

Concepts for 3D Printing-Based Self-Replicating Robot Command and Coordination Techniques

Andrew Jones and Jeremy Straub *

Department of Computer Science, North Dakota State University, Fargo, ND 58102, USA;
andrew.jones.4@ndsu.edu

* Correspondence: jeremy.straub@ndsu.edu; Tel.: +1-701-231-8196

Academic Editor: Steven Y. Liang

Received: 7 December 2016; Accepted: 16 March 2017; Published: 6 April 2017

Abstract: Self-replicating robots represent a new area for prospective advancement in robotics. A self-replicating robot can identify when additional robots are needed to solve a problem or meet user needs, and create them in response to this identified need. This allows robotic systems to respond to changing (or non-predicted) mission needs. Being able to modify the physical system component provides an additional tool for optimizing robotic system performance. This paper begins the process of developing a command and coordination system that makes decisions with the consideration of replication, repair, and retooling capabilities. A high-level algorithm is proposed and qualitatively assessed.

Keywords: self-replicating robots; robot command; robot coordination; multi-agent system; 3D printing; additive manufacturing; robotics

1. Introduction

The concept of self-replicating robots has been around for some time—dating back to the 1940s and earlier [1]. With the advent of 3D printing technology, the development of self-replicating robots seems more feasible to implement than it was previously. This opens the door to research and development opportunities in this area of robotics.

Ellery [2] proposed the use of self-replicating robots for planetary exploration (in particular, focusing on the moon). A major benefit of having robots that are able to make more robots on their own is that the survivability of the multi-robot system increases dramatically. A single surviving robot that can self-replicate could repopulate an entire ‘colony’ of robots with sufficient resources and time. This would give the robots an opportunity to take more risks in trying to accomplish an objective. While suggesting solutions to numerous hardware and logistical issues, Ellery’s work didn’t consider the autonomy of the robotic system. Autonomy is, of course, a key consideration for maximizing the functionality (or allowing functionality in a communications limited/denied environment) of this type of robotic system.

This paper advances previous work by presenting the conceptual framework and an overarching architecture for a self-replication robot system. It also discusses the operation of such a system, by describing an autonomous command and coordination approach. To understand how to efficiently utilize a self-replicating robot system, it is important to analyze the overall concept and command strategies involved. These are discussed in subsequent sections. Then, a prospective command algorithm is proposed and qualitatively evaluated, based on selected metrics and using case studies, to ascertain its prospective efficacy.

2. Background

This section presents prior work in a number of areas that are relevant to the challenge of the development of a command system for a self-replicating robot. First, an overview of prior work on self-replication and its prospective benefits, is presented. Then, prior work on soft robotics, distributed robot system coordination, swarm intelligence, and automated manufacturing, is discussed.

2.1. Self-Replication

Von Neumann is seen by many as the father of self-replicating machines. In the 1940s, he investigated the logical foundations of self-replication [1]. In the 1950s, he proposed a self-replicating structure [3] and this work was presented in *Scientific American* [4], bringing it into the public consciousness. After von Neumann's death [3], Burks completed his design for a 29-state automaton and published it in 1966 [5]. More recently, the technology to actually perform self-replication has become feasible. Beuchat and Haenni [3], for example, created a hardware implementation of cellular automaton and published the results of its analysis in 2000.

As a stepping stone to self-replication, self-configuration and self-reconfiguration were pursued. Von Neumann proposed a self-assembler (that used a cache of spare parts) [1]. Whitesides and Grzybowski [6] demonstrated the similarity of natural and mechanized self-assembly. Butler, Murata, and Rus [7] advanced mechanized self-assembly by developing algorithms for a generic self-reconfiguring robot to divide and reform into different configurations. Sahin, et al. [8] demonstrated how self-assembly and disassembly can be used to allow a swarm of robots to collaborate, in one instance, while retaining the capability to perform tasks independently and in smaller groups, in other instances. Butler, Murata, and Rus [7] presented algorithms for this purpose.

Evolutionary self-assembly was proposed by Jakobsen and Tannum [9], building on the self-improvement concept (for replication) proposed by Koza and Rice [10] in 1995. Cooperation [11] and colonies [12] of self-assembling robots have also been proposed.

In [13], Pfeifer, Lungarella, and Lida augment the self-replication concept by presenting a discussion of how robots will eventually be able to exhibit certain desirable properties of biological organisms, such as adaptivity, robustness, versatility, and agility. In [14], Lee, Moses, and Chirikjian follow a von Neumann-inspired framework and, in this context, define the degree of self-replication and task complexity. Self-replication is presented in terms of an equation that compares the complexity among subsystems and overall system complexity. Mathematically, this can be presented as:

$$D_s = \frac{C_{min}}{C_{max}} * \frac{C_{total}}{C_{ave}} * \frac{1}{C_{ave}} \quad (1)$$

where C_{min} is the module under test with the least complexity and C_{max} is the module with the most complexity. C_{total} and C_{avg} present the sum and mathematical mean, respectively. Entropy (based on Sanderson's model [15]) is used as a measure of task complexity.

In addition to logical and theoretical works, a number of efforts have been made to create hardware systems. Suthakorn, Kwon, and Chirikjian [16], for example, demonstrated the operations of a semi-autonomous robot made from LEGO Mindstorm kit parts, that performs supervised replication. In [16], they built upon previous work (in [17]), where a concept and initial remote-controlled replication was presented. In [18], machine vision and other capabilities are added to the system, to facilitate autonomous operations. Similarly, Zykov, et al. [19] demonstrated real-world replication using modular robots based on specially-produced cubes. These robots collect cubes from feeder troughs and use them to produce equivalent copies. Even more flexible is the work presented in [20], where an algorithm for duplicating shapes using 'smart sand' replicates the shape of presented 3D objects. This work's efficacy was demonstrated via hundreds of simulated test runs.

More practically, the RepRap 3D printer [21] has been used as a template for the creation of numerous consumer-grade 3D printers, based—in part—on the use of parts printed on another 3D printer. Once a user has a working RepRap printer, he or she can produce many of the mechanical parts required to make another RepRap printer for his or her own use or for use by another person.

Unlike the systems proposed herein, which use autonomous replication, RepRap construction requires significant human involvement.

2.2. Robot 3D Printing and the Use of In-Situ Resources

Creating robots with 3D printing is well established. Numerous robots and robot parts have been created using 3D printing. The MU-L8 robot [22], for example, utilized 3D printed limbs to emulate human movements and play robot soccer. For more distant applications, the use of in-situ resources is necessary to prolong mission duration and, potentially, facilitates having a greater ability to take risks. Examples of in-situ resources being used for 3D printing include the use of basalt printing of structures for Martian exploration [23]. The use of a D-shape printer for building infrastructure out of regolith on the Earth's moon [24], and the use of a collection of simple self-replicating robots to exploit lunar material and energy resources [25], have also been previously considered.

2.3. Soft Robots

Soft Robotics refers to robotic devices that are fabricated from soft, flexible, materials, instead of the hard plastics and metals traditionally used in robotics [26,27]. An overview of how they have been designed, fabricated, and controlled is presented in [28]. An overall theme is that they tend to trade precision and deterministic control for bioinspired compliance and physical robustness [29].

Hiller and Lipson [29] demonstrated the automatic design of freeform soft robots for forward locomotion, using soft volumetrically expanding actuator materials. A robot was automatically fabricated and assembled. Performance demonstrated an error of approximately 15%.

In [26], “smart materials” (materials which change their physical properties in response to external stimuli) were demonstrated and used to create a tentacle-like active structure, employed for movement.

In [30], Bartlett et al. employed multi-material 3D printing to manufacture a combustion powered robot whose body transitions from a rigid core to a soft exterior. The robot is powered by the combustion of butane and oxygen, and can perform untethered jumping.

2.4. Multi-Robot Coordination

The study of multi-robot coordinated systems, according to Yan, Jouandeau, and Cherif [31], has recently increased “significantly in size and importance”. They attribute this to the resolution of many previously vexing issues in single robot systems, as well as to specific multi-robot system needs. A number of key decisions define the coordination of a multi-robot system [31]. These include decisions related to the use of static or dynamic coordination, explicit or implicit communications, cooperative or competitive approaches, and centralized or decentralized decision-making. Task and motion planning, and resource conflict resolution techniques, also need to be identified.

Several examples of coordination approaches exist. Nieto-Granda, Rogers, and Christensen [32], for example, compared three exploring and mapping strategies: the reserves, divide and conquer, and buddy system approaches. Under the reserves approach, extra robots wait in the starting area until they are needed and are then given tasks. Under the divide and conquer approach, robots travel in as large of a group as possible and split in half when new navigation goals are uncovered. Finally, with the buddy system approach, robots travel in teams of two, until new navigation goals are detected. Similar to the divide and conquer approach, the team will then split, following both paths.

Portugal and Rocha [33] compared two techniques for the multi-robot system patrolling of a given area. The first technique seeks to optimize local gain (using a Greedy Bayesian Strategy). The other technique seeks to reduce interference and foster scalability (using a State Exchange Bayesian Strategy). They found that both approaches sufficiently solve the problem; however, the state exchange strategy outperformed the greedy strategy.

A wide number of examples of multi-robot coordination use exists. Liu, et al. [34], for example, presented a control system for a collection of life science laboratory mobile robots. Pennisi, et al. [35]

presented the use of multi-robot surveillance (for indoor public places) using a distributed sensor network that combines RFID tags, mobile robots, and RGBD cameras. Starke, et al. [36] demonstrated close-proximity multi-robot operations for a welding automation application.

To meet the challenges presented by distributed systems, a variety of approaches have been suggested. Caliskanelli, Broecker, and Tuyls [37] presented a swarm-inspired method (based on the pheromone signaling behavior of honey bees), called BeePCo, to maximize the total area covered by a robot network in an environment. Swarm control styles are discussed in greater detail in the subsequent subsection. Straub [38] proposed a boundary node-based Blackboard Architecture approach for limiting the data replication traffic, to facilitate local robot decision-making. Jullian, et al. [39] proposed an information theoretic approach that iteratively estimates the environment state using a sequential Bayesian filter and a gradient of mutual information for the purposes of distributed control. Jin, Xingjie, and ZengRong [40] explored the use of robot coordinated adaptive tracking. They presented a control algorithm with the distinctive feature that only a subset of followers need to access the position information of a dynamic leader in the task space, reducing communications and other resource needs.

2.5. *Swarm Robotic Control*

With swarm robotic control, simplistic local rules are used to create complex behaviors [41]. The approach is patterned on insect colonies where groups of insects perform behaviors that are too complicated to be coordinated by any one insect's capabilities [41]. Sahin [42] proffers that swarm robotics involves the use of a "large number of relatively simple physically embodied agents", from which a "desired collective behavior emerges from the local interactions among agents and between the agents and the environment". Practically, this means that members of the robotic system can have simplistic command software and reduced processing capabilities, but still produce a complex outcome. Several efforts to classify swarm robotic systems have been conducted. Abukhalil, Patil, and Sobh [43] define four high-level categories for robotic systems: swarm, self-replicating, self-reconfigurable, and modular. Significant overlap between these categories exists. Groß, Dorigo, and Yamakita [12], for example, combine self-assembly/reconfiguration and swarm control (this approach is also discussed by Barca and Sekercioglu [44]).

Brambilla, et al. [45] take an alternate approach in (like Abukhalil, et al.) categorizing prior articles on swarm intelligence. Unlike Abukhalil, Ptali, and Sobh's approach, Brambilla, et al. classify systems into the categories of method-based and collective behavior-based. The method-based category is further divided into two sub-categories (design and analysis methods), that are further divided into five sub-categories (behavior-based, automatic design, microscopic, macroscopic, and real-robot analysis). Collective behaviors are divided into four sub-categories (spatially-organizing, navigation, collective decision-making, and other), which are further divided into ten sub-categories.

Swarm control has been demonstrated for a variety of applications, including robotic self-assembly [12], dynamic cleaning [46], exploration and mapping [47], foraging [48], object movement and interaction [47], and coordinating cooperation [47,49]. Systems implementing swarm approaches, according to Sahin [42], have benefitted from system robustness, flexibility, and scalability benefits, provided by the approach. Barca and Sekercioglu [44] also identify a number of application-specific benefits.

2.6. *Manufacturing Automation*

Manufacturing automation uses electrical-, mechanical-, and computer-based solutions to operate and control a production process [50]. It is becoming more popular as markets drive rapid product enhancements and the customization of products requires the use of flexible automation infrastructures [51]. To this end, Saliba, Zammit, and Azzopardi [52] discuss a strategy and propose a set of practical guidelines for reconfigurable manufacturing automation.

For a machine to be self-replicating, it must be able to automatically manufacture a replica of itself. Collaborative robots that are used in manufacturing plants are becoming more flexible and

efficient [53]. Robots are now considered, by some, as an integral part of industries, due to their role in improving accuracy, repeatability reliability, preciseness, and efficiency [50].

The software that manages the automation process, referred to as manufacturing automation software projects (MASP), includes information regarding applied automation hardware and is becoming more complex. In [54], an approach for model driven development of automation software, based on the Systems Modeling Language, is discussed. In addition, Vyatkin [55] provides an overview of state-of-the-art software engineering for industrial automation.

The increasing capabilities (hardware and software) created for manufacturing automation make a significant contribution to work on self-replicating robotics. The automation of the replication process is a necessity for robots to be truly self-replicating.

3. Capability Requirements

In the design and development of a multi-robot system, identifying the requirements is a key initial task. A self-replicating multi-robot system must have certain capabilities to be successful. This section outlines these requirements. Relevant capabilities include the ability to: move, communicate, process data, sense the environment, and make more robots. Each will now be discussed.

3.1. Mobility

How robotic mobility is achieved has a bearing on robot replication decision-making and the required supplies. Robot mobility is necessary because making robots requires resources, and mobility is one portion of solving the challenge of obtaining resources.

One option for mobility, perhaps an obvious choice, is the use of wheels. Depending on the quality, wheels are generally reliable and able to support a substantial amount of weight. However, in the absence of appropriate mechanisms, the wheel utility can be easily impaired by unfavorable terrain. Another option is the use of robotic legs. The number of legs and the way in which they are configured on the robot would determine its mobility capabilities. Advanced options include hover, jet propulsion, subterranean digging, and gliding.

3.2. Communication

The coordination of a multi-robot system is heavily dependent on the ability of robots to communicate. Communicating can be accomplished with various techniques and technologies. For instance, installing a radio on each robot would enable them to communicate at a distance, as well as removing the necessity for a line of sight between them (although obstacles and environmental factors could interfere with the signal). Another possibility is using a coded visual system, such as blinking LED lights, to relay a message. A more range restrictive approach would be physically attached wires that temporarily connect robots.

3.3. Processing

Each robot needs a computer processing unit. Fabricating such a device in a factory setting with exact materials, machinery, and a relatively controlled environment, is well understood. For self-replicating robots, especially those designed to forage for resources, fabrication is complicated due to the possible lack of the availability of suitable materials, as well as the need for specialized fabrication equipment and suitable conditions. Currently, processor fabrication requires large, heavy, and expensive production equipment, which makes it impractical for in-situ use. In the longer term, with suitable resources available in-situ, the development of a processor fabrication capability may be possible. In the short term, however, processor availability becomes a limiting factor, that may be the primary determination of how many robots can be produced within a given environment and its available resources, if computer processing units cannot be fabricated locally.

3.4. Sensors

Sensing various aspects of the environment is important for performing mission tasks, gathering resources, and navigating the terrain. These can include visual, audio, touch, and magnetic sensors that aid in detecting various aspects of the environment. Robots may need to relay sensor readings to others, without sensing capabilities, for some applications.

3.5. Replication

The capability to perform replication activities is a core functionality of a self-replicating robot system. The replication process can be performed in a single step or consist of multiple stages of construction, such as the fabrication of parts and their subsequent assembly.

The selection of a replication mechanism must consider the need to duplicate itself during the robot production process. Thus, using sophisticated replication equipment would necessitate the ability to make a copy of the sophisticated equipment.

One promising solution is the use of 3D printing, with an emphasis on printer designs that are simple, yet very capable. For example, currently-available RepRap 3D printers can print parts to make another RepRap printer [21]. However, the assembly of the robot does not need to be performed solely by the printer. In the case of RepRap reproduction, it's achieved through human involvement.

To perform robotic replication, a printing unit could be teamed with robotic arms (either on a single robot, or from a second cooperating robot). These arms could be used to assemble printed, stored, or otherwise obtained parts into their needed configuration. Some sort of assembly capability is critical for allowing robots to print similarly sized robots. If the printing robot cannot produce and connect multiple small parts (or expand the printing area in some way), a subsequent generation of robots would be constrained to be smaller than the originals, so that they could be printed in the printing area.

Techniques for robotic assembly are well understood. Robotic assembly has been used for applications ranging from small parts [56] to buildings [57]. Techniques for both independent and cooperative robotic assembly techniques have been previously proposed [58,59].

In addition to printing and assembly capabilities, a capability for milling [60] may also be needed. Milling capabilities may be required to work with metals and other materials that cannot be 3-D printed, or for rectifying issues detected with a printed object or component. However, milling capabilities will not be required in many cases, as robots can be made with printable materials (plus stock components).

4. Design Considerations

This section examines system design considerations. First, robot replication design choices are considered. Then, the use of homogenous versus heterogeneous robots is discussed. Third, design considerations based on concepts found in nature are outlined.

4.1. Robot Replication

The difference between self-replicating multi-robot system approaches are now considered. The first approach is that of a centralized control node. Second, a peer to peer approach is discussed. These approaches can be implemented for either replication or communication, or perhaps both.

4.1.1. Centralized

With the centralized approach, a special robot or stationary hub is setup such that it acts as the replication center. Having a central node dedicated to the replication process has benefits and drawbacks. One benefit to this approach is that the regular robots that are made by the central node wouldn't have to have the replication equipment installed, such that the additional materials that would not be needed for this could be used for other purposes. It would also mean that all of the replication-related materials collected would end up in the same place, such that the replication process wouldn't get bottlenecked by poor resource exchange between robots (although this could

be remedied by proper implementation of cooperation in non-centralized models). A drawback to this approach is that robots must bring all replication resources to the central node, which becomes problematic for resources that are located far away. This centralized approach also introduces the question as to whether this central node can be made in-situ, or must be specially fabricated and installed in the desired location. If it cannot be made by the multi-robot system, or if an existing central node cannot craft a copy of itself, then this creates a central point of failure. Even if it can copy itself, it would be a single point of failure until the copy was made.

4.1.2. Decentralized

An alternative approach is where all robots can carry out the replication process. A benefit to this approach is that, if one robot survives, and has replication resources available, it can make more robots. This notion leads to the increased survivability of the multi-robot system. It also allows the multi-robot system to split up, and have less dependency on a central hub. One potential drawback, depending on implementation, is that a central hub can be highly versatile in transforming raw materials into usable replication resources. Lacking the ability to use a wide array of raw materials would cause problems in certain environments.

4.2. Homogeneous vs. Heterogeneous

If all the robots in a system are the same, it is considered a homogeneous system. Having multiple types of specialized robots makes the system heterogeneous [31]. Deciding whether to have different autonomous robots with unique designs to fulfill various requirements includes the consideration of many factors.

4.2.1. Material Usage

Using the minimum amount of materials required to produce a functional robot—meeting relevant quality standards—is desirable, in the context of producing a single robot, as it allows the greatest number of robots possible to be constructed, given the available level of in-situ and stored materials. However, while reducing material consumption in robot production is desirable, having plenty of stock (beyond known and projected requirement levels) is also desirable. This excess stock facilitates system resiliency. In particular, it increases the options that the robot system has in regards to new construction and/or repairs, especially during periods where resource collection is scarce (if resources are prone to periods of scarcity and abundance).

Tailoring robots to special roles can ensure that each robot is designed to include only the necessary functionality, thus saving potentially valuable resources. Another consideration is the design's material requirements versus what is available and abundant. Choosing a design that utilizes locally abundant resources would be ideal, to maximize the number of robots that can be produced.

4.2.2. Manufacturing Efficiency

The decision-making process may also need to consider which robot designs are most efficient. For instance, if the equipment that builds robots is tailor-made for efficiently producing a specific design, then deviating from that would come at a cost. Alternatively, a specialized design may also require certain less-abundant resources or a long duration to build.

4.2.3. Specialization Factors

Having specialized designs may affect how the tasks are allocated amongst the robots, such that the capabilities of each robot would need to be accurately considered in this decision-making process. These specialized designs could arise from evolvable hardware, which makes use of evolutionary computation methods to develop a variety of technologies that enable the automatic design, adaptation, and reconfiguration of electrical and mechanical hardware systems, in ways that outperform conventional techniques [61]. However, a potential disadvantage to this is that producing

a robot for a specific task which isn't carried out frequently may render it unhelpful after that task is completed.

4.2.4. Hybrid

It may be possible to combine some of the advantages of specialization and interchangeability. For instance, if robots are comprised of a base unit and interchangeable add-ons that can be removed and added, this could effectively make a robust system of highly adaptable robots.

4.3. Parallels with Biological Organisms

It is beneficial to consider design concepts based on nature. The field of bionics, for example, seeks to do this by designing technology that mimics features of biological structures [13]. For instance, the presence of skin on a hand affects the gripping of objects [13], and an analog may be needed for robotic gripping applications. Jiang, et al. [62] designed an antenna radar cross-section that was inspired by studying insect tentacles. In terms of future applications, Dickinson [63] speculates that, as mechanical capabilities increase and are able to implement such designs, engineers may adopt more and more design concepts from nature.

Biological organisms also provide a model for replication, reproducing in many ways and adapting to their environment through the process of natural selection [64]. Similarly, robots could build other robots that are better suited to the environment or to objective-related needs. For robots, the 'mating' process may be replaced with an automatic analysis of needs, and may only involve one parent.

The behavior of certain species of organisms can also provide design inspiration for robots. For instance, certain social animals, such as ants and birds, exhibit intelligent collective behavior. Observations of these animals provided inspiration for swarm intelligence [45], which was discussed in Section 2.5.

Organic materials can also be used as part of a robot's electronic system. For instance, organic semiconductors such as Organic Thin Film Transistors (OTFT) may be able to be used as printable low-cost materials for a wide variety of applications [65]. This may make producing robots from in-situ resources easier to accomplish, and increase possible fabrication options so that specific resource scarcity may become less burdensome. Furthermore, it may be possible to grow the organic compounds necessary to craft organic electronics.

However, while biology provides many insights, certain design considerations cannot be directly inferred by studying organic life. For instance, robots do not share certain constraints that animals have, such as the need to maintain a running metabolism. Certain technological solutions may also be superior to natural counterparts. Pfeifer et al. [13], for example, suggested that this was the case for the wheel.

5. Task Allocation

Task allocation is critical to the operations of a multi-robot system. Determining which robot does what part of what task can become complicated, as tasks become more complex. This is a field that is reasonably well-explored by the scientific community, known as *distributed task allocation*. This section provides an overview of this challenge and the prior work used to solve it.

5.1. Scheme

Gerkey, et al. [66] provides three criteria that can be used to characterize multi-robot task allocation problems:

- Single-task robots (ST) vs. multi-task robots (MT):
 - ST means that each robot is capable of executing, at most, one task at a time.
 - MT means that some robots can execute multiple tasks simultaneously.
- Single-robot tasks (SR) vs. multi-robot tasks (MR):

- SR means that each task requires exactly one robot to achieve it.
- MR means that some tasks can require multiple robots.
- Instantaneous assignment (IA) vs. time-extended assignment (TA):
 - IA means that the available information concerning the robots, the tasks, and the environment, only permits an instantaneous allocation of tasks to robots, with no planning for future allocations.
 - TA means that more information is available, such as the set of all tasks that will need to be assigned, or a model of how tasks are expected to arrive over time.

It is difficult to definitively categorize a self-replicating multi-robot system based on these criteria. There are benefits and drawbacks to each prospective approach. To simplify the allocation problem, having single task robots is preferable; however, multi-task robots are more capable. This is also similar for the question of designing specialized robots or interchangeable robots, as perhaps some could be designed to be capable of performing multiple tasks concurrently. It is also important to consider how complex disjoint tasks can be, such that robots could be assigned to scan the surrounding environment with visual sensors, while performing an analysis of the soil (assuming that specific visual sensor used for scanning wasn't needed for that). This could warrant having multiple separate task allocation systems, such that one would determine where to scan (i.e., the robots may need to watch a specific area), while another task allocation system could determine movement and environmental interaction.

The category of single robot tasks versus multiple robot tasks will typically lean towards the multiple robot tasks when dealing with self-replicating robots. For most applications, this depends on how the tasks are defined, such that if a task is as broad as 'gather resources to make more robots,' it is very likely that there will be multiple robots tasked with this at one time (the criteria is that some tasks involve multiple robots, not necessarily all of them).

The use of instantaneous assignment versus time-extended assignment is also implementation dependent, but choices will likely fall under the time-extended assignment category, to some degree. This is due to a certain number of future objectives and information being known, or predicted, at the time of assignment.

5.2. Centralized vs. Decentralized

With the centralized approach, each robot connects to a central robot that allocates the tasks to all the robots in the system. Each robot sends all the information they have to the central robot, which in turn evaluates this information and sends the appropriate commands to the other robots in the system, which execute their assigned tasks [67]. The advantages of this approach include the reduction of the duplication of effort, resources, and savings of cost and time [68]. Centralized systems suffer from a lack of redundancy, such that if the central robot fails, the whole system fails. Scalability is also restricted because of the reliance on the central robot, creating a bottleneck for communications and task allocation computation. In addition, centralized approaches generally have a higher communication overhead [69], although this is dependent on the implementation and the size of the system.

In the decentralized system type, each robot communicates relevant information to the other robots [68]. This allows administrative tasks to be dispersed among members of the system. The main advantage of the decentralized approach is redundancy. For instance, if one of the robots fails, the other robots can still work on their own and/or cooperatively with other robots in the system [67]. This approach also allows for a greater scalability, as adding additional robots does not necessarily overload the central communications, processing, and other capabilities of the system. A disadvantage to this approach is that the robots that make decisions may not receive all of the information gathered by the system, and thus, sub-optimal solutions may be produced. However, this could potentially be overcome by increasing the amount of shared information, and along with this, the level of communications required. Thus, a tradeoff between the communication overhead and solution quality exists [69].

The decision of which system type to use may follow the choice of centralized or decentralized replication. This is because, with centralized replication, the system is already impacted by the reliance of the system on a central hub, providing less incentive to use distributed task allocation.

5.3. Optimization vs. Market-Based

Optimization focuses on solving a certain problem, with the aim of finding the best solution for the problem, out of a set of available solutions, given certain decision-making metrics. Optimization techniques are used to maximize the ‘profit’ (maximization problem) or mitigate the ‘loss’ (minimization problem) of a prospective solution [70]. The set of available solutions is restricted by a set of constraints, and the optimum solution is chosen within these constrained solutions, based on certain criteria. These criteria define the objective function, which is a mathematical expression combining variables in order to describe the goal of the system [71].

Alternatively, market-based approaches use the concept of utility functions, which can represent the ability of the agents to measure interest in specific tasks for trading. In these systems, the utility functions show how the robot's skills can match the task's requirements [70].

Either approach could feasibly be used for decision-making for a self-replicating robot system. For instance, optimization techniques have been demonstrated to be effective in allocating resources for automated manufacturing [72].

6. Decision to Replicate

The decision of when a self-replicating robot system should replicate is affected by several factors. These factors include the available resources, replication equipment, current objectives, and robot capacity. These factors are depicted in Figure 1 and are discussed in the following subsections.

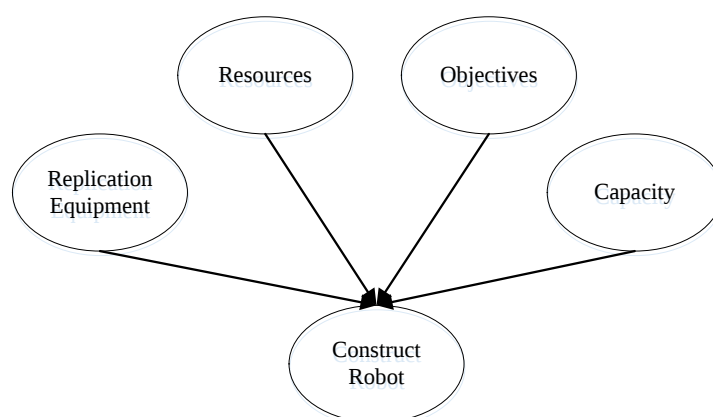


Figure 1. Key decision-making factors for the new robot construction decision.

6.1. Available Resources

Resource availability is critical to the decision to build a robot. Multiple factors, as shown in Figure 2, contribute to the resource availability characterization that is supplied to the decision-making algorithm. A requirement for fabricating a new robot is to have (or project having) the materials that are needed to build it. Another factor to consider is the quality/purity of the resources. Poor quality resources could impact the quality of the finished product, which (ideally) should be considered when deciding whether to move forward with the replication process.

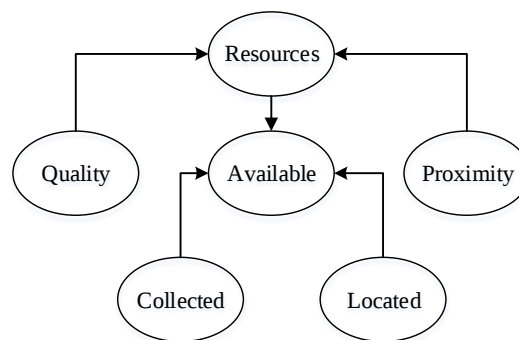


Figure 2. Mission resource characterization.

The acquisition of resources occurs through foraging. Robot foraging is broadly defined as robots searching for and collecting objects, and subsequently bringing them to a collection point [73]. Baldassano and Leonard [74] described measures of performance that can be used to allocate tasks for this purpose. Fibla and Bernardet [75] designed a strategy for robot foraging, based on the behavior of rodents. Cai [76] developed a learning algorithm to handle foraging tasks in completely unknown environments.

Resources can be divided into three categories: collected, located but not collected, and predicted to be in the environment but not located. Collected resources, as the name suggests, are resources that have been collected and stored by the robot system. However, a consideration in this category is the proximity of storage relative to where they would need to be used. The second category, resources that are located but not collected, encompasses resources that have been identified, with some degree of accuracy, to be present in specific locations in the environment. The degree of accuracy of this identification can significantly impact the level of reliance that is appropriate for resources in this category. The last category is resources that are predicted to be in the environment, but have not been located yet. This category's importance will vary in inverse proportion to how much of the environment is currently explored. For instance, at the early stages of exploration, the system may predict that there will be a certain amount of a certain material available, but as soon as the exploration effort progresses, the predicted resources will be transformed into located resources. As exploration is conducted, the resource prediction accuracy may also increase.

The process of predicting the resources available in a given environment may require prior knowledge of certain environmental traits, including geological and other information. To this end, Popa, Screenath, and Lewis [77] discussed algorithms for sampling an environment using mobile robots. Alternatively, certain environments can be continuously monitored by robots. Dunabin and Marques [78] detail the ways in which this has been done in the past, with one example being the monitoring of lakes for specific ecological changes.

Recognizing resources in a given environment can be accomplished in many ways. For example, reflection seismology (similar in concept to radar) has been used to discover oil and natural gas [79]. Magnetic surveys can be used to detect ore deposits [80]. The visual recognition of surface resources can be accomplished by processing images using trained deep convolutional neural networks [81]. The foregoing techniques could identify many of the resources required for robot replication. Additional techniques may be needed for robots with additional resource identification needs.

Once the resources are identified, the robots must be able to autonomously collect them. A number of prior experiments and applications demonstrate relevant capabilities. For example, Green and Vogt [82] proposed a multi-robot system that could cooperatively and autonomously mine ore using rock drills. Similarly, Shaffer and Stentz [83] tested a robotic system for coal mining that could autonomously navigate and reposition itself underground using a laser range finder. Hecker, Carmichael, and Moses [84] described a resource cluster prediction algorithm, inspired by ant foraging behavior, that exploited the natural clustering of resources to efficiently direct robots to find and collect them. Dunker et al. [85] demonstrated a proof of concept for utilizing teams of robots that could automatically gather regolith on the surface of the Moon with an actuated scoop. It would then

bring the collected materials to a central processing station. Each of the foregoing would be a capability that could be integrated into the self-replication robots' design to provide basic functionality. As with resource identification, the resource collection capabilities of a robot (both in terms of physical collection hardware and commanding software) may need to be augmented to support mission-specific collection requirements.

6.2. Replication Equipment

The replication equipment present needs to have the capability to produce the needed design, otherwise the process cannot move forward. There will typically be a probability of success associated with the equipment's capability to perform a given printing task. If this is known, it can be factored into the decision-making process. This chance of success can be calculated, based on a few factors, such as damage to the equipment and the limitations of equipment in pristine condition. Damage can be identified by measuring the results of production and comparing it to the expected results [86]. Even if the equipment can build the design, it may have known limitations or the potential to encounter errors. Issues can include the equipment jamming, printed parts not fitting together properly, and adverse environmental conditions.

6.3. Objectives

The current mission objectives, as shown in Figure 3, are relevant to the decision to build a new robot (or not). These objectives would determine the necessity for fabricating a new robot, and what design it would have. This information would drive the need (if required) for increasing the quantity of robots, or optimizing a design for a specific capability.

An increase in the quantity of robots may be needed for certain exploration efforts, or to support planned future robot fabrication predictions. An alternative consideration is that a design may be needed for a specific task that has a necessary benefit, such as reaching and collecting a resource that is out of reach of the current robots in the system (see the discussion of homogeneous vs heterogeneous selection, above for more on special designs).

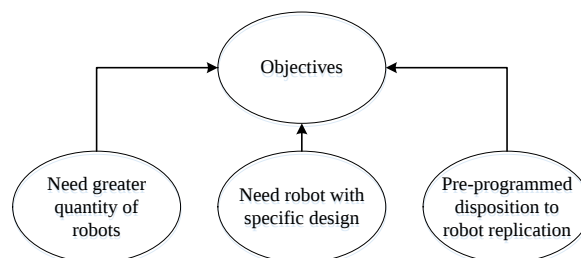


Figure 3. Generic objectives for robot creation.

6.4. Capacity

Depending on user choices or implementation and design constraints, it may be that the robot system is capped at having a certain maximum number of robots. The system capacity factor, depicted in Figure 4, characterizes the ability of the system to support more robots, to inform the build-or-not decision-making process. One example of a restriction would be having centralized command and communications, such that the number of robots that the central robot can command or communicate with is limited. Another restriction would be if robots needed a certain resource to continue functioning over time, such as energy or replacement parts. Finally, the number of robots that a system can have may also be restricted by the space available in the operating environment. Small spaces would obviously necessitate a fewer number of robots for optimal performance. This notion of an optimal number of robots as opposed to a maximum may even be a more general consideration.

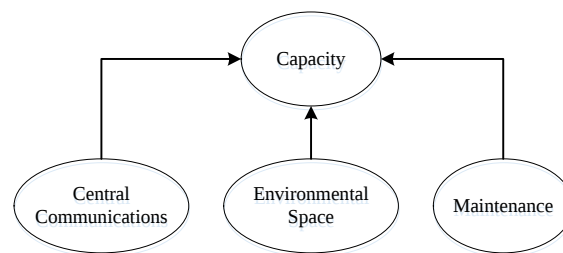


Figure 4. Example constraints on the maximum number of robots needed.

6.5. System Operations

The operations of the system are now considered, holistically. Figure 5 depicts the general decision-making process undertaken by each robot in the system. This diagram assumes a limited set of objectives, including repair robot, build robot, idle, gather resources, explore, and other mission-related objectives.

One point of interest is the flow of locating resources, to collecting resources, to having them available to use as materials (for new robots or for repairs to existing ones). The ‘explore’ objective locates resources and contributes to the terrain map. Located resources can subsequently be gathered by a robot that is pursuing the ‘gather resources’ objective. Gathering resources contributes to the available resources, which may then be used to build or repair robots. These different steps can be performed by the same robot or performed by different robots, depending on the circumstances. Communication amongst robots in the system updates these databases to reflect changes in resource status. This is important because this data affects the decision-making process, such as whether the system carries out a task or not. It also lets a robot know which tasks it has been allocated, based on the task allocation scheme. An example would be communicating that resources from the available resource pool are no longer available, as they were used to construct or repair a robot.

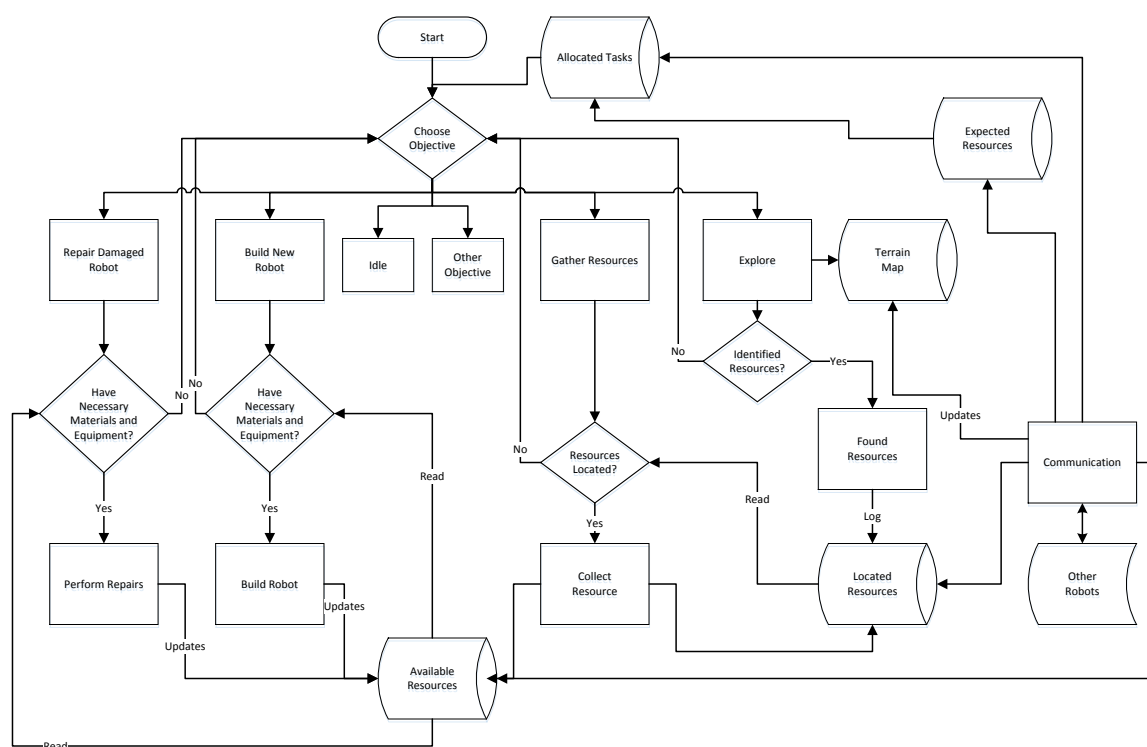


Figure 5. System decision-making diagram.

7. Qualitative Evaluation of the Proposed Approach

The proposed algorithm provides a general model of behavior for each robot. It is generic enough to be applied to a wide array of self-replicating robot systems, serving varied prospective applications. It can be implemented with either a centralized or decentralized paradigm, which can apply to matters of communication or replication. This flexibility makes it adaptable to various situations or user preferences. It also provides scalability, as adding additional robots does not cause the system to become overstrained. The way in which tasks are allocated remains an implementation decision, either due to mission criteria or hardware constraints.

The proposed algorithm assumes that, no matter what the overall mission of the multi-robot system is, it will always involve exploring the environment and gathering resources to replicate. This is based on the notion that one purpose of using a self-replicating robot system involves seeking and using in-situ resources. It also assumes that the capability to repair damaged robots (to some degree) is part of the system's functionality. To assess the proposed system's efficacy, several case studies are now considered. They are evaluated in terms of satisfactorily or unsatisfactorily fulfilling several key metrics. These metrics are:

- **Basic Feasibility:** Whether the proposed approach would be a good behavioral model for the example application.
- **Scalability:** The ease with which the system can handle the addition of new robots.
- **Survivability:** How likely the robots are to survive under adverse circumstances.
- **Exploration Capability:** The ease with which the example can explore and map an environment.
- **Security:** Whether the proposed approach is sufficiently secure for the application.
- **Performance Efficiency:** Whether the performance of the system is efficient or wasteful.
- **Resource Efficiency:** Whether the system can use resources optimally.
- **Proximity Operations Performance:** How well the robots work when they are operating in close proximity, for the application.
- **Distributed Operations Performance:** How well the robots work when they are operating over a wide area, for the application.
- **Cooperative Operations Performance:** How well the robots cooperatively work together, to achieve the application's objective.

Now the efficacy of the proposed approach is compared by assessing its utility for several application areas. The example application areas are discussed below. An overview of their compatibility with the proposed approach is provided in Table 1. This table considers the utility of the system for nine different mission concepts. The performance under each is ranked as 'S' if the proposed approach is satisfactory for the example. A 'U' indicates, on the other hand, that it is unsatisfactory for the application in this area.

Example 1: A self-replicating robot system could be used for planetary exploration. This would be especially useful due to the high cost of bringing materials to those sites. This example works with the proposed approach, due to having the overall mission objectives of exploring and resource gathering. Exploring the planet would provide a non-competitive resource source for the robots, such that they could use whatever they find without competition, promoting scalability. Printing robots could also achieve a faster exploration rate, compared to a fixed number of robots, as well as providing the opportunity to take more risks when exploring. Security wouldn't be a concern due to the non-contention environment of the robot system. The efficiency of the system would depend on the planet being explored and the resources available. The teamwork of the robots should be satisfactory.

Example 2: A second considered use for the system is for military purposes. A key benefit of this application is that lost robots could be replenished (by robots constructing more robots). This works with the proposed approach because materials may be easy to find in combat zones, and the need to explore is usually present, for intelligence gathering and other purposes. The security of this use is very low due to the entire purpose of this system being to engage in combat missions. This may

necessitate more thought being devoted to system security. The efficiency of the system depends on the combat environment and the capabilities of individual robots.

Example 3: Self-replicating nano-robots that can repair larger structures are now considered. This application fits the proposed approach due to the need for exploration and resource gathering for repair materials. The scalability is quite high due to the low amount of material needed to build a small robot. System security needs of this use are low, due to the difficulty (for an attacker) in keeping track of all the nano-bots and their ability to cause harm to individuals. The efficiency of the system and cooperation is satisfactory, due to the robots' ability to work well as a team, as well as alone.

Example 4: Self-replicating visual sensor robots could be tasked with surveying an environment. The proposed approach would perform well, due to the already existent exploration component task. It would also be scalable, because having redundant environmental sensors is a reasonably smart practice to ensure up-time. System security wouldn't be a typical issue, unless it was used for intelligence purposes. The efficiency of gathering resources would be lackluster, due to the presumed low mobility of the robots in this example. System teamwork would be satisfactory (across the board), providing constant performance across varying circumstances.

Example 5: The proposed approach's efficacy for the application of self-replicating satellites that use orbital debris to construct more satellites is considered. The proposed approach may not perform well due to the limited mobility of most satellites, as well as the availability of materials. It would likely not be scalable due to possible distance, line of sight, and communication obstructions. It would present a security risk due to high value asset proximity operations and the high surveillance capability of that vantage point. The performance would generally be satisfactory (as long as it has suitable equipment for dealing with the difficult conditions in space). Craft cooperation would largely be limited to communications and not proximal teamwork, for this application.

Example 6: A system of self-replicating robots tasked with collecting garbage is feasible to implement with the proposed approach. The collected garbage may even be able to be used as material for the robots to replicate with, which would promote scalability. The exploration usefulness, for this example, may be hampered in favor of locating and collecting garbage in a predetermined area. Cooperation, however, would be needed for efficiency and support, for when multiple robots are needed to collect a heavy or otherwise difficult to remove piece of garbage.

Example 7: The use of self-replicating transportation vehicles for commercial use is now evaluated. This example would work well with the proposed approach due to operating in already mapped terrain, such that the transportation would be across frequently visited paths. It would be scalable enough to provide suitable transportation to passengers. The utility of replication capabilities, on a distributed basis, is questionable. Its cooperative teamwork would depend on implementation and design.

Example 8: Robots that collect energy may be implemented as self-replicating, to increase the amount of energy gathered. While the proposed approach is feasible for this example, it would lack the necessity to explore the environment and foraging requirements for replication. Cooperation may be hampered by the lack of exploration goals and perhaps even result in robots in competition for resources, such that proximity operations would become inefficient.

Example 9: Finally, the proposed approach's utility for self-replicating and reconfiguring modular robots that bond together to form shapes is considered. The amount that this use would benefit from the proposed approach is possibly limited by the specific area of application. However, this application would be very scalable and highly cooperative, with an emphasis on proximity operations. In contrast, the performance of this example when the robots are distributed would be poor, due to their modular nature.

The ability of the robot system to switch capabilities (and thus missions), by printing new robots, is also of interest due to the dynamic nature of many prospective operation environments. This can lead to a drastic shift of purpose, such that vastly different missions can be carried out with the same robot system, after it undergoes a design configuration change.

Table 1. Examples (described above) evaluated by performance characteristics. ‘S’ indicates that the proposed approach is satisfactory for the example, while ‘U’ indicates that it is unsatisfactory.

Examples	Basic Feasibility	Scalability	Survivability	Exploration Capability	Security	Performance Efficiency	Resource Efficiency	Proximity Operations Performance	Distributed Operations Performance	Cooperative Operations Performance
Planetary Exploration	S	S	S	S	S	S	S	S	S	S
Military	S	S	S	S	U	S	S	S	S	S
Nano-Repair bots	S	S	S	U	U	S	S	S	S	S
Sensor Network	U	S	S	S	S	S	S	U	S	S
Satellites	U	U	S	U	U	S	U	U	S	U
Garbage Collection	S	S	S	U	S	S	S	S	S	S
Transportation	S	S	U	S	S	S	S	U	S	S
Energy Harvesting	S	S	S	U	S	S	S	U	S	U
Modular Reforming	U	S	S	U	S	S	S	S	U	S

8. Conclusions and Future Work

The concepts, design, and coordination of self-replicating robot systems have been discussed in this paper and a high-level algorithm was proposed and qualitatively assessed. The increasing capabilities of 3D printers are opening the door to feasible 3D printing-based self-replicating robots. This work stems from the general notion of a need for a machine that can assemble anything, like a universal constructor.

There is a need for software that can command these robots, despite their potential for differing designs and capabilities. An interface could be used to determine hardware capabilities, which would then be communicated to the general behavioral command program (the algorithm that was presented herein).

The benefits of a self-replicating robot system stem from being able to replenish lost robots and the capability to create new ones. However, there are issues that arise during the development of these machines, such as the ethical, environmental, and societal ramifications. These issues may need to be adequately addressed before such robot systems become available, and future work plans to discuss this.

Acknowledgments: Facilities, materials, and equipment used for this work have been supplied by the North Dakota State University (NDSU) Department of Computer Science and NDSU College of Science and Math.

Author Contributions: A.J. and J.S. conceived and designed the evaluation approach; A.J. performed the majority of the evaluation work; A.J. and J.S. wrote the paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Sipper, M. Fifty Years of Research on Self-Replication: An Overview. *Artif. Life* **1998**, *4*, 237–257.
2. Ellery, A. Are Self-Replicating Machines Feasible? *J. Spacecr. Rockets* **2016**, *53*, 317–327.
3. Beuchat, J.-L.; Haenni, J.O. Von Neumann's 29-state cellular automaton: A hardware implementation. *IEEE Trans. Educ.* **2000**, *43*, 300–308.
4. Kemeny, J.G. Man Viewed as a Machine. *Sci. Am.* **1955**, *192*, 58–67.
5. Von Neumann, J. *The Theory of Self Reproducing Automata*, 1st ed.; Burks, A.W., Ed.; University of Illinois Press: Champaign, IL, USA, 1966.
6. Whitesides, G.M.; Grzybowski, B. Self-assembly at all scales. *Science* **2002**, *295*, 2418–2421.
7. Butler, Z.; Murata, S.; Rus, D. Distributed Replication Algorithms for Self-Reconfiguring Modular Robots. *Proc. Distrib. Auton. Robot. Syst.* **2002**, *5*, 37–48.
8. Sahin, E.; Labella, T.H.; Trianni, V.; Deneubourg, J.-L.; Rasse, P.; Floreano, D.; Gambardella, L.; Mondada, F.; Nolfi, S.; Dorigo, M. SWARM-BOT: Pattern Formation in a Swarm of Self-Assembling Mobile Robots. In Proceedings of the 2002 IEEE International Conference on Systems, Man and Cybernetics, Hammamet, Tunisia, 6–9 October 2002.
9. Jakobsen, E.; Tannum, C.J. Self-assembly Mechanisms for Evolutionary Robotics. Master's Thesis, Norwegian University of Science and Technology, Trondheim, Norway, 2016.
10. Koza, J.R.; Rice, J.P. Process for Problem Solving Using Spontaneously Emergent Self-Replicating and Self-Improving Entities. U.S. Patent 5,390,282, 14 February 1995.
11. Tuci, E.; Dorigo, M. Cooperation Through Self-Assembly in Multi-Robot Systems. *ACM Trans. Auton. Adapt. Syst.* **2006**, *1*, 115–150.
12. Groß, R.; Dorigo, M.; Yamakita, M. Self-assembly of Mobile Robots: From Swarm-bot to Super-mechano Colony. In *Intelligent Autonomous Systems 9—IAS 9*; IOS Press: Amsterdam, The Netherlands, 2003; pp. 487–496.
13. Pfeifer, R.; Lungarella, M.; Iida, F. Self-organization, embodiment, and biologically inspired robotics. *Science* **2007**, *318*, 1088–1093.
14. Lee, K.; Moses, M.; Chirikjian, G.S. Robotic Self-replication in Structured Environments: Physical Demonstrations and Complexity Measures. *Int. J. Rob. Res.* **2008**, *27*, 387–401.

15. Sanderson, A. Parts entropy methods for robotic assembly system design. In Proceedings of the 1984 IEEE International Conference on Robotics and Automation, Atlanta, GA, USA, 13–15 March 1984; Volume 1, pp. 600–608.
16. Suthakorn, J.; Kwon, Y.T.; Chirikjian, G.S. A Semi-Autonomous Replicating Robotic System. In Proceedings of the 2003 IEEE International Symposium on Computational Intelligence in Robotics and Automation, Kobe, Japan, 16–20 July 2003.
17. Suthakorn, J.; Zhou, Y.; Chirikjian, G. Self-Replicating Robots for Space Utilization. In Proceedings of the 2002 Robosphere Workshop on Self Sustaining Robotic Ecologies, Mountain View, CA, USA, 14–15 November 2002.
18. Suthakorn, J.; Cushing, A.B.; Chirikjian, G.S. An Autonomous Self-Replicating Robotic System. In Proceedings of the 2003 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM 2003), Kobe, Japan, 20–24 July 2003.
19. Zykov, V.; Mytilinaios, E.; Adams, B.; Lipson, H. Robotics: Self-reproducing machines. *Nature* **2005**, *435*, 163–164.
20. Gilpin, K.; Rus, D. What's in the bag: A distributed approach to 3D shape duplication with modular robots. In Proceedings of Robotics: Science and Systems, Sydney, Australia, 9–13 July 2012.
21. Jones, R.; Haufe, P.; Sells, E.; Iravani, P.; Olliver, V.; Palmer, C.; Bowyer, A. RepRap—The replicating rapid prototyper. *Robotica* **2011**, *29*, 177–191.
22. Stroud, A.B.; Morris, M.; Carey, K.; Williams, J.C.; Randolph, C.; Williams, A.B. MU-L8: The Design Architecture and 3D Printing of a Teen-Sized Humanoid Soccer Robot. In Proceedings of the 8th Workshop on Humanoid Soccer Robots, IEEE-RAS International Conference on Humanoid Robots, Atlanta, GA, USA, 15 October 2013.
23. Kading, B.; Straub, J. Utilizing in-situ resources and 3D printing structures for a manned Mars mission. *Acta Astronaut.* **2015**, *107*, 317–326.
24. Cesaretti, G.; Dini, E.; De Kestelier, X.; Colla, V.; Pambaguian, L. Building components for an outpost on the Lunar soil by means of a novel 3D printing technology. *Acta Astronaut.* **2014**, *93*, 430–450.
25. Chirikjian, G.S.; Zhou, Y.; Suthakorn, J. Self-Replicating Robots for Lunar Development. *IEEE/ASME Trans. Mechatron.* **2002**, *7*, 462–472.
26. Walters, P.; McGoran, D. Digital fabrication of “smart” structures and mechanisms: Creative applications in art and design. In *Society for Imaging Science and Technology, Proceedings of International Conference on Digital Printing Technologies and Digital Fabrication, Minneapolis, Minnesota*, 2–6 October 2011; pp. 185–188.
27. Rossiter, J.; Walters, P.; Stoimenov, B. Printing 3D dielectric elastomer actuators for soft robotics. *Proc. SPIE* **2009**, 7287, 1–10.
28. Rus, D.; Tolley, M.T. Design, fabrication and control of soft robots. *Nature* **2015**, *521*, 467–475.
29. Hiller, J.; Lipson, H. Automatic Design and Manufacture of Soft Robots. *IEEE Trans. Robot.* **2012**, *28*, 457–466.
30. Bartlett, N.W.; Tolley, M.T.; Overvelde, J.T.B.; Weaver, J.C.; Mosadegh, B.; Bertoldi, K.; Whitesides, G.M.; Wood, R.J. Robot Powered By Combustion. *Science* **2015**, *349*, 161–165.
31. Yan, Z.; Jouandeau, N.; Cherif, A.A. A survey and analysis of multi-robot coordination. *Int. J. Adv. Robot. Syst.* **2013**, *10*, 399.
32. Nieto-Granda, C.; Rogers, J.G.; Christensen, H.I. Coordination strategies for multi-robot exploration and mapping. *Int. J. Rob. Res.* **2014**, *33*, 519–533.
33. Portugal, D.; Rocha, R.P. Distributed multi-robot patrol: A scalable and fault-tolerant framework. *Rob. Auton. Syst.* **2013**, *61*, 1572–1587.
34. Liu, H.; Stoll, N.; Junginger, S.; Thurow, K. Mobile robot for life science automation. *Int. J. Adv. Robot. Syst.* **2013**, *10*, 1–14.
35. Pennisi, A.; Previtali, F.; Gennari, C.; Bloisi, D.D.; Iocchi, L.; Ficarola, F.; Vitaletti, A.; Nardi, D. *Multi-Robot Surveillance through a Distributed Sensor Network*; Springer International Publishing: Cham, Switzerland, 2015; Volume 604.
36. Starke, G.; Hahn, D.; Pedroza Yanez, D.; Ugalde Leal, L. Self-Organization and Self-Coordination in Welding Automation with Collaborating Teams of Industrial Robots. *Machines* **2016**, *4*, 23.
37. Caliskanelli, I.; Broecker, B.; Tuyls, K. Multi-Robot Coverage: A Bee Pheromone Signalling Approach. In *Artificial Life and Intelligent Agents Symposium*; Springer International Publishing: Cham, Switzerland, 2015; pp. 124–140.

38. Straub, J. A Distributed Blackboard Approach Based Upon a Boundary Node Concept. *J. Intell. Robot. Syst.* **2016**, *82*, 467–478.
39. Julian, B.J.; Angermann, M.; Schwager, M.; Rus, D. Distributed robotic sensor networks: An information-theoretic approach. *Int. J. Rob. Res.* **2012**, *31*, 1134–1154.
40. Zhou, J.; Wu, X.; Liu, Z. Distributed coordinated adaptive tracking in networked redundant robotics systems with a dynamic leader. *Sci. China Technol. Sci.* **2014**, *57*, 905–913.
41. Navarro, I.; Matia, F. An Introduction to Swarm Robotics. *ISRN Robot.* **2013**, *2013*, 1–10.
42. Sahin, E. Swarm robotics: From sources of inspiration to domains of application. *Swarm Robot.* **2004**, doi:10.1007/978-3-540-30552-1_2.
43. Abukhalil, T.; Sobh, T.; Patil, M. Survey on decentralized modular robots and control platforms. *Lect. Notes Electr. Eng.* **2015**, *313*, 165–175.
44. Barca, J.C.; Sekercioglu, Y.A. Swarm robotics reviewed. *Robotica* **2013**, *31*, 345–359.
45. Brambilla, M.; Ferrante, E.; Birattari, M.; Dorigo, M. Swarm robotics: A review from the swarm engineering perspective. *Swarm Intell.* **2013**, *7*, 1–41.
46. Altshuler, Y.; Bruchstein, A.M.; Wagner, I.A. Swarm robotics for a dynamic cleaning problem. In Proceedings of the 2005 IEEE Swarm Intelligence Symposium (SIS 2005), Pasadena, CA, USA, 8–10 June 2005; Volume 2005, pp. 217–224.
47. Mohan, Y.; Ponnambalam, S.G. An extensive review of research in swarm robotics. In Proceedings of the World Congress on Nature & Biologically Inspired Computing (NaBIC 2009), Coimbatore, India, 9–11 December 2009; pp. 140–145.
48. Campo, A.; Dorigo, M. Efficient multi-foraging in swarm robotics. *European Conference on Artificial Life*; Springer: Berlin/Heidelberg, Germany, 2007; pp. 696–705.
49. Mondada, F.; Gambardella, L.M.; Floreano, D.; Nolfi, S.; Deneubourg, J.L.; Dorigo, M. The cooperation of swarm-bots: Physical interactions in collective robotics. *IEEE Robot. Autom. Mag.* **2005**, *12*, 21–28.
50. Iqbal, J.; Islam, R.U.; Abbas, S.Z.; Khan, A.A.; Ajwad, S.A. Automating industrial tasks through mechatronic systems—A review of robotics in industrial perspective. *Teh. Vjesn. Tech. Gaz.* **2016**, *23*, 917–924.
51. Leitão, P.; Colombo, A.W.; Karnouskos, S. Industrial automation based on cyber-physical systems technologies: Prototype implementations and challenges. *Comput. Ind.* **2016**, *81*, 11–25.
52. Saliba, M.; Zammit, D.; Azzopardi, S. Towards practical, high-level guidelines to promote company strategy for the use of reconfigurable manufacturing automation. *Robot. Comput.-Integr. Manuf.* **2016**, in press.
53. Sadrfaridpour, B.; Saeidi, H.; Burke, J.; Madathil, K.; Wang, Y. Modeling and Control of Trust in Human-Robot Collaborative Manufacturing. In *Robust Intelligence and Trust in Autonomous Systems*; Springer: Boston, MA, USA, 2016; pp. 115–141.
54. Vogel-Heuser, B.; Schütz, D.; Frank, T.; Legat, C. Model-driven engineering of Manufacturing Automation Software Projects—A SysML-based approach. *Mechatronics* **2014**, *24*, 883–897.
55. Vyatkin, V.B. Software engineering in industrial automation: State-of-the-art review. *IEEE Trans. Ind. Inform.* **2013**, *9*, 1234–1249.
56. Matthias, B.; Kock, S.; Jerregard, H.; Kallman, M.; Lundberg, I. Safety of collaborative industrial robots: Certification possibilities for a collaborative assembly robot concept. In Proceedings of the 2011 IEEE International Symposium on Assembly and Manufacturing (ISAM), Tampere, Finland, 25–27 May 2011; pp. 1–6.
57. Leyh, W. Experiences with the construction of a building assembly robot. *Autom. Constr.* **1995**, *4*, 45–60.
58. Quaid, A.E.; Hollis, R.L. Cooperative 2-DOF robots for precision assembly. In Proceedings of IEEE International Conference on Robotics and Automation, Minneapolis, MN, USA, 22–28 April 1996; Volume 3, pp. 2188–2193.
59. Bolger, A.; Faulkner, M.; Stein, D.; White, L.; Seung-kook Yun; Rus, D. Experiments in decentralized robot construction with tool delivery and assembly robots. In Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, Taipei, Taiwan, 18–22 October 2010; pp. 5085–5092.
60. Hatna, A.; Grieve, R.J.; Broomhead, P. Automatic CNC milling of pockets: Geometric and technological issues. *Comput. Integr. Manuf. Syst.* **1998**, *11*, 309–330.
61. Lohn, J.; Hornby, G. Evolvable hardware using evolutionary computation to design and optimize hardware systems. *IEEE Comput. Intell. Mag.* **2006**, *1*, 19–27.

62. Jiang, W.; Liu, Y.; Gong, S.; Hong, T. Application of bionics in antenna radar cross section reduction. *IEEE Antennas Wirel. Propag. Lett.* **2009**, *8*, 1275–1278.
63. Dickinson, M.H. Bionics: Biological insight into mechanical design. *Proc. Natl. Acad. Sci. USA* **1999**, *96*, 14208–14209.
64. Fisher, R. The genetical theory of natural selection. *Dover Publ.* **1958**, *22*, 127–130.
65. Klauk, H. Organic Electronics: Materials, Manufacturing, and Applications; John Wiley & Sons: Hoboken, NJ, USA, 2006.
66. Gerkey, B.P.; Matarì, M.J. A formal analysis and taxonomy of task allocation in multi-robot systems. *Intl. J. Robot. Res.* **2004**, *23*, 939–954.
67. Khamis, A.; Hussein, A.; Elmogy, A. Multi-robot task allocation: A review of the state-of-the-art. *Stud. Comput. Intell.* **2015**, *604*, 31–51.
68. Horling, B.; Lesser, V. A survey of multi-agent organizational paradigms. *Knowl. Eng. Rev.* **2005**, *19*, 281.
69. Gerkey, B.P. *On Multi-Robot Task Allocation*. CRES Technical Report CRES-03-012; Center for Robotics and Embedded Systems, University of Southern California, Los Angeles, CA, USA, 2003; pp. 3–12.
70. Badreldin, M.; Hussein, A.; Khamis, A.; Badreldin, M.; Hussein, A.; Khamis, A. A Comparative Study between Optimization and Market-Based Approaches to Multi-Robot Task Allocation. *Adv. Artif. Intell.* **2013**, *2013*, 1–11.
71. Horst, R.; Pardalos, P.M.; Nguyen, V.T. *Introduction to Global Optimization*; Springer: New York, NY, USA, 2000.
72. Gen, M.; Lin, L. Multiobjective evolutionary algorithm for manufacturing scheduling problems: State-of-the-art survey. *J. Intell. Manuf.* **2014**, *25*, 849–866.
73. Winfield, A.F. Foraging robots. *Encycl. Complex. Syst. Sci.* **2009**, *6*, 3682–3700.
74. Baldassano, C. Explore vs. exploit: Task allocation for multi-robot foraging. *Preprint* **2009**, 10–11.
75. Fibla, M.; Bernardet, U. Allostatic control for robot behaviour regulation: An extension to path planning. In *Intelligent Robots and Systems (IROS), Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Taipei, Taiwan, 18–22 October 2010; pp. 1935–1942.
76. Cai, Y. Intelligent Multi-Robot Cooperation for Target Searching and Foraging Tasks in Completely Unknown Environments. Ph.D. Thesis, University of Guelph, Guelph, ON, Canada, 2013.
77. Popa, D.; Sreenath, K.; Lewis, F. Robotic deployment for environmental sampling applications. In *Proceedings of the ICCA 2015 International Conference on Control and Automation*, Budapest, Hungary, 26–29 June 2005.
78. Dunbabin, M.; Marques, L. Robots for environmental monitoring: Significant advancements and applications. *IEEE Robot. Autom. Mag.* **2012**, *19*, 24–39.
79. Waters, K.H. *Reflection Seismology: A Tool for Energy Resource Exploration*, 3rd ed.; Wiley: Hoboken, NJ, USA, 1987.
80. National Research Council. *Evolutionary and Revolutionary Technologies for Mining*; National Academies Press: Washington, DC, USA, 2002; Volume 53.
81. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444.
82. Green, J.J.; Vogt, D. A Robot Miner for Low Grade Narrow Tabular Ore Bodies: The Potential and the Challenge. In *Proceedings of the 3rd Robotics and Mechatronics Symposium (ROBMECH2009)*, Pretoria, South Africa, 8–10 November 2009.
83. Shaffer, G.K.; Stentz, A. A robotic system for underground coal mining. In *Proceedings of the 1992 IEEE International Conference on Robotics and Automation*, Nice, France, 12–14 May 1992; pp. 633–6380.
84. Hecker, J.P.; Carmichael, J.C.; Moses, M.E. Exploiting clusters for complete resource collection in biologically-inspired robot swarms. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, Hamburg, Germany, 28 September–2 October 2015; pp. 434–440.
85. Dunker, P.A. A Biologically Inspired Robot for Lunar Exploration and Regolith Excavation. Master's Thesis, Case Western Reserve University, St. Louis, MO, USA, 2009.
86. Russell, M.; Straub, J. Software Design for an Intelligent Attitude Determination and Control System. Research Experience for Undergraduates Poster Session at the AIAA/USU Conference on Small Satellites, Logan, UT, USA, 8–13 August 2015.

