Motion Planning under Perception and Control Uncertainties with Space Exploration Guided Heuristic Search

Chao Chen[1] and Markus Rickert[1] and Alois Knoll[2]

***Abstract*— Reliability and safety are extremely important for autonomous driving in real traffic scenarios. However, due to imperfect control and sensing, the actual state of the vehicle cannot be flawlessly predicted or measured, but estimated with uncertainty. Therefor, it is important to consider the execution risk advance in motion planning for a solution with a high success rate. The *Space Exploration Guided Heuristic Search (SEHS)* method is extended to deal with perception and control uncertainty in its two planning stages. First, the localization uncertainty is evaluated with a simple probabilistic robot model by the *Space Exploration* to find a path corridor with sufficient localization quality for the desired motion accuracy. Then, a trajectory controller is modeled with nonholonomic kinematics for the belief propagation of a robot state with primitive motions. The dynamic model and the control feedback are approximated in a close neighborhood of the reference trajectory. In this case, the *Heuristic Search* can propagate the state uncertainty as a normal distribution in the search tree to guarantee a high probability of safety and to achieve the required final accuracy.**

The belief-based SEHS is evaluated in several simulated scenarios. Compared to the basic SEHS method that assumes perfection, motions with higher execution successful rate are produced, especially the human-like behaviors for driving through narrow passages and precise parking. This confirms the major contribution of this work in exploiting the uncertainties for motion planning in autonomous driving.

I. Introduction

The primary setup of motion planning requires a model of the robot kinematic and the knowledge about the environment. The former one defines the system state and dynamics, as well as the control inputs. The latter one determines the external constraints of the motion, for example to avoid collisions with obstacles. However, in the real world, these models are usually approximations with simplifications based on certain assumptions of the real systems. The control inputs are limited to a certain precision. The perception also suffers from random noise. Especially in autonomous driving, the sensing of the vehicle state depends on the ego location and motion, e.g., the distance and orientation relative to landmarks or the driving velocity and yaw-rate; the detection of the obstacles relies upon the physical nature of the objects, such as shape, material, and the environmental situation, e.g., temperature or lighting condition. As a result, a motion planned with ideal assumptions may fail in real life due to these uncertainties.

Fig. 1 shows an example of a vehicle driving into a narrow passage. In this case, the simple safety margin approach faces the contradiction that the vehicle may collide with obstacles during execution when considering a small safety margin in planning or that no solution can be found if the collision margin is too conservative. The belief-based SEHS planner on the other hand calculates the uncertainty at each node of the search tree so that a more reliable solution is found. The trajectory is safe with respect to imperfect sensing and execution.

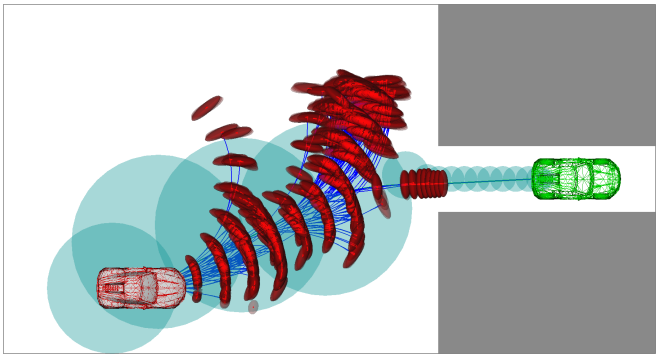


Fig. 1. Space exploration guided heuristic search regarding perception and control uncertainties. A search tree (node in red, edge in blue) is constructed following the path corridor (cyan circles) from the start (red vehicle) to the goal (green vehicle). The red ellipses show the uncertainty at each node.

With imperfect control and sensing, the system takes input $\boldsymbol{u}_t$ to go from state $\boldsymbol{x}_{t-1}$ to $\boldsymbol{x}_t$ with a certain probability. The actual state $\boldsymbol{x}_t$ cannot be detected perfectly, but observed as a measurement $\boldsymbol{y}_t$ with a certain confidence. This procedure is modeled as the *state transition function* f and the *measurement function* g in (1). The uncertainties are captured with random noise $\boldsymbol{\epsilon}_t$ and $\boldsymbol{\delta}_t$.

$$\begin{aligned} \boldsymbol{x}_t &= f(\boldsymbol{x}_{t-1}, \boldsymbol{u}_t) + \boldsymbol{\epsilon}_t \\ \boldsymbol{y}_t &= g(\boldsymbol{x}_t) + \boldsymbol{\delta}_t \end{aligned} \tag{1}$$

The general idea to deal with this probabilistic model is to update the knowledge about the conditional distribution of the state $\hat{p}(\boldsymbol{x}_t) = p(\boldsymbol{x}_t | \boldsymbol{y}_{1:t}, \boldsymbol{u}_{1:t})$, called *belief* [1], iteratively with the control and measurement inputs. As the functions f and g are known, the *state transition probability* $p(\boldsymbol{x}_t | \boldsymbol{x}_{t-1}, \boldsymbol{u}_t)$ and the *measurement probability* $p(\boldsymbol{y}_t | \boldsymbol{x}_t)$ are available. In this case, the belief state can be calculated with (2).

$$\begin{aligned} \bar{p}(\boldsymbol{x}_t) &= \int p(\boldsymbol{x}_t | \boldsymbol{x}_{t-1}, \boldsymbol{u}_t)\, \hat{p}(\boldsymbol{x}_{t-1})\, \mathrm{d}\boldsymbol{x}_{t-1} \\ \hat{p}(\boldsymbol{x}_t) &= \eta\, p(\boldsymbol{y}_t | \boldsymbol{x}_t)\, \bar{p}(\boldsymbol{x}_t) \end{aligned} \tag{2}$$

[1]Chao Chen and Markus Rickert are with fortiss, An-Institut Technische Universität München, Munich, Germany

[2]Alois Knoll is with Robotics and Embedded Systems, Department of Informatics, Technische Universität München, Munich, Germany

The first equation, as *motion update*, gives the prior estimation of the state distribution $\bar{p}(\boldsymbol{x}_t) = p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t-1}, \boldsymbol{u}_{1:t})$ based on the control input $\boldsymbol{u}_t$ and the previous belief $\hat{p}(\boldsymbol{x}_{t-1})$. The second equation, as *measurement update*, fuses the measurement $\boldsymbol{y}_t$ with the prediction using Bayes rules for the posterior belief $\hat{p}(\boldsymbol{x}_t)$. η is a normalizing factor. Thus, starting from the initial knowledge $\hat{p}(\boldsymbol{x}_0)$, the following belief state can be calculated recursively.

At the planning time, the vehicle model is known. Therefore, the uncertainty in the motion update can be modeled. The prior knowledge about the environment provides information about localization uncertainty regarding landmarks or environment conditions. Apart from that, most of the sensing uncertainty is available during the execution in a limited sensing range, which can be considered by an online adaptation or by replanning. The *Space Exploration Guided Heuristic Search (SEHS)* approach [2], [3] provides a general framework for motion planning in real-time. It is able to handle realistic traffic scenarios [4], [5] and dynamic environments [6]. The contribution of this paper is to extend both the exploration and the search phases with belief state updates to achieve robuster solutions for real-life autonomous driving applications.

The chapters after the introduction are organized as follows: First, a short review about the related work is provided in Section II along with the basic Kalman Filter techniques; then Section III and IV elaborate the details about the belief state calculation during Space Exploration and Heuristic Search in the SEHS approach; after that, two examples are presented to demonstrate the benefits of the belief-based SEHS in Section V; finally, Section VI is the conclusion of the research with future aspects.

II. Related Work

The robotic perception and control uncertainties are well introduced in [1] as *probabilistic robotics*. The sensing uncertainty is also addressed in [7] and [8]. If the state transition function and the measurement function are linear (3) and the initial state and the noises are normal distributed, the belief propagation can be solved with the Kalman Filter [9]. If the system is non-linear, the Extended Kalman Filter (EKF) [10] can be applied when the system dynamic can be locally linearized.

$$\begin{aligned} \boldsymbol{x}_t &= \boldsymbol{A}_t\boldsymbol{x}_{t-1} + \boldsymbol{B}_t\boldsymbol{u}_t + \boldsymbol{\epsilon}_t \\ \boldsymbol{y}_t &= \boldsymbol{C}_t\boldsymbol{x}_t + \boldsymbol{\delta}_t \end{aligned} \tag{3}$$

The Kalman filter models the belief state with normal distributions and updates the mean and covariance recursively. First, a prediction step (4) calculates the covariance of the prior belief state $\bar{\boldsymbol{\Sigma}}$ regarding the control noise $\boldsymbol{\epsilon}_t \sim N(\boldsymbol{0}, \boldsymbol{R}_t)$.

$$\begin{aligned} \bar{\boldsymbol{\mu}}_t &= \boldsymbol{A}_t\boldsymbol{\mu}_{k-1} + \boldsymbol{B}_t\boldsymbol{u}_t \\ \bar{\boldsymbol{\Sigma}}_t &= \boldsymbol{A}_t\boldsymbol{\Sigma}_{t-1}\boldsymbol{A}_t^\top + \boldsymbol{R}_t \end{aligned} \tag{4}$$

Then, an update step (5) produces the covariance of the posterior belief state $\boldsymbol{\Sigma}$ with the measurement noise $\boldsymbol{\delta}_t \sim N(\boldsymbol{0}, \boldsymbol{Q}_t)$. $\boldsymbol{K}$ is the Kalman gain.

$$\begin{aligned} \boldsymbol{K}_t &= \bar{\boldsymbol{\Sigma}}_t\boldsymbol{C}_t^\top(\boldsymbol{C}_t\bar{\boldsymbol{\Sigma}}_t\boldsymbol{C}_t^\top + \boldsymbol{Q}_t)^{-1} \\ \boldsymbol{\mu}_t &= \bar{\boldsymbol{\mu}}_t + \boldsymbol{K}_t(\boldsymbol{y}_t - \boldsymbol{C}_t\bar{\boldsymbol{\mu}}_t) \\ \boldsymbol{\Sigma}_t &= \bar{\boldsymbol{\Sigma}}_t - \boldsymbol{K}_t\boldsymbol{C}_t\bar{\boldsymbol{\Sigma}}_t \end{aligned} \tag{5}$$

However, as the measurement $\boldsymbol{y}_t$ is not available during the planning time, it is not possible to update the mean $\boldsymbol{\mu}_t$. Instead, the distribution $\boldsymbol{\mu}_t \sim N(\boldsymbol{0}, \boldsymbol{\Lambda}_t)$ is evaluated. A linear feedback controller is employed for the control input $\boldsymbol{u}_t$ in (6) with $(\hat{\boldsymbol{x}}_t - \tilde{\boldsymbol{x}}_t)$ as the error. By substituting (6) into (4) and (5), the measurement covariance $\boldsymbol{\Lambda}_t$ can be calculated with (7). The detailed derivation can be found in [11] and [12]. Therefore, the combined covariance of the belief state is $\boldsymbol{\Sigma}_t + \boldsymbol{\Lambda}_t$.

$$\boldsymbol{u}_t = -\boldsymbol{D}_t(\hat{\boldsymbol{x}}_t - \tilde{\boldsymbol{x}}_t) \tag{6}$$

$$\boldsymbol{\Lambda}_t = (\boldsymbol{A}_t - \boldsymbol{B}_t\boldsymbol{D}_t)\boldsymbol{\Lambda}_{t-1}(\boldsymbol{A}_t - \boldsymbol{B}_t\boldsymbol{D}_t)^\top + \boldsymbol{K}_t\boldsymbol{C}_t\bar{\boldsymbol{\Sigma}}_t \tag{7}$$

In [13], Lazanas and Latombe simplified the perception and execution uncertainty with landmark regions where the robot can perform perfect motion. A path can be produced with back-projections [14].

In [8], the concept of information space (I-Space) was introduced to model sensing problems, as the actual state is usually unknown in practical motion planning, but observed with sensing and prior-knowledge. In this case, planning is done in I-Space with *Partially Observable Markov Decision Process* (POMDP) by treating the information space as a new kind of state space. However, due to state explosion, only problems with a few state dimensions can be solved in I-Space. A further idea is to simplify the original I-Space to a derived I-Space and to exploit the problem-specific knowledge to aid the planning. In [15], a technique called *Guided Cluster Sampling* is introduced, which divides the belief space into small sub-spaces, so that POMDP is more efficient in these sub-spaces.

In [16], a utility roadmap planner is developed to find a path that minimizes uncertainty and explores the workspace when necessary. Van der Berg et al. [17] evaluated the multiple results from a RRT planner with the LQG method so that the most robust one could be selected. However, if the uncertainty is considered during the planning, it is possible to prune the unlikely branches to accelerate planning. Platt et al. [12] assumed the maximum likelihood observation to achieve deterministic belief-system dynamics. Thus, a LQR method generated motion policies that punishes the action which increases covariance or runs away from the expectation.

A RRT in belief space was developed in [11], which associated each vertex with beliefs, so that the optimal path could be found regarding the probability of reaching the goal without collision. [18] approached path planning in belief space considering the motion uncertainty in state propagation with sensing uncertainty for a success rate of the path. However, the Monte-Carlo-based collision checks are very time consuming.

The Space Exploration Guided Heuristic Search algorithm solves the nonholonomic vehicle motion planning problem in two steps. First, vehicle geometry and kinematics are simplified to a holonomic point robot. Therefore, the workspace can be efficiently evaluated with reachable sets for a path corridor. Then, heuristic search is conducted with adapted motion primitives following the path corridor. Compared to the other search-based methods, the space exploration provides knowledge about free space dimension and topology for heuristics and space decomposition that boost the search performance. In order to plan with uncertainties, the simple space exploration can evaluate the localization in the early phase of the planning. The heuristic search with motion primitives enables belief updates by providing a simple trajectory control model that can be linearized around the reference path. Details are presented in the following sections.

III. Space Exploration with Perception Uncertainties

Most of the sensing uncertainty is only available after performing the sensing action in a certain range. However, some prior knowledge about perception is useful to estimate the uncertainty during planning time, e.g., the position of landmarks or the availability of external localization. Such information can provide a rough idea about the localization precision.

By applying the holonomic kinematic model with a point robot, the motion uncertainty is modeled with a normal distribution whose covariance is proportional to the moving distance. After a motion update, a position measurement is taken with a certain confidence of a normal distribution around the real position. If the control and measurement noise are independent and identically distributed in x and y coordinates, the state uncertainty can be represented with a single standard deviation value, i.e., the states are distributed in a circle around the mean position. In this case, the Kalman filter equations (4) (5) and (7) degenerate to a single dimension version. Assuming in step t the motion noise variance is σ_u^2 and the measurement noise variance is σ_m^2, the prediction is now (8) and the update is (9).

$$\hat{\sigma}_t^2 = \sigma_{t-1}^2 + \sigma_u^2 \tag{8}$$

$$\sigma_t^2 = \frac{\hat{\sigma}_t^2 \sigma_m^2}{\hat{\sigma}_t^2 + \sigma_m^2} \tag{9}$$

If the robot takes a feedback factor d for the position control, regarding (7) the control uncertainty λ_t^2 can be updated with (10). The total uncertainty is the sum of σ_t^2 and λ_t^2.

$$\lambda_t^2 = (1-d)^2 \lambda_{t-1}^2 + \frac{\hat{\sigma}_t^4}{\hat{\sigma}_t^2 + \sigma_m^2} \tag{10}$$

The motion uncertainty remains constant with the same step size, but the localization uncertainty is position dependent. In this case, the robot achieves different state uncertainties through different paths. Space exploration can evaluate the path corridors to find an optimal one that safely passes obstacles and confidently reaches the goal.

Algorithm 1: SpaceExploration($c_{\text{start}}, c_{\text{goal}}$)

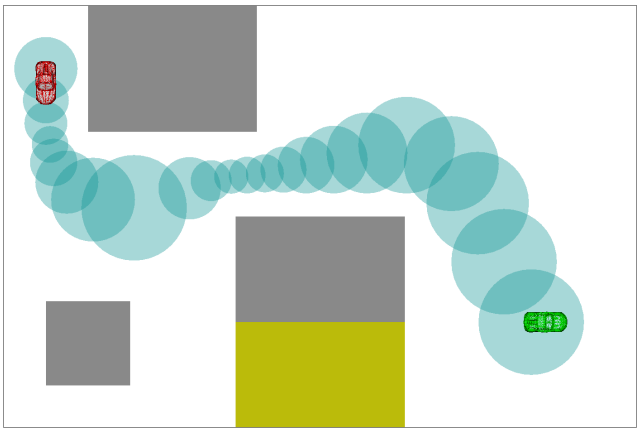
```
1  S_closed ← ∅;
2  S_open ← {c_start};
3  while S_open ≠ ∅ do
4      c_current ← PopTop(S_open);
5      if f[c_goal] < f[c_current] then
6          return success;
7      else if !Exist(c_current, S_closed) then
8          S_open ← Expand(c_current) ∪ S_open;
9          if Overlap(c_current, c_goal) then
10             if σ_current + λ_current ≤ σ_goal then
11                 if f[c_current] < g[c_goal] then
12                     g[c_goal] = f[c_current];
13                     parent[c_goal] = c_current;
14         S_closed ← {c_current} ∪ S_closed;
15 return failure;
```

Algorithm 1 shows the pseudo-code for the belief-based space exploration. Circles are expanded in a best-first order regarding a Euclidean distance heuristic that explores the free workspace between start and goal in an efficient flood-like manner. An open set S_{open} and a closed set S_{closed} are employed to manage the new circles and the already evaluated ones. Details of the subroutines are explained in [3]. The belief-based version holds the heuristic search costs (g, h, f) and the state variance (σ, λ) for each circle. In the function Expand(c_{current}), the child circle uncertainty (σ, λ) is calculated with (8), (9) and (10). It is used as the safety margin to calculate the circle radius. Function Exist($c_{\text{current}}, S_{\text{closed}}$) identifies a redundant circle when its total uncertainty is larger than a circle in the closed set that holds its center point. Thus, a place could be explored several times as a larger confidence can be achieved with a longer traveling distance through another path. The goal is reached when the total uncertainty is less than the desired value.

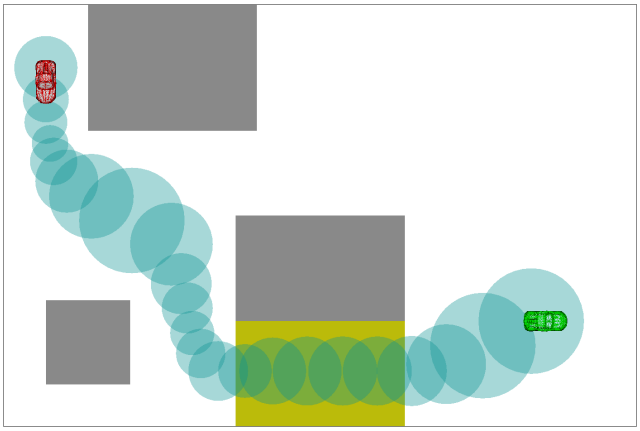
An example is shown in Fig. 2. The robot can localize itself with $\sigma_m = 0.5\,\text{m}$ in the whole area. There are some additional landmarks in the yellow region, where the robot can achieve a better localization with $\sigma_m = 0.1\,\text{m}$. The robot motion has uncertainty $\sigma_u = 0.2\,\text{m}$. The feedback factor is $d = 0.5$. The space exploration results show that when the goal condition requires a greater confidence $\sigma_{\text{goal}} = 0.27\,\text{m}$, the robot will take a longer path to go through the landmark region where it can achieve better localization that reduces the overall uncertainty as in Fig. 2(b). Otherwise, a shorter path is taken as in Fig. 2(a).

IV. Heuristic Search with Control Uncertainties

In heuristic search, the sensing uncertainty is modeled similar to space exploration with an additional orientation measurement. The measurement noise is zero mean normal distributed with location dependent covariance. The control

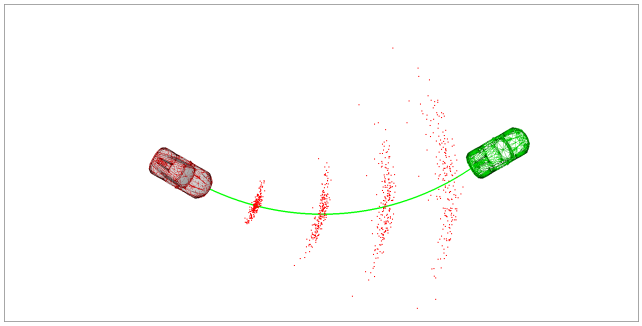


(a) Small goal confidence with $\sigma_{\text{goal}} = 0.30\,\text{m}$

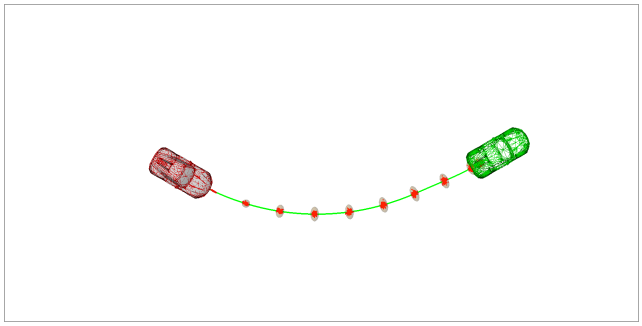


(b) Large goal confidence with $\sigma_{\text{goal}} = 0.27\,\text{m}$

Fig. 2. Space Exploration with localization uncertainty. The exploration result is a path corridor with *cyan circles* from start (*red vehicle*) to the goal (*green vehicle*). Obstacles are *gray*. Localization is more accurate in the *yellow area*.



(a) Turning motion with feed-forward control



(b) Turning motion with feedback control

Fig. 3. Vehicle kinematics take control inputs with additive white Gaussian noise (AWGN). The *green line* is the desired trajectory. The *red dots* are 200 samples following the trajectory. The *orange ellipse* is the estimated covariance of the feedback controller.

uncertainty is modeled with more complex vehicle kinematics and feedback functions.

A. Motion Model with Uncertainty

A nonholonomic vehicle model (11) is applied to the system dynamic. (x, y, θ) are the position and orientation of the vehicle. k is the curvature of the motion. The control input is (v, ω) for velocity and steering (curvature derivative).

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{k} \end{pmatrix} = \begin{pmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ k & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} v \\ \omega \end{pmatrix} \tag{11}$$

This nonholonomic vehicle dynamic violates the Kalman Filter conditions, i.e., although starting from a normal distributed belief state, the next state is not a normal distribution. Fig. 3 shows the propagation of state samples with motion uncertainty following a curve trajectory. The Gaussian noise of steering input results in a distribution in a quasi-polar coordinate system as in Fig. 3(a). In this case, the particle filter would be a better method to estimate the motion uncertainty, but more time consuming and not applicable for online motion adaptation. By applying a feedback controller, the samples are distributed closely around the reference trajectory in Fig. 3(b). The orange ellipses are the estimated covariance of normal distribution, which hold most of the samples. The details about the uncertainty estimation are introduced in the next subsection.

B. Trajectory Control based on Motion Primitives

As the state propagation is performed with primitive motions, the control command of each step is known. In this case, a controller can take the primitive motion as feed-forward input with the feedback input from the errors as (12). The first term on the right is the desired control input $\boldsymbol{u}_t$. The second term is the feedback from the belief state $\hat{\boldsymbol{x}}_t$ with a factor $\boldsymbol{D}_t$. The last term $\boldsymbol{\delta}_t$ is the additive control noise. The control input is normal distributed if $\hat{\boldsymbol{x}}_t$ and $\boldsymbol{\delta}_t$ are normal random variables.

$$\hat{\boldsymbol{u}}_t = \boldsymbol{u}_t - \boldsymbol{D}_t(\hat{\boldsymbol{x}}_t - \boldsymbol{x}_t) + \boldsymbol{\delta}_t \tag{12}$$

Without loss of generality, the vehicle state is mapped to a reference trajectory coordinate. A controller is designed with feedback functions that determine the velocity input based on the longitudinal error and steering input according to the lateral error and orientation difference. If the controller works well, the vehicle will follow the trajectory with a very small deviation. In this case, the vehicle dynamic can be approximated in a close range around the reference trajectory in a small time step t (13). s_t is the moving distance and u_t is

the curvature change in this time step. These control inputs are generated by the controller.

$$\begin{pmatrix} x_{t+1} \\ y_{t+1} \\ \theta_{t+1} \\ k_{t+1} \end{pmatrix} = \begin{pmatrix} x_t \\ y_t \\ \theta_t \\ k_t \end{pmatrix} + \begin{pmatrix} \cos\theta_t & 0 \\ \sin\theta_t & 0 \\ k_t & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} s_t \\ u_t \end{pmatrix} \tag{13}$$

The motion in longitudinal direction can be approximated with $x_{t+1} = x_t + s_t$ when θ is very small. The actual input $\hat{s}_t$ is calculated with (14). $\delta_{s,t}$ is assumed to be additive white Gaussian noise (AWGN), which consists of base-level noise and a part proportional to the control input. In this case, the actual longitudinal state x_{t+1} is normal distributed if the previous one is normal distributed.

$$\hat{s}_t = s_t - \alpha(\hat{x}_t - x_t) + \delta_{s,t} \tag{14}$$

The lateral motion is more complicated. The curvature cannot affect the lateral position or orientation alone, but with the longitudinal input as (15) when θ_t is very small. Here, s_t is a known desired value and k_t is a variable with uncertainty.

$$\begin{aligned} \theta_{t+1} &= \theta_t + s_t k_t \\ y_{t+1} &= y_t + s_t\theta_t + \frac{s_t^2}{2} k_t \end{aligned} \tag{15}$$

The steering input is calculated with (16). The desired input u_t is always zero in the reference trajectory coordinate system. The actual curvature $\hat{k}_t$ can be calculated with (17).

$$\hat{u}_t = u_t - \beta(\hat{y}_t - y_t) - \gamma(\hat{\theta}_t - \theta_t) + \delta_{u,t} \tag{16}$$

$$\hat{k}_t = k_t - \beta(\hat{y}_t - y_t) - \gamma(\hat{\theta}_t - \theta_t) + \delta_{u,t} \tag{17}$$

$\delta_{u,t}$ is assumed to be AWGN. According to (15) and (17), the actual curvature k_{t+1}, lateral state y_{t+1}, and orientation θ_{t+1} are normal distributed if the previous ones are normal distributed.

Therefore, if only (x, y, θ) is considered as state variable and (s, k) as the control inputs, the vehicle dynamics can be linearized with the matrices (18) for the EKF approach. The motion update and measurement update are calculated with (4) and (5). The control uncertainty is updated with (7). As this model is in the reference state frame, the covariance matrix $\boldsymbol{\Sigma}$ and $\boldsymbol{\Lambda}$ should be rotated to the next reference state frame for the next step. The measurement covariance should be mapped to the reference frame.

$$\begin{aligned} \boldsymbol{A} &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & s_t \\ 0 & 0 & 1 \end{pmatrix}, \boldsymbol{B} = \begin{pmatrix} 1 & 0 \\ 0 & \frac{s_t^2}{2} \\ 0 & s_t \end{pmatrix} \\ \boldsymbol{C} &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \boldsymbol{D} = \begin{pmatrix} \alpha & 0 & 0 \\ 0 & \beta & \gamma \end{pmatrix} \end{aligned} \tag{18}$$

C. Heuristic Search Algorithm

Algorithm 2 is the pseudo code for the belief-based heuristic search. Vehicle states are expanded with primitive motions following the path corridor from the space exploration, which build a search tree for a valid motion from start to goal. Each node holds the belief state $(\boldsymbol{\Sigma}, \boldsymbol{\Lambda})$. During expansion, they are updated iteratively with (4) (5) and (7). The total uncertainty $\boldsymbol{\Sigma} + \boldsymbol{\Lambda}$ of the belief state is considered as collision margin in the collision checks. The function $\text{Exist}(\vec{q}_{\text{current}}, c_{\text{current}}, S_{\text{closed}})$ resolves the nodes not only by the vehicle state, but also the uncertainty $\boldsymbol{\Sigma} + \boldsymbol{\Lambda}$. The goal is reached when the final uncertainty is less than the desired goal uncertainty.

Algorithm 2: HeuristicSearch($\{c_i\}, \vec{q}_{\text{start}}, \vec{q}_{\text{goal}}$)

```
S_closed ← ∅;
S_open ← {q_start};
while S_open ≠ ∅ do
    q_current ← PopTop(S_open);
    c_current ← MapNearest(q_current);
    if f[q_goal] < f[q_current] then
        if Σ_current + Λ_current ≤ Σ_goal then
            return success;
    else if !Exist(q_current, c_current, S_closed) then
        S_open ← Expand(q_current, c_current) ∪ S_open;
        if h[q_current] < R_goal then
            GoalExpand(q_current, q_goal);
        S_closed ← {q_current} ∪ S_closed;
return failure;
```

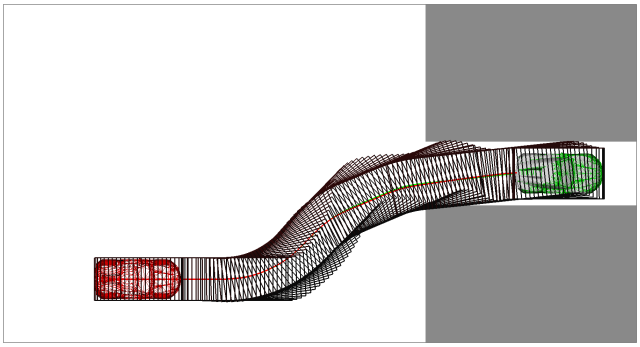
Fig. 4 compares the planning results from a normal heuristic search method and the belief-based heuristic search in the same narrow passage scenario as in Fig. 1. The normal heuristic search returns a shorter path with sharp turns that may cause collision in Fig. 4(a). The planner finds a trajectory that makes the s-turn at the very beginning followed by a rather straight path segment to the goal in Fig. 4(b). Therefore, the vehicle fits better into the narrow passage even with execution and localization errors. The search tree is visualized with the uncertainty ellipse for each node in Fig. 1. Only nodes with small uncertainties can be generated in the narrow passage.

V. Experiments

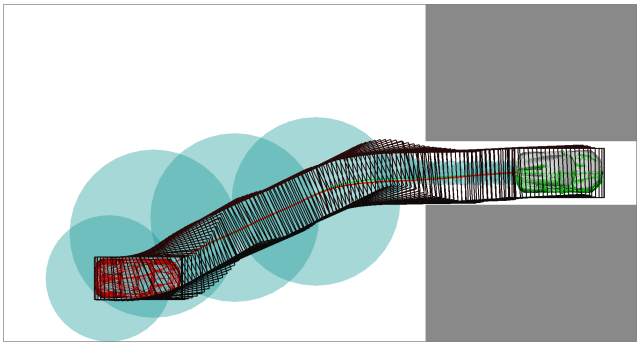
The belief-based SEHS is evaluated in two further experiments, and compared with the basic SEHS method. The experiments are conducted on a computer with an Intel Core i7 2.90 GHz processor and 8 GB RAM running Linux.

A. Precise Parking

In this scenario, the vehicle should park in a position with high precision for a special purpose, e.g., inductive charging. Fig. 5 shows the setup with a marker (yellow circle) placed ahead of the goal position. The vehicle has a 20° field of view (light yellow cone) in a 15 m range to detect the marker. If the marker is in sight, the vehicle can localize itself with extra high precision $\boldsymbol{Q}_1 = \text{diag}(0.002\,5\,\text{m}^2, 0.002\,5\,\text{m}^2, 0.000\,1\,\text{rad}^2)$. Otherwise, the baseline localization has a large uncertainty $\boldsymbol{Q}_0 =$

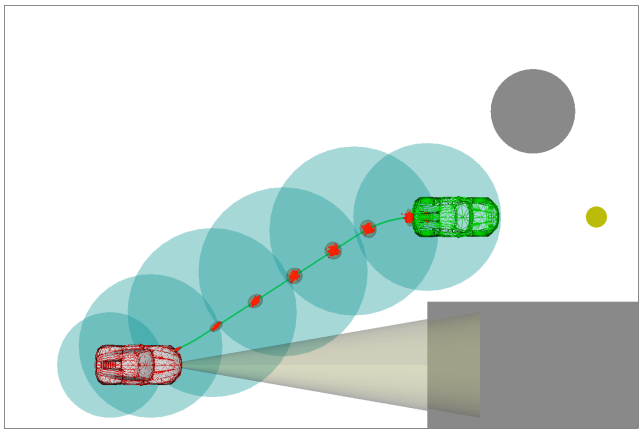


(a) Normal heuristic search

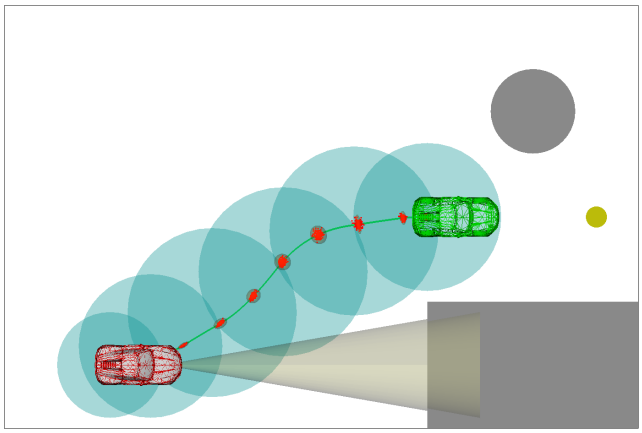


(b) Belief-based heuristic search

Fig. 4. Heuristic search in a narrow passage scenario. The result trajectory is demonstrated with rectangles including the safety margin for uncertainties.



(a) Normal SEHS solution with low final accuracy



(b) Belief-based SEHS solution with high final accuracy

Fig. 5. Precise parking to the goal position with the help of a localization marker (*yellow circle*). The detection range of the vehicle is illustrated as the *light yellow cone*.

$\mathrm{diag}(0.25\,\mathrm{m}^2, 0.25\,\mathrm{m}^2, 0.04\,\mathrm{rad}^2)$. The control factor and motion noise factor is the same as in the previous experiment.

The normal version of SEHS plans a direct path with shortest distance. However, due to the narrow field of view, the vehicle cannot see the marker during most of the parking motion. The final accuracy is $(0.041\,\mathrm{m}, 0.102\,\mathrm{m})$ in (x, y) directions. The belief-based SEHS uses larger steering motion at the beginning, which greatly reduce the bearing angle to the marker, so that the vehicle can detect it during the rest of the motion to the final position. In Fig. 5(b), the point cloud start to converge when the marker is in sight. As a result, it achieves a better accuracy of $(0.025\,\mathrm{m}, 0.055\,\mathrm{m})$ for the final position.

B. Multiple Passages

Fig. 6 shows a scenario where the vehicle should travel through one of the passages to reach the goal position. The passage above is wider than the other two. The passage below is in a precise localization region with $\boldsymbol{Q}_1 = \mathrm{diag}(0.01\,\mathrm{m}^2, 0.01\,\mathrm{m}^2, 0.002\,5\,\mathrm{rad}^2)$. The localization elsewhere is worse with $\boldsymbol{Q}_0 = \mathrm{diag}(0.25\,\mathrm{m}^2, 0.25\,\mathrm{m}^2, 0.01\,\mathrm{rad}^2)$. The control factor is $(0.5, 0.2, 0.2)$ for (α, β, γ). The motion noise has a factor $(0.1, 0.2)$ to the desired control input.

The normal SEHS method just selects the nearest passage in the middle for the shortest path length in Fig. 6(a). However, the passage is too narrow and may cause collisions with respect to the localization and control uncertainties. The belief-based SEHS finds a more robust solution in the wider passage in Fig. 6(b). If a small final uncertainty $(0.062\,5\,\mathrm{m}^2, 0.062\,5\,\mathrm{m}^2, 0.002\,7\,\mathrm{rad}^2)$ is required, the belief-based SEHS takes the longest path with better localization to achieve the goal accuracy in Fig. 6(c).

Table I compares the planning performance between SEHS and the belief-based version in this scenario. The space exploration are all under 10 ms. The normal SEHS is one magnitude faster, however the result is riskier. The belief-based SEHS can plan a safer motion in a 100 ms time frame, which could still be used to adapt the motion in real-time.

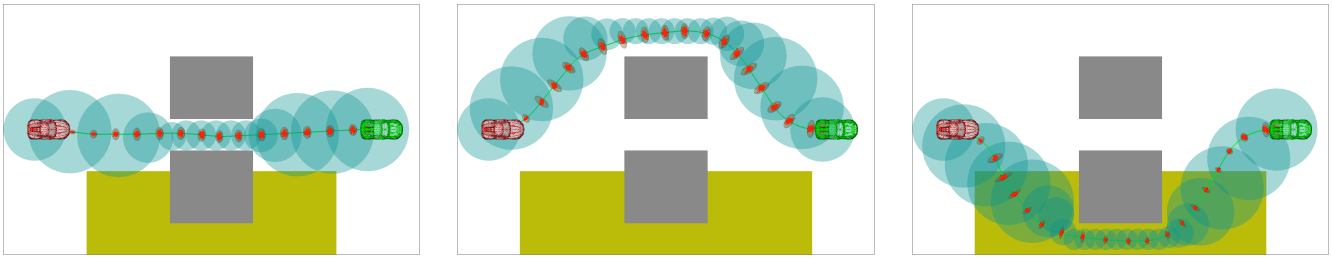
VI. Conclusion and Future Work

In the real world, the sensing and control of a mobile robot or autonomous vehicle are imperfect. The uncertainty will

TABLE I

Results of Multiple Passage Scenario

Fig.	SE Time (ms)	SE Nodes	HS Time (ms)	HS Nodes
6(a)	1	658	16	1822
6(b)	4	1531	129	8022
6(c)	6	1736	159	10424



(a) Normal SEHS planner selects the nearest passage

(b) Belief-based SEHS planner selects the widest passage

(c) Belief-based SEHS planner selects the passage with better localization for better goal confidence.

Fig. 6. Multiple passages are possible between the start (*green vehicle*) and the goal (*red vehicle*). The localization is precise in the *yellow region*.

affect the performance of robot motion, cause collision, or fail to reach the desired goal state. Therefore, it is necessary to consider the stochastic properties of the system and the risk of the path in planning time. Instead of the problems with rather simple dynamics, this paper studies the detailed vehicle kinematics with a trajectory controller in reference path coordinates. The system dynamics is locally approximated to facilitate belief update with the EKF approach. Therefore, the basic SEHS method is extended to evaluate localization uncertainty in the space exploration phase and further on with control uncertainty during the heuristic search procedure. The belief-based SEHS can produce safer solutions with an adaptive safety margin. Furthermore, it can provide the guarantee that the goal can be reached with a certain confidence or accuracy. The motion planning problem with uncertainty can be solved in a rather short time frame with SEHS, which is important for online motion adaptations.

There are still some open points for future study. First, the measurement update is performed with the assumption that the perceptions are independent. However, if two sensing actions are done close together in space and time, they usually correlate. In this case, the second measurement does not provide as much information as the first one. Second, when the heuristic search algorithm takes the traveling distance as cost in SEHS, the state uncertainty does not directly generate reward until it reaches the goal or moves close to obstacles. A better heuristic may also consider to evaluate the state with less uncertainty first. Another important assumption for the EKF in belief-based SEHS is that the vehicle should follow the trajectory with very little deviations. Otherwise, the precondition of EKF is violated and the belief propagation must be done with a Monte-Carlo method, which is time consuming.

References

[1] S. Thrun, W. Burgard, and D. Fox, *Rrobabilistic Robotics*. The MIT Press, 2006.

[2] C. Chen, M. Rickert, and A. Knoll, "Combining space exploration and heuristic search in online motion planning for nonholonomic vehicles," in *Proc. IEEE Intelligent Vehicles Symposium*, June 2013, pp. 1307–1312.

[3] C. Chen, "Motion planning for nonholonomic vehicles with space exploration guided heuristic search," Ph.D. dissertation, Technische Universität München, 2016. [Online]. Available: https://mediatum.ub.tum.de/1292197

[4] C. Chen, M. Rickert, and A. Knoll, "A traffic knowledge aided vehicle motion planning engine based on space exploration guided heuristic search," in *Proc. IEEE Intelligent Vehicles Symposium*, June 2014, pp. 535–540.

[5] C. Chen, M. Rickert, and A. Knoll, "Path planning with orientation-aware space exploration guided heuristic search for autonomous parking and maneuvering," in *Proc. IEEE Intelligent Vehicles Symposium*, June 2015, pp. 1148–1153.

[6] C. Chen, M. Rickert, and A. Knoll, "Kinodynamic motion planning with space-time exploration guided heuristic search for car-like robots in dynamic environments," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct. 2015, pp. 2666–2671.

[7] H. Choset, K. Lynch, S. Hutchinson, G. Kantor, W. Burgard, *et al.*, *Principles of Robot Motion : Theory, Algorithms, and Implementation*. The MIT Press, 2005.

[8] S. LaValle, *Planning Algorithms*. Cambridge University Press, 2006.

[9] R. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.

[10] G. Welch and G. Bishop, "An introduction to the kalman filter," University of North Carolina at Chapel Hill, Tech. Rep., July 1995.

[11] A. Bry and N. Roy, "Rapidly-exploring random belief trees for motion planning under uncertainty," in *Proc. IEEE International Conference on Robotics and Automation*, May 2011.

[12] R. Platt Jr., R. Tedrake, L. Kaelbling, and T. Lozano-Pérez, "Belief space planning assuming maximum likelihood observations," in *Proc. Robotics: Science and Systems*, June 2010.

[13] A. Lazanas and J.-C. Latombe, "Motion planning with uncertainty: a landmark approach," *Artificial Intelligence*, vol. 76, no. 1-2, pp. 287–317, 1995.

[14] A. Lazanas, "Reasoning about uncertainty in robot motion planning," Ph.D. dissertation, Stanford University, 1994.

[15] H. Kurniawati, T. Bandyopadhyay, and N. Patrikalakis, "Global motion planning under uncertain motion, sensing, and environment map," *Autonomous Robots*, vol. 33, no. 3, pp. 255–272, 2012.

[16] B. Burns and O. Brock, "Sampling-based motion planning with sensing uncertainty," in *Proc. IEEE International Conference on Robotics and Automation*, Apr. 2007, pp. 3313–3318.

[17] J. van den Berg, P. Abbeel, and K. Goldberg, "LQG-MP: Optimized path planning for robots with motion uncertainty and imperfect state information," *International Journal of Robotics Research*, vol. 30, no. 7, pp. 895–913, 2011.

[18] D. Lenz, M. Rickert, and A. Knoll, "Heuristic search in belief space for motion planning under uncertainties," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct. 2015, pp. 2659–2665.