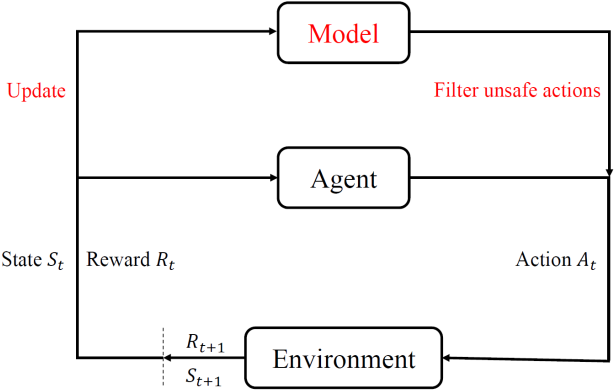

Model
Update
Filter unsafe actions
Agent
State S_t
Reward R_t
Action A_t
R_{t+1}
Environment
S_{t+1}

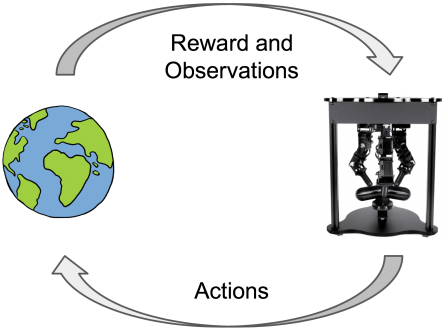
Learnable Model Verification through Reinforcement Learning

December 9, 2021

Yao Feng
yaofeng@andrew.cmu.edu
15624

Anita Li
weihanl1@andrew.cmu.edu
15824

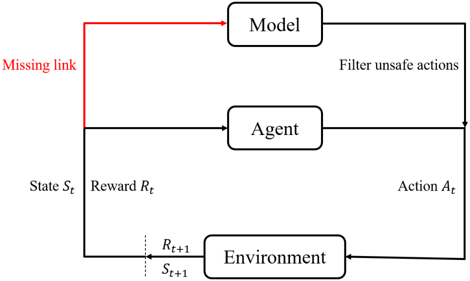
Motivation



- Model-based Reinforcement Learning (MBRL): Learning accurate modeling from the environment
- Formal Verification: Provides safety guarantee for CPS
- Connect MBRL with formal verification:

Safe Reinforcement Learning

A missing link



- Justified Speculative Control (JSC): the first algorithm proposed to incorporate safety guarantees into the learning process of RL
- Verification Preserving Model Updates(VPMU): use a fixed set of models in the learning process

Terms

Given the verified model $\mathtt{init}_\theta \rightarrow [\{\mathtt{ctrl}_\theta ; \mathtt{plant}\}^*]\mathtt{safe}$, we have the following definitions:

Definition

A sequence of tuples $(s^t, a^t, \pi^t, \theta^t)$ describe a **Learning Process**, with the **Transition Function** $T(s^t, a^t) = s^{t+1}$, $a^t \sim \pi^t(s^t)$.

Definition

(Controller Monitor) if $CM(s, a) = true$, $(s, T(s, a)) \in [\![\mathtt{ctrl}]\!]$ [1].

Definition

(Model Monitor) if $MM(s, a, s') = true$, $(T(s, a), s') \in [\![\mathtt{plant}]\!]$ [1].

Definition

Let H be an **update function** of parameter θ such that

$$H(s^t, a^t, s^{t+1}, \theta^t) = \theta^{t+1}$$

Learnable Justified Speculative Learning (LJSL)

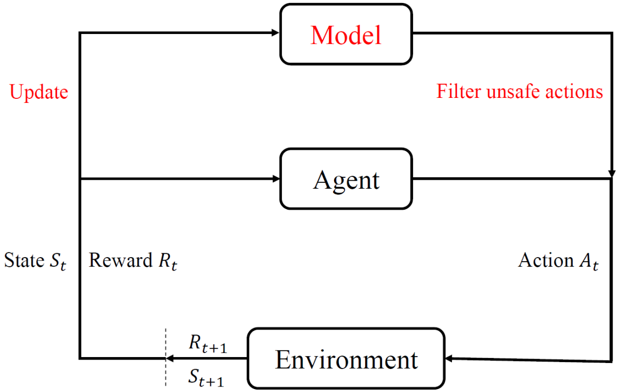


Figure: The framework of LJSL

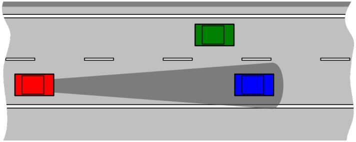
Main Theorem

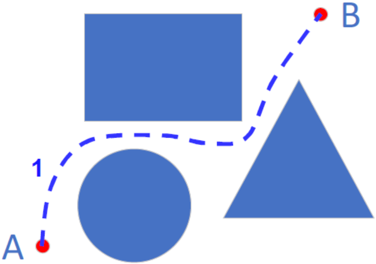
If for all θ, the model $\texttt{init}_\theta \rightarrow [\{\texttt{ctrl}_\theta; \texttt{plant}\}^*]\texttt{safe} \wedge \texttt{init}_\theta$ is valid, CM, MM are accurate control monitor and model monitor and H is a valid update function ($\texttt{init}_{H(\theta,s,a,s')} \rightarrow [\{\texttt{ctrl}_{H(\theta,s,a,s')};\texttt{plant}\}^*]\texttt{safe}$ is accurate for any accurate model $\texttt{init}_\theta \rightarrow [\{\texttt{ctrl}_\theta;\texttt{plant}\}^*]\texttt{safe}$ and $s' = T(s,a)$), and $\texttt{init}_\theta \rightarrow \texttt{init}_{H(\theta,s,a,s')}$, we have $s_t \models \texttt{safe}$ for all $t \geq 0$.

Proof Idea: the model guarantees the initialization of the updated model holds, so we can connect the safety proofs of a sequence of model by induction.

Experiments

- Continuous Adaptive Cruise Control (CACC)
- Robot Motion





Simplified Robot Motion: Goal Finding

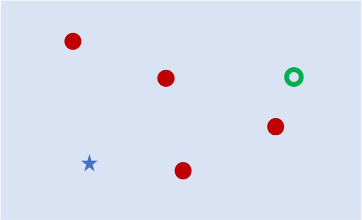


Figure: The star is the moving agent, the green circle is the goal, the red circles are the obstacles to avoid.

Example Model Sketch

$$\theta = (r_{min}, r_{max})$$

$$\texttt{init}_\theta \rightarrow [\{\texttt{ctrl}_\theta;\texttt{plant}\}^*]\texttt{safe}, \ \textit{where}$$

$$\texttt{init}_\theta \equiv \texttt{valid\_env} \wedge \texttt{const\_bounds}_\theta,$$

$$\texttt{valid\_env} \equiv \bigwedge_{i=1}^{n} \texttt{dist\_safe}(agent, obs_i) \wedge \bigwedge_{i\neq j}^{n} \texttt{dist\_safe}(obs_i, obs_j)$$

$$\texttt{ctrl}_\theta \equiv r \text{ satisfies } \{\bigwedge_{i=1}^{n} \texttt{dist\_safe}(agent_r, obs_i)) \wedge r! = 0\}$$

`plant` describes the agent's circular movement

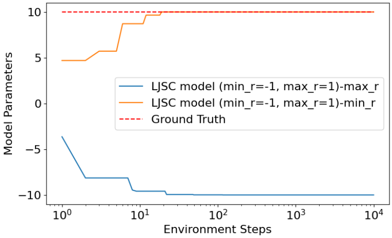
$$\texttt{safe} \equiv \texttt{no\_crash}$$

Experiments

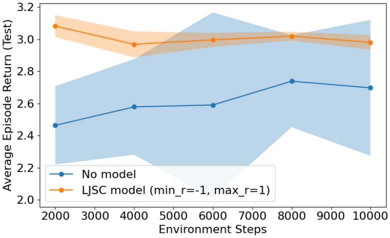
To satisfy the preconditions of our theorem, we need to implement the following functions:

- RL Algorithm: Soft Actor-Critic (SAC)
- Control Monitor and Model Monitor: can be easily induced from the proven model
- Update Function $H(s, a, s', \theta) = \{\min(a_r, r_{min}), \max(a_r, r_{max})\}$

Results: Goal Finding



(a)



(b)

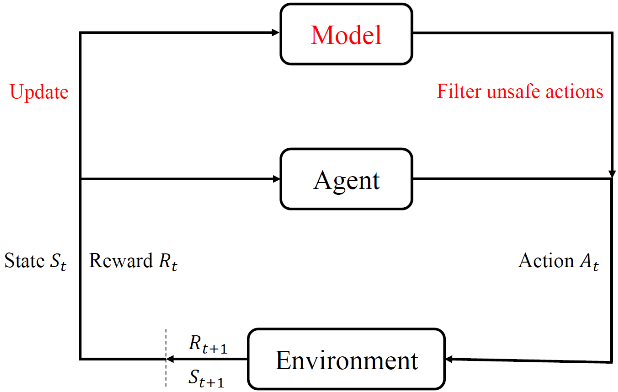
Figure: (a) the learning process of parameters for an imperfect LJSL model (b) test rewards of agents with no model and an LJSL model with imperfect initialization ($r_{min} = -1, \ r_{max} = 1$)

Conclusion

- + Proposed LJSL algorithm and proved the main theorem of safety
- + Implemented 3 different environments and verified effectiveness through experiments

- - The model monitors are conservative, and might still have room for improvements
- - Need much prior information, such as forms of θ and H
- - Simultaneously updating more than one parameter

Teaser Page


Model
Update
Filter unsafe actions
Agent
State S_t
Reward R_t
Action A_t
R_{t+1}
S_{t+1}
Environment