The 2009 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 11-15, 2009 St. Louis, USA

Coordinating Recharging of Large Scale Robotic Teams

Andrew Drenner, Michael Janssen, and Nikolaos Papanikolopoulos
Department of Computer Science and Engineering
University of Minnesota
Minneapolis, MN 55455
Email: {drenner|mjanssen|npapas}@cs.umn.edu

Abstract— **Robotic teams are often proposed for solving a number of problems, ranging from exploring unknown environments to monitoring areas for security or environmental contamination. These teams are composed of individual robots which may lack the capabilities to complete a task on their own. One critical capability required by teams regardless of the mission is the ability to have sufficient battery life to remain active for the duration of the mission.**

We present an approach for maintaining battery life by developing a hierarchical team composed of deployable robots and docking stations. Unlike other approaches, the approach presented here focuses on docking stations supporting multiple deployed robots simultaneously. In order to do so the docking stations must continually optimize their locations with respect to the robots in need of service. Discussion of the optimization is presented, along with simulation in multiple environments to illustrate the scalability of the approach to large robotic teams. The on-going transition of this algorithm to actual hardware is also discussed.

I. Introduction

Robotic teams are often proposed as a means of reducing exposure of humans to hazardous situations. In addition, robotic teams are proposed as a way to increase the effectiveness over a single robot solution, reducing the time until a mission can be completed by distributing the work over many systems. As technology improves and costs are reduced, these teams are becoming a reality.

While they may be effective, distributed robotic teams are not without limitations. In order to accomplish their missions, the members of the team should be able to leverage the sensing, locomotive, communication, and computational capabilities of the other team members to overcome their individual limitations. However, one limitation that is hard to overcome is power. Distributed robotic teams often act with a fixed amount of power, limiting the effective operational lifetime of the team to the smallest battery.

In this paper, a strategy is proposed in which a number of robots can be coordinated using mobile docking stations. The docking stations are capable of maneuvering through the environment while transporting, deploying, recovering, and recharging the other members of the team. It is assumed that the charging stations are capable of transporting significant reserves of energy or are able to maneuver to locations where they themselves can be rapidly recharged in order to have the power to support the distributed team.

In order to support the team, a strategy is required to continually reposition the docking stations in a way that reduces the overall power consumption of the team when they are seeking power. This in turn enables the distributed robots to spend more time on their mission, whether that be mapping a large structure or searching for the source of a hazardous chemical leak.

In order for the coordination strategy to be effective, it must be scalable to support teams consisting of large numbers of robots while still remaining computationally feasible. In many cases, the strategy can be a convex optimization problem. However, in complex environments, convexity may be lost and the problem must be approached using hill climbing or other gradient based optimization techniques.

The remainder of this paper is divided as follows. Section II will discuss other relevant literature. The formulation for the coordination is presented and discussed in Section III. Simulations of the effectiveness of the approach in a variety of environments are presented in Section IV. Conclusions and areas of future exploration are presented in Section V.

II. Related Work

Conserving and reallocating power is hardly a unique concept in robotic teams. In the past this has been done in a number of ways. Conservation of energy will extend the lifetime of a robotic team. In [1], [2], power is saved through communication schemes in which robots only broadcast to nearby neighbors, thus reducing the required signal strength and effectively reducing power consumption across the team. However, as with all conservation approaches, this only postpones the inevitable loss of power.

In other approaches, individual robots must dock with power sources that are known in the environment [3], [4], [5]. These solutions are often limited in their effectiveness as the environment in which a robotic team is deployed may not be known *a priori*, preventing the installation of fixed charging stations. Additionally, even if the locations are known in advance, the dynamic nature of the mission (e.g. responding to a natural disaster) may render the pre-installed charging stations no longer functional, reachable, or optimally placed.

Marsupial robotics have been used in the past [6] where a larger robot shares power with a smaller deployable robot through the use of a tether. However, tethers can be a double-edged sword. While they reduce power consumption in terms of wireless communication and enable continuous power to the deployed system, they also increase the amount

978-1-4244-3804-4/09/$25.00 ©2009 IEEE

of energy used as the deployable robot must expend more energy towing the tether and the tether can get caught further inhibiting movement.

Other robotic systems are able to extract power from their environment such as the NASA Mars Rovers – Sojourner, Spirit, and Opportunity. While solar cells work in certain situations, they may not work in scenarios where robots are needed indoors, in caves, or where environmental contaminants would damage the solar panels.

An approach that is close to the one presented here is found in [7], where a simulated robot acts as a tanker. The tanker traverses an environment to identify and locate robots in need of refueling. However, this approach is limited in that it can only support a single robot at a time.

Another alternative is found in [8] where the robots in the team can physically exchange batteries. This provides an exceptionally high rate of energy transfer between team members. However, the precision alignment for the exchange can be problematic.

III. Team Coordination

This work attempts to identify a means by which a team of robots can be coordinated in order to extend the amount of time that the robots can be "on task" while minimizing the amount of time they spend seeking energy. This can be formally stated as follows: "Given a team of n deployable robots and d docking stations, how can the docking stations be coordinated in order to maximize the longevity of the deployed robots?"

In general, each of the n robots can act independently of one another. If each these robots can choose among n_{alt} alternatives, then in order for a docking station to predict the correct action, it must be able to evaluate n_{alt}^n simultaneous hypothesis about the potential configuration of the n robots. This problem is not computationally tractable as n increases. If the n robots were all stationary when performing their tasks, this could be reduced to the Traveling Salesman Problem (TSP) where the docking station must simply visit each deployable robot in the most efficient path possible. However, the TSP has been proven to be NP-hard, thus a heuristic must be developed.

The following sections walk through a heuristic approach which outlines the actions that a deployed robot may take and how a docking station (or team of docking stations) can be coordinated in a computationally efficient (albeit greedy) method to service them. The underlying model for this team formulation was first discussed in [9], while the work here will focus on more efficient approximations of that approach as well as demonstrations of the approach in more complicated environments.

A. Deployable Robot Modeling

Each of the n robots in this scenario are assumed to be homogeneous, although this is not a requirement. Each of these robots operates with respect to a finite state model governing how they interact with their environment and when they choose to seek additional power in order to continue their mission. Figure 1 shows the states of the model and their relationships to one another. The actual transitions are mission dependent and are based on the deployable robot's estimation of how far it can travel and its knowledge of the location of the docking stations.

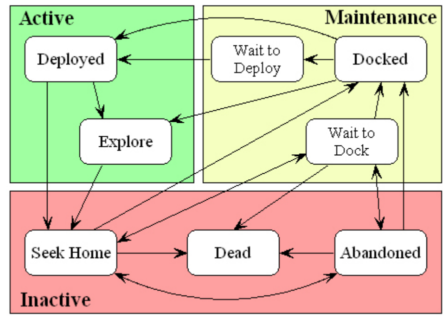


Fig. 1. The state transition model for the deployable robots.

In this model, there are the following groups of states:

- **Active** – This includes the states *Explore* and *Deploy* as the deployed robot is maneuvering through the environment in order to complete a task.
- **Inactive** – This includes the states *Seek Home*, *Abandoned*, and *Dead*. In these states the robot will do what it can to reduce power consumption until it can either make it to a docking station for recharge or remain stationary to conserve power until it is retrieved or dies.
- **Maintenance** – This includes the states *Wait to Dock*, *Docked*, and *Wait to Deploy*. In these states, the robot is either waiting on a full dock, being transported by a dock, or waiting for the dock to allow it to deploy. These states are primarily focused on the actual recharge process of the docking station.

The goal of this model is to maximize the amount of time that the deployed robots spend in the **Active** group while minimizing the time spent in the **Inactive** group.

B. Docking Station Modeling

The docking station must determine which deployed robots are in the **Inactive** state and attempt to assist them by maneuvering towards the robots most in need of its services. The individual robots needing assistance are assumed to be able to communicate an estimation of their position and an estimate of how far they can travel on their remaining energy reserves. The docking station must choose which of these robots are in most need by first clustering the robots using the ISODATA algorithm [10]. ISODATA was selected as it is similar in function to k-means clustering without a predefined number of clusters. Rather, there are an initial guess of k clusters and then a series of iterations occur where clusters are split and merged until the algorithm converges. Each cluster is represented by an artificial average robot which has the mean spatial location and available power of the cluster members. Equation (1) is used to determine the cost to the average robots representing the clusters, the most expensive cluster is chosen as the one needing support.

Equation (1) is then used to determine where the docking station will move with respect to individual members of that cluster.

In the case of multiple docking stations, it is assumed that the docking stations can communicate amongst themselves which robots they are servicing in order to prevent multiple docking stations from attempting to service the same robots concurrently.

Let S_I be the set of inactive robots that the docking station is attempting to service, $\overrightarrow{R_D}$ be a vector representing the location of a docking station at a given time and $\overrightarrow{R_i}$ be a vector representing the location of the i^{th} deployable robot in the set S_I. The objective is to determine the location of $\overrightarrow{R_D}$ at the next time step such that the overall cost of recovering all robots of interest is minimized. The cost function is then

$$f(\overrightarrow{R_D}) = \sum_{i=1}^{n} x e^{\alpha(x-1)} \tag{1}$$

where: $x = \frac{1}{v\varepsilon} dist(\overrightarrow{R_D}, \overrightarrow{R_i})$, $v = R_{i_V}$ or the velocity of the i^{th} robot, and $\varepsilon = R_{i_E}$ or time remaining based upon the remaining energy of the i^{th} robot.

This formulation differs from previous work [9] by introducing a new term α, a scalar used to tune outlier behavior. When $\alpha \to 0$, Equation (1) simply follows the mean of the distribution and spatially close groups of robots will be given the most priority. As $\alpha \to \infty$, the behavior of the cost function changes and more attention will be payed to robots that are outliers. This behavior may be more suitable to mission specific applications. For the examples shown here we have set $\alpha = 1$.

There are many methods for solving this minimization; under open or non-cluttered environments, the optimization of Equation (1) is convex. This can be shown as follows:

The cost function described in Equation (1) is desirable as it is fairly computationally inexpensive and it works in the overall greedy approach described here. Ensuring the global minimum of this cost function over all deployed robots (or over sets of robots that have been clustered) as shown in Equation (1) is reached can be difficult. The following is a look at ways in which this optimization can be solved in a variety of circumstances.

Let $\overrightarrow{R_D} \in \Re^2$.

Minimize

$$f(\overrightarrow{R_D}) = \sum_{i=1}^{n} \frac{1}{v\varepsilon} \left\| \overrightarrow{R_D} - \overrightarrow{R_i} \right\|_2 e^{\frac{1}{v\varepsilon}\left\| \overrightarrow{R_D} - \overrightarrow{R_i} \right\|_2 - 1}$$

such that

$$\|R_D - R_{D_{old}}\|_2 \leq V_D \delta t,$$

where V_D is the maximum velocity of the docking station and δt is the time interval.

Visually, this is shown in Fig. 2 where a single docking station must determine where it must go within a time interval (depicted by the dotted line) in order to best position itself in support of the team of robots deployed around it.

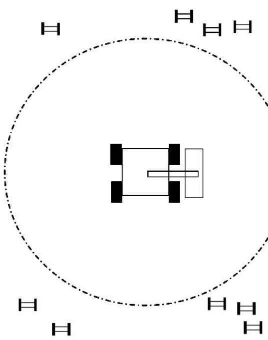


Fig. 2. Depiction of the basic problem in which a docking station must determine the best location for supporting the robots around it. The dotted black circle indicates areas that are reachable within a given time interval.

In this case it can be shown that $f(\overrightarrow{R_D})$ is convex through composition as defined in [11]. Specifically:

$$f(\overrightarrow{R_D}) = h(g_i(R_b))$$

is convex if the following conditions are true:

1) $h(y)$ is convex,
2) $h(y)$ is non-decreasing, and
3) $g_i(R_b)$ is convex.

In order to address the first two consider that

$$h(y) = ye^{y-1}.$$

To show $h(y)$ is convex, the second derivative of $h(y)$ must be positive for all non-negative values of y.

$$\frac{d^2}{dy^2}\left(ye^{y-1}\right) = 2e^{y-1} + ye^{y-1} \tag{2}$$

In this case, there exists no value of y such that it will result in a negative value. Therefore $h(y)$, is convex.

To show that $h(y)$ is non-decreasing, the first derivative of $h(y)$ must be positive for all non-negative values of y.

$$\frac{d}{dy}\left(ye^{y-1}\right) = e^{y-1} + ye^{y-1} \tag{3}$$

Again, there exists no value of y such that this will result in a negative value. Therefore, $h(y)$ is non-decreasing.

To address the third constraint, $g_i(R_b) = \frac{1}{v\varepsilon}\|R_D - R_i\|_2$ must be convex. Given that all norms are convex and that $\frac{1}{v\varepsilon}$ must be positive, this third constraint holds.

Thus, the optimization is convex with respect to a single deployable robot. The function in Equation (1) is also convex over the set of all robots as the sum of convex functions remains convex.

However, convexity only holds when certain conditions are met. For example, the presence of insurmountable obstacles destroys any hope of achieving convexity using the assumptions described so far. In this case, the cost of driving over the obstacle would be infinite and the resulting function would not be continuous and thus non-convex. Extending this, one could switch to an alternative distance metric such as shortest distance. In this case, as shown in Fig. 3, symmetric paths exist between a docking station and a robot, thus there must be multiple local minima, eliminating the possibility of convexity.

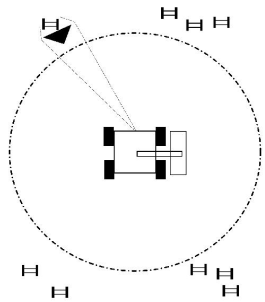
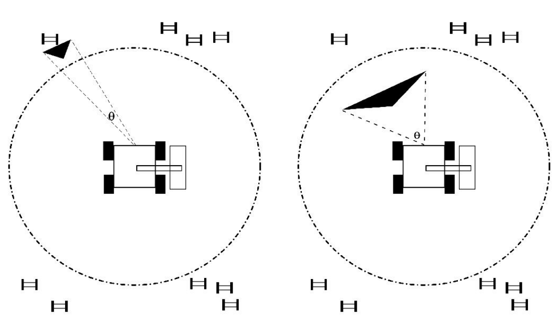


Fig. 3. Convexity will not hold in the presence of symmetric solutions. Here, two equally efficient paths exist for the robot in the upper left.

As a result, as expected from the overall formulation of this problem, the optimization of the function in the general case will be at best an exercise in gradient descent. This gradient descent may be plagued with local minima. The previously discussed clustering helps to partition the space and reduce the likelihood that a local minima will be reached. Used properly, clustering techniques could partition robots into sets that have few if any obstacles between the robots, and when the average robot is computed, would result in an average robot from which a revised cost function could be utilized. This cost function could use an alternate distance function which would remain convex with respect to that single robot.

There are relaxations that would still enable convex optimization to be useful. For instance, if the obstacle could be represented by a polygon, then there exists a left and right boundary of the polygon. From the docking station, the distance between these left and right boundaries can be represented by an angle θ as shown in Fig. 4(a). When θ is sufficiently small, it may be reasonable to assume that a Euclidean distance metric could be utilized. This would hold until the docking station had reached a critical point where $\theta \geq \theta_{max}$, at which point a linear approximation would no longer hold (Fig. 4(b)).



(a) Obstacle boundary creates $\theta < \theta_{max}$ and the linear approximation of distance holds.

(b) Obstacle boundary creates $\theta \geq \theta_{max}$ preventing the linear approximation.

Fig. 4. Relaxation of obstacles in order to approximate convexity.

Other issues preventing convexity could also be relaxed or modified. For instance, symmetric paths could be eliminated through a path-planner with a "commit and forget" strategy. However, the larger issue of where the docking station moves in the presence of obstacles remains. In the simplest approach, the region in which the optimization is being conducted could be scaled (as shown in Fig. 5). This would require more frequent optimizations as the region would not represent the maximum area that the docking station could truly travel and may not be overly optimal. Alternatively, the region could be divided into several convex regions and hypotheses could be created for each of these regions (Fig. 6). This would be required when the scalable area was reduced significantly. However, this option may become more computationally expensive in arbitrarily difficult areas than applying other optimization methods to the same area.

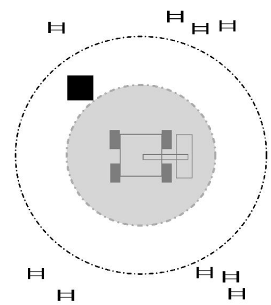
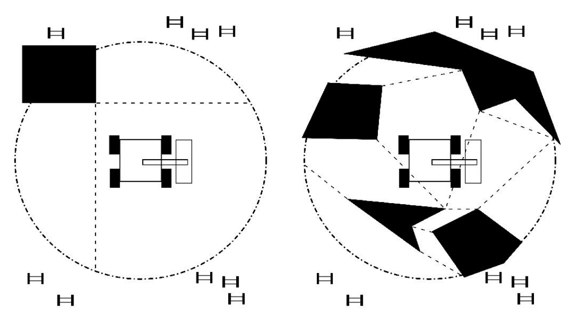


Fig. 5. In the presence of obstacles (black), the docking station's search could theoretically be scaled to avoid them (grey circle).



(a) Relatively easy region division.

(b) Relatively complex region division.

Fig. 6. In the presence of obstacles (black), the docking station's search regions could be divided into convex regions (formed by dashed lines and the reachable area) where multiple hypotheses could be generated.

IV. Simulation

In order to illustrate the coordination strategy outlined in Section III a series of simulation have been developed utilizing the Player/Stage system [12], [13]. In order to simulate the docking, transportation, and recharging of systems, an enhanced Marsupial Player/Stage [14] is also required. The docking station simulated in this environment is similar to the hardware described in [15]. Each of the docking stations are able to transport/recharge a total of six robots simultaneously and have sufficient power to keep the robots in the simulation alive.

In order to illustrate the scalability of the coordination presented in Section III-B, a series of 30 simulations were conducted. In these simulations 64 deployable robots were coordinated with one, two, or four mobile docking stations. The simulations were performed on a Pentium™ Core 2 at 2.4GHz with 4GB of memory, and ran at approximately real time. Each simulation required the use of almost all resources available, as up to 68 devices were simulated simultaneously, each with a separate subsumption-like architecture. Three environments were chosen for running this simulation, the *Open Field* (Fig. 7), *Cave* (Fig. 8), and *Hospital* (Fig. 9).

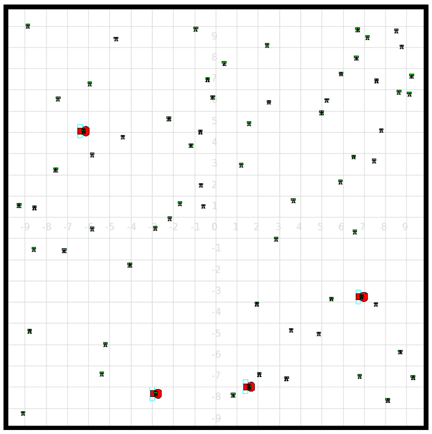


Fig. 7. The *Open Field* environment.

The *Open Field* environment offers only open areas which make it perfect for convex optimizations. The results of this simulation are shown in Table I. The *Cave* environment offers both open areas and areas where obstacles may be present. In this environment the proposed relaxations for convexity could be applied to control the docking stations' movements more efficiently. The results of this simulation are shown in Table II. The *Hospital* environment offers much more restricted access with more obstacles. Small, irregular rooms, increase the difficulty of using the convex approach. The results of this simulation are shown in Table III.

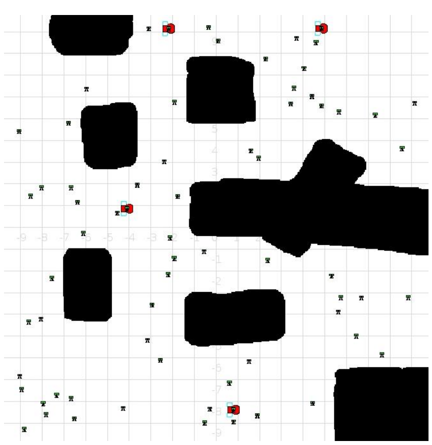


Fig. 8. The *Cave* environment.

As can be seen by the results in Tables I-III, the amount of time the robots are "Active" is greatly increased as the number of docking stations increases. The large number of robots that die in the single docking station case is partially attributable to the fact that all deployed robots are started simultaneously. Thus, they will nearly all attempt to dock for their first dock at the same time and the docking station can not accommodate such a large request.

A second series of simulations were conducted in both the *Open Field* and *Cave* environment to determine the effect of mobility on the docking stations. In this series of simulations, 30 runs were conducted each with four docking stations and

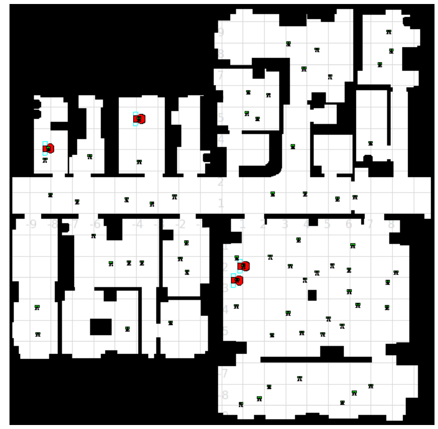


Fig. 9. The *Hospital* environment.

TABLE I
MEAN TIME IN STATE FOR DEPLOYABLE ROBOTS IN THE *Open Field*.

		Number of Docks		
		1	*2*	*4*
Time in State (%)	*Active*	31.09	48.86	74.70
	Inactive	54.69	30.66	8.19
	Maint.	14.22	20.48	17.11
Robots Dead	*Mean*	38.17	22.23	1.67
	Std	0.07	0.38	0.39

TABLE II
MEAN TIME IN STATE FOR DEPLOYABLE ROBOTS IN *Cave*.

		Number of Docks		
		1	*2*	*4*
Time in State (%)	*Active*	30.96	48.31	74.55
	Inactive	56.15	32.33	9.01
	Maint.	12.89	19.37	16.43
Robots Dead	*Mean*	38.53	22.07	1.70
	Std	0.10	0.42	0.59

TABLE III
MEAN TIME IN STATE FOR DEPLOYABLE ROBOTS IN THE *Hospital*.

		Number of Docks		
		1	*2*	*4*
Time in State (%)	*Active*	29.41	43.75	68.96
	Inactive	59.57	40.03	14.09
	Maint.	11.02	16.23	16.95
Robots Dead	*Mean*	41.73	29.10	5.53
	Std	0.30	0.85	0.63

64 deployed robots in each configuration of environment and docking stations (fixed or mobile). The results for this simulation are shown in Table IV.

TABLE IV
RESULTS OF SIMULATION RUNS WITH FIXED DOCKING STATIONS IN THE $Cave$ AND $OpenField$ ENVIRONMENTS.

		Open Field		*Cave*	
		Fixed	*Mobile*	*Fixed*	*Mobile*
Time in State (%)	*Active*	69.82	74.70	68.37	74.55
	Inactive	13.04	8.19	15.43	9.01
	Maint.	17.13	17.11	16.21	16.43
# Robots Dead	*Mean*	4.40	1.67	6.80	1.70
	Std	0.77	0.39	0.96	0.59
Robot Recharges	*Mean*	7.33	7.79	7.14	7.69
	Std	4.69	1.95	7.11	1.99

V. Future Work and Conclusions

As can be seen by the simulation presented in Section IV, the approach presented here is more effective than simply utilizing pre-positioned docking stations in maximizing the amount of time a deployable robot can remain active. Table IV shows that there was an increase of 7% time spent "Active" in the $OpenField$ and 9% in the $Cave$. This also means a corresponding decrease of nearly 37% in the time spent "Inactive" in the $OpenField$ and a decrease of nearly 42% in the $Cave$ environment. The number of robots that die when mobile docking stations are used is approximately 62% less in the $OpenField$ and 75% less in the $Cave$ environment. An ANOVA analysis shows that these results are statistically significant ($p < 0.01$). Also, as expected, the approach is shown to scale with the effectiveness with the number of docking stations available.

Given that the approach presented here has shown to be effective in a variety of simulations in environments of varying complexity, the future work remains focused on moving the algorithms to the actual hardware. In [15], the required mechanisms for the mobile docking station are presented. Figure 10 depicts the physical implementation of a docking station capable of performing the deployment, recovery, transportation, and recharge tasks required for this team coordination. Here, the docking station is mounted on a Pioneer 2 for mobility and capable of transporting and recharging six deployable robots simultaneously. The primary concerns with the physical implementation stem from problems relating to localization of the miniature platforms, ensuring that there is effective communication in real-world environments, and that the vision-based docking algorithm is effective in real-world environments.

In order to truly address the potential of this approach, additional docking stations will be constructed to carry out experimentation with teams of Explorer robots [16]. These docking stations will make use of custom power management systems in order to utilize multiple large lithium-polymer battery packs which will enable operational runtime measuring in days.

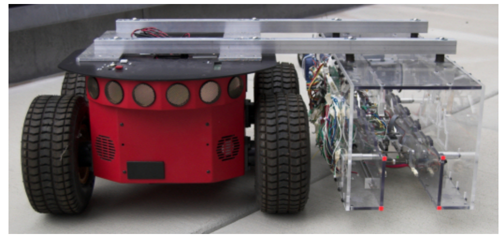


Fig. 10. The physical implementation of the docking station.

VI. Acknowledgments

This work has been supported in part by the National Science Foundation through grants #IIS-0219863, #CNS-0224363, #CNS-0324864, #CNS-0420836, #IIP-0443945, #IIP-0726109, and #CNS-0708344 and a National Science Foundation Graduate Research Fellowship.

References

[1] R. Nowak, U. Mitra, and R. Willett, "Estimating inhomogeneous fields using wireless sensor networks," *IEEE Journal on Selected Areas in Communications*, 2004.
[2] R. Willett, A. Martin, and R. Nowak, "Backcasting: Adaptive sampling for sensor networks," in *Proceedings of Information Processing in Sensor Networks*, 2004, pp. 124–133.
[3] W. G. Walter, *The Living Brain*. New York: W.W. Norton, 1963.
[4] Y. Hada and S. Yuta, "A first experiment of long term activity of autonomous mobile robot - result of repetitive base-docking over a week," in *Proceedings of the ISER 2000 Seventh International Symposium on Experimental Robotics*, Dec. 2000, pp. 235–244.
[5] M. C. Silverman, D. Nies, B. Jung, and G. S. Sukatme, "Staying alive: A docking station for autonomous robot recharging," in *Proceedings of the 2002 IEEE International Conference on Robotics and Automation*, Washington, DC, May 2002, pp. 1050–1055.
[6] R. R. Murphy, "Marsupial and shape-shifting robots for urban search and rescue," *IEEE Intelligent Systems*, vol. 15, no. 2, pp. 14–19, Mar. 2000.
[7] P. Zebrowski and R. Vaughan, "Recharging robot teams: A tanker approach," in *Proceedings of the International Conference on Advanced Robotics (ICAR)*, Seattle, Washington, July 2005.
[8] H. S. T. Ngo, H. Raposo, "Potentially distributable energy: Towards energy autonomy in large population of mobile robots," in *Proceedings of the 2007 IEEE International Symposium on Computational Intelligence in Robotics and Automation*, June 2007, pp. 206–211.
[9] A. Drenner and N. Papanikolopoulos, "Docking station relocation for maximizing longevity of distributed robotic teams," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Orlando, FL, USA, May 2006, pp. 2436–2441.
[10] C. W. Therrien, *Decision Estimation and Classification*. New York: Wiley, 1989.
[11] S. Boyd and L. Vandenberghe, *Convex Optimization*, 1st ed. Cambridge University Press, 2004.
[12] B. Gerkey, R. T. Vaughan, and A. Howard, "The player/stage project: Tools for multi-robot and distributed sensor systems," in *Proceedings of the 11th International Conference on Advanced Robotics*, Coimbra, Portugal, June 2003, pp. 317–323.
[13] T. H. Collett, B. A. MacDonald, and B. P. Gerkey, "Player 2.0: Toward a practical robot programming framework," in *Proceedings of the Australasian Conference on Robotics and Automation*, Sydney, Australia, Dec. 2005.
[14] M. Janssen and N. Papanikolopoulos, "Enabling complex behavior by simulating marsupial actions," in *15th Annual Mediterranean Conference on Control and Automation*, Athens, Greece, June 2007.
[15] C. Carlson, A. Drenner, I. Burt, and N. Papanikolopoulos, "Modular mobile docking station design," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robot Systems*, 2006.
[16] M. Walter, M. Anderson, I. Burt, and N. Papanikolopoulos, "The design and evolution of the eROSI robot," *Proceedings of the 2007 IEEE International Conference on Robotics and Automation*, Apr. 2007.