Uncertainty-aware Safe Exploratory Planning using Gaussian Process and Neural Control Contraction Metric

Dawei Sun[1], Mohammad Javad Khojasteh[2], Shubhanshu Shekhar[3] and Chuchu Fan[2]

[1]University of Illinois at Urbana Champaign [2]Massachusetts Institute of Technology [3]University of California San Diego

Project Homepage: https://realm.mit.edu/L4DC2021

Motivating Example

A car needs to safely explore the environment to learn the effect of different terrains (in different colors) on its dynamics. The light blue regions are pools that the car should avoid.

Problem setup

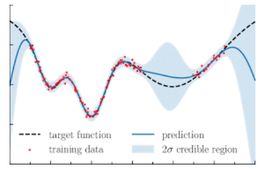
Dynamics: $\dot{\mathbf{x}} = f(\mathbf{x}(t)) + B(\mathbf{x}(t))\,\mathbf{u}(t) + d(\mathbf{x}(t))$

f and B are known, while d is unknown.

The agent gets a noisy observation of $d(\mathbf{x})$ after visiting the state **x**.

The goal is to **safely** collect observations and learn an estimate $\hat{d}$ of the function d from the observations such that the estimation error $||\hat{d}(\mathbf{x}) - d(\mathbf{x})|| \leq \psi_{th}$ for all **x**.

Gaussian process regression



$$\mu_N^{(i)}(\mathbf{x}_*) = K_i(\mathbf{x}_*, \mathbf{x}_{[N]})^\intercal (K_i(\mathbf{x}_{[N]}, \mathbf{x}_{[N]}) + s^2 I_N)^{-1} \mathbf{y}_{i,[N]}$$

$$\sigma_N^{(i)}(\mathbf{x}_*) = \kappa_i(\mathbf{x}_*, \mathbf{x}_*) - K_i(\mathbf{x}_*, \mathbf{x}_{[N]})^\intercal (K_i(\mathbf{x}_{[N]}, \mathbf{x}_{[N]}) + s^2 I_N)^{-1} K_i(\mathbf{x}_*, \mathbf{x}_{[N]})$$

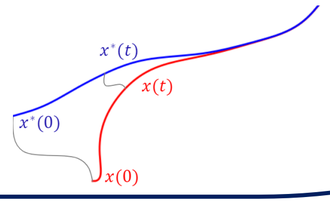
Theorem 1 (Sample complexity): A bound on the number of required samples around a point **x** such that $||\hat{d}(\mathbf{x}) - d(\mathbf{x})|| \leq \psi$ with a high probability.

Learning-based tracking controller

A controller is learned based on the current $\hat{d}$ and certified by a *contraction metric*.

Theorem 3 (Bound on the tracking error): The learned controller can track any reference trajectory $\mathbf{x}^*(t)$. Let the actual trajectory be $\mathbf{x}(t)$. Then the tracking error is bounded as

$$\|\mathbf{x}(t) - \mathbf{x}^*(t)\|_2 \leq \frac{R_0}{\sqrt{\underline{m}}} e^{-\lambda t} + \sqrt{\frac{\overline{m}}{\underline{m}}} \cdot \frac{\psi}{\lambda}(1 - e^{-\lambda t})$$



Safe exploration algorithm and results

The precomputed tracking error can be used to ensure the planned path be safe.

The agent collects more and more data and updates the estimate $\hat{d}$.

The collected data reduces the estimation error ψ and the tracking error, and in turn makes more region safe to explore.

Table 1: Comparison with the baseline method.

Method	Unsafe (%)	Travel time (s)	Tracking error
Algorithm 1	0.3	236	0.051
Baseline ($\mathcal{E} = 0$)	10.3	208	0.243
Baseline ($\mathcal{E} = 0.1$)	5.1	314	0.221
Baseline ($\mathcal{E} = 0.3$)	4.0	515	0.230

Algorithm 1: Safe exploration.

Input: Initial state x; Obstacles $\mathcal{O} \subset \mathcal{X}$;
Input: Error tolerance $\psi_{\texttt{th}}$; Confidence level δ;
Output: Final estimate $\hat{d}$;
Function `Plan` (x, g, $\mathcal{E}$) **:**
  Data: current state x; goal g; bloating factor $\mathcal{E}$;
  Bloating obstacles: $\tilde{\mathcal{O}} = \mathcal{O} \bigoplus \mathcal{B}(0, \mathcal{E})$;
  Plan from x to g while avoiding $\tilde{\mathcal{O}}$;
while *not satisfied* **do**
  Find next goal g to visit using Eq. (5);
  $\rho = \rho_0$; `path` $=$ `null`;
  while `path` *is* `null` **do**
    Compute $\mathcal{E}$ in $\mathcal{B}(\mathbf{x}, \rho)$;
    `path` = `Plan` (x, g, $\mathcal{E}$) ;
    Decrease ρ;
  end
  Move along `path` until reaching the boundary of $\mathcal{B}(\mathbf{x}, \rho)$;
  Enlarge the observation set and update $\hat{d}$;
  Retrain the controller if needed;
end