Energy-Efficient Path Planning for Solar-Powered Mobile Robots

Patrick A. Plonski, Pratap Tokekar, Volkan Isler

Abstract We explore the problem of energy-efficient, time-constrained path planning of a solar powered robot embedded in a terrestrial environment. Because of the effects of changing weather conditions, as well as sensing concerns in complex environments, a new method for solar power prediction is desired. We present a method that uses Gaussian Process regression to build a solar map in a data-driven fashion. With this map, we perform energy-optimal path planning using a dynamic programming algorithm. We validate our map construction and path planning algorithms with outdoor experiments, and perform simulations on our solar maps to determine under which conditions the weight of added solar panels is worthwhile for a mobile robot.

1 Introduction

Mobile robots have the potential to perform many critical outdoor tasks but their potential for long-term deployment is limited due to energy concerns. A possible method to increase the battery life of robots is by harvesting energy from the environment, e.g. with photovoltaic solar panels. Solar harvesting has proven to be useful in marine and extra-terrestrial robotics applications [11, 1] which take place in open space. However, in applications where the robot must operate in complex environments, such as urban search and environmental monitoring, the utility of solar harvesting is not obvious. In this work we focus on extending the battery life of mobile robots using solar panels in such settings.

We study techniques for energy-minimizing path planning for a mobile robot with a photovoltaic panel that uses recent measurements of solar intensity as its

Patrick A. Plonski, Pratap Tokekar, Volkan Isler
Department of Computer Science and Engineering, University of Minnesota, 200 Union Street SE, Minneapolis MN 55455 USA
e-mail: plonski@cs.umn.edu, tokekar@cs.umn.edu, isler@cs.umn.edu

only source of information about future solar power. This is an interesting problem because there are many applications where mobile robots do not necessarily have the sensors or computing power to estimate solar maps using sophisticated techniques such as raytracing on 3d models of the environment. However, energy-efficient paths are still desired. Intuitively, it seems feasible for a good solar map of the environment to be built if the robot is in the field long enough. We provide experimental evidence to support this intuition.

To accomplish energy-efficient path planning, we first build a map of how much solar power the robot is likely to get in its operating environment (Section 2). Next we show how the robot's energy consumption can be modeled and how we can compute energy efficient paths given a solar map (Section 3). We present results from experiments that demonstrate the utility of our techniques (Section 4). We also present simulation results on our solar maps to demonstrate the utility of added solar panels on a robotic platform (Section 5).

1.1 Related Work

Energy efficient planning for mobile robots has received increased attention recently. Mei [8] studied the problem of modeling the power consumption of motion, sensing, communication and embedded hardware for commercially available robots. These power models are then used to compare various strategies for high-level tasks such as coverage, exploration and networking between robots, and increase the lifetime of the system.

Motion is a major source of power consumption for typical robots. Tokekar et al. [15], Wang et al. [17], and Kim and Kim [6] have studied the problem of minimizing the energy consumption by optimizing the velocity profiles for a given path. Sun and Reif [13] studied the problem of finding energy optimal paths between two points on terrains where the cost depends on friction and gravity and is thus direction dependent. They present an approximation algorithm for finding the minimum energy path, but do not optimize the velocity profile along the path. Liu and Sun [7] recently studied the problem of computing energy-efficient paths and trajectory profiles by optimizing the parameters of Bezier curves using an energy-based heuristic. However, the presented method is not guaranteed to minimize energy and the general problem of simultaneously optimizing the path and velocity for given start and goal pose remains unsolved.

Energy efficient motion planning in the context of applications such as coverage and data muling is a subject of recent study. Derenick et al. [2] studied the problem of maintaining persistent coverage using a network of robots by deriving control laws that allow robots with depleted batteries to reach corresponding access points. Similarly, Jensen et al. [5] presented strategies for reconfiguring robot formations for patrolling application.

Sugihara and Gupta [12] presented path planning algorithms for a data muling system for optimizing the trade-off between the energy consumption of the sensors and latency of the data carried by the robot. Tekdas et al. [14] studied the problem of finding time-efficient trajectories for a mobile robot downloading data from a set of wireless nodes, and by setting the parameters proportional to energy cost their approximation algorithm can minimize energy instead of time. In these works, the energy consumption of the robot is not considered. Here we present energy harvesting and path planning techniques that can potentially be useful for such applications.

The aforementioned works have not considered energy harvesting from the environment, and solar-aware path planning has received limited attention. In extraterrestrial applications and some environments on earth (e.g. in Antarctica [10]) collected solar energy can be treated as mostly independent of the path chosen. The TEMPEST mission-level path planner [16] uses ephemeris software to determine the position of the sun and then performs raytracing on known nearby terrain to build a solar map that is used to estimate the energy cost of paths. This is feasible when nearby terrain is known or when it can be accurately detected, but many otherwise feasible platforms for long-term environmental monitoring lack the necessary sensors to do this. In this paper we focus on predicting solar power in complex environments using only the robot's previously recorded position estimates and solar power measurements.

1.2 Problem Statement

Our problem statement is as follows: Suppose we have a mobile solar-powered robot that has been performing a task while also logging the power received from an onboard solar array. Each solar measurement is associated with an estimated robot position. Suppose the robot is required to perform a new task that requires it to reach a goal position within some time limit. How can the robot plan the path that minimizes its net energy consumption?

2 Solar Modeling

In this section we introduce the method we use to predict how much solar power the robot will receive at a given position. Before we present the details of our Gaussian Process (GP) regression, we first cover the basics of predicting electrical output from a photovoltaic panel.

2.1 Basics of Solar Power Prediction

The amount of current I a solar cell will output when it is fixed to a particular voltage V is the solution to the equation

$$I = I_L - I_s(e^{(V+IR_s)/V_T} - 1) - \frac{V+IR_s}{R_{SH}}$$

where I_s is the reverse saturation current of the diode and $V_T = \frac{k*T}{q}$ which is known as the thermal voltage. I_L is proportional to the number of photons that impact the solar cell, and therefore so is I. I decreases with higher voltage, but the effect isn't pronounced until the diode knee voltage is reached at around 0.5 volts for a silicon cell. The knee voltage increases with decreased temperature, but in general the voltage limit varies much less than the current.

Because the voltage of an individual cell is low, cells are usually connected in one or more strings such that each string is electrically in series. These strings have the property that the amount of current output is limited by the *weakest* cell in the string (ignoring the effect of bypass diodes). The weakest cell could be the cell with the smallest dot product between its normal vector and the sun angle vector, or it could be a cell which happens to be in a shadow. This response to partial shading of the array causes the correct solar map to have sharp edges between sun and shade.

Sunlight reaches a solar panel in three different ways[4]: If it comes directly from exactly the part of the sky that contains the sun, it is called direct insolation. If it comes from any other part of the sky, it is called diffuse insolation. Finally, if it comes from anywhere else (i.e. from terrain or objects), it is called reflected insolation. Reflected insolation is most relevant when a solar panel is tilted towards a reflective surface (such as snow), or near a reflective building. On a sunny day direct insolation is high and diffuse insolation is low whereas on a cloudy day direct insolation is low and diffuse insolation is high (and total insolation is much lower than on a sunny day). If a cell has no line of sight to the sun it is in a shadow, and direct insolation drops to zero. However, for diffuse insolation to drop to zero the entire sky must be blocked. Therefore we can expect shadows and therefore the correct solar map to be much sharper on a sunny day than on a cloudy day.

It is challenging to detect the environment and perform raytracing for these three types of insolation so we sidestep and instead construct our solar map using regression from prior measurements of solar power associated with positions.

2.2 Gaussian Process Regression

A Gaussian Process (GP) is defined as a set of random variables such that any subset of the random variables has a joint Gaussian distribution [9]. GP regression is a general regression technique used to predict the most likely value of a function at any point given measured values of the function at some other points, without assuming an explicit parametric model for the function. GP regression, however, requires a suitable covariance function to model the joint Gaussian distribution for points. For more details on GP regression in general see [9].

In our application we associate each measurement of solar power with a position and use GP regression to predict the distribution of solar power at any desired position. When all of the solar cells are horizontal, or if they are otherwise suitably symmetric, the rotation of the robot can be ignored in these position measurements. This makes the solar map easier to learn by eliminating a dimension along which solar power can vary. In this paper we neglect the solar map's time dependence from the changing position of the sun. This is justified when the robot stays in the same environment each day, and can therefore build a separate solar map for various discrete time segments.

In Section 4.4 we present more details of our particular implementation of GP regression, and we empirically compare the performance of different covariance functions.

3 Path Planning

In this section we show how we use a solar map to plan the path that will reach the goal within the time limit while consuming the least amount of energy overall.

Our robot is differential-driven, so it can turn in place, and turning is a relatively expensive operation. We empirically determine in Section 4.3 that for our robot the energy consumption of a path with a certain top speed is well represented as a short initial spike during acceleration, and then a steady cost per meter traveled. the planned path as time-stamped waypoints with straight line segments connecting them, each line segment traversed at a constant speed with instantaneous speed changes between line segments. We model the energy sent to the motors as the following: At any particular speed, there is a constant cost per meter traveled C_s, a constant cost per radian rotated C_r, and an initial acceleration cost C_a. When transitioning from a non-zero speed, the acceleration cost is the C_a for the new speed minus the C_a for the old speed, but with a minimum cost of 0. This makes sense if we assume that acceleration cost is proportional to kinetic energy. We can mathematically state the cost of traversing line segment l_i:

$$cost_i = C_s(speed_i)|l_i| + C_r(speed_i)|\theta_i - \theta_{i-1}| + \max(C_a(speed_i) - C_a(speed_{i-1}), 0)$$

The cost constants as functions of speed are specific to the robot and the terrain. The terrain where our experiments were conducted was flat and uniform, so in this work we do not consider changes in elevation, friction, or rolling resistance.

The total cost of a path is given by the sum over the path $\sum_{i=0}^{n-1} cost_i$ minus the expected amount of solar energy collected while traversing the path. An idle power draw (constant) can be subtracted from the solar power; we do not consider idle power draw because our focus is on path planning and idle power does not affect the optimal path to reach the target in the time scales we consider.

The Algorithm:

The expected value for any particular point in our solar map can be determined in closed form, however there is no convenient closed form model for the entire map as a whole; that is, there is no general geometric model we can use to represent our environment. Therefore some amount of discretization of the solar map is necessary for us to do planning. It is possible in this domain to plan on a set of sampled actions or path shapes (e.g. with an sampling based planner) but since the state space is relatively small we use a complete grid. We then perform dynamic programming to compute the optimal solution for a given resolution. We discretize both space and time, and we also have a dimension in the dynamic programming table for heading and a dimension for whether the robot is moving or the robot is waiting, to account for the cost to rotate and the cost for initial acceleration. In this way we ensure that the output path is always optimal in its resolution, according to our power to drive model. The trajectories generated by our algorithm move at a constant speed when they are on Manhattan edges and a faster constant speed when they are on diagonal edges; traversing to any neighboring state takes the same amount of time.

We observe from the output of this algorithm that optimal trajectories consist of either continuous movement, continuous movement with a wait at the beginning or the end, or continuous movement broken up by a wait in the middle. As more time is allowed the optimal path transitions between those three types: at first there is no time to wait anywhere, then there is time to wait but not enough to compensate for the energy loss from having to re-accelerate, and then finally there is enough time to wait somewhere in the middle for long enough to recoup the extra acceleration cost and possibly enough time to allow deviation from a shortest path. See Figure 3a for examples of planned paths output by our algorithm.

4 Field Experiments

We performed three sets of experiments in the environment shown in Figure 1b: we calibrated our power to drive parameters, we measured solar panel current along paths and used this to construct solar maps using different covariance functions, and we executed energy-minimal paths that were planned on these maps.

4.1 System Description

The chassis of our system was a Husky A100, built by Clearpath Robotics[1]. The A100 is a six wheel, two motor, differential drive machine. The datasheet mass is 35 kg, the maximum payload is 40 kg, and the dimensions are 0.860 meters long by 0.605 meters wide by 0.350 meters tall. In its experimental configuration the A100 was powered by a single lead-acid battery that was nominally 12v and 21 amp hours. See Figure 1a for a photo of the A100 during one of our experiments.

[1] http://www.clearpathrobotics.com/

The solar panels used by our system were two SPM020Ps from Solartech Power[2]. The SPM020P supplies 20w at the optimal voltage of 17.2v under standard test conditions of 1000 w/m^2 insolation and a temperature of 25^{o}C. The panel is wired as a single series string with 36 cells in it. The dimensions are 560x360x18(mm), and each panel nominally weighs 2.5kg.

We placed the panels horizontally on the robot for ease of mounting, for quality in overcast conditions, and to eliminate the dimension of panel rotation in the solar map built. Both panels were connected in parallel with the battery; therefore solar panel current was proportional to solar power. Battery voltage and motor current measurements were provided by the A100, and current from the panel to the battery was measured with a hall-effect current sensor.

Localization of the robot was achieved by using an EKF to fuse GPS measurements with wheel-encoder propagation.

4.2 Terrain Description

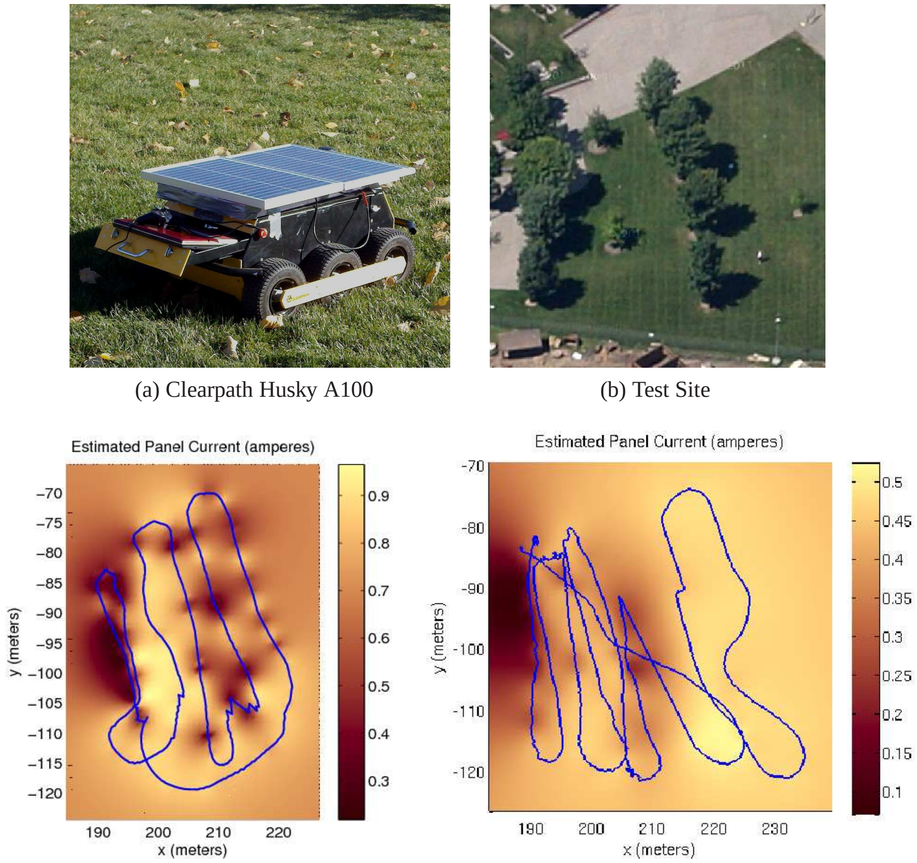
We performed our experiments in the field next to the McNamara Alumni Center, on the Minneapolis campus of the University of Minnesota (see Figure 1b). The field is roughly 40 meters by 30 meters and it is relatively flat, with uniform short grass. Other than a few poles the only objects that occlude the sun are scattered trees. While our calculated power to drive parameters and solar map parameters are likely to change in other environments, the methodology we present here to obtain those parameters remains the same.

We performed our experiments on dry days when there was no snow on the ground. We would expect power to drive to significantly change in wet weather or if there is accumulated snow. All solar parameters except the chosen covariance function were re-learned for each new solar map; this was necessary to account for short term changes from the varying position of the sun, medium term changes from varying weather, and long term changes. One of these long term changes was a seasonal change in solar power that occurred as the leaves fell off the trees as summer turned to winter.

4.3 Power to Drive Experiments

We controlled the forward movement of the A100 by directly setting the motor voltage. We found that this method required less energy than using a closed loop PID speed controller. For a particular motor voltage and on particular terrain, the A100 travels at a particular steady-state speed and consumes a steady amount of energy per unit distance traveled, after a brief acceleration period. To characterize

[2] http://www.solartechpower.com/



(a) Clearpath Husky A100

(b) Test Site

(c) Sunny Solar Map

(d) Cloudy Solar Map

Fig. 1: Our configuration of the A100 (a), top-down view of the test site (b), solar map constructed for 13:42 on November 18, 2011 (c) (this was a sunny day), and solar map constructed for 11:22 on September 16, 2011 (d) (this was a cloudy day). Both solar maps are overlayed with their source paths. The cloudy map was built by sampling with only a single solar panel.

the steady-state cost and acceleration cost we drove straight at a variety of commanded motor voltages and fit a line to the plot of cumulative cost vs. distance for each voltage. The slope of the line determined the steady state cost and the intercept determined the acceleration cost. Then we performed linear regression on the steady state costs as functions of speed and quadratic regression on the acceleration costs as functions of top speed, and ended up with the following equations for our parameters C_s and C_a (see Figure 2):

$C_s = (-17.6624 * speed + 139.4576)$ Joules per meter

$C_a = (321.0671 * speed^2 - 285.3912 * speed + 154.9553)$ Joules to accelerate

Then to characterize turning cost we commanded a tight left turn and tight right turn, and examined the steady state energy per radian.

$C_r = 406.5963$ Joules per radian

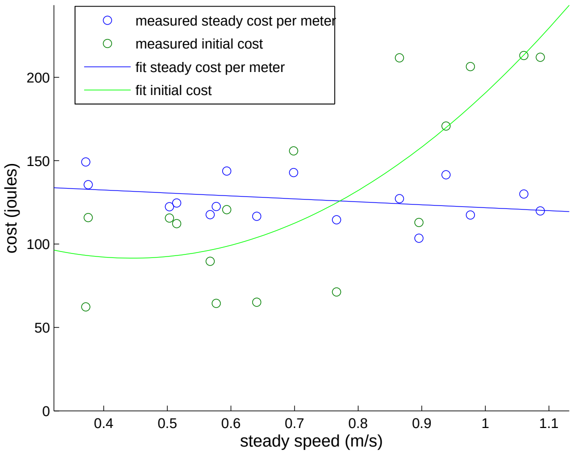


Fig. 2: Power To Drive Test Results

4.4 Solar Map Construction

The input for the solar map is a long path with noisy measurements of solar current taken at 20 Hz, each measurement associated with a position on the path. This accumulates to a very large number of measurements if the robot is embedded in the environment for a long time. As GP regression relies on matrix multiplication of all training points, using all measurements as individual training points becomes infeasible. Fortunately, since we only care about associating solar current to $x-y$ position we can discard information about rotation and time and combine measurements with similar $x-y$ position. In this way the number of measurements considered by the GP regression is bounded by the size of the environment rather than the length of time the robot is collecting data. Also it is valuable for optimizing the GP hyperparameters for measured positions to be weighted equally instead of weighted in proportion to the amount of time the robot has spent there.

In our implementation we placed in a bucket all measurements that were within 0.3 meters of the first measurement and then removed them from the list, and repeated this process until every measurement was in a bucket. The bucket's position was set as the centroid of the positions of the measurements in it, and its value was set as the mean of the values of the measurements in it. We calculated the variance of each bucket from the variance of the measurements in the bucket, treating the bucket solar current as an average of uncorrelated random variables. Then for the regression we treated the noise variance as equal to the average of the variances of the buckets. This was again to induce balanced weighting of different areas; if the robot had waited 20 minutes at the same position we did not want the bucket containing that position to be significantly more valuable than nearby buckets because still only a small portion of the possible points that could go into that bucket would have been explored. The prior mean and prior variance were computed from the mean and variance of the set of buckets.

To perform GP regression we need a covariance function. For this we considered different versions of the Matérn covariance function (detailed in [9]). The Matérn class of covariance functions is given by:

$$k(r) = \frac{2^{1-v}}{\Gamma(v)} \left(\frac{\sqrt{2v}r}{\ell} \right)^{v} K_v \left(\frac{\sqrt{2v}r}{\ell} \right)$$

where v is a positive parameter that affects the smoothness of the process, ℓ is the positive length parameter, and K_v is a modified Bessel function. If v is $1/2$ the function becomes the exponential covariance function, and as $v \to \infty$ the function becomes the squared exponential covariance function. Other than the exponential and the squared exponential, the most commonly used Matérn covariance functions are where $v = 3/2$ and $v = 5/2$, so those are the covariance functions we tested in addition to the exponential and squared exponential.

To optimize the Matérn function's length hyperparameter we performed numerical gradient-descent searches maximizing the likelihood of the observed values given the covariance function. We compared the likelihoods of the different Matérn functions on various data sets we collected and we found that $v = 1/2$ was the most likely on two out of three of the cloudy days tested, and five out of eleven of the sunny days tested, with $v = 3/2$ the most likely on the other days. However, even on the days where it was the most likely, the solar maps constructed using $v = 3/2$ had overshooting at the sharp boundaries between sun and shade. This overshooting made it so that the positions with the most predicted solar power were close to boundaries, and therefore planned energy-minimal paths were pulled towards boundaries. As a real system always has some localization error, a better strategy is to stay away from shadows if possible. This is the behavior that results when we use $v = 1/2$ in our regression, so that is what we did even though it was often less likely given the data and our GP assumption.

Holding $v = 1/2$, the most likely length varied between 2.05 meters and 12.65 meters on sunny days, and between 3.68 meters and 18.67 meters on cloudy days. This difference is because diffuse insolation dominates over direct insolation on cloudy days, and diffuse insolation varies slower than direct with changing position.

4.5 Path Planning and Execution

At 13:10 on February 18, 2012 we drove the A100 around the field in Figure 1b, optimized the length hyperparameter for that dataset with an exponential covariance function, used GP regression to build a solar map, planned paths with our planner detailed in Section 3, and then executed the paths. The A100 had some localization error even when GPS worked well, so a fairly low spatial resolution of 5m was used. Temporal resolution was set to 8 seconds. To calculate the expected solar current in a grid square the expected solar current was calculated on a higher resolution 1m grid and then downsampled. In addition to the planned solar-aware paths the A100 also executed shortest paths after we removed the solar panels (slightly decreasing the power to drive due to decreased weight) from the same start position to the same end position. These paths provide a comparison, allowing us to directly demonstrate the utility of the added panels. See Table 1 for summaries of the executed paths.

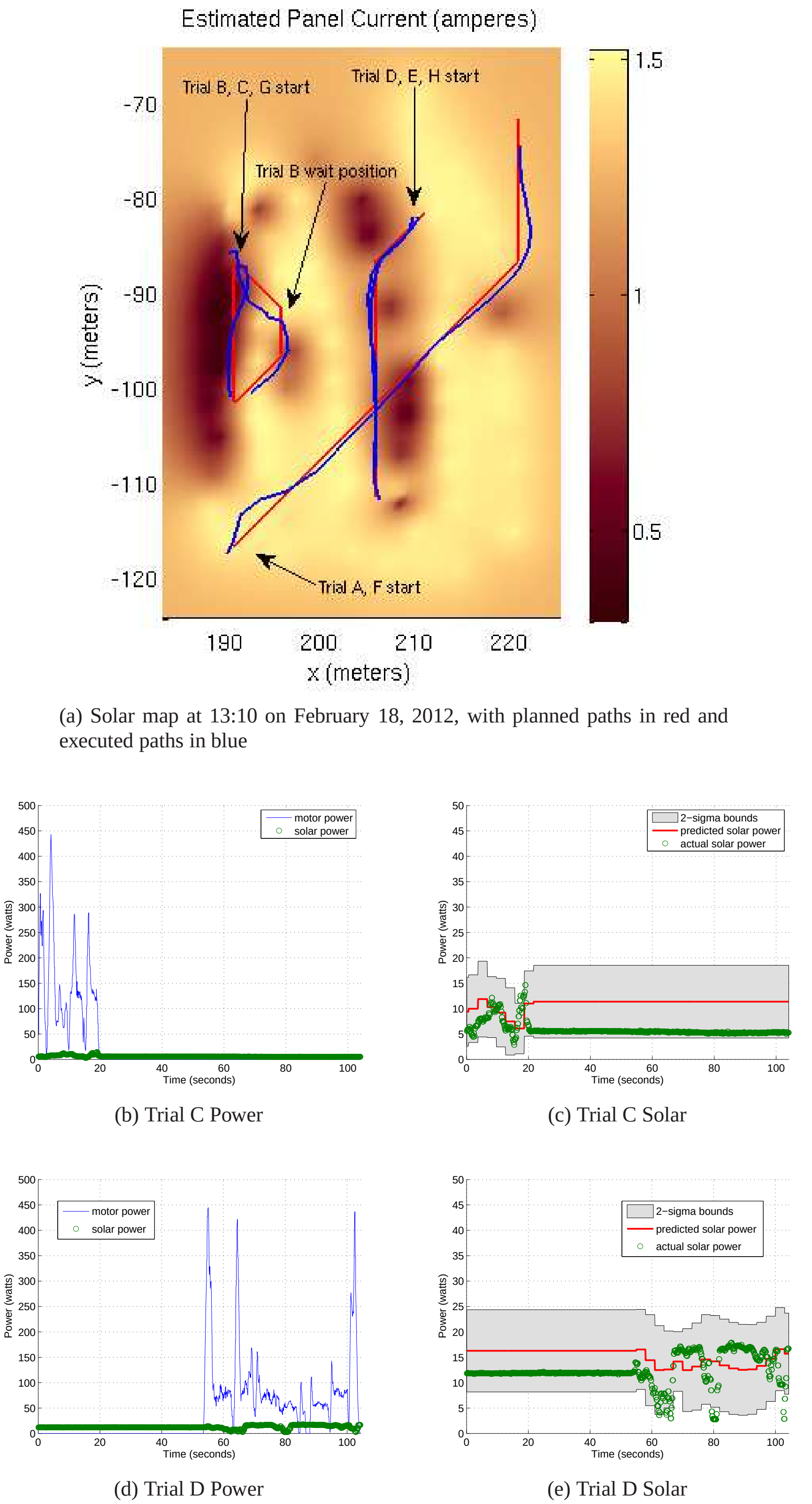
Solar Trial	Duration	Expected Solar	Actual Solar	Expected Cost	Actual Cost	Control Trial	Duration	Cost
A	401 s	7,025.5 J	6,974.1 J	577.16 J	744.6 J	F	45 s	6,295.4 J
B	400 s	6,606.6 J	6,828.6 J	−3,265.9 J	−3,256.7 J	G	19.1 s	2,888.1 J
C	104 s	1,148.3 J	611.26 J	879.99 J	2,253,4 J			
D	104 s	1,600.9 J	1,297.6 J	2,907.3 J	3,480.3 J	H	30.4 s	3,530.4 J
E	104 s	1,600.9 J	1,156.2 J	2,907.3 J	2,822.5 J			

Table 1: Path Execution Results

5 Power Comparison

To further investigate the benefits gained from solar panels, we ran simulated comparisons between our solar powered robot using our path path planner and our robot stripped of its panels driving straight towards the destination. We picked a start position and end position, planned the optimal solar-aware path for a range of time limits, and compared the cost to drive straight without a panel with the distribution of likely solar robot costs. For these simulations we did not consider localization errors, so we increased the resolution of our planning grid to 2 meters per square, 3 seconds per square. We intentionally chose start and end positions in the shade, to see how the system would perform under somewhat adverse conditions.

First we considered a robot traveling from the southwest part of the trees to the northeast part of the trees, at 13:10 on February 18 (the same day as our path execution trials). For details of this simulation see Figures 4a and 4b. The start position was $(188, -109)$ and the end position was $(208, -89)$. At a speed of 1 m/s we expect the baseline path to consume 3,065.0 Joules. For the solar robot to be on average more energy efficient than the baseline it requires at least 42 seconds to execute its path. This is an overall speed of 0.6734 m/s. For the solar robot to be more energy



(a) Solar map at 13:10 on February 18, 2012, with planned paths in red and executed paths in blue

(b) Trial C Power

(c) Trial C Solar

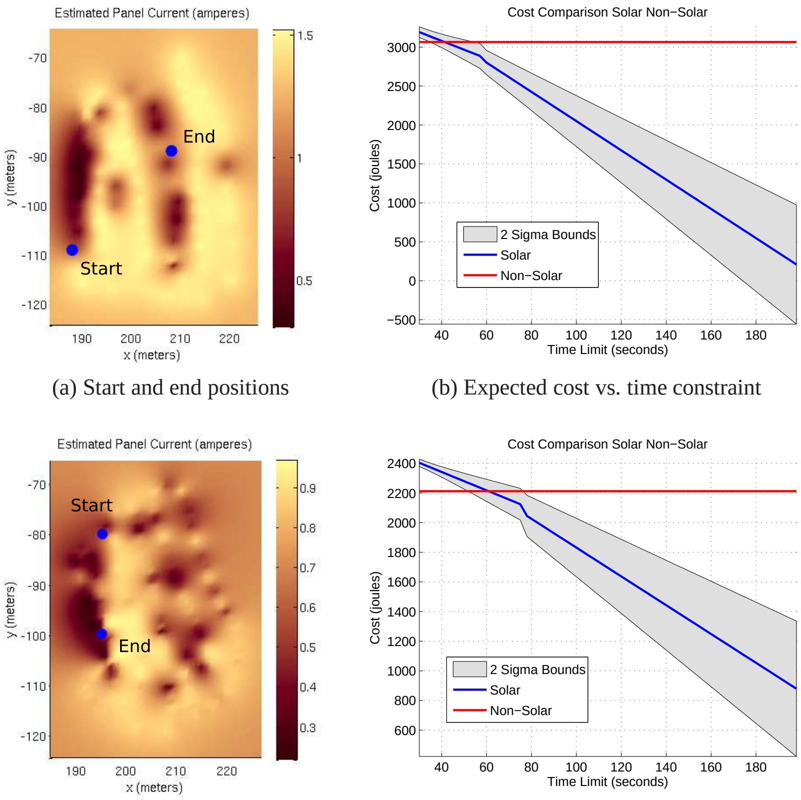
(d) Trial D Power

(e) Trial D Solar

Fig. 3: Planned solar-aware paths and example trials. Note that in trial D the planner chose to wait at the beginning given the information it had but it turned out the position at the end of the path received more solar power.

efficient with 95% confidence, it requires at least 57 seconds which is an overall speed of 0.4962 m/s.

Second we considered a robot traveling south through the shade of the west line of trees, at 13:42 on November 28. For details of this simulation see Figures 4c and 4d. The start position was $(195, -80)$ and the end position was $(195, -100)$. At a speed of 1 m/s we expect the baseline path to consume 2,211.9 J. For the solar robot to be on average more efficient than the baseline it requires at least 63 seconds which is an overall speed of 0.3175 m/s. For the solar robot to be more efficient with 95% confidence it requires at least 78 seconds which is an overall speed of 0.2667 m/s.



(a) Start and end positions

(b) Expected cost vs. time constraint

(c) Start and end positions

(d) Expected cost vs. time constraint

Fig. 4: Simulations for 13:10 on 02-18-2012 (a and b) and 13:42 on 11-28-2011 (c and d). When not much time is allowed the weight of the solar panels ensures that the cost of carrying them is greater than the benefit of solar power, however when the robot is allowed to wait a while in the sun the benefit of panels can be large.

6 Experimental Insights and Concluding Remarks

In our experiments, we observed that true solar energy collected during a trial was close to the expected solar energy obtained from GP regression. However, the predicted probability distributions did not necessarily resemble the true distributions. This is because the probability distribution of sunlight at a point is poorly modeled by a Gaussian distribution: on a sunny day the correct probability distribution of expected solar power at any given point is bimodal, with separate peaks of expected power for the case where the panel is in the sun and the case where it is in the shade. Since Gaussian models cannot capture this behavior well, it may not be best to optimize the covariance function hyperparameters for maximum likelihood. This is an issue we plan to investigate further.

On February 18 the system did not lose much accuracy by neglecting to consider the sun's movement, though the solar map was constructed for 13:10 and the last solar trial (trial E) began at 14:19. The impact of moving shadows may have been mitigated by the fact that shadows were sparse due to bare branches on the trees.

Our power to drive model was reasonably accurate. It tended to underestimate power to drive but not by much: on average it missed by 396.5 J, which was on average 11.2% off from the true value. It underestimated four times and overestimated once. This indicates that our learned parameters were correct and that the A100 waypoint navigation software was not performing too many corrective turns. To get the waypoint navigation software to this state we disallowed backtracking and instead counted the waypoint as reached whenever the plane perpendicular to the path was crossed. This had the effect of slightly decreasing solar prediction accuracy, but also significantly decreasing average power to drive for a trial.

Our path planner worked well at its resolution. If we move to higher resolution there is a danger of the following: the path planner chooses to wait in a position that has sun but due to localization error the A100 ends up waiting in the shade, and an expected good path becomes very bad. With our path planning there was very high cost to deviate from a straight path: the cost of four 45^o turns and at least 10 meters increased distance. Therefore if there is not much time the optimal path will choose to wait at the sunniest spot on the shortest path instead of deviating to a sunnier spot that is slightly off the path. It might be feasible to use something such as Field D* [3] to plan smoother paths that vary only slightly from the shortest path.

Our simulation results show that with our platform and in the environment we tested, the addition of heavy commercial solar panels decreases cost on sunny days in November and February only if the average speed is not required to be greater than 0.6734 m/s for the trial in February or greater than 0.3175 m/s for the trial in November. These were both sunny days, but they were particularly challenging for sunny days: it was the dark part of the year, and the trials both started and ended in the shade. We would therefore expect the addition of solar panels to be feasible in many situations requiring higher average speeds.

In our future work, we will investigate the effect of the varying sun angle on our solar maps, as well as methods to use the known sun angle to improve our predictions. We also plan to further investigate methods of optimizing the hyperparameters, and methods to plan smoother paths on our solar map.

Acknowledgements This material is based upon work supported by the National Science Foundation under grant numbers 1111638, 0916209, 0936710, and 0934327.

References

1. J. Carsten, A. Rankin, D. Ferguson, and A. Stentz. Global path planning on board the mars exploration rovers. In *Aerospace Conference, 2007 IEEE*, pages 1 –11, march 2007.
2. J. Derenick, N. Michael, and V. Kumar. Energy-aware coverage control with docking for robot teams. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 3667 –3672, sept. 2011.
3. D. Ferguson and A. Stentz. Using Interpolation to Improve Path Planning : The Field D * Algorithm. *Journal of Field Robotics*, 23(2):79–101, 2006.
4. D. Y. Goswami, F. Kreith, and J. F. Kreider. *Principles of Solar Engineering*. Taylor & Francis, 2nd edition, 1999.
5. E. Jensen, M. Franklin, S. Lahr, and M. Gini. Sustainable multi-robot patrol of an open polyline. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 4792–4797. IEEE, 2011.
6. C. Kim and B. Kim. Minimum-energy translational trajectory generation for differential-driven wheeled mobile robots. *Journal of Intelligent & Robotic Systems*, 49(4):367–383, 2007.
7. S. Liu and D. Sun. Optimal motion planning of a mobile robot with minimum energy consumption. In *Advanced Intelligent Mechatronics (AIM), 2011 IEEE/ASME International Conference on*, pages 43 –48, july 2011.
8. Y. Mei. *Energy-Efficient Mobile Robots*. PhD thesis, Purdue University, 2006.
9. C. Rasmussen and C. Williams. *Gaussian processes in machine learning*. The MIT Press, 2006.
10. L. Ray, J. Lever, A. Streeter, and A. Price. Design and Power Management of a Solar-Powered Cool Robot for Polar Instrument Networks. *Journal of Field Robotics*, 24(7):581–599, 2007.
11. C. Sauze and M. Neal. Long term power management in sailing robots. In *OCEANS, 2011 IEEE - Spain*, pages 1 – 8, june 2011.
12. R. Sugihara and R. Gupta. Optimizing energy-latency trade-off in sensor networks with controlled mobility. In *INFOCOM 2009, IEEE*, pages 2566–2570. IEEE, 2009.
13. Z. Sun and J. Reif. On finding energy-minimizing paths on terrains. *Robotics, IEEE Transactions on*, 21(1):102–114, 2005.
14. O. Tekdas, D. Bhadauria, and V. Isler. Efficient Data Collection from Wireless Nodes under the Two-Ring Communication Model. *The International Journal of Robotics Research*, 31(6):774–784, 2012.
15. P. Tokekar, N. Karnad, and V. Isler. Energy-optimal velocity profiles for car-like robots. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 1457–1462. IEEE, 2011.
16. P. Tompkins, A. Stentz, and D. Wettergreen. Mission-level path planning and re-planning for rover exploration. *Robotics and Autonomous Systems*, 54(2):174–183, 2006.
17. G. Wang, M. Irwin, H. Fu, P. Berman, W. Zhang, and T. Porta. Optimizing sensor movement planning for energy efficiency. *ACM Transactions on Sensor Networks (TOSN)*, 7(4):33, 2011.