“Provably Safe” in the Wild: Testing Control Barrier Functions on a Vision-Based Quadrotor in an Outdoor Environment

Cherie Ho*, Katherine Shih*, Jaskaran Grover, Changliu Liu, Sebastian Scherer
Email: {cherieh, kshih, jaskarag, cliu6, basti}@cs.cmu.edu

I. Introduction

Quadrotors are increasingly deployed into safety-critical applications, such as drone delivery [3], security [5], and even cinematography [2]. However, maintaining safety is challenging in these applications due to unknown, varying disturbances. In addition, it’s often advantageous to replace conventional sensors (ex. GPS, LIDAR) with cheaper and smaller visual sensors. However, visual sensors are vulnerable to lighting changes and dynamic objects [12]. We must develop systems that can reason around such disturbances and uncertainties to guarantee real-time safety.

Safe control methods guarantee system safety, given that assumptions made during problem setup (e.g., bounded disturbances and bounded state uncertainties) hold. In particular, exponential control barrier functions (ECBFs) enforce high relative degree safety by guaranteeing forward invariance when the system is already in a safe set [1]. While this technique has shown success in safely controlling quadrotors in simulation [14] and in heavily-controlled lab settings [15], there has been limited experimentation in the real world. We extend the work from [15], which uses ECBFs for safe teleoperation of quadrotors. To the best of the authors’ knowledge, this work presents the first implementation of ECBFs on a quadrotor with fully onboard computing and state estimation. With this platform, we evaluate the robustness of ECBFs with actual flight tests to identify key roadblocks in deploying such guaranteed safe methods in the real world. Our main contributions are:

- implementation of ECBFs with fully onboard computing and vision-based state estimation for avoiding virtual obstacles, and
- extensive outdoor field experiments and analysis of key failure points when deploying ECBFs in the real world.

Video of Flight Tests: https://youtu.be/1ohaMHlCmDA
Code: https://github.com/hocherie/cbf_quadrotor

II. Problem Setup & Control Design

Overall Objective: Our objective is to maintain safety while minimizing the intervention of a safe controller. We define intervention as the difference between the applied (safe) control inputs $\hat{\mathbf{u}}$ and nominal control inputs $\mathbf{u}$, $||\hat{\mathbf{u}}-\mathbf{u}||^2$, which becomes a cost function to minimize over given safe control constraints (remaining in safe set).

Exponential Control Barrier Functions: Let $\mathcal{C}_t$ be the safe set of states for a given dynamical system, and define the scalar function $h(t,x)$ so that $\mathcal{C}_t = \{x \in \mathbb{R}^n | h(x,t) \geq 0\}$ for all $t > 0$. $h(t,x)$ is an ECBF for a system of relative degree 2 if the following conditions apply: a) $h(t,x)$ is continuously differentiable; b) a gain matrix $\mathbf{K} \in \mathbb{R}^2$ exists such that the poles of the system $\ddot{h} + \mathbf{K} \cdot [h(t,x)\ \dot{h}(t,x)]$ are on the negative real line; and c) a control input $\mathbf{u}$ exists such that

$$\ddot{h}(t,x,u) + \mathbf{K} \cdot [h(t,x)\ \dot{h}(t,x)]^T \geq 0, \forall x \in \mathcal{C}_t \tag{1}$$

Assuming these conditions hold and initial conditions are such that $h(0) > 0$, the set of possible inputs $\mathbf{u}$ are guaranteed to maintain the safety of the system (that is, positive $h(t,x)$).

Control Design: Fig. 2 describes our method. The nominal trajectory outputs a desired control input capped at $1 m/s^2$; a safe controller then calculates the closest input (acceleration) that satisfies safety constraints defined by ECBF sets. Let $\mathbf{r} \in \mathbb{R}^2$ be the position vector from robot to obstacle in the x-y plane, and let $\mathbf{u}$ directly control $\ddot{r}$. Let $\mathbf{K} = [k_1\ k_2]^T$. The

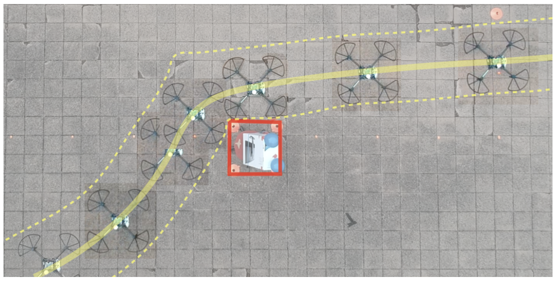


Fig. 1: Timelapse of quadrotor with Exponential Control Barrier Functions (ECBFs) enabled to avoid a virtual obstacle. The quadrotor follows a nominal trajectory toward goal point (orange cone at top right) that would have passed through obstacle without safe control. We use this setup to identify key failure cases of deploying ECBFs in the wild.

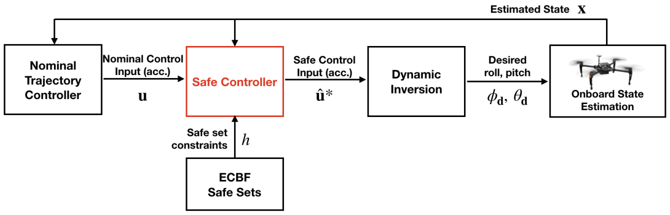


Fig. 2: Overall block diagram. Given a nominal input, the proposed safe controller computes a safe control input that satisfies constraints from a predefined ECBF set. The final control input $\hat{\mathbf{u}}^*$ is then converted from acceleration to desired roll and pitch angles for the quadrotor.

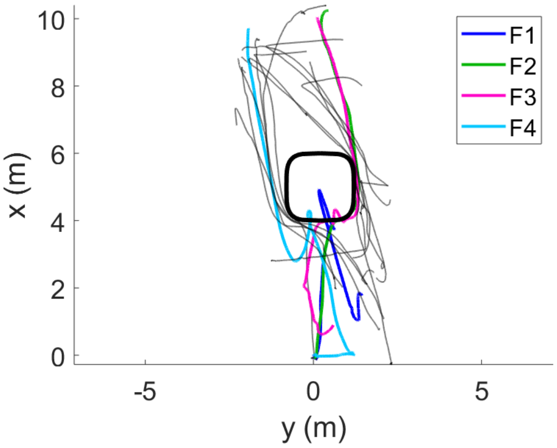


Fig. 3: Trajectories of all 16 flight tests with safe control enabled. Out of all trials, 4 violated safety.

Failure Case	**Notes**
F1	High wind gust. Large abrupt state estimation jump from flying cone, safe control outputted $2000m/s^2$ away from obstacle. **Manual takeover.**
F2	Slight brush against obstacle (max penetration: 4cm).
F3	Small state estimation jump from flying cone, safe control outputted $2m/s^2$ away from obstacle.
F4	Small state estimation jump into obstacle. Able to recover.

TABLE I: Failed trials during real-life testing, with associated color in Fig. 3. The leading causes of failure are: sudden state estimation error, wind/gusts and response delay in roll and pitch angle, further analyzed in discussion.

final safe control $\hat{\mathbf{u}}^*$ is from

$$\hat{\mathbf{u}}^* = \arg\min_{\mathbf{u}_i} ||\hat{\mathbf{u}} - \mathbf{u}||^2; \;\; s.t. \;\; \ddot{h} + \mathbf{K} \cdot [h \;\dot{h}]^T \geq 0 \qquad (2)$$

where $\ddot{h}$ is affine in $\mathbf{u}$. Similarly to [15], we choose a superellipsoid to represent obstacles in our safety index h, with a safety buffer based on size of the quadrotor and tracking error. Details on safe set formulation can be found in the appendix.

III. Experimental Results

1) Experimental Setup: Field tests were conducted on a DJI M100 quadrotor with an Intel T265 for state estimation. All processing is done with an Intel i7 NUC. A virtual obstacle is given, with cones and physical obstacles for easy visualization. Final desired roll and pitch angle to quadrotor is capped at 30^o.

2) Results: We conducted 16 trials with multiple start and goal points with a virtual obstacle to test the robustness of our ECBF implementation. Fig. 3 overlays all trajectories, with highlighted paths for trials that violated safety. Table I details these failure cases. Detailed results of failure cases can be found in the appendix.

IV. Discussion and Potential Solutions

We identified three key reasons for ECBF failures in the real world from our flight tests, and propose potential solutions and recent works that may address these.

Sudden State Estimation Error: Visual Inertial Odometry (VIO) is especially vulnerable to dynamically-changing environments. In several trials with safety violations, state estimation jumps abruptly into the obstacle when objects within the view changed (e.g. cones flying away from quadrotor turbulence). In F1, the safety index h became negative and the drone abruptly commanded a high acceleration away from the obstacle, necessitating a manual takeover to keep operators safe. Flight tests were run under conditions normally ideal for VIO. In conditions with higher lighting variation, such as a partly cloudy day where the sun may be briefly occluded, the errors may be much worse, compounding the problem.

To retain safety in real-world conditions, a potential method is including state estimation uncertainty (i.e., covariance) in a stochastic CBF framework [7, 4]. The proposed change can be robustly tested in TartanAir [12], a photo-realistic dataset that contains challenging light conditions, weather and moving object to stress test vision-based state estimation algorithm.

Exogenous Disturbance - Wind/Gust: During real world testing, the system was subjected to varying wind conditions (up to 10km/h wind). As the model assumed no exogenous disturbance, the computed acceleration is only dependent on the current robot position and velocity. Therefore, the computed control is not able to counteract the wind. One can account for estimated exogenous acceleration in the CBF framework, such as with an L1 Adaptive Controller as used in [8] for Control Lyapunov Functions. Alternatively, provably safe planning methods such as FasTrack [6] guarantee safety, using worst-case scenarios from bounded disturbances as safety buffers.

Response delay in roll, pitch: We assume the system response is affine to the commanded acceleration. However, in reality, there is a delay in the order of $0.1s$ in the acceleration response to the drone. One could use a discrete formulation for the safe control calculation that also includes the maximum expected delay, or as in [9] accounting for an estimated delay.

V. Conclusion

We tested the robustness of Exponential Control Barrier Functions (ECBFs) with outdoor flight tests on a quadrotor with fully onboard computing and state estimation. Field testing showed that a safe controller that performs well under the assumed model (perfect state estimation, no time delay, and little to no disturbances) will fail when these assumptions are violated. A number of recent works are aimed towards relaxing these assumptions for CBFs, such as by incorporating stochasticity [7, 4], accounting for disturbances in an adaptive framework [10], or using a discrete formulation to account for an estimated delay [9]. As robots are increasingly deployed into more challenging situations, there is a growing need for common benchmarking frameworks, such as the one presented in [13], that replicate real life scenarios to robustly test safe control algorithms.

Acknowledgments

This work was supported by the Croucher Foundation.

References

[1] 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)*, pages 3420–3431, June 2019. doi: 10.23919/ECC.2019.8796030.

[2] R. Bonatti, W. Wang, C. Ho, A. Ahuja, M. Gschwindt, E. Camci, E. Kayacan, S. Choudhury, and S. Scherer. Autonomous aerial cinematography in unstructured environments with learned artistic decision-making. *Journal of Field Robotics*, 2019.

[3] Shushman Choudhury, Kiril Solovey, Mykel J. Kochenderfer, and Marco Pavone. Efficient Large-Scale Multi-Drone Delivery Using Transit Networks. *arXiv:1909.11840 [cs]*, April 2020. URL http://arxiv.org/abs/1909.11840. arXiv: 1909.11840.

[4] A. Clark. Control barrier functions for complete and incomplete information stochastic systems. In *2019 American Control Conference (ACC)*, 2019.

[5] M. De-Miguel-Molina. *Ethics and Civil Drones: European Policies and Proposals for the Industry*. Springer, 2018.

[6] Sylvia L. Herbert, Mo Chen, SooJean Han, Somil Bansal, Jaime F. Fisac, and Claire J. Tomlin. Fastrack: a modular framework for fast and guaranteed safe motion planning. *CoRR*, abs/1703.07373, 2017.

[7] Wenhao Luo and Ashish Kapoor. Airborne collision avoidance systems with probabilistic safety barrier certificates, December 2019.

[8] Quan Nguyen and Koushil Sreenath. L1 adaptive control for bipedal robots with control lyapunov function based quadratic programs. In *American Control Conference (ACC)*, pages 862–867, Chicago, IL, July 2015.

[9] Andrew Singletary, Yuxiao Chen, and Aaron D. Ames. Control Barrier Functions for Sampled-Data Systems with Input Delays. 05 2020.

[10] Andrew Taylor and Aaron Ames. Adaptive safety with control barrier functions, 10 2019.

[11] L. Wang, A. D. Ames, and M. Egerstedt. Safe certificate-based maneuvers for teams of quadrotors using differential flatness. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, May 2017.

[12] Wenshan Wang, Delong Zhu, Xiangwei Wang, Yaoyu Hu, Yuheng Qiu, Chen Wang, Yafei Hu, Ashish Kapoor, and Sebastian Scherer. Tartanair: A dataset to push the limits of visual slam. *arXiv preprint arXiv:2003.14338*, 2020.

[13] Tianhao Wei and Changliu Liu. Safe control algorithms using energy functions: A unified framework, benchmark, and new directions. In *Proceedings of the IEEE Conference on Decision and Control (CDC)*, pages 238–243. IEEE, 2019.

[14] G. Wu and K. Sreenath. Safety-critical control of a planar quadrotor. In *2016 American Control Conference (ACC)*, pages 2252–2258, July 2016. doi: 10.1109/ACC.2016.7525253.

[15] B. Xu and K. Sreenath. Safe teleoperation of dynamic uavs through control barrier functions. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7848–7855, May 2018. doi: 10.1109/ICRA.2018.8463194.

Appendix

A. Safe Set Design - Superellipsoids

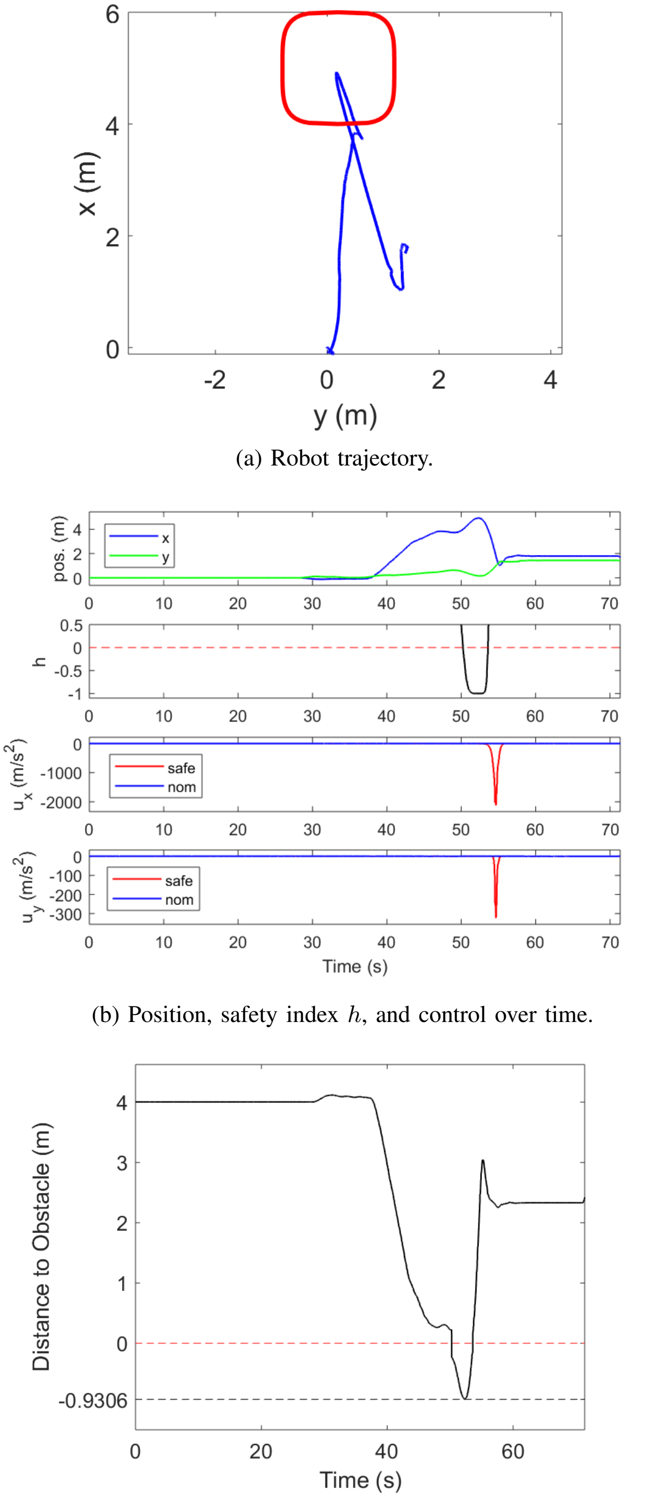
As done in [15], we choose a superellipsoid to represent obstacles in our safety index h. A safety buffer d_s based on the size of the quadrotor and tracking error is incorporated into h, resulting in the safety set and ECBF as follows:

$$C = \{r_i | h(\mathbf{r})\}, \quad i = 1, 2; \quad h(\mathbf{r}) = \left(\frac{r_1}{a_1}\right)^n + \left(\frac{r_2}{a_2}\right)^n - d_s \tag{3}$$

where a_i describe the size of obstacles and n describes the shape. As n increases, the shape of the bounding box becomes less circular and more rectangular; we choose $n = 4$ (as in [11]) to represent a rounded rectangular prism. Rearranging Eq. 3, we replace the constraint equation in Eq. 2 with $A\mathbf{u} \le b$, where $A = \begin{bmatrix} \frac{-4r_1^3}{a_1^4} & \frac{-4r_2^3}{a_2^4} \end{bmatrix}$ and $b = \sum_{i=1}^{2} \left[\left(\frac{12 r_i^2 \dot{r_i}^2}{a_i^4} + k_1 \frac{r_i^4}{a_i^4} + k_2 \frac{4 r_i^3 \dot{r_i}}{a_i^4} \right) \right] - d_s$.

B. Failure Cases

Figs. 4-7 show detailed results from the four trials where safety is violated.

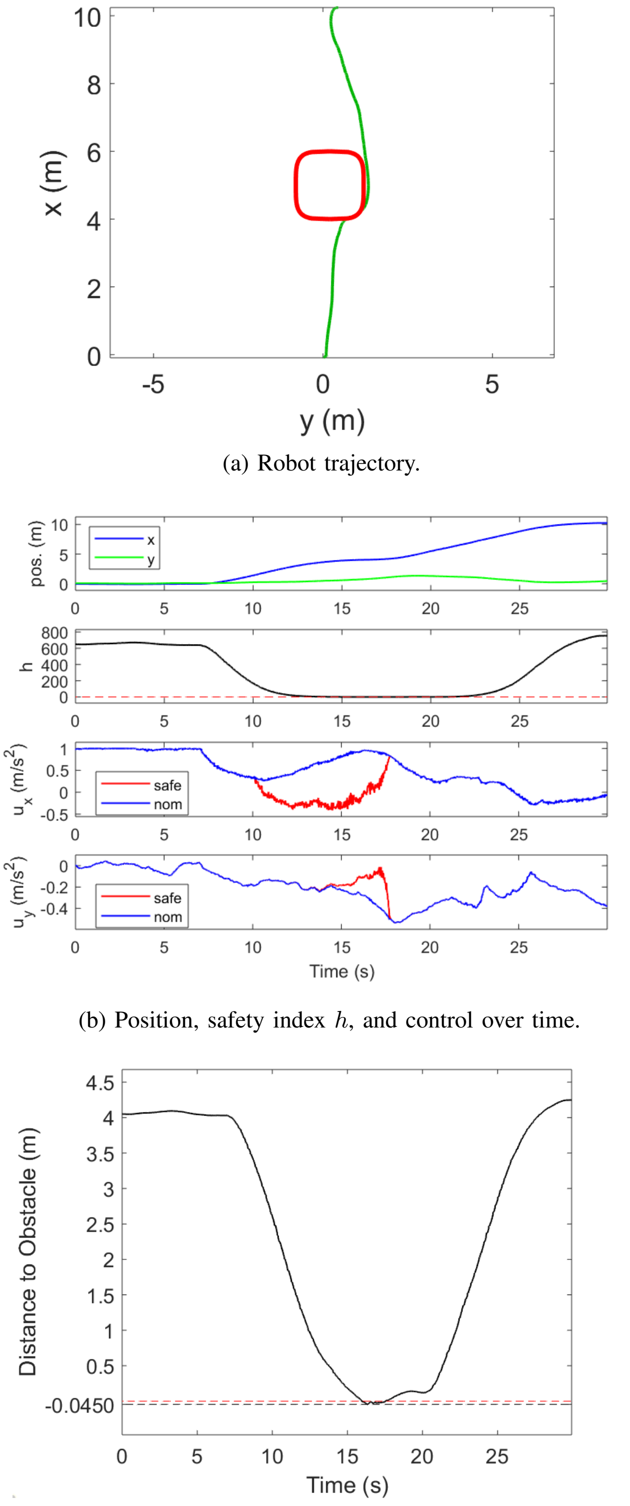


(a) Robot trajectory.

(b) Position, safety index h, and control over time.

(c) Distance from safe set (where negative values indicate collision) over time. Maximum penetration was 0.9306 meters.

Fig. 4: Summary plots of the failed trial F1.

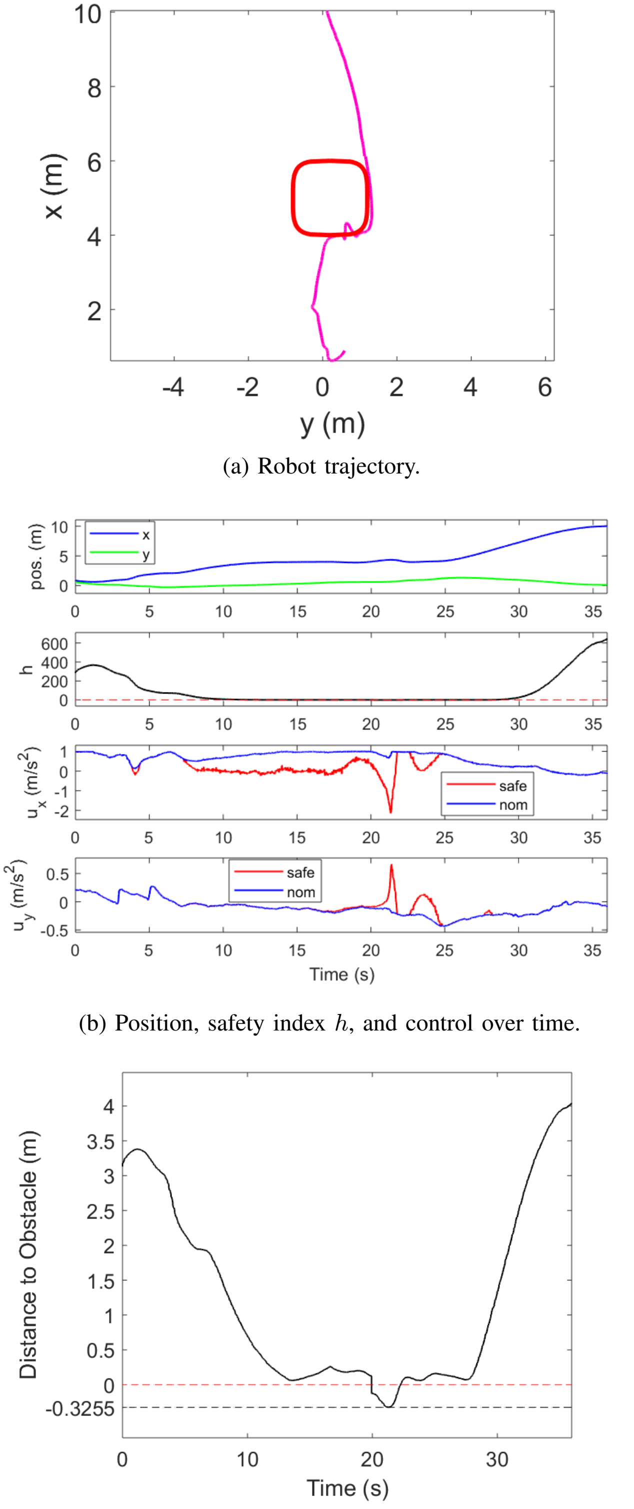


(a) Robot trajectory.

(b) Position, safety index h, and control over time.

(c) Distance from safe set (where negative values indicate collision) over time.

Fig. 5: Summary plots of the failed trial F2.

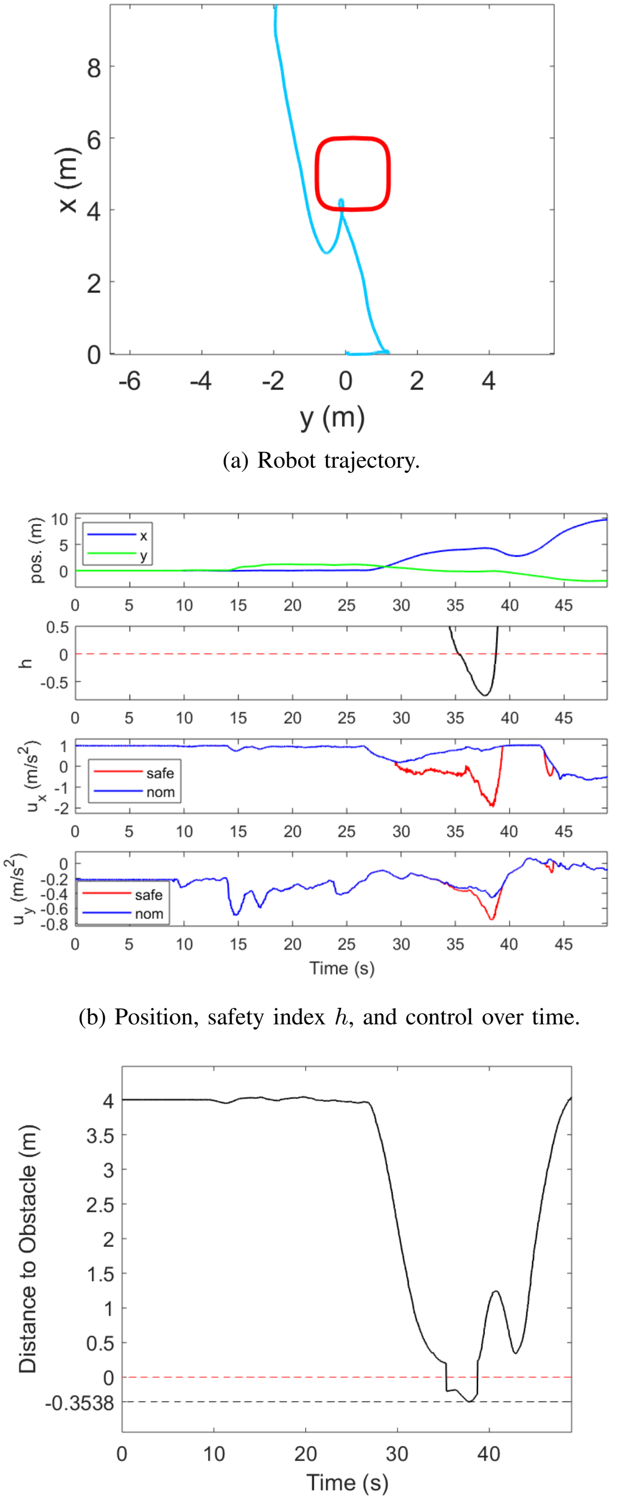


(a) Robot trajectory.

(b) Position, safety index h, and control over time.

(c) Distance from safe set (where negative values indicate collision) over time.

Fig. 6: Summary plots of the failed trial F3.



(a) Robot trajectory.

(b) Position, safety index h, and control over time.

(c) Distance from safe set (where negative values indicate collision) over time. Maximum penetration of set was 0.3538 meters.

Fig. 7: Summary plots of the failed trial F4.