# Planning with Uncertainty in Position: an Optimal Planner

Juan Pablo Gonzalez
Anthony (Tony) Stentz

CMU-RI -TR-04-63

The Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

October 2004

The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies or endorsements, either expressed or implied, of Carnegie Mellon University.

# Abstract

We propose a resolution-optimal planner that considers uncertainty while optimizing any monotonic objective function such as mobility cost, risk, energy expended, etc. The resulting path is a one that minimizes the expected cost value of the objective function, while ensuring that the uncertainty in the position of the robot does not compromise the safety of the robot or the reachability of the goal.

***Keywords: mobile robot, path planning, uncertainty, optimal planner, error propagation***

# Contents

# 1 Introduction

Uncertainty is always present in the position estimation of mobile robots. For path planning, however, its effects are frequently ignored. For most problems, this is an appropriate approach if the uncertainty is smaller or comparable to the size of the robot. However, when the uncertainty is larger, its effects should not be ignored.

The widespread use of GPS (Global Positioning System) has facilitated the position estimation problem for outdoor mobile robots. Nonetheless, there are many scenarios where GPS coverage is limited or unavailable: under tree canopies, in canyons, and in many urban environments. In indoor settings GPS is unavailable, although sometimes landmarks can be used for absolute positioning.

In this paper we propose a resolution-optimal path planner that considers uncertainty, while optimizing any monotonic objective function such as mobility cost, risk, energy expended, etc. The environment is represented as a grid, in which the cost of each cell corresponds to the cost of traveling from the center of the cell to its edge. The resulting path minimizes the expected value of the objective function along the path, while ensuring that the uncertainty in the position of the robot does not compromise the safety of the robot or the reachability of the goal.

The planner uses a single-parameter error propagation model in order to minimize the number of dimensions required. This model is a conservative estimate of the true error projection model and can provide an accurate estimate of the higher-dimension model, depending on the type of error that is predominant in the system. Additionally, the planner is able to handle regions with GPS coverage, where the uncertainty stops propagating and is reduced to a fixed value.

Even though the planner utilized is a 3-D planner, the characteristics of the search space and the simplified error propagation model allow the planner to have a time complexity close to that of a 2-D planner. The planning time is under 1 second for worlds of up to 150x150 cells, and under 10 seconds for worlds of up to 250x250 cells.

This paper is organized as follows: section 2 contains a review the relevant literature in the area of planning with uncertainty. Section 3 explains the motion model and the uncertainty propagation problem. Section 4 explains our approach to the problem. Section 5 shows some results of the planner applied to synthetic and geographic data and analyzes the performance of the algorithm. Section 6 contains the conclusions and explores future directions for this research.

# 2 Related Work

There is abundant work in planning with uncertainty in the research literature. Latombe [1][2] has an extensive review on the state of the art as of 1991. Since then, important contributions by Lazanas and Latombe [3], Bouilly [4][5], Haït et al. [6], Fraichard [7], and others have not only expanded the theoretical approaches to planning with uncertainty, but also addressed some of its practical limitations.

There is, however, little work in creating optimal planners. Although the planner proposed by Bouilly [4] calculates an optimal path with respect to uncertainty or path length (i.e.: a path of minimum uncertainty or minimum path length), the approach is not applicable to finding optimal paths with respect to other important criteria.

To the best of our knowledge, only Roy and Thrun [8] have solved the problem of finding optimal paths for continuous cost worlds in the presence of uncertainty. The planner they propose includes many of the elements of the planner proposed here, but is based on an approximate solution to a POMDP. This approach requires pre-processing all the states in the search space, which later allows for very fast planning. However, the total planning time (including the pre-processing stage) is very long. We are presenting an alternate semi-deterministic approach based on A*, which in average has much lower time complexity.

# 3 Motion Model and Uncertainty Propagation

The first-order motion model for a point-sized robot moving in two dimensions is:

$$\begin{aligned} \dot{x}(t) &= v(t)\cos\theta(t) \\ \dot{y}(t) &= v(t)\sin\theta(t) \\ \dot{\theta}(t) &= \omega(t) \end{aligned} \tag{1}$$

where the state of the robot is represented by $x$(t), y(t) and $\theta$(t) (x-position, y-position and heading respectively), and the inputs to the model are represented by v(t) and $\omega$(t) (longitudinal speed and rate of change for the heading respectively). Equation (1) can also be expressed as:

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{u}(t)) \tag{2}$$

where $\mathbf{x}(t) = (x(t), y(t), \theta(t))$ and $\mathbf{u}(t) = (v(t), \omega(t))$.

If we allow for error terms in (1), the resulting equation is:

$$\begin{aligned} \dot{x}(t) &= (v(t) + w_v(t))\cos\theta(t) + w_{\dot{x}}(t) \\ \dot{y}(t) &= (v(t) + w_v(t))\sin\theta(t) + w_{\dot{y}}(t) \\ \dot{\theta}(t) &= \omega(t) + w_\omega(t) \end{aligned} \tag{3}$$

where $w_v$ and $w_\omega$ are random errors in the inputs $v$(t) and $\omega$(t) (longitudinal control error and gyro drift), and $w_{\dot{x}}$ and $w_{\dot{y}}$ are random model errors.

Using the extended Kalman filter (EKF) analysis for this system, which assumes that the random errors are zero-mean Gaussian distributions, we can model the error propagation as follows:

$$\mathbf{P}(k+1) = \mathbf{A} \cdot \mathbf{P}(k) \cdot \mathbf{A}^T + \mathbf{L} \cdot \mathbf{Q}(k) \cdot \mathbf{L}^T \tag{4}$$

where

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & -v(k\Delta t)\sin(\theta(k)) \cdot \Delta t \\ 0 & 1 & v(k\Delta t)\cos(\theta(k)) \cdot \Delta t \\ 0 & 0 & 1 \end{pmatrix} \tag{5}$$

$$\mathbf{L} = \begin{pmatrix} \cos(\theta(k)) \cdot \Delta t & 0 & \Delta t & 0 \\ \sin(\theta(k)) \cdot \Delta t & 0 & 0 & \Delta t \\ 0 & \Delta t & 0 & 0 \end{pmatrix} \tag{6}$$

and

$$\mathbf{w} = [w_v(k) \quad w_\omega(k) \quad w_{\dot{x}}(k) \quad w_{\dot{y}}(k)]^T \tag{7}$$

Kelly [10] calculated closed-form solutions for some trajectories. For a straight trajectory along the x-axis keeping all inputs constant, the errors behave as follows:

- The error due to the longitudinal control $w_v$ is reflected in the x direction, and is given by $\sigma_x^2 = \sigma_v^2 \cdot vt$, or $\sigma_x = \sigma_v \sqrt{vt}$ (See Figure 1).

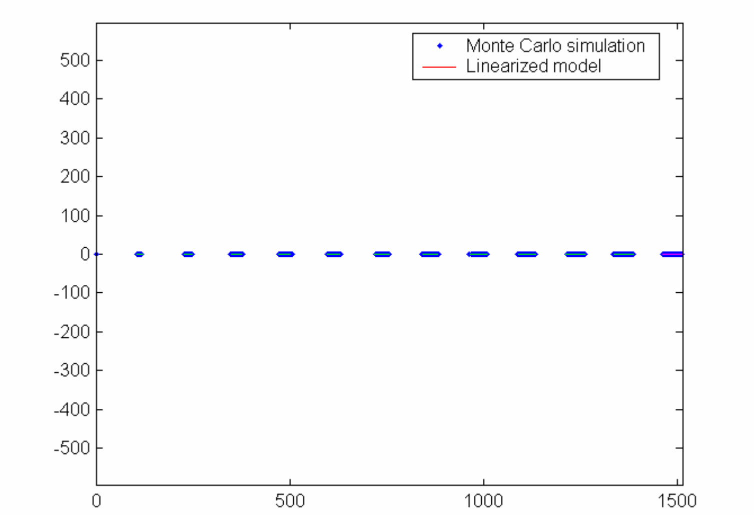

Figure 1. Uncertainty propagation for error in longitudinal control

- The error due to $w_\omega$ is reflected in the y direction, and is given by $\sigma_y^2 = \sigma_\omega^2 \cdot v^2 t^3$, or $\sigma_y = \sigma_\omega \cdot v \cdot t^{3/2}$ (See Figure 2).

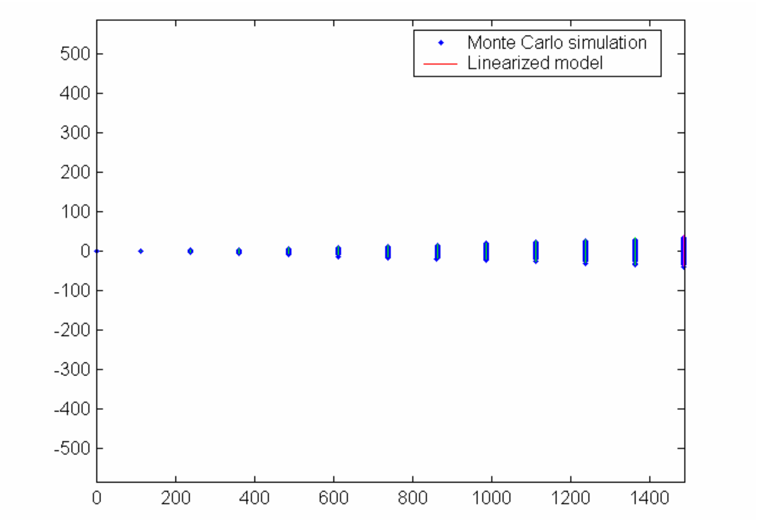

Figure 2. Uncertainty propagation for error due to gyro drift

- The error due to $w_{\dot{x}}$ is reflected in the x direction, and is given by $\sigma_x^2 = \sigma_{\dot{x}}^2 \cdot t$, or $\sigma_x = \sigma_{\dot{x}} \sqrt{t}$ (See Figure 3).

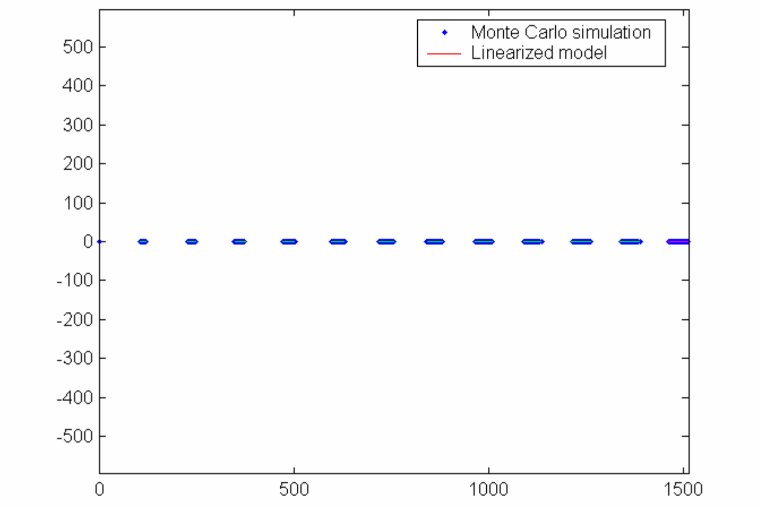

Figure 3. Uncertainty propagation for errors due to $w_{\dot{x}}$

- The error due to $w_{\dot{y}}$ is reflected in the y direction, and is given by $\sigma_x^2 = \sigma_{\dot{x}}^2 \cdot t$ , or $\sigma_x = \sigma_{\dot{x}} \sqrt{t}$ (See Figure 4).

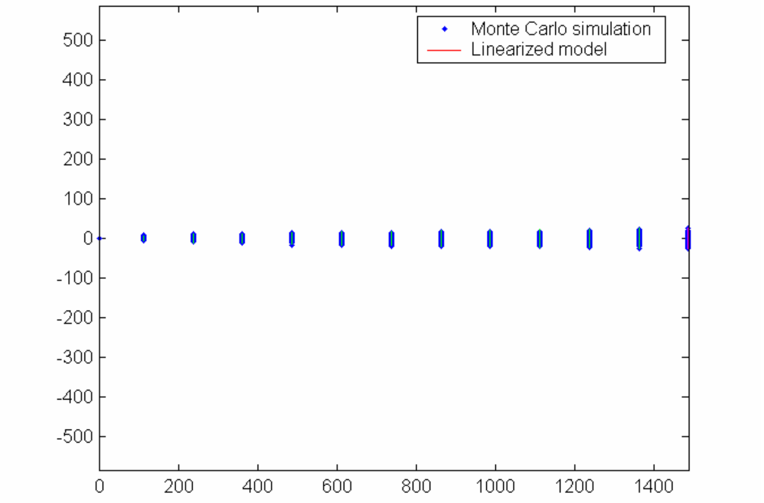

Figure 4. Uncertainty propagation for errors due to $w_{\dot{y}}$

Additionally, there are errors in the initial position of the robot. Errors in x and y do not increase unless there is uncertainty in the model or in the controls. Errors in the heading angle, however, do propagate linearly with *t*. For initial angle errors of less than 15 degrees, the small angle approximation can be used to obtain the expression: $\sigma_y = \sigma_{\theta_o} .vt$ (See Figure 5 and Figure 6).

A straight line trajectory maximizes each one of the error terms (see [10] for the proof) therefore we can use the results of this trajectory as an upper bound on the error for any trajectory. The dominant terms in the error propagation model depend on the navigation system and on the planning horizon for the robot.

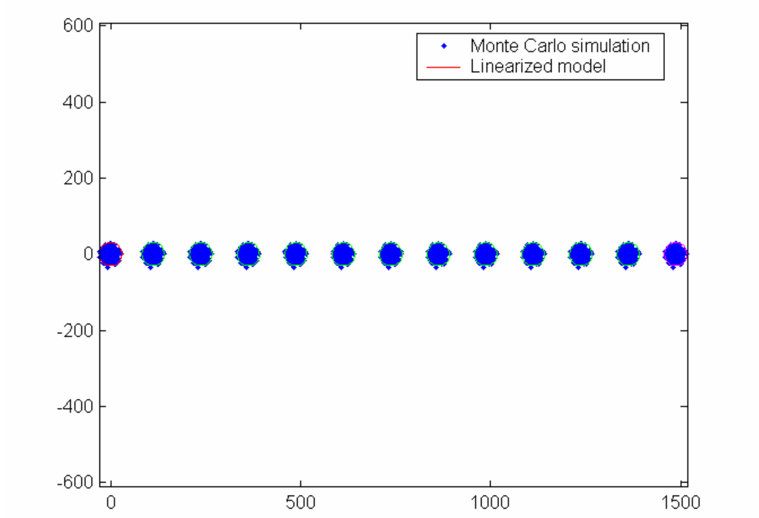

Figure 5. Uncertainty propagation for errors in initial position alone.

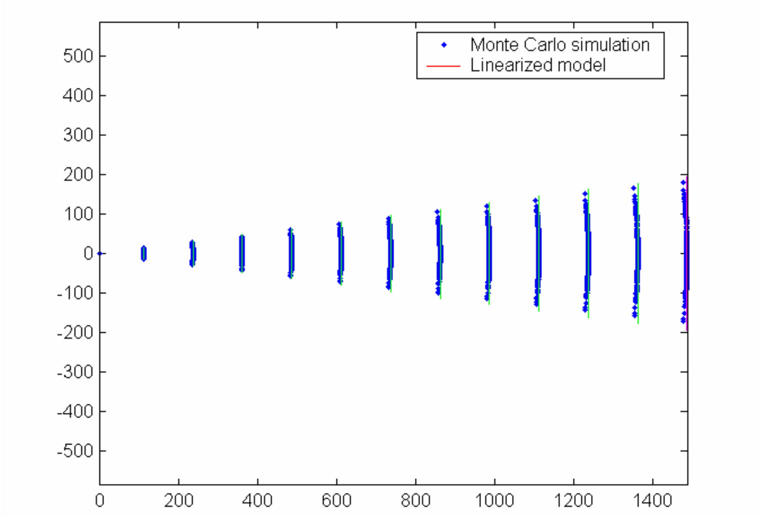

Figure 6. Uncertainty propagation for errors in initial angle alone.

For a planning horizon of 3km at a speed of 5 m/s, with longitudinal control error of 10% of the commanded speed $(\sigma_v = 0.1v = 0.5 m/s)$, gyro drift of 10 deg/hr ($\sigma_\omega$=10 deg/hr = $4.84 \cdot 10^{-5}$ rad/sec) and model error of 0.1m/s in x and y ($\sigma_x = \sigma_y =$ 0.1 m/s), the dominant term is the longitudinal control error (See Figure 7).

However, when considering the error caused by the uncertainty in the initial angle, this becomes the dominant error source for errors as small as 0.25 degrees (See Figure 8).

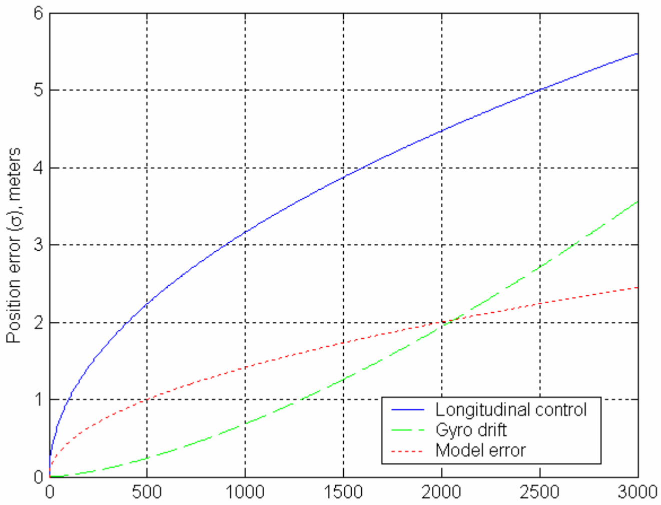

Figure 7. Comparison between different types of error

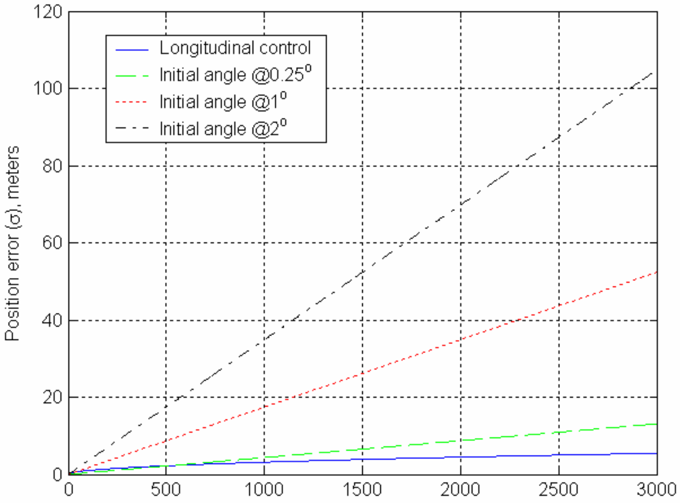

Figure 8. Comparison between different values of initial angle error and longitudinal control error

Figure 9 shows the results of combining all the types of error (with the values mentioned above, and an initial heading angle of 0.25 degrees) for a straight trajectory. Figure 10 shows the same analysis for a random trajectory.

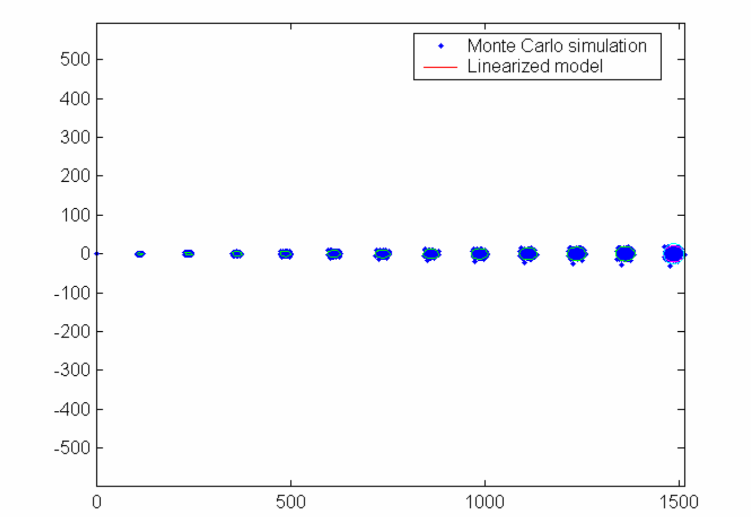

Figure 9. Error propagation for a straight trajectory with all error sources combined

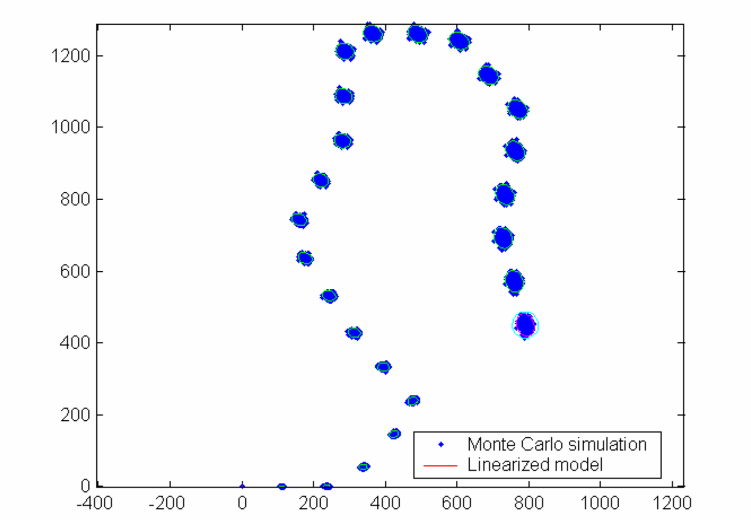

Figure 10. Error propagation for a random trajectory with all error sources combined

# 4 Implementation

## 4.1 Simplified error propagation model

In order to keep the planning problem tractable and efficient, we will approximate the probability density function (pdf) of the error with a symmetric Gaussian distribution, centered at the most likely location of the robot:

$$\begin{aligned} &\mathbf{q} = (x, y) \\ &\mathbf{q} : N(\mathbf{q}_c{}^k, \sigma^k) \end{aligned} \tag{8}$$

where $\mathbf{q}_c^k$ is the most likely location of the robot at step $k$, and $\sigma^k$ is the standard deviation of the distribution.

Let us define:

$$\varepsilon^k = 2 \cdot \sigma^k \tag{9}$$

We can then model the boundary of the uncertainty region as a disk centered at $\mathbf{q}_c^k$ with a radius $\varepsilon^k$. To propagate the uncertainty, we use the model:

$$\varepsilon(\mathbf{q}_c^k) = \varepsilon(\mathbf{q}_c^{k-1}) + k_u d(\mathbf{q}_c^{k-1}, \mathbf{q}_c^k) \tag{10}$$

where $k_u$ is the uncertainty accrued per unit of distance traveled, $\mathbf{q}_c^{k-1}$ is the previous position along the path, and $d(\mathbf{q}_c^{k-1}, \mathbf{q}_c^k)$ is the distance between the two adjacent path locations $\mathbf{q}_c^{k-1}$ and $\mathbf{q}_c^k$.

Equivalently, we can define the uncertainty at location $\mathbf{q}_c^k$ as:

$$\varepsilon(\mathbf{q}_c^k) = \varepsilon(\mathbf{q}_c^0) + k_u D(\mathbf{q}_c^k, \mathbf{q}_c^0) \tag{11}$$

where $\mathbf{q}_c^0$ is the initial most likely location of the robot, and $D(\mathbf{q}_c^k, \mathbf{q}_c^0)$ is the total distance traveled along the path from $\mathbf{q}_c^0$ to $\mathbf{q}_c^k$. The uncertainty rate $k_u$ is typically between 0.01 and 0.1 (1% to 10%) of distance traveled.

By modeling the error propagation in this manner, we are assuming that the dominant term is the uncertainty in the initial angle. Even though we are not explicitly modeling $\theta$ as a state variable, the effects of uncertainty in this variable are accounted for in the uncertainty propagation model for $\mathbf{q}=(x,y)$.

## 4.2 Planning with uncertainty

In a deterministic planner, the position of the robot on the plane is usually defined as $\mathbf{q}=(x,y)$, where $x$ and $y$ are deterministic variables. Because the position of the robot is now a probability distribution, the new representation for the position of the robot is

$$p(\mathbf{q} \mid \mathbf{q}_c{}^k, \varepsilon^k) \tag{12}$$

where $\mathbf{q}$ is a random variable representing the position of the robot, $\mathbf{q}_c{}^k$ is the most likely location of the robot, and $\varepsilon^k$ is the uncertainty in the position. As mentioned in the previous section, this pdf is modeled as a Gaussian distribution centered at $\mathbf{q}_c{}^k$ and with standard deviation $\sigma^k = \varepsilon^k / 2$.

The expected cost of going from a most likely location $\mathbf{q}_c{}^k$ to an adjacent most likely location $\mathbf{q}_c{}^{k+1}$ is

$$E\left[C\left((\mathbf{q}_c{}^k, \varepsilon^k), (\mathbf{q}_c{}^{k+1}, \varepsilon^{k+1})\right)\right] = \sum_{\forall i}\sum_{\forall j} C_o\left(\mathbf{q}_i{}^k, \mathbf{q}_j{}^{k+1}\right) \cdot p\left(\mathbf{q}_i{}^k, \mathbf{q}_j{}^{k+1} \mid \mathbf{q}_c{}^k, \varepsilon_k, \mathbf{q}_c{}^{k+1}, \varepsilon^{k+1}\right) \tag{13}$$

where $C_o(\mathbf{q}_i{}^k, \mathbf{q}_j{}^{k+1})$ is the deterministic cost of traveling from $\mathbf{q}_i{}^k$ to $\mathbf{q}_j{}^{k+1}$.

Expressing the joint probability as a conditional probability:

$$\begin{aligned} & E\left[C\left((\mathbf{q}_c{}^k, \varepsilon^k), (\mathbf{q}_c{}^{k+1}, \varepsilon^{k+1})\right)\right] = \\ & \sum_{\forall i}\sum_{\forall j} C_o\left(\mathbf{q}_i^k, \mathbf{q}_j^{k+1}\right) \cdot p\left(\mathbf{q}_j^{k+1} \mid \mathbf{q}_i^k, \mathbf{q}_c^k, \varepsilon_k, \mathbf{q}_c^{k+1}, \varepsilon^{k+1}\right) \cdot \\ & \qquad p\left(\mathbf{q}_i^k \mid \mathbf{q}_c^k, \varepsilon_k, \mathbf{q}_c^{k+1}, \varepsilon^{k+1}\right) \end{aligned} \tag{14}$$

$$E\left[C\left((\mathbf{q}_c{}^k,\varepsilon^k),(\mathbf{q}_c{}^{k+1},\varepsilon^{k+1})\right)\right]=$$

$$\sum_{\forall i}\sum_{\forall j}C_o\left(\mathbf{q}_i^k,\mathbf{q}_j^{k+1}\right)\cdot p\left(\mathbf{q}_j^{k+1}\mid\mathbf{q}_i^k,\mathbf{q}_c^k,\varepsilon_k,\mathbf{q}_c^{k+1},\varepsilon^{k+1}\right)\cdot \tag{15}$$

$$p\left(\mathbf{q}_i^k\mid\mathbf{q}_c^k,\varepsilon^k\right)$$

If $k_u << \varepsilon^k$, then $\varepsilon^k \approx \varepsilon^{k+1}$ (for a single step increment). We can then model the transition from $\mathbf{q}_c{}^k$ to $\mathbf{q}_c{}^{k+1}$ as a deterministic transition for the distribution $p\left(\mathbf{q}_i^k\mid\mathbf{q}_c^k,\varepsilon^k\right)$.

Under this assumption:

$$p\left(\mathbf{q}_j^{k+1}\mid\mathbf{q}_i^k,\mathbf{q}_c^k,\varepsilon_k,\mathbf{q}_c^{k+1},\varepsilon^{k+1}\right)=\begin{cases}1 & \text{if } j=i' \\ 0 & \text{otherwise}\end{cases} \tag{16}$$

where $i'$ is the image of $i$ under a transformation $T$ such that $T(\mathbf{q}_c^k)=\mathbf{q}_c^{k+1}$ (See Figure 11)

Substituting (16) in (15):

$$E\left[C\left((\mathbf{q}_c{}^k,\varepsilon^k),(\mathbf{q}_c{}^{k+1},\varepsilon^{k+1})\right)\right]=$$

$$\sum_{\forall i}C_o\left(\mathbf{q}_i^k,\mathbf{q}_{i'}^{k+1}\right)\cdot p\left(\mathbf{q}_i^k\mid\mathbf{q}_c^k,\varepsilon^k\right) \tag{17}$$

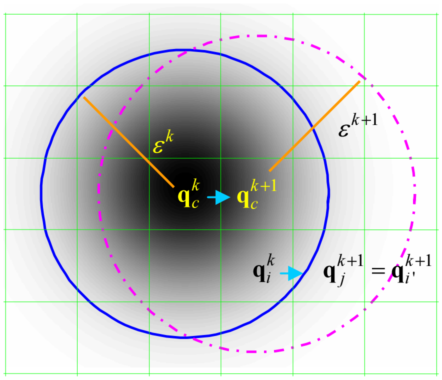

Figure 11. $p\left(\mathbf{q}_i^k\mid\mathbf{q}_c^k,\varepsilon_k,\mathbf{q}_c^{k+1},\varepsilon^{k+1}\right)$ and state transitions when $k_u << \varepsilon$

Since $\mathbf{q}_i{}^k$ and $\mathbf{q}_{i'}{}^k$ are neighbors, we can express $C_o(\mathbf{q}_i^k,\mathbf{q}_{i'}^{k+1})$ as:

$$C_o\left(\mathbf{q}_i^k,\mathbf{q}_{i'}^{k+1}\right)=aC_{\mathbf{q}}(\mathbf{q}_i^k)+bC_{\mathbf{q}}(\mathbf{q}_{i'}^{k+1}) \tag{18}$$

where *a*=*b*=*1* for horizontal or vertical neighboring cells, and *a*=*b*= $\sqrt{2}$ for diagonal neighboring cells.

Substituting (18) in (17):

$$E\left[C\left((\mathbf{q}_c{}^k,\varepsilon^k),(\mathbf{q}_c{}^{k+1},\varepsilon^{k+1})\right)\right]=$$

$$a\sum_{\forall i}C_o\left(\mathbf{q}_i^k\right)\cdot p\left(\mathbf{q}_i^k\mid\mathbf{q}_c^k,\varepsilon^k\right)+b\sum_{\forall i}C_o\left(\mathbf{q}_{i'}^{k+1}\right)\cdot p\left(\mathbf{q}_i^k\mid\mathbf{q}_c^k,\varepsilon^k\right) \tag{19}$$

$$E\left[C\left((\mathbf{q}_c{}^{k},\varepsilon^{k}),(\mathbf{q}_c{}^{k+1},\varepsilon^{k+1})\right)\right]= a\cdot C_E(\mathbf{q}_c^{k},\varepsilon_k)+bC_E(\mathbf{q}_c^{k+1},\varepsilon^{k}) \tag{20}$$

where

$$C_E(\mathbf{q}_c^{k},\varepsilon^{k}\ )=\sum_{\forall i}C_o\left(\mathbf{q}_i^{k}\right)\cdot p\left(\mathbf{q}_i^{k}\mid\mathbf{q}_c^{k},\varepsilon^{k}\right) \tag{21}$$

is the expected cost of traversing cell $\mathbf{q}_c^{k}$ if the uncertainty at this location is $\varepsilon^{k}$ .

Since it is possible to reach a cell $\mathbf{q}_c^{k}$ with different uncertainties, we will create an augmented state vector **r** such that:

$$\mathbf{r}=(\mathbf{q},\varepsilon) \tag{22}$$

hence defining a 3-D configuration space where x and y are the first two dimensions, and ε is the third dimension.

The planner used for planning with uncertainty is a modified version of A* in 3-D. The planner works as follows: we have a start location $\mathbf{q}^{0}$ with uncertainty $\varepsilon^{0}$, an end location $\mathbf{q}^{f}$ with uncertainty $\varepsilon^{f}$, and a 2-D cost map *C*. We form the augmented state variable $\mathbf{r}^{0}=(\mathbf{q}^{0},\varepsilon^{0})$ and place it in the OPEN list. The state $\mathbf{r}^{k}$ with lowest expected total cost to the goal is popped from the OPEN list. Next, $\mathbf{r}^{k}$ is expanded, and its successors $\mathbf{r}_j^{k+1}$ are placed in the OPEN list.

Since ε is function of **q**, the successors of $\mathbf{r}^{\mathbf{k}}$ are calculated only in the 2-D workspace defined by $\mathbf{q}^{\mathbf{k}}$, instead of the 3-D configuration space defined by $\mathbf{r}^{\mathbf{k}}$ (See Figure 12).

As the states are placed in the OPEN list, their uncertainty is updated using (10), and the expected total path cost of $\mathbf{r}_j^{k+1}$ is updated according to (20). However, if any states within the uncertainty region of $\mathbf{r}_j^{k+1}$ are labeled as OBST (obstacles), then the expected total path cost of $\mathbf{r}_j^{k+1}$ is set to infinity, therefore preventing any further expansion of that state.

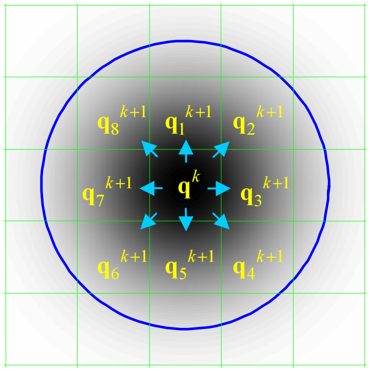

Figure 12. Successors of state $\mathbf{q}^{\mathbf{k}}$

Like in A*, the process described above is repeated until a cell $\mathbf{r}^{k+1}$ such that $\mathbf{q}^{k+1}=\mathbf{q}^{f}$ and $\varepsilon^{k+1}\le\varepsilon^{f}$ is popped out of the OPEN list. The path connecting the backpointers from $\mathbf{r}^{k+1}$ to $\mathbf{r}^{0}$ is the optimal path between $\mathbf{q}^{0}$ and $\mathbf{q}^{f}$ with a final uncertainty lower than $\varepsilon^{f}$ . If the OPEN list becomes empty and no such state has been found, then there is not path between $\mathbf{q}^{0}$ and $\mathbf{q}^{f}$ such that the final uncertainty is lower than $\varepsilon^{f}$ .

## 4.3 Using localization regions

The planner has the capability of using localization regions, such as areas with GPS coverage. If a state propagation yields a distribution that is totally contained in a GPS region (within 2σ), the uncertainty of this state is set to zero (or other constant value associated with the GPS region) instead of using (10) to propagate the uncertainty. This allows the planner to hop from one GPS region to another, if the lowest cost path requires it.

# 5 Results

## 5.1 Route planning example

The following is an example of our planner applied to the problem of finding the best path between two locations on opposite sides of Indiantown Gap, PA. In Figure 13 we can see the results when uncertainty is not being considered. We can use a regular 2-D planner for this task, or we could use our planner with an uncertainty rate of zero. If the uncertainty rate is zero, the third dimension of the planning s

pace is collapsed, and the search space becomes two-dimensional.

Figure 14 shows the resulting path when we use our planner and a propagation model with an uncertainty rate of 2%. Notice how the path is shortened in order to fit through the narrow opening in the lower left corner of the map.

Figure 15 shows the resulting path when the uncertainty rate is increased to 4%. The path cannot pass through the opening in the lower left corner, and now it has to go through the opening in the top right, producing a more costly path.

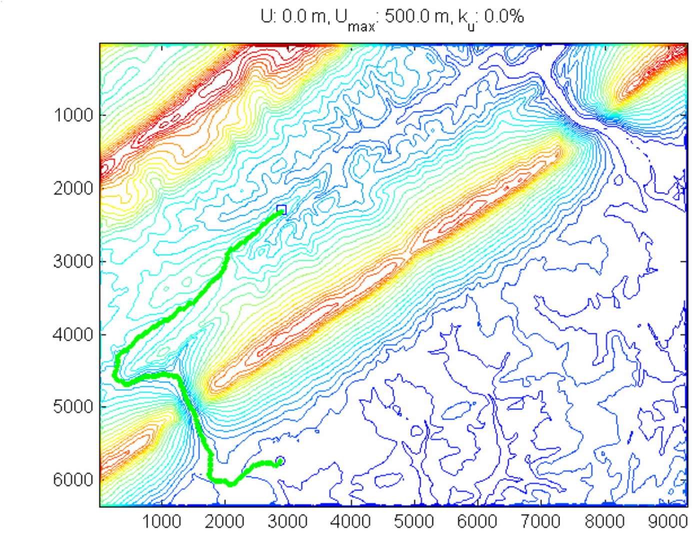

Figure 13. Path calculated with no uncertainty. Total path cost : 9357

Figure 16 shows the resulting path for the same uncertainty rate when there is a GPS region in the left side of the map (small rectangle). In this case the lowest cost path is obtained by going to the GPS region first (which resets the uncertainty to zero) and then going through the gap in the lower left corner (which was not feasible without the GPS region)

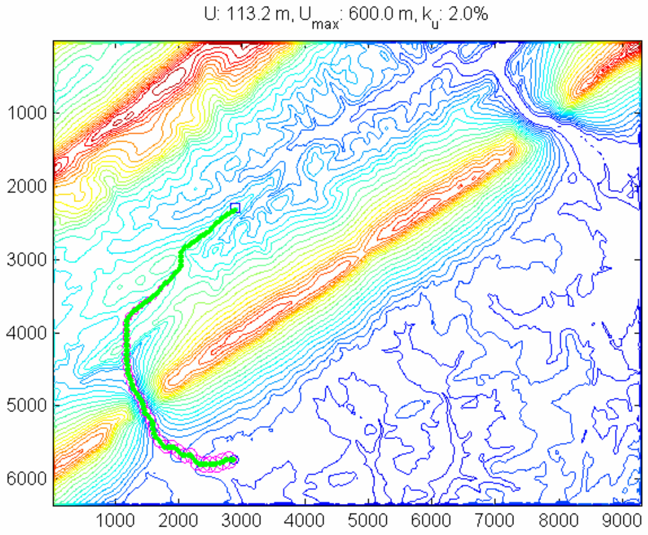

Figure 14. Path calculated with uncertainty rate $k_u$ = 2.0%. Total expected path cost: 20598

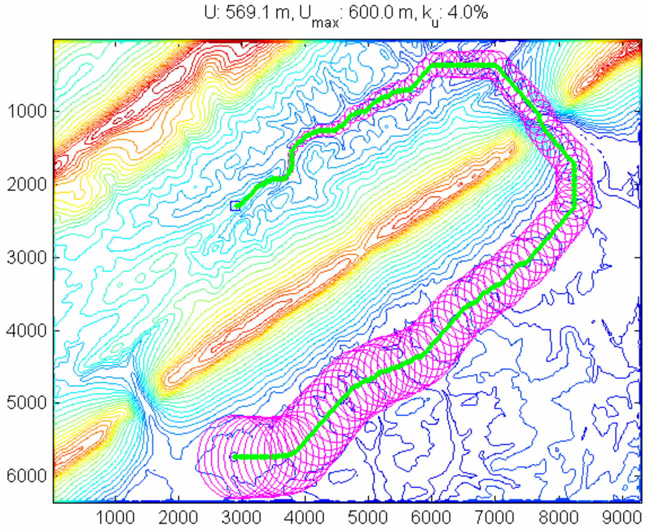

Figure 15. Path calculated with uncertainty rate $k_u$ = 4.0%. Total expected path cost = 44107

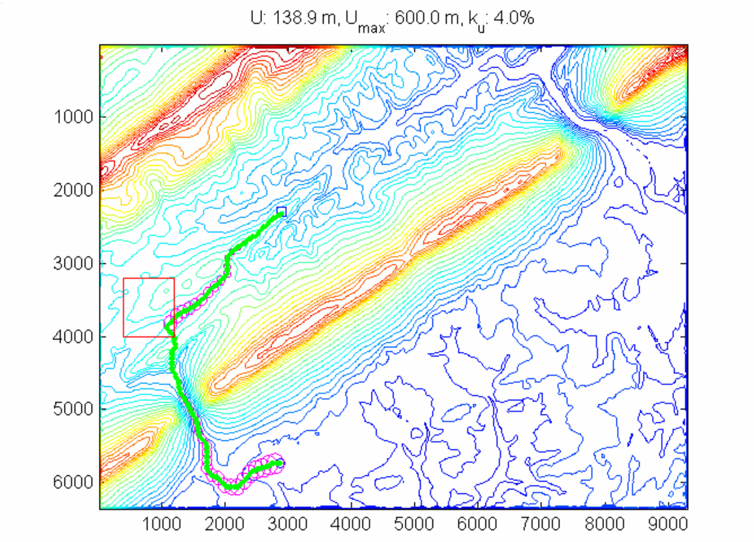

Figure 16. Path calculated with uncertainty rate $k_u$ = 4.0%, and using GPS regions (small square area on the left). Total expected path cost = 19458

## 5.2 Performance

The space complexity of the current implementation of the planner is $O(n_x \cdot n_y \cdot n_u)$, where $n_x$, $n_y$ and $n_u$ are the dimensions along the x, y and u directions.

The time complexity is $O(Q+R)$, where:

- $Q = k_u \cdot (n_x \cdot n_y)^2 \cdot n_u$ is the number of operations required to calculate the expected cost.
- $R = n_x \cdot n_y \cdot n_u \cdot \log(n_x \cdot n_y \cdot n_u))$ is the number of operations required by A* to calculate the path
- $k_u$ is the uncertainty rate.

For $k_u > 0$, the $Q$ term dominates the time complexity of the algorithm. If $k_u = 0$, or the calculation of the expected value is performed beforehand, then the $R$ term becomes the dominant one.

However, because of the uncertainty is dependent on $x$ and $y$, the states expanded by the algorithm are far fewer than $n_x \cdot n_y \cdot n_u$, and the average time complexity of the algorithm is consequently much lower as well.

For binary-cost worlds, the search space for the algorithm is a 2-D manifold, with a cone-like shape around the start location (Figure 17 shows a cross-section of the search space containing the start location). In this type of world, the number of propagations is $O(n_x \cdot n_y)$

For continuous-cost worlds, the search space is something between a 2-D manifold and a thin 3-D volume (Figure 18 shows a cross-section of the search space containing the start location). In this type of world the number of propagations is $O(n_x \cdot n_y \cdot n_{u'})$, where $n_{u'}$ is the average number of propagations per cell. Figure 19 shows $n_{u'}$ vs $n_u$ for a batch of 1800 simulations with $k_u$ varying between 1% and 10%, and $n_x = n_y$ varying from 50 to 250 cells. We can see that even though $n_{u'}$ does increase with $n_u$, it is always a small fraction of $n_u$. For $n_u = 100$ the average value of $n_{u'}$ is 3.4, and the maximum value is 7.9.

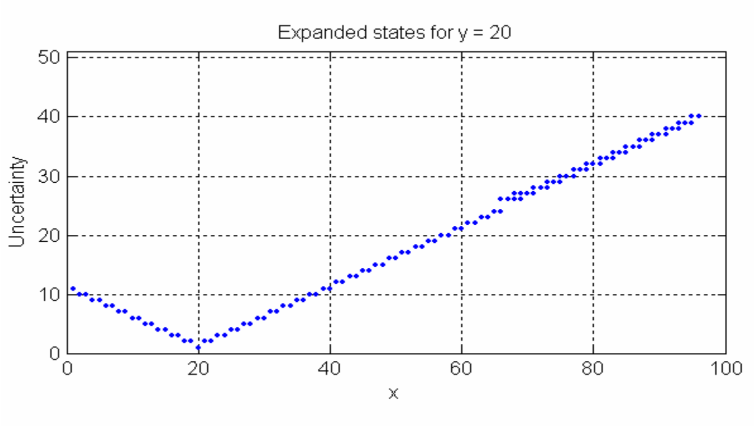

Figure 17. State expansion for binary cost world

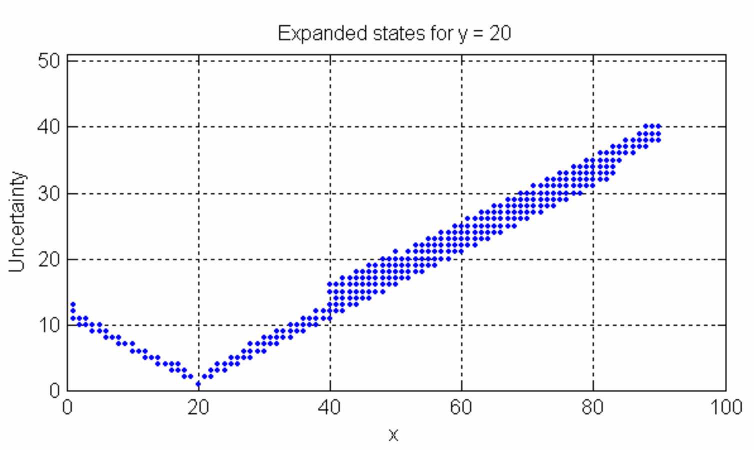

Figure 18. State expansion for continuous cost world

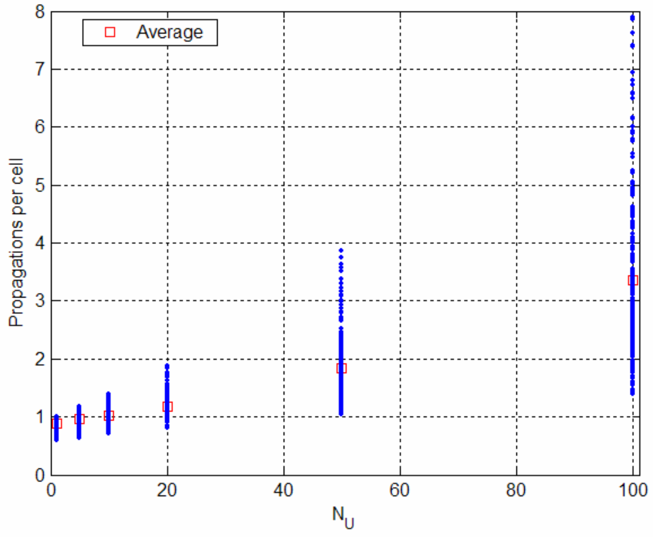

Figure 19. Propagations per cell vs number of uncertainty levels

# 6 Conclusions and Future Work

We have introduced an efficient path planner that calculates resolution-optimal paths while considering uncertainty in position. Even though the planner uses three dimensions to represent the state variables, the computational complexity of the algorithm is very close to that of a 2-D algorithm.

The algorithm takes advantage of the small increase in uncertainty from one step to the next, to model the transitions in a semi-deterministic fashion. This enables the use of a deterministic planner such as A* to solve the planning problem.

Because of the efficient use of the state space, the algorithm does not require a lengthy pre-processing stage as required by Roy and Thrun [8]. The planning time is under 1 second for worlds up to 150x150, and under 10 seconds for worlds up to 250x250. Figure 20 shows the average planning time for 1800 simulations modeling 10 different worlds with sizes from 50x50 to 250x250 (in xy). The uncertainty levels used varied from 1 to 100, and the uncertainty rates from 1% to 10%. The processor used was a Pentium M 1.4GHz.

Our algorithm runs in significantly less time than algorithms that require pre-processing of all states. Depending on the size of the world and the uncertainty rate the speed up factor of our algorithm is between 8 and 80 times, depending on the size and characteristics of the world (See Figure 21).

Future work includes producing a version of the algorithm that allows for more efficient replanning and exploring more complex representations of the error propagation model.

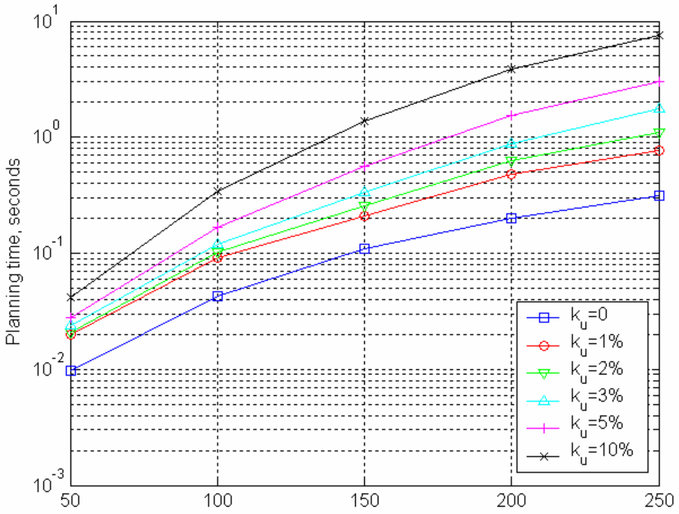

Figure 20. Planning time for different world sizes and uncertainty rates

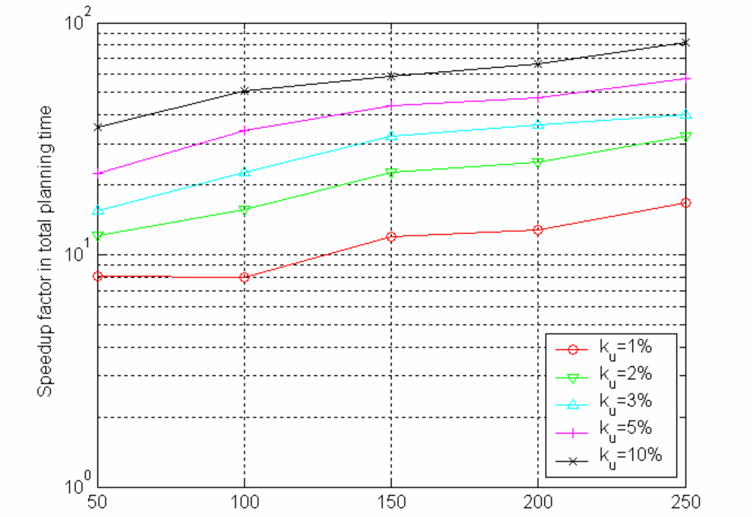

Figure 21. Speed up factor in total planning time compared to preprocessing all states.

# 7 Acknowledgments

This work was sponsored by the U.S. Army Research Laboratory under contract "Robotics Collaborative Technology Alliance" (contract number DAAD19-01-2-0012). The views and conclusions contained in this document do not represent the official policies or endorsements of the U.S. Government.

# References

[1] Latombe, J.C., Lazanas, A., and Shekhar, S., "Robot Motion Planning with Uncertainty in Control and Sensing," Artificial Intelligence J., 52(1), 1991, pp. 1-47.

[2] J. C. Latombe_Robot Motion Planning._Kluwer Academic Publishers.

[3] Lazanas, A., and Latombe, J.-C. 1992. “Landmark-based robot navigation”. In Proc. 10th National Conf. on Artificial Intelligence (AAAI-92), 816--822. Cambridge, MA: AAAI Press/The MIT Press. http://citeseer.nj.nec.com/lazanas92landmarkbased.html

[4] B. Bouilly. Planification de Strategies de Deplacement Robuste pour Robot Mobile. PhD thesis, Insitut National Polytechnique, Tolouse, France, 1997

[5] B. Bouilly, T. Sim'eon, and R. Alami. “A numerical technique for planning motion strategies of a mobile robot in presence of uncertainty”. In Proc. of the IEEE Int. Conf. on Robotics and Automation, volume 2, pages 1327--1332, Nagoya (JP), May 1995. http://citeseer.nj.nec.com/bouilly95numerical.html

[6] Haït, A., Simeon, T. and Taïx, M. "Robust motion planning for rough terrain navigation".Published in IEEE Int. Conf. on Intelligent Robots and Systems Kyongju, Korea, . http://citeseer.nj.nec.com/319787.html

[7] Th. Fraichard and R. Mermond "Integrating Uncertainty And Landmarks In Path Planning For Car-Like Robots" Proc. IFAC Symp. on Intelligent Autonomous Vehicles March 25-27, 1998. http://citeseer.nj.nec.com/fraichard98integrating.html

[8] Roy, N. and Thrun, S. (1999). Coastal navigation with mobile robots. In Advances in Neural Processing Systems 12, volume 12, pages 1043--1049

[9] K. Goldberg, M. Mason, and A. Requicha. Geometric uncertainty in motion planning. Summary report and bibliography. IRIS TR. USC Los Angeles

[10] Kelly, A, "Linearized Error Propagation in Odometry", The International Journal of Robotics Research. February 2004, vol. 23, no. 2, pp.179-218(40).