DOI: 10.1109/TRO.2022.3181055
The final version of this paper is available at https://ieeexplore.ieee.org/document/9813566

Closed-Loop Motion Control of Robotic Swarms – A Tether-Based Strategy

K. Eshaghi, *Student Member, IEEE,* A. Rogers, G. Nejat, *Member, IEEE,* and B. Benhabib

***Abstract*—Swarm robots can achieve effective task execution via closed-loop motion control. However, such a goal can only be realized through accurate localization of the swarm. Past approaches have focused on addressing this issue using external sensors, static sensor networks, or through active localization – requirements that may restrict the motion of the swarm or may not be achievable in practice.**

We present a tether-based strategy that achieves closed-loop swarm-motion control by using a secondary team of mobile sensors. These sensors form a wireless tether that allows the swarm to indirectly sense a home base or a landmark, and to compensate for the accumulated motion errors via a closed-loop control strategy.

The proposed strategy is the first to use a tether of mobile sensors that can dynamically re-shape and re-connect to various points in the environment to achieve closed-loop motion control. The novelty of the strategy is in its ability to adapt to any swarm motion considered, and to be applied to swarms with limited sensing capabilities and knowledge of their environment. The performance of the proposed strategy was validated through extensive experiments.

***Index Terms*— Robotic swarms, dynamic sensors, point-to-point swarm motion paths, localization, motion control.**

I. Introduction

SWARM robotic systems (SRSs) comprise large teams of robots that aim to accomplish complex tasks through tight inter-agent coordination [1]–[3]. Such coordination is, ideally, achieved through a decentralized architecture, where robots make decisions based on information obtained from their neighbors and the environment. This promotes scalability to an increase in the number of member robots, and robustness to failure in robot components.

While the swarm-robotics literature does advocate the overall use of decentralized control systems, there are circumstances that require the use of a centralized component, especially, for simplifying the collaboration of member robots [4]–[13]. One such case is swarms that comprise small-sized robots (*e.g.*, millimeter-scale robots – *millirobot*s), which cannot be equipped with meaningful sensing, communication, and computation hardware as would larger robots be.

Manuscript received January 6, 2022; revised March 31, 2022, accepted May 28, 2022.

This work was supported in part by the Natural Sciences and Engineering Research Council of Canada and in part by the Canada Research Chairs program. *(Corresponding author: Kasra Eshaghi.)*

The authors are with the Department of Mechanical and Industrial Engineering, University of Toronto, Toronto, ON., M5S 3G8, Canada (e-mail: kasra.eshaghi@mail.utoronto.ca, a.rogers@mail.utoronto.ca, nejat@mie.utoronto.ca, beno@mie.utoronto.ca).

Digital Object Identifier (DOI): see top of this page.

This specific challenge constitutes the focus of our paper. Namely, in our proposed tether-based motion control strategy, a swarm of (technologically-limited) robots rely on collaboration with a sensor team to effectively achieve closed-loop control via a centralized sub-system.

The extensive literature review in [5] further supports the above argument. It notes that centralized approaches have been needed and, commonly, used to plan trajectories of swarm members, while local motion planners are used for collision and obstacle avoidance. The paper also highlights that while motion planning can be achieved in a decentralized manner, this would require extensive communications among swarm members, which may cause delays and even result in infeasible or sub-optimal solutions due to the physical limitations of the robots.

Furthermore, in [7], a centralized system was used to plan the motion of the swarm by optimizing the sequence of swarm behaviors that must be completed for the task at hand. Motion planning is achieved intermittently by the central coordinator, while the execution of this motion is completed by member robots in a decentralized manner. In [10], a swarm of robots with limited sensing capabilities are used in a mapping scenario. The exploration of the environment, including obstacle avoidance, is achieved in a decentralized manner individually by each robot. However, the fusion of the obtained sensor measurements, and the subsequent commands are planned by a centralized server. Similarly, in [12], a central system continuously localizes the swarm and plans its motion in an object manipulation scenario.

In addition to the technological limitations of small-sized robots, swarms may also be required to operate in indoor and unstructured environments, for applications such as search and rescue [14] and environment monitoring [15], which inhibit the use of external measurement devices that can be used to overcome these limitations.

Thus, due to all abovementioned and other shortcomings, swarm robots may be unable to individually localize accurately (*i.e.*, estimate their position with respect to a global frame). This would prevent them from achieving closed-loop swarm-motion control, which includes travel along a given path while the swarm maintains its desired formation (*i.e.*, topology) or varies it as needed [16], in a decentralized manner.

Herein, thus, we propose a unique tether-based strategy for closed-loop swarm-motion control with a centralized architecture component. The strategy uses a team of mobile sensors that collaborate with the swarm as it executes its motion. This team of mobile sensors form a wireless tether that dynamically changes its shape to extend the sensing capabilities of the swarm, and to maintain connectivity between the swarm

and a home base or a landmark in its environment. By maintaining connectivity with and indirectly sensing the environment through this wireless tether, the swarm can localize through sensor measurements that are subject to noise that is independent of its traveled distance. This allows it to compensate for the errors that are accumulated during motion, resulting in closed-loop control with bounded motion errors through online corrective relocation of the swarm along its path.

The proposed strategy uses an architecture with a mix of centralized and decentralized components. Namely, a centralized architecture is used to plan the motion of the swarm-sensor team and to localize it during execution, while the calculation/execution of motion commands, obtaining sensor measurements, and communicating these measurements are completed in a decentralized manner.

The proposed strategy represents the first use of a tether of mobile sensors that is dynamically re-configured and re-connected to various points in the environment for achieving closed-loop swarm-motion control. The strategy is novel as it decouples the tether-planning problem from the motion of the swarm. Namely, the motion of the mobile sensors is planned and executed to minimize the swarm's localization error, without impeding its motion. This allows the swarm's motion to be planned for the task at hand, without a need to compromise the task's objectives for the localization requirements of the swarm. Such decoupling also promotes scalability as the paths of the mobile sensors are planned independent of the number of robots in the swarm. This is an important feature as swarms may consists of hundreds of robots, and planning their individual paths is not computationally feasible.

Finally, our strategy can be applied to a homogenous swarm-sensor team with limited onboard sensing technology operating in a partially known environment. This promotes flexibility as it allows the proposed strategy to be applied to swarms of simple (technologically-limited) robots operating in unknown environments, and, also, promotes robustness as mobile sensors and swarm robots can replace each other during motion execution.

It must be noted that, while the proposed strategy is developed, specifically, for swarms with limited onboard hardware, it can be adapted to (non-swarm) multi-robot systems, when members are required to collaborate with each other for enhanced localization.

This paper begins with a detailed discussion of related works in Section II below. The problem addressed is, then, formally defined in Section III. The proposed tether-based motion strategy is presented in Section IV, with details of the strategy implemented for a swarm-sensor team that is equipped with relative position sensing technology. Next, Section V illustrates multiple simulated and physical experiments, and the tether-based strategy is compared to competing approaches in Section VI. The paper concludes in Section VII.

II. Related Works

Swarm-motion control has commonly been achieved through behavioral [17]–[22], virtual structure [23], [24], and follow-the-leader methods [25]–[30]. These approaches focus on developing control laws that allow robots to maintain a given shape while traversing along a path in the environment. Other approaches that coordinate the motion of individual robots to achieve the task at hand while avoiding inter-robot collisions have also been investigated [31]–[37]. The abovementioned approaches, however, assume that the swarm's position, with respect to the global frame, is known. This may not be the case in practice and the swarm must be localized, with tangible uncertainty, during motion execution, to correct its motion commands towards its destination.

Literature in swarm localization has focused on developing methods to combining the inter-robot sensor measurements and their executed motion commands [38]–[51]. For example, in [40], the authors propose an approach to estimating the position of all member robots with respect to a local swarm frame based on the inter-robot sensor measurements. Other methods that also consider the motion commands executed by the swarm have also been proposed [38]. These approaches would allow for swarm localization with respect to a global frame. However, they typically do not bound the swarm's motion errors. Namely, through such approaches, the swarm accumulates motion errors as it executes its motion, and drifts away from its desired path – potentially resulting in ineffective swarm-motion control. Our work aims to develop a strategy that ensures the swarm's motion errors are independent of its travel distance (*i.e.*, are bounded).

The accumulation of motion errors can be minimized using proximity measurements to a stationary reference frame in the swarm's operating environment. One such proposed approach has been to utilize an anchor-based motion strategy that divides the swarm into multiple sub-groups which take turns traveling to their destinations [52]–[60]. Even though such methods enhance swarm localization, they could still be prone to unbounded motion errors since the reference frames used for localization (*i.e.*, the anchors) are temporary. Furthermore, they would inhibit the swarm from maintaining a desired shape while in motion, as a subset of the swarm always remains stationary.

Other approaches that address the accumulation of motion errors consider the use of (*i*) external sensing infrastructures, or (*ii*) *a priori* known maps of the swarm's operating environment. The former, such as overhead camera systems, ground positioning systems (GPSs), and static sensor networks, would allow the swarm to achieve accurate localization by considering proximity measurements made by these devices [61]–[67]. These technologies, however, may not be available in swarm applications that are completed in indoor and unstructured settings, as considered in our work. The proposed tether-based motion strategy does not depend on external sensing infrastructure, providing it with flexibility to operate in indoor and unstructured settings.

The latter approach of using an *a priori* known map would allow the swarm to obtain proximity measurements to

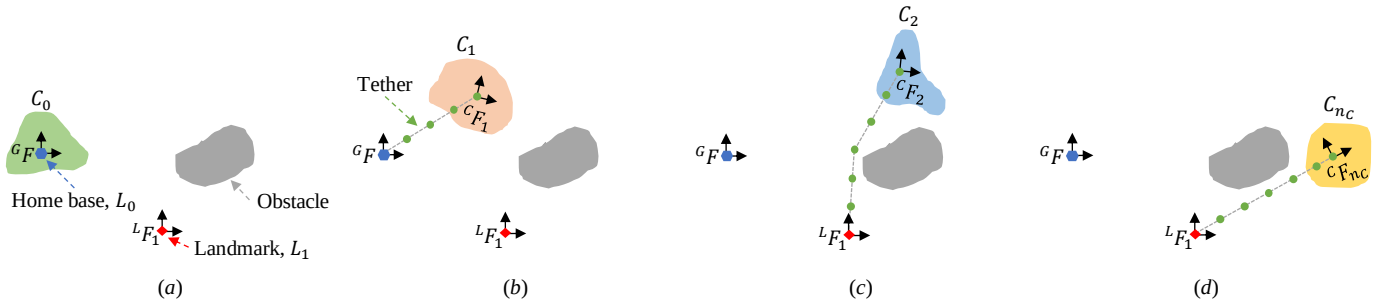


Fig. 1. Tether-based connectivity of the swarm – shown as groups (different overall shapes for different configurations): The swarm is (*a*) at its initial configuration, C_0, at ${}^{G}F$, (*b*) connected to home base, L_0, when it reaches the first desired configuration, C_1, and (*c*)-(*d*) connected to Landmark 1, L_1, at ${}^{L}F_1$ when it reaches its second and last configurations, C_2 and C_{n_C}, respectively.

landmarks as it moves through its known environment, and to localize by comparing these measurements to the map at hand [68], [69]. Though, unlike external infrastructures, landmarks may not always be *visible* to the swarm robots, especially in case of millirobot platforms – such as the *mROBerTO* [70]–[72], Kilobot [73], GRITSBot [74], Alice [75], and TinyTerp [76] – that are equipped with on-board sensing technologies with limited sensing range. In such cases, the swarm may need to divert from its desired motion to visit surrounding landmarks for enhanced localization. This falls in the category of *active localization*, where centralized [77] and decentralized [78]–[80] approaches have been developed that alter the original motion of the swarm, planned for the specific task at hand, in order to reduce the uncertainty in localization. Such alterations to the original motion may result in reduced performance in the swarm's desired task. Thus, when active localization methods are used, the swarm must optimize its motion either for the task at hand, or for enhanced execution. In contrast, our proposed tether-based strategy uses a secondary team of mobile sensors to extend the swarm's sensing range and eliminate its need to divert from the desired motion to sense the surrounding environment. This decouples the swarm's localization requirements from that of its desired motion and allows the swarm's motion to be planned solely for the objectives of the task at hand.

III. Problem Formulation

Our research focuses on developing a tether-based closed-loop motion control strategy for a swarm robotic system traveling along a point-to-point (PTP) path in an environment, potentially, with some *a priori* known landmarks and obstacles. The path is defined as a sequence of n_C desired configurations, $\boldsymbol{C} = \{C_i\}_{i=0}^{n_C}$, that the swarm is expected to attain, Fig. 1. Each configuration is defined by a local frame, ${}^{C}F_i$, positioned at the centroid of the swarm.

The proposed objective is to keep the swarm *connected* to either the home base, L_0, at its starting configuration, defined by a global frame, ${}^{G}F$, or to one of the possible n_L landmarks in the swarm's environment, $\{L_i\}_{i=1}^{n_L}$, Fig. 1. It is assumed that a static sensor that can actively sense the swarm could be placed at home base prior to motion execution, whereas a landmark, L_i, defined by ${}^{L}F_i$, represents a uniquely distinguishable passive object that must be sensed through hardware onboard the robots, such as proximity sensors or cameras.

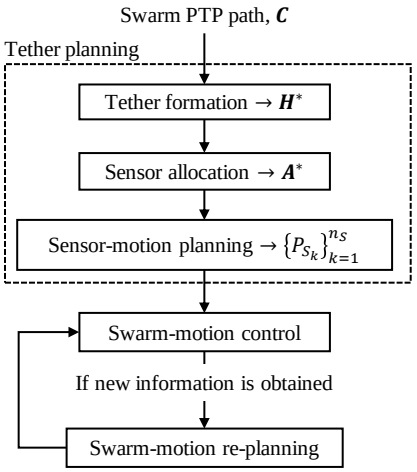


Fig. 2. Sub-problems of tether-based swarm-motion strategy.

Connectivity is to be realized via a (variable-length) wireless tether, comprising a set of (mobile) sensors (***green*** dots in Fig. 1), that could be re-formed dynamically as the swarm travels. It is assumed that the positions of the home base and the landmarks are *a priori* known, with negligible uncertainty. Thus, by maintaining connectivity through the tether, the swarm can indirectly sense the home base or a landmark with noise that is independent of its traveled distance. This allows it to achieve its PTP path with errors that are bounded.

Based on the above preliminary discussion, the swarm-motion planning and control problem at hand can be formulated to comprise three main sub-problems, Fig. 2: tether planning, swarm-motion control, and swarm-motion re-planning, discussed in Sub-sections III.A to III.C, respectively.

A. Tether Planning

The off-line tether-planning stage comprises three sub-steps:

1) *Tether formation*: For each desired configuration, one first needs to determine whether the swarm should continue to maintain its initial connectivity to home base or switch connectivity to a landmark. This is followed by determining the number and the positions of the nodes of the tether at hand.
2) *Sensor allocation*: For each tether, the mobile sensors need to be specifically assigned to the nodes. In our work, it is assumed that sensors which form one tether could be relocated to form the next tether, as needed. Namely, dynamic re-formation of the tethers (determined in Step 1)

would need to be achieved by the deployment of mobile sensors (as opposed to static sensors). Individual sensors could, thus, be assigned to multiple tethers and move from one tether to the next as the swarm travels from one configuration to the next.

3) *Sensor-motion planning*: As the last step of tether planning, based on the formed tethers and corresponding sensor allocations, the mobile sensors' paths from one tether to another would be determined.

1) Tether Formation

In the proposed motion control strategy, the SRS needs to maintain continued connectivity either with a home base or a landmark (*i.e.*, a connectivity point), as it travels from one configuration to the next along the PTP path. Namely, the swarm would start its motion, from C_0 toward C_1, while being connected to home base through the first formed tether. Thereafter, however, as it moves to its subsequent configurations, C_2 and onward, the swarm may switch its connectivity to one of the n_L landmarks, if it becomes (a) more advantageous in terms of enhancing the closed-loop motion control performance of the swarm, and/or (b) necessary due to the limited number of available mobile sensors, n_S.

In this regard, let L_{ci} represent the connectivity point of the swarm once it reaches C_i. For example, as shown in Fig. 1, C_1 is connected to $L_{c1} = L_0$ (*i.e.*, home base), C_2 is connected to $L_{c2} = L_1$, and C_{n_c} is connected to $L_{cn_C} = L_1$.

Thus, the primary problem to be addressed at this stage is to determine whether the swarm should remain connected to home base or switch to a landmark – to be evaluated for every successive desired swarm configuration along the PTP path. This evaluation would, of course, require the formation of the best possible respective tether, $H_i(L_{ci})$, for the swarm configuration at hand, C_i, to the considered landmark, L_{ci}. The optimization objective function for this choice could be, for example, to minimize the localization error of the swarm, e_{li}, at C_i:

$$\text{Min } e_{li} = f\big(L_{ci}, H_i(L_{ci})\big), \qquad i = 1 \dots, n_C, \tag{1}$$

where the positions of a tether's nodes can be designated by:

$$H_i(L_{ci}) = \left\{ {}^G\boldsymbol{x}_{N_j i} \right\}_{j=1}^{n_{Ni}}. \tag{2}$$

Above, the function f describes the dependence of swarm-localization error on the connectivity point of the swarm and the positions of the tether nodes, and ${}^G\boldsymbol{x}_{N_j i}$ is the position of tether Node j, with respect to the global frame, ${}^GF$, for the tether with n_{Ni} nodes connecting configuration C_i to its connectivity point L_{ci}.

The solution of the above optimization problem must ensure connectivity between the swarm and the selected connectivity point through the tether. This constraint must be satisfied while considering the characteristics of the onboard sensing technology of the mobile sensors that are allocated to the tether in the following sensor allocation step. Such characteristics may include the maximum sensing range, line-of-sight requirements, and the connectivity that is required between nodes of the tether for indirect localization of the swarm.

The tethers must also be formed while considering the limited number of available mobile sensors that are allocated to each tether. It is assumed, herein, that there exists a sufficient number of mobile sensors for the swarm to maintain connectivity with at least one landmark at all configurations along its desired PTP path.

Once (1) is solved, the ordered set of optimal tethers, $\boldsymbol{H}^*$, to be achieved for connecting the swarm to its optimal connectivity point at each desired configuration, L_{ci}^*, is determined.

2) Sensor Allocation

Once all tethers have been formed, the next problem at hand is the allocation of the n_S available mobile sensors, $\boldsymbol{S} = \{S_k\}_{k=1}^{n_S}$, to the tether nodes, one sensor per node:

$$A_i = \{a_{ij}\}_{j=1}^{n_{Ni}}, \tag{3}$$

where A_i is the allocation of the robots to the nodes on tether $H_i^*(L_{ci}^*)$, and a_{ij} is the sensor allocated Node j of this Tether. For example, $a_{12} = S_2$ implies that Sensor 2 is allocated to the second node of the tether connecting C_1 to L_{c1}^*, and $a_{22} = S_2$ indicates that the same sensor is, then, allocated to the second node of the (next) tether connecting C_2 to L_{c2}^*.

Thus, the primary problem at hand is to determine the optimal sensor allocations to all tethers, $\boldsymbol{A}^* = \{A_i^*\}_{i=1}^{n_C}$. The optimization objective function for this task could be, for example, to minimize the maximum distance traveled by each sensor over all the formed optimal tethers:

$$\min\left(\max\left(\{d_{S_k}\}_{k=1}^{n_S}\right)\right) = g(\boldsymbol{A}), \tag{4}$$

where the function g describes the dependence of the length of the path planned for each sensor, d_{S_k}, on the allocation of mobile sensors to the formed tethers.

Sensor allocations must be determined while ensuring that the resulting sensor paths allow the swarm to maintain real-time connectivity with the selected connectivity point.

3) Sensor-Motion Planning

The desired PTP path of each mobile sensor S_k, P_{S_k}, with n_P points, must be determined, as it moves from one tether to the next:

$$P_{S_k} = \left\{ {}^G\boldsymbol{x}_{S_k i} \right\}_{i=0}^{n_P}, \tag{5}$$

where the positions of the sensors, ${}^G\boldsymbol{x}_{S_k i}$, are defined with respect to the global frame, ${}^GF$.

The sensor paths must be determined to allow the swarm to maintain real-time connectivity with home base or a landmark during swarm-motion execution. Moreover, paths of sensors that are not allocated to a tether at any given time must also be determined according to an optimal strategy, for example, providing the swarm-motion control strategy with maximum flexibility for potential future tether re-deployment, when necessary.

B. Swarm-Motion Control

The on-line swarm motion control stage synchronizes the desired PTP path of the swarm, and the PTP path of the mobile sensors that are planned during the preceding offline tether planning stage. The synchronization of these paths must address two main problems: (*i*) using the formed tethers to maintain continuous connectivity between the swarm and the selected connectivity points while moving along the PTP path, and (*ii*) switching the connectivity point of the swarm at different configurations along its path (*e.g.*, Fig. 1(*b*) versus Fig. 1(*c*)).

The swarm-motion control stage should consider the objectives of the preceding tether planning stage by synchronizing the motion of the swarm-sensor team to allow for the formation of optimal tethers. Moreover, the requirements of swarm-motion re-planning, that may be invoked if new information is obtained during swarm-motion control, must also be considered. Finally, the swarm-motion control stage should provide modularity by allowing the utilization of any open-loop swarm-motion strategy proposed in literature – *e.g.*, behavior-based [17]–[22], leader-follower [25]–[30], or virtual structure [23], [24] methods.

Once connectivity maintenance and switching are addressed, the swarm must achieve closed-loop motion control at each desired configuration C_i, by localizing based on indirect sensor measurements obtained through the tether, and reducing its motion error, e_{mi}, to below a user-defined threshold, e_{mmax}.

C. Swarm-Motion Re-planning

During swarm-motion control, the swarm robots and/or the mobile sensors may be subjected to events that could not have been *a priori* accounted for during the tether-planning stage. These events could include, encountering physical obstacles whose positions were unknown at the planning stage, unsuccessful switching of swarm connectivity, loss of mobile sensors due to hardware failure, etc. Such events may prevent the swarm from achieving its planned motion. The swarm should, thus, have the ability to re-plan its tethers online subject to formulation outlined above.

IV. Proposed Tether-based Swarm-Motion Strategy

The proposed tether-based swarm-motion strategy requires the swarm to be continually connected to home base, or to a different landmark, while executing its PTP path. As noted above, the successful implementation of such a strategy requires solutions to two primary problems: (*i*) maintaining connectivity to a connectivity point using wireless tethers, and, when needed, (*ii*) switching connectivity from one connectivity point to another.

For the former sub-problem of *connectivity maintenance*, we suggest the use of a team of mobile sensors that travel with the swarm, whose members maybe 'dropped' and left behind. Namely, we consider a robotic swarm that is equipped with a 'bag' of mobile sensors, which it leaves behind one at a time as it travels, forming a tether to a home base, or to a different landmark at a future time. These tethers would need to be re-shaped, including to be shortened or lengthened, as the swarm moves, thus, allowing for the formation of optimal tethers.

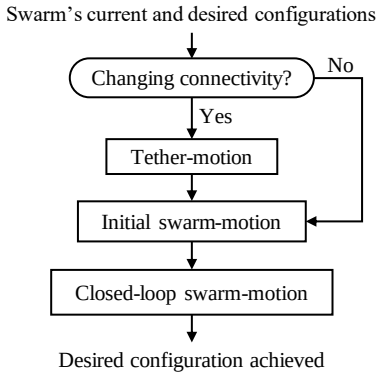


Fig. 3. Proposed tether-based swarm-motion control strategy.

For the latter sub-problem of *connectivity switching*, we suggest the swarm, first, switches its connectivity point from its current configuration to the next landmark, while remaining stationary, before starting its motion toward the next configuration on its PTP path.

A high-level overview of the proposed tether-based swarm-motion strategy, used to move the swarm from its current to its next desired configuration on its PTP path, and to close the motion control loop at this configuration, is shown in Fig. 3. The swarm's motion is completed over three main stages. In the first *tether-motion* stage, the swarm switches its connectivity to the selected connectivity point, if necessary. Then, in the *initial swarm-motion* stage, the swarm moves to its desired configuration, while the tether is dynamically re-shaped to maintain connectivity to the selected connectivity point. Once the swarm reaches its desired configuration, in the *closed-loop swarm-motion* stage, the swarm is localized through the tether, and iterative motion correction steps are completed to minimize the motion error of the swarm.

The proposed tether-based strategy achieves bounded motion errors as the swarm maintains connectivity with a stationary reference point (*i.e.*, the home base or a landmark) at all configurations along its PTP path. Thus, it can estimate its position with uncertainty that is independent of its traveled distance. As detailed in Section IV.A below, by maintaining connectivity to a reference point, the swarm's localization error, and in turn its motion errors, will be bounded by a value that is approximately a multiple of the length of the formed tethers. In contrast, open-loop control strategies that do not use tethers would accumulate motion errors as the swarm travels in its environment.

The proposed tether-based motion strategy is novel as it plans the motion of the mobile sensors to accommodate for the desired motion of the swarm, without impeding with it. This allows the desired motion of the swarm to be planned for the objectives of the swarm's mission, without considering its localization requirements, and executed using any swarm-motion control method, such as behavior-based [17]–[22], leader-follower [25]–[30], or virtual structure [23], [24] ones. This feature also promotes scalability, as the motion of the mobile sensors is planned independent of the number of robots in the swarm.

A. Tether Planning

The offline tether-planning stage, detailed below, first, forms the optimal tether corresponding with each desired swarm configuration, *i.e.*, *tether formation*, Section IV.A.1. The next step is the allocation of mobile sensors to the nodes of the tethers, *i.e.*, *sensor allocation*, Section IV.A.2. Lastly, based on these, the PTP paths of the sensors from one tether to another are determined, *i.e.*, *sensor-motion planning*, Section IV.A.3.

1) Tether Formation

The tether formation stage must choose a connectivity point which the swarm will perceive at every desired configuration along its PTP path, whether it continuing to be the home base or a new landmark. The corresponding tether that provides this connectivity must also be determined.

The connectivity point of the swarm and the corresponding tether are selected herein to minimize the localization error of the swarm at each desired configuration. The localization error of the swarm, once it arrives at configuration C_i, is defined as the average Euclidean distance between the estimated and true positions of all n_R members:

$$e_{li} = \frac{1}{n_R}\sum_{k=1}^{n_R} \left| {}^{G}\boldsymbol{x}_{R_k it} - {}^{G}\hat{\boldsymbol{x}}_{R_k i} \right|, \tag{6}$$

where ${}^{G}\boldsymbol{x}_{R_k it}$ is the true position of swarm Robot R_k and ${}^{G}\hat{\boldsymbol{x}}_{R_k i}$ is the estimate of this position. Minimum swarm localization error, however, can only be achieved by minimizing the localization error of the corresponding tether.

(i) Considered Sensing Models

The formed tethers must provide connectivity between the swarm and a selected connectivity point to allow for indirect sensing of a landmark that is not within the swarm's immediate surrounding. The position of the nodes that provide this connectivity depend on the sensing model of the mobile sensors and swarm robots. Namely, different sensing technologies have different connectivity requirements that must be satisfied between the nodes on a tether, and the corresponding formed tethers would have different parameters that can be selected to minimize the localization error of the swarm.

In this regard, our work considers inter-robot sensing technology that allows a mobile sensor to measure the distance and bearing to its neighbors:

$$z_{S_i S_j} = \left(\hat{d}_{S_i S_j}, \hat{b}_{S_i S_j}\right), \tag{7}$$

where $z_{S_i S_j}$ is the inter-robot sensor measurements that S_i makes of S_j, and $\hat{d}_{S_i S_j}$ and $\hat{b}_{S_i S_j}$ are, respectively, the relative distance and bearing between these sensors, as observed by S_i. Fig. 4.

In practice, such sensor measurements would be equal to the true relative distance and bearing between robots, plus additional zero-mean Normally-distributed noise:

$$\hat{d}_{S_i S_j} = d_{S_i S_j} + N(0, \sigma_{int}^2), \quad \text{and} \tag{8}$$

$$\hat{b}_{S_i S_j} = b_{S_i S_j} + N(0, \sigma_{int}^2), \tag{9}$$

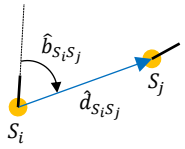


Fig. 4. Inter-robot sensor measurements, thick black lines represent robot orientations.

where $d_{S_i S_j}$ and $b_{S_i S_j}$ are the true distance and bearing between the sensors, and σ_{int}^2 is the variance of the additive Normally-distributed noise. σ_{int}^2 is in units of millimeters and degrees for the distance and bearing measurements, respectively.

We also consider outer-robot sensing technology that allows a mobile sensor to measure the relative distance and bearing to surrounding landmarks, where $z_{S_i L_j} = \left(\hat{d}_{S_i L_j}, \hat{b}_{S_i L_j}\right)$ is the outer-robot proximity measurement between sensor S_i and landmark L_j. Such sensor measurements would also be subject to zero-mean Normally-distributed noise, with their variance denoted by σ_{out}^2. Both sensing models (for inter- and outer-robot measurements) have a limited operating range of z_{max}, and require an obstacle-free line-of-sight between the sensors and/or to a landmark.

In our work, it is assumed that outer-robot measurements can uniquely identify the landmarks that are within its sensing range. Such measurements and identifications can be achieved by individual robots through a variety of sensors, including proximity sensors or cameras. In the absence of individual identification capabilities, existing collective swarm perception methods can be used [81].

(ii) Connectivity Requirements and Optimization Metrics

For the sensing models detailed above, the tether nodes must be placed so that (1) at least one tether node is within detection range of the selected connectivity point, and (2) all other nodes maintain connectivity with at least one neighbor that is closer to the selected connectivity point. Based on these minimum requirements, tethers are formed, herein, by placing nodes along a straight line that connects the swarm to the connectivity point, Fig. 1.

Moreover, a series of investigations were conducted in our work to determine the parameters that affect the localization error of such a tether. The results of these investigations, detailed in Appendix A, validated that localization errors accumulate along the tether. Thus, the length of a tether, l, should be minimized. It was also noted that the rate of the accumulation of localization errors along the tether, δ, depends on where the swarm is connected to. Namely, compared to home base, landmarks may be perceived with higher uncertainty due to the onboard hardware of the mobile sensors. Thus, the uncertainty in measuring the position of a selected connectivity point needs to be considered in the tether formation process.

Based on the above conclusions, the localization error of the swarm, e_{li}, for a considered connectivity point and corresponding straight tether, L_{ci} and $H_i(L_{ci})$, respectively, to be minimized at each desired configuration is approximated by:

$$e_{li}\big(L_{ci}, H_i(L_{ci})\big) \approx \delta(L_{ci})\, l\big(H_i(L_{ci})\big). \quad (10)$$

One may note that, as mentioned above, the node connectivity requirements, which in turn dictate the shape of the formed tether, depend on the sensing technology considered. For example, when distance-only measurements are considered, the mobile sensors on the tether would need to be localized through trilateration. Thus, the tethers would need to be formed to allow for connectivity between a node at least three other neighbors. Similarly, if bearing-only measurements are considered, then, localization is achieved through triangulation, and the tethers must be formed to allow connectivity between a node and at least two other neighbors. Such considerations would also require an investigation of the parameters that affect the localization performance of the formed tether.

(iii) Proposed Search Algorithm for Tether Formation

The tether formation problem needs to be solved, for every desired swarm configuration, C_i, through an algorithm comprising two nested loops.

Outer Optimization Loop: For each desired swarm configuration, the outer loop seeks to determine the optimal connectivity of the swarm by searching through the space of all possible landmarks and the home base. This discrete search space, with $n_L + 1$ possible solutions, can be searched through a discrete-variable search engine, such as Tabu Search [82]. For every connectivity point considered by the outer optimization loop, L_{ci}, the inner loop must, then, determine the corresponding optimal tether, $H_i^*(L_{ci})$, and return the corresponding localization error of the swarm, $e_{li}\big(L_{ci}, H_i^*(L_{ci})\big)$.

Once the search is completed, the optimal connectivity point and the optimal tether for the configuration at hand are determined, L_{ci}^* and $H_i^*(L_{ci}^*)$, respectively.

Inner Optimization Loop: In this loop, an optimal tether is formed by (*i*) finding the shortest path connecting the swarm at C_i to the connectivity point considered by the outer optimization loop, L_{ci}, and (*ii*) placing tether nodes along this path. The shortest path can be determined through existing path planning methods, such as sampling-based Rapidly Exploring Random Trees (*e.g.*, RRT* [83]). The tether nodes are, then, placed along this path through simple geometry.

For a tether connecting the swarm to a connectivity point through a straight path (*e.g.*, Fig. 1(*b*) and Fig. 1(*d*)), the position of all but the last node (*i.e.*, $j = 1, \dots, n_{Ni} - 1$) is determined through:

$${}^G\boldsymbol{x}_{Nji}^* = {}^G\boldsymbol{x}_{L_{ci}} + j\frac{z_{max}}{sf}(\cos\gamma_i, \sin\gamma_i), \quad (11)$$

where ${}^G\boldsymbol{x}_{L_{ci}} = \big({}^Gx_{L_{ci}}, {}^Gy_{L_{ci}}\big)$ is the position of the connectivity point considered, L_{ci}, and γ_i and n_{Ni} are the direction of the tether and the number of required nodes, determined through (12) and (13), respectively:

$$\gamma_i = \text{atan2}\big({}^Gy_{C_i} - {}^Gy_{L_{ci}}, {}^Gx_{C_i} - {}^Gx_{L_{ci}}\big), \text{ and} \quad (12)$$

$$n_{Ni} = \left\lceil \frac{l\big(H_i^*(L_{ci})\big)}{z_{max}/sf} \right\rceil, \quad (13)$$

where $\lceil\ \rceil$ is the ceiling function, $l\big(H_i^*(L_{ci})\big)$ is the length of the optimal tether, ${}^G\boldsymbol{x}_{C_i} = \big({}^Gx_{C_i}, {}^Gy_{C_i}\big)$ is the position of the swarm's local frame, ${}^CF_i$, and z_{max} and sf are the maximum sensing range of the mobile sensors, and a safety factor, respectively. The last node of the tether (*i.e.*, $j = n_{Ni}$) is placed at the swarm's local frame:

$${}^G\boldsymbol{x}_{Nji}^* = {}^G\boldsymbol{x}_{C_i}. \quad (14)$$

Our empirical investigations, as detailed in Appendix A, validated that the number of nodes does not affect the localization error of the tether. By minimizing the number of nodes, additional available mobile sensors can be placed in locations that provide the most flexibility to the swarm.

(iv) Intermediate Tethers for Maintaining Connectivity

The optimization process, described above, may choose to change the swarm's connectivity point at different configurations along its PTP path. In order to achieve this, while providing uninterrupted connectivity as the swarm moves from one configuration to the next, the proposed strategy forms an intermediate tether that connects the swarm to the selected connectivity point, before the swarm begins its motion.

For example, Fig. 5(*a*)-(*d*) shows the motion of the tethers as the swarm moves to the first two desired configurations along its PTP path. The swarm connects to home base at the first desired configuration, $L_{c1}^* = L_0$, Fig. 5(*b*), but changes its connectivity to a landmark for the second configuration, $L_{c2}^* = L_1$, Fig. 5(*d*). Due to this change in connectivity, an intermediate tether, connecting the first desired configuration, C_1, to the landmark selected for the second desired configuration, L_{c2}^*, is formed, $H_2'^*(L_{c2}^*)$, Fig. 5(*c*). During motion execution, this intermediate tether is formed before the swarm begins its motion to the second desired configuration, allowing the swarm to maintain uninterrupted connectivity.

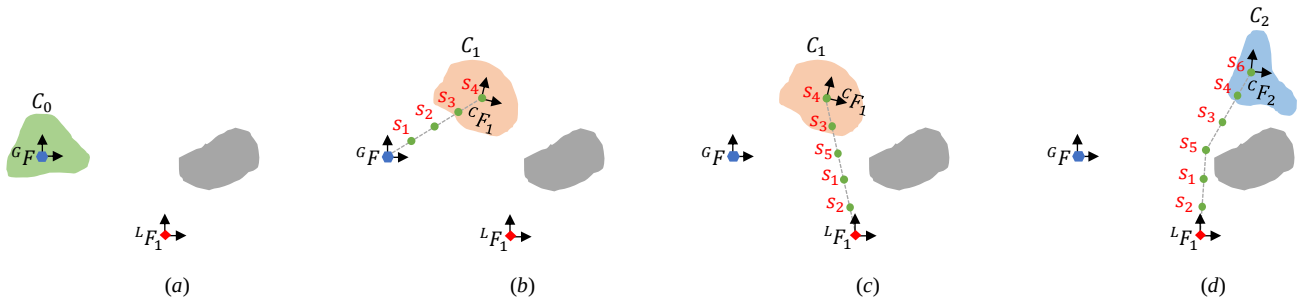
Once all intermediate tethers are formed, the ordered set of optimal tethers to be achieved by the mobile sensors, $\boldsymbol{H}^*$, is determined. For the example at hand, the ordered set would be denoted as $\boldsymbol{H}^* = \Big(H_1^*(L_{c1}^*), H_2'^*(L_{c2}^*), H_2^*(L_{c2}^*), \dots, H_{n_C}^*\big(L_{cn_C}^*\big)\Big)$.

2) Sensor Allocation

Mobile sensors need to be allocated to the nodes of the tethers – one sensor per node, to ensure wireless sensing connectivity between the swarm and the connectivity point. Herein, sensors are allocated with the objective of achieving the formed tethers accurately – minimizing the total travel distance of each sensor. Since mobile sensors are subject to motion errors that accumulate depending on their traveled distance, by minimizing the total travel distance of each sensor, the accumulated motion errors are minimized.

The sensor-allocation step must search through a discrete space, with n_{sol} number of solutions:

$$n_{sol} = \prod\nolimits_{H_k \epsilon \boldsymbol{H}^*} n_S P n_{NH_k}, \quad (15)$$



(*a*) (*b*) (*c*) (*d*)

Fig. 5. (*a*) Initial swarm configuration, (*b*) Tether for C_1, $H_1^*(L_{c1}^*)$, (*c*) intermediate tether for C_2, $H_2'^*(L_{c2}^*)$, (*d*) tether for C_2, $H_2^*(L_{c2}^*)$, and the sensors allocated to them.

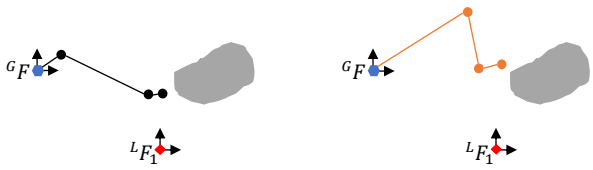
where n_{NH_k} is the number of nodes in tether H_k of the optimally formed tethers found in the tether formation step, $\boldsymbol{H}^*$, and P is the permutation operator. This space can be searched through a combinatoric optimization search engine, such as Genetic Algorithms [84].

Fig. 5 shows the sensors allocated to the tethers planned for the first two configurations of the PTP path. The minimum number of required sensors were considered for this example, determined based on the length of the longest tether, $H_{nc}^*(L_{cnc}^*)$, Fig. 1(*d*).

Heuristic for enhanced scalability of sensor allocation: The allocation of mobile sensors to the tether nodes is a combinatoric optimization problem, which may become computationally intensive with an increase in the number of mobile sensors and/or tether nodes. Herein, a heuristic approach to reducing the size of the sensor allocation search space is presented to address this issue.

The proposed heuristic reduces the size of the sensor allocation search space by simplifying the allocation of sensors to nodes of tethers where the risk of losing connectivity is low. Namely, when the swarm is not changing connectivity during its motion from one configuration to the next (*i.e.*, Fig. 5(*a*)-(*b*)), the risk of losing connectivity to the landmark is low. This is true since the safety factor for tether formation can be used to place multiple sensors within range of the connectivity point. If one sensor loses connectivity due to accumulated motion errors, others can still sense the connectivity point. For these scenarios, sensor allocation can be simplified to maintaining the same order of sensors along consecutive tethers that are connected to the same connectivity point.

However, the risk of losing connectivity is relatively higher when the swarm is planned to disconnect from its previous connectivity point and form an intermediate tether to its connectivity point for the next configuration (*i.e.*, Fig. 5(*b*)-(*d*)). Thus, it is important to allocate sensors to these intermediate tethers to minimize their travel distance and, thus, their accumulated motion errors, from their positions on the previous tether. Doing so would allow sensors to be allocated to ensure wireless connectivity during execution, while enhancing the scalability of sensor allocation, as the number of possible solutions is significantly reduced.



(*a*) (*b*)

Fig. 6. First three points on the PTP path of (*a*) S_1 and (*b*) S_5.

3) Sensor-Motion Planning

The proposed swarm-motion strategy requires that all sensors, whether they are assigned to the latest tether at hand or not, move synchronously with the swarm. For example, in Fig. 5(*a*)-(*b*), as the swarm moves from C_0 to C_1, all the six sensors (available to the swarm) must move to keep up with the swarm: Sensors 1 to 3 are dropped at their respective positions along the way, forming Tether $H_1^*(L_{C1}^*)$, while Sensor 4 remains with the swarm to provide the required connectivity. The unallocated sensors, Sensors 5-6 travel with the swarm. The tether remains stationary while the swarm is localized and closed-loop corrective steps are performed to improve its position at C_1.

Next, for the example at hand, before the swarm begins its motion from C_1 to C_2, all the sensors move to their positions on the intermediate tether, connecting C_1 to the landmark that the swarm connects to at C_2, L_{c2}^*. Sensors 1, 2, 3, and 5 move to their respective positions on the intermediate tether, $H_2'^*(L_{c2}^*)$, while Sensor 4 remains at the centroid of the swarm. The unallocated sensor, Sensor 6, also remains at the centroid of the swarm to allow for swarm-motion re-planning. The swarm, then, begins its motion to C_2, while the sensors move, synchronously, to $H_2^*(L_{c2}^*)$. This is followed by closed-loop control corrective steps.

Per the formulation detailed above, as the final step of tether planning, the desired (PTP) path of each sensor, P_{S_k}, is determined based on the corresponding tethers, and the sensors allocated to them. The initial position of each sensor at home base is set as the origin, ${}^G\boldsymbol{x}_{S_{k0}} = (0,0)$. The mobile sensors' subsequent positions are set to the positions of the nodes that they are allocated to on the ordered tethers, while unallocated sensors are positioned at the centroid of the swarm. Fig. 6(*a*)-(*b*) illustrates the first three points of the PTP path of S_1 and S_5, respectively, for the example at hand.

B. Swarm-Motion Control

The PTP path of the swarm, $\boldsymbol{C}$, and the planned PTP paths of all mobile sensors, $\{P_{S_k}\}_{k=1}^{n_S}$, are synchronized via three stages: *tether-motion*, *initial swarm-motion*, and *closed-loop swarm-motion*. The three-stage procedure is repeated until all swarm configurations on the PTP path are achieved.

At any time during motion execution, if the swarm obtains new information that requires re-planning, motion execution is halted and swarm-motion re-planning is invoked, Section IV.C.

1) Tether-Motion

In the proposed strategy, the swarm begins its motion to its first desired configuration, C_1, while maintaining connectivity with the home base, L_0. Thereafter, however, the swarm may choose to change its connectivity point (*i.e.*, $L_{ci}^* \neq L_{c(i-1)}^*, i \geq 2$). In such a scenario, the tether-motion stage moves the sensors to an intermediate tether, connecting the current swarm configuration, C_{i-1}, to the connectivity point selected for the next configuration, L_{ci}^*.

During the tether-motion stage, the sensors complete motion commands that move them to their respective position on the intermediate tether, $H_i'^*(L_{ci}^*)$, while the swarm remains stationary. The swarm remains stationary during the tether-motion stage in order to act as a static reference, whose position is known with some degree of uncertainty. Namely, by remaining stationary, the swarm can be used by the mobile sensors to localize their positions, in case the desired connectivity is not established, as detailed in Section IV.C.

The sensors' motion commands are executed through an open-loop collaborative strategy, such as behavior-based [17]–[22], leader-follower [25]–[30], or virtual structure [23], [24] methods, that allows the sensors to maintain real-time connectivity with each other and with the swarm. Moreover, while executing their motion, the mobile sensors would interact with their local neighbors to avoid collisions. When collisions are detected, the mobile sensors would adjust their individual motion and re-plan based on onboard sensor data.

2) Initial Swarm-Motion

During their initial motion, the swarm robots move to their next configuration, C_i, while the sensors move to their positions on the corresponding tether, $H_i^*(L_{ci}^*)$. In this regard, a swarm configuration, comprising n_R robots, $\boldsymbol{R} = \{R_k\}_{k=1}^{n_R}$, is defined by the pose (*i.e.*, position and orientation) of the swarm's local frame, ${}^{L}F$, with respect to the global frame, ${}^{G}F$, (${}^{G}\boldsymbol{x}_C$ and ${}^{G}\theta_C$, respectively), and the positions of all member robots with respect to this local frame (*i.e.*, its topology, $T = \{{}^{C}\boldsymbol{x}_{R_k}\}_{k=1}^{n_R}$), Fig. 7:

$$C = ({}^{G}\boldsymbol{x}_C, {}^{G}\theta_C, T). \tag{16}$$

The position of a robot can be described in global coordinates, ${}^{G}\boldsymbol{x}_{R_k}$, through a simple transformation.

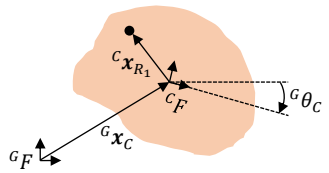


Fig. 7. Swarm configuration.

The initial motion of the swarm-sensor team is completed using any of the aforementioned open-loop collaborative strategies that allows the swarm to maintain real-time connectivity to the selected connectivity point through a tether of mobile sensors. The mobile sensors may use a similar strategy, executing motion commands that allow them to move in a tether formation to their designated positions, while providing real-time connectivity between the swarm and the selected landmark, and without impeding with the motion of the swarm. Both swarm robots and mobile sensors would depend on their local interactions for collision avoidance during motion.

One may note that connectivity is indirectly maintained between the swarm and the selected connectivity point through a mesh sensing network, where swarm robots and mobile sensors only sense their immediate neighbors, without requiring global sensing connectivity amongst all mobile sensors or the connectivity point.

3) Closed-Loop Swarm-Motion

The closed-loop swarm-motion stage aims to minimize the swarm-motion error, e_{mi}, at each desired configuration on the PTP path. Herein, swarm-motion error is defined as the average Euclidean distance between the true and desired position of all swarm robots:

$$e_{mi} = \frac{1}{n_R}\sum_{k=1}^{n_R} \left| {}^{G}\boldsymbol{x}_{R_k it} - {}^{G}\boldsymbol{x}_{R_k i} \right|, \tag{17}$$

where ${}^{G}\boldsymbol{x}_{R_k it}$ is the true position of Robot R_k, with respect to the global frame, after the swarm completes its motion commands to C_i, and ${}^{G}\boldsymbol{x}_{R_k i}$ is the desired position of this robot in C_i.

In practice, the swarm-motion error cannot be directly determined, as the true positions of all robots would be unknown. Thus, as an indirect measure, the proposed closed-loop swarm-motion stage aims to minimize the estimated swarm-motion error, $\hat{e}_{mi}$, calculated based on the estimated positions of all robots:

$$\hat{e}_{mi} = \frac{1}{n_R}\sum_{k=1}^{n_R} \left| {}^{G}\hat{\boldsymbol{x}}_{R_k i} - {}^{G}\boldsymbol{x}_{R_k i} \right|, \tag{18}$$

where ${}^{G}\hat{\boldsymbol{x}}_{R_k i}$ is an estimate of the true position of R_k. The estimated swarm-motion error, $\hat{e}_{mi}$, is a reasonable choice, as the proposed tether-based motion strategy allows the swarm to maintain bounded localization errors independent of the distance traveled by the swarm. Thus, the difference between $\hat{e}_{mi}$ and e_{mi} would also be bounded.

The closed-loop swarm-motion control stage at desired configuration C_i, is detailed in Algorithm 1. Upon reaching its desired configuration via the initial-motion stage, the swarm

begins by localizing based on the inter- and outer- robot measurements obtained from its member robots and mobile sensors. Herein, swarm localization is completed through the approach developed in our previous work [57], a review of which is summarized in Appendix B. Once localized, corrective motion commands are calculated and executed by the swarm robots for better approaching the desired configuration. The localization and corrective motion stages are repeated until the stopping criterion (*i.e.*, the desired swarm-motion error, e_{mmax}, or the maximum number of corrective motion steps, n_{cormax}) are met.

Algorithm 1 - Closed-Loop Swarm-Motion at C_i

1. Localize swarm → $\hat{C}_i$
2. Calculate estimated swarm-motion error → $\hat{e}_{mi}$
3. $n_{cor} = 1$
4. While $\hat{e}_{mi} > e_{mmax}$ and $n_{cor} \leq n_{cormax}$:
5. Calculate and execute corrective motion commands
6. Localize swarm → $\hat{C}_i$
7. Calculate estimated swarm-motion error → $\hat{e}_{mi}$
8. $n_{cor} = n_{cor} + 1$

One may note that the used localization method, detailed in Appendix B, requires transferring the sensor measurements obtained by individual swarm robots/mobile sensors to a central (*i.e.*, leader) robot for data fusion. This information transfer can be completed through a mesh communication network, where robots only communicate with their immediate neighbor, and transfer the information using communication protocols such as flooding [85].

For enhanced scalability, localization can be completed through a divide and conquer (DaC) strategy [40]. DaC divides the swarm into multiple smaller subswarms, each of which has a designated 'leader' robot. Localization is, then, completed through a hierarchical approach, where each subswarm is localized through a mesh network that only spans its own members. Once completed, the leaders of all subswarms create a mesh network within themselves and combine the localized information of their designated subswarms. This hierarchical approach enhances scalability as the number of robots in each subswarm is relatively smaller than the entire swarm, and information can be transferred/processed more efficiently.

C. Swarm-Motion Re-planning

At any time during motion execution, the swarm may be subject to circumstances that might not have been known or anticipated prior to deployment. In such cases, the formation of the tethers, for all remaining swarm configurations that have not been yet achieved, and the corresponding motion of the swarm and/or the mobile sensors may need to be re-planned.

Two example circumstances were considered in our work: (*i*) *encountering new obstacles*, whose positions and topology were unknown prior to the deployment of the swarm, that may prevent the mobile sensors from forming the original tethers, and (*ii*) *unsuccessful change in swarm connectivity* that may happen due to large error in motion execution.

In the former scenario, when unknown obstacles are encountered, the tethers must be re-planned to maintain the optimality of the formed tethers. Such re-planning would be completed using the tether planning process detailed in Section IV.A, through:

(i) Formation of new tethers, and/or

(ii) Use of additional/replacement redundant sensors, if available, and/or

(iii) Re-spacing of the sensors in the tethers, if no redundant sensors exist or replacement may not be feasible.

In the latter scenario, the mobile sensors may fail to obtain the desired connectivity to the swarm's next connectivity point (*i.e.*, are not within detection range of it) after the tether-motion stage, Section IV.B.1. Such an unsuccessful change in connectivity could be due to random errors in motion execution of the mobile sensors. In this scenario, the sensors may recover the desired connectivity in two stages:

(i) First, the sensors correct their positions through iterative localization and corrective motion based on the accurate static swarm location. Namely, as the swarm uses the tether to iteratively correct its position during the closed-loop swarm-motion stage, the mobile sensors use the stationary swarm as a temporary base to localize and correct their own locations, and

(ii) If such correction steps fail to obtain the desired connectivity, then, a search method is used to locate the landmark using, for example, a probabilistic estimate of the landmark's position relative to the swarm [86].

Both scenarios considered above would require the transfer of local sensing data, obtained by individual swarm robots/mobile sensors, to a central (*i.e.*, leader) robot for computation. This information transfer would be completed through a mesh communication network, where robots only communicate with their immediate neighbors.

V. Illustrative Experiments

Numerous experiments, with varying degree of difficulty, were conducted to evaluate the performance of the proposed tether-based motion control strategy. The results of some of these are presented herein.

The proximity sensing and motion characteristics of *mROBerTO* were used as guidance for the simulations [70]–[72]. Namely, all simulations were completed with sensing noise variance of $\sigma_{int}^2 = \sigma_{out}^2 = 2$, and a maximum sensing range of $z_{max} = 200$ mm. As noted in Section IV.A.1.i, the units for the sensing noise are millimeters and degrees for the distance and bearing measurements, respectively. Although the *mROBerTO* does not have the ability to sense its surrounding environment, it was assumed, herein, that outer-robot sensor measurements would be subject to the same uncertainty as the inter-robot sensor measurements.

In all examples presented herein, the swarm robots and mobile sensors are modelled with differential-drive locomotion

and execute their motion commands in all stages of swarm-motion control by (*i*) turning towards their destination and, then, (*ii*) moving along a straight line. Inter-robot collision avoidance was not considered in our work.

A. Motion with One Desired Swarm Configuration

Implementation of the proposed strategy is first discussed herein for a 10-robot swarm traveling from its starting configuration, C_0, to a different-topology configuration, C_1, at a distance of about 250 mm. For C_1, the optimal tether of three (mobile) sensors, extending from home base, L_0, was planned with $sf = 2$. The swarm's motion was, then, executed through the proposed strategy. Corrective closed-loop motion control was repeated until the stopping criterion of $n_{cormax} = 5$ is achieved.

Fig. 8 shows the swarm-motion error, e_{m1}, versus the corrective motion step for the simulated motion described above. The swarm motion-error, used in all the following subsections, is defined as the average Euclidean distance between the desired and true position of all swarm members (17). The results shown are the average of 100 distinct simulations, each subjected to different random noise in motion execution and proximity measurements. The bars on each point represent the upper and lower limits of the 99% confidence intervals for the presented simulations.

As expected, the results clearly illustrate the advantageous use of tether-based closed-loop motion control. The swarm's initial open-loop motion error (at correction step zero) is reduced through iterative corrective motion, where in each iteration, the swarm's true configuration moves closer to its desired. However, localization errors would prevent the swarm from reaching its desired configuration since corrective motion commands are calculated based on the (uncertain) estimate of the swarm's true configuration, and move the swarm to a configuration that is not equivalent to its desired. Thus, the swarm asymptotically reaches a steady-state motion error that is equal, in magnitude, to its localization error. Once steady-state is achieved, further correction steps are unnecessary, as they would not result in improved motion. One may note that in other examples, the number of corrective motion steps required to reach the steady-state motion error may be higher or lower than the one for the example presented herein.

B. A PTP Motion Path with Multiple Swarm Configurations

Herein, a detailed simulated example of the proposed tether-based motion control strategy is presented for a 10-robot swarm that was instructed to move on a PTP path, with $n_C = 8$ distinct desired configurations, Fig. 9. The swarm randomly changed its topology at each desired configuration, and moved directly away from home base, while operating in an environment with $n_L = 2$ distinct landmarks.

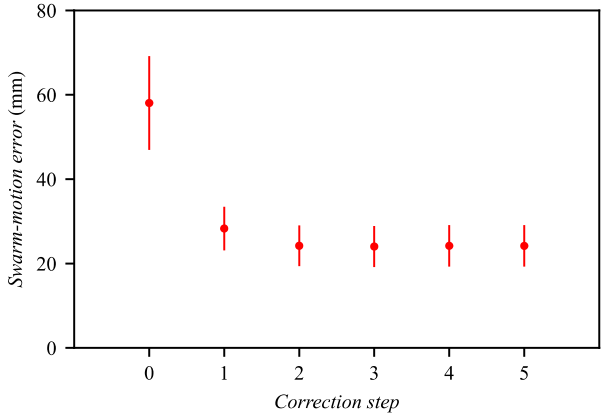


Fig. 8. Closed-loop motion control for one desired swarm configuration.

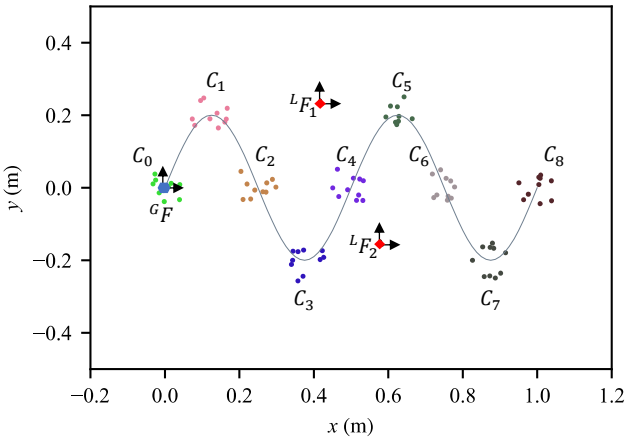


Fig. 9. PTP swarm path.

1) Tether Planning

For the chosen PTP path, the optimal tethers, and the allocation of mobile sensors to these, were determined through the strategy proposed in Section IV.A, with $sf = 2$. Due to the limited number of available sensors, $n_s = 5$, in this specific example, the swarm was connected to home base, L_0, for C_1 and C_2, to L_1 for configurations $C_3 - C_5$, and to L_2 for the remaining configurations, $C_6 - C_8$, respectively. The change in connectivity of the swarm along this PTP path required two intermediate tethers. Three example tethers, and the sensors allocated to these are shown in Fig. 10(*a*)-(*c*), respectively.

2) Swarm-Motion Control

The planned motion was completed through the swarm-motion control strategy detailed in Section IV.B. The implementation of this strategy required the swarm to change its connectivity, via the two intermediate tethers, through the tether-motion stage. Fig. 11 shows the swarm-motion error, e_{mi}, (17), for the open-loop control case (***green*** diamonds), closed-loop control case after the swarm completes the initial motion to its destination (***blue*** stars), and the closed-loop control case after several corrective steps (***red*** dots). The simulations were repeated 100 times with different random noise in motion execution and proximity measurements – the upper and lower limits of the 99% confidence intervals are shown through bars for each data point. Corrective closed-loop motion control of the swarm was repeated until the stopping criterion of $e_{mmax} = 1$ mm or $n_{cormax} = 5$ was achieved.

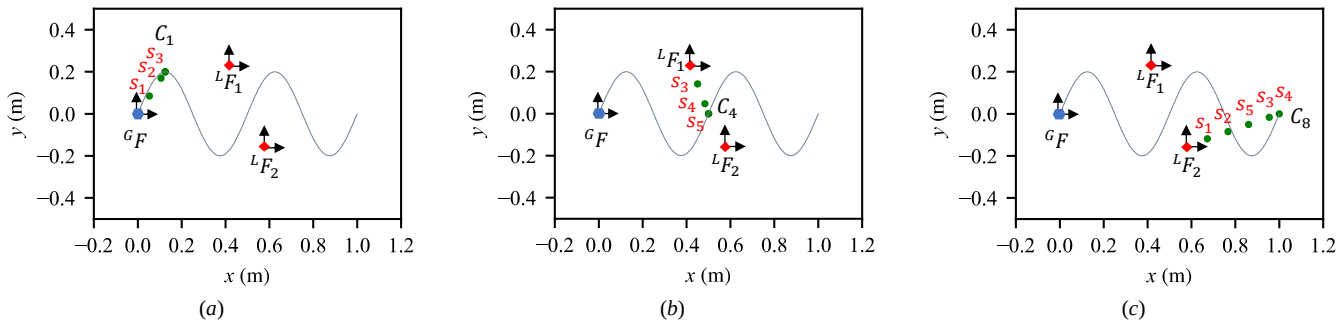


Fig. 10. Tethers (*a*) $H_1^*(L_{c1}^*)$, (*b*) $H_4^*(L_{c4}^*)$, and (*c*) $H_8^*(L_{c8}^*)$, and their allocated sensors.

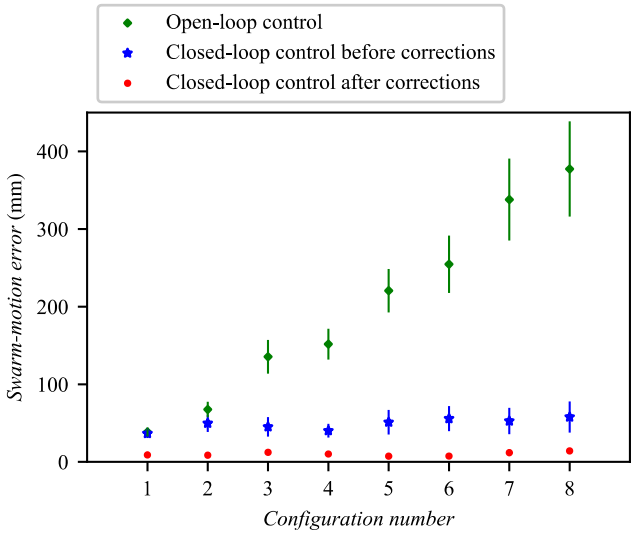


Fig. 11. Swarm motion error along the PTP path in Fig. 9.

C. *Physical Experiments*

In the physical experiments detailed herein, a swarm of four millirobots (*mROBerTOs)* is instructed to follow a desired PTP path – defined by a set of three distinct desired configurations, C_1 to C_3. Additional *mROBerTO* millirobots are used as mobile sensors to form the planned tethers. The *mROBerTO*, developed at the University of Toronto [70]–[72], is a millimeter-scale robot with a sensing module that allows it to accurately identify and measure the relative proximity and bearing of nearby robots through IR communication, Fig. 12.

In the detailed experiment, the swarm maintained connectivity to the home base, represented by a static *mROBerTO* millirobot, at all configurations. Namely, passive landmarks, as well as obstacles, were not included in the experimental setup due to the limitation of *mROBerTO* in not having effective onboard outer-robot proximity sensors capable of detecting these. Moreover, the experiments were conducted through an architecture with a centralized component, where the offline tether planning stage and the online swarm localization step were completed in a centralized manner, while motion calculation, motion execution, obtaining local sensor measurements, and transferring these measurements to the central coordinator were completed individually by each robot in a decentralized manner.

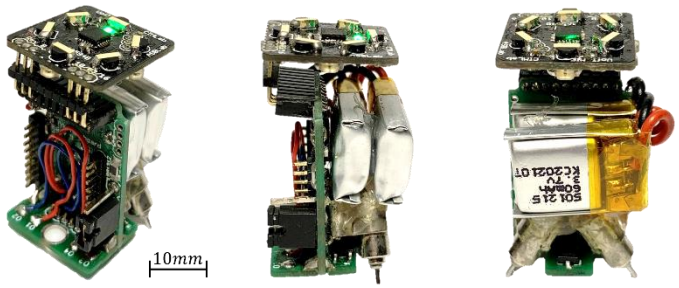


Fig. 12. *mROBerTO* millirobot.

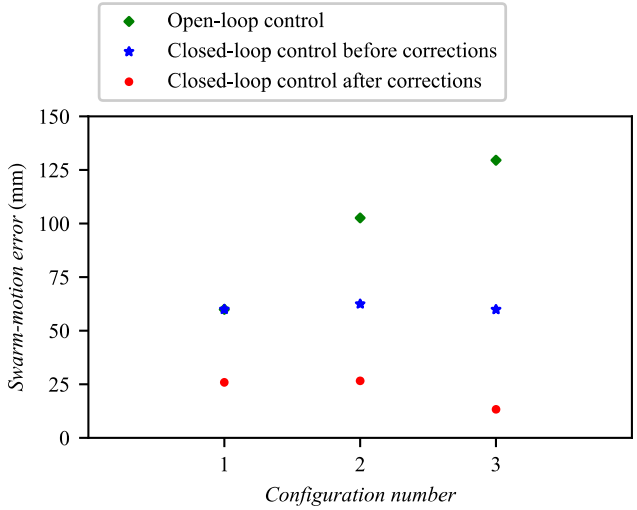


Fig. 13. Swarm motion error for each configuration shown in Fig. 14.

Fig. 13 depicts the swarm-motion error, e_{mi} (17), for the open-loop control case (***green*** diamonds), closed-loop control case after the swarm completes the initial motion to its destination (***blue*** stars), and the closed-loop control case after several corrective steps (***red*** dots). Corrective closed-loop motion control of the swarm was repeated until the stopping criterion of $e_{mmax} = 1$ mm or $n_{cormax} = 5$ were achieved. As expected, the swarm-motion error remains bounded at each successive configuration due to effective closed-loop motion control.

Fig. 14, in turn, illustrates the swarm robots' desired (***blue*** dots) and final true (***red*** stars) positions for each of the three configurations, as well as each tether (***green*** x's). All tethers were planned with $sf = 2$.

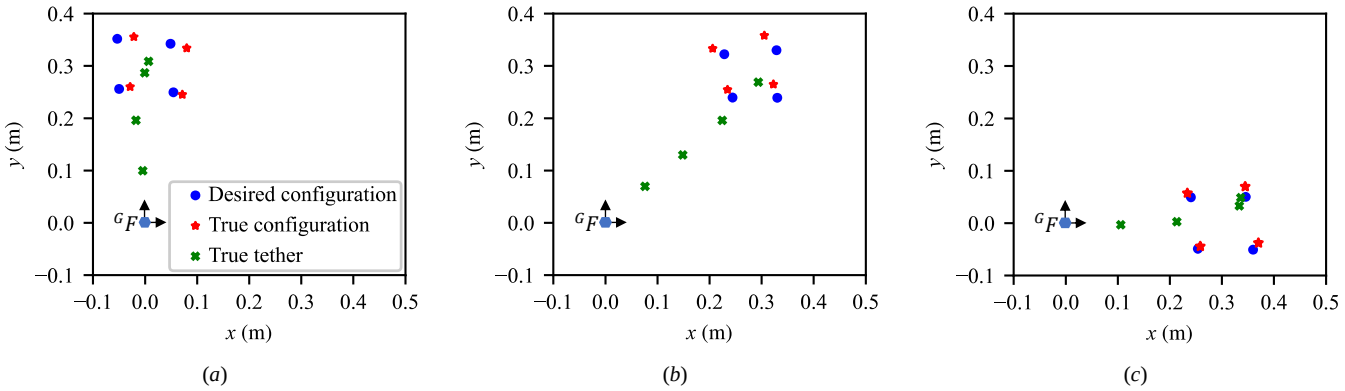


Fig. 14. Swarm configurations after corrective steps for (*a*) C_1, (*b*) C_2, and (*c*) C_3, respectively.

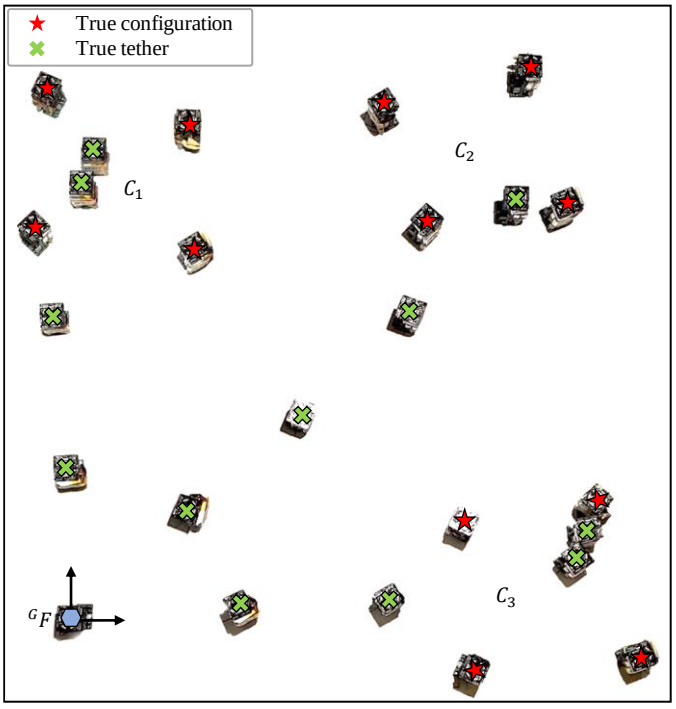


Fig. 15. Superposition of the swarm robots and tethers (in the experiments) after correction steps for all configurations shown in Fig. 14.

Fig. 15 shows the three configurations of the swarm and the corresponding tethers (in the experiments) after completing the necessary correction steps (as illustrated in Fig. 14 for each configuration). A video of an example experiment can be found at https://youtu.be/24WzlfeJWNQ.

VI. Comparative Studies

The proposed tether-based motion control strategy is compared below to alternative approaches. In Section VI.A, it is compared to an approach of using static sensors in forming the tethers. In Section VI.B, a statistical comparison of the proposed strategy to open-loop swarm-motion control is presented.

A. Use of Static Sensors in Tether Formation

The proposed strategy was compared to the alternative approach of using static sensors, that are dropped off as the swarm travels and remain stationary, though, still providing connectivity between the swarm and the home base.

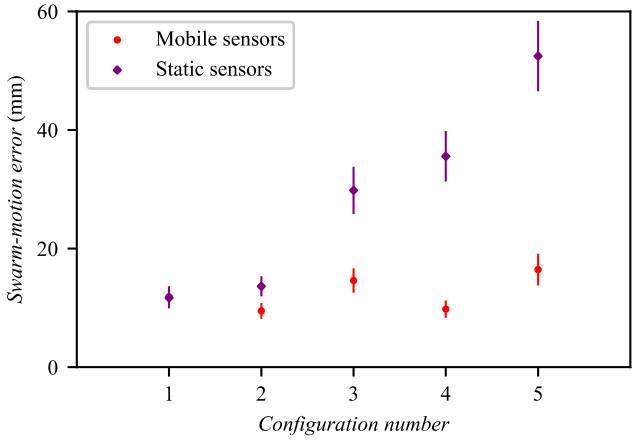


Fig. 16. Swarm-motion error along the PTP path – static versus mobile sensors.

The planned swarm-motion, using static tethers, was simulated through the strategy detailed in Section IV.B for a PTP path with $n_C = 5$ desired configurations that randomly changed topology, and translated between 200 to 300 mm between successive configurations. The PTP path was generated within a bounded radius around the origin (*i.e.*, $(0,0)$) in an environment without any landmarks. As above, once the swarm reached a desired configuration, it completed iterative corrective motion steps to minimize its motion errors.

Fig. 16 shows the swarm-motion error, e_{mi}, (17), versus configuration number: the ***purple*** diamonds correspond to the corrected motion of the swarm using tethers with static sensors, and the ***red*** dots correspond to the corrected motion of the swarm using mobile sensors. The results are the average of 100 simulations, each of which simulated the motion of the swarm along the same PTP path, but with different random noise in motion execution and proximity measurements. The lower and upper limits of the 99% confidence interval are shown as bars for each data point. The correction stopping criterion were set to $e_{mmax} = 1$ mm and $n_{cormax} = 5$.

For this specific example considered, the static sensors yielded higher motion errors than did mobile sensors, while also requiring more sensors. This outcome is due to the fact that tethers created by the static sensors are longer than optimally planned ones, resulting in increased motion errors due to successively increasing tether lengths.

B. Open-loop versus Closed-loop Control

The proposed tether-based motion strategy was also compared to the open-loop control strategy through a statistical analysis. For this purpose, a series of simulations were conducted for a swarm of $n_R = 10$ robots, completing a PTP path with n_C desired configurations that randomly changed topology, and translated between 200 to 300 mm between successive configurations. As in the previous sub-section, the PTP path was generated within a bounded radius around the origin (*i.e.,* $(0,0)$) in an environment without any landmarks. The swarm-motion performance for each simulation was, then, calculated as the total swarm-motion error, normalized with respect to the total average travel distance:

$$p_m = \frac{\sum_{i=1}^{n_C} e_{mi}}{\sum_{i=1}^{n_C} \overline{m}_i}, \tag{19}$$

where $\overline{m}_i$ is the average travel distance of the swarm from C_{i-1} to C_i:

$$\overline{m}_i = \frac{1}{n_R} \sum_{k=1}^{n_R} \left| {}^G\boldsymbol{x}_{R_k it} - {}^G\boldsymbol{x}_{R_k (i-1) t} \right|. \tag{20}$$

The results of 500 random simulations for PTP paths with $n_C = 5$ desired configurations are shown in Table I. In these simulations, the swarm and the sensors were subjected to random noise in their proximity measurements and motion command executions. Comparison of the results shows that the proposed tether-based motion strategy outperforms the open-loop control strategy with a confidence level of at least 99%.

TABLE I. STATISTICAL ANALYSIS – RESULTS.

	Swarm-motion performance, p_m		*Population-mean, μ, 99% confidence interval*
Strategy	*Sample-mean*	*Sample-Std-dev*	
Closed-loop	0.074	0.037	[0.070 – **0.078**]
Open-loop	0.659	0.436	[**0.608** – 0.709]

VII. CONCLUSIONS

This paper presents a novel tether-based strategy that achieves closed-loop swarm-motion control with bounded and minimum motion errors. The proposed strategy deploys a team of mobile sensors that travel with the swarm and provide it with extended sensing capabilities by forming tethers to a home base or landmarks in its surrounding environment. Through such tethers, the swarm obtains proximity measurements that can be used to localize and compensate for the accumulated motion errors. Thus, allowing the swarm to achieve bounded motion errors with respect to its traveled distance.

Tethers are optimized through a dynamic decision-making process (*a*) to choose whether to connect to home base or to a different landmark, and (*b*) to change their shape, in order to use the least possible of resources and to minimize swarm localization errors. The swarm-sensor team can also re-plan its motion online, in cases of unexpected circumstances. These include, for example, encountering *a priori* unknown physical obstacles and/or loss of connectivity within the team.

The proposed strategy presents the first use of wireless sensor tethers that allow the swarm to achieve effective closed-loop motion control. It is novel as it decouples the tether-planning problem from the motion of the swarm – allowing the swarm's desired motion to be planned, by an external motion planner, for the objectives of the application considered. Furthermore, the motion of the swarm can be executed using any existing motion control method. This decoupled nature also promotes scalability, as the computational requirements of the tether planning problem is independent of the number of robots in the swarm.

The proposed tether-based motion control strategy was validated through extensive simulations, and physical experiments using the *mROBerTO* millirobot.

APPENDIX A – LOCALIZATION PERFORMANCE OF TETHERS

A series of simulations were conducted to evaluate the effect of (*i*) the length of the tether, (*ii*) its connectivity point, and (*iii*) the number of nodes on its localization performance.

In each of the simulations, a straight-line tether with length l and direction γ was generated. The tether was either connected to the home base, represented by a static sensor, or a landmark, both of which were placed at the origin, ${}^G F$. Moreover, the n_N nodes of the tether were placed at equal intervals along the tether; the distance between nodes was calculated as the length of the tether divided by the number of nodes. The sensors allocated to the nodes were, then, initialized at the origin, and instructed to execute motion commands to their designated positions. Once completed, sensor measurements were simulated, and used to localize the mobile sensors through the strategy developed in our previous work [57], summarized in Appendix B. The localization error of each mobile sensor S_k, e_{lS_k}, was, then, calculated as the Euclidean distance between its true and estimated position:

$$e_{lS_k} = \left| {}^G\boldsymbol{x}_{S_k t} - {}^G\hat{\boldsymbol{x}}_{S_k} \right|, \tag{A1}$$

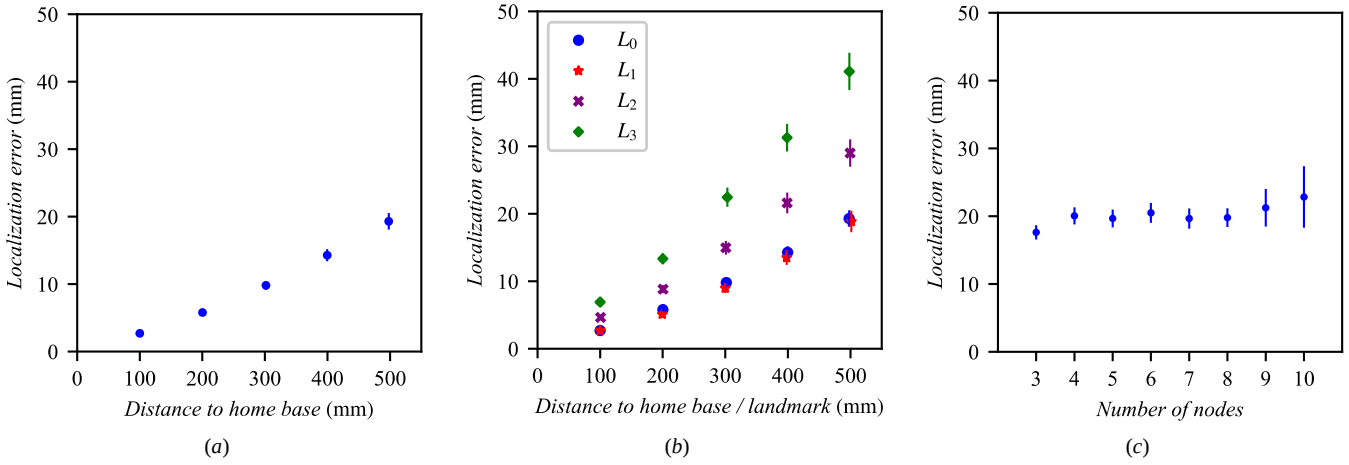
where ${}^G\boldsymbol{x}_{S_k t}$ and ${}^G\hat{\boldsymbol{x}}_{S_k}$ are, respectively, the true and estimated position of sensor S_k.

A. The Effect of Tether Length

It is conjectured that localization errors accumulate along the tether, starting at the first node, closest to the connectivity point, and reaching their maximum at the last (tip) node located at the swarm. Thus, longer tethers have higher localization errors.

1,000 different simulations were conducted for a straight-line tether of length $l = 500$ mm that was connected to the home base and was formed with $n_N = 5$ nodes. The simulations differed in the randomly generated direction of the formed tether, and the noise in motion execution and the sensor measurements of the mobile sensors.

Fig. A1(*a*) shows the localization error of each sensor versus its distance to the home base, averaged over all simulations. The lower and upper limits of the 99% confidence interval are shown as bars for each data point. The results validate the conjecture that localization errors accumulate along the tether and, thus, longer tethers have larger localization errors.



(*a*) (*b*) (*c*)

Fig. A1. The effect of (*a*) tether length, (*b*) connectivity point, and (*c*) number of nodes on the localization performance of the tether.

B. The Effect of Connecting to Landmarks

It is further conjectured that connectivity to a landmark may affect the localization error of the tether as, due to the difference in the required sensing technology, landmarks may be perceived with higher uncertainty than the home base.

Four sets of 1,000 different simulations were conducted for straight-line tethers of length $l = 500$ mm, that were formed with $n_N = 5$ nodes. The 1,000 simulations in each set differed in the random direction of the formed tether and the motion execution and sensing noise that the mobile sensors were subjected to. Furthermore, each set of simulations differed in the connectivity point of the tether – in the first set of simulations the tethers were connected to home base, L_0, that was sensed through inter-robot sensor measurements with noise variance of σ_{int}^2, Section IV.A.1.i. In the second, third, and fourth sets of simulations, the tethers were connected to Landmark L_1, L_2, and L_3, respectively, that were sensed through outer-robot sensor measurements with noise variance of $\sigma_{outL_1}^2 = \sigma_{int}^2$, $\sigma_{outL_2}^2 = 2\sigma_{int}^2$, and $\sigma_{outL_3}^2 = 3\sigma_{int}^2$, respectively.

Fig. A1(*b*) shows the localization error of each sensor versus its distance to the connectivity point, for the four sets of simulations detailed above. The lower and upper limits of the 99% confidence interval are shown as bars for each data point. The results validate the conjecture that, due to the difference in sensing technology, the localization error of the tether may accumulate at a higher rate when connected to a landmark. The rate of error accumulation, δ, can be determined empirically.

C. The Effect of the Number of Nodes

Finally, it is conjectured that the localization error of a tether is independent of the number of nodes. Namely, since localization errors accumulate along the tether with respect to the distance of the sensor to the connectivity point, then, increasing the number of nodes would not impact the localization error of the tether.

Eight sets of 1,000 different simulations were conducted for straight-line tethers of length $l = 500$ mm, that were connected to home base. The 1,000 simulations in each set differed in the random direction of the formed tether and the motion execution and sensing noise that the mobile sensors were subjected to. Each set of simulations differed in the number of nodes used to form the tether, ranging from $n_N = 3$ to $n_N = 10$.

Fig. A1(*c*) shows the localization error of the tip sensor (*i.e.*, the sensor allocated to the last node, furthest from the home base), (A1), versus the number of nodes used to form the tether, for the simulations described above. The lower and upper limits of the 99% confidence interval are shown as bars for each data point. The results validate that the localization error of the tip sensor does not change with increased number of nodes.

APPENDIX B – SWARM LOCALIZATION

The swarm's configuration is estimated during the closed-loop motion stage of the proposed strategy. This localization process is completed by (*i*) merging the swarm robots and mobile/static sensors into one team, and (*ii*) estimating their configuration, with respect to the global frame, using the swarm localization method first introduced in our previous works [57]. The configuration of the swarm, and the estimated position of all mobile sensors are, then, extracted.

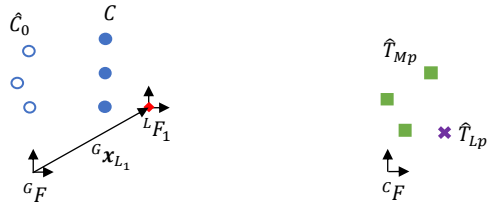
Based on the above, the utilized localization method is detailed, herein, for an arbitrary team with of n_M members, $\boldsymbol{M} = \{M_k\}_{k=1}^{n_M}$, moving from its pre-motion estimated configuration, $\hat{C}_0$, to its desired configuration, C. The localization method includes two phases: (*i*) *data acquisition*, and (*ii*) *data fusion*.

A. Data Acquisition

In the data acquisition stage, (*i*) a preliminary estimate of the configuration of the team and (*ii*) a secondary estimate of the topology of the team and the surrounding landmarks are obtained.

1) Preliminary Estimate of the Team Configuration

A preliminary estimate of the configuration of the team can be obtained by assuming that (*i*) their pre-motion estimated configuration is accurate, and (*ii*) their motion commands were completed without any errors. With these assumptions, a preliminary estimate of the configuration of the team based on motion commands, $\hat{C}_u$, can be calculated as the desired configuration, C.



(*a*) (*b*)

Fig. B1. (*a*) Estimated team configuration based on motion commands, and (*b*) estimated team topology based on sensor measurements.

Due to the noise in motion execution, a measure of the uncertainty in the estimated position of each member, $\lambda_{M_i u}$, can also be obtained. This uncertainty is dependent on the magnitude of the motion command executed by each member. For example, the static sensor at home base would have negligible uncertainty, while the swarm robots and mobile sensors would have higher uncertainty.

Fig. B1(*a*) shows an example preliminary estimate obtained for a team with 3 members, moving near a landmark, L_1, whose position, ${}^{G}\boldsymbol{x}_{L_1}$, is known. The team moves from its initial estimated configuration, $\hat{C}_0$ (hollow ***blue*** dots) to its desired configuration, C (filled ***blue*** dots).

2) Secondary Estimate of the Team and Landmark Topologies

A secondary estimate of the topology of the team, $\hat{T}_{Mp}$, and the surrounding landmarks, $\hat{T}_{Lp}$, can be obtained by clustering the inter-/outer- member proximity measurements. The topology of the team and the landmarks represents their positions with respect to a local swarm frame, ${}^{C}F$. The proximity measurements considered include the distance and bearing between neighboring members and surrounding landmarks, Section IV.A.1.i. The topology estimation process also calculates the uncertainty in the estimated local position of each member, $\lambda_{M_i p}$, and landmark, $\lambda_{L_i p}$.

Fig. B1(*b*) illustrates an example of the estimated topology of the merged team (***green*** squares), and the surrounding landmark (***purple*** x) with respect to the local frame ${}^{C}F$, after executing its motion commands to the desired position in Fig. B1(*a*).

B. Data Fusion

The data fusion stage determines the position of the estimated topologies, with respect to the global frame, by superimposing them onto the preliminary estimated team configuration, and the known position of the sensed landmarks, respectively. Namely, the data fusion step aims to determine the pose of the local frame, $({}^{G}\boldsymbol{x}_C, {}^{G}\theta_C)$, that minimizes the weighted distance between corresponding member positions in $\hat{C}_u$ and $\hat{T}_{Mp}$, and the corresponding landmark positions in the *a priori* known map, $\{{}^{G}\boldsymbol{x}_{L_i}\}_{i=1}^{n_L}$, and $\hat{T}_{Lp}$.

The weights of superimposition are selected to reflect the uncertainty in the estimates obtained through the data acquisition stage. This approach results in landmarks and the static sensor at home base to be weighted higher and, consequently, to have a higher impact on the estimated position of the local frame.

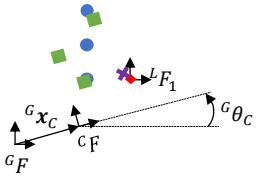


Fig. B2. Superimposition.

Fig. B2 illustrates the topology shown in Fig. B1(*b*) superimposed onto the swarm configuration in Fig. B1(*a*).

Finally, the position of each member is estimated as the weighted average of its estimated position in $\hat{C}_u$, and its estimated position in $\hat{T}_{Mp}$ translated into global coordinates based on the determined pose of the local frame.

ACKNOWLEDGMENT

The authors thank Darie Roman for his input on the tether-based motion strategy.

REFERENCES

[1] M. Brambilla, E. Ferrante, M. Birattari, and M. Dorigo, "Swarm robotics: A review from the swarm engineering perspective," *Swarm Intell.*, vol. 7, no. 1, pp. 1–41, Mar. 2013.

[2] E. Şahin, "Swarm robotics: From sources of inspiration to domains of application," in *Swarm Robotics*, vol. 3342, E. Şahin and W. M. Spears, Eds. Berlin, Heidelberg: Springer, 2004, pp. 10–20.

[3] M. Schranz, M. Umlauft, M. Sende, and W. Elmenreich, "Swarm robotic behaviors and current applications," *Front. Robot. AI*, vol. 7, 2020.

[4] L. Yang, J. Yu, S. Yang, B. Wang, B. J. Nelson, and L. Zhang, "A survey on swarm microrobotics," *IEEE Trans. Robot.*, pp. 1–21, Oct. 2021.

[5] M. Abdelkader, S. Güler, H. Jaleel, and J. S. Shamma, "Aerial swarms: Recent applications and challenges," *Curr. Robot. Rep.*, vol. 2, no. 3, pp. 309–320, Sep. 2021.

[6] S. Shahrokhi, L. Lin, and A. T. Becker, "Planar orientation control and torque maximization using a swarm with global inputs," *IEEE Trans. Autom. Sci. Eng.*, vol. 16, no. 4, pp. 1980–1987, Oct. 2019.

[7] S. Nagavalli, N. Chakraborty, and K. Sycara, "Automated sequencing of swarm behaviors for supervisory control of robotic swarms," in *IEEE Int. Conf. Robot. Autom.*, Singapore, May 2017, pp. 2674–2681.

[8] A. Dirafzoon, A. Bozkurt, and E. Lobaton, "A framework for mapping with biobotic insect networks: From local to global maps," *Robot. Auton. Syst.*, vol. 88, pp. 79–96, Feb. 2017.

[9] A. Dirafzoon and E. Lobaton, "Topological mapping of unknown environments using an unlocalized robotic swarm," in *IEEE/RSJ Int. Conf. Intel. Robot. Syst.*, Tokyo, Japan, Nov. 2013, pp. 5545–5551.

[10] R. Ramaithitima, M. Whitzer, S. Bhattacharya, and V. Kumar, "Automated creation of topological maps in unknown environments using a swarm of resource-constrained robots," *IEEE Robot. Autom. Lett.*, vol. 1, no. 2, pp. 746–753, Jul. 2016.

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

[12] S. Shahrokhi, L. Lin, C. Ertel, M. Wan, and A. T. Becker, "Steering a swarm of particles using global inputs and swarm statistics," *IEEE Trans. Robot.*, vol. 34, no. 1, pp. 207–219, Feb. 2018.

[13] P. Walker, S. A. Amraii, M. Lewis, N. Chakraborty, and K. Sycara, "Human control of leader-based swarms," in *IEEE Int. Conf. Syst., Man, Cybern.*, Manchester, UK, Oct. 2013, pp. 2712–2717.

[14] D. P. Stormont, "Autonomous rescue robot swarms for first responders," in *Proc. IEEE Int. Conf. Comp. Intell. Home. Sec. Pers. Safe.*, Orlando, FL, USA, Mar. 2005, pp. 151–157.

[15] M. L. Elwin, R. A. Freeman, and K. M. Lynch, "Distributed environmental monitoring with finite element robots," *IEEE Trans. Robot.*, vol. 36, no. 2, pp. 380–398, Apr. 2020.

[16] N. Nedjah and L. S. Junior, "Review of methodologies and tasks in swarm robotics towards standardization," *Swarm Evol. Comput.*, vol. 50, p. 100565, Nov. 2019.

[17] D. Izzo and Pettazzi, "Equilibrium shaping: Distributed motion planning for satellite swarm," in *Proc. Int. Symp. Art. Intell. Robot. Autom. Space*, Munich, Germany, Sep. 2005, pp. 1–10.

[18] V. Gazi, "Swarm aggregations using artificial potentials and sliding-mode control," *IEEE Trans. Robot.*, vol. 21, no. 6, pp. 1208–1214, Dec. 2005.

[19] J. R. T. Lawton, R. W. Beard, and B. J. Young, "A decentralized approach to formation maneuvers," *IEEE Trans. Robot. Autom.*, vol. 19, no. 6, pp. 933–941, Dec. 2003.

[20] D. Xu, X. Zhang, Z. Zhu, C. Chen, and P. Yang, "Behavior-based formation control of swarm robots," *Math. Probl. Eng*, vol. 2014, pp. 1–13, Jun. 2014.

[21] H. Zhao, H. Liu, Y.-W. Leung, and X. Chu, "Self-adaptive collective motion of swarm robots," *IEEE Trans. Autom. Sci. Eng.*, vol. 15, no. 4, pp. 1533–1545, Jun. 2018.

[22] H. Wang and M. Rubenstein, "Shape formation in homogeneous swarms using local task swapping," *IEEE Trans. Robot.*, vol. 36, no. 3, pp. 597–612, Feb. 2020.

[23] C.-C. Lin, K.-C. Chen, and W.-J. Chuang, "Motion planning using a memetic evolution algorithm for swarm robots," *Int. J. Adv. Robot. Syst.*, vol. 9, no. 1, p. 19, Mar. 2012.

[24] K. D. Do and J. Pan, "Nonlinear formation control of unicycle-type mobile robots," *Robot. Auton. Syst.*, vol. 55, no. 3, pp. 191–204, Mar. 2007.

[25] C. Bentes and O. Saotome, "Dynamic swarm formation with potential fields and A* path planning in 3D environment," in *Brazilian Robot. Symp. Latin American Robotic. Symp.*, Fortaleza, Brazil, Oct. 2012, pp. 74–78.

[26] T. Soleymani and F. Saghafi, "Behavior-based acceleration commanded formation flight control," in *Int. Conf. Control. Automat. Syst.*, Gyeonggi-do, South Korea, Oct. 2010, pp. 1340–1345.

[27] L. Consolini, F. Morbidi, D. Prattichizzo, and M. Tosques, "A geometric characterization of leader-follower formation control," in *IEEE Int. Conf. Robotic. Automat.*, Roma, Italy, Apr. 2007, pp. 2397–2402.

[28] H. Wang, D. Guo, X. Liang, W. Chen, G. Hu, and K. K. Leang, "Adaptive vision-based leader–follower formation control of mobile robots," *IEEE Trans. Ind. Electron.*, vol. 64, no. 4, pp. 2893–2902, Apr. 2017.

[29] K. Elamvazhuthi, Z. Kakish, A. Shirsat, and S. Berman, "Controllability and stabilization for herding a robotic swarm using a leader: A mean-field approach," *IEEE Trans. Robot.*, vol. 37, no. 2, pp. 418–432, Oct. 2021.

[30] V. S. Chipade and D. Panagou, "Multiagent planning and control for swarm herding in 2-D obstacle environments under bounded inputs," *IEEE Trans. Robot.*, vol. 37, no. 6, Dec. 2021.

[31] I. Vatamaniuk, G. Panina, A. Saveliev, and A. Ronzhin, "Convex shape generation by robotic swarm," in *Int. Conf. Auton. Rob. Syst. Comp.*, Bragança, Portugal, May 2016, pp. 300–304.

[32] M. P. Vicmudo, E. P. Dadios, and R. R. P. Vicerra, "Path planning of underwater swarm robots using genetic algorithm," in *Int. Conf. Humanoid, Nanotechnol., Info. Technol., Commun. Control, Environ. Manage.*, Puerto Princesa, Philippines, Nov. 2014, pp. 1–5.

[33] H.-X. Wei, Q. Mao, Y. Guan, and Y.-D. Li, "A centroidal Voronoi tessellation based intelligent control algorithm for the self-assembly path planning of swarm robots," *Expert Syst. Appl.*, vol. 85, pp. 261–269, Nov. 2017.

[34] Y.-T. Lee, S.-F. Zeng, and C.-S. Chiu, "Distributed path planning of swarm mobile robots," in *Asian Cont. Conf.*, Fukuoka, Japan, Jun. 2019, pp. 49–54.

[35] V. Izhboldina, E. Usina, and I. Vatamaniuk, "A*-based path planning algorithm for swarm robotics," in *Proc. Int. Conf. Inter. Collab. Robot.*, St Petersburg, Russia, Oct. 2020, pp. 107–115.

[36] B. Rivière, W. Hönig, Y. Yue, and S. Chung, "GLAS: Global-to-local safe autonomy synthesis for multi-robot motion planning with end-to-end learning," *IEEE Robot. Autom. Lett.*, vol. 5, no. 3, pp. 4249–4256, Jul. 2020.

[37] W. Hönig, J. A. Preiss, T. K. S. Kumar, G. S. Sukhatme, and N. Ayanian, "Trajectory planning for quadrotor swarms," *IEEE Trans. Robot.*, vol. 34, no. 4, pp. 856–869, Aug. 2018.

[38] D. Inoue, D. Murai, Y. Ikuta, and H. Yoshida, "Distributed range-based localization for swarm robot systems using sensor-fusion technique," in *Int. Conf. Sens. Net.*, Prague, Czech Republic, Feb. 2019, pp. 13–22.

[39] J. Klingner, N. Ahmed, and N. Correll, "Fault-tolerant covariance intersection for localizing robot swarms," *Distrib. Auton. Robot. Syst.*, vol. 122, p. 103306, Dec. 2019.

[40] J. Y. Kim, Z. Kashino, L. M. Pineros, S. Bayat, T. Colaco, G. Nejat, and B. Benhabib, "A high-performance millirobot for swarm-behaviour studies: Swarm-topology estimation," *Int. J. Adv. Robot. Syst.*, vol. 16, no. 6, pp. 1–18, Nov. 2019.

[41] S. Fukui and K. Naruse, "Swarm EKF localization for a multiple robot system with range-only measurements," in *Int. Symp. Syst. Integrat.*, Kobe, Japan, Dec. 2013, pp. 796–801.

[42] A. O. de Sá, N. Nedjah, and L. de M. Mourelle, "Multi-hop collaborative min-max localization," in *IEEE Latin American Symp. Circ. Syst.*, Montevideo, Uruguay, Feb. 2015, pp. 1–4.

[43] Lantao Liu, B. Fine, D. Shell, and A. Klappenecker, "Approximate characterization of multi-robot swarm 'shapes' in sublinear-time," in *IEEE Int. Conf. Robot. Autom.*, Shanghai, China, May 2011, pp. 2886–2891.

[44] D. Ma, M. J. Er, B. Wang, and H. B. Lim, "Range-free wireless sensor networks localization based on hop-count quantization," *Telecommun. Syst.*, vol. 50, no. 3, pp. 199–213, Jul. 2012.

[45] A. Kohlbacher, J. Eliasson, K. Acres, H. Chung, and J. C. Barca, "A low cost omnidirectional relative localization sensor for swarm applications," in *IEEE World Internet Things*, Singapore, Feb. 2018, pp. 694–699.

[46] A. Cornejo and R. Nagpal, "Distributed range-based relative localization of robot swarms," in *Algorithmic Foundations of Robotics XI*, 1st ed., vol. 107, H. L. Akin, N. M. Amato, V. Isler, and A. F. van der Stappen, Eds. Cham, Switzerland: Springer International Publishing, 2015, pp. 91–107.

[47] S. Zhang, E. Staudinger, S. Sand, R. Raulefs, and A. Dammann, "Anchor-free localization using round-trip delay measurements for martian swarm exploration," in *IEEE/ION Pos., Loc. Nav. Symp.*, Monterey, CA, USA, May 2014, pp. 1130–1139.

[48] A. O. de Sá, N. Nedjah, and L. de M. Mourelle, "Distributed and resilient localization algorithm for swarm robotic systems," *Appl. Soft Comput.*, vol. 57, pp. 738–750, Aug. 2017.

[49] L. C. Carrillo-Arce, E. D. Nerurkar, J. L. Gordillo, and S. I. Roumeliotis, "Decentralized multi-robot cooperative localization using covariance intersection," in *IEEE/RSJ Int. Conf. Int. Robot. Syst.*, Tokyo, Japan, Nov. 2013, pp. 1412–1417.

[50] S. I. Roumeliotis and I. M. Rekleitis, "Propagation of Uncertainty in Cooperative Multirobot Localization: Analysis and Experimental Results," *Auton. Robots*, vol. 17, no. 1, pp. 41–54, Jul. 2004.

[51] E. D. Nerurkar, S. I. Roumeliotis, and A. Martinelli, "Distributed maximum a posteriori estimation for multi-robot cooperative localization," in *IEEE Int. Conf. Robot. Autom.*, Kobe, Japan, May 2009, pp. 1402–1409.

[52] R. Kurazume, S. Nagata, and S. Hirose, "Cooperative positioning with multiple robots," in *IEEE Int. Conf. Robotic. Automat.*, San Diego, CA, USA, May 1994, pp. 1250–1257 vol.2.

[53] R. Kurazume and S. Hirose, "Study on cooperative positioning system: optimum moving strategies for CPS-III," in *Proc. IEEE Int. Conf. Robot. Autom.*, Leuven, Belgium, May 1998, vol. 4, pp. 2896–2903.

[54] L. E. Navarro-Serment, C. J. J. Paredis, and P. K. Khosla, "A beacon system for the localization of distributed robotic teams," in *Proc. Int. Conf. Field Serv. Robotic.*, Pittsburgh, PA, USA, 1999, pp. 232–237.

[55] S. Xu, Z. Ji, D. T. Pham, and F. Yu, "Simultaneous localization and mapping: Swarm robot mutual localization and sonar arc bidirectional carving mapping," *J. Mech. Eng. Sci.*, vol. 225, no. 3, pp. 733–744, Mar. 2011.

[56] A. G. Pires, D. G. Macharet, and L. Chaimowicz, "Towards Cooperative Localization in Robotic Swarms," in *Distributed Autonomous Robotic Systems.*, vol. 112, N. Y. Chong and Y. J. Cho, Eds. Tokyo, Japan: Springer, 2016, pp. 105–118.

[57] K. Eshaghi, Z. Kashino, H. J. Yoon, G. Nejat, and B. Benhabib, "An inchworm-inspired motion strategy for robotic swarms," *Robotica*, vol. 39, no. 12, pp. 2283–2305, Apr. 2021.

[58] S. Tully, G. Kantor, and H. Choset, "Leap-frog path design for multi-robot cooperative localization," in *Field and Service Robotics*, vol. 62, A. Howard, K. Iagnemma, and A. Kelly, Eds. Berlin, Heidelberg: Springer, 2010, pp. 307–317.

[59] N. Trawny and T. Barfoot, "Optimized motion strategies for cooperative localization of mobile robots," in *Proc. IEEE Int. Conf. Robot. Autom.*, New Orleans, LA, USA, Apr. 2004, vol. 1, pp. 1027–1032.

[60] K. Hattori, E. Homma, T. Kagawa, M. Otani, N. Tatebe, Y. Owada, L. Shan, K. Temma, and K. Hamaguchi, "Generalized measuring-worm algorithm: high-accuracy mapping and movement via cooperating swarm robots," *Artif. Life Robot.*, vol. 21, no. 4, pp. 451–459, Dec. 2016.

[61] W. Li, Z. Xiong, Y. Sun, and J. Xiong, "Cooperative positioning algorithm of swarm UAVs based on posterior linearization belief propagation," in *IEEE Inf. Tech., Network., Electron. Autom. Cont. Conf.*, Chengdu, China, Mar. 2019, pp. 1277–1282.

[62] Z. Song and K. Mohseni, "A distributed localization hierarchy for an AUV swarm," in *American Cont. Conf.*, Portland, OR, USA, Jun. 2014, pp. 4721–4726.

[63] I. Loefgren, N. Ahmed, E. Frew, C. Heckman, and S. Humbert, "Scalable event-triggered data fusion for autonomous cooperative swarm localization," in *Int. Conf. Info. Fusion*, Ottawa, ON, Canada, Jul. 2019, pp. 1–8.

[64] A. G. Pires, D. G. Macharet, and L. Chaimowicz, "Exploring heterogeneity for cooperative localization in swarm robotics," in *Int. Conf. Adv. Robotic.*, Istanbul, Turkey, Jul. 2015, pp. 407–414.

[65] C. Röhrig, J. Lategahn, M. Muller, and L. Telle, "Global localization for a swarm of autonomous transport vehicles using IEEE 802.15.4a CSS," in *Proc. Int. MultiConf. Eng. Comp. Sci.*, Hong Kong, Mar. 2012, vol. 2, pp. 828–833.

[66] T. Krajnik, M. Nitsche, J. Faigl, P. Vank, M. Saska, L. Peuil, T. Duckett, and M. Mejail, "A practical multirobot localization system," *J. Intell. Robot. Syst. Theory Appl.*, vol. 76, pp. 539–562, Apr. 2014.

[67] A. G. Pires, P. A. F. Rezeck, R. A. Chaves, D. G. Macharet, and L. Chaimowicz, "Cooperative localization and mapping with robotic swarms," *J. Intell. Robot. Syst. Theory Appl.*, vol. 102, no. 47, pp. 1–23, May 2021.

[68] H. Wu, S. Qu, D. Xu, and C. Chen, "Precise localization and formation control of swarm robots via wireless sensor networks," *Math. Probl. Eng.*, vol. 2014, pp. 1–12, Nov. 2014.

[69] S. Mayya, P. Pierpaoli, G. Nair, and M. Egerstedt, "Localization in densely packed swarms using interrobot collisions as a sensing modality," *IEEE Trans. Robot.*, vol. 35, no. 1, pp. 21–34, Feb. 2019.

[70] J. Y. Kim, Z. Kashino, T. Colaco, G. Nejat, and B. Benhabib, "Design and implementation of a millirobot for swarm studies – mROBerTO," *Robotica*, vol. 36, no. 11, pp. 1591–1612, Nov. 2018.

[71] J. Y. Kim, T. Colaco, Z. Kashino, G. Nejat, and B. Benhabib, "mROBerTO: A modular millirobot for swarm-behavior studies," in *Int. Conf. Intel. Robot. Syst.*, Daejeon, Korea, Oct. 2016, pp. 2109–2114.

[72] K. Eshaghi, Y. Li, Z. Kashino, G. Nejat, and B. Benhabib, "mROBerTO 2.0 – An autonomous millirobot with enhanced locomotion for swarm robotics," *Robot. Autom. Let*, vol. 5, no. 2, pp. 962–969, Apr. 2020.

[73] M. Rubenstein, C. Ahler, and R. Nagpal, "Kilobot: A low cost scalable robot system for collective behaviors," in *IEEE Int. Conf. Robotic. Automat.*, Saint Paul, MN, USA, May 2012, pp. 3293–3298.

[74] D. Pickem, M. Lee, and M. Egerstedt, "The GRITSBot in its natural habitat - A multi-robot testbed," in *IEEE Int. Conf. Robot. Autom.*, Seattle, WA, USA, May 2015, pp. 4062–4067.

[75] G. Caprari and R. Siegwart, "Mobile micro-robots ready to use: Alice," in *IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, Edmonton, AB, Canada, Aug. 2005, pp. 3295–3300.

[76] A. P. Sabelhaus, D. Mirsky, L. M. Hill, N. C. Martins, and S. Bergbreiter, "TinyTeRP: A tiny terrestrial robotic platform with modular sensing," in *IEEE Int. Conf. Robot. Autom.*, Karlsruhe, Germany, May 2013, pp. 2600–2605.

[77] V. Indelman, "Towards multi-robot active collaborative state estimation via belief space planning," in *IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, Hamburg, Germany, Sep. 2015, pp. 4620–4626.

[78] T. Regev and V. Indelman, "Multi-robot decentralized belief space planning in unknown environments via efficient re-evaluation of impacted paths," in *IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, Daejeon, South Korea, Oct. 2016, pp. 5591–5598.

[79] L. Ronghua, M. Huaqing, L. Maohai, and H. Qingcheng, "A Method for active global localization in multi-robot system," *Int. J. Adv. Robot. Syst.*, vol. 5, no. 3, pp. 1–10, Sep. 2008.

[80] N. Palomeras, M. Carreras, and J. Andrade-Cetto, "Active SLAM for autonomous underwater exploration," *Remote Sens.*, vol. 11, no. 23:2827, pp. 1–19, Nov. 2019.

[81] S. Kornienko, O. Kornienko, C. Constantinescu, M. Pradier, and P. Levi, "Cognitive micro-Agents: individual and collective perception in microrobotic swarm," in *IJCAI Work. Agent. Real-Time Dyn. Env.*, Edinburgh, Scotland, Jul. 2005, pp. 33–42.

[82] M. Gendreau and J.-Y. Potvin, "Tabu search," in *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques*, E. K. Burke and G. Kendall, Eds. Boston, MA: Springer US, 2005, pp. 165–186.

[83] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *Int. J. Robot. Res.*, vol. 30, no. 7, pp. 846–894, Jun. 2011.

[84] S. Katoch, S. S. Chauhan, and V. Kumar, "A review on genetic algorithm: past, present, and future," *Multimed. Tools Appl.*, vol. 80, no. 5, pp. 8091–8126, Feb. 2021.

[85] I. F. Akyildiz and M. Can Vuran, "Network layer," in *Wireless Sensor Networks*, United Kingdom: John Wiley & Sons Ltd, 2010, pp. 139–165.

[86] A. Dutta, A. Ghosh, and O. P. Kreidl, "Multi-robot informative path planning with continuous connectivity constraints," in *IEEE Int. Conf. Robot. Autom.*, Montreal, QC, Canada, May 2019, pp. 3245–3251.

Kasra Eshaghi (S'19) received his B.A.Sc. degree in Mechanical Engineering in 2018. He is currently a Ph.D. candidate at UofT. His research interests include software and hardware design for robotic swarms.

Andrew Rogers received his B.A.Sc. degree in Mechanical Engineering in 2021. He is currently a M.A.Sc. student at UofT. His research interests include hardware design and application of robotic swarms.

Goldie Nejat (S'03-M'06) is the Canada Research Chair in Robots for Society and a Professor in the Department of Mechanical and Industrial Engineering at UofT. She is the Founder and Director of the ASBLab. She is also a Fellow of the Canadian Institute for Advanced Research. Her research interests include intelligent assistive/service robots, human-robot interactions, and semi-autonomous/autonomous control. She received her B.A.Sc. and Ph.D. degrees in Mechanical Engineering at UofT.

Beno Benhabib received the B.Sc., M.Sc., and Ph.D. degrees in Mechanical Engineering in 1980, 1982, and 1985, respectively. He has been a Professor with the Department of Mechanical and Industrial Engineering at UofT since 1986. His current research interests include the design and control of intelligent autonomous systems.