Adaptive Mobile Charging Stations for Multi-Robot Systems

Alex Couture-Beil   Richard T. Vaughan

Autonomy Lab, Simon Fraser University
Burnaby, British Columbia, Canada
{asc17,vaughan}@sfu.ca

Abstract— **We consider systems of mobile robots that execute a transportation task and periodically recharge from a docking station. The location of the docking station has a considerable effect on task performance. In nonstationary tasks the optimal dock location may vary over the length of the task. In multiple-robot systems, spatial interference between charging and working robots can make it difficult to find an optimal dock location, even in static tasks. We propose a new approach whereby the dock is itself an autonomous robot that attempts to incrementally improve its location. We show simulation results from a simple local controller that adapts to nonstationary tasks and spatial interference, and thus improves overall task performance compared to a static dock.**

I. INTRODUCTION

If a mobile robot is to expend more energy in work than it can store in an initial charge, it must have a means of obtaining more energy during runtime. The most common strategies for powering long-lived autonomous robots are (i) capture ambient energy directly from the environment, e.g. with solar panels, or (ii) transfer energy from a deliberately provided source, such as a charger connected to an electrical outlet. Mechanically coupling with a charging station provides a reasonable recharging rate and straightforward electromechanical design, and so is widely used.

Consider a prototypical autonomous mobile robot task, where robots repeatedly collect resources at one location and drop them off at another. To work for long periods, they must drive to a charging station (a *dock* hereafter) before their stored energy is exhausted, recharge, then return to work. Time spent driving to the dock, charging, and returning is pure overhead and should be minimized. The physical location of the dock can greatly effect the performance of such a system. Placing the dock too far from the worksite will increase travel time; time that could have been spent working.

We might choose therefore to place the dock at or near a worksite - either at a pick-up or drop-off point, or somewhere along the path between them. However, placing the dock *too* close to the working robot's normal trajectory will require the worker at best to detour around the dock, increasing the travel distance, and at worst will block the route, disabling the system completely. In scenarios with multiple workers an obstructing dock may trigger or exacerbate spatial interference between robots, further reducing performance. Even if the dock is designed to minimize interference (e.g. by being built into the floor or ceiling), robots *using* the charging station are a further obstacle. Considering that a typical work/charge time ratio for current lab robots is around 2/1 (e.g. Pioneer 3-DX), 1/1 (e.g. iRobot Roomba/Create) or worse, robots will spend a significant time as stationary obstacles.

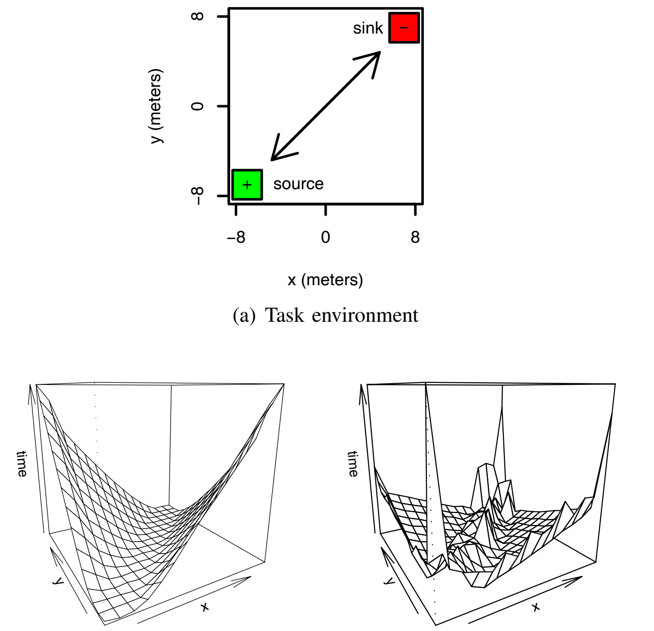
Further, if the dock is shared and multiple robots may queue up waiting to charge, the obstacle presented to working robots is even larger and changes dynamically.

Thus the optimum dock placement will be close to the work trajectory, but not so close as to cause significant interference. The fact that interference is often a complex dynamic feedback process could make identifying the best location in advance very challenging, as illustrated in the next section.

II. THE DIFFICULTY OF DOCK PLACEMENT

To examine the effects of dock location on system performance, we performed a simulation study of two robots transporting pucks between a source and sink in an otherwise empty environment, as depicted by Figure 1(a). We used the well-known *Stage* multi-robot simulator and its standard Pioneer-like mobile robot and SICK LMS-like laser range scanner models [1]. The robots run for 60 simulated minutes, shuttling between the source and sink which are located at (-7,-7) and (7,7) respectively. When a robot's energy store reaches a minimum threshold after approximately three round-trips (five minutes), it drives to the charging station, charges for one minute, then returns to work. At the end of the trial, the global average round-trip time is recorded as the system performance metric. The experiment is repeated with the dock placed at each of the 14×14 points on the integer grid spanning the workspace. This procedure is performed for two scenarios: (i) where the robots do not perceive each other or collide, so that spatial interference is absent; and (ii) where the robots do perceive each other and must drive to avoid collisions, so that spatial interference is possible. Of course, only (ii) is realizable.

The results are plotted in Figure 1(b) and (c). Figure 1(b) shows that the best performance (smallest time) is achieved when the dock is located anywhere along the shortest path connecting the endpoints. As the robots must detour further from their normal route, performance worsens (time increases). As predicted in the previous section, Figure 1(c) shows that the best performance observed is *not* along the robots' working path, but nearby. Also, the results suggest



(a) Task environment

(b) Performance vs. dock placement: without spatial interference

(c) Performance vs. dock placement: with spatial interference.

Fig. 1. Two robots transport pucks along shortest path between endpoints $(-7, -7)$ and $(7, 7)$ (a). The time to perform 100 round-trips, recharging occasionally, is measured for each possible charger location on the spanning integer grid. Results when spatial interference is disabled (b) show that placing the charger on the work route gives best performance. When interference is possible (c) performance is adversely affected and best charger placement is elsewhere.

that dock placement does not predict performance in a straightforward way.

A. Adaptive Mobile Charging Station concept

We have argued that dock placement can influence robot system performance, and that optimal dock placement is a non-trivial problem due to the dynamics of spatial interference. Since we desire practical methods for *good* dock placement, we propose a novel heuristic approach whereby the dock is itself an autonomous robot. Section IV presents a very simple dock position controller which incrementally improves its location in response to local information obtained at run-time. Simulation results in Section VI serve as proof-of-concept of the approach.

This concept is related to our earlier proposal of a 'tanker' robot that visited workers at their static worksites [2]; however in this work the aim of the dock is to find a good place to remain still. The long term motivation of both methods is to increase the overall energy efficiency of the system - though this is not addressed directly in this paper - and we suggest that these two alternative strategies may be suitable in different scenarios.

III. RELATED WORK

Charging stations are common among mobile robots [3] [4] [5]; however, they are constrained by the availability of a wall outlet. Typically, when a robot's stored energy falls below some threshold, the robot visits the charging station to recharge. The threshold can simply be fixed in advance, or it can be calculated at run time based on an estimate of the energy required to travel to the nearest charger. A better, optimal policy is described below, originally in [6].

A docking station capable of transporting, deploying, and coordinating a team of heterogeneous robots is presented in [7]. To facilitate long-term tasks, the docking station is capable of recharging the worker robots. The paper presents an energy efficient way of recharging robots by minimizing a cost function related to the Euclidean distance between the docking station and all worker robots. However, this docking station requires a shared coordinate system between robots, and tracks the positions of all workers. The system described below uses only local information and communication, and no global coordinate system.

In [8] robots are autonomously reconfigured with portable tools or 'effectors' with help from a standardized mounting system. The robot is capable of transporting these tools to different worksites. A charging station is also presented with the same mounting system; however, it is assumed to be stationary as it is requires a wall outlet.

A mobile 'tanker' robot is described in [2], which is used to actively locate and recharge worker robots. Further work in [9] provides a practical heuristic approach to the NP-hard problem of finding the most energy efficient path for a tanker robot to rendezvous with a team of heterogeneous worker robots.

Spatial interference is used as a performance benchmark in [10]. The measurement of interference is used as an evaluative tool while designing multi-agent controller code. Several puck foraging techniques are designed and benchmarked. Further work in [11] has studied the correlation between spatial density and interference. Territorial division is presented as a means to evenly distribute the spatial density of robots, thus minimizing interference. However, territorial division is not consistent with robots sharing a common charging station.

The issue of interference around a charging station is discussed in [12]. When a robot fails to dock with a charging station which is already in use, rather than wait, the robot enters into a random wander mode for a short period before attempting to charge again. This costly behavior reduces robot density commonly found at a charging station.

A mobile charging station that incrementally improves its location online does not appear to have been described before.

IV. TASK AND CONTROLLERS

As a proof of concept of the adaptive dock approach, we present a modification of the experiment above. A $20 * 20$m world contains a fixed obstacle field and a population of five worker robots, as shown in Figure 2. Workers must repeatedly travel between unique fixed source and sink locations. Workers are initially placed at fixed locations near the work site. A single adaptive dock is intially placed at a fixed

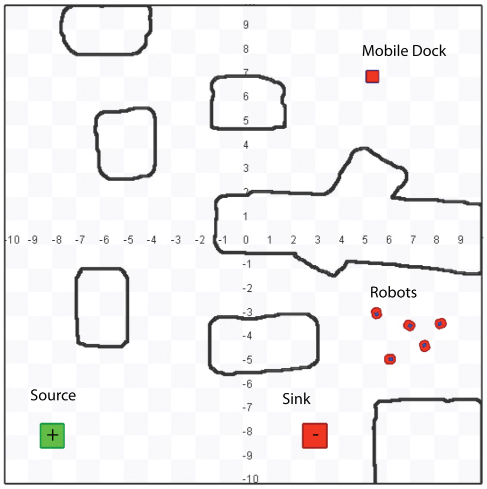


Fig. 2. Demonstration environment - initial conditions.

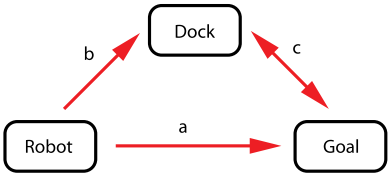


Fig. 3. The robot must maintain at least enough energy to travel b units. If the robot is not capable of travelling $a + c$ units, then it should travel directly to the dock before continuing to the goal.

location some distance from the work trajectory, chosen so that performance is initially much less than optimal. Workers and dock appear in each others' sensors and may collide, so are subject to mutual dynamic interference.

A. Worker Robot Controller

Worker robots are the Stage Pioneer2DX model, with differential-steer velocity control, approximately 50×50×40 cm in size, and equipped with the standard laser rangefinder model with 180 degree field of view. The robots are supplied with an *a priori* occupancy grid map of the fixed obstacle field and the locations of source and sink, and use Dijkstra's algorithm for global navigation along with a simple local obstacle avoidance adapted from the *fasr* demo provided with Stage. Any navigation strategy could be substituted, so details are omitted for brevity.

Robots have a simple recharging policy (Figure 3). Whenever a robot reaches an endpoint (source or sink) it calculates if sufficient stored energy remains to travel first the other endpoint and subsequently to the dock. If so, it travels to the other endpoint. If not, it travels directly to the dock, charges completely, then continues to the other endpoint. We showed previously that this is the optimal policy [6] for a single robot.

B. An Adaptive Dock

The adaptive dock is a mobile robot that can recharge worker robots when physically coupled. For now we assume the dock has sufficient energy stored to complete the experiment without recharging itself. The dock is equipped with a 360 degree range finder with range of 2.5 m with which it avoids obstacles while moving, and a sensor which can identify the state of nearby robots as either seeking charge or working. This could be implemented using various fiducial-based methods. For convenience, our dock has holonomic velocity control. Note that the adaptive dock concept does not require the dock itself to be a robot - it could be an inert dock which is carried into position by another robot.

Every time a worker robot couples with the dock, the worker evaluates the quality of the dock position as described below, then transmits the following two pieces of data to the dock (via a local data link, e.g. RS232, IR or Bluetooth connection).

1) A distance score D, which is used to evaluate the proximity of the dock based on the overhead distance traveled to reach it. In terms of the distances from Figure 3, the worker must travel at least a units to reach the goal. If the worker must first go to the dock before reaching the goal, the total distance traveled is $b + c$. Thus:

$$D = \frac{a}{b+c} \tag{1}$$

To minimized overhead, the dock must be placed along the shortest path. Since $a \leq b + c$, we have bounds $0 \leq D \leq 1$, where 0 is a poor position, and 1 indicates the dock is on the shortest path.

2) A navigability score N, which measures the fraction of time a robot spends navigating (i.e. making progress rather than avoiding obstacles):

$$N = \frac{t_{nav}}{t_{nav} + t_{avoid}} \tag{2}$$

where t_{nav} is the amount of time spent navigating, and t_{avoid} is the time spend avoiding obstacles, both measured since the last recharge. As for D, we have the bounds $0 \leq N \leq 1$, where 0 indicates that no progress has been made, and 1 indicates that ideal progress has been made (and therefore no spatial interference is present),

Meanwhile, the dock has recorded the angle α from which the docked worker approached, and stores a unit vector $\boldsymbol{v}$ with direction α. In our implementation α is the heading to the worker when it approached within 1m from the dock. This measurement is straightforward to achieve using various sensors.

The tuple $[D, N, \boldsymbol{v}]$, called a *vote* is stored for each docking operation. Once a large enough set S of votes has been accumulated, we compute the vector sum $\boldsymbol{V}$ of the set.

$$\boldsymbol{V} = \sum_{i=0}^{|S|} f(D_i, N_i)\boldsymbol{v_i} \tag{3}$$

Quality function $f(D_i, N_i)$ combines our two quality scores into a scalar value as described below. The magnitude of resultant $\boldsymbol{V}$ provides an indication of how well positioned the dock is. A small size indicates robots are approaching the dock equally from all sides, whereas a larger size indicates the majority of the robots are coming from the direction of $\boldsymbol{V}$. If the magnitude of $\boldsymbol{V}$ is above some threshold, the dock moves a short distance (1m in our implementation) in the direction of $\boldsymbol{V}$, while avoiding obstacles. Then the sampling process begins again.

Intuitively, if workers are approaching consistently from some direction, moving the dock in that direction should reduce total worker travel distance. However, if arriving workers are suffering from interference, moving the dock towards them may exacerbate the situation. Thus we need to choose $f(D_i, N_i)$ carefully. Our solution is developed below.

V. Demonstration and Investigation of Metrics

We first examine the behaviour of the system attempting to maximize the distance score alone, i.e. $f(D, N) = 1 - D$. The system is initialized as shown in Figure 2, and runs for 200 minutes. Figure 4 shows the evolution of the system: the lower graph shows the time of delivery of each puck (x axis) plotted against the time interval since the previous delivery (y axis). Triangles indicate *indirect* deliveries where the worker docked to recharge along the way, while circles indicate *direct* deliveries where the robot completed a delivery without recharging (the majority). The graph shows that the direct delivery interval is roughly constant until 120 minutes, while the indirect delivery interval decreases gradually until around 100 minutes. This is due to the dock moving closer to the worksite. Between 100 and 150 minutes, the direct interval increases, indicating that the dock is too close to the worksite and is causing interference that reduces performance.

The upper plot shows the quality measures D and N over the same period. Increases in D appear strongly correlated with reduced indirect delivery intervals, until D approaches 1.0, when direct delivery intervals increase and performance drops. The Navigability score N varies around 0.9 and does not appear well correlated with the delivery interval. These data suggest that D could be used as a control input to maximize performance, and that the ideal value for D in this example is around 0.8, achieved at 120 minutes.

While the data show the navigability score has limited dynamic range and does not predict performance, it can be useful as follows. Consider a worker in a restricted alcove. If the dock obstructs the worker's only exit, then the robot will be unable to navigate, producing a sharp dip in its N value. We can use any N below 0.5 to be a sign of a trapped robot, and convert its vote into a repulsion for the dock instead of the normal attraction. The modified scoring function is:

$$f(D_i, N_i) = \begin{cases} 1 - \mathbf{D}_i & \mathbf{N}_i > 0.5 \\ -1 & \mathbf{N}_i \leq 0.5 \end{cases} \qquad (4)$$

Equation 4 is effective at driving the dock to the worksite and preventing trapped robots. However, it still places the dock too close to the worksite.

One approach could be to drive D to some preset value other than 1.0, perhaps 0.8 as suggested above. However, system performance is quite sensitive to this parameter, and there is no principled way to set it.

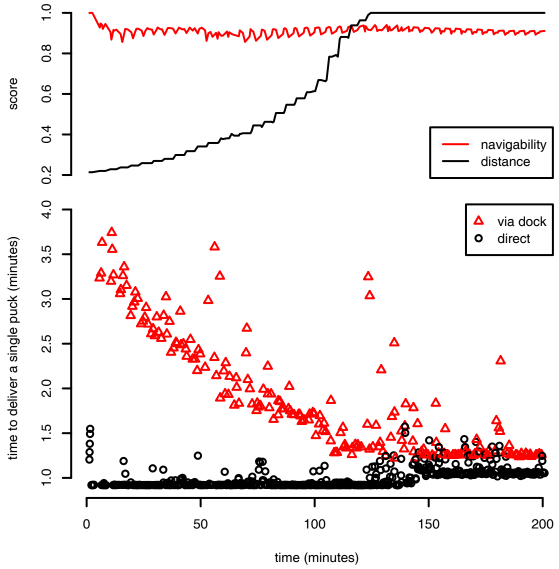


Fig. 4. Maximizing the distance score moves the dock too close to the worksite and reduces the puck delivery rate.

We wish to avoid a sudden increase in interference since the inherent negative feedback often makes them hard to recover from, with drastic loss of performance. Instead we force the dock to stop before it becomes an obstacle for working robots. Once the dock detects a robot that is working and not seeking charge, it will stop maximizing the distance score, using instead the Navigability-only quality function:

$$f^*(D_i, N_i) = \begin{cases} 0 & \mathbf{N}_i > 0.5 \\ -1 & \mathbf{N}_i \leq 0.5 \end{cases} \qquad (5)$$

A. Dynamic Tasks

In situations where the number and location of worksites may change over time, it does not make sense to permanently abandon D maximization once a working robot is detected. Instead, we would like to keep the dock stationary until the point where the task dynamics significantly change, and would benefit from repositioning the dock. To handle this, we monitor the local windowed mean and standard deviation of D, and in the event where a vote is received such that $distance\ score < \mu - 3\sigma$, then we reinitialize the dock controller, again maximizing D with 4 until another worker is detected.

An overview of the dock controller state diagram is illustrated in Figure 5.

B. Dock Queues

Due to the absence of explicit recharge scheduling, multiple robots occasionally arrive at the dock. In order to provide mutual exclusion over dock access and to minimize the obstacle presented by a group of robots, we implemented a queueing behaviour. Details are omitted for brevity, though this issue is of practical importance and will be examined elsewhere.

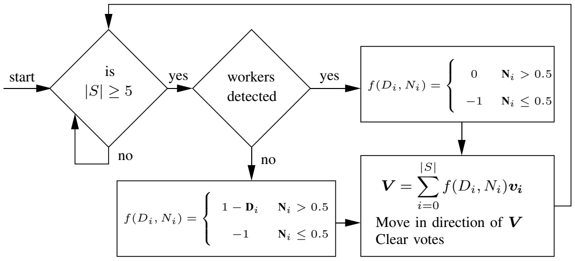
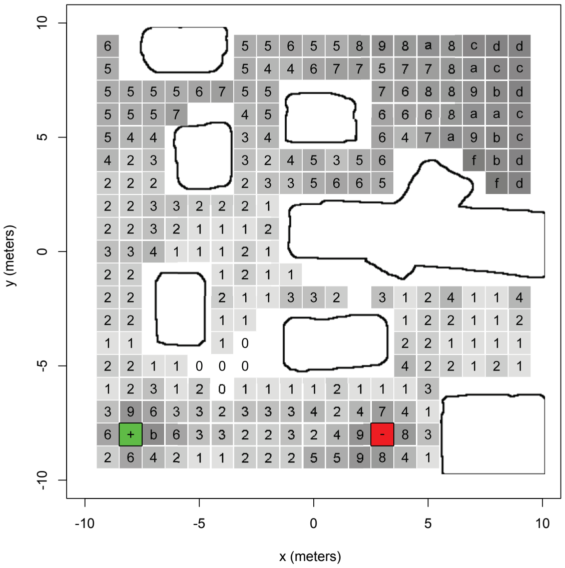

start
is $|S| \geq 5$
yes
no
workers detected
yes
no
$f(D_i, N_i) = \begin{cases} 0 & \mathbf{N}_i > 0.5 \\ -1 & \mathbf{N}_i \leq 0.5 \end{cases}$
$f(D_i, N_i) = \begin{cases} 1 - \mathbf{D}_i & \mathbf{N}_i > 0.5 \\ -1 & \mathbf{N}_i \leq 0.5 \end{cases}$
$\boldsymbol{V} = \sum_{i=0}^{|S|} f(D_i, N_i) \boldsymbol{v_i}$
Move in direction of $\boldsymbol{V}$
Clear votes

Fig. 5. Controller state diagram for the adaptive dock.



6						5	5	6	5	5	8	9	8	a	8	c	d	d
5						5	4	4	6	7	7	5	7	7	8	a	c	c
5	5	5	5	6	7	5	5					7	6	8	8	9	b	d
5	5	5	7			4	5					6	6	6	8	a	a	c
5	4	4				3	4					6	4	7	a	9	b	c
4	2	3				3	2	4	5	3	5	6				f	b	d
2	2	2				2	3	3	5	6	6	5					f	d
2	2	3	3	2	2	2	1											
2	2	3	2	1	1	1	2											
3	3	4	1	1	1	2	1											
2	2				1	2	1	1										
2	2				2	1	1	3	3	2		3	1	2	4	1	1	4
2	2				1	1							1	2	2	1	1	2
1	1				1	0							2	2	1	1	1	1
2	2	1	1	0	0	0							4	2	2	1	2	1
1	2	3	1	2	0	1	1	1	1	2	1	1	1	3				
3	9	6	3	3	2	3	3	3	4	2	4	7	4	1				
6	+	b	6	3	3	2	2	3	2	4	9	-	8	3				
2	6	4	2	1	1	2	2	2	5	5	9	8	4	1				

Fig. 6. Locations for a statically placed dock - as determined by exhaustive search. Values are hexadecimal where 0 indicates best performance (highest delivery rate) and f indicates worst performance.

VI. SIMULATION AND RESULTS

For comparison with our method, we need to find the performance of a well-placed conventional stationary dock. For each location on a 1m grid in navigable space, we placed a dock at that location and ran a complete simulation trial to obtain the average time required to deliver a puck. Figure 6 shows a map of performance by location, with the values normalized from 0 (best) to 15 (worst) and displayed as hexadecimal integers. The best dock locations are around $(-4, -5)$. Note these values are used for evaluation only and were not available to the dock controllers.

After identifying five good locations (indicated by a zero in Figure 6), we ran 600 minutes of simulations with a single static dock for each location, repeated for robot populations of 5, 10 and 15 robots. The average time it took a robot between puck deliveries is displayed in Table I. This is our baseline for “good” performance. Note the large standard deviation for 15 robots, suggesting that a large amount of interference is present. The workspace is very crowded with this large population.

We repeated the experiments with an adaptive dock, and recorded the performance in Table II, along with the time that the adaptive dock stabilized at its final location. The average puck delivery rate after stabilization is similar to our baseline performance with a well-placed static dock. However, in the 15-robot case we see a very large standard deviation.

#robots	avg	std dev
5	0.997	0.299
10	1.12	0.464
15	2.13	1.56

TABLE I

PUCK DELIVERY INTERVALS WITH A WELL-PLACED STATIONARY DOCK.

#robots	stable time	total avg	stable avg	stable std dev
5	109	1.04	0.983	0.207
10	114	1.20	1.13	0.491
15	74.5	2.19	2.03	4.31

TABLE II

PUCK DELIVERY INTERVALS WITH AN ADAPTIVE DOCK. STABLE TIME IS WHEN THE DOCK REACHED ITS FINAL POSITION. TOTAL AVG IS THE AVERAGE DELIVERY INTERVAL OVER THE WHOLE TRIAL. STABLE AVG IS THE AVERAGE DELIVERY INTERVAL AFTER STABILIZATION.

Data from the beginning of an adaptive dock trial with 5 robots is plotted in Figure 7. The indirect delivery interval decreases to around 1.4 minutes at 110 minutes and stays there. The direct delivery interval remains constant, indicating that the dock did not interfere with the work route. The upward curve in the plot of delivered pucks (top) indicates that the delivery rate increases up to about 90 minutes then stabilizes.

A. Dynamic Task demonstration

In a second demonstration, the location of the source and sink changes after 300 minutes, as indicated by “task 2” in Figure 8. The trajectory of the adaptive dock is shown. From its starting position at (6,6), the dock first stabilizes around (-4,-5) until the task locations switch. The dock then detects a significant drop in D following the change, and repositions itself for task 2. Figure 9 shows the change in puck delivery intervals between tasks 1 and 2, the dip in D and the improvement in indirect delivery intervals as the dock repositions itself.

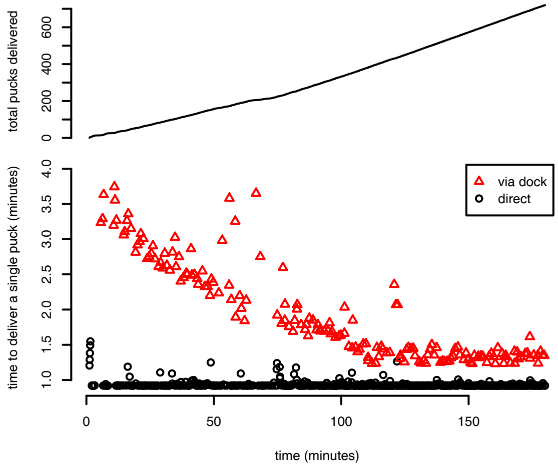


Fig. 7. Rate of puck delivery increases as the adaptive dock improves its location. The direct delivery interval does not change.

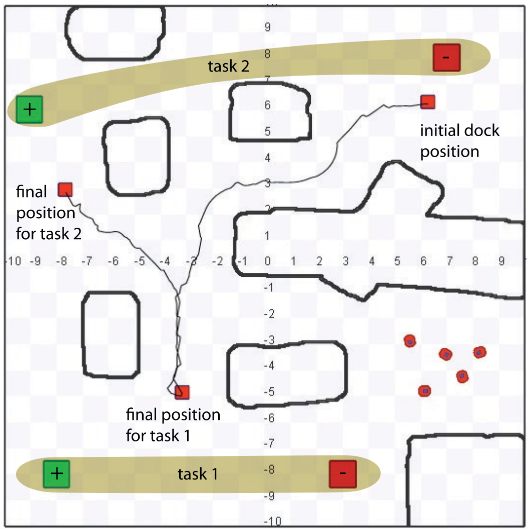


Fig. 8. An example of a path taken by an adaptive dock for two back to back tasks. The first task transported pucks between the source and sink located in the bottom of the figure, and the second task used the source and sink in the top half of the environment. Each task ran for 5 hours. Task 2 started immediately after task 1. The dock successfully detected change in its solution quality and repositioned itself for task 2.

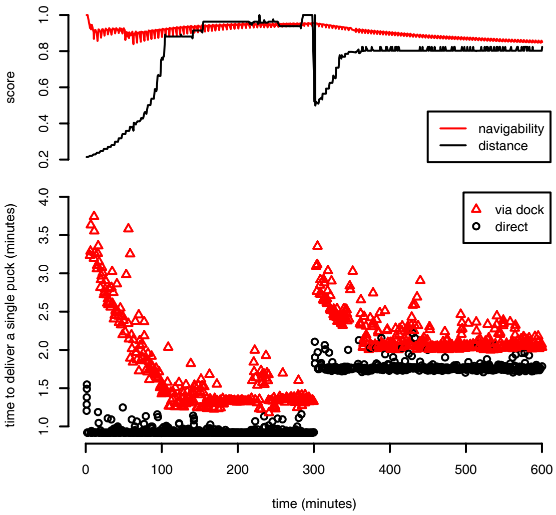


Fig. 9. Rate of puck delivery changes between tasks 1 and 2. The dock successfully detects the change in D and repositions itself to improve the delivery rate.

VII. CONCLUSIONS AND FUTURE WORK

A novel concept for online incremental improvement of charging station location was presented, along with a simple controller for a mobile dock. The method was shown to achieve performance comparable to a well-placed static dock in a simple simulation demonstration. An attractive feature is its ability to adapt to dynamically changing and random task schedules.

Future work will determine the feasibility of using an adaptive dock in a real world environment with rooms, corridors and obstacles. In this case the adaptive dock must be capable of positioning itself in a sufficiently large open area in order to minimize spatial interference.

We have ignored recharging the dock itself here. An obvious extension is to include a service robot capable of resupplying and managing a collection of adaptive docks.

An important general issue for future robotics is in optimizing energy consumption in addition to performance. The reduced travel overhead provided by a mobile dock could potentially have efficiency advantages, depending on the cost of moving the dock and the longevity of the whole system.

An interesting usage scenario is long-lived nomadic robot systems that explore and exploit vast terrains with no permanent installations.

Like our previous work on rendezvous [9], this system is an example of embodied iterated approximation, which we have proposed as a framework for approaching various parallel distributed, spatially embedded problems [13]. We aim to formally analyze its convergence properties and performance bounds as future work, but this paper provides proof of concept. Resources permitting, we will implement a physical adaptive robot mobile charging station.

VIII. ACKNOWLEDGEMENTS

Thanks to members of the Autonomy Lab for their contributions to the idea developed in this paper. This work was supported by NSERC and DRDC in Canada.

REFERENCES

[1] R. T. Vaughan, "Massively multi-robot simulations in stage," *Swarm Intelligence*, vol. 2, no. 2-4, pp. 189–208, 2008.

[2] P. Zebrowski and R. T. Vaughan, "Recharging robot teams: A tanker approach," in *Proceedings of the International Conference on Advanced Robotics (ICAR)*, Seattle, Washington, July 2005.

[3] (2008) iRobot home base. [Online]. Available: http://store.irobot.com/product/index.jsp?productId=2498248

[4] (2008) Robot docking / charging system. [Online]. Available: http://www.activrobots.com/ACCESSORIES/DXDock.html

[5] M. C. Silverman, D. Nies, B. Jung, and G. S. Sukhatme, "Staying alive: a docking station for autonomous robot recharging," in *Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on*, vol. 1, May 2002, pp. 1050–1055.

[6] J. Wawerla and R. T. Vaughan, "Near-optimal mobile robot recharging with the rate-maximizing forager," in *ECAL*, 2007, pp. 776–785.

[7] A. Drenner and N. Papanikolopoulos, "Docking station relocation for maximizing longevity of distributed robotic teams," in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, May 2006, pp. 2436–2441.

[8] E. Takeuchi and T. Tsubouchi, "Portable effector docking mechanism for a service mobile robot and its positioning," in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, May 2006, pp. 3380–3386.

[9] Y. Litus, R. T. Vaughan, and P. Zebrowski, "The frugal feeding problem: Energy-efficient, multi-robot, multi-place rendezvous," in *Proceedings of the IEEE International Conference on Robotics and Automation*, April 2007.

[10] D. Goldberg and M. J. Matarić, "Interference as a tool for designing and evaluating multi-robot controllers," in *In Proceedings, AAAI-97*. AAAI Press, 1997, pp. 637–642.

[11] M. Schneider-Fontan and M. J. Matarić, "Territorial multi-robot task division," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 5, pp. 815–822, Oct. 1998.

[12] A. Munoz, F. Sempe, and A. Drogoul, "Sharing a charging station in collective robotics," 8, rue du Capitaine Scott, 75015 Paris, Tech. Rep. LIP6 2002/026, 2002.

[13] Y. Litus and R. T. Vaughan, "Distributed gradient optimization with embodied approximation," in *Proceedings of the Eleventh International Conference on Artificial Life (ALife XI)*, August 2008.