2024 IEEE Space, Aerospace and Defence Conference (SPACE) | 979-8-3503-6738-6/24/$31.00 ©2024 IEEE | DOI: 10.1109/SPACE63117.2024.10668292

Safe Deep Reinforcement Learning-Based Controller (SDRLC) for Autonomous Navigation of Planetary Rovers

Ravi Kiran Jana
Dept. of Aerospace Engg.
Indian Institute of Science
Bangalore, India
ravikj@iisc.ac.in

Rudrashis Majumder*
ARTPARK
Indian Institute of Science
Bangalore, India
rudrashismajumder@gmail.com

Bharathwaj K S
Dept. of Aerospace Engg.
Indian Institute of Science
Bangalore, India
bharathwaj019@gmail.com

Suresh Sundaram
Dept. of Aerospace Engg.
Indian Institute of Science
Bangalore, India
vssuresh@iisc.ac.in

***Abstract*—Surface exploration and data collection by planetary rovers are challenging due to unknown complex planet terrains. This paper focuses on developing a Deep Reinforcement Learning (DRL)-based controller for rovers to enable safe operation. The necessary control input for safe and efficient vehicle maneuver is derived using the Control Barrier Function (CBF)-based safety protocols. Deep Deterministic Policy Gradient (DDPG) algorithm is used as a DRL framework to find the optimal exploration policies for the rover. Numerical simulations on different vehicle models show the efficacy of the proposed safety method for planetary rovers.**

***Index Terms*—Deep Reinforcement Learning, Path planning, Obstacle avoidance, Control Barrier Function, Rover.**

I. Introduction

In situ exploration of Earth's neighboring planets by sending autonomous space vehicles allows scientists to make more detailed observations and gather more extensive data compared to remote explorations. Planetary rovers [1] are autonomous vehicles designed to land on other planets to collect important surface data on soil composition, mineralogy, atmospheric data, etc. However, remotely controlling the rovers operating on other planets is highly challenging, considering low bandwidth and high latency. The rovers must not only navigate such uneven terrains while avoiding collisions with obstacles such as rocks, but they must also not fall into a crater or turn over. Hence, safe path planning algorithms must be integrated into these autonomous vehicles to explore unknown and non-uniform terrains efficiently.

Avoidance of obstacles or craters includes sensing the environment and deriving adequate control inputs to navigate the vehicles safely. Algorithms for avoiding obstacles need to utilize kinematic or dynamic models of vehicles. Various state-of-the-art methods for collision avoidance are found in the existing literature: Artificial Potential Functions (APF) [2], collision cones [3], Model Predictive Control (MPC) [4], etc. However, the APF algorithm can (a) lead autonomous vehicles to trap at local minima, (b) generate inefficient paths, or (c) lead to collisions with dynamic obstacles. Collision cones for multiple obstacles demand higher computation. Similarly, MPC is a receding horizon controller which is computationally expensive.

A popular development in the literature of safety and obstacle avoidance includes the mathematical concept of Control Barrier Function (CBF) [5] due to its robustness, efficiency, and smoothness of the vehicle trajectory. The research on CBF-based controllers primarily focuses on designing controllers for autonomous vehicles so that the vehicles do not cross the safety barriers to enter unsafe regions. Adequate control inputs for vehicle maneuvers to avoid these unsafe zones are derived by solving a Quadratic Programming (QP) problem with a CBF-based inequality constraint. CBFs have been applied to provide safety guarantees from collisions in various robotic applications such as adaptive cruise control [6], collaborative load transport [7], etc. However, solving multiple QP problems in a cluttered and uncertain environment is challenging due to a higher computational burden.

In this paper, CBF is combined with a Deep Reinforcement Learning (DRL) framework for the rover to find an optimal path that satisfies safety constraints. Traditional CBF design relies on the knowledge of the system dynamics and constraints and can be computationally expensive. The combination of CBF and Reinforcement Learning (RL) tackles both safety and performance aspects of control problems. The CBF prevents the RL agent from exploring unsafe regions while the RL agent learns the best way to navigate within those safe boundaries. Being aware of the CBF constraints through the training process, the RL agent can implicitly understand safety boundaries without needing every specific constraint programmed, thereby reducing the complexity. DRL-based approaches received attention due to not requiring environmental maps, strong learning capabilities, and high dynamic adaptability [8] with uncertainties. The Deep Neural Network (DNN) can efficiently handle high-dimensional information. The trained network can directly generate optimized control

*Corresponding Author

979-8-3503-6738-6/24/$31.00 ©2024 IEEE

Authorized licensed use limited to: J.R.D. Tata Memorial Library Indian Institute of Science Bengaluru. Downloaded on September 27,2024 at 15:17:06 UTC from IEEE Xplore. Restrictions apply.

inputs for the rovers employing sensor information. It is applicable for dynamic path planning of planetary rovers while facilitating exploration of the unknown environment with limited onboard computing resources. This paper uses the DDPG algorithm [9] to derive the optimal policies for safe exploration. Although RL-CBF algorithms using policy gradient methods and CBF can be found in [10], [11], the main focus of these papers is not on the safe navigation of autonomous vehicles. The major contributions of the present paper can be highlighted as the following:

- We explore the potential application of the proposed Safe Deep Reinforcement Learning-based Controller (SDRLC) to solve the QP problem for obstacle avoidance of planetary rovers.
- This study includes different vehicle models to validate the efficacy of the proposed SDRLC framework.

The paper is organized as follows. Section II introduces the essential mathematical concepts. In Section III, the problem formulation for the SDRLC framework is discussed. The numerical simulations are presented in IV. Section V concludes the paper while proposing the future work.

II. Mathematical Preliminaries

This section presents the preliminary mathematical concepts of CBF and DRL algorithm adopted in the present paper.

A. Control Barrier Function (CBF)

CBF provides the constraints in optimization-based control problems by identifying a safe set for the autonomous vehicles. We consider a non-linear control affine system as

$$\dot{x} = f(x) + g(x)u, \tag{1}$$

f and g being Lipschitz globally. The state $x \in \mathrm{R}^n$ and control command $u \in \mathrm{R}^m$ are constrained in closed sets. The initial condition is $x(t_0) = x_0$.

Definition 1 (*Relative degree* [12]): The relative degree of a differentiable function $h : \mathbb{R}^n \to \mathbb{R}$ with respect to system (1) is the number of times it needs to be differentiated along the dynamics of (1) until the control u explicitly comes.

For a continuously differentiable function $h : \mathbb{R}^n \to \mathbb{R}$, let

$$C := \{x \in \mathbb{R}^n : h(x) \geq 0\} \tag{2}$$

be a super-level set of h.

Definition 2 (*Forward Invariance* [12]): A set $C \subset \mathbb{R}^n$ satisfies forward invariance for system (1) if $x(t_0) \in C$ implies $x(t) \in C, \forall t \geq t_0$.

Definition 3 (*Class $\mathcal{K}$ function* [12]): A Lipschitz continuous function $\alpha : [0, a) \to [0, \infty)$ for $a > 0$ is a class $\mathcal{K}$ function if it is strictly increasing and $\alpha(0) = 0$.

Definition 4 (*Control Barrier Function* [12]): Considering set C as given in (2), $h(x)$ is a control barrier function (CBF) for system (1)) if there exists a class $\mathcal{K}$ function α such that

$$\mathcal{L}_f h(x) + \mathcal{L}_g h(x)u + \alpha(h(x)) \geq 0 \tag{3}$$

for all $x \in C$, where $\mathcal{L}_f, \mathcal{L}_g$ stand for Lie derivatives along the direction of f and g, respectively.

Theorem 1 [12]: Given a CBF h associated with set C from (2), any Lipschitz continuous controller $u(t) \in U, t \geq t_0$ satisfying (3) makes the set C forward invariant for (1).

For set C, the CBF $h(\cdot)$ can be designed in such a way that C represents a safe set for the rovers. If the control input u satisfies *Theorem 1*, set C is forward invariant and hence, the rovers will always remain inside C. Hence, the vehicle safety is always guaranteed since C is already a safe set. Since every systems are not first-order in inputs (*Definition 1*), we may need to use High Order Control Barrier Functions (HOCBF) [12] to deal with systems with a relative degree higher than one.

B. Deep Reinforcement Learning

RL agent learns a strategy for sequential decision-making through active interaction with dynamic systems [13]. Markov Decision Processes (MDP) are used to simulate such dynamic systems, which might be fully or partially observable. An MDP is represented as a tuple $\mathcal{S}, \mathcal{A}, \mathcal{O}, T, R, \gamma, P_0$, where (1) $\mathcal{S}$ is a set of agent states in the environment, (2) $\mathcal{A}$ is a set of agent actions, (3) $\mathcal{O}$ is a set of observations in the partially observable case, (4) $T : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is the transition function, (5) R is the reward function, (6) $\gamma \in [0, 1]$ represents the discount factor, and (7) $P_0 : \mathcal{S} \to [0, 1]$ represents the initial state distribution.

A policy $\pi : \mathcal{S} \to \mathbb{P}(\mathcal{A})$ is a function that maps the state space to the probability distribution of actions. $\pi_\theta(a|s)$ represents the probability of taking action a in state s based on the policy parameterized by θ. The objective lies in the maximization of the cumulative reward $J(\theta) = \mathbb{E}_{\tau \sim p_\theta(\tau)} \left[\sum_t \gamma^t R(s_t, a_t)\right]$. The objective of the RL problem is to sample trajectories τ under $\pi_\theta(a|s)$.

To find the policy that maximizes cumulative reward $J(\theta)$, we compute the policy gradient with regard to θ as $\nabla_\theta J(\theta) = \mathbb{E}_{\tau \sim \pi_\theta(\tau)} \left[\nabla_\theta \log \pi_\theta(\tau) G(\tau)\right]$, with $G(\tau) = \sum_t \gamma^t R(s_t, a_t)$ [13]. The Q-function of a policy π is defined as $Q^\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ at each state-action pair (s, a). For a policy π, $Q^\pi(s_0, a_0) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma [R(s_t, a_t)\right]$ represents the trajectory's estimated return. The policy can be deterministic as $\mu_\theta : \mathcal{S} \to \mathcal{A}$.

Since the gradient of the cumulative reward (or, objective function) is determined by the differentiation of actions, a continuous action space is needed. An off-policy approach is used for its better sampling efficiency and stability. Deep Deterministic Policy Gradient (DDPG) [9] is a widely used DRL algorithm for continuous control problems. The Actor-Critic DNNs guarantee in finding a policy that maximizes the objective function. The convex QP problem and safety constraints are embedded into the RL environment. The optimal control action from the QP is stored in the experience buffer and later sampled for the agent to learn safe actions.

uthorized licensed use limited to: J.R.D. Tata Memorial Library Indian Institute of Science Bengaluru. Downloaded on September 27,2024 at 15:17:06 UTC from IEEE Xplore. Restrictions apply

The above mathematical concepts of CBF and DRL are proposed to automate the navigation of planetary rovers as presented in Section III.

III. Problem Framework

In the control affine system (1), SDRLC creates a control input u to meet the goals indicated by the reward function in the MDP while adhering to the safety rules provided by CBF. First, the RL policy creates action without considering the safety guarantee as

$$u_{\text{RL}}(t) = \pi(x_t;\theta^\pi) + \mathcal{N}_t, \tag{4}$$

where $\pi(\cdot;\theta^\pi)$ is a policy characterized by the DNN parameter θ^π and $\mathcal{N}_t$ is a random noise that promotes the exploration of RL agent [14].

Next, CBF ensures that the controller satisfies the safety constraint by solving the following convex quadratic program for control synthesis:

$$\begin{aligned} &\min_{u\in U} ||u-u_{\text{RL}}||^2 \\ \text{subject to}\quad &\sup_{u\in U}\ [\mathcal{L}_f h(x)+\mathcal{L}_g h(x)u+\alpha(h(x))] \geq 0 \end{aligned} \tag{5}$$

where U is the set of admissible control input, $h(x)$ is the CBF, and $\alpha(h(x))$ is a class $\mathcal{K}$ function. The distance-based CBF used in this paper is given as

$$h = (x-x_0)^2 + (y-y_0)^2 - r^2 \geq 0 \tag{6}$$

where (x_0, y_0) is the coordinate of the obstacle centre and r denotes the sum of obstacle radius and safety tolerance.

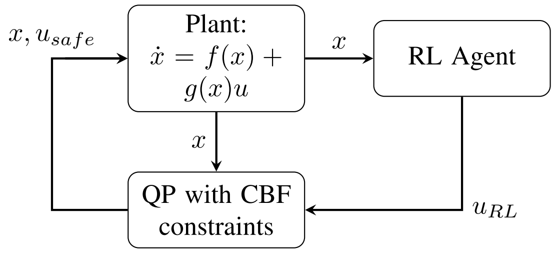


Fig. 1: Block diagram describing the problem formulation

The block diagram of the above formulation can be seen in Fig. 1. Equation (1) describes the nonlinear system, (4) defines the RL policy, and (5) represents the convex QP problem with CBF given as (6).

The vehicle models and reward functions used in training are discussed next. The reward function for the DRL model is built in such a way that it penalizes the squared distance between the agent and the goal states scaled by a coefficient d. It also penalizes each time step by a constant s to reach the goal faster.

A. Point-mass Model

A second-order point-mass model given as

$$\dot{x} = u,\ \dot{y} = v,\ \dot{u} = a_x,\ \dot{v} = a_y \tag{7}$$

is considered as the vehicle kinematics where $[x, y, u, v]$ is the state vector and $[a_x, a_y]$ is the control vector. Since the relative degree of this model is two, a second-order CBF constraint [12] is used in the QP as a constraint.

The reward function [14] for training the point-mass model using the proposed SDRLC algorithm is given below.

$$R_{pm} = -d\|P-P_f\|_2^2 - s \tag{8}$$

where $d = 0.6$ and $s = 1$. Here, P and P_f are the position vectors of rover and goal, respectively.

B. Unicycle Model

A first-order unicycle model is considered as another vehicular model represented as

$$\dot{x} = V\cos\theta,\ \dot{y} = V\sin\theta,\ \dot{\theta} = \omega \tag{9}$$

where x, y, θ are the state parameters, ω is control input, and V is the linear velocity kept constant to $1.5m/s$. Since a model with relative degree one, first-order [6] CBF constraints are used in QP.

The reward function for training the uni-cycle model using SDRLC algorithm is proposed as

$$R_{uc} = -d\|P-P_f\|_2^2 - k\|\Theta\|_2^2 - s \tag{10}$$

where $d = 0.6$, $s = 1$, and $k = 1.2$. Here, P and P_f are the position vectors of rover and goal, respectively, and Θ is the angle between the vehicle direction and line joining the vehicle and goal.

IV. Results and Discussions

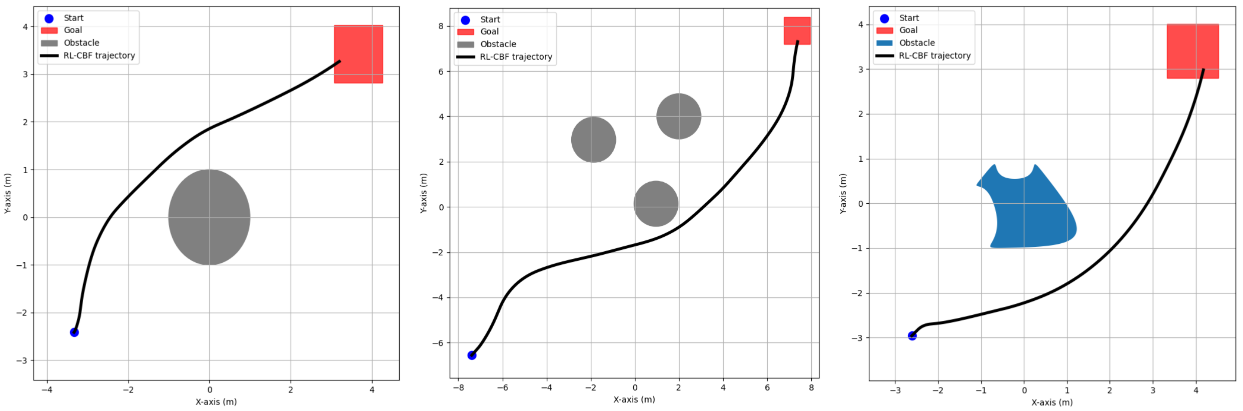
Considering ground vehicles, the motions of the vehicles are simulated in two-dimensional environments. The experiments are simulated using Python and executed on an Nvidia GeForce 3050 GTX. The PyTorch framework is employed for implementing the actor-critic networks (DDPG), with the hyperparameters specified in Table I. RL agent undergoes training for 4000 episodes for both vehicle models, with 1500 steps per episode. An episode is terminated if the goal is reached.

TABLE I: Hyperparameters for training the neural networks

Parameters	Value
Hidden layers of Actor-Critic networks	(128, 64)
Batch size	128
Learning rate of Critic	0.001
Learning rate of Actor	0.0002
Update rate of Target	0.2

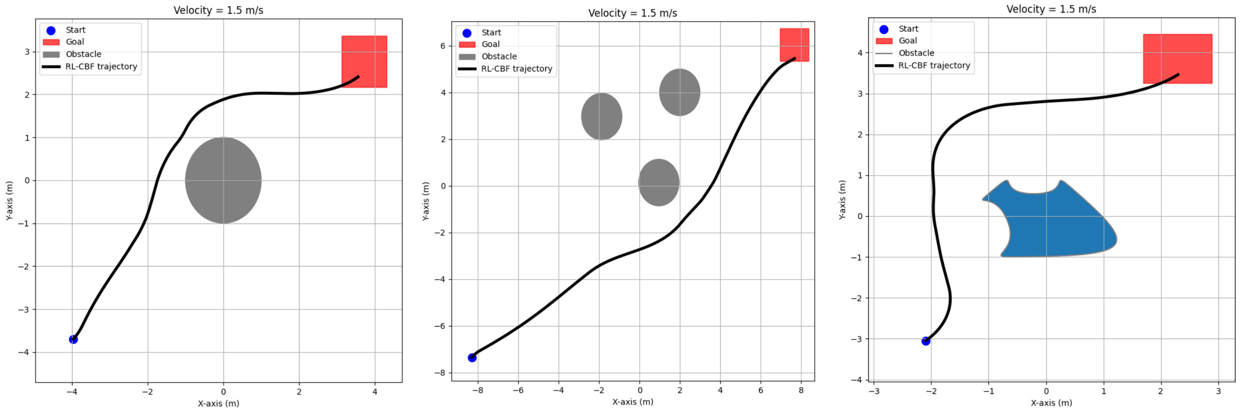
Following the training, the RL agents for both models were evaluated in three distinct environments with (a) a single circular obstacle, (b) multiple circular obstacles, and (c) a

uthorized licensed use limited to: J.R.D. Tata Memorial Library Indian Institute of Science Bengaluru. Downloaded on September 27,2024 at 15:17:06 UTC from IEEE Xplore. Restrictions apply



(a) Avoiding single circular obstacle (b) Avoiding multiple circular obstacles (c) Avoiding non-convex obstacles/craters

Fig. 2: Simulation results for the point-mass model



(a) Avoiding single circular obstacle (b) Avoiding multiple circular obstacles (c) Avoiding non-convex obstacles/craters

Fig. 3: Simulation results for the uni-cycle model

single obstacle/crater of non-convex shape. The results of these simulations are presented in Fig. 2 and Fig. 3 for the respective scenarios of the point-mass and unicycle models, respectively. It is evident that the proposed SDRLC method of combined CBF and DRL framework successfully enables the rover to avoid an obstacle or a crater on planetary surfaces. The proposed SDRLC algorithm is also generalizable to environments with multiple obstacles and non-convex-shaped craters.

V. Conclusions

In this paper, the concepts of CBF and RL are combined to provide a learning platform for planetary rovers for safe and reliable exploration. CBF provides safety guarantee to the vehicle on uneven terrains. RL framework allows the rovers to learn optimal control strategies that maximize performance objectives while respecting the safety constraints enforced by the CBF. This paper employs the DDPG algorithm as the DRL framework. Numerical simulation on two different vehicle models show the efficacy of the SDRLC method in different environments. Extension of this research in the future will consider more extensive simulations with a six-wheeled rover and the implementation of the proposed methodology in hardware.

References

[1] A. Thoesen and H. Marvi, “Planetary surface mobility and exploration: A review,” *Current Robotics Reports*, vol. 2, no. 3, pp. 239–249, 2021.

[2] J. Sun, J. Tang, and S. Lao, “Collision avoidance for cooperative UAVs with optimized artificial potential field algorithm,” *IEEE Access*, vol. 5, pp. 18 382–18 390, 2017.

[3] X. Xu, W. Pan, Y. Huang, and W. Zhang, “Dynamic collision avoidance algorithm for unmanned surface vehicles via layered artificial potential field with collision cone,” *The Journal of Navigation*, vol. 73, no. 6, pp. 1306–1325, 2020.

[4] M. A. Abbas, R. Milman, and J. M. Eklund, “Obstacle avoidance in real time with nonlinear model predictive control of autonomous vehicles,” *Canadian Journal of Electrical and Computer Engineering*, vol. 40, no. 1, pp. 12–22, 2017.

[5] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, “Control barrier function based quadratic programs for safety critical systems,” *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3861–3876, 2016.

uthorized licensed use limited to: J.R.D. Tata Memorial Library Indian Institute of Science Bengaluru. Downloaded on September 27,2024 at 15:17:06 UTC from IEEE Xplore. Restrictions apply

[6] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *2019 18th European Control Conference (ECC)*. IEEE, 2019, pp. 3420–3431.

[7] N. Rao and S. Sundaram, "Integrated decision control approach for cooperative safety-critical payload transport in a cluttered environment," *IEEE Transactions on Aerospace and Electronic Systems*, 2023.

[8] X. Yu, P. Wang, and Z. Zhang, "Learning-based end-to-end path planning for lunar rovers with safety constraints," *Sensors*, vol. 21, no. 3, p. 796, 2021.

[9] H. Tan, "Reinforcement learning with deep deterministic policy gradient," in *2021 International Conference on Artificial Intelligence, Big Data and Algorithms (CAIBDA)*. IEEE, 2021, pp. 82–85.

[10] R. Cheng, G. Orosz, R. M. Murray, and J. W. Burdick, "End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 3387–3395.

[11] H. Zhang, Z. Li, and A. Clark, "Model-based reinforcement learning with provable safety guarantees via control barrier functions," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 792–798.

[12] W. Xiao and C. Belta, "High-order control barrier functions," *IEEE Transactions on Automatic Control*, vol. 67, no. 7, pp. 3655–3662, 2021.

[13] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[14] A. E. Chriat and C. Sun, "On the optimality, stability, and feasibility of control barrier functions: An adaptive learning-based approach," *IEEE Robotics and Automation Letters*, vol. 8, no. 11, pp. 7865–7872, 2023.

uthorized licensed use limited to: J.R.D. Tata Memorial Library Indian Institute of Science Bengaluru. Downloaded on September 27,2024 at 15:17:06 UTC from IEEE Xplore. Restrictions apply