Closed-Loop Aerial Robot-Assisted Navigation of a Cohesive Ground-Based Robot Swarm

Andreagiovanni Reina, Carlo Pinciroli,
Eliseo Ferrante, Ali Emre Turgut, Rehan O'Grady,
Mauro Birattari, and Marco Dorigo

IRIDIA – Technical Report Series

Technical Report No.
TR/IRIDIA/2011-020

October 2011
Last revision: October 2011

IRIDIA – Technical Report Series
ISSN 1781-3794

Published by:

IRIDIA, *Institut de Recherches Interdisciplinaires*
et de Développements en Intelligence Artificielle
Université Libre de Bruxelles
Av F. D. Roosevelt 50, CP 194/6
1050 Bruxelles, Belgium

Technical report number TR/IRIDIA/2011-020

Revision history:

TR/IRIDIA/2011-020.001 October 2011

The information provided is the sole responsibility of the authors and does not necessarily reflect the opinion of the members of IRIDIA. The authors take full responsibility for any copyright breaches that may result from publication of this paper in the IRIDIA – Technical Report Series. IRIDIA is not responsible for any use that might be made of data appearing in this publication.

Closed-Loop Aerial Robot-Assisted Navigation of a Cohesive Ground-Based Robot Swarm

A. Reina, C. Pinciroli, E. Ferrante,
A.E. Turgut, R. O'Grady, M. Birattari, and M. Dorigo

October 17, 2011

Abstract

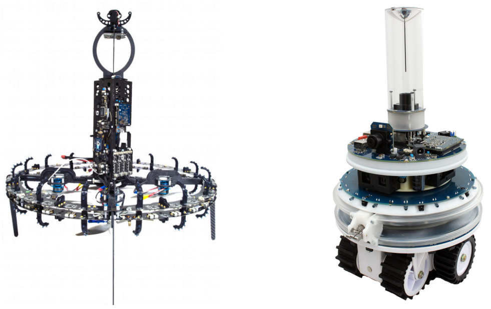
We study how a swarm of ground-based robots, with no knowledge of the environment, can be guided to destination by a group of aerial robots. We show that if the ground-based robots maintain group cohesion, it is possible to create a closed-loop among aerial robots and ground-based robots that results in robust navigation even in presence of high sensory noise. We validate our results through extensive experiments in simulation and provide a proof-of-concept experiment with real robots.

1 Introduction

Swarm robotics focuses on the study of coordination strategies for large groups of relatively simple robots. Most of the research in this field concentrates on homogeneous groups of robots. However, recent work [4, 5, 8, 9] has demonstrated the advantages of heterogeneous robot swarms. One of the most important advantages is the possibility to leverage specialization, which can increase the overall swarm efficiency [1] and makes it thus possible to optimize each robot type for a specific purpose, lowering their complexity and cost.

In this paper, we focus on a navigation task whereby a swarm of ground-based robots must reach a target location that they cannot sense, but that is known by a networked swarm of aerial robots attached to the ceiling. Thus, the latter must guide the ground-based robots to the target area—we refer to this as *assisted navigation*. We propose two novel algorithms to achieve closed-loop control. By closed-loop control, we mean that each aerial robot continuously monitors the ground-based robots in range and sends guidance instructions relative to the current positions of the ground-based robots. In this way, an aerial robot can correct navigation errors as they occur. Hopping from aerial robot to aerial robot, the ground-based robots eventually reach the target.

In both algorithms, we rule out the possibility that an aerial robot calculates individual instructions for each ground-based robot, as this would pose serious scalability issues for large group sizes, and impose high computational power



(a) The eye-bot

(b) The foot-bot

Figure 1: The heterogeneous robotic platform.

requirements on the aerial robots. We equally avoid a leader-based approach, whereby an aerial robot communicates with only one ground-based robot, as this would hinder robustness. Instead, in both algorithms, an aerial robot considers ground-based robots as a unique entity. In this way, an aerial robot needs only to broadcast guidance instructions for the ground-based robot swarm as a whole. Each ground-based robot, in turn, translates swarm-level guidance into individual instructions.

The proposed algorithms rely on the fact that a common frame of reference can be constructed between an aerial robot and the swarm of ground-based robots. The common frame of reference is used by an aerial robot to communicate guidance instructions used by the ground-based robots to move. The difference between the two algorithms is in the way the common frame of reference is realized. In the first algorithm, we assume that the frame of reference is *shared*, i.e., persistent and known by all robots. Experiments show that cohesion is not strictly necessary for the ground-based robots to reach their destination, but it makes navigation robust to high sensory noise. Such high level of robustness allowed us to devise the second algorithm, in which a common, undisputed reference frame is not available (*non-shared* reference). Thus, the robots must reconstruct an individual approximation of it from local sensory data. Results show that, also in this case, cohesion among the ground-based robots is enough to ensure highly robust navigation.

The paper is organized as follows. In Section 2, we describe the robotics platform we used to implement and test our approach. In Sections 3 and 4, we present and validate the two algorithms for closed-loop assisted navigation. In Section 5, we review related work and conclude the paper indicating directions for future work.

2 The Robots

Our robot swarm is composed of aerial robots called *eye-bots* (Fig. 1(a)) and ground-based robots called *foot-bots* (Fig. 1(b)). Eye-bots [13] are quad-rotor equipped robots capable of flying and attaching to the ceiling. They are equipped with a camera which allows them to monitor what happens on the ground. Foot-bots [2] are mobile robots that maneuver with a combined system of track and wheels. They are equipped with proximity sensors and a camera facing towards the ceiling.

Communication between eye-bots and foot-bots can occur in two ways. The first way is visual: both are equipped with an RGB LED ring enabling them to convey information to robots within range. The second way is via a range and bearing system [14] mounted on both types of robot. This system allows the robots to broadcast and receive messages either from neighbors in the same plane, or in a cone above the foot-bots or beneath the eye-bots. Furthermore, the system allows for *situated communication*, meaning that recipients of a message know both the content of the message and the spatial origin of the message (with respect to their own frame of reference).

3 Assisted Navigation With Shared Reference

In classical control theory, a closed-loop is achieved when the difference among the system output and the desired reference is fed back as input to the system and used to correct the system's behavior [11]. In our work, the controlled system is a group of foot-bots and the control system is an eye-bot. The eye-bot continuously monitors the position of the center of mass of the foot-bots beneath. The reference value is the target location that the foot-bots must reach. The output of the eye-bot is the guidance vector $\vec{d}$.

In Section 3.1, we describe the basic communication algorithm between the eye-bot and the foot-bots. In Section 3.2, we describe the cohesion algorithm we employed. In Section 3.3 we analyze the system performance.

3.1 Aerial–Ground Robot Communication

The shared reference frame. Both of the proposed algorithms rely on the fact that an eye-bot and the foot-bots below share a common frame of reference. This first algorithm assumes that the frame of reference is shared, i.e., persistent and known by all robots. A shared frame of reference could be constructed, for example, from an environmental feature accessible to all robots. Alternatively, as proposed in [4, 5], an eye-bot could use its body to indicate a reference vector $\vec{r}$, and foot-bots would sense such vector. In this work, we follow the latter approach. The reference vector is used as the x-axis of a right-handed common reference frame. Therefore, once $\vec{r}$ is known, the other axes are set without ambiguity. We used the LED ring of the eye-bot to achieve this effect, by lighting up two LEDs located at opposite locations on the ring in different

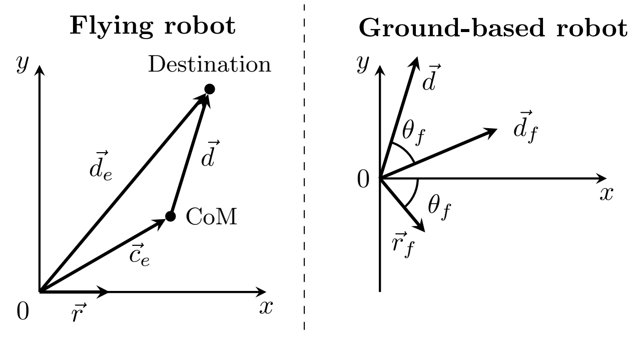


Figure 2: Assisted navigation with shared reference. Calculations by the aerial robot (left) and by the ground-based robot (right). The center of mass of the distribution of ground-based robots calculated by the aerial robot is marked with 'CoM'.

colors. The upwards-looking camera of the foot-bot detects the LEDs as colored blobs, from which the reference vector is constructed.

Aerial robot. An eye-bot sends guidance instructions to the foot-bots using the range and bearing system. Guidance instructions consist of a direction vector $\vec{d}$ expressed with respect to the common reference frame. Since the foot-bots are on the ground, for an eye-bot it is sufficient to perform two-dimensional calculations (see also Figure 2). Thus, both $\vec{r}$ and $\vec{d}$ are two-dimensional vectors. To calculate $\vec{d}$, the eye-bot first calculates the center of mass $\vec{c}_e$ of the distribution of the foot-bots beneath. Each foot-bot keeps its LED ring lit, so that the eye-bot camera can detect the foot-bots as colored blobs. Vector $\vec{c}_e$ is calculated simply as the center of mass of the positions of the sensed blobs.

We indicate with $\vec{d}_e$ the position on the ground where the eye-bot would like the foot-bots to go to. Vector $\vec{d}_e$ is the result of motion planning by the eye-bot (for a distributed algorithm to calculate $\vec{d}_e$, see [12]). If the reference vector $\vec{r}$ is set as the local x-axis of the eye-bot, to obtain the direction vector $\vec{d}$, the eye-bot computes $\vec{d} = \vec{d}_e - \vec{c}_e$.

Ground-based robot. From the point of view of a foot-bot, the reference vector $\vec{r}$ is sensed with respect to its local reference frame as a vector $\vec{r}_f$ (see also Figure 2). Such vector forms an angle θ_f with the local x-axis. The foot-bot must move according to a local direction vector $\vec{d}_f$, that is obtained simply rotating $\vec{d}$ by θ_f.

3.2 Cohesion

Several approaches exist to achieve cohesion with mobile robots [17]. One of the simplest and most effective is based on virtual physics [16]. This approach considers the robots as particles immersed in a virtual potential field. The potential field is "virtual" because each robot uses its sensor readings to calculate

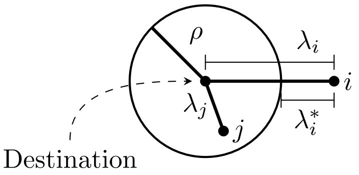


Figure 3: Geometrical representation of the overflow distance. Robot i is outside the optimal distribution at destination, so $\lambda_i^* > 0$. Robot j is inside, so $\lambda_j^* = 0$.

the force of interaction with nearby robots. In the case of cohesion, such force is used as a direction vector and is transformed into motion actuation (e.g., wheel speed).

For our system, we selected the following potential:

$$V(x) = \epsilon \left(\left(\frac{\delta}{x}\right)^{2\alpha} - 2\left(\frac{\delta}{x}\right)^{\alpha} \right),$$

where x is the distance between two robots, δ is the target distance the robots should keep between each other, and ϵ is a factor that accounts for the depth of the minimum of the function in $x = \delta$. When $\alpha = 6$, $V(x)$ corresponds to the Lennard-Jones potential [7]. The magnitude of the virtual force of interaction $\vec{f}(x)$ between two robots is given by:

$$|\vec{f}(x)| = -\frac{\mathrm{d}\,V}{\mathrm{d}\,x} = \frac{2\alpha\epsilon}{x} \left(\left(\frac{\delta}{x}\right)^{2\alpha} - \left(\frac{\delta}{x}\right)^{\alpha} \right).$$

In this work, we used the following values for the parameters: $\delta = 0.4\,\mathrm{m}$, $\epsilon = 8\,500\,\mathrm{J}$ and $\alpha = 0.25$. The foot-bots use the range and bearing system to obtain the distance vectors $\vec{x}_i$ to nearby robots, and calculate the force of interaction $\vec{f_i}(x_i)$ with each of them. The cohesion interaction vector is then computed as the average interaction force. The cohesion interaction vector is summed to $\vec{d_f}$ to obtain the actual movement vector, in turn transformed into wheel actuation.

3.3 Experiments

3.3.1 Experimental Setup

We conducted a set of experiments in simulation to analyze the system. The robots were simulated with ARGoS [10], an accurate physics-based simulator.

In our experiments, a group of foot-bots is deployed in a certain location in the environment. The task is navigating the environment to reach a target location that is 12.5 m far from the deployment location and is unknown by the foot-bots. To assist these robots, 5 eye-bots are attached to the ceiling. They form a line connecting the foot-bot deployment location to the target location.

Each eye-bot's task is to guide the foot-bots within its sensorial range to the next eye-bot. Thus, navigating from eye-bot to eye-bot, the foot-bot group eventually reaches the target.

To study our system, we selected two performance measures. The first considers the distance of the foot-bots to the target at the end of the experiment. However, a simple average of these distances would not be a good performance measure because, the larger the group size, the higher the average is. To obtain a measure not dependent on group size, first consider the case in which the foot-bots are in perfect formation at distance $\delta = 0.4\,\mathrm{m}$ from each other. Each foot-bot can be seen as a circle whose radius is $\delta/2$. With N foot-bots, the total area occupied by the circles is $A = N\pi(\delta/2)^2$. The tightest foot-bot formation possible corresponds to the optimal circle packing [6] in a circular region around the destination point. We can approximate the radius ρ of this circular region with:

$$\rho = \sqrt{\frac{A}{\pi}} = \sqrt{\frac{N\delta^2}{4}}.$$

For each foot-bot i, we calculate its distance to the target λ_i and define the *overflow distance* λ_i^* as

$$\lambda_i^* = \max(\lambda_i - \rho, 0).$$

The *overall overflow distance* λ^* is calculated as the average over the λ_i^*. As shown in Figure 3, the only foot-bots that contribute to the overall overflow distance are those outside the optimal circle packing centered in the destination point. The smaller λ_i^*, the closer the robots are to the target location. The overall overflow distance is our first performance measure.

The second is the number of foot-bots lost. A foot-bot is considered lost when it has not been in communication with an eye-bot for some time, because without directional information it cannot proceed by itself. However, connectivity may be lost temporarily, so a certain degree of tolerance is necessary. Therefore, a foot-bot that has just lost contact with an eye-bot keeps moving along the last received direction for a predefined amount of time. After this time, if the foot-bot still has not restored contact with an eye-bot, the foot-bot stops on spot and declares itself lost.

3.3.2 Results Without Sensory Noise And Bias

In this first set of experiments, we tested our system under the assumption that the robots' sensors are not affected by noise and always return correct readings.

Figure 5(a) compares the distribution of λ_i^* over 100 experimental runs when the robots do not maintain cohesion and when they do. In these experiments, the foot-bot group size ranges from 1 to 25. The box plots show that, even though cohesion is not strictly necessary for the foot-bots to reach their destination, it increases significantly the performance of the system. In fact, with cohesion, the overall overflow distance is zero for all the group sizes we tested.

Furthermore, requiring the foot-bots to maintain cohesion has a dramatic impact on the number of foot-bots lost. As reported in Figure 5(b), without cohesion about a third of the robots lose connectivity with the eye-bots without being able to restore it. On the contrary, cohesion allows all the robots to reach their destination. In fact, even if some of the robots happen to lose connectivity, the rest of the swarm is still able to drag them in the right direction. In this sense, a cohesive swarm has the advantage of covering a large area, thus increasing the probability that some robots are in contact with an eye-bot. At any moment, it is enough that a small portion of the robots is informed of the target direction for the entire group to reach destination [3].

3.3.3 Results With Sensory Noise And Bias

We repeated the previous set of experiments to study the impact of noise and errors on the system performance. In all these experiments, we employed 25 foot-bots.

We focused our analysis on the foot-bots and indentified two main aspects that can be affected by noise on the foot-bots: *(i)* the reference vector $\vec{r}_f$ and *(ii)* the measure of the distance to nearby robots. Noise on $\vec{r}_f$ affects the target direction $\vec{d}_f$, potentially increasing the overall overflow distance. Noise on the distance measures to neighbors could result in the loss of robots when the foot-bots maintain cohesion.

Bias on the sensed reference vector $\vec{r}_f$. In these experiments we assume that the foot-bots have a bias in the way they sense the reference vector $\vec{r}_f$. The bias is realized as a constant rotation applied to $\vec{r}_f$. Such rotation is chosen uniformly at random in a certain range by each robot at the beginning of each run. We tried the following ranges: $[-1^\circ : 1^\circ]$, $[-2^\circ : 2^\circ]$, $[-5^\circ : 5^\circ]$, $[-10^\circ : 10^\circ]$, and $[-20^\circ : 20^\circ]$.

We report in Figure 5(c) a comparison of the results obtained with and without cohesion. Without cohesion, the robots, on average, end their journey with an overall overflow distance λ^* of almost 1 m. This result is fairly stable for all the tested ranges with the exception of the widest bias range, $[-20^\circ : 20^\circ]$. In this case, the average value of λ^* almost doubles. On the contrary, when the foot-bots maintain cohesion, the system is always able to reach destination precisely ($\lambda^* = 0$). In other words, even if the robots suffer from different bias, cohesion allows them to average out their differences and converge on the destination point. In fact, cohesion compensates at the swarm level for individually biased sensory data.

In Figure 5(d), we plot the percentage of foot-bot lost on the way to destination. When the robots do not maintain cohesion, roughly a third of them is lost during navigation for all ranges except the widest. In the latter case, on average, almost half the robots get lost. Cohesion, on the other hand, allows all robots to reach their destination.

Noise on the sensed reference vector $\vec{r}_f$. In this set of experiments, we added noise to the reference vector $\vec{r}_f$ sensed by each foot-bot. Noise consists of a rotation applied to $\vec{r}_f$. The rotation is chosen at random at each control step from a Gaussian distribution with zero mean. We studied the impact of this kind of noise on the system for the following values of the standard deviation: $\{5°, 10°, 20°\}$.

In Figure 5(e), the results when the robots maintain cohesion are identical to those obtained when a bias is applied to $\vec{r}_f$. Interestingly, without cohesion Gaussian noise has a positive effect on λ^*. This counter-intuitive result can be explained by visually analyzing the way the foot-bots move when affected by high noise (standard deviation ¿ 5°.) Two phenomena happen. The first phenomenon is that the robots tend to move erratically and collide with other robots. The second phenomenon is that the robots tend, on average, to go in the right direction, due to the zero mean of the Gaussian noise. For this reason, we noticed that the collisions increase over time and along the right path. In other words, the robots tend to cluster together and form a very tight, cohesive aggregate that drifts towards the right direction. Therefore, the results are in line to those obtained when cohesion is explicitly maintained. The above reasoning also explains why, for high levels of noise, foot-bots are not lost when cohesion is not explicitly maintained.

Noise in cohesion. In this set of experiments, we added noise to the distance measures x_i that each foot-bot takes to calculate the cohesion interaction force $|\vec{f_i}(x_i)|$ (see Section 3.2). The noise was taken from a Gaussian distribution with zero mean. We studied the impact of this kind of noise on the system for the following values of the standard deviation: $\{5, 10, 15, 20, 25\}$ cm. We varied the standard deviation to test its impact on the system performance. In Figure 7(a), we see the evolution over time of the average of the overall overflow distance of the foot-bots for increasing noise levels. Despite the fact that the time to complete the route increases with the noise level, with cohesion the foot-bot eventually always get to destination, with no robots lost. Thus, the system is robust also to this kind of noise.

4 Assisted Navigation With Non-Shared Reference

In the following, we explain how cohesion can be used to achieve robust closed-loop control even without the assumption that the reference frame is shared—the robots cannot exploit a common environmental cue and the eye-bot is not capable to indicate vector $\vec{r}$ with its body. In this second algorithm, we only require that the robots can communicate and locate each other (situated communication). Notice that the robots do not need to sense their mutual *orientations*—just their mutual *positions.*

In Section 4.1, we describe the new communication algorithm. The cohesion algorithm explained in Section 3.2 is unchanged. In Section 4.2 we analyze the properties of the system.

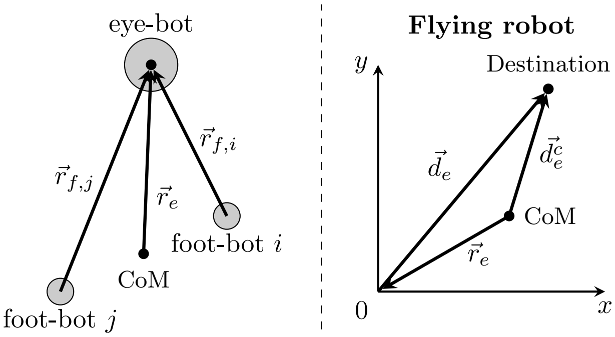


Figure 4: Assisted navigation with non-shared reference. The robots do not agree anymore on the frame of reference (left). Calculations by the aerial robot (right). The center of mass of the distribution of ground-based robots calculated by the aerial robot is marked with 'CoM'.

4.1 Aerial–Ground Robot Communication

The non-shared reference frame. First, let us focus on the case in which one eye-bot is guiding one foot-bot. The robots can sense their mutual positions, that is, they can both sense the vector connecting their positions. In the environment, such vector is three dimensional. However, since navigation is two-dimensional, it is enough for the robots to be able to sense the projection of that vector on the ground (for the eye-bot) or on the ceiling (for the foot-bot), and use such projection as the reference vector $\vec{r}$. To avoid ambiguity, by design $\vec{r}$ points from the foot-bot to the eye-bot. When the group size is greater than 1, the eye-bot calculates the center of mass of the distribution of foot-bots $\vec{c}_e$ and sets its local reference vector to -$\vec{c}_e$. For the foot-bot nothing changes. However, when the foot-bot group size is larger than 1, the robots no longer agree on the exact reference frame (see Figure 4). In fact, while the eye-bot uses the center of mass, the foot-bots calculate the reference vector from their position to the eye-bot, which is in general different from that of the center of mass. Therefore, we will refer to the reference vector calculated by the eye-bot as $\vec{r}_e$ and to that calculated by a foot-bot as $\vec{r}_f$. As we will see, even if the robots do not agree on the reference frame, navigation still works.

Aerial robot. Given the target location for the foot-bots $\vec{d}_e$, expressed with respect to its own local reference frame (see Figure 4), the eye-bot calculates the vector $\vec{d}_e^c = \vec{r}_e + \vec{d}_e$, which represents the movement that the center of mass of the foot-bots must perform to reach the desired position. Vector $\vec{d}_e^c$ is expressed with respect to the eye-bot's local reference frame. The direction vector $\vec{d}$ sent to the foot-bots is obtained by expressing $\vec{d}_e^c$ in the reference frame defined by $\vec{r}_e$.

Ground-based robot. A foot-bot receives vector $\vec{d}$, and performs the same calculations described in Section 3.1.

4.2 Experiments

To test the performance of our system, we used the same experimental setup as in Section 3.3.

4.2.1 Results Without Sensory Noise And Bias

Results in Figure 6(a) show that, without cohesion, the foot-bots are basically unable to reach the destination. With cohesion, instead, results are in line with those found in Section 3.3.2. Practically in all the runs, the robots reach their destination in less than 1500 seconds. The outliers present in Figure 6(a) correspond to few runs in which the robots took longer than 1500 seconds to reach their destination. With respect to the first algorithm, this algorithm performs similarly well for both performance measures. The main difference among the two algorithms is that the first one allows the foot-bots to reach their destination 10 times faster. This is not surprising, because, with a non-shared reference, the foot-bots must spend time to keep cohesion, due to the different perception of the reference vectors $\vec{r}_f$. On the contrary, when the reference is shared, the foot-bots have a lower probability to diverge, so navigation is smoother.

Analyzing the individual $\vec{r}_f$ calculated by the foot-bots, we noticed that, with this algorithm, it is as if each robot is experiencing a bias on $\vec{r}_f$ that varies (relatively slowly) over time. For this reason, then, it can be expected that this algorithm performs somewhat similarly to the case in which a constant bias is applied to the shared reference.

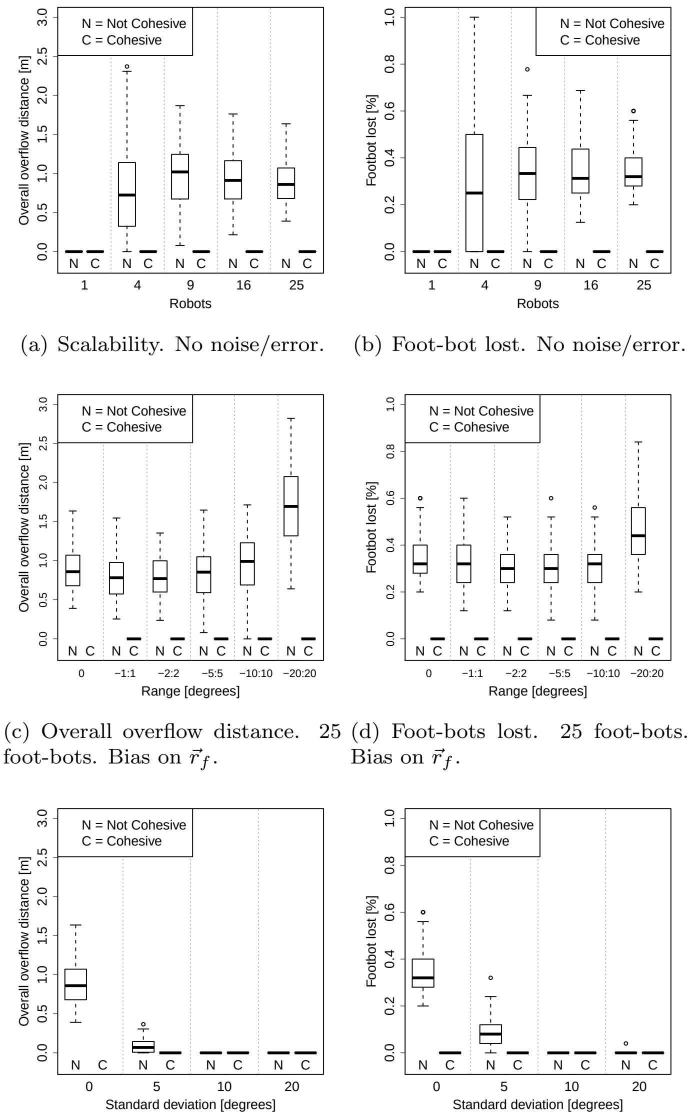
4.2.2 Results With Sensory Noise And Bias

Both in the case of biased and noisy $\vec{r}_f$, the cohesive foot-bot swarm reaches the destination without foot-bots lost (results reported in Figures 6(c)–6(f)). These results are remarkable and highlight how powerful cohesion is in filtering noise and averaging out the different local references $\vec{r}_f$ calculated by the robots. Even when Gaussian noise is applied to cohesion, the foot-bots (reported in Figure 7(b)) reach safely their destination. Increasing levels of noise progressively slow down system, but they do not hinder robustness.

5 Related Work and Conclusions

In this paper, we presented two algorithms to achieve closed-loop control among aerial robots and ground-based robots in a navigation task, in which the destination is known only by the aerial robots. We showed that, if the ground-based robots maintain cohesion, the aerial robots can deal with the ground-based robots as a unique entity, and the resulting navigation is robust to high sensory noise.

Previous work in this matter has primarily been done by Ducatelle *et al.* [4, 5]. They focus on self-organization of a heterogeneous swarm formed by two



(a) Scalability. No noise/error.

(b) Foot-bot lost. No noise/error.

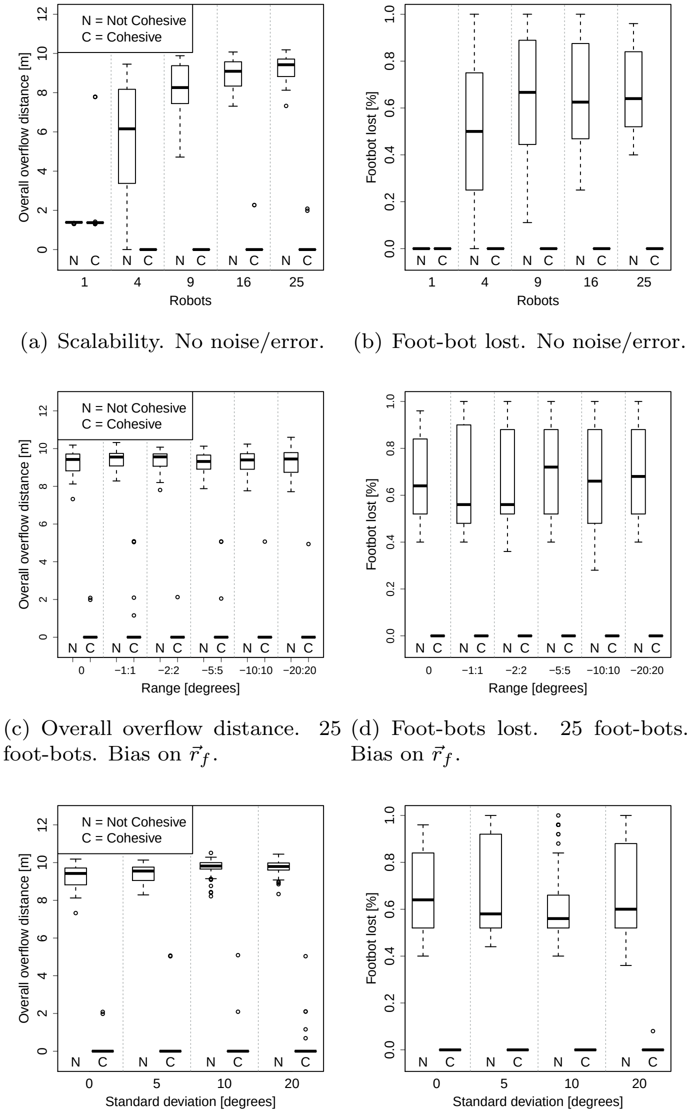
(c) Overall overflow distance. 25 foot-bots. Bias on $\vec{r}_f$.

(d) Foot-bots lost. 25 foot-bots. Bias on $\vec{r}_f$.

(e) Overall overflow distance. 25 foot-bots. Gaussian noise on $\vec{r}_f$.

(f) Foot-bots lost. 25 foot-bots. Gaussian noise on $\vec{r}_f$.

Figure 5: Box plot of the results of the experiments with shared reference after 150 s of simulated time. The bottom and top of the box are the lower and upper quartiles of the data, respectively. The band near the middle of the box is the median. The lower whisker extends till 1.5 times the range of the lower quartile, and the upper whisker till 1.5 times the range of the upper quartile.



(a) Scalability. No noise/error.

(b) Foot-bot lost. No noise/error.

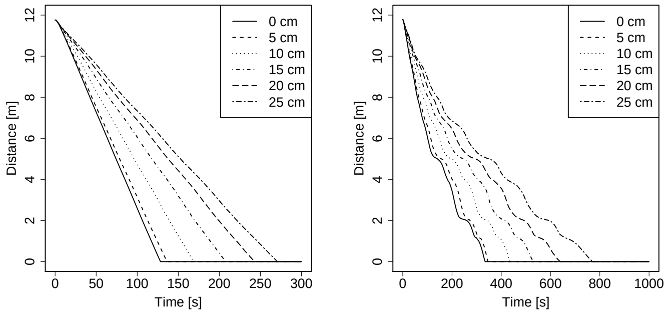
(c) Overall overflow distance. 25 foot-bots. Bias on $\vec{r}_f$.

(d) Foot-bots lost. 25 foot-bots. Bias on $\vec{r}_f$.

(e) Overall overflow distance. 25 foot-bots. Gaussian noise on $\vec{r}_f$.

(f) Foot-bots lost. 25 foot-bots. Gaussian noise on $\vec{r}_f$.

Figure 6: Box plot of the results of the experiments with non-shared reference after 150 s of simulated time. The bottom and top of the box are the lower and upper quartiles of the data, respectively. The band near the middle of the box is the median. The lower whisker extends till 1.5 times the range of the lower quartile, and the upper whisker till 1.5 times the range of the upper quartile.



(a) With shared reference.

(b) With non-shared reference.

Figure 7: Results of navigation experiments with noise in cohesion.

homogeneous sub-swarms: one of aerial robots and one of ground-based robots. The robots construct a shared common frame of reference through the LED ring-equipped body of the aerial robots, in the same way as our first proposed algorithm. The aerial robots communicate guidance information via radio. In [4], Ducatelle *et al.* present an algorithm that enables a group of aerial robots to find the shortest path from a source location to a target location, and back. In this work, the aerial robots are statically attached to the ceiling. In [5], the aerial robots are allowed to move, to find optimal attachment locations on the ceiling. The interesting aspect of this work is that the aerial robots and the ground-based robots create a closed-loop that allows the swarm to optimize the relative placement of the aerial robots to achieve better navigation performance for the ground-based robots.

In these studies, the ground-based robots move individually (not cohesively) and the aerial robots behave as smart waypoints. The complexity of the trajectories the ground-based robots can follow is limited by the number of aerial robots in the system. In contrast, with our system, each aerial robot can send precise guidance instructions whose complexity is arbitrary.

Our system could be also suitable to achieve human-assisted navigation, whereby a human operator takes the role of the aerial robot. Recent work by Robuffo *et al.* [15] demonstrated how a cohesive swarm of flying robots can be controlled by a human through a suitably configured haptic interface. In this system, the user controls a predefined leader directly, while the other robots simply follow it. A similar system could be realized with our algorithms.

References

[1] F. Adler, E. LeBrun, and D. Feener Jr. Maintaining diversity in an ant community: Modeling, extending, and testing the dominance-discovery trade-off. *The American Naturalist*, 169:323–333, 2007.

[2] M. Bonani, V. Longchamp, S. Magnenat, P. Rétornaz, D. Burnier, G. Roulet, F. Vaussard, H. Bleuler, and F. Mondada. The MarXbot, a Miniature Mobile Robot Opening new Perspectives for the Collective-robotic Research. In *International Conference on Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ*, pages 4187–4193. IEEE Press, Piscataway, NJ, 2010.

[3] I. D. Couzin, J. Krause, N. R. Franks, and S. A. Levin. Effective leadership and decision-making in animal groups on the move. *Nature*, 433:513–516, 2005.

[4] F. Ducatelle, G. Di Caro, and L. Gambardella. Cooperative stigmergic navigation in a heterogeneous robotic swarm. In *Proceedings of the 11th International Conference on Simulation of Adaptive Behavior (SAB2010)*, pages 607–617. Publisher, Address, 2010.

[5] F. Ducatelle, G. Di Caro, C. Pinciroli, and L. Gambardella. Self-organized cooperation between robotic swarms. *Swarm Intelligence*, 5:73–96, 2011.

[6] M. Goldberg. Packing of 14, 16, 17, and 20 circles in a circle. *Mathematics Magazine*, pages 134–139, 1971.

[7] J. E. Lennard-Jones. On the determination of molecular fields. *Proceedings of the Royal Society of London*, 106(738):463–477, 1924.

[8] R. O'Grady, C. Pinciroli, A. L. Christensen, and M. Dorigo. Supervised group size regulation in a heterogeneous robotic swarm. In *Proceedings of ROBOTICA 2009 - 9th International Conference on Autonomous Robot Systems and Competitions*, pages 113–119. IPCB, Castelo Branco, Portugal, 2009.

[9] C. Pinciroli, R. O'Grady, A. L. Christensen, and M. Dorigo. Self-organised recruitment in a heterogeneous swarm. In E. Prassel et al., editors, *The 14th International Conference on Advanced Robotics (ICAR 2009)*, page 8. Proceedings on CD-ROM, paper ID 176, 2009.

[10] C. Pinciroli, V. Trianni, R. O'Grady, G. Pini, A. Brutschy, M. Brambilla, N. Mathews, E. Ferrante, G. D. Caro, F. Ducatelle, T. Stirling, A. Gutiérrez, L. M. Gambardella, and M. Dorigo. ARGoS: a modular, multi-engine simulator for heterogeneous swarm robotics. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2011)*, pages 5027–5034. IEEE Computer Society Press, Los Alamitos, CA, 2011.

[11] K. J. Åström and R. M. Murray. *Feedback Systems: An Introduction for Scientists and Engineers*. Princeton University Press, Princeton, NJ, 2008.

[12] A. Reina, G. Di Caro, F. Ducatelle, and L. Gambardella. Distributed motion planning for ground objects using a network of robotic ceiling cameras. In R. Groß, L. Alboul, C. Melhuish, M. Witkowski, T. Prescott,

and J. Penders, editors, *Towards Autonomous Robotic Systems*, volume 6856 of *Lecture Notes in Computer Science*, pages 137–148. Springer Berlin/Heidelberg, 2011.

[13] J. Roberts, T. Stirling, J.-C. Zufferey, and D. Floreano. Quadrotor using minimal sensing for autonomous indoor flight. In *Proceedings of the European Micro Air Vehicle Conference and Flight Competition (EMAV2007)*, 2007.

[14] J. Roberts, T. Stirling, J.-C. Zufferey, and D. Floreano. 2.5d infrared range and bearing system for collective robotics. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2009)*. IEEE Press, Piscataway, NJ, 2009.

[15] P. Robuffo Giordano, A. Franchi, C. Secchi, and H. H. Bülthoff. Experiments of passivity-based bilateral aerial teleoperation of a group of UAVs with decentralized velocity synchronization. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2011)*, pages 163–170. IEEE Computer Society Press, Los Alamitos, CA, 2011.

[16] W. M. Spears, D. F. Spears, J. C. Hamann, and R. Heil. Distributed, physics-based control of swarms of vehicles. *Autonomous Robots*, 17:137–162, 2004.

[17] A. E. Turgut, H. Celikkanat, F. Gokce, and E. Sahin. Self-organized flocking in mobile robot swarms. *Swarm Intelligence*, 2(2–4):97–120, 2008.