Agent Prioritization and Virtual Drag Minimization in Dynamical System Modulation For Obstacle Avoidance of Decentralized Swarms

Agent Prioritization and Virtual Drag Minimization in Dynamical System Modulation For Obstacle Avoidance of Decentralized Swarms

Louis-Nicolas Douce*[1], Alessandro Menichelli*[2], Lukas Huber[1], Anastasia Bolotnikova[2,3], Diego Paez-Granados[4], Auke Ijspeert[2], Aude Billard[1]

***Abstract*— Efficient and safe multi-agent swarm coordination in environments where humans operate, such as warehouses, assistive living rooms, or automated hospitals, is crucial for adopting automation. In this paper, we augment the obstacle avoidance algorithm based on dynamical system modulation for a swarm of heterogeneous holonomic mobile agents. A smooth prioritization is proposed to change the reactivity of the swarm towards the specific agents. Further, a soft decoupling of the initial agent's kinematics is used to design an independent rotation control to ensure the agent reaches the desired position and orientation simultaneously. This decoupling allowed the introduction of a novel heuristic, the *virtual drag*. It minimizes the disturbance influence an agent has when moving through its surrounding. Additionally, the safety module adapts the velocity commands from the dynamical system modulation to avoid colliding trajectories between agents. The evaluation was performed in simulated assisted living and hospital environments. The prioritization successfully increased the minimum distance relative to a moving agent. The safety module is observed to create collision-free dynamics where alternative methods fail. Additionally, the repulsive nature of the safety module augments the convergence rate, thus making the proposed method better applicable to dense real-world scenarios.**

I. Introduction

Recent years have seen growing interest in using mobile robots to assist humans [1]. One promising application is assistive furniture, where autonomous algorithms coordinate the furniture's poses to assist humans with limited mobility. This is useful for people in a wheelchair but also when the displacement becomes exhausting for someone, e.g., an elderly or ill person. Additionally, autonomous furniture can be useful in workspaces where the displacement of furniture takes up an important part of the available time and effort, e.g., in health centers or hospitals [2], [3]. One of the main challenges of coordinating mobile agents while sharing space with humans is to guarantee safety and at the same time ensure to reach the assigned goal. It is paramount to ensure that no collisions occur despite the uncertainty of the environment. Furthermore, the mobile agents must coordinate to avoid blocking each other's path. Otherwise, the system will either be too dangerous for the involved people or useless for application. This paper aims to provide an approach to multi-robot coordination using obstacle avoidance based on dynamic system modulation (DSM). We reduce the disturbance in the environment each agent creates by introducing priority values, a soft decoupling of linear and angular velocity, the virtual rate, and active repulsion. The method eliminates observed collisions in dense environments and increases the convergence rate. The framework enables autonomous agents in an assistive living room environment where the person does not have to adapt to the furniture, but it is the furniture that adapts to the person.

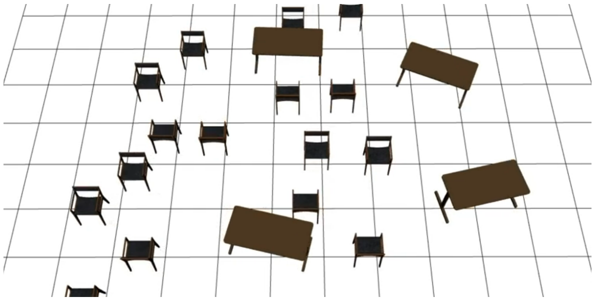


Fig. 1: Autonomous collision aware rearrangement of mobile chairs and tables in an assistive living environment.

We enable multiple mobile agents to coordinate by augmenting control algorithms that allow dynamic collision avoidance for single agents. This assures that agents can avoid the surroundings which we do not control, such as people in an assistive living environment. Furthermore, it allows full decentralization of decision-making and computing. For this work, all agents are assumed to be holonomic and can follow a desired velocity exactly. Furthermore, the state and shape of all obstacles in the environment are known to each agent.

A. Related Works

Navigation strategies can be classified into global and local navigation [4]. Among global navigation, Cell Decomposition (CD) divides the environment into cells that are classified as free or occupied [5]. A feasible trajectory is sampled across the free cells. The path quality increases with decreasing cell size, so does the computational cost, preventing the algorithm to be applied to dynamic, cluttered environments [6].

Rapidly-Exploring Random Trees (RRT) do sampling using a space-filling tree that explores the space to find a feasible path [7]. RRT has been extended to RRT* [8], which adapts the step length of the expanding tree for

* Authors contributed equally

This work was funded by the EU ERC grant SAHR, the EPFL Center for Intelligent Systems (CIS), and the EU H2020 project “Crowdbot” (779942).

[1]LASA Laboratory, [2]BioRob Laboratory, [3]RRL Laboratory, Swiss Federal School of Technology in Lausanne - EPFL, Switzerland

[4]SCAI Laboratory at SPZ, Swiss Federal School of Technology in Zurich - ETH Zurich, Switzerland

improved performance and convergence. Alternatively, Probability Roadmap (PRM) decreases the space explored by sampling points in the environment and connecting the ones which lie in the free space [9], resulting in a connectivity graph from the initial position to the goal. PRM has been extended to multi-robot scenarios by reducing the softening of the collision checks [10]. While RRT and RRT are probabilistic-complete (they will find a feasible path if it exists), the sampling is computationally expensive and cannot be used in dynamic environments.

Roadmaps (RM), in general, provide collision-free path segments. This can be done by incremental application of local obstacle avoidance methods [11], using Vornoi Diagrams [12] or visibility graph [13]. Finally, the method can be stitched together. However, they lack global convergence guarantees.

Artificial Potential Fields (APF) ensure local (reactive) obstacle avoidance by creating a potential field around obstacles that the agent follows [14]. APF has been extended to global path planning [15]. It has been used to track moving targets [16], as well asin multi-robot setups [17]. While APF is efficient and simple, it is prone to local minima in free space. (Global) Navigation Functions (NFs) have been introduced to ensure the convergence around concave obstacles [18]. However, NFs require environment-specific tuning and are often limited to static environments [19].

Dynamical system modulation (DSM) has been used for reactive obstacle avoidance in challenging environments [20]. DSM redirects initial dynamics to avoid collision with obstacles. DSM has been shown to converge around concave obstacles [21]. The method was extended to indoor environments [22], as well as sensor-based obstacle information [23]. As DSM controls only for zero-dimensional points, we had proposed an extension in our previous work which utilizes multiple control points to move rigid bodies [24]. However, the proposed method has a slow orientation convergence rate and has been observed to collide in dense multi-agent environments.

Velocities obstacles (VO) represent the set of velocities resulting in a collision with the environment [25]. The VO are used by Optimal Reciprocal Collision Avoidance (ORCA) to allow for multi-agent coordination to be collision-free [26]. However, the presented methods are designed for circular agents. ORCA has been extended to elliptic obstacles [27], and general shapes constructed from multiple circles [28]. Albeit, the computation of velocity and rotation is decoupled, and hence the approaches do not allow moving with a desired orientation. Simpler methods have been used to guide swarms of drones safely through forests [29]. Yet, as with many swarm avoidance algorithms, the approach is limited to circular collision robot shapes.

B. Contribution

This paper builds on work proposed in [24]. It focuses on the situational awareness of the swarm, i.e., decreasing the disturbance effect each agent has on its environment and enabling DSM to take into account situational factors (agent priority) in its computations. This is achieved with the following contributions:

- Prioritization of the agents (Sec. III)
- Soft decoupling of the agent's linear and angular velocity (Sec. IV)
- Minimization of the virtual drag (Sec. V)
- Safety module (Sec. VI)

The proposed contributions greatly increase the performance of swarms by decreasing collisions and increasing convergence, as is further discussed in Sec. VII-A.

II. Preliminaries

The identity matrix $\mathbb{I}$ is given in appropriate dimensions. While many concepts can be used in arbitrary dimensions $d \geq 2$, this work focuses on $d = 2$. The symbol $\times$ denotes the cross product in two dimensions, such that for two vectors, we have $\mathbf{a} \times \mathbf{b} = a_1 b_2 - a_2 b_1$. We use superscripts for variable names and subscripts for enumeration.

A. Dynamical Systems

Let $\mathbf{x} \in \mathbb{R}^d$ be the state of a mobile robot. It state evolves according to dynamical systems. When undisturbed, it can be described by a state-dependent function $\mathbf{f} : \mathbb{R}^d \rightarrow \mathbb{R}^d$:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) \quad \text{with} \quad \mathbf{f}(\mathbf{x}) = -(\mathbf{x} - \mathbf{x}^a) \tag{1}$$

where $\mathbf{x}^a \in \mathbb{R}^d$ is attractor of the system.

B. Obstacle Avoidance through Modulation

Obstacle avoidance based on Modulation (DSM) for point masses has been proposed in [21], [22]. Dynamic obstacles are avoided by evaluating DSM in the (averaged) moving frame:

$$\dot{\mathbf{x}} = \mathbf{M}(\mathbf{x}) \left(\mathbf{f}(\mathbf{x}) - \dot{\tilde{\mathbf{x}}}^{\text{tot}} \right) + \dot{\tilde{\mathbf{x}}}^{\text{tot}} \tag{2}$$

where $\dot{\tilde{\mathbf{x}}}^{\text{tot}} \in \mathbb{R}^d$ is the averaged velocity of the obstacles [20].

The modulation matrix is constructed as follows:

$$\mathbf{M}(\mathbf{x}) = \mathbf{E}(\mathbf{x}) \mathbf{D}(\mathbf{x}) \mathbf{E}(\mathbf{x})^{-1} \tag{3}$$

where $\mathbf{M}(\mathbf{x})$ is the local modulation matrix composed of the orthonormal basis matrix $\mathbf{E}(\cdot)$, defined as:

$$\mathbf{E}(\mathbf{x}) = [\mathbf{r}(\mathbf{x}) \; \mathbf{e}_1(\mathbf{x}) \; ... \; \mathbf{e}_{d-1}(\mathbf{x})] \tag{4}$$

where the tangent direction $\mathbf{e}_{(\cdot)} \in \mathbb{R}^d$ is perpendicular to the normal of the obstacle surface, and $\mathbf{r}(\mathbf{x}) = (\mathbf{x} - \mathbf{x}^r) / \|\mathbf{x} - \mathbf{x}^r\|$ is the reference direction, with respect to the reference point $\mathbf{x}^r \in \mathbb{R}^d$, a point placed inside the obstacle's boundaries. The diagonal matrix $\mathbf{D}(\mathbf{x})$ is composed of the eigenvalues λ^r and λ^e, as follows:

$$\begin{gathered} \mathbf{D}(\mathbf{x}) = \mathbf{diag}(\lambda^r(\mathbf{x}), \lambda^e(\mathbf{x}), ..., \lambda^e(\mathbf{x})) \quad \text{with} \\ \lambda^r(\mathbf{x}) = 1 - 1/\Gamma(\mathbf{x}), \quad \lambda^e(\mathbf{x}) = 1 + 1/\Gamma(\mathbf{x}) \end{gathered} \tag{5}$$

with the continuous distance function, which has a value of $\Gamma(\mathbf{x}) = 1$ on the boundary of the obstacle, and $\Gamma(\mathbf{x}) > 1$ outside the obstacle.

As $\lambda^r(\mathbf{x}) \leq 1$, the initial dynamics are decreased towards the obstacle, and with $\lambda^e(\mathbf{x}) \geq 1$, the dynamics are increased in tangent direction. It has been shown in [21] that this leads to convergence around (star-shaped) obstacles.

C. Orientation Control of Holonomic Agents

DSM has been used to control holonomic agents by introducing multiple control points [24], referred to as HDSM. By first evaluating the DSM at each of the control points $i \in \{1, .., N^c\}$, and then taking the weighted average with respect to the rigid body constraints (see Eq. 3), the linear velocity $\dot{\mathbf{x}}^c$ and angular velocity ω are obtained with respect to the center of mass $\mathbf{x}^c$.

$$\dot{\mathbf{x}}^c = \sum_{i=1}^{N^c} w_i \dot{\mathbf{x}}_i, \quad \omega = \sum_{i=1}^{N^c} w_i \left(\mathbf{x}^c - \mathbf{x}_i\right) \times \left(\dot{\mathbf{x}}_i - \dot{\mathbf{x}}^c\right) \qquad (6)$$

Each control point has its attractor and a circular margin around it, which is used to compute the distance $\Gamma(\mathbf{x})$, see Fig. 2. Dynamical weights are used to calculate the agent's kinematics to give higher priority to control points closer to collision with their surroundings.

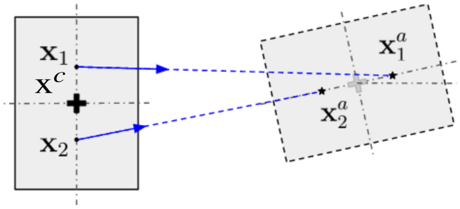


Fig. 2: Each control point of the furniture $\mathbf{x}_{(\cdot)}$ (black dots) has a velocity toward its attractor $\mathbf{x}^a_{(\cdot)}$ (black stars). The blue arrows are the initial dynamics $\mathbf{f}(\mathbf{x})$.

III. Prioritization of the Agent

An agent treating all obstacles in the surrounding equally might have undesired effects, e.g., an elderly person might get irritated when a robot gets close, while a chair would not be disturbed. Hence, we propose to use HDSM to adapt its trajectory based on the *priority* of the obstacles it encounters. In particular, we want the agents to prioritize trajectories staying further away from people to improve their safety.

The prioritization of an agent's safety distance (Eq. 7) can be directly derived from the formula of the eigenvalues proposed in Eq. 5 by introducing $\gamma^s \in \mathbb{R}_+$, the priority of the agent (self), and $\gamma^o \in \mathbb{R}_+$, the priority given to an obstacle (other). The priority values are used in the calculation of the eigenvalues as:

$$\lambda^r(\mathbf{x}) = 1 - \left(\frac{1}{\Gamma(\mathbf{x})}\right)^{\gamma^s/\gamma^o}, \quad \lambda^e(\mathbf{x}) = 1 + \left(\frac{1}{\Gamma(\mathbf{x})}\right)^{\gamma^s/\gamma^o} \qquad (7)$$

We can verify that if $\gamma^s \gg \gamma^o$ then $\mathbf{D}(\mathbf{x}) \approx \mathbb{I}$, as defined in Eq. 5. Hence, if the self-priority of the agent is far greater than the obstacle's priority, its trajectory won't be modulated concerning this obstacle. Moreover, the new Eq. 7 does not break the necessary condition for impenetrability of the DSM given by Eq. 8, since we have:

$$\text{if} \quad \Gamma(\mathbf{x}) = 1 \quad \Rightarrow \quad \lambda^r = 0, \ \lambda^e = 1 \qquad (8)$$

IV. Soft Linear/Angular Control Decoupling

The coupling of the linear and angular velocity (Eq.6) results in the angular velocity being active only when close to the attractor $\mathbf{x}^a$, leading to delayed (angular) convergence. More flexibility of the angular control can be obtained by first computing the initial kinematics at the center of the agent, $\mathbf{x}^c$ (Eq. 9) regarding the pose we want to reach (Fig. 3).

$$\dot{\hat{\mathbf{x}}}^c = -(\mathbf{x}^c - \mathbf{x}^a) \quad \text{and} \quad \omega = -(\phi - \phi^a) \qquad (9)$$

The initial velocity, $\dot{\hat{\mathbf{x}}}^c$, is modulated according to Eq. 2. The prior velocities $\mathbf{x}^c$ and ω are used to obtain the initial dynamics on the control points (Eq. 10).

$$\dot{\mathbf{x}}_i = \dot{\mathbf{x}}^c + \omega \times (\mathbf{x}_i - \mathbf{x}^c), \quad i \in \{1, .., N^c\} \qquad (10)$$

where N^c is number of agent control points. The transformation is shown in Fig. 3. Finally, the velocities at the control points are modulated by Eq. 3 and the agent's final linear and angular velocities are obtained by Eq. 6. This soft decoupling of kinematics brings more flexibility to the orientation control and will be further exploited in the next section.

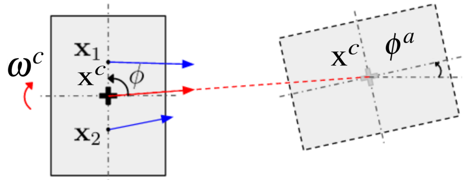


Fig. 3: The agent moves from its initial position (left) to the goal position (right). The prior linear and angular velocities (red arrows) are at the center of the agent $\mathbf{x}^c$ to reach the attractor position $\mathbf{x}^a$ and orientation ϕ^a. These can be transformed to get the control point velocities (blue arrows).

V. Minimization of The Virtual Drag

The drag of a moving object in a viscous fluid determines the disturbance of the surrounding particles. Similarly, the orientation of a swarm-agent moving through a clustered space affects the modulated velocity of its surrounding agents (Fig. 4). Hence, we refer to the obstacle's effect on its neighboring agents as *virtual drag* and desire to minimize it to decrease the disturbance on its environment.

A. Drag Factor

The virtual drag produced by an agent depends on its shape and displacement direction. Circular agents have a constant virtual drag and are invariant to rotation. Depending on its orientation, a rectangular agent with a high length-to-width ratio produces a higher or smaller virtual drag. We introduce the drag factor μ in Eq. 11 to quantify this, where l and L are the shortest and longest axis lengths, respectively.

$$\mu = \frac{L}{l}, \qquad \mu \in [1, \infty) \qquad (11)$$

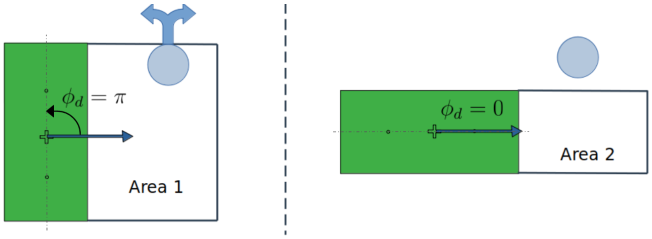


Fig. 4: The agent (green rectangle) needs to pass the blue circle (e.g., a person). If the agent moves perpendicular to its longest axis (left), the person must move out of the way. Whereas, if the agent has its longest axis parallel to the direction of the movement (right), the person does not get disturbed at all.

B. Drag Angle

The drag angle $\phi^d \in [0, \pi]$ is defined as the angle between the direction of the linear velocity and the axis that minimizes the virtual drag produced by the agent. It is 0 for displacement which is in the direction parallel to the minimal drag (and thus has a minimal virtual drag), and π when the agent moves perpendicular to the minimal drag direction (and thus has maximal virtual drag), see Fig. 4.

C. Convergence to Attractor Orientation

Minimizing the drag is important when moving, but as we approach the attractor, the agent needs to approach its desired attractor orientation ϕ^a. This balancing can be quantified as:

$$\begin{gathered}\tilde{\phi} = a_1 \phi^d + a_2 \phi^a \qquad \text{with} \\ a_1 = \frac{1}{2} \frac{d}{d+k} (1 + \tanh(\mu(d - \alpha)) \quad , \quad a_2 = 1 - a_1\end{gathered} \tag{12}$$

where d is the distance from the agent to the goal position, $\mu \in \mathbb{R}_+$ is the drag factor, and $\alpha \in \mathbb{R}_+$ is a parameter that defines the distance to the goal at which $a_1 \leq a_2$, i.e., the agent starts rotating towards the goal orientation. The tuning parameter $k \in \mathbb{R}_+$ is used in $d/(d+k)$ and ensures that a_1 converges to zero at the goal position $d = 0$. The resulting desired angle $\tilde{\phi}$ is used in the initial dynamic (Eq. 9) to obtain the initial angular velocity ω. This algorithm is referred to as drag-HDSM.

VI. Safety Module

Highly dense scenes may lead to situations that cause non-convergence that drag-HDSM cannot resolve. For avoidance in cluttered environments with multiple overlapping obstacles, we introduce a *safety module* to ensure to deviate any colliding trajectory. The safety module becomes active in the *critical* region around the obstacle, where it calculates the avoidance kinematics. This space is defined by the critical distance Γ^c and depends on the distance to the goal d:

$$\Gamma^c(d) = \begin{cases} \Gamma^{\max} & d > d^c \\ \Gamma^{\min} + \frac{d}{d^c}\left(\Gamma^{\max} - \Gamma^{\min}\right) & 0 \leq d \leq d^c \end{cases} \tag{13}$$

The lower bound is set as $\Gamma^{\min} > \Gamma(\mathbf{x}^a)$, i.e., the distance value at the attractor position. This ensures that the agent stays reactive at the goal position. Note that, if $d < d^c$, Γ^c shrinks linearly from $\Gamma^{\max}$ to $\Gamma^{\min}$.

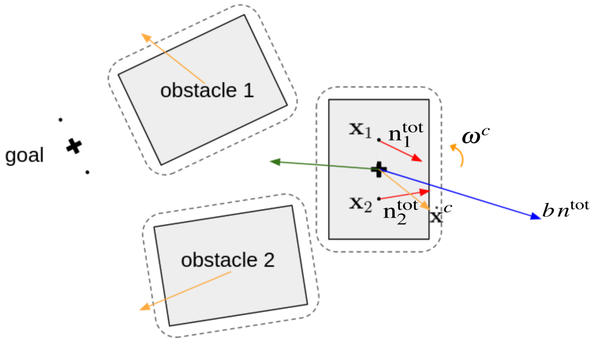


Fig. 5: Even though the planned trajectory (green arrow) is on a colliding path with the two obstacles, the safety module deviates the trajectory and avoids them safely (orange velocity). As all obstacles move away from each other, this leads to a *un-cluttering* of the space and has been observed to increase convergence.

The direction to avoid collisions, $\mathbf{n}_{\text{tot}}$, is computed as a weighted average of all the normal directions $\mathbf{n}_{\text{o}}$ of all surrounding obstacles $o \in \{1, .., N^{\text{obs}}\}$ for which we have $\Gamma^o(\mathbf{x}^c) < \Gamma^c$. The weighted normal is evaluated as:

$$\mathbf{n}^{\text{tot}} = \sum_{o=1}^{N^{\text{obs}}} w_o \mathbf{n}_{\text{o}} \quad \text{with} \quad w_o = \frac{1/\Gamma_{\text{o}}}{\sum_{o=1}^{N^{\text{obs}}} 1/\Gamma_{\text{o}}} \tag{14}$$

Elevating the basic velocity in Eq. 6, the safe linear velocity at the center $\dot{\mathbf{x}}^c$ at the center is defined as:

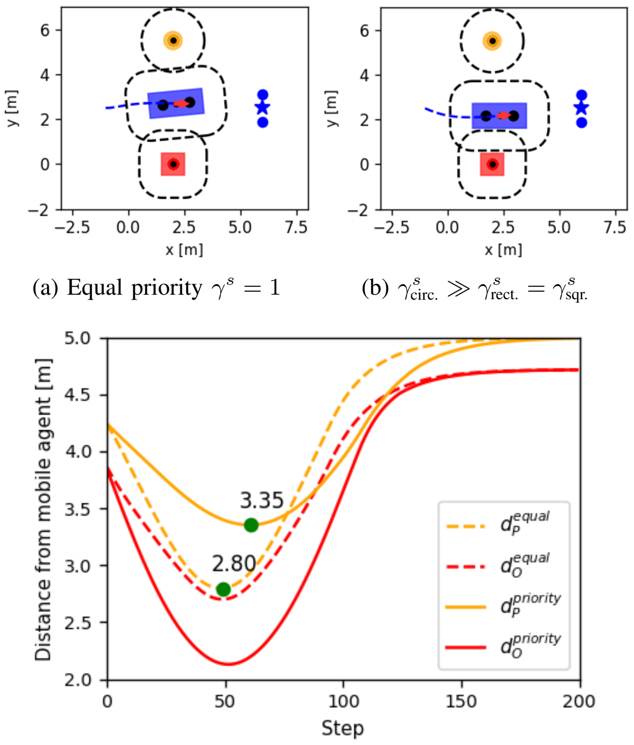
$$\dot{\mathbf{x}}^s = \sum_{i=1}^{N^{\text{c}}} w_i \dot{\mathbf{x}}_i + b\mathbf{n}^{\text{tot}} \tag{15}$$

where $b = (\Gamma^c - 1)\left(\min_o \gamma_o - 1\right)$ defines the strength of the safety response depending on the closest obstacle and the critical space.

Similarly, the angular velocity ω is updated using the parameter b and the correction term ω^c based on the avoidance direction $\mathbf{n}_i^{\text{tot}}$ for each of the control points $i \in \{1, .., N^c\}$ control and their position $\mathbf{x}_i$

$$\begin{aligned} \omega^s &= \sum_{i=1}^{N^{\text{c}}} w_i \left(\mathbf{x}^c - \mathbf{x}_i\right) \times \left(\dot{\mathbf{x}}_i - \dot{\mathbf{x}}^c\right) + b\omega^c \\ \text{with} \quad \omega^c &= \sum_{i=1}^{N^c} w_i \|\mathbf{x}_i \times \mathbf{n}_i^{\text{tot}}\| \end{aligned} \tag{16}$$

where $\mathbf{n}_k^{\text{tot}}$ as described earlier in this section is taking only into account the obstacles in the critical space of the i-th control point. When approaching the limit of the avoidance parameter, i.e., $\min_o \gamma^o \;\rightarrow 1 \quad \Rightarrow \quad b \rightarrow \infty$ as defined in Eq. 16, Consequently, the linear velocity $\mathbf{x}^c$ and angular velocity ω need to be stretched to have a magnitude which is within the desired bounds. Figure 5 shows the effect of the safety module. The safety module is combined with an emergency stop that sets $\dot{\mathbf{x}}^c = \mathbf{0}$ and $\omega = 0$ if any



(a) Equal priority $\gamma^s = 1$

(b) $\gamma^s_{\text{circ.}} \gg \gamma^s_{\text{rect.}} = \gamma^s_{\text{sqr.}}$

(c) Distance from the mobile agent to the person (yellow) and the static obstacle (red) for the case with equal priority (dashed) and higher prioritization of the person (solid).

Fig. 6: The mobile agent (blue) passes between the red rectangle (e.g., a chair) and the yellow circle (e.g., a person) when controlled by drag-HDSM. When the priority is equal (a), the agent passes in the middle between the others, see (c). Conversely, when the person has higher priority (b), the mobile agent's trajectory stays further away from the person.

control point i we have $\Gamma(\mathbf{x}_i) \leq \Gamma^{\text{stop}}$. This algorithm will be referred to as safe-HDSM.

VII. Results

We set he hyper-parameters as $\Gamma^{\text{stop}} = 1.1$, $\Gamma^{\text{min}} = 1.2$, $\Gamma^{\text{max}} = 2$, $d^c = 1$, $\alpha = 1.5$ and $k = 0.01$.

A. Metrics

1) Distance Traveled, $\overline{\mathcal{D}}$: Ideally, the trajectories are closest possible to the shortest distance to the goal. To assess the method performance in terms of how close the final trajectories are to the ideal case, we use the mean relative distance metric. The relative distance is the ratio of the total distance d_i traveled by an agent to the shortest distance D_i. $\overline{\mathcal{D}}$ is the mean relative distance made by all the N^a agents during a scenario.

$$\overline{\mathcal{D}} = \frac{1}{N^a} \sum_{i=1}^{N^a} \frac{d_i}{D_i} \tag{17}$$

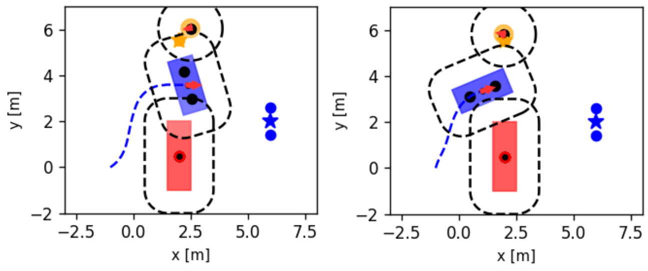
2) Virtual Collisions Rate, $\mathcal{C}$: A virtual collision is registered when an agent's control point breaches another agent's margin. Here, agents are stopped before a collision happens.

B. Agent Priority

We first analyze the influence of the priority value γ^s in a scenario where a mobile agent moves between two static agents (Fig. 6a). In the first setup, all agents have the same priority of $\gamma^s = 1$; in the second setup, we give the circular agent (interpreted as an elderly person) a higher priority, thus having $\gamma^s_{\text{circ.}} = 10^3$, $\gamma^s_{\text{rect.}} = \gamma^s_{\text{sqr.}} = 10^{-3}$. Qualitatively, we can see that the blue agent is making a larger avoidance trajectory around the yellow circle (human) when the latter has a higher priority. Furthermore, the minimum distance between the human (yellow circle) and the mobile agent (blue) decreases by 20% when the human has a higher priority.

C. Drag Minimization

The second comparison scenario includes a mobile agent, which passes between two passive agents (Fig. 7). In this case, the two passive agents can move out of the way if necessary, but they only do so if they are *disturbed*.



(a) HDSM (baseline)

(b) drag-HDSM

Fig. 7: The agent (blue) moves to its goal on the right. The red rectangle (mobile furniture) and the yellow circle (e.g., human) can move but desire to stay in their current position. With the HDSM (baseline), the person (yellow circle) has to move further out of the way than when using drag-HDSM.

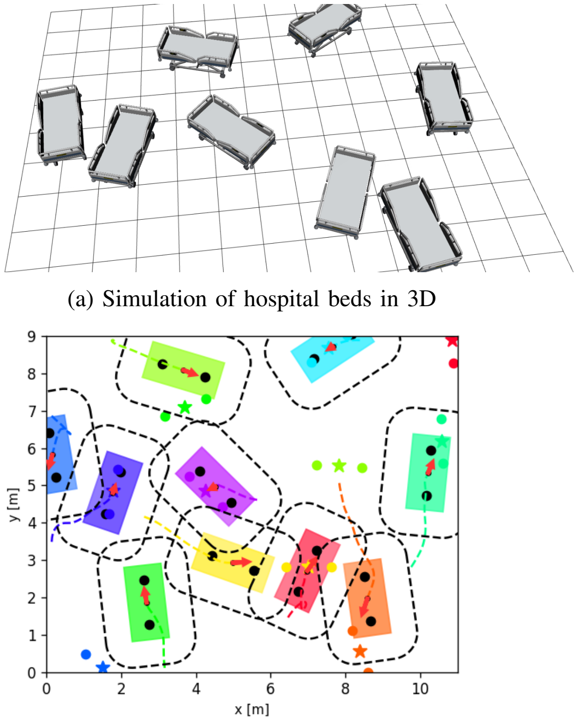
Using drag-minimization, the agent rotates before crossing the narrow passage. Hence, it has less disturbance to the person compared to the situation of the drag not being active. This can be interpreted as the furniture agent performing its trajectory with increased environmental awareness.

D. Quantitative Comparison

The proposed method and the baseline were compared quantitatively by generating 100 random scenarios for 3 to 10 agents. In each scenario, the agent's initial and goal positions are randomly chosen inside the spawn area without overlapping. The size of the spawn area of 11m × 9m, the agents are hospital beds which need to be rearranged of size 2m × 1m (see Fig. 8). No walls constrain the area, and the agents can move outside this area if needed. However, such behavior negatively affects the distance traveled score.[1]

The swarm using the algorithm with virtual drag minimization (drag-HDSM) shows a reduction in the mean relative distance traveled by all the agents compared to the

[1] Source code on `https://github.com/epfl-lasa/autonomous_furniture`



(a) Simulation of hospital beds in 3D

(b) Algorithmic representation of beds

Fig. 8: 100 random scenarios for different numbers of hospital beds were simulated (here, ten beds).

baseline (HDSM). This effect increases with the density of environments (Fig. 9 top plot). In scenarios with 10 agents, the median reduction is 10%. HDSM is expected to use a longer path because the agents disturb each other on the way.

The convergence rate decreases for all algorithms with an increasing number of agents (Fig. 9 middle plot). The virtual drag (drag-HDSM) improves convergence compared to the baseline (HDSM). Since minimizing the drag makes the obstacles align with their direction of motion, they present less surface to get stuck with each other. The safety module (safe-HDSM) shows the biggest improvement in convergence. When obstacles get close to each other, the safety module pushes them away so that they get untangled and converge toward their attractors. For safe-HDSM, even with 10 agents, almost 50% of the scenarios converge towards the attractors, compared to 0% for the other two algorithms.

The baseline method (HDSM) often makes agents collide when trying to reach the goal, and already with three agents, an average of around one-sixth of the simulations collide (Fig. 9 bottom plot). This increases drastically with the number of agents: with nine agents, almost 100% of the simulations result in collisions. The virtual drag (drag-HDSM) decreases the number of colliding scenarios. The effect is more pronounced with fewer agents, and we observe almost no collision for simulations with three obstacles. The greatest contribution comes from the safety module (safe-HDSM), which reduces the ratio of virtual collisions to 1% or less for all simulations.

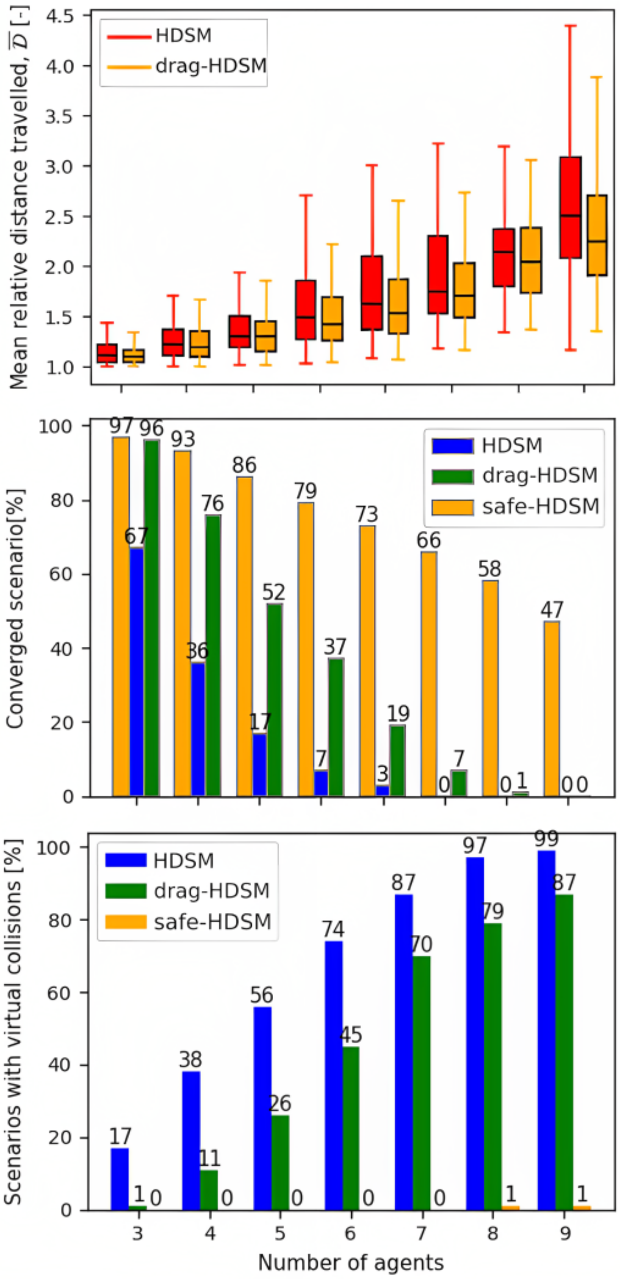


Fig. 9: The proposed algorithm performance comparison to the baseline (HDSM). Only no-collision scenarios were considered for the metrics in the top and middle plots.

VIII. Conclusion

In this paper, we have successfully extended the holonomic dynamical system modulation for obstacle avoidance (HDSM) by adding new features: agent prioritization, soft decoupling of the agent's kinematics, minimization of the virtual drag, and the safety module. The four contributions increase the environmental awareness of each swarm agent.

By augmenting the eigenvalues in the obstacle avoidance with the priority parameter, the safety around higher-priority agents is increased by keeping a larger distance from them.

The agent's linear and angular velocities were softly decoupled by defining their initial values independently. This enabled faster angular convergence. It further allowed the introduction of the minimization of the virtual drag. The latter makes the agent align its longest axis with its desired direction to the velocity, hence disturbing its environment less. This reduced the agents' mean relative displacement and collision probability, especially in dense environments.

The safety module has been shown to decrease the virtual collision and drastically increase the convergence of all agents toward their goal. While the safety module generally has the ability to push away close agents, there still exist cases where agents get blocked. This occurs when an agent is far away from its goal, i.e., $\Gamma^c = \Gamma^{\max}$, but the surrounding agents have reached their respective goals ($\Gamma^c = \Gamma^{\min}$). This blocks the first agent, while the surrounding agents do not react to it and do not move out of the way.

A. Future work

We plan to evaluate the agents' prioritization systematically. This should be evaluated within an assistive environment and together with potential users. It will be of interest to analyze, how the prioritization affects the disturbance of the agents towards each other, but also how it changes the time (and energy) spent to complete the paths.

Extending DSM by utilizing multiple control points results in a decrease of the convergence ratio. As for simulations with 10 obstacles, less than half the obstacles reach their desired goal. More flexible motion planning techniques could be used without compromising the low collision occurrence. We will explore a combination of the obstacle avoidance method and policy learning, such as provided by reinforcement learning. We believe exploiting synergies and strengths of learning algorithms with analytical methods can increase the performance of the system.

Finally, augmentation of the collision problem from 2D to 3D will allow more efficient exploitation of the available space as lower furniture can be under higher ones, and also allows arranging a chair underneath a table.

References

[1] C. Wang, A. S. Matveev, A. V. Savkin, R. Cloutz, and H. T. Nguyen, "A real-time obstacle avoidance strategy for safe autonomous navigation of intelligent hospital beds in dynamic uncertain environments," in *Australasian Conference on Robotics and Automation*, 2013.

[2] J. Brooks, C. Jenkins, D. Kocher, *et al.*, "Before coming home: The value of interaction studies with rehabilitation specialists using low-fidelity, physical prototypes prior to inserting novel assistive technologies into seniors' homes," *Smart Health*, vol. 22, 2021.

[3] A. Fallatah, B. Stoddard, M. Burnett, and H. Knight, "Towards user-centric robot furniture arrangement," in *IEEE International Conference on Robot & Human Interactive Communication*, 2021, pp. 1066–1073.

[4] M. Hoy, A. S. Matveev, and A. V. Savkin, "Algorithms for collision-free navigation of mobile robots in complex cluttered environments: A survey," *Robotica*, vol. 33, no. 3, pp. 463–497, 2015.

[5] E. U. Acar, H. Choset, A. A. Rizzi, P. N. Atkar, and D. Hull, "Morse decompositions for coverage tasks," *The International Journal of Robotics Research*, vol. 21, no. 4, pp. 331–344, 2002.

[6] E. Galceran and M. Carreras, "A survey on coverage path planning for robotics," *Robotics and Autonomous systems*, vol. 61, no. 12, pp. 1258–1276, 2013.

[7] S. M. LaValle, "Rapidly-exploring random trees: A new tool for path planning," *Technical Report. Computer Science Department, Iowa State University (TR 98–11)*, 1998.

[8] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The International Journal of Robotics Research*, vol. 30, no. 7, pp. 846–894, 2011.

[9] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.

[10] G. Sánchez and J.-C. Latombe, "A single-query bi-directional probabilistic roadmap planner with lazy collision checking," in *Robotics Research: The Tenth International Symposium*, 2003, pp. 403–417.

[11] J. F. Canny and M. C. Lin, "An opportunistic global path planner," *Algorithmica*, vol. 10, no. 2-4, pp. 102–120, 1993.

[12] H. Choset and J. Burdick, "Sensor-based exploration: The hierarchical generalized voronoi graph," *The International Journal of Robotics Research*, vol. 19, no. 2, pp. 96–125, 2000.

[13] L. Lulu and A. Elnagar, "A comparative study between visibility-based roadmap path planning algorithms," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005, pp. 3263–3268.

[14] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *The International Journal of Robotics Research*, vol. 5, no. 1, pp. 90–98, 1986.

[15] C. W. Warren, "Global path planning using artificial potential fields," in *IEEE International Conference on Robotics and Automation*, 1989, pp. 316–317.

[16] L. Huang, "Velocity planning for a mobile robot to track a moving target—a potential field approach," *Robotics and Autonomous Systems*, vol. 57, no. 1, pp. 55–63, 2009.

[17] B. Patle, A. Pandey, D. Parhi, A. Jagadeesh, *et al.*, "A review: On path planning strategies for navigation of mobile robot," *Defence Technology*, vol. 15, no. 4, pp. 582–606, 2019.

[18] E. Rimon and D. Koditschek, "Exact robot navigation using artificial potential functions," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 5, pp. 501–518, 1992.

[19] S. G. Loizou, "The navigation transformation," *IEEE Transactions on Robotics*, vol. 33, no. 6, pp. 1516–1523, 2017.

[20] S. M. Khansari-Zadeh and A. Billard, "A dynamical system approach to realtime obstacle avoidance," *Autonomous Robots*, vol. 32, no. 4, pp. 433–454, 2012.

[21] L. Huber, A. Billard, and J.-J. Slotine, "Avoidance of convex and concave obstacles with convergence ensured through contraction," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1462–1469, 2019.

[22] L. Huber, J.-J. Slotine, and A. Billard, "Avoiding dense and dynamic obstacles in enclosed spaces: Application to moving in crowds," *IEEE Transactions on Robotics*, vol. 38, no. 5, pp. 3113–3132, 2022.

[23] L. Huber, J.-J. Slotine, and A. Billard, "Fast obstacle avoidance based on real-time sensing," *IEEE Robotics and Automation Letters*, vol. 8, no. 3, pp. 1375–1382, 2022.

[24] F. M. Conzelmann, L. Huber, D. Paez-Granados, A. Bolotnikova, A. Ijspeert, and A. Billard, "A dynamical system approach to decentralized collision-free autonomous coordination of a mobile assistive furniture swarm," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2022, pp. 7259–7265.

[25] P. Fiorini and Z. Shiller, "Motion planning in dynamic environments using velocity obstacles," *The international journal of robotics research*, vol. 17, no. 7, pp. 760–772, 1998.

[26] 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*, IEEE, 2011, pp. 3475–3482.

[27] A. Best, S. Narang, and D. Manocha, "Real-time reciprocal collision avoidance with elliptical agents," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2016, pp. 298–305.

[28] Y. Ma, D. Manocha, and W. Wang, "Efficient reciprocal collision avoidance between heterogeneous agents using ctmat," in *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, 2018, pp. 1044–1052.

[29] X. Zhou, X. Wen, Z. Wang, *et al.*, "Swarm of micro flying robots in the wild," *Science Robotics*, vol. 7, no. 66, eabm5954, 2022.