2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

From VO to NAO: Reactive Robot Navigation using Velocity and Acceleration Obstacles

Asher Stern, Oz Ben-Yosef, Zvi Shiller[1]

Abstract— **This paper addresses the problem of robot navigation in challenging dynamic environments by extending the Velocity Obstacle (VO) framework to the Nonlinear Acceleration Obstacle (NAO). The NAO represents the set of robot accelerations that would lead to collisions with an obstacle moving along an arbitrary trajectory. By formulating the problem in the acceleration domain, the method allows direct selection of accelerations, the natural control input of second-order systems, to generate safe avoidance maneuvers in complex dynamic environments. Simulation results show that NAO enables real-time collision avoidance while explicitly accounting for both robot kinematics (velocity) and dynamics (acceleration). The proposed framework thus provides a reactive and efficient basis for autonomous navigation in complex dynamic environments.**

I. Introduction

Navigation in dynamic environments has become a central challenge in robotics, with the proliferation of autonomous vehicles, aerial drones, marine vessels, and service robots. These systems must safely maneuver through traffic flows, crowded spaces, and other complex multi-agent settings. A decision system for such tasks must ensure collision avoidance with any number of static and moving obstacles, while generating maneuvers that are dynamically feasible given the robot's kinematic and dynamic constraints.

Early works on motion planning in dynamic environments decomposed the planning problem into two subproblems: path planning and velocity planning [1], [2]. The path planning phase computes a path that avoids all static obstacles, whereas the velocity planning phase adjusts the velocity profile along that path to avoid moving obstacles. Dynamic constraints are satisfied by limiting the slope, direction and curvature of the trajectory in the space-time plane. A similar approach was presented in [3] and [4].

An alternative approach uses what we call "planning in *changing* environments," repeatedly uses a frozen snapshot of the environment at each time step, treating all obstacles, whether moving or not, as static [5], [6], [7]. This approach often produces inefficient, and in some cases unsafe maneuvers [8].

An effective approach to avoiding collisions in dynamic environments is the use of the Velocity Obstacle (VO) [9] that maps obstacles, static or dynamic, to the velocity space of the maneuvering robot. It represents the set of colliding velocities between the robot and an individual obstacle. Selecting a velocity outside the VO of all obstacles ensures collision-free motion while the obstacles are moving at constant velocities.

The VO was later extended to the Nonlinear Velocity Obstacle (NLVO), which accounts for arbitrary known or predicted trajectories of the obstacle [10]. It allows fewer velocity adjustments than the linear version [9] when the obstacle is moving along curved trajectories.

Another variant of the VO is the Reciprocal VO (RVO) [11] [12]. It assumes multi-robot avoidance where each robot is expected to contribute to the avoidance effort. Geometrically, the RVO is a scaled-down version of the original VO so that each robot makes only a partial effort to avoid the other obstacle (by avoiding a smaller VO), letting the other robot reciprocate by sharing the mutual avoidance maneuver.

A comprehensive review of current literature on motion planning using the Velocity Obstacle paradigm can be found in [13].

In this paper, we focus on the obstacle avoidance problem in the acceleration domain by extending the VO to AO (Acceleration Obstacle) and the NLVO to NAO (Nonlinear Acceleration Obstacle). This is motivated by the fact that a robot moving in a dynamic environment is a dynamic and not a kinematic system. The simplest model for such a system is of second order that is driven by acceleration that can be arbitrarily selected subject to the robot's acceleration constraints.

The Acceleration Obstacle, AO, in analogy to the Velocity Obstacle, VO, consists of the constant *accelerations* that would cause collisions between a robot and a moving obstacle. Unlike the VO, the geometric shape of the AO depends on the initial velocities of the robot and the obstacle and the obstacle's initial acceleration, as will be shown later in this paper.

The AO was earlier mentioned in [14], and more recently in [15] in the context of navigation in human crowds. Arguing that AO was impractical since accelerations tend to change frequently and are therefore difficult, the Acceleration Velocity Obstacle, AVO, was proposed [14], which is similar to the VO, except that it accounts for the transition from the current to the target velocity using a proportional feedback law on the acceleration. Our experience suggests that even short bursts of acceleration applied by the moving obstacle can be crucial in selecting the correct avoidance maneuver.

[1] Mechanical Engineering and Mechatronics, Ariel University, Israel. shiller@ariel.ac.il

The AO introduced in [15], was rigorously derived for a robot and obstacle with an initial relative location and velocity, for obstacles moving at a constant relative acceleration. They construct the AO as a union of disks, each expressing the constant relative acceleration that would cause collision between the robot and the obstacle at a specific time. While it was noted that the shape of the AO is curved, the effect of the initial velocities and accelerations on the AO's shape was not fully addressed.

If obstacles (or vehicles) are known to move along general trajectories, then deriving the Nonlinear Acceleration Obstacle (NAO) to reflect their exact motion can result in fewer adjustments by the avoiding vehicle compared to using the AO. In addition, the NAO may be essential in cases where the AO indicates a collision, when in fact there is none because of the vehicle's curved trajectory.

This paper first reviews the derivation of the VO and NLVO and then extends both to the Acceleration Obstacle (AO) and the Nonlinear Acceleration Obstacle (NAO), respectively. The utility of the NAO is demonstrated for a vehicle negotiating successfully through a busy turnaround. The ability to account for arbitrary obstacle trajectories using NAO greatly simplifies the avoidance process, as it requires fewer acceleration adjustments by the maneuvering robot than when using AO, which assumes constant acceleration trajectories.

II. The Velocity Obstacle

We recall the derivation of the velocity Obstacle VO that was first introduced in [9]. It represents this set of forbidden velocities that would cause collision between a robot and an obstacle (static or moving) [9]. The geometry of this set can be easily described in the robot's velocity space. For simplicity, we consider planar robots and obstacles. The robot and obstacles can be of general shapes, however, to reduce the dimensionality of the problem, we assume a circular robot. Growing the obstacles by the radius of the robot transforms the problem into the configuration space where a point robot avoids circular obstacles in the plane, as seen in Fig. 1. It is assumed that the instantaneous states (position and velocity) of the obstacles are either known or measurable.

Henceforth, A denotes a point robot, and B(t) denotes the obstacle at time t with its center located along some arbitrary trajectory $c(t)$.

A. *The Linear Velocity Obstacle, VO*

The VO represents the velocities of A that would result in collision at any time $t = (0, \infty)$. It is useful to identify a subset of VO that would result in collision between A and $B(t)$ at a specific time t.

Consider the relative velocity $v_{A/B}$ shown in Fig. 2. To reach point $p \in B$ at time t, $\boldsymbol{v}_{A/B}$ must satisfy

$$\boldsymbol{v}_{A/B} = \frac{1}{t - t_0} p, \qquad (1)$$

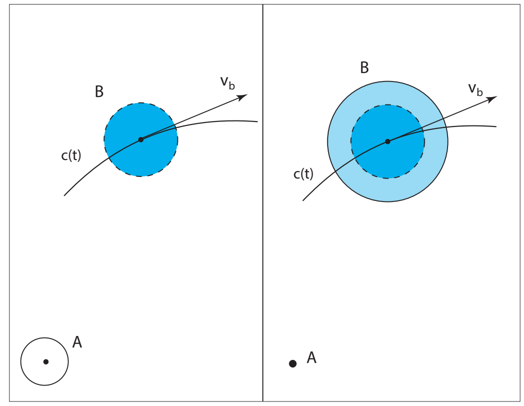

Fig. 1: A moving obstacle

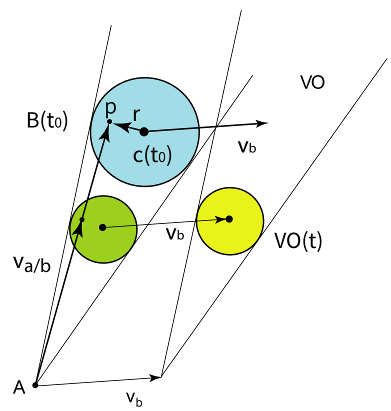

Fig. 2: A Temporal Element of VO.

where p is a position vector relative to A (for brevity, we will denote $A(t_0)$ by A). The mapping of p to $\boldsymbol{v}_{A/B}$ (1) is a homothety transformation $H_{A,k}$ that maps A to itself and maps any other point p to point p' such that Ap and Ap' are collinear and $Ap = kAp'$. A is called the *center* and k the *ratio* of the homothety [16]. Here, homothety $H_{A,k}$ scales the vector p by $k = \frac{1}{t-t_0}$ and positions it at A:

$$\boldsymbol{v}_{A/B} = H_{A,k}(p); k = \frac{1}{t - t_0}. \qquad (2)$$

Substituting for p in (2) the entire set $B(t_0)$, yields the set of all relative velocities, in a frame centered at A, that would result in collision with any point of B at a specific time $t > t_0$, as shown in Fig. 2:

$$RV(t) = H_{A,k}(B(t_0)); k = \frac{1}{t - t_0}. \qquad (3)$$

To emphasize that the shape of $RV(t)$ is a scaled B, located at a distance from A that is inversely proportional to the collision time t, we represent p by

$$p = c(t_0) + r \qquad (4)$$

where $c(t_0)$ is the position of the center of B at time t_0 relative to A, and r is the position of p relative to $c(t_0)$.

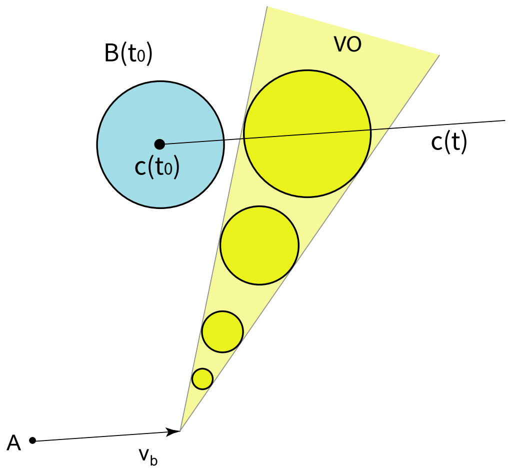

Fig. 3: The Velocity Obstacle VO.

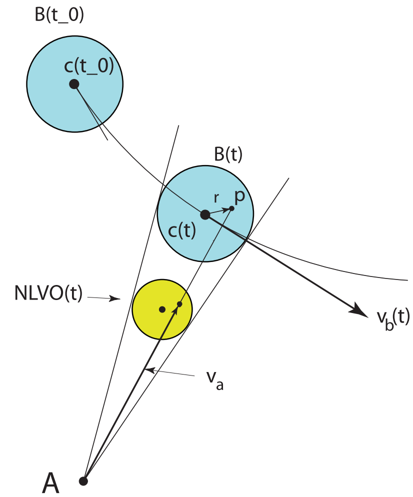

Fig. 4: A temporal element of the non-linear v-obstacle.

The set $RV(t)$ then becomes

$$RV(t) = H_{A,k}(c(t_0)) + H_{c(t_0),k}(B); k = \frac{1}{t - t_0} \tag{5}$$

or

$$RV(t) = \{v|v = \frac{c(t_0) + r}{t - t_0}, r \in B\}. \tag{6}$$

Translating $RV(t)$ by $\boldsymbol{v}_b$ produces the set VO(t), shown in Fig. 2, of all *absolute* velocities of A that would result in collision with any point of B at time $t > t_0$:

$$VO(t) = \boldsymbol{v}_b \oplus RV(t). \tag{7}$$

This leads to the following formal definition of the linear v-obstacle, VO, shown in Fig. 3:

Definition 1: The Linear Velocity Obstacle
Consider at time t_0 a point robot A, located at the origin, and an obstacle B centered at $c(t_0)$ and moving at a constant velocity $\boldsymbol{v}_b$. The linear v-obstacle, VO, consists of the set of all linear velocities of A at time $t = t_0$ that would collide with B at any time $t > t_0$:

$$VO = \boldsymbol{v}_b \oplus \bigcup_t H_{A,k}(B(t_0)); k = \frac{1}{t - t_0}; \forall t > t_0. \tag{8}$$

B. The Non-Linear Velocity Obstacle

The non-linear velocity obstacle (NLVO) applies to the scenario shown in Fig. 4, where, at time t_0, a point robot, A, attempts to avoid an obstacle, B, that at time t_0 is located at $c(t_0)$, and is following a general known trajectory, $c(t)$. The NLVO thus consists of all velocities of A at t_0 that would result in collision with the obstacle at any time $t > t_0$. Selecting a *single* velocity, $\boldsymbol{v}_a$, at time $t = t_0$ outside the NLVO thus guarantees to avoid collision at all times, or

$$(A(t_0) + \boldsymbol{v}_a t) \cap (c(t) + B) = 0 \; ; \; \forall t > t_0. \tag{9}$$

The non-linear v-obstacle is constructed as a union of its temporal elements, NLVO(t), which is the set of all absolute velocities of A, v_a, that would result in collision at a specific time t. Referring to Fig. 4, v_a that would result in collision with point $p \in B(t)$ at time $t > t_0$, expressed in a frame centered at $A(t_0)$, is simply

$$\boldsymbol{v}_a = \frac{c(t) + r}{t - t_0}, \tag{10}$$

where r is the vector to point p in the obstacle's fixed frame.

It is again a homothety transformation, centered at $A(t_0)$ and scaled by $k = \frac{1}{t - t_0}$:

$$\boldsymbol{v}_a = H_{A,k}(c(t) + r); k = \frac{1}{t - t_0}. \tag{11}$$

The set, NLVO(t) of all absolute velocities of A that would result in collision with any point in $B(t)$ at time $t > t_0$ is thus:

$$NLVO(t) = H_{A,k}(B(t)), \;\; k = \frac{1}{t - t_0}. \tag{12}$$

We can rewrite (12) using the Minkowski sum to emphasize the geometric shape of NLVO(t):

$$NLVO(t) = \frac{c(t)}{t - t_0} \oplus \frac{B(t)}{t - t_0}, \;\; \forall t > t_0. \tag{13}$$

Clearly, NLVO(t) is a scaled B, bounded by the cone formed between A and $B(t)$, and located at a distance $\frac{c(t)}{t-t_0}$ from A. Note that the tangency points of the extreme rays of this cone and $B(t)$ are not necessarily the points where A grazes $B(t)$, as discussed later. Note also that the NLVO(t) is independent of $\boldsymbol{v}_b(t)$, since it applies only to $B(t)$ and not to its future or past positions. This leads to the following formal definition of the nonlinear v-obstacle:

Definition 2: The Nonlinear Velocity Obstacle
Let A be a point robot, located at time $t = t_0$ at the origin, and B be an obstacle that is moving along a

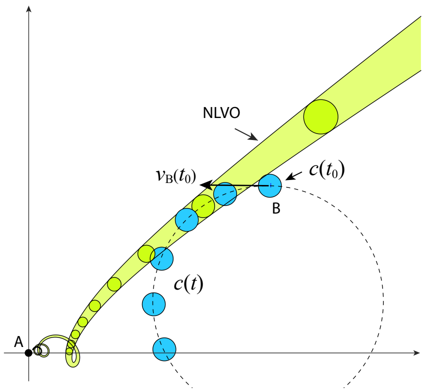

Fig. 5: Construction of the NLVO for obstacle B moving along a circular path $c(t)$ at a constant tangent velocity $\boldsymbol{v}(t_0)$. For a bounded path, the NLVO is a warped cone terminating at $A(t_0)$.

general trajectory $c(t), t = [t_0, \infty)$. The non-linear v-obstacle, NLVO, representing the set of all linear velocities of A that would collide with $B(t)$ at time $t = (t_0, \infty)$, is defined by

$$NLVO = \bigcup_t H_{A,k}(B(t)); k = \frac{1}{t - t_0}; \forall t > t_0. \quad (14)$$

The non-linear v-obstacle is a warped cone as shown in Fig. 5. If $c(t)$ is bounded, then the apex of this cone is at $A(t_0)$. The NLVO can be generated for graphical simulations by drawing its individual temporal elements at discrete time intervals, or by drawing its boundaries [10], which represent velocities that would result in A grazing B.

Fig. 6 demonstrates the importance of NLVO for vehicles moving along a curved road. The VO (blue) anticipates a collision between A and B, whereas the NLVO (yellow) does not, given that both A and B stay in their current lanes. Using NLVO clearly saves unnecessary corrections by A, thus allowing to focus attention only on situations where B deviates from its anticipated trajectory. Fig. 6 shows 4 vehicles moving along a curved road. The maneuvering robot is marked A, and the other three passive vehicles are B, C, and D. The right-hand side of Fig. 6 shows the velocity space of A in its coordinate frame, with the velocity $\boldsymbol{v}_A$ pointing along the vertical axis. Also shown are the linear (in blue) and non-linear (in yellow) velocity obstacles of B, C, and D.

At the position shown, v_A penetrates the linear velocity obstacle of B (in blue), which implies that B is on a collision course with A had both maintained their current velocities. Accounting for B's curved trajectory, which is reflected in its nonlinear velocity obstacle (in yellow), shows no potential collision. Using the linear velocity obstacle would have required A to continuously adjust its velocity to avoid a collision with B. In the case shown, it would require A to slow down or speed up. This, however, is not necessary if B maintains its current course along the middle lane.

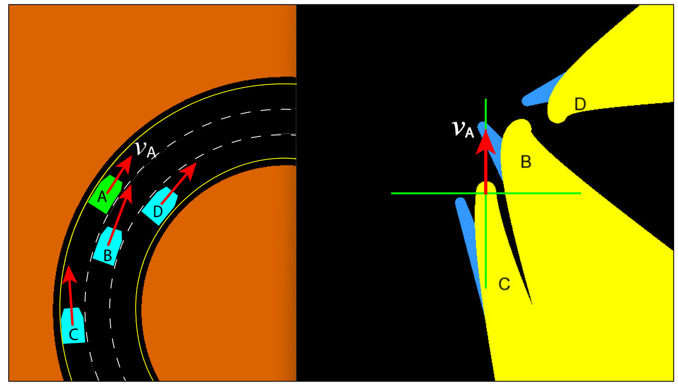

Fig. 6: Comparing linear (blue) and nonlinear (yellow) v-obstacles along a curved road.

C. Static walls

It is often necessary to consider static walls and road boundaries while selecting the velocities of the moving robot. To this end, we specify a time horizon t_h since, without it, the VO of the surrounding walls and boundaries would cover the robot's entire velocity space. This is demonstrated in Fig. 7.

Consider a point robot A at the center of a closed room, as shown on the left in Fig. 7. The velocity obstacle of the room's walls, with an infinite time horizon, covers the entire space, except the origin A, since any nonzero velocity in any direction would eventually result in a collision with the walls. To clear some space for maneuverability, we "push" the VO of each wall by the largest velocity v_h that would traverse the distance d to the wall in some set time horizon t_h:

$$v_h = \frac{d}{t_h}. \quad (15)$$

Any velocity larger than v_h would penetrate the velocity obstacle since it would result in collision at a shorter time than the time horizon t_h. Similarly, any velocity smaller than $\boldsymbol{v}_h$ is safe for the duration $t \le t_h$. The inner boundary of the velocity obstacle of the closed room is a scaled shape of the room's walls, as shown on the right in Fig. 7; the shorter the time horizon, the larger the "free" space. It is easy to show that the inner shape of the truncated VO resembles the inner shape of the boundaries.

III. Acceleration Obstacle (AO)

Following the representation of robot and obstacles in the configuration space (see Fig. 1), we start with the case of a circular obstacle B, initially located at $c(t_0)$ in a coordinate frame centered at A, and moving at a constant acceleration $\boldsymbol{a}_B$ and an initial velocity $\boldsymbol{v}_B(t_0)$, as shown in Fig. 8.

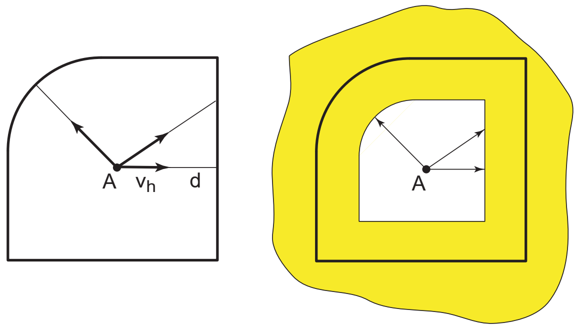

Fig. 7: The velocity obstacle of static walls and boundaries

A. Derivation of the AO

We wish to compute the constant acceleration $a(t)$ of a point robot A, also shown in Fig. 8, that is moving at an initial velocity $\boldsymbol{v}_A(t_0)$ that will reach $c(t)$ at any time $t > t_0$:

$$c(t_0) + \boldsymbol{v}_B(t_0)t + \frac{1}{2}\boldsymbol{a}_B t^2 = \boldsymbol{v}_A(t_0)t + \frac{1}{2}\boldsymbol{a}_A t^2 \tag{16}$$

Solving for $a_A(t_0)$ yields:

$$a_A = \frac{2c(t_0)}{t^2} - \frac{2\boldsymbol{v}_{A/B}(t_0)}{t} + a_B \tag{17}$$

Accounting for all points of B defines the set $\mathrm{AO}(t)$ of

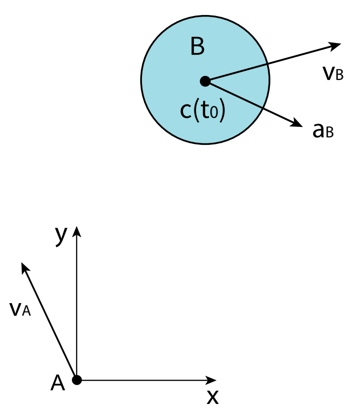

Fig. 8: A moving robot A, and obstacle B that is centered initially at $c(t_0)$ and moving at an initial velocity $\boldsymbol{v}_B$ and a constant acceleration $\boldsymbol{a}_B$.

all constant accelerations that would collide with $B(t)$ at time t:

$$AO(t) = \frac{2B}{t^2} \oplus (\frac{2c(t_0)}{t^2} - \frac{2\boldsymbol{v}_{A/B}(t_0)}{t}) + \boldsymbol{a}_B \tag{18}$$

Integrating (18) over time produces the set $AO \subset \mathbb{R}^2$ of all constant accelerations that would collide with obstacle B at any time $t > t_0$:

$$AO = \bigcup_t AO(t); \quad \forall t > t_0 \tag{19}$$

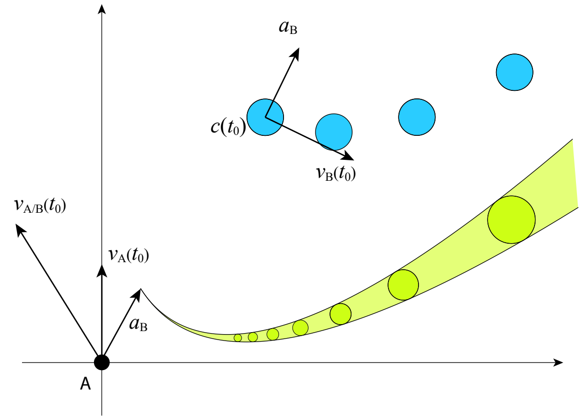

Fig. 9: Constructing the AO for B moving at a constant acceleration with an initial velocity $\boldsymbol{v}_B(t_0)$, and A moving at an initial velocity $\boldsymbol{v}_A(t_0)$.

This leads to the following formal definition of the Acceleration Obstacle AO:

Definition 3: The Acceleration Obstacle (AO)
Consider at time t_0 a point robot A, located at the origin and moving at an initial velocity $\boldsymbol{v}_A(t_0)$, and an obstacle B centered at $c(t_0)$ and moving at an initial velocity $\boldsymbol{v}_B$ and a constant acceleration $\boldsymbol{a}_B$. The Acceleration Obstacle, AO, consists of the set of all accelerations of A at time $t = t_0$ that would collide with B at any time $t > t_0$:

$$AO = \bigcup_t \{\frac{2B}{t^2} \oplus (\frac{2c(t_0)}{t^2} - \frac{2\boldsymbol{v}_{A/B}(t_0)}{t}) + \boldsymbol{a}_B\}; \forall t > t_0 \tag{20}$$

The AO is a warped cone, shifted from A by $\boldsymbol{a}_B$, that is approaching the tangent to the relative velocity $\boldsymbol{v}_{A/B}(t_0)$, as $t \to \infty$, as shown in Fig. 9.

B. Geometric Properties of AO

a) **Similarity between AO and VO**: If $\boldsymbol{v}_B = \boldsymbol{v}_A = 0$, then AO is geometrically identical to VO. The same holds if the relative velocity $\boldsymbol{v}_{A/B} = 0$. It stems from the fact that accelerations with no initial velocities produce straight-line trajectories.

b) **Nonzero initial relative velocity $\boldsymbol{v}_{A/B}$:** if $\boldsymbol{v}_{A/B} \neq 0$ and $\boldsymbol{a}_B = 0$, AO becomes curved. The center line of AO approaches the origin A and coincides with $\boldsymbol{v}_{A/B}$ as $t \to \infty$.
Proof:
Writing Eq. (17) in its x, y components, substituting $a_B = 0$, dividing y/x and taking the limit as $t \to \infty$, yields:

$$\frac{a_y(t)}{a_x(t)} = \frac{c_y(t_0) - v_y(t_0)\,t}{c_x(t_0) - v_x(t_0)\,t} \xrightarrow[t\to\infty]{} \frac{-v_y(t_0)}{-v_x(t_0)},$$

implying that $\boldsymbol{a}(t)$ becomes collinear in opposite direction to $\boldsymbol{v}(t_0)$, as $t \to \infty$.

c) **Relative velocity $\boldsymbol{v}_{A/B}$ pointing toward B**: A self-folding of the AO occurs so that the AO covers the origin while its apex terminates at the origin. This implies that the origin (zero acceleration) is a colliding acceleration.

d) **Geometric interpretation of the apex and the tail of the AO**: Accelerations inside the AO that approach its apex correspond to collisions occurring at distant times, whereas accelerations at the expanding tail of the cone correspond to imminent collisions. Note that the set of reachable accelerations limits the choice of accelerations to a bounded region around the origin. Hence, the expanding portion of the AO may not be reachable in practice.

e) **Obstacle moving at constant acceleration $\boldsymbol{a}_B$**: AO is uniformly offset by $\boldsymbol{a}_B$. From (17) and shown in Fig. 9.

IV. Nonlinear Acceleration Obstacle (NAO)

The Nonlinear Acceleration Obstacle, NAO, accounts for obstacles that are moving on general (nonlinear) trajectories. Its derivation is similar to the derivation of AO except that the left hand side of (16) is replaced with $c(t)$, the actual location of the obstacle center at time t:

$$c(t) = \boldsymbol{v}_A(t_0)t + \frac{1}{2}\boldsymbol{a}_A(t_0)t^2. \qquad (21)$$

Solving for $\boldsymbol{a}_A(t_0)$:

$$\boldsymbol{a}_A(t_0) = \frac{2c(t)}{t^2} - \frac{2\boldsymbol{v}_A(t_0)}{t} \qquad (22)$$

The temporal Nonlinear Acceleration Obstacle is thus:

$$NAO(t) = \frac{2B}{t^2} \oplus (\frac{2c(t)}{t^2} - \frac{2\boldsymbol{v}_A(t_0)}{t}) \qquad (23)$$

Definition 4: The Nonlinear Acceleration Obstacle (NAO)
Consider at time t_0 a point robot A, located at the origin and moving at $\boldsymbol{v}_A$, and an obstacle B centered at $c(t_0)$ and moving along an arbitrary trajectory $c(t), t \in (t_0, t\infty)$. The Nonlinear Acceleration obstacle, NAO, consists of the set of all accelerations of A at time $t = t_0$ that would collide with B at any time $t > t_0$:

$$NAO = \bigcup_t \{\frac{2B}{t^2} \oplus (\frac{2c(t)}{t^2} - \frac{2\boldsymbol{v}_A(t_0)}{t})\}; \forall t > t_0 \qquad (24)$$

Fig. 10 shows the NAO for obstacle B that is moving along a circular trajectory $c(t)$ at a constant tangent velocity v_B. Since $c(t)$ is bounded, as $t \to \infty$, the NAO reduces to a point at the origin A.

A. Geometric Properties of NAO

The NAO properties are similar to those of AO, except that here the obstacle trajectory may be bounded. In this case:

1) NAO is a function of $c(t)$, and therefore cannot be predicted apriori, unlike the AO that is driven by its initial conditions.

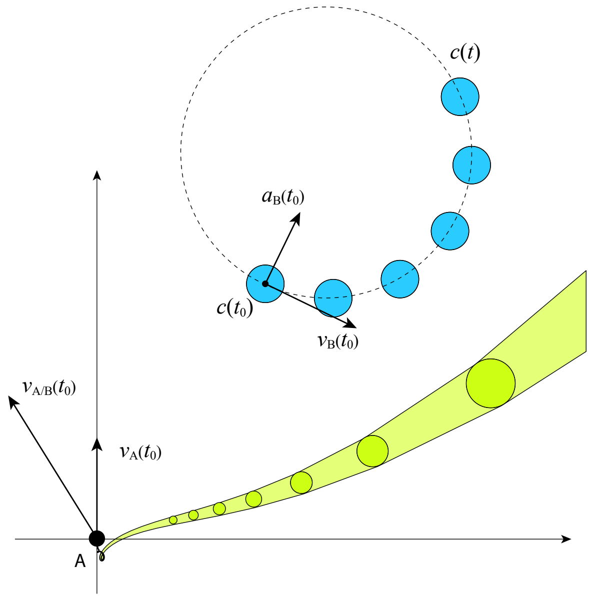

Fig. 10: Constructing the NAO for B moving along a circular trajectory B at a constant tangent velocity, and A moving at an initial velocity $\boldsymbol{v}_A(t_0)$.

2) If $c(t)$ is bounded, the NAO apex converges to the origin.

3) The NAO may curl while approaching the origin, as shown in Fig. 10. This implies that there exist accelerations that may lead to multiple collisions.

V. Selecting an Avoidance Maneuver

The AO and NAO can be used to plan avoidance maneuvers in ways similar to what was suggested in [9] in the context of Velocity Obstacles (VO), namely checking if the current acceleration of the maneuvering vehicle points into the NAO of any obstacle (static or moving), and if it does, selecting a safe acceleration within the set of admissible controls outside of all NAOs, if one exists. This would ensure the avoidance of all obstacles until the next time step when this process would repeat. This can be done using AO if the obstacle is moving at a constant acceleration, or using NAO otherwise to decrease the update rate of the host's acceleration.

It is important to note that AO and NAO account for the host's current velocity. Hence, selecting a safe acceleration, if one exists, would generate a safe avoidance maneuver as long as the obstacle maintains its current trajectory (a constant acceleration with a given initial velocity for AO, and a given forward trajectory for NAO).

We assume the host to be a second-order system for which the acceleration serves as the control variable. This permits the *instantaneous* determination of a safe acceleration, thus ensuring an *immediate* response to an imminent collision. It also offers an exact solution that subsumes velocity-based avoidance [9], [10], [14].

The selection of the safe acceleration can be guided by various heuristics, such as reaching the goal as fast as possible, minimizing the deviation from the current acceleration, or maximizing clearance from other obstacles.

A. Modeling Road Boundaries in the Acceleration Domain

We now wish to model the geometric topology of the admissible space in the acceleration domain, which accounts for the geometry of the road boundaries, similarly to the modeling of static wall boundaries in the velocity domain we introduced earlier in Section II-C. This is crucial when the road boundary is concave, since then the host may run off the road if not modeled properly.

While NAO (and AO) treat obstacles as closed, convex bodies where the robot must remain in the exterior, in the case of road boundaries (e.g., curbs or static walls), the safe physical region is the *interior* of the road, and the unsafe acceleration is located outside the boundary.

The physical boundaries are transformed to a similar shape, scaled, and translated by the relative velocity and time horizon when mapped to the acceleration domain.

Recalling (16), we can compute the acceleration that would cause a point mass to reach the boundary at the time horizon t_h:

$$\boldsymbol{a}_A(t_h) = \frac{2\boldsymbol{d}(t_0)}{t_h^2} - \frac{2\boldsymbol{v}_A(t_0)}{t_h} \qquad (25)$$

where $\boldsymbol{d}(t_0)$ denotes the position vector of a boundary point relative to A, and $\boldsymbol{v}_A(t_0)$ is the current robot velocity.

At an infinite time horizon ($t_h \rightarrow \infty$), the set of admissible accelerations converges to zero, because any non-zero constant acceleration will eventually result in the robot crossing the boundary.

Increasing t_h shrinks the reachable acceleration space by $\frac{1}{t^2}$, and offsets it by $-\frac{\boldsymbol{v}_A}{t}$, as shown in Fig. 11.

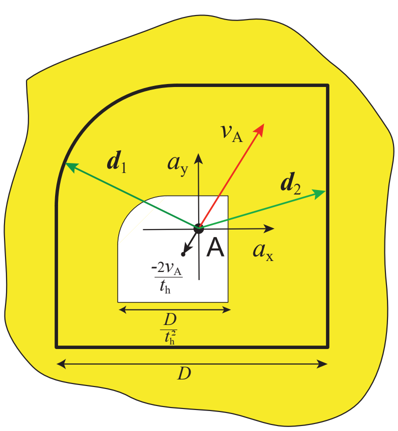

Fig. 11: Transforming the boundaries of a closed space to the acceleration domain. The boundaries are scaled by $1/t_h^2$ and offset by $-2\boldsymbol{v}_A/t_h$.

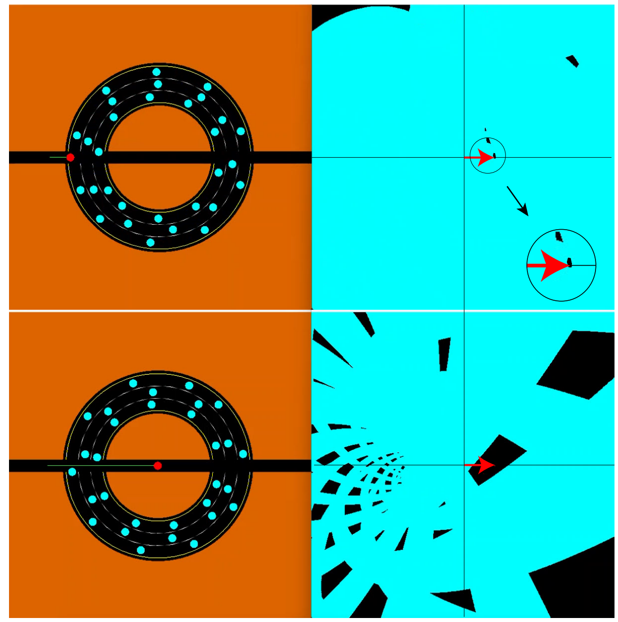

Fig. 12: A vehicle successfully crosses a busy roundabout at a constant horizontal acceleration that pointed to a small collision-free space in the acceleration domain.

VI. Examples

Fig. 12 shows a vehicle (red circle) crossing a busy roundabout at a constant acceleration, while 30 vehicles are moving in three concentric lanes at different speeds. The constant acceleration, shown on the NAO map on the right as a red arrow, was selected to be out of the NAOs of all vehicles, resulting in a successful crossing without collision. Attempting to do the same with AO by frequently adjusting the acceleration to avoid collisions with the nearest vehicles resulted in multiple collisions with the circulating traffic.

Figure 13 shows a maneuvering vehicle (red circle) crossing a busy roundabout with 30 vehicles moving in three concentric lanes at different speeds. At specified time intervals, the vehicle selects accelerations outside the NAOs (not shown), to ensure a safe exit. In each snapshot, the yellow curve represents the local trajectory, generated from the selected acceleration and the current velocity. In this example, the maneuvering vehicle smoothly merged into the roundabout traffic, overtook other vehicles, and reached the exit rapidly and safely without any collisions.

These examples demonstrate the usefulness of the NAO in negotiating complex dynamic environments, and in locally selecting dynamically feasible collision-avoiding accelerations. The resulting vehicle motions are smooth and realistic, resembling the behavior of experienced and careful drivers.

VII. Conclusions

This paper developed a rigorous formulation of the Velocity Obstacle (VO) and its nonlinear extension

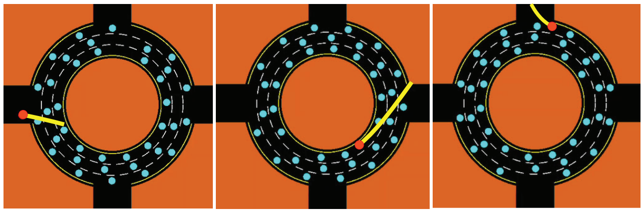

Fig. 13: Navigating a busy roundabout using NAO. The maneuvering vehicle (in red), enters from the left (a) and proceeds through dense traffic to the top exit (c).

(NLVO) using their temporal representations, VO(t) and NLVO(t), respectively. The NLVO can be viewed as a warped cone, a time-scaled map of the obstacle trajectory, where selecting a velocity vector outside the cone guarantees collision avoidance during the defined time horizon. While effective in identifying potential collisions along general trajectories, the NLVO remains insufficient for generating avoidance maneuvers that respect the system's acceleration constraints.

To address this limitation, we extended the VO and NLVO to the acceleration domain by defining the Acceleration Obstacle (AO) and the Nonlinear Acceleration Obstacle (NAO), which represent the robot's colliding accelerations with obstacles moving at constant acceleration (AO) or along arbitrary trajectories (NAO).

The representation in the acceleration domain, and in particular the first introduction of NAO, allows the direct selection of collision-free maneuvers by selecting accelerations within the robot's dynamic constraints. Compared to AO, NAO requires fewer adjustments when obstacles follow general trajectories, resulting in efficient and dynamically feasible maneuvers.

The effectiveness of NAO was demonstrated in challenging navigation scenarios, such as vehicles crossing and navigating busy roundabouts, resulting in safe and smooth maneuvers in environments where earlier VO-based avoidance may fail. This extension is a crucial step toward making collision avoidance compatible with the dynamics of large-scale autonomous systems.

Earlier velocity-based avoidance methods [5], [9], [11], [14] treat the avoidance in the velocity space, whereas the AO and NAO developed here suggest directly the control input of a second-order system, i.e. acceleration, to generate collision-free trajectories. Clearly, acceleration-based avoidance encompasses velocity-based formulations.

It is possible to combine the two approaches by using VO or NLVO for collision detection, and AO or NAO for selecting the control action to avoid the collision.

The Acceleration Obstacle (AO) concept, originally introduced in [15], is generalized here to Nonlinear Acceleration Obstacles (NAO), considering obstacles moving along *arbitrary* trajectories.

The focus here was on constructing the AO and NAO as a union of temporal AO(t) and NAO(t), respectively. Future work will focus on developing analytical definitions of NAO boundaries [17] and extending the framework to more complex robot dynamics and real-world experiments.

VIII. Acknowledgments

The last author wishes to acknowledge the fruitful discussions with Dr. Thierry Fraichard from INRIA, Grenoble, regarding motion planning in the velocity space.

References

[1] K. Kant and S.W. Zucker. Toward efficient trajectory planning: the path-velocity decomposition. *Int. J. Rob. Res.*, 5(3):72–89, 1986.

[2] K. Kant and S.W. Zucker. Planning collision-free trajectories in time-varying environments: a two-level hierarchy. *The Visual Computer*, 3(5):304–313, 1988.

[3] B. H. Lee and C. S. G. Lee. Collision-free motion planning of two robots. *IEEE Trans. Syst. Man Cybern.*, 17(1):21–32, 1987.

[4] C. Ó'Dúnlaing. Motion planning with inertial constraints. *Algorithmica*, 2:431–475, 1987.

[5] D. Fox, W. Burgard, and S. Thrun. The dynamic window approach to collision avoidance. *IEEE Robotics and Automation Magazine*, 4(1):23–33, March 1997.

[6] N. Y. Ko and R. Simmons. The lane-curvature method for local obstacle avoidance. In *IEEE, IROS-98*, pages 1615–1621, 1998.

[7] J. Minguez and L. Montano. Nearness diagram (ND) navigation: collision avoidance in troublesome scenarios. *IEEE Trans. on Robotics and Automation*, 20(1):45–59, February 2004.

[8] R. Parthasarathi and T. Fraichard. An inevitable collision state-checker for a car-like vehicle. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pages 3068–3073. IEEE, 2007.

[9] P. Fiorini and Z. Shiller. Motion planning in dynamic environments using velocity obstacles. *International Journal of Robotics Research*, 17(7):760–772, July 1998.

[10] Z. Shiller, F. Large, and S. Sekhavat. Motion planning in dynamic environments: Obstacle moving along arbitrary trajectories. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2001.

[11] J. van den Berg, Ming L., and D. Manocha. Reciprocal velocity obstacles for real-time multi-agent navigation. In *2008 IEEE International Conference on Robotics and Automation*, pages 1928–1935, 2008.

[12] D. Bareiss and J. van den Berg. Generalized reciprocal collision avoidance. *The International Journal of Robotics Research*, 34(12):1501–1514, 2015.

[13] F. Vesentini, R. Muradore, and P. Fiorini. A survey on velocity obstacle paradigm. *Robotics and Autonomous Systems*, 174:104645, 2024.

[14] J. Van Den Berg, J. Snape, S.J. Guy, and D. Manocha. Reciprocal collision avoidance with acceleration-velocity obstacles. In *2011 IEEE International Conference on Robotics and Automation*, pages 3475–3482. IEEE, 2011.

[15] D.J. Gonon, D. Paez-Granados, and A. Billard. Robots' motion planning in human crowds by acceleration obstacles. *IEEE Robotics and Automation Letters*, 7(4):11236–11243, 2022.

[16] C. W. Dodge. *Euclidean Geometry and Transformations.* Dover Publications, 2004.

[17] Asher Stern and Zvi Shiller. Robot navigation in dynamic environments using acceleration obstacles. `https://arxiv.org/abs/2504.13637`, 2025. arXiv:2504.13637 [cs.RO].