

Chapter 2

Background and Related Work

Abstract This chapter briefly reviews the background and the related work in swarm robotics from the perspective of self-sufficiency. It starts with swarm intelligence and provides few examples of swarm behaviors found in nature that serve as the source of inspiration for the advances in swarm robotics. In this regard, it presents the salient properties of robotic modules in a robot swarm and reviews the design of existing state-of-the-art re-configurable robotic platforms that support power sharing. It then introduces the design features of heterogenous REPLICATOR re-configurable robotic modules. The chapter then covers some of the bio-inspired techniques applied in swarm robotics to gain energy autonomy and briefly introduces the fault tolerance, in the context. And lastly, it briefly describes the simulation tools used in this research work.

2.1 Swarm Intelligence

The word “swarm” is usually designated to a population of apparently disorganized moving individuals that has the tendency to form clusters for different purposes without any centralized control. In nature, such a behavior, the purposeful collaboration of autonomous individuals in a group, can be seen in different species, e.g., ant colonies, bees, birds flocking, fish schooling, bacterial growth, etc. To study the collective behavior of apparently simple but complex individuals, Beni and Wang (1989) were the first who used the term “swarm intelligence” in their work of cellular robotic systems. *Swarm intelligence* focuses on the emergent collective behavior of independent individuals that results from the local interaction of group or swarm members among each other and with the environment without external supervision. Mark (1994), who developed swarm models for application in artificial life has formulated five basic properties of swarm intelligence:

- **Proximity principle:** the swarm members should be able to carry out simple computations concerning space and time,
- **Quality principle:** the swarm members should be able to evaluate the interactions between themselves and with the environment,

- **Diverse response principle:** the interactions among the swarm members must be diverse and distributed instead of concentrated on too narrow behavior,
- **Stability principle:** the swarm members in a swarm are generally less prone to repetitive fluctuations in the environment and therefore their behavior does not oscillate,
- **Adaptability principle:** the swarm is sensitive to rapid changes happening in the environment that may require a change in its behavior.

Bonabeau et al. (1999) defines the term swarm intelligence as,

“any attempt to design algorithms or distributed problem-solving devices inspired by the collective behavior of social insect colonies and other animal societies.”

Later, Bonabeau and Meyer (2001) further explained the term swarm intelligence as,

“Social insects work without supervision. Their teamwork is largely self-organized, and coordination arises from the different interactions among the individuals of the colony. Although these interactions might be primitive, for instance, one ant merely following the trail left by another, taken together they lead to efficient solutions to difficult problems, such as, finding the shortest route to a food source among myriad possible paths. The collective behavior that emerges from a group of social insects has been dubbed swarm intelligence.”

The swarm intelligence principles help to understand and model complex systems through simple rules that are then applied to solve problems in different areas. For instance, the “ant colony optimization” (ACO) that was initially proposed by Dorigo and Caro (1999), Dorigo et al. (1991, 1996). It is a meta-heuristic optimization algorithm that is applied to find approximate solutions to difficult combinatorial optimization problems that can be reduced to finding optimal paths through graphs. The ACO technique mimics the pheromone laying phenomenon found in natural ant colonies to build solutions by moving the artificial ants on the problem graph so the future artificial ants can build better solutions. This is done in an iterative process where the good solutions found by the ants of an iteration should guide the ants of the following iterations. Another related meta-heuristic method, which was introduced by Kennedy and Eberhart (1995), is known as “particle swarm optimization” (PSO). PSO is inspired by social behavior of birds in a flock and fish schooling and optimizes the problem by moving a population of possible or candidate solutions, also called as particles, in the search space. The movement of these particles is then directed to the best particles (evolved over time) found in the search space. It is applied in a variety of continuous optimization problems such as, training of artificial neural networks (ANN), finite element updating, etc. Other commonly known methods include, “stochastic diffusion search” (SDS) by Bishop (1989), and “river formation dynamics” (RFD) by Pablo et al. (2007).

2.1.1 Social Insects

The behavior of ants in social insect colonies is still an inspiring and most popular model in swarm intelligence. Wheeler (1911) described the social insect colony as an organism or as a higher-level organism or a super-organism because of the degree which the individuals appear to operate as a unit, coordinated function and physiology, dedicated to the perpetuation and reproduction of the colony as a whole. Wilson and Hölldobler (1988) further described the behavior of social insect colonies as an integrated unit that possesses the ability to process a large amount of information in a non-centralized manner. The colony members apparently make collective and individual decisions, e.g., task distribution, coordinate the activities of tens to thousands of workers, etc., and exhibits flexibility and robustness in response to external, i.e., environmental, challenges and internal perturbations.

Social insect colonies differ from each other in many ways. Some have only few individuals, whereas some include thousands or even millions of individuals. In some species, the individuals in a colony are short lived or seasonal. In others, they may live for many years (Queller and Strassmann 1998). In a colony, since the workers can perform a variety of tasks, different tasks are dynamically and distributively allocated to swarm or colony members, such as foraging, defense, nest construction, etc. Figure 2.1 shows one such example in which a group of ants collectively build a nest.



Fig. 2.1 Swarming: *Oecophylla smaragdina* workers are collaborating in nest construction, source <http://upload.wikimedia.org/wikipedia/commons/a/aa/SSL11903p.jpg>, by Sean.hoyland, used under license Wikimedia Commons/Desaturated from original, last accessed: 26 Aug. 2013

In literature different theories have been proposed to understand “how do the individuals in a colony or group interact with each other to develop a collective collaborative behavior?” “How does natural selection favor this kind of cooperation?”. In his work of task allocation in social insect colonies, Gordon (1996) pointed out two kinds of factors that determine when and what task an individual worker performs: “internal” and “external” factors. The *internal factors* that are based on attributes of every individual in the colony are often considered to be fixed, i.e., body size, age and genetic factors. Whereas, the *external factors* are based on environmental stimuli and are therefore considered to be dynamic in nature. For example, when to go for foraging, a honeybee’s decision might depend on how much nectar is already collected in the nest. Summarily, the actions of an individual are strongly influenced by at least two types of external cues: actions of other individuals and events happening in the colony’s environment.

In the past couple of decades, different types of swarm or collective behaviors of different natural species were thoroughly investigated to solve or address a variety of problems in different domains, e.g., data mining, optimization, network routing, clustering, scheduling, traveling salesman problem, signal processing, etc. One characteristic feature of social insects, explored in recent years and linked with their foraging behavior is their ability to “organize” themselves in different groups or forms/shapes. This autonomous organization into multiple groups and subgroups involves mutual collaboration and collective efforts, that helps the swarm in several ways, e.g., dynamic task sharing or distribution, improved defense mechanism, better foraging, etc.

Self-organization is a process where a global structure emerges from the local interactions of a group of autonomous individuals without any centralized control mechanism. Bonabeau et al. (1999) defines four basic ingredients of self-organization: positive feedback, negative feedback, balance of exploitation and exploration, and multiple interactions. *Positive feedback* generally promotes the change by amplifying the fluctuations in the system, e.g., the intensity of pheromone laid by several individuals on a successful route. In contrast, *negative feedback* allows a system to tune its response in such a way that reduces changes. According to Garnier et al. (2007), self-organization requires multiple direct or indirect (stigmergy) local interactions among the swarm members to produce deterministic outcomes, and the appearance of large and viable structures.

2.1.2 Swarm Behaviors

Swarm behavior or intelligence is not just limited to social insect colonies. Rather, it can be seen in other species that show similar collaboration and cooperation in the environment, such as fish schools (Grünbaum et al. 2005; Parrish et al. 2002), bird flocks (Reynolds 1987), sheep herds (Gautrais et al. 2007), or even crowds of human beings (Helbing et al. 2001). These species form or arrange themselves in groups or clusters with changing surroundings and environmental conditions, e.g., food supply, change of weather, etc. The clustering of these species for different purposes in nature

evidently provide them several advantages, e.g., better navigation, improved defense against predators, improved or better foraging, etc.

● **Navigation**

In nature, because of changing seasons, every day hundreds and thousands of birds and other animals migrate or travel from one region to another mainly for their survival, e.g., in search of better food, shelter, etc. The most fascinating aspect of their journey lies in their highly coordinated movement without any centralized control mechanism. By obeying simple principles, a flock without any collision travels several hundreds of miles in its everyday life. This collective and coordinated movement of individuals in different sized clusters help them to navigate more efficiently and effectively through different environments. The distributive but collective movement of bird flocks displays a structural order or pattern. The collective actions are so integrated that they appear to move as a single coherent entity, that has the tendency to abruptly change its shape and direction (Couzin et al. 2002). According to Grosan et al. (2006), the main principles of the collective behavior that is observed in a bird flock include:

- *Homogeneity*
All the agents (birds) of a swarm must be of similar species so that they behave identically.
- *Locality*
The movement pattern of an agent can only be influenced by its neighboring flock mates. For this purpose, an active vision among the swarm members acts as a sensor for the flock organization.
- *Collision Avoidance*
By maintaining a sufficient distance with its neighboring flock mates an agent usually avoids collisions in the flock.
- *Velocity Matching*
It is a critical parameter that needs to be dynamically regulated individually and collectively to exhibit a controlled coordinated movement in an aggregation,
- *Flock Centering*
It is a behavior that helps the flock members to stay close to their neighbors.

With these simple rules the flock exhibit an extremely coordinated movement without any centralized control, which in result produces complex motion and interaction that would be extremely hard to create otherwise.

● **Defense**

A significant advantage gained by individuals during swarming is the improved defense mechanism against the potential threats in the environment. For instance, a herd is less vulnerable to predators than an individual.

● **Foraging**

“*Ants aren’t smart, but ant colonies are*” (Gordon 2007). Foraging is among the best examples of decentralized distributed coordination of multiple individuals, found in nature. A colony with simple rules develops the tendency to collectively solve complex problems, such as, finding the optimum path to the best food source,

distributing tasks among colony members, etc.

Foraging is an act of looking or searching for food in the environment individually and collectively for the individuals and group survival. For this purpose, ants use an indirect way of communication mechanism to communicate with the colony members in the environment. While foraging, the forager ant lays pheromone trails to the food source to direct the fellow foraging ants towards the same food source. These pheromone trails later directly and indirectly help the foragers in finding the shortest path to a particular food source and the nest, and thus maximizes the net energy efficiency (Goss et al. 1989). The *pheromone* is a chemical substance, secreted and deposited by a foraging ant on its way to the food source back to the nest. These pheromone trails are then smelled by the fellow foraging ants or newly recruited foragers that then leave their own pheromone trails on their way from food source to the nest. This way the intensity of the pheromone helps the foraging ants to follow an optimum route to the food source and back to the nest. This serves as a positive feedback for the sharing of information about the available food sources in the environment (Sumpter and Beekman 2003). In Fig. 2.2 a group of foraging ants eating a slice of an apple is shown as an example of collective foraging. As already stated, to address the computational problems, e.g., optimal



Fig. 2.2 Collective foraging behavior in an ant colony. Ants eating a slice of an apple, source http://upload.wikimedia.org/wikipedia/commons/c/ca/Ants_eating_fruit.jpg, by Zainichi Gaikokujiin, used under Creative Commons Attribution 3.0 Unported/Desaturated from original, last accessed: 2 Nov. 2013

path finding, data mining, etc., the ant colony optimization algorithm is one such example that is inspired from the foraging phenomenon in natural ant colonies. Johnson (1982) categorizes the foraging strategies into two groups. The “solitary” and “group” foraging. The *solitary foraging* is a strategy in which the animals locate, capture, and consume their prey—food—on their own. It is normally seen in the scenarios where the food resources are in abundance. An example of an exclusive solitary forager is the South American species of the harvester ant, *Pogonomyrmex vermiculatus* (Torres-Contreras et al. 2007). In *group foraging* the animals locate, capture and consume food collectively with other individuals. Therefore, a failure or success in gaining a food resource depends on the mutual behavior of the foragers.

- **Reproduction**

In honey bee colonies, swarming is the natural means of reproduction (Villa 2004). A new honey bee colony or beehive is formed when the queen bee leaves a colony with a large group of worker bees.

2.2 Swarm Robotics

In recent years, enormous research potential has been seen in mimicking the collaborative behavior of socialized animals into artificial systems. In the late 90’s, Beni and Wang (1989) used a population of simple robots, called “cellular robots” to explore and develop the principles of swarm intelligence. Later, Kube and Zhang (1993) used the term “collective robotics” in their work of collective robot intelligence. During recent years, the term *swarm robotics* has emerged as an application of swarm intelligence to multi-robot systems that focuses on physical abilities of the autonomous individuals and their interactions among each other and between the individuals and the environment. Şahin (2005) defined the term as,

“Swarm robotics is the study of how a large number of relatively simple physically embodied agents can be designed such that a desired collective behavior emerges from the local interactions among the agents and between the agents and the environment.”

In general, “swarm” or “collective robotic systems” is a new approach to the coordination of fairly large population of relatively simple robots (Dorigo and Şahin 2004). The approach takes its inspiration from the system-level functioning of social insects—ants, termites, bees and wasps—which demonstrate three essential characteristics for a multi-robotic system: *robustness, adaptability, and scalability*, as mentioned earlier.

- **Robustness** in a system can be defined as the degree to which a system can withstand abrupt internal and external perturbations. In other words, it’s an inherent ability of a system that allows it to continue operating in the presence of system failures or malfunctioning system components. For instance, the organization or distribution of workers in an ant colony continues operating even if they lost several members or if the environmental condition changes notably. The robustness in such

a system can be attributed to several factors; the degree of redundancy in the system, that allows a swarm to tolerate the presence of faults (Kazadi et al. 2004). That is, at macro level, how a colony compensates the loss of its members, e.g., by recruiting new foragers upon the loss of existing ones. The decentralized coordination of individuals, i.e., the lack of any centralized entity or control mechanism, makes them invulnerable to potential break downs in the system (Melhuish 2001).

- **Adaptability** is a system's characteristic, either innate or acquired or both, that enables it to learn and adapt its critical system parameters that effect its stability in a dynamic environment. Adaptability and robustness have partly conflicting definitions (Bayindir and Şahin 2007). The difference between the two arises with the change in environment. For example, to withstand an unexpected change in the surroundings in which the system operates, it should be able to learn and adapt the change for its long term survival. Like in nature, animals change their behavior with the changing climate, e.g., food supply, temperature, shelter, etc., round the year. Adaptability in biological systems can be classified as: structural, behavioral and physiological adaptations (Bayindir and Şahin 2007). *Structural adaptations* point towards the physical appearance of the organism that aid its survival in its natural habitat, like, skin, shape, body covering, teeth, etc. The *behavioral adaptations* are linked with the behavior of an organism in a particular environment, e.g., search for food or a better shelter, etc. And, the *physiological adaptations* help an organism to adapt and regulate its internal system functions, like, secreting lime, body temperature, or like in ants, secreting pheromones. In the context of swarm robotics, it refers to the ability of individuals to modify or change their behavior over time, changes depending on the dynamic environment, changes in the assigned task, changes in the system composition or capabilities, in such a way that the net performance of the entire system can either be improved or at least not degraded (Iocchi et al. 2001).
- **Scalability** can be defined as “*the ability to expand a self-organized mechanism to support larger number of individuals without impacting performance considerably*” (Bayindir and Şahin 2007). In the context of swarm robotics, it can be defined as the collective ability of modules to maintain their performance and efficiency irrespective of the swarm population.

Considering the taxonomy of swarm robotics, Dudek et al. (1993) have proposed a classification of a *robot swarm* in terms of swarm size, communication range, communication topology, communication bandwidth, swarm reconfigurability and swarm unit processing ability. Later, Cao et al. (1997) segregated swarm robotics into: *group architecture*, *resource conflicts*, *origins of cooperation*, *learning* and *geometric problems*. Group architecture is then further divided into centralized and decentralized, homogeneous or heterogeneous robots, communication structure and modeling of other agents, dimensions. The modeling contains an individual's abilities, e.g., sensing, communication, locomotion, etc., to obtain or evolve effective and efficient cooperation in a robot swarm. Iocchi et al. (2001) also presented a taxonomy of multi-robot systems with the focus on the communication and computation aspects. In their classification four different levels are individuated in detail:

a *cooperation level*, a *knowledge level*, a *coordination level*, and an *organization level*. The knowledge level contains “aware” and “unaware” categories. The *aware* category at coordination level is then further subdivided into: “strong”, “weak” and “non” coordinate categories. And finally, the strong coordinated category is further subdivided at organization level into: “strongly centralized”, “weakly centralized” and “distributed” categories.

According to Bayindir and Şahin (2007) the mathematical modeling of swarm robotic systems is generally classified into *sensor-based*, *microscopic*, *macroscopic* and *cellular automata modeling*. The *sensor-based* modeling approach uses the models of the sensors and actuators of a robotic system along with the objects in the environment, e.g., recharge stations, obstacles, etc., as components of the modeled system. It further models the interactions between a robot and its teammates and with the environment. The sensor-based modeling approach is generally carried out using *non-physical* and *physical* simulations. The non-physical simulation does not include the physical properties (dynamics) of the robots and the objects in the environment (Hayes and Dormiani-Tabatabaei 2002; Howard et al. 2002a, b; Trianni et al. 2002, 2003). Whereas, the latter approach models the interactions between the robots and with the environment based on the physical rules, such as, assigning mass and force required to move or drag an object in the environment (Bahgeçi and Sahin 2005; Soysal and Sahin 2005; Trianni and Dorigo 2005; Trianni et al. 2006). The *microscopic modeling* approach captures the dynamics of a robotic module during its interaction with its swarm-mates and with the objects in the environment, but the details about its sub-components, e.g., sensors, actuators, body pose, are abstracted. The behavior of a robotic module is modeled using a state diagram and the transition between the states is triggered using internal (inside a robot), and external (environmental) stimuli. In literature, the microscopic modeling approach has been applied in a variety of applications. Likewise, Ijspeert et al. (2001), Martinoli and Easton (2003), Martinoli et al. (2004) used a probabilistic microscopic modeling approach to model the stick pulling problem in a robotic swarm. Lui (2008) used the probabilistic microscopic modeling approach to model the collective foraging task in a robotic swarm. The *macroscopic modeling* approach differs from microscopic models in terms of its granularity. The macroscopic modeling approach focuses on the behavioral aspects of the whole system, such as a robot swarm. The system behavior is usually defined in terms of difference equations, where each system state represents the average number of robotic modules in a particular state at that instance of time (Bayindir and Şahin 2007). The fourth kind of modeling approach is *cellular automata*. It is the simplest form of mathematical models of complex systems (Ilachinski 2001). It consists of a regular grid/lattice of cells where each cell has a finite number of states, such as *on* and *off*. Each cell interacts only with its neighboring cells based on some fixed rules—generally a mathematical function. The interaction determines the new state of each cell from its current state and the states of its neighboring cells. The studies conducted by Shen et al. (2004) in the context of self-organization and distributed control of robotic swarms can be seen as an example of cellular automata modeling. Figure 2.3 shows the taxonomy of swarm robotics in literature proposed by Bayindir and Şahin (2007).

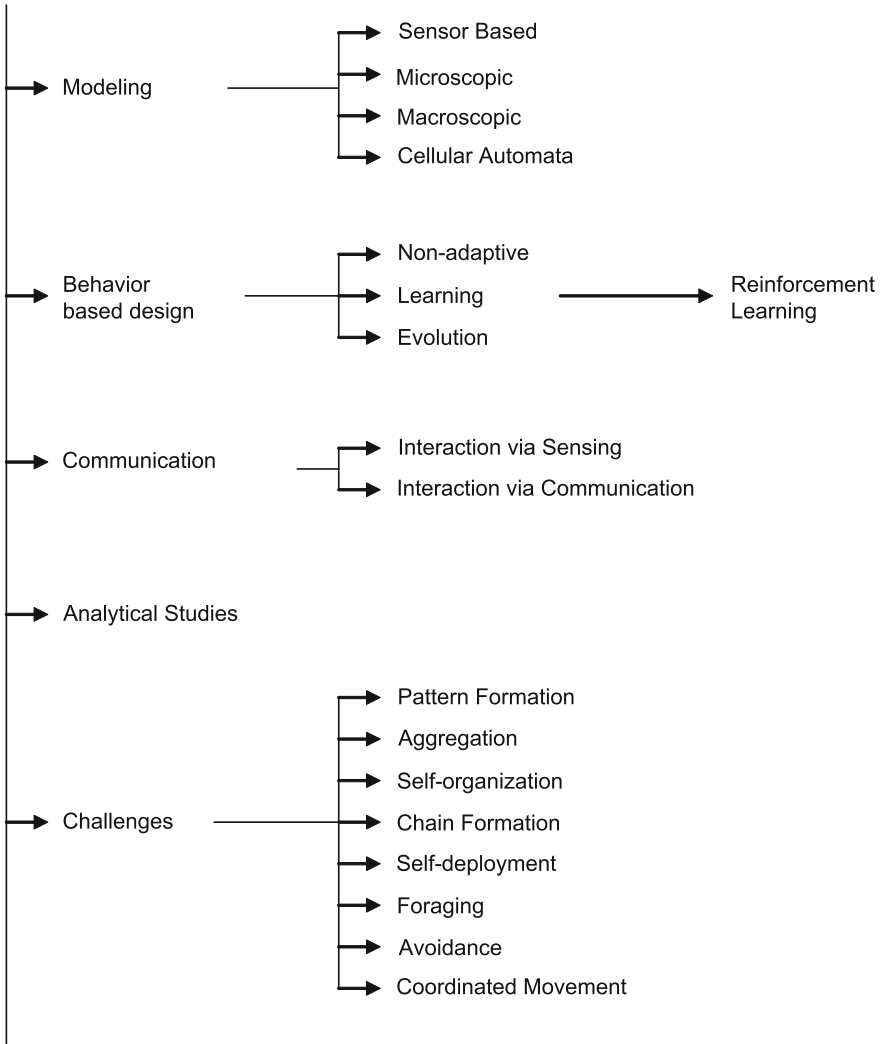


Fig. 2.3 Taxonomy of swarm robotics in literature (Bayindir and Şahin 2007)

The concept of modular robotics along with swarm intelligence principles has been explored in several applications using different platforms. These are broadly classified into two categories: *re-configurable with docking* and *re-configurable without docking*. The term “re-configurable with docking” capability is coined for those robotic platforms that own the ability to physically dock to each other either manually or autonomously and can adapt different morphologies for different purposes. The examples of such systems include PolyBot (Yim et al. 2000), SuperBot (Salemi et al. 2006), ODIN (Lyder et al. 2008), ATRON (Lund et al. 2005), Molecubes (Zykov et al.

2007), etc. Whereas, the second category of platforms though does own the modular structure but lacks the physical docking ability, for example, ePuck (Goncalves et al. 2009), Jasmine (Kornienko et al. 2005), etc. Hence, this category of robotic systems is only able to logically reconfigure themselves in different clusters.

In the context of this research work, the focus will be laid on different design parameters of only those re-configurable robotic systems that have the mechanical docking ability.

2.3 Modular Re-configurable Robotic Systems

The cells in a living creature may itself look quite simple but develop complex behaviors when aggregated in different morphologies—organisms (Fukuda and Nakagawa 1988). In recent years, creating simple robotic systems yet with the ability to adapt to the environment with complex morphing behavior gave birth to a new breed of robotic system known as *modular re-configurable robot*. A re-configurable robotic module with its innate ability of physical aggregation and dis-aggregation can dynamically adapt different morphologies. In the following section, two re-configurable robotic systems supporting some form of power management and energy sharing feature in their hardware design will be briefly described.

2.3.1 SuperBot

The SuperBot is a multi-functional, autonomous, re-configurable, homogeneous modular robot designed and developed by Salemi et al. (2006) at University of Southern California (USC). This modular robotic system was originally developed in collaboration with NASA for space exploration tasks. The salient features of a SuperBot robotic platform includes three degrees of freedom (DOF)—yaw, pitch, and roll—high-speed infra-red LEDs for communication, power sharing, re-configurable and flexible design enable the dynamic configuration of robotic modules to perform efficient locomotion and task manipulation. Figure 2.4a, b show a single SuperBot robotic module and a SuperBot modular robotic organism, respectively. The mechanical design of a SuperBot robotic module comprises two halves. One half of the robotic module contains a battery pack, a motor drive, a control mechanism for three docking sides, i.e., M1, M2, and M3, and a central controller called the “master controller”, i.e., 8-bit 16MHz Atmega128 micro-controller. The master controller is connected to the controller of the second half, called the “slave controller”.

The two controllers in every module are therefore responsible for managing the operations of actuators, communication links, sensors, power flow and the docking interfaces on each respective side. Figure 2.5 shows the block diagram of the hardware control architecture of a SuperBot robotic module. Each module has 6 docking

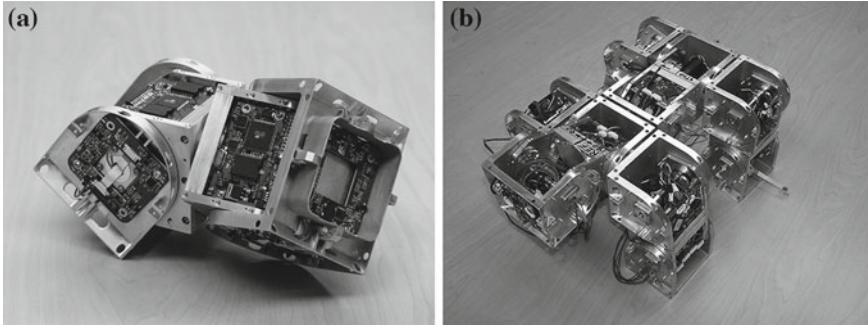


Fig. 2.4 Homogeneous SuperBot robotic system with a hybrid architecture (Salemi et al. 2006). **a** A single SuperBot robotic module. **b** SuperBot modular robotic organism

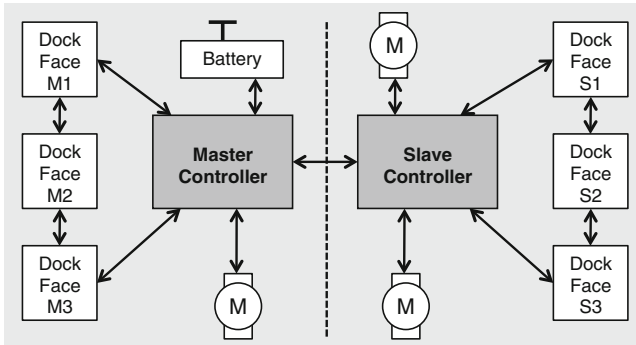


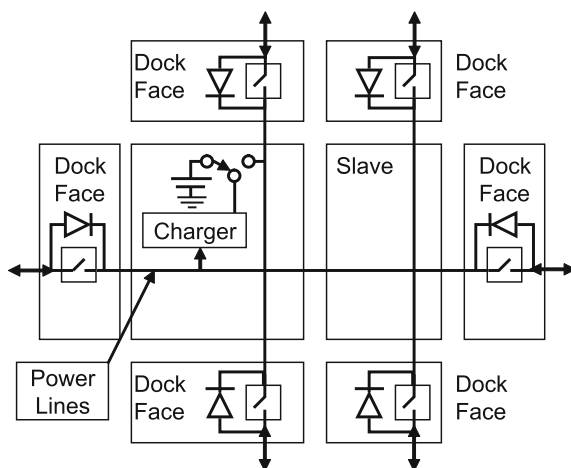
Fig. 2.5 Block diagram of the SuperBot hardware control architecture (Salemi et al. 2006)

connectors (sides) to allow the physical docking of other robotic modules in different configurations.

From the power management perspective, each robotic module is powered with two serially connected lithium polymer cells (2s1p) that provide a nominal voltage of 7.4 v with a charge capacity of 1600 mAh (Salemi et al. 2006). Figure 2.6 shows the SuperBot power sharing schematic. It includes 6 docking interfaces (3 on each half), and a recharging module. Each docking interface includes a diode that allows a uni-directional—inwards—current flow. Whereas, the outward current flow on each side is controlled by the controller with the help of a switch. Further, the local power bus interconnects the 6 docking interfaces thus allowing an omni-directional power flow in the system.

In the default system state the charging switch—the switch that connects the battery with the charging module—remains closed while the rest of the switches present on each docking interface remain open. The application of an external power source on any one of its docking interface, in the default configuration, allows the power to flow into the system’s recharging module, thus allowing the on-board battery

Fig. 2.6 SuperBot power sharing schematic (Salemi et al. 2006)



to recharge, instantly. This configuration of power bus switches in a modular robotic organism thus allows the turn-by-turn recharging of several robotic modules (in case, all the modules are dead) when any one of them is connected with an external power source. The power bus in a SuperBot modular robotic organism not only allows the recharging of robotic individuals through the external power source but also allows the modules to share their energy with each other in different configurations.

2.3.2 ATRON

The ATRON is a self-reconfigurable modular robot originally developed by the Adaptronics group at the University of Southern Denmark (USD) (Lund et al. 2005). Figure 2.7a shows an ATRON re-configurable robotic module. An ATRON robotic module consists of two hemispheres, joint together by a rotating mechanism that provides one DOF to the mechanical design. Each hemisphere has two passive female connectors and two active male connectors, which enable a module to hook to the complementary female connectors of the docking module (Østergaard 2004). Not only that, the two four-sided pyramids with cravng allow the free rotation of robotic modules within an ATRON organism. Figure 2.7b shows an ATRON modular robotic organism constructed from a set of homogeneous, broadly spherical modules, with each of them 11 cm in diameter. The split design architecture, allows every ATRON module to rotate its one hemisphere with respect to the other to generate the collective movement patterns in the organism.

From the perspective of the system's electronic design, each hemisphere of the ATRON module has one 8-bit 16MHz ATmega-128 micro-controller. Figure 2.8 shows the block diagram of electronics in an ATRON module. The northern hemisphere contains the main processor, an I/O processor, two docking connectors, a rota-

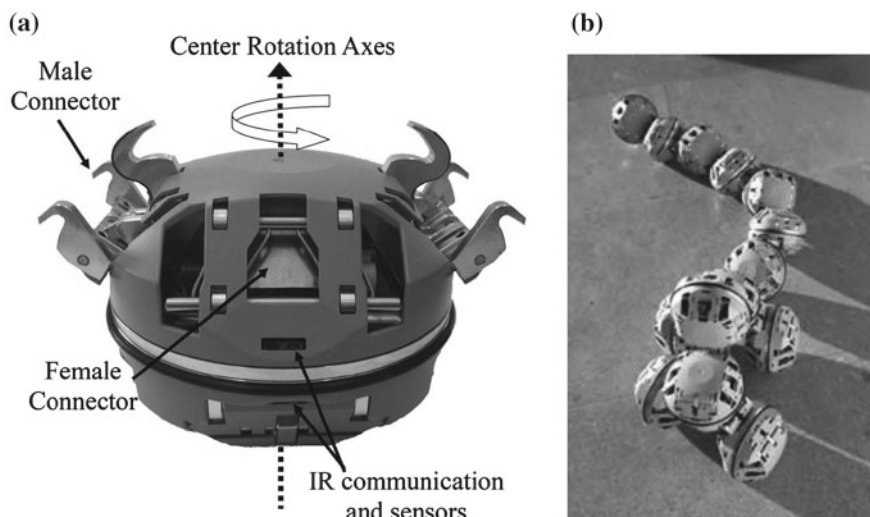


Fig. 2.7 Homogeneous ATRON robotic system (Østergaard 2004). **a** A single ATRON robotic module. **b** An ATRON re-configurable modular robotic organism

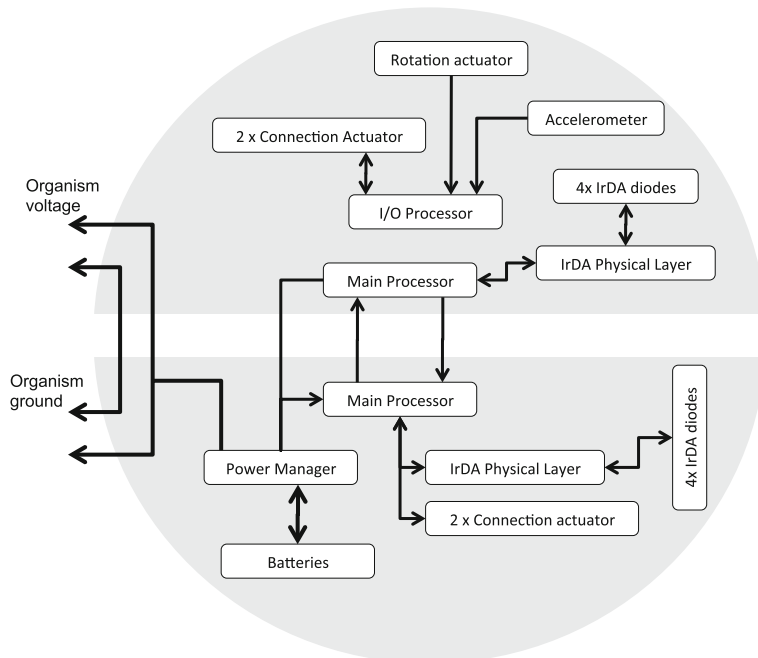


Fig. 2.8 Block diagram of the electronics in an ATRON module (Jørgensen et al. 2004)

tion actuator, an accelerometer, and four IrDA diodes. The main processor is responsible for communication with the northern hemisphere neighbors and the behavior of the entire module. The I/O processor, that is a 8-bit 1-MHz ATmega8 microcontroller, is responsible for reading the accelerometer, controlling the rotational mechanism, rotational lock, two active male connectors, communication and other system functionalities. The southern hemisphere in addition to the main processor includes a power manager, a battery pack, four IrDA communication diodes, and two active male connectors. The southern hemisphere's controller is responsible for intra-organism power sharing, recharging of the on-board battery pack, communication through four IrDA channels and controlling two active male connectors. Further, each ATRON robotic module is energized with two serially connected lithium-polymer cells (2S1P) that provide the nominal voltage of 7.4 V with a max. charge capacity of 980 mAh (Jørgensen et al. 2004). To maintain a consistent power supply to the robotic modules in an organism, the ATRON module's electrical skeleton enables the multiple modules to share or transfer their power with each other. This allows the robotic modules to recharge the neighboring module's battery pack. For this purpose, the entire supporting metal structure forming the skeleton of an ATRON module is electrically grounded. For applying positive electrical potential, one of the docking hooks has an electrically insulated piece of flexible printed circuit board glued on its top, that allows the module to form a power bus when physically connected (Jørgensen et al. 2004).

Figure 2.9 shows the power management system of an ATRON module. It includes a power manager, a charge manager, battery cells, and a CPU. The *power manager* in the system is mainly responsible for monitoring the on-board battery pack supply voltage and the voltage across the organism's power bus to select the best suited power source to power up its on-board electronics. The power sharing in a module is triggered if the organism's voltage (voltage across the power bus) is below a certain threshold with respect to the on-board battery pack's voltage. For collective recharging—parallel recharging of multiple modules in an organism—it is sufficient to connect a single module to an external power source. The modules then by election decide who gets to re-charge its battery pack first. A critical parameter,

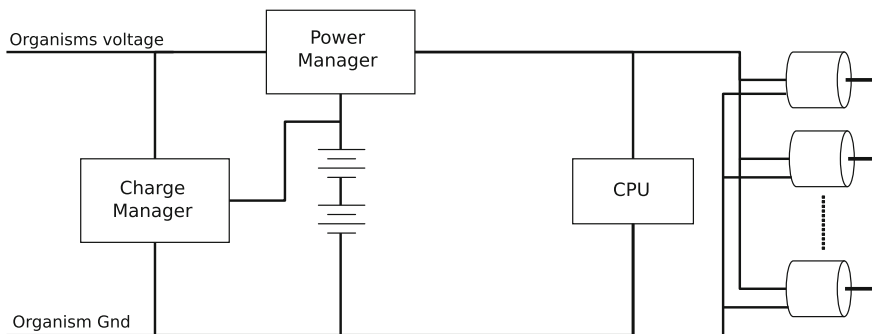


Fig. 2.9 ATRON module's power management system (Jørgensen et al. 2004)

that determines the maximum number of recharging modules in an organism at a time, is the maximum amount of instantaneous current that can flow through the organism's power bus. The ATRON module's system design allows a max. of 7 A of current to flow through. Therefore with 500 mA of recharging current, a maximum of 14 ATRON modules can be recharged simultaneously in an organism.

2.3.3 REPLICATORS

In a dynamic environment, the homogeneity of the robotic modules in a modular robotic organism may limit certain desirable features that are required, for instance, for collective locomotion on different surfaces, avoiding obstacles, lifting objects, exploring environment, etc. To gain such abilities, the idea behind heterogeneous platform design of REPLICATOR (Kernbach et al. 2008) robotic modules was intended to ensure high reliability, self-reconfigurability and adaptability in a modular robotic organism. The three heterogeneous robotic platforms specialized in different features are designed by the REPLICATOR consortium, namely; Kabot, Scout robot and Active wheel.

All three REPLICATOR platforms carry components that they have in common, e.g., docking units, cameras, transceivers for wireless and wired communication, IR sensors, microphones, accelerometers, and others. Among those components in which the three platforms differ due to their respective specialization belong the 3D-actuator, 2D-drive, and laser scanner. It is important to mention here, since the REPLICATOR robotic modules platform was partly designed and developed by the author therefore is used in this book to explore and address the issues related to the self-sufficiency of a modular robotic organism.

2.3.3.1 Kabot

The REPLICATOR *Kabot* is also called “backbone” robot because of its strong 3-D actuation. It was specially added as a feature for more robust, dynamic 3-D movement of robotic modules in the organism. Figure 2.10a shows the fourth generation of Kabot, with a 3-D actuation drive in the center. Its size is roughly $108 \times 108 \times 108 \text{ mm}^3$.

The Kabot mechanical platform is specialized in strong 3D locomotion and actuation capabilities in the organism mode. The robot possesses a strong brushless drive capable of lifting several robots. A single DOF approach reduces the cost and since the main motor is a rotation-symmetric sub-assembly, the available space inside is better suited to support one strong motor instead of several smaller ones. To compensate the reduced number of DOFs, the frame of the robot uses two L-shaped halves, which can be rotated against each other. This allows lifting or rotating connected robots depending on their docked position. A symmetric, genderless, active docking was chosen to eliminate the need to search for compatible docking interfaces. This increases the diversity of possible organism structures that can be built. To use the

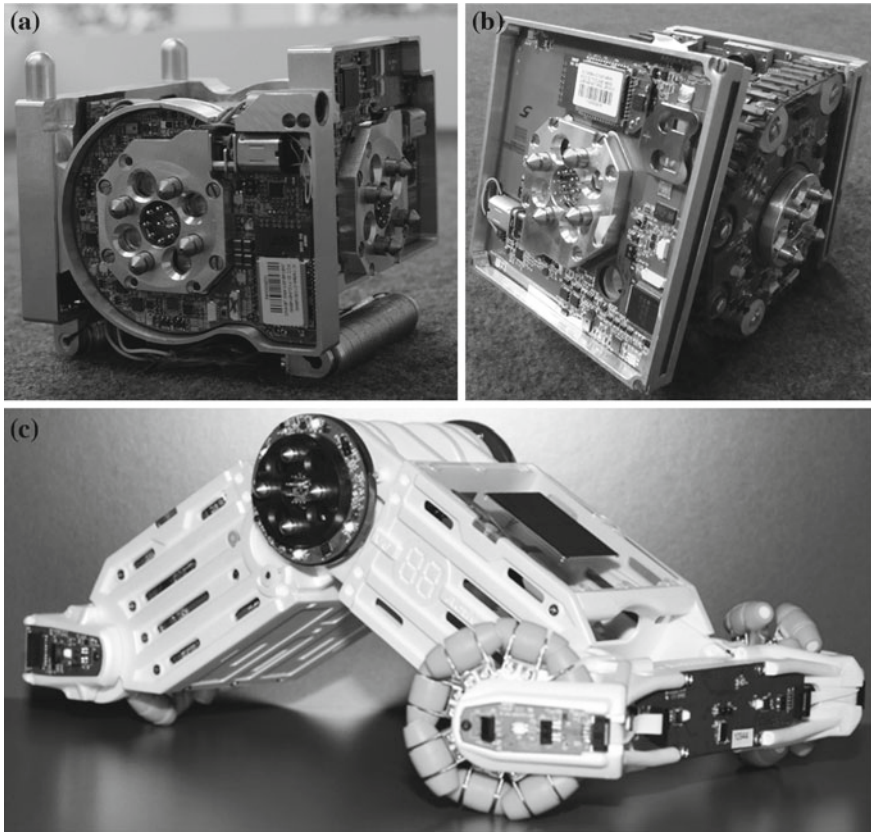


Fig. 2.10 REPLICATOR robotic modules (Kernbach et al. 2008). **a** Kabot. **b** Scout robot. **c** Active wheel

feature of the four mounted docking units, which not only provide for a stable connection but also for power and communication lines, a special 2D drive was included allowing the robot to additionally move sideways. The robot can move freely as a unit and it is possible to use the 2D drives of several connected robots to drive the organism. A cubical shape was chosen because it combines best the requirements for a small sized swarm robot and the requirements for modular self-reconfigurable robots, where a symmetric shape would simplify the reorganization of an organism.

2.3.3.2 Scout Robot

The REPLICATOR *Scout robot* is quite different in mechanical design from the Kabot. Figure 2.10b shows the fifth generation of REPLICATOR Scout robot. Its size is roughly $110 \times 110 \times 110 \text{ mm}^3$. Its platform is specialized in fast locomotion

on challenging terrains for exploration tasks. Therefore, the platform is equipped with additional sensors compared to the other platforms to scan the surroundings and the floor. In particular, the Scout robot has two laser-camera systems, on its front and rear to provide far and short range obstacle detection. The Scout robot's locomotion is based on tracks. Fast locomotion for the exploration on rough terrains is more important for the Scout robot than slow and precise locomotion for aggregation and docking alignment, compared to the Kabot module. Tracks enable the Scout robot to move forward, backward, turn left, turn right and turn on its axis. Moreover, the Scout robot can move on rough terrains, climb slopes and overcome small obstacles. The speed of locomotion is more than one body length per second. With the continuous elastic tracks, the robot can perform locomotion even after overturning accidentally.

Similar to the Kabot, the Scout robot has four docking units, one on each lateral side wall. The docking units are centered on the walls of the robot so that modules can dock regardless of their orientation. This platform has 2 DOF, bending and rotation, with a maximum torque of 4,7 Nm. The bending DOF ($\pm 90^\circ$) allows the robot's rear wall to lift up and down while the rotational DOF ($\pm 180^\circ$) allows the docking unit on the robot's left side to rotate along its axis. The 3D DOF has lower torque than the Kabot since the main role of the platform is exploration and not macro-locomotion. In any case, the Scout robot can lift two other robotic modules. The Scout robot, with higher DOF and lower torque compared to the Kabot, performs best if docked to the tail or to the head of an organism to scan the environment.

2.3.3.3 Active Wheel

The *Active wheel* platform is optimized for specific tasks in the swarm and in the organism and is therefore initially regarded as an active tool in the REPLICATOR swarm. Figure 2.10c shows the third generation Active wheel. With its omnidirectional and fast moving ability, it is specialized in transportation tasks. It is designed to carry and transport an organism consisting of several Scout or Kabots in a most energy-efficient way. The robot itself is able to approach the requested module or organism and can autonomously dock to it. The current design consists of two symmetrically arranged arms. On each end two 90° shifted omni-wheels are attached, which together form the locomotion mechanism of the robot. These arms are connected via a 180° turning hinge. Docking to other robots is provided by 2 docking elements placed on the same axis as the hinge. To allow docking in any position or height, the docking elements can rotate and are actuated by two dedicated motors.

The chassis of the robot has two cavities, which are ideal to place the main electronics and additional battery packs. On one hand, this configuration protects the sensitive components from potential damages during operations; and on the other hand, it provides enough room to place additional sensors for better perception of the environment.

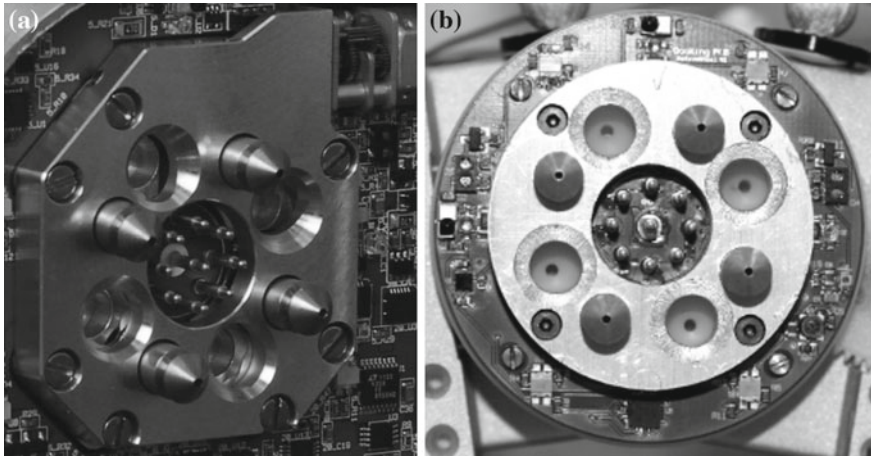


Fig. 2.11 REPLICATOR unified homogeneous docking unit. **a** Active docking unit on a lateral side of Kabot. **b** Passive docking unit on Active wheel

2.3.3.4 Homogeneous Docking Unit

A key element of re-configurable robot platforms is the ability to dock mechanically and electrically to merge into a larger artificial multi-cellular organism. A speciality of the docking unit developed for the REPLICATOR robotic modules, called CoBoLD—Cone Bolt Locking Device—is its genderless design which is made 90° symmetric to handle slight misalignment between the robotic modules during an autonomous docking. Figure 2.11 shows the unified homogeneous docking unit. The design itself is based on 4 cone shaped, spring loaded bolts. Because of the self-locking feature of a worm gear, no energy is required to hold the connection. In the center of the docking unit, spring loaded electrical contacts are placed to assure an automatic connection of power and wired communication between the two docked robotic modules. The contacts are placed on the same PCB holding all the other electrical components on the particular side. This eases the electronic design since no cables or additional contact PCBs are required. The arrangement of these contacts plus a special switching circuit ensures that robots can dock no matter how they are oriented with respect to each other.

2.4 Bio-Inspired Techniques

This section describes those techniques that are inspired from the swarm behaviors of social insect colonies in nature, and applied in swarm robotics to gain energy autonomy and robustness. The bio-inspired techniques includes artificial foraging, trophallaxis, energy harvesting, homeostasis and fault tolerance.

2.4.1 Artificial Foraging

As stated earlier, foraging is not only vital for the life of any living system but also equally important for a non-living system that tries to keep autonomy. *Artificial foraging* (AF) is inspired from the natural foraging behavior found in social insects. It is an act that enables an artificial robotic system to search and regain its replenished energy (food) from the environment. The first known robot that possessed the artificial foraging ability was believed to be the “Tortoise” robot developed by Grey (1963). Here, the foraging behavior was replicated using a simple light following approach that allowed it to guide itself towards the recharge station in the arena. AF behavior is quite analogous to the “basic cycle of work” defined by McFarland and Spier (1997), to maintain the “self-sufficiency” of an autonomous mobile system (McFarland 1994). Foraging and collecting behavior in artificial autonomous robots is replicated in a variety of ways. For instance, a forager robot or a group of foraging robots (Goldberg and Matarić 2000; Parker 1995) has to collect objects scattered in the arena and to bring them collectively or individually either in some random location called the *collection task* (Beckers et al. 1994; Matinoli et al. 1999), or a pre-specified location termed “home” known as the *foraging task* (Goldberg and Matarić 2000; Matarić 1992; Nitz et al. 1993). During the past two decades, several algorithms have been proposed that explore the foraging behavior in robotic systems. The first known implementation of foraging task in a group of real robots was done by Matarić and Marjanovic (1993), Matarić et al. (1995). Figure 2.12 shows Matarić’s robots used for foraging tasks. The proposed foraging behavior defines the following states: *avoiding*, *dispersing*, *searching*, *homing*, and *resting*, which are called behavior primitives.

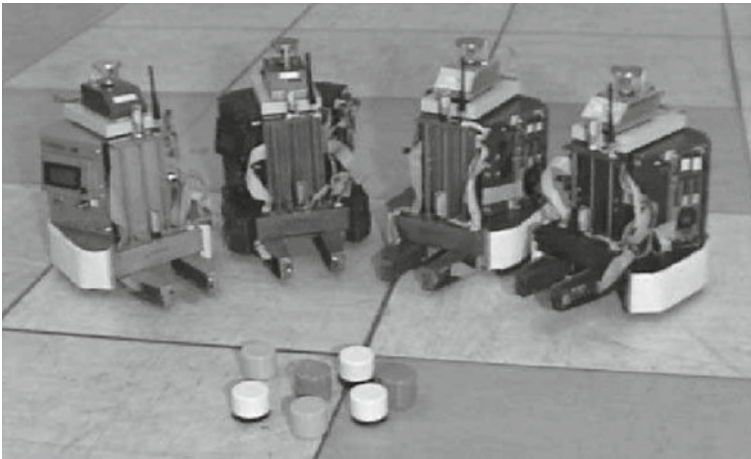


Fig. 2.12 Matarić’s foraging robots (from Matarić 1997)

In biological systems, the *efficiency* of foraging behavior has been defined in terms of “energy” that is measured as the weight of animals before and after foraging to quantify the amount of energy spent and gained (Charnov 1976; Houston et al. 1988; Oster and Wilson 1979). In the context of swarm robotics, the terms such as, *reward*, *income*, *benefit* has been used by Labella et al. (2004), Lerman et al. (2006), Lui et al. (2006), Ulam and Balch (2004).

Lui (2008) in his Ph.D. work modeled a collective foraging scenario in which the food-items were randomly scattered in the arena. A swarm of robots was assigned the task to collect and deposit the food items to a pre-specified location, called *nest*. Each food item that is collected by a robot provides an amount of energy to the swarm, but the foraging activity requires a certain amount of energy for searching and grabbing a food item. The optimal goal of the developed swarm was to maximize the net energy of the swarm.

The collective foraging behavior was modeled using a finite state machine (FSM), shown in Fig. 2.13. Each state in the finite state machine represents a sub-task that a robot undergoes during foraging. The transitions between the states are triggered on the basis of internal, i.e., time, or external, i.e., sensors, events occurred to a robot. The foraging FSM uses two types of timers, i.e., T_s and T_r , to count the time spent by a robot during, *searching* and *resting* in the nest. The transitions from states *randomwalk*, *scanarena*, or *movetofood* to state *homing* are triggered whenever *searching time* T_s reaches the threshold Th_s , i.e., $T_s \geq Th_s$, that ultimately reduces the number of foraging robots to resting robots. The transition between state *resting* and state *leavinghome*, is triggered when the resting time T_r reaches the threshold Th_r , i.e., $T_r \geq Th_r$, thus drives the robot back to foraging. The proposed adaptation that is biologically inspired uses three kinds of cues: *environmental cues*, *internal cues* and *social cues*, to adjust the time threshold parameters Th_s and Th_r that dynamically change the number of foraging and resting robots in the arena to improve the efficiency of the swarm. The performance metric of the swarm was calculated as the energy efficiency given by Eq. 2.1,

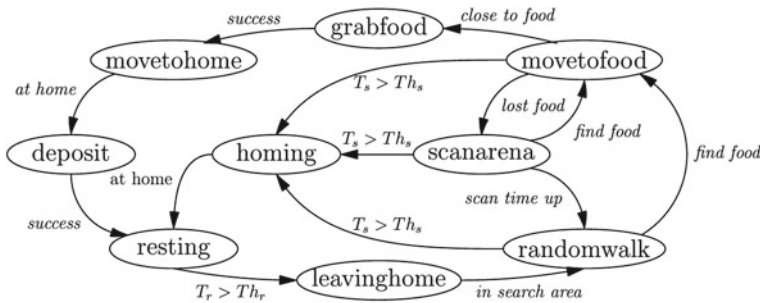


Fig. 2.13 Finite state machine for collective foraging task (from the work of Lui 2008)

$$\text{efficiency} = \frac{\text{net energy income to swarm}}{\text{energy available from environment}}. \quad (2.1)$$

Where the term in the nominator, i.e., *net energy income to swarm*, is the amount of energy gained every time a food item is collected by a forager from the arena minus the energy spent or invested to collect that food item. And, the term in the denominator, *energy available from environment*, is the total number of food items growing over time multiplied by the energy unit a food item provides.

2.4.2 Trophallaxis

“In the social Hymenoptera, continued evolution has placed a premium on the storage of nutrients for the benefit of the society as a whole. They have two central chambers in their digestive tract—the crop—followed by the midgut” (Eisner and Brown 1956). The crop holds the food that can be exchanged between the foraging and non-foraging nest-mates and is therefore called the ant’s *social stomach*. The exchange of food between the individuals of a colony serves as one of the most fundamental bond in the social organization of ants (Eisner and Wilson 1958; Wheeler 1923; Wilson and Eisner 1957).

Trophallaxis in biological systems is a process that allows conspecifics—two or more individual organisms of the same species—to exchange or transfer food through regurgitation via “mouth-to-mouth” known as, *stomodaeal* (Korst and Velthuis 1992) and “anus-to-mouth” known as *proctodeal* (Cabrera and Rust 1999). Figure 2.14a shows food transfer—trophallaxis—between two ants. Figure 2.14b shows the trophallaxis behavior between two honeybees in the hive. The sharing of food between the colony members is not limited to feeding the individuals rather it also provides a means of communication (Camazine 1993; Korst and Velthuis 1992), like in honeybee colonies (Wainseboim and Farina 2003). In addition, it also boosts colony-level immunocompetence, known as “prophylaxis” (Hamilton et al. 2010).

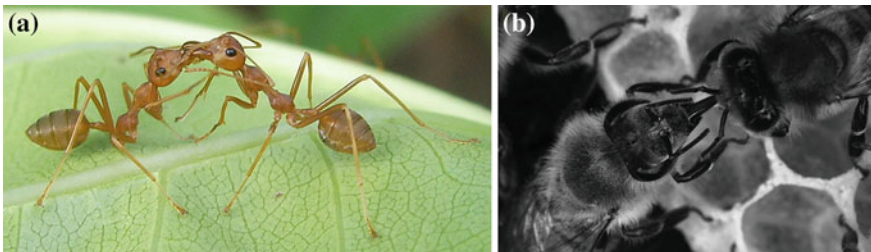


Fig. 2.14 Trophallaxis behavior in nature. **a** Ants engaged in mouth-to-mouth food transfer (source <http://upload.wikimedia.org/wikipedia/commons/d/dd/SSL12022p.jpg>, by Sean.hoyland, used under license Wikimedia Commons/Desaturated from original). **b** Trophallaxis between two honey bees—(mouth-to-mouth) food transfer (with the kind permission of Eric Tournet, author)

In swarm robotics, the term *trophallaxis* has been applied for both energy homeostasis in a swarm (Melhuish and Kubo 2007) and as a means of non-centralized communication within a large robot swarm colony as presented by Schmickl and Crailsheim (2008). Trophallaxis is specifically found useful in insect colonies that need to regulate the internal state of the colony, e.g., the protein supply in the bee hive, without having any centralized control, such as, a brain and communication path-way, like, neurons. Hence, for swarms of disjoint robots it can similarly aid in regulation and self-organization. A re-configurable robot however, does have a dedicated communication channel and may in some settings also provide a central “brain” by using distributed processing in only a few of specialized modules. It is even questionable if the term trophallaxis is valid at all *within* a robotic system consisting of merged¹ robotic modules that simply share their energy on a common energy bus. Trophallaxis may consequently not have the same significance in re-configurable robots, unless these themselves are a member of a swarm of *organisms*. Generally, in a swarm of self-reconfigurable robots one will find single, disjoint robot modules and robot organisms of different morphologies consisting of a variable number of merged modules. If two such robot units, single or merged, dock together for the sake of energy transfer, the term trophallaxis is regarded as being a valid analogy.

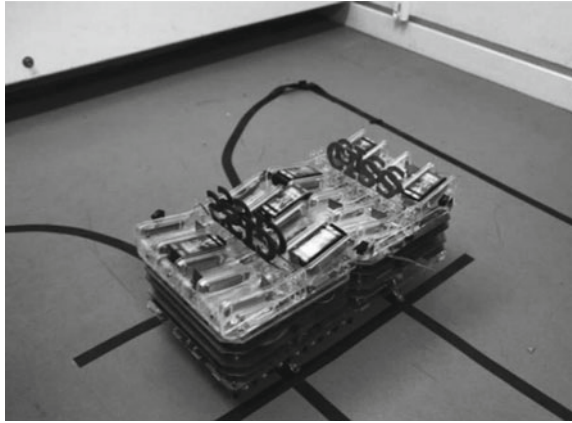
Ngo (2008) in his work, “towards sociable robots”, presented the concept of artificial trophallaxis in swarm robotics. For that, he developed a new robotic platform named “CISSbot” (Ngo and Schiøler 2008). The CISSbot has the physical dimensions of 15×15 cm square shape. The mechanical platform is driven by a differential two wheeled system and a pair of castors to maintain the robot balance.

Trophallaxis based on recharging the other’s battery is quite time-consuming (considering today’s battery technology up to several hours) and inefficient from an energetic perspective. Hence, Ngo (2008) has devised mobile swarm robots that exchange energy through swapping of battery packs. The battery packs are carried on their top out of a specifically designed battery holster unit that can hold up to 8 ejectable batteries, as shown in Fig. 2.15. The idea behind this was to increase the sustainability of the swarm, since the individual robots do not have to interrupt their ongoing tasks for re-fueling through a recharge station. Figure 2.15 shows the artificial energy trophallaxis—swapping of battery packs—between two CISSbots.

In contrast to biological systems, it is important to emphasize that by trophallaxis a swarm of robots does not gain energy. Rather, a significant portion of energy is lost, since the efficiency of energy transfer will always be less than perfect. The energy loss during the exchange is usually not as critical as it is in artificial systems. However, a swarm may prolong its autonomy through trophallaxis, by taking over the role of a “common stomach”, levitating any peaks and dips of energy supply and assuring a more evenly distribution of power supply (Humza and Scholz 2010).

¹ Although the robot hardware cannot merge, this term is being used but to underline that the modules do not only physically aggregate and bond, but also unifies logically—they become one larger entity.

Fig. 2.15 Artificial energy trophallaxis: two CISSbots are in a state of exchanging batteries (with the kind permission of Ngo) (2008)



2.4.3 Energy Harvesting

Robots that move and operate autonomously must have access to some external energy resources to maintain their autonomy. Although in recent years considerable progress has been made regarding the output power capability and the energy weight ratio of batteries (Linden and Reddy 2002), but the stored energy is still very limited compared to the power and energy demand of a meso-scale robot module. As is easily imaginable, this shortage becomes worse when energy-hungry, highly sophisticated sensors and processing systems are involved such as vision systems. In particular for self-reconfigurable robots, it is expected that in general a higher demand of sensing and processing power is required compared to “conventional” swarm robots that only move in 2D space. To increase the energetic autonomy it is hence wise to enable the robots to collect additional energy from the environment, which is often called *energy harvesting*.

Depending on a robotic module’s architecture and its operational environment, different forms of energy harvesting methods have been employed, like solar power (Boletis et al. 2006; Landis and Jenkins 1997), wind power, vibrational power (Wade and Gifford 2007), etc. The robotic systems that employ energy harvesting methods based on fueling from organic substances include: *Gastronome* that uses sugar (Wilkinson 2000), *Ecobot-II* that extracts power from sludges and flies (Ieropoulos et al. 2005b), and slugs eater using microbial fuel cells (MFC) (Kelly et al. 2000). Figure 2.16a shows a *Gastronome* robot and b shows an *Ecobot-II* that uses organic substances to fulfill its power demand. A microbial fuel cell that performed best, provides an average output power of $45.5 \mu\text{W}$ over a period of 10 days and has the dimensions of $6 \times 7 \times 5 \text{ cm}$ (Ieropoulos et al. 2005a).

When considering the power demands of a robot, its size is an extremely important factor since technologies for energy storage and harvesting do not scale down that easily. For example, if a cube shaped robotic module is shrunk in length by a factor x , the surface area would decrease by x^2 , whereas its volume would be reduced

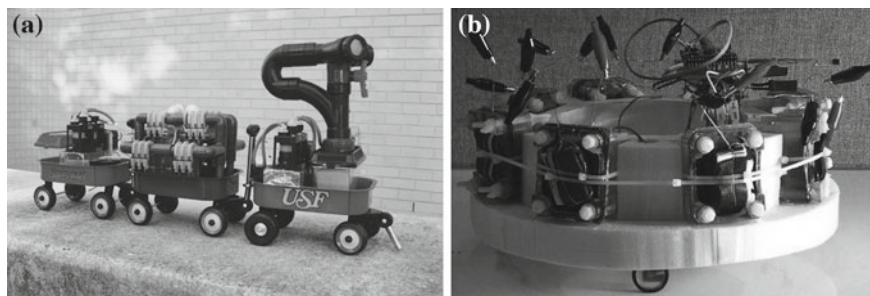


Fig. 2.16 Energy harvesting robots. **a** Gastronome—a prototype MFC powered robot, copied from Wilkinson (2000). **b** An Ecobot-II with O_2 MFCs onboard (Ieropoulos et al. 2003)

by x^3 . In a rough model, the energy consumption due to the weight of the robotic module is more or less proportional to its volume. On the other hand, solar cells generate electrical energy approximately proportionally to their surface area. This implies that it will be much easier to power a small robot with solar cells than a larger one. Reports on energy scavengers confirm the scaling issue. Even though energy harvesting is being employed in more and more marketed mobile electronic devices (vibrational energy scavenging in wristwatches, thermal energy transformers for wearable sensors, etc.) many of these are not very useful for robotic applications in the meso-scale. This conclusion is drawn from the fact that the reported appliances rely on extremely low power demand electronics and lack any actuator, a key component in robotics that due to physical limitations cannot be made energy preserving to an arbitrary level. It is certainly not possible to give exact numbers of robots' power consumption in general. There are several factors that come into play, such as size and hence the weight, processing power and grade of sophistication, principle of locomotion, number of sensors, and many more. In Mei et al. (2005) the author has tried to shed some light onto mobile robots at meso-scale. When looking at the given numbers (10–20 W) it is obvious that certain energy harvesting methods at the current state of art are not able to provide the necessary power. When considering that self-reconfigurable robots will usually be more complex than ordinary swarm robots (sensor fusion for docking approach, docking mechanism, etc.) energy scavenging by integrating harvesting modules within the robot modules is even less promising because of the tighter size restrictions and the expected higher energy demand due to 3D actuation, when lifting other modules.

Nonetheless, photo-voltaic cells are commonly being used for robot energy scavenging, in particular in space missions. The efficiency of solar cells not only depends on the specific technology being used (Si, Ga, thin film, crystalline, etc.) but also on the spectrum of light. Hence, the cells are compared at a well-defined spectrum, which has been standardized, e.g., IEC60904-3, at 1000 W/m^2 . This irradiance prevails approximately at a cloudless noon during spring in central Europe. The highest reported efficiency of PV modules—these are composed of several cells connected together—lie between 8.2% (thin-film poly-crystalline Si) and 22.9% (crystalline

Si) (Green et al. 2009). Consequently, a solar panel made of crystalline Si with a realistic efficiency of 15% under ideal conditions will generate approx. 150 W per square meter on an average bright sunny day. To power a re-configurable robot this may still not be enough, when taking a much smaller available surface area into account. But for a base or charging station to which the robot modules return to recharge this may be a realistic scenario.

2.4.4 Homeostasis

The biological term *homeostasis* or *milieu interieur* was first coined by the American physiologist Walter Bradford Cannon (Cannon 1926, 1929). Later, the following definition of homeostasis comes from his book “*The Wisdom of the Body*” (Cannon 1932):

“The coordinated physiological processes that maintain most of the steady states in the organism are so complex and so peculiar to living beings—involving, as they may, the brain and nerves, the heart, lungs, kidneys and spleen, all working cooperatively—that I have suggested a special designation for these states, homeostasis. It means a condition—a condition that may vary—but is relatively constant.”

Claude Bernard, a French physiologist, was one of the first to discuss what is now called homeostasis, frequently talking about the concept in terms of the stability of the internal environment (*milieu interieur*). Bernard and Dastre (1879) states:

“stability of environment implies an organism so perfect that it can continually compensate for and counter balance external variations. Consequently, far from the higher animals being indifferent to their surroundings, they are on the contrary in close and intimate relation to them, so that their equilibrium is the result of compensation established as continually and as exactly as if by a very sensitive balance.”

Wheeler (1911) described the regulatory behavior of social insects colonies as a “social homeostasis”. *Social homeostasis* is a phenomenon, in which the collective activities of a colony members help them to regulate the colony environment. For example, the regulation of nest atmosphere in termite colonies: the concentrations of carbon dioxide, oxygen, and water (Emerson 1956), respiration in honeybee colony, etc.

In a physiological context, the word “*homeostasis*” describes a property of a living system that is able to maintain a stable internal environment, despite the potentially considerable variations to its internal and external conditions. The *steady states* that Cannon referred to are the relatively constant values at which physiological variables such as blood pressure, body temperature and blood glucose levels are kept, what one might call “organismal homeostasis”. As Cannon states, the stability of these variables is maintained by a number of “coordinated physiological processes”. One such example is the *thermoregulation* of the human body. During thermoregulation, the variable to be maintained is body temperature, which is kept at a near constant 37 °C, despite fluctuations in the external temperature. A classic example of a stable

engineered system is the Watt's governor, a type of centrifugal governor designed in 1788 by James Watts (1736–1819) to regulate the speed of a steam engine. The Watt governor, and other engineered systems involving more state-of-the-art proportional control mechanisms are very good at maintaining the stability within the small task they were designed for, however, they lack the adaptability that stable biological systems possess. Biological homeostatic systems, as has already been shown, are highly adaptive.

Here, it is important to introduce the terminologies required to define homeostatic behavior, later used to define the homeostasis from the perspective of self-sufficiency of a modular robotic organism. Following terminologies largely originate from the two sources: Widmaier et al. (2006)'s book "*Vander's Human Physiology: The Mechanisms of Body Function*" on physiology and Ashby (1960)'s book "*The origin of adaptive behavior*".

- **Essential variables**

It refers to those variables that belong to the internal state of an organism and are closely linked with its survival. The deviation of an essential variable from its normal range will lead to changes in the organism that compromise its survival. Examples of essential variables include: body temperature, blood pressure and blood glucose levels, etc.

- **Set points**

It refers to those values that define the system steady state, around which each essential variable in a healthy organism will tend to remain. Essential variables do not remain constantly fixed at their set point values. In a homeostatic control system, when a variable deviates from its set point, several physiological processes will come into action that tend to bring the variable back to its steady state. For instance, the set point of the human body temperature is 37 °C. It should be noted, the set points are not fixed for the life-time, rather they are adaptable, to accommodate the changes happening in the operating environment. For instance, the human body temperature becomes higher during the day when the body is active.

- **Physiological limits**

They define the normal range of an essential variable. If a variable travels outside its limits then the survival of the organism is put at risk. For example, the physiological limits of the human body temperature are considered to be approximately 35–40 °C, an internal temperature of below 35 °C may lead to hypothermia whereas an individual with a temperature above 40 °C may experience heat stroke, leading to hyper-thermia.

- **Negative feedback**

It is the property of a system in which the increase or decrease in the value of a variable initiates the responses that tend to oppose the change to that variable. Within the endocrine system, for example, a negative feedback loop controls the release of many hormones in the human body. The increase in the concentration of a particular hormone often leads to a series of responses that in the end tend to inhibit or prevent the release of that hormone.

- **Regulatory response**

In response to a stimulus it is a physiological process that acts so as to maintain the essential variables within their physiological limits. Regulatory responses often form part of a negative feedback loop. For example, in response to a drop in body temperature the regulatory response of shivering will bring about changes that tend to increase body temperature. Responses can either be *involuntary*, *unlearned*, *built-in*, or *acquired* and subject to learning and improvement over time. According to Widmaier et al. (2006), since *built-in responses* are subject to learning therefore it is often hard to distinguish the following two types.

- **Feed-forward regulation**

It is a special form of regulatory response that compensates for changes to *essential variables* before any such change has actually occurred. Feed-forward regulation further minimizes the time and the extent which the variables are at risk of breaching their physiological limits.

- **Acclimatization**

Improvement of an already existing homeostatic system due to prolonged exposure to certain environmental conditions. The adaptation of the sweating response in different climates is a good example of acclimatization. The sweating response of an individual that is used to exercising in a hot environment will start quicker and act more aggressively than the response of an individual that is acclimatized to a colder environment.

2.4.5 Fault Tolerance

Tolerance to the presence of faults is an important consideration which an autonomous system must take in to account for its long-term survival. *Fault tolerance* is the ability of a system to continue its operation in the presence of faults and component failures perhaps by gracefully degrading its system performance (Parker 2012). It has been explored in a variety of application scenarios, and attained through different means. In artificial control systems, it is usually addressed through hardware redundancy and redundant control behavior. *Hardware redundancy* or *replication of hardware* is most commonly employed to enhance the system reliability by physically duplicating the critical system components or modules in the system design, that increases the endurance to faults and failures (Johnson 1984; Siewiorek and Johnson 1981). Later, Payton et al. (1992) had investigated a set of redundant control strategies to address high level failures, to gain a high level control and fault tolerance. A *high level failure* refer to those malfunctions that a system can encounter for which it is not explicitly pre-programmed. To deal with unexpected system behaviors, the controller is usually designed with redundant control strategies to perform a task. For instance, if a strategy does not suffice, the controller selects or switches to one of the other defined strategies (Ferrell 1993). Considering the two strategies, the hardware replication strategy is not acceptable to every system design as the redundant system

components not only increase the size of the system but also adds complexity and development cost. On the other side, the redundant control behavior based strategies to encounter faults does not suited well for hardware faults and failures.

In the context of modular robotic systems, the increased system complexity may in some scenarios result in higher failure probability. Carlson and Murphy (2005) report a wide variety of reasons why individual robots fail in the field. According to them, the different causes of failure can be roughly categorized as physical failures, software control failure, or human control errors. Winfield and Nembrini (2006) use the approach of qualitative Failure Mode and Effect Analysis (FMEA) (Dailey 2004) to evaluate the fault tolerance of a robot swarm. Under the FMEA approach, the designer tries to identify different sources of hazards that can cause *internal* and *external* failures in a robotic system. For each hazard, an analysis is performed to determine its impact on the performance of a robot swarm. The tolerance ability mainly depends on the detection, identification and isolation mechanism that allows a system to operate in the presence of malfunctioning components without the need of external intervention. *Fault detection* is a complex process for a number of reasons as pointed out by Parker (2012): the space of possible fault types, sensors, actuators, and the uncertainty in the environment models. *Fault detectors* are commonly employed to monitor a system's essential variables. Willsky (1976) presented a survey of design methods for failure detection in dynamic systems. Upon detection of an unacceptable deviation of a system's parameter, a fault is diagnosed. The *fault diagnosis* locates the fault and establishes its cause to evaluate its effect on the system steady state operations. If the evaluated fault is tolerable, the system may continue its operations; otherwise, the system adapts its behavior accordingly, i.e., through fault isolation and accommodation (Isermann 1984; Paul 1990).

Fault detection and isolation (FDI) is based on the use of *analytical* instead of *physical* redundancy (Paul 1990). It exploits the available information redundancy in the system (Garcia et al. 2000; Jeppesen and Cebon 2004; Leuschen et al. 2002). FDI techniques can be broadly classified into: *model based* and *knowledge based* models. Different model based approaches for fault detection have been developed over the years, see, e.g., (Gertler 1998; Himmelblau 1978; Isermann 1984, 2005; Isermann and Ballé 1997; Paul 1990; Willsky 1976). For knowledge based FDI approaches, see, Milne (1987) and Patton et al. (1989). The concept of *analytical redundancy* is commonly employed for *failure detection* and *isolation* in many systems. Without employing redundant components, it evaluates the inherent system redundancy through information processing under well-featured operating conditions. Figure 2.17 shows the general architecture of the FDI approach based on analytical redundancy, redrawn from Isermann (1984).

The analytical redundancy approach employs residual generation for the detection of faults in the system. The residual generation requires the system models during the three operating modes: *nominal*, or *steady state*, *observed*—actual—and *faulty system*. In the *fault decision logic* the decision on the occurrence of a fault is usually based on the *fault signature* generated from the faulty system model. On the detection of a fault its location, size and cause can be determined using fault diagnosis or inter-

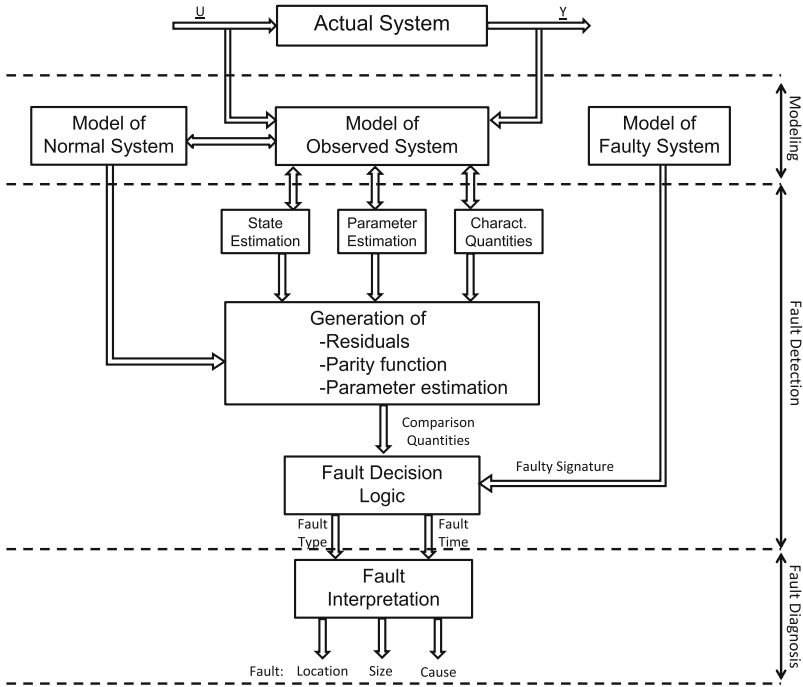


Fig. 2.17 General architecture of FDI based on analytical redundancy, redrawn from Isermann (1984)

pretation. For this a deeper knowledge about the system, e.g., aging, the operational environment, used tools, fault statistics, etc. is required (Paul 1990).

To capture the dynamics of an artificial system especially for the construction of a *faulty system model*, it is important to firstly, distinguish between failure mode and failure mechanism (Larsen 1974). A *failure mode* is a type of *failure* that causes deviations in the essential variables of a homeostatic control system, while, *failure mechanism* indicates the cause of a particular *failure* in the system. For this purpose, H.A. Watson in 1962 at Bell Telephone laboratories first used the *fault tree analysis* approach to analyze the Minuteman Launch Control System (Driessen 1970). *Fault tree analysis* (FTA) is a top-down approach to failure analysis. It identifies the various sequence of events that lead to a component malfunctioning. It starts with a potential undesirable event (failure) as a TOP event and then traces down all the way to its root cause or causes. A fault tree is a logic diagram based on statements that are either true or false, on or off, open or close, etc. (Larsen 1974). An event in the fault tree may either represent a component failure, an external influence/disturbance, or a system operation. The events in a fault tree are connected by a set of logic symbols. Figure 2.18 shows the fault tree analysis symbols.

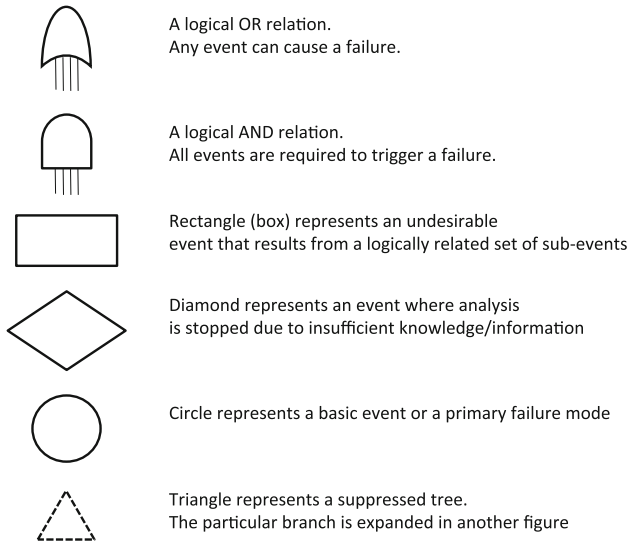


Fig. 2.18 Fault tree analysis symbols (from Larsen 1974)

2.5 Simulation Tools

2.5.1 Player/Stage

The open source project Player/Stage (Gerkey et al. 2003), provides an option for research into robotics and sensor systems. Player is probably one of the most widely used open source robot interfaces in research and post-secondary education (Toby and Bruce 2005). The components of the simulation framework include the Player network server and Stage robot platform simulators.

2.5.1.1 Stage

Stage simulates a population of mobile robots, sensors and objects in a two-dimensional bit mapped environment. It provides fairly simple, computationally cheap models of many devices rather than attempting to emulate any device with great fidelity. This is particularly useful for simulation of a large group of mobile robots. It is often used as a Player plugin module, providing populations of virtual devices for Player. Users write robot controllers and sensor algorithms as ‘clients’ to the Player ‘server’. Typically, clients cannot tell the difference between the real robot devices and their simulated Stage counterparts. After careful design of the simulated robot model in Stage, Player clients developed using Stage will work with little or no modification with the real robots and vice versa. Thus Stage allows rapid

prototyping of controllers destined for real robots. Various sensors and actuators are provided, including sonar, scanning laser range finders, vision (color blob detection), odometer, and a differential steer robot base.

2.5.2 *SPICE*

SPICE (Simulation Program with Integrated Circuit Emphasis) is a simulation and modeling program for analogue circuits and digital logic design. It is used in integrated circuit and board-level design to check the integrity of circuit designs and to mathematically predict the electronics circuits behavior. It provides an environment to simulate components ranging from the most basic passive elements such as resistors and capacitors to sophisticated semiconductor devices such as MESFETs and MOSFETs. Using these intrinsic components as the basic building blocks for larger models, designers and chip manufacturers have been able to define a truly vast and diverse number of SPICE models. Most commercially available simulators include more than 15,000 different components. An electronic circuit in SPICE is defined in the form of a net-list. The net-list is a text description of all circuit elements such as transistors and capacitors, and their corresponding inter-connections. Different schematic capture and simulation tools such as Multisim from National Instruments, LTSpice from Linear Technology, etc., provide a graphical user interface to draw circuit schematics in a user-friendly environment, and automatically translate the circuit diagrams into net-lists.

For the simulation of dynamic power flow and fault tolerance between the robotic modules of modular robotic organisms, in this research work, LTSpice has been used to predict the system behavior at the hardware layer. LTSpice IV is a high performance SPICE simulator, schematic capture and waveform viewer with enhancements and models for easing the simulation of switching regulators. It provides macro models for 80 % of Linear Technology's switching regulators, over 200 op-amp models, as well as resistors, transistors, and MOSFET models. Chapter 4 introduces the simulation framework that uses LTSpice with a simulation front-end designed in Lab-Windows/CVI to simulate power flow in different organism morphologies for this work.

2.6 Summary

This chapter briefly covered the background and the related work carried out in autonomous robotic systems, mainly from the perspective of self-sufficiency. It started with the basic definition of swarm intelligence and its five basic properties. It discussed the swarm behaviors observed in nature in different species that form the basis of swarm intelligence. The common examples of swarm behaviors include navigation, defense and foraging. The chapter then focused on swarm robotics by outlining three essential characteristics of a multi-robotic system: robustness, adapt-

ability, and scalability. After including the taxonomy of swarm robotics in literature, it briefly discussed the platform design and features of some of the re-configurable modular robotic systems that support power sharing. After having a detailed analysis of the state-of-the-art re-configurable modular robotic systems, the following features may be found to be missing in the existing systems:

- determine or measure the current flow through the on-board battery pack to system peripherals and to each docking side—to the docked robotic modules in the organism,
- withstand abrupt system failures, such as, a short circuit, erroneous sensor measurements, component failure, etc.,
- control the recharging of an on-board battery pack of a robotic module in organism mode, especially in case of a faulty robotic module,
- application layer control mechanisms for dynamic power sharing between the robotic modules of an organism.

The current flow measurements at the critical system segments play an important role in controlling the dynamic operations of any battery operated autonomous mobile robotic system. These measurements allow a system to predict system malfunctions/failures even before they occur. The current flow measurements also help the robotic modules to efficiently utilize the available energy distributed in the organism. An important ingredient in the electro-mechanical design of any autonomous system is its ability to withstand system failures. From dynamic power sharing perspective, the system design must have an innate fault tolerance, e.g., current limiters, fuses, switches, etc., to withstand the effects of endogenous and exogenous failures. The control over the current flow path in a modular robotic organism, at one end provides adaptive fault tolerance to an individual and collective systems, on the other hand, allows the robotic modules to efficiently utilize the distributed energy reserves in the organism, for instance, by forming sub-power buses. The chapter then introduced the platform design of state-of-the-art REPLICATOR re-configurable robotic modules. The electronic design of the REPLICATOR robotic modules, more explicitly, the power management system with fault tolerant energy sharing, has been developed and used for experiments in this research work. Later, it reviewed some of the bio-inspired techniques that have been applied in other robotic applications to gain energetic autonomy. The presented bio-inspired techniques include artificial foraging, trophallaxis, homeostasis and fault tolerance. At the end, some of the simulation tools used during the work were briefly described.

Self-Sufficiency of an Autonomous Reconfigurable
Modular Robotic Organism

Qadir, R.H.

2015, XV, 154 p. 79 illus., 16 illus. in color., Hardcover

ISBN: 978-3-319-10288-7