2025 23rd European Control Conference (ECC)
June 24-27, 2025. Thessaloniki, Greece

Communication and Control Co-Design for Risk-Aware Safety of Mobile Robots with Offloaded Localization*

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

***Abstract*— With Edge Computing and 5G, industrial mobile robots will be able to offload computationally expensive algorithms such as sensor-based localization to the edge. However, multiple robots streaming large volumes of data over the network simultaneously will create network congestion, leading to high latencies and data loss, which can severely impact the robot operation. In this paper, we address this problem from a safety perspective by looking at how much communication can be reduced before risking safety violations due to increased localization uncertainty. We propose a co-design approach that adjusts communication and control jointly according to a requirement on localization uncertainty and show that by satisfying this requirement, safety can also be achieved. The method leverages a data-driven model of how the uncertainty depends on both communication and control. The performance of the optimization problem is evaluated experimentally on an ABB Mobile YuMi® Research Platform robot in both simulations and on hardware, and the results indicate that communication can be reduced without compromising safety.**

I. Introduction

With 5G networks expanding, we are looking at more and more applications that can reap the benefits of wireless connectivity. Connected devices will be able to offload heavy processes to edge computers over the network and can thus be simplified while achieving the same, or better, performance. The 5G Alliance for Connected Industries and Automation (5G-ACIA) highlights mobile robots as a key use case for Industrial 5G [1], and in particular the possibility to offload real-time localization as an important use case for industrial 5G edge computing [2]. At the same time, offloading the localization comes with the need to stream sensor data over the network, so if tens or hundreds of robots have to localize simultaneously, each should only use as much bandwidth as it needs. Another aspect to consider is how the robots are controlled, since this can also affect how difficult it is to localize with a given amount of sensor information. The importance of such application-specific control-communication co-design is highlighted as an important step towards factory-scale edge robotic systems in [3].

*This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation. Our experiments were carried out in the WASP Research Arena (WARA)-Robotics, hosted by ABB Corporate Research Center in Västerås, Sweden and financially 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, jose.araujo}@ericsson.com

[2]KTH Royal Institute of Technology, Sweden {amiksits, kallej}@kth.se

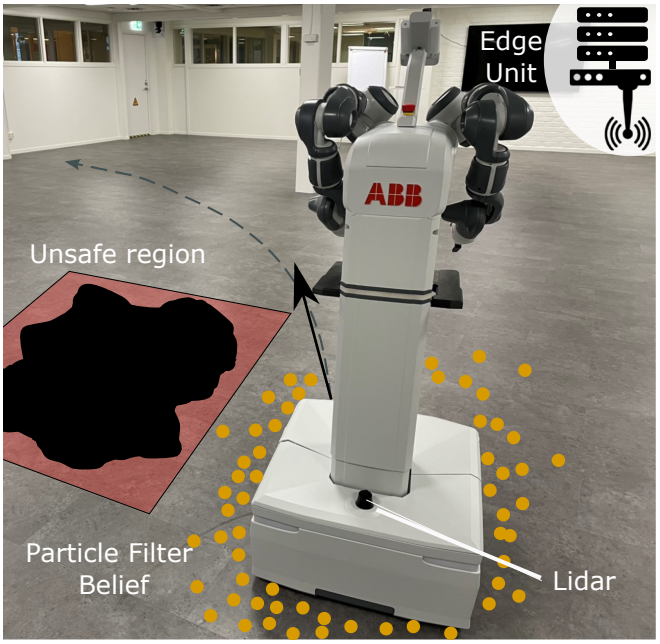

Fig. 1: ABB Mobile YuMi® Research Platform robot following a path plan in dashed gray by navigating based on offloaded localization running in the edge computing unit.

In the case of offloaded localization, an important performance indicator is localization uncertainty, which reflects a region around the estimated position where the true position is likely to be contained. This is different from localization error, which requires knowing the true position of the robot to compute the error of the estimate. Some recent works have looked at how to allocate bandwidth to minimize the total estimation uncertainty over several subsystems when each system needs to send sensor measurements over a network channel with limited capacity [4], [5]. However, in addition to indicating performance, the uncertainty is also related to safety, as shown in the scenario in Fig. 1. The robot is navigating with offloaded localization, in this case using a particle filter that estimates the position using a set of particles, shown as yellow spots in the image, to represent the position probabilistically. To reduce the spread of the particles, and thus also the uncertainty, the robot could request more bandwidth to be able to transmit sensor data to the edge more often. However, if the rest of the resources are already allocated to other robots this might not help.

If the bandwidth for the robot cannot be increased, it could adjust its motion to remain safe instead. Several recent works [6], [7], [8], [9], [10], [11], [12] have looked at safety

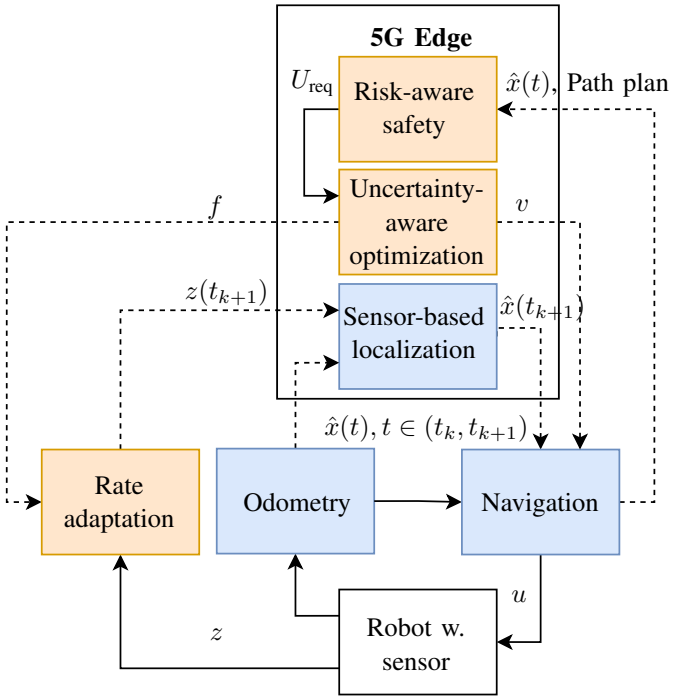

Fig. 2: Proposed framework for safe navigation with offloaded localization.

guarantees for motion control with uncertainties in state estimation. However, if the uncertainty is too high in relation to the nearby obstacles and the allowed deviation from the reference trajectory, solving the problem only at the control level might become infeasible. Then no safe control inputs are available and the robot will be forced to stop until more resources can be used to get a lower uncertainty.

Addressing this problem by only adjusting communication or control might not be enough. Thus, this paper proposes to use a combined approach by co-designing resource allocation and mobile robot control for safety despite uncertainty. We limit ourselves to the single-robot scenario as a start, with the goal of using only as much resource as necessary, and to this end propose the architecture shown in Fig. 2. Here, several new components (in yellow) are integrated into an existing navigation framework (in blue). The robot is running navigation locally, and can update its position estimate $\hat{x}(t)$ between updates using odometry based on for example wheel encoders, represented by $\hat{x}(t), t \in (t_k, t_{k+1})$. When sensor measurements z are sent to the localization algorithm running on the edge, a corrected estimate $\hat{x}(t_{k+1})$ is received.

To prevent unnecessary communication, we introduce an optimization problem that chooses measurement frequency f and robot speed v subject to U_{req}, a requirement on uncertainty. This requirement comes from a safety criteria, i.e. how big of a localization uncertainty is acceptable based on the current estimate $\hat{x}(t)$. By also predicting what the future requirement will be along the path plan, the appropriate configuration can be chosen ahead of time to avoid violations when the requirement decreases faster than the localization uncertainty can adapt.

A. Related Work

Before proceeding, we review related work on allocating resources for offloaded localization as well as different approaches for modeling the uncertainty, and for dealing with it at the control level.

Allocation of limited network resources is a well-studied problem, and scheduling for communication resources is surveyed in [13]. With edge-computing, not only communication resources, but also storage and computational resources can be used, and different ways of scheduling one or more of these resources was surveyed in [14]. Some works have also looked at resource allocation specifically for offloaded localization. In [4], [5], optimal allocation strategies are presented to minimize the estimation error when measurements have to be sent over a network channel with limited capacity. For the related case of offloaded simultaneous localization and mapping, an adaptive method is presented in [15]. By adjusting communication rate based on the magnitude of the correction to the local position estimate, a desired performance is maintained while minimizing the data sent over the network.

However, even with perfect communication there will still be imperfections in localization which mobile robots can be handle by adjusting their control actions (e.g. change direction, reduce speed, etc.) Recent works have looked into characterizing the uncertainty for different kinds of localization, such as Lidar-based [16], vision-based [17], [7], [6], [8], [18] and cellular network-based localization [19]. Given a model of how the uncertainty affects the system, different approaches can then compensate either by adding robustness [7], [6], [8], [18] or by handling it probabilistically [10], [11], [12]. Particularly relevant for this paper is [12], which defines safety for a robot navigating based on global map-based localization with a 2D Lidar and a particle filter.

In the following, we define an uncertainty metric that we will relate to this definition of safety. While only considering a single robot, we take inspiration from [4] by phrasing the communication resource allocation problem as a choice between a fixed set of frequencies for the sensor. Instead of having the network resource as the hard constraint and using the estimation uncertainty as a cost however, we will use the uncertainty as the constraint and treat the communication resource as part of our cost that we want to minimize.

B. Contribution

The contribution of this paper is threefold:

- We define a measure of localization uncertainty for a particle filter belief state, and show that if we satisfy a requirement related to the obstacles in the map, this corresponds to satisfying a safety criteria.
- Using this measure, we generate a model of the expected uncertainty for different combinations of Lidar frequency and robot speed from experimental data.
- Finally, we formulate an optimization problem that chooses Lidar frequency and robot speed based on a user-specified cost, an uncertainty requirement and

the abovementioned model. We demonstrate the performance of this approach in experiments with both a simulated and a real robot.

II. Preliminaries

In this paper, the motion of the robot is modeled as a stochastic differential equation (SDE), and the observations are modeled as noisy discrete-time measurements:

$$\mathrm{d}x = F(x,u)\mathrm{d}t + \sigma(x)\mathrm{d}W \tag{1}$$

$$z(t_k) = h(x(t_k), \nu(t_k)), \;\; \nu(t_k) \sim p(\nu). \tag{2}$$

The state $x \in \mathcal{X} \subseteq \mathbb{R}^n$ evolves in continuous time according to the dynamics prescribed by F and the control input $u \in \mathcal{U} \subseteq \mathbb{R}^m$ and W which is a Brownian motion affecting the system through a diffusion term $\sigma(x)$. However, the state is unknown, and can only be observed indirectly using sensor measurements z that arrive at discrete time instances t_k and are subject to measurement noise $\nu(t_k) \sim p(\nu)$.

A particle filter is used to estimate the state x of the robot, which gives us a set of N weighted particles $X = \{(x_i, w_i)\}_{i=1}^N$, where w_i are normalized weights ($\sum_{i=1}^N w_i = 1$) for the hypothesis that $x = x_i$. To describe the dynamics of the particles, we introduce a piece-wise continuous belief state b as in [12]. At each measurement instance $t = t_k$ the particles are resampled according to the probability of observing z from x_i according to (2). Between measurements, the evolution can be described by propagating each particle in $b(t)$ through the system model (1):

$$\mathrm{d}b = F_b(b,u)\mathrm{d}t + \Sigma\tilde{W}, \;\; \forall t \in (t_k, t_{k+1}) \tag{3}$$

where $F_b(b,u)$ is a column vector corresponding to propagating each particle through F individually, $\Sigma = \mathrm{BD}(\{\sigma\}_{i=1}^N)$, i.e. a block diagonal matrix with σ on the diagonal, and $\tilde{W}$ is a Brownian motion of dimension $N \cdot \dim W$

Based on these dynamics, in [12] a risk-based Control Barrier Function (CBF) is introduced to guarantee safety of the belief state with respect to an obstacle that is to be avoided. While we are not designing a motion controller that guarantees safety by avoiding obstacles, we will use a similar definition of safety based on risk-measures, and thus introduce Value-at-Risk (VaR) and Conditional-Value-at-Risk (CVaR). Both measures assign a real value for a scalar random variable $y \in \mathcal{Y}$, with a probability density function $p(y)$ at a desired risk level $\alpha \in (0,1]$.

$$\mathrm{VaR}_\alpha(y) = \inf_{a \in \mathbb{R}} \{a | \Pr[y \le a] \ge \alpha\},$$

$$\mathrm{CVaR}_\alpha(y) = \mathbb{E}\{y | y \le \mathrm{VaR}_\alpha(y)\}.$$

The VaR is essentially the value of the percentile corresponding to α, while the CVaR corresponds to the mean value of the lower tail, i.e. the part of the distribution below the VaR. Note that both metrics can be computed also for the upper α-tail of the distribution by reversing the inequality signs and taking the $\sup$ instead of the $\inf$ when computing the VaR. To distinguish them, $\underline{\mathrm{VaR}}$ ($\underline{\mathrm{CVaR}}$) and $\overline{\mathrm{VaR}}$ ($\overline{\mathrm{CVaR}}$) will be used in the following when referring to the VaR (CVaR) of the lower and the upper tail respectively.

Since we are computing the risk measures for a discrete distribution we will use the empirical VaR and CVaR instead. For a discrete distribution $\{(y_i, w_i)\}_{i=1}^N$, where the y_i values are sorted in ascending order, the lower tail empirical risk-measures are defined as:

$$\underline{\mathrm{VaR}}_\alpha(\{(y_i,w_i)\}_{i=1}^N) = y_k,$$

$$\text{where } k = \operatorname*{arg\,min}_{1 \le j \le N} \sum_{i=1}^{j} w_i \ge 1-\alpha,$$

$$\underline{\mathrm{CVaR}}_\alpha(\{(y_i,w_i)\}_{i=1}^N) = \frac{\sum_{i=1}^k w_i y_i}{\sum_{i=1}^k w_i}.$$

Finally, we define the Euclidean Distance Field (EDF), which will be used to evaluate the safety of the state estimates. Formally, given a workspace $\mathcal{X}$ divided into free space $\mathcal{X}_{\text{free}}$ and obstacle space $\mathcal{X}_{\text{obs}}$ such that the boundary $\partial\mathcal{X}_{\text{free}}$ is piece-wise smooth, the EDF $d : \mathbb{R}^n \to \mathbb{R}_{\ge 0}$ is defined as

$$d(x) := \inf_{o \in \partial\mathcal{X}_{\text{free}}} ||x - o||_2, \tag{4}$$

with boundary condition $d(x) = 0$ when x is on the boundary of the free space, i.e. $x \in \partial\mathcal{X}_{\text{free}}$.

Using the EDF, we generate a *safety distribution* $\{(h_d(x_i), w_i)\}_{i=1}^N$, where $h_d(\cdot)$ is defined as the value of the EDF at the particle location, subtracted by a scalar safety margin γ that adds inflation for example to compensate for the size of the robot:

$$h_d(x_i) := d(x_i) - \gamma. \tag{5}$$

Risk-based safety at the risk-level α is then fulfilled if

$$\underline{\mathrm{CVaR}}_\alpha\left(\{(h_d(x_i), w_i)\}_{i=1}^N\right) \ge 0. \tag{6}$$

In the following, we will relate this safety requirement to a risk measure we define to quantify the localization uncertainty. We show that if the uncertainty can be kept below a required level based on the EDF, the safety requirement in Eq. 6 is also satisfied. This is achieved in two steps by solving the two problems presented next.

A. Problem Definition

To implement the new components in the framework shown in Fig. 2, there are two main problems that are addressed in this paper:

1) Given the safety requirement in (6), generate a requirement U_{req} such that the safety requirement is satisfied if the localization uncertainty stays below the requirement.
2) Create an uncertainty model $\Delta(f,v)$ so that choosing $(f,v) : \Delta(f,v) \le U_{\text{req}}$ also makes the localization uncertainty stay below the requirement.

Both problems will be addressed by defining a risk measure for localization uncertainty and relating it to the safety requirement. The relation between the two will be formally defined in the next section, but in Fig. 3 a visual interpretation is shown. Based on $d(\hat{x}(t))$ we prescribe a required uncertainty U_{req}, shown as the purple circle. We then pick f

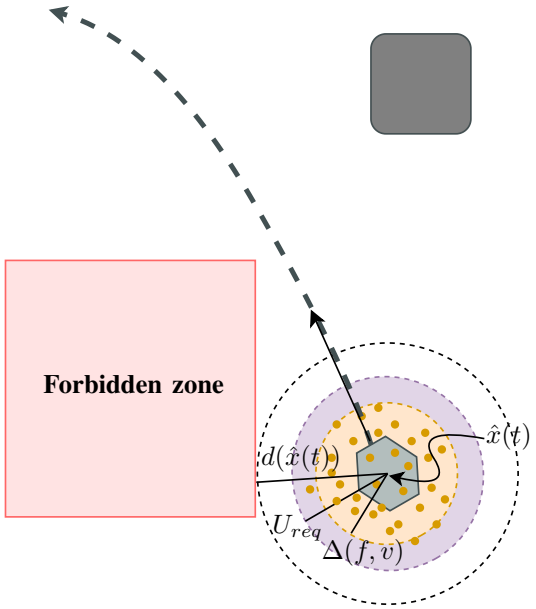

Fig. 3: Top-down view of the scenario in Fig. 1 illustrating how the problems are addressed in this paper.

and v such that the model value $\Delta(f, v)$, shown as the yellow circle, is lower than U_{req}. If the model is accurate enough, the uncertainty, which is based on the particles, shown as yellow spots, should then remain below the required level. Additionally, the chosen f and v should minimize a cost that rewards going as fast as possible and using as little frequency as possible.

Remark 1: Note that while we consider only communication frequency and navigation speed, other factors could also affect the localization uncertainty. In future work, we will look into other aspects that could also have an impact, such as communication delays, packet loss or rotational speed. In this paper we focus on the design of a method that jointly optimizes one variable related to communication and one variable related to control.

III. Risk-Aware Communication and Control Co-Design

To solve the abovementioned problems, we first define what localization uncertainty is. We then show how to generate the requirement U_{req} such that satisfying it also implies risk-based safety is fulfilled according to (6). We also describe how to create a model $\Delta(f, v)$ describing how localization uncertainty depends on f and v. This will finally lead to an optimization problem that chooses f and v such that the safety requirement in Eq. 6 is fulfilled.

A. Localization uncertainty and safety

To compute a risk measure for localization uncertainty, we will first need a scalar measure of the uncertainty. Given a weighted particle set $\{(x_i, w_i)\}_{i=1}^N$ we compute the *uncertainty distribution* $\{(\delta(x_i), w_i)\}_{i=1}^N$, where $\delta(\cdot)$ is defined as

$$\delta(x_i) := ||x_i - \hat{x}||_2, \tag{7}$$

where $\hat{x}$ denotes the estimated position of the robot (which in our case is the weighted sum of the particles, i.e. $\hat{x} = \mu(p) = \sum_{i=1}^N w_i x_i$). Computing the $\overline{\text{CVaR}}_\alpha$ of this distribution then gives us the localization uncertainty at a risk level α. This uncertainty will usually be the largest as $t \to t_{k+1}$, i.e. right before the new measurement update, since propagating the particles through the motion model (1) means integrating noise through the system dynamics. Receiving a new measurement $z(t_{k+1})$ then permits another measurement update, which could help reduce the uncertainty, before it starts increasing again.

We now propose that the localization uncertainty we defined can be used to ensure safety at the same risk level.

Proposition 1: Given a particle distribution $\{(x_i, w_i)\}_{i=1}^N$ with weighted mean $\hat{x}$, assume the particles are distributed such that $d(x_i) \geq d(\hat{x}) - \delta(x_i)$, $\forall i \in [1, N]$ and generate distributions $\{(\delta(x_i), w_i)\}_{i=1}^N$ and $\{(h_d(x_i), w_i)\}_{i=1}^N$. Now, if the localization uncertainty is upper bounded by $d(\hat{x}) - \gamma$, then safety is fulfilled, since $\overline{\text{CVaR}}_\alpha(\{(\delta(x_i), w_i)\}_{i=1}^N) \leq d(\hat{x}) - \gamma \implies \underline{\text{CVaR}}_\alpha(\{(h_d(x_i), w_i)\}_{i=1}^N) \geq 0$.

Proof: From the assumption we have that for each particle $h_d(x_i) = d(x_i) - \gamma \geq d(\hat{x}) - \gamma - \delta(x_i)$. The right side of the inequality generates another distribution $\{(d(\hat{x}) - \gamma - \delta(x_i), w_i)\}_{i=1}^N$, which is equivalent to shifting the uncertainty distribution $\{(\delta(x_i), w_i)\}_{i=1}^N$ by $-(d(\hat{x}) - \gamma)$ and then changing the sign of the new distribution. This means the assumption also implies that the safety measure is lower bounded by the safety measure at the mean subtracted by the localization uncertainty: $\underline{\text{CVaR}}_\alpha(\{(h_d(x_i), w_i)\}_{i=1}^N) \geq d(\hat{x}) - \gamma - \overline{\text{CVaR}}_\alpha(\{(\delta(x_i), w_i)\}_{i=1}^N)$. If the uncertainty $\overline{\text{CVaR}}_\alpha(\{(\delta(x_i), w_i)\}_{i=1}^N)$ is also upper-bounded by $d(\hat{x}) - \gamma$, finally we get $\underline{\text{CVaR}}_\alpha(\{(h_d(x_i), w_i)\}_{i=1}^N) \geq d(\hat{x}) - \gamma - \overline{\text{CVaR}}_\alpha(\{(\delta(x_i), w_i)\}_{i=1}^N) \geq 0$. ■

The proposition means that safety is verified if we can keep the localization uncertainty below $h_d(\hat{x})$, since this implies that the safety requirement in Eq. 6 is fulfilled. This will be used to generate the uncertainty requirement U_{req}, thus addressing the first problem mentioned in Section II-A.

The assumption we make is that the closest obstacle to any particle x_i is never closer than $\delta(x_i)$ to the mean $\hat{x}$ (i.e. $d(x_i) \geq d(\hat{x}) - \delta(x_i)$). This is reasonable as long as the robot is well-localized, since a violation of the assumption, i.e. $d(x_i) < d(\hat{x}) - \delta(x_i)$, would imply that at least one particle is on the opposite side of an obstacle compared to the mean of the particle distribution. That can happen when the robot starts to lose track of its position however, since the particle filter is designed to start spreading the particles more to handle re-localization if the robot was unknowingly moved to a new position.

B. Uncertainty modeling

To solve the second problem, we need to generate a model that, when used in the optimization problem to select

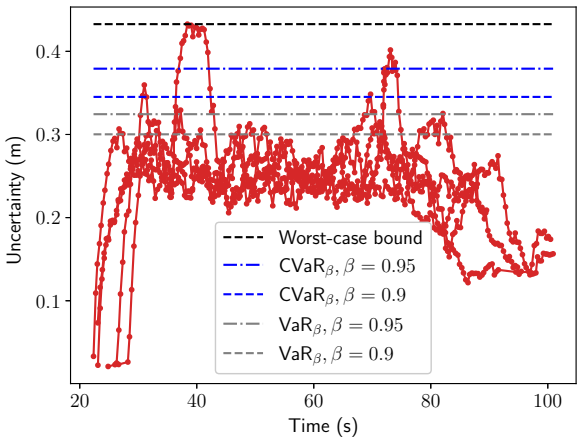

Fig. 4: Example of uncertainty model data for a single configuration and different candidates for the model value.

frequency f and speed v, results in a localization uncertainty low enough to satisfy the given requirement. If the requirement is generated as outlined above, this should then also result in satisfying Eq. 6 according to Proposition 1. In the following, we generate a model $\Delta(f, v)$ using data from localization experiments, which will then be included as a constraint when implementing an optimization problem that selects f and v given a requirement U_{req}.

To get the data for the uncertainty model, in the next section we will run navigation experiments for all possible combinations (f, v). Each time a measurement update is about to happen, we record what the values and weights in particle set were right before the update. For a given risk level α, the $\overline{\text{CVaR}}_\alpha(\{(\delta(x_i), w_i)\}_{i=1}^N)$ is then computed. Repeating this procedure for each iteration of the experiment results in one or more time series of the uncertainty values during the experiment. The experiments are further explained in the next section, but Fig. 4 shows an example of the $\overline{\text{CVaR}}$ over time from five different experiments using one configuration.

By collecting all the data points from each time series into a dataset, we get a set $\mathcal{D}$ with M points, $\mathcal{D} = \{\overline{\text{CVaR}}_\alpha(\{(\delta(x_i), w_i)\}_{i=1}^{N_j})\}_{j=1}^M$. Note that the size M will vary depending on the chosen configuration since a higher frequency means more measurement updates, which means saving the particle set p_j more often. Fig. 4 also shows the upper bound of this dataset, as well as the $\overline{\text{VaR}}_\beta$ and the $\overline{\text{CVaR}}_\beta$ computed at $\beta = 0.9$ and $\beta = 0.95$. We found that $\overline{\text{VaR}}_\beta$ computed at $\beta = 0.95$ results in a model that represents most of the data without being too sensitive to outliers, so for the remainder of the paper we will use this to generate uncertainty model $\Delta(f, v)$.

C. Uncertainty-aware optimization

The last step in addressing the second problem is to integrate the uncertainty model $\Delta(f, v)$ into an optimization problem that selects (f, v) from a set of available options $\mathcal{F} \times \mathcal{V}$, such that $\Delta(f, v) \leq U_{\text{req}}$. Below we present two versions, one reactive and one predictive.

Reactive optimization problem: The reactive optimization problem finds the optimal combination (f, v) with respect to a cost $C(f, v)$, that also (according to the model $\Delta(f, v)$) satisfies the requirement U_{req}

$$\min_{(f,v)\in\mathcal{F}\times\mathcal{V}} \; C(f, v),$$
$$\text{subject to } \; \Delta(f, v) \leq U_{\text{req}},$$

where according to Proposition 1 we choose $U_{\text{req}} = h_d(\hat{x}(t))$.

Predictive optimization problem: The reactive approach will always select the best option that satisfies the current requirement. This might not be enough if U_{req} changes quickly however, because while the model $\Delta(f, v)$ is generated using static combinations (f, v), it could take some time for the localization uncertainty to converge to a lower value when switching between different f and v. Thus, we also propose a predictive version of the method:

$$\min_{(f,v)\in\mathcal{F}\times\mathcal{V}} \; C(f, v),$$
$$\text{subject to } \; \Delta(f, v) \leq \hat{U}_{\text{req}}.$$

Here, $\hat{U}_{\text{req}}$ corresponds to a prediction of the lowest requirement $U_{\text{req}} = h_d(\hat{x}_t)$ within the coming τ seconds. We compute $\hat{U}_{\text{req}}$ by evaluating h_d in each point along the path plan and storing the lowest value, until the summed Euclidean distance between each pair of points exceeds the distance equivalent to the robot driving for τ seconds at max speed.

IV. Experiments

The method is evaluated in experiments with both a simulated and a real ABB Mobile YuMi® Research Platform robot. The robot is running the ROS2 navigation stack (Nav2 [20]) on its onboard computer, an Intel NUC with an eighth generation i7 CPU and 32GB of RAM. Nav2 comes with algorithms for mapping, localization, path planning and path following. In terms of sensors, the robot has wheels equipped with encoders that are used for local odometry estimation. The robot also has two SICK TIM781S Lidars, one facing forward and one facing backward. These are used both to create, and later to localize in, a map of the obstacles, using a particle filter implementation based on Adaptive Monte Carlo Localization (AMCL) [21].

A. Uncertainty modeling

We first run experiments to collect data for the uncertainty model $\Delta(f, v)$. We choose the following options for the optimization problems: $(f, v) \in [15\,\text{Hz}, 7.5\,\text{Hz}, 5\,\text{Hz}, 3.75\,\text{Hz}, 3\,\text{Hz}] \times [1.0, 0.75, 0.5, 0.25]$, where v acts as a scaling factor on the maximum allowed speed. While the Lidars nominally provides new measurements at 15 Hz, in Nav2 AMCL does not update the particle estimate every time a new measurement is received by default. It only performs a measurement update after the robot moved or rotated more than some threshold values.

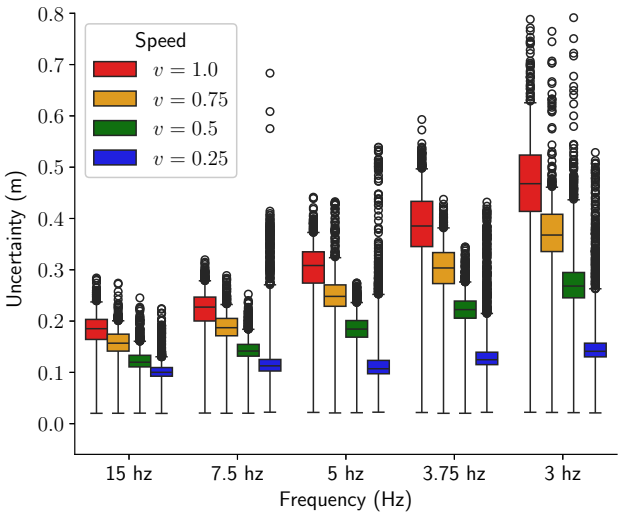

Fig. 5: Shows box plots generated from all the data collected for the uncertainty model grouped by frequency f and color coded by speed v.

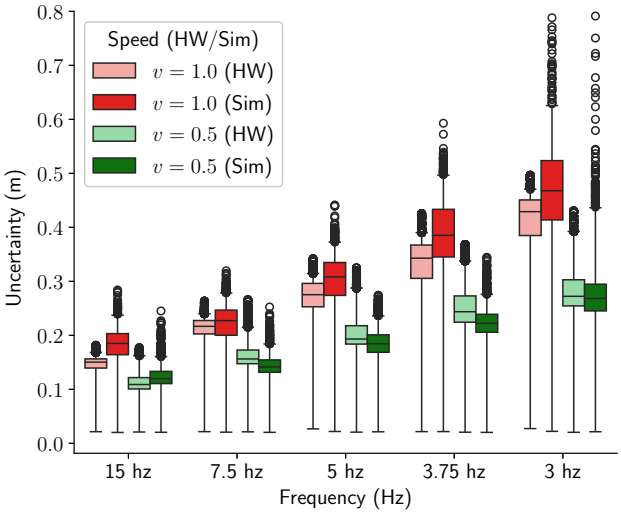

Fig. 6: Comparison with box plots of the data generated in simulations and with the real robot.

TABLE I: The modeled uncertainty values $\Delta(f, v)$ in m based on the $\overline{\mathrm{VaR}}_{0.95}$ of the data from the modeling experiments.

	15 Hz	7.5 Hz	5 Hz	3.75 Hz	3 Hz
$v = 1.0$	0.24 m	0.28 m	0.37 m	0.50 m	0.63 m
$v = 0.75$	0.20 m	0.23 m	0.32 m	0.38 m	0.46 m
$v = 0.5$	0.16 m	0.18 m	0.24 m	0.28 m	0.44 m
$v = 0.25$	0.13 m	0.27 m	0.25 m	0.22 m	0.26 m

In order to force it to update more often, we therefore use smaller distance and angular thresholds than the default values - $1\,\mathrm{cm}$ instead of $25\,\mathrm{cm}$ and $0.02\,\mathrm{rad}$ instead of $0.2\,\mathrm{rad}$ respectively.

1) Simulation experiments: We first collect uncertainty data in simulations. The robot performs 5 different trajectories in one environment using each configuration. The resulting dataset with $\alpha = 0.8$ is visualized using box plots in Fig. 5. The top whisker of the box plot is equivalent to the $\overline{\mathrm{VaR}}_\beta$ with $\beta = 0.95$, which will be used as the values in the model $\Delta(f, v)$. These values are also shown in Tab. I. In general, the results show that slowing down or increasing the frequency reduces the uncertainty. We had some issues with the odometry when the robot is rotating, which sometimes causes larger uncertainties even for the lower velocities. This is the reason why some model values for $v = 0.25$ is worse than for $v = 0.5$ at the same frequency.

2) Hardware experiments: While we do not create a model based on data from the real robot, we perform one experiment to validate that the model created in simulation is also representative for the real robot. We recorded data at $v = 1.0$ and $v = 0.5$ for one trajectory that covers most of the WARA Robotics lab. This is then replayed to AMCL at the different frequencies $f \in [15\,\mathrm{Hz}, 7.5\,\mathrm{Hz}, 5\,\mathrm{Hz}, 3.75\,\mathrm{Hz}, 3\,\mathrm{Hz}]$. The results are compared to the simulation data in Fig. 6. At full speed, the results from the robot ("$v = 1.0$ (HW)" in the plot) are lower than the model values based on simulation data (the upper whiskers of the data labeled with "$v = 1.0$ (Sim)" in the plot). At half speed however the uncertainty was slightly higher on the robot compared to the model values for some frequencies. This indicates that there are discrepancies between the model value and the uncertainties we will get on the robot for the same choice of f and v. One way to compensate could be to enable adapting the model to new data, but we leave this for future work.

B. Optimization problem evaluation

The model corresponding to the values in Tab. I is now used when implementing the two optimization problems. We run these at the nominal sensor rate $f = 15\,\mathrm{Hz}$ to select (f, v) while navigating the robot. For $h_d(x) = d(x) - \gamma$, we choose $\gamma = 0.5\,\mathrm{m}$, since the robot has a rectangular base that is $66\,\mathrm{cm}$ by $60\,\mathrm{cm}$, which means approximately $45\,\mathrm{cm}$ from the center to the corners. Furthermore, the inflation radius used for planning is set to $75\,\mathrm{cm}$ to keep the robot from going so close that no feasible solutions are found in the optimization problems.

1) Simulation experiments: We start by evaluating the two approaches in simulations by running experiments in another environment than the one used for modeling. For the cost function, $C(f, v) = 1 - v + \frac{f}{f_{\max}}$ is used, which equally penalizes slowing down and using more frequency. For the prediction horizon, we empirically determined $\tau = 5\,\mathrm{s}$ to be a good value that allows the uncertainty to reach a lower value in time when a new configuration is picked. Three different trajectories are repeated five times each for both optimization problems. As a comparison, we also repeat the experiments with three static configurations: the maximum frequency and velocity, the minimum frequency and velocity, and the cheapest option that satisfies the strictest requirement experienced during a test run of the same trajectory.

TABLE II: Test results from simulations using each method to select f and v. R-OP and P-OP correspond to the reactive and predictive optimization methods. S-H, S-L and S-C correspond to the highest, lowest and cheapest safe static configuration respectively. The values shown are the mean of the metric over five repetitions of each trajectory, where the best values are marked in bold.

	Trajectory 1			
	Average Cost	**Average v**	**Average f (Hz)**	**Violations**
R-OP	**0.21**	**1.0**	3.21	19.0
P-OP	0.23	**1.0**	3.39	**0.0**
S-H	1.0	**1.0**	15.0	**0.0**
S-L	0.95	0.25	**3.0**	**0.0**
S-C	0.33	**1.0**	5.0	**0.0**
	Trajectory 2			
	Average Cost	**Average v**	**Average f (Hz)**	**Violations**
R-OP	**0.28**	**1.0**	4.16	50.0
P-OP	0.29	**1.0**	4.40	**0.0**
S-H	1.00	**1.0**	15.0	**0.0**
S-L	0.95	0.25	**3.0**	**0.0**
S-C	0.50	**1.0**	7.5	**0.0**
	Trajectory 3			
	Average Cost	**Average v**	**Average f (Hz)**	**Violations**
R-OP	**0.26**	0.98	3.56	99.0
P-OP	0.31	0.98	4.22	4.8
S-H	1.0	**1.00**	15.0	**0.0**
S-L	0.95	0.25	**3.0**	**0.0**
S-C	0.75	0.75	7.5	**0.0**

For each trajectory, four scores are computed: average cost of the cost function, average value of v, average value of f and number of times the uncertainty requirement was violated. The requirement is evaluated 30 times per second by comparing the localization uncertainty of the particle state and with the requirement at that time. The results shown in Tab. II indicate that both of our optimization problems can satisfy the requirement most of the time. The reactive approach has the lowest average cost, but sometimes violates the safety requirement. At the price of a slightly higher cost on average, the predictive method almost always satisfies it, except for the third trajectory. In one iteration on this trajectory the predictive method did violate the requirement several times due to the uncertainty being higher than the model value. This indicates again that the model could benefit from being able to adapt to compensate for mismatch.

2) Hardware experiments: Finally, we implement the different optimization problems and baselines on the real robot and run one experiment using each method. The results, shown in Tab. III, again indicate that our approach can reduce the network usage without slowing down unnecessarily. In the predictive case this is achieved without violating the safety requirement. A more detailed comparison between the results from the reactive and the predictive method is shown in Fig. 7. As expected, the reactive method only increases the frequency and slows down as the requirement gets low, while the predictive approach is more proactive by design. Even if the predictions are not always perfectly aligned with the value of the requirement five seconds into the future, they help in avoiding where the reactive method did not.

TABLE III: Test results from hardware experiments. R-OP and P-OP correspond to the reactive and predictive optimization methods. S-H, S-L and S-C correspond to the highest, lowest and cheapest safe static configuration respectively. The best results for each metric are marked as bold.

	Average Cost	**Average v**	**Average f (Hz)**	**Violations**
R-OP	**0.32**	0.96	4.18	217.0
P-OP	0.41	0.93	5.09	**0.0**
S-H	1.0	**1.0**	15.0	17.0
S-L	0.95	0.25	**3.0**	**0.0**
S-C	1.75	0.25	15.0	**0.0**

V. Conclusions

In this paper, we investigate the problem of reducing network communication without compromising safety for a robot that relies on offloaded localization to correct drift in position estimates. We propose a two-step solution. First, we introduce a risk measure for localization uncertainty and relate it to a risk-aware safety requirement. We show that limiting the uncertainty also implies the safety requirement is fulfilled. We then generate an uncertainty model from experimental data, which enables implementing an optimization problem that adjusts the communication rate and the speed of the robot during runtime. We demonstrate the performance of this optimization problem in experiments both in simulation and with a real robot.

In future work, we want to extend the problem formulation to a multi-robot scenario. While the optimization problem keeps the communication from one robot to a minimum, the issue of congestion remains and has to be addressed if this approach is to implemented at scale. We would also like to investigate improvements for both the uncertainty model and the controller. The model could benefit from being able to adapt to new data to reduce the sim-to-real-gap. Moreover, we plan to extend the model to take continuous inputs for both frequency and speed. When it comes to the improvements for the controller, a natural extension is to consider angular speed. It would also be interesting to integrate a safety filter, such as a control barrier function, could help avoid violations when reducing the uncertainty is not enough. This could however make the robot deviate from the plan, making the requirement harder to predict.

Acknowledgment

We would like to thank Magnus Lindhé for his contributions to the first idea that eventually resulted in the method we proposed in this paper. We would also like to thank Sholeh Yasini for providing feedback and giving input on the paper. Finally, we would like to thank Matti Vahs for providing insights on both the code and the theory behind risk-based safety with particle filter localization.

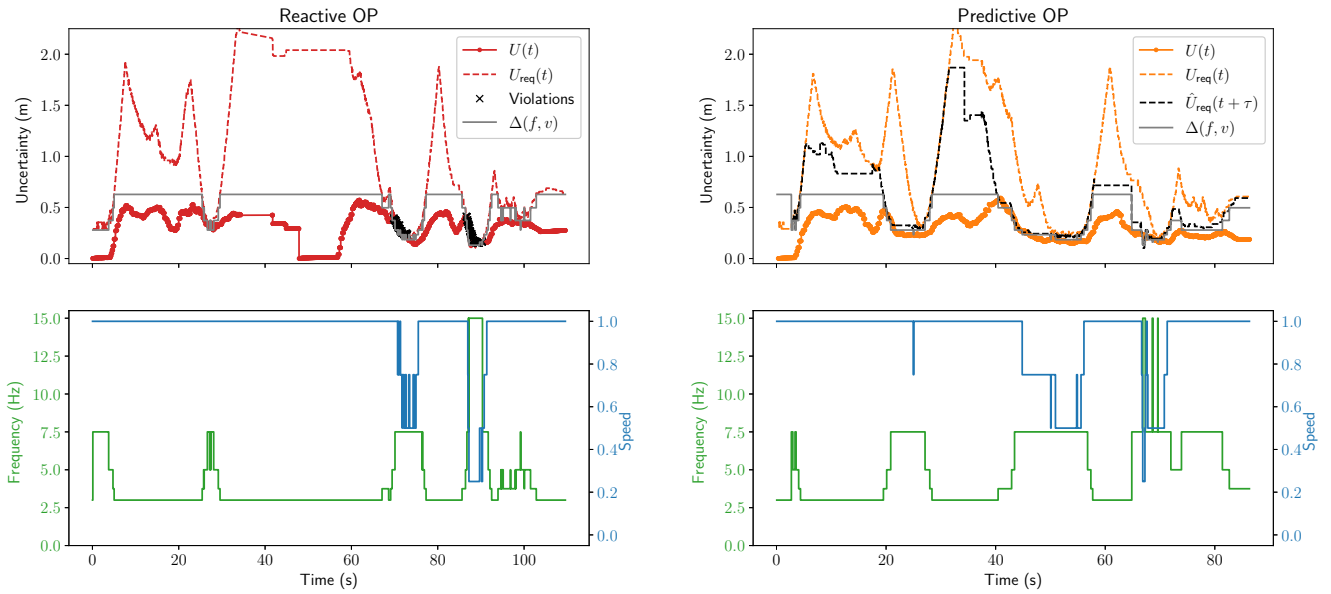

Fig. 7: Results from using the optimization problems (OPs) on the real robot. The upper plots show the $U_{\text{req}}(t)$ (dashed red or orange line) and the localization uncertainty $U(t)$ (solid red or orange line) over time. The model values $\Delta(f, v)$ (solid grey line) are also computed based on f and v, which are shown in green and blue respectively in the lower plot. Additionally, in the left plot violations of the safety requirement are marked as black crosses, and in the right plot $\hat{U}_{\text{req}}(t+\tau)$ (dashed black line) shows the strictest predicted requirement in the coming τ seconds.

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] ——, "Industrial 5G Edge Computing - Use Cases, Architecture and Deployment, White Paper," Feb. 2023. [Online]. Available: https://5g-acia.org/whitepapers/industrial-5g-edge-computing-use-cases-architecture-and-deployment/

[3] 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.

[4] X. Chen, M.-A. Belabbas, and T. Başar, "Optimal capacity allocation for sampled networked systems," *Automatica*, vol. 85, pp. 100–112, 2017.

[5] S. Mathew, K. H. Johannson, and A. Mahajan, "Optimal sampling of multiple linear processes over a shared medium," in *Conference on Decision and Control*. IEEE, 2018, pp. 1712–1718.

[6] 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.

[7] 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.

[8] 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.

[9] L. Lindemann, A. Robey, L. Jiang, S. Das, S. Tu, and N. Matni, "Learning robust output control barrier functions from safe expert demonstrations," *IEEE Open Journal of Control Systems*, vol. 3, pp. 158–172, 2024.

[10] A. Clark, "Control barrier functions for stochastic systems," *Automatica*, vol. 130, p. 109688, 2021.

[11] M. Vahs, C. Pek, and J. Tumova, "Belief Control Barrier Functions for Risk-Aware Control," *IEEE Robotics and Automation Letters*, vol. 8, no. 12, pp. 8565–8572, 2023.

[12] M. Vahs and J. Tumova, "Risk-aware Control for Robots with Non-Gaussian Belief Spaces," in *IEEE International Conference on Robotics and Automation*, 2024, pp. 11 661–11 667.

[13] M. Chitnis, P. Pagano, G. Lipari, and Y. Liang, "A survey on bandwidth resource allocation and scheduling in wireless sensor networks," in *2009 International Conference on Network-Based Information Systems*, 2009, pp. 121–128.

[14] Q. Luo, S. Hu, C. Li, G. Li, and W. Shi, "Resource scheduling in edge computing: A survey," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 4, pp. 2131–2165, 2021.

[15] K.-L. Wright, A. Sivakumar, P. Steenkiste, B. Yu, and F. Bai, "Cloud-slam: Edge offloading of stateful vehicular applications," in *2020 IEEE/ACM Symposium on Edge Computing (SEC)*, 2020, pp. 139–151.

[16] M. Marchi, J. Bunton, Y. Gas, B. Gharesifard, and P. Tabuada, "Sharp performance bounds for pasta," *IEEE Control Systems Letters*, vol. 7, pp. 2401–2406, 2023.

[17] Y. Gao, Y. Tang, H. Qi, and H. Yang, "Closure: Fast quantification of pose uncertainty sets," in *Robotics: Science and Systems 2024*, 07 2024.

[18] 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.

[19] S. Bartoletti, I. Palamà, L. Chiaraviglio, S. M. Razavi, Y. Zhao, G. Bianchi, and N. Blefari-Melazzi, "Positioning integrity via uncertainty quantification," *IEEE Transactions on Vehicular Technology*, pp. 1–14, 2024.

[20] 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.

[21] D. Fox, "Kld-sampling: Adaptive particle filters," *Advances in neural information processing systems*, vol. 14, 2001.