

# Towards A Team of Robots with Reconfiguration and Repair Capabilities

Curt Bererton<sup>1</sup> and Pradeep K. Khosla<sup>2</sup>

In the future, we propose that there will be largely self-sufficient robot colonies operating on distant planets and in harsh environments here on earth. A highly desirable quality of such a colony would be the capability of the robots to repair each other. Towards the goal of autonomous repair, we have designed a robot that can replace the modules composing a similar robot. The final system was teleoperated and module removal and replacement was performed on a test bed. We discuss some of the design trade-offs for such a system and discuss some of the steps required in order to develop a self-sufficient robot colony.

## I. INTRODUCTION

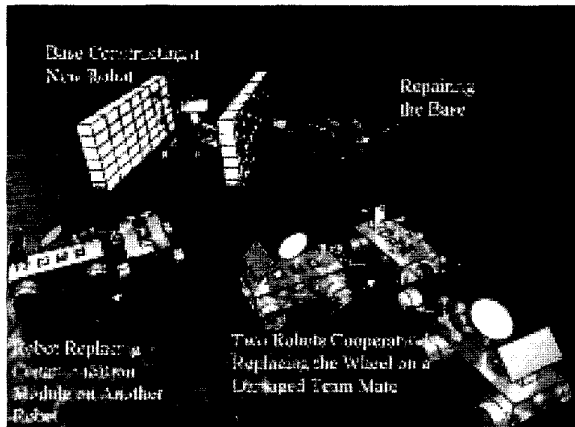


Figure 1: A self-sufficient team of robots

There are several applications for robots capable of cooperative repair. Our main vision is of a colony of robots on Mars (or another planet) preparing a station for human habitation. NASA's technology plan [20] lists self-diagnosing and self-repairing systems as one of the technologies necessary for self-sustained long duration human operations. Such a colony would need to operate without outside assistance for extended periods of time, although there is the possibility for some amount of teleoperation of the robots if humans are present. Other applications include operation in any kind of harsh environment where human intervention is either costly or dangerous. A team with cooperative repair capabilities

would be able to operate longer and more efficiently than a comparable team without such capabilities.

Repairable robots are also useful in non-automated repair tasks. If a robot can be repaired by another robot, then it is likely that the task will be trivial for a human technician. This can lead to smaller downtimes in factory settings as well as decreasing the maintenance cost of robotic systems in general.

In this paper, we present a design for a small mobile robot that has a standard wheeled base, and yet also has the capacity to replace modules on teammates as well as have its own modules be replaced. This type of module replacement is a key step towards *cooperative repair*. In our implementation, the robots also have the ability to perform *cooperative reconfiguration*, by which we mean that different module types can be placed on the robots. The concept of modular redundancy leading to hardware availability is an extremely old concept [5]. Cooperative repair uses modular redundancy between robots so that one need not replicate individual components on a given robot to achieve dependability. The components of failed teammates provide the modular redundancy we require for a dependable robot team.

The closest related work to the concept of systems capable of cooperative repair/reconfiguration is that of metamorphic or reconfigurable robots. The primary difficulties with these systems are due to the exponential increase in planning complexity for a large set of modules [6]. Some of these systems have the capability of self-repair, in that they can expel failed modules and continue to function [7,8]. The key difference is that the above systems must carry the redundant modules with them even if they don't serve any purpose. These modules can then be used to perform self-repair when a module breaks. In our design, the redundancy comes from the other teammates. All modules are in use until that robot fails. These modules can then be used to repair the next robot that fails. Although reconfigurable robots may one day be the solution to many problems, they will be useless in practical situations until an effective means to power them has been developed. All of the reconfigurable and metamorphic robots built to date either use tethered power or quickly drain any battery-powered supply.

It is generally the case that systems made for a specific task are usually cheaper and more efficient. However, in applications where the exact task and environment is not known before hand, a team of robots with repair and reconfiguration capabilities will yield a more versatile solution. The advantages of such a system are

1: Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, [curt@ri.cmu.edu](mailto:curt@ri.cmu.edu)

2: Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA, [pkk@ece.cmu.edu](mailto:pkk@ece.cmu.edu)

similar to those promised by metamorphic robots [6]: Versatility; in that robots can be autonomously reconfigured to optimally perform a task given a limited set of functional modules. Reliability; in that as the number of modules increases in any system, be it a team of robots or one very expensive robot with a large amount of redundancy, the probability of at least one component failing as the number of components increases goes to one. Low Expense; were such a team to be built, a large number of identical modules would lead to decreased production cost per module. The design of such a system also avoids some of the main difficulties with metamorphic systems in that complexity does not directly proportional with the number of modules. Placing the modules on separate robots gives a natural hierarchy that simplifies control and repair. Multi-robot research [6,9,10,11,12] can then be applied to the control of the team as a whole. Perhaps the most appealing feature of this type of system is that redundant components are actually in use prior to a failure. Therefore, no components are sitting idle waiting to take over when another component fails and no productivity or expense is wasted carrying redundant components that are never used.

Repair in dependability and reliability literature is known as *fault removal* [1,2]. In order for fault removal to occur in a self-sufficient robot colony, the robots must first determine that there is some fault present. Once a fault is known to exist, one must then determine the location and nature of such a fault. Action is then taken to remove the fault. Of these three steps: *verification*, *diagnosis*, and *correction*, we have concentrated on the correction aspect. In our case, correction corresponds to replacing a faulty module in a modular design. Firstly, we will discuss the system design itself. Secondly, we will list the desirable features of a system for an automated module replacement task.

## II. SYSTEM DESIGN

As with any system, there are several main aspects to the design. We separated these into mechanical design, electrical design, and software design. Here we concentrate primarily on the mechanical and electrical design. In each section we will discuss the design trade-offs that were made. Another objective of this work was also to make the robot as small as possible, although more specifics of miniature design are discussed in [3].

In designing a robot capable of module replacement there are a large number of trade-offs. Having had experience with designing both large and small robots [3,13,14,15,16,17,18], we believe that it is more difficult to build a robot that is 10cm long than one that is a meter long. This is due mainly to the fact that nearly all of the parts had to be custom designed for prototype that was built.

There were also several assumptions made during the design process. We attempted to concentrate more on the general theory of module replacement rather than building the most capable and robust robot that might fly on a space mission. Thus, to simplify design, we assumed that the robots would be working on a flat workspace such as an office carpet, a concrete floor, or desktop. We also assumed that the floor would be devoid of obstacles that cannot be

seen by the robot sensors. Other assumptions will be mentioned in the appropriate sections.

One of the most important features of a team of repairable robots is that one should not assume that the robot being repaired can assist in the fault removal process. If the robot can in fact assist, then this simplifies the problem, however, in most situations (such as power failure) the robot being repaired will be unable to assist in any of the three steps of fault removal. Thus, when the electrical and mechanical systems were designed for module replacement we attempted to keep the design free of inter-robot dependencies.

Though one should not rely on a damaged robot to assist in the repair task, one should attempt to use any remaining capabilities it has to facilitate the procedure. For example, an external system should be able to verify (in some manner) all components of the faulty robot. If the faulty robot has maintained enough functionality however, then it may be able to perform its own internal verification and fault diagnosis. These results can then be communicated to the repair robot or repair system.

### A. Functional Module Division

The most crucial design decision we faced was to decide which components of the robots were to be replaceable. Ideally, any component of the system should be replaceable. The main difficulty with this approach is that the implementation is extremely difficult. One could envision a current mobile robot design being used today. Add to such a robot a high degree of freedom manipulator that could solder and unsolder any electrical component as well as replace any mechanical component on another robot of the same design. This design might well work, but in requiring so much accuracy and strength in the manipulator, the design of the robot has become immensely complicated.

In the earlier phase of the design process, we determined that the most feasible solution was to divide the robot into replaceable components according to their primary function. These components would then be grouped both physically and electrically into *modules*. Of course "primary function" is a vague term. In our case, we chose a two-dimensional mapping and navigation task as our motivation. A common task in the mobile robot domain [19], this gave some insight as to what primary functions were required by the robot.

As in any team-mapping task, the typical format to map a new area follows a set of basic steps. Go to an unexplored area, take a sensor reading, communicate with teammates and decide where to go next. This led to several apparent primary functions: locomotion, sensing, and communication. We added two separate functions, namely main processing and power, in order to complete the functionality required by the robot. In order to achieve repair, we also required the ability to manipulate the components that must be replaced. This led to the following list of primary functions: Locomotion, Sensing, Power, Communications, Main processing, and Manipulation. Ideally, each one of these primary functions should be part of a replaceable component. Although a design for such a robot

was considered, the implementation of such a robot required a significant expenditure of resources. As we are interested more in the theory of cooperative repair we decided to implement a relatively simple system.

Functional module division simplifies all three of the steps required for fault removal. Verification is simplified because we need only verify one module of the robot at a time. When verification fails, one will typically know which module is at fault and hence diagnosis is usually easier. If verification fails, we need not know exactly which component in the module has failed. The knowledge that anything is wrong with a particular module is sufficient to repair the robot, since we need only replace that functional module. This is not strictly true as there are and always will be dependencies between the modules. In general however, the division of the robot into functional modules simplifies any given verification and diagnosis procedure. Functional division also allows a 'swap and test' procedure for verification. If the fault cannot be narrowed to a single module, each module can be removed and tested in a separate robot. This allows for external verification of each removable module, and also a 'divide and conquer' scheme for fault diagnosis.

The prototype is presented in Figure 2. In this system we combined the locomotion and manipulation into one module that cannot be replaced (henceforward this module will be referred to as the 'chassis'). Shown on the robot are three of the modules that have been implemented. From front to back these modules are: Main processing, Communication, and the power module. Of primary concern was the one degree of freedom manipulator on the front of the robot. The manipulator has linear motion in the z direction thus leading to the name "forklift". To perform module replacement, the two forklift pins are inserted into a module receptacle that is used to lift the module off of the chassis. In order to place a working module on the robot, the procedure is reversed.

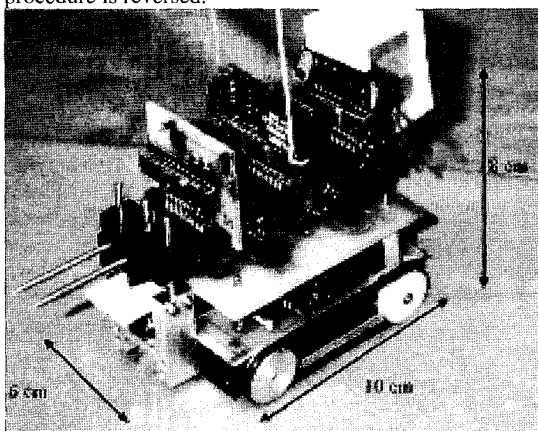


Fig 2. Prototype chassis with three functional modules.

### B. Mechanical Design

In Figure 2 we see a prototype for the repair robot. The length of the base is approximately 10cm long. Note that all actuation is contained in the chassis. The chassis is skid

steered via two MicroMo DC gear-motors with integrated encoders and a third identical motor powers the forklift on the front. Note that in this design the chassis cannot be replaced. There were two primary concerns for the mechanical design of this system. First, the robot required a mobile base with which to move. Second, it required some means of replacing the functional modules that were implemented.

The chassis design is relatively straightforward. It is a simple skid steered base. The two drive motors were mounted inside the chassis itself, leaving more space for electronics. The idlers on each side are also contained inside the chassis and are made from Delrin to minimize friction. This design has several disadvantages including the skids' tendency to pull off when driving in thick carpet and the poor dead reckoning accuracy when performing turns. The only other feature of interest in the chassis is the mounting holes with which the forklift or a towing attachment can be mounted.

The more challenging design decision was the implementation of the removal and replacement of modules. Our main objective was to make the module exchange process as simple as possible, and thus more robust to errors. Our secondary goal was to make the required mechanisms as simple as possible so they would be easy to manufacture and more robust. We have made all replaceable components mechanically identical. All modules have identical connectors and docking receptacles.

To achieve truly interchangeable modules, the connector design was crucial. We chose to use commercially available connectors available from Hypertronics. The main feature of these connectors is that they had guide pins on either side of the connectors that allowed for error in module positioning when inserting a module into the robot. Figure 3 is a close up view of a male connector with the guide pins.

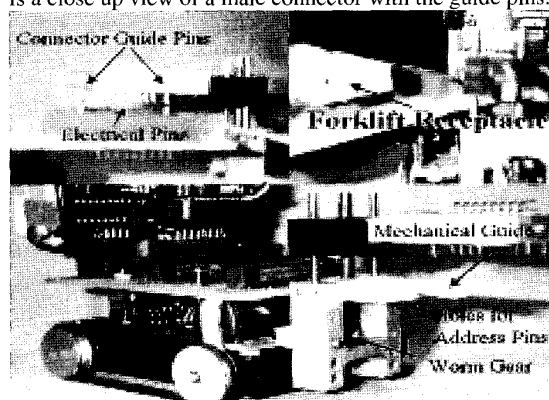


Fig 3. Close up of connectors and forklift receptacle.

Figure 2 and 3, indicate that the modules are removed and inserted using the forklift mechanism on the front of the robot. The forklift has only one degree of freedom in the Z-axis, thus the modules are inserted and removed by pulling up or pushing down with the forklift. The drive chain for forklift consists of the dc gear motor followed by a 15:1 reduction at the worm/worm gear interface. The worm gear then turns a lead screw which generates the upwards and downwards force required to

remove and replace the modules. The current implementation can travel 2cm and lift a 1.5kg payload.

The only forces attaching the modules to the base is the friction force of the 17 connector pins and the two guide pins. This means that this friction must be overcome when lifting the module out of its socket. Since this friction force is greater than the weight of the entire robot, the connectors hold well, but an extra piece had to be added to stop the forklift from lifting the entire robot. This piece serves the dual purpose of holding the second robot down in addition to serving as a mechanical guide when docking with the second robot. The other option considered was to have a custom designed connector that would be zero force when the forklift pins were inserted and a lock that would be enabled when the pins were removed. Had this been necessary, we would have designed the required connectors, but fortunately, the commercial connectors functioned well. In a more realistic scenario however, a superior connector design (both electrical and mechanical) would be advised, but the size of the robot limited selection essentially to the above connectors or custom design.

The forklift receptacle and pins were designed to allow maximal orientation error when docking with the module to be removed. The more error in angular orientation that is allowed, the easier it is to automate the docking process. Three of the modules are approached from one side of the robot, and the remaining two must be approached from the other. The staggering of modules in this manner allows more room for angular positioning error.

Lastly, several sensors were added to ensure that the module is correctly inserted and removed from its socket. To this end, the forklift motor was selected to have encoders that are used to ensure that the module is lifted to the correct height. More importantly, misalignment errors could be detected when the motor did not travel far enough during module insertion. Several address connections were added underneath each module on the underside of the circuit board. When power is applied to one pad, some combination of four other pads will also be powered yielding a four bit module address. These connections are activated when the forklift pins are properly inserted, thus ensuring the correct module is removed.

There are several disadvantages of this type of design. Foremost is the fact that the forklift and chassis cannot be replaced. When the forklift fails, the robot is no longer capable of repairing teammates. Although alternating the docking direction for the modules simplifies the mechanical docking alignment problem, it will complicate most autonomous docking control strategies.

Although all robots are assumed to have a forklift in this particular team, a more general system might have fewer robots with repair capabilities. In general, this type of design lends itself to highly heterogeneous teams. The team can be configured with any combination of modules in the inventory. In this case, it would be advisable for the robots to have the ability to tow one another back to a repair station or closer to a working repair robot. We plan to have such towing capability in the future.

Additional advantages of this design are that the power module can be replaced and modules can be scavenged

off of failed chassis. Because the batteries are physically attached to the power module, this allows the robots to literally change their own batteries. In an autonomous task this is crucial, as the robots need not wait extended periods of time charging their batteries. In a space scenario there could be batteries being charged by solar power as the robot performed another task. If the chassis were to fail, the useful modules could be scavenged off of the robot in case of future failures in a different teammate.

### *C. Electrical Design*

Similar to the mechanical design, the electrical design should also be entirely modular with the components being interchangeable. We must first ensure that the robot has a sufficient operating time. Secondly, we must address the issue of electrical interchangeability. This problem has been addressed frequently with any number of commercial busses that have been developed, and thus we use a bus to connect the modules in the robot.

The robot draws approximately 280mA at 7.2 volts when both drive motors are running and all modules are in active mode. When modules are active but no motor is powered the robot draws 60mA. The forklift motor draws a peak current of 120mA, but is never operated at the same time as the drive motors. Given that the current battery capacity is 280mAH, this yields total operating times of between 1 and 3 hours depending on how much motion is required.

The first choice to be made is to determine whether a single master bus or a multi master bus is preferred. The advantages of a single master bus are similar to the advantages of centralized control in multi-robot systems [9]. The primary disadvantage however is that there is a single point of failure which can disable the whole robot. Fault diagnosis is also complicated because testing for faults will require a functioning main processing module.

A significant advantage of having a modular design with a multi-master bus was that should the main-processing unit on the robot fail, the robot could be controlled remotely from the communications module. In fact, due to the homogeneity of the robot team, a robot with an auxiliary main processing module could use its surplus processing power to control another robot with a disabled main processing module. Having the capability for the robot to be controlled directly from its onboard processor or remotely via the communications module is also a significant advantage with respect to fault verification and fault diagnosis. If the communications module is still operating when another module fails, remote fault verification and diagnosis can be performed.

The final design has a multi-master bus with 6 lines. For data transfer we used the inter-integrated circuit (I<sup>2</sup>C) bus that requires only two lines and is a multi-master bus. Battery voltage, 5 volts, ground, and one other line are brought down the bus. The last line is for future extensions or coordinating time critical operations between modules. Clearly any multi-master bus would have been acceptable here, I<sup>2</sup>C was chosen because even inexpensive microprocessors can handle acceptable data rates.

The primary disadvantage to using a multi-master bus is that each module must have some level of processing capability. In a mass-produced system, this could be achieved using FPGAs or ASICs. For our prototype system we chose to use Microchip PIC (Model 16C73A, 20MHz) processors for the modules, although any processor that could handle I<sup>2</sup>C data rates would have performed equally as well. This architecture is summarized in Fig 4 below. Though having processing on each module is expensive, it enables a form of internal fault diagnosis. Inter-module verification queries can be sent during an internal verification procedure. If a given module fails to respond or responds incorrectly then some types of faults can be easily discovered in this manner.

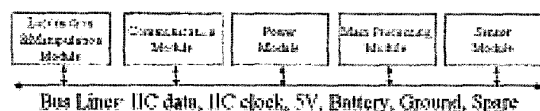


Fig. 4: Block diagram of the electrical architecture

There are also many disadvantages to such a bus that were not accounted for in this design. A power spike down any of these lines could damage multiple components. Thus surge protection should be provided on all lines. The power module assumes that there is only one power module, although clearly some applications might require extended run time, or the ability to carry spare power modules for a particularly power hungry teammate (possibly a teammate with a high energy consumption due to the number or type of functional modules being carried).

The primary design issue, however, is that the I<sup>2</sup>C bus is not designed for fault tolerant operation. Any module can halt all inter-module communication with a 'stuck at zero fault' on either the I<sup>2</sup>C data or the I<sup>2</sup>C clock lines. In our implementation of the bus, there is only one trace bringing each line to each module. Encasing the wires/traces in a protective substance and having redundancy in the electrical path would reduce the likelihood of having to scrap a chassis due a short or broken trace. Lastly, a superior design would have a redundant bus of some sort. A short range RF bus would be ideal due to its resistance to the problems of a physical bus connection (broken traces, faulty connectors, etc.).

### III. DESIRABLE CHARACTERISTICS IN AN AUTONOMOUS MODULE REPLACEMENT TASK

Most of this list was made before the robot was designed, but almost all were refined as the process of designing the robot progressed. We feel that should a similar system be required that these will be useful guidelines when designing the system. This is by no means a comprehensive list and should thus be viewed only as a guide.

#### 1) *All replaceable components should be mechanically and electrically interchangeable where possible*

This reduced the complexity in the average module replacement procedure. The same manipulator and docking algorithms can be used to replace any component. There should be no "exception" cases when replacing specific components.

#### 2) *All components should be replaceable*

This is not the case in our design. It would be especially useful to have all mechanical components (the forklift, wheels, idlers, chassis components) also be replaceable. Clearly, this is no simple task and to our knowledge there is no design that has yet achieved this.

#### 3) *The removal and replacement of all components should be simple and robust*

We have strived to accomplish this in the above design, but several features such as zero insertion force connectors and proper design of mechanical guides would improve the design.

#### 4) *The resolution at which components can be replaced should be as fine as possible*

For today's technology, the finest resolution would correspond to replacing chips, resistors, and manipulating standard machine screws and bolts. However, there is a tradeoff between fine resolution and being able to replace them in a simple and robust manner. Here we made the replacement as simple and robust as possible, but replacement only occurs at a coarse resolution.

#### 5) *All bus lines should be protected against both mechanical and electrical faults*

An accident, in this case, refers to: power spikes, the ability of one faulty module to stop all communication, and simple wear and tear on the physical lines. Physical redundancy in the bus lines would also be desirable. Of course this feature is geared mainly towards extended duration autonomous operation in harsh environments.

#### 6) *A secondary method of inter-module communication should be available*

If some portion of the main bus should fail, a redundant system would allow the system to assist in its own diagnosis and perhaps even "call for help" using the communications module.

#### 7) *A component should be replaceable regardless of physical orientation and robust to environmental conditions*

This is clearly not the case in the above design. If the robot has fallen on its side, repair is not possible. If the connector on the module is plugged, it cannot be replaced. In a more extreme environment (i.e. the moon or mars etc.) the design should have several features to deal with such situations. Again this is the 'ideal' goal. No system designer can truly achieve this goal, but one should consider this when designing repairable robots.

### IV. EXPERIMENTAL RESULTS

The main result of this work was the construction of the prototype. Using remote control, we were able to use the first robot to remove and replace a functional module on a mock robot (a portion of a second robot in a clamp). This task is presented in Figure 5.

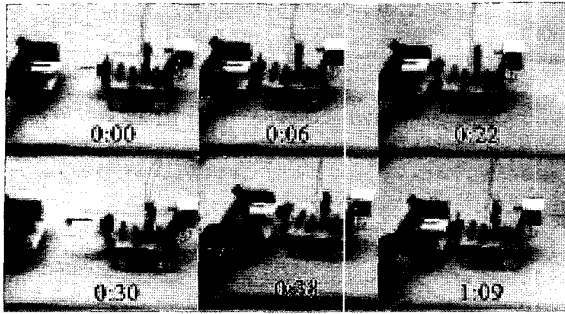


Figure 5: The removal and replacement of a module.

Unfortunately, if a picture can convey a thousand words then a video of the robot in operation can convey thousands of images. We have a video of the robot in operation available at our website <http://www.cs.cmu.edu/~aml/>. One removal and replacement of a module takes between 1 to 4 minutes (depending on the skill of the operator) and this is only a minor portion of the operating time of the robot.

## V. CONCLUSION

The physical module replacement is only part of the fault removal process. We still require a localization and docking system in order to position the first robot reliably with respect to the second robot. We have already developed a suitable localization system in [4], and we are planning to use visual servo control [21,22] for the high precision control required during docking. If prominent visual patterns are added to each robot then they may be used for the visual servoing as well as teammate identification.

Most prominently lacking are the verification and diagnosis steps of the fault removal process. We are currently implementing a cooperative verification and diagnosis procedure whereby the robots communicate with each other in an attempt to discover faults and diagnose them. We are also working towards automating the docking process.

There are a large number of open research topics in repairable systems. Clearly, there are other possible designs for module replacement, verification and diagnosis. Model-based verification has been performed in several areas and will be a useful tool in the verification and diagnosis steps. Another important issue is the development of useful metrics by which we can compare the efficiency and quality of different types of repair.

We are planning to use these particular robots in a two-dimensional mapping task. We can then perform experiments to determine the performance gains that can be achieved using a team of robots capable of cooperative repair. The ultimate goal is to look at designs where every component of the robot can be replaced and perform similar experiments with that platform.

The problem of designing a robot team with cooperative repair and reconfiguration capabilities is complex and not easily solved. We believe that such capabilities will be extremely useful in harsh environments where human assistance is either impossible, extremely costly, or in factory setting where downtime is crucial. It seems that the

necessary tools and resources exist to implement a dependable team of robots using the concept of cooperative repair. We hope that this paper will generate some interest in the field of autonomously repairable systems and speed the development of such systems.

## VI. REFERENCES

- [1] J.-C. Laprie et al., "Dependability - It's Attributes, Impairments and Means", Predictably Dependable Computing Systems, pp.3-24, ISBN: 3-540-59334-9, 1995
- [2] B. Randell, "Facing Up to Faults", Turing Lecture, 2000
- [3] C. Bererton, L.E. Navarro-Serment, R. Grabowski, C. J.J. Paredis & P. K. Khosla, "Millibots: Small Distributed Robots for Surveillance and Mapping", Government Microcircuit Applications Conference, 2000
- [4] L.E. Navarro-Serment, C.J.J. Paredis, P.K. Khosla, "A Beacon System for the Localization of Distributed Robotic Teams", in Proceedings of the International Conference on Field and Service Robotics, 1999
- [5] J. Gray, "Why Do Computers Stop and What Can Be Done About It?", Technical Report 85.7, PN87614, 1985
- [6] M. Yim, D. Duff, K. Roufas, "PolyBot: a Modular Reconfigurable Robot", Proceedings of the IEEE Conference on Robotics and Automation, 2000
- [7] S. Murata et al., "Self-Repairing Mechanical System", Proceedings of SPIE Conference on Sensor Fusion and Decentralized Control in Robotic Systems II, 1999
- [8] E. Yoshida et al., "Experiment of Self-repairing Modular Machine", Proceedings of SPIE Conference on Sensor Fusion and Decentralized Control in Robotic Systems II, 2000
- [9] P. Stone, M. Veloso, "Multiagent systems: A survey from a machine learning perspective", Autonomous Robots, Volume 8, number 3, July 2000
- [10] T. Balch, R. Arkin, "Motor schema-based formation control for multiagent robot teams", Proceedings of the First International Conference on Multi-Agent Systems, pp.17-24, 1995
- [11] K. Decker, "Distributed problem solving: A survey.", IEEE Transactions on Systems, Man, and Cybernetics, 17(5):729-740, 1987
- [12] K. Decker, "Environment Centered Analysis and Design of Coordination Mechanisms", PhD thesis, University of Massachusetts, 1995
- [13] L.E. Navarro-Serment, R. Grabowski, C.J.J. Paredis, P.K. Khosla, "Modularity in Small Distributed Robots", Proceedings of the SPIE conference on Sensor Fusion and Decentralized Control in Robotic Systems II, 1999.
- [14] C. J. J. Paredis, H. B. Brown, and P. K. Khosla, "A Rapidly Deployable Manipulator System", Robotics and Autonomous Systems, Vol. 21, pp. 289-304, 1997
- [15] C. J. J. Paredis, and P. K. Khosla, "Designing Fault Tolerant Manipulators: How Many Degrees-of-Freedom?", International Journal of Robotics Research, Vol. 15, No. 6, pp.611-628, 1996
- [16] A. Soto, M. Sapharishi, A. Trebi-Ollennu, J. Dolan, and P. Khosla, "Cyber-ATVs: Dynamic and Distributed Reconnaissance and Surveillance Using All-Terrain UGVs", Proceedings of the International Conference on Field and Service Robotics, pp. 329-334, 1999
- [17] S.-J. Tsai, E.D. Ferreira, C.J.J. Paredis, "Control of the Gyrover: A Single-Wheel Gyroscopically Stabilized Robot", Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'99), 1999
- [18] Ünsal, C., P. K. Khosla, "Mechatronic Design of a Modular Self-Reconfiguring Robotic System", Proceedings of the IEEE International Conference on Robotics and Automation, pp. 1742-1747, 2000
- [19] S. Thrun, D. Fox, and W. Burgard, "A Probabilistic Approach to Concurrent Mapping and Localization for Mobile Robots", Machine Learning 31, 29--53 and Autonomous Robots 5, 253--271
- [20] "NASA Technology plan", available at <http://technologyplan.nasa.gov/>, pp. 100-118, 2000
- [21] S. Hutchinson, G. Hager, P. "A Tutorial on Visual Servo Control", IEEE Transactions on Robotics and Automation, Vol. 12, No. 5, 1996
- [22] J. Shi, C. Tomasi, "Good Features to Track", IEEE Conference on Computer Vision and Pattern Recognition, 1994