Efficient Safe Learning for Robotic Systems in Unstructured Environments

Sara Pohland, Sylvia Herbert, Claire Tomlin

***Abstract*—Learning-based control schemes that can preserve safety without overly constricting the learning process are beneficial to ensure safe operation while a robotic system is learning about its environment and to enable the system to update safety guarantees as it obtains new information. A previously developed safe learning control scheme was successful in utilizing reachability analysis to enable a system to safely generate a model of its environment and update its safety guarantees based on this model [1]. However, this scheme was not able to efficiently update the safety analysis online, due to the computational intensity of reachability analysis. This limited the applicability of this control scheme to high-dimensional systems, which suffer most from high computation times. Recent work has developed a method of warm-start reachability analysis to improve computation time of safety updates by initializing safety analysis with previous computations [2]. The work in this paper improves upon the previous safe learning control scheme by utilizing warm-start reachability analysis to decrease computation time and create an efficient safe learning framework for robotic systems. This lowered computation time presents the potential to apply the efficient safe learning control scheme to high-dimensional systems in unstructured environments.**

I. Introduction

Through reinforcement learning, a user can give a system a task to complete without specifying the specific series of actions it must perform to complete that task. Instead, the system learns how to complete the task with minimal human intervention by receiving rewards for choosing actions that progress it towards the completion of the task. This is extremely beneficial when the system operates in an unstructured environment and needs to be able to adapt to changes in its environment. However, since the system is not designed to handle constraint satisfaction, systems that operate solely through reinforcement learning are unemployable in safety-critical situations. Conversely, Hamilton-Jacobi (HJ) reachability analysis is a technique that determines the safe region within a state space by assuming the system operates under the worst-case conditions. Reachability analysis is robust and computationally intensive, specifying the constraints of the system and ensuring performance and safety guarantees. While reachability analysis differs from reinforcement learning in that it can allow for safety guarantees, this technique works most effectively when the dynamics of the system and the environment are known and unchanging. This is due to the fact that HJ reachability analysis is computationally intensive and therefore is typically used offline as a precomputation. This presents the desire to combine reinforcement learning with reachability analysis to enable systems to learn about their environment, while guaranteeing safety.

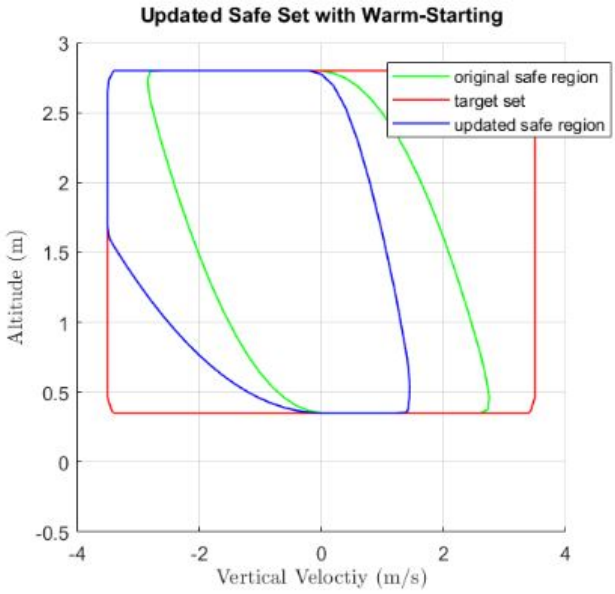

Fig. 1: Computations of safe sets for a quadcopter in a room. The red set defines the target set: the set of states that the quadcopter must remain in. In this case the target set represents the constraints on altitude and velocity of the quadcopter. The original safe region (green) is computed using a given disturbance bound. When inside of this bound the quadcopter is guaranteed to not violate the target set despite worst-case disturbances. As the knowledge of the disturbance grows, the updated safe set (blue) is computed using warm-start reachability.

Previous work has been done to create a safe learning framework that combines reinforcement learning with reachability analysis by using Gaussian Processes (GPs) to compute a model of the environment and HJ reachability analysis to guarantee safety [1]. This work was successful in guaranteeing the safe operation of a quadcopter exploring an environment with unexpected disturbances, while minimally interfering with the learning process. Furthermore, the authors were successful in refining the safety analysis as the system learned more about its environment, forcing the system to operate more or less conservatively as necessary. However, this work found difficulty in recomputing and refining this safety analysis in real time due to the computational burden of HJ reachability analysis.

Since this safe learning framework was established, addi-

This research is supported by the NSF SUPERB REU program.
Pohland is with the Department of Electrical and Computer Engineering at the University of Maryland; Herbert and Tomlin are with the Department of Electrical Engineering and Computer Sciences at UC Berkeley. Contact info: spohland@umd.edu, {sylvia.herbert, tomlin}@berkeley.edu

tional work has been completed to combat the computational intensity that limits the ability of a system to update safety guarantees online using HJ reachability analysis. A technique termed *warm-start reachability analysis* is able to produce the same solution as the original method of reachability analysis in less time by *updating* the safety analysis from the previously computed solution, rather than restarting the analysis from scratch [2]. This presents the opportunity to improve on the issues present in the original safe learning framework by using warm-start reachability techniques to shorten computation time.

In this paper we describe a framework for combining safe learning with warm-start reachability and use a simulated quadcopter to demonstrate the improved efficiency in updating the safety analysis in response to unexpected disturbances. Fig. 1 depicts the ability combine the safe learning framework with warm-start reachability to use the original safety analysis (green) to generate an updated safety analysis (blue).

II. Problem Formulation

A. Hamilton-Jacobi Reachability Analysis

Hamilton-Jacobi (HJ) reachability analysis is a formal verification method that ensures systems operate according to pre-specified safety constraints. In reachability analysis, the target set is defined as the region of the state space that the system aims to reach or the region it aims to avoid to ensure the safety of the system. In Fig. 2, the target set is in red and represents safety constraints for a quadcopter (comprised of constraints on altitude and vertical velocity). The goal of HJ reachability analysis is to find the set of states from which the system can reach the target set if the system applies the optimal control under the worst-case disturbance. This provides either the set of states the system must remain within or the set of states the system must avoid to guarantee safety for all possible disturbance values. The control action is bounded within a range of values $u \in \mathcal{U}$, and the optimal control is chosen as the best action within this range to ensure that the system reaches or avoids the target set. Similarly, the disturbance is also bounded within a range of values $d \in \mathcal{D}$, and the worst-case disturbance is the disturbance within this range that acts against the optimal control action most significantly. The dynamics of the system are provided as differential equations that describe how the state of a system changes $\dot{x} = f(x, u, d)$. In the setup described by Fig. 2, the disturbance can be, for example, external wind.

In HJ reachability analysis, the cost function $l(x)$ is defined as a measure of how far the system is from the target set for a given state x. The cost function is negative when the system is within the target set, positive when the system is outside of the target set, and zero when the system is on the boundary of the target set. If the control aims to minimize the cost function, and the disturbance aims to maximize the cost, the value of a state at time t is given by

$$V(x,t) = \min_{u} \max_{d} l(x). \tag{1}$$

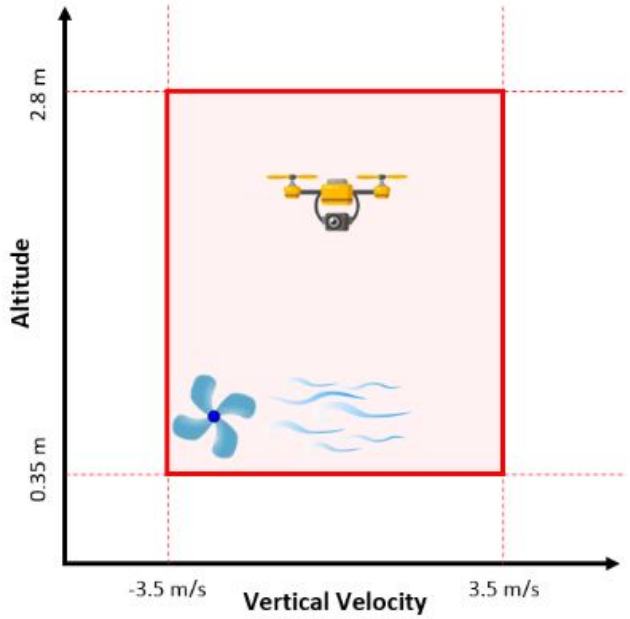

Fig. 2: Scenario of simulated demonstration. The red box represents the target set, where the quadcopter must remain to avoid crashing into the ceiling or floor.

If a state has a negative value, optimal trajectories originating from that state will end within the target set. Therefore, the set of states that have a negative value comprise the safe region in which the system can operate to ensure that it will remain within the target set under the worst-case disturbance. The solution $V(x,t)$ is found by solving the Hamilton-Jacobi-Isaacs variational inequality (HJI VI)

$$\min \Big\{ D_t V(x,t) + H\Big(V(x,t), f(x,u,d)\Big), \; l(x) - V(x,t) \Big\} = 0. \tag{2}$$

where H is the Hamiltonian,

$$H = \min_{u} \max_{d} \langle \nabla V(x(t),t), f(x,u,d) \rangle. \tag{3}$$

For more information on the derivation of the equations used for HJ reachability analysis, see [3–5].

This verification tool is very beneficial because it is robust and can compute strong safety guarantees if the dynamics of the system and the bounds on the disturbance are known. While it is can also be applied to numerous different systems with any number of dimensions, there are limitations on the ability to apply reachability analysis to some real-world systems. First, reachability analysis is computationally intensive, which leads to high computation times, particularly for high-dimensional systems. The safety analysis is also only valid if the assumptions on the dynamics of the system and bounds on the disturbance hold true.

B. Warm-Start Reachability Analysis

One of the challenges of HJ reachability analysis is improved upon through a method termed warm-starting, which lowers computation time when updating the safety analysis

[2]. Assumptions about the dynamics of a system and the bounds on the disturbance acting on the system often change with time. As new knowledge is acquired, it is desirable to update the safety analysis to more accurately reflect the dynamics of the system and the bounds on the disturbance to improve safety guarantees. Rather than restarting the safety analysis as new assumptions are made, warm-start reachability presents a method to initialize reachability computations with the previously computed safe set. Previous work has demonstrated that this technique is effective in producing exact solutions in less time than similar reachability methods that do not utilize warm-starting [2]. Warm-start reachability analysis is a general analysis method that can be applied to a variety of systems to update safety guarantees when the system obtains information about its dynamics or its environment.

C. Safe Learning

Through safe learning, initial assumptions about the disturbance bounds can be updated as a system learns more about its environment [1]. This process begins by performing HJ reachability analysis, using the initial assumptions about the bounds on the disturbance, to compute the safe region of operation. The control of the system is chosen to allow the system to learn about its environment, while remaining within this safe region. If the system learns that the disturbance is not within the bounds it originally assumed, the system becomes more conservative, staying in a smaller region within the calculated safe set where initial assumptions are still valid. The system then continues to gather data about its environment within this confined region of the safe set until it has enough information to compute a model of the disturbance within its environment. It uses GPs to generate a model of disturbance over the state space; the 95th percent confidence intervals on the resulting GP are set as the new bounds on the disturbance. These new assumptions are then used to recompute the safety analysis to better reflect the true disturbances.

This safe learning framework is effective in ensuring the safe operation of a system and recomputing the safety analysis [1]. However, this updating of the safety analysis is time-consuming, and it is often not feasible to perform these computations online. By introducing warm-starting techniques, safety analysis can be updated from its initial assumptions, rather than recomputing safety analysis without these initializations. This has the potential to improve computation time and improve the feasibility of performing these computations online.

III. Efficient Safety Updates in Uncertain Systems

We aim to combine previous safe learning work with warm-start reachability analysis to create a framework for efficient safety updates in uncertain robotic systems. The goal is to efficiently update the safe set as the system acquires information about the disturbance in its environment.

Through our framework, the safe set is initially computed using our assumptions of the bounds on the disturbance. The system then aims to create a model of the disturbance to update this safe set. Until the system has collected enough data to create a model, it explores its environment. The system begins by calculating the disturbance at its given state. If this disturbance is not within the bounds it originally assumed, the system becomes more conservative and contracts its safety region. The system then calculates the relative safety level of its current state to produce a safe control action that attempts to follow a specified trajectory. This control is then used to update the state of the system. Upon collecting enough data about the disturbance, Gaussian Processes are used to update the bounds of the disturbance, which are then used to compute the updated safety analysis. See Algorithm 1 for details on the efficient safety update framework.

Algorithm 1: Efficient safe learning demonstration

```
Result: Updated safe set
Initialize: dynamics, uBounds, dBounds, target,
  trajectory, dSamples;
[SafetyController, safe_set] = HJReachability(dynamics,
  uBounds, dBounds, target);
t = 0;

while length(dSamples) < minimum_samples do
    while t < t_experiment do
        d(x) = MeasureDisturbance(x(t), x(t-1));
        dSamples(end+1) = d(x);

        if d(x) ∉ dBounds then
            safe_set = ContractSet(x, safe_set);
        end

        safety_level = ComputeSafetyLevel(x, safe_set);
        u_perf = PerformanceController(x, dynamics,
          trajectory);
        u_safe = SafetyController(x, dynamics);
        u = VerifierController(u_perf, u_safe,
          safety_level);
        x = UpdateState(x, dynamics, u, t);
    end

    dBounds = GaussianProcessUpdate(dSamples);
    target = safe_set;
    [SafetyController, safe_set] =
      HJReachability(dynamics, uBounds, dBounds,
      target);
end
```

IV. Demonstration

To test the performance of our efficient safety updates, we are simulating a scenario in which a quadcopter moves vertically within a room, learning about its environment while

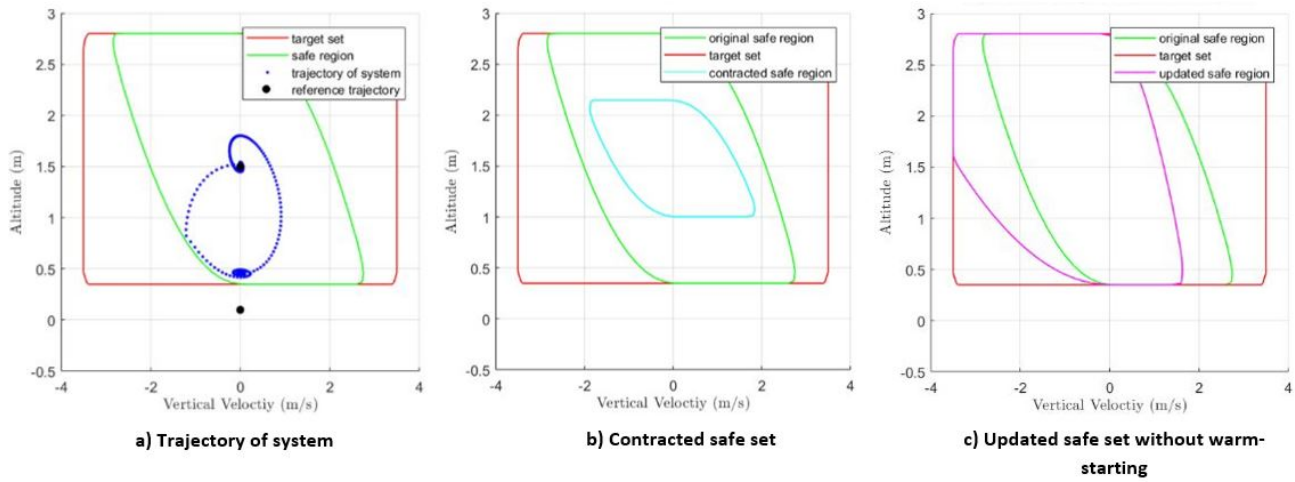

Fig. 3: (a) The trajectory of the system uses a combination of the safety and performance controller. The performance controller aims to move between the points in the reference trajectory, and the safety controller overrides this control when the system is at risk of leaving the safety region. (b) The safe region contracts when the system observes a disturbance that is stronger than expected, limiting the space in which the system can safely move. (c) Original safe region computed using assumed disturbance and updated safe region computed using model of disturbance without warm-starting.

remaining safe. The goal is for the quadcopter to learn how to move around within the space and avoid crashing into the floor or ceiling, despite the fact that the system is not fully aware of the disturbance in the environment. This scenario can be seen in Fig. 2.

The quadcopter operates under the following dynamics:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ k_T * u + g + k_0 + d \end{bmatrix} \tag{4}$$

with altitude x_1 and vertical velocity x_2, where k_T, g, and k_0 are constants. The control $u \in [5, 8]$ is the motor thrust command, and the disturbance is initially assumed to be bounded such that $d \in [-2, 2]$. In order to avoid crashing into the floor or ceiling, the center of the quadcopter must remain between an altitude of $0.35m$ and $2.8m$, while its vertical velocity must remain between $-3.5m/s$ and $3.5m/s$. This is the target set, which can be visualized by the red box in Fig. 2.

While we are initially assuming that the disturbance is within the range $d \in [-2, 2]$, the actual disturbance present in the environment is different than our assumptions. We are considering a case in which there is a fan on the ground that produces a disturbance of which the quadcopter is initially unaware. This disturbance is modeled by the function

$$d(x_1) = \frac{3}{1 + e^{2x_1 - 4}} \tag{5}$$

which produces a disturbance that is stronger than we had anticipated near the ground and approaches zero as altitude increases as shown in Fig. 4. Intuitively, we would expect that increased wind speeds at lower altitudes would cause this region of the state space to be less safe, whereas the decreased wind speeds at higher altitudes would cause this region to be more safe.

Before the quadcopter can begin its trajectory, we compute the safe region (Fig. 3a), given the assumed disturbance with no knowledge of the actual disturbance. We then create

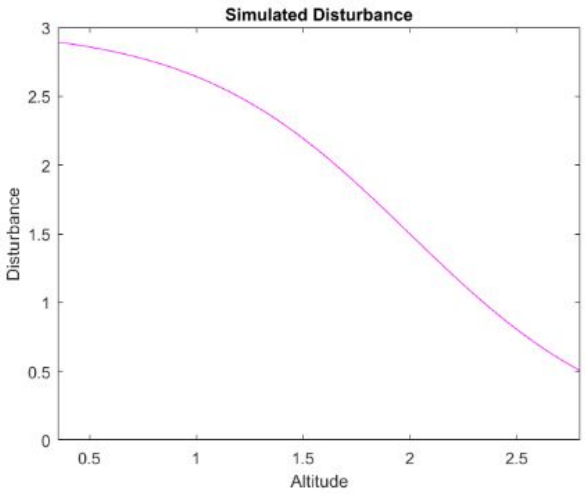

Fig. 4: Model of actual disturbance for simulation, where disturbance is high for low altitudes and approaches zero for higher altitudes.

a performance controller that computes the linear-quadratic regulator (LQR) control to enable the system to follow a given trajectory, and a safety controller that computes the optimal control to remain within the safe region. These controllers are averaged based on the systems relative safety level to create a controller that enables the system to follow the given trajectory with minimal resistance, until the system comes close to the boundary of the safe region. The trajectory of the system is shown in Fig. 3a.

As the system moves within the state space, it calculates the disturbance at various altitudes. If the disturbance is higher in magnitude than the maximum or minimum disturbance that it originally assumed, the system becomes more conservative, immediately contracting the safe region (Fig. 3b). By using the systems current state to recompute the boundary of the safe region, we ensure that the system remains safe as it continues to learn about its environment.

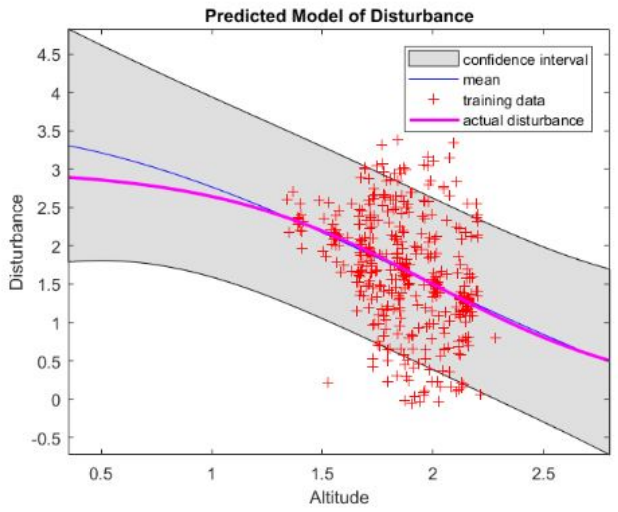

Fig. 5: Model of disturbance and actual disturbance for various altitudes. Model provides the range of disturbances in which we assume the actual disturbance resides with 95% confidence.

Once the system has computed a sufficient number of disturbance measurements for various altitudes, Gaussian Processes (GPs) are used to generate a model of the disturbance for every altitude within the target set. For every altitude within the target set, this model provides a prediction of the disturbance and the range of disturbances that the actual disturbance resides in with 95% confidence. This model is shown in Fig. 5.

Using this model of disturbance, the safe region can be recomputed, using the bounds of the confidence interval as the new assumptions of the maximum and minimum disturbance. These computations have been performed both with and without warm-starting. The updated safe region that used warm-starting techniques to update it from the previously computed safe region (Fig. 1) is nearly identical to the updated region that was computed by restarting the computations without warm-starting (Fig. 3c). The time it took to complete safety analysis without warm-starting was 5.169 seconds, while the time required to complete safety analysis without warm-starting was 4.758 seconds.

V. Discussion & Conclusion

This work creates an efficient method for a system to learn about its environment and update safety guarantees by combining the previous safe learning framework with warm-start reachability analysis. This presents the potential to improve computation time when updating safety analysis, which would allow computations to be performed online in real-time. For a system with only two dimensions, the difference in computation time with and without warm-starting is evident but minimal. However, this safety framework is applicable to high-dimensional systems, which benefit much more from warm-starting [2]. Previously, the safe-learning framework could not effectively be applied to systems with a large number of dimensions due to the high computation time. Combining the safe learning framework with warm-start reachability would allow high-dimensional systems to benefit from improved methods of safe learning in uncertain robotic systems.

Future work aims to demonstrate this improved safe learning framework on real-world systems with the goal of applying this work to more realistic, high-dimensional systems. Additionally, we aim to implement a performance controller that uses reinforcement learning, rather than an LQR controller, to improve the system's ability to travel in and learn about its environment. We also aim to perform reachability analysis online, performing safety analysis and collecting disturbance data in parallel to allow for real-time updates to the safe set.

References

[1] Jaime F Fisac, Anayo K Akametalu, Melanie N Zeilinger, Shahab Kaynama, Jeremy Gillula, and Claire J Tomlin. A general safety framework for learning-based control in uncertain robotic systems. *IEEE Transactions on Automatic Control (TAC)*, 2018.

[2] Sylvia L Herbert, Shromona Ghosh, Somil Bansal, and Claire J Tomlin. Reachability-based safety guarantees using efficient initializations. *IEEE Conference on Decision and Control (CDC)*, 2019.

[3] Somil Bansal, Mo Chen, Sylvia Herbert, and Claire J Tomlin. Hamilton-Jacobi reachability: A brief overview and recent advances. In *Conf. on Decision and Control*. IEEE, 2017.

[4] Ian M. Mitchell, A. M. Bayen, and C. J. Tomlin. A time-dependent Hamilton-Jacobi formulation of reachable sets for continuous dynamic games. *IEEE Transactions on Automatic Control*, 2005.

[5] Jaime F Fisac, Mo Chen, Claire J Tomlin, and S Shankar Sastry. Reach-avoid problems with time-varying dynamics, targets and constraints. In *Int. Conf. on hybrid systems: computation and control*. ACM, 2015.