IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received February 5, 2020, accepted March 7, 2020, date of publication March 10, 2020, date of current version March 26, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2979929

Robotic Swarm Motion Planning for Load Carrying and Manipulating

ODED MEDINA[1], **SHLOMI HACOHEN**[1], **AND NIR SHVALB**[2]
[1]Department of Mechanical Engineering, Ariel University, Ariel 40700, Israel
[2]Department of Industrial Engineering, Ariel University, Ariel 40700, Israel

Corresponding author: Oded Medina (odedmedina@gmail.com)

ABSTRACT Certain species of ants can carry out tasks in dense work spaces while maintaining their ability to accurately manipulate heavy loads, and these advantages are of interest to the robotics community. We consider a robotic swarm of $N \geq 6$ agents that assumes the task of moving a load through a cluttered space. This forces the swarm to carefully manipulate the orientation of the load, while transporting it to its destination point. We model this scenario as a 6-PPSS (Prismatic-Prismatic-Spherical-Spherical) redundant mobile platform, having six degrees of freedom. As with insects, the multitude of agents enables sharing the burden of the load in the case that one or more agents are blocked by an obstacle. We model this by a semi-algebraic set of constraints on the distances between the agents and the load. We apply an Extended Kalman Filter routine, in order to estimate their relative locations. We show how the estimation-error is reduced when position-information is shared among the agents. These estimations are then used to calculate the full configuration and investigate the effect of position estimation error on the platform heading error. We show how motion planning can then be calculated in the model's full configuration space and demonstrate this with a distributed control scheme. To reduce the search time, we introduce a variant of the crawling probabilistic road map motion planning algorithm under a set of kinematic constraints and work-space obstacles. Finally, we exemplify our algorithms on several simulated scenarios.

INDEX TERMS Swarm, load, extended Kalman filter, parallel platform, crawling probabilistic road map.

I. INTRODUCTION

A. ENTOMOLOGY PARALLELS

Ants demonstrate the ability to cooperatively transport objects without prior knowledge of the objects' shape and mass [1]. Careful observation reveals that their modus operandi demonstrates substantial adaptability to varying object sizes and terrains while withstanding failure of individual insects. At least in some cases and to some extent, this may involve information sharing regarding agents' forces exerted upon them [2]. An individual insect may hold the object firmly (e.g. apply force and torque) or alternatively, may just pull the object (e.g. apply force) ([3] and [2], respectively).

B. ROBOTIC SWARMS THAT TRANSPORT LOADS

In the robotics literature (e.g. [4, §9], [5]), a swarm usually refers to a multitude of agents that assume a cooperative mission and that act individually without a central controller. In most cases, the agents have limited communication capabilities. Among the tasks investigated are mapping [6], [7], searching [8], and loads manipulating [9].

In *non-prehensile* approaches to cooperative manipulation, the load is transported via pushing or rolling while stabilizing the direction of the movement [10]. Ardakani *et al.* investigated the task of transporting a plate by a swarm of omni-directional mobile robots by means of friction [11]. They assumed Coulomb friction at the contact points between the robots and the plate. A distributed force and torque controller for such agents was proposed [12]. Rus *et al.* [13] investigated global control of a set of agents that can communicate and that are given the task of pushing objects. A similar task was studied in [14], in which a decentralized group of robots moved objects while maintaining "object closure" (that is, the agents enclosed the object so that it could be manipulated). The incorporation of disc-obstacles was considered in [15]. There, the researchers constructed an artificial potential field (APF) that holds the agents at the object's

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

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

boundary, such that obstacles were avoided, while heading to the target position using navigation function. Kalat *et al.* [16] presented a decentralized, communication-free load carrying approach where each agent relies on the behavior of a cooperative virtual agent, so the real team formation is not needed.

Reference [17] proposed a distributed motion planner for the task. The agents surrounded the object (which may have an arbitrary shape) and thus managed to estimate the object's centroid. The motion controller proposed in [17] is capable of both translating and rotating the object, by specifying the agents' speed with respect to the centroid of the object. Cooperative manipulation can also be achieved without communication, using vision occlusion [18], passive caster [19], and inter-robot force measurements [20], [21].

C. LOAD ORIENTING

Agents that carry a load and orient it can be modeled as a redundant parallel manipulator, where each of its base points can traverse the plane. Parallel platform, such as the Stewart-platform and Delta, have been extensively documented in the literature (c.f. [22], [23]). It is widely agreed that for these mechanisms, the payload-to-weight ratio (i.e. the load's weight over the mechanism's total weight, calculated in a regular configuration) is higher than that of serial chains, as the payload distributes among the actuators. Accuracy is also higher since the joint errors do not accumulate. Additionally, as the actuators are usually positioned at the base, they excel also in their structural rigidity [24]. On the other hand, the work-space of a parallel-platform is limited, as it is a subspace of the intersection of the work-spaces of its limbs [25]. To overcome the above, researchers have suggested several structural modifications, for example, the 6- degree of freedom (DOF) mechanisms with linear base-point-actuators such as the 3-PRPS parallel-platform [26] by Shim *et al.* and the 3RPRS introduced in [27]. Totally mobile parallel platforms (i.e. those that possess 2-translational degrees of freedom at their base) have also been investigated. For example, Ben-Horin and Shoham [24] introduced a six-DOF 3PPRS mechanism, the 3PPSR platforms, which were previously introduced by Tahmasebi and Tsai [28] and by Pernette *et al.* [29]. The 3-PPSP was built by Byun and Cho [30], and the 3PPRS by Kim *et al.* [31].

D. LOCALIZATION

The task of carrying a load using a swarm requires cooperation between the agents. In most cases, an agent's motion depends on some/all other agents' positions. The term *localization* is used in the literature to denote the process of sensing the agents' absolute locations, or sensing their (affine) relative locations (e.g. to one of the agents which may act as a leader). For clarity, we shall refer to the latter as the *swarm's distribution*. Information about the swarm's distribution is needed in many of the previously described cooperative manipulation strategies. Basiri *et al.* [32] suggested a sensing scheme to detect the agents' positions – a robot generates chirps at a predefined rate and frequency. The sound waves are received by an on-board microphone array and the data is used to estimate the target's direction. Soysal and Sahin [33] studied the possible clustering-configurations (aggregation) of the swarm's distribution, when confined to a closed arena. Using a probabilistic aggregation model inspired by social insects, they proposed a macroscopic model for predicting the final distribution of aggregates, in terms of the parameters of the aggregation behavior, the arena size, and the sensing characteristics of the robots.

Contribution and Paper Organization: As indicated, the load carrying swarm problem is of interest to the research community. A number of papers have described swarms and mechanical systems that may be considered as swarms of agents carrying loads (i.e. parallel platforms that are mobile to some extent). Table 1 compares the main characteristics of these solutions and swarm solutions with the proposed solution.

Here we address the problem that arises when the carried load meets an obstacle (as for example, when insects carry a load inside a tunnel). To this end, the swarm is capable of fully controlling the load's spatial orientation as well as its planar location. This is addressed in the first column of Table 1. To avoid obstacles in the work space, redundant mobility is preferable. This is facilitated by the participation of a large number of agents in the task. One possibility is that $N > 6$ agents carry the load; alternatively, agents may disengage from the load (i.e. an *agent translating freely*, see Table 1, fourth column). Note that in a case of a large workspace obstacle, the agents may address the problem by transporting the load around the obstacle while moving together and keep-

TABLE 1. Literature review comparison.

	Load DOF	# Agents	Grasp type	Free agents	Obstacles avoidance	Communication	Decentralized control
Herein	6	$N \geq 6$	Spatial forces	✓	CPRM	✓	✓
Gelblum et al. [2015] ants in certain cases	6	$N \geq 6$	Spatial forces	✓	✓	very limited	✓
Ardakani et al. [2017]	3	$N \geq 3$	Planar Friction	✓	APF	✓	✓
Wang et al. [2018]	3	$N \geq 3$	Planar Friction	✗	Local	✗	✓
Pereira et al. [2004]	3	$N \geq 4$	Planar Friction	✗	✗	✗	✓
Fink et al. [2008]	2	$N \geq 3$	Enclosing	✗	APF	✓	✓
Habibi et al. [2015]	3	$N \geq 3$ and $N \gg 6$,	Enclosing	✗	✓	✓	✗
Chen et al. [2015]	2	$N \geq 3$	Push/pull forces	✓	✗	✗	✓
Stilwell and Bay [1993]	2	$N \geq 3$	Push/pull forces	✓	✗	✗	✓
Groß et al. [2006]	2	$N = 6$	Spatial forces	✗	✗	✗	✓
Shoval and Shoham [2001]	6	$N = 3$	Spatial forces and torques	✗	✗	-	✗

ing the formation. Small obstacles in the workspace necessitate a different approach; in such a case, each agent is forced to avoid the obstacles on its way while continuing to transport the load towards the destination. A centralized/decentralized, local/global motion planner can enable a swarm to overcome obstacles. Here we introduce a decentralized approach for local maneuvers, together with global *crawling probabilistic road map* (CPRM) [35] motion planning. As such, the road map (i.e. the way-points) is known to the entire swarm, as with ant swarms (see Subsection II-B below). The last three columns of Table 1 refer to this issue. In [2] researchers introduced a set of experiments in which a single informed ant leads the entire carrying swarm to the target position. This is equivalent to saying that that the leading ant provides a set of load-way-points along the way. Here, we introduce a similar concept where each given way-point also includes the agents' positions.

Note that the number of swarm agents that participate in the task of carrying the load may exceed six. Nevertheless, to fully determine a spatial load in the non-prehensile case, six agents will suffice. Thus, we assumed that at each moment, only six agents are actively carrying the load. We assumed that the base points (agents) can traverse the plane, and thus addressed the task of controlling both the agents and the orientation of the load while carrying it. To maximize flexibility of the solution, the agents are modeled as having the ability to switch roles from time to time (e.g., nonparticipating agents can join the transport team in order to help maneuver the load around an obstacle). This can become essential in congested environments where agents may reach a dead end. To enable orientation of the load, the agents should know their location. Thus, the task of self-locating is also addressed in this paper. The method introduced here resembles the behavior of cooperative transport in ants since it enables more than six agents, is decentralized by nature, and enables spatial re-orientation of the load, which is essential.

This paper is organized as follows: Section II introduces the main assumptions of our model (i.e. the sensors that are available, the means by which the agents communicate and estimate their location, and the underlying kinematic model). In Section III, we show how an Extended Kalman Filter can be applied to solve the swarm's distribution problem (i.e. the means of efficiently extracting position estimations of the agents). Section IV is dedicated to the swarm-load kinematic model that we used to move the load in six dimensions. Section V introduces a variant of the CPRM motion planning scheme that is applied to evade planar obstacles and spatial obstacles. Section VI provides a set of simulated experiments that demonstrate our approach and algorithms in both obstacle-free and congested workspaces. Section VII concludes the paper.

II. MODEL ASSUMPTIONS

A. SENSORS

We assume that each agent is equipped with a magnetometer (compass) and a sensory system that measures the relative angle to all other agents (as with a standard camera). These measurements enable global orientation for all agents, such that the coordinate systems of all agents are parallel. The global configuration of the platform (center-of-gravity and orientation), or at least the position of one of the connected robots, is assumed to be known.

B. COMMUNICATION

In the literature, a swarm commonly refers to simple physical agents with minimal communication needs (or having no communication needs as in [36]). This applies to ants, up to the point of global motion planning [2]:

> *"…we describe field experiments of cooperative transport by P. longicornis ants. By combining the analysis of the load motion with single-ant trajectory data, we find that while the combined force of the group determines the speed of the load, it is individual informed ants that steer the direction of movement."*

More precisely, their experimentation shows that informed ants which carry and steer the load, "lose" their orientation (as their antenna is obstructed by the load) but may regain it by disengaging for a while. In other words, no specific ant stays informed along the way, rather several ants may steer the load along the trajectory. Nevertheless, for our purposes it suffices to say that global information is available.

Here we shall assume that all agents are fully aware of the (estimated) position of all other agents (i.e. they share only their estimated position, not their planned velocities). Thus, in each time-step an agent proceeds towards its own goal position (or a way-point) while orienting and moving the platform (as much as it can) towards the corresponding desired goal position (or a way-point) of the platform. This is of course done while avoiding obstacles and satisfying mechanical constraints (e.g., avoiding collisions between rods).

C. THE KINEMATIC MODEL

An agent i, located at $\boldsymbol{x}_i \in \mathbb{R}^2$, which holds the load at point $\boldsymbol{x}_i^p \in \mathbb{R}^3$, may drop the load, provided that another agent j, located at $\boldsymbol{x}_i \neq \boldsymbol{x}_j$, replaces it by holding the load at $\boldsymbol{x}_j^p$, which is not necessarily equal to $\boldsymbol{x}_i^p$. We apply a non-prehensile approach, in which the agents may only apply forces upon the load. To prevent an individual agent from applying torques upon the load, the connection may be modeled as a sphere joint (S), while the agents that carry the load can traverse the plane. This sums to a 6-PPSS model of a redundant mobile platform (as the *mobility* equals 12), having six degrees of freedom. In this model, each of the 6 base points can traverse the plane (PP) - these are the agents that carry the load at a given moment. As the inverse kinematics of mobile platform is often hard to calculate, and since this is not the main focus of this paper, differential kinematic formalism will be used.

III. LOCALIZATION AND THE SWARM DISTRIBUTION

In what follows, the set $\{1, \ldots, N\}$ of all agents in the swarm is denoted by $\mathcal{A}$. A tilde decoration $\tilde{\zeta}$ indicates an estimation of a quantity ζ. The time-step k is indicated in parentheses and a $i^{th} \in \mathcal{A}$ agent related quantity, by a subscript $\zeta_i(k)$. Locations and measurements are marked by $\boldsymbol{x} \in \mathbb{R}^2$ and $z \in \mathbb{R}$, respectively, and a Gaussian distribution with an expected value μ and a covariance σ will be denoted throughout the paper by $\mathcal{N}(\mu, \sigma)$.

A. THE STATE MODEL

In every time-step k, each agent i measures the set of $N-1$ angles $Z_i(k) := \{z_{i,j}(k)\}_{i,j\in\mathcal{A}}$ towards all other agents $j \in \mathcal{A}$ with respect to a global x-axis. This is essential for the estimation process during which the measurement-updates assume estimated locations $\tilde{X}(k) = [\tilde{\boldsymbol{x}}_1(k), \tilde{\boldsymbol{x}}_2(k), \ldots, \tilde{\boldsymbol{x}}_N(k)]$, which are obviously given in a predetermined local coordinate-system. The agents communicate only their estimated positions, so each agent knows the estimated positions of all other agents.

For real world implementations, a stochastic scenario (i.e. the agents' position can only be estimated) is more realistic than a deterministic one. In such scenarios, the noise of the odometry measurements $\mathbf{d}_i(k)$, $i \in \mathcal{A}$, as well as the measured angles, must be considered.

We assume a discrete-time state model as the location equation for the i^{th} agent:

$$\boldsymbol{x}_i(k) = \boldsymbol{x}_i(k-1) + \mathbf{d}_i(k) + \nu_i(k) \tag{1}$$

where $\mathbf{d}_i(k)$ is a single time-step translation vector, $\nu_i(k) \sim \mathcal{N}(0, \Sigma_i)$ is the process-noise and Σ_i is a semi-positive-definite two-dimensional square matrix that is assumed to be known. The measured angle from one agent to another is a random variable that should be summed with the measurement-noise $\omega(k) \sim \mathcal{N}(0, \sigma_\theta)$. Therefore, the angle measured between the i^{th} agent located at $\boldsymbol{x}_i = (a_i, b_i)$ and some other agent j located at $\boldsymbol{x}_j = (a_j, b_j)$ is given by:

$$z_{i,j}(k) = atan2(b_i - b_j, a_i - a_j) + \omega(k) \tag{2}$$

Thus, the measurement function for agent i is:

$$Z_i(k) = h(X(k)) \tag{3}$$

Here $h(X)$ returns the vector of all relative angles between the agents (see Eq.2). Our algorithm considers the locations of all agents as benchmarks for the estimation process. Since the agents' locations are random variables, and the estimated angles $h(\tilde{X}(k))$ depend on these locations, the uncertainties can be thought of as inflated measurement noises, resulting in location-uncertainty, as well as in sensor-noise.

The position uncertainties can be thought of as noise. So, we add the position covariance to the noise covariance. This results in an "inflated" Gaussian.

To account for the location uncertainties of the i^{th} and j^{th} agents, their location distributions must be combined by a convolution (see [37] for further discussion). For a normal distribution, the convolution product distribution covariance is the sum of the distributions' covariance. Thus, the approximated standard deviation of the measurement-noise (i.e. the angle measured from the i^{th} agent to the j^{th} agent) can be selected as:

$$\sigma_{i,j} = \sigma_\theta + \frac{\lambda_{max}(P_i(k)) + \lambda_{max}(P_j(k))}{d_{i,j}} \tag{4}$$

where $\lambda_{max}(A)$ is the maximal eigenvalue of the matrix A, $P_i(k)$ is the estimation error covariance matrix of the i^{th} agent at time k and $d_{i,j}$ is the estimated distance between the i^{th} agent and the j^{th} agent.

The most common estimation algorithm for non-linear systems is the Extended Kalman Filter (EKF). We apply a variant of this filter for the purpose of position estimation. We assume that the location uncertainty is the error-estimation matrix P_i, which is assumed to be normally distributed and computed by the EKF, as described below. So the matrix $P_i(k)$ is a 2×2 symmetric semi-positive definite matrix.[1]

B. ESTIMATION PROCEDURE

The estimated set of locations of all agents $\in \mathcal{A}$ at time k (estimated by the agent r) is the ordered set $(\tilde{\boldsymbol{x}}_1(k), \tilde{\boldsymbol{x}}_2(k), \ldots)$.

The estimation procedure, calculated for an agent i, is as follows: at each time-step, the agent's estimated location $\tilde{\boldsymbol{x}}_i(k)$ is propagated based on the measured locomotion and using Eq.1. The estimation error propagated covariance is $P_i(k) = P_i(k-1) + \Sigma_i$. The prediction step follows

$$\boldsymbol{x}_i(k) = \boldsymbol{x}_i(k-1) + \mathbf{d}_i(k)$$

the state is then updated:

$$K_i(k) = P_i(k)H_i(k)^\top (H_i(k)P_i(k)H_i(k)^\top + R_i(k))^{-1}$$

and

$$\tilde{\boldsymbol{x}}_i(k) \mapsto \tilde{\boldsymbol{x}}_i(k) + K_i(k)(Z_i(k) - h(\tilde{X}(k)))$$

where R_i is a $(N-1)$-dimensional square diagonal matrix with the diagonal $\sigma_{i,j}$ for all $j \in \mathcal{A} \setminus i$. Here $H_i(k)$ is the Jacobian of $h(X)$ at $\tilde{X}(k)$.

Assume for simplicity $N = 3$ and assume for discussion that the location-uncertainty of the second agent is relatively small. The locations of all agents can be estimated at once (the entire state). But note that such a strategy would in some sense average the covariance of all agents to find the best fit of the measured angles and would then miss the opportunity to use the accurate estimation of the second agent. A better choice would be to estimate the agents' locations one at a time. That is, the first agent would estimate the (say) second agent's location and then (say) the third's. In this case, accurate estimation of the second agent would be used to better estimate the third's. In the general case, the *Rao-Blackwell theorem* states that the variance of the conditional distribution is less or equal to the variance of the estimator of the entire

[1] Here we discuss only the planar case. The spatial case will be discussed in an extended paper.

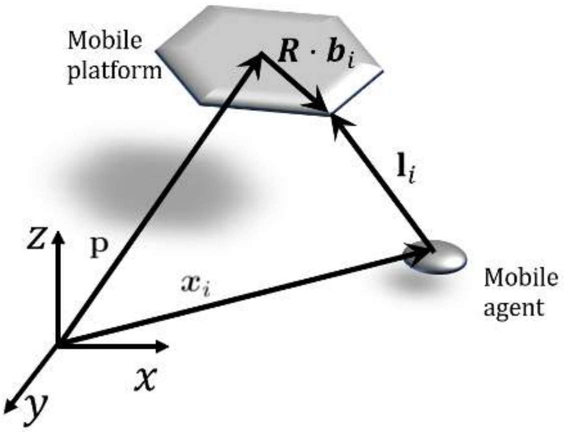

FIGURE 1. The forward kinematics of the six agents that carry the load: A mobile agent at the base is connected by a passive sphere joint to a rod, which is connected by a second passive sphere joint to the load. This is equivalent to a 6PPSS parallel platform. The rods are passive prismatic joints which attain their minimal length due to the platform's weight. These prismatic joints are allowed to extend in the simulation to enable the agents "to lose grip" of the platform in cases where $N > 6$.

state (see [38, pp. 475]). This is exactly the foundation of our algorithm. Note that if the implementation is involved with communication failure, the EKF estimation error increases until this failure is solved. In Section VI, we shall examine the sufficiency of the resulting accuracies for a load carrying swarm.

IV. LOAD ORIENTATION

Figure 1 defines the vectors $\boldsymbol{x}_i, \boldsymbol{x}_i^p$, l_i and p, which are given in global coordinates, with vector b_i written in the load coordinate system. The 3×3 load's rotation matrix R sets:

$$\boldsymbol{x}_i + l_i = p + Rb_i$$

for all $i = 1, 2, \ldots, 6$. Differentiating with respect to time and noting that $|l_i| = const$, yields:

$$l_i^\top \dot{\boldsymbol{x}}_i = l_i^\top \dot{p} + l_i^\top \dot{R} b_i$$

which can be written in the matrix form:

$$J_1 \begin{pmatrix} \dot{p} \\ \boldsymbol{\omega} \end{pmatrix} = J_2 \dot{\boldsymbol{x}}$$

where we define the gripper's Jacobian matrix J_1 as

$$\left[\begin{array}{c|ccc} l_1^\top & l_1^\top \dfrac{\partial R}{\partial \psi} b_1 & l_1^\top \dfrac{\partial R}{\partial \theta} b_1 & l_1^\top \dfrac{\partial R}{\partial \phi} b_1 \\ l_2^\top & l_2^\top \dfrac{\partial R}{\partial \psi} b_2 & l_2^\top \dfrac{\partial R}{\partial \theta} b_2 & l_2^\top \dfrac{\partial R}{\partial \phi} b_2 \\ \vdots & \vdots & \vdots & \vdots \\ l_6^\top & l_6^\top \dfrac{\partial R}{\partial \psi} b_6 & l_6^\top \dfrac{\partial R}{\partial \theta} b_6 & l_6^\top \dfrac{\partial R}{\partial \phi} b_6 \end{array}\right]$$

the load angular velocity vector as $\boldsymbol{\omega} = (\dot{\psi}, \dot{\theta}, \dot{\phi})^\top$ (Euler angular velocities) and the actuation Jacobian matrix as $J_2 = diag\left(l_1^\top, l_2^\top, \ldots, l_6^\top\right)$.

V. MOTION PLANNING

A problem arises as we assume that the agents do not communicate their decision making (direction of motion), while, on the other hand, the load's configuration obviously complies with the agent's position. Following the notion that all the swarm's members act "together", we solve the motion planning scheme in the configuration space. The second computational complexity problem arises since we apply a probabilistic road map to a multidimensional configuration space. These two problems are mitigated in this Section.

A. THE CONFIGURATION SPACE

The motion planner will be formulated in the configuration space $\mathcal{C}$ to include all $N > 6$ agents. Recall that a configuration c is a vector that fully defines the posture of a mechanism (for a rigorous discussion on this matter see [39]). The configuration space $\mathcal{C}$ is the space of all configurations of the mechanism, which is usually also equipped with a metric (here we shall use the Euclidean metric). In the current case, a configuration $c \in \mathcal{C}$ is defined as an ordered set of the planar coordinates of all the agents, followed by the spatial coordinates of all the nodes of the load (i.e. points on the load that the agents hold). Thus, for N agents, a configuration $c \in \mathcal{C} \subset \mathbb{R}^{5N}$ is:

$$(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N, \boldsymbol{x}_1^p, \ldots, \boldsymbol{x}_N^p) \tag{5}$$

for some $N \geq 6$. This is obviously a redundant description of $\mathcal{C}$. To ensure rigidity of the load, we added 4 diagonal constraints. Note that $3N - 6$ distance constraints should be imposed to enable all rigid transformations of the load. The platform is modeled as a planar N polygon embedded in $\mathbb{R}^3$, so $3N - 6$ constraints will rigidify the platform leaving 6 DOF for the load transformation. Since rod lengths constitute N more constraints, each configuration must hold a set of distance constraints $G = \{g_i(c)\}_{i=1}^{4N-6}$, where $g_i : \mathcal{C} \to \mathbb{R}$ is the Euclidean norm. We added the work-space obstacles $\mathcal{O}_i$ to G, which keeps the system from: (1) agent-agent collision; (2) rod-rod collision; (3) platform-obstacle collision; (4) agent-obstacle collision; (5) rod-obstacle collision. The forward kinematics problem is:

> *Let $A = \{a \in \mathcal{C} | g_i(a) = \ell_i, \forall i \in \{1, 4N-6\}\}$ be the set of all admissible configurations with the desired set of lengths $\{\ell_i\}_1^{4N-6}$. Find a configuration $c \in A$ such that $\Phi(c) = \max\{\Phi(a), \forall a \in A\}$, where $\Phi : \mathcal{C} \to \mathbb{R}$ is the proximity to singularity.*

Here $\ell_i = \|l_i\|$. In this way, we confine the motion in $\mathcal{C}$ to the boundary of the fixed-valued constraint functions. This is also essential for avoiding configurations in which small movements of the agents result in large changes in the platform's orientation (see Subsection VI-A below).

B. RATIONALE

Our approach may be summarized according to the pseudo-code presented in Algorithm 1, which is a variant of the CPRM crawling strategy (see [35]). The CPRM is a probabilistic road map that entails the additional trait that in order to populate an arc between way-points $c^{(m)}$ and $c^{(n)}$, the local planner crawls on the obstacles that obscure the path between the way-points. Here we assume that the agents do not communicate their decision making: **each agent advances on its own without knowing what the other agents will do**. In other words, the motion planning algorithm discussed here differs from the CPRM in that in every time-step, all elements in the direction vector in the configuration space $\mathcal{C}$ are set to be zero except in two dimensions; that is, the two corresponding coordinates of one agent.

In Algorithm 1, we define the objective function for the i^{th} agent as:

$$\zeta_i := v_i^{(m,n)} - (\ell_i - L)\nabla\hat{g}_i$$

where $v_i^{(m,n)}$ is the projection π_i of the vector from $c^{(m)}$ to $c^{(n)}$ onto the i^{th} agent's coordinates. $\nabla\hat{g}_i$ is the i^{th} rod length gradient. L indicates the minimal rod's length. We define the matrix G_{rods} such that its i^{th} column is the i^{th} rod length gradient. η (in Algorithm 1) is the number of agents available to hold the load. In other words, a free agent (i.e. one that does not carry the load) traverses towards its goal configuration while reducing its rod's length. We defined the subspace $\mathcal{C}_{free} \subset \mathcal{C}$ as the set of obstacle-free configurations.

Algorithm 1 Swarm-CPRM. Note That the Individual Agents Are Not Aware of the Actions of the Other Agents

```
Result: find a path between $c_0$ to $c_f$.
initialize a road map $\Gamma := (E, V)$;
$V(\Gamma) \leftarrow$ the initial configuration;
$V(\Gamma) \leftarrow$ the goal configuration;
for all pairs of way-points $m, n$ in $\Gamma$ do
    for each agent $i$ do
        $v_i^{(m,n)} \leftarrow \pi_i(c^{(m)} - c^{(n)})$;
        compute $\zeta_i$;
        if the line from $c^{(m)}$ to $c^{(n)}$ is $\in \mathcal{C}_{free}$ then
            advance in $\zeta_i$ direction;
        else
            crawl on the obstacle minimizing $\|v_i^{(m,n)}\|$,
            if $\eta > 6$ release load if needed;
        end
    end
    $E(\Gamma) \leftarrow E(c^{(m)}, c^{(n)})$ if succeeded;
end
find shortest path on $\Gamma$.
```

Such an approach is preferable when the configuration space is of high dimension, especially when it is also congested with obstacles as in the given case. The number of randomly chosen way-points that are required to sufficiently model the configuration space increases rapidly with the dimension. Applying such a crawling scheme reduces this number substantially. The described scheme increases the chances of finding a connecting arc between a pair of way-points in cases where a straight line would fail. In many cases, this may mean reduction by a few orders of magnitude in the number of road-map arcs and consequently in the required calculation time.

As described in Subsection II-B, Algorithm 1 enables non-synchronized motion, while maneuvering between two way-points $c^{(m)}$ and $c^{(n)}$. In this case, each agent translates at a predefined step-size ε separately (i.e. independent of the actions of other agents) in the direction of the projection $\pi_i(c^{(n)} - c^{(m)})$, onto its own coordinates. Notably, in the case of individual-decision making, the step-size is equal for all agents. This contrasts with a cooperative decision scheme, by which the step-sizes of the agents differ since the projection of $(c^{(n)} - c^{(m)})$ onto each agent's position coordinates is generally different.

C. LOCAL MOTION PLANNER

Assume a configuration c_1, a neighboring configuration c_2, and the set of constraints $g_i \in G$. Motion from c_1 to c_2, restricted to the boundary of G, can be maintained by calculating ∇g_i at a current c, and restricting motion to their joint null space $\mathcal{K}_G$. Explicitly, setting a desired instantaneous direction vector V_d, and projecting it onto the null space $\hat{\mathcal{K}}$ yields:

$$V = \sum_{i\in N+6-M} \hat{\mathcal{K}}_i^\top \cdot V_d \cdot \hat{\mathcal{K}}_i \tag{6}$$

where M is the number of obstacles to crawl on at c, and $\hat{\mathcal{K}}_i$ is the i^{th} base vector of $\mathcal{K}_G$. Thus, at each time-step, advancement is by

$$c(k+1) = c(k) + \epsilon V \tag{7}$$

where $\epsilon \ll 1$ is a predetermined step size in $\mathcal{C}$. Here, V_d can be a vector in $\mathcal{C}$ that is associated with movement of the agents that does not move the load (for the forward kinematics calculation). Clearly, the load would eventually comply with the agent's motion, as V is constrained by G. Alternatively, V_d can be a vector in $\mathcal{C}$ that is associated to motion of the load alone (for the inverse kinematics); also here, the agents should eventually comply. Hence, motion in $\mathcal{C}$ is obtained by applying a configuration space motion planner and repeatedly calculating (Equations 7 and 6) until reaching the desired load configuration.

We expect an infinite solution set; hence, the proximity $\Phi(c)$ to a singular configuration will also be incorporated. To estimate the proximity to singularity at $c \in \mathcal{C}$, the condition number of the associated Jacobian J_1 may be calculated. However, since, as mentioned, a redundant coordinate system is chosen for the configuration space (i.e. that includes the platform's anchoring points), we defined a *configuration space Jacobian* $J_{\mathcal{C}}$, and calculated its condition number $\Phi(c) = cond(J_{\mathcal{C}})$ to estimate the proximity to singularity (for the 6 carrying agents, the following applies):

$$J_{\mathcal{C}} \triangleq \left[G_{agents},\ G_{rods},\ G_{load}\right]_{30\times 30}$$

where G_{agents} columns are simply the standard basis. This implies that the agents may traverse only on the plane. The column vectors of G_{rods} and G_{load} are the 30-dimensional gradients of the distance map $(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2$, which expresses that the i^{th} and the j^{th} nodes are confined by a rigid rod. For regular configurations, the null-space of these vectors is an empty set. However, a configuration $c \in \mathcal{C}$, in which a node (of the load/agents/rods) may alter its position while all others hold theirs, is singular.

Assume N agents are carrying a regular polygon. For each configuration in $\mathcal{C}$, $N - 1$ identical configurations can represent the rotation of the robot in the plane, such that each robot i replaces the location of its neighboring robot. During each step, we calculated these *congruent configurations* and proceeded in the direction towards the closest one (in terms of Euclidean metric in $\mathcal{C}$).

D. GLOBAL MOTION PLANNER

We used the CPRM as a motion planner [35]. Note that this is a rather congested configuration space. Obviously, the motion in $\mathcal{C}_{free}$ is confined to maintain the mechanical constraints, as described above. As in a standard Probabilistic Road Map, we began by generating k random way-point configurations in $\mathcal{C}_{free}$. The direction vector between these configurations is defined as the straight line in $\mathcal{C}$, connecting a pair of way-point configurations.

The case in which $N > 6$ is statically indeterminate. As we are interested in the $N > 6$ case, we changed the algebraic constraints of the fixed rod lengths to a semi-algebraic constraint; that is, the distance ℓ_i between an agent and the corresponding node at the load-platform is in a predetermined range $\ell_i \in [L, L + \Delta L]$ for every $i = 1, \ldots, N$ (here, ΔL was set to be 20% of the rod's minimal length). In other words, we assume the rods are passive prismatic elements which are effective as long as their length is L, but can extend in order to enable the agents to lose grip of the platform in cases where $N > 6$. Thus, in any configuration, at least six rods are under a compression force due to the weight of the load. In such a case, their lengths assume the minimum value. All other $N - 6$ agents are free to change their distance from their corresponding load nodes. Hence, during such a motion, the swarm moves towards the goal configuration while decreasing rod lengths to the minimum value. This is done by subtracting vectors $g_i \in G_{rods}$ from the direction vector, towards the goal configuration. Subsequently, the vector $-g_i$ reduces the i-th rod length.

VI. SIMULATION RESULTS

A. SWARM's DISTRIBUTION STUDY

To demonstrate the efficiency of the proposed swarm's distribution algorithm, a simulated experiment was implemented. The scene workspace of size 100 × 100 [cm] with 10 agents traverses in random directions. Each agent is equipped with a sensor that measures the relative angle to all other agents, with sensor-noise variance $\sigma_\theta = 5°$. The process-noise corresponds to the error in the direction and the length of the step size in each time-step, simulated as a normal distributed vector with variance of 1[mm], while the step size is $\|\mathbf{d}_i\| = 10$[mm]. In the initialization stage, the locations of the agents, and of the estimated states, were randomly chosen over the entire space. The results for $N = 10$ agents are depicted in Figure 2. The graph presents the statistics of 50 simulation runs for 1, 500 time-steps. Notably, the estimation error converges properly. A set of tests shows that this robust property does not depend on the initial conditions.

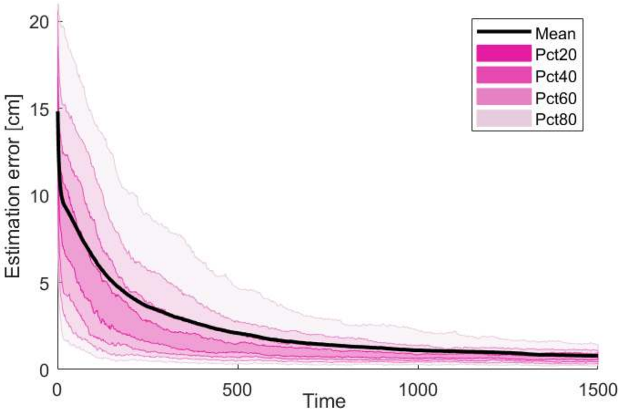

FIGURE 2. A *Fanplot* of the algorithm performances for $N = 10$ over time-steps. The middle solid line indicates the mean estimated error, which after 1000 steps (∼ 30 seconds for a standard camera) converges to 8mm.

The estimation error of the agents' position converges rapidly to 8mm. We then performed a set of 10, 000 simulation experiments to examine the applicability of such accuracies to a load carrying swarm. In each experiment, a random configuration c was chosen. Then, the estimation errors $\{\epsilon\}_i$ were added to the agents' positions.

We defined the heading error angle as the angle between the platform's normal with the estimation errors included, and the platform's normal without the estimation errors. Similarly, we defined the load's position error. Note that singular configurations may result in large heading error angles. Recall that the condition number constraint $\Phi(c) < \Phi_0$, imposed on motion planning, was determined exactly to avoid such a configuration (Φ_0 was chosen by generating random configurations and averaging the condition number of the chosen ones). The Jacobian condition number is a useful tool for avoiding mechanism singularities. These may result in instabilities of the whole mechanism in some cases or require the agents to apply forces beyond their ability in others. The underlying logic behind the condition number is that it relates to the mechanism's sensitivity to erroneous input actuation. So, in order to overcome the kinematic instability, it suffices to set an arbitrary bound for the condition number which we set to 2000. This was found satisfactory during our experimentation.

Generalized force analysis may be provided also by the Jacobian formalism. Nevertheless, note that since agents are free to move on the plane, our mathematic formalism is redundant (think of an agent traversing a base of a cone around its anchoring point). The Jacobian formalism should, therefore, be altered such that the radial distance to the cone center would constitute the inputs. Yet, as we focus our attention here to the kinematic problem we make do with a simple force analysis where the load which the platform experiences is of its own weight.

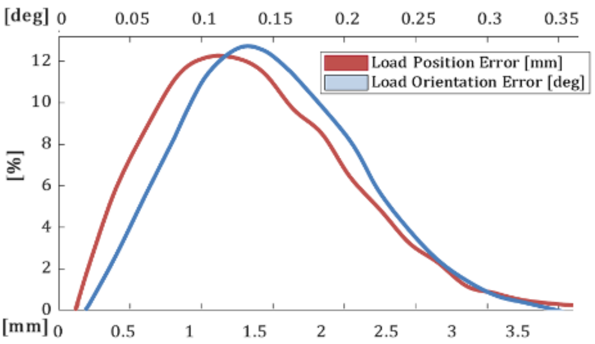

FIGURE 3. The histogram of the load's position and orientation errors (smoothed) due to the estimation errors of the position of the agents depicted in Figure 2.

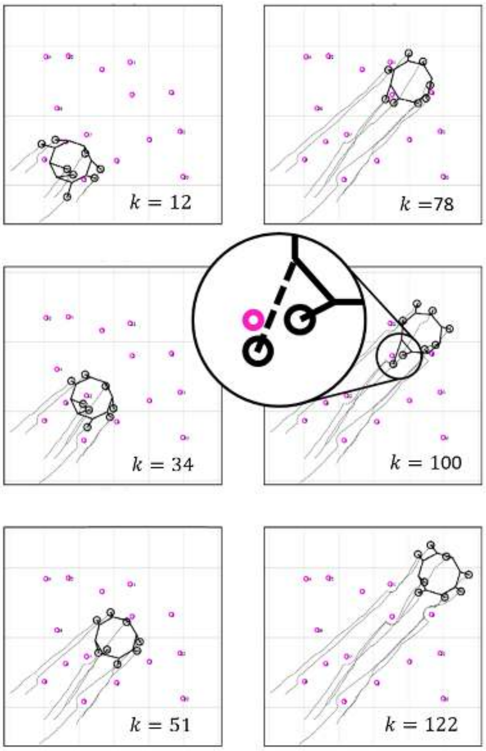

FIGURE 4. Eight agents carrying a load with 15 ground-obstacles. Throughout the motion, exactly six agents hold the load while others follow the group. An agent may withdraw (dashed line) from the load to avoid an obstacle (see $k = 100$ time-step) while another agent attaches to it.

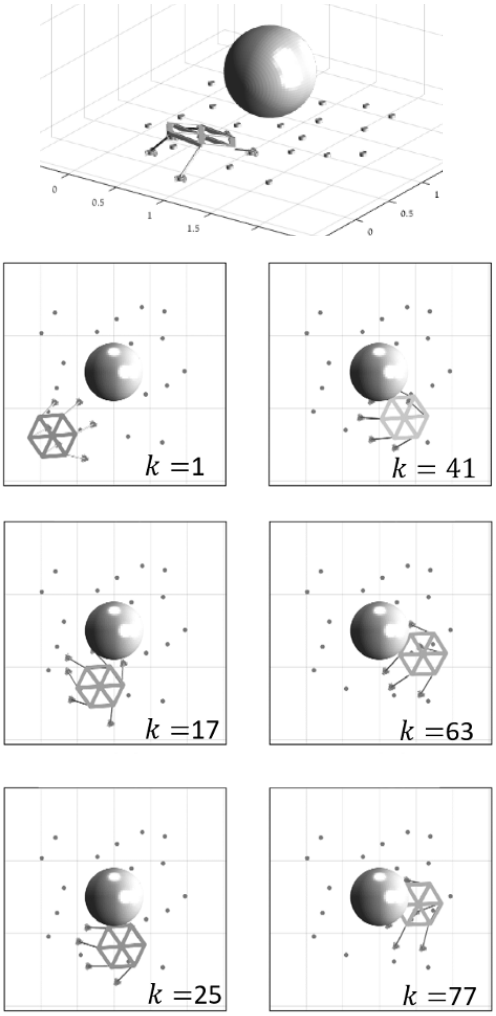

FIGURE 5. Six agents are carrying a load with 20 ground-obstacles and a hovering sphere obstacle. A solution was attained with only 4 way-points. Top: Isometric view. Bottom: Overhead view.

The resulting heading angle error histogram, excluding the configuration that violates the condition number constraint, is presented in Figure 3. These are χ^2 distributed with a mean orientation error of $\mu \sim 0.14^\circ$. The resulting position error histogram, excluding the configuration that violates the condition number constraint, is presented in Figure 3. These are also χ^2, distributed with mean position error of $\mu \sim 1.4mm$.

Thus, to evade platform obstacles, these obstacles should be "inflated" accordingly by means of the *Minkowski Sum*. The position error was calculated as the error of the load's CoG. So, to avoid the load's obstacle one should "inflate" the obstacle twofold - by the position error value and the load radius multiplied with the orientation error as well as by the load geometry at all possible orientations.

B. EVADING PLANAR OBSTACLES WHILE MAINTAINING THE LOAD's HORIZONTAL ORIENTATION

We simulated eight agents; at each time-step, six were actively carrying a load-platform, while others were moving freely and were available for a "switch". The frame sequence in Figure 4 presents the swarm in a work space having 15 ground obstacles.

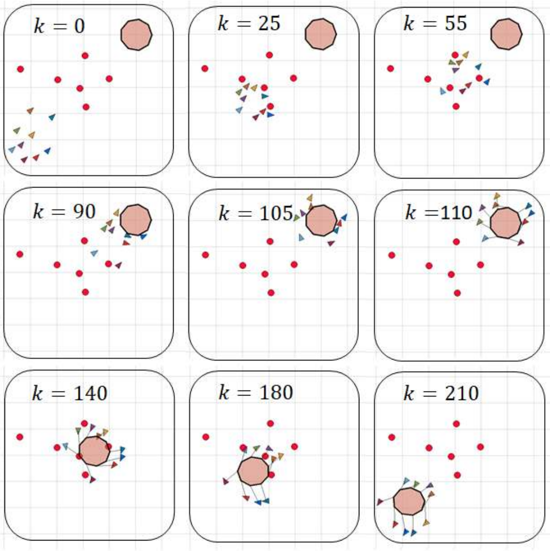

FIGURE 6. A swarm that assumes the task of picking up a load and transferring it through a cluttered work-space.

Here we constrained the platform to horizontal orientation throughout the motion. Minimal rod lengths were set as 0.3 of a unit length. Throughout the scenario, agents that actively carry the load are those whose prismatic-rods attain a minimal length L. Other agents, (i.e. those that do not carry the load at that given moment) are simulated as being connected to the load via prismatic rods that were not at their minimal length. The algorithm proceeds such that all agents pursue the target configuration while shortening their rods to the minimal length. To be more precise, we simplified this statically undetermined problem by setting the rods of $N - 6$ agents to be of length $L + \varepsilon$. This enables any $N - 6$ agent to release the load at will.

A "switch" between agents may occur when an agent meets an obstacle and is unable to proceed at the pace of the other agents. The rod related to this agent would thus lengthen, transferring the carrying task to the other agents. This may repeat until exactly six agents are left to carry the load. Figure 4 in the $k = 100$ time-step depicts such a scenario.

Note that since the load is carried by the agents via rods with spherical joints at their ends, an agent may move in a circle that maintains its distance from the load, such that the load stays still. Such a maneuver is carried out throughout the motion and is notable in the (f) frame of Figure 4.

C. EVADING SPATIAL OBSTACLES

We simulated six agents carrying the load. For convenience, other agents wandering around freely are not depicted. The frame sequence in Figure 5 presents a swarm that assumes the task of evading 20 ground obstacles and a large $r = 0.4$ spherical obstacle, with its center hovering at $z = 0.55$ of a unit length. Motion is achieved by applying Algorithm 1. Here, we allow the platform to tilt freely up to 12°. To overcome the dense obstacle space, the agents may change the platform orientation as in Figure 5. Figure 6 presents a swarm that assumes the task of picking up a load and transferring it through a cluttered work-space.

VII. CONCLUSION

We introduced a load carrying swarm that is capable of avoiding spatial and planar obstacles. Inspired by insects that have low communication capabilities, we showed how a robotic swarm may transport a load while evading obstacles, and while communicating only their position estimations.

Our discussion was limited to scenarios in which $\geq$ 6 agents participate in the load carrying task. Alternative carrying models such as caging are efficient, according to a purely kinematic perspective. However, the kinematic model applied here concurs better with the entomological case in which insects reorient a load while avoiding obstacles (i.e. the *piano movers problem*).

Thus, knowing the next desired way-point c_m in the road-map graph, each agent moves towards its own (projected) goal configuration in c_m, while avoiding obstacles in its way and while controlling the load as much as it can. To realize this, individual agents receive, by turn, the opportunity to advance. Apparently, the order in which the agents act should be considered (c.f. [40]). Nevertheless, our experiments indicate that randomly setting this order suffices, such that one of the agents may communicate and randomly set the motion's order. Future work will include implementation of the algorithms introduced in this paper to a real robotic swarm.

REFERENCES

[1] S. Berman, Q. Lindsey, M. S. Sakar, V. Kumar, and S. Pratt, "Study of group food retrieval by ants as a model for multi-robot collective transport strategies," *Robotics: Sci. Syst. VI*, p. 259, 2011.

[2] A. Gelblum, I. Pinkoviezky, E. Fonio, A. Ghosh, N. Gov, and O. Feinerman, "Ant groups optimally amplify the effect of transiently informed individuals," *Nature Commun.*, vol. 6, no. 1, p. 7729, Nov. 2015.

[3] T. J. Czaczkes and F. L. Ratnieks, "Cooperative transport in ants (hymenoptera: Formicidae) and elsewhere," *Myrmecol. News*, vol. 18, pp. 1–11, 2013.

[4] K. Eugene, S. Nir, and B.-G. Irad, *Autonomous Mobile Robots and Multi-Robot Systems: Motion-Planning, Communication, and Swarming*, 1st ed. Hoboken, NJ, USA: Wiley, 2019.

[5] M. D. Kennedy, L. Guerrero, and V. Kumar, "Decentralized algorithm for force distribution with applications to cooperative transport," in *Proc. Int. Design Eng. Tech. Conf. Comput. Inf. Eng. Conf. (ASME)*, 2015, Art. no. V05CT08A013.

[6] G. Grisetti, R. Kummerle, C. Stachniss, and W. Burgard, "A tutorial on graph-based SLAM," *IEEE Intell. Transp. Syst. Mag.*, vol. 2, no. 4, pp. 31–43, Winter 2010.

[7] B. B. Moshe, N. Shvalb, J. Baadani, I. Nagar, and H. Levy, "Indoor positioning and navigation for micro uav drones—Work in progress," in *Proc. IEEE 27th Conv. Electr. Electron. Eng. (ISR)*, Nov. 2012, pp. 1–5.

[8] M. S. Couceiro, P. A. Vargas, R. P. Rocha, and N. M. F. Ferreira, "Benchmark of swarm robotics distributed techniques in a search task," *Robot. Auto. Syst.*, vol. 62, no. 2, pp. 200–213, Feb. 2014.

[9] H. Farivarnejad, S. Wilson, and S. Berman, "Decentralized sliding mode control for autonomous collective transport by multi-robot systems," in *Proc. IEEE 55th Conf. Decis. Control (CDC)*, Dec. 2016, pp. 1826–1833.

[10] K. M. Lynch, "Nonprehensile robotic manipulation: Controllability planning," Carnegie Mellon University Pittsburgh, Pittsburgh, PA, USA, Tech. Rep. CMU-RI-TR-96-05, 1996.

[11] E. S. Ardakani, H. Ebel, and P. Eberhard, "Transporting an elastic plate using a group of swarm mobile robots," in *Proc. IEEE Int. Conf. Adv. Intell. Mechatronics (AIM)*, Jul. 2017, pp. 1393–1398.

[12] Z. Wang, G. Yang, X. Su, and M. Schwager, "Ouijabots: Omnidirectional robots for cooperative object transport with rotation control using no communication," in *Distributed Autonomous Robotic Systems*. Cham, Switzerland: Springer, 2018, pp. 117–131.

[13] D. Rus, B. Donald, and J. Jennings, "Moving furniture with teams of autonomous robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. Hum. Robot Interact. Cooperat. Robots*, vol. 1, Aug. 1995, pp. 235–242.

[14] G. A. S. Pereira, M. F. M. Campos, and V. Kumar, "Decentralized algorithms for multi-robot manipulation via caging," *Int. J. Robot. Res.*, vol. 23, nos. 7–8, pp. 783–795, Aug. 2004.

[15] J. Fink, M. A. Hsieh, and V. Kumar, "Multi-robot manipulation via caging in environments with obstacles," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2008, pp. 1471–1476.

[16] S. T. Kalat, S. G. Faal, and C. D. Onal, "A decentralized, communication-free force distribution method with application to collective object manipulation," *J. Dyn. Syst., Meas., Control*, vol. 140, no. 9, Sep. 2018, Art. no. 091012.

[17] G. Habibi, Z. Kingston, W. Xie, M. Jellins, and J. McLurkin, "Distributed centroid estimation and motion controllers for collective transport by multi-robot systems," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2015, pp. 1282–1288.

[18] J. Chen, M. Gauci, W. Li, A. Kolling, and R. Gros, "Occlusion-based cooperative transport with a swarm of miniature mobile robots," *IEEE Trans. Robot.*, vol. 31, no. 2, pp. 307–321, Apr. 2015.

[19] D. J. Stilwell and J. S. Bay, "Toward the development of a material transport system using swarms of ant-like robots," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 1993, pp. 766–771.

[20] R. Gross, F. Mondada, and M. Dorigo, "Transport of an object by six pre-attached robots interacting via physical links," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2006, pp. 1317–1323.

[21] R. Groß and M. Dorigo, "Group transport of an object to a target that only some group members may sense," in *Proc. Int. Conf. Parallel Problem Solving Nature*. Cham, Switzerland: Springer, 2004, pp. 852–861.

[22] J.-P. Merlet, *Parallel Robots*, vol. 128. Cham, Switzerland: Springer, 2006.

[23] G. Gogu, *Structural Synthesis of Parallel Robots*, vol. 866. Cham, Switzerland: Springer, 2008.

[24] R. Ben-Horin, M. Shoham, and S. Djerassi, "Kinematics, dynamics and construction of a planarly actuated parallel robot," *Robot. Comput.-Integr. Manuf.*, vol. 14, no. 2, pp. 163–172, Apr. 1998.

[25] N. Shvalb, M. Shoham, G. Liu, and J. C. Trinkle, "Motion planning for a class of planar closed-chain manipulators," *Int. J. Robot. Res.*, vol. 26, no. 5, pp. 457–473, May 2007.

[26] J. Shim, D.-S. Kwon, and H. S. Cho, "Kinematic analysis and design of a six DOF 3prps in-parallel manipulator," *Robotica*, vol. 17, no. 3, pp. 269–281, 1999.

[27] G. Yang, I.-M. Chen, W. Chen, and W. Lin, "Kinematic design of a six-DOF parallel-kinematics machine with decoupled-motion architecture," *IEEE Trans. Robot.*, vol. 20, no. 5, pp. 876–884, Oct. 2004.

[28] F. Tahmasebi and L. Tsai, "Workspace and singularity analysis of a novel six-DOF parallel minimanipulator," Inst. Syst. Res., College Park, MD, USA, Tech. Rep. T.R. 93-55, 1993.

[29] E. Pernette, S. Henein, I. Magnani, and R. Clavel, "Design of parallel robots in microrobotics," *Robotica*, vol. 15, no. 4, pp. 417–420, Jul. 1997.

[30] Y. K. Byun and H. S. Cho, "Analysis of a novel 6-DOF, 3-PPSP parallel manipulator," *Int. J. Robot. Res.*, vol. 16, no. 6, pp. 859–872, Dec. 1997.

[31] J. Kim, F. C. Park, S. J. Ryu, J. Kim, J. C. Hwang, C. Park, and C. C. Iurascu, "Design and analysis of a redundantly actuated parallel mechanism for rapid machining," *IEEE Trans. Robot. Autom.*, vol. 17, no. 4, pp. 423–434, Aug. 2001.

[32] M. Basiri, F. Schill, D. Floreano, and P. Lima, "Audio-based relative positioning system for multiple micro air vehicle systems," Inst. Syst. Robot., Instituto Superior Tecnico, Lisboa, Portugal, Tech. Rep., 2013.

[33] O. Soysal and E. ahin, "A macroscopic model for self-organized aggregation in swarm robotic systems," in *Proc. Int. Workshop Swarm Robot.* Cham, Switzerland: Springer, 2006, pp. 27–42.

[34] S. Shoval and M. Shoham, "Sensory redundant parallel mobile mechanism," in *Proc. ICRA. IEEE Int. Conf. Robot. Autom.*, vol. 3, May 2001, pp. 2273–2278.

[35] N. Shvalb, B. B. Moshe, and O. Medina, "A real-time motion planning algorithm for a hyper-redundant set of mechanisms," *Robotica*, vol. 31, no. 8, pp. 1327–1335, Dec. 2013.

[36] S. Hacohen, S. Shoval, and N. Shvalb, "Multi agents' multi targets mission under uncertainty using probability navigation function," in *Proc. 13th IEEE Int. Conf. Control Autom. (ICCA)*, Jul. 2017, pp. 845–850.

[37] 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.

[38] A. Gelman and D. B. Rubin, "Inference from iterative simulation using multiple sequences," *Stat. Sci.*, vol. 7, no. 4, pp. 457–472, Nov. 1992.

[39] N. Shvalb, M. Shoham, H. Bamberger, and D. Blanc, "Topological and kinematic singularities for a class of parallel mechanisms," *Math. Problems Eng.*, vol. 2009, pp. 1–12, 2009.

[40] A. Y. Carmi and D. Moskovich, "Tangle machines," 2014, *arXiv:1404.2862*. [Online]. Available: http://arxiv.org/abs/1404.2862

ODED MEDINA received the Ph.D. degree. He is currently a Lecturer of mechanical engineering with Ariel University. He deals with motion planning problems and general computational problems in robotics, like configuration space representation and minimal actuation problems of redundant robots. Amongst his early research projects are a real-time motion planner for hyper redundant serial robot aimed for pole climbing tasks. He also investigates the motion of hyper-redundant parallel mechanisms, bio-mimicking of thin muscles (such as fins or the human diaphragm) which is due to the flexibility of the muscle which requires large number of actuators. Most projects also included designing and fabrication of actual mechanisms. His main interest includes robotics.

SHLOMI HACOHEN received the Ph.D. degree. He is currently a Lecturer of mechanical engineering with the Department of Industrial Engineering and Management, Ariel University. His expertise is in the field of robotics and control. Specifically, his expertise is related to decision making under uncertainty conditions applied for robot's motion planning and for pedestrian behavior modeling. He is engaged in the research in the field of multiagents control under uncertain conditions. He also deals with classical control theory and non-linear estimation methods which may be applied to indoor positioning problems. He is also the Co-Founder of a commercial company IndoorGo Ltd.

NIR SHVALB received the Ph.D. degree. He is currently an Associate Professor with the Faculty of Engineering, Ariel University, Israel. He heads the Robotics Research Laboratory. He is also the Founder of several startup companies, including MeMic medical robotics, Innoging medical imaging, and IndoorGo. He wrote a robotics textbook, published more than 60 scientific articles and patents. His main interests include medical robotics global path planning and theoretical foundations of robotics.