A Teleoperation Framework for Robots Utilizing Control Barrier Functions in Virtual Reality

Aref Hebri
The University of Texas at Arlington
Arlington, Texas, USA

Sneh Acharya
The University of Texas at Arlington
Arlington, Texas, USA

Michail Theofanidis
The University of Texas at Arlington
Arlington, Texas, USA

Fillia Makedon
The University of Texas at Arlington
Arlington, Texas, USA

ABSTRACT

This paper describes a novel shared control teleoperation framework for mobile robots that utilizes Control Barrier Functions (CBFs) as filtering mechanism to prevent a human operator from making dangerous actions. The proposed framework demonstrates the potential to create a CBF controller that enables users with no prior knowledge of robotics to safely tele-navigate mobile robots with limited situational awareness. As formal methods, we utilize a hand-crafted CBF, which acts as a repulsive field to describe unsafe regions withing the robot's vicinity. The implementation of the application was deemed possible by creating a Virtual Reality (VR) simulation in the Unity Engine with the SUMMIT-XL STEEL mobile base as an experimental platform. Preliminary experimental results show the ability of the framework to enable safe teleoperation.

CCS CONCEPTS

• **Human-centered computing** → **Virtual reality**; • **Computer systems organization** → **Robotic autonomy**.

KEYWORDS

virtual reality, simulation, teleoperation, robots, control barrier functions, shared control

ACM Reference Format:
Aref Hebri, Sneh Acharya, Michail Theofanidis, and Fillia Makedon. 2023. A Teleoperation Framework for Robots Utilizing Control Barrier Functions in Virtual Reality. In *Proceedings of the 16th International Conference on PErvasive Technologies Related to Assistive Environments (PETRA '23), July 05–07, 2023, Corfu, Greece.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3594806.3596522

1 INTRODUCTION

The coexistence and collaboration of humans and robots have been the aspiration of many scientific endeavors during the past century. Research on the topic of Human-Robot Interaction (HRI) has created a plethora of different human-robot collaboration frameworks, which utilize different mathematical paradigms to model the interactions of a human-robot team. Our work is concerned with robot teleoperation, in which a human operator controls a robot remotely in real-time. In traditional teleoperation systems, the human operator has full control over all of the robot's actions. Direct control is ineffective in applications where the user must control the robot remotely, in real-time, but cannot attain sufficient and reliable information about the robot's environment. For instance, this may happen if the user can only access information through the robot's sensors (typically a front-facing camera), leading to ample blind spots and collisions with obstacles.

A common way to reduce the mismatch between the capabilities of the user and the capabilities of the robot is to enhance the robotic system with an autonomy controller, which imposes and mediates the communication between the user and the robot. In shared control scenarios, the controller can assist the user by modifying the input commands from the user to the robot in order to prevent failures (e.g., collisions), therefore assisting in teleoperation [1], [6]. Considering this, we aim to propose an autonomy controller that prevents the robot from taking unsafe actions, by filtering the operator's commands to the robot through a correcting force which is generated by a CBF [4], [2], that the shared controller utilizes to describe undesired locations. As such, in the context of this application, CBFs can be thought of as repulsive fields [16] that prevent the robot from visiting undesirable regions, such as an obstacle, a human, or another robot. As the operator takes control of the mission, the CBF controller evaluates the robot's environment and encapsulates unsafe regions with a CBF, which projects the correcting force that prevents the robot from reaching the unsafe region. To facilitate the effectiveness of the proposed CBF controller we created a virtual environment with the Unity game engine to simulate the interaction between a human operator and a virtual mobile robot as it traverses the environment.

2 RELATED WORK

A considerable amount of research has been conducted in the field of human-robot interaction to create user-friendly interfaces for robot teleoperation. Whereas for industrial, exploration or medical applications, the field of HRI utilizes different technologies to assist robot operators. The variety of the techniques depend on the level of autonomy of the robot and the task it must accomplish. These techniques range from direct physical interaction with the robot [9] to computer aided graphical interfaces (GUI) [22],[8]. Shared control frameworks have proven to be promising in helping human operators to remotely control a robot. Traditionally, these

---

PETRA '23, July 05–07, 2023, Corfu, Greece

ACM ISBN 979-8-4007-0069-9/23/07.
https://doi.org/10.1145/3594806.3596522

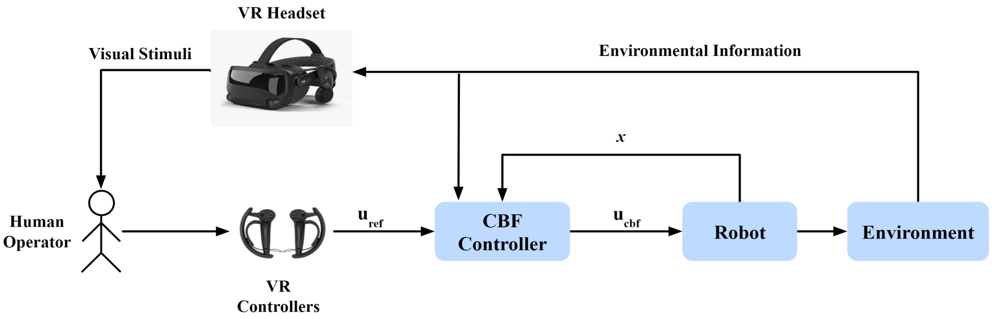

Figure 1: Proposed Framework Architecture

frameworks [14],[15] break the link between the human's control command and the robot's action. The reduced transparency of the internal control loop can reduce user satisfaction but it enables safety and stability for the robotic system.

Recent literature has been shown that CBFs are formulated in a variety of different ways depending on the type of the robot and the task it must accomplish. CBFs can take polynomial [2], [21] or exponential [2], [13] form. Experiments that perform obstacle avoidance [23], suggest that CBFs can take the form of potential fields [16] or any function that relates the position of the obstacle and the robot, such as the signed distance function [17]. Moreover, CBFs have been applied to safely cruise mobile robots [4], enable flying robots [19], [20] and robotic swarm to avoid obstacles [18], [5] and support bibedal robots to walk on uneven surface [12]. Very recently, some studies have used RL in combination with CBFs to model imperfections [7] or learn the boundaries of safe spaces [10], [11]. In our work we propose a shared control framework that at it's epicenter utilizes a CBF controller that provides autonomy in the context of collision and obstacle avoidance.

3 PRELIMINARIES

In this section, we will summarize the CBF framework as described in [2]. Assuming that the robot can be modeled as an input-affine dynamics of the form:

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

where x is the state (e.g., position, velocity) of the robot, u is the control input to the robot, and both $f : D \to \mathbb{R}^n$, $u \in U^m$ and $g : D \to \mathbb{R}^{n \times m}$ are locally Lipschitz continuous vector fields. The autonomy controller keeps the robot state inside a safe space $C = \{x \in D \mid h(x) \geq 0 \;\; and \;\; h \in C(D;R)\}$, which is defined as a set of robot states by the super zero level-set of function h, with the boundary of the safe set as the zero level-set $\partial C = \{x \in D \mid h(x) = 0\}$. During the operation of the autonomy controller the robotic system 1 is considered safe if for all $t \geq 0, x(t) \in C$ when $x(0) \in C$ that is defined by $h(x)$. As detailed in [2], [3], $h(x)$ is a CBF function and it can be used to guarantee that the robotic system will not visit any unsafe states outside of the C, given that $h(x)$ is bound by an extended class K function δ which is strictly increasing and $\delta(0) = 0$ as $\delta(\dot{h}(x)) \geq 0$. We consider the set K_{cbf} consisting of all control values that render C safe as

$$K_{cbf}(x) = \left\{ u \in U \mid L_f h(x) + L_g h(x) + \delta(h(x)) \geq 0 \right\} \tag{2}$$

This implies that the safety of the system can then be guaranteed under the action of a suitable control input $u(x) \in u_{cbf}(x)$ for all $x \in D$. Then given a reference control u_{ref} (e.g., from the user), the basic CBF formulation computes a safe control u_{cbf} according to:

$$u_{\text{cbf}} = \arg\min_{u} ||u - u_{\text{ref}}||^2 \tag{3}$$

which keeps the robot in $C = \{x \in D \mid h(x, \theta) \geq 0\}$. Equation 3 can be solved as a quadratic program (QP) [2].

4 SYSTEM ARCHITECTURE

Fig. 1 provides an overview of the proposed framework. The system is composed by an interface which includes a VR controller and a VR headset, the CBF controller which mediates the communication between the user and the robot, and the robotic platform. Initially, the human operator interacts with the VR controller, which transforms the user's intentions into a low-level reference control signal u_{ref} that accelerates the robot. The CBF controller filters u_{ref} by interposing a correcting force u_{cbf} which is generated by a CBF that describes the location of undesirable states. Specifically, the CBF controller mediates the communication by receiving the control signal of the user and issues a correcting control signal u_{cbf} that prevents the robot from visiting undesirable states, such as obstacles. Thus, if the operator issues an unsafe control command, the CBF controller will help guide the operator towards the closest safe input command. The CBF controller is aware about the state of the environment, such as the potion of the robot and the position of obstacles. In this paper the environment is a 3D simulated in-doors setting, which can also be observed by the human operator through a VR headset.

4.1 CBF Controller

In the context of this application, CBFs act as repulsive fields that describe robot states which should be avoided. As mentioned in the related work section, there are many CBFs that can achieve such

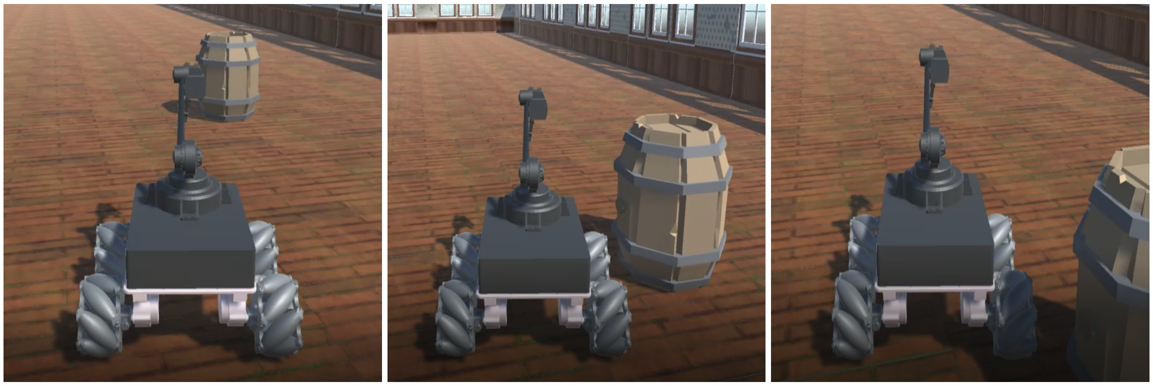
(a) (b) (c)

Figure 2: Simulated renders of the result (a) The robot is approaching the obstacle (b) CBF force is applied to the robot (c) After the obstacle is avoided, the robot follows the command of the human

results such as potential fields [16]. Throughout the experimental section we utilized the following CBF function:

$$h(x) = (x - x_{\text{obs}})^2 - r_{\text{obs}}^2 + 2(x - x_{\text{obs}})\dot{x} \tag{4}$$

where x is the position of the robot, x_{obs} is the position of the obstacle, r_{obs} and r_{obs} is the radius of the obstacles. Equation 4 describes the distance of the robot to the obstacle but also takes into consideration the speed that robot approaches the obstacle. The controller issues the safe command u_{cbf} though equation 3 by knowing the full state of the robot and the environment and the state dynamics of the robot.

4.2 Robot

The state dynamics of the robot where implemented as

$$\dot{x} = A * x - b * x + g * u, \tag{5}$$

where x is the state of the robot, and is expressed in matrix form as $x = [p_{\text{x}}, v_{\text{x}}, p_{\text{y}}, v_{\text{y}}]$ that concatenates the position p and velocity v of the robot. b is a friction coefficient that we utilize to make the simulation of the robot in VR more realistic and is selected as 0.2 for the purposes of this study, and A and g are selected so that the system behaves as a two dimensional double integrator.

5 EXPERIMENTAL SETUP AND RESULT

5.1 Setup

The virtual environment, the CBF controller and the robot were simulated using Unity game engine. The simulation code was implemented using C#. We used a 3D model of SUMMIT-XL STEEL mobile robot with Panda Arm attached on top of it. The robot has omnidirectional kinematics based on 4 high-power drive wheels. The simulation is modeled after an in-door environment and includes a single spherical obstacle. Fig 2 shows the simulated robot, obstacle and the environment.

To fully immerse the human operator in the environment and to help with the teleoperation of the robot, a virtual reality setup was used. In the VR setup, we utilized the Valve Index headset and controllers. The operator will see the simulation through the Valve headset in a VR setting and uses the joystick on this controller to control the direction and acceleration of the robot to navigate the environment. This controller is also capable of generating haptic feedback which we will use in our future setups. Fig 3 depicts the VR setup.

The goal of this setup is for the human operator to drive the robot toward the obstacle and for the CBF controller to issue a command that will guarantee the safety of the robot. This command will be in the form a control signal u_{cbf} which will be different from the user input if there is any danger in the path of the robot and it will be the same as the user input if there is no danger in the path of the robot. If there is any undesirable region within the robot's vicinity, this safe command will drive the robot away from the obstacle thus avoiding collision. It should be noted that when the robot is near an obstacle, the CBF controller will only generate a different signal than the operator when the robot is actively approaching the obstacle with an unsafe velocity and acceleration.

5.2 Result

Fig 2 and Fig 4 illustrate the result of the experiment. Specifically, Fig 2 shows screenshots of the robot as it is driven towards the obstacle by a human operator in VR. When the robot gets to a certain distance from the object, the CBF will apply a force u_{cbf} that will be different from the input of the operator u_{ref}in order to avoid the obstacle. The robot then changes its path and avoids colliding with the object. After passing the obstacle, since there is no other danger, the output of the CBF controller will be the same as the input of the human operator.

Fig 4 shows a plot of the position of the robot and the obstacle in the x-y axis. As can be seen in the graph,the robot starts moving towards the obstacle and at a certain point, the robot changes trajectory to avoid the obstacle. Without the CBF controller, the robot would have collided with the obstacle. After the obstacle is passed, the robot continues on its intended path. This result indicates that

Figure 3: The human operator uses a VR headset and controllers to control the robot in a simuulated in-doors environment.

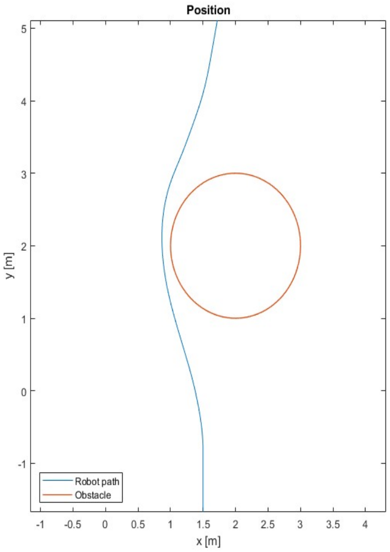

Figure 4: Plot of robot and obstacle position

CBF-based teleoperation can help the operator navigate the environment safely. Although more investigation is needed, this result is a good staring point for further studies and experiments.

6 FUTURE WORK

In this paper, we presented a novel shared control teleoperation framework for mobile robots utilizing CBFs to generate a safe signal for a human operators to prevent dangerous actions. The result shows a promising approach that can improve a human operator's ability to safely teleoperate a robot.

Our next steps will be to add a haptic feedback component to our system. Instead of directly applying the control signal generated by CBF to the robot, we will also apply the difference between the command issued by the human operator and the safe command returned by the CBF as a haptic feedback to the operator through the VR Controller. Other future investigating avenues include the implementation of a CBF controller with more complex CBF functions that enable obstacle avoidance in more complex and unstructured virtual environments.

Apart from that, we want to conduct a user study to evaluate the effects of haptic feedback generated through CBFs on the performance of the human operator while navigating a robot and using the panda arm to pick up and drop off objects in a simulated 3D environment.

REFERENCES

[1] Jake J Abbott, Panadda Marayong, and Allison M Okamura. 2007. Haptic virtual fixtures for robot-assisted manipulation. In *Robotics research*. Springer, 49–64.

[2] Aaron D Ames, Samuel Coogan, Magnus Egerstedt, Gennaro Notomista, Koushil Sreenath, and Paulo Tabuada. 2019. Control barrier functions: Theory and applications. In *2019 18th European control conference (ECC)*. IEEE, 3420–3431.

[3] Aaron D Ames, Kevin Galloway, Koushil Sreenath, and Jessy W Grizzle. 2014. Rapidly exponentially stabilizing control lyapunov functions and hybrid zero dynamics. *IEEE Trans. Automat. Control* 59, 4 (2014), 876–891.

[4] Aaron D Ames, Jessy W Grizzle, and Paulo Tabuada. 2014. Control barrier function based quadratic programs with application to adaptive cruise control. In *53rd IEEE Conference on Decision and Control*. IEEE, 6271–6278.

[5] Urs Borrmann, Li Wang, Aaron D Ames, and Magnus Egerstedt. 2015. Control barrier certificates for safe swarm behavior. *IFAC-PapersOnLine* 48, 27 (2015), 68–73.

[6] Stuart A Bowyer, Brian L Davies, and Ferdinando Rodriguez y Baena. 2013. Active constraints/virtual fixtures: A survey. *IEEE Transactions on Robotics* 30, 1 (2013), 138–157.

[7] Jason Choi, Fernando Castaneda, Claire J Tomlin, and Koushil Sreenath. 2020. Reinforcement learning for safety-critical control under model uncertainty, using control lyapunov functions and control barrier functions. *arXiv preprint arXiv:2004.07584* (2020).

[8] Raúl Crespo, René García, and Samuel Quiroz. 2015. Virtual reality simulator for robotics learning. In *Interactive Collaborative and Blended Learning (ICBL), 2015 International Conference on*. IEEE, 61–65.

[9] Scott Devine, Karen Rafferty, and Stuart Ferguson. 2016. Real time robotic arm control using hand gestures with multiple end effectors. In *Control (CONTROL), 2016 UKACC 11th International Conference on*. IEEE, 1–5.

[10] Haitong Ma, Jianyu Chen, Shengbo Eben Li, Ziyu Lin, Yang Guan, Yangang Ren, and Sifa Zheng. 2021. Model-based Constrained Reinforcement Learning using Generalized Control Barrier Function. *arXiv preprint arXiv:2103.01556* (2021).

[11] Zahra Marvi and Bahare Kiumarsi. 2020. Safe reinforcement learning: A control barrier function optimization approach. *International Journal of Robust and Nonlinear Control* (2020).

[12] Quan Nguyen, Ayonga Hereid, Jessy W Grizzle, Aaron D Ames, and Koushil Sreenath. 2016. 3d dynamic walking on stepping stones with control barrier functions. In *2016 IEEE 55th Conference on Decision and Control (CDC)*. IEEE, 827–834.

[13] Quan Nguyen and Koushil Sreenath. 2016. Exponential control barrier functions for enforcing high relative-degree safety-critical constraints. In *2016 American Control Conference (ACC)*. IEEE, 322–328.

[14] Allison M Okamura. 2009. Haptic feedback in robot-assisted minimally invasive surgery. *Current opinion in urology* 19, 1 (2009), 102.

[15] Claudio Pacchierotti, Leonardo Meli, Francesco Chinello, Monica Malvezzi, and Domenico Prattichizzo. 2015. Cutaneous haptic feedback to ensure the stability of robotic teleoperation systems. *The International Journal of Robotics Research* 34, 14 (2015), 1773–1787.

[16] Andrew Singletary, Karl Klingebiel, Joseph Bourne, Andrew Browning, Phil Tokumaru, and Aaron Ames. 2020. Comparative analysis of control barrier functions and artificial potential fields for obstacle avoidance. *arXiv preprint arXiv:2010.09819* (2020).

[17] Mohit Srinivasan, Amogh Dabholkar, Samuel Coogan, and Patricio Vela. 2020. Synthesis of control barrier functions using a supervised machine learning approach. *arXiv preprint arXiv:2003.04950* (2020).

[18] Li Wang, Aaron Ames, and Magnus Egerstedt. 2016. Safety barrier certificates for heterogeneous multi-robot systems. In *2016 American Control Conference (ACC)*. IEEE, 5213–5218.

[19] Li Wang, Aaron D Ames, and Magnus Egerstedt. 2017. Safe certificate-based maneuvers for teams of quadrotors using differential flatness. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 3293–3298.

[20] Guofan Wu and Koushil Sreenath. 2016. Safety-critical control of a planar quadrotor. In *2016 American Control Conference (ACC)*. IEEE, 2252–2258.

[21] Xiangru Xu, Paulo Tabuada, Jessy W Grizzle, and Aaron D Ames. 2015. Robustness of control barrier functions for safety critical control. *IFAC-PapersOnLine* 48, 27 (2015), 54–61.

[22] Basem Fayez Yousef. 2012. A Robot for Surgery: Design, Control and Testing. In *Advances in Robotics and Virtual Reality*. Springer, 33–59.

[23] Dawei Zhang, Roberto Tron, and Rebecca P Khurshid. 2021. Haptic Feedback Improves Human-Robot Agreement and User Satisfaction in Shared-Autonomy Teleoperation. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 3306–3312.