delta}(u) \subseteq \mathcal{X}_0,\ \bigwedge_{k=0}^{N-1} u_k \in \mathcal{U},\ \mathcal{X}_0^{t,\delta} = \{x(t)\},$$



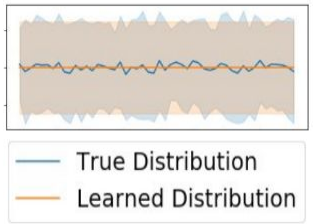
Offline: meta-train

1. Collect data
2. Train model

$$\theta_i^\top \phi_i(x, u)$$



3. Verify accuracy

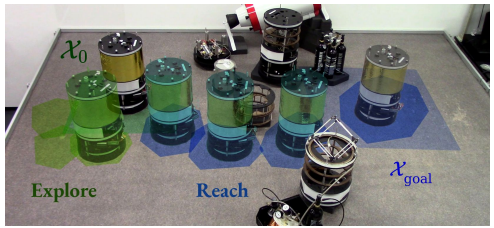


$$\theta_i^\top \phi_i(x, u)$$

Online: exploration - exploitation

1. If **ReachOCP** is **feasible**:
   Reach $\mathcal{X}_{\text{goal}}$
2. Else, for different horizons N:
   Solve **ExploreOCP(N)** to $\mathcal{X}_0$
   Apply controls & gather data
   Adapt parameters θ_i

while $x \notin \mathcal{X}_{\text{goal}}$



This approach is a feasible solution to the original problem

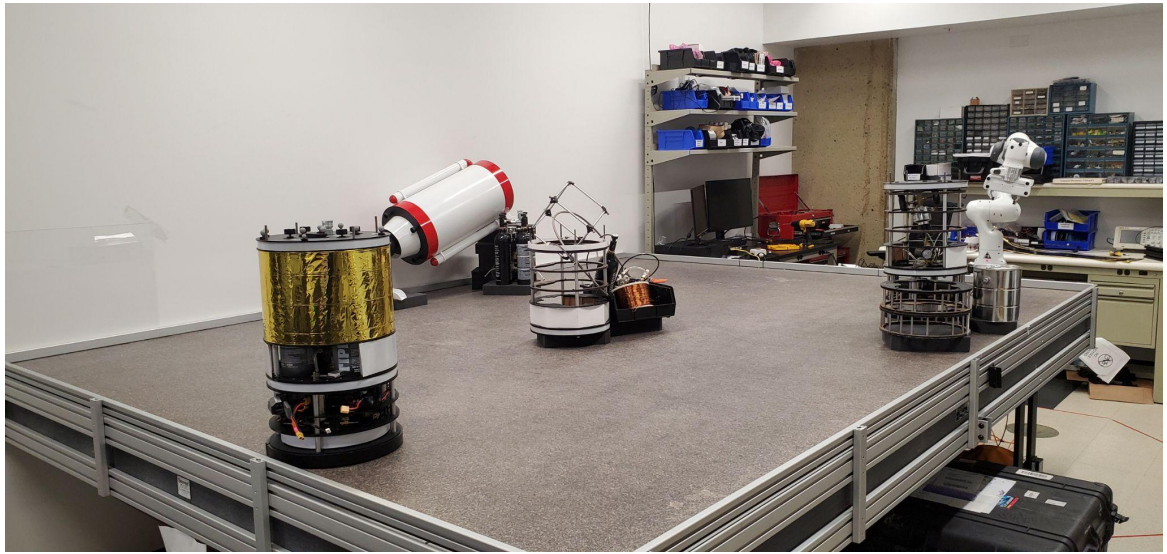
Theorem: *assume that the* ϵ_t *are* σ_i*-subGaussian and bounded and that* $g \in \mathcal{H}_{\phi_i} = \left\{ \hat{g}_{i,\theta_i} = \theta_i^\top \phi_i \ : \ \|\theta_i - \bar{\theta}_{i,0}\|^2_{\Lambda_{i,0}} \leq \sigma_i^2 B \right\}$.

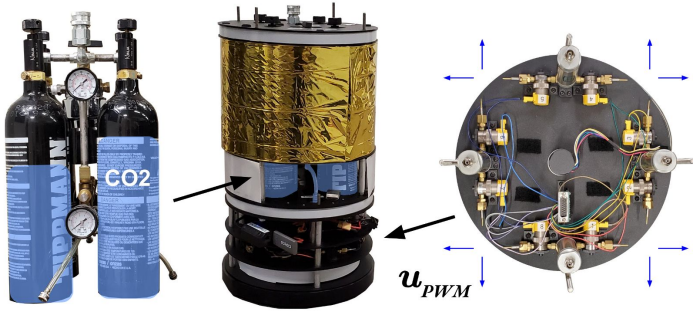
Then, with probability at least $(1-\delta)$ *:*

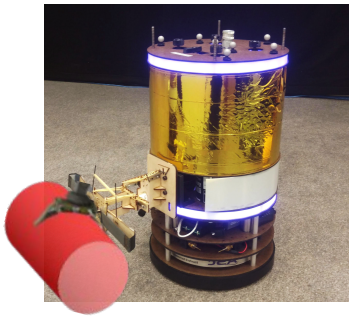
- *each* ***Explore-OCP*** *is feasible*
- *all constraints are always satisfied*
- *if* ***Reach-OCP*** *is eventually feasible, then*

$$\mathbb{P}\left((x_t \in \mathcal{X} \wedge u_t \in \mathcal{U} \ \forall t), \ (x_T \in \mathcal{X}_{\text{goal}}) \right) \geq (1-\delta)$$

Results on a free-flyer robotic testbed



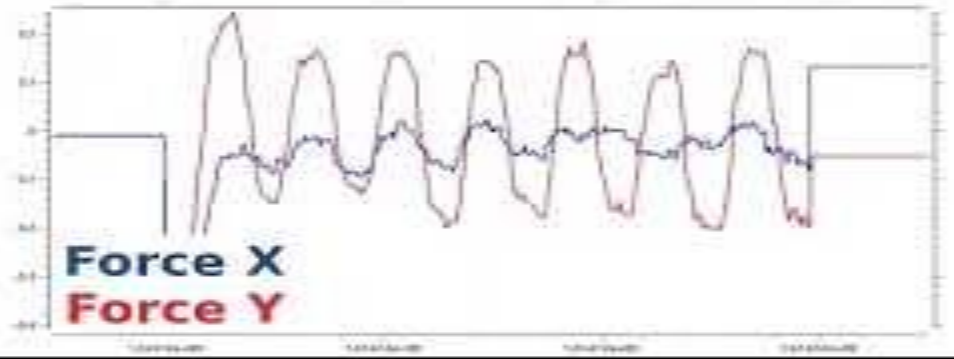


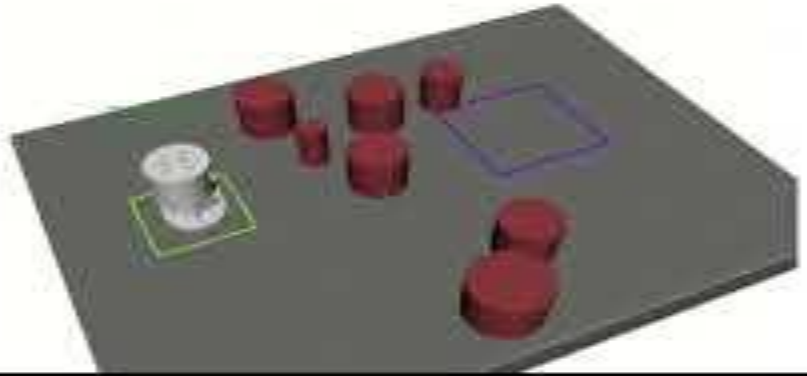


uncertainty:

- CoM offset
- Table tilt (constant force)

Explore & learn inertial properties



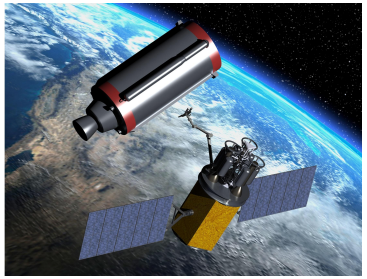


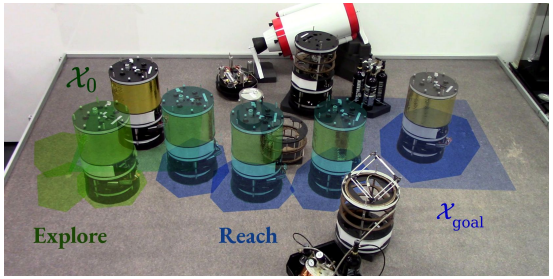
Conclusion (1st part):

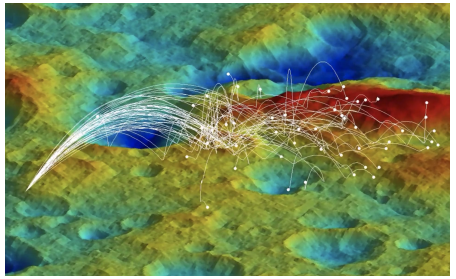
- Need for active learning to safely perform tasks under uncertainty
- Exploit linear model structure to quantify uncertainty
- Open questions:
  - Time to solve the problem? Regret bounds?
  - Efficient computation of the reachable sets accounting for parametric uncertainty?

$$\mathcal{X}_k^{t,\delta}(u) = \left\{ \begin{array}{c} x_k = f(\cdot, u_{k-1}, \theta, \epsilon_{k-1}) \circ \cdots \circ f(x_0, u_0, \theta, \epsilon_0) \\ \text{with} \quad x_0 = x(t),\ \theta_i \in \mathcal{C}_{i,t}^{\delta},\ \epsilon_{i,s} \in \mathcal{E}_i\ , \\ s=1,\ldots,k-1,\ i=1,\ldots,n \end{array} \right\}$$

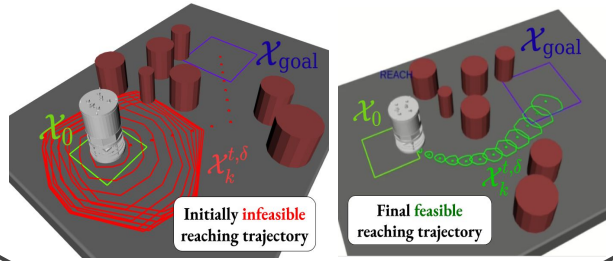
Towards safe learning-based robotic autonomy



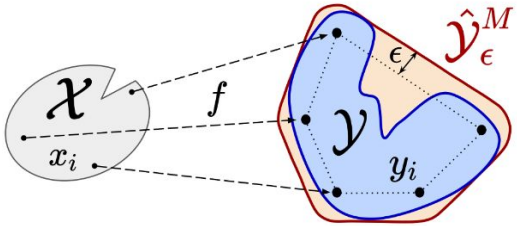




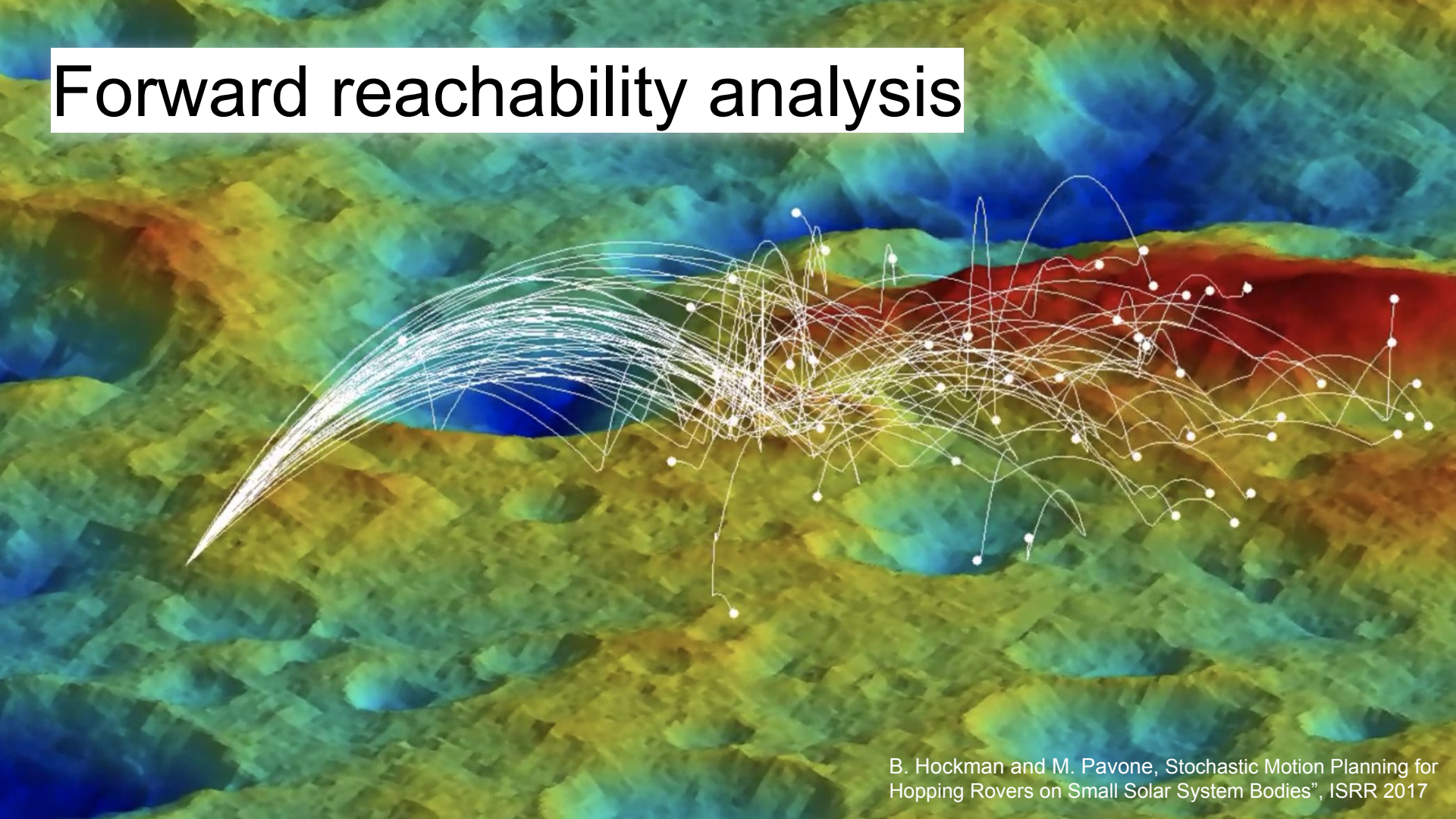
1 safe active dynamics learning and control



2 **sampling-based reachability analysis**

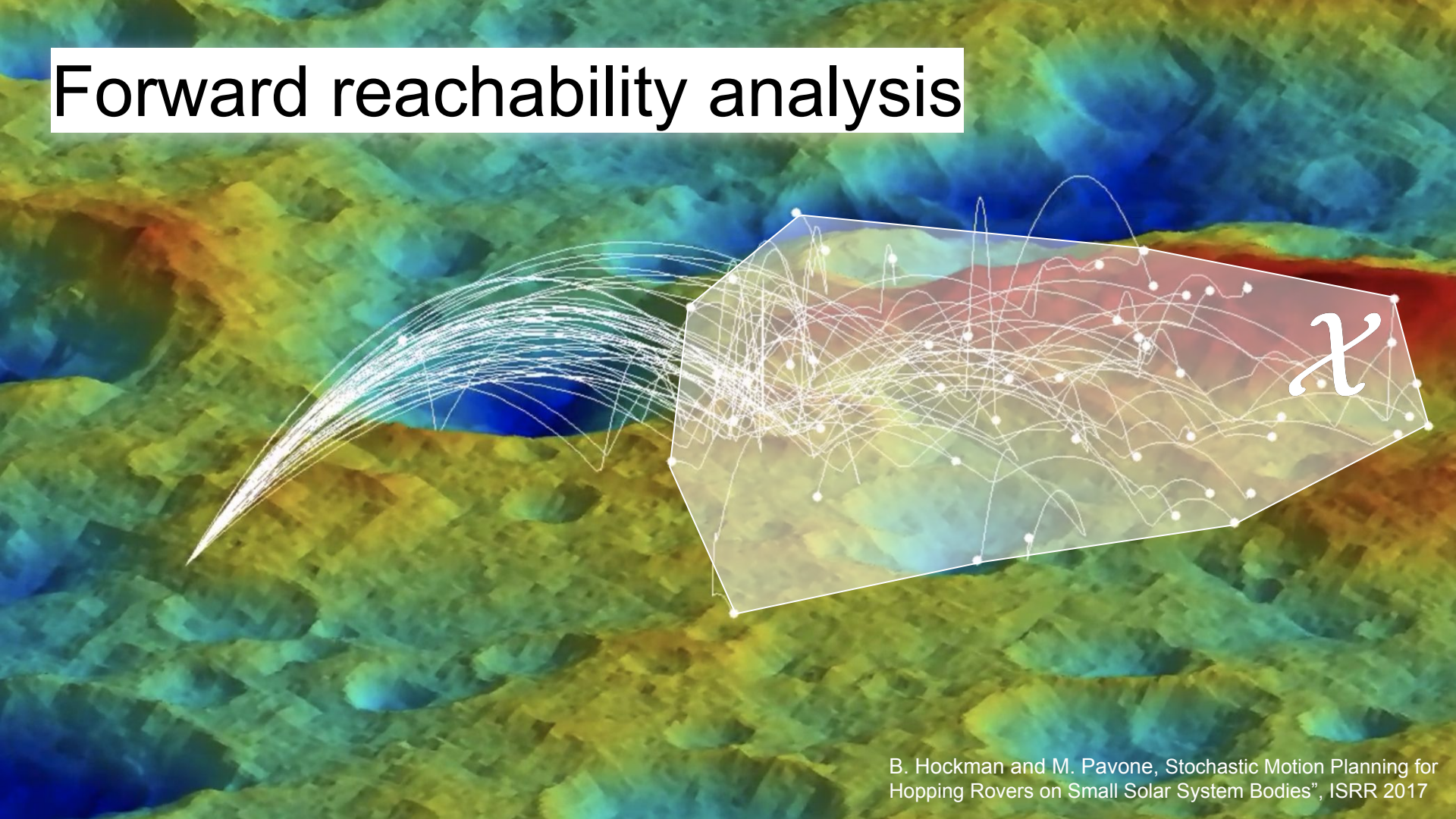


Forward reachability analysis



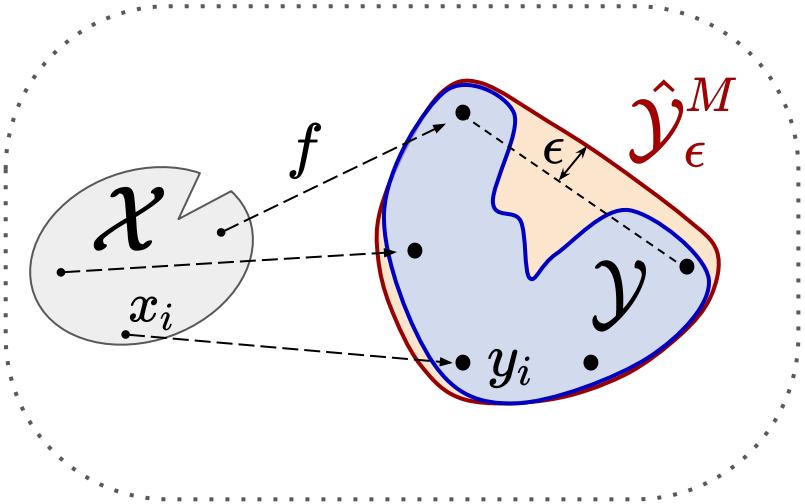
B. Hockman and M. Pavone, Stochastic Motion Planning for Hopping Rovers on Small Solar System Bodies", ISRR 2017

Forward reachability analysis

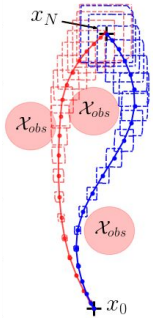


B. Hockman and M. Pavone, Stochastic Motion Planning for Hopping Rovers on Small Solar System Bodies", ISRR 2017

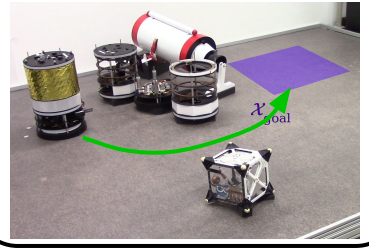
Applications of reachability analysis



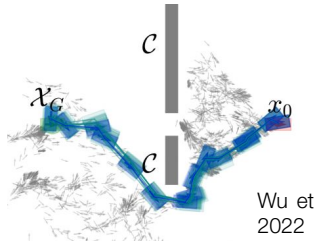
Robust Optimal Control (SCP)



Robust MPC

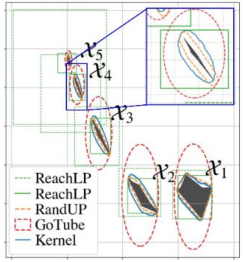


Robust RRT



Wu et al, 2022

NN Verification



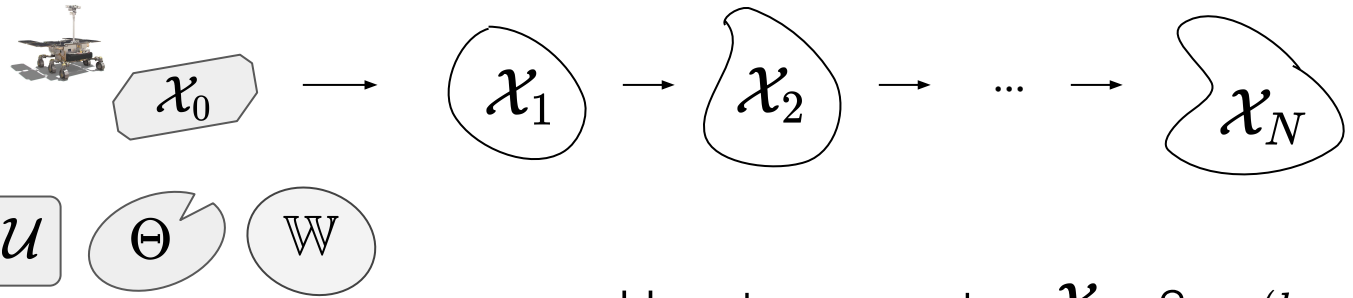
Lew, Janson, Bonalli, Pavone, *A Simple and Efficient Sampling-based Algorithm for General Reachability Analysis*, L4DC, 2022

Forward reachability analysis

$$x_{k+1} = f(x_k,\, u_k,\, \theta,\, w_k)$$

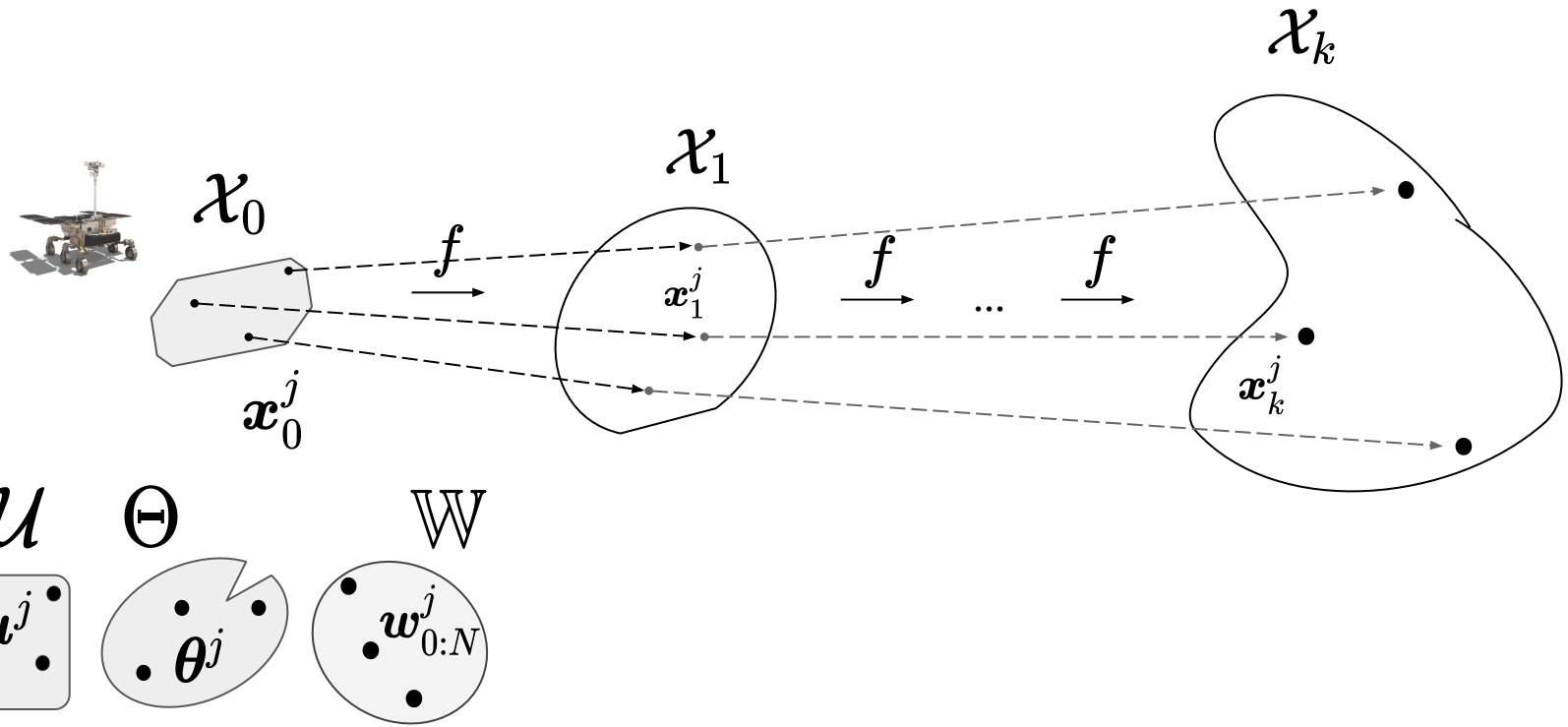
with $(x_0, u, \theta, w) \in \mathcal{X}_0 \times \mathcal{U} \times \Theta \times \mathcal{W}$ compact

x State
u Control
θ Parameters
w Disturbances

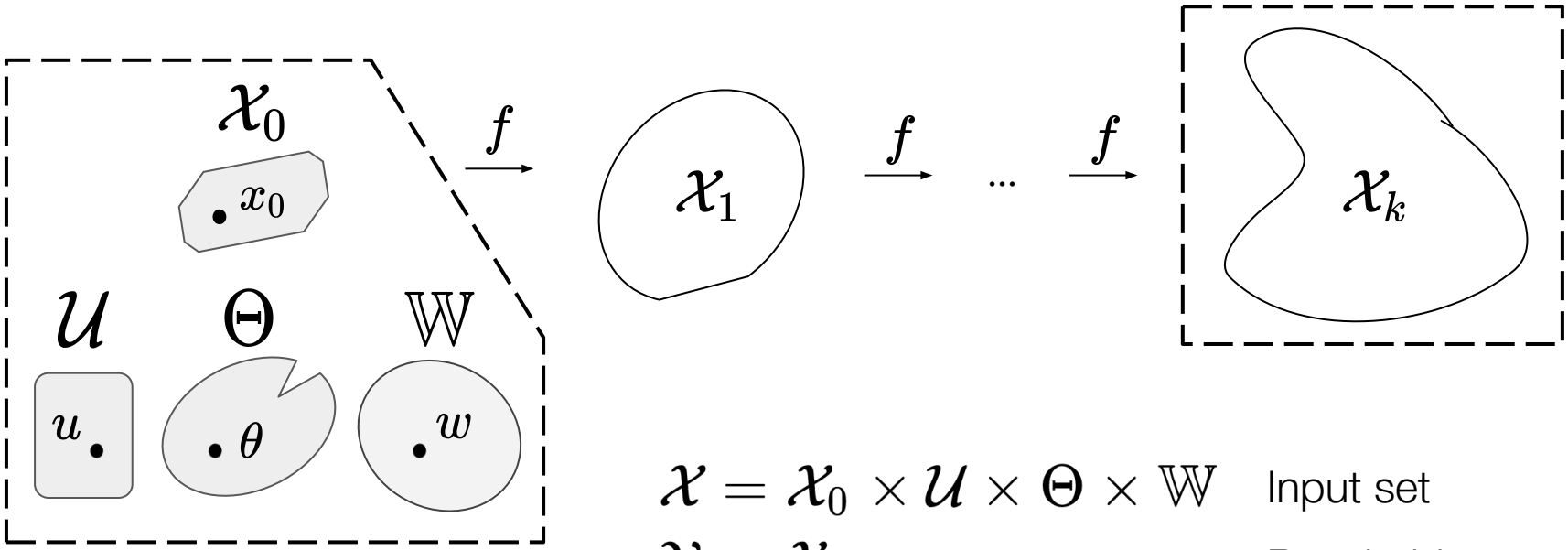


How to compute $\mathcal{X}_k$? $(k = 1, \ldots N)$

Sampling-based forward reachability analysis



General formulation of reachability analysis



$$\mathcal{X} = \mathcal{X}_0 \times \mathcal{U} \times \Theta \times \mathbb{W}$$ Input set

$$\mathcal{Y} = \mathcal{X}_k$$ Reachable set

Reachability map:

$$f : \mathcal{X} \to \mathcal{Y}$$

$$(x_0, u, \theta, w) \mapsto f(\cdot, u_{k-1}, \theta, w_{k-1}) \circ \cdots \circ f(x_0, u_{k-1}, \theta, w_{k-1})$$

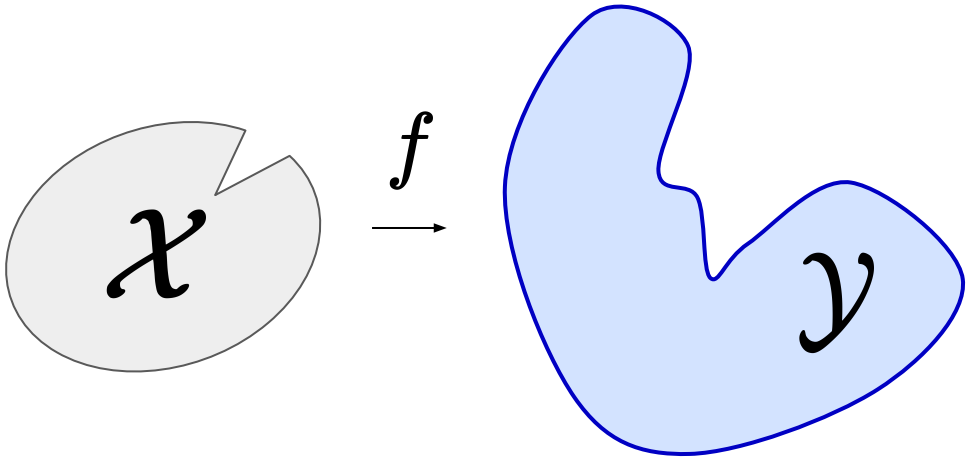
General formulation of reachability analysis

$\mathcal{X} \subset \mathbb{R}^p$ compact

$f : \mathbb{R}^p \to \mathbb{R}^n$ continuous

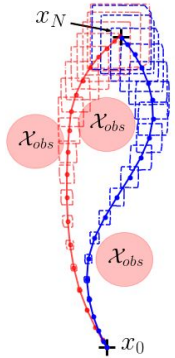
reachable set to estimate

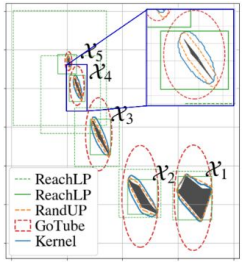
$$\mathcal{Y} = f(\mathcal{X})$$



Related work

- Applications: robust trajectory optimization, robust MPC, …
  - *Convex over-approximations are often sufficient*
- Polytopic & ellipsoidal reachable sets for systems with disturbances
  - Problem-specific: linear/polynomial dynamics (Girard, 2005), (Chen et al, 2013), (Althoff et al, 2021)
  - Nonlinear: techniques that use monotonicity or bound linearization error via Lipschitz constants (Koller et al., 2018)
  - Difficult to account for parametric uncertainty: time correlations along state trajectories
- Neural network verification community
  - (Chen et al, 2013), (Weimer et al, 2019), (Fazlyab et al, 2019), (Tran et al, 2019), (Ivanov, 2019), (Everett et al, 2021) developed libraries for NN verification (Flow*, Verisig, LipSDP)
- How can we compute accurate approximations in milliseconds?
  - Sampling-based approaches (Devonport & Arcak, 2020, Thorpe et al, 2021)





Sampling-based reachability analysis

$\mathcal{X} \subset \mathbb{R}^p$ compact

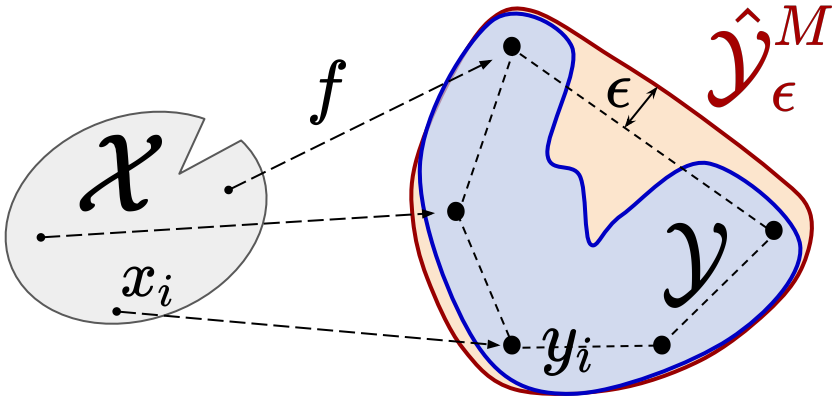
$f : \mathbb{R}^p \to \mathbb{R}^n$ continuous

reachable set to estimate

$$\mathcal{Y} = f(\mathcal{X})$$

3 parameters:

1) a number of samples $M \in \mathbb{N}$
2) a padding constant $\epsilon > 0$
3) a probability measure $\mathbb{P}_{\mathcal{X}}$ on $(\mathbb{R}^p, \mathcal{B}(\mathbb{R}^p))$



Sampling-based reachability analysis

$\mathcal{X} \subset \mathbb{R}^p$ compact

$f : \mathbb{R}^p \to \mathbb{R}^n$ continuous

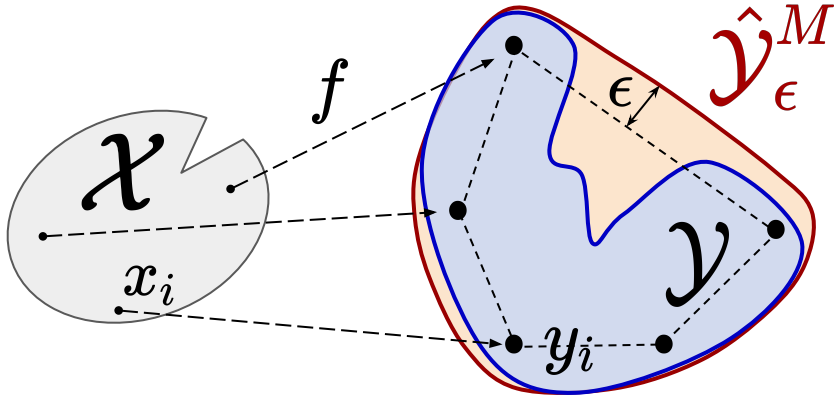
reachable set to estimate

$$\mathcal{Y} = f(\mathcal{X})$$

3 steps:

1) Sample M inputs $x_i \sim \mathbb{P}_{\mathcal{X}}$
2) Evaluate outputs $y_i = f(x_i)$
3) Evaluate estimator

$$\hat{\mathcal{Y}}_\epsilon^M = \mathrm{H}(\{y_i\}_{i=1}^M) \oplus B(0, \epsilon)$$



Sampling-based reachability analysis

$\mathcal{X} \subset \mathbb{R}^p$ compact

$f : \mathbb{R}^p \to \mathbb{R}^n$ continuous

reachable set to estimate

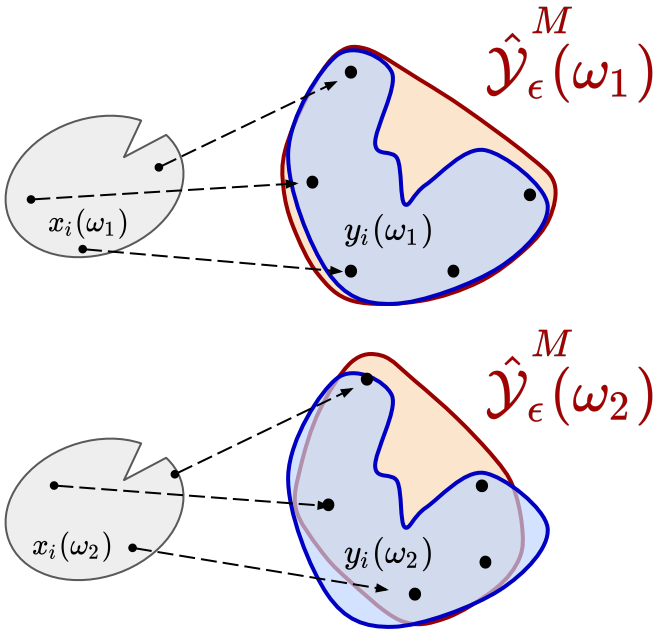
$$\mathcal{Y} = f(\mathcal{X})$$

Let $(\Omega, \mathcal{G}, \mathbb{P})$ be a probability space

Let $\mathcal{K}$ denote the family of nonempty compact subsets of $\mathbb{R}^n$

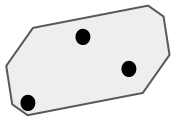
$\hat{\mathcal{Y}}_\epsilon^M : \Omega \to \mathcal{K}$ is a

random compact set

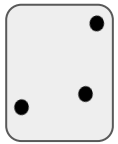


Asymptotics

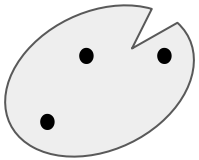
$\mathcal{X}_0$



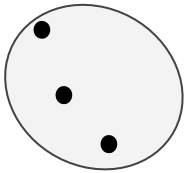
$\mathcal{U}$

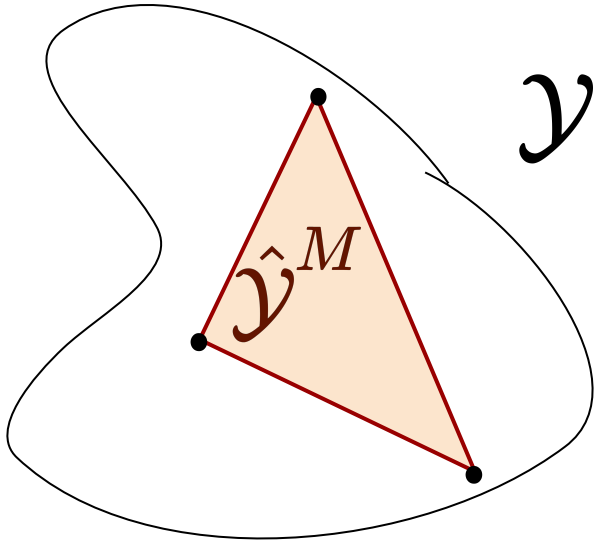


Θ



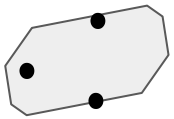
$\mathbb{W}$



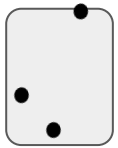


Asymptotics

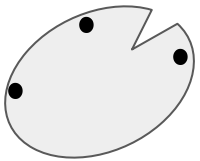
$\mathcal{X}_0$



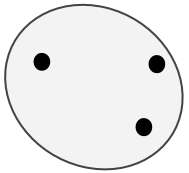
$\mathcal{U}$

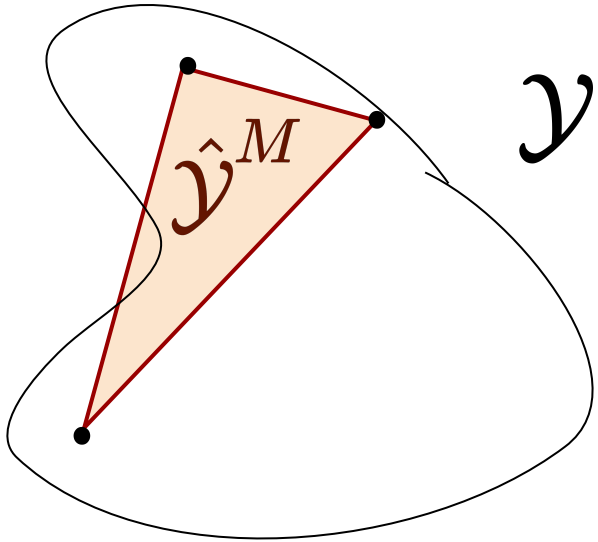


Θ



$\mathbb{W}$





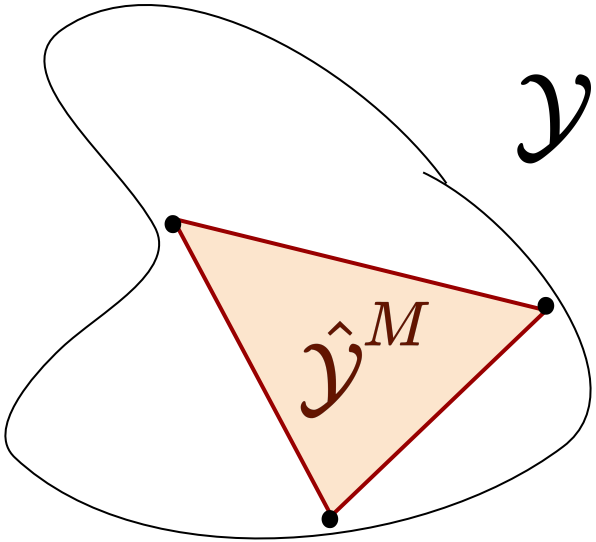
Asymptotics

$\mathcal{X}_0$

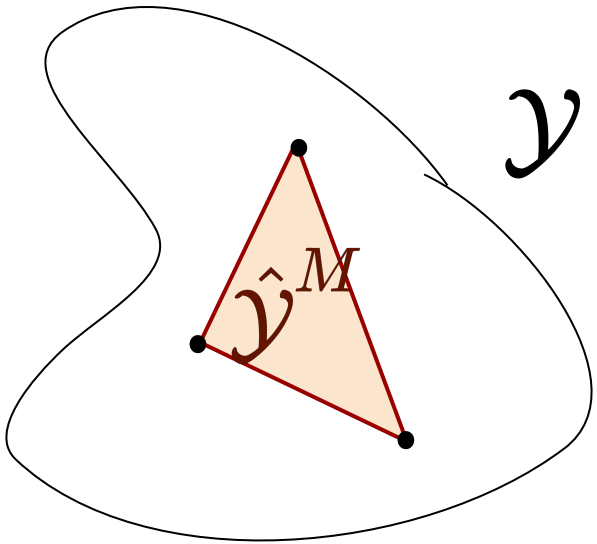
$\mathcal{U}$

Θ

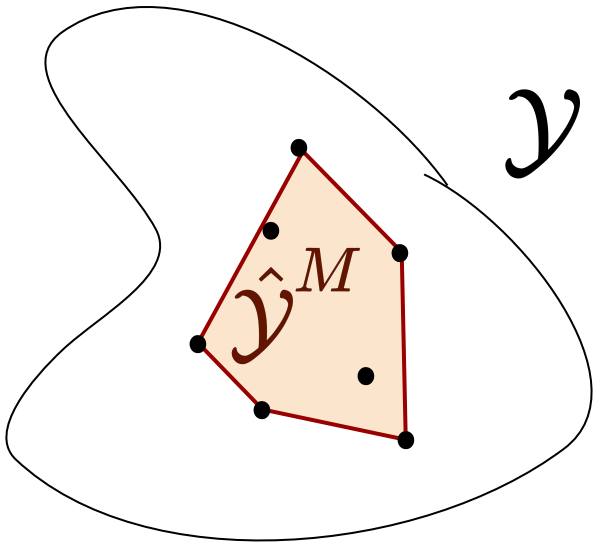
$\mathbb{W}$



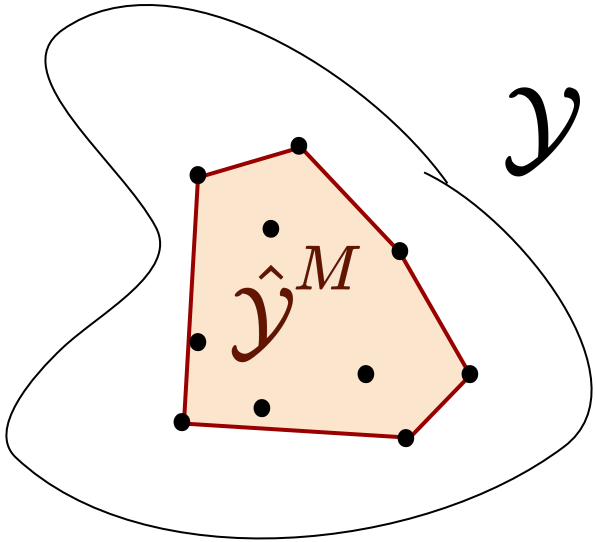
Asymptotics



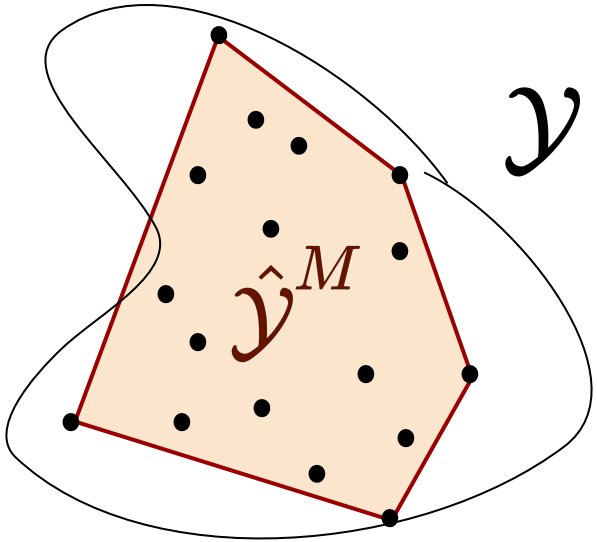
Asymptotics



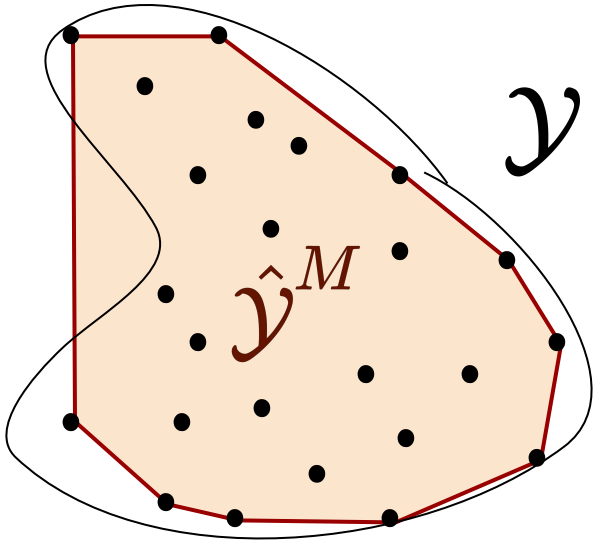
Asymptotics



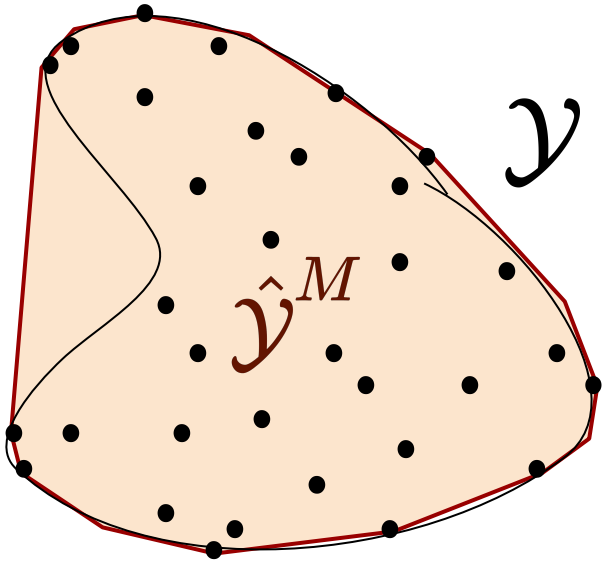
Asymptotics



Asymptotics



Asymptotics

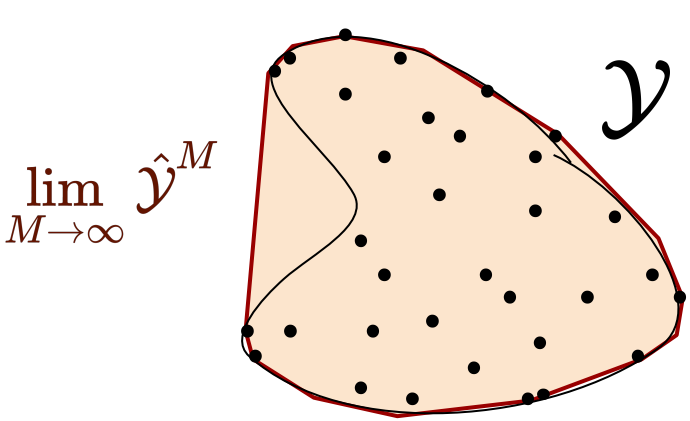


Characterizing the accuracy of the random compact set estimator

Asymptotic analysis

$\mathbb{P}$-almost surely, as $N \to \infty$,

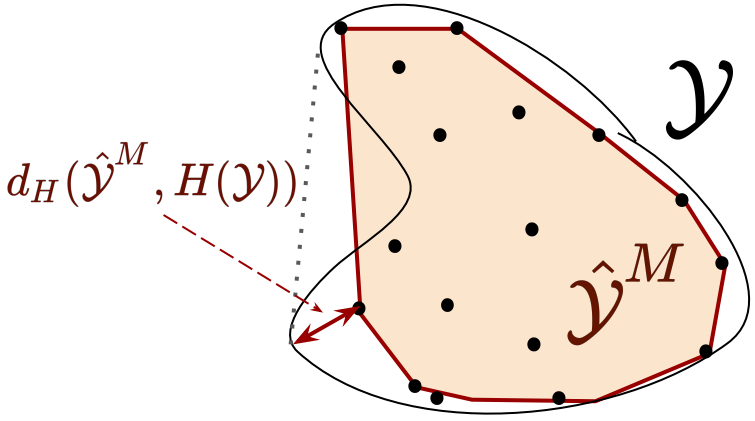
$$d_H\left(\hat{\mathcal{Y}}_\epsilon^M, \mathrm{H}(\mathcal{Y}) \oplus B(0, \bar{\epsilon})\right) \to 0$$



Finite-sample analysis

with $\mathbb{P}$-probability at least $1 - \delta$,

$$d_H\left(\hat{\mathcal{Y}}^M, \mathrm{H}(\mathcal{Y})\right) \leq \epsilon$$

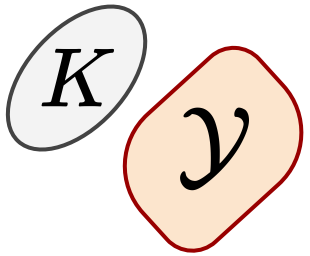


Theorem: Let $\mathcal{Y} \in \mathcal{K}$ and $(\hat{\mathcal{Y}}^M)_{M\in\mathbb{N}}$ be a seq. of random compact sets.

Assume that

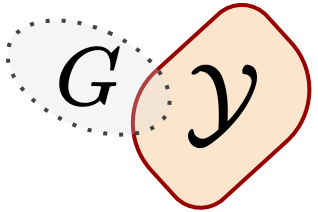
- For any $K \in \mathcal{K}$ such that $\mathcal{Y} \cap K = \emptyset$:

$$\mathbb{P}(\hat{\mathcal{Y}}^M \cap K \neq \emptyset \text{ i.o.}) = \mathbb{P}\left(\bigcap_{N=1}^{\infty} \bigcup_{M=N}^{\infty} \{\hat{\mathcal{Y}}^M \cap K \neq \emptyset\}\right) = 0.$$



- For any open $G \subset \mathbb{R}^n$ such that $\mathcal{Y} \cap G \neq \emptyset$:

$$\mathbb{P}(\hat{\mathcal{Y}}^M \cap G = \emptyset \text{ i.o.}) = \mathbb{P}\left(\bigcap_{N=1}^{\infty} \bigcup_{M=N}^{\infty} \{\hat{\mathcal{Y}}^M \cap G = \emptyset\}\right) = 0.$$



Then, $\mathbb{P}$-almost surely,

$$d_H(\hat{\mathcal{Y}}^M, \mathcal{Y}) \to 0 \quad \text{as} \quad M \to \infty.$$

Ilya Molchanov. Theory of Random Sets. Springer, 2017

Theorem: Assume that $f^{-1}(\partial\mathcal{Y}) \subseteq \mathrm{supp}\,(\mathbb{P}_{\mathcal{X}})$.

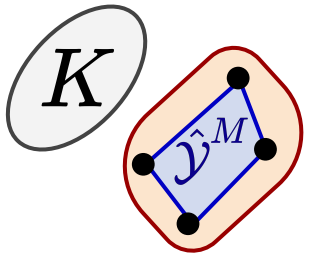
Then,

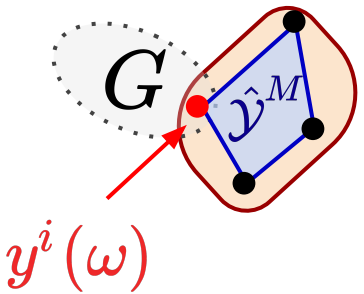
- For any $K \in \mathcal{K}$ such that $\mathrm{H}(\mathcal{Y}) \cap K = \emptyset$:

$$\mathbb{P}(\hat{\mathcal{Y}}^M \cap K \neq \emptyset \text{ i.o.}) = \mathbb{P}\left(\bigcap_{N=1}^{\infty} \bigcup_{M=N}^{\infty} \{\hat{\mathcal{Y}}^M \cap K \neq \emptyset\}\right) = 0.$$

- For any open $G \subset \mathbb{R}^n$ such that $\mathrm{H}(\mathcal{Y}) \cap G \neq \emptyset$:

$$\mathbb{P}(\hat{\mathcal{Y}}^M \cap G = \emptyset \text{ i.o.}) = \mathbb{P}\left(\bigcap_{N=1}^{\infty} \bigcup_{M=N}^{\infty} \{\hat{\mathcal{Y}}^M \cap G = \emptyset\}\right) = 0.$$



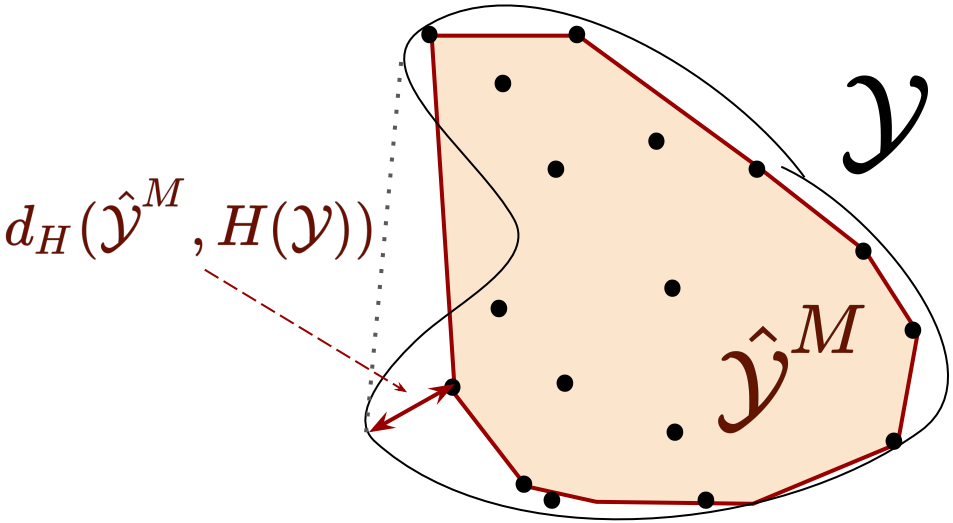


Thus, $\mathbb{P}$-almost surely,

$$d_H(\hat{\mathcal{Y}}^M, \mathcal{Y}) \to 0 \quad \text{as} \quad M \to \infty.$$

Finite-sample error bounds

$$\mathbb{P}\left(d_H\left(\hat{\mathcal{Y}}^M, \mathrm{H}(\mathcal{Y})\right) \leq \epsilon\right) \geq 1 - \delta$$



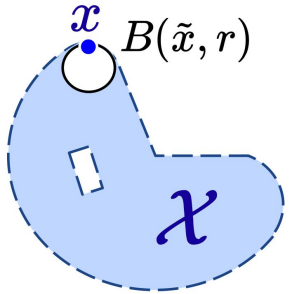
Finite-sample error bounds

Assume:

$\mathcal{X}^c$ is r - convex

f is L - Lipschitz

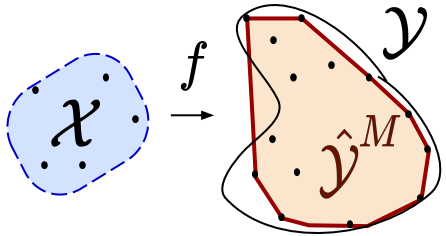
$\mathbb{P}_{\mathcal{X}}(A) \geq p_0 \lambda(A)$ for all $A \subset \mathcal{X}$



Then,

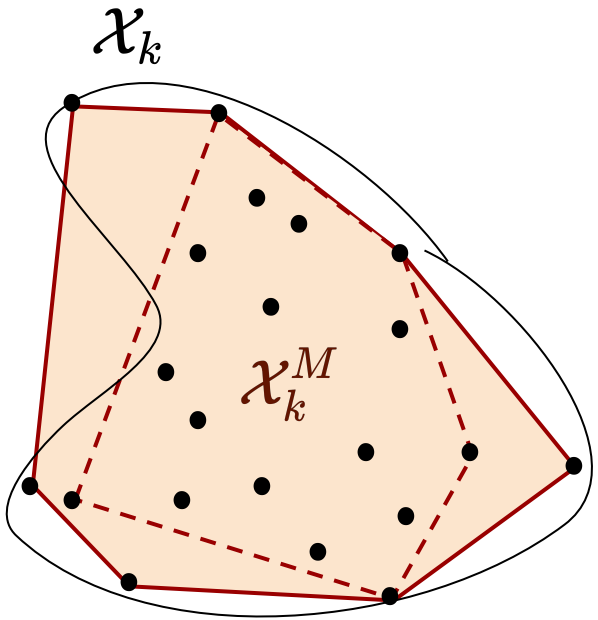
$$\mathbb{P}\left(d_H\left(\hat{\mathcal{Y}}^M, \mathrm{H}(\mathcal{Y})\right) \leq \epsilon\right) \geq 1 - \delta_M$$

with $\delta_M = D(\mathcal{X}, \epsilon/2L)(1 - p_0 \Lambda_\epsilon^{r,L})^M$



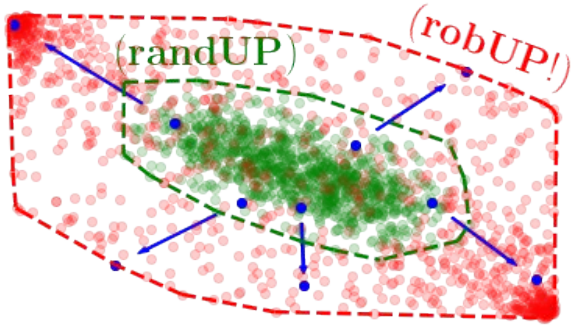
(proof via covering argument. Assumptions can be relaxed along the boundary of the input set.)

Adversarial Sampling

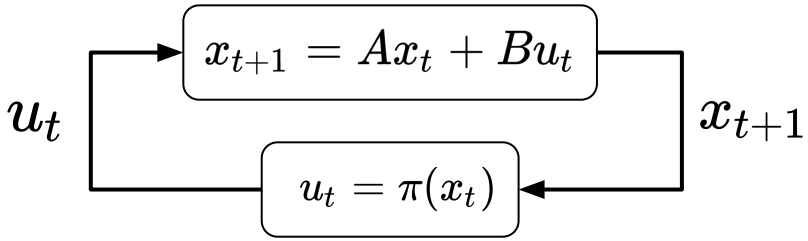


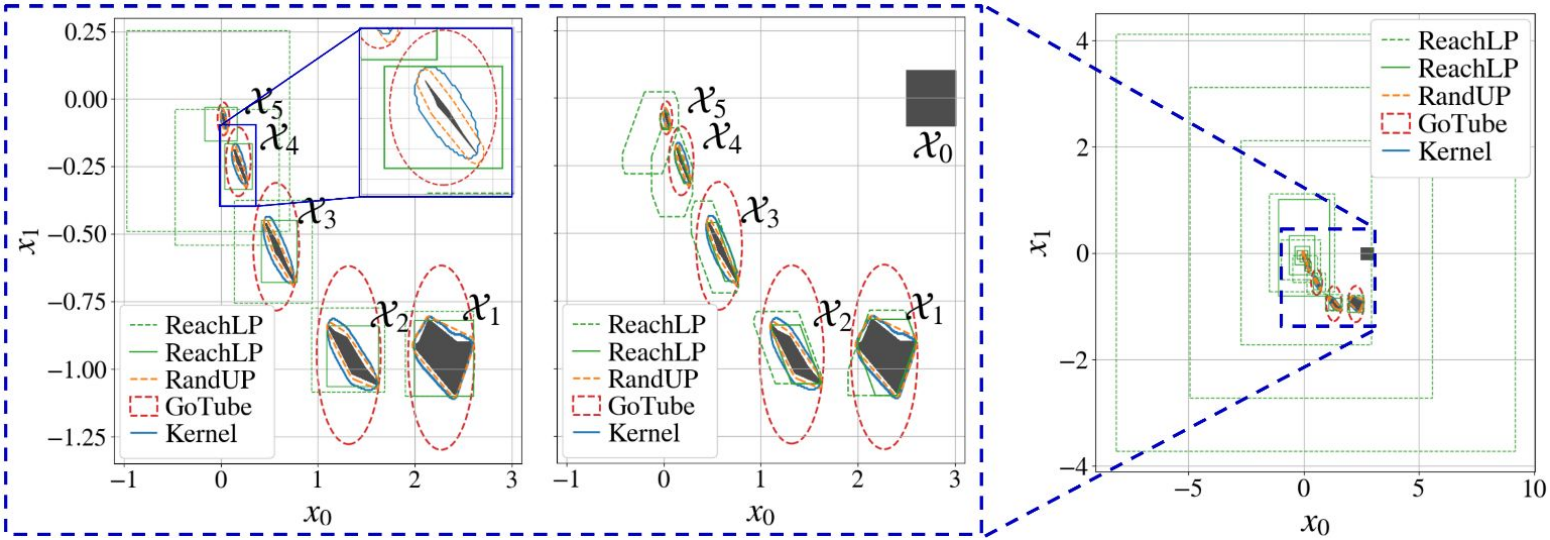
Accuracy depends on samples

=> **Adversarial sampling**

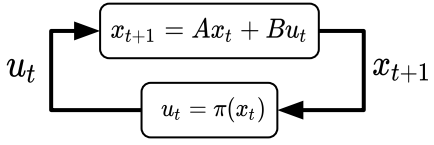


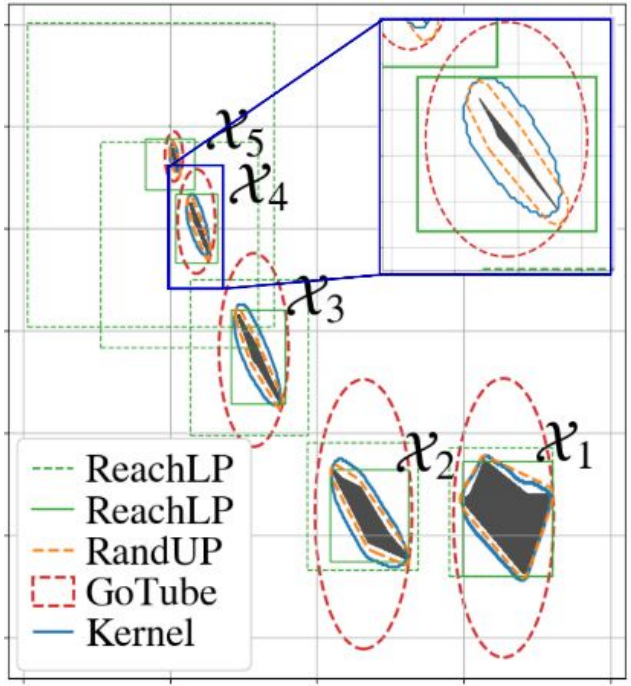
Results for neural network verification

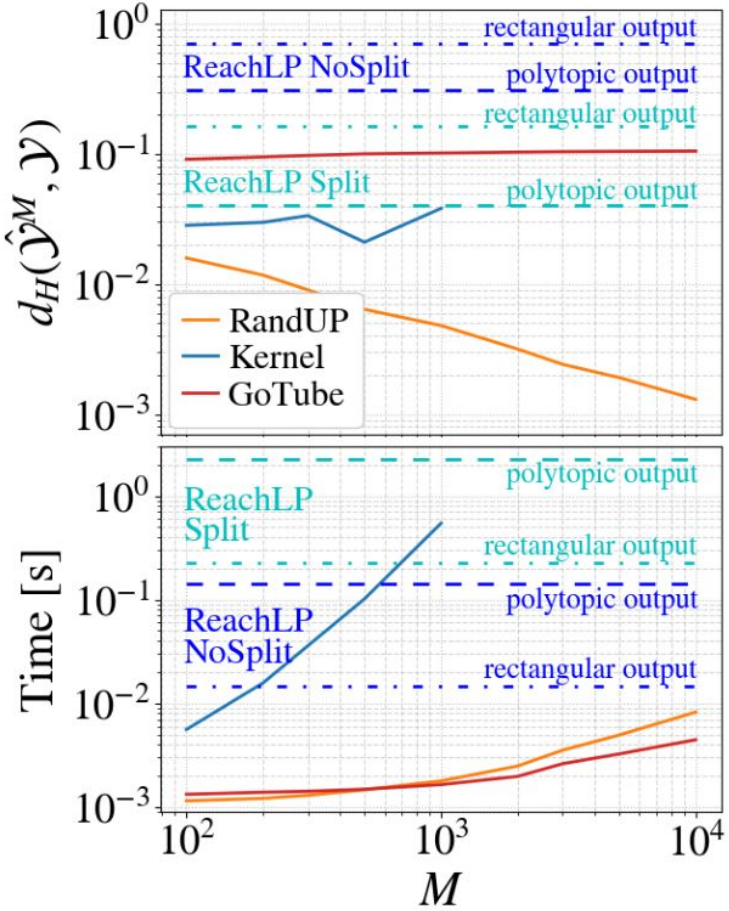




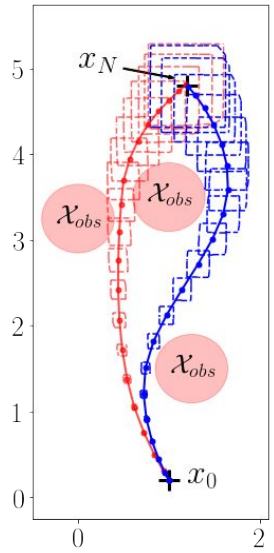
Results for neural network verification

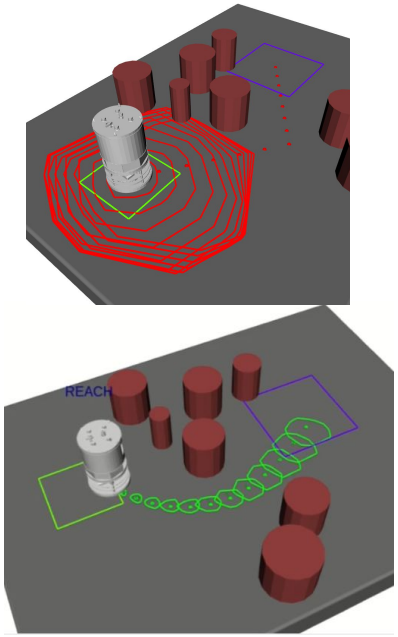


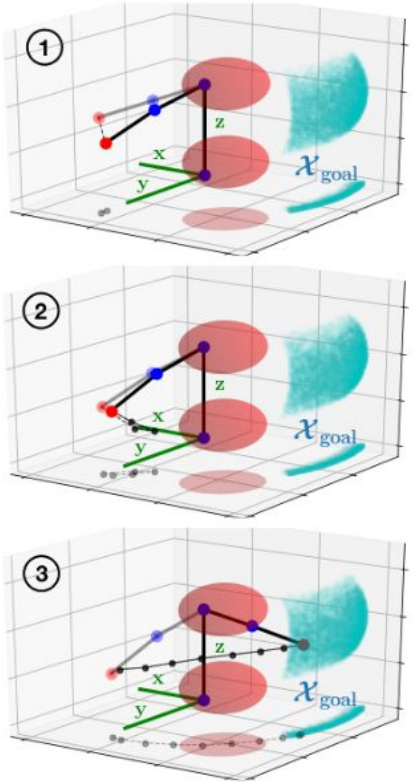


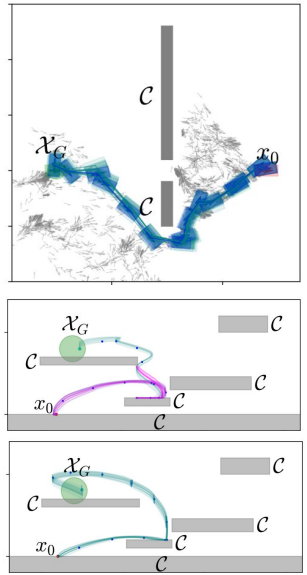


Applications to robust planning & control



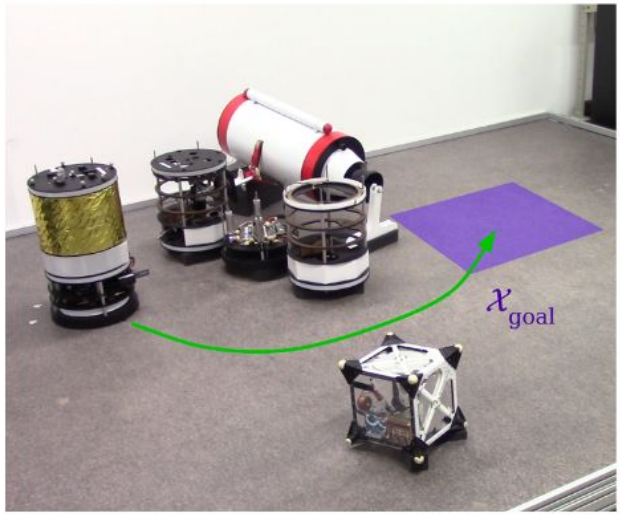


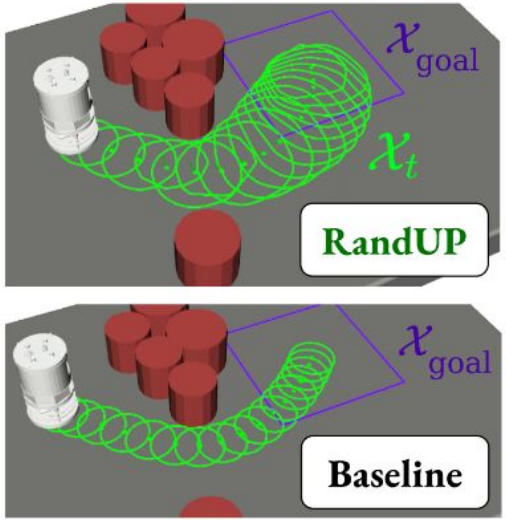


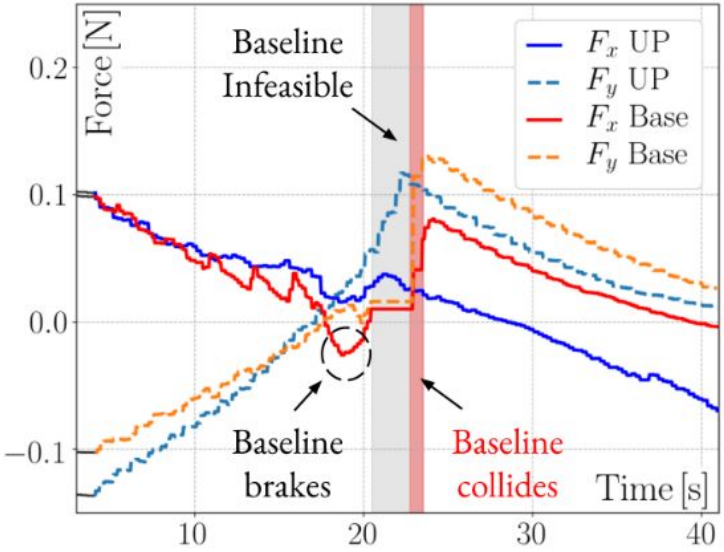


Wu, Lew, Solovey, Schmerling, Pavone, *Robust-RRT,* 2022 (to appear)

Results: robust MPC







RandUP-MPC

Baseline-MPC

MPC feasible

constraints satisfied

MPC infeasible

constraints violated

obstacle collision

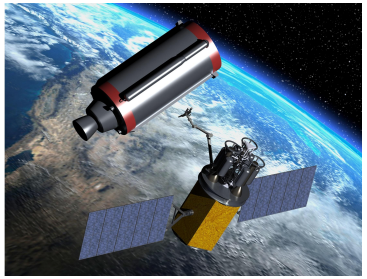
Sampling-based reachability analysis

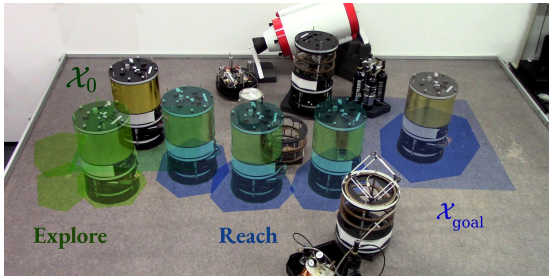
- simple + efficient approach for general reachability analysis
- analysis using random set theory
- can be used for uncertainty-aware planning and control
- hardware-tested

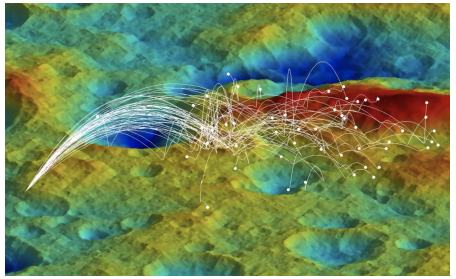
Future work:

- Tighter rates of convergence
- Parallelization on GPUs

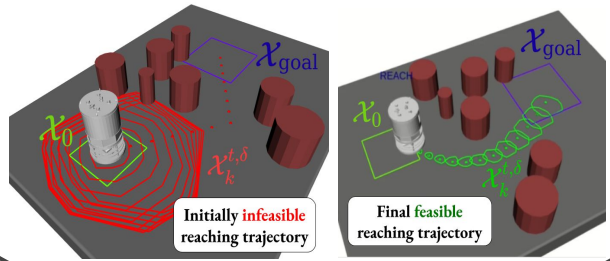
Towards safe learning-based robotic autonomy



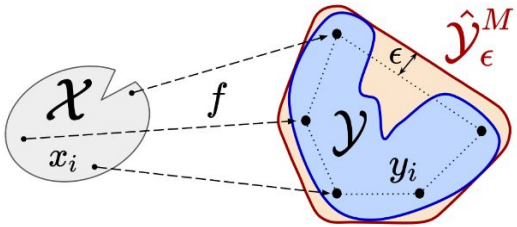




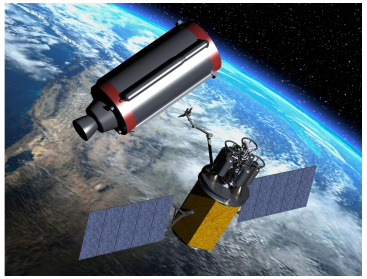
1 safe active dynamics learning and control

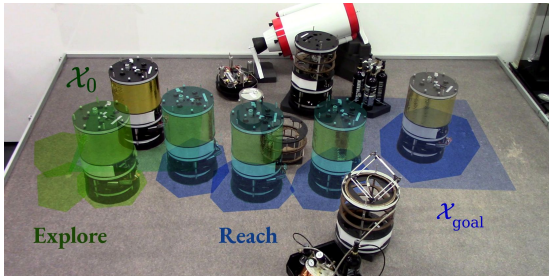


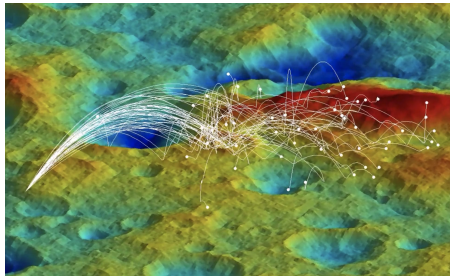
2 sampling-based reachability analysis



Towards safe learning-based robotic autonomy







- Extensions: (1) continuous-time analysis for systems modeled as stochastic differential equations (2) tighter analysis to close the theory-to-practice gap
- New tools to guarantee the performance of complex autonomous systems with learning-in-the-loop components

Acknowledgements


Marco Pavone


Riccardo Bonalli

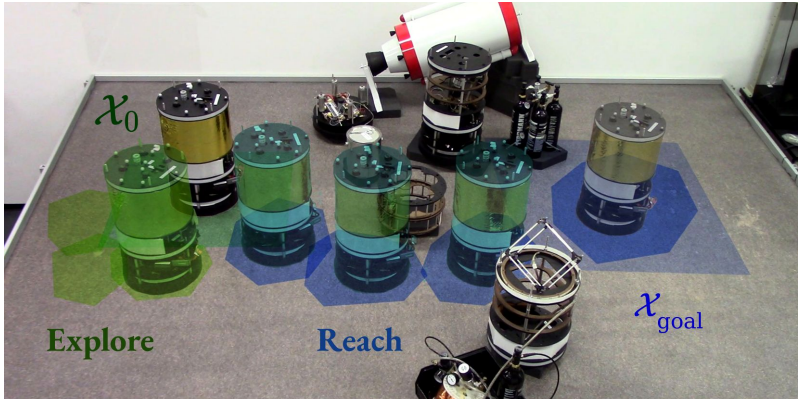

Lucas Janson

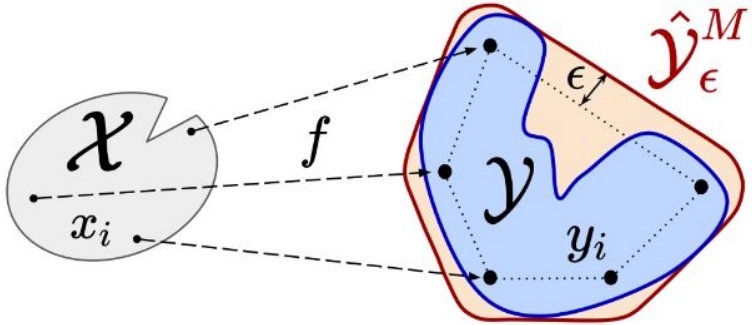

Apoorva Sharma


James Harrison


Andrew Bylard

Autonomous Systems Lab





Uncertainty-aware control strategies for safe learning-based robotic autonomy

Thomas Lew

Autonomous Systems Lab