Value Function Learning for AutoRally Racing

Nolan Wagener, Panagiotis Tsiotras, Byron Boots

Georgia Tech

UNIVERSITY of WASHINGTON

Summary

- We reframe model predictive control (MPC) as a reinforcement learning (RL) approach where each decision is a control sequence.
- This formalizes value learning approaches which rely on MPC to generate the value function targets.
- We show incorporating a value function can improve racing performance of simulated AutoRally car

Model Predictive Control as Reinforcement Learning

Given MDP $\mathcal{M}=(\mathcal{X},\mathcal{U},p,c,\gamma)$

- $\mathcal{X}$: state space
- $\mathcal{U}$: control space
- $p(x_{t+1}|x_t,u_t)$: transition probability
- $c(x_t,u_t,x_{t+1})$: cost function
- γ: discount factor

RL problem: Find policy $\pi(u_t|x_t)$ that minimizes sum of accumulated costs while having high entropy (i.e., noisiness).

$$\min_{\pi} \mathbb{E}_{\pi,p} \sum_{t=0}^{\infty} \gamma^t \left(c(x_t,u_t,x_{t+1}) - \lambda \mathcal{H}\left(\pi(u_t|x_t)\right)\right)$$

Solving this problem may be too difficult, so we form a relaxed RL problem that is potentially easier.

We instead work with a policy $\tilde{\pi}(u_t,\dots,u_{t+H-1}|x_t)$ that gives feedback every H steps. The corresponding MDP is $\widetilde{\mathcal{M}}=(\widetilde{\mathcal{X}},\widetilde{\mathcal{U}},\tilde{p},\tilde{c},\tilde{\gamma})$.

- $\widetilde{\mathcal{X}}=\mathcal{X}$, $\quad \widetilde{\mathcal{U}}=\mathcal{U}^H$, $\quad \tilde{\gamma}=\gamma^H$, $\quad \tilde{\lambda}=\gamma^{H-1}\lambda$
- $\tilde{p}(\tilde{x}_{k+1}|\tilde{u}_k,\tilde{x}_k)=\int \prod_{h=0}^{H-1} p(x_{kH+h+1}|x_{kH+h},u_{kH+h})\, dx_{kH+(1:H-1)}$
- $\tilde{c}(\tilde{x}_k,\tilde{u}_k,\tilde{x}_{k+1})=\mathbb{E}_{x_{kH+(1:H-1)}|x_{kH},u_{kH},x_{kH+H}} \sum_{h=0}^{H-1} \gamma^h c_{kH+h}$

The following RL problem is an upper bound of the original one:

$$\min_{\tilde{\pi}} \mathbb{E}_{\tilde{\pi},\tilde{p}} \sum_{k=0}^{\infty} \tilde{\gamma}^k \left(\tilde{c}(\tilde{x}_k,\tilde{u}_k,\tilde{x}_{k+1}) - \tilde{\lambda}\mathcal{H}\left(\tilde{\pi}(\tilde{u}_k|\tilde{x}_k)\right)\right)$$

Optimal value function $\tilde{V}^*$ satisfies the recurrence

$$\tilde{V}^*(\tilde{x}) = -\tilde{\lambda}\log\int \exp\left(-\frac{1}{\tilde{\lambda}}\mathbb{E}_{\tilde{p}(\tilde{x}'|\tilde{x},\tilde{u})}[\tilde{c}(\tilde{x},\tilde{u},\tilde{x}') + \tilde{\gamma}\tilde{V}^*(\tilde{x}')]\right) d\tilde{u}$$

This recurrence can used to train an approximator of the optimal value function.

\+ Can get accelerated convergence of the value function

\+ Value function targets can be efficiently estimated with algorithms like model predictive path integral (MPPI)

\- Some performance degradation since we represent policy as H-step open loop distribution

Simulated Results

- Cost function encodes that car minimizes lap time while not crashing
- Control algorithm: MPPI used in receding horizon
  - We vary planning horizon and whether we use value function as terminal cost
- Value function is a 2-layer neural network trained with an optimization horizon of 0.5 seconds
- Value function can improve performance and even stabilize car in case of short planning horizon

Planning horizon = 0.5 seconds
No value function used
Lap time: 12.56 ± 1.90 s
5 crashes in 5 trials

Planning horizon = 0.5 seconds
Value function used
Lap time: 6.43 ± 0.29 s
2 crashes in 5 trials

Planning horizon = 1 second
No value function used
Lap time: 7.28 ± 0.42 s
0 crashes in 5 trials

Planning horizon = 1 second
Value function used
Lap time: 6.50 ± 0.22 s
0 crashes in 5 trials