Architecture, Abstractions, and Algorithms for Controlling Large Teams of Robots: Experimental Testbed and Results

Nathan Michael, Jonathan Fink, Savvas Loizou, and Vijay Kumar

University of Pennsylvania
Philadelphia, Pennsylvania 19104-6228
{nmichael,jonfink,sloizou,kumar}@grasp.upenn.edu

Summary. We describe the architecture, algorithms, and experimental testbed for the deployment of large numbers of cooperating robots, and applications to tasks like manipulation and transportation. The coordination between robots is completely decentralized to enable scaling up to large numbers of robots. There is no labeling or identification of robots and all robots (and their software) are identical allowing robustness to failures, ease of programming, and modularity enabling the addition or deletion of robots from the team. Our approach requires minimal communication and sensing and the proposed controllers are based only on local information. Moreover, our architecture facilitates asymmetric communication from one or more supervisory agents that can broadcast information to all robots and close the loop by acquiring abstract, high level information related to the supervised robots. We discuss the hardware and software implementation, the architecture, and present recent experimental results.

1 Introduction

Recent advances in communication, sensing, and networking, as well as increases in the ratio between performance and price of computers are reflected in the increased presence of embedded computers and sensors in homes and factories. Further, wireless ad-hoc networks or plug-and-play wired networks are becoming commonplace. Applications for networked robots where one can exploit such technology and infrastructures include environmental monitoring [1], surveillance and reconnaissance for security and defense [2–4], and support for first responders in a search and rescue operation [5, 6].

The emerging need for large-scale networked multi-robot systems has prompted a growing response by the research community for architectures and methodologies for balancing the increased algorithmic requirements of controlling distributed systems with sufficient centralization to permit human interfacing. In this paper, we describe our approach to the architectural design of large scale multi-robot algorithms. We motivated this design with a discussion of the pragmatic considerations required when applying multi-robot algorithms to larger systems. We discuss a control framework that abstracts the system complexity to a manageable level by promoting algorithmic invariance to the scaling size of the system through an *Asymmetric Broadcast Control* (ABC). To elucidate these concepts we discuss the development of an experimental

M. Kaneko and Y. Nakamura (Eds.): Robotics Research, STAR 66, pp. 409–419.
springerlink.com

robotics infrastructure with a focused discussion on the application of such methodologies in experimentation. Recent results provide validation of the appropriateness of the control framework and the use of behaviors as a means to reduce the algorithmic complexity of large multi-robot systems.

2 Motivating Design Principles

We are interested in the deployment of potentially large numbers of cooperating robots with applications to tasks such as persistent surveillance, object manipulation, and transportation. When applying distributed control algorithms to robotic system attributes such as *decentralization*, *anonymity*, and *uniform modularity* aid in implementation. Decentralization means that the algorithm does not require access to the full global state. Anonymity suggests that each robot does not require an identifying attribute of other robots in order to continue the computation of the algorithm. Uniform modularity in algorithmic application extends the idea of anonymity to further promote the notion that each robot executes an instance of the same uniform algorithmic module. Modularity permits a higher level of interoperability between differing control algorithms and often reduces the complexity of the control algorithm simplifying the implementation. These attributes often improve the efficiency and interoperability of algorithms by permitting computations to execute in parallel across the robot network. Additionally, a robot is rendered similar to the other robots in the system. The algorithm is made robust by ensuring that no robot plays a role of vital importance and each robot easily replaceable in the case of failure.

While wishing to maintain these distributed properties we acknowledge the need for the centralization of certain aspects of the control law to permit higher level interaction with the system by a supervisory agent such as a human operator. It is often impractical for large numbers of robots to share information and to have every robot access global state information. While the communication network facilitates the design of distributed estimators, these estimators may not be robust to intermittent failures of the network. As such failures can reduce the robustness of the control algorithm, we are also interested in an approach that alleviates the burden of communication and sensing within the technological constraints of modern networking protocol with controllers that decouple the performance of multi-robot controllers from that of decentralized estimators.

3 Control Architecture

A number of multi-robot control architectures have been developed in the last decade [7–11], many of which were inspired by behavior-based control paradigms [12]. The need to have decentralized control to enable scalability to large numbers is emphasized [9, 13]. Many architectures rely on hierarchy to manage the complexity of the task and the control software [14]. However, to task large groups of robots, we include a component of centralization to allow a human (or an appropriate supervisory agent) to interact with and task the team.

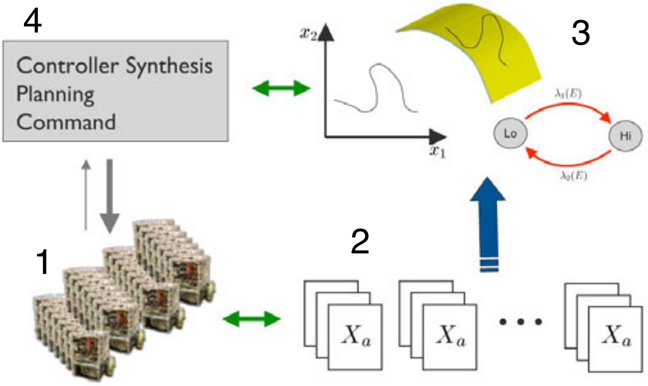


Fig. 1. Asymmetric Broadcast Control (ABC) Architecture. The mathematical state description (**2**) of a team of robots (**1**) reduces the complexity of representing the control of the system in an abstract space (**3**; here depicted as a vector, geometric manifold, or stochastic space). The control synthesis, planning, and command (**4**) need only consider the abstract space rather than the state of the full system of robots.

3.1 The Asymmetric Broadcast Control Architecture

When considering teams of robots nearing the scale characteristic of sensor networks, it is necessary to consider approaches to program, command, control, and monitor the robot teams. However, it is also desirable in such large teams to not require knowledge of the specifics of each robot and the number of robots in the team. We advocate a *broadcast paradigm* in which all robots have uniform software and receive common instructions but have the software-enabled intelligence to adopt roles and perform the required tasks.

Accordingly we propose the *Asymmetric Broadcast Control* architecture shown in Fig. 1. Since the configuration space and the state space of the multi-robot system grows with the number of robots, and the complexity of most centralized algorithms for coordination are at least quadratic or cubic in the number of nodes, it is necessary to allow the supervisory agent to interact with the system in a reduced dimensional space. We argue that it is mathematically necessary to define an *abstraction*, a function that maps the large state space into an abstract state, whose dimension is small and independent of the number of robots and has a physical interpretation that is easy to control and command. Of course there may be multiple abstractions based on the task. Formally, define the state space of the n robot system as n copies of X_{a_i}, the state space of the i^{th} robot:

$$\mathbf{X} = X_{a_1} \times X_{a_2} \times \ldots \times X_{a_n},$$

and the abstract space $\mathbf{A}$ whose dimension is smaller and independent of the dimension of $\mathbf{X}$.

$$\Phi : \mathbf{X} \rightarrow \mathbf{A}, \quad \Phi(\mathbf{x}) = \mathbf{a}, \tag{1}$$

where Φ is a mapping of the higher-dimensional state $\mathbf{x} \in \mathbf{X}$ to the lower-dimensional abstract state $\mathbf{a} \in \mathbf{A}$. In Fig. 1, we show three different abstractions. $\mathbf{A}$ can be an m-dimensional vector space and the evolution of the multi-robot system can be viewed as the evolution of the abstract state **a** in this vector space. **A** may be endowed with the geometric properties of a manifold as in [15–17]. When scaling to very large numbers, it may be productive to view the group as a stochastic process and the appropriate abstraction may be a continuous-time, q-state, Markov chain [18].

Such an abstraction permits the designer or programmer to consider the motion planning and controller synthesis problem on this lower-dimensional space **A**. This approach necessitates a supervisory controller interacting with the group at this higher level in the abstract space **A**. Tasks, plans, and trajectories are specified at the abstraction level permitting supervisory interaction in a reduced-complexity space. Further, distributed networking protocols such as broadcasting is readily incorporated since the higher level abstraction provides common information allowing individual robots to interpret these commands based on their individual states and the state of their local environment.

The requirement of a supervisory agent may arguably detract from the decentralization of the control algorithms. However, the *ABC* architecture permits a level of centralization without requiring the individual robots to lose the desirable properties of distributed control algorithms. Additionally, this framework accommodates the (often pragmatic) need to provide global information to the team of robots such as information regarding the external world (e.g., maps of the environment) and information about the abstract state of the group. However, the broadcast protocol ensures that the information is independent of the number of robots in the team and depends only on the complexity of the tasks that need to be performed. At the local inter-agent level, any interactions and communications between the robots are transparent to the supervisory agent.

There may also be a need to provide to the supervisory agent some information about the state of the system. This can be achieved at low data rates by individual agents. Alternatively it can also be accomplished by equipping supervisory agents with appropriate sensors allowing them to measure the abstract state of the system at a coarse level of granularity without worrying about specifics regarding individual states. Such a framework ensures that the volume of data transfer from the supervisory agent to the team of robots greatly exceeds the data transfer from the individual robots to the supervisory agent in an *asymmetric* manner. This design permits the addition or deletion of robots with little effect on the total data transfer.

We acknowledge that our emphasis in the discussion of the architectural issues is on the behavioral architecture and control, rather than estimation, localization, and mapping. Clearly, there is an entire set of issues regarding how individual robots exchange information with other robots [11], how information from different robots and sensors are seamlessly integrated [3], and how this information can be used for mapping and exploration [4]. However, in this architecture individual robots can talk to each other in addition to talking to a supervisory agent. Additionally, the integration of information and the sharing of this information may be facilitated by a broadcast agent.

3.2 Design and Composition of Local Behaviors

While the motivating principles discussed in Sect. 2 emphasize the abstraction of the control algorithms to a space that is invariant to system size, local interactions must also be considered for many practical reasons. Following the same motivation to maintain

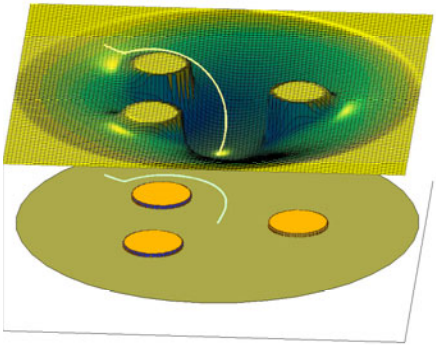


Fig. 2. Navigation function with three obstacles and the resulting gradient following path

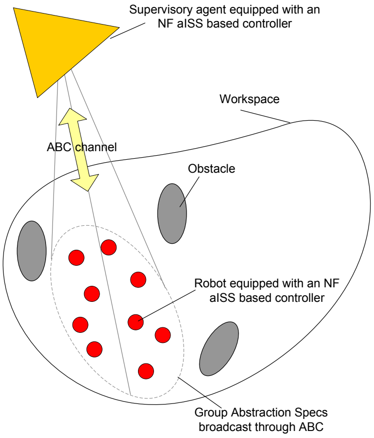


Fig. 3. Exploiting the *ABC* concept in conjunction with navigation function (NF) based aISS controllers.

decentralized, anonymous, and uniformly modular control algorithms for each robot we propose the use of local controllers for each of the individual robots that satisfy the fundamental properties of safety and stability. These properties will guarantee that the group will be able to carry out the high level orders of the supervisor (stability), while the supervisor will not have to worry about collisions between the robots (safety).

One class of controllers that provide the fundamental properties of safety and stability are the navigation function based controllers which inherit their safety and stability properties from the underlying navigation functions [19]. Navigation functions are real valued maps realized through cost functions whose negated gradient fields are attractive towards the goal configuration and repulsive with respect to obstacles or other robots as depicted in Fig. 2.

Although navigation functions provide a way to create navigation behaviors for individuals, this is not adequate when facing complex tasks that require the joint "intelligence" of several systems (including human users) or when the system needs to concurrently satisfy several initiatives. This is true not only for the local robot controllers, but is also often an indispensable capability for the supervisory agent. In [20] a compositional framework is presented where the controllers inherit the safety and stability properties from the underlying navigation functions while the interconnections are handled under the almost Input-to-State Stability (aISS) framework. The resulting system maintains the fundamental properties of safety and (global asymptotic) stability of the initial navigation function system. The almost global characterization is another trait that is inherited from the navigation functions. The case of mixed initiative control, where humans are interacting with the planners, can be seen as a special case of this compositionality framework [21].

A compositional control strategy that promotes the use of the *ABC* architecture at a high level with the local interaction guarantees of the navigation function based aISS

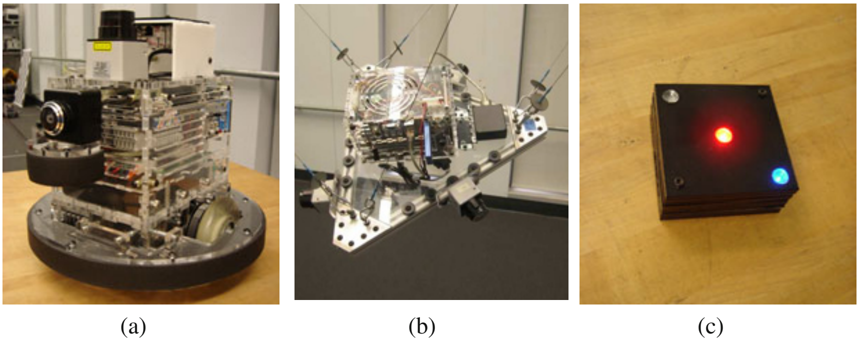

(a) (b) (c)

Fig. 4. The 20 × 13.5 × 22.2 cm^3 SCARAB platform is shown in Fig. 4(a). The KHEPRI robot, Fig. 4(b), is controlled by six cables and has a full suite of sensing and computational abilities making it well suited for emulation of an UAV in indoor environments. Fig. 4(c) depicts an LED target used for localization.

controllers is shown in Fig. 3. At a global level, the supervisory agent assesses abstract properties describing the workspace and environment of the team of robots. This abstract description defines the local aISS control laws. Since these controllers are navigation function based aISS controllers, mixed initiative control can be implemented, enabling human users to seamlessly interact with the system at any level of the hierarchy while preserving the properties of the distributed *ABC* architecture and the stability and safety guarantees of the underlying control laws.

4 Testbed Design for Multi-robot Experimentation

The following discussion briefly describes the design of an experimental testbed with an emphasis on the influence of the design principles discussed in Sect. 2 on the system architecture. We detail the realization of distributed control algorithms in modular software as well as the physical infrastructure developed to implement the algorithms in light of these principles.

4.1 Software

We adhere to the well-known object-oriented programming paradigm for writing hardware and algorithmic code modules. We use the *Player* server [22] of the open-source *Player/Stage/Gazebo* (PSG) project to ensure interfacing between the code modules that run on each of the robots. The development of networked code modules permits the evaluation of algorithms in a distributed manner.

4.2 Hardware and Infrastructure

We have designed two robots for use in distributed multi-robot experiments. The SCARAB is a small (20 × 13.5 × 22.2 cm^3) differential drive ground platform, shown in Fig. 4(a), equipped with an embedded computer, sensors, and motors. The robot was designed to be easily manufactured, economic, and repairable due to the modularity in the mechanical design. The on-board computational and sensor capabilities of the SCARAB permit each robot to perform independent calculations based on their local environment. Additionally, a local wireless network permits communications between each of the robots or the global system. With these capabilities, the SCARAB serves as an ideal platform for testing distributed control algorithms on mobile ground robots. Further, the small form-factor of the robot permits the size of the group of robots to scale based on the experiment design.

The second robot is the cable-driven parallel mechanism shown in Fig. 4(b) called KHEPRI. In the *ABC* framework, either KHEPRI or a human act as the supervisory agent. KHEPRI is equipped with an embedded computer, camera, lasers, and an inertial measurement unit, with cables driven by motors equipped with encoders. Such a sensor suite permits KHEPRI to be used for both control and observation. Further, since KHEPRI is enabled to communicate on the same wireless network as the SCARABs, interaction between the two types of robots is possible. Given the discussion of Sects. 3.1 – 3.2, it is apparent that KHEPRI and the SCARABs facilitate the testing of distributed control algorithms for teams of robots with a global observer.

The LED localization target of Fig. 4(c) is used to localize the pose of the team of ground robots. By equipping each of the SCARABs (and other ground objects) with a target, we are able to identify and localize each of the robots via overhead cameras. Each target flashes with a unique pattern that permits real-time tracking based on the LED position, and identification based on the unique pattern. This technology permits the verification of the *ABC* framework without requiring the additional complexity of pose estimation.

A dedicated wireless network is used to accommodate the networking requirements of the distributed robotics experiments. The SCARABs, KHEPRI, and external computers are interfaced on this network, which permits inter-robot communication and external human interaction during experimentation.

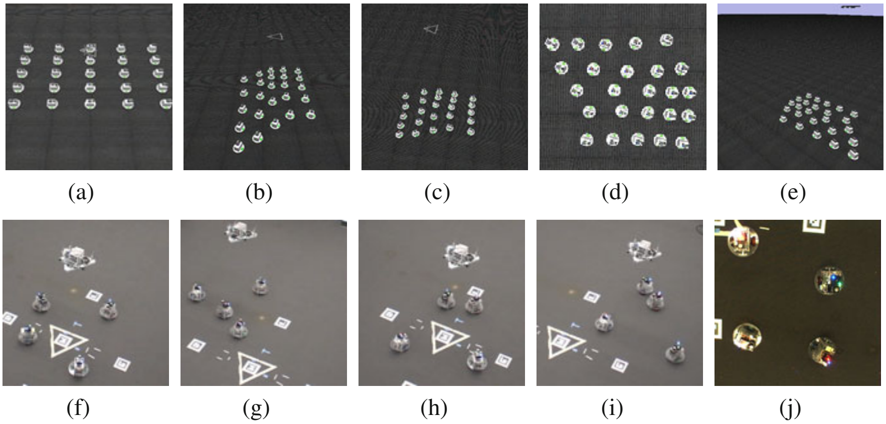
4.3 Simulation and Integration

As mentioned in Section 4.1, we use the software developed by the PSG project, which defines interfaces for our distributed system and provides communication between the robots. Additionally, *Player* provides a layer of hardware abstraction that permits algorithms to be tested in simulated three-dimensional *Gazebo* environments. By adhering to well-defined interface specifications, each algorithmic implementation (for example, those discussed in Sects. 5.1 and 5.2) is identical for both simulation and experimentation on hardware.

5 Experimental Verification

5.1 Formation Control

We are interested in controlling a large team of nonholonomic ground robots in a decentralized fashion that is invariant to the number of ground robots. We briefly present experimental results using the KHEPRI aerial robot and a team of SCARAB robots.



(a) (b) (c) (d) (e)

(f) (g) (h) (i) (j)

Fig. 5. Formation Control in Simulation and Experimentation. Figures 5(a)–5(e) depict the control of a team of twenty-five robots (in simulation) by an aerial robot using the control law discussed in Sect. 5.1. The aerial robot is controlling the team of ground robots by defining a predefined trajectory in a lower-dimensional abstract space. Figures 5(f)–5(j) show a similar scenario on a team of ground robots and an aerial cable robot using hardware.

In our previous work [17, 23], we developed an abstraction for the team that includes a gross model of the shape of the formation of the team and information about the position and orientation of the team in the plane. Thus in (1), we can define $\mathbf{A} = SE(2) \times \mathcal{S}$, where $SE(2)$ is the special Euclidean group in two dimensions and $\mathcal{S} \subset R^2$ is the shape space consisting of the major and minor axes of a ellipsoidal approximation [15]. We also derived controllers that allow the team of robots to move in formation while avoiding collisions and respecting the abstraction commanded by the aerial platform [23]. Thus KHEPRI, the supervisory agent, is able to control not only the gross position (μ) and orientation (θ) of the formation, but also the shape $(s_1,\ s_2) \in \mathcal{S}$ by simply broadcasting the current abstract state $\mathbf{a} = (\mu_x,\ \mu_y,\ \theta,\ s_1,\ s_2)$, and the desired abstract state to the ground robots.

Thus, commands of the abstract state $\mathbf{a}$ are translated into commands of individual robots. Details on the control laws and derivations are available in [15, 17] and on the practical implementation including collision avoidance in [23]. The formation controller was implemented on a team of four SCARABS and KHEPRI in simulations and experiments. A representative control scenario is shown in Fig. 5.

5.2 Cooperative Manipulation

For cooperative manipulation we are interested in a decentralized algorithm that manipulates an object via caging [24]. The complexity of the control problem is reduced by abstracting the control to a composition of vector fields [16, 25]. Since these vector fields are defined with respect to the object's geometric shape, the resulting control law is independent of the number of agents and only requires the assumption that the number of agents is sufficiently large to surround the object for caging purposes. The

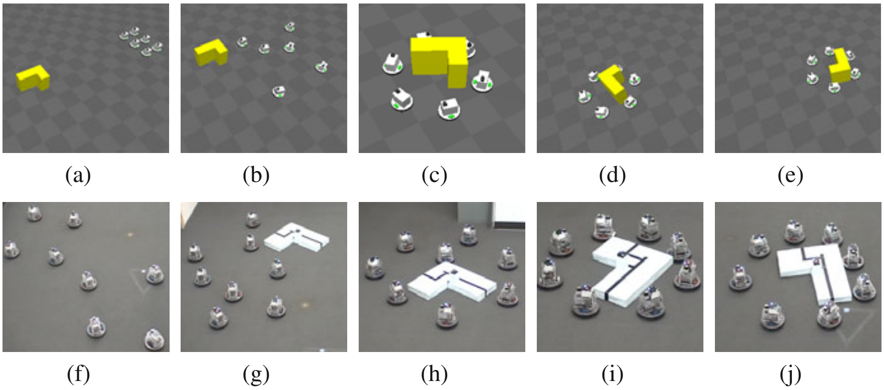


Fig. 6. Figures 6(a)–6(e) present a representative distributed manipulation of an "L-shaped" object simulated in *Gazebo*. The robots approach (Figs. 6(a) – 6(b)), surround (Fig. 6(c)), and manipulate (Figs. 6(d) – 6(e)) an object. Figures 6(f)–6(j) depict a similar scenario as above with the approach (Figs. 6(f) – 6(g)), surround (Fig. 6(h)), and manipulate (Figs. 6(i) – 6(j)) behaviors.

control law is anonymous in that the identification of individual agents is unnecessary and the number of robots can change dynamically.

We refer to this composition of vector fields as the construction of behaviors where individual components can be described by common semantics such as approach, surround, and manipulate. Further, each of these components have been shown to provide guarantees on the convergence and stability of the control law [26].

We have been able to demonstrate through simulation and experimentation that a switching system between these behaviors is robust to both the type of object being manipulated and the number of robots available for manipulation. In fact, hundreds of simulations have been conducted with different object types and different numbers of available agents with success. In experimentation on real hardware we have conducted tens of trials with four to eight robots manipulating an object along both linear and sinusoidal trajectories as depicted in Fig. 6.

6 Conclusion

We presented our architecture for controlling large teams of robots. The architecture emphasizes the use of abstractions to reduce the complexity of control algorithms for distributed systems. Additionally, we demonstrated how this framework enables the inclusion of global and local interactions in the form of supervisory agents and local behaviors, respectively. An experimental testbed was discussed which adheres to the motivating principles underlying the control framework in software and hardware. The flexibility of the system was shown in two experiments which serve as more concrete examples to the meaning of the notion of an abstraction.

We are currently considering the application of the *ABC* framework to a larger class of problems. We are broadening the controllers of Sect. 5.1 to capture more diverse abstract descriptions beyond ellipsoidal approximations. Additionally, we extending upon the work of Sect. 5.2 to take into account robot failures during manipulation by incorporating local sensing into the behavior descriptions. Another area of interest is the estimation of abstract parameters to supplement or replace requirements on the supervisory agent.

Acknowledgment

This research was supported by: NSF grants CCR02-05336, NSF IIS-0413138, and IIS-0427313, and ARO Grants W911NF-04-1-0148 and W911NF-05-1-0219.

References

1. Sukhatme, G.S., Dhariwal, A., Zhang, B., Oberg, C., Stauffer, B., Caron, D.A.: Design and development of a wireless robotic networked aquatic microbial observing system. Environmental Engineering Science 24(2), 205–215 (2007)
2. Rybski, P.E., Stoeter, S.A., Erickson, M.D., Gini, M., Hougen, D.F., Papanikolopoulos, N.: A team of robotic agents for surveillance. In: Proc. of Int. Conf. on Autonomous Agents, Barcelona, Spain, June 2000, pp. 9–16 (2000)
3. Howard, A., Parker, L.E., Sukhatme, G.S.: Experiments with a large heterogeneous mobile robot team: Exploration, mapping, deployment and detection. The Int. Journal of Robotics Research 25(5-6), 431–447 (2006)
4. Fox, D., Ko, J., Konolige, K., Limketkai, B., Schultz, D., Stewart, B.: Distributed multirobot exploration and mapping. Proc. of the IEEE 94(7), 1325–1339 (2006)
5. Corke, P., Peterson, R., Rus, D.: Networked Robots: Flying Robot Navigation using a Sensor Net. Springer Tracts in Advanced Robotics, vol. 15, pp. 234–243. Springer, New York (2005)
6. Kumar, V., Rus, D., Singh, S.: Robot and sensor networks for first responders. IEEE Pervasive Computing 3(4), 24–33 (2004)
7. Balch, T., Parker, L.E. (eds.): Robot Teams: From Diversity to Polymorphism. A K Peters, Ltd., Wellesley (2002)
8. Beard, R.W., Lawton, J., Hadaegh, F.Y.: A coordination architecture for spacecraft formation control. IEEE Transactions on Control Systems Technology 9(6), 777–790 (2001)
9. Parker, L.E.: Alliance: An architecture for fault tolerant multi-robot cooperation. IEEE Transactions on Robotics and Automation 14(2), 220–240 (1998)
10. Mackenzie, D.C., Arkin, R.C., Cameron, J.M.: Multiagent mission specification and execution. Autonomous Robots 4, 29–52 (1997)
11. Makarenko, A., Brooks, A., Williams, S., Durrant-Whyte, H., Grocholsky, B.: A decentralized architecture for active sensor networks. In: Proc. of the IEEE Int. Conf. on Robotics and Automation, New Orleans, LA, vol. 2, pp. 1097–1102 (2004)
12. Arkin, R.C.: Behavior Based Robotics. The MIT Press, Cambridge (1998)
13. Feddema, J.T., Schoenwald, D.: Decentralized control of cooperative robotic vehicles. In: Proc. of SPIE, vol. 4364, pp. 136–146 (September 2001)
14. Simmons, R.G., Singh, S., Hershberger, D., Ramos, J., Smith, T.: First Results in the Coordination of Heterogeneous Robots for Large-Scale Assembly. Lecture Notes in Control and Information Sciences, vol. 271, pp. 323–332. Springer, New York (2000)
15. Belta, C., Kumar, V.: Abstraction and control for groups of robots. IEEE Transactions on Robotics 20(5), 865–875 (2004)

16. Chaimowicz, L., Michael, N., Kumar, V.: Controlling swarms of robots using interpolated implicit functions. In: Proc. of the IEEE Int. Conf. on Robotics and Automation, Barcelona, Spain, April 2005, pp. 2487–2492 (2005)
17. Michael, N., Belta, C., Kumar, V.: Controlling three dimensional swarms of robots. In: Proc. of the IEEE Int. Conf. on Robotics and Automation, Orlando, FL, May 2006, pp. 964–969 (2006)
18. Julius, A., Halasz, A., Kumar, V., Pappas, G.J.: Finite state abstaction of a stochastic model of the lactose regulation system escherichia coli. In: Proc. of the IEEE Conference on Decision and Control, San Diego, CA, December 2006, pp. 19–24 (2006)
19. Koditschek, D.E., Rimon, E.: Robot navigation functions on manifolds with boundary. Advances in Applied Mathematics 11(4), 412–442 (1990)
20. Loizou, S.G., Kumar, V.: Weak input-to-state stability properties for navigation function based controllers. In: Proc. of the IEEE Conference on Decision and Control, San Diego, CA, pp. 6506–6511 (2006)
21. Loizou, S.G., Kumar, V.: Mixed initiative control of autonomous vehicles. In: Proc. of the IEEE Int. Conf. on Robotics and Automation, Rome, Italy, April 2007, pp. 1431–1436 (2007)
22. Gerkey, B.P., Vaughan, R.T., Howard, A.: The player/stage project: Tools for multi-robot and distributed sensor systems. In: Proc. of the Int. Conf. on Advanced Robotics, Coimbra, Portugal, June 2003, pp. 317–323 (2003)
23. Michael, N., Fink, J., Kumar, V.: Controlling a team of ground robots via an aerial robot. In: Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, San Diego, CA, October 2007, pp. 965–970 (2007)
24. Fink, J., Michael, N., Kumar, V.: Composition of vector fields for multi-robot manipulation via caging. In: Robotics: Science and Systems, Atlanta, GA (June 2007)
25. Hsieh, M.A., Kumar, V.: Pattern generation with multiple robots. In: Proc. of the IEEE Int. Conf. on Robotics and Automation, Orlando, FL, May 2006, pp. 2442–2447 (2006)
26. Hsieh, M.A., Loizou, S.G., Kumar, V.: Stabilization of multiple robots on stable orbits via local sensing. In: Proc. of the IEEE Int. Conf. on Robotics and Automation, Rome, Italy, April 2007, pp. 2312–2317 (2007)