Fast Obstacle Avoidance Based on Real-Time Sensing

Lukas Huber, *Member, IEEE*, Jean-Jacques Slotine, and Aude Billard, *Fellow, IEEE*

***Abstract*—Humans excel at navigating and moving through dynamic and complex spaces, such as crowded streets. For robots to do the same, it is crucial that they are endowed with highly reactive obstacle avoidance which is adept at partial and poor sensing. We address the issue of enabling obstacle avoidance based on sparse and asynchronous perception. The proposed control scheme combines a high-level input command provided by either a planner or a human operator with fast reactive obstacle avoidance (FOA). The sampling-based sensor data can be combined with an analytical reconstruction of the obstacles for real-time collision avoidance. Thus, we can ensure that the agent does not become stuck when a feasible path exists between obstacles. Our algorithm was evaluated experimentally on static laser data from cluttered, indoor office environments. Additionally, it was used in shared-control mode in a dynamic and complex outdoor environment in the center of Lausanne. The proposed control scheme successfully avoided collisions in both scenarios. During the experiments, the controller took 1 ms to evaluate over 30000 data points.**

***Index Terms*—Collision avoidance, autonomous agents, dynamical systems, crowd navigation, mobile robots.**

I. Introduction

LOCAL avoidance algorithms must take into account new sensory information in real time and adapt their global paths to ensure safe navigation in various environments. With walking speeds in crowds of around 1.4 m/s and distances between proximal agents as low as 30 cm, robots need a reaction time, i.e., control update rate, well below a quarter of a second to ensure collision-free movement.

One approach is to directly elevate distance sensor data using vector field histograms (VFH), which has been applied on mobile robots to navigate autonomously [1]. VFH has the advantage of taking measurement uncertainties into account. Furthermore, the short evaluation time of VFH makes it applicable to shared control on wheelchairs [2], [3]. However, VFH has the shortcoming of the control velocity being zero in narrow pathways, for example, in front of (narrow) doorways.

Manuscript received 10 May 2022; accepted 14 December 2022. Date of publication 26 December 2022; date of current version 31 January 2023. This letter was recommended for publication by Associate Editor L. L. S. Wong and Editor A. Bera upon evaluation of the reviewers' comments. This work was supported by European Union (EU) through European Research Council (ERC) under Grant SAHR. *(Corresponding author: Lukas Huber.)*

Lukas Huber and Aude Billard are with the LASA Laboratory, Swiss Federal School of Technology in Lausanne - EPFL, 6312 Steinhausen, Switzerland (e-mail: lukas.huber@epfl.ch; aude.billard@epfl.ch).

Jean-Jacques Slotine is with the Nonlinear Systems Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139-4307 USA (e-mail: jjs@mit.edu).

This letter has supplementary downloadable material available at https://doi.org/10.1109/LRA.2022.3232271, provided by the authors.

Digital Object Identifier 10.1109/LRA.2022.3232271

Alternatively, sampling-based methods have been used for fast self-collision avoidance for robots in joint spaces [4]. A collision-free environment is taught using sampled data. As the models require training before execution, the methods cannot adapt to fast-changing environments.

Conversely, the velocity obstacle (VO) approach is designed for dynamic environments. VO takes into account the motion of the obstacles and their potential future positions to calculate collision-free motion [5], [6]. VO has been successfully used in shared control on wheelchairs [7]. However, VO is prone to local minima, which predominantly becomes apparent in static, cluttered environments.

Recently, real-time optimization algorithms have become feasible for implementation onboard robots thanks to the improved computational performance of hardware. This allows shared control of wheelchairs with model predictive control [8] or quadratic programming [9]. However, the proposed algorithms cannot guarantee convergence to a feasible solution within a constrained update rate.

Recent work on control barrier functions (CBFs) can ensure a feasible solution by softening some constraints [10] or using slack variables [11]. Nevertheless, these CBFs require a Lyapunov candidate function to reach the goal position and cannot be applied to dynamic environments.

Conversely, artificial potential fields create a closed-form, repulsive field to avoid collision with obstacles [12]. They can be elevated with navigation functions to ensure convergence to the desired goal around concave obstacles [13]. Recent work has expanded navigation capabilities to include discovering obstacles' positions at runtime, but the approach relies on prior knowledge of obstacles' shapes [14]. However, it is difficult to tune navigation functions for dynamic obstacles. Moreover, navigation functions have not been combined with high-level control inputs, as is the case for shared control.

Similarly, [15] creates a dynamical system-based algorithm inspired by harmonic potential fields to ensure collision-free motion in dynamic environments. This has been extended to ensure convergence around concave obstacles by guiding an initial (linear) system around all obstacles to a global goal [16], [17], [18]. However, such approaches require the analytical description of the obstacles and interpreting sensor data onboard a robot often introduces a significant delay [18]. Hence, while the dynamical system algorithms have theoretically guaranteed impenetrability, but delayed reaction times can lead to collisions in highly dynamic environments.

In this work, we propose a fast obstacle avoidance algorithm (FOA) with the following contributions:

- *Low computational complexity* ensures fast evaluation in cluttered environments, such as crowds (see Section III).

- *Obstacle avoidance using sampled data* makes the method directly applicable to Lidar data (see Section IV).
- *Unifying disparate obstacle descriptions,* such as sample-based sensor data and (delayed) analytic reconstruction of obstacles, is enabled through asynchronous evaluation and dynamic weighting (see Section V).

II. Preliminaries

The state of the robotics system $\boldsymbol{\xi} \in \mathbb{R}^d$ refers to the position of a d-dimensional state throughout this work. A dynamical system governs the evolution of the position

$$\dot{\boldsymbol{\xi}} = \mathbf{f}(t, \boldsymbol{\xi}) \quad \text{with} \quad \mathbf{f}(t, \boldsymbol{\xi}) : (\mathbb{R}_{>0}, \mathbb{R}^d) \to \mathbb{R}^d \tag{1}$$

We further introduce a linear (autonomous) dynamical system which is asymptotically stable at the attractor position $\boldsymbol{\xi}^a$:

$$\mathbf{f}(\boldsymbol{\xi}) = -(\boldsymbol{\xi} - \boldsymbol{\xi}^a) \qquad \text{with} \quad \boldsymbol{\xi}^a \in \mathbb{R}^d \tag{2}$$

A. Obstacles and Gamma Function

As in [16], a continuous distance function $\Gamma(\boldsymbol{\xi}) : \mathbb{R}^d \setminus \mathscr{X}^i \mapsto \mathbb{R}_{\geq 1}$ is defined around each obstacle and divides the space into three regions:

$$\begin{aligned}
&\text{Exterior points:} && \mathscr{X}^e = \{\boldsymbol{\xi} \in \mathbb{R}^d : \Gamma(\boldsymbol{\xi}) > 1\} \\
&\text{Margin exterior points:} && \mathscr{X}^g = \{\boldsymbol{\xi} - (R + D^{\text{gap}})\mathbf{r}(\boldsymbol{\xi}) \in \mathscr{X}^e\} \\
&\text{Boundary points:} && \mathscr{X}^b = \{\boldsymbol{\xi} \in \mathbb{R}^d : \Gamma(\boldsymbol{\xi}) = 1\} \\
&\text{Interior points:} && \mathscr{X}^i = \{\boldsymbol{\xi} \in \mathbb{R}^d \setminus (\mathscr{X}^e \cup \mathscr{X}^b)\}
\end{aligned} \tag{3}$$

where $D^{\text{gap}} \in \mathbb{R}_{>0}$ is the margin distance, and $R \in \mathbb{R}_{>0}$ the robot radius. By construction, $\Gamma(\cdot)$ increases monotonically (C^1-smoothness) with increasing distance from the center.

The obstacle avoidance algorithm [17] is defined for *star-shapes*, i.e., a shape where there exists a reference point $\boldsymbol{\xi}^r \in \mathbb{R}^d$ within the obstacle, such that:

$$\exists \boldsymbol{\xi}^r \in \mathscr{X}^i \quad : \quad \langle \mathbf{r}(\boldsymbol{\xi}),\, \mathbf{n}(\boldsymbol{\xi}) \rangle > 0 \quad \forall \boldsymbol{\xi} \in \mathscr{X}^e \tag{4}$$

where $\mathbf{r}(\boldsymbol{\xi}) = (\boldsymbol{\xi} - \boldsymbol{\xi}^r)/\|\boldsymbol{\xi} - \boldsymbol{\xi}^r\|$ with $\mathbf{r}(\boldsymbol{\xi}) \in \mathbb{R}^d$, $\|\mathbf{r}(\boldsymbol{\xi})\| = 1$ is denoted as the normalized reference direction, and $\mathbf{n}(\boldsymbol{\xi})$ is the normal to the surface. An environment made up of *star-shaped* obstacles is referred to as a *star world*.

B. Obstacle Avoidance Through Modulation

Assume the agents' nominal velocity command as $\mathbf{v} \in \mathbb{R}^d$, which results from a user's input or a high-level planner evaluated at position $\boldsymbol{\xi}$. As in [17], the collision-free velocity around a single obstacle is obtained by applying a dynamic modulation matrix $\mathbf{M} \in \mathbb{R}^{d \times d}$:

$$\dot{\boldsymbol{\xi}} = \mathbf{M}(\boldsymbol{\xi}, \mathbf{v})\mathbf{v}, \quad \mathbf{M}(\boldsymbol{\xi}, \mathbf{v}) = \mathbf{E}(\boldsymbol{\xi})\mathbf{D}(\boldsymbol{\xi})\mathbf{E}(\boldsymbol{\xi})^{-1} \tag{5}$$

The basis matrix is defined as:

$$\mathbf{E}(\boldsymbol{\xi}) = [\mathbf{r}(\boldsymbol{\xi}) \;\; \mathbf{e}_1(\boldsymbol{\xi}) \;\; .. \;\; \mathbf{e}_{d-1}(\boldsymbol{\xi})] \tag{6}$$

where $\mathbf{e}_{(\cdot)}$ form an orthonormal basis of the tangent plane.[1] The diagonal eigenvalue matrix $\mathbf{D}(\boldsymbol{\xi})$ defines the stretching in each direction.

[1]The decomposition matrix of spherical obstacles is orthonormal, i.e., $\mathbf{E}^{-1} = \mathbf{E}^T$.

III. Fast Obstacle Avoidance

When navigating in cluttered environments, such as dense crowds requires the application of modulation for all obstacles at each time step [18] This results in a high computational cost which increases linearly with the number of obstacles. We propose a computationally cheaper algorithm by creating a single *virtual* obstacle which encapsulates all obstacles. The modulation is applied once for all obstacles while ensuring collision avoidance. The virtual obstacle has a single modulation matrix $\mathbf{M}(\boldsymbol{\xi})$ with corresponding decomposition matrix $\mathbf{E}(\boldsymbol{\xi})$ and diagonal stretching matrix $\mathbf{D}(\boldsymbol{\xi})$

A. Single-Modulation Obstacle Avoidance

To evaluate the decomposition matrix $\mathbf{E}(\boldsymbol{\xi})$ as defined in (6), we use the normalized reference direction $\mathbf{r}(\boldsymbol{\xi})/\|\mathbf{r}(\boldsymbol{\xi})\|$, and the tangent directions $\mathbf{e}_i(\boldsymbol{\xi})$ to form the orthonormal basis to $\mathbf{n}(\boldsymbol{\xi})$. Hence, the normal is evaluated as

$$\mathbf{M}(\boldsymbol{\xi}, \mathbf{v}) = \begin{cases} \mathbf{E}(\boldsymbol{\xi})\mathbf{D}(\boldsymbol{\xi}, \mathbf{v})\mathbf{E}(\boldsymbol{\xi})^{-1} & \text{if } \|\mathbf{r}(\boldsymbol{\xi})\| \neq 0 \\ \mathbf{I} & \text{otherwise} \end{cases} \tag{7}$$

Note, that using the identity matrix at $\|\mathbf{r}(\boldsymbol{\xi})\| = 0$ is the continuous expansion of the modulation, see Appendix A.

1) Averaged Reference Direction: The averaged reference direction $\mathbf{r}(\boldsymbol{\xi})$ is evaluated as the weighted sum:

$$\mathbf{r}(\boldsymbol{\xi}) = \frac{1}{\Gamma^{\min}} \sum_{o=1}^{N^{\text{obs}}} w_o(\boldsymbol{\xi})\mathbf{r}_o(\boldsymbol{\xi}) \;, \quad \Gamma^{\min} = \min_{o \in 1..N^{\text{obs}}} \Gamma_o(\boldsymbol{\xi}) \tag{8}$$

and the reference directions $\mathbf{r}_o(\boldsymbol{\xi})$ are evaluated for all obstacles as described in (4). Additionally, the influence weights $w_o(\boldsymbol{\xi})$ are given by

$$w_o(\boldsymbol{\xi}) = \begin{cases} \hat{w}_o(\boldsymbol{\xi})/\hat{w}^{\text{sum}} & \text{if } \hat{w}^{\text{sum}} > 1 \\ \hat{w}_o(\boldsymbol{\xi}) & \text{otherwise} \end{cases}, w^{\text{sum}} = \sum_o \hat{w}_o(\boldsymbol{\xi}) \tag{9}$$

which is a function of the distance Gamma, see (3), and evaluated for each obstacle o as follows:

$$\hat{w}_o(\boldsymbol{\xi}) = \left(\frac{D^{\text{scal}}}{D_o(\boldsymbol{\xi})} \right)^s \qquad \text{with} \quad D_o(\boldsymbol{\xi}) = \Gamma_o(\boldsymbol{\xi}) - 1 \tag{10}$$

where $s \in \mathbb{R}_{>0}$ is the scaling potential and $D^{\text{scal}} \in \mathbb{R}_{>0}$. We choose $s = 2$ and $D^{\text{scal}} = 1$.

2) Summing of the Normal Direction: The basis matrix $\mathbf{E}(\boldsymbol{\xi})$ in (6) needs to have full rank; thus, the tangent basis $\mathbf{e}_{(\cdot)}(\boldsymbol{\xi})$ and the reference direction $\mathbf{r}(\boldsymbol{\xi})$ have to be linearly independent (see Appendix A). The normal direction $\mathbf{n}(\boldsymbol{\xi})$, which is used to obtain the tangent basis, is designed to never be perpendicular to the reference direction, i.e., $\langle \mathbf{n}(\boldsymbol{\xi}),\, \mathbf{r}(\boldsymbol{\xi}) \rangle > 0, \; \{\boldsymbol{\xi} \in \mathbb{R}^d : \mathbf{r}(\boldsymbol{\xi}) \neq \mathbf{0}\}$.

For this, we first compute the normal offset for all obstacles:

$$\mathbf{n}^{\Delta}(\boldsymbol{\xi}) = \sum_{o=1}^{N^{\text{obs}}} w_o(\boldsymbol{\xi}) \left(\mathbf{n}_o(\boldsymbol{\xi}) - \mathbf{r}_o(\boldsymbol{\xi}) \right) \tag{11}$$

where $N^{\text{obs}} \in \mathbb{N}_{\geq 0}$ is the number of obstacles.

Finally, the weighted normal vector is obtained as:

$$\mathbf{n}(\boldsymbol{\xi}) = \hat{\mathbf{n}}(\boldsymbol{\xi})/\|\hat{\mathbf{n}}(\boldsymbol{\xi})\| \quad \text{with} \quad \hat{\mathbf{n}}(\boldsymbol{\xi}) = c\frac{\mathbf{r}(\boldsymbol{\xi})}{\|\mathbf{r}(\boldsymbol{\xi})\|} + \mathbf{n}^{\Delta}(\boldsymbol{\xi})$$

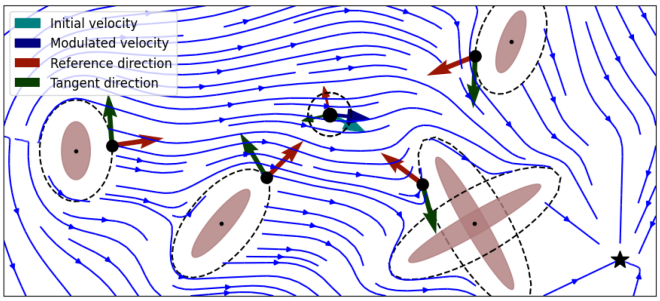


Fig. 1. A single *virtual* obstacle is constructed with corresponding reference and tangent direction. This ensures the absence of minima in *star worlds* around analytical obstacles (brown).

$$c = \begin{cases} 1 & \text{if } p < \frac{\sqrt{2}}{2} \\ \sqrt{2}p & \text{otherwise} \end{cases}, \quad p = (-1)\frac{\langle \mathbf{r}(\boldsymbol{\xi}),\, \mathbf{n}^{\Delta}(\boldsymbol{\xi}) \rangle}{\|\mathbf{r}(\boldsymbol{\xi})\|\, \|\mathbf{n}^{\Delta}(\boldsymbol{\xi})\|} \tag{12}$$

The scaling factor $c \in [1, \sqrt{2}]$ ensures that the decomposition matrix $\mathbf{E}(\boldsymbol{\xi})$ is invertible.

B. Eigenvalue Matrix

The diagonal eigenvalue matrix is given as:

$$\mathbf{D}(\boldsymbol{\xi}) = \text{diag}\left(\lambda^r(\boldsymbol{\xi}), \lambda^e(\boldsymbol{\xi}), \ldots, \lambda^e(\boldsymbol{\xi})\right) \tag{13}$$

with eigenvalues in reference direction $\lambda^r(\boldsymbol{\xi}) \in]0, 1[$ and in tangent $\lambda^e(\boldsymbol{\xi}) \in]1, 2[$ are a function of the averaged reference direction $\mathbf{r}(\boldsymbol{\xi})$:

$$\lambda^r(\boldsymbol{\xi}) = 1 - \|\mathbf{r}(\boldsymbol{\xi})\|^{\rho} \qquad \lambda^e(\boldsymbol{\xi}) = 1 + \|\mathbf{r}(\boldsymbol{\xi})\|^{\rho} \tag{14}$$

where $\rho \in \mathbb{R}_{>0}$ is the reactivity, see [16] for more information.

Theorem 1: Consider a star- world as defined in (4) with N^{obs} obstacles. Any trajectory $\{\boldsymbol{\xi}\}_t$, which starts within the free space $\mathscr{X}^e$ and evolves according to (5) and (12), will stay in free space, i.e., $\{\boldsymbol{\xi}\}_t \in \mathscr{X}^e \; \forall\, t$. Furthermore, if the motion results from linear nominal dynamics $\mathbf{v}$ as given in (2), the final dynamics are stable and have a unique minimum at the attractor $\boldsymbol{\xi}^a \in \mathscr{X}^f$, i.e., $\{\boldsymbol{\xi} : \|\dot{\boldsymbol{\xi}}\| = 0, \nabla \dot{\boldsymbol{\xi}} > 0\} = \{\boldsymbol{\xi}^a\}$

Proof: see Appendix A.

The proposed approach allows collision avoidance in a star-shaped world similar to [18], but only executes a single modulation, resulting in lower computational cost (see Fig. 1).

IV. Sample-Based Obstacle Avoidance

In many scenarios, the analytical obstacle description is not known at runtime, but the obstacles are perceived as a large number of data points, $\boldsymbol{\xi}_p \in \mathbb{R}^d$, $p = 1, \ldots, N^{\text{pnt}}$, where $N^{\text{pnt}} \in \mathbb{N}_{>0}$ is the total number of points.[2] We extend the notion of virtual obstacle to be applicable to a large number of data points. Furthermore, throughout this section, we assume a circular d-dimensional robot with radius $R \in \mathbb{R}_{>0}$ and a sampling angle of $\delta \ll 1$, see Fig. 2.

[2]In this work we focus on Lidar data, but this could be a point-cloud model.

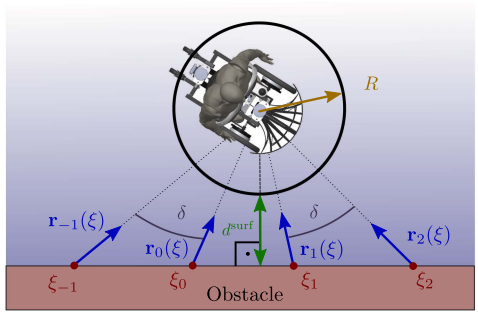


Fig. 2. Mobile robot in front of a flat obstacle.

A. Reference Summing for Sampled Data

Let us define the weighted reference direction

$$\mathbf{r}(\boldsymbol{\xi}) = \sum_{o=1}^{N^{\text{pnt}}} w_p(\boldsymbol{\xi}) \mathbf{r}_p(\boldsymbol{\xi}) \quad \text{with} \quad \mathbf{r}_p(\boldsymbol{\xi}) = \frac{\boldsymbol{\xi}_p - \boldsymbol{\xi}}{\|\boldsymbol{\xi}_p - \boldsymbol{\xi}\|} \tag{15}$$

where the weights $w_p(\boldsymbol{\xi})$ are according to (9), but use the distance measure

$$D_p(\boldsymbol{\xi}) = \|\boldsymbol{\xi}_p - \boldsymbol{\xi}\| - R \qquad \forall\, p = 1..N^{\text{pnt}} \tag{16}$$

and limit the weighted sum

$$w^{\text{sum}} = \min\left(\sum_{p=1}^{N^{\text{pnt}}} \hat{w}_p(\boldsymbol{\xi}), w^{\text{norm}}\right) \tag{17}$$

where $w^{\text{norm}} > 0$ is the maximum distance weight; for more information, see Appendix A.

B. Decomposition Matrix

Each data point be seen as an infinitesimally small circle. Hence, normal and reference direction are the same, and the decomposition matrix $\mathbf{E}(\boldsymbol{\xi})$ is constructed to be orthonormal, i.e., the tangents $\mathbf{e}_{(\cdot)}(\boldsymbol{\xi})$ are orthonormal to the reference direction $\mathbf{r}(\boldsymbol{\xi})$.

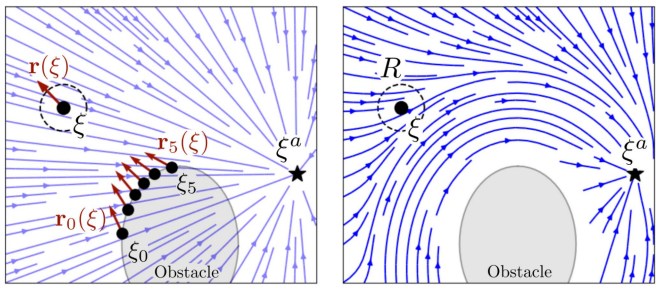
Alternatively, an improved surface estimate can be obtained by clustering the sample points. DBSCAN [19] is a suitable clustering method, as the number of clusters is predicted by the algorithm and the choice of $\epsilon = 2R^{\text{robo}}$ for the neighborhood distance ensures the passage between two clusters. The surface point of each cluster is used to evaluate the normal direction $\boldsymbol{n}(\boldsymbol{\xi})$, and the geometric center for the reference direction $\mathbf{r}(\boldsymbol{\xi})$. Similarly to (12), it is also ensured that the decomposition matrix $\mathbf{E}(\boldsymbol{\xi})$ is invertible.

C. Eigenvalue Matrix

1) Eigenvalue Design Constraints: The magnitude of the reference direction $\|\mathbf{r}(\boldsymbol{\xi})\|$ reaches zero far away from an obstacle, and approaches infinity when on the surface.

Accordingly, the modulated velocity $\dot{\boldsymbol{\xi}}$ needs to be tangent to the surface, or directed away from it when approaching the obstacle to ensure collision avoidance. Conversely, the controller should have no effect when far away from obstacles. These two constraints can be summarized with respect to the eigenvalues in tangent $\lambda^e(\boldsymbol{\xi})$ and reference direction $\lambda^r(\boldsymbol{\xi})$ from (13) as follows

$$\|\mathbf{r}(\boldsymbol{\xi})\| = 0 \quad \Rightarrow \quad \lambda^r(\boldsymbol{\xi}) = \lambda^e(\boldsymbol{\xi}) = 1$$
$$\|\mathbf{r}(\boldsymbol{\xi})\| = 1 \quad \Rightarrow \quad \lambda^r(\boldsymbol{\xi}) / \lambda^e(\boldsymbol{\xi}) = 0$$



(a) Initial vector field **(b)** Modulated vector field

Fig. 3. At each position, the algorithm is only aware of the sample points $\boldsymbol{\xi}_p,\ p = 1..N^{\text{pnt}}$ and not the actual shape of the obstacle in grey (a). The summed reference direction $\mathbf{r}(\boldsymbol{\xi})$ is used to guide the nominal velocity around the obstacle in grey (b).

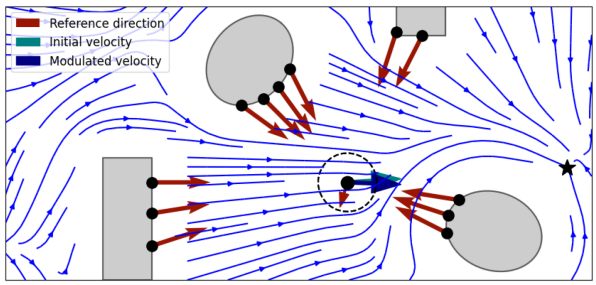


Fig. 4. The algorithm does not need to estimate the number or shape of each individual obstacle. Calculating the weighed reference direction is sufficient to navigate safely in cluttered environments.

$$\|\mathbf{r}(\boldsymbol{\xi})\| \to \infty \quad \Rightarrow \quad |\lambda^e(\boldsymbol{\xi})|/\lambda^r(\boldsymbol{\xi}) = 0 \tag{18}$$

Additionally, $\lambda^e(\boldsymbol{\xi}) \in \mathbb{R}_{>0}$ as the tangent direction should not be reflected.

2) Eigenvalues Based on Reference Direction: Based on the previous boundary conditions, we choose the eigenvalue in reference direction $\lambda^r(\boldsymbol{\xi}) \in [1,\ -1[$ as:

$$\lambda^r(\boldsymbol{\xi}) = \begin{cases} \cos(\frac{\pi}{2}\|\mathbf{r}(\boldsymbol{\xi})\|) & \text{if } \|\mathbf{r}(\boldsymbol{\xi})\| < 2 \\ -1 & \text{otherwise} \end{cases} \tag{19}$$

For close obstacles, and, hence, large averaged reference directions, the eigenvalue is inverted:

$$\lambda^r(\boldsymbol{\xi}) \leftarrow (-1)\lambda^r(\boldsymbol{\xi}) \quad \text{if } \langle \mathbf{r}(\boldsymbol{\xi}),\ \mathbf{v} \rangle < 0,\ \|\mathbf{r}(\boldsymbol{\xi})\| > 1 \tag{20}$$

Additionally, the eigenvalues in tangent direction $\lambda^r(\boldsymbol{\xi}) \in [1,\ 2]$ are given by

$$\lambda^e(\boldsymbol{\xi}) = \begin{cases} 1 + \sin\left(\frac{\pi}{2}\|\mathbf{r}(\boldsymbol{\xi})\|\right) & \text{if } \|\mathbf{r}(\boldsymbol{\xi})\| < 1 \\ 2\sin\left(\frac{\pi}{2\|\mathbf{r}(\boldsymbol{\xi})\|}\right) & \text{otherwise} \end{cases} \tag{21}$$

where π is the circle constant.

Note that the eigenvalues are C^1 smooth, because of $\frac{d}{dx}\sin(x)|_{x=\pi/2} = 0$ and $\frac{d}{dx}\cos(x)|_{x=\pi} = 0$. This enables collision-free movement around single (Fig. 3) or multiple obstacles (Fig. 4) based on point samples only.

Theorem 2: Consider an agent with radius R and a nominal velocity $\{\mathbf{v} \in \mathbb{R}^d : \mathbf{v} \neq \mathbf{0}\}$. Let us assume the existence of N^{obs} obstacles with corresponding boundary points $\mathscr{X}^b$ and exterior points $\mathscr{X}^e$, i.e. the free space, as given in (3), from which the sampled surface points $\boldsymbol{\xi}_p \in \mathbb{R}^d,\ p \in 1,\ \ldots N^{\text{pnt}}$ are known at each time step t.[3]

Any trajectory $\{\boldsymbol{\xi}\}_t$ that starts in free space $\mathscr{X}^e$, and evolves according to (5), (19) and (21), will stay in free space, i.e., $\{\boldsymbol{\xi}\}_t \in \mathscr{X}^e$, and will not stop outside of the gap distance, i.e., $\|\dot{\boldsymbol{\xi}}\| > 0\ \ \forall \boldsymbol{\xi} \in \mathscr{X}^g$.

Proof: see Appendix B.

V. Disparate Obstacle Descriptions

Detection algorithms can fail to identify certain obstacles. Hence, we are left with a mix of analytic obstacles and sampled sensor data. Furthermore, the analytic reconstruction of the obstacles often comes with a significant delay. This results in the asynchronous reception of information on obstacles, with fast sample-based information and slow analytical model acquisition. We address this by proposing an approach to fuse analytic obstacle descriptions (see Section III) with sample-based sensor data (see Section IV).

The first step is to remove sampled obstacle data which has been identified as an analytic obstacle. Hence, the laser scan weights of a point $p = 1..N^{\text{pnt}}$ defined in (17) are set to zero if they belong to an obstacle o:

$$\hat{w}_p(\boldsymbol{\xi}) \leftarrow 0 \quad \text{if } \exists o \in \{1,\ ..,\ N^{\text{obs}}\} : \Gamma_o(\boldsymbol{\xi}_p) \leq 1 \tag{22}$$

A. Fusion Weights of Sampled and Analytic Obstacles

The weight is used to evaluate the reference direction of the mixed environment:

$$\mathbf{r}(\boldsymbol{\xi}) = w^p(\boldsymbol{\xi})\mathbf{r}^p(\boldsymbol{\xi}) + w^o(\boldsymbol{\xi})\mathbf{r}^o(\boldsymbol{\xi}) \tag{23}$$

where $\mathbf{r}^o(\boldsymbol{\xi})$ is the averaged reference direction from all the obstacles, as given in (8), and $\mathbf{r}^p(\boldsymbol{\xi})$ the averaged reference direction from the sampled data points, as given in (15). The magnitude of the reference direction of the sampled and analytic obstacles gives us information about the proximity to obstacles. Hence, the reference magnitude can be used to calculate the importance weight of the sampled data $w^p \in [0, 1]$ and analytic obstacles $w^o \in [0, 1]$ as:

$$\begin{bmatrix} w^p \\ w^o \end{bmatrix} = \frac{1}{p+o}\begin{bmatrix} p \\ o \end{bmatrix} \quad \text{with } \begin{cases} p = 1/\left(1 - \|\bar{\mathbf{r}}^p(\boldsymbol{\xi})\|\right) - 1 \\ o = 1/\left(1 - \|\bar{\mathbf{r}}^o(\boldsymbol{\xi})\|\right) - 1 \end{cases} \tag{24}$$

As a result, navigation is possible in disparate environments, as shown in Fig. 5.

The analytical description has the additional advantage that it often provides information about the velocity and shape of the obstacles. This can be used to ensure collision avoidance in dynamic environments. We propose including the velocity similar to [18] for a fused scenario, as follows:

$$\dot{\boldsymbol{\xi}} = \mathbf{M}(\boldsymbol{\xi}, \mathbf{v})\left(\mathbf{v} - \dot{\boldsymbol{\xi}}^{\text{d,tot}}\right) + \dot{\boldsymbol{\xi}}^{\text{d,tot}} \quad \dot{\boldsymbol{\xi}}^{\mathbf{d},\text{tot}} = \mathbf{w}^{\mathbf{o}}\dot{\boldsymbol{\xi}}^{\text{tot}} \tag{25}$$

where $\dot{\boldsymbol{\xi}}^{\text{tot}} \in \mathbb{R}^d$ is the mean velocity of all analytical obstacles.

B. Asynchronous Evaluation

The disparate data is combined using a controller with three inputs: the velocity command from the human operator $\mathbf{v}$, the

[3]The sensor update rate $f = 1/(t_{i+1} - t_i)$ is expected to be larger than the relative motion $\Delta\mathbf{v}$ and the inverse of the distance, i.e., $f \gg \|\Delta\mathbf{v}\|/\|\boldsymbol{\xi} - \boldsymbol{\xi}_p\|$.

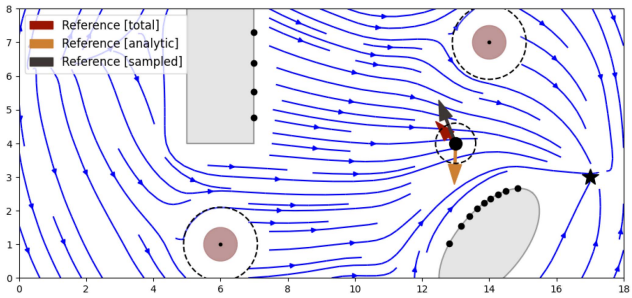


Fig. 5. Analytical obstacle descriptions (brown) and sampled obstacle data (black dots on the gray obstacles) are fused by the algorithm. The weighting uses the corresponding reference directions.

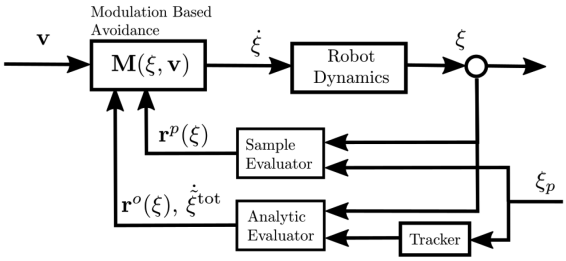


Fig. 6. FOA uses the latest information from the sampled and analytic obstacle description. The reference directions are obtained at different rates and are combined asynchronously.

laser scan reading $\boldsymbol{\xi}_p$, $p = 1..N^{\text{pnt}}$, and the estimated position of the robot $\boldsymbol{\xi}$ (see Fig. 6).

The two sensor loops run at different frequencies, with the sample-based evaluator running at a higher update rate than the analytic evaluator. The latter is delayed due to the processing time required to generate analytic obstacle descriptions. However, the analytic descriptions enable improved behavior due to the information about the obstacle's shape and velocity.

C. Uniform Importance Scaling

An obstacle described by sampled data points or an analytic function should have the same effect on the modulation. Adequate scaling is necessary since a single obstacle often corresponds to multiple data points.

Let us consider an obstacle that can be approximated by radius R^{obs} at a distance D. The number of sampling data points N^{pnt} can be approximated as:[4]

$$N^{\text{pnt}} \sim \left(R^{\text{obs}}/\tan(\delta)D\right)^{d-1} \sim \left(R^{\text{obs}}/\delta D\right)^{d-1} \tag{26}$$

It follows that the scaling weight of the sampled data is set to be proportional to the sampling angle δ:

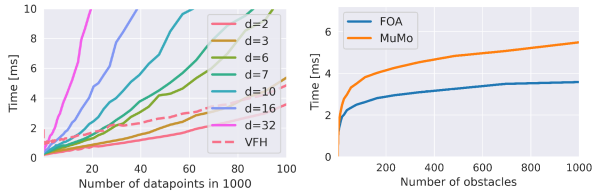
$$s^{\text{pnt}} \approx \delta^{d-1} \tag{27}$$

Moreover, the scaling weight of the analytic obstacle description is dependent on the size and distance.

$$s^{\text{obs}} \approx \left(R^{\text{obs}}/D\right)^{d-1} \tag{28}$$

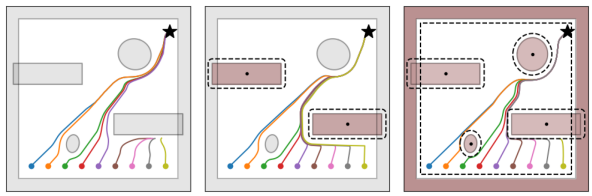
As a result, we set the distance scaling to $D^{\text{scal}} = 2\pi/\delta^{(d-1)}$ in mixed environments.

[4]If the obstacle sampling is volumetric, e.g., when obtaining a collision model of the robot in higher dimensions, the power value of the right side is d, i.e., $N^{\text{pnt}} \sim (\cdot)^d$.



(a) Sample data (b) Analytic data

Fig. 7. The computational complexity of the FOA grows linearly with the number of data points (a), but is lower than for VFH [2], [3] in two dimensions. Similarly, using analytic obstacles, the FOA outperforms the baseline with multiple modulations (MuMo) [18].



(a) Sampled (b) Disparate (c) Analytic

Fig. 8. FOA can access sampled information of the obstacles' surface in gray only (a), an analytic obstacle model (c), or a disparate mix of sampling and analytic description (b).

VI. Experimental Validation

A. Computational Speed

FOA has a computational complexity of $\mathcal{O}(dN^{\text{obs}})$. Thus, it is lower than when compared to similar approaches that have a complexity of $\mathcal{O}(d^{2.4}N^{\text{obs}})$, see [18]. This was confirmed by the low computational time observed during the experiments for sampled data and analytic obstacle descriptions (see Fig. 7). Often the most important decrease of computation time of the FOA compared to the baseline is due to using sampled sensor data directly without the need for an analytic obstacle description, see the experiments in Section VI-C2.

B. Convergence Analysis Using Disparate Sensor Data

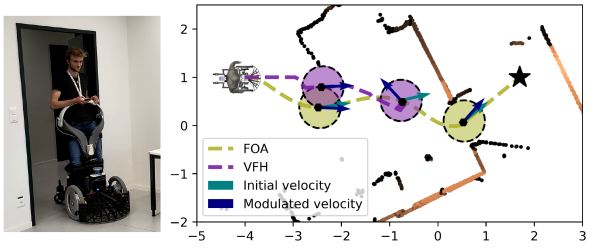
FOA and two baselines were applied to simulated, two-dimensional environments with four obstacles and a surrounding wall (Fig. 8).[5] The shape and pose of the elliptical obstacles were randomly obtained in the top-right and bottom-left corners, respectively. The positions of the two square obstacles and the wall are fixed. The sample-based description (obstacles in gray) is obtained by the algorithm from a simulated laser scan with a sampling angle 0.12 rad.

With increasing environment observations (from sampled to analytic), the convergence rate R increases (Table I). While the Raw-FOA has an increased convergence rate compared to the VFH, the Full-FOA is outperformed by MuMo. However, in both cases, the evaluation time T of the FOA is lower: by a factor of 7 for the sampled environment, and a factor of 2 for the analytic environment. The convergence distance D is similar for all algorithms, but Raw-FOA has a lower mean velocity $\bar{v}$ compared to the other algorithms. Additionally, the FOAs have higher velocity variation Δv to the baselines.

[5]Source code on https://github.com/hubernikus/fast_obstacle_avoidance.git

TABLE I
IN THE THREE SCENARIOS FROM FIG. 8, THE FOAS ARE COMPARED TO THE SAMPLE-BASED APPROACH VFH [2], [3], AND ANALYTIC-OBSTACLES APPROACH USING MULTIPLE MODULATIONS (MUMO) [18]. THE RATIO OF CONVERGENCE R IS WITH RESPECT TO THE TOTAL OF 100 RUNS. THE COMPUTATION TIME C IN $1e^{-}4$ S IS THE AVERAGE EVALUATION OF ONE TIMESTEP. THE DISTANCE TRAVELED D IN M IS THE TOTAL OF ALL TIME STEPS, THE MEAN VELOCITY $\bar{v}$ IN M / S IS AVERAGED FROM ALL TIME STEPS, AND Δv IN M / S IS THE VARIATION IN THE VELOCITY MAGNITUDE

	Sampled		Disparate	Analytic	
	Raw-FOA	VFH	Partial-FOA	Full-FOA	MuMo
R	62%	53%	76%	79%	85%
T	0.3 ± 0.0	2.2 ± 0.2	1.8 ± 0.2	0.9 ± 0.1	1.8 ± 0.3
D	18.9 ± 1.6	18.2 ± 1.6	18.4 ± 1.5	18.5 ± 1.5	18.3 ± 1.5
$\bar{v}$	13.8 ± 1.3	14.9 ± 0.0	14.9 ± 0.0	14.7 ± 1.2	14.9 ± 0.0
Δv	7.3 ± 2.3	5.5 ± 5.1	4.7 ± 5.2	6.0 ± 9.5	4.1 ± 1.8



(a) Doorway **(b)** Comparison of the algorithms passing the doorway

Fig. 9. During the experiments with QOLO passing a narrow doorway (a), FOA allowed the safe navigation of the passage, while VFH was blocked in front of the door (b).

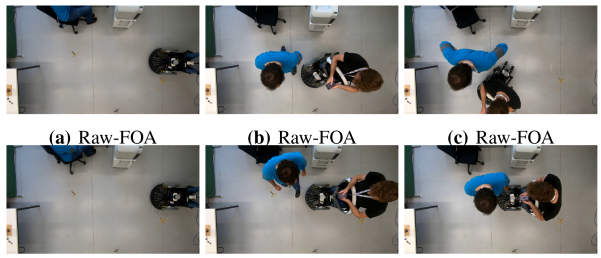
C. Evaluation Using a Semi-Autonomous Wheelchair QOLO

The experimental validation was performed with the semi-autonomous, standing wheelchair QOLO [18]. The nominal velocity $\mathbf{v}$ was obtained from the onboard operator. The wheelchair was non-holonomic with two rigid main wheels and two passive back wheels; its geometry is approximated with a circle of radius $0.45\,\mathrm{m}$. The circle's center $\boldsymbol{\xi}$ denotes the front of the middle of the main wheels at a distance of $d = 6.25 \times 10^{-2}\,\mathrm{m}$. The linear l and angular a velocity command were obtained using the Jacobian $\mathbf{J}$:

$$\begin{bmatrix} l & a \end{bmatrix}^T = \mathbf{J}^{-1}\dot{\boldsymbol{\xi}} \quad \text{with} \quad \mathbf{J}^Q = \mathrm{diag}\,(1 \;\; d) \tag{29}$$

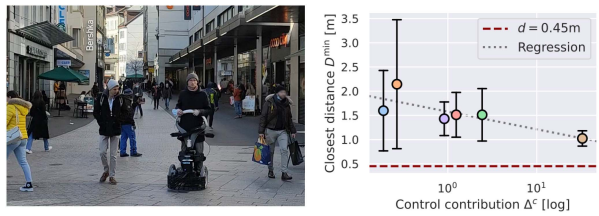
1) Passing Narrow Spaces Based on Sensor Data: The QOLO is equipped with two 3D *Velodyne* Lidars, which are both interpreted as a laser scan by extracting the horizontal 2D row of the Lidar. During the recording, the sensor was placed approximately $30\,\mathrm{cm}$ above ground. The resulting laser scans have an angle of view of $\pm 0.75\,\pi$ rad and an angle increment of $\delta = 7 \times 10^{-3}$ rad, resulting in approximately 650 data points each. While both FOA and the baseline VFH use the (raw) sensor data, VFH often does not outputs smooth control command in cluttered environments. Conversely, FOA controls the robot with a smooth velocity command, even when passing through narrow doorways (Fig. 9).

2) Control Update Rate in Fast-Changing Environments: FOA was implemented in *Python* and evaluated on the onboard computer of QOLO (*Intel(R) Atom(TM) Processor E3950 @* 1.60 GHz), taking 0.4 ms to evaluate a scenario with over 1000 data points. Furthermore, the main control loop ran at 100 Hz, with Lidar information obtained at 20 Hz and the pedestrian tracker providing an update at a frequency of 1 Hz to 5 Hz (depending on the number of tracked persons in close proximity). The operator control input was obtained at a frequency of 20 Hz.



(a) Raw-FOA **(b)** Raw-FOA **(c)** Raw-FOA
(d) MuMo **(e)** MuMo **(f)** MuMo

Fig. 10. The person was initially not detected by the algorithms as they were hidden behind the shelf, (a) and (d). However, as they moved abruptly in front of QOLO, see (b) and (e), the much-shorter reaction time of the pipeline using the Raw-FOA ensured collision avoidance (c), while with MuMo a collision occurred (f).



(a) QOLO in the city center **(b)** Shared control analysis

Fig. 11. During the evaluation with the QOLO robot in Lausanne (a), the control contribution ensured that the closest distance to any pedestrian stayed above the save margin of $0.45\,\mathrm{m}$ during six runs (b). The control contribution is the relative input of the controller, evaluated as $\Delta^c = \|\dot{\boldsymbol{\xi}} - \mathbf{v}\|/\|\mathbf{v}\|$. The closest distance $D^{\min}$ is the average distance of the ten closest Lidar measurements. The relation between the mean of the two measurements was observed to be $\Delta^c = \exp(10.0 - 6.3\,D^{\min})$.

FOA was directly applied on the sensor data; hence, the control delay was approximately $1\,\mathrm{ms}$ (algorithm only). Whereas, MuMo relies on the detection of the person; hence, the delay is approximately 200 ms to 1000 ms (tracker interpretation and control loop).[6] Comparing FOA to MuMo, this is an effective speed up of data reception to control output of over 200.

In the experiment, where a person suddenly appeared in front of the robot at a distance of approximately $0.4\,\mathrm{m}$, the algorithm is required to update the motion in a fraction of a second to ensure the continuous movement of the robot at a speed of approximately $0.5\,\mathrm{m/s}$. While FOA has can ensure a collision-free trajectory, the evaluation time of MuMo is too long to avoid collisions (Fig. 10).

3) Navigation in Outdoor Crowds: The qualitative evaluation using the robot was conducted in the city center of Lausanne, Switzerland.[7] The chosen location is an intersection of six streets and is pedestrianized. This results in a large diversity in pedestrian speeds and directions of movement. The operator navigated with QOLO up and down the street along a transect with a distance of 15 meters (Fig. 11). During the experiments, the controller supported the operator while ensuring collision-free motion.

[6]Moreover, the tracker cannot guarantee detecting pedestrians, and often fails when they suddenly appear in close proximity.

[7]Approved by the EPFL ethics board and the city of Lausanne.

VII. Discussion

This work presents a modulation-based approach for fast obstacle avoidance (FOA) based on dynamical systems. FOA excels in its speed and can be applied to sampled data, such as Lidar or laser scans combined with analytic obstacle descriptions. We proved the absence of local minima in free space, and an agent controlled by FOA only stopped when directly driving towards obstacles. Furthermore, FOA is scalable to higher dimensions. Experimental validation was performed in simulation, static sensor data, and the standing wheelchair QOLO. It was shown that the algorithm avoids collisions while navigating in static and dynamic environments. The evaluation with sampled sensor data showed that the algorithm could pass narrow doorways. Moreover, the algorithm had a short evaluation time onboard the robot.

A. Future Work

Most of the theoretical developments in this work concern multiple-dimensional collision avoidance. While, the implementation was on a mobile agent in crowds for navigation in two dimensions, future implementations will focus on scalability to higher dimensions. Sampled data can be obtained using distance sensors for three-dimensional scenarios. Conversely, in a higher dimensional space, e.g., joint space, the collision data can be obtained in through sampling in simulation [4].

The algorithm was tested in a shared-controller setup. Future work will include a quantitative controller evaluation compared to existing algorithms in various crowd scenarios. The quantitative controller evaluation is intended to include several operators and identify how different driving styles influence the overall behavior. Finally, we believe that the algorithm's quantitative data will enable the design of an intelligent shared controller, which can learn and adapt its behavior based on the operator and the environment, similar to [3].

Appendix

A. Proof of Theorem 1

As the eigenvalues $\lambda^e(\boldsymbol{\xi})$ are equal in all tangent directions $\mathbf{e}_i$, we only analyze the two-dimensional case. Generalization is the immediate result; see [17].

1) Continuously Defined Vector Field: Special consideration has to be given to the case where $\mathbf{r}(\boldsymbol{\xi}) \to \mathbf{0}$, as stated in (7). The final velocity is obtained as:

$$\begin{aligned}\lim_{\mathbf{r}(\boldsymbol{\xi})\to\mathbf{0}} \dot{\boldsymbol{\xi}} &= \lim_{\mathbf{r}(\boldsymbol{\xi})\to\mathbf{0}} \mathbf{E}(\boldsymbol{\xi})\mathbf{D}(\boldsymbol{\xi},\mathbf{v})\mathbf{E}(\boldsymbol{\xi})^T\mathbf{v} \\ &= \lim_{\mathbf{r}(\boldsymbol{\xi})\to\mathbf{0}} \mathbf{E}(\boldsymbol{\xi})\mathrm{diag}\left(\lambda^r(\boldsymbol{\xi},\,\mathbf{v})\,,\lambda^e(\boldsymbol{\xi})\right)\mathbf{E}(\boldsymbol{\xi})^T\mathbf{v} \\ &\approx \mathbf{E}(\boldsymbol{\xi})\mathrm{diag}\left((1+\sin(0)),\,\cos(0)\right)\mathbf{E}(\boldsymbol{\xi})^T\mathbf{v} \\ &= \mathbf{E}(\boldsymbol{\xi})\,\mathbf{I}\,\mathbf{E}(\boldsymbol{\xi})^T\mathbf{v} = \mathbf{v} \end{aligned} \tag{30}$$

Hence, the identity matrix continuously extends the modulation at $\mathbf{r}(\boldsymbol{\xi}) -> \mathbf{0}$.

2) Absence of Local Minima in Free Space: The modulation (7) does not introduce any local minima, if the modulation matrix and, hence, its components $\mathbf{E}(\boldsymbol{\xi})$ and $\mathbf{D}(\boldsymbol{\xi})$ have full rank [17]. Let us first analyse the normal direction $\mathbf{n}(\boldsymbol{\xi})$ and reference direction $\mathbf{r}(\boldsymbol{\xi})$ which are used to construct the decomposition matrix. We have

$$\bar{\mathbf{r}}(\boldsymbol{\xi})^T\hat{\mathbf{n}}(\boldsymbol{\xi}) = \bar{\mathbf{r}}(\boldsymbol{\xi})^T(c\bar{\mathbf{r}}(\boldsymbol{\xi}) + \mathbf{n}^\Delta(\boldsymbol{\xi})) = c + \bar{\mathbf{r}}(\boldsymbol{\xi})^T\mathbf{n}^\Delta(\boldsymbol{\xi}) \tag{31}$$

where $\bar{\mathbf{r}}(\boldsymbol{\xi}) = \mathbf{r}(\boldsymbol{\xi})/\|\mathbf{r}(\boldsymbol{\xi})\|$ denotes the normalized reference direction. Further, from the star-shape constraint in (4), we have $\|\mathbf{n}^\Delta(\boldsymbol{\xi})\| < \sqrt{2}$. Consider the case where $\langle\bar{\mathbf{r}}(\boldsymbol{\xi}),\,\mathbf{n}^\Delta(\boldsymbol{\xi})\rangle/\|\mathbf{n}^\Delta(\boldsymbol{\xi})\| > -\sqrt{2/2}$:

$$\bar{\mathbf{r}}(\boldsymbol{\xi})^T\hat{\mathbf{n}}(\boldsymbol{\xi}) = 1 + \bar{\mathbf{r}}(\boldsymbol{\xi})^T\mathbf{n}^\Delta(\boldsymbol{\xi}) \geq 1 - \|\mathbf{n}^\Delta(\boldsymbol{\xi})\|\sqrt{2}/2 > 0 \tag{32}$$

For the case of $\langle\mathbf{r}(\boldsymbol{\xi}),\,\mathbf{n}^\Delta(\boldsymbol{\xi})\rangle/\|\mathbf{n}^\Delta(\boldsymbol{\xi})\| \leq -\sqrt{2}/2$ we have:

$$\begin{aligned}\bar{\mathbf{r}}(\boldsymbol{\xi})^T\hat{\mathbf{n}}(\boldsymbol{\xi}) &= -\sqrt{2}\bar{\mathbf{r}}(\boldsymbol{\xi})^T\mathbf{n}^\Delta + \bar{\mathbf{r}}(\boldsymbol{\xi})^T\mathbf{n}^\Delta(\boldsymbol{\xi}) \\ &= \left(\sqrt{2}-1\right)\left(-\bar{\mathbf{r}}(\boldsymbol{\xi})^T\mathbf{n}^\Delta(\boldsymbol{\xi})\right) \\ &\geq \left(\sqrt{2}-1\right)\|\mathbf{n}^\Delta(\boldsymbol{\xi})\|\sqrt{2}/2 > 0 \end{aligned} \tag{33}$$

It follows that the decomposition matrix $\mathbf{E}(\boldsymbol{\xi})$ has full rank.

Let us ensure that the eigenvalues of the stretching matrix $\mathbf{D}(\boldsymbol{\xi})$ are strictly positive. In tangent direction, from (14), we have $\lambda^e(\boldsymbol{\xi}) \geq 1$. Conversely, the eigenvalue in reference direction is limited as:

$$\lambda^r(\boldsymbol{\xi})1 - \|\mathbf{r}(\boldsymbol{\xi})\|^\rho = 1 - \underbrace{\frac{1}{\Gamma^{\min}}}_{<1,\ \boldsymbol{\xi}\in\mathscr{X}^e,\ (3)} \underbrace{\left\|\sum_{o=1}^{N^{\mathrm{obs}}} w_o(\boldsymbol{\xi})\mathbf{r}_o(\boldsymbol{\xi})\right\|}_{\sum_o w_o=1\,,\ \|\mathbf{r}_o\|=1\ \forall\, o} > 0 \tag{34}$$

Thus, $\mathbf{D}(\boldsymbol{\xi})$ has full rank in free space. Since both matrices have full rank, it follows that the modulation has a non-trivial result as long as the input velocity is nonzero.

3) Saddle Point Only on the Obstacles' Surfaces: The avoidance of multiple obstacles is defined as the modulation with respect to a single *virtual* obstacle with boundary function $\hat{\Gamma}(\boldsymbol{\xi})$. As we approach an obstacle o, we have $\hat{\Gamma}(\boldsymbol{\xi}) \to \Gamma_o(\boldsymbol{\xi})$ (see Section III-A). Using the proofs from [17], we can conclude that: (1) there is no minimum on the surface of the obstacle o; hence, there are no minima for the virtual obstacle; and (2) there is impenetrability of the virtual obstacle $\hat{o}$ and, hence, any obstacle o as we approach its surface.

Finally, any trajectory starting in free space $\mathscr{X}^e$ will remain in free space. ■

B. Proof of Theorem 2

Analogously to the proof in Appendix A, the two-dimensional analysis generalizes to higher dimensions.

1) Absence of Local Minima in Free Space: The modulated DS from (5) has the same equilibrium points as the original system, if $\det(\mathbf{M}(\boldsymbol{\xi})) \neq 0$; see [18] for further information. Since the basis matrix $\mathbf{E}(\boldsymbol{\xi})$ is orthonormal, the modulation matrix is only *rank-deficient* if any of the eigenvalues are 0. From (19), we know that the only trivial eigenvalues are at $\lambda^e(\boldsymbol{\xi})|_{\|\mathbf{r}(\boldsymbol{\xi})\|\to\infty} = 0$ and $\lambda^r(\boldsymbol{\xi})|_{\|\mathbf{r}(\boldsymbol{\xi})\|=1} = 0$.

It is, hence, sufficient to limit the magnitude of the reference direction to $\|\mathbf{r}(\boldsymbol{\xi})\| < 1$. Let us analyze the *maximum repulsive environment* i.e., where the distance to the surface D_p in one

half-plane is D^{gap}, and infinite in the other half-plane:

$$D_p = \begin{cases} D^{\text{gap}} & \text{if } \langle \mathbf{r}(\boldsymbol{\xi}),\, \boldsymbol{\xi}_p \rangle < 0 \\ \infty & \text{otherwise} \end{cases} \quad p = 1..N^{\text{pnt}} \tag{35}$$

Hence, the magnitude of the reference direction is evaluated as:

$$\begin{aligned} \|\mathbf{r}(\boldsymbol{\xi})\| &\leq \sum_{i=0}^{N/2} \frac{D^{\text{scal}}}{D^{\text{gap}}} w^{\text{sum}} \sin(i\delta) \approx w^{\text{sum}} \frac{D^{\text{scal}}}{D^{\text{gap}}} \int_0^{\pi/\delta} \sin(\phi\delta)\, d\phi \\ &= w^{\text{sum}} \frac{D^{\text{scal}}}{D^{\text{gap}}} \frac{2}{\delta} \leq w^{\text{norm}} \frac{D^{\text{scal}}}{D^{\text{gap}}} \frac{2}{\delta} \end{aligned} \tag{36}$$

The approximation of the finite sum as an integral holds for a small sampling angle δ and many sample points, as is the case for most scanners.

By choosing $w^{\text{norm}} = (D^{\text{gap}}\delta)/(2D^{\text{scal}})$, the modulated DS does not vanish further than D^{gap} away from any obstacle.

2) Impenetrability: Let us observe the modulated velocity $\dot{\boldsymbol{\xi}}$ as we approach the surface, i.e., $\exists p : D_p \to 0$ (see Fig. 2), from (15) follows that $\|\mathbf{r}(\boldsymbol{\xi})\| \to \infty$ and hence $\lambda^e(\boldsymbol{\xi}) = 2\sin(\pi/(2\|\mathbf{r}(\boldsymbol{\xi})\|)) = 0$. With this, we can rewrite the modulated velocity as

$$\begin{aligned} \dot{\boldsymbol{\xi}} &= \mathbf{E}(\boldsymbol{\xi}) \text{diag}\left(\lambda^r(\boldsymbol{\xi}),\ \lambda^e(\boldsymbol{\xi})\right) \mathbf{E}(\boldsymbol{\xi})^T \mathbf{v} \\ &= \left[\bar{\mathbf{r}}(\boldsymbol{\xi})\ \mathbf{e}(\boldsymbol{\xi})\right] \text{diag}\left(\lambda^r(\boldsymbol{\xi}),\ 0\right) \left[\bar{\mathbf{r}}(\boldsymbol{\xi})\ \mathbf{e}(\boldsymbol{\xi})\right]^T \mathbf{v} \\ &= \lambda^r(\boldsymbol{\xi}) \left[\bar{\mathbf{r}}(\boldsymbol{\xi})\ \mathbf{0}\right] \text{diag}(1,\ 0) \left[\bar{\mathbf{r}}(\boldsymbol{\xi})\ \mathbf{0}\right]^T \mathbf{v} \\ &= \lambda^r(\boldsymbol{\xi}) \langle \bar{\mathbf{r}}(\boldsymbol{\xi}),\, \mathbf{v} \rangle \bar{\mathbf{r}}(\boldsymbol{\xi}) \end{aligned} \tag{37}$$

Furthermore, from (19), we have

$$\lambda^r(\boldsymbol{\xi}) \begin{cases} > 0 & \text{if } \langle \mathbf{r}(\boldsymbol{\xi}),\, \mathbf{v} \rangle > 0 \\ \leq 0 & \text{otherwise} \end{cases} \tag{38}$$

Consequently, the Neuman boundary condition holds true:

$$\begin{aligned} \langle \mathbf{n}(\boldsymbol{\xi}),\, \dot{\boldsymbol{\xi}} \rangle &= \mathbf{n}(\boldsymbol{\xi})^T \left(\lambda^r \langle \bar{\mathbf{r}}(\boldsymbol{\xi}),\, \mathbf{v} \rangle \bar{\mathbf{r}}(\boldsymbol{\xi})\right) \\ &\geq \underbrace{\arccos\left(\delta/2\right)}_{\approx 1,\ \text{since } \delta \ll 1} \underbrace{\lambda^r \langle \bar{\mathbf{r}}(\boldsymbol{\xi}),\, \mathbf{v} \rangle}_{\geq 0,\ \text{from (38)}} \geq 0 \end{aligned} \tag{39}$$

As a result, we have a smooth vector field with no local minima away from the obstacles and proven impenetrability. ■

C. Practical Considerations

1) Tail Negligence: The modulation-based obstacle avoidance is active even when the nominal dynamics $\mathbf{v}$ are already moving away from an obstacle. This can be an undesired effect since, in this case, the unconstrained velocity already ensures collision avoidance. This *tail* effect can be reduced by modifying the wake eigenvalues while still ensuring a smooth vector field by adapting the eigenvalues as follows:

$$\begin{aligned} \tilde{\lambda}^e(\boldsymbol{\xi}) &= p\, s + (1 - p\, s)\lambda^e(\boldsymbol{\xi}) \\ \tilde{\lambda}^r(\boldsymbol{\xi}) &= \text{sgn}(s) p\, \tilde{\lambda}^e(\boldsymbol{\xi}) \ + (1 - \text{sgn}(s) p)\lambda^r(\boldsymbol{\xi}) \end{aligned} \tag{40}$$

with the weights given as

$$p = \min\left(1, \frac{1}{\|\mathbf{r}(\boldsymbol{\xi})\|}\right) \quad s = \max\left(0, \frac{\langle \mathbf{r}(\boldsymbol{\xi}),\, \mathbf{v} \rangle}{\|\mathbf{r}(\boldsymbol{\xi})\|\, \|\mathbf{v}\|}\right)^c \tag{41}$$

where $c \in \mathbb{R}_{>0}$ is the power weight; we choose $c = 0.2$.

2) Decreasing Tail Weight: The importance of obstacles in the wake of the nominal velocity $\mathbf{v}$ can be further decreased by modifying the weight. For this, we restate the weight from (9) as:

$$\begin{aligned} &\hat{w}_o \leftarrow \hat{w}_o \left(\frac{\hat{w}_o}{\sum_i \hat{w}_o}\right)^{1/k} \text{ with } \ k = 1 - \frac{\langle \mathbf{v},\, \mathbf{r}_o(\boldsymbol{\xi}) \rangle}{\|\mathbf{v}\|\, \|\mathbf{r}_o(\boldsymbol{\xi})\|} \\ &\forall\, o = 1..N^{\text{obs}} \end{aligned} \tag{42}$$

ACKNOWLEDGMENT

The authors would like to thank David Gonon, Albéric de Lajarte and Loïc Niederhauser for their support in running the experiments on QOLO.

REFERENCES

[1] J. Borenstein and Y. Koren, "The vector field histogram-fast obstacle avoidance for mobile robots," *IEEE Trans. Robot. Automat.*, vol. 7, no. 3, pp. 278–288, Jun. 1991.
[2] S. P. Levine, D. A. Bell, L. A. Jaros, R. C. Simpson, Y. Koren, and J. Borenstein, "The NavChair assistive wheelchair navigation system," *IEEE Trans. Rehabil. Eng.*, vol. 7, no. 4, pp. 443–451, Dec. 1999.
[3] Q. Li, W. Chen, and J. Wang, "Dynamic shared control for human-wheelchair cooperation," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2011, pp. 4278–4283.
[4] M. Koptev, N. Figueroa, and A. Billard, "Real-time self-collision avoidance in joint space for humanoid robots," *IEEE Robot. Automat. Lett.*, vol. 6, no. 2, pp. 1240–1247, Apr. 2021.
[5] P. Fiorini and Z. Shiller, "Motion planning in dynamic environments using velocity obstacles," *Int. J. Robot. Res.*, vol. 17, no. 7, pp. 760–772, 1998.
[6] D. K. M. Kufoalor, E. F. Brekke, and T. A. Johansen, "Proactive collision avoidance for ASVS using a dynamic reciprocal velocity obstacles method," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 2402–2409.
[7] E. Prassler, J. Scholz, and P. Fiorini, "A robotics wheelchair for crowded public environment," *IEEE Robot. Automat. Mag.*, vol. 8, no. 1, pp. 38–45, Mar. 2001.
[8] G. Bardaro, L. Bascetta, E. Ceravolo, M. Farina, M. Gabellone, and M. Matteucci, "MPC-based control architecture of an autonomous wheelchair for indoor environments," *Control Eng. Pract.*, vol. 78, pp. 160–174, 2018.
[9] D. J. Gonon, D. Paez-Granados, and A. Billard, "Reactive navigation in crowds for non-holonomic robots with convex bounding shape," *IEEE Robot. Automat. Lett.*, vol. 6, no. 3, pp. 4728–4735, Jul. 2021.
[10] A. D. Ames, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs with application to adaptive cruise control," in *Proc. 53rd IEEE Conf. Decis. Control*, 2014, pp. 6271–6278.
[11] M. F. Reis, A. P. Aguiar, and P. Tabuada, "Control barrier function-based quadratic programs introduce undesirable asymptotically stable equilibria," *IEEE Control Syst. Lett.*, vol. 5, no. 2, pp. 731–736, Apr. 2021.
[12] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *Int. J. Robot. Res.*, vol. 5, no. 1, pp. 90–98, 1986.
[13] E. Rimon and D. E. Koditschek, "Exact robot navigation using artificial potential functions," *IEEE Trans. Robot. Automat.*, vol. 8, no. 5, pp. 501–518, Oct. 1992.
[14] S. Loizou and E. Rimon, "Mobile robot navigation functions tuned by sensor readings in partially known environments," *IEEE Robot. Automat. Lett.*, vol. 7, no. 2, pp. 3803–3810, Apr. 2022.
[15] H. J. S. Feder and J.-J. Slotine, "Real-time path planning using harmonic potentials in dynamic environments," in *Proc. IEEE Int. Conf. Robot. Automat.*, 1997, pp. 874–881.
[16] S. M. Khansari-Zadeh and A. Billard, "A dynamical system approach to realtime obstacle avoidance," *Auton. Robots*, vol. 32, no. 4, pp. 433–454, 2012.
[17] L. Huber, A. Billard, and J.-J. Slotine, "Avoidance of convex and concave obstacles with convergence ensured through contraction," *IEEE Robot. Automat. Lett.*, vol. 4, no. 2, pp. 1462–1469, Apr. 2019.
[18] L. Huber, J.-J. Slotine, and A. Billard, "Avoiding dense and dynamic obstacles in enclosed spaces: Application to moving in crowds," *IEEE Trans. Robot.*, vol. 38, no. 5, pp. 3113–3132, Oct. 2022.
[19] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.