Learning-based Control for Hybrid Systems

Hybrid Systems Laboratory

Jason J. Choi, Alonso Marco, Claire J. Tomlin

Data-driven Safety Filters for eVTOL Flight Envelope Protection

Motivation for Data-driven Safety Filters

Main two limitations of traditional safety filters based on certificate functions:
- Such certificate functions are hard to verify for high-dimensional and hybrid systems.
- Such safety filters are not robust to the uncertainties of the actual plant.

Methods

GP-regression based safety filter

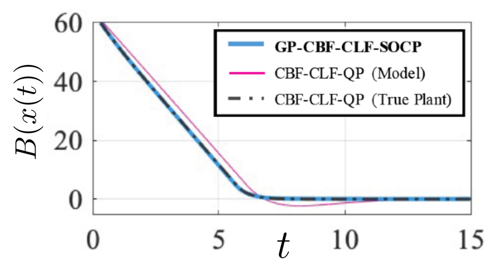
- Classical safety filters based on CBFs (CBF-QP) is designed based on the system model.

$$\underbrace{\widehat{\dot{B}}(x,u)}_{\text{Model-based estimate of true } \dot{B}(x,u)} + \gamma B(x) \geq 0$$

- With the collected data $\{(x_k, u_k, \dot{x}_k)\}_{k=1}^{N}$, model-plant mismatch term $\Delta(x,u) := \dot{B}(x,u) - \widehat{\dot{B}}(x,u)$ can be learned by GP regression, which results in GP-CBF-SOCP safety filter that is convex and has probabilistic safety guarantee [1, 2, 3].

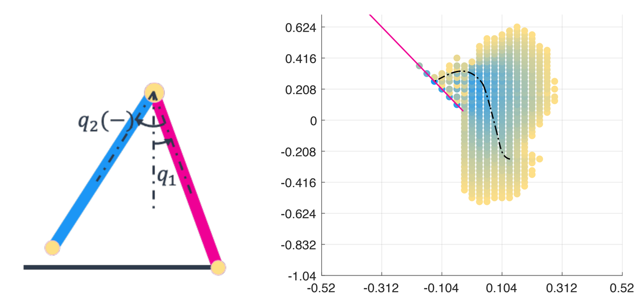
$$\widehat{\dot{B}}(x,u) + \mu(x,u) - \beta(\delta)\sigma(x,u) + \gamma B(x) \geq 0$$

- Example: Adaptive Cruise Control [2, 3]



HJ reachability analysis for hybrid systems [4, 5]

- [4]: HJ reachability frameworks for systems involving transitions between multiple modes.
- [5]: HJ reachability frameworks are extended to **hybrid systems with discontinuous reset maps** by incorporating the value remapping principle.
- Applied to the **stabilization of walking robots**.
- Example: Compass-gait walker



Maximal region-of-attraction verified from HJ reachability

- **On-going work:** Extending [5] to high-dimensional hybrid systems by using neural-network-based approximate dynamic programming [6].

Application: Flight Envelope Protection of NASA Tiltwing Concept Vehicle (eVTOL) [7]

Collaboration with NASA and Bechamo LLC with supports in part from NASA SBIR, RVLT, and TTT MCAAD programs.

Motivation: The need for efficient and automated flight test pipeline for eVTOL control system development.

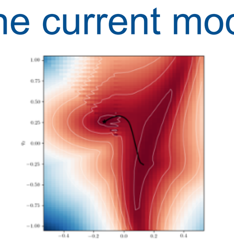
Proposed Framework [8]:

Main Challenges
Hybrid system (hover vs. cruise)
High-dimensional system.

Near-term Deliverable
Apply HJ reachability to the reduced-order model of Tiltwing.

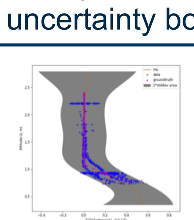
HJ reachability analysis

Verification of the flight envelope based on the current model $\hat{f}_k(x,u,d)$



Flight Envelope $V_{\text{safety}}(x)$

Updated Model with uncertainty bounds $\hat{f}_{k+1}(x,u,d), d \in D(x)$



System Identification based on new data

New Data $\{(x_k, u_k, \dot{x}_k)\}_{k=1}^{N}$

Model Improvement

Flight test & Data Collection

Test controller $\rightarrow u \rightarrow$ Safety Filter $\rightarrow u_{safe}$

References

[1] **GP-CLF-SOCP:** Castaneda*, Choi* et al., Gaussian Process-based Min-norm Stabilizing Controller for Control-Affine Systems with Uncertain Input Effects and Dynamics, ACC 2021.
[2] **GP-CBF-CLF-SOCP & Feasibility Analysis:** Castaneda*, Choi* et al., Pointwise feasibility of gaussian process-based safety-critical control under model uncertainty, CDC 2021.
[3] **Safe online learning with GP-CBF-CLF-SOCP:** Castaneda*, Choi et al., Probabilistic Safe Online Learning with Control Barrier Functions, submitted to CDC 2022.
[4] **Reachability analysis for hybrid systems with multiple modes:** Tomlin et al., Computational techniques for the verification of hybrid systems, IEEE Access, 2003.
[5] **Reachability analysis for hybrid systems with state jumps:** Choi et al., Computation of RoA for Hybrid Limit Cycles Using Reachability: An Application to Walking Robots, RA-L, 2022.
[6] **Deep Neural Network-based HJ Reachability:** Bansal, Tomlin, Deepreach: A deep learning approach to high-dimensional reachability, ICRA 2021.
[7] Whiteside et al., Design of a Tiltwing Concept Vehicle for Urban Air Mobility, NASA, 2021
[8] **Safe learning framework based on HJ reachability:** Herbert, Choi, et al., Scalable learning of safety guarantees for autonomous systems using Hamilton-Jacobi reachability, ICRA 2021

Model-based Continual Learning For Quadruped Locomotion

Learn new model & policy at deployment time

Robust/zero-shot policies trained in simulation assume access to the distribution of the real-world data
- There will always be new situations (e.g., corner cases) not seen at training time
- Re-learn dynamical models and policies at deployment time

Goal-reaching task

Environment $f^i \rightarrow f^{i+1}$

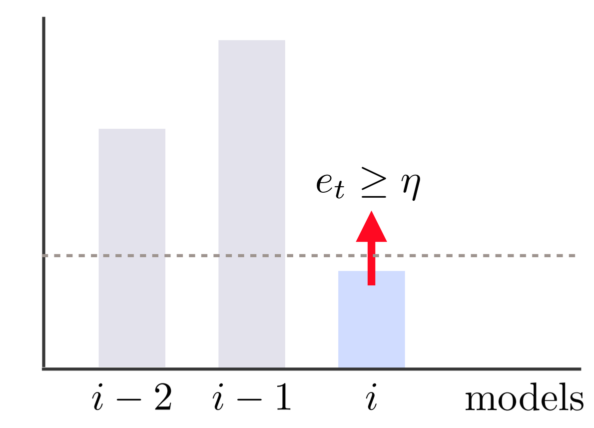
Policy $\pi(\cdot|\theta_i)$

Current environment $x_{t+1} = f^j(x_t, u_t)$

Models $\{\psi_0, \dots, \psi_i\}$

Policies $\{\pi(\cdot|\theta_0), \dots, \pi(\cdot|\theta_i)\}$

Compute prediction error for all models



Trigger learning when $e_t \geq \eta$

$$e_t = \min_{j=\{0,\dots,i\}} -\log p(y_t | x_{t-1}, u_{t-1}; \psi_j)$$

Learn new model & policy using Model-based RL [1]

Cautious exploration

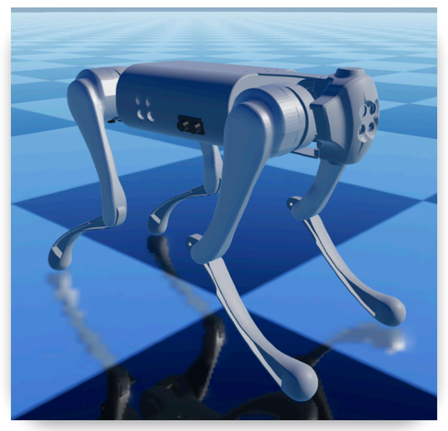
Environment f^{i+1}

Data

Policy $\pi(\cdot|\theta_{i+1})$

Model ψ_{i+1}

Dynamics: Gaussian process + informed prior



Motivation
- Achieve $\mathcal{O}(N)$ complexity with dataset size
- Data-driven prior imposes less structure than white priors

GP model [2]

$$\begin{bmatrix} x_{t+1}^{(1)} \\ \vdots \\ x_{t+1}^{(D)} \end{bmatrix} = \begin{bmatrix} \beta_1^\top \\ \vdots \\ \beta_D^\top \end{bmatrix} \phi(x_t, u_t) + \varepsilon$$

$$\beta_d \sim \mathcal{N}(m_d, \Lambda_d)$$

- Posterior

$$\hat{x}_{t+1}^{(d)} = \hat{\beta}_d^\top \phi(x_t, u_t)$$

$$\hat{\beta}_d \sim \mathcal{N}(\hat{\beta}_d; \mu, \Sigma)$$

- Random Fourier Features [3]

$$\phi(x_t, u_t) = \sqrt{2}\cos(\omega_{\mathrm{x}}^\top x_t + \omega_{\mathrm{u}}^\top u_t + b)$$

$$b \sim \mathrm{U}(0, 2\pi)$$

$$\omega_{\mathrm{x}}, \omega_{\mathrm{u}} \sim S_d(\omega_{\mathrm{x}}, \omega_{\mathrm{u}}) \propto |G_d(j\omega_{\mathrm{x}}, j\omega_{\mathrm{u}})|$$

Spectral density: data-driven

$$G_d(j\omega) = \int_{\mathcal{X},\mathcal{U}} e^{-j\omega_{\mathrm{x}}^\top x_t - j\omega_{\mathrm{u}} u_t} f_d(x_t, u_t) \mathrm{d}x_t \mathrm{d}u_t$$

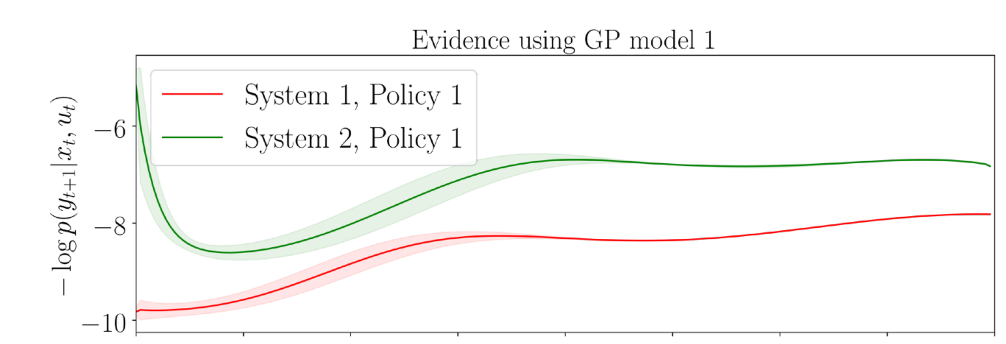
$$\approx \sum_{t=0}^{T-1} e^{-j\omega_{\mathrm{x}}^\top \tilde{x}_t - j\omega_{\mathrm{u}} \tilde{u}_t} \tilde{x}_{t+1}^{(d)}$$

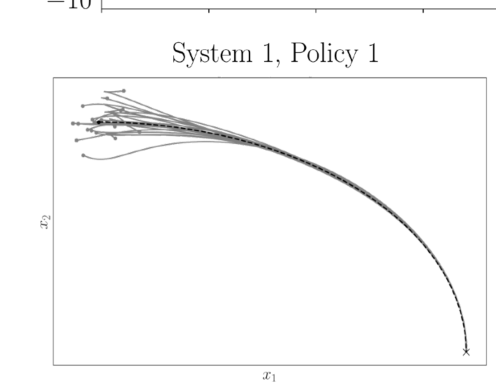
$$\propto \sum_{t=0}^{T-1} \cos(\xi_t)\tilde{x}_{t+1}^{(d)} - j\sin(\xi_t)\tilde{x}_{t+1}^{(d)}$$

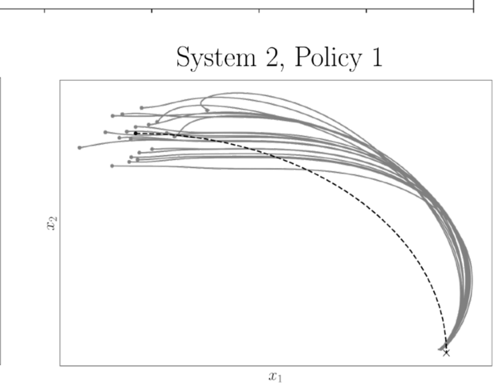
$$\xi_t = \omega_{\mathrm{x}}^\top \tilde{x}_t + \omega_{\mathrm{u}} \tilde{u}_t$$

Toy example

- Goal-reaching task with unknown disturbance
- Model prediction error as triggering mechanism







References

[1] Chua, K., et al. Deep reinforcement learning in a handful of trials using probabilistic dynamics models, NeurIPS 2018
[2] W, Christopher, C. E. Rasmussen, Gaussian processes for machine learning, MIT press, 2006
[3] R, Ali, B. Recht, Random features for large-scale kernel machines, NeurIPS, 2007

The NASA University Leadership initiative (grant #80NSSC20M0163) provided funds to assist the authors with their research, but this article solely reflects the opinions and conclusions of its authors and not any NASA entity.