Barrier Certificates through Reinforcement Learning

Papers Consulted:

Learning Control Barrier Functions and their application in Reinforcement Learning: A Survey

Safe and Efficient Reinforcement Learning Using Disturbance-Observer-Based Control Barrier Functions

Safe Reinforcement Learning via Shielding

Reinforcement Learning:

Powerful technique for developing robot behaviour.
Here,

- Agent interact with enviornment.
- Take action **(a).**
- Get reward or penalty.
- Learn a policy.

Steering / Danger Compass:

B(x) = ∇ B(x) . f(s,a) <= 0

Here,

- ∇ B(x) = Time derivative of B(x)
  (It is a vector of partial derivatives)

∇ B(x) = [d/dx1(B) , d/dx2 (B) . . . d/dxn (B)]

Barrier function : B(x) = x1^2 , x2^2

Example in 2D
Assume , your vehicle is in the origin.
∇ B(x) = d/dx1^2 , d/dx2^2

S = (x1 , x2)	∇ B(x)	Direction	Notes
(1 , 0)	[2 , 0]	Points Right	
(0 , 1)	[0 , 2]	Points Up	
(-1 , 0)	[-2 , 0]	Points Left	
(0 , -1)	[0 , -2]	Points Down	

Assume:

Current Motion = Vector = f (s , a)

f (s , a.forward) = [0 , 1] // Moving North

Turn Right

f (s , a.right) = [1 , 0] // Moving East

Markov Decision Process (MDP) : Environment Model

An agent , given an unknown environment , can be modelled by MDP.

MDP = (S , s1 , A , P , R)

Here,

S = Finite set of states (Safe + Unsafe).

s1 belongs to S = Unique initial state (Start State).

A = Finite set of actions (Brake , Speed , Turn).

P = Probabilistic transition function (Rules).

R = Reward function.

Shield is computed by the reactive synthesis of MDP & Preemptive Shielding .

Preemptive Shielding

At every time-step ‘ t ‘ , Shield computes a set of all safe steps { a (t1) , a (t2) . . a (tn) }. From all the available actions (safe / unsafe). Now the environment executes an action a(t) and moves to the next state S (t + 1) . And provide reward R (t + 1) .

Self Driving Car Example

Deep Q - Network (DQN) with a Boltzmann exploration policy

- Type of RL algorithm.
- Estimates how good an action is in a state.

Here Agent ,

➔ See a state ‘S’
➔ Compute ‘Q’ values for all the actions
➔ Pick highest ‘Q’

Boltzmann exploration policy

It convert the value of ‘Q’ into probabilities using **“Softmax Formulae”**

$$P(a/s) = \frac{e^{Q(s,a)/T}}{\Sigma a' e^{Q(s,a')/T}}$$

P(a/s): Probability of picking action ‘a’ in state ‘s’
Q(s,a): Q value for that action ‘a’ in state ‘s’
T: Temperature parameter
- High
- Low
- 0

e ^ Q (s,a) / T: Convert Q into positive probability
Σa’ e ^ Q (s,a’) / T: Make all probabilities add up to 1