IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 17 February 2024, accepted 9 March 2024, date of publication 12 March 2024, date of current version 26 April 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3376604

RESEARCH ARTICLE

Navigation Function for Multi-Agent Multi-Target Interception Missions

SHLOMI HACOHEN, SHRAGA SHOVAL, AND NIR SHVALB
Department of Mechanical Engineering, Ariel University, Ariel 40700, Israel

Corresponding author: Nir Shvalb (nirsh@ariel.ac.il)

This work involved human subjects or animals in its research. Approval of all ethical and experimental procedures and protocols was granted by the Ariel University Institutional Ethics Committee for Non-Clinical Human Experiments.

ABSTRACT Missions across a variety of disciplines require the interception of multiple targets. In defence scenarios, targets may pose a threat to sites, while in agriculture the targets may be invasive pests or fruit ready to harvest. This paper focuses on the cooperative control of a robot swarm for interception missions of multiple static and dynamic targets while avoiding collisions. We formulate two modifications of the classical Navigation-Function for a swarm interception mission which are suitable for deterministic and stochastic scenarios: the Swarm Navigation Function (S-NF) for the deterministic case, and the Swarm Probabilistic Navigation Function (S-PNF) for the stochastic case. Both functions provide a simultaneous solution for the problems of target assignment and motion-planning as opposed to the classical approaches that solve each problem independently. We demonstrate the effectiveness of these functions through extensive simulations and real-world experiments, comparing their performance with optimal solutions and human decision-making in similar scenarios. We show analytically that by following the Swarm-Navigation-Function gradient, the swarm will intercept all static targets while avoiding agent-agent and agent-obstacle collisions and similarly following the gradient of the Probabilistic-Navigation-Function will almost surely converge to a target in finite time, while the probability for agent-agent and agent-obstacles collisions is limited to a predefined value. The complexity of both schemes is linear with the number of targets and robots, and therefore it is scalable. Although not optimal, these solutions are simple and efficient, making them suitable for an extended set of real-time and real-life applications. We compare the resulting Swarm-Navigation-Function trajectories to that of a human in a catch game and an interception virtual game, the comparison indicates that as the trajectories are similar, human decision-making performs better. We conclude the paper with a set of simulated experiments and real-world experiments demonstrating the efficiency of the proposed scheme for dynamic targets.

INDEX TERMS Path planning, robotic swarm, S-PNF, uncertainty.

I. INTRODUCTION

For some robot missions, time and process constraints require the use of multiple cooperative agents working simultaneously as a swarm [1], [2], [3]. Individual members of a robotic swarm appear to be independent, but work together to create a highly complex performance, where the performance of the whole is greater than the sum of its parts. In this paper, we consider the interception of multiple targets by a robotic swarm. The swarm robotics approach offers distinct advantages over other methods, particularly in scalability and robustness. It has the potential to efficiently handle complex and expanding tasks, a feat challenging for individual robots. The redundancy in a swarm ensures mission success even with individual robot failures, providing resilience unmatched by single or few robots. Cost-effectiveness is also a key factor, as using multiple simpler robots often outweighs the investment in few sophisticated ones. These benefits collectively make swarm robotics a highly effective choice for diverse applications, underpinning

The associate editor coordinating the review of this manuscript and approving it for publication was Yilun Shang.

our research focus. Note that targets can be static and/or dynamic, operating in deterministic or stochastic scenarios. For example, in an exploration mission in an unknown environment, the unexplored zones are the (static) targets [4], [5]. In some scenarios, all targets require interception. Defense missions might require the elimination of all (dynamic) targets intruding into strategic zones; whereas in agricultural missions, interception might only require approaching the targets, for example scaring away water birds to prevent them raiding aquaculture tanks [6].

The traditional approach to these kinds of problems [7], is to (a) formulate, decompose, and allocate tasks among a group of (possibly heterogeneous) agents; (b) enable a predefined level of communication between the agents; and (c) ensure coherent behavior. However, the behavior of a cooperative team becomes harder to predict as the number of interactions between the team members increases combinatorically, and each interaction may result in an unexpected behavior. Furthermore, a failure in an interface between two agents can have a major effect on the entire swarm, and these interfaces between agents are often the most susceptible element in the mission. As a result, the performance of the team may be lower than the sum of its parts. Thus, the traditional *assignment problem* of agents to tasks is considered to be a fundamental combinatorial optimization problem [8]. This work, therefore, aims to integrate task allocation and motion planning into a unified framework, avoiding the traditional separation of these components, in order to enhance the overall performance and predictability of the swarm's behavior.

We consider a dynamic scenario where a team of autonomous mobile agents is required to intercept a set of targets (static and/or dynamic) while avoiding collisions with obstacles that are scattered in the environment and with each other. The locations of the agents, the targets, and the obstacles may be given in a deterministic manner or by a Probabilistic Distribution Function due to uncertainties (e.g., limited resolution of sensors, noisy environment). Beard et al. [9] suggest decomposing such a mission into three sub-missions: (i) assignment of all agents to targets; (ii) determining a feasible trajectory for each agent; (iii) determining asymptotically stable controllers for the motion of each agent such that all targets are intercepted within a bounded time period. The problem is known to be NP-hard as the number of possible assignments grows exponentially with the number of targets and agents.

For a single robot motion planning task, one can construct a special suited artificial potential field and follow its gradient. Providing that the potential field is chosen properly, this method guarantees task completion with computational simplicity. Such a function was first introduced as Navigation Function in [10]). A few attempts have been made to implement this method for swarm motion planning. For example, [11] use an artificial potential function to assign a control law for all swarm members in an aggregational assignment. The researchers set a repulsive potential for collision avoidance and an attractive potential that forces the agents to stay close to each other. In [12] the authors introduce a method to derive paths for all agents, realizing a set of distinguishable mutual paths (in the topological sense, i.e., a set of non-homotopic braids that represent such paths) by means of time-varying density functions. Note, however, that the potential field method inherently suffers from the possibility of local minima, and the extension of the Navigation Function for a single agent to swarm scenarios is not trivial, particularly in addressing uncertainties. Therefore, in this work we present a solution for swarms that not only (partially) guarantees the avoidance of local minima for all agents but also accommodates uncertainties, thereby enhancing the robustness and applicability of swarm motion planning.

Zhang et al. [13] introduce a method for coordinating the trajectories of multiple vehicles in real-world driving scenarios. They use Bayesian nonparametric learning to create essential building blocks, and then employ a sampling-based path planning algorithm to translate the change points of traffic patterns into actionable paths on the road. In [14], the authors present an algorithm that extends three types of single-agent deep reinforcement learning algorithms–policy gradient, temporal-difference error, and actor-critic–to cooperative multi-agent systems. To efficiently apply these algorithms to a large number of agents, the study combines them with a multi-agent version of curriculum learning.

Variants of the well-known *Navigation-Function* were presented by Dimarogonas, and Kyriakopoulos [15], [16]. The authors presented a scheme for guaranteed collision avoidance and global convergence of the swarm to the target configuration. They considered agents' dynamic models with holonomic, nonholonomic, and mixed constraints. In their work, each agent is provided with a designated static target - a solution that requires a preliminary sub-missions assignment stage (which as indicated above, we wish to avoid).

In this paper, we provide an extension of the well-established method for swarm motion planning using the navigation-function. More precisely, we propose an efficient motion planner for all swarm members, by extending both the classical Navigation-Function and the *Probabilistic-Navigation-Function* [17]. Our solution does not require decomposition of the problem into sub-missions, and it guarantees asymptotic convergence even for a very large number of agents and targets, as the complexity of the solution is linear to the number of agents plus the number of un-intercepted targets. Throughout the paper, we assume that:

1) The only information shared is the agents' and targets' locations.
2) Each agent separately performs their own motion planning.

A. CONTRIBUTION OF THIS PAPER

This paper offers several key contributions to the field of swarm robotics. Firstly, we introduce the Swarm-Navigation-Function and the Swarm-Probabilistic-Navigation-Function specifically designed for deterministic

and stochastic scenarios to enhance multi-agent target interception and collision avoidance. Secondly, we provide a thorough analytical validation of these navigation functions, demonstrating their effectiveness in dynamic environments. Thirdly, our research presents scalable control algorithms with linear complexity relative to the number of agents and targets, suitable for large-scale applications. Additionally, we validate our algorithms through extensive simulations and real-world experiments, highlighting their practical applicability. We also compare our swarm navigation strategies with human decision-making in similar tasks, offering insights into their efficiencies. Finally, the potential of these methods in diverse real-world applications is emphasized.

B. ORGANIZATION & ACRONYMS

The paper is organized as follows: Section I outlines the motivation and background of the study. Section II details the development of the Swarm Navigation Function (S-NF) and the Swarm Probabilistic Navigation Function (S-PNF) which are swarm-suited extensions of the deterministic navigation function (NF) and the Probabilistic one (PNF), respectively. In Section II we introduce our solution for multi-agent multi-target missions, extending the classical Navigation Function for swarm applications. This section is divided into subsections discussing deterministic cases, stochastic cases, and scenarios involving dynamic targets. Section III presents the findings from various sets of simulations and real-world experiments. It includes planar and spatial simulated experiments, real-world robotic swarm experiments, and experiments involving human decision-making, all designed for validating the proposed schemes. The final section interprets the results, explores their implications, and discusses the potential applications of the study. We conclude with a summary of key findings and contributions to the field of swarm robotics.

II. SWARM INTERCEPTION MISSIONS

The main contribution of this article is the introduction of a solution for a multi-agent multi-target mission. Our solution is closely related to the classical NF, which solves the problem of intercepting a single target by a single agent. We formulate a Swarm Navigation Function (S-NF) and prove its convergence for the deterministic scenario where all information is assumed to be known. In Section II-B, we show how a probabilistic version of the S-NF can be extended for the more realistic stochastic scenarios [6], where the locations of the agents and the targets are given by their probability density functions. Throughout, the geometries are assumed to be perfectly known.

We mark a bounded workspace by $\mathcal{W} \subset \mathbb{R}^d$ in which a set of agents $\mathcal{A}$ is required to intercept a set of mobile targets $\mathcal{T}$ while avoiding a set of obstacles $\mathcal{O}$. More precisely, the $i \in \mathcal{A}$ agent, located at $x_i^{(a)} \in \mathcal{W}$, should avoid all other agents as well as a set of mobile obstacles which together constitute the set of *entities* $\mathcal{O}_i$. We further denote by $x_m^{(\tau)} \in \mathcal{W}$ and by $x_j^{(o)} \in \mathcal{W}$ the locations of the $m \in \mathcal{T}$ target and the $j \in \mathcal{O}$ obstacle, respectively. The free portion of workspace $\mathcal{W}_i$ for the i^{th} agent is defined as the subset of $\mathcal{W}_{free}$ which is the interior of the space that is free from entities. The proposed scheme considers spherical agents. One can include other geometries by following Rimon and Koditschek's rational [18] for the deterministic case and [17] for the stochastic case, nevertheless this is out of the scope of the current paper.

The Deterministic Case: in this scenario all the entities' locations are assumed to be perfectly known. So, the problem to be solved can be stated as follows:

> *Given a closed environment with a set of agents, obstacles and targets of known locations, generate a path for each agent such that after finite time all targets are intercepted, while collisions with other entities are avoided.*

The Stochastic Case: in this scenario the locations refer to the expected positions of the centers of the entities, and are considered to be random variables. We assume that only the set of independent probabilistic distribution functions of the locations are available for all entities. Also, we assume that the random variables are normally distributed, and we denote such a probabilistic distribution function by $p(x)$ for $x \in \mathcal{W}$. The problem to be solved is follows:

> *Given a closed environment with a set of probabilistic distribution functions that characterize the uncertainties of the locations of the agents, obstacles, and the targets, generate a path for each agent that maintains the probability for collision with every other entity below a given allowable probability Δ, and increases the probability for intercepting a non-intercepted target, such that after finite time, all targets are intercepted.*

We further assume that:

1) All random variables are normally distributed and have a scalar-matrix as their covariance.
2) All agents are aware of the location probabilistic distribution functions of the other agents, obstacles, and targets.
3) The entities are far enough apart. In other word, for all pairs of entities j_1, j_2, there is no point $x \in \mathcal{W}$ where both probabilities $p_{j_1}(x)$ and $p_{j_2}(x)$ are greater than Δ.

The third assumption is required to avoid non-convex obstacles, which make sense since the agents are expected to avoid undesirable collisions.

A. THE DETERMINISTIC CASE

The classical NF is formulated in [19] for static deterministic scenarios with a single agent and a single target as:

$$\varphi(x) = \left(\frac{\gamma_d^K(x)}{\gamma_d^K(x) + \beta(x)} \right)^{1/K} \tag{1}$$

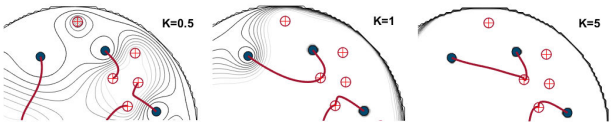


FIGURE 1. The effect of K on the path curvature and length. Increasing K decreases the path length. The agents are marked by dark discs, the targets are the red ⊕, the red lines are the planned paths.

where $\gamma_d(x) = \|x - x^{(t)}\|_2^2$ is the *target function* which calculates the distance to the target location $x^{(t)}$. The *obstacles function* is defined as:

$$\beta(x) = \left(\rho_0^2 - \|x\|_2^2\right) \prod_{j\in\mathcal{O}} \|x - x_j^{(o)}\|_2^2 - \rho_j^2 \tag{2}$$

where ρ_j and $x_j^{(o)}$ are the radius and the location of the j^{th} obstacle, respectively. Note that the environment is defined as the zero obstacle with a radius ρ_0 located at the origin, ensuring that agents do not cross the environment's boundary.

A function is said to be NF if it satisfies the following conditions: (a) it is polar; (b) it is admissible; (c) it is a Morse function; and (d) it is smooth. These conditions are shown to guarantee convergence from any initial condition to the target. Because the NF is Morse in nature, it should be of full rank at any point in $\mathcal{W}_{free}$ (i.e. its Hessian should be regular at all its critical points). Note however, that the NF nominator is γ_d^K where a proper selection of K guarantees that φ is indeed a NF. Typically, K is chosen to be at least the number of obstacles (note that this is not the case in all our experiments where we set $K = 1$). This makes the NF degenerate at the target point (a simple equivalent example would be $y = x^4$). To overcome this, the composition of the above with the function $x \mapsto x^{1/K}$ is applied (which intuitively and effectively eliminates the power K of the nominator. For the mathematical line of thinking see Appendix V-A).

To accommodate the classical NF for the swarm case, we introduce the S-NF φ_i for the i^{th} agent, which incorporates a multi-target function γ and an obstacles function β which accounts for all entities $\mathcal{O}_i$ (i.e. all obstacles and other agents). The S-NF deals with degeneracy in a different way from that for the classical NF. It has a similar structure to the traditional NF except that the $1/K$ power is eliminated, and the targets function is defined as:

$$\gamma(x) = \prod_{m\in\mathcal{T}} \gamma_m^{K_m}(x) = \prod_{m\in\mathcal{T}} \|x - x_m^{(t)}\|_2^{2K_m} \tag{3}$$

where $x_m^{(t)}$ and K_m are the location of the m^{th} target and a parameter related to this target respectively. Note that γ may be computed once, for all agents. The structure of the obstacles function, denoted by β, retains its classical NF configuration, but now encompasses all obstacles and other agents except the i^{th} agent. Consequently, an agent following $-\nabla\varphi_i(x)$ experiences an increase in distance from all entities as β diminishes with distance.

Definition 1: The swarm navigation function for the deterministic case, defined for the i^{th} agent is:

$$\varphi_i(x) = \frac{\gamma(x)}{\gamma(x) + \beta_i(x)} \tag{4}$$

where $\gamma(x)$ is defined in Eq. 3, and $\beta_i(x)$ is:

$$\beta_i(x) = \left(\rho_0^2 - \|x\|_2^2\right) \prod_{j\in\mathcal{O}_i} \|x - x_j^{(o)}\|_2^2 - \rho_j^2 \tag{5}$$

Note that since the set of targets is the same for all agents, for simplicity of notations, we do not provide subscript i in our notations for γ function, while we do for β_i.

Following the gradient of Eq. 4 above for a scenario with fixed targets where the locations of all agents are known for every timestamp $x(t)$, one can state (the proof is provided in Appendix V-A):

Theorem 1: For the deterministic case where all targets are static, there is an ordered set $(K_1, K_2, \ldots, K_{\|\mathcal{T}\|})$ such that traversing the gradient of $\varphi_i(x)$ (Eq. 4) in every time step, with a step size $a < max\left\{\left[\sup_{x\in\mathbb{R}^d} \|\nabla^2\varphi_i(x)\|_2\right]^{-1}\right\}$ almost surely will converge to a target in $\mathcal{T}$, while avoiding all entities in $\mathcal{O}_i$.

It should be noted that for the purpose of proving Theorem 1, and Theorem 2 we used the values of K_m as a mathematical tool, nevertheless in all experiments their value was set to 1, so for all practical uses this value will do.

It also should be noted that an agent that initially begins to track a specific target can alter its choice and switch to another target. However, this shift occurs only if another agent, which is not currently pursuing a target, is closer to the initial target. Essentially, the target remains engaged by the first agent unless a closer, unassigned agent is available.

B. THE STOCHASTIC CASE

To solve the stochastic case we follow the same rationales in the previous subsection - we extend the *Probabilistic Navigation Function* (PNF) developed in [17] to the swarm scenario.

In [17] the authors introduce the PNF for the stochastic case which is defined similarly to the NF (i.e. as given in Eq. 1 and Eq. 3) with the exception that the obstacles' function β is a function of the probability for collision rather than a geometric function used for the NF. In this study, the authors prove that the PNF is a navigation function. Following the gradient of the PNF will reach the target with probability 1 while bounding the probability for a collision to a pre-defined probability Δ. To accommodate the swarm case, we introduce the S-PNF (which is also marked by φ_i for the i^{th} agent):

Definition 2: The swarm probability navigation function (S-PNF) for the stochastic case is given in Eq. 4 with γ as defined in Eq. 3 and:

$$\beta(x) = \prod_{j\in\mathcal{O}} \left(\Delta - p_j(x)\right) \tag{6}$$

where $p_j(x)$ is the probability for a collision with the j^{th} entity at position x, and Δ is a predefined bound on the probability for collision. (Formal expressions for the probabilities for collision $p_j(x)$ are given in [17]).

Following the gradient of the S-PNF for a scene with fixed targets, one can state (the proof is provided in Appendix V-A):

Theorem 2: For the case where all targets are static, there is an ordered set $(K_1, K_2, \ldots, K_{\|\mathcal{T}\|})$ such that traversing the spacetime gradient descent of the S-PNF given in Def. 2 with step size $a < \left[\sup_{x\in\mathbb{R}^d} \|\nabla^2\varphi_i(x,t)\|_2\right]^{-1}$ almost surely converges to a target in $\mathcal{T}$, while avoiding all entities in $\mathcal{O}_i$ up to a probability below the given allowable probability for collision (Δ).

C. THE DYNAMIC TARGETS CASE

The solution provided for the case where all targets are static is very similar by nature to the solution which will emerge for the case where the targets are moving slowly. For more details see Palis and Smale 1970 discussion on structurally stable flows [20]. It is important to clarify that an agent successfully intercepting a target in a static scenario may not necessarily achieve the same in a slow dynamic scenario. That is, the overall convergence properties are maintained. In other words, when considering the swarm and the targets as a unified system, the nature of convergence points and their associated basins remain consistent. Additionally, the vector fields within the configuration spaces of both static and dynamic systems can be viewed as continuous deformations of one another.

In the proposed methodology, each agent independently recalculates its own S-NF (S-PNF) at every time step. This autonomous recalibration enables the agents to effectively intercept the targets while ensuring their safety, particularly as the targets continue to advance. This approach consistently ensures safety of the agents as well as target interception (when an agent is in close enough proximity to the target). It is pertinent to mention, however, that while this method does not serve as a definitive proof of interception in dynamic scenarios, our analysis shows that in every case studied all the targets were successfully intercepted by the agents.

As discussed below, we used simulations to test our algorithm for the case of moving targets. We applied two scenarios:

1) The targets move in a linear fashion and rebound off the workspace boundary at random angles.
2) The targets move according to a repulsive potential field, where the agents are the source of repulsion.

We noticed that in all cases the gradient did not vanish. We also noticed that the algorithm easily converged for any given speed of the targets as long as they did not exceed the speed of the agents.

In all these experiments K was set to a unit. However, note that for the case of fast-moving targets, one can set K to be such that the length of the trajectories is shorter, thereby improving convergence success (see Figure 1).

III. RESULTS

Two large sets of simulations were performed. Firstly, a set of real-world robotic swarm experiments, and secondly, two types of experiments involving human decision-making to investigate the performance of our interception solution.

A. METHODOLOGY

In order to validate the proposed scheme, both for the deterministic and stochastic cases, we conducted several experimental sets. The experiments consisted of a simulation set using Matlab framework, and real-world scenario sets. In the first simulation set, a swarm of simulated agents is required to intercept 100 simulated targets (static and dynamic) in 2D scenarios. The number of agents in the swarm varies from 10 to 100 and all operational parameters (interception times and trajectory lengths of all agents) are recorded. In the next simulation set, a swarm of 3D simulated agents is required to intercept a set of 100 simulated static and dynamic targets, where the number of agents in the swarm varies again from 10 to 100. As with the 2D simulation scenarios, all operational parameters are continuously recorded. The aim of the simulation sets is to verify the validity of the proposed scheme, and also to compare its real-time performance with the optimal off-line solution. In the first set of real-world scenarios, we use a swarm of 2-6 drones that are required to intercept a set of 1-6 static and dynamic targets, represented by small colored balls that are located manually in the experimental arena. In this set of experiments, several static objects are also located in the arena. The positions of all agents, targets, and obstacles are determined in real-time using a set of 18 pre-positioned "OptiTrack" Flax13 cameras, providing accurate 3D estimates of the entities' positions. In the second set of real-world scenarios, we compared the performance of the proposed scheme to the performance of a "human swarm". In the first experiment in this set, a group of children paly the "Tag" game. In this game, one group of 5 children chases another group of 5 children while the other group is trying to escape. A camera mounted at the top of the experimental arena records the trajectories of the children from the two groups participating in the game. The trajectories of the escaping group are then used as an input to the motion algorithm scheme, and the trajectories of the chasing group are compared with the trajectories generated by the algorithm. In the final set of real-world scenarios, a human user is playing a simple computer game where the aim is to intercept 10 targets by a swarm of 4 agents (3 computerized agents on one human agent). The three computerized agents move according to the proposed scheme, while the actual trajectory of the human agent is recorded and compared with the trajectory generated by the scheme. The aim of the experimental setups that involve human agents is to compare the performance of the human agents, which is close to optimal, with the performance of the proposed scheme.

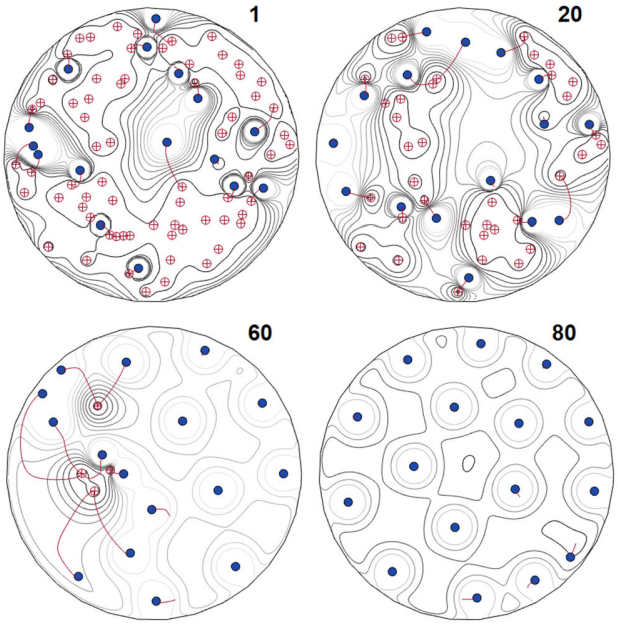


FIGURE 2. Four timestamps of the interception process of 70 dynamic targets by 16 agents applying the S-PNF. Agents are represented as dark discs, ⊕ marking the targets (no obstacles were introduced). The gray contour lines represent the values of S-PNF of the agent, and the resulting trajectories are marked depicted as red lines. Targets are canceled out once intercepted. At time stamp 60 two agents are heading to the same target. This do not pose any problem, since the agent with the longer path will at some point will change its goal.

B. PLANAR SIMULATED EXPERIMENTS

The motion planning schemes introduced in this paper hold for any dimension of the ambient space (e.g. $d = 2$ for a swarm of ground robots and $d = 3$ for a swarm of drones). To examine them, planar and spatial experiments were devised. We also examined the implementation of our motion planner for dynamic targets.

The experimental setup consists of deterministic as well stochastic scenarios. The workspace is set to be a planner circular workspace of radius 1 unit length. The agents' radii are set to be 0.02. All constant K_i's are set to be equal to the total number of agents and obstacles divided by the number of targets. Here we follow the 'rule of thumb' provided by [19] for the classical NF and by [21] for the PNF. For uncertain scenarios, we chose $\Delta = 0.9$ and the standard deviation for the estimation error is simulated as 0.01 length unit. The maximal agent velocity in both the deterministic and the stochastic cases is set to be 0.02 length unit/time unit.

(i) Two sets of planar experiments were conducted under the same conditions with agents intercepting 100 *dynamic targets*. The targets' maximal velocities were chosen to be standard normal random variables with a mean of 0.5 length unit or time unit.

(ii) Next, two sets of the planar experiments tested the S-NF and S-PNF schemes, where agents are required to intercept 30 *idle targets* that are fixed to their initial random locations; These experiments were repeated with a varying number of agents from 2 to 16.

Figure 2 depicts a snapshot set of the stochastic scenario. At the initial timestamps (and up to a timestamp of about 35) the number of targets is larger than the number of agents. During this period, each agent is directed towards a different target. Once all targets have been intercepted and the mission is completed, the agents spread out evenly across the entire playground. Repeating the experiments for the stochastic case, it was found that the agents maintained safe distances from each other and from the environment's boundary, as expected.

It should be noted that an additional set of experiments were conducted. These demonstrated that our algorithm is well suited also for the case where the agents share their sensory data and locations only to the agents in their vicinity. Nevertheless, this is out of the scope of this paper.

Figures 3 compare the average of the agents' path lengths required for convergence (i.e. when all targets were intercepted) for the set of experiments for the deterministic (solid red line) and the stochastic scenarios (dashed black line). Each point on the graphs represents the average of 100 Monte-Carlo runs.

To evaluate the efficiency of our algorithm, we compare the convergence graphs in the case of a deterministic static scenario with that of the optimal simulated-annealing (SA) based motion planner (dashed gray). The optimal "traveling salesman" graphs and the resulting optimal paths' overall length was calculated using a SA procedure, minimizing the overall length of these graphs. The graphs demonstrate a tight agreement of the convergence rate to the optimal one from $\approx$ 10 agents on, for the deterministic scenario with static targets.

For the SA based algorithm for a swarm with $N = 16$ agents, the average running time is $\mu = 9.88$ seconds with a standard deviation of $\sigma = 3.72$ seconds, while for $N = 50$ the running time is $\mu = 14.81, \sigma = 9.87$ seconds (on a Core i7 Processor). The S-NF and the P-NF are based on analytic function calculations, so their resulting running times are considerably shorter and more consistent. For example, for $N = 16$ agents the running time is $\mu = 0.04, \sigma = 0.002$ seconds while for $N = 50$ it is $\mu = 0.25, \sigma = 0.003$ seconds.

C. SPATIAL SIMULATED EXPERIMENTS

For the spatial case, the world is set to be a cylindrical playground. The agents' radii, the playground radius and K_i's are set to the same values as for the planar experiments. The maximal agent velocities set to be 0.02 length-unit/time-unit.

We conducted a set of simulated experiments for testing the S-NF scheme, where agents were required to intercept 30 idle targets fixed to their initial random locations; These experiments were repeated with varying number of agents from 2 to 22. Figure 4.(a) depicts a snapshot of the S-NF experiment, with black drones and red drones representing agents and targets respectively.

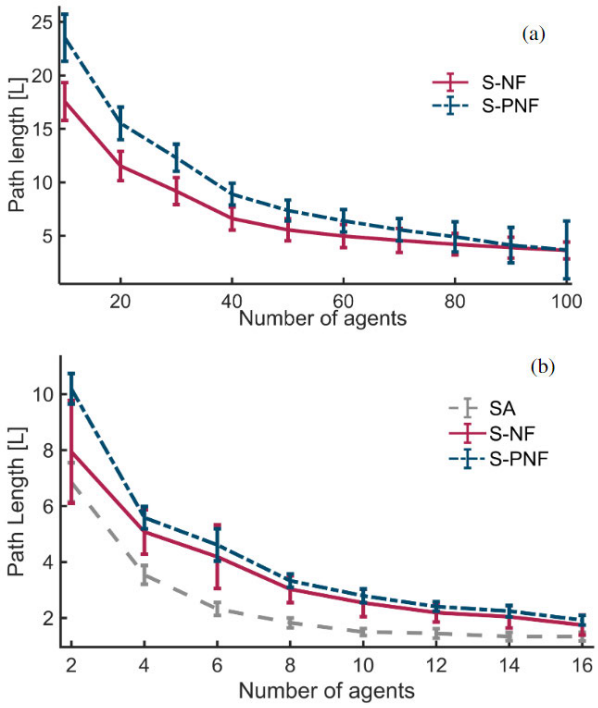


FIGURE 3. Convergence for various numbers of agents of radius 0.02 in a disc workspace of radius 1 unit length. (a) The mean path length per agent of the S-NF (solid red) and the S-PNF (dashed black) intercepting 100 dynamic targets. Each experiment was repeated 100 times, and the uncertainty is $\sigma = 0.01$; (b) Performances of the S-NF, the S-PNF compared to the optimal path computed by the SA (dashed gray) for various number of agents intercepting 30 static targets. Each experiment was repeated 100 times. The performances are the longest path between the swarm's members.

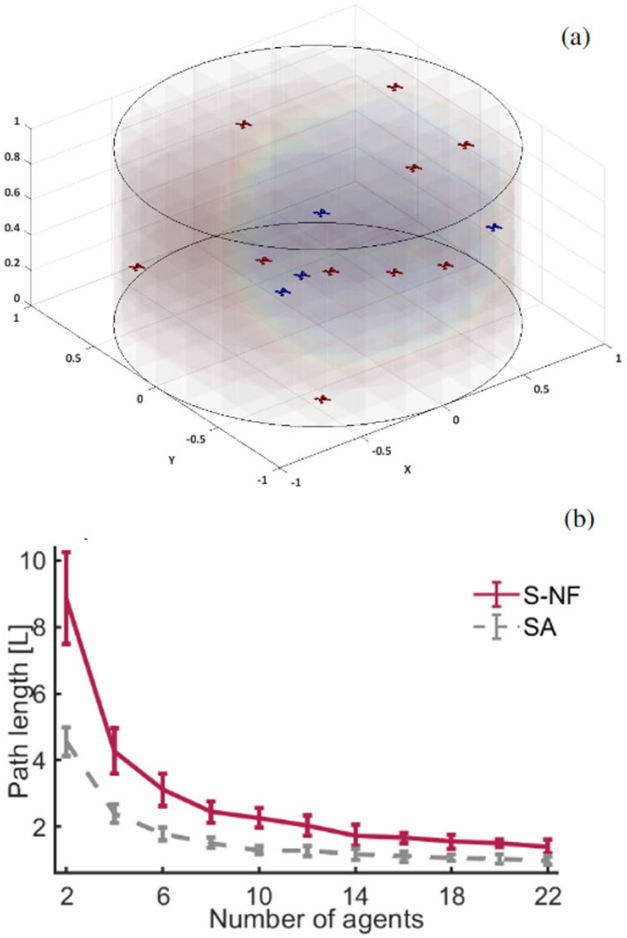


FIGURE 4. (a) The spatial experiment set; (b) Performances of the S-NF (solid red) and the simulated annealing (dashed gray) for a varying number of idle agents intercepting 30 idle targets. The graph depicts an average of the swarm's total path length calculated over 100 Monte-Carlo runs.

To evaluate the efficiency of our algorithm, we compare the convergence graph with an optimal convergence line (gray) calculated in the same manner as for the planar case. Figure 4.(b) presents the path-length required for convergence (i.e. when all targets were intercepted) for the set of spatial experiments for the deterministic scenario compared with the SA based algorithm optimal convergence time. Each point in the graphs represents the average of 100 Monte-Carlo runs.

D. REAL-WORLD ROBOTIC SWARM EXPERIMENTS

We examined our algorithmic scheme by using an experimental setup of a real-world, multi-agent multi-target interception mission scenario. A swarm of $3-6$ small unmanned aerial vehicles (UAVs) (DJI Tello edu) was free to travel in a cylindrical playground of radius 1.8m and length of 2m and "intercept" several targets.

The UAVs locations were measured by 18 pre-positioned "OptiTrack" Flax13 cameras. As this system provides accurate positions with a small localization error ($< 2mm$), we implemented the S-NF rather than the S-PNF probabilistic version. To ease the implementation stage, the entire algorithm computation was independently calculated for each agent (using Matlab) and the control signals were then transmitted via Wi-Fi to each agent.

In these experiments we also incorporated four static cylindrical obstacles ($r = 7.5cm, h = 2m$) which were randomly located in the workspace see Figure 6.(a) and (b). The targets (red discs) were placed randomly by the experimenters. Figure 5 depicts a sequence of frames of the experiment (A video demonstration is available in [22]).

Figure 7 shows the average time-to-interception for a varying number of agents intercepting the four targets while avoiding collision with the static obstacles. As shown, the average interception time is reduced linearly with the increase in the number of agents.

E. TWO SETS OF HUMAN BEHAVIOR EXPERIMENTS

The schemes introduced here are not optimal, moreover, optimality is very hard to assess for dynamic scenarios. To have a sense of the optimality of our solution, an experimental setup was designed to provide a comparison with human decisions in similar scenarios. Two experimental setups were

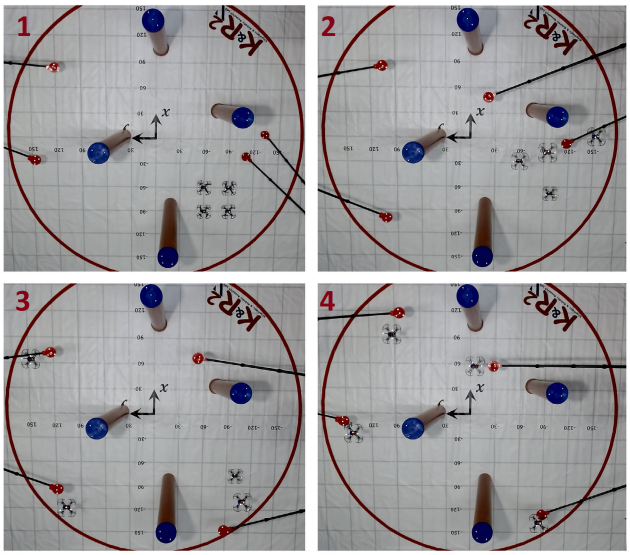


FIGURE 5. A sequence of frames for a four agents experiment with four targets and four static obstacles. Random initial conditions (upper left) and complete interception (lower right).

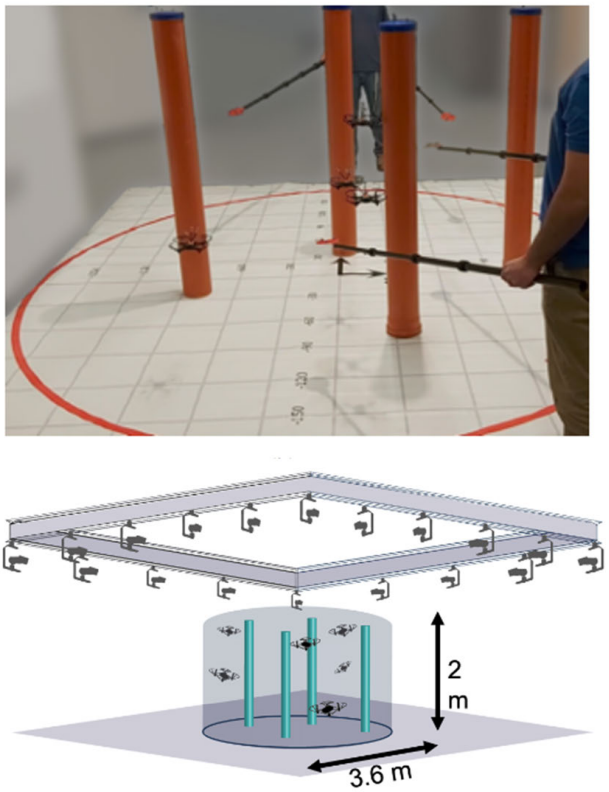


FIGURE 6. An illustration of the real-world experiment setup and typical view of an experiment with four agents and four static obstacles.

examined: (i) a real-world game and (ii) a video-game setup. All participants' custodians provided an informed consent after the nature and possible consequences of the study were explained. In addition, the experiment was confirmed from the university ethics committee.

(i) A 2.5 meter radius scene was defined on a planar concrete floor (see Figure 8.(a)). Ten young participants

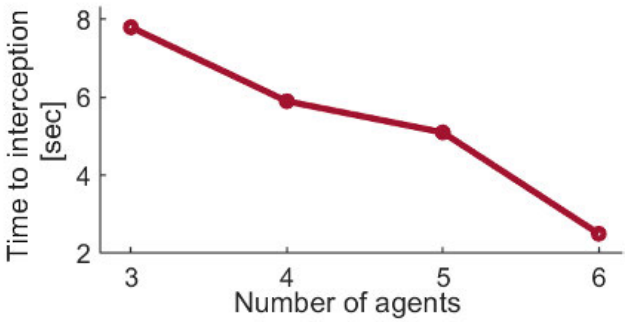


FIGURE 7. The average time to interception of 3 – 6 agents in real-world interception experiments.

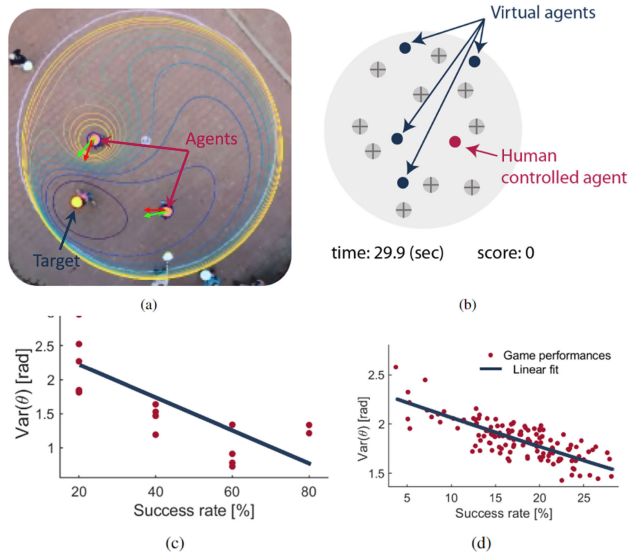


FIGURE 8. (a) A snapshot of the real-world, human decision making experimental setup. Red arrows indicate the participant's current direction while green arrows indicate the algorithm's preferred direction; (b) The setup of the human decision-making computer simulated game. The human participant's agent is marked as a red ball, other agents are the other four black balls, and the targets are indicated by colorful bubbles with plus signs. (c) Success rates of the tag players vs. the angular differences between the algorithm decisions and the player decisions. (d) Success rates of a human player vs. the angle differences between the algorithm decisions and the human decisions in the computer simulated game. The red markers are averaged results over 128 participants, the solid line is their resulting linear fit.

(all 12 years old) were randomly divided into two groups, representing the agents and the targets, playing a variation of the game ''tag''. The agent participants were guided to intercept the target participants. All participants were directed to maintain a constant speed throughout the game. A hovering camera was placed above the scene and all movements were recorded. Each game lasted until all ''targets'' were intercepted, which took 2 – 3 minutes. This procedure was repeated 20 times to eliminate bias.

To measure the agreement between human decisions, which we believe is somewhat optimal under dynamic conditions, we analyzed the agreement of human participants' decisions with our algorithm, and the corresponding

success rate (i.e. the number of interceptions divided by the number of targets). We used the variance $var(\theta)$ of the angular difference between the human participant trajectories and the algorithm's chosen trajectories as a measure of agreement with our algorithm. The results are provided in Figure 8.(c) (a video demonstration is available as movie in [23]). Clearly, participants whose trajectories corresponded with our algorithm were more successful.

(ii) A simulated game was designed to examine the algorithm agreement with human behavior. The game was designed as having a disc arena with the same proportions as with the real-world experiment. A single, human-controlled agent and 4 additional computerized agents were given the task of intercepting 10 moving targets. The targets moved in a linear manner in $\mathcal{W}_{free}$ and randomly changed direction after a collision with the boundaries. When a target was intercepted, a new one was randomly generated. The participant's score was computed as the total score of the swarm. Figure 8 depicts the agreement (i.e. the standard deviation of the angle's absolute differences between the algorithm and the human decisions) vs. the success rates (i.e. how many targets were intercepted by the human participant). As shown in Figure 8, the success rate of the human participants improves linearly with the reduction of the difference between the decision making of the human and the algorithm.

IV. DISCUSSION

The motivation of this work is to introduce an efficient, non-central solution for a cooperative multi-target interception mission by a swarm of agents. Each agent constructs its own artificial potential field and follows its gradient. We formulated two such functions, one for the deterministic case where the locations of all the agents, obstacles and targets are known; and a stochastic function where these are known as probability distributions. We proved that following the gradient of such functions results in intercepting all static targets in a finite time span. Nevertheless, for situations where the targets move, such proof is out of reach. We therefore conducted numerical experiments to validate the efficiency of this scheme.

Figure 3.(b) demonstrates the agreement of the resulting trajectories of the swarm agents in the planar case compared with a non-real-time SA procedure which minimizes the overall length of these graphs. We consider the latter as a benchmark for optimality for the deterministic static-target case (in the sense of path length). It was demonstrated that our solution provides a real-time solution with path lengths similar to that of the optimal solution but two orders of magnitude faster.

With regard to the overall resulting path length, the S-NF, the S-PNF schemes and the optimal solution demonstrate very close behaviors which almost coincide for large swarms. For example, for a swarm with $N = 16$, the average overall path length an agent traversed was 1.4 unit lengths when following S-NF based motion planning; and an agent of an S-PNF controlled swarm traversed along 1.8 unit lengths compared with the optimal average path of 0.7 unit lengths. For larger swarms, task allocation is responsible for the main difference between the optimal solution and that of the suggested solution.

That is, as more agents or targets are introduced in the workspace, the resulting trajectories become shorter (see Figures 3 and 4.(b)) and therefore the governing differences in behavior may be attributed to the choice of targets. Furthermore, in such situations, the task allocation mission is reduced to choosing the nearest target. The same behavior was observed for the spatial case (see Figure 7), with slightly less agreement with the optimal solution. This is so since in the spatial case, the number of potential neighboring agents grows by $n : n^2$ compared to the planar case. Therefore, a given agent's choice of a target in the former case is affected by more agents than in the planar case. It should be further noted that in all simulations all targets were intercepted. This was also the case in all the UAV swarm experiments when the targets were either stationary or randomly traversing.

The above experiments were conducted to evaluate the efficiency of the algorithm in terms of the trajectory length. Nevertheless, optimality is a much more intricate term that may optimize any combination of the resulting path length, energy loss, required acceleration (which will manifest as smooth trajectories), interception success rate etc. Human decisions, on the other hand, are perceived as close to the optimal with regard to the above criteria. This is the rationale behind the last two sets of experiments. It should be noted that the game depicted in Figure 8.(a), did not accurately simulate the underlying situation we discussed in this paper, as the participants did not sense the entire surrounding. For this reason, we designed a second human-decision game (Figure 8.(b)) in which the participants could see all targets and agents simultaneously.

Figure 8.(c) and (d) demonstrate the fact that in both experiments the closer a player's decisions are to the algorithm, the more successful the outcome. Here, we consider the portion of interceptions of each participant out of the overall number of interceptions as a measure of success.

Furthermore, the resulting trajectories are also not optimal, as the other agents are avoided without one knowing the predictions of their trajectories. But it should be noted that [21] demonstrated how the NF and the PNF solutions can be further improved by accounting for predictions.

In this paper we consider the case where agents share their sensory data (i.e. their location, targets locations), to all other agents. Nevertheless, it is important to note, although out of the scope of this paper, that additional experiment sets where conducted with similar results, where the agents share their knowledge only to agents that are located in their vicinity.

One of the main challenges in multi-robot multi-target missions is reducing the computation complexity. Planning a multi-robot multi-target mission is usually divided into two steps [24]:

1) Assignment of targets to agents, where the suggested solutions may be based on optimal search or heuristics.

Obviously, as the number of targets and/or agents increases, the optimal solution is more complex and often is not feasible in a bounded time constraint.

2) Trajectory planning: having a set of sub-missions for the agents, a trajectory must be generated for each agent while considering collision avoidance with obstacles and other agents.

When executed independently, the target assignment step does not consider the trajectory planning step and vise-versa. As a result, an optimal solution of one step may cause inferior performance of the other. So, for optimality, these two steps should be executed simultaneously. Thus, a realistic comparison of the proposed swarm interception algorithm with other models requires adding up the complexities of target assignment algorithms with the complexity of path planning algorithms.

Note also that the complexity of target assignment algorithms is usually given by the count of the number of ''iterations'' rather than the number of floating point operations, though the details of an ''iteration'' may vary across architectures [25]. For the following discussion, the number of agents is marked by n, the number of targets by m and the dimension of $\mathcal{W}$ is d (the number of obstacles has a minor effect on the complexity and therefore is not considered in the discussion).

Gerkey and Matarić [25] provide a thorough survey on algorithms for task allocation and their respective complexities. For example, [26] introduces a method named MURDOCH, having a complexity of $O(n)$ for a single mission task with n agents. In the swarm interception problem discussed here, the complexity of the MURDOCH algorithm for the task allocation alone sums up to $O(nm)$. An additional complexity in the range of $O(nm)$ is also required for communication between the agents. The Alliance algorithm [27] also demonstrates a complexity of $O(mn)$ for the task allocation itself, and an additional $O(m)$ for communication. Note that the term ''iteration'' is interpreted differently in cases where the tasks are computed simultaneously or sequentially.

Known path planning algorithms are divided into those that construct a graph (called a *roadmap*) that represents the $\mathcal{W}_{free}$, and to those that use the idea of a potential field. The complexity for constructing a Voronoi roadmap is $O(n^{\lceil d/2 \rceil})$ - see [28] (e.g. for $d = 3$ the complexity is $O(n^2)$). The complexity of the searching through this roadmap is $O(E)$ where E is the number of edges in the graph (see [29]). For example, the case of d-simplex enclosing each entity, requires $E \geq (d+1)m$.

The navigation function algorithm, which exploits the potential field idea, has a complexity of $O(d^2n + n^2)$ (see [10]). The methods available in the literature and discussed above provide the following:

1) Performing an optimal task allocation that is followed by a Voronoi graph approach results in a complexity of $O(mn + n^{\lceil d/2 \rceil} + nE)$.
2) Performing an optimal task allocation that is followed by a navigation function results in a computational load for each agent in the range of $O(mn + d^2n^2 + n^3)$.

On the other hand, Eq. 5, Eq. 3 and Eq. 4 involve computations of m distances from the targets, and n distances from the obstacles. Thus, following [10], the overall complexity of the proposed swarm interception algorithm is $O(d^2(m+n) + (n+m)^2)$. Note that this suffices for solving the problem for the entirety of swarm agents. In other words, this estimation should not be multiplied by n since Eq. 5 and Eq. 3 are computed only once (because $\beta_i = \frac{\beta}{\|x - x_i^{(a)}\|_2^2 - \rho_i^2}$). A simple calculation leads to the conclusion that our proposed algorithm is preferable as long as $m < n^{\alpha}$ with $\alpha = 1 + log(n-1)/log(n)$, which equates for situations where the number of targets is about the square of the number of agents.

V. CONCLUSION & FUTURE WORK

This paper focuses on multi-agent multi-target interception missions. We introduce two new versions of a navigation function for a swarm interception mission: the S-NF for the deterministic case and S-PNF for the stochastic case. We show that these functions have similar properties to the original Navigation Function and the Probability Navigation Function respectively, for scenarios with multiple agent and multiple targets. We provide analytic proofs that for scenarios with stationary targets for both schemes the S-NF and the S-PNF. The S-NF scheme avoids the obstacles for the deterministic case, and the S-PNF minimizes the probability of a collision for the stochastic case. The experimental results show that convergence time decreases linearly with the number of agents and is robust to uncertainties and to the dynamics of the targets. We further show that the computational complexity grows linearly with the number of agents, since for every additional agent only a single function is added, which is of practical importance.

The proposed schemes can be implemented in various practical missions, ranging from defense related tasks where a swarm of agents is required to protect a strategic zone from hostile invaders to agricultural missions where a swarm of pests can damage the crop. In both cases, the threatening targets can be static or dynamic, operating around various types of obstacles in 2D as well as 3D environments. The paper also compares the performance of the proposed schemes to the behavior of humans during interception-style games (see Figure 8) and shows that the agents' trajectories produced by the S-NF and the S-PNF are similar to, and in some cases even superior to the trajectories of humans.

The analytical proofs we provided are applicable to cases where the targets are static. Interestingly, we also observed convergence in scenarios involving dynamic targets. This phenomenon, which warrants further exploration, was consistently noted across all scenarios we examined. Recalling Farber's 2003 work on robotics related topological complexity [30], it becomes apparent that a deeper

understanding of dynamic targets necessitates an investigation into non-structurally stable gradient flows within the theory of dynamic systems. The authors of this paper intend to focus future research efforts in this direction.

Additionally, it was observed that our extended NF approaches for swarm behavior aligns closely with the strategies humans employ in interception tasks. This parallel suggests a potential area for further study, as it may provide valuable insights into both human cognitive processes and swarm dynamics.

A. DATA AVAILABILITY

The datasets generated and analysed during the current study are available in the Google-Drive repository, at https://bit.ly/3AduVjB.

APPENDIX - PROOF OF CONVERGENCE FOR THE CASE OF STATIC TARGETS

To formally prove Theorem 1, we need the following perquisites:

Definition 3: Given $\phi : \mathbb{R}^d \to \mathbb{R}$*, a point* $c \in \mathbb{R}^d$ *satisfying* $\nabla\phi(c) = 0$*, is called a critical point of* ϕ*. If the Hessian* $\nabla^2\phi$ *is positive (negative) definite at c, then it is a local minimum (maximum) of* ϕ *(For semi-positive and semi-negative definite Hessians this test is not conclusive). If c is neither a minimum nor a maximum point, it is said to be a saddle point. A critical point is called isolated if all its eigenvalues do not vanish. A saddle point at c is called a strict saddle if* $\nabla^2\phi(c)$ *has at least one negative eigenvalue.*

Theorem 3 (Panageas & Piliouras): Let $\phi : \mathbb{R}^d \to \mathbb{R}$ *be a twice continuously differentiable function in* $\mathcal{W}$ *such that all its saddle points are strict saddles,* $\sup_{x\in\mathcal{W}} \|\nabla^2\phi(x)\|_2 \leq L < \infty$*, and* $\mathcal{W}$ *is forward invariant, then a gradient descent procedure with step-size* $0 < a < 1/L$ *from any initial condition almost surely will not converge to a saddle point.*

The intuitive idea behind the claim that $\varphi_i(x)$ (say, the NF) will almost surely converge to a minimal point, can be made clear if one thinks of the initial position as being exactly on a ridge that points towards a saddle point. That is, being in a subspace W which is of co-dimension 1. In such a case, the agent is due to converge to the saddle point. Nevertheless, if one randomly chooses an initial position. this will happen with zero probability. A more exact description of the above is that the map $g : \mathcal{W} \to \mathcal{W}$ defined as $g(x) = x - a\nabla\varphi$, deforms a neighborhood of each critical point diffeomorphically. This implies the existence of a local stable center manifold W that contains all points that are locally non-escaping (including the critical point itself). Since W is of a dimension equal to the number of non-negative eigenvalues of $\nabla^2\varphi$, and φ has the strict saddle property, one concludes that W has a positive co-dimension which includes the set of points promised by the *Stable Manifold Theorem*, that actually converge to the critical point is a subset of W. The theorem of Panageas and Piliouras circumvents the requirement for φ being globally Lipschitz by requiring $\mathcal{W}$ to be forward invariant which is obvious in our case.

Note that the swarm situation differs from the traditional motion planning problem of a single agent by the fact that while the agent of interest moves, so do the other agents. This implies that one cannot apply the Panageas and Piliouras theorem directly to φ_i since it changes in time. To overcome this, we think of the swarm movement as being performed sequentially. That is, for the sake of the proof alone, we examine how the swarm performs where at every time step only one agent moves while all others remain still. It should be emphasized that this does not imply anything about the practical implementation of our scheme since as far as the analytical proof concern dt is infinitesimal.

Specifically, at every time step t the i-th agent moves following the gradient descent of $\varphi_i(x)$ at t.. Excluding the case where the agent will fall into a local minima of $\varphi_i(x)$ (we shall resolve this later), one should refute a claim that the agent may converge into a local minimum in one of the following cases: (1) when some other agent $j \neq i$ moves; (2) when the i agent moves while all other agents are still.

(1) The function φ_i domain contains subsets W which, due to the Panageas & Piliouras theorem, we prefer the agent not to enter. Assuming that at time t this did not occur, the question is whether this may occur due to the movement of the j-th agent. But note that since W are of Lebesgue measure zero and that the j-th agent moves in steps of finite length α, this has zero probability of occurrence.

(2) Applying the Panageas & Piliouras theorem again, implies that this would not end with the ith agent falling into a local minima.

So, in order to guarantee that the gradient descent scheme over φ_i, will converge to saddle points with probability zero, it remains to be shows that:

- At every time step only the targets constitute the minimum points of $\varphi_i(x)$ in $\mathcal{W}$.
- $\varphi_i(x)$ satisfies the strict saddle property in $\mathcal{W}$.
- $\sup_{x\in\mathcal{W}} \|\nabla^2\varphi_i(x)\|_2 < \infty$ (see Lemma 1).

We shall provide the two first requirements in Remark 1 and Proposition 1.

Remark 1: Since φ_i *is the composition* $\sigma \circ \hat{\varphi}$ *where the function* $\sigma(x) = \frac{x}{1+x}$ *is strictly increasing, the critical points of* $\hat{\varphi}$ *and* φ_i *and their corresponding indices coincide [19]. Thus, from now on we shall consider only* $\hat{\varphi}$*:*

$$\hat{\varphi} = \frac{\gamma}{\beta} \tag{7}$$

The gradient of $\hat{\varphi}$ is:

$$\nabla\hat{\varphi} = \frac{\nabla\gamma}{\beta} - \frac{\gamma\nabla\beta}{\beta^2} \tag{8}$$

So the following is also essential:

Proposition 1: There is an ordered set $(K_1, K_2, \ldots, K_{\|\mathcal{T}\|})$*, such that all the critical points of* $\hat{\varphi}_i(x)$ *that are not located at the target points, are strict saddles.*

Proof: We start the proof by induction and define:

$$\hat{\varphi}_n := \frac{1}{\beta}\gamma_1^{K_1}\gamma_2^{K_2}\cdots\gamma_n^{K_n}$$

where $1 \leq n < \|\mathcal{T}\|$ and $\mathscr{C}_n$ as the set of critical points of $\hat{\varphi}_n$ that are not at any of the target points. As per Eq. 3 we can rewrite $\hat{\varphi}$ in terms of $\hat{\varphi}_n$:

$$\hat{\varphi} = \hat{\varphi}_n\gamma_{n+1}^{K_{n+1}}\gamma_{n+2}^{K_{n+2}}\cdots\gamma_{\|\mathcal{T}\|}^{K_{\|\mathcal{T}\|}}$$

Note that $\nabla\hat{\varphi}_{n+1} = \nabla\hat{\varphi}_n\gamma_{n+1}^{K_{n+1}} + \hat{\varphi}_n\nabla\left(\gamma_{n+1}^{K_{n+1}}\right)$. Since $\nabla\hat{\varphi}_{n+1}$ vanishes at $c \in \mathscr{C}_{n+1}$,

$$\nabla\hat{\varphi}_n = -K_{n+1}\frac{\hat{\varphi}_n\nabla\gamma_{n+1}}{\gamma_{n+1}}$$

A simple differentiation yields the expression for the Hessian at c that is given by:

$$\begin{aligned}&\nabla^2\hat{\varphi}_{n+1}\\ &= \gamma_{n+1}^{K_{n+1}}\left[\nabla^2\hat{\varphi}_n\right.\\ &\quad\left. - \frac{K_{n+1}\hat{\varphi}_n}{\gamma_{n+1}}\left(\frac{(\nabla\gamma_{n+1})(\nabla\gamma_{n+1})^T}{\gamma_{n+1}}(K_{n+1}+1) - 2I\right)\right] \quad (9)\end{aligned}$$

Note that the Hessian of $\gamma_i = \|x - x_i^{(t)}\|_2^2$ for all i is the identity matrix multiplied by two, which corresponds to the last term of Eq. 9. We pursue the eigenvalue of $\nabla^2\hat{\varphi}$ which is a sum of two matrices. Recall Weyl's inequality [31] that states that if A and B are two Hermitian matrices then $C = A + B$ is also Hermitian and $a_1 \geq a_2 \geq \cdots \geq a_n$, $b_1 \geq b_2 \geq \cdots \geq b_n$ and $c_1 \geq c_2 \geq \cdots \geq c_n$ be their respective eigenvalues, then, $c_{i+j-1} \leq a_i + b_j$

Next, note that a quadratic form vv^T of a vector $v \in \mathbb{R}^d$ has a zero eigenvalue of multiplicity $N-1$, and an additional eigenvalue $\|v\|^2$. To confirm this, note that all the columns of vv^T are dependent which associatively implies that $(vv^T)v = v\|v\|^2$. Therefore, the single non-vanishing eigenvalue of $\frac{(\nabla\gamma_{n+1})(\nabla\gamma_{n+1})^T}{\gamma_{n+1}}$ is 4 (a trivial derivation due to γ_m definition). Moreover, note that given a matrix M with its eigenvalues $\lambda_1, \cdots, \lambda_n$, the eigenvalues of $M - I$ are $\lambda_1 - 1, \cdots, \lambda_n - 1$. This is so since if $Mv = \lambda v$ is satisfied and $(M-I)v = \lambda' v$, then $\lambda' = \lambda - 1$ are eigenvalues of $M - I$. The maximal eigenvalue of the right term is

$$\frac{K_{n+1}\hat{\varphi}_n}{\gamma_{n+1}}\left(\frac{(\nabla\gamma_{n+1})(\nabla\gamma_{n+1})^T}{\gamma_{n+1}}(K_{n+1}+1) - 2I\right),$$

in Eq. 9 is equal to $\lambda_{max} = \frac{K_{n+1}\hat{\varphi}_n}{\gamma_{n+1}}(4K_{n+1}+2)$.

In order for $\nabla^2\hat{\varphi}_{n+1}$ to have at least one negative eigenvalue in a critical point c, the maximal eigenvalue of $\nabla^2\hat{\varphi}_n$ (the leftmost term of Eq. 9) must be smaller than λ_{max}. For convergence, K_{n+1} should be set to ensure this requirement, which is possible since (1) $\nabla^2\hat{\varphi}_n$ is bounded as will be proven in Lemma 1; (2) the step size is finite; (3) at each induction step, a target is added in a given finite distances from the other targets implying that there is a minimal finite $\hat{\varphi}_n(c) > 0$. This implies that c is not a minimal point of $\hat{\varphi}_{n+1}$. Explicitly, c may be a maximum or a strict saddle of $\hat{\varphi}_{n+1}$. To complete the proof note that inductively one starts with $\hat{\varphi}_{n=1}$ which is exactly the classical Navigation Function, and recall that the original Navigation Function has a single minimum point at its target. Applying this proposition inductively proves that one may add targets without damaging the convergence property. ■

Since the set of critical points of φ coincide with the critical points of $\hat{\varphi}$ (Remark 1), applying the Panageas and Piliouras Theorem proves our claim also for φ.

Eq. 8 implies that the critical points are not located at the boundaries of the obstacles which are not close to the targets. Based on that and examining Eq. 9, the constant K_{n+1} needs to be a relatively small number. Experimentation shows that the values for all K_n's can be equal 1 for convergence.

Lemma 1: The Hessian of φ_i is bounded by $\sup_{x\in\mathcal{W}}\|\nabla^2\varphi_i(x)\|_2 < \infty$.

Proof: Note that

$$\nabla\varphi = \frac{\beta\nabla\gamma - \gamma\nabla\beta}{(\beta+\gamma)^2}, \quad (10)$$

so the Hessian $\nabla^2\varphi$ is:

$$\begin{aligned}&\frac{1}{(\beta+\gamma)^2}\left(\beta\nabla^2\gamma - \gamma\nabla^2\beta\right.\\ &\quad - \frac{2}{\beta+\gamma}\left(\beta\nabla\gamma\nabla\gamma^T - \gamma\nabla\beta\nabla\beta^T\right.\\ &\quad\left.\left. + \gamma\nabla\gamma\nabla\beta^T - \beta\nabla\beta\nabla\gamma^T\right)\right) \quad (11)\end{aligned}$$

But note that β vanishes only on the boundary of the obstacles $\partial\mathcal{O}_i$. Also note that γ vanishes only at the target points $\mathcal{T}$. Given that both are positive in $\mathcal{W}\setminus(\partial\mathcal{O}_i\cup\mathcal{T})$, their sum cannot vanish in $\mathcal{W}$ which completes the proof. ■

To complete the proof of Theorem 1, note that a full discussion and proof that the minimum point of the PNF (with a single agent) is at the target can be found at [17]. So, Remark 1 and Proposition 1 also holds for the S-PNF.

Lemma 1 holds as β vanishes if one of the probabilities for a collision $p_j(x)$'s equals Δ. (e.g. the agent is in the vicinity of an entity such that its probability for collision accedes the allowable probability for collision Δ). So in any other case $\beta(x) + \gamma(x) \neq 0$ in $\mathcal{W}_i$ which completes the proof.

To complete the proof of Theorem 2, note that a full discussion and proof that the minimum point of the PNF (with a single agent) is at the target can be found at [17]. So, Remark 1 and Proposition 1 holds also for the S-PNF.

Lemma 1 holds as β vanishes if one of the probabilities for a collision $p_j(x)$s equals Δ. That is, if the agent is in the vicinity of some entity such that its probability for collision accedes the allowable probability for collision Δ. So in any other case $\beta(x) + \gamma(x) \neq 0$ in $\mathcal{W}_i$ which completes the proof.

REFERENCES

[1] E. Kagan, N. Shvalb, S. Hacohen, and A. Novoselsky, "Multi-robot systems and swarming," in *Autonomous Mobile Robots and Multi-Robot Systems: Motion-Planning, Communication and Swarming*. Hoboken, NJ, USA: Wiley, 2019, p. 199.

[2] T. A. Phan and R. A. Russell, "A swarm robot methodology for collaborative manipulation of non-identical objects," *Int. J. Robot. Res.*, vol. 31, no. 1, pp. 101–122, Jan. 2012.

[3] G.-Z. Yang, J. Bellingham, P. E. Dupont, P. Fischer, L. Floridi, R. Full, N. Jacobstein, V. Kumar, M. McNutt, R. Merrifield, B. J. Nelson, B. Scassellati, M. Taddeo, R. Taylor, M. Veloso, Z. L. Wang, and R. Wood, "The grand challenges of science robotics," *Sci. Robot.*, vol. 3, no. 14, Jan. 2018.

[4] K. N. McGuire, C. De Wagter, K. Tuyls, H. J. Kappen, and G. C. H. E. de Croon, "Minimal navigation solution for a swarm of tiny flying robots to explore an unknown environment," *Sci. Robot.*, vol. 4, no. 35, Oct. 2019, Art. no. eaaw9710.

[5] C. Nieto-Granda, J. G. Rogers, and H. I. Christensen, "Coordination strategies for multi-robot exploration and mapping," *Int. J. Robot. Res.*, vol. 33, no. 4, pp. 519–533, Apr. 2014.

[6] M. Dorigo, G. Theraulaz, and V. Trianni, "Reflections on the future of swarm robotics," *Sci. Robot.*, vol. 5, no. 49, 2020, Art. no. eabe4385.

[7] L. Garattoni and M. Birattari, "Autonomous task sequencing in a robot swarm," *Sci. Robot.*, vol. 3, no. 20, 2018, Art. no. eaat0430.

[8] D. Morgan, G. P. Subramanian, S.-J. Chung, and F. Y. Hadaegh, "Swarm assignment and trajectory optimization using variable-swarm, distributed auction assignment and sequential convex programming," *Int. J. Robot. Res.*, vol. 35, no. 10, pp. 1261–1285, Sep. 2016.

[9] R. W. Beard, T. W. McLain, M. A. Goodrich, and E. P. Anderson, "Coordinated target assignment and intercept for unmanned air vehicles," *IEEE Trans. Robot. Autom.*, vol. 18, no. 6, pp. 911–922, Dec. 2002.

[10] E. Rimon, "A navigation function for a simple rigid body," in *Proc. IEEE Int. Conf. Robot. Autom.*, Jan. 1991, pp. 546–551.

[11] D. V. Dimarogonas and K. J. Kyriakopoulos, "Connectedness preserving distributed swarm aggregation for multiple kinematic robots," *IEEE Trans. Robot.*, vol. 24, no. 5, pp. 1213–1223, Oct. 2008.

[12] Y. Diaz-Mercado and M. Egerstedt, "Multirobot mixing via braid groups," *IEEE Trans. Robot.*, vol. 33, no. 6, pp. 1375–1385, Dec. 2017.

[13] W. Zhang, W. Wang, J. Zhu, and D. Zhao, "Multi-vehicle interaction scenarios generation with interpretable traffic primitives and Gaussian process regression," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Oct. 2020, pp. 1197–1204.

[14] J. K. Gupta, M. Egorov, and M. Kochenderfer, "Cooperative multi-agent control using deep reinforcement learning," in *Proc. Int. Conf. Auton. Agents Multiagent Syst.*, São Paulo, Brazil. Heidelberg, Germany: Springer, May 2017.

[15] D. V. Dimarogonas and K. J. Kyriakopoulos, "A feedback control scheme for multiple independent dynamic non-point agents," *Int. J. Control*, vol. 79, no. 12, pp. 1613–1623, Dec. 2006.

[16] S. G. Loizou and K. J. Kyriakopoulos, "Navigation of multiple kinematically constrained robots," *IEEE Trans. Robot.*, vol. 24, no. 1, pp. 221–231, Feb. 2008.

[17] S. Hacohen, S. Shoval, and N. Shvalb, "Probability navigation function for stochastic static environments," *Int. J. Control, Autom. Syst.*, vol. 17, no. 8, pp. 2097–2113, Aug. 2019.

[18] E. Rimon and D. E. Koditschek, "The construction of analytic diffeomorphisms for exact robot navigation on star worlds," *Trans. Amer. Math. Soc.*, vol. 327, no. 1, pp. 71–116, 1991.

[19] D. E. Koditschek and E. Rimon, "Robot navigation functions on manifolds with boundary," *Adv. Appl. Math.*, vol. 11, no. 4, pp. 412–442, Dec. 1990.

[20] J. Palis and S. Smale, "Structural stability theorems," in *The Collected Papers of Stephen Smale*, vol. 2. Singapore: World Scientific, 2000, pp. 739–747.

[21] S. Hacohen, S. Shoval, and N. Shvalb, "Applying probability navigation function in dynamic uncertain environments," *Robot. Auto. Syst.*, vol. 87, pp. 237–246, Jan. 2017.

[22] *Multi Agent Multi Target Interception Using S-NF*. Accessed: May 14, 2019. [Online]. Available: https://www.youtube.com/watch?v=9y5fUu443GY

[23] *S-PNF As a Model for Human Cooperative Behavior*. Accessed: May 14, 2019. [Online]. Available: https://www.youtube.com/watch?v=sZFPfzwEsZQ

[24] Z. Yan, N. Jouandeau, and A. A. Cherif, "A survey and analysis of multi-robot coordination," *Int. J. Adv. Robotic Syst.*, vol. 10, no. 12, p. 399, Dec. 2013.

[25] B. P. Gerkey and M. J. Mataric, "Multi-robot task allocation: Analyzing the complexity and optimality of key architectures," in *Proc. IEEE Int. Conf. Robot. Autom.*, Sep. 2003, pp. 3862–3868.

[26] B. P. Gerkey and M. J. Mataric, "Sold! Auction methods for multirobot coordination," *IEEE Trans. Robot. Autom.*, vol. 18, no. 5, pp. 758–768, Oct. 2002.

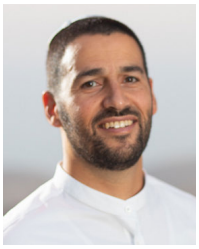
[27] L. E. Parker, "ALLIANCE: An architecture for fault tolerant multirobot cooperation," *IEEE Trans. Robot. Autom.*, vol. 14, no. 2, pp. 220–240, Apr. 1998.

[28] M. Sharir, "Arrangements in higher dimensions: Voronoi diagrams, motion planning, and other applications," in *Proc. Workshop Algorithms Data Struct.*, Ottawa, ON, Canada, Aug. 1995, pp. 109–121.

[29] M. Thorup, "Undirected single-source shortest paths with positive integer weights in linear time," *J. ACM*, vol. 46, no. 3, pp. 362–394, May 1999.

[30] M. Farber, "Topological complexity of motion planning," *Discrete Comput. Geometry*, vol. 29, no. 2, pp. 211–221, Jan. 2003.

[31] H. Weyl, "Inequalities between the two kinds of eigenvalues of a linear transformation," *Proc. Nat. Acad. Sci. USA*, vol. 35, no. 7, pp. 408–411, Jul. 1949.



SHLOMI HACOHEN received the M.Sc. degree in electrical engineering and the M.B.A. degree in economics and business management from Ariel University, Israel. He is currently a Lecturer with the Engineering Faculty, Ariel University. He is also the Head of the Robotics and Kinematics Laboratory. His main research interests include uncertain dynamic environments, nonlinear estimation, and robot motion planning.

SHRAGA SHOVAL is currently a Professor and the Dean of the Faculty of Engineering, Ariel University, Israel. He has authored more than 200 scientific publications. His current research interests include the development of mobile robotics and autonomous systems, motion planning, traffic safety, trusted autonomy, systems engineering, and surface science.

NIR SHVALB is currently an Associate Professor with the Faculty of Engineering and the Vice Dean of Ariel University, Israel. He heads the Robotics Research Laboratory. He is the Founder of several robotics startup companies amongst are Momentis Surgical and W Endoluminal Robotics. His main research interests include medical robotics, global path planning, and theoretical foundations of robotics.