Towards A Team of Robots with Repair Capabilities: A Visual Docking System

Curt Bererton
Robotics Institute, Carnegie Mellon University
Pittsburgh, Pennsylvania, USA
curt@cs.cmu.edu

Pradeep K. Khosla
Electrical and Computer Engineering, Carnegie Mellon University
Pittsburgh, Pennsylvania, USA
pkk@ece.cmu.edu

Abstract: 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 designed a robot that can replace the modules composing a similar robot. In this paper we highlight a visual docking system for the repairable robot design that allows the robots to autonomously replace their teammate's modules. The primary contribution of this work lies in the application of known techniques to the more constraining platforms of very small robots. This forces the use of very simple hardware and algorithms that perform robustly. The results obtained consist of initial configurations from which the robots could successfully complete the docking operation and the average time required to dock.

1. Introduction

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 (Figure). NASA's technology plan [12] 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 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.

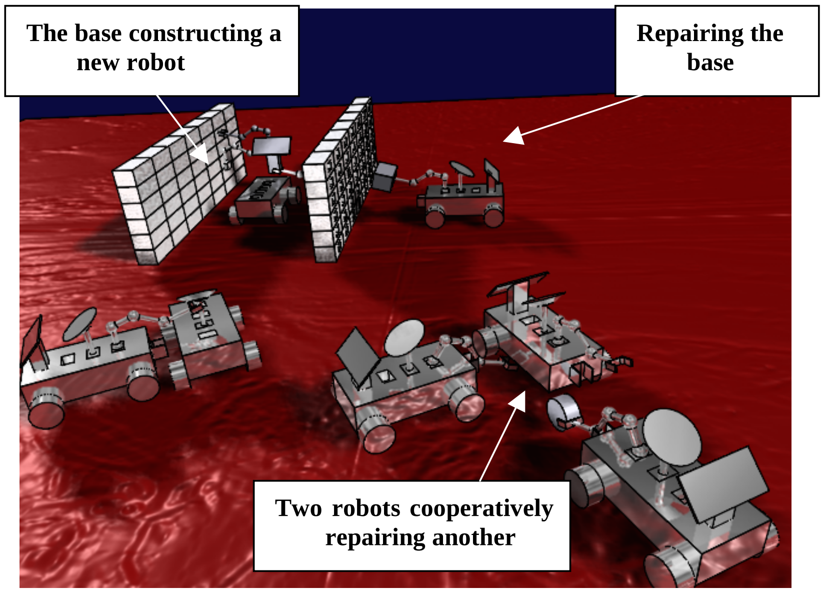
The base constructing a new robot
Repairing the base
Two robots cooperatively repairing another

Figure 1. Possible tasks for a self-sufficient team of 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. In previous work [16], we have designed a small mobile robot that has a standard skid-steered 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 in a wide variety of configurations.

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 due to the large number of actuators required.

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 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 planning complexity is 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] 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. In order to correct any fault in this case, we must replace the faulty module. To replace the faulty module, we must be able to dock with the failed robot and replace the module. This paper concentrates on the docking system that was designed for this purpose.

There is a relatively small amount of literature that is directly applicable to docking mobile robots. Most of the literature about pose-estimation and navigation based on landmarks is highly relevant to docking [20][21]. The main difference between this type of work and the systems presented here has to do with complexity. The typical task in such work is to accurately determine the robot's cartesian location. This work could be used to perform docking tasks, but tends to be overly complex for a simple docking task and often does not yield the required accuracy. Other docking tasks specifically designed for mobile robots tend to be highly task-specific [22], or use complex sensors and algorithms unsuitable for use on small robots [23].

2. Task Description

In [4] we developed a localization system based on ultrasonic trilateration. The system could position a robot reliably within one or two centimeters. Thus we required a docking system to dock from a start position that was anywhere within that area. In the results section, we describe the initial positions from which the robot can begin the docking procedure and successfully complete the docking task. These initial positions consist of an X and Y offset from the goal position. We will henceforward refer to this as the initial configuration of the robot.

In this case, the task definition was very precise. This task was to dock a forklift robot (the repair robot) with a second (stationary) robot and remove a module. The sensor used to accomplish this was a black and white wireless camera. The robot was required to fully insert the forklift pins into the forklift receptacle on the other robot. Contact with the second robot caused a bump switch to be triggered, indicating that docking was complete. The two robots are shown in Figure . In this task, the visual target was identified and tracked autonomously in order to dock the robot.

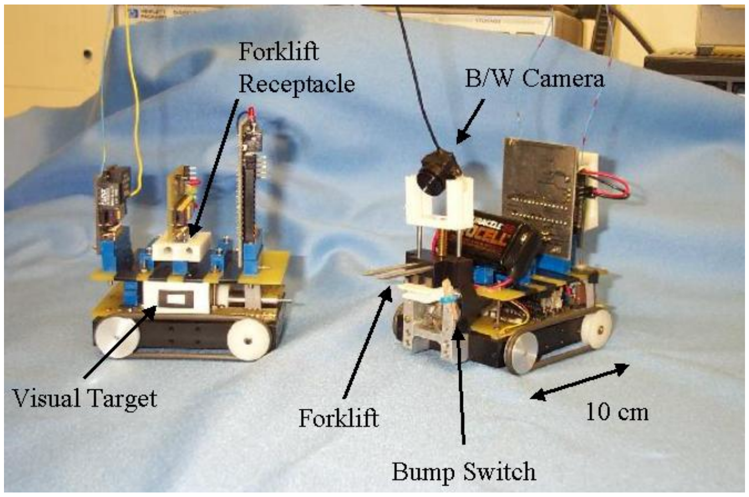

Figure 2. Repair robot and the robot to be repaired, including the visual target

The repair robot was placed in an arbitrary initial configuration and was required to complete the docking procedure, including the removal of the module, or it should indicate to the user that it could not complete the task. Success in this task occurred when the robot successfully removes the module on the second robot.

3. Hardware and Algorithms

3.1. Hardware Description

The basic platform for the repairable and repair capable is discussed in detail in [16]. To this platform we added a black-and-white wireless camera and a bump switch, as indicated in Figure . The camera transmitted a 320x240 pixel video stream to a Pentium II 400MHz PC, which then digitized the image using a Matrox Meteor II capture card. The PC performed the image processing and target tracking and then sent control commands to the robot to perform the docking task via an RF link. Most of the image processing was done using the Matrox imaging library.

One of the key features of the forklift and the forklift receptacle was that they were designed to have approximately a 30 degree allowable error angle between the line formed by the forklift pin and the axis of the hole in which the pin was to be inserted. This was partially achieved through rounding the pins of the forklift and shaping each of the two holes in the forklift receptacle as cones. The other main feature that allowed the 30 degree error is the fact that the robot is skid-steered. Thus, if the pins were partially inserted into the receptacle and the robot pushes straight ahead, then the wheels would slip on the ground and allow the robot to center the pins in the receptacle.

Although the image processing was done on the PC, our goal is to eventually perform all processing on the robot itself. To this end, the image processing, machine vision algorithms, and robot control strategies were deliberately kept as simple as possible.

3.2. Image processing and machine vision algorithms

The location of the visual target in the image could be identified given only a single image. The basic idea for this type of visual target came from [18]. In that example they used it to control a 5-degree of freedom manipulator. In order to determine the pixel coordinates of the target within the image, the following steps were performed:

- Capture a single grayscale image
- Perform histogram equalization on the image (equalization of the image intensity distribution)
- Binarize the image using a constant threshold which is determined empirically
- Find all 8-connected white regions (blobs) in the image and calculate the center of gravity of each region (i.e. the center of the region)
- Find all 8-connected black regions (blobs) in the image and calculate their centers of gravity
- Exclude all regions that are less than 30 pixels in size (30 pixels is the area the visual target occupies when it can barely be detected by this algorithm). This step eliminates noise in the image and reduces the processing required by the following steps
- Exclude all black regions that do not have exactly one white hole inside them. Since the target only has one white hole then all other regions are not of interest.
- For all remaining black regions, compare their centers of gravity to the centers of gravities of the white regions
- If a white center of gravity matches a black center of gravity to within 2 pixels, then that is our visual target. The coordinates of the black region's center of gravity give the center of the visual target. The value of 2 pixels was determined empirically.

The image processing routine returned the x and y location of the target in the image. We only had one 1cm tall target in the image and it was tracked extremely reliably up to a distance of 50cm from the target. At distances of greater than 50cm, the connected regions composing the target became too small to discriminate from the background. For comparison, a human observer was unable to observe the connected regions in the image (shown on the screen) at approximately 70cm. Obviously, if we had a larger target or a higher resolution camera, the distance at which the target could be tracked would have been greater.

3.3. Controlling the robot using the results from the machine vision system

The control of the robot followed a simple state machine:

- Acquire target state: This is the initial state in which the robot begins the docking procedure. Essentially, the robot turns counter-clockwise (clockwise would work equally as well) until the target is seen. This allows the robot to start from an arbitrary orientation. After the target is acquired, the robot proceeds to the approach state
- Approach state: First we determine the position of the visual target in the image. If the target is to the left of center, turn slightly to the left while moving forward. If it is to the right of center, turn slightly to the right while moving forward. Repeat until the target approaches the bottom of the image at which point we move to the centering state
- Centering state: In this state the robot turns left or right very slowly until the visual target is directly centered below the forklift pins and then proceeds to the insertion state
- Insertion state: The robot drives straight ahead until either the bump switch is triggered or a timeout occurs. If the bump switch is triggered, the docking is complete and the forklift can be raised to remove the module. This is considered a successful completion of the docking task. If a timeout occurs, the robot enters the retry state.
- Retry state: The robot drives directly backwards until the visual target is within view or a timeout occurs. If a timeout occurs during the retry state, then the docking procedure is considered a failure. If the visual target is seen again, then we proceed to the approach state. If this is the third retry then the docking procedure has failed.

There are several features to note about this state machine. Firstly, the robot was only allowed to retry 3 times before the docking procedure is considered to have failed. The justification here was that there is likely something blocking the forklift pins from entering the holes. Not mentioned in the above state machine is the fact that each state has an associated timeout. If the robot takes too long to perform any given state, the docking procedure is considered to have failed.

Perhaps the weakest feature of this control algorithm was that it relied on the bump switch for confirmation of a successful dock. Though we only checked the state of the bump switch in the Insertion state, it may have been possible for something to have accidentally triggered the switch. The hardware actually supported a superior method for determining docking success though it was unimplemented in these experiments. Just below the forklift pins on the repair robot, the white docking guide has connections for one power pin and four input pins. When properly positioned on the failed robot, these four input pins generate a four-bit address for each module, thus we would know if we had successfully docked with the failed robot as well as exactly which module on the robot we had docked with. This will be implemented in future work.

4. Docking System Results

Our primary concern was to determine the initial configurations from which the robot could successfully dock with and remove the module from the second robot The second robot is stationary at the goal position is at 0,0. Note in Figure that we do not show the initial orientation of the robot, as the acquire target state mentioned above will find the target if it is within the recognition range of the machine vision algorithm or it will timeout and indicate a failure.

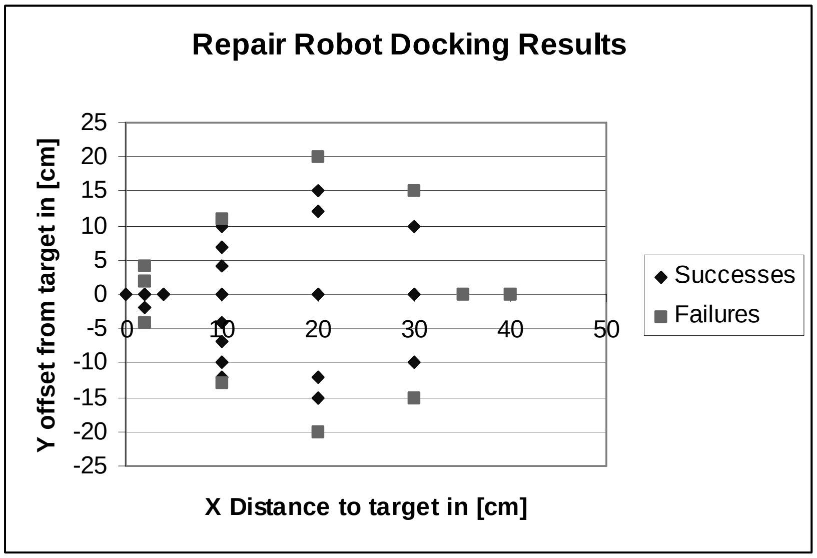

Figure 3. Docking Results

In Figure we performed 31 trials to determine the initial configurations from which the repair robot could successfully dock with the failed robot. Note that the area in which the robot succeeded in docking forms a cone-like shape. Thus, if we could position the robot anywhere within this cone we could successfully dock with the failed robot and remove the module. In the trials shown in Figure we measured the average time of a successful docking procedure to be 80 seconds, and the average time to notify the user of a failed docking procedure to be 75 seconds. From the furthest point that the robot could successfully dock without retries, the robot required 80 seconds if it performed no retries. A retry on average took 45 seconds, and retries were only required if the robot was on the edge of the area in which a docking procedure could be successfully performed. The system is quite robust; all the trials that were not on the border of the cone were successful. Considering that the robot operates for approximately one and a half to two hours on a single battery charge, the docking time was a minor portion of the operational time of the robot.

5. Discussion

5.1. Docking System Failure Modes

The failure modes of this system were due almost entirely to the visual system and the machine vision algorithms being used. Firstly, one should note that the camera was positioned high on the robot and looking towards the ground at an angle. This configuration allowed us to use the y position of the target in the image as a rough estimation of the distance to the target. The disadvantage was that we were unable to see the target at a distance of further than 40cm. The only reason for placing the camera there (as opposed to mounting the camera lower on the chassis) was that we wanted to have the camera be a replaceable module.

The binarization of the image was a large source of error. Though a human could clearly distinguish the target in the image at 55 cm, the binarization caused the outlines of the target to become indistinguishable from the background and the remaining steps in the algorithm to fail. There are several solutions to this. In [17], they used a self-similar pattern that could be easily recognized in the image. This approach could also be used here, but is significantly more processing intensive than our approach. Another approach would be to use a color camera and a target of a unique color as has been used in several other systems.

5.2. Strengths and Weaknesses of the Docking System

5.2.1. System advantages:

Visual target tracking explicitly determines whether or not it was possible to see the target from the current position, thus we always knew if it was indeed possible to successfully dock from the starting position. If we did not see the target then we could alert the user or a higher-level planner of the error. If the target is lost during the docking procedure, the robot can undo previous steps to find the target and reattempt the docking procedure. In many cases the robot being repaired will have lost power, if this is the case the visual system will still function correctly. In general one cannot make the assumption that the robot to be repaired can assist in the repair procedure in any way.

5.2.2. System disadvantages:

The hardware required for the docking system is expensive compared to the cost of the robot: Wireless B/W Camera \$250, Receiver \$100, Capture Card \$100-\$600. The robot itself costs approximately \$500, most of which is in the cost for the motors. Though in this particular case we performed the vision processing off-board, a more realistic scenario would be to have the vision processing done on-board. Vision is a computing-intensive application, and thus we would need a fair amount of processing power on the robot, which is difficult for a robot of this size.

5.3. Docking Systems for Repairable Robots

The above system was a second attempt at a docking system. The first system used infrared light emitting diodes (LEDs) to perform the docking procedure. This docking system was very inexpensive and computationally simple, but had several properties that were unacceptable in a docking system for cooperative repair. In order to find a damaged robot, the damaged robot had to turn on two LEDs in order for the repair robot to dock with it. Obviously, if the damaged robot does not have power, this is impossible. As a general rule, one cannot assume that a robot in need of repair can assist in the repair procedure in any way. Thus any docking system used in a repair procedure should be passive with respect to the damaged robot.

For our specific class of repairable robots, where repair consists of replacing failed modules, the docking system should allow the repair robot to dock with any replaceable module. Though not included in this system, the extension is simple. The addition of a visual bar code around the border of the visual target will allow us to have a visual target for every replaceable module. A more general approach might be to have one or more reference markers on the robot from which the relative position of all the replaceable modules could be determined.

Another crucial feature of docking systems for repair robots is that there must be a retry feature. There are an infinite number of possible failures that may occur during the docking phase. Many of these failures are only transient, and thus can be overcome if the docking procedure is repeated. Also, if the retries fail, it should be possible to re-plan at a higher level. In our case, this means using our ultrasonic localization system to move the robot to a configuration from which it can attempt the docking procedure again.

6. Conclusion

We have presented the concept of *cooperative repair*. Cooperative repair requires three distinct steps: *verification*, *diagnosis*, and *correction*. The docking system in conjunction with the mechanical design of the robot will allow us to complete the correction aspect of these three steps. Our current goal is to integrate the docking system with the localization system developed in [4]. We must then further develop the system to be able to distinguish and dock with multiple modules and various teammates.

The overall objective is to build a platform with which to develop the remaining two aspects required for cooperative repair, namely fault detection and fault diagnosis. Once the platform is complete, we will have a real platform to test the algorithms and procedures being developed for fault detection and diagnosis.

References

[1] J. -C. Laprie et al.. Dependability – It's Attributes, Impairments and Means. In 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 and 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 and 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 and K. Roufas. PolyBot: a Modular Reconfigurable Robot. In proceedings of the IEEE Conference on Robotics and Automation, 2000.

[7] S. Murata et al. Self-Repairing Mechanical System. In 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. In proceedings of SPIE Conference on Sensor Fusion and Decentralized Control in Robotic Systems II, 2000.

[9] P. Stone and M. Veloso. Multiagent systems: A survey from a machine learning perspective. In Autonomous Robots, Volume 8, number 3, July 2000.

[10] T. Balch and R. Arkin. Motor schema-based formation control for multiagent robot teams. In proceedings of the First International Conference on Multi-Agent Systems, pp.17-24, 1995.

[11] 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.

[12] NASA Technology plan. Available at http://technologyplan.nasa.gov/, pp. 100-118, 2000.

[13] S. Hutchinson and G. Hager. A Tutorial on Visual Servo Control. In IEEE Transactions on Robotics and Automation, Vol. 12, No. 5, 1996.

[14] J. Shi and C. Tomasi. Good Features to Track. In IEEE Conference on Computer Vision and Pattern Recognition, 1994.

[15] L. Navarro, R. Grabowski, C. Paredis, and P. Khosla, 1999. Modularity in Small Distributed Robots. In proceedings of the SPIE conference on Sensor Fusion and Decentralized Control in robotic Systems II.

[16] C. Bererton and P. Khosla. Towards A Team of Robots with Reconfiguration and Repair Capabilities. In proceedings International Conference on Robotics and Automation, 2000.

[17] A. Briggs, D. Scharstein and S. Abbott. Reliable Mobile Robot Navigation From Unreliable Visual Cues. Workshop on the Algorithmic Foundations of Robotics (WAFR 2000).

[18] D. Hershberger, Burridge, D. Kortenkamp and R. Simmons,. Distributed Visual Servoing with a Roving Eye. In proceedings, IROS 2000.

[19] C. Colombo, B. Allotta and P. Dario. Affine Visual Servoing for Robot Relative Positioning and Landmark-Based Docking, AdvRob(9), No. 4.

[20] S. Hutchinson, G. Hager and P. Corke. A Tutorial on Visual Servo Control, RA(12)

[21] W. Wilson, C. Hulls and G. Bell. Relative End-Effector Control Using Cartesian Position Based Visual Servoing, RA(12), No. 5 1996.

[22] S. Mascaro and H. Asada. Docking control of holonomic omnidirectional vehicles with applications to a hybrid wheelchair/bed system. In proceedings IEEE International Conference on Robotics and Automation, Volume: 1, 1998.

[23] H. Roth and K. Schilling. Navigation and docking maneuvers of mobile robots in industrial environments. Industrial Electronics Society. IECON '98. In proceedings of the 24th Annual Conference of the IEEE, Volume: 4 , 1998.