

A Realistic Model of Frequency-Based Multi-Robot Polyline Patrolling

Yehuda Elmaliach, Asaf Shiloni, and Gal A. Kaminka
The MAVERICK Group
Computer Science Department
Bar Ilan University, Israel
{elmaley,shilonia,galk}@cs.biu.ac.il

ABSTRACT

There is growing interest in multi-robot frequency-based patrolling, in which a team of robots optimizes its frequency of point visits, for every point in a target work area. In particular, recent work on patrolling of open polygons (e.g., open-ended fences) has proposed a general cooperative patrolling algorithm, in which robots move back and forth along the polygon, in an synchronized manner, such that their assigned areas of movement overlap. If the overlap factor is carefully chosen—based on the motion models of the robots—specific performance criteria are optimized. Unfortunately, previous work has presented analysis of motion models in which there are no errors in the movement of the robots, and no velocity changes. We go a step beyond existing work, and develop a realistic model of robot motion, that considers velocity uncertainties. We mathematically analyze the model and show how to use it to find optimal patrolling parameters, given known bounds of uncertainty on the motion. We then use the model to analyze the independently-programmed patrolling movements of physical robots, in extensive experiments. We show that the model predicts the behavior of the robots much more accurately than previously-described models.

Categories and Subject Descriptors

I.2.9 [Robotics]: Autonomous vehicles
; I.2.11 [Distributed Artificial Intelligence]: Multiagent Systems

General Terms

Algorithms, Experimentation, Security

Keywords

Multi-Robotics, Multi-robot path planning, Motion planning, Team planning

1. INTRODUCTION

There is growing interest in frequency-based multi-robot patrolling (also known as repeated coverage) in which a team of robots optimizes its frequency of point visits, for every point in a target work area [2, 3, 7], or a target polyline circumscribing the work area [17]. Several optimization criteria are possible, such as

Cite as: A Realistic Model of Frequency-Based Multi-Robot Polyline Patrolling, Elmaliach, Shiloni, and Kaminka, *Proc. of 7th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2008)*, Padgham, Parkes, Müller and Parsons (eds.), May, 12-16., 2008, Estoril, Portugal, pp. 63-70.

Copyright © 2008, International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). All rights reserved.

uniformity of point-visit frequency, increased average frequency, etc. [7]. Frequency-based patrolling is a task useful for applications such as waste cleaning and monitoring [4, 14], and surveillance [17].

Recent work has begun to address patrolling of polylines that describe an open polygon, where the two extremities of the polylines are not connected (e.g., as in open-ended fences). A general algorithm for cooperative multi-robot patrolling in such cases is described in [8]. Here, robots move back and forth along the polygon, in an synchronized manner, such that their assigned areas of movement overlap. Patrolling of open polygons pose significant challenges to frequency-based performance criteria, because in open polygons, robots reaching the endpoints necessarily have to backtrack over points that they just visited, thus visiting them again immediately after the previous visit. By carefully choosing the overlapping factor, the negative effects of the endpoints can be mitigated.

Unfortunately, previous work has presented analysis of motion models in which there are no errors in the movement of the robots, no velocity changes, and no uncertainty in the planned trajectories. In particular, the analysis assumes that robot movement along longer segments of the polylines are handled at exactly the same velocity as shorter distances, i.e., that the velocity is independent of the number of polyline segments assigned to the robot. This does not allow analysis of patrolling performance in realistic settings, where traveling longer distances often involve accumulating motion errors, which result in increasing delays and velocity shifts.

We thus go beyond existing work, and develop a realistic model of robot motion, that considers real-world velocity uncertainties and accumulating motion errors. We mathematically analyze the model and show how to use it to find optimal patrolling parameters, given the velocity profile of the robot for different travel distances. The model we develop can therefore account for accumulating errors in motion, and handle bounds on the expected performance of the robots.

To evaluate the efficacy of the realistic model, we have conducted extensive experiments, using independently-programmed physical robots, patrolling a mock fence in our lab. The robots were programmed to follow the general overlapping patrol algorithm [8], without imposing any a-priori restrictions on their motion. Post-hoc analysis revealed that the model presented in this paper accurately predicts the actual behavior of the robots, and is significantly more accurate than the theoretical model described in previous work.

2. RELATED WORK

Many previous investigations have focused on multi-robot patrol *inside an area*. Elmaliach et. al [7] pose patrolling as a

visit-frequency optimization problem, and describe three optimization criteria: Maximizing the minimal visit frequency (bounding the worst frequency from below), maximizing the average frequency, and maintaining maximally-uniform frequency (e.g., with the smallest frequency standard deviation). They present an algorithm for multi-robot patrol inside a closed area where movement direction and velocity constraints may change between different portions. Their algorithm generates a minimal-cost cyclic path visiting all points in the area, satisfying all frequency optimization criteria they describe. In contrast to this work, this paper focuses on patrolling along an open polygon, rather than inside an area. Here, no circular path is possible, and thus the algorithms previously presented cannot apply.

Another approach to patrolling an area is to divide the area between the robots to sub areas, where each robot is responsible for the sub area it is positioned in. Ahmadi and Stone [2] describe a negotiation-based approach for dividing the area between the robots. Guo et. al ([12, 13]) also divide the area between the robots while focusing on their localization and sensing capabilities. In this paper, the division of the polygon into segments is done based frequency optimization criteria, rather than negotiations. The robots are assumed to be cooperative, and acting as a team.

Ryale et al. [16] and Girard et al. [11] describe architectures for multiple robot patrolling a border (a polyline), using unmanned aerial vehicles. These systems focus on allowing a single operator to operate and command multiple robots. They do not address point visit optimization criteria. In contrast to these investigations, our work focuses on autonomous patrolling, taking into account velocity constraints along the path, and optimizing frequency-related criteria.

Correll and Martinoli [5, 6] describe a distributed coverage for swarm robots, which combines elements of both area and boundary coverage. They implement their idea on swarm robots to cover turbine elements aligned on a surface area. Thus the robots have to cover the work area completely, yet each detected element is to be circumscribed by the robots. The main idea is to combine probabilistic and deterministic models in order to achieve better real-world performance. However, this work does not address frequency of repeated coverage.

We emphasize that patrolling, as studied in this paper, is investigated from the point of view of optimizing point visit frequency. There are alternative optimization criteria for patrolling. For instance, Paruchuri et. al. [15] and Agmon et al. [1] study patrolling in adversarial environments, in which the robots' goal is to maximize their rewards. These rewards are received if the robots manage to observe the adversary, which tries to evade the patrolling robots.

3. POINT VISIT FREQUENCY

We first briefly remind the reader of frequency-based patrolling, and the general frequency-based overlapping patrol algorithm (Section 3.1). We then discuss a robot motion model that takes motion errors into account (Section 3.2), and use it to analyze the performance of the algorithm in realistic settings (Section 3.3).

3.1 Frequency-Based Overlapping Patrol

In frequency-based patrolling, a key challenge is to assign trajectories to robots, such that the robots repeatedly visit points in a target area or line, while optimizing some point visit-frequency criteria. Earlier work on patrolling introduced three frequency-based performance criteria for frequency-based patrolling [7]:

- **Uniformity.** The goal is to decrease the variance between

the frequencies in which each target is visited, i.e., all targets should ideally be visited with uniform frequency f .

- **Maximal average.** The goal is to increase the average frequency f in which targets are visited.
- **Maximal minimum frequency (under-bounded frequency).** The goal is to increase the minimal frequency f with which any target point is visited, such that every target is visited with frequency of at least f . In other words, all targets should be monitored at least once every $1/f$ cycles.

Unfortunately, for a single robot, perfect uniformity of point visit frequency is impossible to achieve in open polygon patrolling. The fact that the fence is not circular prevents the robot's path from being continuous and thus at some point the robot needs to change direction. The direction change forces the robot to back-track over points in the path that it has visited only moments before, and therefore the visit frequency is non-uniform along the path.

From a more formal perspective, the argument is as follows. The basic motion for a single robot along a fence is monotonic movement from left to right and vice versa. Suppose a robot is patrolling a fence of length L . Let us focus on an arbitrary point of interest along the fence, at a distance X from one of the endpoints of the fence. Assuming a naive motion model, where turning does not take any time and there is no uncertainty in motion, the times at which the point will be visited form the following series $2X, 2(L - X), 2X, 2(L - X)$ (assuming unit velocity for simplicity). The frequency of visits is therefore uniform only in the midpoint of the fence (when $X = L/2$), while the variance in visit times grows towards the endpoints of the fence.

Previous work [8] has shown that using r multiple robots, we can improve the uniformity of the point visit-frequency (hereinafter, *frequency* for brevity). All robots execute the frequency-based overlapping patrol (FOP) algorithm (Algorithm 1). The key idea in FOP is that each robot patrols more than a single segment. Each robot begins by moving along its own segment, but then, depending on the *overlap factor*, may move into adjacent segment (while the robot in this segment is moving into the next segment, etc.). Thus the robots trajectories overlap in space, but not in time.

The FOP algorithm (Algorithm 1) calculates the patrol movement for a robot in a fence where the overlap factor is o , robot i is initially located in segment i , and the number of segments is r (equal to the number of robots). Each robot i (of the r robots that participate in the patrol) runs this algorithm in a distributed fashion. The algorithm assumes that the open polygon has already been divided into r equal-time segments, and that all robots start at the beginning of their assigned segments, facing towards the direction of movement. Also, the algorithm assumes perfect communications (to allow the robots to synchronize their turns) and localization along the fence.

The first step of the algorithm moves the robot o segments (by the overlap factor). In case where the robot arrive to the right last segment, the $r - i + 1$ value ensure that the robot will not move beyond the fence boundary. The second step ensures that the robots will synchronized and wait until the entire robots arrive to their destination. The third step occurs only for the robots that collect at the fence endpoint. They must wait until the other robots have left the segment. In the fourth step each robot returns to the segment that it started from. Finally, all robots turn in place, synchronized again, and repeat the process.

The behavior of the robots in FOP is thus dictated by a single parameter, the overlapping factor o . Figure 1 shows the trajectories assigned to a fixed set of robots, for a given open polygon, using different overlap factors.

Algorithm 1 FOP(overlap factor o , robot id i , number of robots r)

1. Move $\min(o, r - i + 1)$ segments, using the velocity constraints of each segment.
 2. Turn in place and synchronize with others
 3. If you are in the right edge segment wait until your left robot neighbor is one segment far
 4. Move to your base segment
 5. Turn in place and synchronize with others
 6. Return to step 1.
-

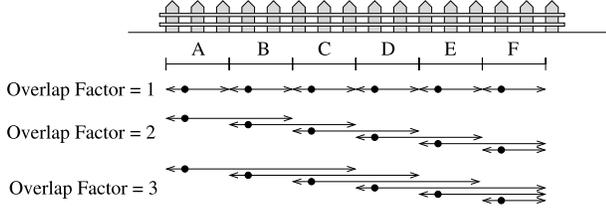


Figure 1: An illustration of the effect of overlap factor on FOP behavior.

By carefully choosing the value of the overlap factor o , different trajectories are generated, and as a result, different results along the three optimization criteria. For instance, if we assume point robots with precise positioning, and instantaneous turns, increasing o leads to improved uniformity and maximal minimum frequency in all segments which are visited by o robots (called *middle segments*, while sacrificing the results in the segments adjacent to the endpoints (*edge segments*) [8]. Since many applications have many more middle segments than edge segments, this trade-off may be beneficial.

The selection of an optimal overlap factor thus critically depends on the model for the robot motion characteristics. Of course, in reality, robots take non-zero space, do not necessarily have precise positioning, and take time to turn. In the next section we will explore a realistic model that considers uncertainty in robot motion and turning times.

3.2 Analysis of Point Visit Frequency

In analyzing the behavior of the FOP algorithm, we utilize the following notations and definitions. In representing the polygon, we use l for the polygon length, and r for the number of robots (and the number of segments). We denote a point on a given segment by p , defined by a fraction of the length of the segment, $p \in [0, 1)$, where for the leftmost point, $p = 0$ (left is arbitrary chosen for the polygon).

DEFINITION 1 (OVERLAP FACTOR). *The overlap factor is the number of segments visited by each robot. We denote this factor by o . Note that in the case of no overlap, $o = 1$.*

When $o > 1$, not all segments are covered by an equal number of robots. We use s_i to denote the number of robots that visit a specific segment i . This creates a distinction between *edge* and *middle* segments:

DEFINITION 2 (EDGE AND MIDDLE SEGMENTS). *A segment i is called an edge segment if $o \neq s_i$, or middle segment,*

otherwise.

In Figure 1, for $o = 1$, all segments are middle segments. For $o = 2$ (second set of trajectories), segment A is covered by a single robot and is an edge segment, while the others are middle segments. For $o = 3$, segments A, B are edge segments.

To represent the motion characteristics of the robots, we use t for the time it takes robots to turn, and v to denote the robots' velocity. We assume homogeneous robots, and thus all robots turn and move at the same velocity.

We depart from earlier work in using a function $d(x)$ to account for the accumulation of errors in robot motion. Let $T(x)$ denote the time it would take a robot to pass a distance x . Under assumption of no errors, $T = \frac{x}{v}$ where v is the robot's constant velocity. However, in realistic settings, due to acceleration changes and accumulating errors in motion, the actual travel time is going to be different: $T = \frac{x}{v} + d(x)$. By choosing to represent the error in travel time directly in terms of time, we bypass modeling the different factors accounting for delays, and focus on the symptoms. Note that we assume the $d(x)$ is non decreasing function. Although in principle it is possible that a robot will travel too fast due to errors, in reality, this is rarely the case. For instance, a common source of velocity errors in research-grade robots is battery decay. This causes slowed motion, rather than acceleration.

We now analyze the visit frequency of a given point p using FOP, given the robots' motion characteristics t and d . The function $time_p(l, r, p, v, o, s, n, t, d)$ calculates the time that passes between two subsequent visits $(n - 1, n)$ to the point p in a given segment i . By minimizing this function we improve the frequency of visits to the point p . The function $time_p(l, r, p, v, o, s_i, n, t, d)$ is defined as follows.

$$\left\{ \begin{array}{ll} 2\frac{l}{r} \frac{p}{v} + t + d(o\frac{l}{r}) - d((o-p)\frac{l}{r}) + d(p\frac{l}{r}) & \text{if } n \bmod 2s_i = 1 \\ 2(1-p)\frac{l}{rv} + 2(o-s_i)\frac{l}{rv} + t + d(o\frac{l}{r}) - d((s_i+p-1)\frac{l}{r}) + d((1-p+o-s_i)\frac{l}{r}) & \text{if } n \bmod 2s_i = s_i + 1 \text{ or } s_i = 1 \\ \frac{l}{rv} + d(\frac{l}{r}([(n-1) \bmod s_i] + p)) - d(\frac{l}{r}([(n-2) \bmod s_i] + p)) & \text{otherwise} \end{array} \right. \quad (1)$$

The first condition in the formula (Eq. 1) is satisfied when the robots change direction at the left edge of a segment ($\frac{l}{r}$ is a length of a segment). The second condition is satisfied when the robots change direction at the right edge of the segment, and the third condition is satisfied in the overlapping regions.

Using Eq. 1, we can now construct the cyclic series which describes the times at which a point p is visited. The series is given in Eq. 2 below.

The first element is:

$$2\frac{l}{r} \frac{p}{v} + t + d(o\frac{l}{r}) - d((o-p)\frac{l}{r}) + d(p\frac{l}{r})$$

Then the next $s_i - 1$ elements are of the form:

$$s_i - 1 \left\{ \begin{array}{l} \frac{l}{rv} + d((1+p)\frac{l}{r}) - d(p\frac{l}{r}) \\ \frac{l}{rv} + d((2+p)\frac{l}{r}) - d((1+p)\frac{l}{r}) \\ \vdots \\ \frac{l}{rv} + d((s_i-1+p)\frac{l}{r}) - d((s_i-2+p)\frac{l}{r}) \end{array} \right.$$

Then, one element:

$$2(1-p)\frac{l}{rv} + 2(o-s_i)\frac{l}{rv} + t + d(o\frac{l}{r}) - d((s_i+p-1)\frac{l}{r}) + d((1-p+o-s_i)\frac{l}{r})$$

And finally, $s_i - 1$ elements of the form:

$$s_i - 1 \begin{cases} \frac{l}{rv} + d((1+p)\frac{l}{r}) - d(p\frac{l}{r}) \\ \frac{l}{rv} + d((2+p)\frac{l}{r}) - d((1+p)\frac{l}{r}) \\ \vdots \\ \frac{l}{rv} + d((s_i-1+p)\frac{l}{r}) - d((s_i-2+p)\frac{l}{r}) \end{cases} \quad (2)$$

The first element of the series is the result of the first condition in Eq. 1. It matches the situation where a robot returns to its base segment and turns back until it meets the point p again: $2\frac{l}{r} \frac{p}{v} + t + d(o\frac{l}{r}) - d((o-p)\frac{l}{r}) + d(p\frac{l}{r})$. It is constructed from three components: (i) The time to arrive at p , based on the distance and robot velocity, the time it takes the robot to turn and the time error function, d ; (ii) the time t it take the robot to turn around; and (iii) the error (in travel time) due to the robot motion. The first component is given by $2\frac{l}{r} \frac{p}{v}$ where $\frac{l}{r} \frac{p}{v}$ is the time it takes the robot to arrive from point p to the left edge of the segment. We multiply it by 2 since the robot needs to return from this edge to point p . The third component is given by $d(o\frac{l}{r}) - d((o-p)\frac{l}{r}) + d(p\frac{l}{r})$, which is a reduction of $d(o\frac{l}{r}) - d((o-1)\frac{l}{r} + (1-p)\frac{l}{r}) + d(p\frac{l}{r})$. This value has two parts: uncertainty from point p to the left edge and the uncertainty from the left edge back to point p . The uncertainty from point p to the left edge is equal to $d(o\frac{l}{r}) - d((o-1)\frac{l}{r} + (1-p)\frac{l}{r})$ which is the uncertainty of moving along all the (overlap) segments ($d(o\frac{l}{r})$) minus the uncertainty of moving along all the (overlap) segments until point p . The uncertainty of moving from the left edge to point p is equal to $d(p\frac{l}{r})$. Note that this model assumes that motion time errors are set to zero once a robot halts and turns.

The second element in the cyclic series (Eq. 2), corresponds to a neighboring robot that visits the point p , due to any overlap. The value is $\frac{l}{rv} + d((1+p)\frac{l}{r}) - d(p\frac{l}{r})$. It is composed of two factors: The time to arrive at the point (based on the distance and velocity), and the travel-time error function d . Since the robots are in $\frac{l}{r}$ distances from one another, the pure time it takes an adjacent robot to arrive the point p (after it has just been visited by another robot) is $\frac{l}{rv}$. The component $d((1+p)\frac{l}{r})$ comes from the adjacent robot's movement until it reaches point p . We then subtract from it $d(p\frac{l}{r})$, the error in time originating with the first robot movements as it leaves the point p behind it.

Overall, the segment i in question is visited by s_i robots, each twice (when moving left to right, and when moving right to left). The previous two paragraphs described the visit times due to the first two of these visits: The visit by original (first) robot, and a visit by an adjacent robot. Any other $s_i - 2$ robots follow the behavior of the adjacent robot, thus overall there are $s_i - 1$ elements due to adjacent robots in the first part of the series, as robots move left to right.

The other s_i elements in the cycle of series are due to the robots turning and repeating the movement, but from right to left. The next value (number $s_i + 1$) is $2(1-p)\frac{l}{rv} + 2(o-s)\frac{l}{rv} + t + d(o\frac{l}{r}) - d((s+p-1)\frac{l}{r}) + d((1-p+o-s)\frac{l}{r})$. This is a reduction of $2(1-p)\frac{l}{rv} + 2(o-s)\frac{l}{rv} + t + d(o\frac{l}{r}) - d((s-1)\frac{l}{r} + p)\frac{l}{r}) + d((1-p)\frac{l}{r} + (o-s)\frac{l}{r})$. This value is similar in form to the first value of the series.

3.3 Analysis of Patrolling Performance

In this section we analyze the performance of the FOP algorithm, under the point-visit model proposed in Section 3.2. We remind the reader of the three key performance criteria: *Maximal average frequency*, *maximal under-bounding frequency*, and *maximal uniformity*. We show how an optimal o may be computed for each of these performance criteria.

3.3.1 Optimizing average frequency

The average time to visit a point p in the i 'th segment is given the function avg_p , shown in Eq. 3. The function avg_p averages the previously shown series in Eq. 2, by relying on the function sum_p (Eq. 4) to sum the times between visits to point p in segment i .

$$avg_p(l, r, p, v, o, s_i, t, d) = \frac{1}{2s_i} sum_p(l, r, p, v, o, s_i, t, d) \quad (3)$$

$$sum_p(l, r, p, v, o, s_i, t, d) = \frac{2ol}{rv} + 2t + 2d(o\frac{l}{r}) - d(p\frac{l}{r}) - d((o-p)\frac{l}{r}) + d((s_i-1+p)\frac{l}{r}) + d((1-p+o-s_i)\frac{l}{r}) \quad (4)$$

In order to find the optimal overlap factor o that minimizes the global average along all fence, we need to summarize all visit sequence of each point in all fence segment and minimizing this value respectively to o . We do this in two stages. First, we use an integral from 0 to 1 on function sum_p , with respect to p , to sum all of the visit intervals of all points in a specific segment. We then sum all such integrals in all segments, and divide by the length of the entire polygon.

Equations Eq. 5–6 show this process. The function sum in equation Eq. 5 sums all the visit intervals of all points in all segments. Here, S is the set of all s_i . Then, the function avg (Eq. 6) divides the sum by l to get the average time between visits, over the entire length of the open polygon.

$$sum(l, r, p, v, o, S, t, d) = \sum_{i=1}^r \int_0^1 sum_p(l, r, p, v, o, s_i, t, d) dp \quad (5)$$

$$avg(l, r, p, v, o, S, t, d) = \frac{1}{l} sum(l, r, p, v, o, S, t, d) \quad (6)$$

To find the optimal o value that minimizes the avg function 6 by looking for a minimal value, e.g., by using the first and second derivative with respect to o to determine minimum points. Since in this article we did not place any restrictions on the structure of d , we refrain from doing so here. In practice, it should be done only once d is known.

3.3.2 Maximal minimum frequency

For the under-bounded frequency criteria it is easy to determine that the segment i for which $o - s_i$ is greatest, has the lowest frequency of visits to segment points. The edge where this occurs is known in advance—it is the leftmost edge segment ($i = 1$), which is left alone for long periods of time when the robot responsible for it is busy in the overlapping portions of its trajectory, and no other robots visit it. As we move right, and more robots patrol the segments, the better this measure becomes.

The worst time of visiting a point in the leftmost segment is (as appears in equation Eq. 2):

$$2(1-p)\frac{l}{rv} + 2(o-s_1)\frac{l}{rv} + t + d(o\frac{l}{r}) - d((s_1+p-1)\frac{l}{r}) + d((1-p+o-s_1)\frac{l}{r})$$

Thus if the maximal minimum criteria is important at the polygon level (i.e, across all segments), then no overlap should be used. setting $o = 1$ maximizes the minimal frequency in this case.

However, suppose we are instead seeking to examine the maximal minimum frequency per segment. In middle segments (where $o = s_i$) the lowest frequency (the greatest time between visits) of visiting a point will be the maximal value from the following options:

1. $2\frac{l}{r} \frac{p}{v} + t + d(o\frac{l}{r}) - d((o-p)\frac{l}{r}) + d(p\frac{l}{r})$
2. $max_{i=1}^{o-1} (\frac{l}{rv} + d((i+p)\frac{l}{r}) - d((i-1+p)\frac{l}{r}))$
3. $2(1-p)\frac{l}{rv} + t + d(o\frac{l}{r}) - d((o-1+p)\frac{l}{r}) + d((1-p)\frac{l}{r})$

The maximal value depends on the form of d function and the point p . We therefore leave further derivation out of this paper.

3.3.3 Maximal frequency uniformity

Following [7], we measure the uniformity of visit frequency by the standard deviation of frequency values. Lower values indicate improved uniformity, as it means that the frequency values for different points p are clustered more closely around the average frequency.

The standard deviation of visiting a point along the polygon is shown in equation 7. In order to find the o value that optimizes the uniform frequency criteria (which decreases the standard deviation) we need to differentiate equation 7 with respect to by o , and find the value the minimizes this function.

$$\sqrt{\frac{\sum_{i=1}^r (\int_0^1 (\sum_{j=1}^{2s_i} (time_p(l,r,p,v,o,s_i,j,t,d) - avg(l,r,p,o,s_i,t,d))^2) dp)}{l \sum_{i=1}^r 2s_i}} \quad (7)$$

Summary. The analysis in this section has shown how in principle it is possible to optimize the selection of the overlapping factor o to maximize performance across three frequency-based criteria. Thus given environment parameters (e.g., length of fence, number of available robots) and robot motion characteristics (turning duration t , the error function d), it is possible to determine the optimal o . Naturally, by manipulating the analysis, it might be possible to instead determine the optimal number of robots for a given overlapping factor, or the optimal turn duration for a given number of robots and overlapping factor, etc. We leave this for future work.

4. EXPERIMENTS

To evaluate the usage of the new model we conducted a series of experiments on physical robots, during which the robots are performing patrols of $o = 1$ and $o = 2$. The model discussed in this paper predicted that an overlap of one ($o = 1$) would be better for the given length of the fence. The robots were programmed by students, without knowledge of the motion model developed here. The experiment settings are discussed in Section 4.1. We recorded the point visit frequencies in extensive trials, and conduct post-hoc analysis, in which we compare the predictions of the model to the actual behavior of the robots (Section 4.2).

4.1 Experiment Settings

Our experiments utilized a team three robots, patrolling a mock fence, using Friendly robotics' RV-400 [9] vacuum cleaning robots (Figure 2). Each commercial robot was modified to be controlled

by a small Linux-running computer, sitting on top of it. A generic interface driver for the RV-400 robot was built in the Player robotics API [10], and a client program was built to control it. The robots have 8 short-range sonar sensors, pointing forward and sideways, which we utilize for maintaining distance to the mock fence. The robot interface also provides rudimentary odometry readings (coordinates and pose), which are unfortunately fairly inaccurate.



Figure 2: The RV-400 vacuum cleaner robot, with our lab's computer overriding its commercial control software.

The experiment settings consisted of a carton-box mock fence, 5.40 meters in length. The fence was divided into three equal-length segments (180cm each). Figure 3 shows a birds-eye view of the mock fence. Because the picture was taken at an angle, it is difficult to see that the three robots are equidistant from each other.

We ran two sets of experiments. In the first, the overlapping factor was set to one ($o = 1$), and in the second set, the overlapping factor was doubled ($o = 2$). The FOP algorithm—in its abstract form—was described to students carrying out course projects in the lab. Thus their implementation of the FOP algorithm is untainted by our expectations, given our own knowledge of the motion characteristics model. This is a critical point in the design of the experiment.

Independent variables. In the first set of nine patrol runs, the robots had to complete one back and forth round with an overlapping factor of 1, i.e., no overlap at all. In the second set of seven experiments (originally, nine, but two were dismissed because a weak battery caused noticeable slowdown in their movements), the



Figure 3: A snapshot from experiments.

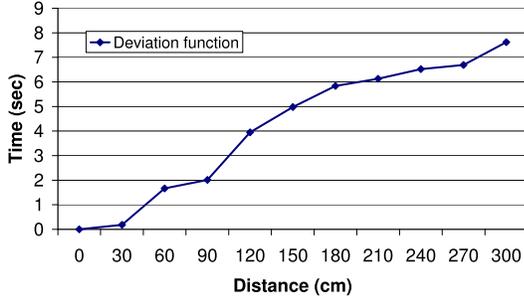


Figure 4: The deviation function $d(x)$ which measures the extra time that the robot delays when moving a distance x , as a result of uncertainty in movement and accumulating errors.

robots also had to complete one round trip, but this time with an overlapping of 2, in which the leftmost and middle robot are also responsible for the segment to their right.

Dependent variables. To measure frequency, we recorded movies of the robots patrolling the fence, and later analyzed the video recordings to determine the duration between subsequent visits to points within each segment. Four sampling points p were selected at a distance of 45cm, 225cm, 255cm, and 285cm from the left edge of the fence. The 45cm point is on the leftmost edge segment, at $p = \frac{45}{180} = \frac{1}{4}$. The other three points are respectively in the middle segment. Each patrolling run consisted of a number of visits to each point, the results (below) are averaged over approximately 30 data-points (when $o = 2$), or 20 data-points ($o = 1$).

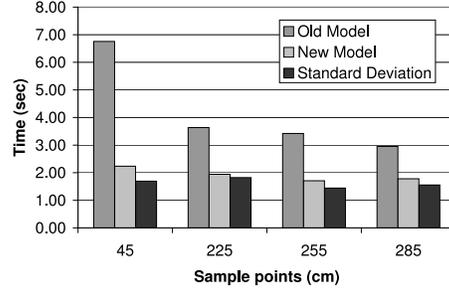
To compare the measured results, we estimated the t and d parameters for the robots, from the recorded videos. We determined that the turning duration for the robots was approximately 6 seconds. The travel time error function $d(x)$ is shown in Figure 4. It was estimated based on measurements of 10 different distances. Here, the x -axis shows the distance x , and the y -axis measures the error in travel time, compared to the predictions based on the robot velocity alone. Note the monotonically increasing error function: We remind the reader that this error function was not intentionally built in, but was discovered ad-hoc, providing empirical support for the assumptions we made earlier on regarding the monotonically-increasing nature of d .

4.2 Experiment Results

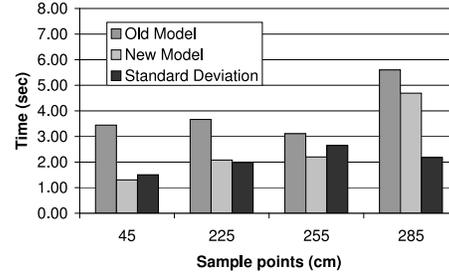
We calculated the predicted intervals of visitation according to both the new model in this paper, and the older motion model in earlier work [8]. This older model assumes robots maintain their velocity precisely, and does not allow for the deviation function d . We contrasted the predictions of both models with the empirical results observed in practice. In this comparison, adjustments were made to the predictions of both new and old models, to account for the size of the robot compared to the length of its segment (essentially, the robot size was deducted from the distance traveled). Both models assume that the robot size is insignificant relative to the fence size, which is not true in laboratory conditions.

Point-level predictions.

We begin by a direct comparison of the two models, by specifically focusing on the errors in their predictions for the sampling



(a) $o = 2$



(b) $o = 1$

Figure 5: Errors in predictions, and the sample standard deviation. Lower values are better. The new model is always better than that previously published.

points we chose earlier. In Figure 5 we present the average errors of each model with respect to the observed results, and the standard deviation for the observed results. In both sub-figures, the x -axis shows the 4 sampled points along the fence, denoted by their distance in centimeters from the left edge of the fence. In the y -axis, we present the error in seconds. The left bar shows the average error in the prediction of the previous model. The middle bar shows the error in prediction for the new model, described in Section 3.2. The right bar shows the sample standard deviation for the observed results, for comparison.

Several conclusions can be reached based on Figure 5. First, for both overlapping factor settings, the new model is clearly more accurate than the older model; both Figure 5(a) and Figure 5(b) clearly shows a substantial reduction in error in the new model (middle bar), compared to the old model (left bar).

Second, in most cases, the average errors of the new model are approximately equal to the standard deviation of the observed results. This suggests that the new model is not just *relatively accurate* (being superior to the previous model), but also *absolutely accurate*, in that the error is indistinguishable from the normal observed varying measurements.

Segment-level predictions.

We now abstract away from specific points along the fence, and turn to examine the predictions of the model at the segment level. Figure 6 contrasts the segment level average frequency, uniformity (as measured by the frequency standard deviation) and maximal minimum frequency for two segments: The leftmost segment (which, when $o = 2$ is an edge segment), and the middle segment. Each sub-figure shows two groups of bars. The left group of

bars shows the results for the leftmost (edge) segment. The right group of bars shows the results for the middle segment. Within each group, the leftmost bar shows the predictions of the old model [8], the middle bar shows the predictions of the new model, and the final bar shows the actual results (averaged over the different runs). Sub-figures should be contrasted vertically: The top row of figures show the frequency criteria measurements for $o = 1$, and the bottom row shows the measurements, for the same criteria, for $o = 2$.

Again, multiple issues are raised by the comparison. First, by contrasting predictions of the old and new models with the actual results (i.e., within each group of bars), we again see that the new model, developed in this paper, accurately predicts the observed behavior of the robots in practice. To see this, we look at the statistical significance of the differences between the predictions of the old model and the experiment results, versus the significance of the difference between the predictions of the new model and the results.

We find that generally, in terms of average frequency and uniformity of frequency, the predictions for the *old* model are statistically significantly different than the experiment results (two-tailed Z-test, $p < 0.05$). This implies that the old model does not accurately predict the experiment results. However, there is no statistically significant difference between the predictions of the new model and the results. While this does not prove they are the same (i.e., that the new model is accurate), it does lend support to this hypothesis. For the maximal minimal-frequency criteria, both the old and new model are not, in general, significantly different than the results.

Second, by examining the left and right groups of bars within each sub-figure, and contrasting them vertically. We see a clear qualitative difference between the middle and edge segments (when $o = 2$), which is not present in the case $o = 1$. This distinction between the results of the FOP algorithm in middle and edge segments, for $o > 1$ is theoretically predicted by the analytical models developed previously and in this paper. In this respect, selecting $o = 1$ may be a better choice, if one does not allow variability in patrolling frequency at the segment level.

Third, by contrasting individual groups of bars vertically, we can begin to see where there may be an advantage to the overlapping group ($o = 2$). For the middle segments, the results of the average and maximum minimum frequency are essentially the same for $o = 2$ as for $o = 1$. However, we can also see that the uniformity of the middle segments in the case of $o = 2$ is much improved compared to the case of $o = 1$ (Figures 6(c) and 6(f)).

Polyline-level predictions.

Finally, we discuss the results of the experiments in terms of the entire length of the polyline, encompassing all three segments. Based on Equation 7 above, the model predicts that for the given length (5.40cm), number of robots (3), and their motion parameters, an overlap of 1 ($o = 1$) is preferable. The results of the experiments with real robots show that the standard deviation of visits in the case of $o = 1$ is 10.945, while in the case of $o = 2$, the standard deviation is 19.745. Thus as predicted, an overlap factor one is preferable in terms of uniformity of point-visit frequency.

Note that these results are for the specified parameters. For instance, if there were 5 robots, an overlap of two ($o = 2$) would have been better. And had there been a longer polyline, the overlap could have changed as well: The longer the polyline—the greater the number of middle segments—the more advantageous it would likely be to use $o > 1$: The effects we are seeing here for the middle segment would be the same for all middle segments, however many; while there will only be a few edge segments regardless of

the length of the polyline.

5. CONCLUSIONS

Frequency-based multi-robot patrolling, in areas or on their boundaries, is an area of significant interest for defense and civilian applications. Recent previous work has begun to examine boundary patrolling of open polygons (i.e., visiting points on a polyline). Such patrolling is very challenging, because the robots must inherently backtrack over their position when they reach an endpoint, and thus uniformity of point-visit timing cannot be perfectly maintained. A skeleton algorithm for patrolling, which allows robot trajectories to overlap in space (but not in time) has been proposed, but has not been evaluated in physical robots. Instead, an analytical model of its behavior was presented in [8].

We develop a more realistic analytical treatment of the performance of the algorithm with multiple patrolling robots. We develop a realistic model of robot motion, that considers real-world uncertainties and accumulating motion errors. We mathematically analyzed the model, and then used it to predict the empirically observed behavior of robots patrolling with different overlapping factors; robots that have not been developed with the model in mind. The results of extensive experiments show that the new model is not only more accurate relative to previous models, but is also accurate on an absolute scale.

Much remains for future work. Given the inherent difficulty of maintaining uniform patrolling frequency, we are considering ways of using it to the robot's advantage. We are also interested in extending the algorithm to handle more detailed models of errors, and their causes.

Acknowledgments

Thanks to K. Ushi. for nourishment and support during the final stretch of 36 straight hours of writing this article. This research was supported in part by ISF Grant #1357/07.

6. REFERENCES

- [1] N. Agmon, V. Sadov, G. A. Kaminka, and S. Kraus. The impact of adversarial knowledge on adversarial planning in perimeter patrol. In *Proceedings of the Seventh International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS-08)*, 2008.
- [2] M. Ahmadi and P. Stone. A multi-robot system for continuous area sweeping tasks. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA-06)*, 2006.
- [3] D. Carrola, C. Nguyena, H. Everetta, and B. Frederickb. Development and testing for physical security robots. In *SPIE*, Orlando, 2005.
- [4] J. Colegrave and A. Branch. A case study of autonomous household vacuum cleaner. In *AIAA/NASA CIRFFSS*, 1994.
- [5] N. Correll and A. Martinoli. Robust Distributed Coverage using a Swarm of Miniature Robots. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA-07)*, pages 379 – 384, 2007.
- [6] N. Correll, S. Rutishauser, and A. Martinoli. Comparing Coordination Schemes for Miniature Robotic Swarms: A Case Study in Boundary Coverage of Regular Structures. In *The 10th International Symposium on Experimental Robotics (ISER)*, Springer Tracts in Advanced Robotics, to appear, 2006.

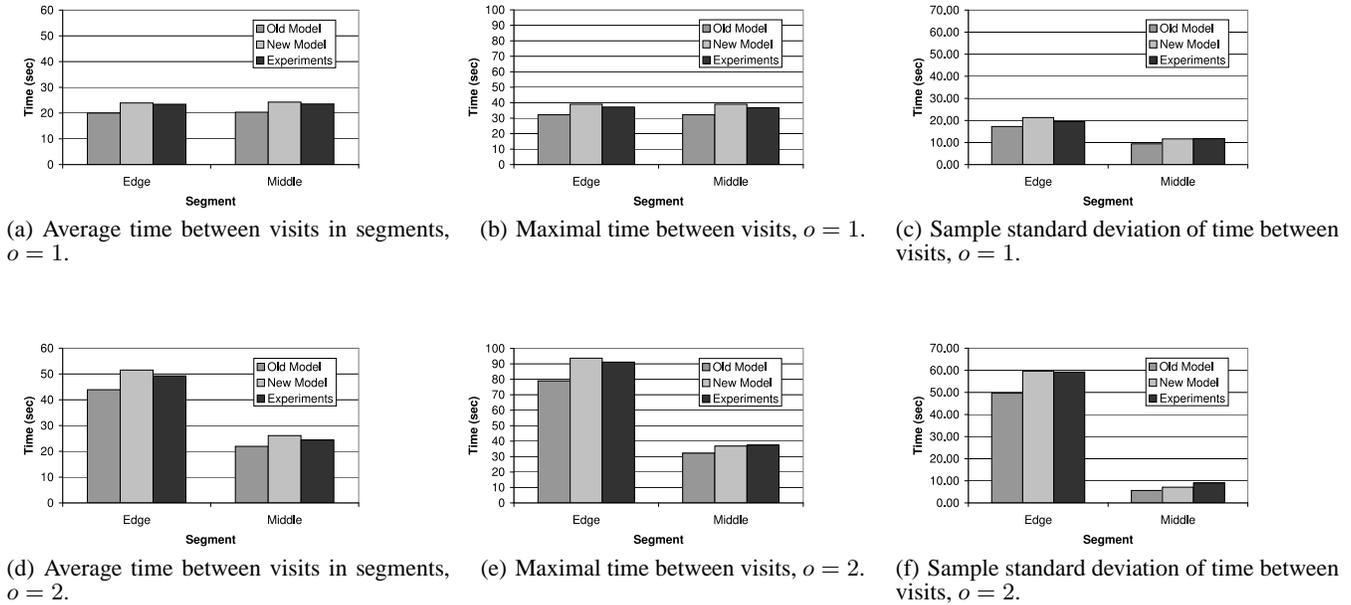


Figure 6: Frequency optimization criteria. Measurement at the segment level.

- [7] Y. Elmaliach, N. Agmon, and G. A. Kaminka. Multi-robot area patrol under frequency constraints. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA-07)*, 2007.
- [8] Y. Elmaliach, A. Shiloni, and G. A. Kaminka. Frequency-based multi-robot fence patrolling. Technical Report MAVERICK 2008/01, Bar Ilan University, Computer Science Department, MAVERICK Group, 2008.
- [9] Friendly Robotics®, Ltd. Friendly robotics vacuum cleaner. http://www.friendlyrobotics.com/friendly_vac/.
- [10] B. P. Gerkey, R. T. Vaughan, and A. Howard. The player/stage project: Tools for multi-robot and distributed sensor systems. In *Proceedings of the International Conference on Advanced Robotics*, pages 317–323, Coimbra, Portugal, Jul 2003.
- [11] A. Girard, A. Howell, and J. Hedrick. Border patrol and surveillance mission using multiple unmanned air vehicles. In *Proceedings of the 43rd IEEE Conference on Decision and Control*, pages 620–625, 2004.
- [12] Y. Guo, L. Parker, and R. Madhavan. Towards collaborative robots for infrastructure security applications. In *Proceedings of the 2004 International Symposium on Collaborative Technologies and Systems (CTS-04)*, pages 235–240, 2004.
- [13] Y. Guo and Z. Qu. Coverage control for a mobile robot patrolling a dynamic and uncertain environment. In *Proceedings of the Fifth World Congress on Intelligent Control and Automation (WCICA-04)*, volume 6, pages 4899–4903, 2004.
- [14] S. Hedberg. Robots cleaning up hazardous waste. *AI Expert*, pages 20–24, May 1995.
- [15] P. Paruchuri, J. P. Pearce, M. Tambe, F. Ordonez, and S. Kraus. An efficient heuristic approach for security against multiple adversaries. In *Proceedings of the Sixth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS-07)*, 2007.
- [16] A. Ryan, X. Xiao, S. Rathinam, J. Tisdale, M. Zennaro, D. Caveney, R. Sengupta, and K. Hedrick. A modular software infrastructure for distributed control of collaborating UAVs. In *Proceedings of the AIAA Conference on Guidance, Navigation, and Control*, Keystone, 2006.
- [17] K. Williams and J. Burdick. Multi-robot boundary coverage with plan revision. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA-06)*, 2006.