IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 8 August 2022, accepted 31 August 2022, date of publication 12 September 2022, date of current version 27 September 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3205611

RESEARCH ARTICLE

Ensuring Robot-Human Safety for the BD Spot Using Active Visual Tracking and NMPC With Velocity Obstacles

SAMUEL KARLSSON, BJÖRN LINDQVIST, AND GEORGE NIKOLAKOPULOS
Department of Computer Science, Electrical and Space Engineering, Luleå University of Technology, 971 87 Luleå, Sweden

Corresponding author: Samuel Karlsson (samkar@ltu.se)

This work was supported by the European Unions Horizon 2020 Research and Innovation Program through illuMINEation and NEXGEN-SIMS under Grant 869379 and Grant 101003591.

ABSTRACT When humans and robots operate in and occupy the same local space, proximity detection and proactive collision avoidance is of high importance. As legged robots, such as the Boston Dynamics (BD) Spot, start to appear in real-world application environments, ensuring safe robot-human interactions while operating in full autonomy mode becomes a critical gate-keeping technology for trust in robotic workers. Towards that problem, this article proposes a track-and-avoid architecture for legged robots that combines a visual object detection and estimation pipeline with a Nonlinear Model Predictive Controller (NMPC) based on the Optimization Engine, capable of generating trajectories that satisfy the avoidance and tracking problems in real-time operations where the computation time never exceeded 40 ms. The system is experimentally evaluated using the BD Spot, in a custom sensor and computational suite, and in fully autonomous operational conditions, for the robot-human safety scenario of quickly moving noncooperative obstacles. The results demonstrate the efficacy of the scheme in multiple scenarios where the maximum safety distance violation was only 9 cm for an obstacle moving at 2.5 m/s while affected by both state estimation and object detection uncertainty and noise.

INDEX TERMS Human robot interaction, NMPC, object detection, object tracking, spot, velocity obstacle.

I. INTRODUCTION

Quadruped robots such as the Boston Dynamics Spot depicted in Figure 1 are, due to now being available on the public market and due to their advantageous capabilities in carrying large and flexible payloads over uneven terrain, appearing in more and more industry applications. As their use go from remote operations into fully autonomous mission execution, the need for onboard intelligence paired with perception systems, is increasing as well. And as humans and autonomous robotic workers start occupying the same areas, special attention has to be provided for the safe robot-human interaction problem in a general collision avoidance context.

The ability for a robot to navigate a constrained environment while avoiding collisions with any obstacles encountered along the way, is one of the key ingredients to allow for any fully autonomous mission scenario, and as such, robot path planning is an extensively studied subject [1], [2]. To allow complex mission execution in a dynamic

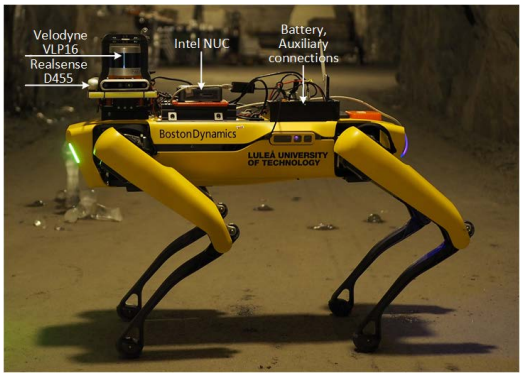

FIGURE 1. Boston dynamics spot with custom added sensor and computation suit, in an subterranean operating environment.

The associate editor coordinating the review of this manuscript and approving it for publication was P. Venkata Krishna.

or unknown environments map-based path planners, such as heuristic grid-search algorithms [3] or sampling based methods [4], provide the high-level paths that should be followed to reach the desired goal (area for inspection, next frontier to visit in the exploration context [5], etc.).

While map-based path planning can provide low-risk paths, and can incorporate notions of risk-aware behavior [6], it is common to combine them with a layer of reactive collision avoidance schemes, operating at the local level and at a high frequency, and preferably linked more directly to the onboard perception and control system for efficient maneuvering. In the context of a dynamic environment, such as a local work space in the construction or mining industry, where other vehicles, human workers, or possibly other robots are constantly moving around, such reactive local navigation schemes are of great importance to maintain a safe environment.

Lately, the idea of incorporating moving obstacles explicitly in the local navigation problem has gained significant attention using a variety of methods [7], [8], [9], [10], [11], [12]. This allows not only for a reactive but a proactive response to a future collision with the moving obstacle. Model Predictive Control is, due to its already predictive nature and direct link to the dynamics of the robot, the perfect fit for the explicit consideration of both cooperative [13] and non-cooperative [14] moving obstacles.

It should be noted: the significance of all of these works, and the proposed method in this article, as opposed to legacy approaches to obstacle avoidance is the ability to react to an incoming obstacle before it has entered the radius of influence, or desired safety distance, of the robot. A proper consideration of dynamic obstacles by also including predicted future obstacle states lessens the underlying assumption that the robot will have time to move out of the way of an incoming obstacle as it enters the desired safety distance by acting *proactively*, while optimally also considering the dynamics of the robot.

While there exist a wide range of modern methods for constructing and solving MPC problems that take model and obstacle uncertainty into account; Chance Constraints [15], Robust Scenario MPC [16], the fundamental link between onboard perception systems and MPC for velocity obstacles, and consequent experimental verification, is relatively underrepresented in the literature, while being a necessity for a real-world implementation in an unknown environment. Surprisingly, to the best of the authors capacity, we could cite only one other work with onboard tracking/estimation and experimental results in a real robot-human safety scenario for velocity obstacles [17]. In the robot-human safety context, a visual approach utilizing Convoluted Neural Networks [18] (CNNs) to detect and classify objects of interest (humans, other robots, vehicles) can provide a discrete bounding box measurement of object positions. In this work, we utilize the state-of-the-art CNN detector YOLOv4 [19] and combine it with an object localizer from depth-images. As noise-reduction is critical when we want to use real obstacle measurements for our velocity obstacle predictions, we apply a Kalman filter [20] to properly estimate the obstacle position and velocity states.

The fundamentally similar work on legged robot MPC is [21], while there exist some significant differences. In that work, the MPC is focused on solving for the more complex dynamics of quadruped walking motion, resulting in a lower horizon for real-time operations. Furthermore, the MPC is linear, as opposed to our nonlinear MPC based proposed framework. The method of object detection also differs: [21] uses occupancy mapping and LiDAR detection of nearby obstacles, while we use a visual CNN to specifically focus on objects of interest. This also leads to the track-and-avoid problem in our formulation, as opposed to a pure collision avoidance problem. We also place our evaluation focus on critical moving obstacle scenarios in the robot-human context to demonstrate fast and efficient proactive avoidance behavior.

As such, the fundamental scope of this article is to provide the link between onboard perception, CNN-based object detection, object tracking and filtering, and constraint-based NMPC obstacle avoidance with a focus on the applicability and evaluation of the framework in a fully autonomous and infrastructure-free approach for quickly moving obstacles, that results in a complete and functioning framework for providing safe human-robot interactions of a quadruped robot.

A. CONTRIBUTIONS

Based on the state-of-the-art, this article presents the following contributions towards the track-and-avoid problem, and general quadruped robot autonomous operations in the presence of humans in the immediate surrounding environment of the robot:

- A novel high-level NMPC framework designed specifically for the track-and-avoid problem for quadruped robots and velocity obstacles. This includes the developed robot model, cost function, and practical additions to facilitate the predictive tracking and re-detection of objects of interest that proved significant for improving performance in the real-world experiment scenarios. The proposed framework allows for a large prediction horizon, thus implying an earlier response to velocity obstacles which is critical for any scenario of an encounter with a quickly moving obstacle as the response time and maximum maneuvering speed of the quadruped robot is limited.
- A full pipeline for obstacle tracking that utilizes visual CNN-based object detection, velocity estimation and filtering by Kalman Filters, and velocity-obstacle predictions that consistently and rapidly feeds the relevant predicted obstacle trajectory information to the NMPC module using only onboard sensors. The framework rapidly and quickly detects and tracks human obstacles in the environment.
- Experimental evaluations that offer a significant contribution to the area of velocity-obstacle avoidance and

safe robot-human interactions. The evaluations are performed in a full autonomy mode (and as such we also present the utilized full autonomy stack) that does not rely on any infrastructure or assistance from outside sources of state estimation or computing. We present, and place focus on, realistic scenarios of critical robot-human safety scenarios in multiple experiments of quickly moving non-cooperative human obstacles that properly demonstrate the track-and-avoid architecture in the presence of real-world real-time challenges.

II. TRACKING & AVOIDANCE ARCHITECTURE

The avoidance architecture, depicted in figure 5, can be divided into three steps: 1) CNN-based object detection using YOLO, 2) the object localization and object state estimation pipeline, and 3) the proposed NMPC to perform the active tracking and avoidance maneuvers. To supplement full autonomy alongside the proposed framework we use the Lidar-Inertial Odometry LIO-SAM [22] used to estimate the robot state $\hat{x}$. The outputs of this architecture are the velocity and heading rate commands that satisfy the avoidance and tracking problems, sent to the built in low level motion controller on Spot.

A. HUMAN DETECTION

Detection of humans is a task that can be solved by a convolution neural network (CNN), and in our case we are utilizing the You Only Look Once version 4 (YOLO) [19]. YOLO takes an RGB image I as its input, and based on a selected training dataset is capable of on that I to detect multiple objects of different kinds. In this work we used pre-trained weights for YOLO, specifically ones developed for the Tiny model. YOLOv4 is a further development of its predecessor YOLOv3, that now is constructed with a CSPDarknet53 backbone, SPP additional module, and PANet path-aggregation neck. A set of weights provided by the authors of [19] were trained with the Microsoft Coco data set [23] to facilitate the detection of humans. The weights were trained using decay learning rate scheduling strategy, with a learning rate of 0.01 which is multiplied by 0.1 after $400,000$ steps and $450,000$ steps. Monument and weight decay were set at 0.9 and 0.0005 respectively. The detection was limited to one human, by selection the human-detection with the highest probability, for tracking and avoidance. When YOLO detects an object it generates a bounding box as $BB = [x^{\mathcal{I}}, y^{\mathcal{I}}, H, W]$ (position x, y, height and width) in pixels as in Figure 2.

B. TRACKER

To transform the BB from image frame $\mathcal{I}$ to the global frame of reference $\mathcal{G}$ the distance to the object is needed, and that is provided by a depth image D that is frame aligned with I. The distance z to the obstacle is retrieved by the depth values from the pixels in D that correspond to the BB's x, y positions. From a square of 3×3 pixels in D, where x, y is the center pixel, the average value is used as the discrete depth to the

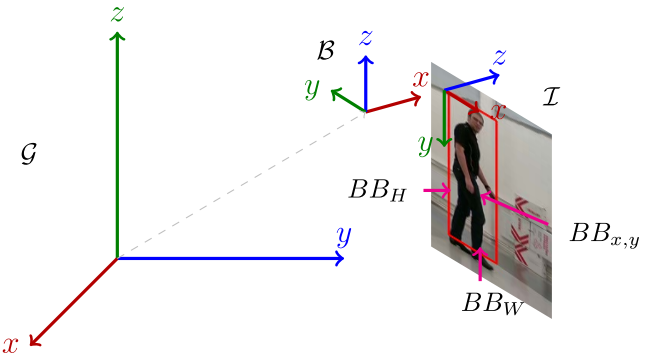

FIGURE 2. The coordinate frames used. $\mathcal{G}$ is the global frame, $\mathcal{B}$ is the body fixed frame for Spot, $\mathcal{I}$ is the camera image frame. $\mathcal{I}$ has a static relationship to $\mathcal{B}$. Whiles $\mathcal{B}$ moves relative to $\mathcal{G}$. The images *I* and *D* are captured in $\mathcal{I}$, where YOLO detects objects and produces *BB*'s.

detected human. This is an assumption that the centroid of BB, where x, y is taken from, is within the actual human that is being detected. Transforming the obstacles position $p_{obs} = [x, y, z]$ from pixels to meters is done with the cameras intrinsic matrix $\mathbf{K}$

$$\mathbf{K}^{-1} p_p = p_m \tag{1}$$

The obstacles width is the distance between p_{min} and p_{max} in meters, where $p_{min} = [x - W/2, y]$ and $p_{max} = [x + W/2, y]$, calculated as in equation 1, resulting in an expected obstacle width of $W_{\text{obs}} = (p_{max,x} - p_{min,x})/2$. To transform p from $\mathcal{I}$ to $\mathcal{G}$, the translation matrix ${}^{\mathcal{G}}\mathbf{T}_{\mathcal{I}}$ is utilized.

$$p^{\mathcal{I}\mathcal{G}}\mathbf{T}_{\mathcal{I}} = p^{\mathcal{G}} \tag{2}$$

${}^{\mathcal{G}}\mathbf{T}_{\mathcal{I}}$ is retrieved from the robot state estimation $\hat{x}$.

1) STATE ESTIMATION

We are interested in estimating the obstacle state from a sequence of detections of the same object or $x^{\mathcal{G}}_{\text{obs}} = [p^{\mathcal{G}}_{\text{obs}}, v^{\mathcal{G}}_{\text{obs}}]$. By tracking the position change over time, the object velocity v_{obs} can be calculated, which is clearly needed for the consideration of a velocity obstacle avoidance scenario. In this case, the problem is that p_{obs} is noisy and is generated with a high frequency (16Hz) that causes the calculated v_{obs} to be unreliable. A Median filter can mitigate this issue, but it also introduces some inertia to the system, where a speed change will take several measurements to show in v_{obs}. Instead a Kalman Filter is implemented to estimate the full obstacle state $\hat{x}_{\text{obs}}$, based on measurements of $p^{\mathcal{G}}_{\text{obs}}$ that are provided by the localiser, with the added benefit of filtering our measurements of $p^{\mathcal{G}}_{\text{obs}}$. A Kalman Filter is a statistical optimal state estimator when measurements have a Gaussian noise, that make them ideal for filtering noisy measurements and estimating non measurable states. It is assumed that detected humans state $\hat{x}^{\mathcal{G}}_{\text{obs}} = [p^{\mathcal{G}}_{\text{obs}}, v^{\mathcal{G}}_{\text{obs}}]$ evolves according to a constant acceleration model

$$\hat{x}^{\mathcal{G}}_{\text{obs},k+1} = \mathbf{F}\hat{x}^{\mathcal{G}}_{obs,k} + \mathbf{G}a \tag{3}$$

where

$$\mathbf{F} = \begin{bmatrix} 1 & \delta t \\ 0 & 1 \end{bmatrix} \tag{4}$$

$$\mathbf{G} = \begin{bmatrix} \frac{\delta t^2}{2} \\ \delta t \end{bmatrix} \tag{5}$$

and a is the acceleration. The state $\hat{x}^{\mathcal{G}}_{\text{obs}}$ is divided per axis for the tracking problem, so in practice there are three identical Kalman filter running in parallel.

C. NMPC

1) TASK DEFINITION & CONTROL OBJECTIVE

In this article the objective is to design an NMPC framework that can be used as a baseline position controller, while utilizing the already existing walking motion of the Boston Dynamics Spot, commanded via velocity and heading rate inputs. One can see this as a higher-level controller built in a cascaded fashion on top of the out-of-the-box controller running on Spot. Our proposed controller is designed such that if an object of interest (human) comes into view the controller should both avoid and track the obstacle while keeping it in camera field of view, and if not, it should act as a normal position controller that can be given waypoints in a global frame from any path planning or exploration module added on top of it. Since the NMPC by definition is already considering future predicted states of the system, one can easily incorporate the addition of a moving obstacle, or velocity obstacle. But, importantly, we want to use the predictive nature of the NMPC to also track heading angles that look towards the obstacle at future predicted time instants. Thus, the control objectives are threefold: 1) reach the desired reference state, 2) avoid detected velocity obstacles, 3) predictively track detected human obstacles.

2) MODEL AND COST FUNCTION

As the NMPC uses the built-in Spot walking controller, we can ignore the complex low-level dynamics of autonomous quadruped walking movements. Instead, we define a high level model with states $x = [p^{\mathcal{G}}_x, p^{\mathcal{G}}_y, v^{\mathcal{B}}_x, v^{\mathcal{B}}_y, \psi]^T$ and with control inputs $u = [u^{\mathcal{B}}_{vx}, u^{\mathcal{B}}_{vy}, u_\omega]^T$. Here, the denoted frame $\mathcal{G}$ is in a global frame of reference, and robot body frame velocities in $\mathcal{B}$ simply represent velocities with a rotation about the z-axis with the heading angle ψ. The model for the legged robot then becomes:

$$\begin{aligned}
\dot{p}^{\mathcal{G}}_x(t) &= \cos\psi(t) v^{\mathcal{B}}_x(t) - \sin\psi(t) v^{\mathcal{B}}_y(t) \\
\dot{p}^{\mathcal{G}}_y(t) &= \sin\psi(t) v^{\mathcal{B}}_x(t) + \cos\psi(t) v^{\mathcal{B}}_y(t) \\
\dot{v}^{\mathcal{B}}_x(t) &= {}^1/_{\tau_{vx}}(\kappa_{vx}(u^{\mathcal{B}}_{vx}(t) - v^{\mathcal{B}}_x(t)) \\
\dot{v}^{\mathcal{B}}_y(t) &= {}^1/_{\tau_{vy}}(\kappa_{vy}(u^{\mathcal{B}}_{vy}(t) - v^{\mathcal{B}}_y(t)) \\
\dot{\psi}(t) &= \kappa_\omega u_\omega(t)
\end{aligned} \tag{6}$$

where the evolution of velocity states is modelled as a first-order system with gains $\kappa_{vx}, \kappa_{vy} \in \mathbb{R}$ and time constants $\tau_{vx}, \tau_{vy} \in \mathbb{R}$, representing the response of the low-level Spot controller when acting on control input u, while $\kappa_\omega \in \mathbb{R}$ is the gain related to the rotational movements. The system dynamics are discretized with the forward Euler using sampling time T_s to obtain the predictive form $x_{k+1} = \zeta(x_k, u_k)$ used as the discrete prediction model for the NMPC.

The cost function defines the control objective of the NMPC. In this case we would like to reach the desired state reference $x_{ref} = [p^{\mathcal{G}}_{ref}, v^{\mathcal{B}}_{ref}, \psi_{ref}]^T$, while minimizing actuation and delivering smooth control signals. Commonly, a quadratic cost can be imposed on states and inputs, and we will do the same for the position states and velocity states/inputs. A problem with quadratic costs is that for the heading state we do not capture properly the 2π-modularity of such an angle/rotation state, for example, if we assume that the state estimation provides $\psi \in [-\pi, \pi]$, there is a clear discontinuity at $\psi \sim \pm\pi$ and an incorrect cost associated with $\psi_{ref} - \psi$ for a subset of initial conditions and references. Now, most works on NMPC for mobile robots either use a model that doesn't consider the evolution of the heading state [24] or ensures that a situation with sub-optimal control behavior, e.g. taking the long way around due to not considering the modularity, doesn't arise by continuous coordinate frame shifts.

We would like to be able to predict how the robot should track future heading references for a large horizon and a moving obstacle, a concept shown in Figure 3 displaying the desired behavior of ψ tracking future obstacle positions, and this requires us to properly penalize the heading state error under the consideration of how such a state actually evolves, as it has to be consistent for the full prediction regardless of the relative predicted motion of robot and obstacle. There is a wide range of trigonometric functions that result in the desired costs, but we will penalize the heading state as $(-\cos(\psi_{ref} - \psi) + 1)Q_\psi$, e.g. it is at its minimum for $(\psi_{ref} - \psi) \mod 2\pi = 0$, continuous, differentiable, and positive real valued.

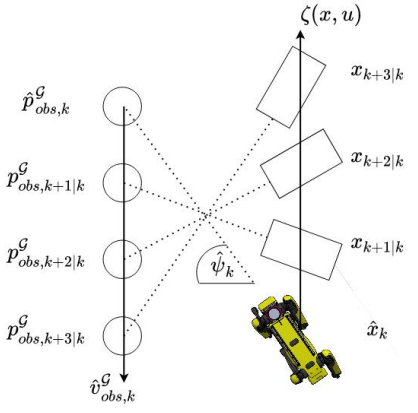

FIGURE 3. Tracking of future obstacle positions $\hat{p}^{\mathcal{G}}_{obs}$ by aligning the predicted heading state ψ towards the predicted obstacle trajectory at each time step $k = 0 \ldots N$.

Let $\boldsymbol{u}_k$ denote the full sequence of control inputs along the horizon and $\boldsymbol{x}_k$ the predicted states. Also let $k + j|k$ denote

a prediction j steps into the future, performed at time step k. The total cost function becomes:

$$J(\boldsymbol{x}_k, \boldsymbol{u}_k; u_{k-1|k}) = \sum_{j=0}^{N} \|p^{\mathcal{G}}_{\text{ref,j}} - p^{\mathcal{G}}_{k+j|k}\|^2_{Q_p} + \|v^{\mathcal{B}}_{\text{ref,j}} - v^{\mathcal{B}}_{k+j|k}\|^2_{Q_v} + (-\cos(\psi_{ref} - \psi) + 1)Q_\psi + \|u_{\text{ref,j}} - u_{k+j|k}\|^2_{Q_u} + \|u_{k+j|k} - u_{k+j-1|k}\|^2_{Q_{\Delta u}}, \quad (7)$$

where $Q_p, Q_v \in \mathbb{R}^{2\times2}$, $Q_\psi \in \mathbb{R}$, and $Q_u, Q_{\delta u} \in \mathbb{R}^{3\times3}$ are diagonal weight matrices related to position, velocity, and heading states, as well as input and input rates respectively. $J(\boldsymbol{x}_k, \boldsymbol{u}_k; u_{k-1|k})$ denote the total cost function, and N is the prediction horizon of the receding horizon NMPC problem. Additionally, we pose hard bounds on control inputs to match the predicted behavior to how the real Spot robot can move as:

$$u_{\min} \leq u_{k+j|k} \leq u_{\max}. \quad (8)$$

3) OBSTACLE AVOIDANCE

Obstacle avoidance of moving or dynamic obstacles using NMPC has been considered before for velocity obstacles [8], [14], [25]. We follow the approach in [26] and impose a set-exclusion constraint [27] on the robot position states representing a circular area with center $p^{\mathcal{G}}_{obs}$ and radius r_{obs}. We can write this as an equality constraint using the $max(a, 0) = [a]_+$ operator, such that the constraint is violated (non-zero) if the obstacle and the robot are closer by distance than r_{obs}:

$$C_{\text{circle}}(p^{\mathcal{G}}, p^{\mathcal{G}}_{obs}, r_{obs}) = [r^2_{\text{obs}} - (p^{\mathcal{G}}_x - p^{\mathcal{G}}_{\text{obs,x}})^2 - (p^{\mathcal{G}}_y - p^{\mathcal{G}}_{\text{obs,y}})^2]_+ = 0. \quad (9)$$

We model spot as two circles, one over the shoulder and one over the rear, as in the concept shown in Figure 4. As the source of state estimation is the shoulder-area, the center of that circle is already defined by $p^{\mathcal{G}}$. The rear circle is defined in the global frame as: $p^{\mathcal{G}}_{\text{rear}} = [p^{\mathcal{G}}_x - l\cos\psi, p^{\mathcal{G}}_y - l\sin\psi]$, with l denoting the distance from shoulders to rear on the Spot robot. We define the exact same constraint (9) for $p^{\mathcal{G}}_{\text{rear}}$, and denote it $C_{\text{circle},2}$. For the consideration of a moving obstacle,

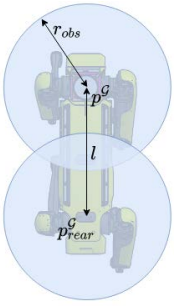

FIGURE 4. Spot modelled as two circles with radius r_{obs}, one over the shoulder aligned with the LiDAR frame and one over the rear with a static body-frame offset. The circles represent the position space where if the obstacle enters this space, $C_{\text{circle}}(p^{\mathcal{G}}, p^{\mathcal{G}}_{obs}, r_{obs}) \geq 0$.

we use a constant velocity model as $\dot{p}^{\mathcal{G}}_{\text{obs}} = v^{\mathcal{G}}_{\text{obs}}$ to predict future obstacle positions from the estimated $\hat{x}_{\text{obs}}$ to generate the obstacle position-trajectory denoted as $\boldsymbol{p}^{\mathcal{G}}_{\text{obs}} = (p^{\mathcal{G}}_{\text{obs,j}})_j$ along the prediction horizon. The obstacle radius is calculated as:

$$r_{obs} = r_s + W_{\text{obs}}/2 + c(\sigma_p + \sigma_v) \quad (10)$$

with r_s denoting a minimum safety radius, W_{obs} the width of the detected obstacle bounding box translated to meters, and σ_p, σ_v are the Euclidean norms of covariances of the tracking Kalman filter along each axis for the position and velocity states respectively. The idea is to increase the obstacle radius if the measurement (and thus also prediction) uncertainty of obstacle states are high. A similar idea was used in [14] using robot state uncertainties, while in our case the $\hat{x}_{\text{obs}}$ covariances are directly linked to how uncertain the predicted obstacle positions are.

4) PREDICTIVE TRACKING

As keeping the object of interest in the camera field of view is of great importance we are interested in tracking it over the full predicted horizon in the NMPC problem. Based on (7) object tracking heading references are generated as:

$$\psi_{\text{ref}} = \arctan(p^{\mathcal{G}}_{obs,y} - p^{\mathcal{G}}_y, p^{\mathcal{G}}_{obs,x} - p^{\mathcal{G}}_x). \quad (11)$$

We can also evaluate this expression based on future predicted robot states and predicted obstacle states via the discrete prediction model $x_{k+1} = \zeta(x_k, u_k)$. Here there are two approaches: either internally evaluating (11), as part of the NMPC problem, or using the previous solution $\boldsymbol{u}_{k-1}$, under the assumption that NMPC trajectories will not differ significantly from one time step to the next (a concept successfully used in [13] for real-time dynamic obstacle prioritization) during fast run-time operations. Thus, we can generate $\psi_{\text{ref,j}}, j = 0 \ldots N$ along the horizon from predicted obstacle positions $\boldsymbol{p}^{\mathcal{G}}_{\text{obs}}$, estimated position $\hat{p}^{\mathcal{G}}$, and previous solution $\boldsymbol{u}_{k-1}$. We select the second approach based on initial testing, and as we are already using $\boldsymbol{u}_{k-1}$, as the initial guess for the optimizer to promote a consistent generated trajectory for the avoidance maneuver.

We also impose the tracking problem as a constraint. Based on the onboard camera, the NMPC should command the heading towards the object of interest to be at least within the field of view, if perfect tracking is impossible. This ensures that the robot does not loose track of the obstacle during an aggressive avoidance maneuver or for a fast-moving obstacle, and that we do not have to pose an extremely large Q_ψ to be able to track a fast-moving obstacle. Using the same expression as for the heading costs/penalties, but allowing a specified range of $\psi_{\text{ref}} - \psi$ we get:

$$C_{\text{track}}(\psi, \psi_{\text{ref}}, \beta) = [-\cos(\psi_{ref} - \psi) + 1 - \beta]_+ = 0 \quad (12)$$

e.g. allowing a heading error, representing a value smaller than half the camera field of view as to have a margin for not loosing track of the object. The actual value of β is just

the evaluation of $-\cos(\Delta\psi)+1$ where $\Delta\psi$ is the desired maximum offset, for example for the 30° offset used in Section III, $\beta \sim 0.13$. Our final consideration for actively tracking the object is to use the already predictive nature of the scheme, and impose that if we have a loss of tracking frames for some moments (for example if the object is momentarily too close), then instead of ending the tracking process of the object, we update the obstacle states as: $\hat{x}_{\text{obs,k}} = [p_{\text{obs,k|k-1}}, v_{\text{obs,k-1}}]^T$ and proceed as normal for the obstacle predictions, avoidance, and tracking e.g. guessing that the object will keep moving along its predicted trajectory with the last measured velocity value. This process continues for $\sim$ 2s or until the object is re-detected. The problem is formulated such that both C_{track} and Q_ψ can easily be turned off (set to zero) if there is no obstacle detection (nor a desired heading for normal reference tracking).

5) OPTIMIZATION

This work focuses on the implementation and performance aspects of the NMPC and not on the optimization itself, as it follows established methods, while the reader should investigate the cited works in this section for more information. The resulting NMPC problem from the cost function in (7), obstacle avoidance constraints (9), tracking constraints (12) and hard input bounds (8) can be written as:

$$
\begin{aligned}
&\underset{\boldsymbol{u}_k,\boldsymbol{x}_k}{\text{Minimize}}\ J(\boldsymbol{x}_k,\boldsymbol{u}_k,u_{k-1|k})\\
&\text{s. t.:}\quad x_{k+j+1|k} = \zeta(x_{k+j|k},u_{k+j|k}),\\
&\qquad j=0,\dots,N,\\
&\qquad u_{\min} \le u_{k+j|k} \le u_{\max},\ j=0,\dots,N,\\
&\qquad C_{\text{sphere}}(p^{\mathcal{G}}_{\text{k+j|k}}, p^{\mathcal{G}}_{obs,j}, r_{obs}) = 0,\\
&\qquad j=0,\dots,N,\\
&\qquad C_{\text{sphere,2}}(p^{\mathcal{G}}_{\text{rear,k+j|k}}, p^{\mathcal{G}}_{obs,j}, r_{obs}) = 0,\\
&\qquad j=0,\dots,N,\\
&\qquad C_{\text{track}}(\psi_{\text{k+j|k}}, \psi_{\text{ref,j}}, \beta) = 0,\\
&\qquad j=0,\dots,N,\\
&\qquad x_{k|k} = \hat{x}_k.
\end{aligned}
\tag{13}
$$

This NMPC problem fits into the framework of the Optimization Engine [28], or OpEn, where the PANOC [29] (Proximal Averaged Newton-type method for Optimal Control) algorithm is combined with a Penalty Method [27] to solve for trajectories that satisfy the obstacle and tracking constraints. The solving of the NMPC problem follows previous works [8], [24], [26], [28].

III. RESULTS

A. EXPERIMENT SET-UP

To facilitate a realistic robot-human avoidance we require that the proposed schemes operates in a fully autonomous mode, without external infrastructure or assistance. The full autonomy hardware-software architecture used for experiments can be found in Figure 5 with the corresponding robot in Figure 1, where we use the state-of-the-art tightly coupled LiDAR-Intertial Odometry LIO-SAM [22] for robot state-estimation. The robot is equipped with a RealSense D455 RGB-D camera providing RGB frames at 30 FPS with 90° Horizontal FOV and 65° Vertical FOV, as well as depth images at 30 FPS with 86° Horizontal FOV and 57° Vertical FOV, used in the object tracking pipeline. A Velodyne VLP16 PuckLite 3D LiDAR provides 3D pointclouds $\{\boldsymbol{P}\}$ at 10 Hz with 360° Horizontal FOV and 30° Vertical FOV with a range of 100 m, here used only for the Lidar-Inertial odometry together with IMU data from a Vectornav VN-100. Spot carries an additional computational payload consisting of an Intel NUC - NUC10i5FNKPA processor, as well as the Intel Neural Compute Stick 2 (for accelerating YOLO computations).

The experiments are carried out in a laboratory environment at Luleå University of Technology in a larger empty space. We should note that, based on Figure 5, position waypoints $p^{\mathcal{G}}_{\text{ref}}$ can be provided from any kind of higher level mission module, such as path planning, exploration, or inspection-trajectory modules for any scenario where Spot could be operating nearby human workers. But in this evaluation they will simply be provided manually by an operator. The NMPC and obstacle prediction sampling time is set to $T_s = 0.1$ s, and as such the main control loop also runs at 10 Hz, while the NMPC prediction horizon is set to 40 implying a prediction of 4 s. The modelling constants are obtained by a step-response experiment, the result being gains $\kappa_{vx}, \kappa_{vy}, \kappa_\omega = 1$ (as could be assumed), while time constants $\tau_{vx}, \tau_{vy} = 0.4$.

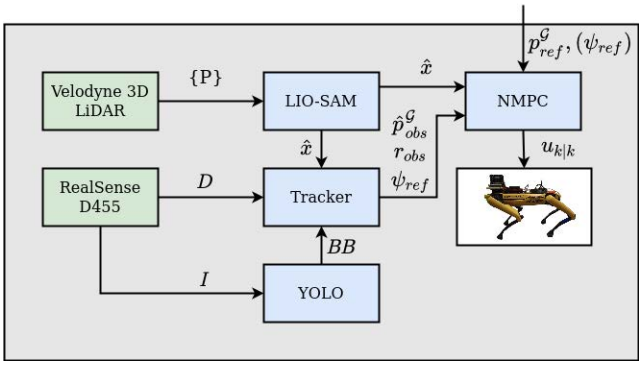

FIGURE 5. The proposed full autonomy hardware-software architecture used for experiments. From RGB image *I* and depth image *D* together with estimated robot state $\hat{x}$ the tracker provides to the NMPC module the predicted obstacle trajectory $\hat{p}^{\mathcal{G}}_{\text{obs}}$ and radius r_{obs} as well as predicted heading references ψ_{ref} looking towards the obstacle.

B. EXPERIMENTAL EVALUATIONS

The proposed complete kit is evaluated in a series of obstacle avoidance and tracking scenarios consisting of: 1) evaluating the detection pipeline and predictive tracking of a quickly moving object, 2) collision avoidance and tracking of a static object, 3) position hold where a moving object enters the robots space, 4) a meeting scenario

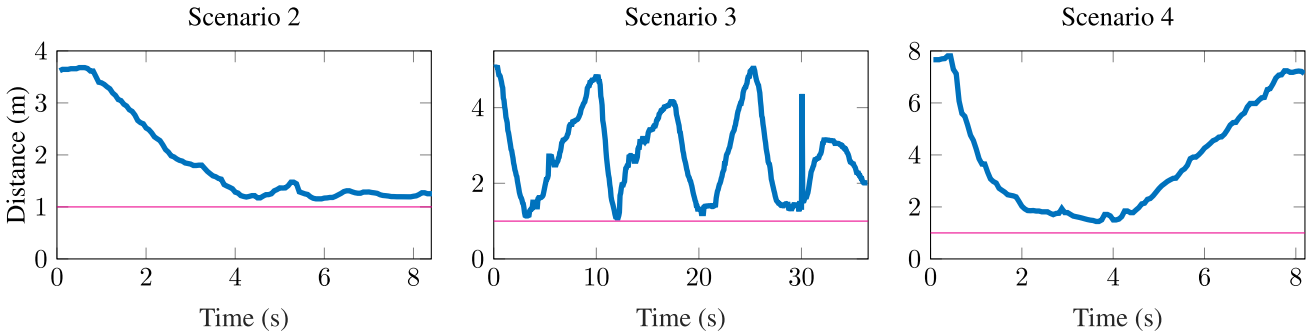

FIGURE 6. The distance between Spot and the human in the different scenarios measured in the depth image D, compared to the desired minimum safety-distance of 1 m. A median filter has been applied to the depth measurements to eliminate outliers and noise, which the Kalman Filter usually does for the tracking of the estimated $\hat{x}_{obs}$. The smallest distance between spot and the human that were 0.9m and occurred in scenario 3.

where the robot encounters a moving human while following position waypoints. Autonomous operations with multiple moving objects are hard to visualize in static figures, so we recommend the reader to watch the experiment video found at `https://youtu.be/_NaRNfnSmks` displaying properly the real-time behavior of the scheme in the mentioned full-autonomy scenarios. The proposed system correctly detects, predicts, tracks, and avoids the human obstacle in every aspect and provided efficient avoidance maneuvers while never losing sight of the object. Clearly, there are some instants of suboptimal behavior, for example related to minor jumps in obstacle trajectories in the static scenario (2), but in general the proposed scheme could maintain a clearly safe distance to the object in the challenging and realistic full autonomy scenarios. The measured robot-human safety distance can be found in Figure 6, where the minimum distance was 0.9 m, corresponding to a slight constraint violation below our minimum safety radius of 1 m, and occurred during the fastest moving obstacle interaction in Scenario (3). We should note that these are the raw depth-camera obstacle distances based on the onboard camera (pre-Kalman filter) as no ground truth was available, while we have removed some momentary outlier detection hits and incorrect detection hits (e.g. where the detector would momentarily classify another object as a human) with a median filter to more clearly visualise how safety distances are maintained.

The NMPC computation time from scenario (3) can be found in Figure 9, where we can see that the OpEn module easily solves the NMPC problem within the 0.1 s sampling time, spiking only once to 37 ms. Other scenarios looked very similar with a general solver time of around 5-10 ms during the avoidance maneuver, with the occasional spike (maximally 56 ms). This implies that we can likely run the NMPC with a significantly higher horizon, if that is needed in an application scenario, for even earlier reactions to moving obstacles. In table 1 are the max and average velocities of the tracked human in the different experiments shown (excluding (2) as the object is static). These are the max estimated momentary velocities while the highest average approaching speed in scenario (3) was around 2.5 m/s,

TABLE 1. A table depicting the max and average estimated velocities (in m/s) of the human in the different scenarios.

Scenario	1	3	4
Max v	6.2	4.7	3.0
Average v	1.5	1.4	1.5

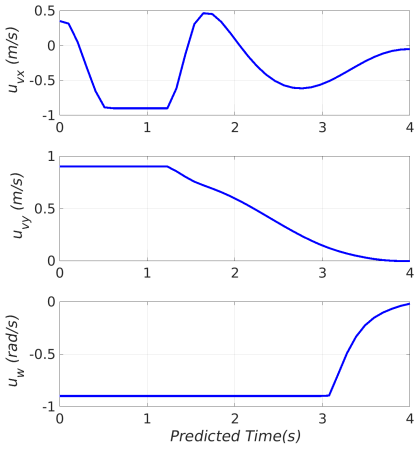

FIGURE 7. Predicted control actions along the horizon as the avoidance maneuver is initiated in scenario (4) where the human and spot are walking towards each other.

which is a major result in the context of velocity obstacle avoidance for a real-world no infrastructure experiment. Spot manages to avoid all obstacles in the demonstrated tests while successfully keeping them in the field-of-view during the avoidance maneuver. An example of a predicted obstacle avoidance maneuver is depicted in Figures 7 and 8. These show data from the specific time instant in scenario (4) (meeting scenario) where Spot first estimated that the moving obstacle is on a collision course, and initiates the proactive collision avoidance maneuver. Figure 7 shows the computed control input trajectory along the horizon at that moment, where we can see that the maneuver is planned such that Spot rotates maximally via u_w as to track the rapidly moving obstacle visually, while it is then forced to move backwards

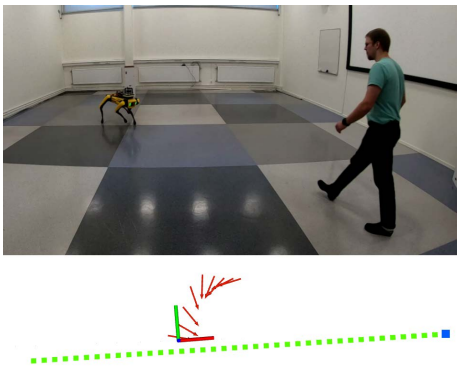

FIGURE 8. A snapshot of predicted robot poses at the instant of a collision-course trajectory. Spot is predicted to move along the red arrows (showing the predicted heading state ψ) where the frame shows the current robot pose and the green dots display the predicted pedestrian approaching from the blue dot (current object measurement).

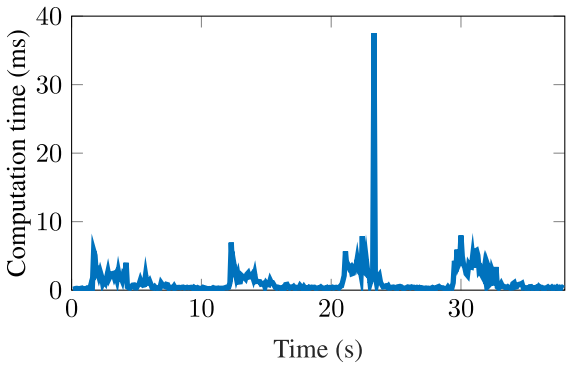

FIGURE 9. Computation time of the NMPC module during scenario (3) where spot is holding position and a human walks towards it. Four avoidance maneuvers are visible as increases in computation time due to Penalty Method iterations being applied. At one instant the computation time spiked to 37ms.

in u_{vx} as well as heavily actuate in u_{vy} to both avoid and look towards the object simultaneously. Figure 8 shows the scenario set-up as avoidance is initiated, and the predicted poses along the prediction horizon of Spot, and the predicted obstacle trajectory. We see that even at the first moment of detecting an obstacle on a collision course, the NMPC module plans a full avoidance maneuver that both tracks and avoids the obstacle within the horizon.

In scenario (2) the human obstacle is standing still and should therefore have a constant position in an ideal case. We centralize measured object position by their median position value and the offset distance calculated from that point as a measure of the consistency of the obstacle tracking pipeline. The offset distance is displayed as a box plot in figure 10, where a total distribution of 0.3 m and a IQR of 0.12 m is shown. The median offset distance is 0.09 m can partly be explained by the size of the object and that the distance to it is measured from different angles thought the experiment, while the position median manage to eliminate this offset.

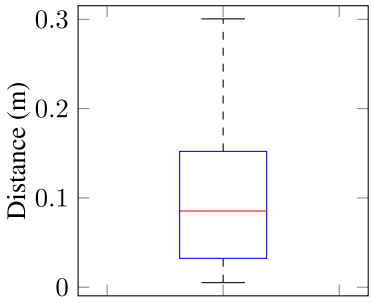

FIGURE 10. Box plot of the humans tracked distance form its median position in scenario (2), with a static obstacle. The distance to the median position visualise the level of noise in detection and tracking. In this case were the median error 0.09 m and LQR 0.12 m.

IV. LIMITATIONS & FUTURE WORK

While we can demonstrate successful results in the specified trials, the general novel proposed framework of ensuring safe robot-human interactions can still be improved with multiple directions of future works that attack the limitations of the framework. First, we have a very large reliance on the simple velocity obstacle prediction model. The avoidance, tracking, and re-detection schemes rely on the idea of constant-velocity movements, which does have surprisingly good performance under the condition of fast feedback corrections, but clearly limits the performance and proactive nature of the scheme if the obstacle is moving with very irregular movements. Interestingly, this is an underlying problem for avoidance of moving non-cooperative obstacles: how do we predict future obstacle states based on a buffer of previous measurements. We should investigate and take inspiration both from the ideas of Chance Constraints [15] as well as ideas of *intentions*, from man-in-the-loop systems [30]. On the detection side; incorporating multiple cameras for larger total field of view, tracking other key objects such as vehicles and other robots, multi-obstacle (crowd) tracking, and keeping a memory of previous obstacle detections is a clear direction for future implementation work. This could pose another interesting optimization problem: how should the robot orient itself to keep the most relevant *prioritized* obstacles in the field of view, while also gaining new information on the movements of as many previously detected objects as possible in a crowd scenario? Another direction for future work is to increase the use of CNNs and deep learning to have a CNN that does more than only detect the obstacle. It could for example also be extended track the objects and to predict the objects future movement based on a buffer of measurements as opposed to our method of Kalman filtering plus assuming a constant-velocity model. Additionally, the evaluations presented in this article are focused on the pure track-and-avoid scenario. Future works should place the robot in the context of performing a relevant mission in close proximity to humans (ex. inspection of an active construction site or underground mine) where the presented local avoidance scheme has to operate in parallel to higher level modules for mission execution. In this context, comparisons with both classical approaches and

other state-of-the-art methods of perception-based obstacle avoidance should also be performed to demonstrate the need for the consideration of dynamic obstacles in maintaining safe robot-human interactions.

V. CONCLUSION

This work has presented a tracking and avoidance framework based on NMPC and a CNN for human detection combined with Kalman filtering. The main focus of the article is providing a full-autonomy framework without any reliance on external sensing, computation or assistance for ensuring safe robot-human interactions, while we pose a series of practical implementations to solve the track-and-avoid problem. We demonstrate the efficacy of the scheme in multiple realistic and challenging laboratory experiments using the quadruped BD Spot robot, for quickly moving obstacles that stress-test both the avoidance and tracking functionalities. In the multiple performed experiments of velocity-obstacle avoidance, the desired safety distance of 1 m was maintained, except for one instance in the scenario with the most rapidly moving obstacle, approaching the robot at 2.5 m/s where we saw a violation of around 0.09 m. As we are operating under real-world conditions with significant measurement/detection and state-estimation noise such small violations are realistic. Additionally, in the performed trials, the computation time of the NMPC module never exceeded 40 % of the 0.1 s sampling time, and on average was well below 10 % of the sampling time during the avoidance maneuvers implying the scheme easily runs online. While the trials show promising results, the development of the proposed track-and-avoid framework has multiple directions for future work.

ACKNOWLEDGMENT

(Samuel Karlsson and Björn Lindqvist contributed equally to this work.)

REFERENCES

[1] S. M. LaValle, *Planning Algorithms*. Cambridge, U.K.: Cambridge Univ. Press, 2006.
[2] K. Cai, C. Wang, J. Cheng, C. W. D. Silva, and M. Q.-H. Meng, "Mobile robot path planning in dynamic environments: A survey," 2020, *arXiv:2006.14195*.
[3] F. Duchoň, A. Babinec, M. Kajan, P. Beňo, M. Florek, T. Fico, and L. Jurišica, "Path planning with modified a star algorithm for a mobile robot," *Proc. Eng.*, vol. 96, pp. 59–69, Jan. 2014.
[4] M. Otte and E. Frazzoli, "RRT^X: Asymptotically optimal single-query sampling-based motion planning with quick replanning," *Int. J. Robot. Res.*, vol. 35, no. 7, pp. 797–822, 2016.
[5] S.-K. Kim, A. Bouman, G. Salhotra, D. D. Fan, K. Otsu, J. Burdick, and A.-A. Agha-Mohammadi, "PLGRIM: Hierarchical value learning for large-scale exploration in unknown environments," in *Proc. Int. Conf. Automated Planning Scheduling*, vol. 31, 2021, pp. 652–662.
[6] S. Karlsson, A. Koval, C. Kanellakis, A.-A. Agha-Mohammadi, and G. Nikolakopoulos, "D^*_+: A generic platform-agnostic and risk-aware path planing framework with an expandable grid," 2021, *arXiv:2112.05563*.
[7] Y. Xinyi, Z. Yichen, L. Liang, and O. Linlin, "Dynamic window with virtual goal (DW-VG): A new reactive obstacle avoidance approach based on motion prediction," *Robotica*, vol. 37, no. 8, pp. 1438–1456, 2019.
[8] B. Lindqvist, S. S. Mansouri, A.-A. Agha-Mohammadi, and G. Nikolakopoulos, "Nonlinear MPC for collision avoidance and control of UAVs with dynamic obstacles," *IEEE Robot. Autom. Lett.*, vol. 5, no. 4, pp. 6001–6008, Oct. 2020.
[9] B. H. Lee, J. D. Jeon, and J. H. Oh, "Velocity obstacle based local collision avoidance for a holonomic elliptic robot," *Auton. Robots*, vol. 41, no. 6, pp. 1347–1363, Aug. 2017.
[10] B. Gopalakrishnan, A. K. Singh, M. Kaushik, K. M. Krishna, and D. Manocha, "PRVO: Probabilistic reciprocal velocity obstacle for multi robot navigation under uncertainty," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 1089–1096.
[11] U. Patel, N. K. S. Kumar, A. J. Sathyamoorthy, and D. Manocha, "DWA-RL: Dynamically feasible deep reinforcement learning policy for robot navigation among mobile obstacles," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May/Jun. 2021, pp. 6057–6063.
[12] M. Mohanan and A. Salgoankar, "A survey of robotic motion planning in dynamic environments," *Robot. Auton. Syst.*, vol. 100, pp. 171–185, Feb. 2018.
[13] B. Lindqvist, P. Sopasakis, and G. Nikolakopoulos, "A scalable distributed collision avoidance scheme for multi-agent UAV systems," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep./Oct. 2021, pp. 9212–9218.
[14] M. Kamel, J. Alonso-Mora, R. Siegwart, and J. Nieto, "Robust collision avoidance for multiple micro aerial vehicles using nonlinear model predictive control," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 236–243.
[15] X. Zhang, J. Ma, Z. Cheng, S. Huang, S. S. Ge, and T. H. Lee, "Trajectory generation by chance-constrained nonlinear MPC with probabilistic prediction," *IEEE Trans. Cybern.*, vol. 51, no. 7, pp. 3616–3629, Jul. 2020.
[16] I. Batkovic, U. Rosolia, M. Zanon, and P. Falcone, "A robust scenario MPC approach for uncertain multi-modal obstacles," *IEEE Control Syst. Lett.*, vol. 5, no. 3, pp. 947–952, Jul. 2021.
[17] J. Lin, H. Zhu, and J. Alonso-Mora, "Robust vision-based obstacle avoidance for micro aerial vehicles in dynamic environments," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May/Aug. 2020, pp. 2682–2688.
[18] Z. Cai, Q. Fan, R. S. Feris, and N. Vasconcelos, "A unified multi-scale deep convolutional neural network for fast object detection," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, Oct. 2016, pp. 354–370.
[19] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," 2020, *arXiv:2004.10934*.
[20] G. Bishop and G. Welch, "An introduction to the Kalman filter," *Proc. SIGGRAPH, Course*, vol. 8, nos. 27599–23175, p. 41, 2001.
[21] M. Gaertner, M. Bjelonic, F. Farshidian, and M. Hutter, "Collision-free MPC for legged robots in static and dynamic scenes," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May/Jun. 2021, pp. 8266–8272.
[22] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, and R. Daniela, "LIO-SAM: Tightly-coupled LiDAR inertial odometry via smoothing and mapping," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Jan. 2021, pp. 5135–5142.
[23] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, Sep. 2014, pp. 740–755.
[24] A. Sathya, P. Sopasakis, R. Van Parys, A. Themelis, G. Pipeleers, and P. Patrinos, "Embedded nonlinear model predictive control for obstacle avoidance using PANOC," in *Proc. Eur. Control Conf. (ECC)*, Jun. 2018, pp. 1523–1528.
[25] S. Subramanian, S. Nazari, M. A. Alvi, and S. Engell, "Robust NMPC schemes for the control of mobile robots in the presence of dynamic obstacles," in *Proc. 23rd Int. Conf. Methods Models Autom. Robot. (MMAR)*, 2018, pp. 768–773.
[26] E. Small, P. Sopasakis, E. Fresk, P. Patrinos, and G. Nikolakopoulos, "Aerial navigation in obstructed environments with embedded nonlinear model predictive control," in *Proc. 18th Eur. Control Conf. (ECC)*, 2019, pp. 3556–3563.
[27] B. Hermans, G. Pipeleers, and P. P. Patrinos, "A penalty method for nonlinear programs with set exclusion constraints," *Automatica*, vol. 127, May 2021, Art. no. 109500.
[28] P. Sopasakis, E. Fresk, and P. Patrinos, "Open: Code generation for embedded nonconvex optimization," Int. Fed. Autom. Control, Laxenburg, Austria, 2020.
[29] L. Stella, A. Themelis, P. Sopasakis, and P. Patrinos, "A simple and efficient algorithm for nonlinear model predictive control," in *Proc. IEEE 56th Annu. Conf. Decis. Control (CDC)*, Dec. 2017, pp. 1939–1944.
[30] Y. Pang, G. Zhang, and H. Xia, "Autonomous driving traffic control based on human-in-the-loop decisions," in *Proc. 33rd Chin. Control Decis. Conf. (CCDC)*, 2021, pp. 1116–1121.

SAMUEL KARLSSON received the master's degree in computer science and engineering with specialization in industrial computer systems from the Luleå University of Technology, Sweden, in 2020, where he is currently pursuing the Ph.D. degree with the Department of Computer Science, Electrical and Space Engineering, Robotics and AI Team, working in the field of areal and ground robotic. His research interests include risk aware path planning for safe navigation of atrial and ground robots.

BJÖRN LINDQVIST received the master's degree in space engineering with a specialization in aerospace engineering from the Luleå University of Technology, Sweden, in 2019, where he is currently pursuing the Ph.D. degree with the Department of Computer Science, Electrical and Space Engineering, Robotics and AI Team, working in the field of aerial robotics. His research interests include collision avoidance and path planning for single and multi-agent unmanned aerial vehicle systems, and field applications of such technologies. He has worked as a part of the JPL-NASA led Team CoSTAR in the DARPA Sub-T Challenge on subterranean UAV exploration applications, specifically in the search-and-rescue context.

GEORGE NIKOLAKOPULOS currently works as a Chair Professor in robotics and artificial intelligence (RAI), while heading the Division of Systems and Interaction, Department of Computer Science, Electrical and Space Engineering, Robotics Team, Luleå University of Technology, Sweden. In the past, he was also affiliated with the NASA Jet Propulsion Laboratory (JPL), Pasadena, CA, USA, for contacting collaborative research on Aerial Planetary Exploration. His team participated in the DARPA Grand challenge on Sub-T exploration with the COSTAR1 team of NASA, where they have won the second stage of the competition, in February 2020. He is also a member of the Board of Director at euRobotics, a member of the Scientific Council of ARTEMIS in the field of robotics and AI, a member of the IFAC TC on Robotics, an Elected Expert for the Permanent Working Group (PWG) of A.SPIRE with a focus on process optimization and ultra carbon coal, and an Elected Member on the Aeneas-XECS in embedded control systems.