2023 62nd IEEE Conference on Decision and Control (CDC)
December 13-15, 2023. Marina Bay Sands, Singapore

Safe Navigation of Networked Robots under Localization Uncertainty using Robust Control Barrier Functions*

Adam Miksits[1,2] Fernando S. Barbosa[1] Magnus Lindhé[1] José Araújo[1] Karl H. Johansson[2]

***Abstract*— 5G networks have the potential to provide external sensor data and offload computations for future industrial mobile robots. To enable these benefits while maintaining safety, we propose a modular architecture, including an onboard safety filter for the velocity control loop. The safety filter leverages robust control barrier functions to guarantee safety from collisions under bounded localization uncertainty. Initial experiments are performed to quantify the localization uncertainty and generate suitable bounds for the safety filter. We then derive the safety filter, and analyze its conservatism numerically. Finally, the method is demonstrated in experiments using an ABB Mobile YuMi® Research Platform robot.**

I. Introduction

A. Motivation

As 5G networks are expanding, more applications are being adapted to run partially or entirely over the network, instead of locally on individual devices. At the same time, connected devices can benefit from receiving additional information from other sensors, which they would otherwise not have access to. This is possible thanks to previously unachievable network characteristics, such as lower latency and guarantees on quality of service. For industrial applications, mobile robots are highlighted as one of the key use cases for 5G in a recent white paper by the 5G Alliance for Connected Industries and Automation (5G-ACIA) [1].

Connected mobile robots could get access to information about their environment from external sensors, and at the same time leverage the computational power available in edge or cloud computers to run more complex algorithms, enabling simplifications of the robots. To enable these benefits while maintaining safety, a method for taking external sensor information into account is required. This method needs to consider the current uncertainty in localization to avoid obstacles it cannot detect with its own sensors. Furthermore, the method should allow the robot to continue operating, even if a new path plan is not yet available from the offloaded planner, as long as it is safe.

An example of this is shown in Fig. 1, where an oil spill has been detected by an external sensor and communicated to the robot over the network, and the robot now has to deviate from the previously planned path to remain safe. The path plan is shown as a blue dashed line, with the green cross representing the navigation goal. The red arrow illustrates the best path-following velocity for the robot, while the green arrow represents a safe velocity input that deviates from the plan to make sure the robot does not enter the area containing the oil spill. The safe input takes into account the localization uncertainty shown in pink, guaranteeing that the robot remains safe even if its true position would be closer to the obstacle.

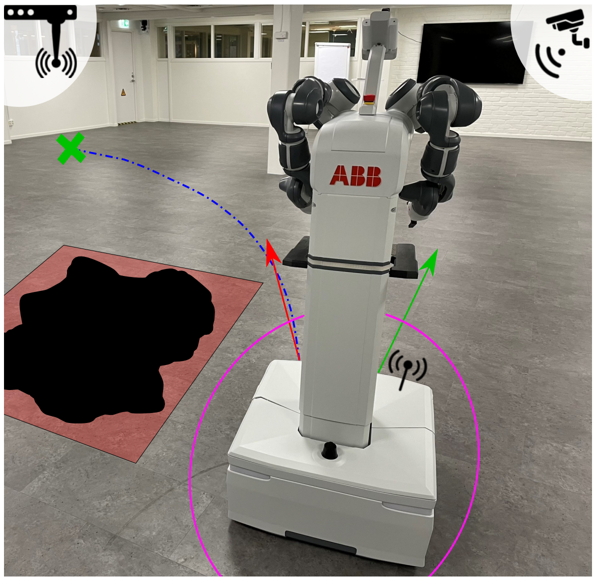

Fig. 1: ABB Mobile YuMi® Research Platform robot avoiding an obstacle identified by an external sensor. The robot is following a path plan in dashed blue, with the navigation goal marked by a green cross. The red arrow indicates an unsafe input that tracks the path, while the green arrow is a safe input generated by our method. The safe input takes into account the localization uncertainty shown in pink, and therefore deviates from the path to avoid entering a forbidden zone illustrated as the red area with the oil spill on the ground.

To address the scenario shown in Fig. 1, this paper proposes a method for safe navigation under localization uncertainty. We build on the popular open-source navigation framework Nav2 [2], and implement a robust safety filter to ensure the velocity inputs sent to the robot are safe when navigating based on the localization estimates. We leverage Control Barrier Functions (CBFs) [3] to implement this safety filter, and evaluate our method in numerical simulations as well as in experiments on a real mobile robot.

*This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation

[1]Ericsson Research, Sweden, {adam.miksits, fernando.dos.santos.barbosa, magnus.lindhe, jose.araujo}@ericsson.com
[2]KTH Royal Institute of Technology, Sweden {amiksits, kallej}@kth.se

B. Related Work

Before presenting the contribution of this paper, we review related work on networked mobile robots and safe navigation in general, as well as more specific approaches of modeling the perception uncertainty and compensating for it in the controller to guarantee safety.

Recently, several works [4], [5], [6] have investigated the possibility of offloading algorithms from industrial mobile robots, including those related to the lowest levels of control and safety. However, if a lot of data is sent over the same network by several devices, bandwidth limitations might have to be taken into account in the controller [7], [8]. Alternatively, dynamic bandwidth allocation can be used to ensure that time-criticality is taken into account when distributing network resources [9]. In this paper, we focus on safe control running locally on the robot, while allowing less time-critical algorithms, such as motion planning, to be offloaded.

To safely navigate a mobile robot, one approach is to ensure the path plan is safe from collisions, and then track the path plan with the motion controller. Numerous methods for planning collision-free trajectories are covered in many textbooks, e.g., [10]. Another approach is to separate planning from safe control by defining a set of safe states, and then ensuring safety using tools such as Hamilton–Jacobi reachability analysis [11] or CBFs [3], deviating from the plan only when it is unsafe. There are also works combining the two approaches, for example, model predictive control together with a CBF to ensure the robot follows a plan guaranteed to be safe [12].

If perception is included in the control loop, the uncertainty introduced from sensor-based estimation also needs to be taken into account, which has been addressed by recent works on so-called perception-based control. If safety is taken into account in the path planning level, there are methods to guarantee tracking despite uncertainties in state estimation [13], [14], [15]. For safe navigation, this also requires taking the uncertainty into account when planning however, which could be difficult if the uncertainty in position changes. For the separated planning and control approach, authors have recently developed methods using CBFs to take various perception uncertainties into account, either from sensor-based localization [16], [17] or from direct obstacle detection [18], [19]. We are inspired by this line of work. In particular, we extend the setup in [18], where all obstacles in the environment are represented as a single CBF, to a robust setting with uncertain localization.

C. Contribution

The contribution of this paper is threefold. First, we experimentally characterize the localization uncertainty in a navigation system on a mobile robot. By comparing a trajectory estimated from localizing in a previously built map with the same trajectory recorded by a motion capture system, we obtain a model for the localization uncertainty.

Second, we introduce a method for safe navigation taking into account the identified localization uncertainty, which we will refer to as a safety filter. This is achieved by including the entire environment in a CBF, representing obstacles identified both by the robot, as well as by other sensors in the same network. The localization uncertainty is then accounted for when using this CBF as a constraint in an optimization problem to generate safe control inputs.

Third, we implement and evaluate this safety filter in navigation experiments. We first analyze the robustness of the proposed approach in numerical simulations, and then validate the achieved safety in an experimental campaign using an ABB Mobile YuMi®Research Platform robot.

D. Outline

Section II introduces and motivates the problem of safe navigation under localization uncertainty with an experiment. A system architecture for navigation including a safety filter is then proposed to address the problem of uncertain localization. In Section III, we present the theory necessary to implement the safety filter in the proposed system architecture. The method is evaluated in Section IV, both in simulations and in real experiments to show the challenges and benefits of being robust to localization uncertainty. Finally, in Section V we conclude the paper and discuss possible future work.

II. Problem Formulation

In this section, we start with a motivating example to show the importance of considering localization uncertainty for safe navigation. We then propose a system architecture capable of taking the uncertainty into account, and finally define the problem considered in this paper - the design of the safety filter in the proposed architecture.

A. Motivating Example

Throughout this paper, we will use an ABB Mobile YuMi® Research Platform robot to further explore the scenario shown in Fig. 1. The robot is running the ROS2 navigation stack (Nav2 [2]) on its onboard computer, an Intel NUC with an eighth generation i7 CPU and 32GB of RAM. The navigation stack enables functions such as mapping, localization, path planning and path following. In terms of sensors, the robot has two SICK TIM781S Lidars, one facing forward and one facing backward, which it can use to create, or localize in, a map of the obstacles. The robot has four steerable wheels, enabling omnidirectional drive. The wheels are also equipped with encoders that are used for local odometry estimation.

By combining scan matching of Lidar measurements against the obstacle map with wheel encoder information, the localization algorithm estimates the position of the robot. This position estimate, which we will denote $\hat{x}$ in the following, enables feedback control for path following, which the robot uses to follow a path plan generated by one of the path planners available in Nav2. However, while the localization estimate can be used to drive the robot towards its goal, it will not correspond exactly to the true position x. Due to uncertainties in the estimation it will deviate with some error e such that $\hat{x} = x + e$, which could have an impact on safety.

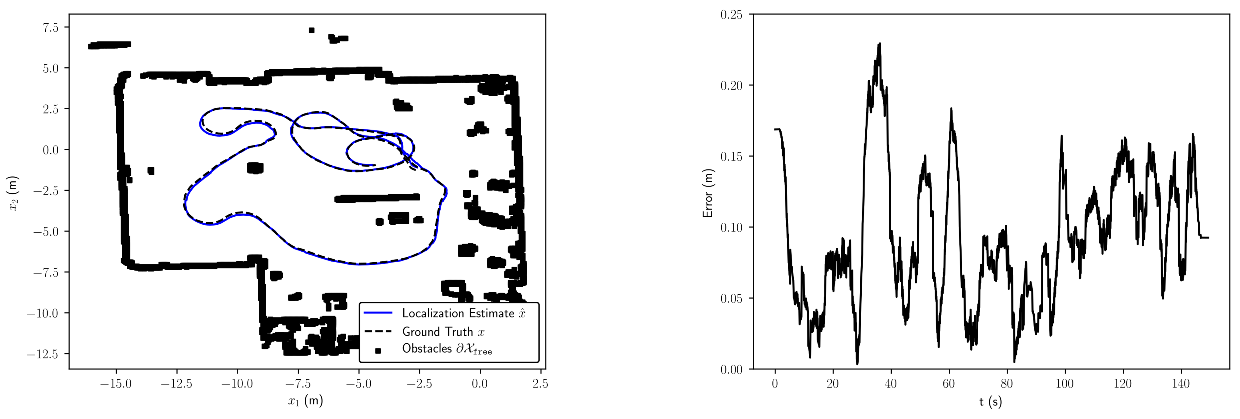

Fig. 2: Results from running a localization experiment in WARA Robotics. (Left) Trajectories generated by remotely controlling the robot. The solid blue curve indicate the localization estimate and the dashed black the ground truth obtained from the motion capture system. (Right) The localization error as a function of time for the experiment shown to the left.

To investigate the localization uncertainty in the navigation system, we perform a localization experiment with the robot in the WASP Research Arena for Robotics (WARA Robotics) at ABB. The robot is remote-controlled to first create a map, and we then proceed to drive the robot around while running localization in the map. Whenever a new estimate is available, we record it together with the latest position available from the Qualisys motion capture system installed in WARA Robotics. The resulting trajectories are shown in the left part of Fig. 2, and the corresponding localization error is shown in the right part of the Figure. The error was computed by aligning the trajectories and calculating position differences $||\hat{x} - x||_2$ using evo [20]. Most of the time, the localization error varies from $0\,\text{cm}$ to $15\,\text{cm}$, but in one case it comes close to $25\,\text{cm}$.

B. System Architecture

We propose the system architecture for safe navigation illustrated in Fig. 3. The blue boxes indicate features already available in Nav2 [2]. By combining scan matching of Lidar measurements against the obstacle map with wheel encoder information, the localization algorithm estimates $\hat{x}$, which is the current position of the robot in the free space $\mathcal{X}_{\text{free}}$. This estimate is then used by the path following module to make the robot follow an externally computed path plan, which is achieved by applying the velocity control u_{ref}.

A safety filter is included in the proposed navigation system. It ensures that a safe velocity input u_{safe} is implemented. This is achieved by comparing the path following velocity u_{ref} with a distance function generated from obstacle measurements. The boundary of the free space $\partial\mathcal{X}_{\text{free}}$ represents obstacles in the robot map, as well as obstacles identified by external sensors.

C. Problem Definition

The problem in this paper concerns the design of the safety filter with respect to the identified uncertainty and the uncertainty model $\hat{x} = x + e$. More specifically, we want to design a filter that outputs the closest safe velocities u_{safe} to input reference velocities u_{ref}, based on localization estimates $\hat{x}$ and obstacle measurements $\partial\mathcal{X}_{\text{free}}$, which are assumed to be noise-free.

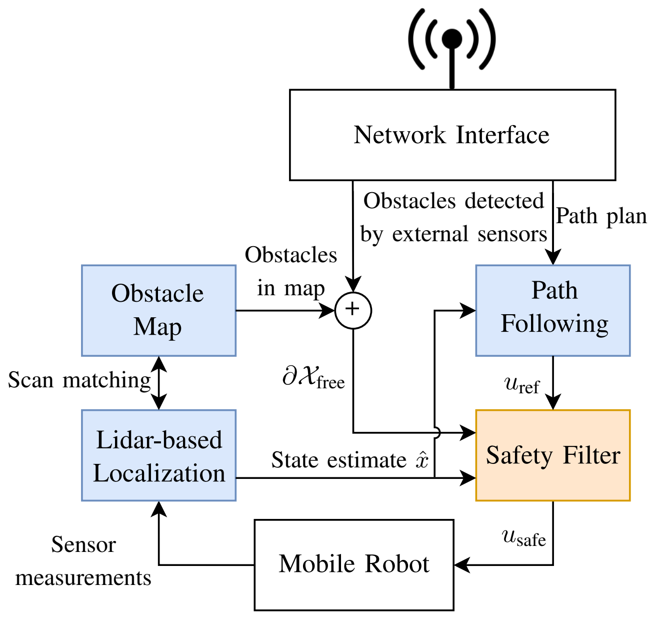

Fig. 3: Proposed modular system architecture for safe navigation of a networked mobile robot.

With some assumptions on the uncertainty e based on the localization experiment, we will show in the next section how this can be solved as an optimization problem that guarantees safe navigation. This is achieved by ensuring the true position x remains in a set $\mathcal{C} \subset \mathcal{X}_{\text{free}}$, where $\mathcal{C}$ is chosen so that the entire robot avoids collisions with $\partial\mathcal{X}_{\text{free}}$. In the derivation of the safety filter, a single integrator system model $\dot{x} = u$ will be considered for the robot, but the method can also be generalized to other models, which is covered briefly.

III. Safe Navigation under Localization Uncertainty

In this section, the necessary tools to create a robust safety filter are introduced. We first review basic safety and CBF formalism, and then show how this can be used to achieve safe navigation. Finally, the robust safety filter is derived.

A. Preliminaries

For general nonlinear systems, CBFs can be used to guarantee safety. Consider a nonlinear control affine system:

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

where $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$, $f : \mathbb{R}^n \to \mathbb{R}^n$ and $g : \mathbb{R}^n \to \mathbb{R}^{n \times m}$ are locally Lipschitz continuous on $\mathbb{R}^n$. Given a locally Lipschitz continuous state-feedback controller $k : \mathbb{R}^n \to \mathbb{R}^m$, the closed-loop system dynamics of the system can be defined as:

$$\dot{x} = f_{cl}(x) = f(x) + g(x)k(x). \tag{2}$$

We can then define the *safe set* $\mathcal{C} \subset \mathbb{R}^n$ as the 0-superlevel set of a continuously differentiable function $h : \mathbb{R}^n \to \mathbb{R}$ yielding:

$$\mathcal{C} = \{x \in \mathbb{R}^n : h(x) \geq 0\}. \tag{3}$$

Further assuming $\mathcal{C}$ is nonempty and contains no isolated points, enables the following definition:

Definition 3.1: (Forward Invariant & Safe) A set $\mathcal{C} \subset \mathbb{R}^n$ is forward invariant if for every initial condition $x(t_0) \in \mathcal{C}$, the solution $x(t)$ to system (2) satisfies $x(t) \in \mathcal{C}$ for all $t \geq t_0$. The system (2) is safe with respect to the set $\mathcal{C}$ if the set $\mathcal{C}$ is forward invariant.

Using the notion of extended class $\mathcal{K}_\infty$ functions ($\alpha \in \mathcal{K}_{\infty,e}$) and that c is a regular value of a continuously differentiable function $h : \mathbb{R}^n \to \mathbb{R}$ if $h(x) = c \longrightarrow \frac{\partial h}{\partial x}(x) \neq 0$, we can define a zeroing CBF as follows:

Definition 3.2: (Control barrier function (CBF), [21]) Let $\mathcal{C} \subset \mathbb{R}^n$ be the 0-superlevel set of a continuously differentiable function $h : \mathbb{R}^n \to \mathbb{R}$ with 0 a regular value. The function h is a CBF for system (1) if there exists $\alpha \in \mathcal{K}_{\infty,e}$ such that for all $x \in \mathcal{C}$:

$$\sup_{u \in \mathbb{R}^m} \frac{\partial h}{\partial x}(x) f(x) + \frac{\partial h}{\partial x}(x) g(x) u > -\alpha(h(x)). \tag{4}$$

Given such a CBF $h(x)$ further enables defining a set of control inputs that render the safe set $\mathcal{C}$ forward invariant:

$$K_h(x) = \left\{ u \in \mathbb{R}^m : \frac{\partial h}{\partial x}(x) f(x) + \frac{\partial h}{\partial x}(x) g(x) u \geq -\alpha(h(x)) \right\}. \tag{5}$$

To obtain a CBF for safe navigation, we use a smooth under-approximation of the Euclidean Distance Field (EDF) from [18]. This method generates a function $d(x)$ representing the distance to the closest obstacle p from a given position x in the free space $\mathcal{X}_{\text{free}}$ (which is assumed to have a smooth boundary)

$$d(x) := \inf_{p \in \partial \mathcal{X}_{\text{free}}} ||x - p||_2 \tag{6}$$

where the gradient of d satisfies the Eikonal equation

$$|\nabla d| = 1 \tag{7}$$

as well as the boundary constraints $d(x) = 0$ and $\frac{\partial d(x)}{\partial \mathbf{n}} = 1$ on $x \in \partial \mathcal{X}_{\text{free}}$. Here, $\frac{\partial d(x)}{\partial \mathbf{n}}$ is the gradient of $d(x)$ along $\mathbf{n}$, the normal of the boundary. This allows defining the safe set and barrier function as follows

$$\mathcal{C} := \{x \in \mathcal{X}_{\text{free}} \subset \mathbb{R}^n : h_d(x) \geq 0\}, \tag{8}$$
$$h_d(x) = d(x) - \gamma \tag{9}$$

where $\gamma > 0$ is the safety margin, i.e. the minimal distance to maintain from all obstacles.

In a navigation scenario, we assume the controller $k(x)$ in (2) is designed such that the robot is following a path plan using a nominal, potentially unsafe control input $u_{\text{ref}}(x)$, which is assumed to be locally Lipschitz. As proposed in [22], we define a set of safe inputs $K_{h_d}(x)$ as in (5), but based on $h_d(x)$. This enables filtering the output of the nominal controller using optimization, to select a u that deviates minimally from the reference, while ensuring the control signal remains in the set of safe inputs

$$\begin{aligned} u^* = \underset{u \in \mathbb{R}^m}{\arg\min} \quad & \frac{1}{2} ||u - u_{\text{ref}}(x)||_2^2 \\ \text{s.t.} \quad & u \in K_{h_d}(x). \end{aligned} \tag{CBF OP}$$

B. Robust Control Barrier Functions for Localization Uncertainty

In Section II, we showed that when the robot navigates based on state estimates $\hat{x}$ from localization, the localization error is not negligible, and needs to be taken into account in the controller to guarantee safety. We also defined a model for the relationship between localization estimates and true positions of the robot, and stated that the design of the safety filter has to take into account how this uncertainty affects the system model, and the barrier function we use to keep the system safe.

If we assume to know an upper bound ϵ on the error, i.e. $||e||_2 = ||\hat{x} - x||_2 \leq \epsilon$, the effects of controlling based on $\hat{x}$ instead of x can be investigated. For example, the state influences the control signal through the CBF and its gradient. In Fig. 4, the gradient of h_d is visualized as arrows, and an example of possible $\hat{x}$ and x is shown in blue and black respectively, given $\epsilon = 0.5$. From Fig. 4, it is clear that this uncertainty could mean the true state x is $0.5\,\mathrm{m}$ closer to an obstacle than the estimate $\hat{x}$. Even more importantly, it is also shown that an uncertainty in x could mean the gradients differ, meaning $\frac{\partial h_d}{\partial x}(x) \neq \frac{\partial h_d}{\partial x}(\hat{x})$, which needs to be considered when choosing safe inputs.

We now propose the optimization problem for the robust safety filter before proceeding to derive the condition used

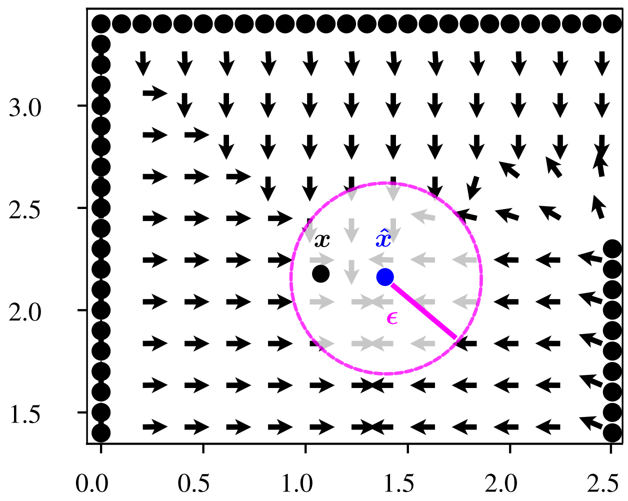

Fig. 4: Illustrates impact of localization uncertainty on the gradient of an EDF. A 2D obstacle map is shown, with arrows representing the direction of the gradient and obstacle measurements represented as circles.

in the optimization problem to guarantee safety

$$u_{\text{safe}}(\hat{x}) = \underset{u\in\mathbb{R}^m}{\arg\min} \quad \frac{1}{2}||u - u_{\text{ref}}(\hat{x})||_2^2 \qquad \text{(RCBF OP)}$$
$$\text{subject to} \quad u \in K_{h_d,\text{robust}}(\hat{x})$$

where

$$K_{h_d,\text{robust}}(\hat{x}) =$$
$$\{u \in \mathbb{R}^m \;:\; \frac{\partial h_d}{\partial x}(\hat{x})u - \epsilon\mathcal{L}_\alpha - 2.0||u||_2 > -\alpha(h_d(\hat{x}))\}.$$

In the remainder of this section, we derive the safety condition in (RCBF OP), and show that choosing $u \in K_{h_d,\text{robust}}(\hat{x})$ will ensure the pair (x, u) satisfies the original CBF condition (4). This safety condition follows as a natural extension to the condition in the optimization based controller (MR-OP) from [16], but certain adaptations have been made for the safe navigation problem and the specific CBF we consider in this work.

Showing that the optimization-based controller guarantees safe navigation is relatively straight-forward. We first define a function $c(x,u) = \frac{\partial h_d}{\partial x}(x)f(x) + \frac{\partial h_d}{\partial x}(x)g(x)u + \alpha(h_d(x))$. To show that $u \in K_{h_d,\text{robust}}(\hat{x})$ ensures safety of system (1) is equivalent to showing $c(x,u) \geq 0$. Using the relation $\hat{x} = x + e$ with $||\hat{x} - x||_2 \leq \epsilon$, we can lower-bound $c(x,u)$ for all x in the set $\{x \in \mathcal{C} : ||\hat{x} - x||_2 \leq \epsilon\}$

$$\begin{aligned}\inf_{||\hat{x}-x||_2\leq\epsilon} c(x,u) &= \inf_{||e||_2\leq\epsilon} c(\hat{x}-e,u)\\ &= c(\hat{x},u) + \inf_{||e||_2\leq\epsilon} c(\hat{x}-e,u) - c(\hat{x},u)\\ &\geq c(\hat{x},u) - \sup_{||e||_2\leq\epsilon} |c(\hat{x}-e,u) - c(\hat{x},u)|.\end{aligned}$$

We then assume that $\alpha(h_d(x))$ is Lipschitz continuous on $\mathcal{C}$ with coefficient $\mathcal{L}_\alpha$. We also assume that the velocities are inputs for a single integrator system model, i.e. $f(x) = 0$ and $g(x) = \mathcal{I}$. Combining these assumptions with the properties of $h_d(x)$ enables bounding $|c(\hat{x}-e,u) - c(\hat{x},u)|$:

$$\begin{aligned}&|c(\hat{x}-e,u) - c(\hat{x},u)| =\\ &= \left|\frac{\partial h_d}{\partial x}(\hat{x}-e)u + \alpha(h_d(\hat{x}-e)) - \left(\frac{\partial h_d}{\partial x}(\hat{x})u + \alpha(h_d(\hat{x}))\right)\right|\\ &\leq \left|\left|\frac{\partial h_d}{\partial x}(\hat{x}-e) - \frac{\partial h_d}{\partial x}(\hat{x})\right|\right|_2 ||u||_2 + \mathcal{L}_{\alpha\circ h_d}||(\hat{x}-e) - \hat{x}||_2\\ &\leq 2.0\,||u||_2 + \epsilon\mathcal{L}_\alpha\end{aligned}$$

where in the last step, $||\frac{\partial h_d}{\partial x}(\hat{x} - e) - \frac{\partial h_d}{\partial x}(\hat{x})||_2$ was bounded by 2.0 since the gradient of $h_d(x)$ satisfies the Eikonal equation (Eq. 7). Also, for the same reason $\mathcal{L}_{\alpha\circ h_d} = \mathcal{L}_\alpha\mathcal{L}_{h_d} = \mathcal{L}_\alpha$, since $\mathcal{L}_{h_d} = 1$. Finally, this means that

$$\inf_{||\hat{x}-x||_2\leq\epsilon} c(x,u) \geq c(\hat{x},u) - 2.0\,||u||_2 - \epsilon\mathcal{L}_\alpha,$$

and since we enforce $c(\hat{x},u) - 2.0||u||_2 - \epsilon\mathcal{L}_\alpha \geq 0$ in (RCBF OP), $\inf_{||\hat{x}-x||_2\leq\epsilon} c(x,u) \geq 0$, and the true system remains safe.

The two major differences to previous work concern the bound $||\frac{\partial h_d}{\partial x}(\hat{x} - e) - \frac{\partial h_d}{\partial x}(\hat{x})||_2$ and the assumption that the control inputs are velocities for a single integrator system model. The bound is necessary for CBFs with a discontinuous gradient, which will be present if it represents a map with more than one obstacle. The bound reduces to a very simple form when assuming a single integrator model, and while the assumption is limiting, the method can be extended to guarantee safe navigation for more complex system models. This was demonstrated in [23] for non-robust CBFs, where a reference control input for another model was translated into single integrator inputs for safety filtering, and then transformed back. However, the bound might still be overly conservative in most cases, which is discussed more in the following remark.

Remark 1: While the same Lipschitz argument as in previous work was used to bound the term involving $\alpha(h_d(x))$, this argument is not valid for $||\frac{\partial h_d}{\partial x}(\hat{x}-e) - \frac{\partial h_d}{\partial x}(\hat{x})||_2$ in our case, since the barrier function $h_d(x)$, despite being smooth, has a discontinuous gradient. This is illustrated in Fig. 4, where the gradient switches when moving between areas closer to obstacles along different walls. Instead, we find a global robustness bound based on the maximum change of direction of the gradient $\frac{\partial h_d}{\partial x}$ in the safe set, which yields an upper bound for all $x \in \{x \in \mathcal{C} : ||\hat{x} - x||_2 \leq \epsilon\}$, namely

$$\left|\frac{\partial h_d}{\partial x}(\hat{x})u - \frac{\partial h_d}{\partial x}(x)u\right| \leq 2.0||u||_2.$$

Note that there is no explicit relation to the uncertainty, but if $\epsilon = 0$, we would have $\frac{\partial h_d}{\partial x}(\hat{x}) = \frac{\partial h_d}{\partial x}(x)$. The same would be true if the robot is moving alongside a flat wall, since the gradient would remain constant despite localization uncertainty. In most cases, this bound will be overly conservative, and inspired by the motivation in [24], where a trade-off between safety and performance was investigated for robust control barrier functions, we will consider replacing the factor 2.0 with a coefficient $\beta \in [0.0, 2.0]$. Therefore, in

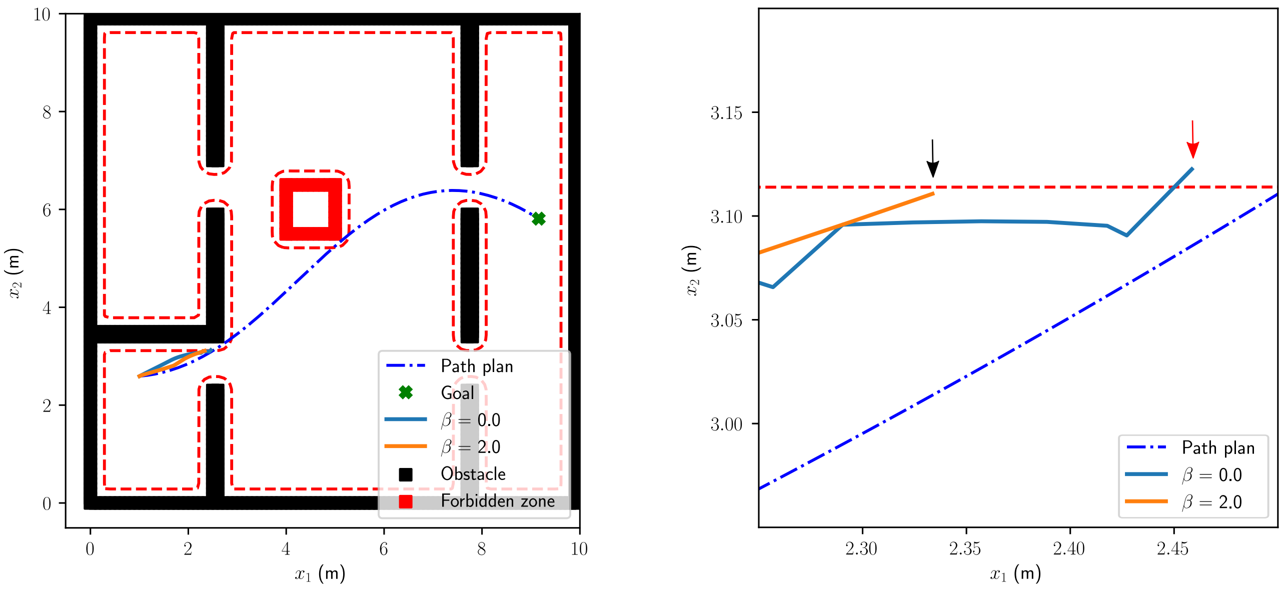

Fig. 5: Results from numerical navigation experiment, dashed red lines indicating the safety margin. (Left) Trajectories with β set to zero or the worst-case value 2.0. (Right) Zoomed in plot showing the opening where the robot either got stuck due to conservatism ($\beta = 2.0$) or violated the safety constraint ($\beta = 0.0$), indicated with a black and red arrow respectively.

the next section we will first perform numerical experiments to find a suitable value for β, and then implement the safety filter with this coefficient β on the real robot.

IV. Experiments

In this section, the robust CBF is implemented and used for safe navigation. Safety is achieved by filtering reference tracking velocities on the robot, based on an EDF generated from already mapped obstacles in the environment, as well as obstacles detected by external sensors. We first analyze the conservatism of the robust safety filter in numerical experiments, and then compare the robust safety filter with a nominal safety filter in navigation experiments on the mobile robot.

A. Numerical Experiments

To analyze the conservatism of the robust safety filter, the robot is simulated as a single integrator in a 2D environment with fixed obstacles, and a path was planned so that following the plan perfectly will violate safety conditions. The simulations as well as the safety filter runs at 100 Hz. We implement a relaxed version of the (RCBF OP), adding slack with a large penalty to ensure feasibility as was done in [16]. The robustness term is chosen as $\epsilon = 0.25\,\mathrm{m}$, and $h_d(x)$ defined as in (9), using a safety margin $\gamma = 0.29\,\mathrm{m}$. This makes safe navigation to the goal difficult, but not impossible, since the most narrow openings along the path are $1.1\,\mathrm{m}$ wide. To simulate localization uncertainty, we subtract ϵ from the x_2-coordinate in the robot's state estimates that are used for path following.

In the left of Fig. 5, results are shown from a navigation experiment with a safety filter using either a worst-case robustness term ($\beta = 2.0$) or no compensation ($\beta = 0.0$) for uncertainty in gradient. Both robots moved towards the top of the opening due to the localization uncertainty in x_2. A zoomed-in view of this opening is shown in the right of Fig. 5. While $\beta = 2.0$ formally guarantees safety with respect to the worst-case change in gradient direction, this value is too conservative and the robot gets stuck, as indicated by the black arrow. On the other hand, $\beta = 0.0$ results in a safety violation in the point indicated by the red arrow, after which the experiment was aborted. This indicates that a $\beta > 0$ is necessary for safety.

Intuitively, if the reference velocities from the path following module brings the robot close to the boundary of the safe set, at some point the terms not related to u in the CBF condition in (RCBF OP) will cancel, i.e. $-\alpha(h_d(\hat{x})) + \epsilon\mathcal{L}_\alpha \to 0$, requiring $\frac{\partial h_d}{\partial x}(\hat{x})u - \beta||u||_2 > 0$. If $\beta \geq 1$, there will be no theoretically feasible u, since the gradient is normalized, resulting in $\left|\left|\frac{\partial h_d}{\partial x}(\hat{x})u\right|\right|_2 - \beta||u||_2 \leq 0$. However, for any $\beta < 2.0$, the theoretical safety guarantees no longer hold - by changing β we make a trade-off between safety and performance.

To ensure this trade-off prioritizes safety while still achieving acceptable performance, we repeat the experiment for some different values of β around and below 1.0, and again interrupt experiments when safety is violated. Results are shown in the left of Fig. 6, zooming in on the same narrow opening as before. As expected, the trajectories associated with both $\beta = 1.05$ and $\beta = 1.0$ got stuck close to the safety margin showed in dashed red. On the other hand, lower values of β resulted in safety violations. As shown in the right of Fig. 6, only the trajectory generated with $\beta = 0.95$ passed the opening safely (and reached the goal), so for this reason, the safety filter with $\beta = 0.95$ will be implemented and evaluated in the experiment with the real robot.

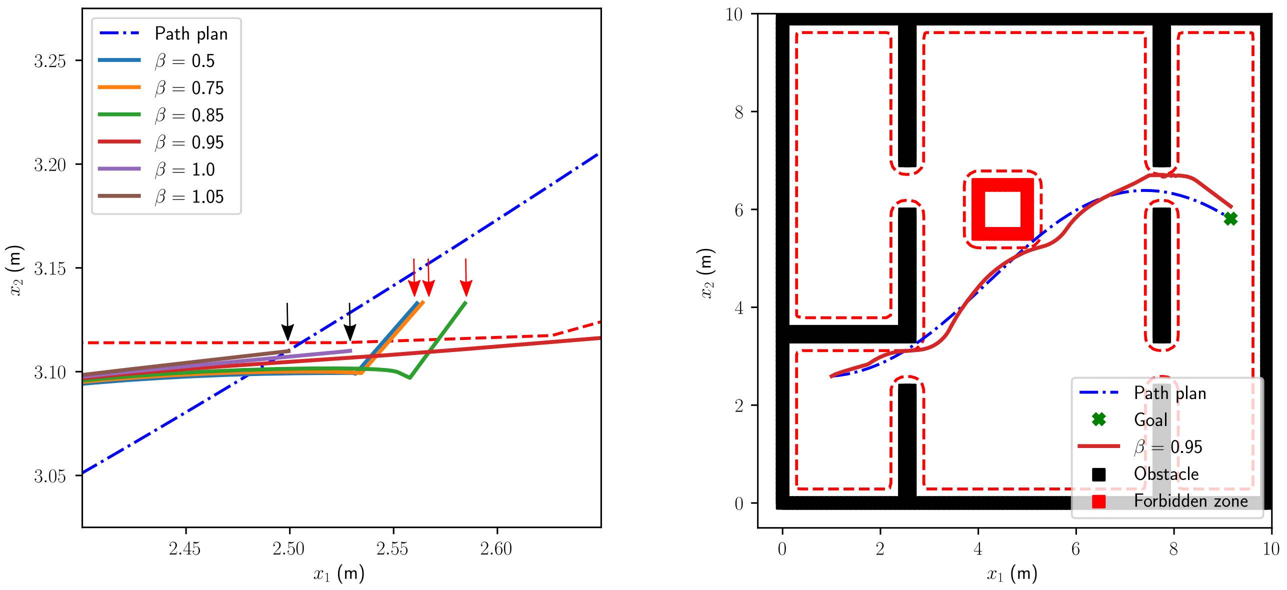

Fig. 6: Conservatism analysis of safety filters with varying values of the robustness coefficient. The dashed red line indicates the safety margin. (Left) Zoomed in plot of resulting trajectories with varying values of β in the same opening as in Fig. 5. Black arrows indicate the ends of trajectories from experiments where a robot got stuck due to overly conservative parameter choice, and red arrows indicate where an experiment was interrupted due to a safety violation. (Right) Shows the trajectory with $\beta = 0.95$ for which the robot successfully reached the goal without any safety violations.

B. Mobile Robot Experiments

In the following, the safety filter is implemented in the navigation stack of an ABB Mobile YuMi® Research Platform robot according to the architecture shown previously in Fig. 3. The robot is running path following at 10 Hz on its onboard computer. We implement our safety filter taking into account the uncertainty identified before by setting $\epsilon = 0.25\,\text{m}$, and use a robustness term $\beta = 0.95$. In the experiment, we compare our method with a safety filter based on the nominal CBF optimization problem (CBF OP) using $\hat{x}$ as input. For both of these, we choose $\gamma = 0.5\,\text{m}$ in the CBF $h_d(x)$, since the robot is 33 cm wide and 30 cm long.

The safety filter is then evaluated in a navigation scenario as illustrated in Fig. 1. The robot is given coordinates in its map representing the boundary of the forbidden zone, which it needs to avoid in addition to the obstacles already in the map. The results for both methods are shown in Fig. 7. While no safety violations occurred using our method, using a safety filter not taking the uncertainty into account resulted in violations, both near the forbidden zone and near an obstacle in the map. Furthermore, even if the nominal safety filter is supposed to keep at least $\hat{x}$ safe, it still failed to ensure $\hat{x} \in \mathcal{C}$. This was likely due to model inaccuracies, i.e. the mobile robot having inertia which was not considered by guaranteeing safety of the single integrator model.

Another probable explanation is that this is an artifact of using safety guarantees designed for continuous systems with a path following algorithm running at 10 Hz, which also resulted in jerky movement close to the safety margin. The main reason for this choice of frequency was the EDF implementation being computationally cumbersome, thus limiting the rate at which the safety filter receives distance information.

It would be interesting to investigate how the computational efficiency could be improved. For example, if structure in the environment could be used to make the distance field representation more efficient, this could also be exploited to design separate CBFs for obstacles with smooth or constant distance field gradients, which could also reduce the need for conservatism as discussed in the previous section. This could also be a good middle ground between representing each obstacle as one CBF and representing all obstacles with one CBF.

V. CONCLUSIONS

In this paper, we showed the importance of taking into account localization uncertainty when controlling mobile robots to maintain a safe distance with respect to unsafe regions, such as physical obstacles and forbidden areas. We proposed a navigation system architecture, and quantified the localization uncertainty for this system. We then derived a robust safety filter to compensate for the identified uncertainty, analyzed its robustness properties in simulation, and finally demonstrated its performance on a real mobile robot.

Aside from looking into a more efficient EDF implementation, future work could consider discrete-time control barrier functions to remedy the issue with the safety filter's low update frequency. We would also like to extend the problem formulation to time-varying and non-symmetric localization uncertainties, and investigate how localization uncertainty and safe navigation is affected by other aspects, for instance bandwidth availability, when localization is offloaded.

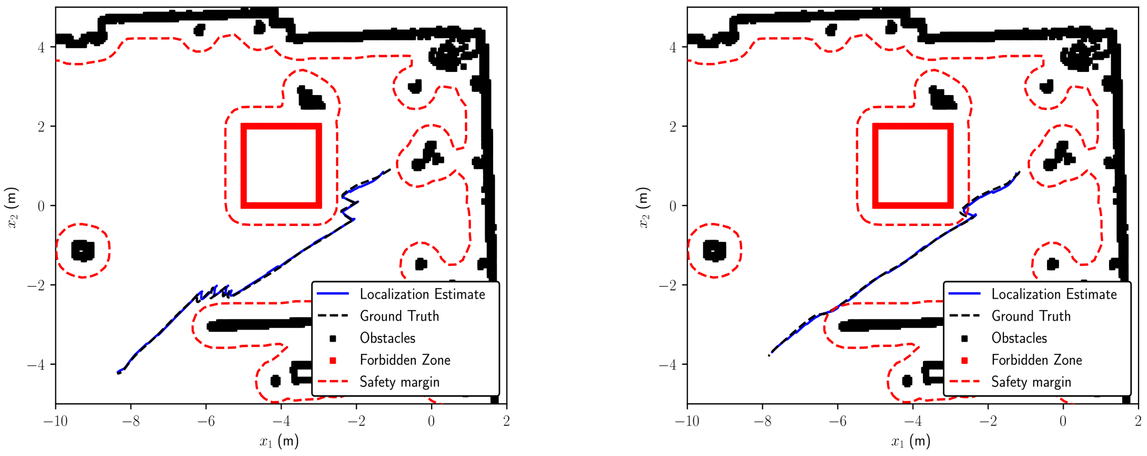

Fig. 7: Results from experiments in WARA Robotics. (Left) True and estimated trajectory from navigation experiment with robust safety filter. (Right) True and estimated trajectory from navigation experiment with nominal safety filter.

ACKNOWLEDGMENT

We would like to thank Aris Synodinos and Jonas Larsson at ABB for helping us set up the experiments in WARA Robotics. We also want to thank Ryan Cosner for taking the time to explain how Measurement-Robust Control Barrier Functions and their implementation works. Finally, we would like to thank Erik Warberg for the help with implementing a nominal Control Barrier Function based on the Euclidean Distance Field in ROS2.

REFERENCES

[1] 5G-ACIA, "Key 5G Use Cases and Requirements," Frankfurt am Main, Germany, Tech. Rep., May 2020. [Online]. Available: https://5g-acia.org/wp-content/uploads/5G-ACIA_WP_Key-5G-Use-Cases-and-Requirements_SinglePages.pdf

[2] S. Macenski, F. Martín, R. White, and J. G. Clavero, "The Marathon 2: A navigation system," in *International Conference on Intelligent Robots and Systems*. IEEE, 2020, pp. 2718–2725.

[3] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *European control conference*. IEEE, 2019, pp. 3420–3431.

[4] 5G-SMART, "5G-SMART Final Report," Stockholm, Sweden, Tech. Rep. D7.4, 2022. [Online]. Available: https://5gsmart.eu/wp-content/uploads/5G-SMART-D7.4-v1.0.pdf

[5] M. Balogh, A. Vidács, G. Fehér, M. Maliosz, M. Á. Horváth, N. Reider, and S. Rácz, "Cloud-controlled autonomous mobile robot platform," in *International Symposium on Personal, Indoor and Mobile Radio Communications*. IEEE, 2021, pp. 1–6.

[6] A. Baxi, M. Eisen, S. Sudhakaran, F. Oboril, G. S. Murthy, V. S. Mageshkumar, M. Paulitsch, and M. Huang, "Towards factory-scale edge robotic systems: Challenges and research directions," *Internet of Things Magazine*, vol. 5, no. 3, pp. 26–31, 2022.

[7] H. Li, G. Chen, T. Huang, and Z. Dong, "High-performance consensus control in networked systems with limited bandwidth communication and time-varying directed topologies," *Transactions on Neural Networks and Learning Systems*, vol. 28, no. 5, pp. 1043–1054, 2016.

[8] Z. Lu and G. Guo, "Control and communication scheduling co-design for networked control systems: a survey," *International Journal of Systems Science*, vol. 54, no. 1, pp. 189–203, 2023.

[9] M. H. Mamduhi, D. Maity, J. S. Baras, and K. H. Johansson, "A cross-layer optimal co-design of control and networking in time-sensitive cyber-physical systems," *Control Systems Letters*, vol. 5, no. 3, pp. 917–922, 2020.

[10] L. E. Kavraki and S. M. LaValle, "Motion Planning," in *Springer Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. Berlin, Heidelberg: Springer, 2008, pp. 109–131.

[11] A. Bajcsy, S. Bansal, E. Bronstein, V. Tolani, and C. J. Tomlin, "An efficient reachability-based framework for provably safe autonomous navigation in unknown environments," in *Conference on Decision and Control*. IEEE, 2019, pp. 1758–1765.

[12] P. Roque, W. S. Cortez, L. Lindemann, and D. V. Dimarogonas, "Corridor MPC: Towards optimal and safe trajectory tracking," in *American Control Conference*. IEEE, 2022, pp. 2025–2032.

[13] S. Dean, N. Matni, B. Recht, and V. Ye, "Robust guarantees for perception-based control," in *Learning for Dynamics and Control*. PMLR, 2020, pp. 350–360.

[14] L. Jarin-Lipschitz, R. Li, T. Nguyen, V. Kumar, and N. Matni, "Robust, perception based control with quadrotors," in *International Conference on Intelligent Robots and Systems*. IEEE, 2020, pp. 7737–7743.

[15] L. Lindemann, A. Robey, L. Jiang, S. Tu, and N. Matni, "Learning robust output control barrier functions from safe expert demonstrations," *arXiv preprint arXiv:2111.09971*, 2021.

[16] S. Dean, A. Taylor, R. Cosner, B. Recht, and A. Ames, "Guaranteeing safety of learned perception modules via measurement-robust control barrier functions," in *Conference on Robot Learning*. PMLR, 2021, pp. 654–670.

[17] R. K. Cosner, A. W. Singletary, A. J. Taylor, T. G. Molnar, K. L. Bouman, and A. D. Ames, "Measurement-robust control barrier functions: Certainty in safety with uncertainty in state," in *International Conference on Intelligent Robots and Systems*. IEEE, 2021, pp. 6286–6291.

[18] F. S. Barbosa and J. Tumova, "Risk-Aware Navigation on Smooth Approximations of Euclidean Distance Fields Among Dynamic Obstacles," 2021.

[19] R. K. Cosner, I. D. J. Rodriguez, T. G. Molnar, W. Ubellacker, Y. Yue, A. D. Ames, and K. L. Bouman, "Self-supervised online learning for safety-critical control using stereo vision," in *International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 11 487–11 493.

[20] M. Grupp, "evo: Python package for the evaluation of odometry and SLAM," 2017. [Online]. Available: https://github.com/MichaelGrupp/evo

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

[22] A. D. Ames, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs with application to adaptive cruise control," in *Conference on Decision and Control*. IEEE, 2014, pp. 6271–6278.

[23] T. G. Molnar, R. K. Cosner, A. W. Singletary, W. Ubellacker, and A. D. Ames, "Model-free safety-critical control for robotic systems," *Robotics and Automation Letters*, vol. 7, no. 2, pp. 944–951, 2021.

[24] R. Cosner, M. Tucker, A. Taylor, K. Li, T. Molnar, W. Ubelacker, A. Alan, G. Orosz, Y. Yue, and A. Ames, "Safety-aware preference-based learning for safety-critical control," in *Learning for Dynamics and Control*. PMLR, 2022, pp. 1020–1033.