Saferl Research Proposal

hongyic

February 2021

1 Current Method

Currently, we adopt the model-based TD3 algorithm and test in safety gym. The difference of our approach is that at every step, we will select multiple actions from action distribution, and for each action we unroll the state for 10 steps using the trained model and pick the action gives us lowest cost.

The biggest disadvantage is the slow training and testing speed, it takes two times longer (nearly 40 hours in my laptop) than the normal model-based TD3 algorithm to finish 100 episodes training. On the other hand, we notice that this method achieves higher accumulated reward and cost. The reason might be that the model unrolling step facilitates training, thus achieves higher rewards and costs.

2 My Proposal

The overview is shown in Figure 1. The detail explanation of each module is described in next sub-sections.

2.1 Path Prediction

For dynamic moving objects in safety gym, we will use recursive least square parameter adaptation algorithm (RLS-PAA) to predict their long-term path and short-term uncertainty. For multiple dynamic objects, we can train only one network but adjust the weights of last layer for each of the objects.

For ego robot, whose action sequence is determined by the current state and policy, we use dynamic model unrolling to predict its path. Because the robot action is selected from a Gaussian distribution, its uncertainty comes from the standard deviation of Gaussian distribution.

2.2 Planner

From previous prediction module, we get the paths with uncertainty of ego robot and dynamic objects. In planning module, we will check whether ego robot will get too close to objects. We have different strategies for collision checking, like

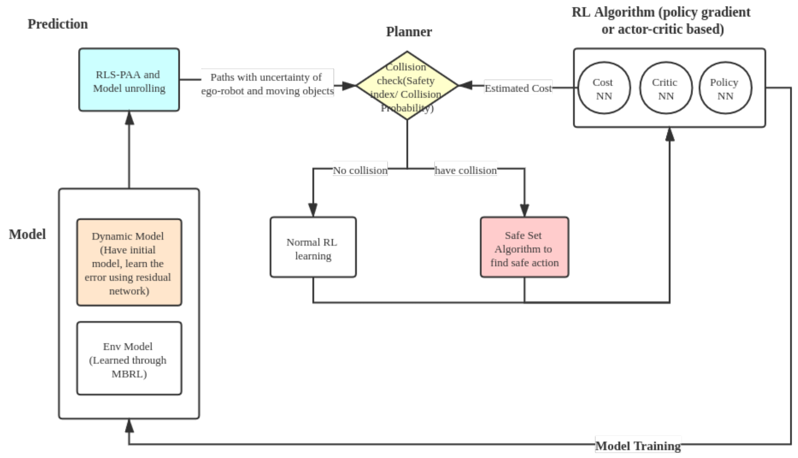
Prediction
RLS-PAA and Model unrolling
Paths with uncertainty of ego-robot and moving objects
Planner
Collision check(Safety index/ Collision Probability)
Estimated Cost
RL Algorithm (policy gradient or actor-critic based)
Cost NN
Critic NN
Policy NN
No collision
have collision
Normal RL learning
Safe Set Algorithm to find safe action
Model
Dynamic Model (Have initial model, learn the error using residual network)
Env Model (Learned through MBRL)
Model Training

Figure 1: Safe RL Workflow

calculating the safety index, calculating the probability of collision, using cost network in reinforcement learning algorithm module to predict future cost or check if the path are overlapped.

If they are collision free, our reinforcement learning policy controls the robot without any modification. Otherwise, we use the safe set algorithm (SSA) to find out the safe action. (If we use SSA, I guess the collision check we can only use safety index method)

2.3 Dynamic Model and Environment Model Training

In original Model-based RL work, dynamic model and environment model are trained together, which make the training process more difficult and less accurate compared to training them separately using the prior information we have.

We train the environment model (¡state, action¿ -¿ ¡reward, cost¿) using probabilistic ensemble model training method, that is we train multiple networks to for environment model for representing uncertainty when lacking data.

For dynamic model, we use the bicycle dynamic model as our prior information (for car robot) and learn the residual between our prior model and real dynamic. As you can see in Figure 2

2.4 Reinforcement Learning (RL) Algorithm

Except for the commonly used actor network and critic network, we will train another cost network, which is becoming popular recently, for example openAI benchmark paper [2] is using cost network in PPO, TRPO algorithms.

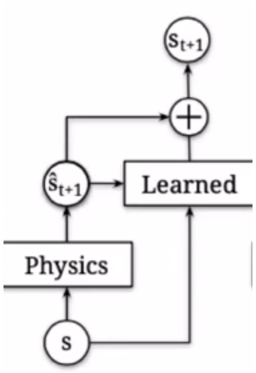

Figure 2: Residual Learning for Dynamic Model

3 Other Ideas

3.1 Long Term Planning

As described in [1], the convex feasible set (CFS) algorithm can be applied here. However, if we use CFS to generate an optimal trajectory, we need to change the RL algorithm to goal-indexed one. The advantage of this method is effective, but the number of training samples will decrease dramatically. We will first implement the basic version and see if we need to do this.

References

[1] Changliu Liu, Te Tang, Hsien-Chung Lin, Yujiao Cheng, and Masayoshi Tomizuka. Serocs: Safe and efficient robot collaborative systems for next generation intelligent industrial co-robots, 2018.

[2] Alex Ray, Joshua Achiam, and Dario Amodei. Benchmarking Safe Exploration in Deep Reinforcement Learning. 2019.