

Spatially targeted communication in decentralized multirobot systems

Nithin Mathews · Gabriele Valentini ·
Anders Lyhne Christensen · Rehan O’Grady ·
Arne Brutschy · Marco Dorigo

Received: 7 June 2014 / Accepted: 3 January 2015 / Published online: 18 January 2015
© Springer Science+Business Media New York 2015

Abstract Spatially targeted communication (STC) allows a message sender to choose message recipients based on their location in space. Currently, STC in multirobot systems is limited to centralized systems. In this paper, we propose a novel communication protocol that enables STC in decentralized multirobot systems. The proposed protocol dispenses with the many aspects that underpin previous approaches, including external tracking infrastructure, a priori knowledge, global information, dedicated communication devices or unique robot IDs. We show how off-the-shelf hardware components such as cameras and LEDs can be used to establish ad-hoc STC links between robots. We present a Markov chain model for each of the two constituent parts of our proposed protocol and we show, using both model-based analysis and experimentation, that the proposed protocol is highly scalable. We also present the results of extensive experiments carried out on an autonomous, heterogeneous multirobot system composed of one aerial robot and numerous ground-

based robots. Finally, two real world application scenarios are presented in which we show how spatial coordination can be achieved in a decentralized multirobot system through STC.

Keywords Decentralized multirobot systems · Robot swarms · Air-ground robot teams · Spatial coordination · Inter-robot communication

1 Introduction

Communication often plays an integral role in multirobot systems that require spatial coordination among the robots. In such systems, the type of messages that can be transmitted, and hence the kind of coordination that can take place, fully depends on the modality of communication available to the system. One important axis of variation in communication technologies involves the extent to which transmitted messages include information about the spatial positioning of communicating robots. In many technologies, the spatial location of communication partners is largely unavailable. Robots using wireless Ethernet, for example, are not aware of the location of their communication partners. By contrast, *situated communication* (Støy 2001) modalities allow the recipient of a message to locate the sender in space within its own frame of reference. An example of situated communication would be for a robot to send a message by illuminating its LEDs in a particular color, and for nearby robots to use their cameras to detect the message while also computing their position relative to the message sending robot. Situated communication increases the spatial coordination capability of a robotic system by making messages of the type “*come to me*” or “*stay away from here*” immediately meaningful.

N. Mathews (✉) · G. Valentini · R. O’Grady · A. Brutschy ·
M. Dorigo
IRIDIA-CoDE, Université Libre de Bruxelles,
Avenue Franklin Roosevelt 50, CP 194/6, 1050 Brussels, Belgium
e-mail: nmathews@ulb.ac.be

G. Valentini
e-mail: gvalenti@ulb.ac.be

R. O’Grady
e-mail: rogrady@ulb.ac.be

A. Brutschy
e-mail: arne.brutschy@ulb.ac.be

M. Dorigo
e-mail: mdorigo@ulb.ac.be

A. L. Christensen
Instituto de Telecomunicações, Instituto Universitário de Lisboa
(ISCTE-IUL), Avenida Forças Armadas, 1649-026 Lisboa, Portugal
e-mail: anders.christensen@iscte.pt

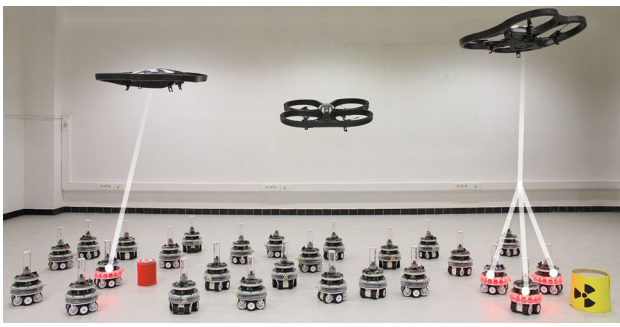


Fig. 1 An illustration of spatially targeted communication. The overlays show the ad-hoc communication links the aerial robots establish to communicate with specific robots on the ground that are situated next to the objects of interest

Situated communication modalities are, however, insufficient for spatial coordination in certain tasks as messages of the type “*reverse direction – the three of you are about to fall off a cliff!*” cannot be communicated. Such messages need to be delivered to particular message recipients based on their spatial location to be meaningful (as opposed to situated communication where the location and the number of message recipients cannot be controlled by the message sender). We coin the term *spatially targeted communication* (or STC) to describe this communication modality where messages are targeted at particular robots based on their location in the environment. STC is particularly useful (but not limited) to heterogeneous systems where spatial cooperation is required between different robot types that have different views of the environment and/or operate in different parts of the environment. In Fig. 1, for example, aerial robots are able to detect objects of interest on the ground and to instruct nearby ground-based robots accordingly.

In existing multirobot research, STC is limited to systems that either rely on centralized solutions or consider robots that have a global view of the system. For instance, some systems (Vaughan et al. 2000; Stentz et al. 2002; Kushleyev et al. 2013) use external tracking infrastructure in conjunction with global information or centralized control to enable communication between robots based on their location. Alternatively, robots may use dedicated on-board hardware suited for situated communication (Gutiérrez et al. 2008; Pugh et al. 2009; Roberts et al. 2012) to identify themselves by broadcasting centrally provisioned unique IDs, so that, in a second step, favorably located communication partners can be determined based on unique IDs and associated relative positions. Such centralized solutions are, however, unsuitable for certain types of multirobot systems, in particular for systems with large numbers of robots (Dorigo et al. 2014) or systems that need to operate in a priori unknown environments. External tracking infrastructure is potentially disadvantageous as it either requires access to satellite-based localization such as GPS or an a priori modified environment. Satellite-based

solutions are not practical indoors or for tasks that require higher position resolution than that provided by such systems. Dedicated communication hardware is potentially disadvantageous for three reasons. Firstly, the cost of dedicated hardware can become prohibitive as the numbers of robots in the system increases. Secondly, the use of dedicated hardware means that heterogeneous systems can only be composed of robots that have been explicitly designed a priori to communicate with each other (Dorigo et al. 2013). Thirdly, using dedicated hardware for situated communication can impose extra computational overheads in the form of sensor fusion—in Fig. 1, for instance, an aerial robot using dedicated communication hardware would have to map coordinates used by its communication device into the coordinates of whatever other sensor it was using to query the environment for interesting features (Kumar and Michael 2012).

To the best of our knowledge, there has been no proposal tabled for STC in decentralized multirobot systems to date. The protocol presented in this paper is the first of its kind to provide STC to decentralized systems as it does not require external tracking infrastructure, global information, centralized control, dedicated hardware, or the exchange of unique IDs. The core idea of our approach is to leverage existing situated communication modalities to establish ad-hoc communication links between robots based on their relative positions. Our approach assumes a situated communication modality capable of sending and receiving at least three distinguishable situated signals that are broadcast within a limited communication range. The protocol we propose is based around an iterative elimination process that results in a dedicated communication link being established between the robot initiating communication (henceforth the “*initiator robot*”) and the desired communication target (henceforth the “*target robot*”). In this process, the initiator robot iteratively eliminates robots within communication range (henceforth the “*potential recipient robots*”) it does not wish to communicate with until only the target robot is left. Further communication links to robots neighboring the target robot (henceforth the “*target group*”) can be established through an iterative growth process.

Our approach has a number of advantages. Firstly, the protocol we propose can be implemented using standard, low-cost components. In our experimentation, we use off-the-shelf cameras and LEDs in combination with wireless Ethernet. Secondly, our approach can be accurately modeled with Markov chains. We present separate models for the elimination process and the growth process. Thirdly, our approach is highly scalable: both model-based analysis and experimentation confirm that the elimination process at the heart of our approach scales logarithmically with respect to the number of potential recipient robots, while the growth process scales logarithmically with respect to the size of the target group. We present real robot experiments with a het-

erogeneous robotic system, composed of up to ten ground-based robots named marXbots (Bonani et al. 2010), that we use as potential recipient robots, and one aerial robot named AR.Drone (Bristeau et al. 2011), that we use as an initiator robot.¹ Our experiments show that our protocol can be successfully implemented on real robotic hardware, and that it can be used to achieve spatial coordination between robots in a decentralized multirobot system.

2 Communication and spatial coordination in multirobot systems

Multirobot systems employ a number of different wireless communication technologies to achieve spatial coordination. Wireless Ethernet (IEEE 802.11) is a widely used medium for inter-robot communication. Many systems that use the wireless Ethernet medium consider “tightly coupled” coordination (Chaimowicz et al. 2001; Gerkey and Mataric 2002; Parker et al. 2004; Zhang and Parker 2013) to direct broadcast messages from message sending robots to message receiving robots. In tightly coupled systems, message sending robots share their perceptual or computational capabilities with peer robots using broadcast messages while identifying themselves and their capabilities in the messages. The robots that process a certain message are often predetermined or are determined by the internal state of each robot. Stentz et al. (Stentz et al. 2002) as well as Vaughan et al. (Vaughan et al. 2000) use wireless Ethernet to coordinate a team composed of aerial and ground-based robots by combining GPS with environment maps or georeferenced coordinates. Kushleyev et al. (Kushleyev et al. 2013) use ZigBee technology (IEEE 802.15.4) to coordinate a team of micro quadrotors to fly formations in 3D space. They use an external motion capture system connected to a central base station to compute and control each team member’s position. Other researchers have proposed decentralized solutions based on Bluetooth to allow simultaneous communication between pairs of robots (Barnhard et al. 2004; McClain et al. 2004). In these studies, robot localization is determined using landmarks distributed in the environment and thus require a priori knowledge of the environment.

Many authors focus on the coordination of robots by considering onboard relative localization systems. While Franchi et al. (Franchi et al. 2010) proposed a probabilistic framework to enable relative localization in multirobot systems, other researchers developed ultrasound-based hardware sys-

tems (Grabowski and Khosla 2001; Rivard et al. 2008) for the same purpose. Pugh and Martinoli (Pugh et al. 2009) presented an infrared-based technology that allows for simultaneous localization and communication (hence situated communication) between neighboring robots. Today, derivations based on this work are available for a variety of platforms (Gutiérrez et al. 2008; Roberts et al. 2012). Although not explicitly designed for the purpose, such devices can be used by robots to enable STC by first exchanging robot IDs and then following up with communication data marked with addressee IDs.

Vision-based systems often use unique identifiers such as markers (Parker et al. 2004; Zhang and Parker 2013) or barcodes (Bolla et al. 2011) mounted on the robots to identify individual robots. Stegagno et al. (Stegagno et al. 2013) presented a system in which an aerial robot used its camera to identify marked ground-based peer robots and compute their relative location w.r.t. one another. Although there are several adverse conditions (such as occlusion, low illumination, etc.) under which the identification may not be feasible or reliable, systems that consider the exchange of robot IDs and support simultaneous robot localization have the potential to engage in STC by employing a wireless communication technology to address interestingly located peers directly. Other vision-based systems use a combination of cameras and LEDs to detect neighboring robots (without detecting their identities) and to enable robots to communicate with each other using differently colored LEDs. Camera and LED-based communication has been successfully applied to study a series of tasks that require spatial coordination such as path formation (Nouyan et al. 2008), collective transport (Campo et al. 2006), and morphology control (Christensen et al. 2007).

In our previous work (Mathews et al. 2010a), we have shown that a ground-based robot can use situated communication to initiate and establish STC links with peer robots in a homogeneous multirobot system composed of five ground-based robots. In simulation, we have also studied the benefits of cooperation through STC (Mathews et al. 2010b). In (Mathews et al. 2012a, b), we presented potential applications of STC. Our current work differs from our previous works in that it is a comprehensive work in which (i) we extend our real robot experiments to a heterogeneous robotic system composed of ground-based robots and one aerial robot, (ii) we combine cameras and LEDs-based communication with the high throughput medium of wireless Ethernet to enable high-level coordination between communicating robots while also establishing STC links much faster, (iii) we test the approach extensively on a real robotic hardware, (iv) we formulate two Markov chain models that characterize the performance of the processes used to establish STC links, and (v) we show two examples of real world applications that benefit from spatial coordination achieved through STC.

¹ Note that this is the allocation of roles used in the experiments presented in this paper; it is not a requirement of our protocol. For example, in previous work we showed the protocol working where both the initiator and potential recipient robots were both ground based (Mathews et al. 2010a).

3 Experimental platforms

Our heterogeneous multirobot testbed is composed of the two different robot platforms shown in Fig. 2. The AR.Drone 1.0 (Bristeau et al. 2011) is a commercially available quadcopter that can fly at speeds up to 18 km/h (Fig. 2 top). It has an autonomy of up to 12 min and a maximum flight weight of 420 g. It consists of a central plastic body housing a sensor and control board, one front-facing and another downward-pointing camera, a carbon-fiber crossbar connecting four brushless motors, and one removable indoor and another outdoor hull. The sensory equipment also includes a 6-DoF inertial measurement unit and an ultrasound altimeter. The API provided by the manufacturer gives access to sensory information, such as the altitude, battery level, and images from the cameras, and allows communication with the AR.Drone at 30 Hz via an ad-hoc wireless Ethernet network. An ARM9 processor running at 468 MHz with 128 MB of DDR RAM is the platform's main computational unit.

The marXbot (Bonani et al. 2010) is a ground-based modular robot that has a circular chassis with a 17 cm diameter (Fig. 2 bottom). A variety of hardware modules can be



Fig. 2 Our heterogeneous multirobot testbed consists of one AR.Drone (on the top) and multiple marXbots (on the bottom). RGB colors displayed by the marXbots can be detected by the AR.Drone using its downward-pointing camera. Signal transmission to the marXbots occurs over wireless Ethernet

successively mounted on the base module. The base module offers electrical power from a removable battery to all modules and simultaneously provides the robot with differential drive capabilities. The configuration used in this study also includes an attachment module that enables marXbots to physically attach to one another. Multiple marXbots can thus self-assemble into larger physical structures (Groß and Dorigo 2008). Additionally, 12 RGB-colored LEDs are distributed around the attachment module. A range and bearing module provides high-speed communication (1 Mbps) and relative range and bearing estimates to neighboring marXbots. The topmost module is the main computer module. It features an on-board ARM 11 processor (i.MX31 clocked at 533 MHz with 128 MB RAM) running a Linux-based operating system. This module also supports wireless Ethernet and is equipped with an upward-pointing camera and an omnidirectional camera with a resolution of 3 MP each.

The AR.Drone communicates with the marXbots using broadcasts over wireless Ethernet. Each signal broadcast by the AR.Drone is received by all marXbots at all times. Note that this is due to the wireless Ethernet-based broadcast communication we have chosen for the aerial robot in this particular study and not a general requirement of the presented protocol. The marXbots use their LEDs to locally broadcast situated signals by displaying different colors. LED signals displayed by the marXbot can be detected both by the AR.Drone (using its downward-pointing camera) and by other marXbots (using their omnidirectional camera). LED signal detection on both platforms is accomplished by detecting color blobs in the captured images.

In our experiments, we use one AR.Drone and up to ten marXbots. For safety reasons, we use a colorless, transparent plexiglass platform installed at 40 cm height from the ground to shield the ground robots from the aerial robot. Vision algorithms are not run directly on the AR.Drone, but are executed on a remote PC with video streamed from the AR.Drone in real time using the standard API. The same API is used to communicate the resulting position control data back to the AR.Drone in real-time (refer to (Krajník et al. 2011) for details). Experiments with larger number of robots are conducted in a physics-based simulator developed for multirobot simulation (Pinciroli et al. 2012).

4 Establishing a one-to-one STC link

In this section, we detail our proposed approach to establish a one-to-one STC link between an initiator robot and a target robot. We give a microscopic (i.e., a robot level) description of the approach using finite state machines (FSMs). From this microscopic description, we derive a macroscopic (i.e., a system level) model using a time-homogeneous Markov

chain. We analyze the scalability of the approach using data collected from physics-based simulations, real robot experiments, and predictions of the macroscopic model. We also demonstrate how an established one-to-one STC link can be applied to solve a real world task.

An STC link is established between an initiator and a target robot by means of an iterative elimination process. We assume a set $C := \{c_1, \dots, c_s : s \geq 3\}$ of distinct signals. The set C consists of signal c_1 , used to start and terminate the elimination process, and of the subset $C_s := \{c_2, \dots, c_s\}$ that contains the signals used in the iterative elimination process. In the heterogeneous multirobot system considered in the study, RGB colors are used as signals to define the sets $C := \{red, blue, green\}$ and $C_s := \{blue, green\}$. The initiator robot initializes the elimination process by emitting the signal c_1 . The robots that perceive the c_1 signal enter the elimination process by acknowledging with c_2 . Then, the initiator robot starts the elimination process by emitting a matching handshake signal c_2 . At each iteration, robots still in the process individually and independently choose a signal to emit from the set C_s . The initiator robot responds by matching the signal emitted by the target robot. At the end of the iteration, only those robots whose signal matches that of the initiator robot remain in the process. The other (non-matching) robots no longer emit any signal, and thus take no further part in the selection process. The elimination process continues until the target robot is the only robot remaining. At this point, the initiator robot indicates the termination of the elimination process by repeating c_1 . The target robot acknowledges the termination by signaling c_1 . The initiator robot has now established an STC link to the target robot.

In our experiments, robots are controlled by the behavior-based controllers shown in Fig. 3. The controller for the initiator robot is executed by the AR.Drone and the controller for the potential recipient robots by the marXbots. Both controllers are autonomous and are implemented as FSMs with $C := \{red, blue, green\}$.

4.1 Microscopic description of the elimination process

The FSM running on the AR.Drone, shown in Fig. 3a, consists of three states: STA (start); EP (elimination process); and LE (link established). When the AR.Drone needs to establish an STC link with a particular marXbot, it enters the STA state and transmits the signal *red* (c_1) to start the elimination process. The transition τ_{a1} is triggered if at least the target robot acknowledges the STA signal. This guarantees that the target robot is available for the elimination process and is not currently occupied with a different task. While in state EP, the AR.Drone continuously takes the transition τ_{a2} and matches the color displayed by the target robot. However, if the target robot is the only robot displaying a color because all other robots have been eliminated, the transition τ_{a3}

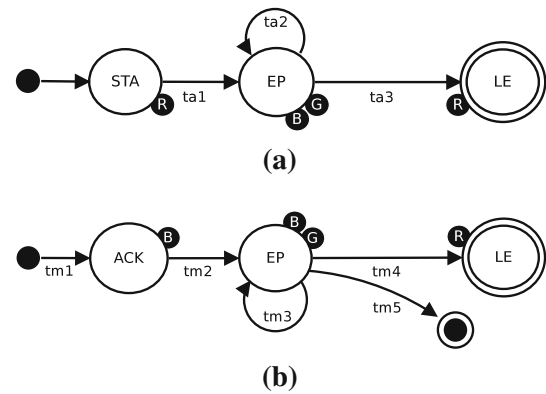


Fig. 3 Establishing a one-to-one STC link: FSMs running on **a** the AR.Drone, and **b** the marXbots. **R**, **G**, and **B** stand for the RGB color signals emitted in the respective states

taken to enter the state LE. The signal *red* is transmitted to confirm an established STC link to the target robot.

The controller executed by the marXbots is shown in Fig. 3b. Three states compose this FSM: ACK (acknowledge); EP (elimination process); LE (link established). When the signal *red* is received from the AR.Drone, a marXbot enters the ACK state by taking transition τ_{m1} . The color *blue* (c_2) is displayed to acknowledge the marXbot's availability to enter the elimination process. The transition τ_{m2} is triggered as soon as the AR.Drone matches the color *blue*. When entering the state EP, a marXbot randomly displays either *green* or *blue*. Simultaneously, it starts incrementing an internal counter t . Whenever t exceeds a fixed threshold τ , the marXbot examines the most recent color transmitted by the AR.Drone. If this color matches that displayed by the marXbot, it remains in the process by taking τ_{m3} . Otherwise, the marXbot terminates the behavior by taking τ_{m5} as a result of mismatching colors. Because all marXbots enter the state EP at the same time, and because they all use the same τ , their actions remain synchronised. The value of τ is chosen such that the AR.Drone has sufficient time to perceive, process, and respond to the colors displayed by the marXbots. If a marXbot still in the elimination process receives the signal *red* from the AR.Drone, it knows that all other robots have been eliminated and that it is indeed the target of the AR.Drone. In this case, the transition τ_{m4} is triggered and the marXbot confirms the termination of the elimination process by displaying the color *red*.

4.2 Macroscopic model of the elimination process

For a given number of robots N in the communication range of the initiator robot, we are interested in the number of iterations necessary to establish a one-to-one STC link with a target robot. In our model, each time step n corresponds to one iteration in the iterative elimination process. An iteration

ends when all remaining robots in the elimination process (i) have chosen and emitted a random signal, and (ii) have detected the match or mismatch of this signal with the one emitted by the initiator robot. Note that each iteration has a cost (i.e., duration of each iteration) that is independent of the number of robots in the system as the computation is done by each robot in parallel.

The iterative elimination process as described so far is a *memory-less* process. At each iteration, a potential recipient robot acts only on the basis of its current state and the current signal emitted by the initiator robot. Hence, the future state of the robot is independent of its past, fulfilling the Markov assumption (Kemeny and Snell 1976). Moreover, the synchronization provided by the timer mechanism described in the previous section allows us to consider the potential recipient robots from a macroscopic point of view. That is, we abstract from the states of individual robots and, instead, we focus on groups of robots that are in the same state simultaneously. As a consequence, we employ a time-homogeneous Markov chain to define a macroscopic model of the iterative elimination process. In particular, we use an absorbing Markov chain characterized by a set of transient states and one absorbing state. While the transient states represent all intermediate states of the iterative elimination process, the existence of an absorbing state guarantees the termination of the iterative elimination process in finite time.

We define our process as a Markov chain $\{X_n, n \in \mathbb{N}\}$ with $N + 1$ states. The random variable $X = x \in \Omega := \{1, \dots, N + 1\}$ represents the state of the process. Each state is characterized by the number $\eta_x = N + 1 - x$ of robots that are still part of the elimination process. All the states of the chain are transients except for $X = N + 1$ which is an absorbing state. In state $X = N + 1$, the number of robots involved in the process is $\eta_x = 0$, which corresponds to the completion of the elimination process. At time step n , the process will move from state $X_n = i$ to the next state $X_{n+1} = j$ with probability π_{ij} . We define the stochastic transition matrix $\Pi := (\pi_{ij}: i, j \in \Omega)$ of the Markov chain as

$$\begin{cases} \pi_{ij} = C_{\eta_i-1}^{\eta_j-1} p^{\eta_j-1} (1-p)^{\eta_i-\eta_j}, & \text{for } i < N, i \leq j \leq N, & \text{(1a)} \\ \pi_{ij} = 1, & \text{for } i \geq N, j = N + 1, & \text{(1b)} \\ \pi_{ij} = 0, & \text{otherwise.} & \text{(1c)} \end{cases}$$

Case 1a defines the transition probability between transient states by providing the probability π_{ij} that $\eta_i - \eta_j$ robots will leave the elimination process during the transition $i \rightarrow j$. This probability follows a binomial distribution with parameters $\eta_i - 1$, the number of robots that are still part of the elimination process without considering the target robot, and $p = 1/|C_S|$, the probability for one of them to emit the same signal as the target robot. Case 1b has a twofold significance. On the one hand, it states that the transition from the state

$X_n = N$ to the absorbing state $X_{n+1} = N + 1$ occurs with probability $\pi_{ij} = 1$. This deterministic step models the final handshake between the initiator and the target robot at the last iteration. On the other hand, it defines the state $X = N + 1$ as an absorbing state. Finally, Case 1c states that transitions not covered by previous conditions will never occur.

4.3 Model analysis

The stochastic transition matrix Π formally describes the dynamics of the iterative elimination process. Such a formal description allows us to predict the performance of the process. In what follows, we validate the accuracy of our macroscopic model by comparing model predictions with data acquired from physics-based simulations and real robot experiments. We use the model to analyze the scalability of the iterative elimination process. We first define the random variable ϑ as the number of iterations necessary to establish a one-to-one STC link to the target robot. Second, to study the distribution of ϑ , we consider the matrices Γ and Σ of the canonical decomposition of Π (Kemeny and Snell 1976). Γ provides the transition probabilities between transient states. The fundamental matrix Σ , given by $(I - \Gamma)^{-1}$ with I representing the identity matrix, gives the expected number of visits to each transient state.

4.3.1 Model validation

We validate the accuracy of the model’s predictions by comparing the distribution of ϑ with the empirical data acquired from simulation-based experiments. The cumulative distribution function $F(\vartheta) = P(\vartheta \leq n, x_0)$ can be obtained for a given initial state $X_0 = x_0$ as the infinite series

$$F_n(\vartheta) = 1 - \sum_{j \in \Omega} \Gamma_{x_0, j}^n. \tag{2}$$

The subcomponent $\sum_{j \in \Omega} \Gamma_{x_0, j}^n$ of Eq. 2 gives the probability that the process will be in a transient state at iteration n . The complement of this value provides the probability of entering the absorbing state prior to iteration n .

Figure 4a shows the cumulative distribution function $F(\vartheta)$ provided by the model together with the empirical distribution function $\hat{F}(\vartheta)$ obtained from 1,000 simulation runs for $N = 50$ and different values of $|C_S|$. The predictions of the model are shown to closely match the empirical observations for each value of $|C_S|$. Also the probability mass function $f(\vartheta) = P(\vartheta = n, x_0)$ shows a close agreement between the two data sets (see Fig. 4b), indicating reliable model predictions. The probability mass function is right-skewed. This implies the possibility of long executions of the elimination process. However, by increasing the number $|C_S|$ of signals available to the elimination process, the variance of ϑ shrinks

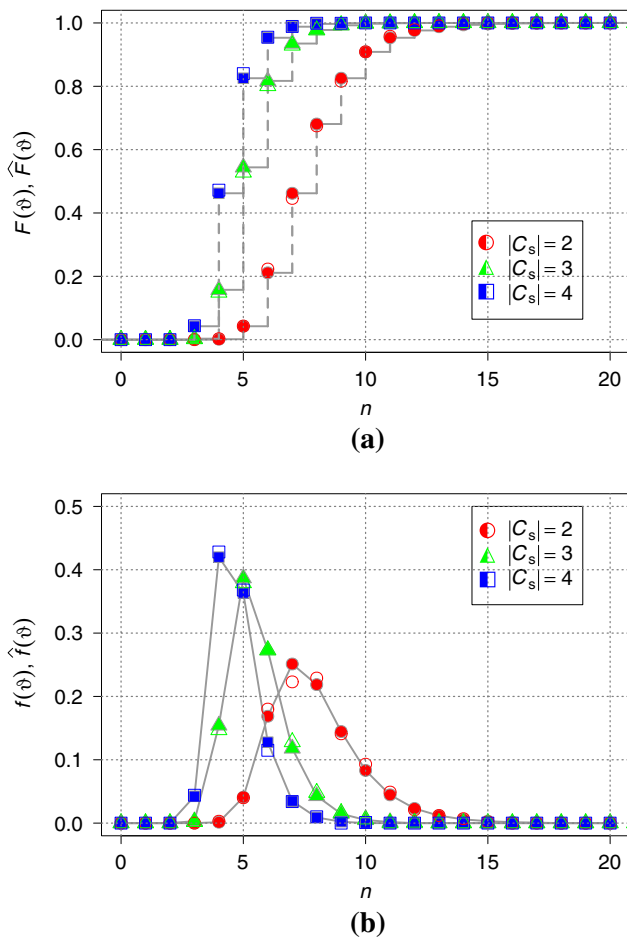


Fig. 4 **a** Cumulative distribution function $F(\vartheta)$ and **b** probability mass function $f(\vartheta)$ plotted against their empirical counterparts acquired from simulation: $\hat{F}(\vartheta)$ and $\hat{f}(\vartheta)$, respectively. Full symbols represent model predictions, empty symbols show the outcome of 1,000 simulation runs where $N = 50$

considerably reducing the probability mass under the right tail, and thus, the occurrence of long executions.

4.3.2 Real robot experiments

We carried out experiments with one AR.Drone placed on a plexiglass platform, and $N \in \{2, 4, 6, 8, 10\}$ marXbots distributed in a $1\text{ m} \times 1.5\text{ m}$ arena with $|C_S| = 2$ colors. For each value of N , we executed 30 runs resulting in a total of 150 runs. In each run, the marXbots were placed in the arena with random orientations and positions together with a light source embodying the object of interest to the AR.Drone. The AR.Drone identifies the marXbot closest to the object of interest as the target robot. The marXbots remain static while they execute the controller described in Sect. 4.1. The AR.Drone, on the other hand, executes the following behavior: (i) it elevates to a height of approximately 1.4 m above the plexiglass platform (i.e., ca. 1.8 m above ground

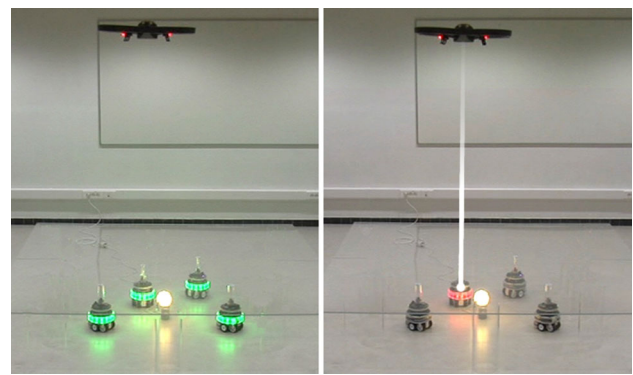


Fig. 5 Snapshots from a real robot experiment. Left an AR.Drone and four marXbots executing the one-to-one behavior. Right the AR.Drone establishes an STC link to the marXbot closest to the object of interest

level), (ii) hovers above the object of interest by executing a manually tuned PID controller that continuously minimizes the distance between the object of interest and the center of the image received from the downward-pointing camera (see Fig. 5 left), and (iii) it establishes a one-to-one STC link to the marXbot closest to the object of interest (see Fig. 5 right) before landing on the plexiglass platform. While hovering, we occasionally observed lateral drifts of the AR.Drone caused by impeded airflow in the arena. Therefore, we introduced an additional signal for our real robot experiments. The AR.Drone issued a *freeze* signal if not all marXbots in the experiment were in its field of view due to drifting. The marXbots responded to this signal by pausing their internal timers as long as the *freeze* signal was perceivable. Hence, the *freeze* signal prevents the fixed threshold τ (set to 200 ms) from being reached when the AR.Drone is unfavorably positioned (i.e., beyond the communication range of the marXbots) to continue the elimination process.² For each considered number N of marXbots, a video showing ten runs can be found in the online supplementary material (Mathews et al. 2013).

Figure 6 shows the results plotted as boxplots together with predictions of the model plotted as error bars. They show the expected value $\mathbb{E}[\vartheta]$ and the variance $\sqrt{\mathbb{V}[\vartheta]}$ of the number of iterations necessary to establish a one-to-one STC link. We compute $\mathbb{E}[\vartheta]$ and $\mathbb{V}[\vartheta]$ from the matrices Γ and Σ :

$$\mathbb{E}[\vartheta] = \xi \Sigma, \tag{3}$$

$$\mathbb{V}[\vartheta] = (2\Sigma - I)\mathbb{E}[\vartheta] - \mathbb{E}_{sq}[\vartheta]. \tag{4}$$

The term ξ in Eq. (3) represents a column vector of all 1s. That is, for every (transient) initial state, the expected value

² We have shown in previous work (Mathews et al. 2010a) that our approach can be implemented without the *freeze* signal if the initiator robot and the potential recipient robots do not leave each others communication ranges.

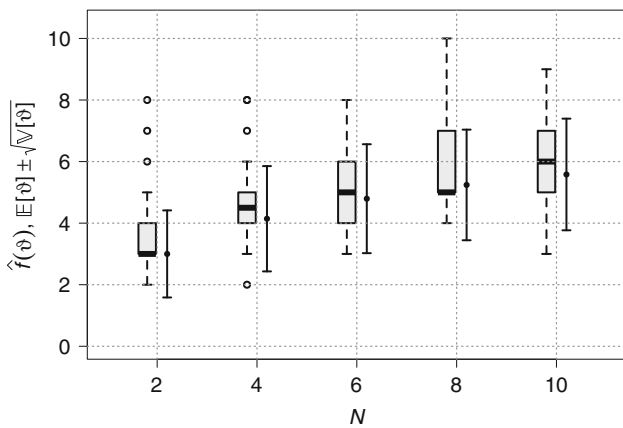


Fig. 6 Empirical probability mass function $\hat{f}(\vartheta)$ computed from real robot experiments (boxplots) plotted against model predictions (error bars) for varying sizes of N and for $|C_s| = 2$. Each boxplot represents 30 runs while error bars correspond to $\mathbb{E}[\vartheta] \pm \sqrt{\mathbb{V}[\vartheta]}$

of ϑ is given by the row sum of the fundamental matrix Σ . In Eq. (4), I denotes the identity matrix and $\mathbb{E}_{sq}[\vartheta]$ is $\mathbb{E}[\vartheta]$ with squared entries.

By means of Eqs. (3) and (4), we compare the empirical distribution of the results obtained from real robot experiments with the predictions of the Markov chain model defined above. The results in Fig. 6 show close agreement between the empirical observations and the theoretical predictions. In particular, the median values obtained from the real robot experiments (shown in boxplots) correspond closely to the theoretical expectation of ϑ . However, in the case of $N \in \{6, 8, 10\}$, the empirical observations have a larger variance than the theoretical predictions. This discrepancy can be explained by the limited number of observations gathered from real robot experiments. While the size of N in this set of experiments was increased by a factor of 5 (i.e., from 2 to 10 marXbots), the average number of iterations only experienced an increase by a factor of 1.6 (i.e., from 3.8 for 2 marXbots to 6.1 iterations for 10 marXbots). These results, thus, give a preliminary indication that the process scales well. Below, we further investigate scalability using the Markov chain model.

4.3.3 Scalability

We use Eq. (3) and Eq. (4) to study how the distribution of the expected value of ϑ scales for increasing values of N and $|C_s|$. As the results presented in Fig. 7 show, $\mathbb{E}[\vartheta]$ is characterized by a logarithmic trend that indicates high scalability for increasing values of N . The same trend applies to the variance of $\sqrt{\mathbb{V}[\vartheta]}$: given a certain value for $|C_s|$, $\mathbb{V}[\vartheta]$ grows logarithmically for increasing N .

The results also show that as $|C_s|$ increases, the variance of ϑ decreases considerably, increasing the reliability of the expected value as an aggregated indicator of the performance

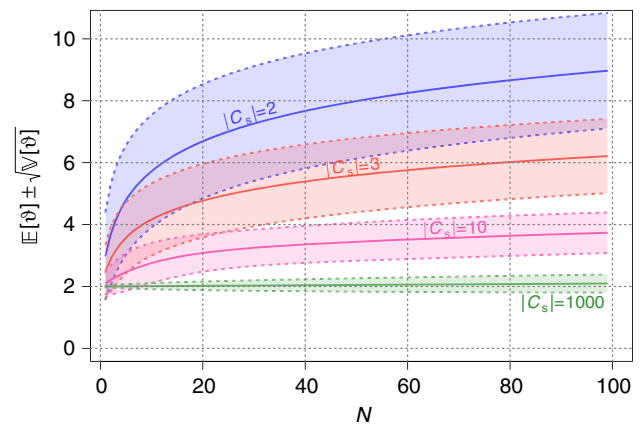


Fig. 7 Scalability study for increasing values of N and $|C_s|$. The shaded areas correspond to $\mathbb{E}[\vartheta] \pm \sqrt{\mathbb{V}[\vartheta]}$

of the process (e.g., compare the width of the shaded areas for $N = 80$ between $|C_s| = 2$ and $|C_s| = 1,000$ in Fig. 7). On the one hand, when the number of available signals tends to infinity, i.e., $|C_s| \rightarrow \infty$, the probability for a potential recipient robot to randomly choose the same signal as the target robot tends to zero. As a consequence, the iterative elimination process approaches a deterministic behavior that has exactly two iterations: at the first iteration all potential recipient robots except the target robot are eliminated, while in the second iteration the target robot performs the final handshake with the initiator robot. On the other hand, in an application scenario where the number of available signals is limited, the elimination process still ends in finite time as guaranteed by the absorbing state of the Markov chain. The number of iterations necessary to complete the process decreases as the number of available signals increases.

4.4 One-to-one STC application scenario

We study how a one-to-one STC link can be applied in a real world task using one AR.Drone and six marXbots. We consider a task that requires the AR.Drone to control morphology growth (Christensen et al. 2007) of the marXbots on the ground.

We deploy the robots on a mission towards an object of interest represented by a light source placed on the ground. The task consists of forming a specific morphology by multiple marXbots next to the object of interest. The size and shape of a suitable morphology are neither known to the marXbots nor can be determined from their vantage point. Hence, the marXbots rely on the AR.Drone to (i) determine a target robot to seed morphology growth, and (ii) to send the instructions needed to create the morphology required by the task. The AR.Drone flies to the object of interest in advance and waits for the marXbots to arrive. When all marXbots are in its field of view, the AR.Drone enters the state STA and

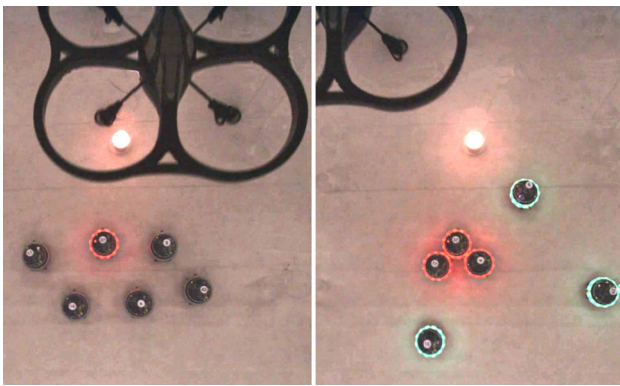


Fig. 8 Two frames captured from a ceiling camera of an experiment in which a one-to-one STC link is set up and used by the AR.Drone to send self-assembly related data: (left) STC link established to the marXbot closest to the object of interest; (right) the marXbot executes the instructions received from the AR.Drone to self-assemble an arrow-like morphology

issues the signal *red* that brings all marXbots on the ground to a halt. Then, the AR.Drone establishes a one-to-one STC link with the marXbot closest to the object of interest (see Fig. 8 left). Subsequently, the established STC link is used to transmit morphology control instructions (Christensen et al. 2008). The target robot receives and executes the instructions to grow an arrow-like morphology (see Fig. 8 right) using the connection forming mechanism presented in (Mathews et al. 2011). Note that while the instructions for this particular morphology were preloaded on the AR.Drone, we have shown in our previous work (Mathews et al. 2012b) that suitable task-dependent morphologies can be determined by aerial robots based on observed environmental features. In this particular experiment, the AR.Drone was flown manually while the communication with the marXbots was entirely autonomous.

In the experiment, the STC link was established with three iterations of the elimination process or within 600 ms (excluding the time during which the *freeze* signal was issued by the AR.Drone) as the fixed threshold τ was set to 200 ms, and hence, $3 \times 200 \text{ ms} = 600 \text{ ms}$. The successful execution of this proof-of-concept experiment demonstrates how real robots can use STC to achieve spatial coordination. Video footage of this experiment can be found in the online supplementary material (Mathews et al. 2013).

5 Establishing a one-to-many STC link

In multirobot systems, spatially targeted messages often need to be conveyed to groups of robots. In this section, we present an extension to our protocol that allows an initiator robot (the AR.Drone in our case) to expand an existing one-to-one STC link to a one-to-many link. To communicate with a group of co-located robots, the initiator first establishes a one-to-one

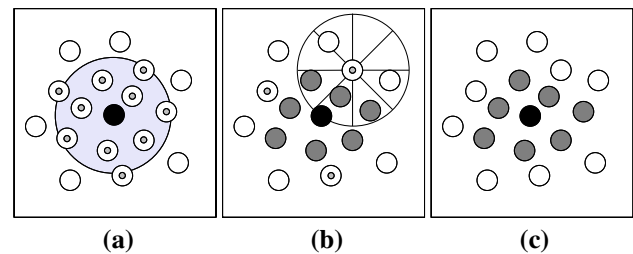


Fig. 9 The exclusion mechanism executed by candidate robots to achieve cohesive target groups. ● = member, ● = closest candidate robot, ⊙ = candidate robot, and ○ = hibernating robot. **a** Candidate robots within communication range of a robot already in the group. **b** Division of perception range into eight equally sized sections visualized on a candidate robot. Three candidate robots exclude themselves and do not become closest candidate robots, as they detect another candidate robot closer to the target robot in a single section. **c** The target robot and its remaining closest candidate robots

STC link to a favorably located robot and then iteratively expands the link to include more neighbors. Note that we are not interested in which individual robots are in the target group, but only in the size of the group and its spatial cohesiveness. We validate the model using simulation-based studies and extensive experimentation on real robots. We present the results of our scalability studies. Finally, we present an experiment exemplifying the usefulness of one-to-many STC in real world tasks.

Given a set $C := \{c_1, c_2, c_3\}$ of distinct signals, a one-to-one STC link between an initiator and a target robot can be expanded into a one-to-many STC link through an iterative growth process. At the start of the process, the target robot is the only member of the target group. At each iteration, the initiator robot may emit the signal c_3 to request a growth of the target group. From the robots that perceive this signal, those that have a target group member within perception range respond with signal c_2 . We refer to these robots as *candidate robots*. Next, each candidate robot determines if any other robot is located between itself and the closest target group member, and if so, leaves the process. We refer to the remaining candidate robots as *closest candidate robots*. As shown in Fig. 9, this exclusion mechanism amongst the candidate robots ensures the spatial cohesiveness of the target group. The closest candidate robots communicate their candidacies to the initiator robot by emitting c_3 . If the number of the closest candidate robots plus the number of current group members (hereafter referred to as the “*potential target group size*”) exceeds the target group size, the initiator robot issues the signal c_2 to request the closest candidate robots to relinquish their candidacies probabilistically. The c_2 signal is issued until the potential target group size is smaller or equal to the target group size. Otherwise, if the potential target group size does not exceed the target group size, the initiator robots grants group membership to the closest

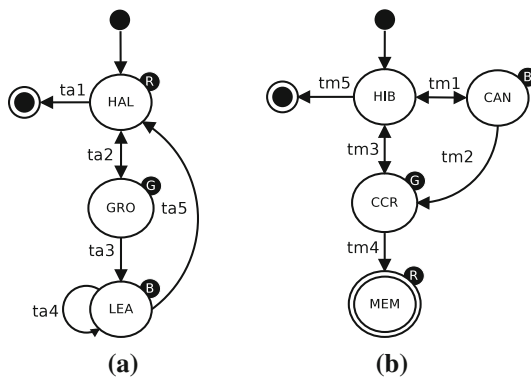


Fig. 10 Establishing a one-to-many STC link: FSMs running **a** on the AR.Drone and **b** the marXbots. The RGB color signal emitted in each state is indicated by **R**, **G**, or **B**

candidate robots by emitting c_1 . That is, the initiator robot may halt the growth process at an intermediate group size that is smaller than the target group size. Subsequently, the initiator robot restarts the growth process. This incremental growth is repeated until the target group size is reached.

5.1 Microscopic description of the growth process

The FSM of the initiator robot (the AR.Drone) and the FSM of the potential recipient robots (the marXbots) are shown in Fig. 10. In what follows, we detail both robot controllers under the assumption that $C := \{red, blue, green\}$. Note that the iterative growth process requires potential recipient robots to communicate with each other. To this end, the marXbots use their LEDs and omnidirectional cameras.

Figure 10a shows the FSM implemented on the AR.Drone. It consists of three states: HAL, to halt the growth process; GRO to request growth of target group size; and LEA, to request closest candidate robots to leave the process probabilistically. The AR.Drone first enters the HAL state and sends the signal *red*. If the current group size matches the target group size, the τ_{a1} transition is triggered and the AR.Drone terminates the execution of the controller. By contrast, if the current group size is smaller than the target group size, transition τ_{a2} is taken and the AR.Drone enters the GRO state and transmits the signal *green* to signal a growth of the group to the marXbots. From the state GRO, the AR.Drone can move to two states as needed if no candidate robot is perceivable (i.e., there are no more ongoing executions of the exclusion mechanism). If the potential target group size is equal or less than the target group size, transition τ_{a2} is taken to return to state HAL and to halt the growth process at an intermediate group size. Alternatively, by moving to state LEA through transition τ_{a3} and transmitting *blue*, excessive closest candidate robots can be requested to probabilistically relinquish their candidacies. The AR.Drone iterates in the LEA state by taking transition τ_{a4} until the potential target group size

is equal or less than the target group size. If this condition is satisfied, the AR.Drone triggers the transition τ_{a5} to the HAL state to grant membership to remaining closest candidate members.

Figure 10b shows the controller running on the marXbots. It consists of four distinct states: HIB, hibernate; CAN, candidate robot; CCR, closest candidate robot; and MEM, member of the target group. The LEDs are switched off while in state HIB, whereas the *blue* color is illuminated while in state CAN, *green* while in state CCR, and *red* while in state MEM. The controller is initiated in the HIB state. The transition τ_{m5} enables robots to leave the growth process permanently when the AR.Drone indicates an excess of closest candidate robots using the signal *blue*. When the signal *green* is received from the AR.Drone, marXbots that are able to detect at least one target group member take the transition τ_{m1} to enter the CAN state. In this state, the following procedure is executed to maintain group cohesion. As shown in the example in Fig. 9b, each candidate robot divides its 360° view of the environment into eight equally sized sections with an angular range of 45° each. Subsequently, transition τ_{m1} is taken to leave the growth process if another candidate robot is perceived closer than the nearest target group member in one of the sections. Otherwise, transition τ_{m2} is triggered to enter state CCR. When in state CCR, a marXbot responds to a received *blue* signal (indicating too many closest candidate robots) from the AR.Drone by considering the outcome of an independent Bernoulli trial with success probability $p = 0.5$. In case of success, the marXbot stays in the state CCR, otherwise it takes the transition τ_{m3} to HIB state. In state HIB, a single bit of memory is used to store a binary value that represents the state prior to reaching HIB: CAN or CCR. This allows closest candidate robots from previous iterations to re-enter the state CCR by taking the transition τ_{m3} while avoiding the CAN state entirely.³ Finally, a marXbot in state CCR triggers transition τ_{m4} when it receives a *red* signal from the AR.Drone.

5.2 Macroscopic model of the growth process

We are interested in the time required for the initiator robot to grow a one-to-one link into a one-to-many STC link to a target group of size D . For the same reasons presented in Sect. 4.2, we consider time to be discrete. For clarity, we refer to the exemplary growth process illustrated in Fig. 11 with $N = 25$ and $D = 12$, where we conduct the analy-

³ When the intermediate group size is close to the target group size, the AR.Drone may need to repeatedly request the same set of marXbots to candidate and withdraw candidacies probabilistically until the target group size is reached. Storing the prior state allows robots in state HIB that were previously in state CCR to respond to such a request by avoiding the re-execution of the exclusion mechanism and thus reduces the wall clock time of an iteration.

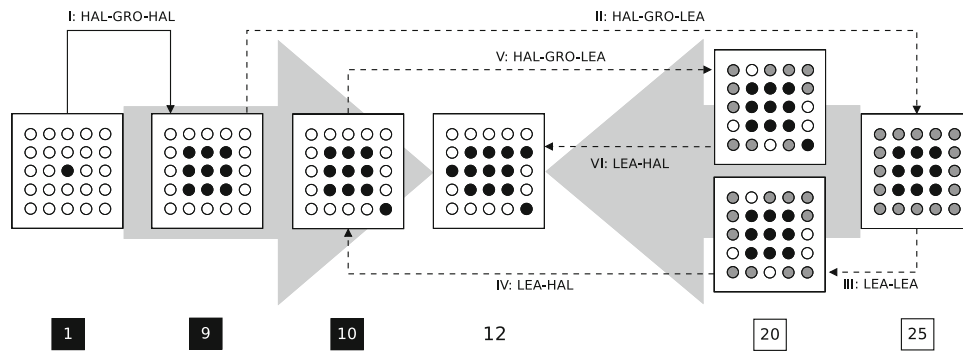


Fig. 11 A schematic of an iterative growth process in a square lattice where $N = 25$ and $D = 12$. Iteration in the deterministic phase is indicated by a *solid arrow*, and iterations in the stochastic phase are indicated by *dashed arrows*. Intermediate group sizes (*filled boxes*) and

potential target group sizes (*empty boxes*) are given in the *bottom*. The *opposing arrows* depict confining upper and lower bounds. ● = MEM, ◐ = CCN, ○ = HIB

sis assuming the potential recipient robots to be distributed in a square lattice with each robot able to perceive at least the robots in its Moore neighborhood. In Sect. 5.3, we assess the impact of such an assumption on model predictions and discuss possible relaxations. In what follows, we detail three concepts underlying the Markov chain model before presenting the model: iterations, growth phases, and the interaction graph.

Iterations: we characterize the number of iterations required by the initiator robot to reach target group size D . We define each step n in the Markov chain as an occurrence of one of the following four state transition sequences of the initiator robot:

HAL→GRO→HAL, HAL→GRO→LEA, LEA→LEA, LEA→HAL.

This set of state transition sequences has three important properties. Firstly, it is complete, that is, any possible run of the growth process (i.e., for any D and N) can be decomposed into a series of these state transition sequences (see FSM-based description of growth process in Sect. 5.1). Figure 11 shows a growth process decomposed into this set of state transition sequences—in the example illustrated in the figure, six iterations are required to reach target group size. Secondly, all of these state transition sequences have the same time and computation costs, making it meaningful to analyse the number of iterations. To see why the cost is constant, note that each state transition sequence ends with a transition that is triggered when the number of closest candidate robots decreases or increases in the system. Consequently, the cost of detecting an iteration by the AR.Drone can be kept constant by applying a color-based similarity matching technique (Goshtasby 2012) of constant cost on two consecutive images retrieved from its downward-pointing camera. Thirdly, characterizing iterations using this set of state transition sequences disregards candidate robots from the modeling process entirely. The state space of our Markov chain

thus becomes simpler as it does not need to represent the states of the candidate robots.

Growth phases: we distinguish two distinct phases. The first phase is deterministic and is composed of a series of HAL→GRO→HAL steps that occur repeatedly until the potential target group size is greater than or equal to D . Each of these steps is deterministic, as at each step, the number of closest robots is a priori predictable as a function of the relative positioning of the potential recipient robots. If the potential target group size equals D , then the process terminates. The second phase is stochastic, and starts if and when the potential target group size exceeds D . Each step in this phase is an occurrence of one of the three sequences (HAL→GRO→LEA, LEA→LEA, LEA→HAL). These steps are stochastic because they each contain a probabilistic response to a leave request (LEA). This stochastic phase continues until the target group size is reached (see iterations II–VI in Fig. 11). During the stochastic phase, the potential target group size acts as a monotonically decreasing upper bound on the number of robots involved in the growth process. The intermediate group size always (i.e., also during the deterministic phase) represents a monotonically increasing lower bound on the number of robots involved in the growth process. The two bounds meet at the target group size (see the opposing arrows in Fig. 11).

We model the deterministic and stochastic phases of the iterative growth process separately. We can thus write $D = D_{det} + D_{sto}$, where D_{det} is the number of robots added to the intermediate group during the deterministic phase, and D_{sto} is the number of robots added to the intermediate group during the stochastic phase.

Interaction graph: to derive a non-spatial characterization of the growth process, we consider an interaction graph in which two connected nodes correspond to a target group member and one of its closest candidate robots. Using such an

interaction graph, we further consider a sequence $\{a_0, a_1, \dots\}$ of which each element corresponds to the total number of closest candidate robots available at each intermediate group size. For the intermediate group sizes 1, 9, and 10 shown in the example in Fig. 11, the sequence corresponds to $\{8, 16, 15\}$. At intermediate group size 1, for instance, the total number of closest candidate robots is 8. This is a result of the exclusion mechanism shown in Fig. 9 that limits a group member’s closest candidate robots to its Moore neighborhood regardless of communication range.

5.2.1 Deterministic phase model

In an interaction graph resulting from a square lattice distribution with Moore neighborhood, each node (i.e., a target group member) is connected to 8 neighbors (i.e., has 8 closest candidate robots). The sequence of closest candidate robots for intermediate group sizes is thus given by $\{8k, \forall k \in \mathbb{N} > 0\}$ in an infinite lattice while the number of iterations required to exhaust the deterministic phase is

$$\arg \max_{k>0} \left(1 + \sum_{i=1}^k 8i < D \right). \tag{5}$$

At the end of a deterministic phase, the target group size is defined by $D = D_{det} = 1 + \sum_{i=1}^k 8i$.

5.2.2 Stochastic phase model

We define a Markov chain $\{Y_n, n \in \mathbb{N}\}$ with the random variable $Y \in \Omega := \{1, \dots, w\}$ that represents the state of the growth process and Ω that represents the state space with $w = 2(N - D)(D - D_{det}) + 1$ states. Each state $Y = y$ is characterized by:

- η_y^m , current number of target group members;
- η_y^c , current number of closest candidate robots;
- $\eta_y^{c'}$, number of closest candidate robots at the previous iteration;
- η_y^h , number of hibernating robots.

Although the growth process uses a single bit of memory, we apply the formalism of time-homogeneous Markov chains using a chain of order 2. The memory is modeled using the variable $\eta_y^{c'}$: when the memory is necessary, $\eta_y^{c'}$ equals the number of closest candidate robots at the previous step of the chain, when it is not necessary, $\eta_y^{c'} = -1$. All states are transient with the exception of $Y = y$ such that $\eta_y^m = D, \eta_y^c = 0, \eta_y^h = N - D, \eta_y^{c'} = -1$ which is the only absorbing state.

At step n , the process on the chain will move from the current state $Y_n = i$ to the next state $Y_{n+1} = j$ with probability π_{ij} . We define the stochastic transition matrix

Table 1 The adjacency conditions between states i and j of the stochastic transition matrix Π_Y

	η_j^m	η_j^c	η_j^h	$\eta_j^{c'}$
$c_{1.1}$	$=\eta_i^m$	$=0$	$=\eta_i^c + \eta_i^h$	$=\eta_i^c$
$c_{1.2}$	$=\eta_i^m$	$\leq \eta_i^c$	$=\eta_i^c + \eta_i^h - \eta_j^c$	$= -1$
$c_{1.3}$	$> \eta_i^m$	$=0$	$-$	$=(\eta_i^c + \eta_i^m - \eta_j^m) \vee -1$
c_2	$=\eta_i^m$	$\leq \eta_i^c$	$=\eta_i^h - \eta_i^c$	$= -1$
c_3	$=D$	$=0$	$=N - D$	$= -1$

$\Pi_Y := (\pi_{ij} : i, j \in \Omega)$ of the chain $\{Y_n, n \in \mathbb{N}\}$ as

$$\begin{cases} \pi_{ij} = C_r^q p^q (1 - p)^r, & \text{if } c_{1.1} \vee c_{1.2} \vee c_{1.3}, & (6a) \\ \pi_{ij} = 1, & \text{if } c_2 \vee c_3, & (6b) \\ \pi_{ij} = 0, & \text{otherwise,} & (6c) \end{cases}$$

where the adjacency conditions $c_{1.1}, c_{1.2}, c_{1.3}, c_2, c_3$ between states i and j are summarized in Table 1. Case 6a corresponds to a binomial distribution where $r = \eta_i^c$, i.e., the number of closest candidate robots withdrawing candidacies probabilistically (in state $Y_n = i$) with success probability p , and $q = \eta_j^c + \eta_i^m - \eta_j^m$, i.e., the number of closest candidate robots retaining candidacies (in state $Y_{n+1} = j$). The adjacency conditions $c_{1.1}, c_{1.2}, c_{1.3}$ define all possible states i and j such that the transition $i \rightarrow j$ corresponds to an iteration $LEA \rightarrow LEA$ or $LEA \rightarrow HAL$. Case 6b defines the transition probability between adjacency conditions c_2 and c_3 to 1. The adjacency condition c_2 covers all possible pairs of states (i, j) such that the transition $i \rightarrow j$ corresponds to an iteration $HAL \rightarrow GRO \rightarrow LEA$. The adjacency condition c_3 defines the absorbing state of the system. As defined in Case 6c, transitions not covered by the adjacency conditions never occur. Note that this Markov chain model can be applied to any spatial distribution of the robots given the interaction graph by solely changing the adjacency conditions.

5.3 Model analysis

We analyze the performance of the growth process through the canonical decomposition of Π_Y in the transition probability matrix Γ_Y and the fundamental matrix Σ_Y . We define the random variable φ as the number of iterations necessary to reach a target group size D . That is, $\varphi = \varphi_{det} + \varphi_{sto}$, where φ_{det} and φ_{sto} account for the number of iterations during the deterministic and stochastic phase, respectively. φ_{det} is given by Eq. (5), while φ_{sto} is derived from the Markov chain model that provides the number of state transitions before the process reaches its only absorbing state. We analyze the cumulative distribution function $F(\varphi)$ using Eq. (2) and the variance $\mathbb{V}[\varphi]$ using Eq. (4), while the expected value is given by

$$\mathbb{E}[\varphi] = \arg \max_{k>0} \left(1 + \sum_{i=1}^k 8i < D \right) + \xi \Sigma_Y, \tag{7}$$

where ξ is a column vector with all values set to 1.

In what follows, we first assess the reliability of the Markov chain model assuming a Moore neighborhood-based interaction graph with eight neighbors per node. We compare model predictions against data collected from simulation-based studies. Second, we assess the quality of our spatial assumption using data acquired from both real robots experiments and simulation-based experiments in which we vary the spatial distribution of the robots from a square lattice to a random distribution. We then present the results of our scalability studies of the growth process.

5.3.1 Model validation

We validate the reliability of the Markov chain model using multiple probability–probability (P–P) plots, in which we compare the cumulative distribution function $F(\varphi)$ with the empirical distribution $\hat{F}(\varphi)$ collected from simulation for varying sizes of potential recipient robots N and target group size D .

In Fig. 12, we present the results using nine different P–P plots organized in a 3×3 matrix. Plots in the same row have identical N values while plots in the same column have target group size $D = D_{det} + d \cdot (N - D_{det})$, where d is a constant fraction of robots remaining in the stochastic phase $(N - D_{det})$. For each configuration of parameters N and D , we execute 1,000 independent simulation runs and record the number of iterations required before D was reached. As a P–P plot is confined to the unit square, the distributions under examination agree when the points in the plot lie on the diagonal of the square. In all nine plots shown in Fig. 12, the function $(F(\varphi), \hat{F}(\varphi))$ produces data points in the close proximity of the diagonal of the respective squares indicating a good agreement between theoretical and empirical distributions for each combination of parameters considered. We depend on this observation to confirm the reliability of our model.

5.3.2 Real robot experiments and spatial assumptions

We conducted a series of experiments with real robots to study the duration of the growth process. In particular, we studied how a random distribution of the robots may affect the duration of the growth process with respect to the model predictions that assume an interaction graph-based on square lattice distribution. We assumed an already established one-to-one STC link between the AR.Drone and one marXbot. We performed experiments with $N = 9$ marXbots and one AR.Drone with an experimental setup similar to that presented in Sect. 4.3.2. However, unlike the experiments

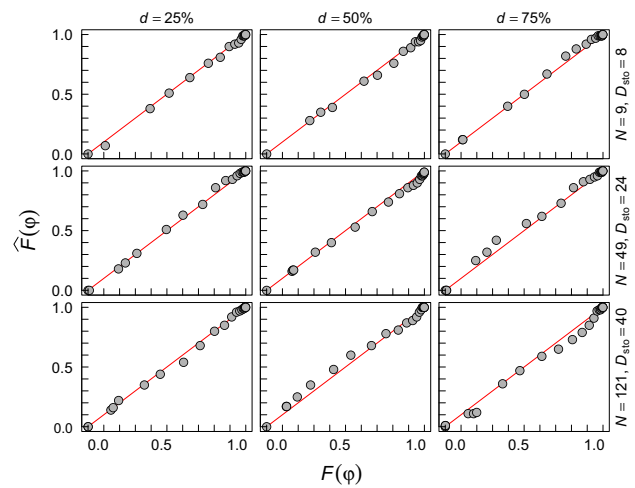


Fig. 12 P–P plots comparing theoretical and empirical cumulative distribution functions of φ with $N \in \{9, 49, 121\}$ and $D = D_{det} + d \cdot (N - D_{det})$ with $d \in \{25, 50, 75\% \}$. Each data point is an average of 1,000 simulation runs

presented in Sect. 4.3.2, no light source was considered in this set of experiments. The AR.Drone therefore used the center of the bounding box that includes all marXbots in the field of view as the input to its PID controller. The *freeze* signal was issued when fewer than nine marXbots were visible to the AR.Drone. We considered two spatial distributions of the marXbots: random (Fig. 13a left) and a square lattice with a side length of 0.4 m (Fig. 13a right). For each spatial distribution, we varied the target group size D in the set $\{2, 4, 8\}$. We executed 30 repetitions for each combination of parameters resulting in a total of 180 runs. Video footage of this series of experiments is available in the online supplementary material (Mathews et al. 2013).

Figure 13b is a plot of the results obtained using random distribution, square lattice distribution, and model predictions. The results show that the median value of the square lattice distribution and the expectation value of the model predictions for $D = 2$ and $D = 8$ do not vary significantly. Also, the amplitude of the standard deviation is well approximated. In the case of $D = 4$, however, the empirical data does not correspond well to the theoretical expectation. This may be explained by random fluctuations of the estimated distributions caused by the limited sample size of 30 runs. When comparing the results obtained using random distributions, we observe the growth process to terminate minimally faster than the expectation value of the model. This observation is a direct consequence of the sizes chosen for D . When nine marXbots are distributed in a square lattice around the target robot, all eight neighbors of the target robot are closest candidate robots and hence no deterministic iteration of the type HAL→GRO→HAL can occur. By contrast, when distributed randomly, some marXbots may occlude others and hence render intermediate group sizes more probable.

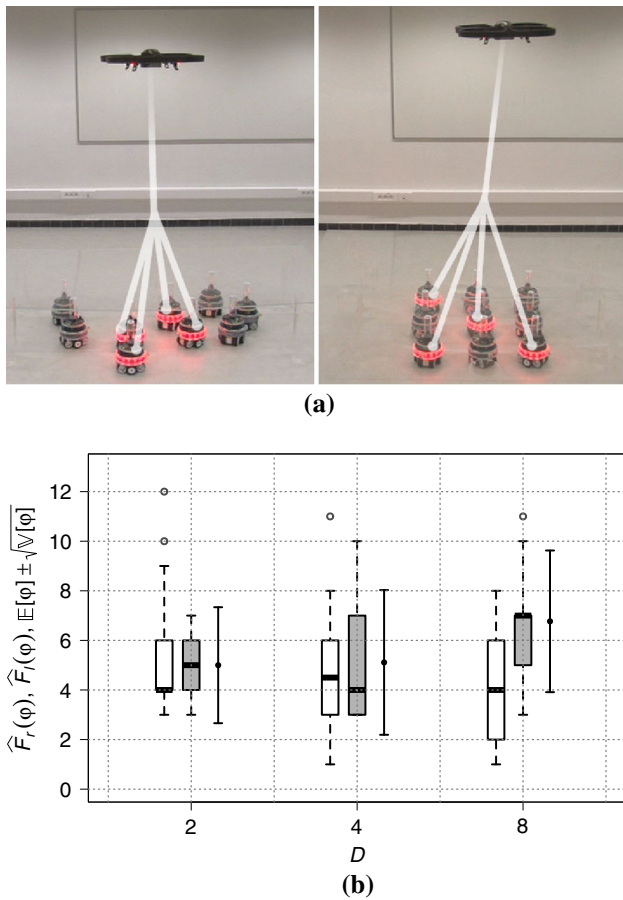


Fig. 13 **a** Snapshots of experiments where $N = 9$ and $D = 4$: (left) random distribution and (right) square lattice distribution. **b** Empirical cumulative distribution functions of real robot experiments with $\hat{F}_r(\varphi)$ for randomly distributed marXbots, $\hat{F}_l(\varphi)$ for marXbots distributed in a square lattice plotted against theoretical expectation $\mathbb{E}[\varphi] \pm \sqrt{\mathbb{V}[\varphi]}$ for $N = 9$ and $D \in \{2, 4, 8\}$. Each boxplot represents 30 runs. Empty boxes are for $\hat{F}_r(\varphi)$ and filled boxes are for $\hat{F}_l(\varphi)$. The error bars represent the model predictions

Consequently, the AR.Drone may reach the target group size with deterministic phase iterations only, and may avoid stochastic phase iterations entirely. We also confirm this observation using the video footage of the experiments (Mathews et al. 2013). We conclude that for small N , the square lattice-based interaction graph underlying the Markov chain model overestimates the duration of the growth process as random distributions often do not result in the dense, structured neighborhoods inherent to the square lattice.

In simulation-based studies, we further investigated the difference in the duration of the growth process between a random distribution and a square lattice, and compare the results with the predictions of the model for increasing $N \in \{9, 25, 49, 81, 121\}$. Figure 14 shows the results in which the target group size $D = D_{det} + 50\% \cdot (N - D_{det})$ is set to a constant fraction of the stochastic phase. To mitigate the effect of random fluctuations, we performed 1,000

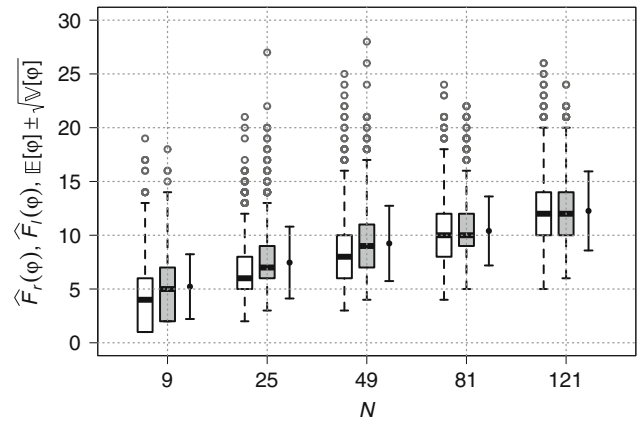


Fig. 14 Empirical cumulative distribution functions computed for $N \in \{9, 25, 49, 81, 121\}$ and $D = D_{det} + 50\% \cdot (N - D_{det})$ with 1,000 simulation runs each. $\hat{F}_r(\varphi)$ represents randomly distributed marXbots, $\hat{F}_l(\varphi)$ represents square lattice distribution, and the theoretical expectation $\mathbb{E}[\varphi] \pm \sqrt{\mathbb{V}[\varphi]}$

simulation runs for each parameter configuration. The results show that for smaller $N \in \{9, 25, 49\}$, the growth process terminates faster when robots are randomly distributed. For larger $N \in \{81, 121\}$, the Markov chain model provides a good approximation of the outcome of the simulations regardless of the spatial arrangement of the potential recipient robots. This can be explained by the high density of robots that makes it more probable for a robot to perceive at least one neighbor in each of its eight sections of perception. The average degree of connectivity in the resulting interaction graph may hence resemble that of a square lattice with eight nodes. In summary, the data shows that even if the spatial assumption underlying the Markov chain model is not met, the model provides a reliable and quantitative approximation of the duration of the growth process provided the potential recipient robots are densely packed.

5.3.3 Scalability

An STC link to a target group can be established by executing the elimination process presented in Sect. 4 followed by either the growth process or by repeating the elimination process multiple times such that one robot is added to the target group after each repetition. Using the Markov chain model, we study the scalability properties of both these approaches by computing the expectation value of φ where $N = 121$ and $D \in \{2, \dots, 100\}$.

In Fig. 15, we present the results of the study in which we also vary the number of distinct signals C_s available to the iterative elimination process. The results show that the iterative growth process clearly outperforms the iterative elimination process regardless of the size $|C_s|$. This dominance still holds when the theoretical case is considered in which $|C_s|$ tends to infinity. For larger values of D , we observe that the

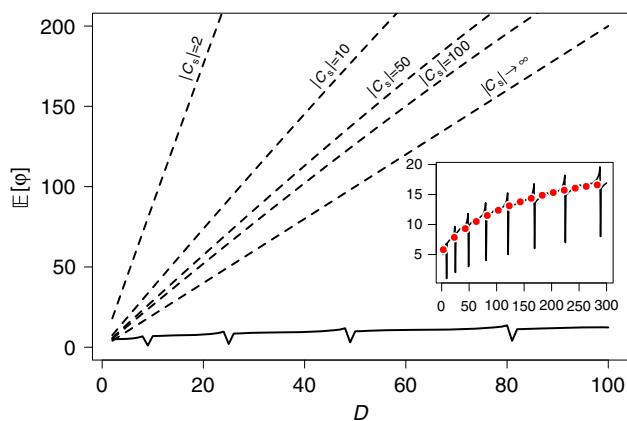


Fig. 15 Scalability study showing the expectation value $\mathbb{E}[\varphi]$ for $N = 121$ robots and target group size $D \in \{2, \dots, 100\}$ in a square lattice distribution. Results obtained by repeating the elimination process with varying $|C_s|$ (dashed line) are plotted against those obtained from the growth process (solid line). The inset shows the performance of the growth process for larger values of D together with the fitted values shown in circular markers

iterative growth process scales approximately as $D \cdot \log(D)$ (see fitted values shown in the inset in Fig. 15). When D can be reached through iterations in the deterministic phase only (e.g., $D \in \{9, 25, 49, \dots\}$) the iterative growth process terminates faster than for other values of D which require iterations to occur in the stochastic phase resulting in the spikes shown in Fig. 15 and its inset. Although establishing an STC link to a group of robots through the iterative growth process is faster in all cases considered, growing a target group by incrementally adding one robot after the other may still remain an option in applications that require STC links to particular robots rather than to a cohesive group of a specific size.

5.4 One-to-many STC application scenario

We consider a segregation task as an example of a real world application that requires spatial coordination among the robots. In this experiment, we deploy eight marXbots and one AR.Drone on a mission in which a subgroup consisting of four marXbots closest to an object of interest needs to be segregated from the rest of the group and sent to the object of interest. We use a light source placed on the ground as the object of interest. The marXbots are, unlike the AR.Drone, neither aware of the total number of robots in the system nor of the number of robots that may be required for upcoming tasks. After deployment, the AR.Drone follows the group of marXbots until the object of interest is in its field of view (see Fig. 16 left). The AR.Drone then establishes a one-to-one STC link to the marXbot closest to the object of interest, before launching the growth process to expand the link to grow a target group that includes the

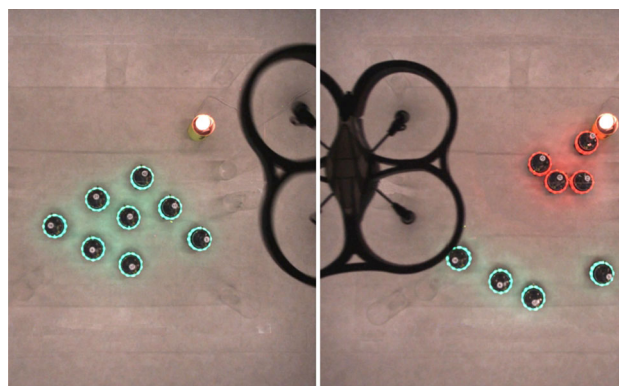


Fig. 16 Two top-down snapshots of the segregation experiment. *Left* the group of robots reaches the object of interest. *Right* the AR.Drone uses STC to segregate the four marXbots closest to the object of interest from the group and to direct them towards the object

four marXbots closest to the object of interest. Finally, the AR.Drone uses the established one-to-many STC link to achieve segregation between the two groups by broadcasting instructions (Christensen et al. 2008) that require the target group to perform a phototaxis behavior while the remaining robots are commanded to execute a retreat behavior (see Fig. 16 right). In this experiment, the marXbot behavior is fully autonomous. The AR.Drone performs autonomous sensing and communication, but was flown manually.

In the proof-of-concept experiment we conducted, the one-to-one STC was established within six iterations (i.e., within $6 \times 200\text{ms}=1,200\text{ms}$, as the fixed threshold τ was set to 200 ms). The growth process took only one iteration of type HAL→GRO→HAL to form the required target group (i.e., within $3 \times 300\text{ms}=900\text{ms}$ as marXbots use three sequentially retrieved frames from their omnidirectional cameras for the decision-making required in the exclusion mechanism shown in Fig. 9 while a frame is retrieved each 300 ms). The wall clock time given in ms discounts the time during which not all eight marXbots are in the field of view of the AR.Drone and the freeze signal is issued. A video recording of the experiment is available in the online supplementary material (Mathews et al. 2013). Further examples of applications for one-to-many STC can be found in our previous work (Mathews et al. 2012a).

6 Discussion

The STC protocol as described in this paper requires that any potential recipient robot in the communication range of the initiator robot must also be able to communicate back to the initiator robot. If this condition is not respected, it creates a problem of asymmetrical communication: potential recipient robots can respond to the initiator robot without the initiator robot being aware of them. As a result, in some

cases the initiator robot can believe that it has established a one-to-one link with a single target robot when in fact it has unknowingly selected multiple target robots. Similarly, when establishing a one-to-many link extra potential recipient robots can add themselves to the forming group without the initiator's knowledge.

With the hardware we use in our experiments there is an asymmetry between the communication ranges of the initiator and potential recipient robots: the AR.Drone transmits using wireless Ethernet and can therefore send signals to all the marXbots. The marXbots transmit using LEDs, thus the only marXbots that can send signals to the AR.Drone are those in the AR.Drone's field of vision. In our experimental setup we have avoided the asymmetrical communication problem by placing all the marXbots within the AR.Drone field of vision. In practice, this means that in our experiments all communication in the system is global.

Instead of solving the asymmetric communication problem by using global communication to establish a one-to-one STC link it would be enough to let the initiator robot use the macroscopic model (see Sect. 4.2) to calculate the probability that only one target robot has been selected. This calculation would use the number of elimination iterations already conducted together with an estimate of the total number of robots⁴ that can receive the initiator robot's signals to produce the required probability. Should this probability be not low enough, the protocol could be repeated for an additional number of iterations until the probability falls below the required threshold.

This solution does not extend to the establishment of one-to-many STC links. In some cases a behavioral solution could solve the problem. With the hardware we use, for example, the AR.Drone could ensure that robots outside of its field of vision cannot become part of the group. This can be achieved by initiating and then expanding the group in such a way that the boundaries of the group never expand beyond its own field of vision. Even in cases in which such behavioral solutions are not feasible, small errors in the size of the group will often not be important, especially given that the resulting group will anyway be spatially coherent. This is for example true in almost all swarm robotics applications (Nouyan et al. 2009; Dorigo et al. 2013, 2014).

A remaining issue with the applicability of the STC protocol, even in its extended form, to our hardware would be if two or more AR.Drones wished to establish STC links in parallel. The fact that they each use non-situated broadcast communication (wireless Ethernet) would make it impossi-

ble for the marXbots to distinguish between signals from the different initiator robots. However, this is no longer an issue when both the initiator and potential recipient robots use situated communication. In previous work we have shown the STC protocol working with this type of bidirectional situated communication (Mathews et al. 2010a, 2012b). In such a setup, multiple initiator robots can establish STC links in parallel—the potential recipient robots can use the inherent properties of situated communication to distinguish between signals from the different initiator robots (e.g., based on distance) and therefore choose which STC process to participate in.

Another potential issue with our protocol is related to the noise inherent to most technologies that transmit situated communication signals (e.g., ultrasound, infrared, or LED light). Our STC protocol can cope with wrongly detected signals caused by noise. During the elimination process, if a wrong signal causes the target robot to eliminate itself at any point, the initiator robot stops signalling until all potential recipient robots quit the process, then simply starts the elimination process again. During the growth process, if a signal is wrongly detected by any of the robots, the process simply requires a few additional iterations to correct the error. Both these corrective measures have time costs. We therefore also used filtering techniques to reduce the number of wrong signals: to filter out wrongly detected signals, we let each signal receiving robot make decisions only after processing at least three subsequently received camera frames. This simple filtering mechanism proved sufficient to completely eliminate transmission errors in all of our experiments.

7 Conclusions and future work

Multirobot systems are starting to consist of large numbers of robots. Additionally, such systems often consider a heterogeneous group of robots that operate in increasingly large environments pursuing different tasks in parallel. Scalable spatial coordination between different types of robots is thus becoming increasingly important. In this paper, we identified STC as an essential form of communication that enables spatial coordination in multirobot systems. We presented a protocol that enables robots in a decentralized multirobot system to establish ad-hoc communication links to particular target robots based on their location in the environment. The protocol can be implemented on standard hardware found on many existing robotic platforms. We formally described the protocol's constituent processes using Markov chain formalisms and showed that the elimination process scales logarithmically with respect to the number of potential recipient robots while the growth process scales logarithmically with respect to the target group size. We presented results of 332 experimental runs carried out using a heterogeneous

⁴ This estimate need not be precise. One can even round up the estimate by an order of magnitude to be on the safe side. The logarithmic nature of the relationship between the number of robots in the system and the number of iterations in the STC protocol (see Fig. 7) ensures that the cost in terms of additional iterations will be very low.

multirobot system composed of autonomous aerial and ground-based robots—a type of heterogeneous multirobot system that has recently gained much attention (Dorigo et al. 2013) and may potentially benefit the most from our approach. We presented two application scenarios to demonstrate the usefulness of STC when solving real world tasks. Our approach does not require any dedicated hardware and also functions in GPS-denied environments. Thus, it offers high portability to a variety of existing and future robotic systems that include any kind of situated signal emitting technology.

In future research, we are incorporating the approach presented in this work in studies that require coordination between teams of aerial and ground-based robots that operate in parallel. We are also interested in developing platform specific models that consider characteristics inherent to the used platform, for instance probability of miscommunication (due to wrongly detected signals) or probability of malfunctioning robots, to enhance the robustness of model predictions when operating on different real robotic hardware. Such models will allow us to reliably apply STC to a multitude of platforms to study a series of complex tasks that require spatial coordination ranging from multirobot construction to dynamic task allocation under the supervision of aerial robots.

Acknowledgments This work was partially supported by the European Research Council through the ERC Advanced Grant “E-SWARM: Engineering Swarm Intelligence Systems” (Contract 246939). Nithin Mathews acknowledges support from Wallonia-Brussels-International (WBI) through a Scholarship for Excellence Grant. Anders Lyhne Christensen acknowledges support from Fundação para a Ciência e a Tecnologia (FCT) through the Grants PEst-OE/EEI/LA0008/2013 and EXPL/EEI-AUT/0329/2013. Rehan O’Grady, Arne Brutschi, and Marco Dorigo acknowledge support from the Fund for Scientific Research F.R.S.—FNRS of Belgium’s French Community, of which they are a postdoctoral researcher, a research fellow, and a research director respectively.

References

- Barnhard, D. H., McClain, J. T., Wimpey, B. J., & Potter, W. D. (2004). Odin and Hodur: Using bluetooth communication for coordinated robotic search. In *Proceedings of the 2004 International Conference on Artificial Intelligence (IC-AI 2004)* (pp. 365–371). Athens, GA: CSREA Press.
- Bolla, K., Istenes, Z., Kovacs, T., & Fazekas, G. (2011). A fast image processing based robot identification method for Surveyor SRV-1 robots. In *Proceedings of 2011 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM 2011)* (pp. 1003–1009). New York, NY: ASME.
- Bonani, M., Longchamp, V., Magnenat, S., Rétornaz, P., Burnier, D., Roulet, G., et al. (2010). The marXbot, a miniature mobile robot opening new perspectives for the collective-robotic research. In *Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2010)* (pp. 4187–4193). Los Alamitos, CA: IEEE Computer Society Press.
- Bristeau, P. J., Callou, F., Vissière, D., & Petit, N. (2011). The navigation and control technology inside the AR.Drone micro UAV. In *Proceedings of the 18th IFAC World Congress, IFAC-PapersOnLine* (pp. 1477–1484). Centerville, OH.
- Campo, A., Nouyan, S., Birattari, M., Groß, R., & Dorigo, M. (2006). Negotiation of goal direction for cooperative transport. In *Ant Colony Optimization and Swarm Intelligence—Proceedings of ANTS 2006 – Fifth International Workshop, vol 4150* (pp. 191–202). Berlin, Germany: Springer.
- Chaimowicz, L., Sugar, T., Kumar, V., & Campos, M. F. M. (2001). An architecture for tightly coupled multi-robot cooperation. In *Proceedings of the 2001 IEEE International Conference on Robotics and Automation (ICRA 2001)* vol 3, (pp. 2992–2997). Piscataway, NJ: IEEE Press.
- Christensen, A. L., O’Grady, R., & Dorigo, M. (2007). Morphology control in a self-assembling multi-robot system. *IEEE Robotics & Automation Magazine*, 14(4), 18–25.
- Christensen, A. L., O’Grady, R., & Dorigo, M. (2008). SWARMORPH-script: A language for arbitrary morphology generation in self-assembling robots. *Swarm Intelligence*, 2(2–4), 143–165.
- Dorigo, M., Floreano, D., Gambardella, L. M., Mondada, F., Nolfi, S., Baaboura, T., et al. (2013). Swarmanoid: A novel concept for the study of heterogeneous robotic swarms. *IEEE Robotics & Automation Magazine*, 20(4), 60–71.
- Dorigo, M., Birattari, M., & Brambilla, M. (2014). Swarm robotics. *Scholarpedia*, 9(1), 1463.
- Franchi, A., Oriolo, G., & Stegagno, P. (2010). Probabilistic mutual localization in multi-agent systems from anonymous position measures. In *49th IEEE Conference on Decision and Control, Atlanta, GA, USA* (pp. 6534–6540).
- Gerkey, B. P., & Mataric, M. J. (2002). Sold!: Auction methods for multi-robot coordination. *IEEE Transactions on Robotics and Automation*, 18(5), 758–768.
- Goshtasby, A. (2012). Similarity and dissimilarity measures. In *Image Registration, Advances in Computer Vision and Pattern Recognition* (pp. 7–66). London, UK: Springer.
- Grabowski, R., & Khosla, P. (2001). Localization techniques for a team of small robots. In *Proceedings of the 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2001)* vol 2, (pp. 1067–1072). Piscataway, NJ: IEEE Press.
- Groß, R., & Dorigo, M. (2008). Self-assembly at the macroscopic scale. *Proceedings of the IEEE*, 96(9), 1490–1508.
- Gutiérrez, A., Campo, A., Dorigo, M., Amor, D., Magdalena, L., & Monasterio-Huelin, F. (2008). An open localization and local communication embodied sensor. *Sensors*, 8(11), 7545–7563.
- Kemeny, J. G., & Snell, J. L. (1976). *Finite markov chains*. New York, NY: Springer-Verlag.
- Krajník, T., Vonásek, V., Fišer, D., & Faigl, J. (2011). AR-Drone as a platform for robotic research and education. In *Research and Education in Robotics—EUROBOT 2011* vol 161, (pp. 172–186). Berlin, Germany: Springer.
- Kumar, V., & Michael, N. (2012). Opportunities and challenges with autonomous micro aerial vehicles. *The International Journal of Robotics Research*, 31(11), 1279–1291.
- Kushleyev, A., Mellinger, D., Powers, C., & Kumar, V. (2013). Towards a swarm of agile micro quadrotors. *Autonomous Robots*, 35(4), 287–300.
- Mathews, N., Christensen, A. L., Ferrante, E., O’Grady, R., & Dorigo, M. (2010). Establishing spatially targeted communication in a heterogeneous robot swarm. In *Proceedings of 9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2010)* (pp. 939–946). Richland, SC: IFAAMAS.
- Mathews, N., Christensen, A. L., O’Grady, R., & Dorigo, M. (2010). Cooperation in a heterogeneous robot swarm through spatially targeted communication. In *Proceedings of the 7th International Con-*

- ference on Swarm Intelligence (ANTS 2010) (pp. 400–407). Berlin, Germany: Springer.
- Mathews, N., Christensen, A. L., O'Grady, R., Rétornaz, P., Bonani, M., Mondada, F., et al. (2011). Enhanced directional self-assembly based on active recruitment and guidance. In *Proceedings of the 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2011)* (pp. 4762–4769). Los Alamitos, CA: IEEE Computer Society Press.
- Mathews, N., Christensen, A. L., O'Grady, R., & Dorigo, M. (2012). Spatially targeted communication and self-assembly. In *Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2012)* (pp. 2678–2679). Los Alamitos, CA: IEEE Computer Society Press.
- Mathews, N., Stranieri, A., Scheidler, A., & Dorigo, M. (2012). Supervised morphogenesis – morphology control of ground-based self-assembling robots by aerial robots. In *Proceedings of 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)* (pp. 97–104). Richland, SC: IFAAMAS.
- Mathews, N., Valentini, G., Christensen, A. L., O'Grady, R., Brutschy, A., & Dorigo, M. (2013). Spatially targeted communication in multirobot systems. Online supplementary material: <http://iridia.ulb.ac.be/supp/IridiaSupp2013-005>.
- McClain, J. T., Wimpey, B. J., Barnhard, D. H., & Potter, W. D. (2004). Distributed robotic target acquisition using Bluetooth communication. In *Proceedings of the ACM Southeast Regional Conference 2004* (pp. 291–296). New York, NY: ACM.
- Nouyan, S., Campo, A., & Dorigo, M. (2008). Path formation in a robot swarm: Self-organized strategies to find your way home. *Swarm Intelligence*, 2(1), 1–23.
- Nouyan, S., Groß, R., Bonani, M., Mondada, F., & Dorigo, M. (2009). Teamwork in self-organized robot colonies. *IEEE Transactions on Evolutionary Computation*, 13(4), 695–711.
- Parker, L. E., Kannan, B., Tang, F., & Bailey, M. (2004). Tightly-coupled navigation assistance in heterogeneous multi-robot teams. In *Proceedings of 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2004)* (pp. 1016–1022). Piscataway, NJ: IEEE Press.
- Pincioli, C., Trianni, V., O'Grady, R., Pini, G., Brutschy, A., Brambilla, M., et al. (2012). ARGoS: A modular, parallel, multi-engine simulator for multi-robot systems. *Swarm Intelligence*, 6(4), 271–295.
- Pugh, J., Raemy, X., Favre, C., Falconi, R., & Martinoli, A. (2009). A fast onboard relative positioning module for multirobot systems. *IEEE/ASME Transactions on Mechatronics*, 14(2), 151–162.
- Rivard, F., Bisson, J., Michaud, F., & Létourneau, D. (2008). Ultrasonic relative positioning for multi-robot systems. In *Proceedings of the 2008 IEEE International Conference on Robotics and Automation (ICRA 2008)* (pp. 323–328). Piscataway, NJ: IEEE Press.
- Roberts, J. F., Stirling, T., Zufferey, J. C., & Floreano, D. (2012). 3-D relative positioning sensor for indoor flying robots. *Autonomous Robots*, 33(1–2), 5–20.
- Stegagno, P., Cognetti, M., Rosa, L., Peliti, P., & Oriolo, G. (2013). Relative localization and identification in a heterogeneous multi-robot system. In *Proceedings of the 2013 IEEE International Conference on Robotics and Automation (ICRA 2013)* (pp. 1857–1864). Piscataway, NJ: IEEE Press.
- Stentz, A. T., Kelly, A., Herman, H., Rander, P., Amidi, O., & Mandelbaum, R. (2002). Integrated air/ground vehicle system for semi-autonomous off-road navigation. In *Proceedings of the AUVSI Unmanned Systems Symposium*.
- Støy, K. (2001). Using situated communication in distributed autonomous mobile robotics. In *Proceedings of the Seventh Scandinavian Conference on Artificial Intelligence (SCAI 2001)* (pp. 44–52). Amsterdam, The Netherlands: IOS Press.
- Vaughan, R. T., Sukhatme, G. S., Mesa-Martinez, F. J., & Montgomery, J. F. (2000). Fly spy: Lightweight localization and target tracking for cooperating air and ground robots. In *Distributed Autonomous Robotic Systems 4* (pp. 315–324).
- Zhang, Y., & Parker, L. (2013). IQ-ASyMTR: Forming executable coalitions for tightly coupled multirobot tasks. *IEEE Transactions on Robotics*, 29(2), 400–416.



Nithin Mathews studied computer science and holds a Dipl.-Ing degree from the University of Applied Sciences and Arts Northwestern Switzerland, Switzerland (February 2006) and a Master degree from the University of Freiburg, Germany (July 2008). He was also awarded a Master of Advanced Studies degree in Applied Sciences (May 2010) by Université Libre de Bruxelles, Belgium. He currently is a Ph.D. student at IRIDIA-CoDE, the artificial intelligence

research laboratory of the Université Libre de Bruxelles in Belgium. His research interests lie in the areas of swarm robotics, self-assembling robots, and air-ground robot teams.



Gabriele Valentini is a Ph.D. student at IRIDIA-CoDE, Université Libre de Bruxelles, Belgium. He obtained Bachelor and Master degrees in Engineering of Computing Systems at Politecnico di Milano, Italy, respectively in 2008 and 2011. In 2013, he received a Master of Advanced Studies in Applied Sciences from the Faculty of Engineering of the Université Libre de Bruxelles, Belgium. His research interests focus on swarm intelligence and swarm robotics with particular emphasis on mathematical modeling of self-organized collective decision-making systems.



Anders Lyhne Christensen is an assistant professor at ISCTE-IUL and a researcher at Instituto de Telecomunicações, Portugal. In 2002, he obtained a Master's degree in computer science and bioinformatics from Aalborg University, Denmark. Later that year, he received a Marie Curie Fellowship hosted by the Dependable Systems Group at the Universidade de Coimbra and Critical Software, Portugal. He completed his Diplôme d'Études Approfondies at the Université Libre de Bruxelles in 2005. In 2008, he obtained the doctoral degree from IRIDIA-CoDE, Université Libre de Bruxelles, Belgium. Dr. Christensen has spent several years in the private sector and worked on software projects ranging from 3D acoustics to high performance computing. His research interests include swarm robotics, autonomous self-assembly, and evolutionary computation.



Rehan O'Grady is a post-doctoral researcher at the IRIDIA-CoDE laboratory of the Université Libre de Bruxelles, Belgium. He graduated in 1999 from Edinburgh University with First Class Honours in Mathematics and Computer Science. He was awarded the Kevin Clark memorial prize, awarded each year to the top Mathematics and Computer Science graduate. He subsequently worked in the software industry for several years. During his time at Micromuse

PLC he developed a network monitoring system which was subsequently patented. He received his doctorate from the Faculty of Engineering of the Université Libre de Bruxelles, Belgium in 2010. His current research interests include self-assembling systems and human-swarm-interaction.



Arne Brutschy received a Diploma in computer science from the University of Leipzig, Germany. In December 2014 he received a Ph.D. in applied sciences at IRIDIA, Université Libre de Bruxelles, Belgium. Arne Brutschy received a scholarship from the Fund for Scientific Research F.R.S.-FNRS of Belgium's French Community for the duration of his Ph.D. studies. His research interests are in artificial intelligence and swarm robotics, with a focus on self-

organized task allocation and task partitioning.



Marco Dorigo received the Laurea, Master of Technology, degree in industrial technologies engineering in 1986, and the Ph.D. degree in electronic engineering in 1992 from the Politecnico di Milano, Milan, Italy, and the title of Agrégé de l'Enseignement Supérieur, from ULB, in 1995. From 1992 to 1993, he was a Research Fellow at the International Computer Science Institute, Berkeley, CA. In 1993, he was a NATO-CNR Fellow, and from 1994 to

1996, a Marie Curie Fellow. Since 1996, he has been a tenured Researcher of the FNRS, the Belgian National Funds for Scientific Research, and co-director of IRIDIA, the artificial intelligence laboratory of the ULB. He is the inventor of the ant colony optimization metaheuristic. His current research interests include swarm intelligence, swarm robotics, and metaheuristics for discrete optimization. He is the Editor-in-Chief of Swarm Intelligence, and an Associate Editor or member of the Editorial Boards of many journals on computational intelligence and adaptive systems. Dr. Dorigo is a Fellow of IEEE, AAAI, and ECCAI. He was awarded the Italian Prize for Artificial Intelligence in 1996, the Marie Curie Excellence Award in 2003, the Dr. A. De Leeuw-Damry-Bourlart award in applied sciences in 2005, the Cajastur International Prize for Soft Computing in 2007, an ERC Advanced Grant in 2010, and the IEEE Frank Rosenblatt Award in 2015.