Robust Area Coverage using Hybrid Control *

Manuel Mazo Jr.[†] Karl Henrik Johansson[‡]

Abstract

Efficient coverage of an area by a mobile vehicle is a common challenge in many applications. Examples include automatic lawn mowers and vacuum cleaning robots. In this paper a vehicle with uncertain heading is studied. Five control strategies based on position measurements available only when the vehicle intersects the boundary of the area are compared. It is shown that the performance depends heavily on the heading error. The results are evaluated through extensive Monte Carlo simulations. An experimental implementation on a mobile robot is also presented.

1 Introduction

Mine detecting robots, search-and-rescue missions, and snow removing vehicles are applications in which it is important to efficiently cover a given area. Recent commercial implementations in consumer products include automatic vacuum cleaners [1] and automatic lawn mowers [2]. Several solutions to the area coverage problem are proposed in the literature, see Choset [3] for a recent survey. Many existing algorithms consider the decomposition problem, in which the main task is to find intelligent ways to decompose a given large irregular area into pieces easily covered by a default coverage path. An example of such an algorithm is proposed by Hert and Lumesky [4]. There also exist heuristic coverage methods, for example, behavior-based algorithms with one or more robots [5, 6]. The efficiency of an algorithm can be measured through the time it takes to complete the coverage. In the work of Huang et al. [7], it is argued that a reasonable optimization criterion is the total number of turns needed for a complete coverage. This is based on the natural assumptions that for mobile robots and other vehicles, turns are costly due to the need to decelerate, turn, and accelerate. It seems like actuator and sensor errors have not been considered in the area coverage literature, though they play an important role for the performance in many applications.

The main contribution of this paper is to introduce an area coverage problem that has an uncertain and dynamic vehicle model. Based on this model and the assumption that position measurements are only available at the boundary of the area to be covered, five control strategies are analyzed through extensive simulations and experiments. The setup is quite realistic, for example, for mobile robots which might suffer from unreliable position readings and limited sensor capacity. We consider the problem of minimizing the total number of turns needed to cover a given area, cf., [7]. The control strategies are evaluated by comparing this number for various system uncertainties. It is shown that for large uncertainties, a randomized strategy is the best one, which seems intuitive since the system state does not reveal much information in that case. For small uncertainties, a heuristic strategy sweeping the area by a simple back-and-forth motion is sufficient. The interesting case, however, is for uncertainties of middle range. We present three robust control algorithms that then outperforms the randomized and the heuristic strategies. The computational tools are mainly based on computational geometry software [8, 9]. The complexity of the coverage algorithms is not studied in the paper. In general, one can probably say, however, that the presented solutions do not scale well. A complexity analysis of the algorithms used in the geometrical operations can be found in [10]. In this context, we also remind

*This work was supported by European Commission and Swedish Research Council.

[†]Department of Signals, Sensors & Systems, Royal Institute of Technology, Stockholm, Sweden, manuelme@ieee.org

[‡]Department of Signals, Sensors & Systems, Royal Institute of Technology, Stockholm, Sweden, kallej@s3.kth.se. Corresponding author.

of the art gallery problem, which is a somehow related coverage problem that has been extensively studied also regarding algorithmic complexity [11].

It is interesting to notice that the proposed closed-loop control system for the area coverage can be modeled as a hybrid automata. In this way, it is possible to have a low-order model that still can capture the complexity of the problem. The efficiency to use hybrid control in robotics is also illustrated in time-optimal tracking control problems for Dubin's vehicle [12].

The outline of the paper is as follows. The area coverage problem is formulated in Section 2. Five control strategies for solving the problem is presented in Section 3. In Section 4 it is shown by extensive Monte Carlo simulations that the preferable control strategy depends on the error bound of the steering actuator. Experimental results are presented in Section 5, where an implementation on a mobile robot is shown. Section 6 concludes the paper.

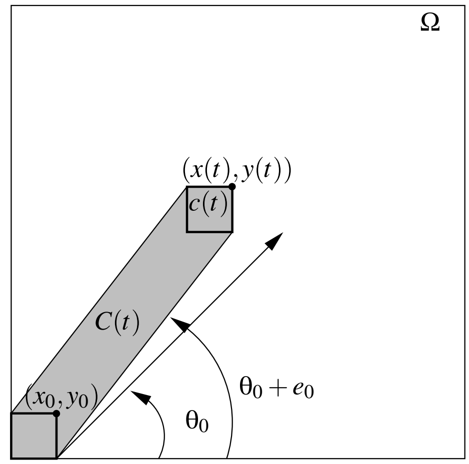

Figure 1: Area coverage problem. A square vehicle with uncertain dynamics should cover the area of Ω as fast as possible.

2 Problem Formulation

Consider the problem of covering the set

$$\Omega = [0,L] \times [0,L] \subset \mathbb{R}^2, \quad L > 1,$$

by a square vehicle, as illustrated in Figure 1. The vehicle covers a unit square, which is positioned with its upper-right corner at coordinate (x,y). For simplicity, we assume that the vehicle starts in the lower-left corner $(x(0),y(0)) = (1,1)$. At time $t \geq 0$, the vehicle covers the set

$$c(t) = [x(t)-1, x(t)] \times [y(t)-1, y(t)].$$

The accumulated covered set is denoted

$$C(t) = \bigcup_{s \in [0,t]} c(s).$$

Assuming a dynamical model of the vehicle of the unicycle type

$$\begin{aligned} \dot{x}(t) &= \cos(\theta(t)) \\ \dot{y}(t) &= \sin(\theta(t)) \\ \dot{\theta}(t) &= \omega(t) \\ \dot{v}(t) &= F/m \\ \dot{\omega}(t) &= \tau/J, \end{aligned} \tag{1}$$

where the force over mass, F/m, and torque over inertia momentum, τ/J are the input signals. Now assume each independently actuated wheel applies a force F_1, F_2 against the ground. The force and torque can then be expressed as

$$\begin{pmatrix} F \\ \tau \end{pmatrix} = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} \\ -\frac{2}{l} & \frac{2}{l} \end{pmatrix} \begin{pmatrix} F_1 \\ F_2 \end{pmatrix} \tag{2}$$

Consider two cases: the robot moving straight in the heading direction θ constant unit velocity of the vehicle, its dynamics is given by

$$\begin{aligned} \dot{x}(t) &= \cos(\theta(t) + e(t)) \\ \dot{y}(t) &= \sin(\theta(t) + e(t)), \end{aligned} \tag{3}$$

where $\theta \in [-\pi, \pi)$ is the controlled heading and e an unknown angular error. The error, which thus affects the actuation of the control, is bounded by a known constant $\varepsilon \in [0, \pi)$.

The vehicle localization is constrained, such that the vehicle position is known only at moments when the vehicle hits the boundary of Ω, i.e., for $t > 0$ such that $c(t) \cap \partial\Omega \neq \emptyset$. This can be implemented in practice by marking the boundary in a suitable way; compare current systems used for automatic cleaning robots [1] and lawn movers [2]. The control strategies studied in the paper are limited to piecewise constant controls triggered by the events $c(t) \cap \partial\Omega \neq \emptyset$, which corresponds to moments when the vehicle turns. Also the error e is piecewise constant and should be interpreted as the uncertainty in the actuation of the turning angle. We suppose that the turning events are separated in time and denoted $0 = t_0 < t_1 < \ldots$. The control θ at turn k is denoted θ_k and the corresponding error e_k. We suppose that $\theta_k + e_k$ never drives the vehicle outside Ω, i.e., for all $t \geq 0$ we have $c(t) \subset \Omega$.

In order to efficiently cover the area of Ω, denoted $A(\Omega) = \int_\Omega dz$, it is reasonable to try to minimize the total number of turns $N > 1$ made by the vehicle to complete the coverage, cf. [7, 3]. The feedback controls θ_k, $k = 0, 1, \ldots, N$, can be written as

$$\theta_k = f(x_k, y_k, C_k),$$

where $(x_k, y_k) = (x(t_k), y(t_k))$ is the position at the turning point and $C_k = C(t_k)$ is the total covered set up till time t_k.

The closed-loop system can be described as the hybrid automaton in Figure 2. When the guard condition $c(t) \cap \partial\Omega \neq \emptyset$ is fulfilled, a discrete-event is generated. It updates the control θ and the error e according to the indicated reset maps. A control law f that solves the coverage problem in N turns corresponds to a family of hybrid trajectories, which each consists of N straight lines. A hybrid automaton describing the area coverage control problem. When the guard condition $c \cap \partial\Omega \neq \emptyset$ is enabled (i.e., the vehicle coverage intersects the boundary of Ω), a discrete event takes place. At the event, the control θ_k and error e_k are updated according to a control law f and an error set $[-\varepsilon, \varepsilon]$, respectively.

An interesting hybrid differential game problem is to find a feedback control law f that minimizes N, given hard constraint on the error $|e_k| < \varepsilon$. The authors are not aware of a general solution to this robust control problem. Instead we present a few intuitive algorithms in next section, and the rest of the paper is devoted to their evaluations.

3 Area Coverage Control Strategies

Five feedback control strategies for area coverage are presented in this section. They are denoted *nominal*, *guaranteed*, *possible*, *heuristic*, and *randomized*. The first three of them are "greedy" in the sense that they try to maximize the area covered between two turns given different constraints. The fourth strategy is heuristic and basically mimics a traditional way of covering an area when there are no actuator errors; a boustrophedon path [3, 13]. The fifth control strategy is a randomized solution, which is inspired by commercial implementations in automatic vacuum cleaners and lawn movers [1, 2].

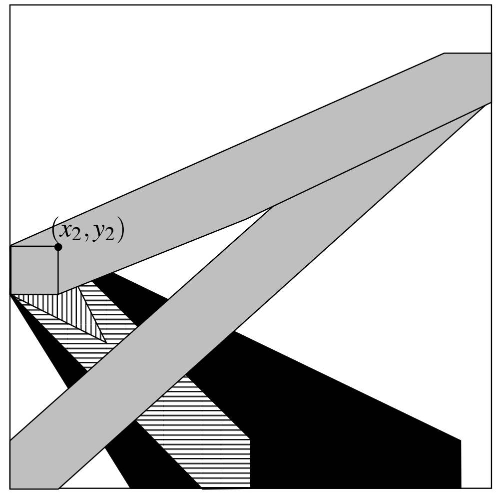

Figure 2: Comparison of area coverage control algorithms at $t = t_2$.

Figure 2 shows a comparison of how the first three control strategies are derived. The snapshot is taken at turn $k = 2$. The current coverage $c(t_2)$ of the vehicle is marked by a small square. The gray area corresponds to the accumulated coverage $C(t_2)$. At this moment, the hybrid control strategies maximize the area to be covered till turn $k = 3$, i.e., search for the best control θ_2 over the interval $[-\pi, \pi)$. Figure 2 shows areas for $\theta_2 = -\pi/4$. How these are derived is further described below.

3.1 Nominal Control

The *nominal* control strategy maximizes the new area covered by the vehicle between turn k and $k+1$ neglecting the influence of the error. The feedback control is given by

$$\theta_k^{\mathrm{N}} = \arg \max_{\theta \in [-\pi, \pi)} A(B(x_k, y_k, \theta) \cup C_k),$$

where

$$B(x,y,\theta) = \bigcup_{z \in \ell(x,y,\theta):\, \overline{c}(z) \subset \Omega} \overline{c}(z)$$

denotes the set to be covered till next turn if the error was zero. Here $\overline{c}(z)$ denotes the set covered by a vehicle in position $z \in \Omega$ and $\ell : \mathbb{R}^2 \times [-\pi, \pi) \mapsto \mathbb{R}^2$ is the line $\ell(x,y,\theta) = \{(x + s\cos\theta, y + s\sin\theta) : s \geq 0\}$. The union of the striped areas corresponds $B(x_2, y_2, -\pi/4)$ in Figure 2.

3.2 Guaranteed Control

The *guaranteed* control strategy maximizes the new area that is guaranteed to be covered by the vehicle between turn k and $k+1$. This feedback control law is given by

$$\theta_k^{\mathrm{G}} = \arg \max_{\theta \in [-\pi, \pi)} A(B_\cap(x_k, y_k, \theta) \cup C_k),$$

where

$$B_\cap(x,y,\theta) = \bigcap_{\alpha \in [-\varepsilon, \varepsilon]} B(x,y,\theta + \alpha)$$

denotes the set guaranteed to be covered regardless of the actual error executed at t_k. In Figure 2, the vertically striped area corresponds to $B_\cap(x_2, y_2, -\pi/4)$. Note that this algorithm will not work for large ε. In that case, when a sufficiently large part of the area has been covered at time t_k, say, the guaranteed new area to cover is equal to zero, i.e., $A(B_\cap(x_k, y_k, \theta_k) \cup C_k) = A(C_k)$. (When this happens in our implementation, a random control action is issued.)

3.3 Possible Control

The *possible* control strategy maximizes the area that corresponds to the nominal control but evaluated over the union of all possible errors less than ε. This feedback control law is given by

$$\theta_k^{\mathrm{P}} = \arg \max_{\theta \in [-\pi, \pi)} A(B_\cup(x_k, y_k, \theta) \cup C_k),$$

where

$$B_\cup(x,y,\theta) = \bigcup_{\alpha \in [-\varepsilon, \varepsilon]} B(x,y,\theta + \alpha).$$

The union of the black and the stripped areas in Figure 2 corresponds to $B_\cup(x_2, y_2, -\pi/4)$.

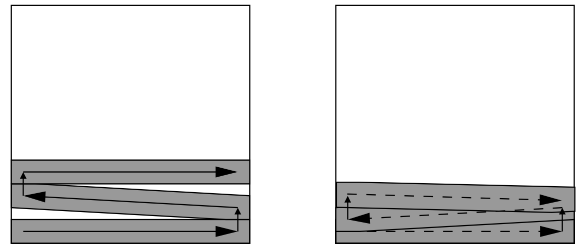

Figure 3: The left picture shows that a boustrophedon path does not succeed to cover the set Ω, when there is a non-zero steering error ε. The proposed heuristic strategy, however, performs conservative movements to guarantee complete coverage, as shown in the right picture.

3.4 Heuristic Control

The *heuristic* control strategy mimics a boustrophedon path, which is the simple back-and-forth motion an ox follows when dragging a plow in a field [3]. The only difference here is that the heuristic control θ_k^{H} is choosing conservatively, so that $C(t)$ is guaranteed to be a connected set for all $t \geq 0$, see Figure 3. Note that a pure boustrophedon strategy, without the error compensation, might not succeed in covering the whole set Ω. For small ε, the heuristic control strategy is efficient in the sense that N is close to optimal. When ε grows, however, the strategy rapidly deteriorates. For ε larger than $\varepsilon_c = 2^{-1} \arctan L^{-1}$ it happens that the path makes a closed orbit, which thus does not contribute to the area coverage.

3.5 Randomized Control

The *randomized* control strategy is simply to let θ_k^{R} take a random value from the uniform distribution $\mathcal{U}(-\pi, \pi)$. This algorithm is easy to implement, since no state information, such as current position or covered area, is needed. The Electrolux automatic cleaning robot Trilobite [1] and the Husqvarna automatic lawn mover Solar Mover [2] apply similar randomized navigation schemes.

3.6 Computational Implementation

The greedy control algorithms require the calculation of the area covered by the polygons generated from the ve-

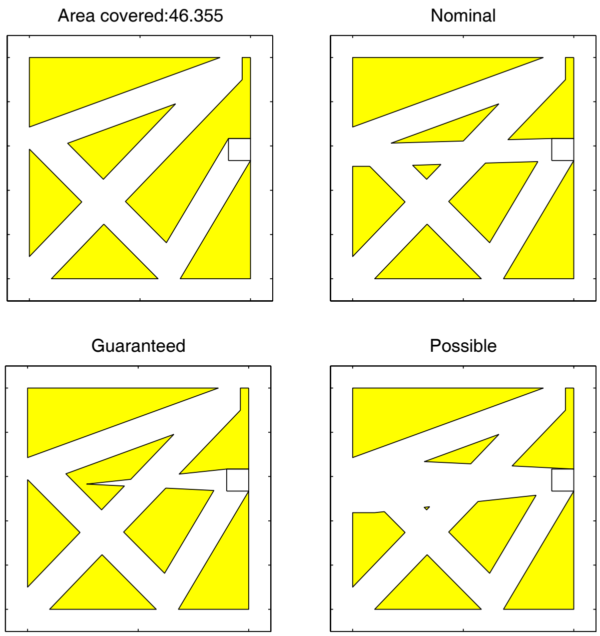

Figure 4: Snapshot of an area coverage simulation at turn $k=4$. About half of the area is covered, as indicated by the white part of Ω. The plots illustrates the nominal, guaranteed, and possible control strategies.

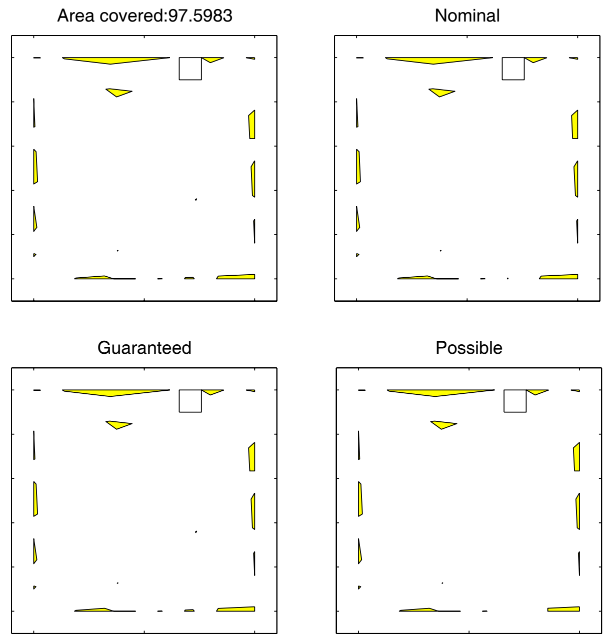

Figure 5: Snapshot of an area coverage simulation after $k=25$ turns. Almost all of the area is covered.

hicle movements. These algorithms are implemented in Matlab. They are based on Vatti's algorithms for polygon clipping [8] as implemented by Murta in the General Polygon Clipping Library [9]. Functions for area calculation and convex hull generation by Pankratov [14] are also used.

4 Simulation Results

To evaluate the area coverage control strategies, the results from Monte Carlo simulations are presented in this section. The size of the set Ω to be covered is set to $L=10$. The turning error e_k, $k=0,\ldots,N$, is drawn from a uniform distribution $\mathcal{U}(-\varepsilon,\varepsilon)$ (except for the last part of the section, where a comparison with normally distributed errors is made).

Figure 4 shows a snapshot of a simulation with $\varepsilon=0.078$ at $t=t_4$. The upper left plot shows the accumulated covered set $C(t_4)$ in white, with the current coverage $c(t_4)$ marked by a small square. At this stage the covered area is equal to $A(C(t_4))\approx 46\,\%$. Recall that the vehicle starts in the lower left corner $(x(0),y(0))=(1,1)$. Note that the error e_0 leads to that the vehicle is not able to steer exactly to the upper right corner. The upper right plot indicates the estimation for the nominal control, while the lower left and the lower right shows the guaranteed and possible controls, respectively. Figure 5 shows a snapshot of this same simulation at $t=t_{25}$. At this much later state of the simulation almost all of Ω is covered, namely, $A(\Omega)\approx 98\,\%$.

An extensive simulation comparison of the five area cover control strategies is shown in Figure 6. For each value of the error bound ε marked in the figure, one hundred Monte Carlo simulations were done and the average N was derived for $98\,\%$ coverage. The ε-axis can roughly be divided into four regions. For small errors ($\varepsilon<0.07$), the heuristic control strategy gives the best result. Note

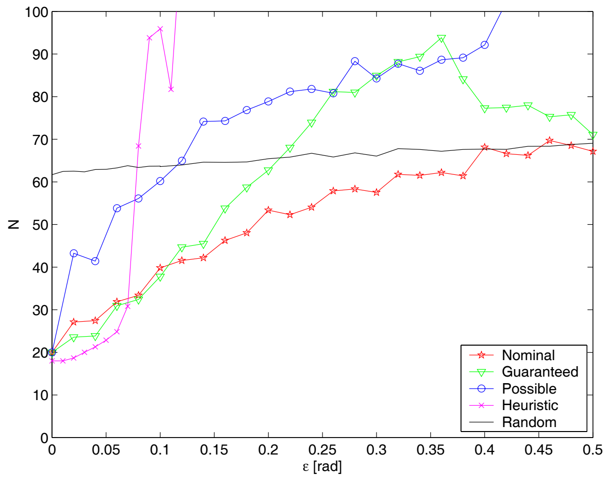

Figure 6: The average number of turns N required for 98 % coverage versus error bound ε. Five control strategies are compared. Each mark corresponds to one hundred Monte Carlo simulations. The control strategy that gives the best performance depends on ε.

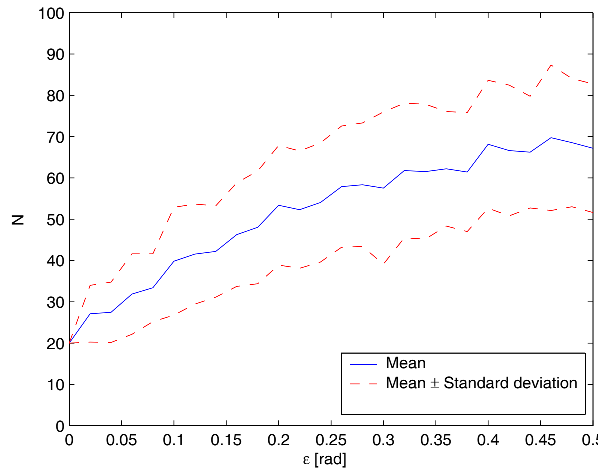

Figure 7: Similar simulations as in Figure 6 for the nominal control strategy. The dashed curves indicate the standard deviation.

that for $\varepsilon = 0$, it gives $N = 18$, which is the optimal. For $\varepsilon > 0.07$, the strategy shows quickly bad performance. This is related to the parameter $\varepsilon_c = 0.05$, see Section 3. For $\varepsilon \in (0.07, 0.10)$, the nominal and the guaranteed control strategies are equally good. Then for $\varepsilon \in (0.1, 0.4)$, the nominal control is the best. For large errors ($\varepsilon > 0.4$), the randomized strategy perform similarly, which is natural because the worst-case error is then larger than 23 degrees. For a given vehicle model, Figure 6 indicates hence preferable choices of feedback controls. Though it should be emphasized that the implementation complexity varies for the different control strategies.

The nominal control strategy shows a quite good performance over a large range of error bounds. Figure 7 shows the same result as in Figure 6 for this algorithm, but includes the standard deviations.

It is interesting to see how influential the error distribution is on the results. Figure 8 shows a comparison between errors e_k from the uniform distribution $\mathcal{U}(-\varepsilon, \varepsilon)$ (marked with rings) and errors from the normal distribution $\mathcal{N}(0, \varepsilon/\sqrt{3})$ (asterisks). The distributions thus have the same means and standard deviations. As expected, the normal distribution yields a slightly lower N, but still the results are comparable.

5 Experimental Results

The experimental setup is based on the Khepera II mobile robot, see Figure 9. The diameter of the Khepera robot is 55 mm and the area to be covered has $L = 550$, so the experimental setup and the simulation study have roughly the same quota $A(c)/A(\Omega)$. A camera and image processing software are used for the localization of the robot at the boundary of Ω. From modeling experiments, the error bound for the Khepera II robot was determined to be $\varepsilon = 0.078$,

As illustrated by the snapshot in Figure 10, the experiment follows the behavior quite well of the corresponding simulations (Figure 4). When running an experiment for a long time, it has been noticed however that the error model used in the simulation is not accurate. The error distribution tends to change over time. This is particularly the case if the localization error at the boundary of Ω is not negligible.

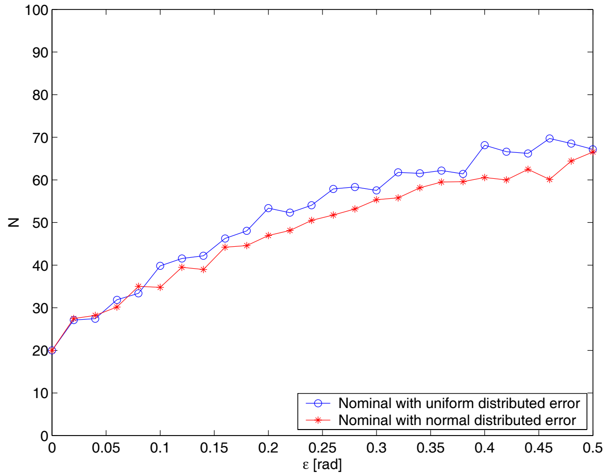

Figure 8: Similar simulations as in Figure 6 for the nominal control strategy. The rings indicate the results for uniformly distributed errors, while the asterisks indicate normally distributed errors.

6 Conclusions

Motivated by the need for robust control algorithms for area coverage under uncertain vehicle models, we presented and analyzed a few possible strategies. It was shown that the number of turns needed in order to cover an area is increasing with the error bound ε of the turns. Moreover, which algorithm that performed the best depends on ε. For example, for a bad steering actuator (large ε), a randomized algorithm performed as well as the more intelligent ones, while for a better actuator considerable improvements can be achieved by using the proposed robust strategies.

The closed-loop control system for the area coverage was presented as a hybrid automata. In this way, it was possible to have a low-order model that still can capture the complexity of the problem. It would be interesting to apply existing verification tools in order to analyze this so called timed automata [15], which the area coverage problem in the paper led to. Another possible extension of the work is to consider collaborating vehicles.

Figure 9: Experimental setup for evaluation of the area coverage robot.

References

[1] Electrolux, "Trilobite", http://www.electrolux.com/trilobite.

[2] Husqvarna AB, "Husqvarna solar mower", April 1998, http://www.solarmower.com.

[3] H. Choset, "Coverage for robotics - a survey of recent results", *Annals of Mathematics and Artificial Intelligence*, , no. 31, pp. 113–126, 2001.

[4] S. Hert, S. Tiwari, and V.Lumelsky, "A terrain-covering algorithm for an auv", *Autonom. Robots*, vol. 3, pp. 91–119, 1996.

[5] T. Balch and R. C. Arkin, "Communication in reactive multiangent robotic system", *Autonom. Robots*, vol. 1, no. 1, 1995.

[6] D. MacKenzie and T. Balch, "Making a clean sweep: Behaviour based vacuuming", in *AAAI Fall Symposium*, 1996.

[7] Y.Y. Huang, Z.L. Cao, and E.L. Hall, "Region filling operations for mobile robot using computer graphics", in *Proceedings of the IEEE Conference on Robotics ans Automation*, 1986, pp. 1607–1614.

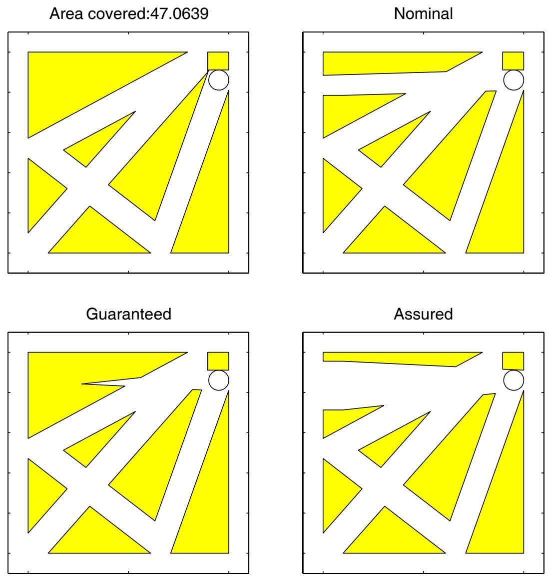

Figure 10: Snapshot of an area coverage experiment at turn $k = 4$. The experimental results agrees well with the simulations.

[8] B. R. Vatti, "A generic solution to polygon clipping", *Communications of the ACM*, vol. 35, no. 7, pp. 56–63, July 1992.

[9] A. Murta, "A general polygon library", Jan 2003, http://www.cs.man.ac.uk/aig/staff/alan/software/.

[10] G. Greiner and K. Hormann, "Efficient clipping of arbitrary polygons", *ACM Transactions on Graphics*, vol. 17, no. 2, pp. 71–83, April 1998.

[11] H.H. Gonzalez-Banos and J.C. Latombe, "A randomized art-gallery algorithm for sensor placement", in *Proc. 17th ACM Symp. on Computational Geometry*. ACM, 2001, pp. 232–240.

[12] A. Balluchi and P. Souères, "Optimal feedback control of Dubins' car tracking circular reference paths", in *Proc. 35th IEEE Conference on Decision and Control*, Kobe, Japan, December 1996, pp. 3558–3563.

[13] H. Choset and P. Pignon, "Coverage path planning: The boustrophedon decomposition", in *Proceedings of the International Conference on Field and Service Robotics, Camberra, Australia*. IEEE, December 1997.

[14] K. K. Pankratov, "Saga matlab toolbox", 1995, http://puddle.mit.edu/ glenn/kirill/saga.html.

[15] R. Alur and D. L. Dill, "A theory of timed automata", *Theoretical Computer Science*, vol. 126, no. 2, pp. 183–235, 1994.