Multimedia Contents

53. Multiple Mobile Robot Systems

Lynne E. Parker, Daniela Rus, Gaurav S. Sukhatme

Within the context of multiple *mobile, and networked* robot systems, this chapter explores the current state of the art. After a brief introduction, we first examine architectures for multirobot cooperation, exploring the alternative approaches that have been developed. Next, we explore communications issues and their impact on multirobot teams in Sect. 53.3, followed by a discussion of networked mobile robots in Sect. 53.4. Following this we discuss swarm robot systems in Sect. 53.5 and modular robot systems in Sect. 53.6. While swarm and modular systems typically assume large numbers of homogeneous robots, other types of multirobot systems include heterogeneous robots. We therefore next discuss heterogeneity in cooperative robot teams in Sect. 53.7. Once robot teams allow for individual heterogeneity, issues of task allocation become important; Sect. 53.8 therefore discusses common approaches to task allocation. Section 53.9 discusses the challenges of multirobot learning, and some representative approaches. We outline some of the typical application domains which serve as test beds for multirobot systems research in Sect. 53.10. Finally, we conclude in Sect. 53.11 with some summary remarks and suggestions for further reading.

Researchers generally agree that multirobot systems have several advantages over single-robot systems [53.1, 2]. The most common motivations for developing multirobot system solutions are that:

1. The task complexity is too high for a single robot to accomplish.
2. The task is inherently distributed.
3. Building several resource-bounded robots is much easier than having a single powerful robot.
4. Multiple robots can solve problems faster using parallelism.
5. The introduction of multiple robots increases robustness through redundancy.

The issues that must be addressed in developing multirobot solutions are dependent upon the task requirements and the sensory and effector capabilities of the available robots.

The types of robots considered in the study of multiple *mobile* robot systems are those robots that move around in the environment, such as ground vehicles, aerial vehicles, or underwater vehicles. This chapter focuses specifically on the interaction of multiple *mobile* robots, as distinguished from other types of multirobot interaction. For example, a special case of multiple mobile robot systems are the *reconfigurable* or *modular* robots that interconnect with each other for the purposes of navigation or manipulation. Algorithmic aspects of these systems are covered in Sect. 53.5, while hardware aspects are covered in Chap. 22. *Networked* robotics is very closely related to multiple mobile robot systems; the focus in networked robotics is on systems of robots, sensors, embedded computers, and human users that are all connected by networked communication. Another variant of multirobot cooperation is multiple manipulator arm cooperation. Chapter 39 describes these systems in detail.

53.1 History

Since the earliest work on multiple mobile robot systems in the 1980s, the field has grown significantly, and covers a large body of research. At the most general level, approaches to multiple mobile robot systems fall into one of two broad categories: *collective swarm* systems and *intentionally cooperative* systems. *Collective swarm* systems are those in which robots execute their own tasks with only minimal need for knowledge about other robot team members. These systems are typified by the assumption of a large number of homogeneous mobile robots, in which robots make use of local control laws to generate globally coherent team behaviors, with little explicit communication among robots. On the other hand, robots in *intentionally cooperative* systems have knowledge of the presence of other robots in the environment and act together based on the state, actions, or capabilities of their teammates in order to accomplish the same goal. Intentionally cooperative systems vary in the extent to which robots take into account the actions or state of other robots, and can lead to either strongly or weakly cooperative solutions [53.3]. *Strongly cooperative* solutions require robots to act in concert to achieve the goal, executing tasks that are not trivially serializable. Typically, these approaches require some type of communication and synchronization among the robots. *Weakly cooperative* solutions allow robots to have periods of operational independence, subsequent to coordinating their selection of tasks or roles. Intentionally cooperative multirobot systems can deal with heterogeneity in the robot team members, in which team members vary in their sensor and effector capabilities. In these teams, the coordination of robots can be very different from collective swarm approaches, since robots are no longer interchangeable.

Most of the work specific to multiple mobile robot cooperation can be categorized into a set of key topics of study. These topics, which are the foci of this chapter, include *architectures*, *communication*, *swarm robots*, *heterogeneity*, *task allocation*, and *learning*. *Architectures* and *communication* in multirobot systems are relevant for all types of multirobot systems, as these approaches specify how the robot team members are organized and interact. *Swarm robots* is a particular type of multirobot system, typified by large numbers of homogeneous robots that interact implicitly with each other. Such systems are often contrasted with *heterogeneous* robots, in which team members may vary significantly in their capabilities. When robots vary in capabilities, challenges arise in determining which robots should perform which tasks – a challenge commonly referred to as *task allocation*. Finally *learning* in multirobot teams is of particular interest in designing teams that are adaptive over time and can learn new behaviors. Illustrating the advances in each of these areas often takes place in a set of representative application domains; these *applications* are the final major topic of discussion in this chapter.

53.2 Architectures for Multirobot Systems

The design of the overall control architecture for the multirobot team has a significant impact on the robustness and scalability of the system. Robot architectures for multirobot teams are composed of the same fundamental components as in single-robot systems, as described in Chap. 12. However, they also must address the interaction of robots and how the group behavior will be generated from the control architectures of the individual robots in the team. Several different philosophies for multirobot team architectures are possible; the most common are *centralized*, *hierarchical*, *decentralized*, and *hybrid*.

Centralized architectures that coordinate the entire team from a single point of control are theoretically possible [53.4], although often practically unrealistic due to their vulnerability to a single point of failure, and due to the difficulty of communicating the entire system state back to the central location at a frequency suitable for real-time control. Situations in which these approaches are relevant are cases in which the centralized controller has a clear vantage point from which to observe the robots, and can easily broadcast group messages for all robots to obey [53.5].

Hierarchical architectures are realistic for some applications. In this control approach, each robot oversees the actions of a relatively small group of other robots, each of which in turn oversees yet another group of robots, and so forth, down to the lowest robot, which simply executes its part of the task. This architecture scales much better than centralized approaches, and is reminiscent of military command and control. A point of weakness for the hierarchical control architecture is recovering from failures of robots high in the control tree.

Decentralized control architectures are the most common approach for multirobot teams, and typically require robots to take actions based only on knowledge local to their situation. This control approach can be highly robust to failure, since no robot is responsible for the control of any other robot. However, achieving global coherency in these systems can be difficult, because high-level goals have to be incorporated into the local control of each robot. If the goals change, it may be difficult to revise the behavior of individual robots.

Hybrid control architectures combine local control with higher-level control approaches to achieve both robustness and the ability to influence the entire team's actions through global goals, plans, or control. Many multirobot control approaches make use of hybrid architectures.

A plethora of multirobot control architectures have been developed over the years. We focus here on three early approaches that illustrate the spectrum of control architectures. The first, the Nerd Herd, is representative of a pure *swarm* robotics approach using large numbers of homogeneous robots. The second, ALLIANCE, is representative of a behavior-based approach that enables coordination and control of possibly heterogeneous robots without explicit coordination. The third, distributed robot architecture (DIRA), is a hybrid approach that enables both robot autonomy and explicit coordination in possibly heterogeneous robot teams.

53.2.1 The Nerd Herd

One of the first studies of social behaviors in multirobot teams was conducted by *Matarić* [53.6], with results being demonstrated on the Nerd Herd team of 20 identical robots (shown in Fig. 53.1). This work is an example of swarm robotic systems, as described further in Sect. 53.4. The decentralized control approach was based on the subsumption architecture (Chap. 12), and assumed that all robots were homogeneous, but with relatively simple individual capabilities, such as detecting obstacles and *kin* (i.e., other robot team members). A set of basic social behaviors were defined and demonstrated, including obstacle avoidance, homing, aggregation, dispersion, following, and safe wandering. These basic behaviors were combined in various ways to yield more composite social behaviors, including flocking (composed of safe wandering, aggregation, and dispersion), surrounding (composed of safe wandering, following, and aggregation), herding (composed of safe wandering, surrounding, and flocking), and foraging (composed of safe wandering, dispersion, following, homing, and flocking). The behaviors were

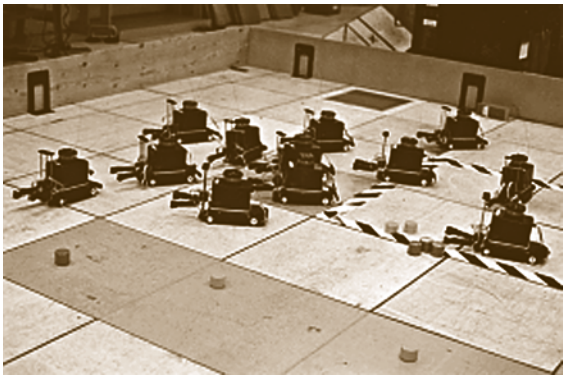

Fig. 53.1 The Nerd Herd robots

implemented as rules, such as the following rule for aggregate:

```
Aggregate:
  If agent is outside aggregation
  distance
    turn toward aggregation centroid
    and go.
  Else
    stop.
```

This work showed that collective behaviors could be generated through the combination of lower-level basic behaviors. Related work on this project studied issues such as using bucket brigades to reduce interference [53.7], and learning [53.8].

53.2.2 The ALLIANCE Architecture

Another early work in multirobot team architectures is the ALLIANCE architecture (Fig. 53.2), developed by *Parker* [53.9] for fault-tolerant task allocation in heterogeneous robot teams. This approach builds on the subsumption architecture by adding behavior sets and motivations for achieving action selection without explicit negotiations between robots. Behavior sets group low-level behaviors together for the execution of a particular task. The motivations consist of levels of impatience and acquiescence that can raise and lower a robot's interest in activating a behavior set corresponding to a task that must be accomplished.

In this approach, the initial motivation to perform a given behavior set is set to zero. Then, at each time step, the motivation level is recalculated based on:

1. The previous motivation level
2. The rate of impatience
3. Whether the sensory feedback indicates the behavior set is needed
4. Whether the robot has another behavior set already activated
5. Whether another robot has recently begun work on this task
6. Whether the robot is willing to give up the task, based on how long it has been attempting the task.

Effectively, the motivation continues to increase at some positive rate unless one of four situations occurs:

1. The sensory feedback indicates that the behavior set is no longer needed.
2. Another behavior set in the robot activates.
3. Some other robot has just taken over the task for the first time.
4. The robot has decided to acquiesce the task.

In any of these four situations, the motivation returns to zero. Otherwise, the motivation grows until it crosses a threshold value, at which time the behavior set is activated and the robot can be said to have selected an action. When an action is selected, cross-inhibition within that robot prevents other tasks from being activated within that same robot. When a behavior set is active in a robot, the robot broadcasts its current activity to other robots at a periodic rate.

The L-ALLIANCE extension [53.10] allows a robot to adapt the rate of change of the impatience and ac-

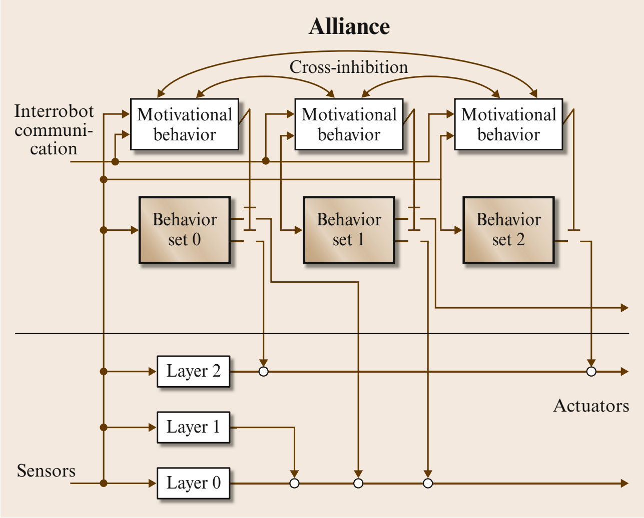

Fig. 53.2 The ALLIANCE architecture

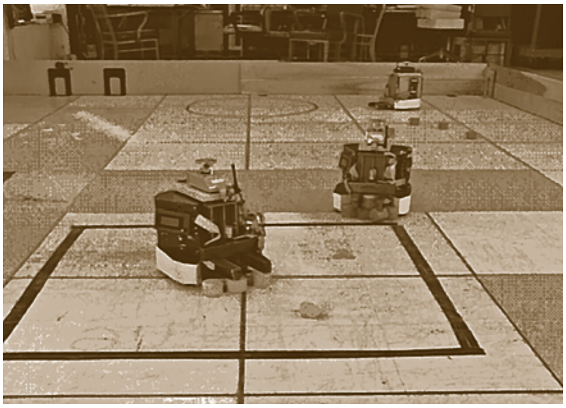

Fig. 53.3 Robots using the ALLIANCE architecture for a mock clean-up task

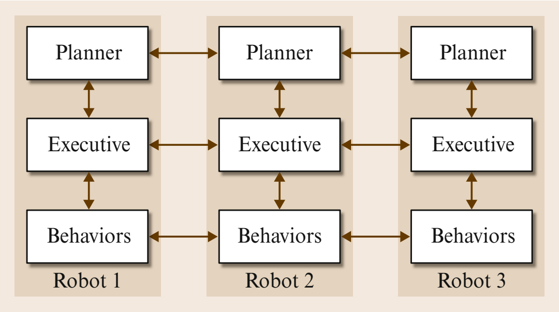

Fig. 53.4 The distributed robot architecture

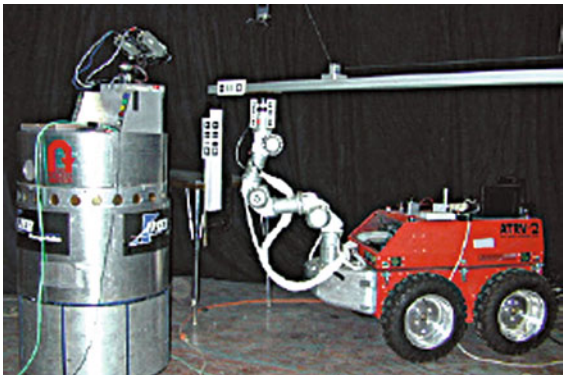

Fig. 53.5 Robots using the distributed robot architecture for assembly tasks

quiescence values depending on the quality with which that robot is expected to accomplish a given task. The result is that robots that have demonstrated their ability to better accomplish certain tasks are more likely to choose those tasks in the future. Additionally, if problems occur during team performance, then robots may dynamically reallocate their tasks to compensate for the problems. This approach was demonstrated on a team of three heterogeneous robots performing a mock clean-up task, two robots performing a box-pushing task, and four robots performing a cooperative target observation problem. The approach has also been demonstrated in the simulation of a janitorial service task and a bounding overwatch task. Figure 53.3 shows robots using ALLIANCE to perform the mock clean-up task.

53.2.3 The Distributed Robot Architecture

Simmons et al. [53.11] have developed a hybrid architecture called the distributed robot architecture (DIRA). Similar to the Nerd Herd and ALLIANCE approaches, the DIRA approach allows autonomy in individual robots. However, unlike the previous approaches, DIRA also facilitates explicit coordination among robots. This approach is based on layered architectures that are popular for single-robot systems (Chap. 12). In this approach (Fig. 53.4), each robot's control architecture consists of a *planning* layer that decides how to achieve high-level goals; an *executive* layer that synchronizes agents, sequences tasks, and monitors task execution; and a *behavioral* layer that interfaces to the robot's sensors and effectors. Each of these layers interacts with those above and below it. Additionally, robots can interact with each other via direct connections at each of the layers.

This architecture has been demonstrated in a team of three robots – a crane, a roving eye, and a mobile manipulator – performing a construction assembly task (Fig. 53.5). This task requires the robots to work together to connect a beam at a given location. In these demonstrations, a *foreman* agent decides which robot should move the beam at which times. Initially, the crane moves the beam to the vicinity of the emplacement based on encoder feedback. The foreman then sets up a behavioral loop between the roving eye and the crane robot to servo the beam closer to the point of emplacement. Once the beam is close enough, the foreman tasks the roving eye and the mobile manipulator to servo the arm to grasp the beam. After contact is made, the foreman tasks the roving eye and the mobile manipulator to coordinate to servo the beam to the emplacement point, thus completing the task.

53.3 Communication

A fundamental assumption in multirobot systems research is that globally coherent and efficient solutions can be achieved through the interaction of robots lacking complete global information. However, achieving these globally coherent solutions typically requires robots to obtain information about their teammates' states or actions. This information can be obtained in a number of ways; the three most common techniques are:

1. The use of implicit communication *through the world* (called *stigmergy*), in which robots sense the effects of teammate's actions through their effects on the world [53.6, 12–16]

2. Passive action recognition, in which robots use sensors to directly observe the actions of their teammates [53.17]
3. Explicit (intentional) communication, in which robots directly and intentionally communicate relevant information through some active means, such as radio [53.9, 18–21] – an area widely studied in the field of networked robot systems.

Each of these mechanisms for exchanging information between robots has its own advantages and disadvantages [53.22]. *Stigmergy* is appealing because of its simplicity and its lack of dependence upon explicit communications channels and protocols. However, it is limited by the extent to which a robot's perception of the world reflects the salient states of the mission the robot team must accomplish. *Passive action recognition* is appealing because it does not depend upon a limited-bandwidth, fallible communication mechanism. As with implicit cooperation, however, it is limited by the degree to which a robot can successfully interpret its sensory information, as well as the difficulty of analyzing the actions of robot team members. Finally, the *explicit communication* approach is appealing because of its directness and the ease with which robots can become aware of the actions and/or goals of its teammates. The major uses of explicit communication in multirobot teams are to synchronize actions, exchange information, and to negotiate between robots. Explicit communication is a way of dealing with the *hidden-state* problem [53.23], in which limited sensors cannot distinguish between different states of the world that are important for task performance. However, explicit communication is limited in terms of fault tolerance and reliability, because it typically depends upon a noisy, limited-bandwidth communications channel that may not continually connect all members of the robot team. Thus, approaches that make use of explicit communications must also provide mechanisms to handle communication failures and lost messages.

Selecting the appropriate use of communication in a multirobot team is a design choice dependent upon the tasks to be achieved by the multirobot team. One needs to carefully consider the costs and benefits of alternative communications approaches to determine the method that can reliably achieve the required level of system performance. Researchers generally agree that communication can have a strong positive impact on the performance of the team. One of the earliest illustrations of this impact was given in the work of *MacLennan* and *Burghardt* [53.24], which investigates the evolution of communication in simulated worlds and concludes that the communication of local robot information can result in significant performance improvements. Interestingly, for many representative applications, researchers have found a nonlinear relationship between the amount of information communicated and its impact on the performance of the team. Typically, even a small amount of information can have a significant impact on the team, as found in the study of *Balch* and *Arkin* [53.25]. However, more information does not necessarily continue to improve performance, as it can quickly overload the communications bandwidth without providing an application benefit. The challenge in multirobot systems is to discover the *optimal* pieces of information to exchange that yield these performance improvements without saturating the communications bandwidth. Currently, no general approaches to identifying this critical information are available; thus, the decision of what to communicate is an application-specific question to be answered by the system designer. *Dudek* et al.'s taxonomy of multirobot systems [53.26] includes axes related to communication, including communication range, communication topology, and communication bandwidth. These characteristics can be used to compare and contrast multirobot systems.

Several related issues of active research in communications for multirobot teams deal with dynamic network connectivity and topologies; for example, robot teams must either be able to maintain communications connectivity as they move, or employ recovery strategies that allow the robot team to recover when the communications connectivity is broken. These concerns may require robots to adapt their actions in response to the anticipated effects on the communications network, or in response to knowledge of the anticipated propagation behavior of information through the dynamic network. These and related issues are discussed next in the context of *networked robot systems*.

53.4 Networked Mobile Robots

Networked robots are multiple robots operating together coordinating and cooperating by *networked communication* to accomplish a specified task. Multiple robots enable new capabilities and the communication network enables new approaches and solutions that are difficult with just perception and control. Communication enables new control and perception capabilities in the system (e.g., access to information outside the perception range of the robot system). Conversely, control enables solutions for problems that are difficult without mobility (e.g., localization). Section 53.4.1 defines the field, examines the benefits of networking in robot coordination, and discusses applications. Section 53.4.2 highlights a few projects focused on networked robotics and discusses the application potential of the field. Section 53.4.3 discusses the research challenges at the intersection of control, communication, and perception. Section 53.4.4 defines a model for the control of a networked system which is used in Sects. 53.4.5–53.4.8 to examine specific research issues and opportunities facilitated by the interplay between communication, control, and perception.

53.4.1 Overview

The term *networked robots* refers to multiple robots operating together coordinating and cooperating by *networked communication* to accomplish a specified task. Communication between entities is fundamental to cooperation (and coordination), hence there is a central role for the communication network in networked robots. Networked robots may also involve coordination and cooperation with stationary sensors, embedded computers, and human users. The central feature of networked robots is the ability of the system to perform tasks that are well beyond the abilities of a single robot or multiple uncoordinated robots.

The IEEE (Institute of Electrical and Electronics Engineers) *Technical Committee on Networked Robots* has adopted the following definition of a networked robot:

A networked robot is a robotic device connected to a communications network such as the Internet or local-area network (LAN). The network could be wired or wireless, and based on any of of a variety of protocols such as the transmission control protocol (TCP), the user datagram protocol (UDP), or 802.11. Many new applications are now being developed ranging from automation to exploration. There are two subclasses of networked robots:

1. Teleoperated, where human supervisors send commands and receive feedback via the network. Such systems support research, education, and public awareness by making valuable resources accessible to broad audiences.
2. Autonomous, where robots and sensors exchange data via the network. In such systems, the sensor network extends the effective sensing range of the robots, allowing them to communicate with each other over long distances to coordinate their activity. Sensing, actuation, and computation need no longer be collocated. A broad challenge is to develop a science base that couples communication, perception, and control to enable such new capabilities.

This definition of autonomous networked robots also includes a third class of distributed systems, mobile sensor networks, which is a natural evolution of sensor networks. Robot networks allow robots to measure spatially and temporally distributed phenomena more efficiently. The robots in turn can deploy, repair, and maintain the sensor network to increase its longevity, and utility. The focus of this chapter is *autonomous networked robots*.

Embedded computers and sensors are becoming ubiquitous in homes and factories, and increasingly wireless ad hoc networks or plug-and-play wired networks are becoming commonplace. Human users interact with embedded computers and sensors to perform tasks ranging from monitoring (e.g., supervising the operation of a factor and surveillance in a building) to control (e.g., running an assembly line consisting of sensors, actuators, and material-handling equipment). In most of these cases, the human users, embedded computers, and sensors are not collocated and the coordination and communication happens through a network. *Networked robots* extends this vision to multiple robots functioning in a wide range of environments performing tasks that require them to coordinate with other robots, cooperate with humans, and act on information derived from multiple sensors.

Figure 53.6 shows prototype concepts derived from academic laboratories and industry. In all these exam-

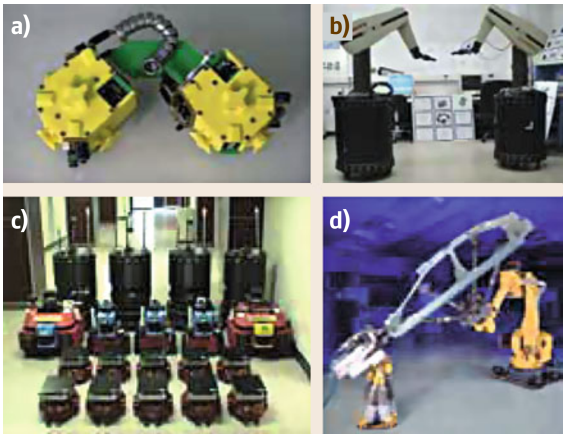

Fig.53.6a–d Small modules (after [53.27]) can automatically connect and communicate information to perform locomotion tasks **(a)**; robot arms (after [53.28]) on mobile bases can cooperate to perform household chores **(b)**; swarms of robots (after [53.29]) can be used to explore an unknown environment **(c)**; and industrial robots can cooperate in welding operations **(d)**

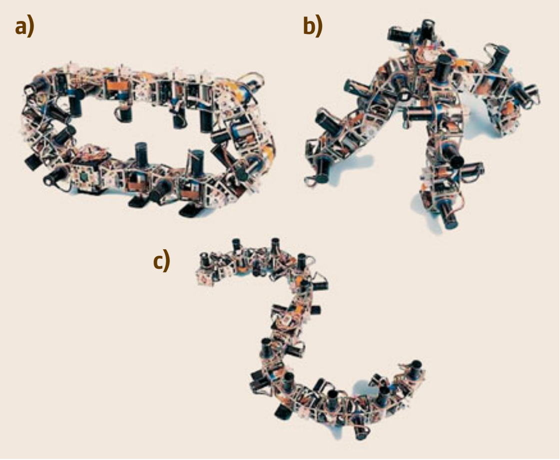

Fig.53.7a–c Robotic modules (after [53.30]) can be reconfigured to morph into different locomotion systems including **(a)** a wheel-like rolling system, **(b)** a snake-like undulatory locomotion system, and **(c)** a four-legged walking system

ples, independent robot or robotic modules can cooperate to perform tasks that a single robot (or module) cannot perform. Robots can automatically couple to perform locomotion tasks (also Fig. 53.7) and manipulation tasks that either a single robot cannot perform, or that would require a special-purpose larger robot to perform. They can also coordinate to perform search and reconnaissance tasks exploiting the efficiency that is inherent in parallelism. They can also perform independent tasks that need to be coordinated. Examples in the manufacturing industry include, for example, fixturing and welding.

Besides being able to perform tasks that individual robots cannot perform, networked robots also result in improved efficiency. Networking gives each robot access to information outside its perception range. Tasks such as searching or mapping can, in principle, be performed faster with an increase in the number of robots. A speed up in manufacturing operations can be achieved by deploying multiple robots performing operations in parallel but in a coordinated fashion.

Another advantage of using the network to connect robots is the ability to connect and harness physically removed assets. Mobile robots can react to information sensed by other mobile robots at a remote location. Industrial robots can adapt their end-effectors to new parts being manufactured upstream in the assembly line. Human users can use machines that are remotely located via the network.

The ability to network robots also enables fault tolerance in design. If robots can dynamically reconfigure themselves using the network, they are more tolerant to robot failures. This is seen in the Internet where multiple gateways, routers, and computers provide for a fault-tolerant system (although the Internet is not robust in other ways). Similarly, robots that can *plug* and *play* can be swapped in and out automatically to provide for a robust operating environment.

Finally, networked robots have the potential to provide great synergy by bringing together components with complementary benefits and making the whole greater than the sum of the parts.

Applications for networked robots abound. The US military routinely deploys unmanned vehicles that are reprogrammed remotely based on intelligence gathered by other unmanned vehicles, sometimes automatically. The deployment of satellites in space, often by astronauts in a shuttle with the shuttle robot arm, requires the coordination of complex instrumentation onboard the space shuttle, human operators on a ground station, the shuttle arm, and a human user on the shuttle. Home appliances now contain sensors and are becoming networked. As domestic and personal robots become more commonplace, it is natural to see these robots working with sensors and appliances in the house while cooperating with one or more human users. Networked robots will likely be used as critical ingredients in the environmental observatories of the future. Large-scale ecological monitoring precludes the use of monolithic infrastructure, and is envisioned to be built as a distributed, networked robotic system.

53.4.2 State of the Art and Potential

The growth in networked robot systems is broad-based, across many industries. There is a strong connection between this industry and the industry connected to sensor networks. Sensor networks have been projected to grow dramatically in terms of commercialization and market value [53.31]. Robot networks are analogous to sensor networks except that they allow sensors to have mobility and allow the geographical distribution of the sensors to be adapted based on the information acquired.

A system of robots, embedded computers, actuators, and sensors has tremendous potential in civilian, defense, and manufacturing applications. Nature provides the proof of concept of what is possible [53.32]. Group behaviors in nature can be found in organisms that are only microns to those that are several meters in length. There are numerous examples of simple animals that execute simple behaviors with modest sensors and actuators but communicate with and sense nearest neighbors to enable complex emergent behaviors that are fundamental to navigation, foraging, hunting, constructing nests, survival, and eventually growth. As seen

Fig. 53.8 Ants are able to cooperatively manipulate and transport objects often in large groups, without identified or labeled neighbors, and without centralized coordination

in Fig. 53.8, relatively small agents are able to manipulate objects that are significantly larger in terms of size and payload by cooperating with fairly simple individual behaviors. The coordination between agents is completely decentralized, allowing scaling up to large numbers of robots and large objects [53.33]. Individuals do not recognize each other. In other words, there is no labeling or identification of robots. The number of agents in the team is not explicitly encoded. Agents are identical, enabling robustness to failures and modularity. There is minimal communication, and even that which is present is only between neighbors. Furthermore, the optimal mode of group coordination may be scale dependent. Studies of wasps show strong evidence of centralized coordination among species with small colony sizes, but a distributed, decentralized coordination in larger colonies [53.34]. All these attributes are relevant to networked robots.

Biology has shown how simple decentralized behaviors in unidentified individuals (e.g., insects and birds exhibiting swarming behaviors) can exhibit a wide array of seemingly intelligent group behaviors. Similarly networked robots can potentially communicate and cooperate with each other, and even though individual robots may not be sophisticated, it is possible for networked robots to provide a range of intelligent behaviors that are beyond the scope of intelligent robots.

The significance and potential impact of networked robots is apparent from the following examples.

The manufacturing industry has always relied on integration between sensors, actuators, material-handling equipment, and robots. Today companies are finding it easier to reconfigure existing infrastructure by networking new robots and sensors with existing robots via wireless networks. There is also an increasing trend toward robots interacting with each other in operations like welding and machining, and robots cooperating with humans in assembly and material-handling tasks. Workcells consist of multiple robots, numerous sensors and controllers, automated guided vehicles, and one or two human operators working in a supervisory role. However, in most of these cells, the networked robots operate in a structured environment with very little variation in configuration and/or operating conditions.

There is a growing emphasis on networking robots in applications of field robotics, for example, in the mining industry. Like the manufacturing industry, operating conditions are often unpleasant and the tasks are repetitive. However, these applications are less structured and human operators play a more important role.

In the health care industry, networks allow health care professionals to interact with their patients, other professionals, expensive diagnostic instruments, and surgical robots. Telemedicine is expected to provide a major growth impetus for remote networked robotic devices that will take the place of today's stand-alone medical devices.

There are already many commercial products, notably in Japan, where robots can be programmed via and communicate with cellular phones. For example, the MARON robot developed by Fujitsu lets a human user dial up their robot and instruct it to conduct simple tasks including sending pictures back to the user via a cellular phone. Indeed these robots will interact with other sensors and actuators in the home – door openers equipped with Bluetooth cards and actuators and computer-controlled lighting, microwaves, and dishwashers. Indeed the Network Robot Forum [53.35] is already setting standards for how stationary sensors and actuators can interact with other robots in domestic and commercial settings.

Environmental monitoring is a key application for networked robots. By exploiting mobility and communication, robotic infrastructure enables observation and data collection at unprecedented scales in various aspects of ecological monitoring. This is significant for environmental regulatory policies (e.g., clean air and water legislation), as well as an enabler of new scientific discovery. For example, it is possible to obtain maps of salinity gradients in oceans, temperature and humidity variations in forests, and chemical composition of air and water in different ecological systems [53.36]. In addition to mobile sensor networks, it is also possible to use robots to deploy sensors and to retrieve information from the sensors. Mobile platforms allow the same sensor to collect data from multiple locations while communication allows the coordinated control and aggregation of information. Examples include systems built for aquatic [53.37], terrestrial [53.38], and subsoil monitoring [53.39]. There are many efforts to developed networked underwater platforms [53.40–42].

Networks of static and robotic devices have been developed for aquatic monitoring [53.37] and to obtain high-resolution information on the spatial and temporal distributions of plankton assemblages and concomitant environmental parameters. The RiverNet project [53.43] at Rensselaer Polytechnic Institute (RPI) has focused on the development of robotic sensor networks for monitoring a river ecosystem. Recent work at University of California, Los Angeles (UCLA), University of Southern California (USC), University of California, Riverside, and University of California, Merced on the networked infomechanical system project [53.38] has focused on the development of robotic networks for monitoring the forest canopy, with a view to providing data for modeling canopy and undercover growth. Networked robotic mini-rhizotrons [53.39] are being deployed in the forest to monitor root growth in the soil.

In the defense industry, countries like the USA have invested heavily in the concept of networked, geographically distributed assets. Unmanned aerial vehicles like the Predators are operated remotely. Information from sensors on the Predators triggers the deployment of other vehicles and weapon systems at a different remote location and allows commanders in a third location to control and command all these assets. The US military engaged in the large Future Combat Systems initiative to develop network-centric approaches to deploying autonomous vehicles. The network-centric tactical paradigms for modern warfare have created networked robots for defense and homeland security. While networked robots are already in operation, current approaches are limited to human users commanding a single vehicle or sensor system. However, it takes many human operators (between 2–10 depending on the complexity of the system) to deploy complex systems like unmanned aerial vehicles. A Predator unmanned aerial vehicle (UAV) is operated from a tactical control station, which may be on an aircraft carrier, with a basic crew of 3–10 operators.

The eventual goal, however, is to enable a single human user to deploy networks of unmanned aerial, ground, surface, and underwater vehicles. There have been several recent demonstrations of multirobot systems exploring urban environments [53.44, 45] and the interiors of buildings [53.46, 47] to detect and track intruders, and transmit all of the above information to a remote operator. These examples show that it is possible to deploy networked robots using an off-the-shelf 802.11b wireless network and have the team be remotely tasked and monitored by a single operator. An example of a project with heterogeneous vehicles in an urban setting is shown in Fig. 53.9. An example of a project with heterogeneous vehicles in an indoor

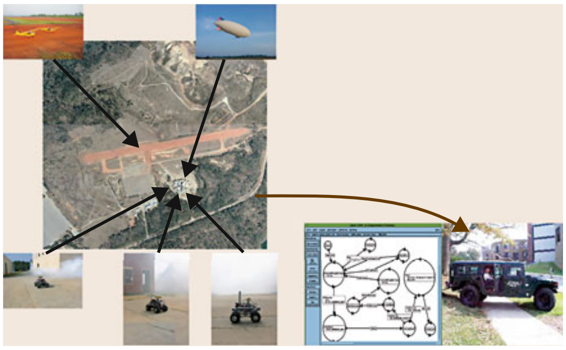

Fig. 53.9 A single operator commanding a network of aerial and ground vehicles from a command and control vehicle in an urban environment for scouting and reconnaissance in a recent demonstration by the University of Pennsylvania, Georgia Tech. and University of Southern California (after [53.48])

setting is shown in Fig. 53.10 wherein robots map an environment and deploy themselves to form a sensor network to detect intruders.

Many research projects are addressing group behaviors or collective intelligence by realizing swarming behaviors observed in nature. For example, the European Union (EU) funded several EU-wide coordinated projects on collective intelligence or swarm intelligence. The I-Swarm project in Karlsruhe [53.49] and the Swarm-Bot project at Ecole Polytechnique Fédérale de Lausanne (EPFL) [53.50] are examples of swarm intelligence. The Laboratory for Analysis and Architecture of Systems (LAAS) has a strong group in robotics and artificial intelligence. This group has had a long history of basic and applied research in multi-robot systems. The integration of multiple unmanned vehicles for applications such as terrain mapping and fire-fighting is addressed in [53.51]. A multi-university US project addressed the development of networked vehicles for swarming behaviors [53.52]. Projects such as these are exploring the scalability of the basic concepts to large numbers of robots, sensors, and actuators.

53.4.3 Research Challenges

While there are many successful embodiments of networked robots with applications to manufacturing industry, the defense industry, space exploration, domestic assistance, and civilian infrastructure, there are significant challenges that have to be overcome.

The problem of coordinating multiple autonomous units and making them cooperate creates problems at the intersection of communication, control, and perception. Who should talk to whom and what information should be conveyed, and how? How does each unit

Fig. 53.10 Under the DARPA SDR program, a team from the University of Southern California, the University of Tennessee, and Science Applications and International Corporation (SAIC) demonstrated mapping, and intruder detection by a team of networked robots (after [53.46])

move in order to accomplish the task? How should the team members acquire information? How should the team aggregate information? These are all basic questions that need basic advances in control theory, perception, and networking. In addition, because humans are part of the network (as in the case of the Internet), we have to devise an effective way for multiple humans to be embedded in the network and command/control/monitor the network without worrying about the specificity of individual robots in the network. Thus the underlying research challenges lie at the intersection of control theory, perception, and communication/networks, as shown in Fig. 53.11.

It is also worth noting that robot networks are dynamic, unlike networks of sensors, computers or machines which might be networked together in a fixed topology. When a robot moves, its neighbors change and its relationship to the environment changes. As a consequence, the information it acquires and the actions it executes must change. Not only is the network topology dynamic, but the robot's behavior also changes as the topology changes. It is very difficult to predict the performance of such dynamic robot networks, yet it is this analysis problem that designers of robot networks must solve before deploying the network.

This notion of a changing topology inevitably leads us to complicated mathematical models. Traditionally, models of group behavior have been built on continuous models of dynamics of individuals, including local interactions with neighbors, and models of control and sensing with a fixed set of neighbors. While dynamics at the level of individual units may be adequately described by differential equations, the interactions with neighbors are best described by edges on a graph. Modeling, analysis, and control of such systems will require a comprehensive theoretical framework and new representational tools. New mathematical tools that marry dynamical system theory, switched systems, discrete mathematics, graph theory, and computational geometry are needed to solve the underlying problems. We need a design methodology for solving the *inverse*

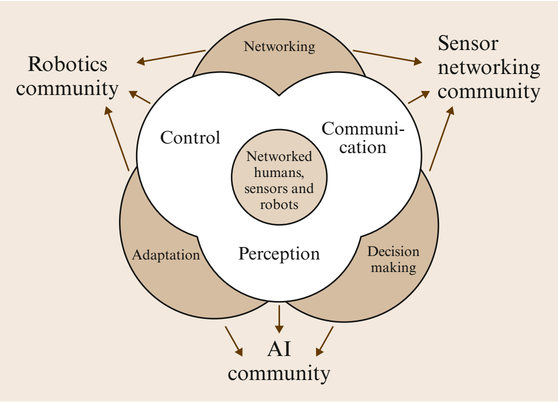

Fig. 53.11 The paradigm of *networked robots* introduces fundamental challenges at the intersection of control, perception, and communication that is of interest to the robotics, sensor networks, and artificial intelligence communities

problem in navigation – behaviors for controlling individuals to achieve a specified aggregate *motion* and *shape* of the group, and the application to active perception and coverage. An overview of some of these methods is provided in Sect. 53.4.4.

Problems of perception have been studied extensively in the robotics community. However, the perception problems in a system of networked, mobile sensor platforms bring a new set of challenges; for example, consider the problem of estimating the state of the network. State estimation requires the estimation of the state of robots and the environment based on local, limited-range sensory information. Localization of n vehicles in an m-dimensional configuration space requires $O((nm)^k)$ computations, where k is somewhere between 3 and 6, depending on the algorithm and domain-specific assumptions. The estimation problem is further exacerbated by the fact that not all robots in the network may be able to get the necessary information in a time-critical fashion. There are deep issues of representation and algorithmic development, which are discussed in Sect. 53.4.6.

The paradigm of active perception [53.53] links the control of sensor platforms to perception, bringing control theory and perception together in a common framework. Extending this paradigm to networked robots requires approaches of distributed control to be merged with decentralized estimation. Robots can move in order to localize themselves with respect to their neighbors, to localize their neighbors, and also to identify, localize, and track features in the environment. These problems are discussed in Sect. 53.4.7.

As discussed earlier, the communication network is central to the functioning of a network of robots.

However, if the network consists of mobile agents with transmitters and receivers with finite power, there is no guarantee that all agents can talk to teach other. Unlike a static sensor network, robots in a network can move toward each other to facilitate communication and adaptively maintain a communication network. Some basic algorithmic problems and several pertinent results are provided in Sect. 53.4.8.

53.4.4 Control

The control of individual robots is critical to the performance and scope of robot networks. Indeed motion coordination algorithms have been proposed for the purpose of improving communication performance [53.54, 55], localization [53.56, 57], information integration, deployment [53.58], and coverage [53.59–61], among other tasks. Mobility allows the group of robots to self-deploy, and self-organize by relocating themselves in support of communication, sensing, or task needs; for example, they can reconfigure to guarantee a desired communication bandwidth, k-hop connectivity, or algebraic connectivity, enabling message delivery from one robot to another. The group can also self-organize to position sensors so as to cover a desired area and adapt to shifts in the focus of monitoring activities. Controlling sensor position also supports map making, tracking of objects and events, and goal-directed navigation for users of the network. Finally, mobility allows robots to accomplish tasks such as navigation, reconnaissance, transportation, and search and rescue.

Given a group of mobile sensors, we would like to have distributed control capabilities that realize desirable global specifications. Thus, it is necessary to be able to automatically determine the necessary position and orientation of the group members and/or the distribution of group members, and their motion to achieve the desired task. At a lower level, the robots must be able to use information from the communication network and from their own sensors to derive local estimates, reason about the spatial network (their neighbors and their relationship to the environment), and then use the appropriate control policies to achieve the desired group specifications. We briefly outline the simplest mathematical model that is necessary to formulate such problems in order to provide a better sense of the underlying challenges.

In a robot network, we have multiple agents or nodes in which each agent is a physical entity that can be a robot, a vehicle with actuators and sensors, a sensor platform (possibly static) or even a communication relay node. Each agent A_i is characterized by an identifier, $i \in I \subset Z$, a state $x_i \in X_i \subset \mathbb{R}^n$, and control inputs $u_i \in U_i \subset \mathbb{R}$, with $f_i : X_i \times U_i \to TX_i$ specifying the dynamics

$$\dot{x}_i = f_i(x_i, u_i) . \tag{53.1}$$

The state x_i will consist of the position (and orientation), r_i in some d-dimensional space, and its velocity, $\dot{r}_i : x_i = \left(r_i^{\mathrm{T}}, \dot{r}_i^{\mathrm{T}}\right)^{\mathrm{T}}$, with $n = 2d$. $\mathcal{N}^{\mathrm{c}}(r_i)$ and $\mathcal{N}^{\mathrm{s}}(r_i)$ are neighborhoods of r that define the range and field of view of the communication hardware and sensors, respectively.

A network of robots S consists of N agents with a *sensing graph* and a *communications graph* that is defined by the physical distribution of the agents. The sensing graph (and similarly the communications graph) is defined by a map $E^{\mathrm{s}} : X^1 \times X^2 \cdots X^N \to I \times I$, where the edges of the graph are formed dynamically depending on the physical proximity of pairs of agents. Specifically, the $N \times N$ adjacency matrix, $\mathcal{A}^{\mathrm{s}}$ (and similarly $\mathcal{A}^{\mathrm{c}}$) has entries

$$\mathcal{A}^{\mathrm{s}}{}_{ij} = \begin{cases} 1, & \text{if } r_j \in \mathcal{N}^{\mathrm{s}}(r_i) , \\ 0, & \text{otherwise} . \end{cases} \tag{53.2}$$

Agent A_i has estimates of its own state and the states of neighbors (e.g., A_j), and these estimates are derived from information associated with edges in the sensing and communication graph

$$\hat{x}_j^{(i)} = h(x_i, z_{ij}) , \tag{53.3}$$

where z_{ij} represents measurements of the state of agent A_j available to A_i by sensing or communication channels and h is the estimator used by A_i. Note that z_{ij} may have dimension less than n and may therefore not contain complete information about $x_{ij} = x_i - x_j$. Clearly the relative position vector denoted by $r_{ij} = r_i - r_j$ and its magnitude are important quantities that may need to be estimated for biological and artificial agents.

Finally, A_i can encode n_{b_i} behaviors, which we will denote by $\mathcal{B}_i = B_1, B_2, \ldots, B_{n_{b_i}}$. Each behavior B_j is a controller, a function $k_j : \mathbb{R} \times X_i \to U_i$. All agents can be assigned identical or different behaviors. Each behavior represents a set of unsynchronized, locally executed computations (for control or estimation) being carried out for some collective purpose, with each processor using in its computations only data from its *neighboring* processors. Furthermore, even for a fixed assignment of behaviors, each processor's neighbors typically *change* with time because the processors are moving in and out of the sets $\mathcal{N}^{\mathrm{c}}$ and $\mathcal{N}^{\mathrm{s}}$. Thus the methodology for modeling and analyzing such systems will require the merging of graph theory and dynamical system theory at a fundamental level.

The reader is directed to the many survey articles on this subject for further information. An overview of challenges for the controls community is presented in [53.62]. The underlying theory for networked mobile systems has been explored in the context of automated highway systems [53.63], cooperative robot reconnaissance [53.46] and manipulation [53.64], formation flight control [53.65], and the control of groups of unmanned vehicles [53.45]. Our goal in the following sections is to explore the connections between communication, perception, and control.

53.4.5 Communication for Control

Communication networks allow physically disconnected entities to exchange information. At the lowest level, when groups of vehicles coordinate their actions, communication allows vehicles to exchange state information [53.66–68]. At a higher level, robots can plan navigation and exploration tasks based on an integrated map of the world derived from information acquired from different robots [53.52].

The use of communication for control in the multivehicle context has been addressed in the PATH project where formations of inline vehicles were studied [53.63]. Problems of the stability of the formation [53.69], the convergence of the formation to shapes [53.70], and the overall performance of the system [53.71] are of great interest. The performance of the system is directly influenced by the interconnections between agents. In addition to impacting on stability [53.63], feedback of states from different agents and feedforward information from the plans of different agents affects the rates at which the system of agents can respond to external stimuli [53.71] or to commands from human operators [53.72].

In addition, communication can be used for high-level control and planning of robots. There is great interest in using static sensor nodes as beacons to guide robot navigation. In [53.73], the problem of coverage and exploration of an unknown dynamic environment using a mobile robot is considered. An algorithm is presented which assumes that global information is not available (neither a map, nor global positioning system (GPS) information). The algorithm deploys a network of radio beacons that assists the robot in coverage. The network is also used by the robot for navigation. The deployed network can also be used for applications other than coverage (such as multirobot task allocation). A similar idea was presented using potential-field-based navigation in [53.52]. In this work the notion of no-go or danger areas was incorporated into the navigation cost function. Recent work along these lines with experimental data from sensor nodes is reported in [53.74].

In such communication-enabled cooperative control and planning (see also [53.75]), the communication network plays an important role in the creation of a shared representation of information. This notion of a shared representation is important to the scaling of coordinated control algorithms to large numbers of devices. For example, in [53.67], the information form of the Kalman filter is used to derive a framework for decentralizing estimation and fusion algorithms. This approach was shown to be applicable to multiple heterogenous ground and aerial platforms [53.56]. Such methodologies are transparent to the specificity and identity of the cooperating vehicles. This is because vehicles share a common representation, which consists of a certainty grid that contains information about the probability of detection of targets, and an information vector–matrix pair that is used in the information form of the Kalman filter [53.45]. Observations are propagated through the network by changing both the certainty grid and the information vector/matrix. This allows each vehicle to choose the action that maximizes a utility function, which is the combined mutual information gain from onboard sensors towards the detection and localization of features in the environment.

Thus, in summary, at the lowest level, communication enables either partial or complete state feedback of the network and allows agents to exchange information for feedforward control. At the higher levels, agents can share information for planning and for control. This is also discussed in Sect. 53.4.6 where the communication network is shown to enable a network-centric approach to perception.

53.4.6 Communication for Perception

While individual robots have sensors and the ability to build maps and models by integrating sensory information, networked robots can exchange information and leverage sensory data, maps, and models from other robots. The challenge is to exploit communication for perception in tasks such as distributed mapping in the presence of the delays, limited bandwidth, and disruption that are typical of communication networks.

Distributed localization is the term used to describe the merging of communication and perception for state estimation. Localization is an essential tool for the development of low-cost robot networks for use in location-aware applications and ubiquitous networking [53.76]. Location information is needed to track the placement of the nodes and to correlate the values measured by the node with their physical location. Distributed computation and robustness in the presence of measurement noise are key ingredients for a practical localization algorithm that will give reliable results over a large-scale network.

The methods for distributed localization can be classified into two broad classes: algorithms that rely on anchor nodes for localization and algorithms that use no beacons. Localization may be computed using range information between nodes, bearing information, or both.

In [53.54] a theoretical foundation for network localization in terms of graph rigidity theory is provided. The problem is solved when nodes have perfect range information and it is show that a network has a unique localization if and only if its underlying graph is *generically globally rigid*. In [53.77] the Cramér–Rao lower bound (CRLB) for network localization is derived. This work computes the expected error characteristics for an ideal algorithm, and compares this to the actual error in an algorithm based on multilateration, drawing the important conclusion that the error introduced by the algorithm is just as important as the measurement error in assessing end-to-end localization accuracy. In [53.78] a distributed algorithm that uses no beacons and is guaranteed to compute correct location information under measurement noise for nodes that can range to neighbors is presented. This algorithm relies on the notion of robust quadrilaterals to compute robustly a global system of coordinates among the nodes. The computation supports moving nodes. Extensions of this work to passive tracking have been discussed in [53.79]. Localization based on the propagation of location information from known reference nodes based on connectivity includes [53.80, 81]. Mobility-assisted localization is introduced in [53.82]. Other techniques use distributed propagation of location information using multilateration [53.77, 83]. In a recent paper [53.84] the problem of evaluating the rigidity of a planar network is treated while satisfying common objectives of real-world systems: decentralization, asynchronicity, and parallelization.

Two approaches for cooperative relative localization of mobile robot teams are given in [53.85, 86]. Neither method uses GPS, landmarks, or maps of any kind; instead, robots make direct measurements of the relative pose of nearby robots and broadcast this information to the team as a whole. In [53.85], each robot processes this information independently to generate an egocentric estimate for the pose of other robots using a Bayesian formalism with a particle filter implementation. In [53.86], maximum-likelihood estimation (MLE) and numerical optimization is used to achieve a similar result.

A key issue is to be able to scale these computations for building a shared representation to large numbers of robots and sensors. This problem was studied in experiments under the US Defense Advanced Research Projects Agency (DARPA-funded software for distributed robotics (SDR) program. The goal of these experiments was to develop and demonstrate a multi-robot system capable of carrying out a specific mission. This required the ability to deploy a large number of robots into an unexplored building, map the building interior, detect and track intruders, and transmit all of the above information to a remote operator. A report on one set of experiments is presented in [53.46]. A tiered strategy for deploying the robots is described, where highly capable robots formed the first wave to enter and map a building, followed by a second wave which used the resulting map to self-deploy and monitor the environment for intruders. Both approaches relied extensively on networking the robots using commercial 802.11b wireless technology. This task involved both communication for building a shared representation as well control for perception.

Another important set of problems arises when robot networks are used for identifying, localizing, and then tracking targets in a dynamic setting. An embedded stationary wireless sensor network is like a virtual sensor spread over a large geographical area. Such a network can provide information to mobile robots about remote locations. Robot networks allow this virtual sensor to move in response to external stimuli and to track moving targets. Indeed, it is possible to cast this scenario as a pursuit-evasion game with robotic sensor networks [53.87]. For example, the Tenet project at USC addresses the design of network primitives and abstractions for tiered network architectures, with robotic pursuit evasion as one of the target applications. Algorithms for guiding the sampling strategy of a robotic boat to model and locate phenomena of interest (e.g., hotspots) in aquatic environments are discussed in [53.37]. The networked infomechanical systems (NIMS) project has focused on sensor-assisted techniques for mobile robot-based adaptive sampling for event response [53.88] and field reconstruction [53.89].

The information collected by the nodes in a sensor network can be processed at a central location or in a decentralized fashion. Such in-network data processing techniques make better use of network communication and computation resources than centralized processing. This also enables the network to compute accurate and up-to-date global pictures of the global perception landscape that are available to all the robots in the system. Methods for in-network data processing with static nodes include artificial potential-field computation, gradient computations, particle filters, Bayesian inference, and signal processing. Algorithms have been developed for computing maps, paths, and predictors [53.52, 73, 90].

A recent DARPA demonstration showed how communication networks can be used effectively in perception tasks involving heterogenous robots [53.44]. In cooperative search, identification, and localization unmanned aerial vehicles (UAVs) can be used to cover large areas, searching for targets. However, sensors on UAVs are typically limited in their accuracy of localization of targets on the ground. On the other hand, ground robots can be deployed to accurately locate ground targets but have the disadvantage of not being able to move rapidly and see through obstacles such as buildings or fences. In [53.56], the synergy between these two devices is exploited by creating a seamless network of UAVs and unmanned ground vehicles (UGVs). As discussed in Sect. 53.2, the key to such network-centric approaches for search and localization is a shared representation of state information, which in this case is easily scalable to large numbers of UAVs and UGVs and is transparent to the specificity of individual platforms. However, how to do this more generally and for more unstructured information remains an issue for future research.

53.4.7 Control for Perception

Networked mobile robots enable the exploration of dynamic environments and the recovery of three-dimensional information via distributed active perception [53.53]. Since the nodes are mobile, a natural question is: where should the nodes be placed in order to ensure successful integration of information from multiple nodes, and to maximize the quality of the estimates returned by the team? Since there is a cost associated with transmitting and processing data, it is important to consider which sensor readings should be used in the state estimation and what information should be communicated to the rest of the system. The quality of the information computed by the network depends on the locations of the sensor platforms both in an absolute and relative sense. The quality also depends on the noise characteristics of each sensor, and the communication network.

A robot network goes well beyond a fixed sensor network, which can only collect data at fixed positions in space; for example, when an event is detected at a specific location it is possible to direct more sensors toward the location of observation of the event for more information (for example, higher-resolution data or higher sampling frequency). Reconfiguring the node locations for adaptive resolution sampling relies on distributed control strategies.

Various strategies have been introduced for controlling mobile sensor network coverage. Mobile sensing agents are controlled using gradients of information-based objective functions [53.91]. Stability results are derived without concerns for the optimality of the network configuration, but local guarantees are provided. Topology aware coordinated behavior is treated in [53.92]. A body of results reported in [53.93] and [53.94] describes decentralized control laws for positioning mobile sensor networks optimally with respect to a known event distribution density function. This approach is advantageous because it guarantees that the network (locally) minimizes a cost function relevant to the coverage problem. However, the control strategy requires that each agent have a complete knowledge of the event distribution density, thus it is not reactive to the sensed environment. The work by [53.95, 96] generalizes these results to situations in which the nodes estimate rather than know ahead of time the event distribution density function. A local (decentralized) control law requires that each agent can measure the value and gradient of the distribution density function at its own position. This results in a sensor network that is reactive to its sensed environment while maintaining or seeking a near-optimal sensing configuration. In addition, the distribution density function approximation yields a closed-form expression for the control law in terms of the vertices of an agent's Voronoi region. This eliminates the need for the numerical integration of a function over a polynomial domain at every time step, thereby providing a significant reduction in computational overhead for each agent. Other work in event monitoring for unknown distributions includes [53.59]. *Krause* et al. [53.97] have recently proposed an approach for sensor placement that considers both the sensing quality and communication cost of imperfect sensing and communication components. They use a parametric model for link reception rate that assumes no acknowledgement and no temporal correlation of lossy links.

Beginning with the art gallery problem, there have been multiple efforts to determine an optimal configuration of sensors to cover a given region [53.98–100]. A variant which allows the use of mobile sensors is known as the *watchmen tours* problem. In these approaches the sensor model is abstract and not well suited to real environments and cameras. Distributed geometric optimization methods [53.94] have also been used for mobile sensor network reconfiguration. A related class of methods is the use of estimation-theoretic optimization metrics and the application of information filters to coordinate network-wide motion [53.56]. There are other distributed optimization methods which use a distributed control law and show that it optimizes a global metric of interest, such as using a potential field or other linear control law based only on local neighbor interactions [53.101]. Research focusing on the control of cameras with pan, tilt, and zoom capabilities is due to [53.60, 102, 103]. In [53.102] an approach is developed to calibrate a pan–tilt–zoom camera automatically over its full zoom range and to build very high-resolution panoramas. In [53.60], the cameras are constantly moved to track observed targets, using a factor graph. A recent algorithm due to [53.104] significantly improves on this by positioning cameras to make the network better suited to detect and classify targets as they emerge. Pan–tilt–zoom cameras allow the construction of far more flexible vision systems than static cameras.

53.4.8 Control for Communication

In Sect. 53.4.5, we briefly discussed the benefits of using the communication network to synthesize and improve controller design. Conversely, the movement of robots affects the network and data transmission in the network. This gives rise to many challenges. If the controllers for individual robots are known, can we provide guarantees about communication in the network and can we develop robust information routing and networking algorithms in the presence of robot motion? Another challenge concerns how information propagates and diffuses in these networks. If the robots move under a given control model, how does a piece of information propagate through the network and what can we say about when and where that piece of information will be heard? If we know the answers to such questions, it may be possible to design controllers to realize desired communication network characteristics.

One simple control strategy that can affect network performance is to control the robot motion to ensure messages are transmitted between designated nodes. The movement of robots in a network of robots and sensors may cause network partitioning when nodes go out of range. However, the ability of the robots to move in a controlled way also leads to an opportunity to address the information routing problem in disconnected networks by turning the robots into relay nodes. The key idea here is to enable the robot holding a current message to an unavailable destination to modify their trajectory in order to relay a message. This problem has been formulated as an optimization problem. The goal is to minimize the trajectory modifications necessary to send a message to its destination. Several solutions have been proposed depending on the information that is available to the robots. If the robots' trajectories are known, path planning techniques can be used to compute who moves where to relay what. If the robots' trajectories are not known, a distributed spanning tree can be created to enable robots to keep track of each other. Each robot is assigned a region of movement and a parent in the spanning tree. When the robot leaves its region, the parent is informed. When the robot moves too far away, the spanning tree is modified.

Mobile robots can be used to create desired network topologies under suitable models of network communication. If a robot is used to emplace nodes in an environment (or if sensor nodes robotically self-deploy) to build a network, the problem is referred to as deployment. It is possible to control the motion of individual nodes to guarantee that a specified topology is maintained [53.55]. It is also possible to reposition nodes with the explicit aim of changing the network topology – the so-called mobility-based topology control problem.

A distributed algorithm for the deployment of mobile robot teams has been described by [53.105] using the concept of virtual pheromones: localized messages from one robot to another. These messages are used to generate either a gas expansion or a guided growth deployment model. Similar algorithms based on artificial potential fields are described in [53.106, 107], where the latter incorporates a connectivity constraint. An incremental deployment algorithm for mobile sensor networks is given in [53.58]; nodes are deployed one at a time into an unknown environment, with each node making use of information gathered by previously deployed nodes to determine its deployment location. The algorithm is designed to maximize network coverage while ensuring that nodes retain line of sight with one another.

Most work on network topology control has dealt with uncontrolled deployments, where there is no explicit control of the positions of individual nodes. The primary mechanisms proposed are power control and sleep scheduling. These methods involve pruning an already existing, well-connected communication graph in order to save power while ensuring that the resultant subgraph preserves connectivity. Given a network that is connected when all nodes are operating at maximum power, the aim of power control is to use the minimum power level at each node for which the network remains connected [53.108]. Given an overdeployed network, sleep scheduling seeks to activate a minimal subset of nodes to maintain connectivity and achieve other desired metrics [53.109]. In contrast, controlled deployments are feasible when the positions of individual nodes can be altered. Such deployments are interesting for two reasons. First, network topology with wireless communication relates directly to proximity relations and hence the position of the nodes. Second, there is increasing evidence that a large number of deployments are likely to involve careful, nonrandom placement of nodes. The positioning of nodes is controlled either by the nodes themselves or by external agents. Such networks present a different and interesting scenario for topology control since it is possible to exploit control of the motion and placement of the nodes to build efficient topologies. A local, completely decentralized technique for topology control using mobility is given in [53.110].

An important application for networked robots is in monitoring and surveillance, where it is important that the robots cover the space while remaining within communication range [53.111–113]. In a recent development [53.114] the problem of how to design communication models and scheduling protocols for choosing the appropriate path planning algorithms for robotic data collection is discussed. Probing environment and adaptive sleeping protocol (PEAS) was one of the first attempts to address communication connectivity and sensing coverage simultaneously using heuristic algorithms [53.115]. *Wang* et al. [53.116] proposed a new coverage configuration protocol (CCP) to produce an approach that simultaneously optimizes coverage and connectivity while maximizing the number of nodes that are placed into sleep mode. Furthermore, they also identified three different classes of coverage–connectivity problems with respect to the ratio of radio and sensing ranges and recognized the critical ratio where the former range is twice as long as the latter. *Zhang* and *Hou* proved that, if the communication range is at least twice the sensing range, complete coverage of a convex area guarantees network communication connectivity, and then used this theorem as a basis for a localized density control algorithm [53.109]. This was subsequently generalized to show that the condition that the communication range is twice the sensing range is sufficient and is the tight lower bound to guarantee that complete coverage preservation implies communication connectivity among nodes if the original network topology is connected [53.117].

In summary, if the state of the communication network and the desired state of the communication network is known to each agent, it should be possible to synthesize distributed controllers to move agents to attain desired network characteristics. However, the assumptions on the global state are clearly not justified. Also, the desired motion to optimize network characteristics will conflict with the motion that is required to perform the desired task. However, as the brief discussion above illustrates, there are many interesting studies that point to promising directions for future work in this very fertile research field.

53.5 Swarm Robots

Historically, some of the earliest work in multirobot systems [53.12, 13, 118–125] dealt with large numbers of homogeneous robots, called *swarms*; swarm robotics continues to be a very active area of research. Most swarm approaches obtain inspiration from biological societies – particularly ants, bees, and birds – to develop similar behaviors in multirobot teams. Because biological societies are able to accomplish impressive group capabilities, such as the ability of termites to build large complex mounds, or the ability of ants to collectively carry large prey, robotics researchers aim to reproduce these capabilities in robot societies.

Swarm robotics systems are often called *collective* robotics, indicating that individual robots are often unaware of the actions of other robots in the system, other than information on proximity. These approaches aim to achieve a desired team-level global behavior from the interaction dynamics of individual robots following relatively simple local control laws. Swarm robotic systems typically involve very little explicit communication between robots, and instead rely on stigmergy (i. e., communication through the world) to achieve emergent cooperation. Individual robots are assumed to have minimal capabilities, with little ability to solve meaningful tasks on their own. However, when grouped with other similar robots, they are collectively able to achieve team-level tasks. Ideally, the entire team should be able to achieve much more than individual robots working alone (i. e., it is *superadditive*, meaning that the whole is bigger than the sum of the parts). These systems assume very large numbers of robots (at least dozens, and often hundreds or thousands) and explicitly address issues of scalability. Swarm robotic approaches achieve high levels of redundancy because robots are assumed to be identical, and thus interchangeable with each other.

Many types of swarm behaviors have been studied, such as foraging, flocking, chaining, search, herding, aggregation, and containment. The majority of these swarm behaviors deal with spatially distributed multirobot motions, requiring robots to coordinate motions either:

1. Relative to other robots
2. Relative to the environment
3. Relative to external agents
4. Relative to robots and the environment
5. Relative to all (i. e., other robots, external agents, and the environment).

Table 53.1 categorizes swarm robot behaviors according to these groupings, citing representative examples of relevant research.

Table 53.1 Categories of swarm behaviors

Relative motion requirements	Swarm behaviors
Relative to other robots	Formations [53.126, 127], flocking [53.121], natural herding (as in herds of cattle), schooling, sorting [53.14], clumping [53.14], condensation, aggregation [53.128], dispersion [53.129]
Relative to the environment	Search [53.130], foraging [53.131], grazing, harvesting, deployment [53.58], coverage [53.132], localization [53.133], mapping [53.134], exploration [53.135]
Relative to external agents	Pursuit [53.136], predator–prey [53.137], target tracking [53.138], forced herding/shepherding (as in shepherding sheep)
Relative to other robots and the environment	Containment, orbiting, surrounding, perimeter search [53.139]
Relative to other robots, external agents, and the environment	Evasion, tactical overwatch, soccer [53.140]

Much of the current research in swarm robotics is aimed at developing specific solutions to one or more of the swarm behaviors listed in Table 53.1 (VIDEO 214). Some of these swarm behaviors have received particular attention, notably formations, flocking, search, coverage, and foraging. Section 53.10 discusses these behaviors in more detail. In general, most current work in the development of swarm behaviors is aimed not just at demonstrating group motions that are similar to biological systems, but also at understanding the formal control theoretic principles that can predictably converge to the desired group behaviors, and remain in stable states.

Demonstration of physical robot swarms is both a hardware and a software challenge. As discussed in Sect. 53.2, the first demonstrations were by *Matarić* [53.6], involving about 20 physical robots performing aggregation, dispersion, and flocking. This work defined composable *basis behaviors* as primitives for structuring more complex systems. In later research, *McLurkin* [53.141] (VIDEO 215) developed an extensive catalog of swarm behavior software, and demonstrated these behaviors on about 100 physical robots (called the SwarmBot robots), developed by iRobot, as shown in Fig. 53.12. He created several group behaviors, such as avoidManyRobots, disperseFromSource, disperseFromLeaves, disperseUniformly, computeAverageBearing, avoidManyRobots, followTheLeader, orbitGroup, navigateGradient, clusterOnSource, and clusterIntoGroups. A swarm of 108 robots used the developed dispersion algorithms in an empty schoolhouse of area of about 300 m^2, and were able to locate an object of interest and lead a human to its location [53.129].

The European Union has sponsored many swarm robot projects, leading toward decreasingly smaller-sized individual robots, and increasingly larger-sized robot teams. The I-SWARM project, for instance, aimed at developing millimeter-sized robots with full onboard sensing, computation, and power for performing biologically inspired swarming behaviors, as well as collective perception tasks. This project was both a hardware and a software challenge, in that developing microscale robots that are fully autonomous and can perform meaningful cooperative behaviors will require significant advances in the current state of the art.

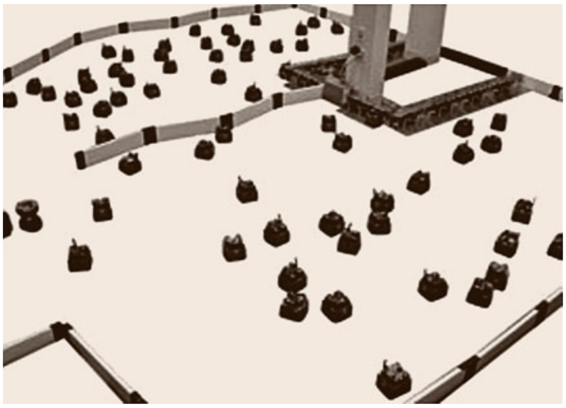

Fig. 53.12 The SwarmBot robots

The EU-based SWARM-BOTS project studied new concepts in the design and implementation of self-organizing and self-assembling robots (VIDEO 195). In this work [53.142], *s-bot* robots are developed that have grippers enabling the robots to create physical links with other s-bots or objects, thus creating assemblies of robots. These assemblies can then work together for navigation across rough terrain, or to collectively transport objects. The s-bots are cylindrical, with a flexible arm and toothed gripper that can connect one s-bot to another. An interesting application of object transport using SWARM-BOTs is shown in Figure 53.13, which shows the robots self-assembling into four chains in order to pull a child across the floor (VIDEO 212).

Another notable effort in swarm robotics research is the US multi-university SWARMS initiative led by the University of Pennsylvania. Research in this project aimed at developing a new system-theoretic frame-

Fig. 53.13 SWARM-BOTs self-assembling to move a child across the floor

work for swarming, developing models of swarms and swarming behavior, analyzing swarm formation, stability, and robustness, synthesizing emergent behaviors for active perception and coverage, and developing algorithms for distributed localization.

Besides the hardware challenges of dealing with large numbers of small robots, there are many important software challenges that remain to be solved. From a practical perspective, a common approach to creating homogeneous multirobot swarms is to hypothesize a possible local control law (or laws), and then study the resulting group behavior, iterating until the desired global behavior is obtained. However, the longer-term objective is to be able to both predict group performance based on known local control laws, and to generate local control laws based upon a desired global group behavior.

More recent research has focused on the development of analytical techniques that can synthesize distributed controllers that achieve the desired macro-level system behaviors. *Mather* and *Hsieh* [53.143] address this challenge by proposing a technique that first identifies robot–robot interactions at the macroscopic level; they then use this analysis to improve local robot control policies by filtering out spurious robot–robot interactions. Another top-down design approach is presented by *Chen* et al. [53.144] who show how to automatically synthesize control and communication strategies for a robot team based on global specifications of the desired system-level behavior, stated using regular expressions. The resulting control strategies are formally proven to correctly achieve the desired global behavior. The work of *Melo* and *Veloso* [53.145] takes a different approach to the synthesis of local decision policies by making use of the decentralized sparse-interaction Markov decision process. This technique allows agents to recognize states when interactions with other robots might occur, thus enabling them to choose better motions based on possible future inter-robot actions. *Tsiotras* and *Castro* [53.146] synthesize local controllers by generalizing the standard consensus algorithm, applied to geometric pattern formation.

A common theme in most of the works cited above is that they first make use of formal methods to describe the desired macro-level behavior, and then show how to use this macro-level goal to synthesize individual robot controllers. Another use of formal methods is to show how the individual goals of robot team members can be considered collectively, with the objective of maximizing the system's achievement of individual goals. Toward this end, game-theoretic techniques have been shown useful in a variety of distributed robot formulations. *Cheng* and *Dasgupta* [53.147] make use of the game-theoretic technique called Weighted Voting Games to address the problem of multi-robot team formation control amidst obstacles. *Taheri* et al. [53.148] also make use of game-theoretic principles, building upon the Local Interaction Game diffusion model to investigate how a small number of agents can influence the global society's behavior through local interactions.

An interesting question in the design of distributed robot coordination mechanisms is the extent to which identical controllers can lead to diversity, specialization, or changes in robot behavior. *Hsieh* et al. [53.149] study the emergence of specialization in robot swarms by making use of a distributed adaptation algorithm. They present a top-down analytical approach that defines the system equilibrium using waiting time parameters, and then present adaptive optimization strategies that converge to the optimal configurations that achieve system equilibrium. Temporal changes in system-level swarm behavior are addressed by *Hoff* et al. [53.150] who show how a swarm can change and improve its foraging behavior by switching between algorithms based on the environment in which the swarm finds itself.

Once the distributed controller is synthesized, most of the works mentioned above presume that individual robots execute their controller successfully. The typical presumption is that large swarms of interchangeable robots automatically result in robust and scalable swarm behavior. However, this presumption is challenged by *Winfield* and *Nembrini* [53.151], who illustrate that overall swarm reliability quickly falls in the presence of worst-case, partially failed robots. They conclude that future large scale swarm systems must develop new approaches for achieving high levels of fault tolerance. One example approach is shown in VIDEO 194.

53.6 Modular Robotics

The modular robotics field began with a paper presented at International Conference on Robotics and Automation in the Spring of 1988 by *Fukuda* and *Nakagawa* [53.152] describing the abstract concept of a reconfigurable robotic system that can assume different shapes and envisioned a robot system composed of different types of modules that can combine to accomplish a variety of tasks. Over the past twenty years, modular robotics research developed many facets: hardware design; planning and control algorithms; the trade-off between hardware and algorithmic complexity; efficient simulation; and system integration.

Modular robots are collections of physically connected, electromechanically active modules that, as a whole, form robotic systems that exhibit capabilities greater than those of the individual modules. Typically modular robots can change their shape or configuration in order to adapt to a variety of different tasks. For example, a collection of modules could reconfigure from a closed chain that rolls quickly over open ground to a legged robot that more easily traverses rough terrain. Modular robots are typically touted for their adaptability, their fault-tolerance, and the relative simplicity of the unit modules. Modular robotic systems can be described and classified on several axes using a variety of properties. In what follows, we choose the traditional route of classifying modular robotic systems by the geometry of the system: chain, lattice, truss, or free form. For a more detailed history of the modular robotics field, consult [53.153, 154].

53.6.1 Chain Systems

The defining characteristic of chain-type modular robot systems is the fact that the modules, when connected to their neighbors, are arranged in a chain. These chains may be one-dimensional, or two-dimensional, but three-dimensional chains are not as common. The fact that a chain-type modular robot is two-dimensional, or even one-dimensional, does not mean that it cannot operate in three dimensions. In fact, snake-like modular robots composed of segments with orthogonal joints are quite common.

One of the first chain-type modular robotic systems was the polypod system developed by *Yim* [53.155, 156] (VIDEO 196). The polypod system was composed of two types of modules: segments and nodes. It could form a variety of shapes including rolling loops and hexapods, and it went on to inspire many other chain-based systems. One was the CONRO system [53.157–159] in which each module was composed of two orthogonal servo motors controlling each module's pitch and yaw.

Murata et al. developed the M-TRAN modular robotic system [53.160–163] which has undergone multiple revisions and improvements. In [53.161], Kamimura et al. employ a set of interconnected, out of phase oscillators to achieve walking gaits in the M-TRAN system. *Marbach* and *Ijspeert* improved upon the ability of systems like M-TRAN to generate gaits in real-time by applying function optimization to their modular system, YaMoR [53.164]. *Murata* et al. added cameras to the M-TRAN system so that a set of M-TRAN modules could separate, perform independent tasks, and then rejoin into a larger structure [53.163].

The ATRON system [53.165, 166] was developed to improve upon the M-TRAN. Lund et al. wanted to keep M-TRAN's ability to form dense lattices while taking advantage of the two orthogonal degrees of freedom, (pitch and yaw), found in the CONRO system. The Superbot system [53.167] also builds upon on the mechanical design of M-TRAN by adding an additional degree of rotational freedom between the two existing rotation axes.

The PolyBot is chain-type modular robot [53.168, 169] with a single rotational degree of freedom. PolyBot evolved into CKBot which has demonstrated the ability to reassemble itself after being accidentally or intentionally destroyed [53.170]. The Molecube system [53.171], developed by *Lipson* et al. is another example of a chain-type modular system with only one degree of freedom but still able to achieve interesting three-dimensional (3-D) configurations. Lipson et al. showed that a short chain of Molecube modules, along with some free modules, can self-replicate.

Yim et al. designed another unique chain-type system named RATChET [53.172] which uses a connected chain of inter-latching right angle tetrahedrons to form structures. Neighboring RATChET modules latch together when the angle between them passes some critical value, and they unlatch through the use of shape memory alloy (SMA) springs when heated beyond 70 °C. Interestingly, the RATChET modules possess no intelligence. Instead, they rely on an intelligent external actuator which rotates to control one end of the dangling chain. One unique property of the RATChET system is its relatively strength.

53.6.2 Lattice Systems

Lattice-type modular robot systems are collections of interconnected robotic modules in which the units are situated at the intersection points of a two or three dimensional grid. (A 1-D (one-dimensional) lattice system is simply a chain-type robot.) The main characteristic separating a lattice system from a densely configured chain-type robot is the density of the interconnections between the modules. In a lattice-type system, each module is typically connected to all of its neighbors. In a dense chain-type system, two modules may be neighbors, but they will not be physically connected.

Additionally, lattice-type systems tend to be built with modules that contain no rotational degrees of freedom. While the modules in a lattice system typically have mechanisms which enable the modules to move relative to, and bond with, their neighbors, they generally cannot bend themselves. In comparison, chain-type systems are often built from modules that contain one or more rotational degrees of free so that the modules can flex like links in a chain. There is some overlap between between the two types of system.

Chirikjian et al. developed one of the first lattice-based modular robotic systems [53.173–175] (VIDEO 198) in which the modules are deformable hexagons capable of bonding with their neighbors. Others, such as *Walter* et al. [53.176] further analyzed these hexagonal type systems to create distributed motion planners capable of reconfiguring the system from one state to another.

Murata et al. were also early contributors to the development of lattice-based modular robotic systems with their development of a roughly hexagonal module capable of rolling around its neighbors in two dimensions [53.177, 178]. *Kurokawa* et al. presented a three dimensional adaptation [53.179] composed of cubes with six protruding arms capable of rotation. *Yoshida* et al. improved on this system with a new design that used shape-memory alloy actuators to rotate one robot module around the perimeter of a neighbor [53.180].

One of the simplest lattice systems is the the Digital Clay project [53.181]. The system was a set of completely passive modules that relied on the user to make changes to it topology. The 2.5 cm rhombic dodecahedrons were able to sense and communicate with their neighbors in order to create a virtual model of the physical arrangement of modules.

Rus et al. also explored the idea of 3-D modules capable reconfiguration through a series of latchings, rotations, and unlatchings with the Molecule system [53.182–185]. In [53.186, 187], *Vona* and *Rus* describe a different type of deformable lattice system. The Crystal system is composed of square modules able to expand and contract by a factor of two in the x–y plane. *Suh* et al. expanded on the Crystal concept with the Telecubes [53.188] that could move in three dimensions by expanding all six faces.

Chiang and *Chirikjian* analyzed how to perform motion planning in a lattice of rigid cubic modules able to slide past each other [53.189]. The CHOBIE robot developed by *Koseki* [53.190] is able to actually perform the sliding motion assumed by *Chiang* and *Chirikjian* in [53.189]. More recently, *An* developed the EM-Cube system [53.191] which is also capable of sliding motion.

Another unique lattice is the I-Cube developed by *Khosla* et al. [53.192, 193]. The 3-D I-Cube system consists of passive cubes which are connected by active links with three rotational degrees of freedom that are able to grab, reposition, and release the cubes. The 3-D I-Cube system was an improvement of the two-dimensional (2-D) system [53.194] developed by *Hosokawa* et al. for rearranging cubic modules in a vertical plane.

Goldstein et al. initiated the Claytronics project by publishing several papers [53.195, 196] proposing lattice-based *claytronic atoms* or catoms. These vertically-oriented cylindrical robots, which were incapable of independent motion, used 24 electromagnets around their perimeters to achieve rolling locomotion about their neighbors. Goldstein et al. envisioned a system in which millions of smaller catoms could form arbitrary shapes using a randomized algorithm that avoided conveying a complete description of the shape to each module in the system.

The catoms continue to evolve. One of the newest instantiations [53.197] employs hollow cylinders rolled from SiO_2 rectangles patterned with aluminum electrodes. The authors hope that two of these cylinders, when placed in close proximity with their axes aligned, will be able to rotate with respect to one another using electrostatic forces. Specifically, the electrodes, (which reside on the inside of each cylinder and are electrically isolated by the SiO_2), will be charged so that they attract and repel mirror charges on the neighboring cylinder in a way that causes rotation. Currently, the system appears to be constrained to form 2-D structures. The authors claim the completed system will have a yield strength similar to that of plastic and that the modules will be able to transfer power and communication signals capacitively from neighbor to neighbor.

The Claytronics project has proposed, but not yet demonstrated with hardware, the use of sub-millimeter intelligent particles as sensing and replication devices [53.198]. In particular, Pillai et al. present a theoretical 3-D fax machine in which the object to be *faxed* is immersed in a container of intelligent particles that sense and encode the object's dimensions. At the receiving end, these same Claytronic particles decode the shape description sent by the transmitter and bond together to replicate the original object. Unlike our approach, Pillai's approach is completely centralized and relies on an external computer for computation.

White et al. developed hardware and algorithms for several 2-D stochastically-driven self-assembling systems [53.199]. To form specific shapes, each module is provided with a representation of the desired shape and decides, based on its location in the structure, whether to allow other modules to bond to its faces. *Lipson* et al. extended their 2-D system to 3-D [53.200–202] by using cubic modules suspended in turbulent fluid to achieve self-assembly and reconfiguration. As the free modules circulate in the fluid, they pass by a growing structure of assembled modules. When they come close enough, they are accreted onto the structure. The modules attract or repel each other with fluid suction or positive pressure. Early versions of the system used modules with interval values that could redirect these suction forces. More recently, *Lipson*'s group has worked to move the intelligence and actuation capabilities from the modules to the tank in which the modules circulate [53.203].

The Miche system [53.204] consisting of 45 mm cubic modules capable of mating with their neighbors using mechanically switchable permanent magnets. Each module contains three switchable magnets, each of which mated with a steel face on a neighboring module. Because the connectors were gendered, any collection of modules had to be assembled by hand so that the connectors were always oriented correctly, but the system was capable of self-disassembling to form 3-D structures. The Robot Pebbles (VIDEO 211) are based, at least in principle, on the Miche modules.

One of the newest lattice-type modular robotics is an aerial system composed of identical, hexagonal, single-rotor modules [53.205]. A group of modules may connect to form a flying platform with an arbitrary arrangement of multiple rotors. In addition to the ability to fly, each module contains wheels so that the system may self-reconfigure on the ground for the specific task at hand.

53.6.3 Truss Systems

Truss systems, as their name implies, are modular robotic systems in which the modules are nodes and edges in a truss structure. Both the trusses and connectors may be active in such systems. Unlike the lattice-based systems, truss-based systems do not need to operate on any regular lattice. Most truss-based systems under development employ struts that expand or contract to achieve structural deformation. One of the first such system to do so was Tetrobot [53.206]. The Odin system, conceived by *Lyder* et al. [53.207, 208] consists of three physically different types of modules: active strut modules capable of changing their length; passive strut modules of fixed length; and joint modules. The biologically inspired Morpho system [53.209] developed by *Nagpal* et al. is similar to Odin. It also uses active links, passive links, and connector cubes.

53.6.4 Free-Form Systems

Free-form systems are able to aggregate modules in at least semi-arbitrary positions. One such system is the Slimebot [53.210, 211]. The system consists of identical vertical cylindrical modules that move on a horizontal plane. The perimeter of each module is covered by six gender-less hook and loop patches used to bond with neighboring modules. These patches oscillate radially in and out from the center of the body. By controlling the frequency and phase of the oscillations between neighbors, the system can achieve aggregate motion in a given direction.

Researchers are also developing algorithms for free-form systems. *Funiak* et al. developed a localization algorithm [53.212] that is capable of localizing tens-of-thousands of irregularly packed modules in 3-D. Rubenstein and Shen developed a number of shape formation algorithms for collections of two-dimensional modules. These algorithms allow an arbitrary-sized collection of modules to form arbitrary scale-independent shapes [53.213, 214]. Once the shape is formed, modules can be added to or removed from the system, and the system will reconfigure itself to incorporate the new modules. The resulting shape will grow or shrink, but its basic form will remain unchanged. Recently, *Rubenstein* et al. developed a 1000 modules hardware platform on which to deploy these algorithms [53.215].

Researchers have also explored the use of folding to create reconfigurable foldable systems [53.216, 217]. These systems use flexible wiring and shape memory alloy actuators embedded in composite sheets to programmatically create origami-inspired shapes. By controlling which actuators are energized, the system can form multiple different shapes.

53.6.5 Self-Assembling Systems

Self-assembling modular robotic systems are collection of modules that are capable of autonomously coalescing and bonding with their neighbors to form a greater structure. The result is often robotic, but it need not be. Whether a system is capable of self-assembling is independent of whether it is free-form, a chain, a lattice, or a truss-based system. Almost all of aforementioned modular robot systems rely on human intervention to assemble. In an attempt to automate the process of creating intricate modular robotic systems, researchers have attempted to mimic and improve upon natural self-assembling systems. *Whitesides* et al. investigated a wide variety of engineered self-assembling systems [53.218–220].

Miyashita et al. performed a more theoretical analysis of self-assembly using pie-shaped pieces to form complete circles [53.221] from pie-shaped pieces. In the process, they followed *Hosokawa* et al.'s lead [53.222] and modeled the system as a chemical reaction. *Shimizu* and *Suzuki* developed a system of passive modules capable of self-repair when placed on a vibrating table [53.223].

Computer scientists have also investigated theoretical aspects of self-assembly in the context of 2-D tiles which selectively bond with their neighbors to form simple well-defined shapes like squares [53.224–226]. Each side of every tile in the system has an associated bonding strength. When two tiles collide, they remain attached only if their cumulative bond strength exceeds a globally defined system entropy. To form a specific shape, one must design a set of tiles with the appropriate bonding strengths.

Klavins et al. worked to develop intelligent self-assembling systems that employ triangular modules driven by oscillating fans on an air table to self-assemble different shapes [53.227]. The authors employ knowledge of the module's local topology and internal module state so that each module decides, in a distributed fashion, when to maintain or break a connection with its immediate neighbors. *Griffith* et al. also worked with intelligent modules capable of selective bonding to show that self-assembling systems may self-replicate [53.228].

Rus et al. [53.185] present the first generic rule-based approach to self-assembly, shape formation, and locomotion by reconfiguration. The rules can be used on any modular robot system that can implement the sliding cube model of relative motion. The result is an abstract set of rules for each of these tasks, that can be compiled down to module motions, taking into account how the physical module implementes translation and convex and concave transitions.

Jones and *Matarić* [53.229] presented rule-based approach to self-assembly termed transition rule sets. In particular, they present a method that, given a goal structure, produces a set of rules shared among all modules that govern when and where new modules are allowed to attach to the growing structure. *Kelly* and *Zhang* [53.230] expanded on this work by optimizing the size of the rule sets used to form a specific shape. *Werfel* [53.231] also applied the idea of a transition rule set when studying the use of swarms to assemble complex structures from passive materials.

Other groups have attempted to make self-assembly more deterministic. The MEMS (microelectromechanical system) robots developed by *Donald* et al. [53.232, 233] consists of thin (7–20 μm), rectangular ($\approx 260\,\mu\text{m} \times 60\,\mu\text{m}$), scratch-drive devices capable of moving on an insulating substrate embedded with electrodes. The authors used four of these robots to build larger composite structures. The Sitti group has developed a similar system of micro-meter sized robots [53.234]. Instead of using a scratch drive for locomotion, the robots are manipulated by external magnetic fields. The authors can electrostatically clamp any number of robots to the stage on which they move. With all but one robot immobilized, the remaining robot may be moved independently. The system naturally self-assembles because the robots contain permanent magnets that attract their neighbors.

The majority of existing self-assembly systems aim to form structures in one of two ways. Some systems such as [53.221, 223–226] use a collection of application specific differentiated modules, that are only capable of assembling in a particular fashion to form a specific shape. In contrast, other systems such as [53.199–201, 227, 229–231, 235] use completely generic modules with more computation and communication ability embedded in each module. Both types of systems aim to form complex shapes in a direct manner: as these structures grow from a single module, new modules are only allowed to attach to the structure in specific locations.

An alternative approach eliminates many of the complexities of shape formation by active assembly by using dissasemly for shape formation. The Smart Pebble system [53.154, 204, 236, 237] employs a set of distributed algorithms to perform two discrete steps: 1) rely on stochastic forces to self-assemble a close-packed crystalline lattice of modules and 2) use the process of self-disassembly to remove the extra material from this block leaving behind the goal structure. By approaching shape formation in this manner, the entire shape-formation process is sped-up and the robustness of the system is increased.

53.7 Heterogeneity

Robot heterogeneity can be defined in terms of variety in robot behavior, morphology, performance quality, size, and cognition. In most large-scale multirobot systems work, the benefits of parallelism, redundancy, and solutions distributed in space and time are obtained through the use of homogeneous robots, which are completely interchangeable (i. e., the swarm approach, as described in Sect. 53.4). However, certain complex applications of large-scale robot teams may require the simultaneous use of multiple types of sensors and robots, all of which cannot be designed into a single type of robot. Some robots may need to be scaled to smaller sizes, which will limit their payloads, or certain required sensors may be too expensive to duplicate across all robots on the team. Other robots may need to be large to carry application-specific payload or sensors, or to navigate long distances in a limited time. These applications, therefore, require the collaboration of large numbers of heterogeneous robots.

The motivation for developing heterogeneity in multirobot teams is thus twofold: heterogeneity may be a design feature beneficial to particular applications, or heterogeneity may be a necessity. As a design feature, heterogeneity can offer economic benefits, since it can be easier to distribute varying capabilities across multiple team members rather than to build many copies of monolithic robots. Heterogeneity can also offer engineering benefits, as it may simply be too difficult to design individual robots that incorporate all of the sensing, computational, and effector requirements of a given application. Heterogeneity in behavior may also arise in an emergent manner in physically homogeneous teams, as a result of behavior specialization.

A second compelling reason to study heterogeneity is that it may be a necessity, in that it is nearly impossible in practice to build a truly homogeneous robot team. The realities of individual robot design, construction, and experience will inevitably cause a multirobot system to drift to heterogeneity over time. This is recognized by experienced roboticists, who have seen that several copies of the same model of robot can vary widely in capabilities due to differences in sensor tuning, calibration, etc. Over time, even minor initial differences among robots will grow due to individual robot drift and wear and tear. The implication is that, to employ robot teams effectively, we must understand diversity, predict how it will impact performance, and enable robots to adapt to the diverse capabilities of their peers. In fact, it is often advantageous to build diversity explicitly into the design of a robot team.

There are a variety of research challenges in heterogeneous multirobot systems. A particular challenge to achieving efficient autonomous control is when overlap in team member capabilities occurs, thus affecting task allocation or role assignments [53.238]. Techniques as described in Sect. 53.6 can typically deal with heterogeneous robots for the purposes of task allocation. Another important topic in heterogeneity is how to recognize and quantify heterogeneity in multirobot teams. Some types of heterogeneity can be evaluated quantitatively, using metrics such as the social entropy metric developed by *Balch* [53.239]. Most research in heterogeneous multirobot systems assumes that robots have a common language and a common understanding of symbols in their language; developing a common understanding of communicated symbols among robots with different physical capabilities is a fundamental challenge, addressed by *Jung* and *Zelinsky* [53.240] (VIDEO 200).

As discussed in Sect. 53.2, one of the earliest research demonstrations of heterogeneity in physical robot teams was in the development of the ALLIANCE architecture by *Parker* [53.9]. This work demonstrated the ability of robots to compensate for heterogeneity in team members during task allocation and execution. *Murphy* has studied heterogeneity in the context of marsupial robot deployment, where a *mothership* robot assists smaller robots in applications such as search and rescue [53.241] (VIDEO 206). *Grabowski* et al. [53.134] developed modular millibots for surveillance and reconnaissance that could be composed of interchangeable sensor and effector components, thus creating a variety of different heterogeneous teams. *Simmons* et al. [53.11] demonstrated the use of heterogeneous robots for autonomous assembly and construction tasks relevant to space applications. *Sukhatme* et al. [53.242] demonstrated a helicopter robot cooperating with two ground robots in tasks involving marsupial-inspired payload deployment and recovery, cooperative localization, and reconnaissance and surveillance tasks, as shown in Fig. 53.14. *Parker* et al. [53.243] demonstrated assistive navigation for sensor network deployment using a more intelligent leader robot for guiding navigationally challenged simple sensor robots to goal locations, as part of a larger demonstration by *Howard* et al. [53.244] of 100 robots performing exploration, mapping, deployment, and detection. *Chaimowicz* et al. [53.245] demonstrated a team of aerial and ground robots cooperating for surveillance applications in urban environments. *Parker* and *Tang* [53.246] developed ASyMTRe (automated

Fig. 53.14 Heterogeneous team of an air and two ground vehicles that can perform cooperative reconnaissance and surveillance

synthesis of multirobot task solutions through software reconfiguration), which enables heterogeneous robots to share sensory resources to enable the team to accomplish tasks that would be impossible without tightly coupled sensor sharing.

Many open research issues remain to be solved in heterogeneous multirobot teams; for example, the issue of optimal team design is a very challenging problem. Clearly, the required behavioral performance in a given application dictates certain constraints on the physical design of the robot team members. However, it is also clear that multiple choices may be made in designing a solution to a given application, based upon cost, robot availability, ease of software design, flexibility in robot use, and so forth. Designing an optimal robot team for a given application requires significant analysis and consideration of the tradeoffs in alternative strategies.

53.8 Task Allocation

In many multirobot applications, the mission of the team is defined as a set of tasks that must be completed. Each task can usually be worked on by a variety of different robots; conversely, each robot can usually work on a variety of different tasks. In many applications, a task is decomposed into independent subtasks [53.9], hierarchical task trees [53.247], or roles [53.11, 245, 248, 249] either by a general autonomous planner or by the human designer. Independent subtasks or roles can be achieved concurrently, while subtasks in task trees are achieved according to their interdependence. Once the set of tasks or subtasks have been identified, the challenge is to determine the preferred mapping of robots to tasks (or subtasks). This is the *task allocation* problem.

The details of the task allocation problem can vary in many dimensions, such as the number of robots required per task, the number of tasks a robot can work on at a time, the coordination dependencies among tasks, and the time frame for which task assignments are determined. *Gerkey* and *Matarić* [53.250] defined a taxonomy for task allocation that provides a way of distinguishing task allocation problems along these dimensions, which is referred to as the multirobot task allocation (MRTA) taxonomy.

53.8.1 Taxonomy for Task Allocation

Generally, tasks are considered to be of two principal types: *single-robot tasks* (SR, according to the MRTA taxonomy) are those that require only one robot at a time, while *multirobot tasks* (MR) are those that require more than one robot working on the same task at the same time. Commonly, single-robot tasks that have minimal task interdependencies are referred to as *loosely coupled tasks*, representing a *weakly cooperative* solution. On the other hand, multirobot tasks are often considered to be sets of subtasks that have strong interdependencies. These tasks are therefore often referred to as *tightly coupled tasks* that require a *strongly cooperative* solution. The subtasks of a loosely coupled multirobot task require a high level of synchronization or coordination between subtasks, meaning that each task must be aware of the current state of the coordinated subtasks within a small time delay. As this time delay becomes progressively larger, coordinated subtasks become more loosely coupled, representing weakly cooperative solutions.

Robots can also be categorized as either *single-task robots* (ST), which work on only one task at a time or *multitask robots* (MT), which are able to make progress on more than one task at a time. Most commonly, task allocation problems assume robots are single-task robots, since more capable robots that perform multiple tasks in parallel are still beyond the current state of the art.

Tasks can either be assigned to optimize the instantaneous allocation of tasks (IA), or to optimize the assignments into the future (TA, for time-extended assignment). In the case of instantaneous assignment, no consideration is made for the effect of the current assignment on future assignments. Time-extended assignments attempt to assign tasks so that the performance of the team is optimized for the entire set of tasks that may be required, not just the current set of tasks that need to be achieved at the current time step.

Using the MRTA taxonomy, triples of these abbreviations are used to categorize various task allocation approaches, such as SR-ST-IA, which refers to an assignment problem in which single-robot tasks are assigned once to single-task robots. Different variations of the task allocation problem have different computational complexities. The easiest variant is the ST-SR-IA problem, which can be solved in polynomial time since it is an instance of the optimal assignment problem [53.251]. Other variants are much more difficult, and do not have known polynomial time solutions. For example, the ST-MR-IA variant can be shown to be an instance of the set partitioning problem [53.252], which is strongly *NP*-hard. The ST-MR-TA, MT-SR-IA, and MT-SR-TA variants have also all been shown to be *NP*-hard problems. Because these problems are computationally complex, most approaches to task allocation in multirobot teams generate approximate solutions.

53.8.2 Representative Approaches

Approaches to task allocation in multirobot teams can be roughly divided into behavior-based approaches and market-based (sometimes called negotiation-style or auction-based) approaches. The following subsections describe some representative architectures for each of these general approaches. Refer to [53.250] for a comparative analysis of some of these approaches, in terms of computation and communications requirements and solution quality.

Behavior-Based Task Allocation

Behavior-based approaches typically enable robots to determine task assignments without explicitly discussing individual tasks. In these approaches, robots use knowledge of the current state of the robot team mission, robot team member capabilities, and robot actions to decide, in a distributed fashion, which robot should perform which task.

One of the earliest architectures for multirobot task allocation that was demonstrated on physical robots was the behavior-based ALLIANCE architecture [53.9] and the related L-ALLIANCE architecture [53.10]. ALLIANCE addresses the ST-SR-IA and ST-SR-TA variants of the task allocation problem without explicit communication among robots about tasks. As described in Sect. 53.2.2, ALLIANCE achieves adaptive action selection through the use of motivational behaviors, which are levels of impatience and acquiescence within each robot that determine its own and its teammates' relative fitness for performing certain tasks. These motivations are calculated based upon the mission requirements, the activities and capabilities of teammates, and the robots' internal states. These motivations effectively calculate utility measures for each robot–task pair.

Another behavior-based approach to multirobot task allocation is broadcast of local eligibility (BLE) [53.253], which addresses the ST-SR-IA variant of task allocation. BLE uses a subsumption style behavior control architecture [53.254] that allows robots to efficiently execute tasks by continuously broadcasting locally computed eligibilities and only selecting the robot with the best eligibility to perform the task. In this case, task allocation is achieved through behavior inhibition. BLE uses an assignment algorithm that is very similar to *Botelho* and *Alami*'s M+ architecture [53.255].

Market-Based Task Allocation

Market-based (or negotiation-based) approaches typically involve explicit communications between robots about the required tasks, in which robots bid for tasks based on their capabilities and availability. The negotiation process is based on market theory, in which the team seeks to optimize an objective function based upon individual robot utilities for performing particular tasks. The approaches typically greedily assign subtasks to the robot that can perform the task with the highest utility.

Smith's contract net protocol (CNP) [53.256] was the first to address the problem of how agents can negotiate to collectively solve a set of tasks. The use of a market-based approach specifically for multirobot task allocation was first developed by *Botelho* and *Alami* with their M+ architecture [53.255]. In the M+ approach, robots plan their own individual plans for the task they have been assigned. They then negotiate with other teammates to incrementally adapt their actions to suit the team as a whole, through the use of social rules that facilitate the merging of plans.

Since these early developments, many alternative approaches to market-based task allocation have been developed. A thorough survey on the current state of the art in market-based techniques for multirobot task allocation is given in [53.257], comparing alternative approaches in terms of solution quality, scalability, dynamic events and environments, and heterogeneous teams.

Most of the current approaches in market-based task allocation address the ST-SR problem variant, with some approaches (e.g., [53.11, 258–260]) dealing with instantaneous assignment (IA), and others (e.g., [53.135, 261–263]) addressing time-extended assignments (TA). More recent methods are beginning to address the coaltion formation problem, which is the allocation of multirobot tasks (i. e., the MR-ST problem variant), including [53.246, 264–269]. An example approach to the MR-MT problem variant is found in [53.270].

Some representative market-based techniques include MURDOCH [53.258], TraderBots [53.247, 263], and Hoplites [53.265]. The MURDOCH approach [53.258] employs a resource-centric, publish–subscribe communication model to carry out auctions, which has the advantage of anonymous communication. In this approach a task is represented by the required resources, such as the environmental sensors. The methods for how to use such a sensor to generate satisfactory results is preprogrammed into the robot.

The TraderBots approach [53.247, 263] applies market economy techniques for generating efficient and robust multirobot coordination in dynamic environments. In a market economy, robots act based on selfish interests. A robot receives revenue and incurs cost when trying to accomplish a task. The goal is for robots to trade tasks through auctions/negotiations such that the team profit (revenue minus cost) is optimized.

The Hoplites approach [53.265] focuses on the selection of an appropriate joint plan for the team to execute by incorporating joint revenue and cost into the bid. This approach couples planning with passive and active coordination strategies, enabling robots to change coordination strategies as the needs of the task change. Strategies are predefined for a robot to accomplish a selected plan.

Some alternative approaches formulate the objects to be assigned as *roles*, which typically package a set of tasks and/or behaviors that a robot should undertake when acting in a particular role. Roles can then be dynamically assigned to robots in a similar manner as in the auction-based approaches [53.11, 248].

53.9 Learning

Multirobot learning is the problem of learning new cooperative behaviors, or learning in the presence of other robots. The other robots in the environment, however, have their own goals and may be learning in parallel [53.271]. The challenge is that having other robots in the environment violates the Markov property that is a fundamental assumption of single-robot learning approaches [53.271]. The multirobot learning problem is particularly challenging because it combines the difficulties of single-robot learning with multiagent learning. Particular difficulties that must be considered in multirobot learning include continuous state and action spaces, exponential state spaces, distributed credit assignment, limited training time and insufficient training data, uncertainty in sensing and shared information, nondeterministic actions, difficulty in defining appropriate abstractions for learned information, and difficulty of merging information learned from different robot experiences.

The types of applications that have been studied for multirobot learning include multitarget observation [53.272, 273], air fleet control [53.274], predator–prey [53.137, 275, 276], box pushing [53.277], foraging [53.23], and multirobot soccer [53.140, 278]. Particularly challenging domains for multirobot learning are those tasks that are *inherently* cooperative. Inherently cooperative tasks are those that cannot be decomposed into independent subtasks to be solved by individual robots. Instead, the utility of the action of one robot is dependent upon the current actions of the other team members. This type of task is a particular challenge in multirobot learning, due to the difficulty of assigning credit for the individual actions of the robot team members.

The credit assignment problem is a particular challenge, since it is difficult for a robot to determine whether the fitness (either good or bad) is due to its own actions, or due to the actions of another robot. As discussed by *Pugh* and *Martinoli* in [53.279], this problem can be especially difficult in situations where robots do not explicitly share their intentions. Two different variations of the credit assignment problem are common in multirobot learning. The first is when robots are learning individual behaviors in the presence of other robots that can affect their performance. The second is when robots are attempting to learn a task with a shared fitness function. It can be difficult to determine how to decompose the fitness function to appropriately reward or penalize the contributions of individual robots.

While learning has been explored extensively in the area of single-robot systems (see, for example, the discussion of learning in behavior-based systems in Chap. 13, and a discussion of fundamental learning techniques in Chap. 15) and in multi*agent* systems [53.280], much less work has been done in the area of multi*robot* learning, although the topic is gaining increased interest. Much of the work to date has focused on reinforcement learning approaches. Some examples of this multirobot learning research include the work by *Asada* et al. [53.281], who propose a method for learning new behaviors by coordinating previously learned behaviors using Q-learning, and apply it to soccer-playing robots. *Matarić* [53.8] introduces a method for combining basic behaviors into higher-level behaviors through the use of unsupervised reinforcement learning, heterogeneous reward functions, and progress estimators. This mechanism was applied to a team of robots learning to perform a foraging task. *Kubo* and *Kakazu* [53.282] proposed another reinforcement learning mechanism that uses a progress value for determining reinforcement, and applied it to simulated ant colonies competing for food. *Fernandez* and *Parker* [53.272] apply a reinforcement learning algorithm that combines supervised function approximation with generalization methods based on state-space discretization, and apply it to robots learning the multi-object tracking problem. *Bowling* and *Veloso* [53.271] developed a general-purpose, scalable learning algorithm called GraWoLF (gradient-based win or learn fast), which combines gradient-based policy learning techniques with a variable learning rate, and demonstrated the results in the adversarial multirobot soccer application.

Other multirobot learning approaches not based on reinforcement include *Parker*'s L-ALLIANCE architecture [53.10], which uses parameter tuning, based on statistical experience data, to learn the fitness of different heterogeneous robots in performing a set of tasks. *Pugh* and *Martinoli* [53.279] apply particle swarm optimization techniques to distributed unsupervised robot learning in groups, for the task of learning obstacle avoidance.

53.10 Applications

Many real-world applications can potentially benefit from the use of multiple mobile robot systems. Example applications include container management in ports [53.283], extraplanetary exploration [53.284], search and rescue [53.241], mineral mining, transportation, industrial and household maintenance, construction [53.11], hazardous waste cleanup [53.9], security [53.285, 286], agriculture, and warehouse management [53.287] (VIDEO 210). Multiple robot systems are also used in the domain of localization, mapping, and exploration; Chap. 46 mentions some of the work in multirobot systems applied to these problems. Parts F and G of this handbook outline many application areas that are relevant not only to single-robot systems, but also to multiple mobile robot systems. To date, relatively few real-world implementations of these multirobot systems have occurred, primarily due to the complexities of multiple robot systems and the relative newness of the supporting technologies. Nevertheless, many proof-of-principle demonstrations of physical multirobot systems have been achieved, and the expectation is that these systems will find their way into practical implementations as the technology continues to mature.

Research in multiple mobile robot systems is often explored in the context of common application test domains. While not yet elevated to the level of benchmark tasks, these common domains do provide opportunities for researchers to compare and contrast alternative strategies to multirobot control. Additionally, even though these common test domains are usually just laboratory experiments, they do have relevance to real-world applications. This section outlines these common application domains; see also [53.2] and [53.288] for a discussion of these domains and a more detailed listing of related research.

53.10.1 Foraging and Coverage

Foraging is a popular testing application for multirobot systems, particularly for those approaches that address swarm robotics, involving very large numbers of mobile robots. In the foraging domain, objects such as pucks or simulated food pellets are distributed across the planar terrain, and robots are tasked with collecting the objects and delivering them to one or more gathering locations, such as a home base. Foraging lends itself to the study of weakly cooperative robot systems, in that the actions of individual robots do not have to be tightly synchronized with each other. This task has traditionally been of interest in multirobot systems because of its close analogy to the biological systems that motivate swarm robotics research. However, it also has relevance to several real-world applications, such as toxic waste cleanup, search and rescue, and demining. Additionally, since foraging usually requires robots to completely explore their terrain in order to discover the objects of interest, the *coverage* domain has similar issues to the foraging application. In coverage, robots are required to visit all areas of their environment, perhaps searching for objects (such as landmines) or executing some action in all parts of the environment (e.g., for floor cleaning). The coverage application also has real-world relevance to tasks such as demining, lawn care, environmental mapping, and agriculture.

In foraging and coverage applications, a fundamental question is how to enable the robots to explore their environments quickly without duplicating actions or interfering with each other. Alternative strategies can include basic stigmergy [53.14], forming chains [53.120], and making use of heterogeneous robots [53.131]. Other research demonstrated in the foraging and/or coverage domain includes [53.23, 132, 289–294].

53.10.2 Flocking and Formations

Coordinating the motions of robots relative to each other has been a topic of interest in multiple mobile robot systems since the inception of the field. In particular, much attention has been paid to the flocking and formation control problems (VIDEO 217, VIDEO 293). The flocking problem could be viewed as a subcase of the formation control problem, requiring robots to move together along some path in the aggregate, but with only minimal requirements for paths taken by specific robots. Formations are stricter, requiring robots to maintain certain relative positions as they move through the environment. In these problems, robots are assumed to have only minimal sensing, computation, effector, and communications capabilities. A key question in both flocking and formation control research is determining the design of local control laws for each robot that generate the desired emergent collective behavior. Other issues include how robots cooperatively localize themselves to achieve formation control [53.133, 295], and how paths can be planned for permutation-invariant multirobot formations [53.296].

Early solutions to the flocking problem in artificial agents were generated by *Reynolds* [53.297] using a rule-based approach. Similar behavior- or rule-based approaches have been used physical robot demonstrations and studies, such as in [53.121, 298]. These earlier solutions were based on human-generated local control rules that were demonstrated to work in practice. More recent work is based on control theoretic principles, with a focus on proving stability and convergence properties in multirobot team behaviors. Examples of this work include [53.128, 299–307]. Refer to [53.308, 309] for surveys of relevant control theoretic work.

53.10.3 Object Transportation and Cooperative Manipulation

Some of the earliest work in swarm robotics was aimed at the object transportation task [53.13, 123, 310–313], which requires a team of robots to move an object from its current position in the environment to some goal destination (VIDEO 193). The primary benefit of using collective robots for this task is that the individual robots can combine forces to move objects that are too heavy for individual robots working alone or in small teams. However, the task is not without its challenges; it is non-trivial to design decentralized robot control algorithms that can effectively coordinate robot team members during object transportation. A further complication is that the interaction dynamics of the robots with the object can be sensitive to certain object geometries [53.314, 315] and object rotations during transportation [53.315], thus exacerbating the control problem.

Object transportation and cooperative manipulation are popular domains for demonstrating multirobot cooperation, because they offer a clear domain where close coordination and cooperation is required. A common type of object transportation – box pushing – requires robot teams to move boxes from their starting positions to defined goal configurations, sometimes along specified paths. Typically, box pushing operates in the plane, and the assumption is made that the boxes are too heavy or too long to enable single robots to push alone. Sometimes there are several boxes to be moved, with ordering dependencies constraining the sequence of motions. Cooperative manipulation is similar, except it requires robots to lift and carry objects to a destination. This test bed domain lends itself to the study of strongly cooperative multirobot strategies, since robots often have to synchronize their actions to successfully execute these tasks. The domain of box pushing and cooperative manipulation is also popular because it has relevance to several real-world applications [53.288], including warehouse stocking, truck loading and unloading, transporting large objects in industrial environments, and assembly of large-scale structures.

Researchers usually emphasize different aspects of their cooperative control approach in the box pushing and cooperative manipulation domain. For example, *Kube* and *Zhang* [53.13] (VIDEO 199) demonstrate how swarm-type cooperative control techniques could achieve box pushing (Fig. 53.15), *Parker* [53.10, 316] illustrates aspects of adaptive task allocation and learning, *Donald* et al. [53.317] (VIDEO 208) illustrates concepts of information invariance and the interchangeability of sensing, communication, and control, and *Simmons* et al. [53.11] demonstrate the feasibility of cooperative control for building planetary habitats.

In general, the manipulation techniques used for collective object transportation can be grouped into three primary methods [53.318]: pushing, grasping, and caging. The pushing approach [53.10, 11, 13, 316, 317] requires contact between each robot and the object, in order to impart force in the goal direction; however, the robots are not physically connected with the object. In the grasping approach [53.123, 142, 310–312, 319–322], each robot in the team physically attaches to the object being transported. See for example Fig. 53.16

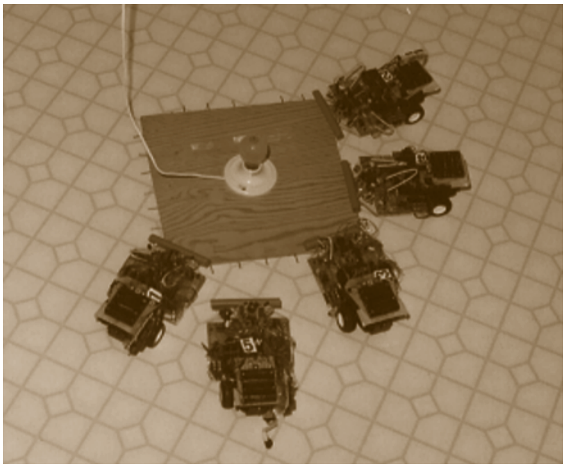

Fig. 53.15 Collective pushing of lighted box

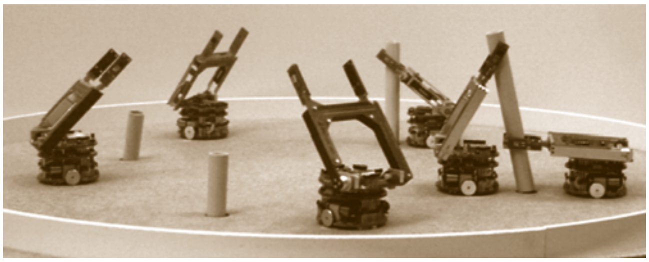

Fig. 53.16 Cooperative stick pulling

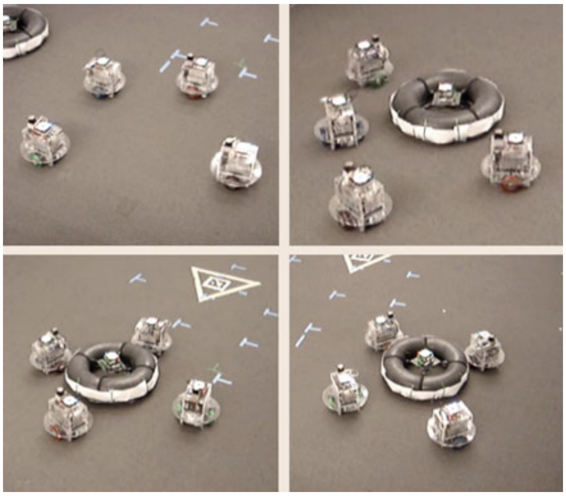

Fig. 53.17 Collective transport via caging

for cooperative stick pulling work of [53.320]. Finally, the caging approach [53.323–326] (VIDEO 292) involves robots encircling the object so that the object moves in the desired direction, even without the constant contact of all the robots with the object. See Fig. 53.17 for an example of collective transport via caging, from [53.323].

A closely related task is that of collective construction and wall building. The objective of the collective construction and wall building task is for robots to build structures of a specified form, in either 2-D or 3-D. This task is distinguished from self-reconfigurable robots, whose bodies themselves serve as the dynamic structure. *Werfel* and *Nagpal* have extensively explored this topic (VIDEO 216), developing distributed algorithms that enable simplified robots to build structures based on provided blueprints, both in 2-D [53.327–329] and in 3-D [53.330]. In their 3-D approach, the system consists of idealized mobile robots that perform the construction, and smart blocks that serve as the passive structure. The robots' job is to provide the mobility, while the blocks' role is to identify places in the growing structure at which an additional block can be placed that is on the path toward obtaining the desired final structure. The goal of their work is to be able to deploy some number of robots and free blocks into a construction zone, along with a single block that serves as a seed for the structure, and then have the construction to proceed autonomously according to the provided blueprint of the desired structure.

Hardware challenges of collective robot construction are addressed by *Terada* and *Murata* [53.331]. In this work, a hardware design is proposed that defines passive building blocks, along with an assembler robot that constructs structures with the robots. Other related work on the topic of collective construction includes the work of *Wawerla* et al. [53.332], in which robots use a behavior-based approach to build a linear wall using blocks equipped with either positive or negative Velcro, distinguished by block color. Their results show that adding 1 bit of state information to communicate the color of the last attached block provides a significant improvement in the collective performance. The work by *Stewart* and *Russell* [53.333, 334] proposes a distributed approach to building a loose wall structure with a robot swarm.

Another type of construction is called blind bulldozing, which is inspired by a behavior observed in certain ant colonies. Rather than constructing by accumulating materials, this approach achieves construction by removing materials. This task has practical application in site clearing, such as would be needed for planetary exploration [53.335]. Early ideas of this concept were discussed by *Brooks* et al. in [53.336], which argues for large numbers of small robots to be delivered to the lunar surface for site preparation. *Parker* et al. [53.337], further develop this idea by proposing robots using force sensors to clear an area by pushing material to the edges of the work site.

A significant body of additional research has been illustrated in this domain; representative examples include [53.3, 6, 123, 258, 284, 310, 323, 338–344].

53.10.4 Multitarget Observation

The domain of multitarget observation requires multiple robots to monitor and/or observe multiple targets moving through the environment. The objective is to maximize the amount of time, or the likelihood, that the targets remain in view by some team member throughout task execution. The task can be especially challenging if there are more targets than robots. This application domain can be useful for studying strongly cooperative task solutions, since robots have to coordinate their motions or the switching of targets to follow in order to maximize their objective. In the context of multiple mobile robot applications, the planar version of this test bed was first introduced in [53.345] as cooperative multirobot observation of multiple moving targets (CMOMMT). Similar problems have been studied by several researchers, and extended to more complex problems such as environments with complex topography or three-dimensional versions for multiple aerial vehicle applications. This domain is also related to problems in other areas, such as art gallery algorithms, pursuit evasion, and sensor coverage. This domain has practical application in many security, surveillance, and reconnaissance problems. Research applied to the multitarget observation problem in multirobot systems includes [53.138, 253, 346–353].

53.10.5 Traffic Control and Multirobot Path Planning

When multiple robots are operating in a shared environment, they must coordinate their actions to prevent interference. These problems typically arise when the space in which robots operate contains bottlenecks, such as networks of roadways, or when the robots take up a relatively large portion of the navigable space. In these problems, the open space can be viewed as a resource that robots must share as efficiently as possible, avoiding collisions and deadlocks. In this domain, robots usually have their own individual goals, and must work with other robots to ensure that they receive use of the shared space to the extent needed to achieve their goals. In some variants, the entire paths of multiple robots need to be coordinated with each other; in other variants, robots must simply avoid interfering with each other.

A variety of techniques have been introduced to address this problem, including traffic rules, subdividing the environment into single-ownership sections, and geometric path planning ([53.354] for an overview). Many of the earliest research approaches to this problem were based on heuristic approaches, such as predefining motion control (or traffic) rules that were shown to prevent deadlock [53.355–358], or using techniques similar to mutual exclusion in distributed computing [53.359, 360]. These approaches have the benefit of minimizing the planning cost for obtaining a solution. Other, more formal, techniques view the application as a geometric multirobot path planning problem that can be solved precisely in configuration space–time. Chapter 7 includes a discussion of motion planning for multiple robots relevant to this domain. While geometric motion planning approaches provide the most general solutions, they can often be too computationally intensive for practical application, impractical due to the dynamic nature of the environment, or simply unnecessary

Fig. 53.18 Legged robot teams competing in robot soccer

for the problem at hand. In these cases, approximation approaches may be sufficient, such as centralized techniques that limit the search space through roadmapping [53.361, 362], and decoupled approaches that use either prioritized planning [53.363–365] (i. e., generating robot paths one by one) or path coordination (i. e., first planning individual paths for robots, then handling collision avoidance).

53.10.6 Soccer

Since the inception of the RoboCup multirobot soccer domain as a proposed challenge problem for studying coordination and control in multirobot systems [53.366], research in this domain has grown tremendously. This domain incorporates many challenging aspects of multirobot control, including collaboration, robot control architectures, strategy acquisition, real-time reasoning and action, sensor fusion, dealing with adversarial environments, cognitive modeling, and learning. Annual competitions show the ever-improving team capabilities of the robots in a variety of settings, as shown in Fig. 53.18. A key aspect of this domain that is not present in the other multirobot test domains is that robots must operate in *adversarial* environments. This domain is also popular because of its educational benefits, as it brings together students and researchers from across the world in competitions to win the RoboCup challenges. The RoboCup competitions have added an additional search-and-rescue category to the competition [53.367], which has also become a significant area of research (Chap. 66 for more details on this field). Annual proceedings of the RoboCup competitions document much of the research that is incorporated into the multirobot soccer teams. Some representative research works include [53.368–372] (VIDEO 202, VIDEO 209).

53.11 Conclusions and Further Reading

This chapter has surveyed the current state of the art in multirobot systems, examining architectures, communications issues, swarm robot systems, heterogeneous teams, task allocation, learning, and applications. Clearly, significant advances have been made in the field in the last decade. The field is still an active area of research, however, since many open research issues still remain to be solved. Key open research questions remain in the broad areas of system integration, robustness, learning, scalability, generalization, and dealing with heterogeneity.

For example, in the area of system *integration*, an open question is how to effectively allow robot teams to combine a spectrum of approaches toward achieving complete systems that can perform more than a limited set of tasks. We are a long way from creating robust robot networks that can perform physical tasks in the real world. In the area of *robustness*, multirobot teams still need improvements in the ability to degrade gracefully, to reason for fault tolerance, and to achieve complexity without escalating failure rates. The area of *learning* in multirobot teams is still in its infancy, with open questions including how to achieve continual learning in multirobot teams, how to facilitate the use of complex representations, and how to enable humans to influence and/or understand the results of the team learning. *Scalability* is still a challenging problem, in terms of more complex environments as well as ever-larger numbers of robots. We do not yet have a methodology for creating self-organizing robot networks that are robust to labeling (or numbering), with completely decentralized controllers and estimators, and with provable emergent response. This requires basic research at the intersection of control, perception, and communication. Open issues in *generalization* include enabling the robot team to reason about context and increasing the versatility of systems so that they can operate in a variety of different applications. In dealing with *heterogeneity*, open questions include determining theoretical approaches to predicting system performance when all robots are not equal, and determining how to design a robot team optimally for a given application. Advances over the last decade have provided human users with the ability to interact with hundreds or thousands of computers on the Internet. It is necessary to develop similar network-centric approaches to interfacing, both for control and for monitoring. Finally, a major challenge is to create systems that are proactive and anticipate our needs and commands rather than reacting (with delays) to human commands.

For further reading on the topic of multiple mobile robot systems, the reader is referred to survey articles in the field, including [53.2, 288, 373, 374]. Additionally, several special journal issues on this topic have appeared, including [53.1, 375–377]. Some taxonomies of multirobot systems are given in [53.26, 288, 378]. A variety of symposia and workshops have been held on a regular basis on the topic of multirobot systems, in particular the DARS (distributed autonomous robotic systems) series of symposia. Recent proceedings of these workshops and symposia include [53.379–388]. Additional texts on this topic include [53.389–391]. For some excellent further background on networked robotics we direct the reader to [53.31, 35, 46, 52, 55, 75, 90, 104].

Video-References

VIDEO 192 Agents at play: Off-the-shelf software for practical multi-robot applications
available from http://handbookofrobotics.org/view-chapter/53/videodetails/192

VIDEO 193 Handling of a single object by multiple mobile robots based on caster-like dynamics
available from http://handbookofrobotics.org/view-chapter/53/videodetails/193

VIDEO 194 Synchronization and fault detection in autonomous robots
available from http://handbookofrobotics.org/view-chapter/53/videodetails/194

VIDEO 195 Self-assembly and morphology control in a swarm-bot
available from http://handbookofrobotics.org/view-chapter/53/videodetails/195

VIDEO 196 CKBOTS reconfigurable robots
available from http://handbookofrobotics.org/view-chapter/53/videodetails/196

VIDEO 197 Biologically inspired multi vehicles control algorithm
available from http://handbookofrobotics.org/view-chapter/53/videodetails/197

VIDEO 198 Metamorphic robotic system
available from http://handbookofrobotics.org/view-chapter/53/videodetails/198

VIDEO 199 Multi-robot box pushing
available from http://handbookofrobotics.org/view-chapter/53/videodetails/199

VIDEO 200 Elements of cooperative behavior in autonomous mobile robots
available from http://handbookofrobotics.org/view-chapter/53/videodetails/200
VIDEO 201 Coordination of multiple mobile platforms for manipulation and transportation
available from http://handbookofrobotics.org/view-chapter/53/videodetails/201
VIDEO 202 Robots in games and competition
available from http://handbookofrobotics.org/view-chapter/53/videodetails/202
VIDEO 203 A robotic reconnaissance and surveillance team
available from http://handbookofrobotics.org/view-chapter/53/videodetails/203
VIDEO 204 MARS (multiple autonomous robots)
available from http://handbookofrobotics.org/view-chapter/53/videodetails/204
VIDEO 205 A method for transporting a team of miniature robots
available from http://handbookofrobotics.org/view-chapter/53/videodetails/205
VIDEO 206 Reconfigurable multi-agents with distributed sensing for robust mobile robots
available from http://handbookofrobotics.org/view-chapter/53/videodetails/206
VIDEO 207 Miniature air vehicle cooperative timing missions
available from http://handbookofrobotics.org/view-chapter/53/videodetails/207
VIDEO 208 Distributed manipulation with mobile robots
available from http://handbookofrobotics.org/view-chapter/53/videodetails/208
VIDEO 209 Autonomous robot soccer – Through the wormhole with Morgan Freeman
available from http://handbookofrobotics.org/view-chapter/53/videodetails/209
VIDEO 210 A day in the life of a Kiva robot
available from http://handbookofrobotics.org/view-chapter/53/videodetails/210
VIDEO 211 Robot Pebbles – MIT developing self-sculpting smart sand robots
available from http://handbookofrobotics.org/view-chapter/53/videodetails/211
VIDEO 212 Transport of a child by swarm-bots
available from http://handbookofrobotics.org/view-chapter/53/videodetails/212
VIDEO 213 Towards a swarm of nano quadrotors
available from http://handbookofrobotics.org/view-chapter/53/videodetails/213
VIDEO 214 Swarm robotics at CU-Boulder
available from http://handbookofrobotics.org/view-chapter/53/videodetails/214
VIDEO 215 Swarm robot system
available from http://handbookofrobotics.org/view-chapter/53/videodetails/215
VIDEO 216 Swarm construction robots
available from http://handbookofrobotics.org/view-chapter/53/videodetails/216
VIDEO 217 Multi robot formation control – Khepera team
available from http://handbookofrobotics.org/view-chapter/53/videodetails/217
VIDEO 292 Experiments of escorting a target
available from http://handbookofrobotics.org/view-chapter/53/videodetails/292
VIDEO 293 Formation control via a distributed controller-observer
available from http://handbookofrobotics.org/view-chapter/53/videodetails/293

References

53.1 T. Arai, E. Pagello, L.E. Parker: Editorial: Advances in multi-robot systems, IEEE Trans. Robotics Autom. **18**(5), 655–661 (2002)

53.2 Y. Cao, A. Fukunaga, A. Kahng: Cooperative mobile robotics: Antecedents and directions, Auton. Robots **4**, 1–23 (1997)

53.3 R.G. Brown, J.S. Jennings: A pusher/steerer model for strongly cooperative mobile robot manipulation, IEEE/RSJ Int. Conf. Intell. Robots Syst. (1995) pp. 562–568

53.4 D. Milutinović, P. Lima: Modeling and optimal centralized control of a large-size robotic population, IEEE Trans. Robotics **22**(6), 1280–1285 (2006)

53.5 B. Khoshnevis, G.A. Bekey: Centralized sensing and control of multiple mobile robots, Comput. Ind. Eng. **35**(3/4), 503–506 (1998)

53.6 M.J. Matarić: Issues and approaches in the design of collective autonomous agents, Robotics Auton. Syst. **16**, 321–331 (1995)

53.7 E. Ostergaard, G.S. Sukhatme, M.J. Matarić: Emergent bucket brigading, 5th Int. Conf. Auton. Agents (2001)

53.8 M. Matarić: Reinforcement learning in the multi-robot domain, Auton. Robots **4**, 73–83 (1997)

53.9 L.E. Parker: ALLIANCE: An architecture for fault-tolerant multi-robot cooperation, IEEE Trans. Robotics Autom. **14**(2), 220–240 (1998)

53.10 L.E. Parker: Lifelong adaptation in heterogeneous teams: Response to continual variation in individual robot performance, Auton. Robots **8**(3), 239–267 (2000)

53.11 R. Simmons, S. Singh, D. Hershberger, J. Ramos, T. Smith: First results in the coordination of het-

erogeneous robots for large-scale assembly, Proc. ISER 7th Int. Symp. Exp. Robotics (2000)
53.12 J. Deneubourg, S. Goss, G. Sandini, F. Ferrari, P. Dario: Self-organizing collection and transport of objects in unpredictable environments, Jpn.-USA Symp. Flex. Autom. (1990) pp. 1093–1098
53.13 C.R. Kube, H. Zhang: Collective robotics: From social insects to robots, Adapt. Behav. **2**(2), 189–219 (1993)
53.14 R. Beckers, O. Holland, J. Deneubourg: From local actions to global tasks: Stigmergy and collective robotics, Proc. 14th Int. Workshop Synth. Simul. Living Syst. (1994) pp. 181–189
53.15 S. Onn, M. Tennenholtz: Determination of social laws for multi-agent mobilization, Artif. Intell. **95**, 155–167 (1997)
53.16 B.B. Werger: Cooperation without deliberation: A minimal behavior-based approach to multi-robot teams, Artif. Intell. **110**(2), 293–320 (1999)
53.17 M.J. Huber, E. Durfee: Deciding when to commit to action during observation-based coordination, Proc. 1st Int. Conf. Multi-Agent Syst. (1995) pp. 163–170
53.18 H. Asama, K. Ozaki, A. Matsumoto, Y. Ishida, I. Endo: Development of task assignment system using communication for multiple autonomous robots, J. Robotics Mechatron. **4**(2), 122–127 (1992)
53.19 N. Jennings: Controlling cooperative problem solving in industrial multi-agent systems using joint intentions, Artif. Intell. **75**(2), 195–240 (1995)
53.20 M. Tambe: Towards flexible teamwork, J. Artif. Intell. Res. **7**, 83–124 (1997)
53.21 R.T. Vaughan, K. Stoy, G.S. Sukhatme, M.J. Matarić: LOST: Localization-space trails for robot teams, IEEE Trans. Robotics Autom. **18**(5), 796–812 (2002)
53.22 L.E. Parker: The Effect of action recognition and robot awareness in cooperative robotic teams, IEEE/RSJ Int. Conf. Intell. Robots Syst. (1995) pp. 212–219
53.23 M. Matarić: Behavior-based control: Examples from navigation, learning, and group behavior, J. Exp. Theor. Artif. Intell. **19**(2/3), 323–336 (1997)
53.24 B. MacLennan, G.M. Burghardt: Synthetic ethology and the evolution of cooperative communication, Adapt. Behav. **2**, 161–188 (1993)
53.25 T. Balch, R.C. Arkin: Communiation in reactive multiagent robotic systems, Auton. Robots **1**(1), 27–52 (1995)
53.26 G. Dudek, M. Jenkin, E. Milios, D. Wilkes: A taxonomy for multi-agent robotics, Auton. Robots **3**, 375–397 (1996)
53.27 Z. Butler, K. Kotay, D.L. Rus, M. Vona: Self-reconfiguring robots, Commun. ACM **45**(3), 39–45 (2002)
53.28 O. Khatib, K. Yokoi, K. Chang, D. Ruspini, R. Holmberg, A. Casal: Coordination and decentralized cooperation of multiple mobile manipulators, J. Robotic Syst. **13**(11), 755–764 (1996)
53.29 L.E. Parker: The effect of heterogeneity in teams of 100+ mobile robots. In: *Multi-Robot Systems Volume II: From Swarms to Intelligent Automata*, ed. by A. Schultz, L.E. Parker, F. Schneider (Kluwer, Dordrecht 2003)
53.30 M. Yim, Y. Zhang, D. Duff: Modular robots, IEEE Spectrum **39**(22), 30–34 (2002)
53.31 D. Estrin: *Embedded, Everywhere* (National Academies Press, Washington 2001)
53.32 J. Parrish, S. Viscido, D. Grünbaum: Self-organized fish schools: An examination of emergent properties, Biol. Bull. **202**, 296–305 (2002)
53.33 N. Franks, S. Pratt, E. Mallon, N. Britton, D. Sumpter: Information flow, opinion polling and collective intelligence in house-hunting social insects, Philos. Trans. R. Soc. B **357**, 1567–1584 (2002)
53.34 R.L. Jeanne: Group size, productivity, and information flow in social wasps. In: *Information Processing in Social Insects*, ed. by C. Detrain, J.M. Pasteels, J.L. Deneubourg (Birkhauser, Basel 1999)
53.35 T. Akimoto, N. Hagita: Introduction to a network robot system, IEEE Int. Symp. Intell. Signal Proc. Comm. (2006)
53.36 Argo Floats: *A global array of 3000 Free-Drifting Profiling Floats for Environmental Monitoring* (Argo Information Center, Ramonville 2007)
53.37 G.S. Sukhatme, A. Dhariwal, B. Zhang, C. Oberg, B. Stauffer, D.A. Caron: The design and development of a wireless robotic networked aquatic microbial observing system, Environ. Eng. Sci. **24**(2), 205–215 (2006)
53.38 W. Kaiser, G. Pottie, M. Srivastava, G.S. Sukhatme, J. Villasenor, D. Estrin: Networked infomechanical systems (NIMS) for ambient intelligence. In: *Ambient Intelligence*, ed. by W. Weber, J.M. Rabaey, E. Aarts (Springer, Berlin, Heidelberg 2005)
53.39 Amarss: *Networked Minirhizotron Planning and Initial Deployment* (Center for Embedded Networkedsensing, Los Angeles 2007)
53.40 H. Singh, J. Catipovic, R. Eastwood, L. Freitag, H. Henriksen, F.F. Hover, D. Yoerger, J.G. Bellingham, B.A. Moran: An integrated approach to multiple AUV communications, navigation and docking, MTS/IEEE Conf. Proc. OCEANS (1996) pp. 59–64
53.41 I. Vasilescu, M. Dunbabin, P. Corke, K. Kotay, D. Rus: Data collection, storage, and retrieval with an underwater sensor network, Proc. ACM Sens. Syst. (2005)
53.42 N. Leonard, D. Paley, F. Lekien, R. Sepulchre, D.M. Fratantoni, R. Davis: Collective motion, sensor networks and ocean sampling, Proc. IEEE **95**(1), 48–74 (2006)
53.43 D.O. Popa, A.S. Sanderson, R.J. Komerska, S.S. Mupparapu, D.R. Blidberg, S.G. Chappell: Adaptive sampling algorithms for multiple autonomous underwater vehicles, IEEE/OES AUV2004: A Workshop Multiple Auton. Underw. Veh. Oper. (2004)
53.44 L. Chaimowicz, A. Cowley, B. Grocholsky, M.A. Hsieh, J.F. Keller, V. Kumar, C.J. Taylor: Deploying air-ground multi-robot teams in urban environments, 3rd Multi-Robot Syst. Workshop (2005)

53.45 B. Grocholsky, R. Swaminathan, J. Keller, V. Kumar, G. Pappas: Information driven coordinated air-ground proactive sensing, Proc. IEEE Int. Conf. Robotics Autom. (ICRA) (2005)
53.46 A. Howard, L.E. Parker, G.S. Sukhatme: Experiments with a large heterogeneous mobile robot team: Exploration, mapping, deployment and detection, Int. J. Robotics Res. **25**(5/6), 431–447 (2006)
53.47 D. Fox, J. Ko, K. Konolige, B. Limketkai, D. Schulz, B. Stewart: Distributed multirobot exploration and mapping, Proc. IEEE **94**(7), 1325–1339 (2006)
53.48 M.A. Hsieh, A. Cowley, J.F. Keller, L. Chaimowicz, B. Grocholsky, V. Kumar, C.J. Talyor, Y. Endo, R. Arkin, B. Jung, D. Wolf, G. Sukhatme, D.C. MacKenzie: Adaptive teams of autonomous aerial and ground robots for situational awareness, J. Field Robotics **24**(11/12), 991–1014 (2007)
53.49 J. Seyfied, M. Szymanski, N. Bender, R. Estana, M. Theil, H. Worn: The I-swarm project: Intelligent small world autonomous robots for micromanipulation, SAB 2004 Int. Workshop (2004)
53.50 F. Mondada, G.C. Pettinaro, A. Guignard, I.W. Kwee, D. Floreano, J.-L. Deneubourg, S. Nofli, L.M. Gambardella, M. Dorigo: Swarm-Bot: A new distributed robotic concept, Auton. Robots **17**, 193–221 (2004)
53.51 A. Ollero, S. Lacroix, L. Merino, J. Gancet, J. Wiklund, V. Remuss, I. Veiga, L.G. Gutierrez, D.X. Viegas, M.A. Gonzalez, A. Mallet, R. Alami, R. Chatila, G. Hommel, F.J. Colmenero, B. Arrue, J. Ferruz, R. Martinez de Dios, F. Caballero: Architecture and perception issues in the COMETS multi-UAV project, IEEE Robotics Autom. Mag. **12**(2), 46–57 (2005)
53.52 Q. Li, D. Rus: Navigation protocols in sensor networks, ACM Trans. Sens. Netw. **1**(1), 3–35 (2005)
53.53 R. Bajcsy: Active perception, Proc. IEEE **76**, 996–1005 (1988)
53.54 T. Eren, D. Goldenberg, W. Whitley, Y.R. Yang, S. Morse, B.D.O. Anderson, P.N. Belhumeur: Rigidity, computation, and randomization of network localization, Proc. IEEE INFOCOM (2004)
53.55 A. Hsieh, A. Cowley, V. Kumar, C.J. Taylor: Towards the deployment of a mobile robot network with end-to-end performance guarantees, Proc. IEEE Int. Conf. Robotics Autom. (ICRA) (2006)
53.56 B. Grocholsky, S. Bayraktar, V. Kumar, C.J. Taylor, G. Pappas: Synergies in feature localization by air-ground teams, Proc. 9th Int. Symp. Exp. Robotics (2004)
53.57 B. Grocholsky, E. Stump, V. Kumar: An extensive representation for range-only SLAM, Int. Symp. Exp. Robotics (2006)
53.58 A. Howard, M.J. Matarić, G.S. Sukhatme: An incremental self-deployment algorithm for mobile sensor networks, Auton. Robots **13**(2), 113–126 (2002)
53.59 Z. Butler, D. Rus: Controlling mobile networks for monitoring events with coverage constraints, Proc. IEEE Int. Conf. Robotics Autom. (ICRA) (2003)
53.60 M. Chu, J. Reich, F. Zhao: Distributed attention for large video sensor networks, Intell. Distrib. Surveill. Syst. Semin (2004)
53.61 B. Jung, G.S. Sukhatme: Tracking targets using multiple robots: The effect of environment occlusion, Auton. Robots **13**(3), 191–205 (2002)
53.62 R.M. Murray, K.J. Åström, S.P. Boyd, R.W. Brockett, G. Stein: Future directions in control in an information-rich world, IEEE Control Syst. Mag. (2003)
53.63 A. Pant, P. Seiler, K. Hedrick: Mesh stability of look-ahead interconnected systems, IEEE Trans. Autom. Control **47**, 403–407 (2002)
53.64 T. Sugar, J. Desai, V. Kumar, J.P. Ostrowski: Coordination of multiple mobile manipulators, Proc. IEEE Int. Conf. Robotics Autom. (ICRA) (2001) pp. 3022–3027
53.65 R.W. Beard, J. Lawton, F.Y. Hadaegh: A coordination architecture for spacecraft formation control, IEEE Trans. Control Syst. Technol. **9**, 777–790 (2001)
53.66 A. Das, J. Spletzer, V. Kumar, C. Taylor: Ad hoc networks for localization and control, Proc. IEEE Conf. Decis. Control (2002) pp. 2978–2983
53.67 J. Manyika, H. Durrant-Whyte: *Data Fusion and Sensor Management: An Information-Theoretic Approach* (Prentice Hall, Upper Saddle River 1994)
53.68 H.G. Tanner, A. Jadbabaie, G.J. Pappas: Stable flocking of mobile agents, Part I: Fixed topology, Proc. IEEE Conf. Decis. Control (2003) pp. 2010–2015
53.69 J.M. Fowler, R. D'Andrea: Distributed control of close formation flight, Proc. IEEE Conf. Decis. Control (2002) pp. 2972–2977
53.70 J.P. Desai, J.P. Ostrowski, V. Kumar: Modeling and control of formations of nonholonomic mobile robots, IEEE Trans. Robotics Autom. **17**(6), 905–908 (2001)
53.71 H.G. Tanner, V. Kumar, G.J. Pappas: Leader-to-formation stability, IEEE Trans. Robotics Autom. **20**(3), 443–455 (2004)
53.72 S. Loizou, V. Kumar: Relaxed input to state stability properties for navigation function based systems, Proc. IEEE Conf. Decis. Control (2006)
53.73 M. Batalin, G.S. Sukhatme: Coverage, exploration and deployment by a mobile robot and communication network, Telecommun. Syst. J. **26**(2), 181–196 (2004)
53.74 K.J. O'hara, V. Bigio, S. Whitt, D. Walker, T.R. Balch: Evaluation of a large scale pervasive embedded network for robot path planning, Proc. IEEE Int. Conf. Robotics Autom. (ICRA) (2006) pp. 2072–2077
53.75 V. Kumar, N. Leonard, A.S. Morse (Eds.): *Cooperative control*, Lecture Notes in Control and Information Sciences, Vol. 309 (Springer, Berlin, Heidelberg 2004)
53.76 J. Chen, S. Teller, H. Balakrishnan: Pervasive pose-aware applications and infrastructure, IEEE Comput. Graph. Appl. **23**(4), 14–18 (2003)
53.77 A. Savvides, C.-C. Han, M. Srivastava: Dynamic fine-grained localization in ad-hoc networks of sensors, Proc. 7th Annu. Int. Conf. Mobile Comput. Netw. (MOBICOM-01) (2001) pp. 166–179

53.78 D. Moore, J. Leonard, D. Rus, S.J. Teller: Robust distributed network localization with noisy range measurements, Proc. 2nd Int. Conf. Embed. Netw. Sens. Syst. (SenSys) (2004) pp. 50–61
53.79 C. Detweiler, J. Leonard, D. Rus, S. Teller: Passive mobile robot localization within a fixed beacon field, Proc. Int. Workshop Algorithmic Found. Robotics (2006)
53.80 N. Bulusu, J. Heidemann, D. Estrin: Adaptive beacon placement, Proc. 21st Int. Conf. Distrib. Comput. Syst. (ICDCS-01) (2001) pp. 489–498
53.81 S.N. Simic, S. Sastry: Distributed Localization in Wireless ad hoc Networks, Tech. Rep. UCB/ERL M02/26 (2001), http://www.eecs.berkeley.edu/Pubs/TechRpts/2002/4010.html
53.82 P. Corke, R. Peterson, D. Rus: Communication-assisted localization and navigation for networked robots, Int. J. Robotics Res. **4**(9), 116 (2005)
53.83 R. Nagpal, H.E. Shrobe, J. Bachrach: Organizing a global coordinate system from local information on an ad hoc sensor network, Lect. Notes Comput. Sci. **2634**, 333–348 (2003)
53.84 R.K. Williams, A. Gasparri, A. Priolo, G.S. Sukhatme: Evaluating network rigidity in realistic systems: Decentralization, asynchronicity, and parallelization, IEEE Trans. Robotics **30**(4), 950–965 (2014)
53.85 A. Howard, M.J. Matarić, G.S. Sukhatme: Putting the 'i' in 'team': An ego-centric approach to cooperative localization, Proc. IEEE Int. Conf. Robotics Autom. (ICRA) (2003) pp. 868–892
53.86 A. Howard, M.J. Matarić, G.S. Sukhatme: Localization for mobile robot teams using maximum likelihood estimation, IEEE/RSJ Int. Conf. Intell. Robots Syst. (2002) pp. 434–459
53.87 R. Vidal, O. Shakernia, H.J. Kim, D.H. Shim, S. Sastry: Probabilistic pursuit-evasion games: Theory, implementation and experimental evaluation, IEEE Trans. Robotics Autom. **18**(5), 662–669 (2002)
53.88 M. Batalin, M.H. Rahimi, Y. Yu, D. Liu, A. Kansal, G.S. Sukhatme, W. Kaiser, M. Hansen, G. Pottie, M. Srivastava, D. Estrin: Call and response: Experiments in sampling the environment, Proc. 2nd Int. Conf. Embed. Netw. Sens. Syst. (SenSys) (2004) pp. 25–38
53.89 M.H. Rahimi, W. Kaiser, G.S. Sukhatme, D. Estrin: Adaptive sampling for environmental field estimation using robotic sensors, IEEE/RSJ Int. Conf. Intell. Robots Syst. (2005) pp. 747–753
53.90 F. Zhao, L. Guibas: *Wireless Sensor Networks: An Information Processing Approach* (Morgan Kaufmann, New York 2004)
53.91 F. Zhang, B. Grocholsky, V. Kumar: Formations for localization of robot networks, IEEE Int. Conf. Robotics Autom. (2004)
53.92 R.K. Williams, G.S. Sukhatme: Constrained interaction and coordination in proximity-limited multiagent systems, IEEE Trans. Robotics **29**(4), 930–944 (2013)
53.93 J. Cortes, S. Martinez, T. Karatas, F. Bullo: Coverage control for mobile sensing networks, IEEE Trans. Robotics Autom. **20**(2), 243–255 (2004)
53.94 J. Cortes, S. Martinez, F. Bullo: Spatially-distributed coverage optimization and control with limited-range interactions, ESAIM Control Optim. Calc. Var. **11**, 691–719 (2005)
53.95 M. Schwager, J. McLurkin, D. Rus: Distributed coverage control with sensory feedback for networked robots, Proc. Robotics Sci. Syst. (RSS) (2006)
53.96 M. Schwager, J.-J. Slotine, D. Rus: Decentralized adaptive control for coverage for networked robots, Proc. Int. Conf. Robotics Autom. (2007)
53.97 A. Krause, C. Guestrin, A. Gupta, J. Kleinberg: Near-optimal sensor placements: Maximizing information while minimizing communication cost, 5th Int. Conf. Inf. Process Sens. Netw. (IPSN) (2006)
53.98 V. Chvatal: A combinatorial theorem in plane geometry, J. Comb. Theory Ser. **18**, 39–41 (1975)
53.99 J. O'Rourke: *Art Gallery Theorems and Algorithms* (Oxford Univ. Press, New York 1987)
53.100 S. Fisk: A short proof of Chvatal's watchmen theorem, J. Comb. Theory Ser. **24**, 374 (1978)
53.101 A. Jadbabaie, J. Lin, A.S. Morse: Coordination of groups of mobile autonomous agents using nearest neighbor rules, IEEE Trans. Autom. Control **48**(6), 988–1001 (2003)
53.102 S. Sinha, M. Pollefeys: Camera network calibration from dynamic silhouettes, Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (2004)
53.103 R. Collins, A. Lipton, H. Fujiyoshi, T. Kanade: Algorithms for cooperative multisensor surveillance, Proc. IEEE **89**(10), 1456–1477 (2001)
53.104 A. Kansal, W. Kaiser, G. Pottie, M. Srivastava, G.S. Sukhatme: Reconfiguration methods for mobile sensor networks, ACM Trans. Sens. Networks **3**(4), 1–28 (2007)
53.105 D. Payton, M. Daily, R. Estkowski, M. Howard, C. Lee: Pheromone robotics, Auton. Robots **11**, 319–324 (2001)
53.106 A. Howard, M.J. Matarić, G.S. Sukhatme: Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem, Proc. Int. Symp. Distrib. Auton. Robotic Syst. (2002) pp. 299–308
53.107 S. Poduri, G.S. Sukhatme: Constrained coverage for mobile sensor networks, IEEE Int. Conf. Robotics Autom. (2004) pp. 165–172
53.108 R. Wattenhofer, L. Li, P. Bahl, Y.M. Wang: A cone-based distributed topology-control algorithm for wireless multi-hop networks, IEEE/ACM Trans. Netw. **13**(1), 147–159 (2005)
53.109 H. Zhang, J.C. Hou: On deriving the upper bound of alphalifetime for large sensor networks, ACM Trans. Sens. Netw. **1**(6), 272–300 (2005)
53.110 S. Poduri, S. Pattem, B. Krishnamachari, G.S. Sukhatme: Using local geometry for tunable topology control in sensor networks, IEEE Trans. Mobile Comput. **8**(2), 218–230 (2009)
53.111 M.A. Hsieh, V. Kumar: Pattern generation with multiple robots, Proc. IEEE Int. Conf. Robotics Autom. (ICRA) (2006)
53.112 R.N. Smith, M. Schwager, S.L. Smith, B.H. Jones, D. Rus, G.S. Sukhatme: Persistent ocean mon-

itoring with underwater gliders: Adapting spatiotemporal sampling resolution, J. Field Robotics **28**(5), 714–741 (2011)
53.113 J. Das, F. Py, T. Maughan, T. O'Reilly, M. Messié, J. Ryan, G.S. Sukhatme, K. Rajan: Coordinated sampling of dynamic oceanographic features with AUVs and drifters, Int. J. Robotics Res. **31**(5), 626–646 (2012)
53.114 G.A. Hollinger, S. Choudhary, P. Qarabaqi, C. Murphy, U. Mitra, G.S. Sukhatme, M. Stojanovic, H. Singh, F. Hover: Underwater data collection using robotic sensor networks, IEEE J. Sel. Areas Commun. **30**(5), 899–911 (2012)
53.115 F. Ye, G. Zhong, J. Cheng, L. Zhang, S. Lu: Peas: A robust energy conserving protocol for long-lived sensor networks, Int. Conf. Distrib. Comput. Syst. (2003)
53.116 X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, C. Gill: Integrated coverage and connectivity configuration in wireless sensor networks, 1st ACM Conf. Embed. Netw. Sens. Syst. (SenSys) (2003)
53.117 D. Tian, N.D. Georganas: Connectivity maintenance and coverage preservation in wireless sensor networks, Ad Hoc Networks **3**(6), 744–761 (2005)
53.118 G. Theraulaz, S. Goss, J. Gervet, J.-L. Deneubourg: Task differentiation in Polistes wasp colonies: A model for self-organizing groups of robots, Proc. 1st Int. Conf. Simul. Adapt. Behav. (1990) pp. 346–355
53.119 L. Steels: Cooperation Between Distributed Agents Through Self-Organization, Proc. IEEE Int. Workshop Intell. Robots Syst. (IROS) (1990)
53.120 A. Drogoul, J. Ferber: From tom thumb to the dockers: Some experiments with foraging robots, Proc. 2nd Int. Conf. Simul. Adapt. Behav. (1992) pp. 451–459
53.121 M.J. Matarić: Designing emergent behaviors: From local interactions to collective intelligence, Proc. 2nd Int. Conf. Simul. Adapt. Behav. (1992) pp. 432–441
53.122 G. Beni, J. Wang: Swarm intelligence in cellular robotics systems, Proc. NATO Adv. Workshop Robots Biol. Syst. (1989)
53.123 D. Stilwell, J. Bay: Toward the development of a material transport system using swarms of ant-like robots, Proc. IEEE Int. Conf. Robotics Autom. (1993) pp. 766–771
53.124 T. Fukuda, S. Nakagawa, Y. Kawauchi, M. Buss: Self organizing robots based on cell structures – CEBOT, Proc. IEEE Int. Workshop Intell. Robots Syst. (1988) pp. 145–150
53.125 J.H. Reif, H.Y. Wang: Social Potential fields: A distributed behavior control for autonomous robots, Robotics Auton. Syst. **27**(3), 171–194 (1999)
53.126 L.E. Parker: Designing control laws for cooperative agent teams, Proc. IEEE Int. Cont. Robotics Autom. (1993) pp. 582–587
53.127 K. Sugihara, I. Suzuki: Distributed algorithms for formation of goemetric patterns with many mobile robots, J. Robotic Syst. **13**(3), 127–139 (1996)
53.128 V. Gazi: Swarm aggregations using artificial potentials and sliding-mode control, IEEE Trans. Robotics **21**(6), 1208–1214 (2005)
53.129 J. McLurkin, J. Smith: Distributed algorithms for dispersion in indoor environments using a swarm of autonomous mobile robots, Symp. Distrib. Auton. Robots Syst. (2004)
53.130 D. Gage: Randomized search strategies with imperfect sensors, Proc. SPIE Mobile Robots VIII (1993) pp. 270–279
53.131 T. Balch: The impact of diversity on performance in robot foraging, Proc. 3rd Ann. Conf. Auton. Agents (1999) pp. 92–99
53.132 Z.J. Butler, A.A. Rizzi, R.L. Hollis: Cooperative coverage of rectilinear environments, Proc. IEEE Int. Conf. Robotics Autom. (ICRA) (2000)
53.133 A.I. Mourikis, S.I. Roumeliotis: Performance analysis of multirobot cooperative localization, IEEE Trans. Robotics **22**(4), 666–681 (2006)
53.134 R. Grabowski, L.E. Navarro-Serment, C.J. Paredis, P.K. Khosla: Heterogeneous teams of modular robots for mapping and exploration, Auton. Robots **8**(3), 271–298 (2000)
53.135 M. Berhault, H. Huang, P. Keskinocak, S. Koenig, W. Elmaghraby, P. Griffin, A. Kleywegt: Robot exploration with combinatorial auctions, Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS) (2003) pp. 1957–1962
53.136 J. Kim, J.M. Esposito, V. Kumar: An RRT-based algorithm for testing and validating mulit-robot controllers, Proc. Robotics Sci. Syst. I (2005)
53.137 Z. Cao, M. Tin, L. Li, N. Gu, S. Wang: Cooperative hunting by distributed mobile robots based on local interaction, IEEE Trans. Robotics **22**(2), 403–407 (2006)
53.138 R.W. Beard, T.W. McLain, M. Goodrich: Coordinated target assignment and intercept for unmanned air vehicles, Proc. IEEE Int. Conf. Robotics Autom. (ICRA) (2002) pp. 2581–2586
53.139 J. Clark, R. Fierro: Cooperative hybrid control of robotic sensors for perimeter detection and tracking, Proc. Am. Control Conf. (2005) pp. 3500–3505
53.140 P. Stone, M. Veloso: A layered approach to learning client behaviors in the RoboCup soccer server, Appl. Artif. Intell. **12**, 165–188 (1998)
53.141 J. McLurkin: Stupid Robot Tricks: Behavior-Based Distributed Algorithm Library for Programming Swarms of Robots, Ph.D. Thesis (Massachusetts Institute of Technology, Cambridge 2004)
53.142 F. Mondada, L.M. Gambardella, D. Floreano, S. Nolfi, J.L. Deneuborg, M. Dorigo: The cooperation of swarm-bots: Physical interactions in collective robotics, IEEE Robotics Autom. Mag. **12**(2), 21–28 (2005)
53.143 T.W. Mather, M.A. Hsieh: Macroscopic modeling of stochastic deployment policies with time delays for robot ensembles, Int. J. Robotics Res. **30**(5), 590–600 (2011)
53.144 X.C. Ding, M. Kloetzer, Y. Chen, C. Belta: Automatic deployment of robot teams, IEEE Robotics Autom. Mag. **18**(3), 75–86 (2022)

53.145 F.S. Melo, M. Veloso: Decentralized MDPs with sparse interactions, Artif. Intell. **175**(11), 1757–1789 (2011)
53.146 P. Tsiotras, L.I.R. Castro: Extended multi-agent consensus protocols for the generation of geometric patterns in the plane, Am. Controls Conf. (2011) pp. 3850–3855
53.147 K. Cheng, P. Dasgupta: Weighted voting game based multi-robot team formation for distributed area coverage, Proc. 3rd Int. Symp. Pract. Cognit. Agents Robots (2010) pp. 9–15
53.148 A.A. Taheri, M. Afshar, M. Asadpour: Influence maximization for informed agents in collective behavior, Distrib. Auton. Robotic Syst. (2013) pp. 389–402
53.149 M.A. Hsieh, A. Halász, E.D. Cubuk, S. Schoenholz, A. Martinoli: Specialization as an optimal strategy under varying external conditions, Proc. IEEE Int. Conf. Robotics Autom. (2009) pp. 1941–1946
53.150 N. Hoff, R. Wood, R. Nagpal: Distributed colony-level algorithm switching for robot swarm foraging, Distrib. Auton. Robotic Syst. (2013) pp. 417–430
53.151 A. Winfield, J. Nembrini: Safety in numbers: Fault-tolerance in robot systems, Int. J. Model. Identif. Control **1**(1), 30–37 (2006)
53.152 T. Fukuda, S. Nakagawa: Dynamically reconfigurable robotic system, IEEE Int. Conf. Robotics Autom. (1988) pp. 1581–1586
53.153 M. Yim, W.-M. Shen, B. Salemi, D. Rus, M. Moll, H. Lipson, E. Klavins, G.S. Chirikjian: Modular self-reconfigurable robot systems: Challenges and opportunities for the future, IEEE Robotics Autom. Mag. **14**(1), 43–52 (2007)
53.154 K. Gilpin, D. Rus: Modular robot systems: From self-assembly to self-disassembly, IEEE Robotics Autom. Mag. **17**(3), 38–53 (2010)
53.155 M. Yim: A reconfigurable modular robot with many modes of locomotion, JSME Int. Conf. Adv. Mechatron. (1993) pp. 283–288
53.156 M. Yim: New locomotion gaits, Proc. IEEE Int. Conf. Robotics Autom. (ICRA) (1994) pp. 2508–2514
53.157 A. Castano, P. Will: Mechanical design of a module for reconfigurable robots, Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS) (2000) pp. 2203–2209
53.158 W.-M. Shen, P. Will: Docking in self-reconfigurable robots, Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS) (2001) pp. 1049–1054
53.159 A. Castano, A. Behar, P. Will: The conro modules for reconfigurable robots, IEEE Trans. Mechatron. **7**(4), 403–409 (2002)
53.160 S. Murata, E. Yoshida, A. Kamimura, H. Kurokawa, K. Tomita, S. Kokaji: M-TRAN: Self-reconfigurable modular robotic system, IEEE/ASME Trans. Mechatron. **7**(4), 431–441 (2002)
53.161 A. Kamimura, H. Kurokawa, E. Yoshida, S. Murata, K. Tomita, S. Kokaji: Automatic locomotion design and experiments for a modular robotic system, IEEE/ASME Trans. Mechatron. **10**(3), 314–325 (2005)
53.162 H. Kurokawa, K. Tomita, A. Kamimura, E. Yoshida, S. Kokahji, S. Murata: Distributed self-reconfiguration control of modular robot M-TRAN, IEEE Int. Conf. Mechatron. Autom. (2005) pp. 254–259
53.163 S. Murata, K. Kakomura, H. Kurokawa: Docking experiments of a modular robot by visual feedback, Proc. IEEE/RSJ Int. Conf. Intell. Robots Systems (IROS) (2006) pp. 625–630
53.164 D. Marbach, A.J. Ijspeert: Online optimization of modular robot locomotion, IEEE Int. Conf. Mechatron. Autom. (2005) pp. 248–253
53.165 E.H. Østergaard, H.H. Lund: Evolving control for modular robotic units, IEEE Int. Symp. Comput. Intell. Robotics Autom. (2003) pp. 886–892
53.166 M.W. Jørgensen, E.H. Østergaard, H.H. Lund: Modular ATRON: Modules for a self-reconfigurable robot, Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS) (2004) pp. 2068–2073
53.167 B. Salemi, M. Moll, W.-M. Shen: SUPERBOT: A deployable, multi-functional, and modular self-reconfigurable robotic system, Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS) (2006) pp. 3636–3641
53.168 M. Yim, D.G. Duff, K.D. Roufas: PolyBot: A modular reconfigurable robot, Proc. IEEE Int. Conf. Robotics Autom. (ICRA) (2000) pp. 514–520
53.169 M. Yim, Y. Zhang, K. Roufas, D. Duff, C. Eldershaw: Connecting and disconnecting for self-reconfiguration with PolyBot, IEEE/ASME Trans. Mechatron. **7**(4), 442–451 (2003)
53.170 M. Yim, B. Shirmohammadi, J. Sastra, M. Park, M. Dugan, C.J. Taylor: Towards robotic self-reassembly after explosion, IEEE/RSJ Int. Conf. Intell. Robots Syst. (2007) pp. 2767–2772
53.171 V. Zykov, E. Mytilinaios, M. Desnoyer, H. Lipson: Evolved and designed self-reproducing modular robotics, IEEE Trans. Robotics **23**(2), 308–319 (2007)
53.172 P.J. White, M.L. Posner, M. Yim: Strength analysis of miniature folded right angle tetrahedron chain programmable matter, Proc. IEEE Int. Conf. Robotics Autom. (ICRA) (2010) pp. 2785–2790
53.173 G.S. Chirikjian: Kinematics of a metamorphic robotic system, Proc. IEEE Int. Conf. Robotics Autom. (ICRA) (1994) pp. 449–455
53.174 G. Chirikjian, A. Pamecha, I. Ebert-Uphoff: Evaluating efficiency of self-reconfiguration in a class of modular robots, J. Robotic Syst. **13**(5), 317–388 (1996)
53.175 A. Pamecha, I. Ebert-Uphoff, G.S. Chirikjian: Useful metrics for modular robot motion planning, IEEE Trans. Robotics Autom. **13**(4), 531–545 (1997)
53.176 J.E. Walter, E.M. Tsai, N.M. Amato: Algorithms for fast concurrent reconfiguration of hexagonal metamorphic robots, IEEE Trans. Robotics **21**(4), 621–631 (2005)
53.177 S. Murata, H. Kurokawa, S. Kokaji: Self-assembling machine, Proc. IEEE Int. Conf. Robotics Autom. (ICRA) (1994) pp. 441–448
53.178 E. Yoshida, S. Murata, K. Tomita, H. Kurokawa, S. Kokaji: Distributed formation control for a modular mechanical system, Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS) (1997) pp. 1090–1097

53.179 H. Kurokawa, S. Murata, E. Yoshida, K. Tomita, S. Kokaji: A 3-D self-reconfigurable structure and experiments, Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS) (1998) pp. 860–865

53.180 E. Yoshida, S. Murata, S. Kokaji, A. Kamimura, K. Tomita, H. Kurokawa: Get back in shape! A hardware prototype self-reconfigurable modular microrobot that uses shape memory alloy, IEEE Robotics Autom. Mag. **9**(4), 54–60 (2002)

53.181 J. Gargus, B. Kim, I. Llamas, J. Rossignac, C. Shaw: Finger Sculpting with Digital Clay, Tech. Rep. GIT-GVU-02-22 (2002)

53.182 K. Kotay, D. Rus, M. Vona, C. McGray: The self-reconfiguring robotic molecule, Proc. IEEE Int. Conf. Robotics Autom. (ICRA) (1998) pp. 424–431

53.183 K. Kotay, D. Rus: Motion synthesis for the self-reconfiguring robotic molecule, IEEE Int. Conf. Intell. Robots Syst. (1998) pp. 843–851

53.184 K. Kotay, D. Rus: Algorithms for self-reconfiguring molecule motion planning, Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS) (2000)

53.185 Z.J. Butler, K. Kotay, D. Rus, K. Tomita: Generic decentralized control for lattice-based self-reconfigurable robots, Int. J. Robotics Res. **23**(9), 919–937 (2004)

53.186 D. Rus, M. Vona: A basis for self-reconfiguring robots using crystal modules, Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS) (2000) pp. 2194–2202

53.187 D. Rus, M. Vona: Crystalline robots: Self-reconfiguration with compressible unit modules, Int. J. Robotics Res. **22**(9), 699–715 (2003)

53.188 J.W. Suh, S.B. Homans, M. Yim: Telecubes: Mechanical design of a module for self-reconfigurable robotics, Proc. IEEE Int. Conf. Robotics Autom. (ICRA) (2002) pp. 4095–4101

53.189 C.-J. Chiang, G.S. Chirikjian: Modular robot motion planning using similarity metrics, Auton. Robots **10**, 91–106 (2001)

53.190 M. Koseki, K. Minami, N. Inou: Cellular robots forming a mechanical structure (evaluation of structural formation and hardware design of CHOBIE II), Proc. 7th Int. Symp. Distrib. Auton. Robotic Syst. (DARS) (2004) pp. 131–140

53.191 B.K. An: Em-cube: Cube-shaped, self-reconfigurable robots sliding on structure surfaces, Proc. IEEE Int. Conf. Robotics Autom. (ICRA) (2008) pp. 3149–3155

53.192 C. Ünsal, P.K. Khosla: Mechatronic design of a modular self-reconfiguring robotic system, Proc. IEEE Int. Conf. Robotics Autom. (ICRA) (2000) pp. 1742–1747

53.193 K.C. Prevas, C. Ünsal, M.Ö. Efe, P.K. Khosla: A hierarchical motion planning strategy for a uniform self-reconfigurable modular robotic system, Proc. IEEE Int. Conf. Robotics Autom. (ICRA) (2002) pp. 787–792

53.194 K. Hosokawa, T. Tsujimori, T. Fujii, H. Kaetsu, H. Asama, Y. Kuroda, I. Endo: Self-organizing collective robots with morphogenesis in a vertical plane, Proc. IEEE Int. Conf. Robotics Autom. (ICRA) (1998) pp. 2858–2863

53.195 S. Goldstein, J. Campbell, T. Mowry: Programmable Matter, IEEE Computer **38**(6), 99–101 (2005)

53.196 S.C. Goldstein, J.D. Campbell, T.C. Mowry: Programmable matter, Computer **38**(6), 99–101 (2005)

53.197 M.E. Karagozler, S.C. Goldstein, J.R. Reid: Stress-driven MEMS assembly + electrostatic forces = 1 mm diameter robot, Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS) (2009) pp. 2763–2769

53.198 P. Pillai, J. Campbell, G. Kedia, S. Moudgal, K. Sheth: A 3-D fax machine based on claytronics, Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS) (2006) pp. 4728–4735

53.199 P. White, K. Kopanski, H. Lipson: Stochastic self-reconfigurable cellular robotics, IEEE Conf. Robotics Autom. (2004) pp. 2888–2893

53.200 P. White, V. Zykov, J. Bongard, H. Lipson: Three dimensional stochastic reconfiguration of modular robots, Robotics Sci. Syst. (2005)

53.201 M. Tolley, J. Hiller, H. Lipson: Evolutionary design and assembly planning for stochastic modular robots, Proc. IEEE/RSJ. Int. Conf. Intell. Robotics Syst. (IROS) (2009) pp. 73–78

53.202 M. Tolley, H. Lipson: Fluidic manipulation for scalable stochastic 3-D assembly of modular robots, Proc. IEEE Int. Conf. Robotics Autom. (ICRA) (2010) pp. 2473–2478

53.203 M.T. Tolley, H. Lipson: Programmable 3-D stochastic fluidic assembly of cm-scale modules, Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS) (2011) pp. 4366–4371

53.204 K. Gilpin, K. Kotay, D. Rus, I. Vasilescu: Miche: Modular shape formation by self-disassembly, Int. J. Robotics Res. **27**, 345–372 (2008)

53.205 R. Oung, F. Bourgault, M. Donovan, R. D'Andrea: The distributed flight array, Proc. IEEE Int. Conf. Robotics Autom. (ICRA) (2010)

53.206 G.J. Hamlin, A.C. Sanderson: Tetrobot: A modular system for hyper-redundant parallel robotics, Proc. IEEE Int. Conf. Robotics Autom. (ICRA) (1995) pp. 154–159

53.207 A. Lyder, R.F.M. Garcia, K. Stoy: Mechanical design of Odin, an extendable heterogeneous deformable modular robot, Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS) (2008) pp. 883–888

53.208 A. Lyder, H.G. Peterson, K. Stoy: Representation and shape estimation of Odin, a parallel under-actuated modular robot, Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS) (2009) pp. 5275–5280

53.209 C.-H. Yu, K. Haller, D. Ingber, R. Nagpal: Morpho: A self-deformable modular robot inspired by cellar structure, Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS) (2008) pp. 3571–3578

53.210 M. Shimizu, A. Ishiguro, T. Kawakatsu: A modular robot that exploits a spontaneous connectivity control mechanism, Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS) (2005) pp. 1899–1904

53.211 M. Shimizu, T. Mori, A. Ishiguro: A Development of a modular robot that enables adaptive reconfiguration, Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS) (2006) pp. 174–179

53.212 S. Funiak, P. Pillai, M.P. Ashley-Rollman, J.D. Campbell, S.C. Goldstein: Distributed localization of modular robot ensembles, Int. J. Robotics Res. **28**(8), 946–961 (2009)
53.213 M. Rubenstein, W.-M. Shen: Scalable self-assembly and self-repair in a collective of robots, Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS) (2009) pp. 1484–1489
53.214 M. Rubenstein, W.-M. Shen: Automatic scalable size selection for the shape of a distributed robotic collective, Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS) (2010) pp. 508–513
53.215 M. Rubenstein, C. Ahler, R. Nagpal: Kilobot: A low cost scalable robot system for collective behaviors, IEEE Int. Conf. Robotics Autom. (ICRA) (2012)
53.216 E. Hawkes, B. An, N.M. Benbernou, H. Tanaka, S. Kim, E.D. Demaine, D. Rus, R.J. Wood: Programmable matter by folding, Proc. Natl. Acad. Sci. USA **107**(28), 12441–12445 (2010)
53.217 B. An, D. Rus: Programming and controlling self-folding sheets, IEEE Int. Conf. Robotics Autom. (ICRA) (2012)
53.218 G. Whitesides, B. Grzybowski: Self-assembly at all scales, Sci. USA **295**, 2418–2421 (2002)
53.219 G.M. Whitesides, M. Boncheva: Beyond molecules: Self-assembly of mesoscope and macroscopic components, Proc. Natl. Acad. Sciences **99**(8), 4769–4774 (2002)
53.220 D.H. Garcias, J. Tien, T.L. Breen, C. Hsu, G.M. Whitesides: Forming electrical networks in three dimensions by self-assembly, Science **289**(5482), 1170–1172 (2000)
53.221 S. Miyashita, M. Kessler, M. Lungarella: How morphology affects self-assembly in a stochastic modular robot, IEEE Int. Conf. Robotics Autom. (2008) pp. 3533–3538
53.222 K. Hosokawa, I. Shimoyama, H. Miura: Dynamics of self-assembling systems: Analogy with chemical kinematics, Artif. Life **1**(4), 413–427 (1994)
53.223 M. Shimizu, K. Suzuki: A Self-repairing structure for modules and its control by vibrating actuation mechanisms, IEEE Int. Conf. Robotics Autom. (ICRA) (2009) pp. 4281–4286
53.224 P.W.K. Rothemund, E. Winfree: The program-size complexity of self-assembled squares, 32rd Annu. ACM Symp. Theory Comput. (2000) pp. 459–468
53.225 L. Adleman, Q. Cheng, A. Goel, M.-D. Huang: Running time and program size for self-assembled squares, 33rd Annu. ACM Symp. Theory Comput. (2001) pp. 740–748
53.226 G. Aggarwal, M.H. Goldwasser, M.-Y. Kao, R.T. Schweller: Complexities for generalized models of self-assembly, 15th Annu. ACM-SIAM Symp. Discrete Algorithms (2004) pp. 880–889
53.227 J. Bishop, S. Burden, E. Klavins, R. Kreisberg, W. Malone, N. Napp, T. Nguyen: Programmable parts: A demonstration of the grammatical approach to self-organization, Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS) (2005) pp. 3684–3691
53.228 S. Griffith, D. Goldwater, J.M. Jacobson: Robotics: Self-replication from random parts, Nature **437**, 636 (2005)
53.229 C. Jones, M.J. Matarić: From local to global behavior in intelligent self-assembly, IEEE Int. Conf. Robotics Autom. (ICRA) (2003) pp. 721–726
53.230 J. Kelly, H. Zhang: Combinatorial optimization of sensing for rule-based planar distributed assembly, IEEE Int. Conf. Intell. Robots Syst. (2006) pp. 3728–3734
53.231 J. Werfel: Anthills Built to Order: Automating Construction with Artificial Swarms, Ph.D. Thesis (MIT, Cambridge 2006)
53.232 B. Donald, C.G. Levey, C.D. McGray, I. Paprotny, D. Rus: An untethered, electrostatic, globally controllable MEMS micro-robot, J. Microelectromech. Syst. **15**(1), 1–15 (2006)
53.233 B.R. Donald, C.G. Levey, I. Paprotny: Planar microassembly by parallel actuator of MEMS microrobots, J. Microelectromech. Syst. **17**(4), 789–808 (2008)
53.234 C. Pawashe, S. Floyd, M. Sitti: Assembly and disassembly of magnetic mobile micro-robots towards 2-D reconfigurable micro-systems, Int. Symp. Robotics Res. (2009)
53.235 N. Napp, S. Burden, E. Klavins: The statistical dynamics of programmed self-assembly, IEEE Int. Conf. Robotics Autom. (ICRA) (2006) pp. 1469–1476
53.236 K. Gilpin, D. Rus: A distributed algorithm for 2-D shape duplication with smart pebble robots, IEEE Int. Conf. Robotics Autom. (ICRA) (2012)
53.237 K. Gilpin, A. Knaian, D. Rus: Robot pebbles: One centimeter robotic modules for programmable matter through self-disassembly, IEEE Int. Conf. Robotics Autom. (ICRA) (2010)
53.238 L.E. Parker: The effect of heterogeneity in teams of 100+ mobile robots. In: *Multi-Robot Systems Volume II: From Swarms to Intelligent Automata*, ed. by A. Schultz, L.E. Parker, F. Schneider (Kluwer, Dordrecht 2003)
53.239 T. Balch: Hierarchic social entropy: An information theoretic measure of robot team diversity, Auton. Robots **8**(3), 209–238 (2000)
53.240 D. Jung, A. Zelinsky: Grounded symbolic communication between heterogeneous cooperating robots, Auton. Robots **8**(3), 269–292 (2000)
53.241 R.R. Murphy: Marsupial robots for urban search and rescue, IEEE Intell. Syst. **15**(2), 14–19 (2000)
53.242 G. Sukhatme, J.F. Montgomery, R.T. Vaughan: Experiments with cooperative aerial-ground robots. In: *Robot Teams: From Diversity to Polymorphism*, ed. by T. Balch, L.E. Parker (A K Peters, Natick 2002)
53.243 L.E. Parker, B. Kannan, F. Tang, M. Bailey: Tightly-coupled navigation assistance in heterogeneous multi-robot teams, Proc. IEEE Int. Conf. Intell. Robots Syst. (2004)
53.244 A. Howard, L.E. Parker, G.S. Sukhatme: Experiments with a large heterogeneous mobile robot team: Exploration, mapping, deployment, and detection, Int. J. Robotics Res. **25**, 431–447 (2006)

53.245 L. Chaimowicz, B. Grocholsky, J.F. Keller, V. Kumar, C.J. Taylor: Experiments in multirobot air-ground coordination, Proc. IEEE Int. Conf. Robotics Autom. (ICRA) (2004)

53.246 L.E. Parker, F. Tang: Building multi-robot coalitions through automated task solution synthesis, Proc. IEEE **94**(7), 1289–1305 (2006)

53.247 R. Zlot, A. Stentz: Market-based multirobot coordination for complex tasks, Int. J. Robotics Res. **25**(1), 73–101 (2006)

53.248 J. Jennings, C. Kirkwood-Watts: Distributed mobile robotics by the method of dynamic teams. In: *Distributed Autonomous Robotic Systems 3*, ed. by T. Lueth, R. Dillmann, P. Dario, H. Wörn (Springer, Berlin, Heidelberg 1998)

53.249 E. Pagello, A. D'Angelo, E. Menegatti: Cooperation issues and distributed sensing for multirobot systems, Proc. IEEE **94**, 1370–1383 (2006)

53.250 B. Gerkey, M.J. Matarić: A formal analysis and taxonomy of task allocation in multi-robot systems, Int. J. Robotics Res. **23**(9), 939–954 (2004)

53.251 D. Gale: *The Theory of Linear Economic Models* (McGraw-Hill, New York 1960)

53.252 E. Balas, M.W. Padberg: On the set-covering problem, Oper. Res. **20**(6), 1152–1161 (1972)

53.253 B.B. Werger, M.J. Matarić: Broadcast of local eligibility for multi-target observation. In: *Distributed Autonomous Robotic Systems 4*, ed. by L.E. Parker, G. Bekey, J. Barhen (Springer, Tokyo 2000)

53.254 R.A. Brooks: A robust layered control system for a mobile robot, IEEE J. Robotics Autom. **RA-2**(1), 14–23 (1986)

53.255 S. Botelho, R. Alami: M+: A scheme for multi-robot cooperation through negotiated task allocation and achievement, Proc. IEEE Int. Conf. Robotics Autom. (1999) pp. 1234–1239

53.256 R.G. Smith: The contract net protocol: High-level communication and control in a distributed problem solver, IEEE Trans. Comput. **C-29**(12), 1104–1113 (1980)

53.257 B. Dias, R. Zlot, N. Kalra, A. Stentz: Market-based multirobot coordination: A survey and analysis, Proc. IEEE **94**(7), 1257–1270 (2006)

53.258 B.P. Gerkey, M.J. Matarić: Sold! Auction methods for multi-robot coordination, IEEE Trans. Robotics Autom. **18**(5), 758–768 (2002)

53.259 H. Kose, U. Tatlidede, C. Mericli, K. Kaplan, H.L. Akin: Q-learning based market-driven multi-agent collaboration in robot soccer, Proc. Turk. Symp. Artif. Intell. Neural Netw. (2004) pp. 219–228

53.260 D. Vail, M. Veloso: Multi-robot dynamic role assignment and coordination through shared potential fields, multi-robot systems: From swarms to intelligent automata, Proc. Int. Workshop Multi-Robot Syst. (2003) pp. 87–98

53.261 M. Lagoudakis, E. Markakis, D. Kempe, P. Keshinocak, A. Kleywegt, S. Koenig, C. Tovey, A. Meyerson, S. Jain: *Auction-based multi-robot routing*, *Robotics: Science and Systems I* (MIT Press, Cambridge 2005)

53.262 G. Rabideau, T. Estlin, S. Schien, A. Barrett: A comparison of coordinated planning methods for cooperating rovers, Proc. AIAA Space Technol. Conf. (1999)

53.263 R. Zlot, A. Stentz, M.B. Dias, S. Thayer: Multi-robot exploration controlled by a market economy, Proc. IEEE Int. Conf. Robotics Autom. (ICRA) (2002) pp. 3016–3023

53.264 J. Guerrero, G. Oliver: Multi-robot task allocation strategies using auction-like mechanisms, Proc. 6th Congr. Catalan Assoc. Artif. Intell. (2003) pp. 111–122

53.265 N. Kalra, D. Ferguson, A. Stentz: Hoplites: A market-based framework for planned tight coordination in multirobot teams, Proc. IEEE Int. Conf. Robotics Autom. (ICRA) (2005)

53.266 L. Lin, Z. Zheng: Combinatorial bids based multi-robot task allocation method, Proc. IEEE Int. Conf. Robotics. Autom. (ICRA) (2005) pp. 1145–1150

53.267 C.-H. Fua, S.S. Ge: COBOS: Cooperative backoff adaptive scheme for multirobot task allocation, IEEE Trans. Robotics **21**(6), 1168–1178 (2005)

53.268 E.G. Jones, B. Browning, M.B. Dias, B. Argall, M. Veloso, A. Stentz: Dynamically formed heterogeneous robot teams performing tightly-coupled tasks, Proc. IEEE Int. Conf. Robotics Autom. (ICRA) (2006) pp. 570–575

53.269 Y. Zhang, L.E. Parker: IQ-ASyMTRe: Forming executable coalitions for tightly coupled multirobot tasks, IEEE Trans. Robotics **29**(2), 400–416 (2012)

53.270 L. Vig, J.A. Adams: Multi-robot coalition formation, IEEE Trans. Robotics **22**(4), 637–649 (2006)

53.271 M. Bowling, M. Veloso: Simultaneous adversarial multi-robot learning, Proc. Int. Joint Conf. Artif. Intell. (2003)

53.272 F. Fernandez, L.E. Parker: A reinforcement learning algorithm in cooperative multi-robot domains, J. Intell. Robots Syst. **43**, 161–174 (2005)

53.273 C.F. Touzet: Robot awareness in cooperative mobile robot learning, Auton. Robots **2**, 1–13 (2000)

53.274 R. Steeb, S. Cammarata, F. Hayes-Roth, P. Thorndyke, R. Wesson: *Distributed Intelligence for Air Fleet Control*, Rand Corp. Tech. Rep. R-2728-AFPA (1981)

53.275 M. Benda, V. Jagannathan, R. Dodhiawalla: On Optimal Cooperation of Knowledge Sources, Boeing AI Center Tech. Rep. BCS-G2010-28 (1985)

53.276 T. Haynes, S. Sen: Evolving behavioral strategies in predators and prey. In: *Adaptation and Learning in Multi-Agent Systems*, ed. by G. Weiss, S. Sen (Springer, Berlin, Heidelberg 1986) pp. 113–126

53.277 S. Mahadevan, J. Connell: Automatic programming of behavior-based robots using reinforcement learning, Proc. AAAI (1991) pp. 8–14

53.278 S. Marsella, J. Adibi, Y. Al-Onaizan, G. Kaminka, I. Muslea, M. Tambe: On being a teammate: Experiences acquired in the design of RoboCup teams, Proc. 3rd Annu. Conf. Auton. Agents (1999) pp. 221–227

53.279 J. Pugh, A. Martinoli: Multi-robot learning with particle swarm optimization, Proc. 5th Int. Jt.

Conf. Auton. Agents Multiagent Syst. (2006) pp. 441–448
53.280 P. Stone, M. Veloso: Multiagent systems: A survey from a machine learning perspective, Auton. Robots **8**(3), 345–383 (2000)
53.281 M. Asada, E. Uchibe, S. Noda, S. Tawaratsumida, K. Hosoda: Coordination of multiple behaviors acquired by a vision-based reinforcement learning, Proc. IEEE/RSJ/GI Int. Conf. Intell. Robots Syst. (1994) pp. 917–924
53.282 M. Kubo, Y. Kakazu: Learning coordinated motions in a competition for food between ant colonies, Proc. 3rd Int. Conf. Simul. Adapt. Behav. (1994) pp. 487–492
53.283 R. Alami, S. Fleury, M. Herrb, F. Ingrand, F. Robert: Multi-robot cooperation in the MARTHA project, IEEE Robotics Autom. Mag. **5**(1), 36–47 (1998)
53.284 A. Stroupe, A. Okon, M. Robinson, T. Huntsberger, H. Aghazarian, E. Baumgartner: Sustainable cooperative robotic technologies for human and robotic outpost infrastructure construction and maintenance, Auton. Robots **20**(2), 113–123 (2006)
53.285 H.R. Everett, R.T. Laird, D.M. Carroll, G.A. Gilbreath, T.A. Heath-Pastore, R.S. Inderieden, T. Tran, K.J. Grant, D.M. Jaffee: *Multiple resource host architecture (MRHA) for the mobile detection assessment response system* (MDARS), SPAWAR Systems Technical Documen 3026, Revision A (2000)
53.286 Y. Guo, L.E. Parker, R. Madhavan: Towards collaborative robots for infrastructure security applications, Proc. Int. Symp. Collab. Technol. Syst. (2004) pp. 235–240
53.287 C. Hazard, P.R. Wurman, R. D'Andrea: Alphabet Soup: A testbed for studying resource allocation in multi-vehicle systems, Proc. AAAI Workshop Auction Mech. Robot Coord. (2006) pp. 23–30
53.288 D. Nardi, A. Farinelli, L. Iocchi: Multirobot systems: A classification focused on coordination, IEEE Trans. Syst. Man Cybern. B **34**(5), 2015–2028 (2004)
53.289 K. Passino: Biomimicry of bacterial foraging for distributed optimization and control, IEEE Control Syst. Mag. **22**(3), 52–67 (2002)
53.290 M. Schneider-Fontan, M. Matarić: Territorial multi-robot task division, IEEE Trans. Robotics Autom. **15**(5), 815–822 (1998)
53.291 I. Wagner, M. Lindenbaum, A.M. Bruckstein: Mac vs. PC – Determinism and randomness as complementary approaches to robotic exploration of continuous unknown domains, Int. J. Robotics Res. **19**(1), 12–31 (2000)
53.292 K. Sugawara, M. Sano: Cooperative behavior of interacting simple robots in a clockface arranged foraging field. In: *Distributed Autonomous Robotic Systems*, ed. by H. Asama, T. Arai, T. Fukuda, T. Hasegawa (Springer, Berlin, Heidelberg 2002)
53.293 P. Rybski, S. Stoeter, C. Wyman, M. Gini: A cooperative multi-robot approach to the mapping and exploration of Mars, Proc. AAAI/IAAI (1997)
53.294 S. Sun, D. Lee, K. Sim: Artificial immune-based swarm behaviors of distributed autonomous robotic systems, Proc. IEEE Int. Conf. Robotics Autom. (2001) pp. 3993–3998
53.295 A.I. Mourikis, S.I. Roumeliotis: Optimal sensor scheduling for resource-constrained localization of mobile robot formations, IEEE Trans. Robotics **22**(5), 917–931 (2006)
53.296 S. Kloder, S. Hutchinson: Path planning for permutation-invariant multirobot formations, IEEE Trans. Robotics **22**(4), 650–665 (2006)
53.297 C.W. Reynolds: Flocks, herds and schools: A distributed behavioral model, ACM SIGGRAPH Comput. Gr. **21**, 25–34 (1987)
53.298 T. Balch, R. Arkin: Behavior-based formation control for multi-robot teams, IEEE Trans. Robotics Autom. **14**(6), 926–939 (1998)
53.299 A. Jadbabaie, J. Lin, A.S. Morse: Coordination of groups of mobile autonomous agents using nearest neighbor rules, IEEE Trans. Autom. Control **48**(6), 988–1001 (2002)
53.300 C. Belta, V. Kumar: Abstraction and control for groups of robots, IEEE Trans. Robotics **20**(5), 865–875 (2004)
53.301 C.M. Topaz, A.L. Bertozzi: Swarming patterns in two-dimensional kinematic model for biological groups, SIAM J. Appl. Math. **65**(1), 152–174 (2004)
53.302 J.A. Fax, R.M. Murray: Information flow and cooperative control of vehicle formations, IEEE Trans. Autom. Control **49**(9), 1465–1476 (2004)
53.303 J.A. Marshall, M.E. Broucke, B.R. Francis: Formations of vehicles in cyclic pursuit, IEEE Trans. Autom. Control **49**(11), 1963–1974 (2004)
53.304 S.S. Ge, C.-H. Fua: Queues and artificial potential trenches for multirobot formations, IEEE Trans. Robotics **21**(4), 646–656 (2005)
53.305 P. Tabuada, G. Pappas, P. Lima: Motion feasibility of multi-agent formations, IEEE Trans. Robotics **21**(3), 387–392 (2005)
53.306 G. Antonelli, S. Chiaverini: Kinematic control of platoons of autonomous vehicles, IEEE Trans. Robotics **22**(6), 1285–1292 (2006)
53.307 J. Fredslund, M.J. Matarić: A general algorithm for robot formations using local sensing and minimal communication, IEEE Trans. Robotics Autom. **18**(5), 837–846 (2002)
53.308 Y. Cao, W. Yu, W. Ren, G. Chen: An overview of recent progress in the study of distributed multi-agent coordination, IEEE Trans. Ind. Inf. **9**(1), 427–438 (2012)
53.309 R.M. Murray: Recent research in cooperative control of multivehicle systems, J. Dyn. Syst. Meas. Control **129**(5), 571–583 (2007)
53.310 P.J. Johnson, J.S. Bay: Distributed control of simulated autonomous mobile robot collectives in payload transportation, Auton. Robots **2**(1), 43–63 (1995)
53.311 Z. Wang, E. Nakano, T. Matsukawa: Realizing cooperative object manipulation using multiple behaviour-based robots, Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS) (1996) pp. 310–317

53.312 K. Kosuge, T. Oosumi: Decentralized control of multiple robots handling an object, Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS) (1996) pp. 318–323

53.313 N. Miyata, J. Ota, Y. Aiyama, J. Sasaki, T. Arai: Cooperative transport system with regrasping car-like mobile robots, Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS) (1997) pp. 1754–1761

53.314 C.R. Kube, E. Bonabeau: Cooperative transport by ants and robots, Robotics Auton. Syst. **30**(1), 85–101 (2000)

53.315 R. Groß, M. Dorigo: Towards group transport by swarms of robots, Int. J. Bio-Inspired Comput. **1**(1), 1–13 (2009)

53.316 L.E. Parker: ALLIANCE: An architecture for fault tolerant, cooperative control of heterogeneous mobile robots, Proc. IEEE/RSJ/GI Int. Conf. Intell. Robots Syst. (1994) pp. 776–783

53.317 B. Donald, J. Jennings, D. Rus: Analyzing teams of cooperating mobile robots, Proc. IEEE Int. Conf. Robotics Autom. (1994) pp. 1896–1903

53.318 Y. Mohan, S.G. Ponnambalam: An extensive review of research in swarm robotics, World Congr. Nat. Biol. Insp. Comput. (2009) pp. 140–145

53.319 A.J. Ijspeert, A. Martinoli, A. Billard, L.M. Gambardella: Collaboration through the exploitation of local interactions in autonomous collective robotics: The stick pulling experiment, Auton. Robots **11**(2), 149–171 (2001)

53.320 A. Martinoli, K. Easton, W. Agassounon: Modeling swarm robotic systems: A case study in collaborative distributed manipulation, Int. J. Robotics Res. **23**(4-5), 415–436 (2004)

53.321 S. Berman, Q. Lindsey, M.S. Sakar, V. Kumar, S.C. Pratt: Experimental study and modeling of group retrieval in ants as an approach to collective transport in swarm robotic systems, Proc. IEEE **99**(9), 1470–1481 (2011)

53.322 J.M. Esposito: Distributed grasp synthesis for swarm manipulation with applications to autonomous tugboats, IEEE Int. Conf. Robotics Autom. (2008) pp. 1489–1494

53.323 Z. Wang, V. Kumar: Object closure and manipulation by multiple cooperating mobile robots, Proc. IEEE Int. Conf. Robotics Autom. (ICRA) (2002) pp. 394–399

53.324 Z. Wang, Y. Hirata, K. Kosuge: Control a rigid caging formation for cooperative object transportation by multiple mobile robots, Proc. of IEEE Int. Conf. Robotics Autom. (ICRA) (2004) pp. 1580–1585

53.325 J. Fink, N. Michael, V. Kumar: Composition of vector fields for multi-robot manipulation via caging, Robotics Sci. Syst. Conf. (2007)

53.326 F. Arrichiello, H.K. Heidarsson, S. Chiaverini, G.S. Sukhatme: Cooperative caging and transport using autonomous aquatic surface vehicles, Intell. Serv. Robotics **5**(1), 73–87 (2012)

53.327 J. Werfel: Building patterned structures with robot swarms, Proc. 19th Int. Jt. Conf. Artif. Intell. (IJCAI) (2005) pp. 1495–1502

53.328 J. Werfel, Y. Bar-Yam, D. Rus, R. Nagpal: Distributed construction by mobile robots with enhanced building blocks, IEEE Int. Conf. Robotics Autom. (2006) pp. 2787–2794

53.329 J. Werfel, R. Nagpal: Extended stigmergy in collective construction, IEEE Intell. Syst. **21**(2), 20–28 (2006)

53.330 J. Werfel, R. Nagpal: Three-dimensional construction with mobile robots and modular blocks, Int. J. Robotics Res. **27**(3/4), 463–479 (2008)

53.331 Y. Terada, S. Murata: Automatic modular assembly system and its distributed control, Int. J. Robotics Res. **27**, 445–462 (2008)

53.332 J. Wawerla, G.S. Sukhatme, M.J. Mataric: Collective construction with multiple robots, Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS) (2002) pp. 2696–2701

53.333 R.L. Stewart, R.A. Russell: Building a loose wall structure with a robotic swarm using a spatio-temporal varying template, Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS) (2004) pp. 712–716

53.334 R.L. Stewart, R.A. Russell: A distributed feedback mechanism to regulate wall construction by a robotic swarm, Adapt. Behav. **14**(1), 21–51 (2006)

53.335 T. Huntsberger, G. Rodriguez, P. Schenker: Robotics challenges for robotic and human Mars exploration, Proc. Robotics 2000 (2000) pp. 340–346

53.336 R.A. Brooks, P. Maes, M.J. Mataric, G. More: Lunar based construction robots, Proc. IEEE Int. Workshop Intell. Robots Syst. (IROS) (1990) pp. 389–392

53.337 C.A.C. Parker, H. Zhang, C.R. Kube: Blind bulldozing: Multiple robot nest construction, Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS) (2003) pp. 2010–2015

53.338 S. Sen, M. Sekaran, J. Hale: Learning to coordinate without sharing information, Proc. AAAI (1994) pp. 426–431

53.339 B. Tung, L. Kleinrock: Distributed control methods, Proc. 2nd Int. Symp. High Perform. Distrib. Comput. (1993) pp. 206–215

53.340 Z.-D. Wang, E. Nakano, T. Matsukawa: Cooperating multiple behavior-based robots for object manipulation, Proc. IEEE/RSJ/GI Int. Conf. Intell. Robots Syst. (IROS) (1994) pp. 1524–1531

53.341 D. Rus, B. Donald, J. Jennings: Moving furniture with teams of autonomous robots, Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (1995) pp. 235–242

53.342 F. Hara, Y. Yasui, T. Aritake: A kinematic analysis of locomotive cooperation for two mobile robots along a general wavy road, Proc. IEEE Int. Conf. Robotics Autom. (1995) pp. 1197–1204

53.343 J. Sasaki, J. Ota, E. Yoshida, D. Kurabayashi, T. Arai: Cooperating grasping of a large object by multiple mobile robots, Proc. IEEE Int. Conf. Robotics Autom. (1995) pp. 1205–1210

53.344 C. Jones, M.J. Matarić: Automatic synthesis of communication-based coordinated multi-robot systems, Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS) (2004) pp. 381–387

53.345 L.E. Parker: Cooperative robotics for multi-target observation, Intell. Autom. Soft Comput. **5**(1), 5–19 (1999)

53.346 A.W. Stroupe, M.C. Martin, T. Balch: Distributed sensor fusion for object position estimation by multi-robot systems, Proc. IEEE Int. Conf. Robotics Autom. (ICRA) (2001) pp. 1092–1098

53.347 S. Luke, K. Sullivan, L. Panait, G. Balan: Tunably decentralized algorithms for cooperative target observation, Proc. 4th Int. Jt. Conf. Auton. Agents Multiagent Syst. (2005) pp. 911–917

53.348 S.M. LaValle, H.H. Gonzalez-Banos, C. Becker, J.-C. Latombe: Motion strategies for maintaining visibility of a moving target, Proc. IEEE Int. Conf. Robotics Autom. (1997) pp. 731–736

53.349 T.H. Chung, J.W. Burdick, R.M. Murray: A decentralized motion coordination strategy for dynamic target tracking, Proc. IEEE Int. Conf. Robotics Autom. (ICRA) (2006) pp. 2416–2422

53.350 A. Kolling, S. Carpin: Multirobot cooperation for surveillance of multiple moving targets – A new behavioral approach, Proc. IEEE Int. Conf. Robotics Autom. (ICRA) (2006) pp. 1311–1316

53.351 B. Jung, G. Sukhatme: Tracking targets using multiple mobile robots: The effect of environment occlusion, Auton. Robots **13**(3), 191–205 (2002)

53.352 Z. Tang, U. Ozguner: Motion planning for multitarget surveillance with mobile sensor agents, IEEE Trans. Robotics **21**(5), 898–908 (2005)

53.353 M.A. Vieira, R. Govindan, G.S. Sukhatme: Scalable and practical pursuit-evasion with networked robots, Intell. Serv. Robotics **2**(4), 247–263 (2009)

53.354 L.E. Parker: Path planning and motion coordination in multiple mobile robot teams. In: *Encyclopedia of Complexity and System Science*, ed. by R.A. Meyers (Springer, Berlin, Heidelberg 2009)

53.355 D. Grossman: Traffic control of multiple robot vehicles, IEEE J. Robotics Autom. **4**, 491–497 (1988)

53.356 P. Caloud, W. Choi, J.-C. Latombe, C. Le Pape, M. Yim: Indoor automation with many mobile robots, IEEE Int. Workshop Intell. Robots Syst. (IROS) (1990) pp. 67–72

53.357 H. Asama, K. Ozaki, H. Itakura, A. Matsumoto, Y. Ishida, I. Endo: Collision avoidance among multiple mobile robots based on rules and communication, Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS) (1991) pp. 1215–1220

53.358 S. Yuta, S. Premvuti: Coordinating autonomous and centralized decision making to achieve cooperative behaviors between multiple mobile robots, Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (1992) pp. 1566–1574

53.359 J. Wang: Fully distributed traffic control strategies for many-AGV systems, IEEE Int. Workshop Intell. Robots Syst. (1991) pp. 1199–1204

53.360 J. Wang, G. Beni: Distributed computing problems in cellular robotic systems, Proc. IEEE Int. Workshop Intell. Robots Syst. (1990) pp. 819–826

53.361 P. Svestka, M. Overmars: Coordinated path planning for multiple robots, Robotics Auton. Syst. **23**, 125–152 (1998)

53.362 M. Peasgood, C. Clark, J. McPhee: A complete and scalable strategy for coordinating multiple robots within roadmaps, IEEE Trans. Robotics **24**(2), 283–292 (2008)

53.363 M. Erdmann, T. Lozano-Perez: On multiple moving objects, Algorithmica **2**, 477–521 (1987)

53.364 C. Ferrari, E. Pagello, J. Ota, T. Arai: Multirobot motion coordination in space and time, Robotics Auton. Syst. **25**, 219–229 (1998)

53.365 M. Bennewitz, W. Burgard, S. Thrun: Finding and optimizing solvable priority schemes for decoupled path planning techniques for teams of mobiel robots, Robotics Auton. Syst. **41**(2), 89–99 (2002)

53.366 H. Kitano, M. Asada, Y. Kuniyoshi, I. Noda, E. Osawa, H. Matasubara: RoboCup: A challenge problem of AI, AI Magazine **18**(1), 73–86 (1997)

53.367 H. Kitano, S. Tadokoro: RoboCup rescue: A grand challenge for multiagent and intelligent systems, AI Magazine **22**(1), 39–52 (2001)

53.368 B. Browning, J. Bruce, M. Bowling, M. Veloso: STP: Skills, tactics and plays for multi-robot control in adversarial environments, IEEE J. Control Syst. Eng. **219**, 33–52 (2005)

53.369 M. Veloso, P. Stone, K. Han: The CMUnited-97 robotic soccer team: Perception and multiagent control, Robotics Auton. Syst. **29**(2/3), 133–143 (1999)

53.370 T. Weigel, J.-S. Gutmann, M. Dietl, A. Kleiner, B. Nebel: CS Freiburg: Coordinating robots for successful soccer playing, IEEE Trans. Robotics Autom. **5**(18), 685–699 (2002)

53.371 P. Stone, M. Veloso: Task decomposition, dynamic role assignemnt, and low-bandwidth communicaiton for real-time strategic teamwork, Artif. Intell. **110**(2), 241–273 (1999)

53.372 C. Candea, H.S. Hu, L. Iocchi, D. Nardi, M. Piaggio: Coordination in multi-agent Robocup teams, Robotics Auton. Syst. **36**(2), 67–86 (2001)

53.373 L.E. Parker: Current state of the art in distributed autonomous mobile robotics, Distrib. Auton. Robotic Syst. **4**, 3–12 (2000)

53.374 K.R. Baghaei, A. Agah: Task allocation and communication methodologies for multi-robot systems, Intell. Autom. Soft Comput. **9**, 217–226 (2003)

53.375 T. Balch, L.E. Parker: Guest editorial, special issue on heterogeneous multi-robot systems, Auton. Robots **8**(3), 207–208 (2000)

53.376 M. Dorigo, E. Sahin: Guest editorial, special issue on swarm robotics, Auton. Robots **17**(2/3), 111–113 (2004)

53.377 M. Veloso, D. Nardi: Special issue on multirobot systems, Proc. IEEE **94**, 1253–1256 (2006)

53.378 T. Balch: Taxonomies of multi-robot task and reward. In: *Robot Teams: From Diversity to Polymorphism*, ed. by T. Balch, L.E. Parker (A K Peters, Natick 2002)

53.379 L.E. Parker, G. Bekey, J. Barhen (Eds.): *Distributed Autonomous Robotic Systems 4* (Springer, Berlin, Heidelberg 2000)

53.380 H. Asama, T. Arai, T. Fukuda, T. Hasegawa (Eds.): *Distributed Autonomous Robotic Systems 5* (Springer, Berlin, Heidelberg 2002)
53.381 R. Alami, R. Chatila, H. Asama (Eds.): *Distributed Autonomous Robotic Systems 6* (Springer, Berlin, Heidelberg 2006)
53.382 M. Gini, R. Voyles (Eds.): *Distributed Autonomous Robotic Systems 7* (Springer, Berlin, Heidelberg 2006)
53.383 H. Asama, H. Kurokawa, J. Ota, K. Sekiyama: *Distributed Autonomous Robotic Systems 8* (Springer, Berlin, Heidelberg 2009)
53.384 A. Martinoli, F. Mondada, N. Correll, G. Mermoud, M. Egerstedt, M.A. Hsieh, L.E. Parker, K. Stoy (Eds.): *Distributed Autonomous Robotic Systems*, Springer Tracts in Advanced Robotics, Vol. 83 (Springer, Berlin, Heidelberg 2013)
53.385 A. Schultz, L.E. Parker (Eds.): *Multi-Robot Systems: From Swarms to Intelligent Automata* (Kluwer, Dordrecht 2002)
53.386 A. Schultz, L.E. Parker, F. Schneider (Eds.): *Multi-Robot Systems Volume II: From Swarms to Intelligent Automata* (Kluwer, Dordrecht 2003)
53.387 E. Sahin, W.M. Spears (Eds.): *Swarm Robotics: SAB 2004 Int. Workshop* (Springer, Berlin, Heidelberg 2004)
53.388 L.E. Parker, F. Schneider, A. Schultz (Eds.): *Multi-Robot Systems Volume III: From Swarms to Intelligent Automata* (Kluwer, Dordrecht 2005)
53.389 T. Balch, L.E. Parker (Eds.): *Robot Teams: From Polymorphism to Diversity* (A K Peters, Natick 2002)
53.390 F. Bullo, J. Cortés, S. Martinez: *Distributed Control of Robotic Networks: A Mathematical Approach to Motion Coordination Algorithms* (Princeton University Press, Princeton 2009)
53.391 S. Kernbach (Ed.): *Handbook of Collective Robotics: Fundamentals and Challenges* (CRC, Boca Raton 2013)