IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 16 January 2023, accepted 31 January 2023, date of publication 6 February 2023, date of current version 23 February 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3243043

RESEARCH ARTICLE

Decentralized Motion Planning for Load Carrying and Manipulating by a Robotic Pack

HAREL COHEN, SHLOMI HACOHEN, NIR SHVALB, AND ODED MEDINA
Department of Mechanical Engineering, Ariel University, Ariel 4077625, Israel

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

ABSTRACT In many cases, a pack of robots holds an advantage over a single robot such when an oversized or over-weighted load is to be carried. In such cases, a single robot will not do. Nevertheless, this may not be an easy task for a pack of robots as well, especially when the load needs to be lifted off the ground making the cooperative task less tolerant of errors. The limited research on such a load can be attributed to the mechanical complexity of the problem. Notably, previous studies have not considered the spatial, decentralized, communication-free scenario. We, therefore, consider a robotic pack of six agents that assumes the task of spatially moving a load through a cluttered space. As it transports the load, the pack carefully avoids planar obstacles while maintaining its orientation. To do so, we model the whole system as a six Prismatic-Prismatic-Spherical-Spherical (6-PPSS) redundant mobile platform, having twelve degrees of freedom. This paper focuses on a decentralized control scheme where no mutual communication is needed. Each agent calculates its ego movements according to the height of its corresponding load-node; the surrounding obstacles, and; the goal's relative position. To avoid numerical errors appearing in the vicinity of singular configurations, we calculate the platform's forward kinematics in the model's full configuration space. We then show how this rationale can be further extended to formulate a distributed control scheme for the motion planner. We demonstrate our algorithms in several simulated scenarios and in a set of real-world experiments using specially designed omnidirectional robot agents. We test the ability of the pack to maintain the load's orientation just by measuring the load's height at the holding node of each agent. Lastly, we measured the time required for the pack to assume a desired load orientation. Results indicated that even in the presence of a 10-degree tilt error, the load was able to be restabilized within a maximum of 15 seconds in simulated conditions and 20 seconds in real-life experiments.

INDEX TERMS Swarm robotics, parallel robots, motion planning, decentralized control.

I. INTRODUCTION

The ability to cooperatively transport objects is common for several arthropods [3] where an individual insect holds the object firmly by applying generalized forces or alternatively, push/pull the object [7], [13]. Typically, this involves limited information sharing with regard to the exerted generalized forces each experience [7], [13].

In the Robotics literature (e.g., [8], [9]§, [21], [23]), a *swarm* refers to multiple robotic agents with a cooperative mission where usually a decentralized controller with minimal communication capabilities is applied. Such cooperative missions may include mapping, loads manipulating, and searching [6], [9], [10], [15], [26]. In general, motion in the plane requires at least two degrees of freedom. For wheeled agents, these can be given using: Omniwheels that enable motion to any direction regardless of the agent's orientation; or Differential (or skid) steering where opposite wheels (or tracks) can be independently actuated.

A. RELATED WORK

Early research mainly considered centralized control for the load carrying task. For example, in [33], where the researchers considered a swarm of communicating agents

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

that manipulate planar loads via cooperative pushing. Since then, most work has been focused on decentralized control schemes. These can be further categorized according to predefined information-sharing assumptions:

1) DECENTRALIZED WITH INFORMATION-SHARING

Carrying a load may rely on friction. For example, in [1] the researchers considered a swarm of omnidirectional mobile robots transporting a planar elastic load by means of Coulomb friction. The motion planner introduced was based on an artificial potential field calculated assuming the agents share their locations. Carrying arbitrarily shaped loads were considered in [18]. In their work, the agents first surround the object, and by sharing the locations of all agents, each agent estimates the object's centroid. Then, by choosing their motion with respect to that centroid, the agents manipulate the load. The researchers in [42] proposed a decentralized control scheme for a multi-robot system using a deep Q-network controller for each agent. This was used to transport an oversized object on the plane. The researchers demonstrated a two-robots scenario where an oversized rod was carried through a doorway. The information regarding the rod's orientation, as well as the fellow agent position, was assumed to be known. In [28] the authors of this paper recently introduced a spatial-load-carrying swarm where a decentralized control scheme with information sharing was applied. The swarm was given the task of maneuvering through a planar-cluttered space with additional obstacles in the load's path.

2) DECENTRALIZED WITH NO INFORMATION-SHARING

Pereira et al. [31] considered an "object closure" strategy for translating and orienting planar objects. They assumed that only local information is available to the robots (relative position and orientation of their neighbors) and that each robot knows the object's shape. Similarly, manipulating planar loads was considered in [12] where the researchers constructed an artificial decentralized navigation function to hold the object's boundary and traverse it towards the goal position. Researchers in [16], [17] addressed a cooperative planar transport mission where the group members lack knowledge about the goal position. The agents exploit the physical interactions with other group members to imitate a passive caster. Kalat et al. [22] proposed a decentralized control algorithm for a planar transport problem where each robot coordinates its actions with a virtual teammate located at the robot's mirror position with respect to the payload's center. A communication-free manipulation was also proposed in [5] where each agent is given the simple mission of pushing an object on the plane at positions where a direct line of sight to the goal is occluded by the object. Note that all cases above refer to the load-carrying task as a planar one, meaning, the load can be manipulated in three degrees of freedom. In such cases, pushing or enclosing an object in the plane is insensitive to discontinuities, that is, instances where one agent or more stop participating in the task of carrying the load.

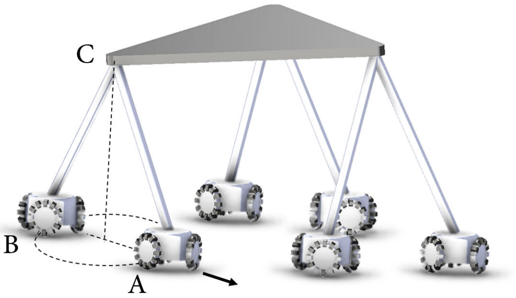


FIGURE 1. An illustration of six omnidirectional agents carrying a triangular load. The equivalent kinematic model is a 6-PPSS redundant mobile platform. As A and B follow the dashed circle and all other agents will stay still, point C will stay in place. The motion of A and B together radially into the circle will elevate point C.

3) SPATIAL TASKS

The problem at hand is to translate a *spatial* load and maintain its orientation while avoiding planar obstacles. Unlike planar load-carrying tasks, the spatial case is more involved and requires a complex robot-load formation referred to in the literature as a 'Parallel Manipulator'. Manipulating a spatial load requires six planar agents at a minimum as presented in Figure 1. One can therefore model the entire agents-load system as a 6-PPSS parallel manipulator which has been extensively studied (e.g., [14], [29]). Parallel mechanisms are characterized by a limited work-space volume (the subspace of the intersection of the work-spaces of its limbs [36]). Thus, various structural modifications have been implemented to increase the work-space volume. For example, a 3-PRPS parallel platform with linear base-point-actuators [34], and the 3-RPRS introduced in [41]. In our case, this deficiency is overcome by the planar translation capabilities of the agents such as been investigated in [2] and similarly, as with the 3-PPSR platforms which were introduced by Tahmasebi and Tsai [39] and by Pernette et al. [32] (c.f., [4], [25]).

B. CONTRIBUTION

As presented, carrying a load by a pack of robotic agents is of interest to the research community. Obviously, six or more agents may be involved in the hauling assignment, however, six will suffice to properly control a spatial load in a non-prehensile situation.

In this paper, we shall consider only six agents who are actively bearing the load. The load's orientation is controlled as the agents move over the plane, transporting it to the goal position. While information sharing can greatly facilitate the task, the risk of communication failure can lead to problems, such as the payload falling. Thus, our proposed approach involves a motion planning scheme that does not require communication. Under this assumption, with the agents unaware of the positions of the others, the ability of the individual agents to solve the forward kinematics and act accordingly is out of hand. Figure 1 illustrates this: agents A and B hold the load at point C. As A and B follow the dashed circle and all other agents will stay still, point C will stay in place. If all agents can coordinate their actions they can elevate point C or lower it at will. For example, all agents but A and B stay in place while A and B should move together radially into the circle or outwards, respectively. Nevertheless, the agents do not communicate and agent A does not know what will agent B does nor what the others do. Under these circumstances, the proposed scheme relies on a heuristic algorithm designed via a large set of simulation experiments calculated in the configuration space. Here, we apply a Potential Field method [24] for each of the agents as a motion planner due to its computational efficiency and ease of implementation. The scalar potential field for each agent is a weighted sum of the distance to the target, the distance to nearby obstacles, and the holding point height. These sum up to solving the pack's heuristic local inverse kinematics while avoiding obstacles and pursuing a target position.

Since the agents are unaware of their mutual positions and cannot calculate the kinematics, avoiding singularities is out of hand. Instead, we concentrate on maintaining the payload in a predefined orientation while satisfying some kinematic thumb rules. Nevertheless, singularities are bound to appear so to avoid numerical errors in the vicinity of such singular configurations we present a robust, error-free method for the time propagation in the mechanism's configuration space.

C. PAPER ORGANIZATION

This paper is organized as follows: Section II introduces the model and the main assumptions (i.e. the sensor capabilities and the mechanical structure). Section III introduces the associated kinematic model used to manipulate the load in the spatial space. We show its vulnerability and present a method that overcomes the Jacobian-ill-condition errors by calculating the kinematics in the configuration space. Section IV introduces a potential field motion planning scheme that we applied to maintain the load's orientation while evading planar obstacles. The simulation study, which served as the basis for our proposed scheme, is detailed in Section V. Section VI presents the set of simulated experiments demonstrating our approach and algorithms in congested workspaces and real-world experiments using robotic agents specially designed and fabricated in our laboratory. Section VII concludes the paper.

II. MODEL AND ASSUMPTIONS

The use of swarms to carry heavy and oversized loads offers significant advantages over the use of a single agent. In addition to the load being distributed among multiple agents, the need for each agent to exert high torques to manipulate the load in the spatial space is eliminated, making the task possible in a mechanical sense. From a kinematic perspective, each agent within the swarm can be conceptualized as holding the load via a rod that is connected at both of its ends to spherical joints (see Figure 2). It is clear that utilizing less than six agents will necessitate the incorporation of torques, which is beyond the scope of this paper. On the other hand, incorporating more than six agents will result in a non-determinate problem, thus in this paper, we consider six agents for the task.

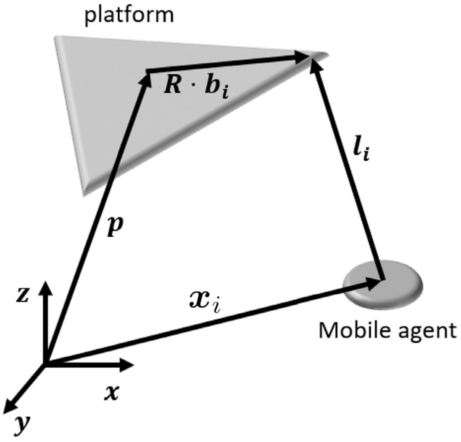


FIGURE 2. The kinematic model of a single agent (out of six) carrying a load. A planar mobile agent located at x_i holds a rod via a passive sphere joint connected by a second passive sphere joint to the platform at $x_i + l_i$.

We now detail the assumptions made regarding the agents' sensing capabilities and their communication abilities.

A. SENSORS

We assume that each agent is equipped with a compass and a sensory system that measures the distance to nearby obstacles (other agents are considered obstacles as well). Each agent holds the platform by a rod having two spherical joints at its ends. The agent's rod relative orientation is assumed to be known; thus at each time-step an agent is aware of the height of its rod's endpoint on the platform. Measurement noise is out of the scope of this paper.

B. COMMUNICATION

In the literature, a swarm commonly refers to simple physical agents with minimal communication needs (or having no communication needs as in [19]).

In our precursory research [28] we assumed 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). However, in the current research, each time step an agent proceeds towards its own goal while no information regarding the other agents' position is available.

C. PROCESSING

Our method is decentralized and communication free. Therefore, it is assumed that all calculations and decisions are made individually in each agent's local CPU. This includes sensory processing and motion control calculation.

TABLE 1. A comparison of the state-of-the-art research. The first row lists the current research. Note that the first three rows are the only attempts to manipulate a spatial load, whereas the current paper is the only one that provides a decentralized, communication-free solution.

	Load DOF	# Agents	Grasp type	Sensors	Obstacles avoidance	Communication	Decentralized control
Herein	6	$N = 6$	Spatial forces	Nearby obstacles & angle to load	PF	✗	✓
[28]	6	$N \geq 6$	Spatial forces	Nearby obstacles & load's state	CPRM	✓	✓
[35]	6	$N = 3$	Spatial forces & torques	Full knowledge	✗	✓	✗
[1]	3	$N \geq 3$	Planar Friction	Nearby obstacles	PF	✓	✓
[40]	3	$N \geq 3$	Planar Friction	Force, torques & load's velocities	✗	✗	✓
[31]	3	$N \geq 4$	Enclosing	Nearby obstacles location and orientation	✗	✗	✓
[12]	2	$N \geq 3$	Enclosing	Nearby obstacles	PF	✓	✓
[18]	3	$N \geq 3$ and $N \gg 6$,	Enclosing	Nearby obstacles	✗	✓	✓
[5]	2	$N \geq 3$	Push/pull forces	Target position	✗	✗	✓
[38]	2	$N \geq 3$	Push/pull forces	Force sensor to the load	✗	✗	✓
[17]	2	$N = 6$	Spatial forces	Full local knowledge	✗	✗	✓

III. KINEMATICS

We use the kinematic model provided in [28]: Figure 2 depicts the i^{th} agent position vector x_i, the load center of gravity position $\boldsymbol{p}$, the i^{th} rod vector $\boldsymbol{l}_i$ and the i^{th} load's node $\boldsymbol{b}_i$, which are given in global coordinates, with vector $\boldsymbol{b}_i$ written in the load's coordinate system. The 3×3 load's rotation matrix R sets:

$$x_i + \boldsymbol{l}_i = \boldsymbol{p} + \mathrm{R}\boldsymbol{b}_i$$

for all $i = 1, 2, \ldots, 6$. Differentiating with respect to time and noting that $|\boldsymbol{l}_i| = \text{const}$, yields:

$$\boldsymbol{l}_i^\top \dot{x}_i = \boldsymbol{l}_i^\top \dot{\boldsymbol{p}} + \boldsymbol{l}_i^\top \dot{\mathrm{R}}\boldsymbol{b}_i$$

which can be written in the matrix form:

$$J_1 \begin{pmatrix} \dot{\boldsymbol{p}} \\ \boldsymbol{\omega} \end{pmatrix} = J_2 \dot{x} \tag{1}$$

where we define the platform's Jacobian matrix J_1 as

$$\left[\begin{array}{c|ccc} \boldsymbol{l}_1^\top & \boldsymbol{l}_1^\top \dfrac{\partial R}{\partial \psi}\boldsymbol{b}_1 & \boldsymbol{l}_1^\top \dfrac{\partial R}{\partial \theta}\boldsymbol{b}_1 & \boldsymbol{l}_1^\top \dfrac{\partial R}{\partial \phi}\boldsymbol{b}_1 \\ \boldsymbol{l}_2^\top & \boldsymbol{l}_2^\top \dfrac{\partial R}{\partial \psi}\boldsymbol{b}_2 & \boldsymbol{l}_2^\top \dfrac{\partial R}{\partial \theta}\boldsymbol{b}_2 & \boldsymbol{l}_2^\top \dfrac{\partial R}{\partial \phi}\boldsymbol{b}_2 \\ \vdots & \vdots & \vdots & \vdots \\ \boldsymbol{l}_6^\top & \boldsymbol{l}_6^\top \dfrac{\partial R}{\partial \psi}\boldsymbol{b}_6 & \boldsymbol{l}_6^\top \dfrac{\partial R}{\partial \theta}\boldsymbol{b}_6 & \boldsymbol{l}_6^\top \dfrac{\partial R}{\partial \phi}\boldsymbol{b}_6 \end{array}\right] \tag{2}$$

the load's angular velocity vector as $\boldsymbol{\omega} = (\dot{\psi}, \dot{\theta}, \dot{\phi})^\top$ (Euler angular velocities) and the actuation (agents) Jacobian matrix as $J_2 = diag\left(\boldsymbol{l}_1^\top, \boldsymbol{l}_2^\top, \ldots, \boldsymbol{l}_6^\top\right)$.

Note that if more than six robots are involved in carrying the load it will result in a statically indeterminate system. Nevertheless, load carrying is also possible with multiple agents, for example, by adding a passive prismatic joint on each rod. In such a case, six rods will always attain their minimal length due to the platform's weight, while the rest of the rods will comply by extending to enable the agents holding them to ''lose grip'' of the platform momentarily (see [28] for further information). This may be useful for avoiding obstacles.

A. SINGULAR CONFIGURATIONS

The singularity of parallel platforms can be divided into two types, depending on the two Jacobian matrices J_1 and J_2 (c.f. [37]). In general, in the case where J_1 is singular (or close to singularity) activating the platform's actuators will result in no motion of the platform. In the case where J_2 is singular (or close to singularity), the platform may move even if no actuation is introduced (see also [30]).

Note that in the case of load carrying by a swarm having decentralized control, it is possible to avoid some simple singular configurations (e.g., avoiding configuration where the holding rods are vertical). Of course, some other ill-conditioned configurations may come about and result in bad performances. Thus, training the swarm to carry the load using simulations will not suffice since significant numerical errors are expected.

Figure 3 depicts the swarm in a singular configuration where all agents independently decide their next step. In this configuration, however, the platform's velocities vector cannot be accurately calculated by Equation 1 because the corresponding jacobian is ill-conditioned. But note that in order to implement any kind of a heuristic motion planner these inaccuracies should be carefully considered (or avoided altogether).

B. TIME PROPAGATION IN THE CONFIGURATION SPACE

To overcome this, we consider the entire mechanism's configuration rather than focusing on the workspace (as in Eq. 1). The configuration space $\mathcal{C}$ is a continuous set of vectors $c \in \mathcal{C}$ equipped with a Euclidean metric. Each vector c represents a posture of the mechanism. A configuration c is defined as an ordered set of the planar coordinates of all the agents concatenated by the spatial coordinates of all the platform's vertices.

Here we assume that the agents hold the load at three contact points (equivalent to a ternary link). Moreover, each contact point is held by a pair of agents. Thus, for 6 agents, a configuration $c \in \mathcal{C} \subset \mathbb{R}^{21}$ is:

$$(x_1, \ldots, x_6, x_1^p, \ldots, x_3^p) \tag{3}$$

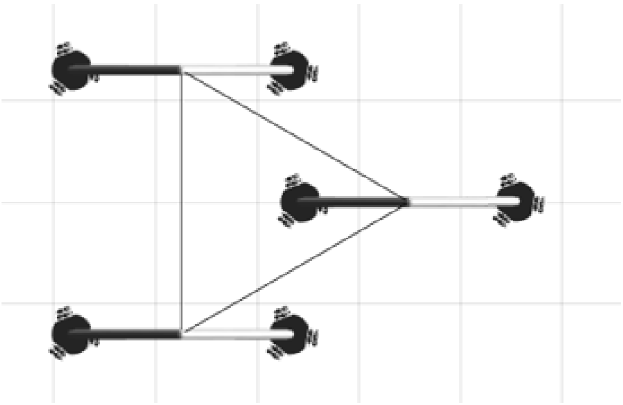


FIGURE 3. A top view of a singular configuration. If for example, all agents move left so will the load. Still, the platform's velocities vector cannot be accurately calculated directly from the jacobian since the configuration is singular.

which is obviously a redundant description of the system's configuration. Fixing the agents' planar locations, 9 distance constraints should be imposed to ensure the rigidity of the mechanism. Since the platform is modeled as a triangle embedded in $\mathbb{R}^3$, three constraints will rigidify the platform leaving 6 DOF for the load transformation. The carrying rods' lengths constitute 6 more constraints. Therefore, each configuration must hold a set of distance constraints $G = \{g_i(c) = l_i\}_{i=1}^{9}$, where g_i is the Euclidean norm. Obviously, one may also add external obstacles constraints set $\{o_j(c) = \epsilon_j\}_{j=1}^{m}$ to the set G if exist, to keep the system from agent-agent collision and agent-obstacle collision (see [28]).

Assume a configuration c_1, and neighboring configuration c_2. Motion from c_1 to c_2, restricted to the boundary of G, can be obtained by calculating ∇g_i at the current configuration $c = c_1$, and restricting motion to the joint null space $\mathcal{K}_G$ (provided as the intersection of all null spaces $\{\mathcal{K}_i\}_{i=1}^{1}2$. Explicitly, motion direction v_d is set to a desired instantaneous direction vector projected onto the null space $\hat{\mathcal{K}_G}$ yields:

$$v = \sum_{i=1}^{12-m} \hat{\mathcal{K}}_i^\top \cdot v_d \cdot \hat{\mathcal{K}}_i \tag{4}$$

where m is the number of the additional workspace 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, the system advances by the following:

$$c(k+1) = c(k) + \epsilon v \tag{5}$$

where ϵ is a predetermined small step size in $\mathcal{C}$.

Let us assume, for example, a scenario with no obstacles where one defines $v_d = (\beta_1, \beta_2)^\top$ where $\beta_1 = (1, 0, 2, 0, 3, 0, \vec{0}_{1\times 6})$ and $\beta_2 = \vec{0}_{1\times 9}$. So agents 1 to 3 should advance only in the x direction and such that agent 2 moves twice as fast and agent 3 moves three times as fast from agent one as implied by the vector β_1. All other agents stay still as well as the platform. In such a case, the dimension of the null space $\mathcal{K}_G$ is 12. Thus, the projection of v_d on $\mathcal{K}_G$ will not necessarily result in the desired agents' motion. Rather the resulting motion will only comply with the imposed constraints. To overcome this, we define a subspace in $\mathcal{C}$ where its vector base is spanned by the columns of the following matrix:

$$A = \left[\begin{array}{c|c} {\beta_1}^\top \ldots {\beta_1}^\top & {\beta_1}^\top \\ \hline I_{9\times 9} & 0_{9\times 1} \end{array}\right]_{21\times 10}$$

By projecting v_d on $\mathcal{K}_G \cap A$, agents' motion is maintained to the desire motion: The agents' motion is confined to the 12-element vector β_1 while the platform is free to move in any direction in $\mathcal{C}$ (which corresponds to the identity sub-matrix) or to stay still (the null vector). Note that in this example, we do not consider collisions (c.f., [28]).

Alternatively, the inverse kinematics is calculated by associating $v_d \in \mathcal{C}$ with the desired motion of the load.

IV. MOTION PLANNING

As stated, the main contribution of this paper is the assumption that the agents do not communicate their location nor their motion direction. The only data available for the agents are the desired load's 3D trajectory (position and orientation) and its current state. Instead, the agents are assumed to have the ability to measure their relative position to the load. We shall now provide an algorithm for each agent such that it obeys the following rules as much as it can:

1) Keeping the load steady,
2) Avoiding obstacles,
3) Advancing the load to its desired destination.

A. POTENTIAL FIELD METHOD

The Potential Field (PF) method is one of the most important and useful methods in robot motion planning. PF is an artificial potential field defined in $\mathcal{C}$ which models mission targets as points of attraction in $\mathcal{C}$ and obstacles (and other agents) as repulsion. Here, we apply six PF's – one for each agent.

At each time step, each agent advances towards the direction of the field's gradient. This way each agent is repelled one from the others and from the obstacles as $\propto 1/o_j(c)$ for all $j \in \{1, \ldots, m\}$.

The main flaw of the PF method is the emergence of local minima. That is, points in the configuration space where the field's gradient vanishes. Here each agent follows an independent PF, so a local minimum for the entire system means that at a certain point in $\mathcal{C}$ all PFs assume a local minima. This indeed may occur, but since solutions are available (e.g., [19], [20]) and since in all the experiments conducted, no such point was observed, we shall not consider such scenarios. However, since the agents' sensors are of limited range, faraway obstacles are not being considered, reducing the chance of local minima.

We shall use the PF method to maintain the aforementioned rules. In addition to the traditional repulsion and attraction forces, we define a force-vector whose purpose is to maintain the load balanced which obviously influences the movement of the pack members as it forces them to take the load's balance into account.

B. BALANCING WHILE MANEUVERING

Recall that the agents can sense objects only in their vicinity (as with ants [7], [11]). Following this, it is natural to assume that each agent can measure the orientation of the connecting-rod it holds and the position of the load-center. The latter assumption implies that each agent can deduce its general movement direction $\hat{v}_{goal}$ (for ants, this assumption is apparently not true, but it is mitigated by following a leader ant strategy).

To balance the load, an agent i calculates the height differences between the far end of the rod it holds $x_i + l_i$, and the desired height of the load-center $p = p_d$ at the destination (see Figure 2). The aim of each will then be to reduce this difference as much as possible.

To do so, we define imaginary circles lying on the floor-plane for each of the holding-nodes as follows (the inner circle demonstrated in Figure 6): The circles' centers $\tilde{x}_i^p$ are positioned at the projection of $x_i + l_i$ on to the plane. The radii of which are the horizontal distances between the agents and the corresponding centers of the circles $\|x_i - \tilde{x}_i^p\|$.

Each agent tries to maintain the load at the desired height. That is, an agent that finds the load to be tilted downwards with respect to it, tries to lift the holding-node by advancing towards the center of its corresponding circle (moves in the direction $v_b = \tilde{x}_i^p - x_i$). If the load is inclined upwards, the agent lowers the node by moving in the opposite direction. To verify the effectiveness of this scheme, a set of simulative experiments were conducted. Unfortunately, the results indicate that this approach does not always work (as detailed in Section V). There may be configurations where moving towards (away from) the center of the circle will lower (elevate) the load (opposite to the desired result). Of course, sharing more information may resolve this, but this would deviate from the scope of this paper. Thus we shall make do with the above and try to improve the balancing scheme and examine the algorithm's performance statistics. To do so, each agent can 'close its loop' simply by measuring its resulting node motion. To elevate the holding-node, the agent first moves radially as described above $\delta\hat{v}_b$ (initialized as $\delta = 1$), and checks whether the node moved in accordance with intentions (i.e. if $\Delta z_k = z_d - z$ is with the desired sign). If the node should be further elevated, the agent will continue moving in the same direction. Otherwise, it will toggle the heuristic $\delta = -\delta$.

Finally, an agent follows:

$$k_1\delta\hat{v}_b + k_2 v_{obs} + k_3\hat{v}_{goal} \tag{6}$$

where v_{obs} is the sum of forces over all detected obstacles. Each is inversely proportional to the square of the distance between the agent and the obstacle.

Figure 4 depicts the calculated directions and the weighted vector sum. Figure 8 presents the algorithm performance in the obstacle workspace while moving towards the goal (green star). Under these assumptions, the expected deviation from a desired orientation will attain its maximal value when the predefined orientation is horizontal and two triangle nodes deviate upwards while the latter deviates downwards. The maximal error in each of the nodes will be assumed when two agents are positioned on opposite sides of the projected circle demonstrated in Figure 6. The i-th node height is given as $h_i^2 = l^2 - {v_b}_i^2$ and so $\Delta h_i = -\frac{v_{bi}}{h_i}\Delta v_{bi}$. The resultant deviation of the load from its designated orientation is $\Delta\theta \sim \frac{4\Delta h_i}{3\|b_i\|}$ (see Figure 2). For the values we used in our simulations ($b = 0.3$, $\Delta v_{bi} = 0.01$, $v_{bi} = 0.21$) the maximal deviation is $\Delta\theta \sim 2.6^o$.

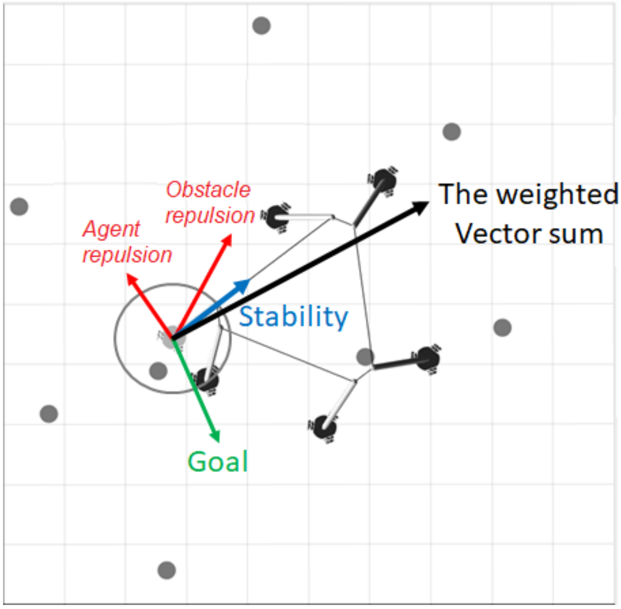


FIGURE 4. Each agent individually decides its current motion direction according to the detected obstacles, detected agents and goal direction.

V. SIMULATIVE STUDY

We first conducted a set of simulated experiments aiming to examine the efficiency of the aforementioned scheme for a pack of agents to avoid obstacles and maintain the load's balance while having no communication between the agents. We first tested the direction of movement of each agent that results in a lift of its corresponding load's node.

Figure 5 exemplifies the directions of movement for all agents, which according to the rationale above, should balance the load by moving each of the corresponding nodes upwards (marked as green half) or downwards (red half). Note that for all agents but agent #1, moving away from the node's projection on the plane lowers the corresponding node. This indicates impossible to unequivocally determine the nodes' vertical direction.

Next, we tested whether it is possible to unequivocally know the rise or fall of the node when a robot walks into the projection circle when located in different initial positions on the circle. In each experiment, all agents were fixed but one, the agent in question, then circumvent the circle completing a

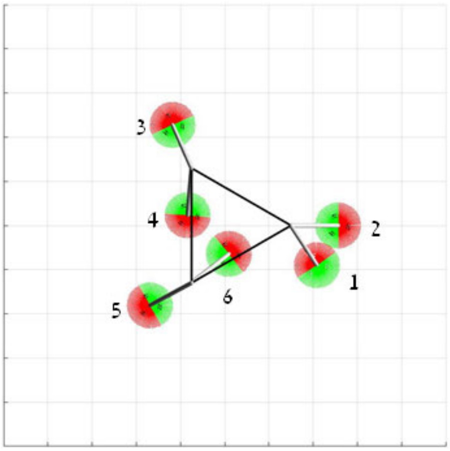


FIGURE 5. Examining whether moving towards the projection circle lifts the node (green) or lowers the node (red). We initialized the load to be horizontal. Here, at each test, only one agent moved. Green is the direction resulting with a node lift.

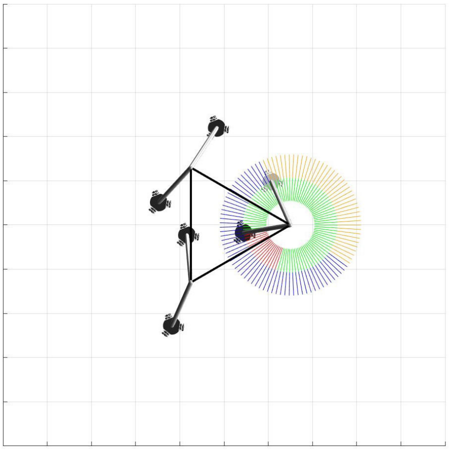


FIGURE 6. Load balance test when agent #2 is located in different positions along the projection circle while agent #1 moves inward into the circle. The Green color indicates that agent #1 node rises, downward movement is indicated by red. Blue and yellow indicates the same for its neighboring agent.

complete turn as depicted in Figure 6. Each starting position of the robot was marked green (red) if moving inwards from that point resulted in the node moving upwards (downwards).

The upward/downward motion of the load depends on the position of the agents that do not move as well as depicted in Figure 7.

We therefore may conclude that since the agents do not communicate and agents do not know the location of the others, it is not possible for an individual agent to determine how it can correct the load orientation by its ego motions. So we resorted to maintaining balance by means of trial and error.

VI. EXPERIMENTAL RESULTS

We conducted two sets of experiments. The first was designed to test our stabilizing scheme while no translational constraints are imposed on the load. The other tested the stabilizing performance during motion planning takes place. It should be noted that in all experiments the load's orientation error is of concern while translation errors that were straightforward to correct are not presented here.

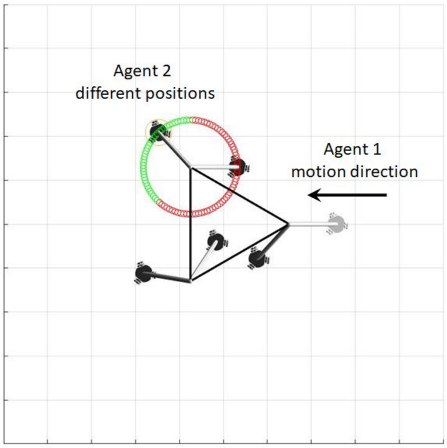


FIGURE 7. Testing whether moving towards the projection circle lifts the node. The load was initialized to be horizontal. At each test all agents were fixed but one. Green is the direction resulting in an upwards motion of the node. Note that in the current configuration agent #1 should move away from the projected circle to lift its node.

A. SIMULATIONS

We conducted a set of over 1, 000 experiments to test the performance of our algorithm in balancing the load and measured the time it took the agents to reorient a load tilted in an initiate angle to an angle under 2°. To quantify the performance of our scheme we measured the resulting error of the load's orientation (in degrees) and used the elapsed time to resolve it, as our metric.

Each test was initiated with random load orientation and with different agents positions. Figure 9 depicts the elapsed time it took for the pack to stabilize the load from an initial inclination angle of 10° degrees measured between the load's normal and the z axis. The results provided in Figure 9 show that the algorithm performs well, and the load can be stabilized to a tilt angle under 2° degrees in less than 15 seconds in all cases tested.

B. REAL-WORLD EXPERIMENTS

In addition to the simulations, a set of real-world experiments were conducted. In these experiments, we tested our scheme or real load which was constructed as an equilateral triangular frame with edges of length 35 cm. We tested the ability to maintain a steady load and to follow a simple motion planner. The load was made of Carbon-fiber beams having a hollow rectangular cross-section of size 2.5 × 2.5 cm. A pack of six robot agents was specially fabricated. Each robot with three omni-directional wheels is capable of translating the plane in all directions. Each robot was controlled via ESP32 microcontroller and motion commands were sent via WiFi. Each robot held the load via a 35cm Carbon-fiber thin rod with magnetic spherical joints at its ends.

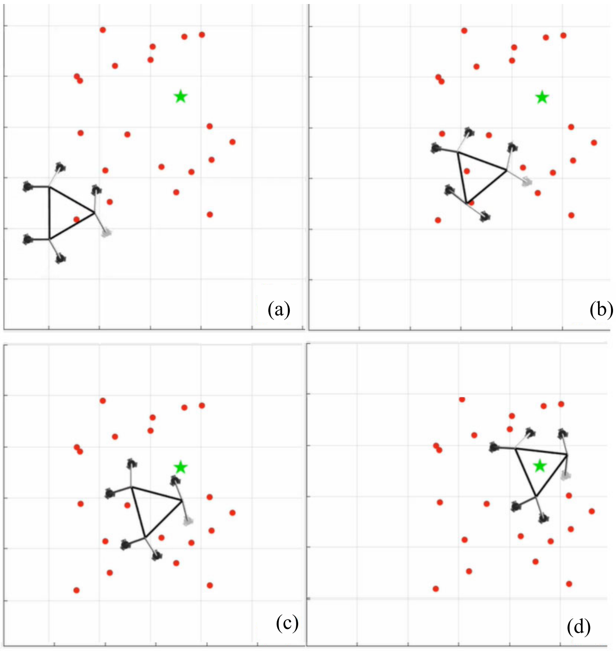


FIGURE 8. Motion towards the goal position (green star) while avoiding obstacle collision and keeping the load stable.

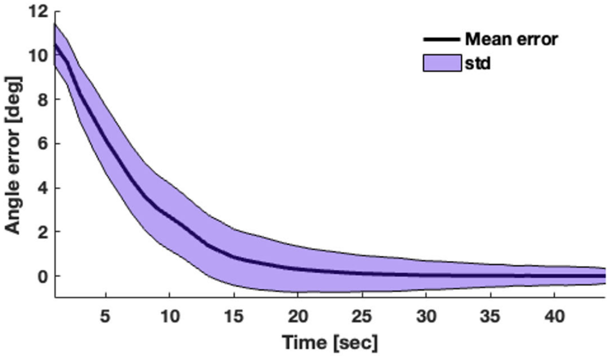


FIGURE 9. Load stabilizing. A typical load stabilizing process. An initial inclination angle of ten degrees measured between the load's normal and the z-axis was eliminated in less than 15 seconds.

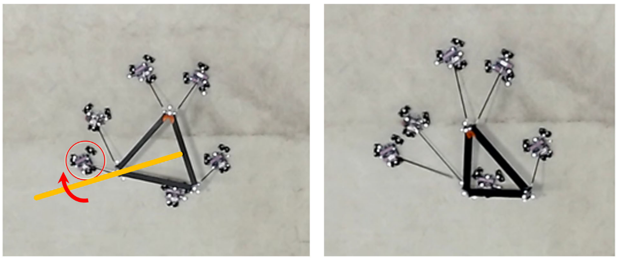


FIGURE 10. For keeping the rods under compression, we add a position constraint to each agent such that an agent should not cross an imaginary line between the projections on the floor of the agent's node on the load and the load's center. On the left - an agent (circled in red) crosses the boundary line causing the load to collapse (right figure).

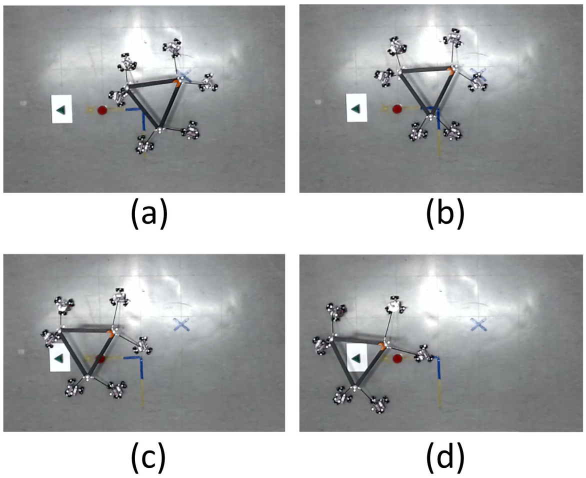


FIGURE 11. A snapshot from a real-world experiment. An obstacle (red) was positioned in the workspace between the initial position of the pack and the target (white rectangle). The system traversed the plane while avoiding mutual collisions and the static obstacle and while keeping the load balanced.

We used a set of 18 OptiTrack surveillance cameras, which provide us with real-time spatial locations of the entire system with an accuracy of two millimeters. All calculations were performed in real-time, and the processing time was negligible. Four markers were connected to each agent for the cameras to detect. From which, the OptiTrack algorithm then constructed a rigid body. In addition, we attached three markers to each of the corners of the ternary-link load. It is important to note that during the experiment, the main computer holds the information regarding all agents' locations. Yet, each agent is shared with only the relevant information according to the algorithm described in Section IV. We first tested the balancing algorithm alone. Six agents were placed on the floor randomly, holding the triangular load by six rods. The allowable range in which we define the load as stable is when each of its nodes is at a distance of up to ±2cm from its desired height. The desired height was set to be 44cm from the ground (33cm from the top of the agents). Upon activating the algorithm, the agents started moving in accordance with the balancing algorithm until the system was balanced and the agents stopped. In order to challenge the system, we randomly moved two agents so that the load would lose its balance. After the system recognizes that the load orientation was altered (using the OptiTrack) the agents balanced the load (see a short video of the experiment in [27]). In all conducted experiments the load was balanced after less than 20sec which is in accordance with the simulation results. Note that the rods' lengths are given as a constraint, so kinematically, both tension and compression forces in the rods are acceptable. In the real world, though, this is true up to a certain limit. Obviously, it is preferable to keep the rods under compression. Thus, we add a position constraint to each agent such that an agent should not cross an imaginary line between the projections on the floor of the agent's node on the load and the load's center. Figure 10 depicts the realization of the said imaginary line.

Indeed, a set of real experiments confirmed the efficacy of this approach. We conducted a set of 15 real-life experiments in which the robotic pack carried the load to the destination, and maintained its desired orientation while avoiding planar obstacles (a static obstacle and the other agents) on the way. In most cases (12 out of 15) the algorithm performed well, and the agents were able to traverse the balanced load to its desired location while avoiding the obstacles. In the unsuccessful experiments, the compression force in one of the rods caused an undesirable tilt of the robot and its lack of ability to control its motion, eventually resulting in the load falling. Figure 11 depicts a snapshot from one of the experiments. A video of the experiment is available in [27].

VII. CONCLUSION

We introduced a novel heuristic motion planning scheme for a load-carrying pack of agents which is a continuation of the author's previous paper on a load-carrying swarm [28]. Here, we drew inspiration from ants' behavior and assumed the agents do not communicate their positions. Furthermore, the agents are considered as such that are able to sense only their vicinity (i.e., the spatial angle at which they hold the load and nearby obstacles).

Unlike most load-carrying research that traverses the load in 2D, in our research, we carry the load spatially. Furthermore, when load carrying is performed in 3D, and a communication failure occurs, the load-agents structure may collapse. Thus, a communication-free motion planning scheme is of interest. The system is designed as a parallel mechanism and consequently suffers from errors in the time propagation calculation using Jacobian formalism. We demonstrated how formalizing the problem in the mechanism's configuration space solves this.

To test our scheme, we fabricated six specially designed robotic agents and exemplified the efficiency of the scheme through a set of simulations and real-world experiments. We first showed that the pack was able to stabilize the load from an initial inclination angle of 10° degrees to a tilt angle under 2° degrees in less than 15 seconds. We then showed that using our scheme the system can avoid obstacles and balance the load while carrying it to a target destination.

In summary, following some simple heuristic rules suffices to complete the vast majority of the missions: Each agent closes its control loop by measuring the height of its load's node; Agents follow toggled potential field (Equation 6); In the real-world experiments, additional constraints were added to to avoid tension in the rods that support the load.

Future work will focus on enlarging the number of agent so they could switch positions or dispatch when required.

REFERENCES

[1] 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.

[2] 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.

[3] 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," *Robot., Sci. Syst.*, vol. 6, p. 259, Aug. 2011.

[4] 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.

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

[6] 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.

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

[8] 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.

[9] H. Farivarnejad and S. Berman, "Multirobot control strategies for collective transport," *Annu. Rev. Control, Robot., Auto. Syst.*, vol. 5, no. 1, pp. 205–219, May 2022.

[10] 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.

[11] O. Feinerman, I. Pinkoviezky, A. Gelblum, E. Fonio, and N. S. Gov, "The physics of cooperative transport in groups of ants," *Nature Phys.*, vol. 14, no. 7, pp. 683–693, Jul. 2018.

[12] 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.

[13] 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, Jul. 2015.

[14] G. Gogu, *Structural Synthesis of Parallel Robots*, vol. 866. Berlin, Germany: Springer, 2008.

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

[16] 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.

[17] 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)*, Jun. 2006, pp. 1317–1323.

[18] 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.

[19] 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.

[20] S. Hacohen, S. Shoval, and N. Shvalb, "Navigation function for multi-agent multi-target interception missions," Res. Square, Durham, NC, USA, Tech. Rep., 2022.

[21] S. Hacohen, O. Medina, T. Grinshpoun, and N. Shvalb, "Improved GNSS localization and Byzantine detection in UAV swarms," *Sensors*, vol. 20, no. 24, p. 7239, 2020.

[22] 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.

[23] M. D. Kennedy, L. Guerrero, and V. Kumar, "Decentralized algorithm for force distribution with applications to cooperative transport," in *Proc. 39th Mech. Robot. Conf.*, Aug. 2015, p. 10.

[24] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *Autonomous Robot Vehicles*. Berlin, Germany: Springer, 1986, pp. 396–404.

[25] 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.

[26] K. M. Lynch, *Nonprehensile Robotic Manipulation: Controllability and Planning*. Pittsburgh, PA, USA: Carnegie Mellon Univ. Pittsburgh, 1996.

[27] O. Medina. *Robotic Pack Load Carrying*. Youtube. [Online]. Available: https://youtu.be/b4_8i8f1a4o

[28] O. Medina, S. Hacohen, and N. Shvalb, "Robotic swarm motion planning for load carrying and manipulating," *IEEE Access*, vol. 8, pp. 53141–53150, 2020.

[29] J.-P. Merlet, *Parallel Robots*, vol. 128. Berlin, Germany: Springer, 2006.

[30] B. M. St-Onge and C. M. Gosselin, "Singularity analysis and representation of the general gough-stewart platform," *Int. J. Robot. Res.*, vol. 19, no. 3, pp. 271–288, Mar. 2000.

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

[32] 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.

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

[34] 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.

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

[36] 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.

[37] 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, Jan. 2009.

[38] 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.

[39] F. Tahmasebi and L. Tsai, "Workspace and singularity analysis of a novel six-DOF parallel minimanipulator," ISR, Inst. Syst. Res., Tech. Rep. TR 93-55, 1993.

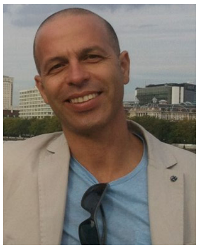
[40] Z. Wang, G. Yang, X. Su, and M. Schwager, "OuijaBots: Omnidirectional robots for cooperative object transport with rotation control using no communication," in *Distributed Autonomous Robotic Systems*. Berlin, Germany: Springer, 2018, pp. 117–131.

[41] 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. 876–887, Oct. 2004.

[42] L. Zhang, Y. Sun, A. Barth, and O. Ma, "Decentralized control of multi-robot system in cooperative object transportation using deep reinforcement learning," *IEEE Access*, vol. 8, pp. 184109–184119, 2020.

HAREL COHEN is currently pursuing the Graduate degree. He is also a Teaching Assistant with the Mechanical Engineering Department, with a specialization in robotics. His research interest includes related to robotic multi-agent systems. His early research project included implementation and development of a motion planning algorithm for the Stewart platform in a robotic swarm.

SHLOMI HACOHEN is currently a Lecturer in mechanical engineering with Ariel University. His expertise are 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 multi agents control under uncertain conditions. He also deals with classical control theory and non-linear estimation methods which may be applied to indoor positioning problems.



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 also the Founder of several robotics startup companies amongst are Momentis Surgical and W Endoluminal Robotics. His research interests include medical robotics, global path planning, and theoretical foundations of robotics.

ODED MEDINA received the Ph.D. degree from Ben Gurion University. He is currently a Lecturer with the Mechanical Engineering Department, Ariel University. In his research laboratory, 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 flexible mechanisms, which due to their flexibility require a large number of actuators. Most projects also included the designing and fabrication of actual mechanisms.