Recharging Robot Teams: A Tanker Approach

Pawel Zebrowski, Richard T. Vaughan
School of Computing Science, Simon Fraser University, Burnaby, British Columbia
{pzebrows,vaughan}@sfu.ca

***Abstract*— We examine the use of a tanker robot to distribute energy in a system of autonomous robots. Consider a team of autonomous mobile worker robots performing some task, each with a finite but rechargeable energy supply such as a battery or fuel cell. To work, the robots must expend energy. To expend more energy than is contained in a single battery charge, the robots must recharge. We examine the use of a special-purpose, energy-transporting "tanker" robot that finds and recharges worker robots, extending their working life. We examine the requirements of such a robot and compare some candidate designs in simulation.**

I. Introduction

Energy maintenance is a key requirement in creating long-lived autonomous robots. An autonomous robot, no matter how sophisticated its Artificial Intelligence, will have its life-span and work-load limited by the available energy. This problem is in common with all living things, and is so fundamental that we believe it may place interesting constraints on the design of intelligent autonomous systems. The Autonomy Lab at Simon Fraser University was created to study these issues.

The usual approach to solving this problem is to outfit a robot with means to recharge itself, usually by visiting a charging device at some fixed location [1]–[3]. This approach places certain demands on the sensing and computation of the robot, for example finding a fixed known location requires localization, which is certainly possible but can be very computationally expensive to achieve in an indoor robot, and may require high-resolution sensors such as laser scanners [4]. Yet this technology is mature enough that at least one off-the-shelf recharging system is available for research robots [http://www.activrobots.com/ACCESSORIES/DXDock.html], as is a low-cost domestic floor-sweeping robot, the Roomba Discovery [http://www.irobot.com/consumer] with autonomous base-station recharging.

We propose an alternative approach: a robot 'tanker' system in which a special-purpose robot collects energy from a source and distributes it to one or more worker robots. This may have two advantages compared to conventional autonomous recharging: first in terms of cost and complexity of worker robots, and second in overall system efficiency.

One benefit of the tanker system is an economy of scale. It is usually cheaper per Joule to move energy around in large quantities, as the overhead costs are lower. This is seen in the domestic vehicle fuel supply, where cars and trucks are driven to a local gas station to refuel, but the gas is delivered from a distant repository using a large tanker truck. Gas may be delivered to the repository using an ocean tanker which is filled from an oil pipeline. In each step up the tree, the delivery mechanism is more expensive, but moves fuel much more efficiently.

Another benefit of the tanker system is purely functional. Consider a fighter aircraft. It must be small and light to be maneuverable, yet it needs extended range to perform its task. It cannot carry all the fuel it needs, so it is refueled in flight by a tanker aircraft.

Another possible benefit is the decoupling of the final user of the energy from the fixed source. The tanker robot could change the form of the energy, perhaps obtaining hydrogen and providing electric charge, and it can change the mechanical coupling properties. This may allow the worker robots to have a more simple mechanical design.

This paper describes some of the complexity issues involved with robot recharging, as well as the benefits of recharging using a tanker. A tanker controller is proposed and design details explained. A series of experiments are performed, in simulation, to evaluate the performance for this tanker. Design decisions are then compared and tradeoffs explored.

A. Recharging Complexity

The robot-recharging problem consists of several hurdles that need to be overcome. Any robot capable of recharging itself must have the ability to

1) Realize its need to recharge
2) Locate a charging station
3) Approach a charging station
4) Interface with a charging station
5) Avoid several emergent problems, such as interference around a charging station

Implementing methods to overcome these hurdles increases physical and algorithmic complexity. Any robot that is capable of performing this task will have special circuitry monitoring battery level, localization allowing for the location and movement to a charging station, adequate sensing to allow for approach and interfacing with a charging station, and algorithmic sophistication allowing for the avoidance of interference, such as [5].

This additional complexity can quickly exceed the scope of a system of robots. Consider a system of simple robots designed to wander and collect samples in their environment. Such a task requires minimal computation, mobility, and sensing power. In order to achieve recharging with such robots, complexity is likely to increase well beyond the initial scope and budget.

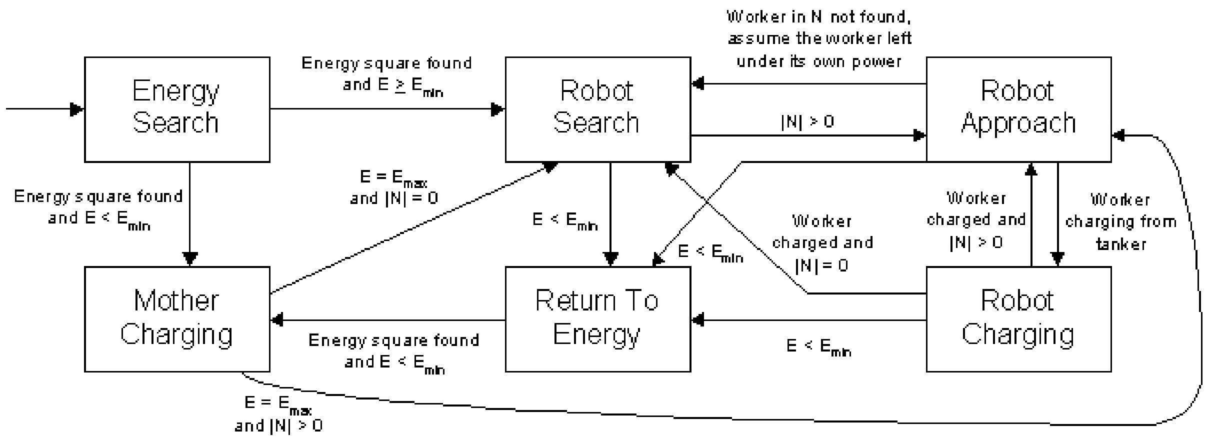

Fig. 1. Tanker controller

B. Tanker Approach

The tanker based approach is suggested as a useful alternative to individual recharging approaches for systems of autonomous robots. In the most simple form, an energy-transporting robot ("tanker") coexists in an environment consisting of worker robots ("workers") designed to perform some task (ex: forage) and is required to find and recharge worker robots in need of energy, while maintaining its own energy level. Such a tanker must be able to locate and approach worker robots, as well as return to a charging location and recharge itself.

The benefits of the tanker approach over a self-charging system can be seen in the division of labor. Worker robots are able to largely retain their task scope, with no need for extra sensing or algorithmic complexity. Tankers have only one task, to search and recharge workers. Further, this division of labor leads to a division in complexity. For worker robots, the coupling of their intended tasks and the recharging task is not required. As an example, a worker can stop working and wait for a tanker, instead of concerning itself with the complexities of returning to a charging station. A certain level of modularity is also introduced, consistent with the idea of interchangeable parts. This is likely to lead to cost saving and ease of replacement in case of failure.

II. Experiments

A. World

The experiments in this paper demonstrate tanker based recharging in simulation using Player/Stage [6]. The world is a two dimensional floorplan map of a hospital (a standard Stage environment) constructed from a blueprint of this hospital. Within the world is one tanker robot and a number of worker robots, each able to traverse the hospitals rooms and corridors.

Each robot in this world is equipped with an energy device which allows for the simulation of energy depletion and acquisition. The energy device consumes energy at a rate proportional to the number of devices on a robot and the mass of the robot (as defined in the simulation). For example, a Pioneer 3-DX based robot consumes less energy per time unit than a Nomadic Technology 200 based robot equipped with a laser range finder. It should be noted that a motionless robot continues to consume energy at a slow rate.

The world contains within it energy squares - locations which have the ability to give unlimited amounts of energy. Unlike a robot, which can acquire, store, and use a finite amount of energy, an energy square models a wall outlet or other public utility connected energy source. Each robot is equipped with a 'nose' that is used to transfer energy between charging location and robot, or robot and robot. This is an abstraction from the physical mechanisms required to achieve such a system. See future work for possible implementations.

B. Devices

The tanker is modeled after a Nomadic Technology 200. It is simulated by a sixteen-sided polygon with sixteen sonar range sensors. The tanker is also equipped with a laser range finder modeled after a SICKLMS laser range finder. These devices are used for tanker navigation throughout the hospital. The workers are modeled after a Pioneer 3-DX. They contain sonar range sensors, which are used for navigation.

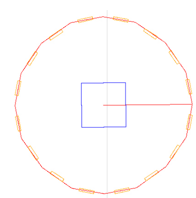

Fig. 2. Simulated Tanker

Each robot is equipped with a fiducial device, used primarily to distinguish a worker in need of energy from all other workers. Any worker robot with fiducial id $fid > 200$ is considered to be in need of energy. On real robots, this could be achieved using colour indicator lights and a blobfinder device. Fiducial id's are also used for statistical purposes, and to keep a list of energy awaiting worker robots when the tanker

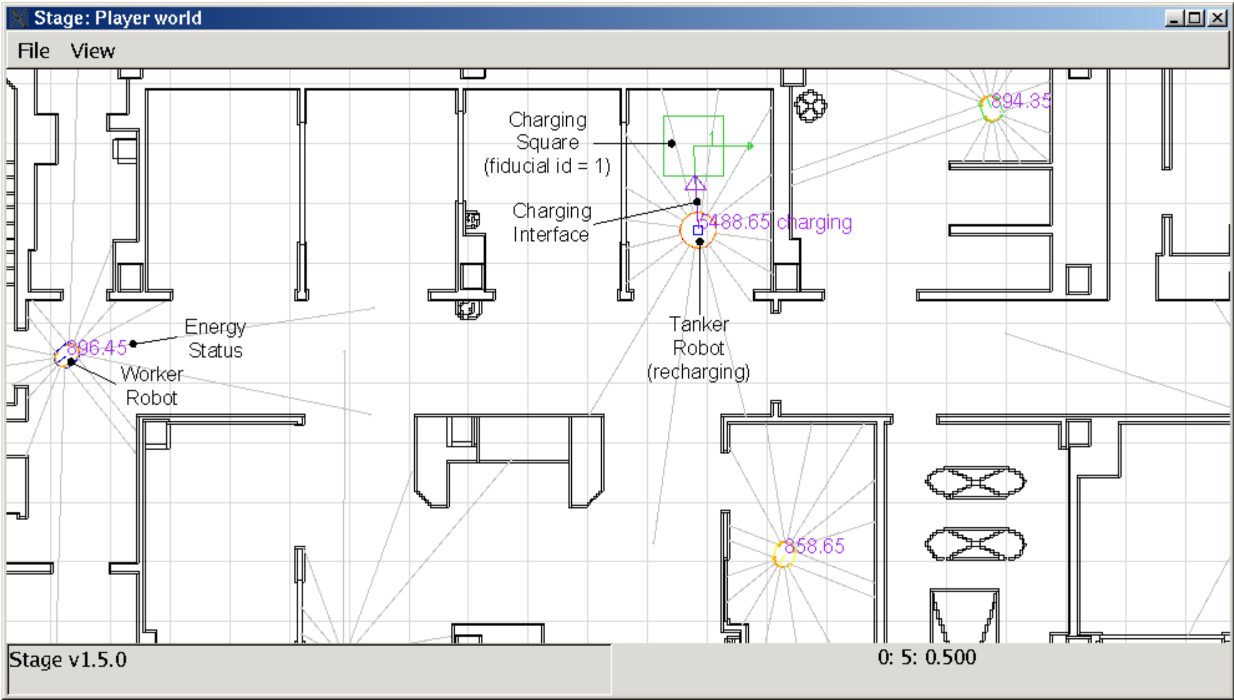

Fig. 3. Simulated World

cannot immediately recharge them. This, however, is not crucial in the successful implementation of this experiment.

C. Controller

The tanker controller is implemented based on the Subsumption Architecture [7]. This approach is found to be suitable as the robot needs to perform a series of simple tasks which have a static priority that is easily defined. The general groups of tasks are defined as follows, in order of priority:

1) Maintain mobility by avoiding obstacles and cyclic motion.
2) Find charging locations, and return to a charging location if tanker energy is below threshold.
3) Find worker robots in need of energy and recharge them.

The state transition diagram in Figure 1 precisely defines the states and transitions of a tanker. Notice that this state transition diagram is not dependent on any particular method, but is adaptable based on environment. For example, the "return to energy" state is implemented using a breadcrumbs trail-following method that combines some features of planning and mapping [8] in this paper, but can easily be replaced with a more complex localization scheme.

Execution begins in the energy search state, where the tanker's only goal is to find a charging location. Finding a charging location is key to sustaining energy beyond the initial amount provided, since the tanker depends on a reliable source of energy. No assumption is made that the energy locations are known a priori. Once a charging location is found, the tanker begins a cycle of searching and charging worker robots, and returning to some charging locations to recharge itself. This cycle continues indefinitely. It should be noted that the tanker always chooses to recharge itself before recharging workers, in order to sustain functionality.

D. Implementation

1) Data Structures: The tanker depends on several data structures that define the precise action of the robot. Let the worker vector N be a vector of $< fid, x, y, theta >$ (fiducial id and set of coordinates) which identifies the last seen location of all workers discovered as being in need of energy. Let the breadcrumb trail vector B be a vector of $< x, y, theta >$ (a set of coordinates) which identifies a series of points along the path to the last seen charging square. Let the energy threshold E_{min} be the amount of energy below which the tanker beings seeking a charging location to recharge, and E_{max} be the maximum attainable energy level.

2) Program Flow: In the energy search state, the tanker moves around its environment in an attempt to locate a source of energy. An energy square is identifiable by a range of specific fiducial ids. While in this state, the tanker adds the location of any workers in need of energy to the worker vector N, but does not proceed to help them at this time. Upon finding a charging station, the tanker begins dropping breadcrumbs during all subsequent actions in order to reliably return to this location when needed. When a charging location is once again identified, a new breadcrumb trail is started.

In the robot search state, the tanker looks for worker robots in need of recharging. Upon locating one, the tanker adds the robot's location information to the worker vector N. When the worker vector N is non-empty, the tanker proceeds directly to the location of some robot in N. During this worker approach state, the tanker continues noting other worker robots in need of energy. In the worker recharging state, the tanker transfers energy from itself to the worker robot.

When the tanker's energy level $E < E_{min}$, the tanker immediately begins following breadcrumbs in reverse order to get back to the last seen charging location. Once at the charging location, the tanker is able to recharge itself. When finished, it resumes looking for robots to add to the worker list N, or approaching some robot in N.

3) Navigation: The tanker achieves obstacle avoidance with help of the vector field histograms (VFH) method [9]. VFH allows the tanker to traverse its environment from its current pose p to some specified pose p' while avoiding static and dynamic obstacles.

Using VFH for obstacle avoidance introduces several cyclic motion problems. It is found that the tanker tends to traverse short paths repetitively and do so until some external factor interferes. To suppress this cyclic motion, a cyclic motion detection technique is used. This technique is achieved by placing a virtual tail on the tanker consisting of n breadcrumbs, where n is the length of the tail. Let p_i be the position of the tanker at time i. Let T_i be the tail at time i. Let j and k be times. $T_j =< p_0, p_1, ..., p_j >$ for $j < n$ and $T_k =< p_0, p_1, ..., p_n >$ for $k \geq n$. The tail can then be thought of as a series of the last n breadcrumbs taken at some regular interval. The algorithm compares the current pose p_c to every location in the tail T_c. Confidence C that the tanker is in a cycle is then defined as the number of p_i that are near p_c. In this implementation, p_i is considered near p_c if it is within one meter from p_c.

Cyclic motion confidence C remains low as long as the tanker traverses an area not seen within the last nt time units. However, if the tanker becomes trapped in a short cycle, the number of tail points p_i near p_c will increase. When confidence C reaches some threshold the tanker begins taking evasive action, in this case moving towards an open area until the confidence drops below the threshold. This occurs as soon as the tanker leaves the area in which the cycle occurred, thereby stopping the cyclic motion and returning control to normal navigation.

4) Searching: The tanker spends most of its time searching for worker robots and energy locations. Two searching techniques are explored: right wall following and random walking.

Right wall following is achieved with the help of VFH, by setting the destination in front and to the right of the tankers current location. See Figure 6 for a typical path taken during one simulated hour.

Random walking is defined as follows. Every $t = 10$ seconds, a new destination d is chosen in an area $x = -10...10$ and $y = -10...10$ where the tanker is defined to be at $(0, 0)$. A bias towards destinations in front of the robot is added to promote non-cyclic movements. Should the tanker come within 1 meter of destination d before time t elapses, a new destination d' is chosen immediately. See Figure 5 for a typical path taken during one simulated hour.

5) Return to Energy: Reliable return to a charging location is assured with the help of a breadcrumb trail algorithm. A breadcrumb $b_t =< x, y, theta >$ is a coordinate at time t. The breadcrumb trail T_n after last seeing the charging station n seconds ago is defined as $T_n =< b_c, b_0, b_1, ..., b_n >$ where b_c is the location of the last seen charging station. When the tanker reaches an energy level $E < E_{min}$, it stops performing its current action and proceeds to the last dropped breadcrumb b_n. Upon coming within 1 meter of b_n, T is truncated after $b_n - 1$, and the tanker proceeds to $b_n - 1$. This continues until reaching b_c and therefore the charging location. An optimization is added such that should the tanker be within 1 meter of any other breadcrumb b_j where $j < n$ and $j \leq k$ for all b_k within 1 meter of the tanker, the list of breadcrumbs is truncated after $b_j - 1$. This results in the tanker not repeating any cyclic behavior it performed during searching.

6) Worker Robot Recharging: The tanker identifies that a worker is charging from it by monitoring energy usage per second (watts). When watts used increases above the nominal rate, the tanker concludes that some robot is charging. The tanker then immediately stops to facilitate the charging. Every 30 seconds, the tanker initiates a 360-degree rotation during which all robots within 1 meter and facing the tanker are identified as charging. Their id's are noted and removed from the worker vector N. When the tanker's watts fall back to the nominal rate, the tanker resumes its previous task.

E. Procedure

Experiments are run in simulation using Player/Stage to demonstrate tanker based recharging. An experiment consists of 7 simulated one-hour trials. Each trial begins by placing the tanker and 9 worker robots in the simulated hospital world. A worker robot is to explore its environment, avoiding obstacles, until its energy level reaches some threshold, at which point it is to indicate the need for energy, stop moving, and wait for the tanker. Even while motionless, the worker continues to consume energy.

A record is kept of the tanker's state every simulated second. From this, metrics are extracted to be used in analysis. The following metrics are used to gauge experiment performance:

1) Number of worker robots in need of energy found
2) Number of worker robots recharged
3) Percentage of time spent in each state
4) Mean energy

These metrics are used to gauge the effectiveness of several performance enhancements as well as certain design tradeoffs. Two performance enhancements tested are breadcrumb trails when seeking energy, and the worker vector. The design tradeoffs tested are two different search strategies, and two processing techniques for the worker vector.

III. Results

A. Searching

Two techniques are explored for searching: random walking and right wall following. Right wall following provides for a detailed search with a slow coverage of the entire environment,

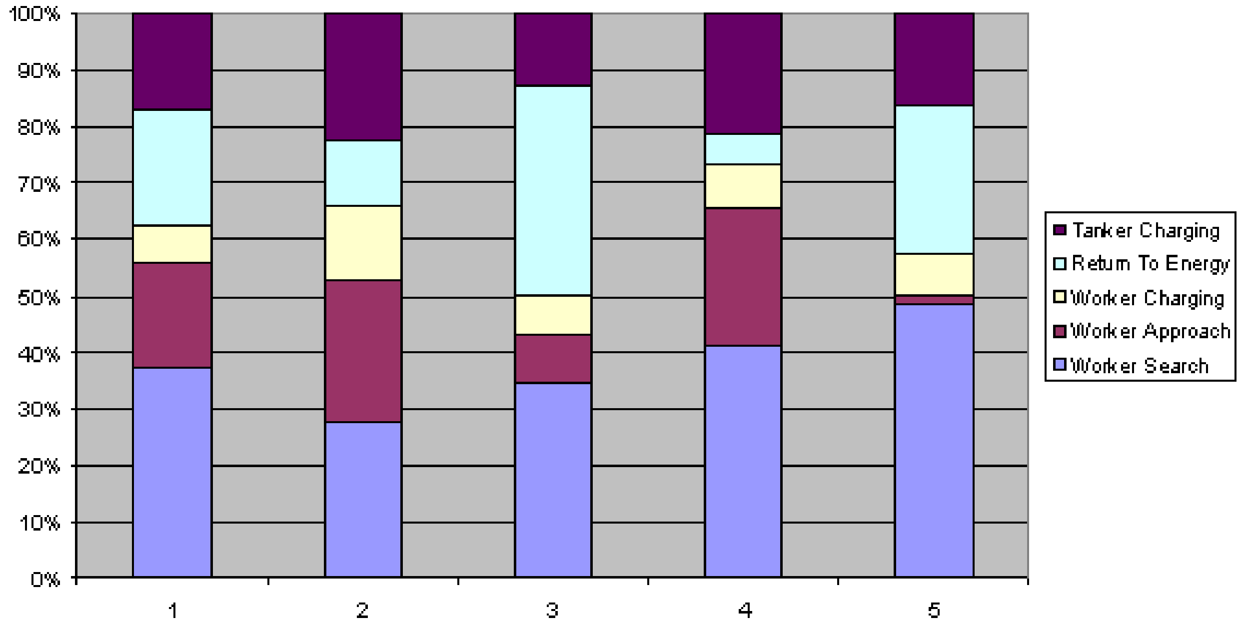

Fig. 4. Percent Time Spent In Each State: 1. Random walking, breadcrumbs, worker vector, FIFO. 2. Right wall following, breadcrumbs, worker vector, FIFO. 3. Random walking, breadcrumbs disabled, worker vector, FIFO. 4. Random walking, breadcrumbs, worker vector disabled, FIFO. 5. Random walking, breadcrumbs, worker vector, LIFO.

while the random algorithm covers the entire environment with little detailed search. As a result, robots further away are usually only discovered by the random algorithm, while robots hidden in rooms are only discovered by right wall following. Figure 5 and 6 demonstrate the coverage of each searching algorithm. In our experiments, right wall following performed significantly better than random walking, finding an average of 14 worker robots in need of charging, as compared to 4.5 using random walking. Further, robot searching only consumed 28% of the tankers time while using right wall following, compared to 37% with random walking.

TABLE I

SEARCH TECHNIQUE

	Average number of workers found
Random Walking	4.5
Right Wall Following	14

B. Worker Vector

It is found that the worker vector reduces the average time spent searching for depleted workers by 4%. It is interesting to note that a significant reduction of time spent in the energy seeking state is also noticeable. With the worker vector enabled, 21% of time is spent returning to a charging location, as compared to 5% with the vector disabled. Examining a typical path taken with the list disabled (Figure 7) shows that this is a result of the tanker staying closer to the charging location, and often retracing its path several times after recharging.

TABLE II

WORKER VECTOR

	Average number of workers recharged
Enabled	3.25
Disabled	3.66

C. Breadcrumb Trail

Experiments were performed with the breadcrumb trail enabled and disabled, storing only the pose of a charging location. Out of 7 trials with breadcrumb trail disabled, 2 terminated prematurely because the tanker could not return to a charging station before becoming depleted of energy. Further, the percent average energy level in trials with breadcrumb trails enabled was 72% compared to 55% with breadcrumb trails disabled. The percent of time spent returning to a charging location with breadcrumbs enabled was 21%, compared to 38% disabled. This leads to a conclusion that keeping a breadcrumb trail reduces the amount of time spent returning to charging locations, which allows for more time spent searching for depleted workers.

TABLE III

BREADCRUMB TRAIL

	Average percent energy level
Enabled	72%
Disabled	55%

D. Charging Order

Two methods of charging depleted worker robots are tested: first in, first charged (FIFO) and charge last seen robot first

(LIFO). First in, first charged guarantees that each robot will eventually be recharged, while charge last seen robot first makes the optimization of attending to closest robots first, resulting in better performance, but potential starvation. On average, first in first charged resulted in 19% of time being spent seeking depleted workers, while charge last seen robot first spent only 2%. First in first charged resulted in 37% of time spent searching for depleted workers, as compared to charge last seen robot first, with 49%. Starvation did not occur in any of the trials.

IV. Discussion

We have shown that the tanker method for recharging systems of autonomous mobile robots is a feasible solution to the recharging problem. Several techniques are explored and compared to yield a functioning tanker robot in simulation. This section discusses the significance of results obtained as well as possible improvements.

A. Search Techniques

The two search techniques tested produced greatly varying results. Right wall following can be seen to charge significantly more workers than random walking, and it does so with less time spent searching. However, the advantage of right wall following over random walking does not indicate a superior search technique for this problem. Both methods perform nearly identical tasks: navigate the environment due to some algorithm without storing historical data, planning, or otherwise intelligent reasoning. The reason that right wall following proves superior is because the majority of worker robots tend to remain in rooms, often far away from corridors. It is unclear whether this is an accurate estimate of what real worker robots would do. Consider a system of transport robots. It is foreseeable that such a system will favor spending time in corridors between rooms, making random walking a more effective search algorithm. Given this, a provably superior search algorithm may need to be based on a search strategy similar to frontier-based exploration [10] or some other more complex method.

B. Worker Vector

The worker vector does not result in an increase in the number of worker robots recharged. It is however witnessed that a slight reduction (4%) in the percent of time spent searching for workers does occur. It is conjectured that the lack of a perceivable performance benefit results from the following phenomenon.

It is observed that without the worker vector, the tanker stays significantly closer to the last seen charging station than with the vector enabled. This is once again attributed to the simplicity of the search technique. After tanker recharging is complete, the tanker commences its search strategy. Having no worker vector to remind it where it left off, it begins just as before, retracing large portions of the same path already traversed after the last recharge. In this sense, the worker vector acts more like a bookmark to remind the tanker where it left off before returning to recharge. This results in a more complete coverage of the entire environment. A more sophisticated search algorithm would likely account for this.

By remaining close to the last seen charging station, the tanker is able to traverse an area populated with robots at more frequent intervals, thereby having more opportunity to recharge workers in this area. This results in workers near the charging location receiving immediate assistance. With the worker vector enabled (and therefore broader coverage) a worker must wait longer before being recharged. It is therefore reasoned that the lack of perceivable performance benefit resulting from the worker vector is due to the tanker recharging the same group of robots repeatedly, thereby artificially increasing the charged worker count, instead of recharging all robots less frequently. This, of course, is not the desired result, as the goal is to sustain the energy level of the entire system by recharging all worker robots. Without the worker vector, a subset of workers is never discovered, and therefore never recharged.

C. Breadcrumb Trails

It can be seen from the experiments that a tanker is capable of reaching some distant target without the help of breadcrumb trails. This is witnessed when, with the worker vector enabled, the tanker is able to find the last worker seen, even if the location is a long distance away. The same cannot be expected in real world experiments with imperfect localization. The breadcrumb trail is used as a mechanism for reliably returning to an energy source, regardless of localization properties. It is shown in [8] that breadcrumb trails are reliable even with imperfect localization, which allows the tanker to make no assumptions about localization properties a priori. Why then are breadcrumbs not used in the worker vector as well? The worker vector does not contain breadcrumbs since it is used as a guideline only. The tanker does not need a precise path to each worker found.

D. Charging Order

Charging order is seen to have a significant influence on the percentage of time spent seeking workers in the worker vector. The reason for this is clear: having traversed a path encountering n worker, LIFO allows for the robots to be charged in reverse order. However, with FIFO, the tanker needs to traverse the path again, in reverse, then begin charging robots in the original order added. This extra traversal consumes time that can be used for other tasks. Further, should the tanker encounter any new workers while traversing this path, it will have to perform yet another cycle.

It should be noted that LIFO is in theory susceptible to starvation. The tanker may never get the chance to approach the first worker seen if others keep being recharged. While not witnessed in experimentation, it could conceivably be easy to set up a starvation scenario. For this reason, LIFO should not be considered a superior algorithm unless starvation is acceptable.

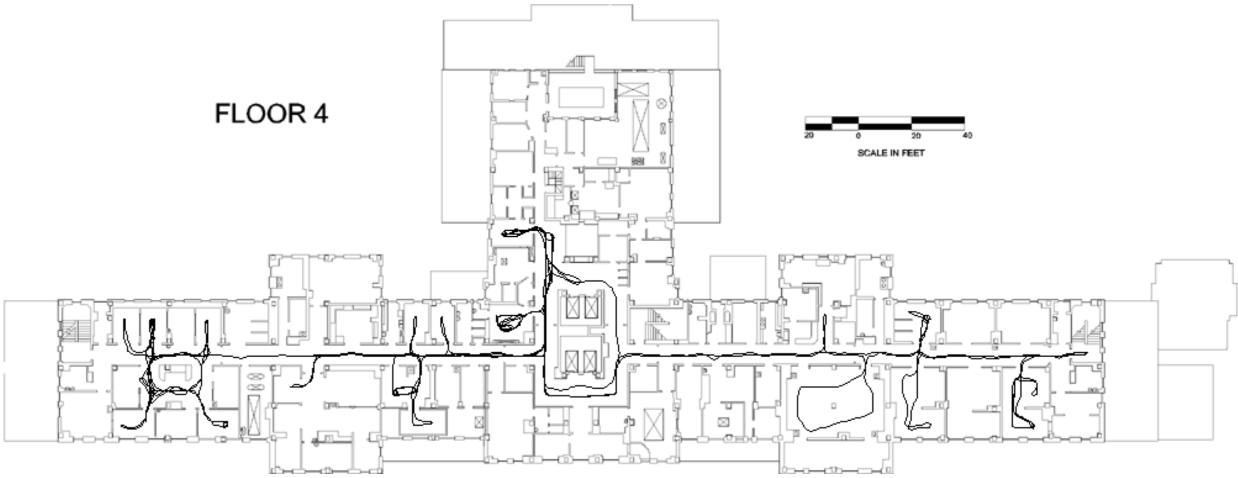

Fig. 5. Typical path taken with random walking

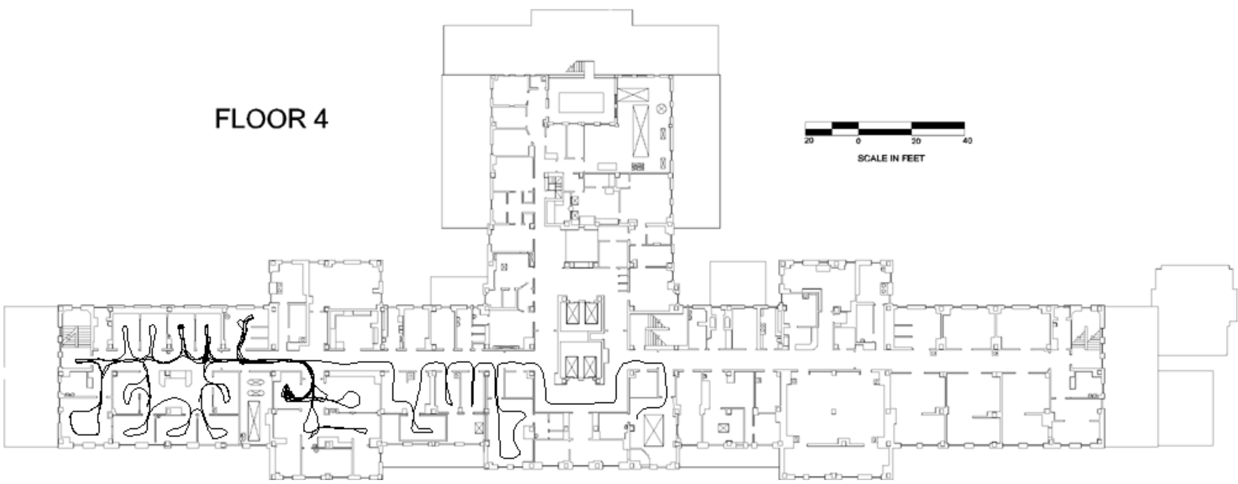

Fig. 6. Typical path taken with right wall following

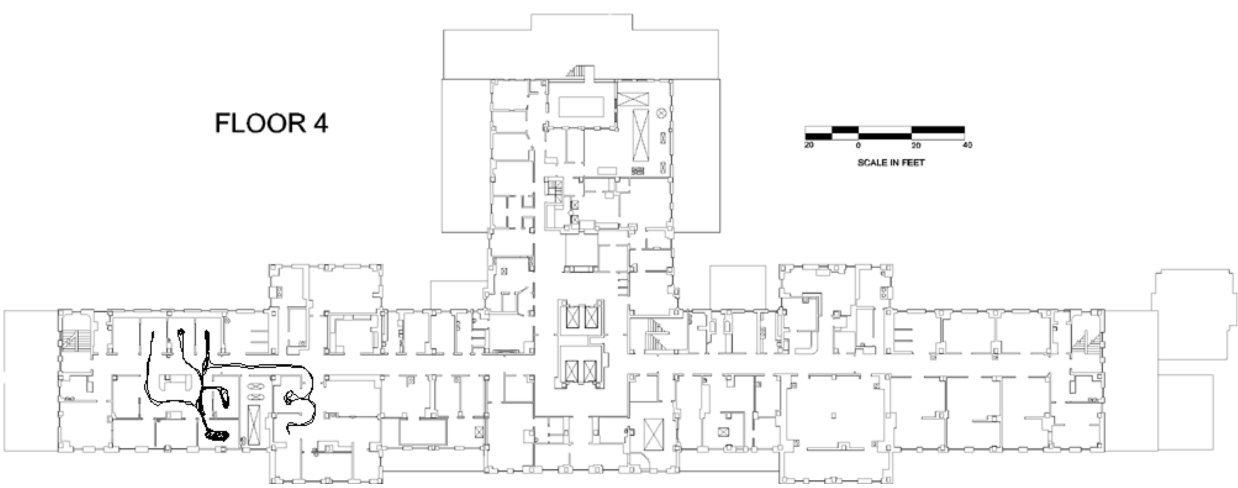

Fig. 7. Typical path taken with worker vector disabled

A more suitable algorithm would be one that allows for the path optimization of LIFO, but is not susceptible to starvation. For example, an algorithm that incorporates priorities increasing with time could be used.

V. Conclusion

The use of an energy distributing tanker robot in a system of autonomous worker robots as a solution to achieving long term multi-robot systems is demonstrated. We argue for decreased complexity, both algorithmic and physical, as well as increased modularity as a result of the division of labor resulting from this technique.

A tanker robot is constructed in simulation and demonstrated to be adequate at performing the prescribed task. We describe both a general tanker architecture suitable for adaptation given some target environment as well as design details used in achieving the tanker used in our experiments.

Several design techniques are implemented and evaluated for suitability and performance. Metrics are shown that demonstrate the utility of each technique, as well as a discussion about the intended and emergent behavior.

VI. Future Work

We are currently in the process of implementing these controllers on a real multi-robot system. The tanker is based on a pioneer 3-DX outfitted with grippers that allow it to grip both a charging post and worker robots. With the help of the grippers, the tanker is able to receive energy from a charging post or donate energy to a worker robot.

Work is currently planned for the exploration of the tanker approach for recharging in more detail. In particular, it is hoped that several optimality results can be obtained. For example, given a team of workers, at what location should worker and tanker meet to achieve optimal aggregate energy usage.

References

[1] W. G. Walter, *The Living Brain*. New York: W.W. Norton, 1963.

[2] M. Silverman, D. M. Nies, B. Jung, and G. S. Sukhatme, "Staying alive: A docking station for autonomous robot recharging," in *IEEE International Conference on Robotics and Automation*, Washington D.C., May 2002, pp. 1050–1055.

[3] A. Oh and K. Taylor, "Autonomous battery recharging for indoor mobile robots." [Online]. Available: citeseer.ist.psu.edu/oh00autonomous.html

[4] D. Fox, W. Burgard, and S. Thrun, "Markov localization for mobile robots in dynamic environments," *Journal of Artificial Intelligence Research*, vol. 11, pp. 391–427, 1999. [Online]. Available: citeseer.nj.nec.com/fox99markov.html

[5] R. Vaughan, K. Sty, G. Sukhatme, and M. Mataric, "Go ahead, make my day: Robot conflict resolution by aggressive competition," 2000. [Online]. Available: citeseer.ist.psu.edu/vaughan00go.html

[6] B. Gerkey, R. Vaughan, K. Stoy, A. Howard, G. Sukhatme, and M. Matarić, "Most valuable player: A robot device server for distributed control," 2001. [Online]. Available: citeseer.ist.psu.edu/gerkey01most.html

[7] R. A. Brooks, "A robust layered control system for a mobile robot," Massachusetts Institute of Technology, Tech. Rep., 1985.

[8] R. T. Vaughan, K. Stoy, G. S. Sukhatme, and M. J. Matarić, "Lost: Localization-space trails for robot teams," *IEEE Transactions on Robotics and Automation, Special Issue on Multi-Robot Systems*, vol. 18, no. 5, pp. 796–812, Oct 2002.

[9] J. Borenstein and Y. Koren, "The vector field histogram - fast obstacle avoidance for mobile robots," *IEEE Transactions on Robotics and Automation*, vol. 7, no. 3, pp. 278–288, 1991. [Online]. Available: citeseer.ist.psu.edu/borenstein91vector.html

[10] B. Yamauchi, "Frontier-based exploration using multiple robots," in *Proceedings of the second international conference on Autonomous agents*. ACM Press, 1998, pp. 47–53.