Negotiating Visibility for Safe Autonomous Navigation in Occluding and Uncertain Environments

Jacob Higgins and Nicola Bezzo

***Abstract*—Navigation through an occluded environment is a challenging task for autonomous mobile robots (AMR), since they must balance both safety and speed in an attempt to fluidly steer around occlusions in uncertain environments. This is because real world environments have dynamic actors that may be occluded to the robot during motion, introducing uncertainty. One key element of eliminating this uncertainty is moving in such a way to maximize perception around these occlusions. This letter presents a novel control framework that combines both perception and safety constraints, resulting in motion that is quick and safe when occlusions are present. Perception is satisfied using a model predictive control (MPC)-based approach to provide inputs that increase visibility around occlusions while safety is promoted by modeling uncertainties as projected probabilities of occupancy derived from current observation and expected traffic motion. Improvements in visibility, safety, and speed are shown in simulations and are experimentally validated using an unmanned ground vehicle.**

***Index Terms*—Motion planning, mobile robots, optimal control.**

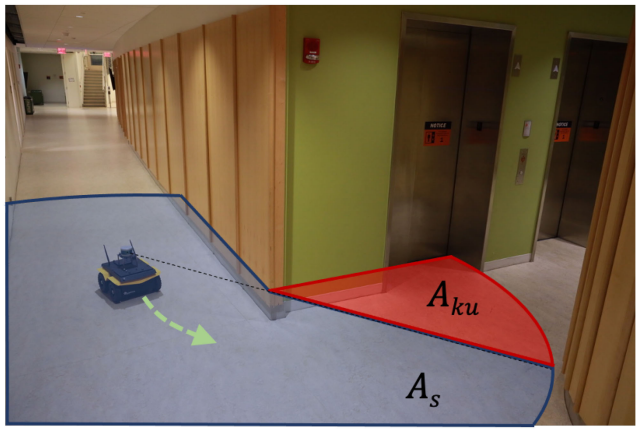


Fig. 1. Pictorial representation of the problem covered in this work. A_s represents the visible region while A_{ku} is the "known-unknown" not visible sensed area by the robot

I. Introduction

OVER the years, robots have carved a bigger space in our every day lives, and the current state of the world has only accelerated this pace. The on-going pandemic has fueled interest in minimizing human-to-human contact in many different sectors using autonomous mobile robots (AMR), from ride-sharing to food/drug delivery. As AMR are introduced in more indoor spaces shared with other actors (e.g., humans, other robots), one hurdle they must overcome is to guarantee safety while promoting speed in typically cluttered occluding environments. Indeed, when one sees videos of deployed AMR in action, it is immediately apparent how slow they are programmed to move through the environment, presumably due to the kinds of uncertainties that traversing a co-human environment presents.

When human beings are tasked to plan a route through a cluttered environment, visibility is a key factor in their decision-making process. Although the exact path is different for each person, there are several factors that most consider:

- How much of the path ahead is visible?
- How likely is a collision with an obstacle that is not currently visible?
- What is a reasonable balance between moving to increase visibility and cutting corners to decrease travel time?

Humans answer these questions with split-second decision making informed by heuristic reasoning and experience, but translating these sensibilities into an algorithmic framework for AMR is a more challenging problem.

Consider a robot deployed in a complex environment, approaching busy intersections of hallways and aiming to take a turn in the presence of square walls as depicted in Fig. 1. A naive navigation approach that has the robot stopping at every intersection would be at best safe but extremely inefficient. However, if the robot could quantify the limits of its perception based on surrounding occlusions, then it could attempt to reduce the unknowns by taking mitigating actions like reducing its speed and adjusting its trajectory to take a wider angle when approaching the corner.

To solve these challenges, we propose a control framework that not only satisfies both safety and perception constraints, but also uses both of these objectives to inform the importance of the other in an intuitive and natural way. This control framework captures the complex notions of safety and visibility, yet is kept simple so that it may be applied in a variety of contexts.

This work presents two main contributions: 1) the formulation of an analytic perception objective that, when used in the cost function of an optimal control problem, results in motion that increases perception around occlusions, and 2) an occupancy grid-based technique to determine if the robot can safely

Manuscript received October 15, 2020; accepted February 20, 2021. Date of publication March 24, 2021; date of current version April 9, 2021. This letter was recommended for publication by Associate Editor J.-M. Lien and Editor N. Amato upon evaluation of the reviewers' comments. This work was supported in part by the DARPA under Contract FA8750-18-C-0090, and in part by the NSF under Grant #1823325. *(Corresponding author: Jacob Higgins.)*

The authors are with the Department of Electrical and Computer Engineering, University of Virginia, Charlottesville, VA 22903 USA (e-mail: jdh4je@virginia.edu; nbezzo@virginia.edu).

This letter has supplementary downloadable material available at https://doi.org/10.1109/LRA.2021.3068701, provided by the authors.

Digital Object Identifier 10.1109/LRA.2021.3068701

navigate around the occlusion in the presence of traffic from other moving actors.

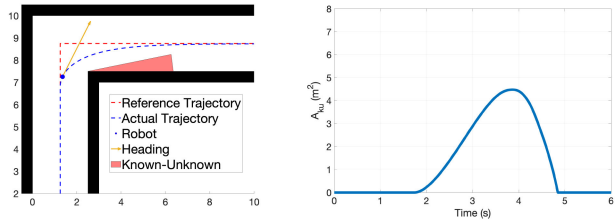
The rest of the paper is organized as follows: in Section II we provide an overview of the current state of the art in motion planning. In Section III we formally define the problem of motion planning under safety and perception constraints. Section IV presents our policy framework that can safely navigate in occluded and uncertain environments which is tested with both simulations and experiments in Sections V and VI, respectively. Finally, we draw conclusions and discuss future work in Section VII.

II. Related Work

Motion planning is one of the most studied problem in robotics [1], and many different approaches have been developed over the years [2]. When the environment is completely known, there are a variety of techniques that can generate paths around obstacles [3], [4]. In many real world scenarios, however, the presence of occlusions means complete knowledge of the environment cannot be guaranteed. To address this, much research over the past decade has focused on motion planning that emphasizes visibility throughout the trajectory. For example, [5] plans a UAV trajectory that minimizes snap while avoiding obstacles and keeping a target object within the field of view of an attached camera. If the optimization is performed as part of an online MPC, similar work can be found in [6] where the authors use a pinhole camera model to include camera image dynamics inside the model of the MPC, keeping objects of interest at the center of the camera while moving. [7] achieves the same goal, but instead uses a neural network to predict the motion of points of interest on the camera image, reducing computational complexity. Visibility of multi-UAV systems has been explored in [8], where their framework is inspired by the coordinated motion of the human eye. Work has also been done on observing the surface area of large 3-dimensional objects that cannot be inspected entirely with a single camera image [9]. Authors in [10] even constructed a specialized unmanned ground vehicle (UGV) for the purpose of classifying objects that may be partially or totally occluded.

Despite this volume of work, less progress has been made on addressing the inherent uncertainty that occlusions introduce. [11] presents a formal discussion on optimal path planning when obstacles in the environment are unknown due to the possibility of occlusion by other obstacles. Their discussion, however, is limited only to static obstacles and does not consider dynamic obstacles that may appear from behind the occlusions. In the context of autonomous transportation vehicles, [12] computes the reachability sets of cars that may be hidden behind occlusions to avoid possible collision states. These reachable sets assume the occluded vehicles follow reasonable rules for cars obeying traffic laws, but do not generalize to other non-transportation settings.

In contrast with this previous work, our proposed framework considers occlusions to be a source of uncertainty, especially when dynamic obstacles may appear from behind the occlusions. This uncertainty triggers the framework to increase visibility, and in turn the increased visibility serves to reduce uncertainty. To the best of our knowledge, this is the first work formally addressing these concerns.



(a) (b)

Fig. 2. Baseline simulation with highlighted A_{ku} over time in (b).

III. Problem Formulation

In this work we are interested in finding a control policy for a robot to negotiate occlusions while considering visibility, safety, and speed constraints. The framework should be able to apply to any general autonomous mobile system, as well as be able to generalize to any occluded environment.

There are many ways of approaching this multi-faceted problem. Our approach decouples this research question into two sub-problems whose solution strongly affects the overall motion of the autonomous system. Formally, these two problems can be defined as follows:

Problem 3.1: *Safe Navigation* **:** A robot must be able to avoid collision with any obstacle within its visibility range at any time over a time horizon t_T, or mathematically:

$$||\boldsymbol{p}(t') - \boldsymbol{o}_i|| > 0, \forall t' \in [t, t + t_T], \forall i \in [1, n_o] \tag{1}$$

in which $\boldsymbol{p}(t) = [x, y]^\intercal$ is the position of the robot, $\boldsymbol{o}_i(t) = [o_x, o_y]^\intercal$ is the position of the i^{th} obstacle in the $x - y$ plane at time t, and n_o is the number of obstacles.

In the context of this work, this problem implies that the robot must be able to stop and avoid a collision with actors and obstacles that are occluded, and hence may suddenly appear in in the field of view (FOV) of the vehicle.

Now, consider the case depicted in Fig. 1 that shows a situation that an autonomous robot will most likely encounter while moving through an indoor setting. The shaded area corresponds to the robot's (FOV) $F(t)$, while $A(t) \in F(t)$ is the traversable area that is not occupied or blocked by obstacles. The portion of $A(t)$ visible to the robot is defined by $A_s(t) \in A(t)$, while the area that is traversable but not visible – which we call the "known-unknown" region – is denoted as $A_{ku}(t) = A(t) - A_s(t)$. The second problem that we propose to solve in this work is then as follows:

Problem 3.2: *Minimizing Known-unknown Occlusions:* Given the safety constraint defined in Problem 3.1, find a control policy $\mathcal{P}_u$ which at runtime maximizes visibility in an occluded environment, or equivalently minimizes $A_{ku}(t)$:

$$\mathcal{P}_u(t) = \arg\min_{\boldsymbol{u}} A_{ku}(t), \forall t \tag{2}$$

where $\boldsymbol{u}$ is the commanded input to the robot.

A. Running Example: Two-Corridor Hallway

To better explain and guide the reader through the proposed approach, throughout the paper we will use the general occluded environment shown in Fig. 2 characterized by 90° corners, typical of most environments that surround us. Additionally, the robot model for the proposed running example is given by the

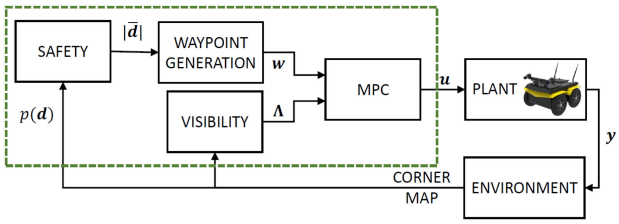


Fig. 3. Diagram showing the different components of the proposed controller. The contributions of this letter are within the green box.

following double integrator dynamics:

$$\ddot{\boldsymbol{p}} = \boldsymbol{u} \tag{3}$$

where $\boldsymbol{u} = (\ddot{x}, \ddot{y})^\intercal$ is the input acceleration vector.

Since this work is focused on motion planning in the presence of potentially occluded dynamic obstacles, we assume that the shape, size and location of the corridors are detectable by onboard range/vision sensors or known to the robot a priori, reflecting the situation in which the robot is already familiar with a particular building.

In Fig. 2 we show a baseline simulation for the robot following a trajectory using a pure-pursuit approach [13] through the proposed environment. Fig. 2(a) shows the entire trajectory of the robot, and Fig. 2(b) shows the known-unknown area over time.

IV. Approach

The diagram in Fig. 3 shows the architecture of our framework. At the core, a model predictive controller (MPC) is chosen because it can predict future values of an objective function and work to minimize these future values. In the context of this work, an MPC can recognize when the visibility will be negatively affected in future states and proactively and optimally maneuver to mitigate such occlusion. MPC can also be fed state constraints on the system that must be respected in the state prediction horizon. These constraints are used extensively to promote safe motion.

A pure pursuit-based module is included to dynamically generate waypoints over the desired trajectory for the robot. This module both guarantees progress in the navigation toward the end goal and includes safety constraints when influencing the motion of the robot. An equally important feature of the proposed controller is the ability to incorporate uncertainties of the upcoming hallway into its motion via a probabilistic occupancy grid-based approach, slowing down the vehicle in the presence of upcoming uncertainties and maintaining/increasing its speed when the path ahead is seen (or known a prior) to be clear. For ease, in this work we assume that a desired trajectory is built and known a priori. How to build trajectories is not the scope of this letter and our proposed technique complements any trajectory generation procedure (e.g., visibility graph, virtual potential field) by modifying the path to increase visibility and safety.

In the next sections, we provide details about each of these modules that make up the proposed controller depicted in Fig. 3, starting with the MPC formulation for perception to minimize the known-unknown areas at runtime.

A. MPC-Based Known-Unknown Perception Minimization

In this work, we are interested in designing a single model predictive controller (MPC) framework that negotiates occluded intersections of varying shapes and sizes.

MPC operates on solving an online optimal control problem (OCP) that optimizes over a moving prediction horizon. This OCP requires two main components: (1) a cost function J that the OCP minimizes at each time step, and (2) feasibility regions that the OCP must respect. The cost function J at time t for our specific problem is expressed as:

$$\begin{aligned} J = {} & (\boldsymbol{x}_{t+T} - \boldsymbol{w}_t)^\intercal \boldsymbol{Q}_{t+T}(\boldsymbol{x}_{t+N} - \boldsymbol{w}_t) \\ & + \sum_{i=1}^{T-1} (\boldsymbol{x}_{t+i} - \boldsymbol{w}_t)^\intercal \boldsymbol{Q}_{t+i}(\boldsymbol{x}_{t+i} - \boldsymbol{w}_t) + \boldsymbol{u}_{t+i-1}^\intercal \boldsymbol{R}\boldsymbol{u}_{t+i-1} \\ & + w_\Lambda \Lambda^2(x_{t+i}, y_{t+i}). \end{aligned} \tag{4}$$

where $\boldsymbol{x}_i$ is the state space vector for the ith prediction step, $\boldsymbol{w}_t$ is the reference set in state space, and $\boldsymbol{u}_i$ is the ith control input that the MPC computes at each sampling period. $\boldsymbol{Q}$ and $\boldsymbol{R}$ are the cost weighting matrices for state space position and control input reference tracking, respectively.

The OCP is then formulated as:

$$\begin{aligned} \underset{\boldsymbol{u}_0,\ldots,\boldsymbol{u}_{N-1}}{\arg\min} \quad & J(\boldsymbol{x}_0, \boldsymbol{u}_0, \ldots \boldsymbol{u}_{N-1}) \\ \text{subj. to} \quad & \boldsymbol{u}_{t+i} \in \mathcal{U}_t, \quad \forall i = [0, N-1] \\ & \boldsymbol{x}_{t+i} \in \mathcal{X}_t, \quad \forall i = [1, N] \end{aligned} \tag{5}$$

where $\boldsymbol{x}_{t+i}$ is the model-based predicted state at future time $t + i \times \Delta t$, Δt is the sampling time and T is the prediction horizon over N steps. The feasibility regions for the control inputs and state-space variables are denoted as $\mathcal{U}_t$ and $\mathcal{X}_t$, respectively. In this letter, control inputs are restricted by simple min-max inequalities $\boldsymbol{u}_{\min} \leq \boldsymbol{u} \leq \boldsymbol{u}_{\max}$. The feasibility region for the position component of the state is denoted by $\mathcal{P}_t$ and defined by an H-Polyhedron:

$$\mathcal{P}_t = \{\boldsymbol{p} \in \mathbb{R}^2 : A\boldsymbol{p}(t) \leq b\} \tag{6}$$

where the matrices A and b that define $\mathcal{P}_t$ depend on the position of the robot $\boldsymbol{p}(t)$, as well as the width of the current corridor w_0 and the width of the next corridor w_1:

$$A = \begin{pmatrix} -1 & 0 \\ 0 & 1 \\ 1 & -y/x \end{pmatrix}; \quad b = \begin{pmatrix} w_0 \\ w_1 \\ 0 \end{pmatrix} \tag{7}$$

Motion through the environment is produced by a pure pursuit approach of a waypoint $\boldsymbol{w}_t$ that changes over time. Details on how this waypoint is chosen will be discussed in Section IV-B.

The novelty about our MPC framework is the inclusion of a new term $\Lambda(x, y)$ in (4), which defines the perception objective. The purpose of the perception objective $\Lambda(x, y)$ is to value the current position based on this occluded area.

An exact analytical expression for the known-unknown area $A_{ku}(t)$ of an occluded environment is in general a piecewise continuous function that is not differentiable and computationally intractable. Instead, this work finds use in a simple analytical expression that closely correlates to $A_{ku}(t)$. This analytical expression is defined by distances Δx and Δy relative to the occluding corner, shown in Fig. 4. The angle between

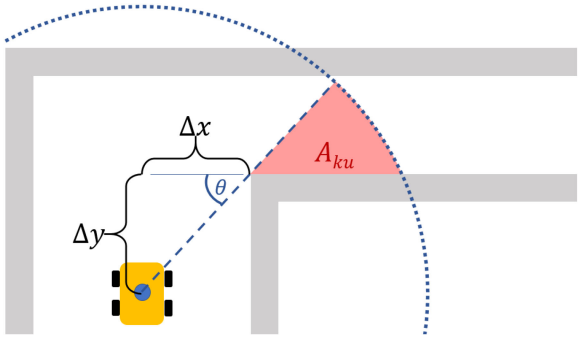
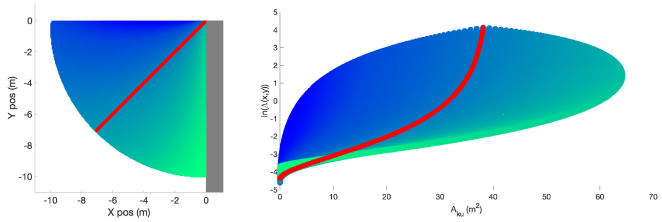


Fig. 4. Defining the perception objective in terms of the occluding corner, relative to the current location of the robot.



(a) (b)

Fig. 5. Mapping points in the $x-y$ plane that are occluded by a corner (a) to values of the known-unknown area $A_{ku}(x,y)$ and the logarithm of the perception objective $\Lambda(x,y)$ (b).

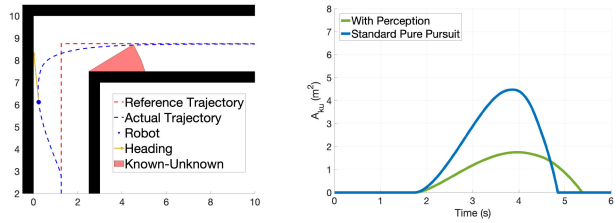
the position of the robot and the upcoming corridor is defined as $\theta = \arctan(\Delta y/\Delta x)$. With these quantities, the perception objective is defined as:

$$\Lambda(\Delta x, \Delta y) = \frac{\theta}{\Delta y} = \frac{\arctan(\Delta y/\Delta x)}{\Delta y} \tag{8}$$

In other words, the perception objective is defined as the ratio between the angle θ and the parallel distance Δy. This expression has two main properties that make it appealing to use as a perception objective:

- $\lim_{\Delta y \longrightarrow \infty} \Lambda(\Delta x, \Delta y) = 0$, meaning that this perception objective naturally tends to zero if the robot is far away from the occluding corner. In this case, the robot's motion will not be affected by the perception objective.
- $\lim_{\Delta y \longrightarrow 0, \Delta x \neq 0} \Lambda(\Delta x, \Delta y) = 1/\Delta x$, meaning that as the robot approaches the occluding corner, the perception objective takes on the value $1/\Delta x$ and the MPC tries to increase Δx in order to minimize the perception objective. By increasing Δx, the occluded area is naturally minimized.

We will now show that the analytical expression in (8) mimics the desired behavior that an ideal perception objective would have and provides a differentiable cost function that any nonlinear MPC can handle. Fig. 5 shows a mapping between spatial points occluded by an upcoming corner, the desired value $A_{ku}(x,y)$ and perception objective $\Lambda(x,y)$. The values of $A_{ku}(x,y)$ are considered the "ground truth" that the analytical perception objective $\Lambda(x,y)$ is approximating. The quantity $\log(\Lambda(x,y))$ is plotted against $A_{ku}(x,y)$ to address the non-linear relationship between them. Highlighted in Fig. 5 is a red line of locations along which $\ln(\Lambda(x,y))$ and $A_{ku}(x,y)$ share a correlation of 0.96, implying a strong relationship between the two values. In other words, as the MPC decides to increase/decrease the perception objective as



(a) (b)

Fig. 6. Effect of perception objective on the robot motion and comparison of A_{ku} with the baseline case in Fig. 2.

defined in (8), this correlates to an increase/decrease in the known-unknown area.

Analyzing the effects of the perception constraint in (4), we note that a large weight w_Λ causes the MPC to return commanded inputs whose overall effect is to provide a better vantage down occluded hallways. Fig. 6 demonstrates this effect, in which the A_{ku} is significantly reduced from the baseline case in Fig. 2 when approaching the occlusion.

1) Stability Properties: One important observation that can be made is that since the MPC balances the two possibly conflicting objectives of reference tracking and maximizing perception, stability of this controller cannot be cast as a reference tracking problem simply because the minimum of the cost function will *not lie on the reference, but at a point between the reference and maximizing perception*. Instead, stability is defined in the following sense:

Definition IV.1: Let $\boldsymbol{x}_m(t)$ be a minimum of the cost function defined in (4) at time t. Suppose $\boldsymbol{x}_m(t') = \boldsymbol{x}_m(t) = \boldsymbol{x}_m, \quad \forall t' > t$. A controller is defined to be stable if it can guarantee $\boldsymbol{x}(t') \longrightarrow \boldsymbol{x}_m$.

In other words, the controller is said to be stable if the minimum $\boldsymbol{x}_m(t)$ is asymptotically stable at every instant t. Note that this does not mean the robot will ever reach $\boldsymbol{x}_m(t)$, since the minimum is always changing (i.e., the tracked reference point is constantly changing over time).

The rest of this section will focus on supporting the following lemma:

Lemma 1: The optimal control policy defined in (5) results in asymptotically stable motion that approaches the minimum of the cost function (4).

Proof: In order to show this, only the position dependence of the cost function $J(\boldsymbol{x})$ is considered. As the higher derivatives of position, e.g., $\dot{\boldsymbol{p}}$ and $\ddot{\boldsymbol{p}}$, are quadratic in $J(\boldsymbol{x})$ with positive coefficients, it is clear that the minimum of $J(\boldsymbol{x})$ corresponds to a system that has no motion (any non-zero motion only adds to the cost function). What needs to be explained is how equilibrium values of (x_{eq}, y_{eq}) are affected by the addition of the perception objective $\Lambda(x,y)$ defined in (8). To this end, define this $x-y$ dependent portion as $J_{xy}(x,y)$:

$$J_{xy}(x,y) = w_x(x-x_w)^2 + w_y(y-y_w)^2 + w_\Lambda \Lambda^2(x,y) \tag{9}$$

In order to prove Lemma 1, it must be demonstrated that the minima of J_{xy} are well-defined at all times. One simple approach is to solve where the gradient of the cost function vanishes. The gradient of the cost function can be shown to be:

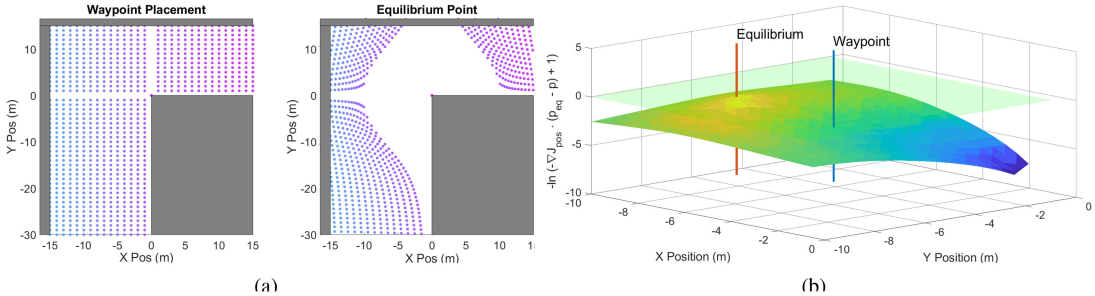


Fig. 7. Establishing stability of the MPC controller defined in (5). Fig. 7(a) shows the relationship between the placement of the waypoint $\boldsymbol{w}$ and the minimum value (i.e. equilibrium value) of the spatial portion of the cost function $J_{xy}(\boldsymbol{x})$. Fig. 7(b) shows a numerical argument for why this system exhibits stability. Also shown is the waypoint location at $\boldsymbol{w} = (-5, -5)^\intercal$ and resulting equilibrium location $\boldsymbol{p}_{eq} = (-7.2, -6.3)^\intercal$.

$$\frac{\delta J_{xy}}{\delta x} = (x - x_w) - \frac{\arctan(y/x)}{x^2 y(y^2/x^2 + 1)}$$

$$\frac{\delta J_{xy}}{\delta y} = (y - y_w) - \frac{\arctan^2(y/x)}{y^3} + \frac{\arctan(y/x)}{xy^2(y^2/x^2 + 1)}$$

where $\boldsymbol{w} = (x_w, y_w)$ is the location of the waypoint tracked by the robot at position $\boldsymbol{p} = (x, y)$. Setting these equations to zero results in transcendental equations, which cannot be solved analytically. Instead, numerical methods may be used to find minima; Fig. 7(a) shows the results of such an analysis. This minimum is referred to as the equilibrium point $\boldsymbol{p}_{eq}$ of the MPC, as it is the point at which cost cannot be minimized anymore, and so no motion is commanded by the MPC. It is clear from this figure how the perception objective causes the MPC to steer away from the corner and gain a better vantage around the occluding corner. For any point $\boldsymbol{p}$, the MPC finds a policy that minimizes the cost function $J(\boldsymbol{x})$ at each step over the prediction horizon. Stability is thus achieved when the MPC selects a policy that moves the system towards $\boldsymbol{p}_{eq}$ at all times. Assuming it is always possible to choose such a policy, in order to prove stability it is sufficient to show that movement towards $\boldsymbol{p}_{eq}$ directly minimizes the cost function at all points in the $x - y$ plane. For an $x - y$ point $\boldsymbol{p}$, this condition is true if and only if the following inequality holds:

$$\nabla J_{xy}(x, y)^\intercal \cdot (\boldsymbol{p}_{eq} - \boldsymbol{p}) < 0.$$

Here, the vector $(\boldsymbol{p}_{eq} - \boldsymbol{p})$ points from the $x - y$ position $\boldsymbol{p}$ to the MPC equilibrium position $\boldsymbol{p}_{eq}$. The above inequality asserts that motion toward $\boldsymbol{p}_{eq}$ must always decrease the cost function. Fig. 7(b) shows this to be true numerically for certain choices in waypoint placement and relative perception weight w_Λ. The green plane corresponds to $\nabla J_{xy} \cdot (\boldsymbol{p}_{eq} - (x, y)^\intercal) = 0$, meaning that since the projection remains below this plane, it stands to reason that moving towards the equilibrium position always results in decreasing J_{xy}. Assuming such a policy is feasible, this implies the MPC will always choose to move towards the equilibrium position. □

B. Waypoint Generation

As mentioned in Section IV-A, the MPC is given a waypoint $\boldsymbol{w}(t)$ that serves as a pure pursuit objective for the robot to follow. In the proposed framework, safety is enforced by adapting $\boldsymbol{w}(t)$ depending on the uncertainty of the upcoming environment.

Section IV-B1 discusses how uncertainty is modeled in our framework, followed by Sec. IV-B2 that discusses waypoint placement. Note that this work assumes some prior knowledge about the expected traffic of other actors in an environment. This notion can be acquired from prior experience/assumptions. If no prior knowledge is available, and safety is paramount, one would select the uncertain case discussed below.

1) Expected Distance to Collision: In the presence of uncertainty due to possible incoming traffic of actors from around occlusions, safety is determined through probabilistic means by computing the average distance until collision. In order to calculate this expectation value, one must know the probabilities of collision for different locations in space at particular instances in time. While this may seem unwieldy at first, such a task becomes more manageable by making the following connection: *the probability of colliding with an object at a given point in space is the same as that location's probability of occupancy*. Occupancy grids are a common and well-studied approach for mapping [14], and readily give the probability of occupancy for any grid location. The proposed framework borrows this idea of occupancy-for-mapping and instead applies it towards calculating the likelihood of collision in an occluded hallway.

In choosing a waypoint $\boldsymbol{w}$ in an uncertain environment, it is beneficial to have a function that directly connects to the safety of $\boldsymbol{w}$. In this framework, this function is the expected distance to collision beyond the corner and is calculated using estimated future probabilities of occupancy in the upcoming section of the hallway.

Define $\Delta\boldsymbol{d} = \boldsymbol{p} - \boldsymbol{w}$, and divide $\Delta\boldsymbol{d}$ up into n sufficiently small line segments $\boldsymbol{d}_i = \boldsymbol{p} + \frac{i}{n}\Delta\boldsymbol{d}, i = 1, \ldots, n$. Additionally, let $\mathcal{S}_{\text{next}} \in \mathbb{R}^2$ define the area of the upcoming hallway. Define the set $\mathcal{D}$ as,

$$\mathcal{D} = \{\boldsymbol{d}_i : \boldsymbol{d}_i \in \mathcal{S}_{\text{next}}\} \tag{10}$$

In other words, $\mathcal{D}$ contains all points $\boldsymbol{d}_i$ that lie beyond the occluding corner.

An occupancy grid map takes as input the $x - y$ position in space and returns the probability that a region is occupied by an obstacle. At each sampling time, this occupancy grid is updated by what is observed by the robot. Define $p_t(\boldsymbol{d}_i)$ as the current probability function, with $p_t(\boldsymbol{d}_i) = 1$ representing the knowledge that $\boldsymbol{d}_i$ is occupied by an obstacle with certainty. In order to address situations with dynamic obstacles (e.g., a person walking into view from around the corner), future occupancy $p_{t+\Delta t}(\boldsymbol{d}_i)$ at some time Δt later is predicted from $p_t(\boldsymbol{d}_i)$ via convolution with a probabilistic motion model. For ease of discussion, the probabilistic motion model is assumed to be a simple uni-directional motion down the corridor. The parameter

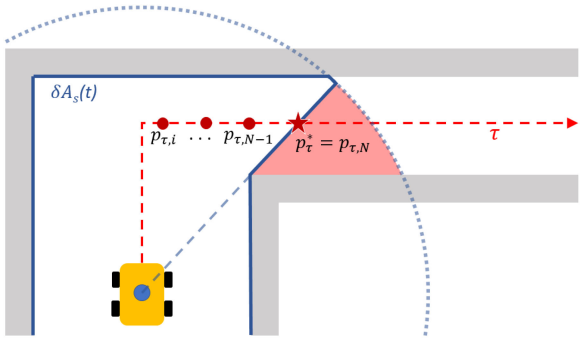


Fig. 8. Waypoint placement in safe and unsafe situations.

Δt is estimated as the distance to $\boldsymbol{w}$ divided by the speed the robot is currently moving in that direction:

$$\Delta t = \frac{|\boldsymbol{d}|}{\dot{\boldsymbol{p}} \cdot \hat{\boldsymbol{d}}^{\mathsf{T}}} \tag{11}$$

The expected location of collision $\bar{\boldsymbol{d}}$ can be defined as:

$$\bar{\boldsymbol{d}} = \sum_{i \in \mathcal{D}} \boldsymbol{d}_i p_{t+\Delta t}(\boldsymbol{d}_i) \prod_{j=1}^{i} [1 - p_{t+\Delta t}(\boldsymbol{d}_j)] \tag{12}$$

Here, (12) can be thought in terms of a generalized geometric probability distribution, in that the term $p(\boldsymbol{d}_i)\Pi_{j=1}^{i}[1-p(\boldsymbol{d}_j)]$ is interpreted as the probability that the robot travels along $\Delta \boldsymbol{d}$ from its current position $\boldsymbol{p}$ and collides with an obstacle only at $\boldsymbol{d}_i$. It should be noted that (12) is a conservative approximation of the actual expected distance to collision, as the motion of the robot may not be along $\boldsymbol{d}$.

The safety of a waypoint $\boldsymbol{w}$ is determined by comparing the expected distance to collision $|\bar{\boldsymbol{d}}|$ to the stopping distance of the robot. If we assume that the motion of the robot is limited to a maximum velocity $|\dot{\boldsymbol{p}}|_{\max}$ and acceleration $|\ddot{\boldsymbol{p}}|_{\max}$, a maximum stopping distance can be computed as:

$$d_{\text{stop, max}} = \frac{|\dot{\boldsymbol{p}}|_{\max}^2}{2|\ddot{\boldsymbol{p}}|_{\max}} \tag{13}$$

A waypoint $\boldsymbol{w}$ is considered safe when the expected distance to collision is greater than the maximum stopping distance. In this way, the robot has enough room to stop completely before its anticipated collision. To indicate safety we introduce a binary variable $\Theta = 1(0)$ when safe (unsafe):

$$\Theta = \begin{cases} 1, & \text{if } |\bar{\boldsymbol{d}}|/d_{\text{stop, max}} > 1 \\ 0, & \text{else} \end{cases} \tag{14}$$

2) Waypoint Placement: Consider now a predefined desired trajectory $\boldsymbol{\tau}$ used as a means of routing the robot around known static obstacles, such as walls, towards the robot's goal position. Points along $\boldsymbol{\tau}$ that are currently visible to the robot are considered as candidate waypoints, with (14) used to ensure that any particular candidate is safe to move towards.

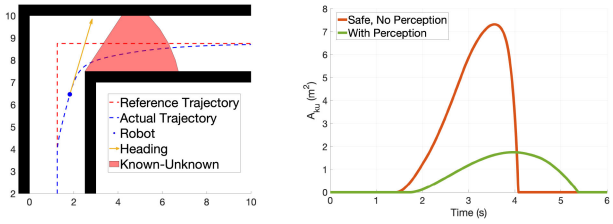
Recall from the problem formulation in Sec III that $A_s(t)$ is defined as the area visible to the robot at time t. Let us define the boundary of this area as $\delta A_s(t) \in \mathbb{R}^2$ as depicted in Fig. 8.

Let $\boldsymbol{\tau}' = \{\boldsymbol{p}_{\tau,0}, \ldots, \boldsymbol{p}_{\tau,i} \ldots, \boldsymbol{p}_{\tau,N}\}$ with $i = \{1, \ldots, N\}$ be an indexed set of N discrete points that make up the visible model trajectory $\boldsymbol{\tau}'$. The waypoint location $\boldsymbol{w}$ is placed at the furthest point along $\boldsymbol{\tau}'$ that is considered safe according to (14). The first point considered is the farthest visible point $\boldsymbol{p}_{\tau,N} = \delta A_s \cap \boldsymbol{\tau}$. If $\boldsymbol{p}_{\tau,N}$ is unsafe, then a closer visible point is considered, $\boldsymbol{p}_{\tau,N-1}$. Fig. 8 shows how this process may be repeated until a safe waypoint $\boldsymbol{w} = \boldsymbol{p}_{\tau,N-k}$ is found. If no point on $\boldsymbol{\tau}$ is considered safe in the upcoming corridor, the waypoint is placed at the entrance of the upcoming corridor $\boldsymbol{p}_{\tau,0}$ and the upcoming corridor is considered unsafe. In situations where $\delta A_s \cap \boldsymbol{\tau} = \emptyset$, the waypoint is chosen as the closest point $\boldsymbol{v}^* \in \delta A_s$ to the desired trajectory $\boldsymbol{\tau}$.

$$\boldsymbol{w} = \begin{cases} \boldsymbol{p}_{\tau,N-k}, \quad k \in [1, N] & \text{if } \delta A_s \cap \boldsymbol{\tau} \neq \emptyset \\ \boldsymbol{v}^* \in \delta A_s \; s.t. \; |\boldsymbol{p} - \boldsymbol{v}^*| \leq |\boldsymbol{p} - \boldsymbol{v}| & \\ \qquad \forall \boldsymbol{v} \in \delta A_s(t), & \text{else} \end{cases} \tag{15}$$

When the waypoint is placed at $\boldsymbol{w} = \boldsymbol{p}_{\tau,0}$, it means that there is some non-negligible probability of expected traffic coming from around the occluding corner. In this situation, it is advantageous for the robot to gain visibility around the occluding corner to increase the certainty that the upcoming hallway is unoccupied, thereby lengthening the expected distance to collision in (14). Thus the perception objective is activated and the waypoint is fixed at $\boldsymbol{p}_{\tau,0}$. By bringing the waypoint closer to the position of the robot, a desired side effect is that the MPC will reduce the speed of the robot to a safe stop at the selected waypoint if needed.

Fig. 9(a) shows the resulting motion through the two-corridor system in the running example using this waypoint generation. Here the perception objective does not activate because the environment is known to be safe. Also shown in Fig. 9(b) is the plot of the known-unknown area used to characterize its visibility (or lack-thereof) around the occluding corner during the operation, in comparison with the case in which the visibility constraint is active (see also Fig. 6).



(a) (b)

Fig. 9. The overall trajectory of the robot in safe situations with no perception objective (a) and in (b) the relative A_{ku} compared to the A_{ku} for the perception case in Fig. 6.

V. Simulations

The first case study investigated in this work is a simple "L"-shaped corridor with one occluding corner. The simulated UGV robot uses a common differential drive motion model [15] and has a maximum velocity of 2 m/s and maximum acceleration of 1 m/s^2. Thus, from (13), $d_{\text{stop, max}} = 2$ m. The FOV of the UGV is limited to a 5 m radius. The optimal control problem was solved using ACADO toolkit [16] and qpOASES [17] as the solver. Fig. 10 shows a series of snapshots of the proposed framework navigating the robot through the occluded intersection. The gray grid cells represent the estimated future probabilities of occupancy. Current occupancy probabilities are determined through a log-odds Bayesian update, and used to

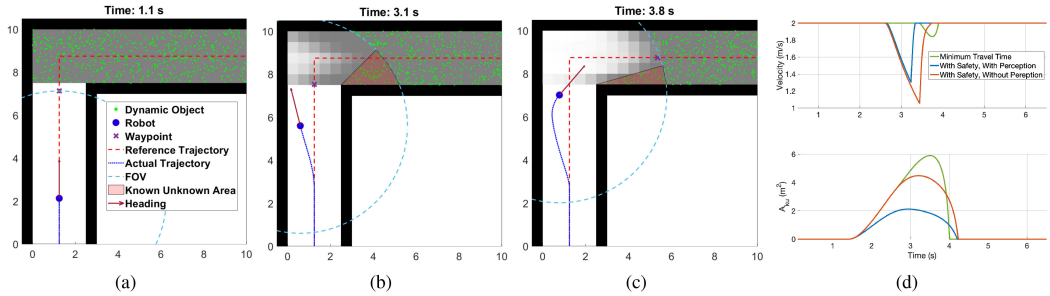


Fig. 10. Snapshots of a simulation case study in which a UGV navigates an occluding corner, considering uncertainties. The gray shaded region on the second corridor indicates the probability that a certain cell is occupied by other actors. (d) shows plots of velocity and A_{ku} for our policy framework with and without perception compared to a minimum travel time implementation.

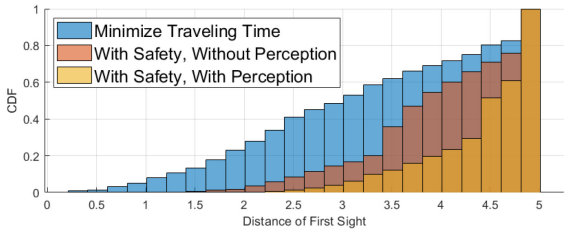


Fig. 11. The cumulative distribution function of the distance that the robot first senses a dynamic object while negotiating the L-shaped corridor.

estimate future occupancy probabilities by convolving with a known motion model.

Also shown in Fig. 10 are simulated dynamic objects, uniformly ranging in speed from 0 to 5 m/s traveling to the left. The convolution model used to temporally propagate the occupancy probabilities correlates to this uniform probability in dynamic obstacle velocity. As the simulation ran, the UGV could "sense" a dynamic object only when it was within the UGV's FOV, recording its distance. This distance serves as a conservative estimate on safety of the proposed framework since it is designed to provide ample reaction time in uncertain situations, and not provide a policy that plans around dynamic obstacles. For these reasons, the dynamic obstacles are removed from simulation when they are first observed.

Fig. 11 shows these results as a cumulative distribution function over the distance that a dynamic obstacle was first sensed. As a point of comparison, also shown are the results of a UGV moving to minimize traveling time (i.e., quickly cutting the corner) as well as a UGV following the safety module without a perception objective to help with visibility. The figure shows how motion that minimizes traveling time has a 23% chance of sensing a dynamic obstacle under $d_{\text{stop, max}} = 2$ m., which is unacceptable in a safety-critical situation. With the full safety and perception framework, there was no situation where the UGV sensed a dynamic obstacle within its maximum stopping distance. Fig. 10(d) plots velocities and known-unknown areas for these simulations for comparison.

It is apparent from Fig. 10(d) that the proposed policy framework performs best at maximizing visibility. What may be surprising is that by considering visibility constraints, the proposed framework *also moves faster around the corner than motion guided only by the safety module*. The UGV is able to do this because having visibility around the occluding corner helps reduce the uncertainty in occupancy, establishing safety more quickly than using the safety module alone.

The second case-study focused on a real-world situation in which an industrial UGV was tasked to navigate a series of hallways inside a warehouse to retrieve an item from a stockroom and take it to a specified location. In this case study, the UGV must reach the stockroom via a main hallway that is often occupied by dynamic obstacles (e.g., people, other robots), and thus has some uncertainty of occupancy. Two scenarios were tested: (1) the main hallway is known to be clear of dynamic obstacles (e.g. it is night-time and no other actors are present in the warehouse) and thus this main hallway is known to be safe a priori, and (2) occupancy in the main hallway is unknown, with a probabilistic motion model that assumes all dynamic obstacles move down the hallway. Fig. 12 shows the setup and results for these scenarios.

As Fig. 12 shows, the main difference of trajectories between the two scenarios is when the UGV enters the main hallway. When there is uncertainty, the UGV moves to gain visibility up the hallway, and when the main hallway is known to be safe, it instead cuts the corner. Fig. 12(e) shows how the known-unknown area is reduced when occupancy is uncertain in the main hallway.

VI. Experiments

Experimental validations were performed with a Clearpath Robotics Jackal UGV inside our lab. As a proof of concept, different occluded geometries were created to showcase how the proposed framework adapted to different scenarios. For each scenario, a Vicon motion capture system was used to measure state-space values of the Jackal, which were fed into the policy framework outlined above. The MPC executed at 10 Hz, producing commanded velocities which were fed to a lower level controller executing at 100 Hz. As the Jackal can follow commanded velocities, it can instantly stop moving at any point and $d_{\text{stop, max}}$ is effectively zero. Because of this feature, only the effect of the perception constraint on motion was explored in experiments.

Fig. 13 shows snapshots for the case study of a UGV approaching the intersection of two hallways where its sensing capabilities are occluded by a sharp corner. Two different objectives were tested: (1) minimum time and (2) perception. Fig. 13 shows snapshots of these two experiments, as well as known-unknown areas recorded by the Jackal. Additionally Figs. 13(a,b) show laser scan data recorded by the onboard lidar. The impact of including perception is highlighted by the additional laser scan points around the occluding corner.

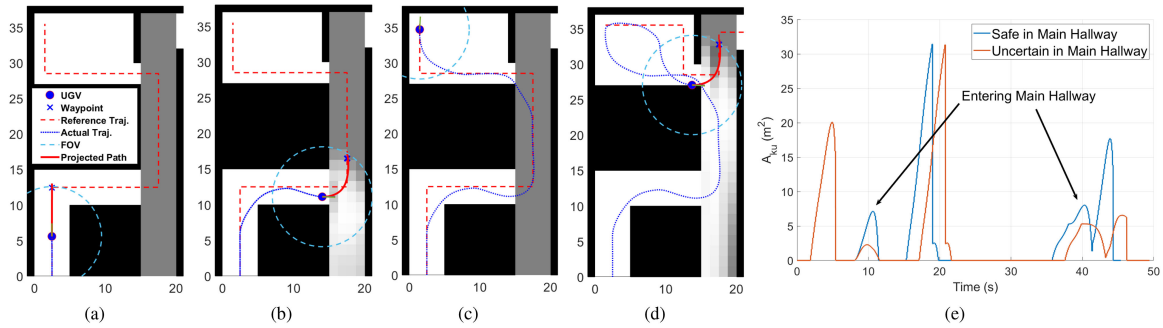
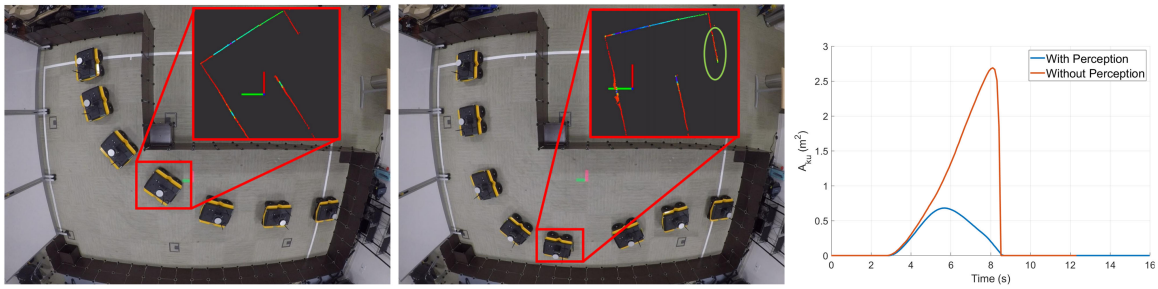


Fig. 12. Sequence of snapshots for a simulation of a robot operating in a occluding environment with variable expected traffic of dynamic objects (a-d). In (e) it is depicted the comparison between the known-unknown areas over time of the case where the long hallway is safe vs unsafe.



(a) Motion with minimum traveling time. (b) Motion with perception. (c) A_{ku} over time

Fig. 13. Snapshots of experiments and data for the two-corridor scenario. Highlighted in (a) and (b) are the lidar point-cloud data before passing the corner showing a decreased known-unknown in (b). In (c) the plots show the difference in A_{ku} between (a) and (b).

Note: More simulations and experiments with different environments and occlusion conditions are available in the following link: https://youtu.be/ErtXafqdzJg.

VII. Conclusions and Future Work

In this work, we have presented a novel MPC-based framework to navigate occluding environments that increases visibility while considering uncertainties. Uncertainties are considered through an occupancy mapping-based approach to autonomously decide if it is safe to move around an occluding corner. If it is not safe, the policy framework chooses a motion that promotes visibility as it approaches the occlusion, thereby reducing uncertainty while navigating around a corner safer and faster.

From here, future theoretical work includes addressing the challenge of tuning MPC parameters for different scenarios and runtime modeling of the probabilistic motion model used to estimate safety. Further experimenting with more complex robots like aerial vehicles is also in our agenda.

References

[1] H. M. Choset *et al.*, *Principles of Robot Motion: Theory, Algorithms and Implementation*. Cambridge, MA, USA: MIT Press, 2005.

[2] M. Elbanhawi and M. Simic, "Sampling-based robot motion planning: A review," *IEEE Access*, vol. 2, pp. 56–77, 2014.

[3] L. Kong, Q. Liu, and C. B. Yu, "Range-limited, distributed algorithms on higher-order voronoi partitions in multi-robot systems," in *Proc. IEEE Int. Conf. Intell. Robots Syst.*, no. 1, 2019, pp. 6541–6546.

[4] N. Bezzo, B. Griffin, P. Cruz, J. Donahue, R. Fierro, and J. Wood, "A cooperative heterogeneous mobile wireless mechatronic system," *IEEE/ASME Trans. on Mechatronics*, vol. 19, no. 1, pp. 20–31, Feb. 2014.

[5] B. Penin, P. R. Giordano, and F. Chaumette, "Vision-based reactive planning for aggressive target tracking while avoiding collisions and occlusions," *IEEE Robot. Automat. Lett.*, vol. 3, no. 4, pp. 3725–3732, Oct. 2018.

[6] D. Falanga, P. Foehn, P. Lu, and D. Scaramuzza, "Pampc: Perception-aware model predictive control for quadrotors," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 1–8.

[7] K. Lee, J. Gibson, and E. A. Theodorou, "Aggressive perception-aware navigation using deep optical flow dynamics and pixelmpc," *IEEE Robot. and Automat. Lett.*, vol. 5, no. 2, pp. 1207–1214, Apr. 2020.

[8] Y. Chang, H. Zhou, X. Wang, L. Shen, and T. Hu, "Cross-drone binocular coordination for ground moving target tracking in occlusion-rich scenarios," *IEEE Robot. Automat. Lett.*, vol. 5, no. 2, pp. 3161–3168, Apr. 2020.

[9] J. Zhang, "Occlusion-aware uav path planning for reconnaissance and surveillance in complex environments," in *Proc. IEEE Int. Conf. Robot. Biomimetics*, 2019, pp. 1435–1440.

[10] D. Fehr, W. J. Beksi, D. Zermas, and N. Papanikolopoulos, "Occlusion alleviation through motion using a mobile robot," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2014, pp. 3179–3184.

[11] L. Janson, T. Hu, and M. Pavone, "Safe motion planning in unknown environments: Optimality benchmarks and tractable policies," in *Proc. Robot.: Sci. Syst.*, Pittsburgh, Pennsylvania, Jun. 2018. *arXiv:1804.05804*.

[12] P. F. Orzechowski, A. Meyer, and M. Lauer, "Tackling occlusions & limited sensor range with set-based safety verification," in *Proc. 21st Int. Conf. Intell. Transp. Syst.*, 2018, pp. 1729–1736.

[13] E. Horváth, C. Hajdu, and P. Kőrös, "Novel pure-pursuit trajectory following approaches and their practical applications," in *Proc. 10th IEEE Int. Conf. Cogn. Infocommunications*, 2019, pp. 000597–000602.

[14] D. J. Kim, S.-H. Lee, and C. C. Chung, "Object vehicle motion prediction based on dynamic occupancy grid map utilizing cascaded support vector machine," in *Proc. 19th Int. Conf. Control, Automat. Syst.*, 2019, pp. 496–500.

[15] S. M. LaValle, *Planning Algorithms*. Cambridge, U.K.: Cambridge University Press, 2006.

[16] B. Houska, H. J. Ferreau, and M. Diehl, "ACADO toolkit - An open-source framework for automatic control and dynamic optimization," *Optimal Control Appl. Methods*, vol. 32, no. 3, pp. 298–312, 2011.

[17] H. J. Ferreau, C. Kirches, A. Potschka, H. G. Bock, and M. Diehl, "qpOASES: A parametric active-set algorithm for quadratic programming," *Math. Program. Comput.*, vol. 6, no. 4, pp. 327–363, 2014.