Landmark Placement for Accurate Mobile Robot Navigation

Maximilian Beinhofer Jörg Müller Wolfram Burgard
Department of Computer Science, University of Freiburg, Germany

***Abstract*— Being able to navigate accurately is one of the basic capabilities a mobile robot needs to effectively execute a variety of tasks, including collision avoidance, docking, manipulation, and path following. One popular approach to achieve the desired accuracy is to use artificial landmarks for a sufficiently accurate localization. In this paper, we consider the problem of optimally placing landmarks for robots navigating repeatedly along a given set of trajectories. Our method aims at minimizing the number of landmarks for which a bound on the maximum deviation of the robot from its desired trajectory can be guaranteed with high confidence. The proposed approach incrementally places landmarks utilizing linearized versions of the system dynamics of the robot, thus allowing for an efficient computation of the deviation guarantee. To verify this guarantee for the real and possibly non-linear system dynamics of the robot, our method then performs a Monte Carlo simulation. We evaluate our algorithm in extensive experiments carried out both in simulation and with a real robot. The experiments demonstrate that our method requires substantially fewer landmarks than other approaches to achieve the desired accuracy.**

***Index Terms*— Localization, Autonomous Navigation, Service Robots**

I. Introduction

One of the main challenges for mobile service robots is autonomous navigation. Especially in industrial applications, a high degree of accuracy is required to avoid collisions and, e.g., to ensure reliable docking maneuvers. However, robots navigating autonomously tend to deviate from their desired trajectory due to uncertainty in motion and position. Typically, they are equipped with on-board sensors to estimate the deviation and react according to a feedback control law. In service tasks, robots usually have to repeatedly execute a fixed number of trajectories. As real world environments often contain dynamic and ambiguous areas, many practical applications rely on artificial landmarks placed along these trajectories to allow for accurate self-localization [6, 7]. In these applications, placing the artificial landmarks is often expensive or the computational power of the robot is limited. Therefore, it is beneficial to select the smallest possible number of landmark positions which still guarantees an accurate navigation.

In this paper, we present a novel algorithm for landmark placement. It computes a landmark configuration which guarantees that the deviation of the robot from its desired trajectory stays below a user-defined threshold $d_{\max}$ with high confidence. To check if the guarantee holds, we use the specific properties of the robot and its navigation task in the landmark placement algorithm. The algorithm works in two stages. In a first stage, it uses a linearized motion model and observation model of the robot to efficiently place landmarks in an incremental fashion. This stage aims at placing the smallest number of landmarks for which the deviation guarantee holds. In a second stage, the algorithm employs a Monte Carlo simulation for the computed landmark configuration to check if the guarantee also holds for the possibly non-linear models.

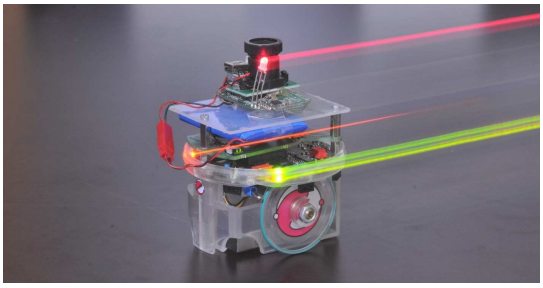

Fig. 1. The miniature e-puck robot we use in the experiments. Mounted on top is a wireless webcam detecting uniquely identifiable visual markers attached to the ceiling.

Our approach has several characteristics which make it especially useful for mobile robot navigation. It can deal with arbitrary trajectories, and the maximum allowed deviation of the robot can be defined individually for every part of the trajectories. Taking into account the properties of the individual robotic system results in customized landmark sets: while high-precision robots only need few landmarks for reaching the deviation guarantee, low cost systems typically require more landmarks. As our placement algorithm efficiently evaluates the guarantee using linear models, it can deal even with large instances of the landmark placement problem (i.e., long trajectories). Note that our incremental method simultaneously determines the number of landmarks needed and their positions to meet the desired guarantee.

This paper is organized as follows. After discussing related work in the following section, we formalize the problem definition in Section III. In Section IV we describe the prediction of the deviation from the trajectory in linearized systems. Afterwards, in Section V, we present our incremental landmark placement algorithm. Finally, we provide extensive experiments in which we evaluate the algorithm in simulation as well as with the real e-puck robot [9] depicted in Fig. 1.

II. Related Work

In the past, the problem of finding an optimal set of landmark positions has been addressed from several points

of view. Sala *et al.* [12] select landmark positions so that at every position in the map, at least k landmarks are observable. Jourdan and Roy [8] consider a set of possible target positions. They place sensors on the walls of buildings to minimize the average position error bound in the sensor network. Unlike these methods, our approach takes into account the full specification of the robot and its navigation task.

Vitus and Tomlin [15] also consider the full problem specification to place sensors in the environment. They approximate the a-priori covariances with the a-posteriori covariances of the most likely run of the robot. Similar to our approach, van den Berg *et al.* [3] evaluate sensor positions using the exact a-priori distributions in a linearized system. As they focus mainly on path planning, they restrict themselves to randomly sampled positions of a single sensor. Our previous work [1] selects landmarks maximizing the mutual information between the sensor readings and the states of the robot. It solely applies Monte Carlo simulations to estimate the a-priori distributions, which makes it computationally more demanding.

While all of the approaches above place artificial landmarks or sensors before operation of the robot, the following approaches decide whether to utilize observed landmarks during operation. In contrast to our method, their decisions are based on a-posteriori distributions, i.e., the information already gathered by the robot. Thrun [14] selects the subset of the observed landmarks for localization which minimizes the average posterior localization error. Strasdat *et al.* [13] and Zhang *et al.* [17] both consider landmark selection in the context of the simultaneous localization and mapping (SLAM) problem. Strasdat *et al.* use reinforcement learning to create a landmark selection policy whereas Zhang *et al.* minimize the entropy of the a-posteriori distributions.

In contrast to the above-mentioned approaches, our method optimizes the positions of the landmarks so that the robot stays within a user-defined region around the trajectory with high a-priori probability.

III. Problem Definition

We consider the problem of placing landmarks for localization and control of a mobile robot. We assume the time to be discretized into steps of equal duration. At each time step t the state of the robot is defined by a vector $\mathbf{x}_t \in \mathcal{X}$, which changes over time according to the stochastic motion model

$$\mathbf{x}_t = f(\mathbf{x}_{t-1}, \mathbf{u}_{t-1}, \mathbf{m}_t) \tag{1}$$

where $\mathbf{u}_t \in \mathcal{U}$ is the control command at time t. Thereby, the motion is disturbed by Gaussian noise $\mathbf{m}_t \sim \mathcal{N}(\mathbf{0}, M_t)$. For self-localization, we assume that the robot is equipped with a sensor taking measurements of a set of landmarks $\mathcal{A} = \{l_1, ..., l_n\}$ according to the measurement function

$$\mathbf{z}_t = h(\mathbf{x}_t, \mathbf{n}_t, \mathcal{A}) \tag{2}$$

where the sensor signal is disturbed by Gaussian noise $\mathbf{n}_t \sim \mathcal{N}(\mathbf{0}, N_t)$. The covariances M_t and N_t model the uncertainty in the motion and the measurements, respectively.

We define a navigation task as a trajectory that the robot should follow. A trajectory $\mathcal{T} = (\langle \mathbf{x}_0^\star, \mathbf{u}_0^\star \rangle, \ldots, \langle \mathbf{x}_T^\star, \mathbf{u}_T^\star \rangle)$ can be considered as a series of states and desired controls the robot should execute to reach these states. In this navigation task, we assume that the trajectory will be executed using a linear-quadratic regulator (LQR) [4] feedback controller. At each time step t the LQR controller selects the control command $\mathbf{u}_t$ which minimizes the quadratic cost function

$$\mathbb{E}\Big[\sum_{\ell=t}^{T}((\mathbf{x}_\ell - \mathbf{x}_\ell^\star)^T C(\mathbf{x}_\ell - \mathbf{x}_\ell^\star) + (\mathbf{u}_\ell - \mathbf{u}_\ell^\star)^T D(\mathbf{u}_\ell - \mathbf{u}_\ell^\star))\Big], \tag{3}$$

where C and D are positive definite weight matrices.

The localization uncertainty and, as a result, also the deviation from the desired trajectory strongly depend on the specific configuration of landmarks $\mathcal{A} = \{l_1, ..., l_n\}$ which are observed during operation. Our approach selects landmarks l_i from a continuous space of possible landmark locations $\mathcal{L}$. We evaluate the quality of a landmark configuration based on the deviation of the (real) state $\mathbf{x}_t$ from the desired state $\mathbf{x}_t^\star$ at each time step t (ignoring the control part $\mathbf{u}_{0:T}^\star$ of the trajectory). In particular, we consider the Euclidean distance between the part of the state $\mathbf{x}_t^{\text{pos}}$ describing the position of the robot and $\mathbf{x}_t^{\star\text{pos}}$. We focus on limiting the deviation

$$d^{\text{pos}}(\mathbf{x}_t, \mathbf{x}_t^\star) = \|\mathbf{x}_t^{\text{pos}} - \mathbf{x}_t^{\star\text{pos}}\|_2 \tag{4}$$

of the robot from its trajectory at all time steps $t \in [0, T]$. Our approach aims at finding the landmark configuration $\mathcal{A}$ with the fewest elements for which the *deviation guarantee*

$$\forall t \in [0, T] : \; p\left(d^{\text{pos}}(\mathbf{x}_t, \mathbf{x}_t^\star) \leq d_{\max}(\mathbf{x}_t^\star) \mid \mathcal{A}\right) \geq p_{\min} \tag{5}$$

holds. This guarantee ensures that the probability of deviating at most $d_{\max}$ from the desired trajectory is at least $p_{\min}$. Note that $d_{\max}$ can be either a globally constant value or depend on the position or time.

IV. Predicting the Deviation from the Trajectory

To validate the guarantee (5) for a certain landmark configuration $\mathcal{A}$, we need to compute $p\left(d^{\text{pos}}(\mathbf{x}_t, \mathbf{x}_t^\star) \leq d_{\max}(\mathbf{x}_t^\star) \mid \mathcal{A}\right)$. For this, we consider the a-priori probability distribution

$$p(\mathbf{x}_t - \mathbf{x}_t^\star \mid \mathcal{A}) = \int\int p(\mathbf{x}_t - \mathbf{x}_t^\star \mid \mathbf{u}_{0:t-1}, \mathbf{z}_{1:t}, \mathcal{A}) \cdot p(\mathbf{u}_{0:t-1}, \mathbf{z}_{1:t} \mid \mathcal{A}) \, \mathrm{d}\mathbf{u}_{0:t-1} \, \mathrm{d}\mathbf{z}_{1:t} \,, \tag{6}$$

which averages over the observations $\mathbf{z}_{1:t}$ and controls $\mathbf{u}_{0:t-1}$ that are not yet available during landmark placement.

For general, non-linear systems, the a-posteriori distributions $p(\mathbf{x}_t - \mathbf{x}_t^\star \mid \mathbf{u}_{1:t-1}, \mathbf{z}_{1:t}, \mathcal{A})$ can be used to estimate the a-priori distributions (6) via Monte Carlo simulation by sampling observations and controls and averaging over numerous runs [1]. However, this is computationally expensive for large instances of the landmark placement problem.

A. A-Priori State Estimation in Linearized, Gaussian Systems

In the main part of our landmark placement algorithm, we locally linearize the system and approximate all distributions by Gaussians. This allows for an analytical evaluation of the guarantee (5), making it more efficient than Monte Carlo simulations. Linearizing the motion model (1) and the sensor

model (2) around the desired trajectory $(\mathbf{x}^*_{0:T},\ \mathbf{u}^*_{0:T})$ by first order Taylor expansion leads to

$$\begin{aligned}\mathbf{x}_t &= f(\mathbf{x}^\star_{t-1}, \mathbf{u}^\star_{t-1}, \mathbf{0}) + A_t(\mathbf{x}_{t-1} - \mathbf{x}^\star_{t-1}) \\ &\quad + B_t(\mathbf{u}_{t-1} - \mathbf{u}^\star_{t-1}) + V_t\mathbf{m}_t\,, \qquad (7)\\ \mathbf{z}_t &= h(\mathbf{x}^\star_t, \mathbf{0}, \mathcal{A}) + H_t(\mathbf{x}_t - \mathbf{x}^\star_t) + W_t\mathbf{n}_t\,, \qquad (8)\end{aligned}$$

with the Jacobians

$$\begin{aligned}A_t = \frac{\partial f}{\partial \mathbf{x}}(\mathbf{x}^\star_{t-1}, \mathbf{u}^\star_{t-1}, \mathbf{0}),\ B_t = \frac{\partial f}{\partial \mathbf{u}}(\mathbf{x}^\star_{t-1}, \mathbf{u}^\star_{t-1}, \mathbf{0}),\\ V_t = \frac{\partial f}{\partial \mathbf{m}}(\mathbf{x}^\star_{t-1}, \mathbf{u}^\star_{t-1}, \mathbf{0}),\\ H_t = \frac{\partial h}{\partial \mathbf{x}}(\mathbf{x}^\star_t, \mathbf{0}, \mathcal{A}),\ W_t = \frac{\partial h}{\partial \mathbf{n}}(\mathbf{x}^\star_t, \mathbf{0}, \mathcal{A})\,. \qquad (9)\end{aligned}$$

In this linearized system, the Gaussian a-posteriori distribution $p(\mathbf{x}_t - \mathbf{x}^\star_t \mid \mathbf{u}_{1:t-1}, \mathbf{z}_{1:t}, \mathcal{A}) \sim \mathcal{N}(\mu_t - \mathbf{x}^\star_t, P_t)$ of the deviation from the trajectory can be computed recursively using a Kalman filter [16]. The Kalman filter propagates a given initial Gaussian distribution $p(\mathbf{x}_0 - \mathbf{x}^\star_0 \mid \mathcal{A}) \sim \mathcal{N}(\mu_0 - \mathbf{x}^\star_0, P_0)$ according to the actual control commands in the *motion update*

$$\begin{aligned}\bar{\mu}_t - \mathbf{x}^\star_t &= A_t(\mu_{t-1} - \mathbf{x}^\star_{t-1}) + B_t(\mathbf{u}_{t-1} - \mathbf{u}^\star_{t-1})\\ \bar{P}_t &= A_t P_{t-1} A_t^T + V_t M_t V_t^T\,. \qquad (10)\end{aligned}$$

and according to the measurements in the *observation update*

$$\begin{aligned}K_t &= \bar{P}_t H_t^T (H_t \bar{P}_t H_t^T + W_t N_t W_t^T)^{-1} \qquad (11)\\ \mu_t - \mathbf{x}^\star_t &= \bar{\mu}_t - \mathbf{x}^\star_t + K_t(\mathbf{z}_t - h(\mathbf{x}^\star_t, \mathbf{0}, \mathcal{A}) - H_t(\bar{\mu}_t - \mathbf{x}^\star_t))\\ P_t &= (I - K_t H_t)\bar{P}_t\,. \qquad (12)\end{aligned}$$

Note that the covariance P_t and the Kalman gain K_t depend, via the Jacobians, on $\mathbf{x}^\star_{0:t}$ and $\mathbf{u}^\star_{0:t-1}$ but not on the actual values of $\mathbf{u}_{0:t-1}$ and $\mathbf{z}_{1:t}$ (see (10), (11), (12)). Therefore they can be calculated before the robot starts operation (a-priori).

The minimization of the expected deviation from the desired trajectory (3) in the LQR controller can also be solved a-priori, linearly relating the control command $\mathbf{u}_t$ to the estimated state μ_t via a feedback matrix L_t:

$$\mathbf{u}_t - \mathbf{u}^\star_t = L_t(\mu_t - \mathbf{x}^\star_t)\,. \qquad (13)$$

L_t depends on the a-priori known Jacobians (9) and the weight matrices (3) and is derived explicitly in [4].

As described above, we express the whole navigation algorithm, which consists of executing a motion command, making an observation, localizing, and selecting the next motion command depending on the localization, by linear functions. For this linear navigation system, van den Berg *et al.* [2] proved that the a-priori joint distribution of $\mathbf{x}_t$ and μ_t is a Gaussian

$$\begin{bmatrix}\mathbf{x}_t - \mathbf{x}^\star_t\\ \mu_t - \mathbf{x}^\star_t\end{bmatrix} \sim \mathcal{N}(\begin{bmatrix}0\\0\end{bmatrix}, R_t = \begin{bmatrix}S_t & \mathrm{Cov}(\mathbf{x}_t, \mu_t)\\ \mathrm{Cov}(\mathbf{x}_t, \mu_t)^T & U_t\end{bmatrix}),$$

and that its covariance R_t can be computed recursively by

$$R_0 = \begin{bmatrix}P_0 & 0\\ 0 & 0\end{bmatrix},\ R_t = F_t R_{t-1} F_t^T + G_t \begin{bmatrix}M_t & 0\\ 0 & N_t\end{bmatrix} G_t^T,$$

with

$$\begin{aligned}F_t &= \begin{bmatrix}A_t & B_t L_{t-1}\\ K_t H_t A_t & A_t + B_t L_{t-1} - K_t H_t A_t\end{bmatrix}, \qquad (14)\\ G_t &= \begin{bmatrix}V_t & 0\\ K_t H_t V_t & K_t W_t\end{bmatrix}. \qquad (15)\end{aligned}$$

R_t only depends on a-priori known variables, namely the Jacobians (9), the Kalman gain (11), and the feedback matrix (13). These variables can be computed a-priori since we linearize the models around the (a-priori known) desired states $\mathbf{x}^\star_{0:T}$ and not around the (a-priori unknown) estimates $\mu_{0:T}$ as it is done for example in the extended Kalman filter [16].

B. Evaluation of the Deviation Guarantee

In the linearized system we can efficiently check whether the deviation guarantee (5) holds. Let S_t^{pos} be the part of the a-priori covariance S_t of $p(\mathbf{x}_t - \mathbf{x}^\star_t \mid \mathcal{A})$ corresponding to the position of the robot. The length $a_t(\mathcal{A})$ of the major semi-axis of the $p_{\min}$-confidence ellipsoid of S_t^{pos} can be calculated using

$$a_t(\mathcal{A}) = c\sqrt{\lambda_t}\,, \qquad (16)$$

where λ_t is the largest eigenvalue of S_t^{pos} and c is a scaling factor corresponding to $p_{\min}$ via the regularized Gamma function as described in [2]. If $a_t(\mathcal{A}) \leq d_{\max}$, then the $p_{\min}$-ellipsoid of S_t^{pos} is inside a circle with radius $d_{\max}$ and guarantee (5) holds for the linearized system. Note that this test is a conservative approximation and is exact if the $p_{\min}$-ellipsoid is a sphere.

C. Visibility of Landmarks

For robots with a limited sensor range, the non-linearity of the sensor model at the border of the field of view of the robot induces a large discrepancy between the real model and its linearization. To avoid this, we conservatively estimate for every landmark if the robot will observe it at time t, following the approach of Vitus and Tomlin [15]: We consider a landmark as visible at time t only if it is $p_{\min}$-*visible*, i.e., it is visible from every position inside the $p_{\min}$-ellipsoid of S_t^{pos} around $\mathbf{x}_t^{\star\text{pos}}$. If a landmark configuration satisfies the guarantee (5) when only $p_{\min}$-visible landmarks are observed, it also satisfies it when using all visible landmarks. Note that for general state spaces and fields of view (like in the framework proposed in [11]), checking if a certain landmark is $p_{\min}$-visible is highly non-trivial and could for example be done using a sampling based approach.

However, for the two dimensional case (i.e., $\mathbf{x}_t^{\text{pos}} = [x\ y]$) and for a robot with a circular field of view with radius r, there exists a closed form solution to checking if a given landmark l is $p_{\min}$-visible. Consider the $p_{\min}$-ellipse of S_t^{pos} centered at the origin of the coordinate system. We apply a principal axis transformation on the ellipse so that afterwards its semi-axes lie on the axes of the coordinate system. Applying this transformation on S_t^{pos} yields a diagonal matrix $S_t^{\text{pos}\prime} = \mathrm{diag}(\lambda_1, \lambda_2)$ with the diagonal elements identical to the eigenvalues of the matrix. Any point $\mathbf{x}' = [x'_1,\ x'_2]^T$ on the transformed ellipse $\mathcal{E}$ can then be described as $\mathbf{x}' = c \cdot \mathrm{diag}(\sqrt{\lambda_1}, \sqrt{\lambda_2}) \cdot \mathbf{x}$ for a point $\mathbf{x} = [x_1,\ x_2]^T$ on the unit circle $\mathcal{C}$ and the scaling factor c from (16). To check if

landmark l is $p_{\min}$-visible given an a-priori estimate $(\mathbf{x}_t^\star, S_t)$, we apply the principal axis transformation also on the relative position $(l - \mathbf{x}_t^\star)$ of the landmark, resulting in $l' = [l'_1, l'_2]$. Checking if l is $p_{\min}$-visible means checking if

$$\begin{aligned}
&\max_{\mathbf{x}' \in \mathcal{E}} \|\mathbf{x}' - l'\|_2 \leq r \\
\Leftrightarrow &\max_{\mathbf{x} \in \mathcal{C}} \|c \operatorname{diag}(\sqrt{\lambda_1}, \sqrt{\lambda_2})\mathbf{x} - l'\|_2 \leq r \\
\Leftrightarrow &\max_{x_1 \in [-1,1],\, \operatorname{sgn} \in \{-1,1\}} \big((c\sqrt{\lambda_1}x_1 - l'_1)^2 \\
&\qquad + (c\sqrt{\lambda_2} \operatorname{sgn} \sqrt{1 - x_1^2} - l'_2)^2 - r^2\big) \leq 0\,.
\end{aligned}$$

Applying a distinction of cases for sgn, we set the derivative with respect to x_1 of the function inside the max-operator to 0 and reorder the resulting equation. This yields a quartic term, which can be solved analytically in an efficient way.

V. Incremental Landmark Placement Algorithm

Our landmark placement approach aims at minimizing the number of landmarks that have to be placed for the deviation guarantee to hold. Since the dimensionality of the search space grows with the length of the trajectory, in general, globally searching for the optimal landmark configuration is computationally intractable. However, using an incremental placement algorithm, we can efficiently find an approximate solution to the landmark placement problem.

A. Landmark Placement for the Linearized System

In a first stage, our algorithm employs the linearized system to incrementally place landmarks. Considering linearized Gaussian models has the advantage that the a-priori distributions can be efficiently calculated analytically. The objective of our approach is to minimize the number of landmarks needed for the deviation guarantee to hold on the whole trajectory $\mathbf{x}_{0:T}^\star$. We approximate this minimum by maximizing the number of time steps every single landmark guarantees (5). Let

$$t_{\max}(\mathcal{A}, \mathbf{x}_{0:T}^\star) = \max\{t \mid a_s(\mathcal{A}) \leq d_{\max}\ \forall s \leq t\} \tag{17}$$

be the maximum time step for which the landmark set $\mathcal{A}$ guarantees (5) in the linearized system for the first part of the trajectory $\mathbf{x}_{0:t_{\max}}^\star$. In every iteration our algorithm adds the landmark $l^\star$ which maximizes $t_{\max}$ to the already selected set of landmarks $\mathcal{A}$. In some cases, one additional landmark is not enough to increase $t_{\max}$. This can happen for example if $d_{\max}(\mathbf{x}_{t_{\max}+1}^\star)$ is chosen considerably smaller than $d_{\max}(\mathbf{x}_{t_{\max}}^\star)$. In these cases, the algorithm selects the landmark which minimizes $a_{t_{\max}}(\mathcal{A})$ instead. Reducing $a_{t_{\max}}(\mathcal{A})$ increases the likelihood that in the next step a landmark can be found which increases $t_{\max}$ again (see (17)). Algorithm 1 describes the incremental landmark placement for the linearized system.

B. Monte Carlo Validation

In a second stage, we check the computed landmark configuration $\mathcal{A}$ for the deviation guarantee via Monte Carlo simulation using the real (possibly non-linear, non-Gaussian) models. This is necessary to account for approximation errors due to the linearization and the Gaussian assumption. The Monte Carlo simulation samples robot states $\mathbf{x}_{0:T}$, controls $\mathbf{u}_{0:T}$, and observations $\mathbf{z}_{1:T}$ of the landmarks in $\mathcal{A}$ and counts the number of time steps t in which $d^{\text{pos}}(\mathbf{x}_t, \mathbf{x}_t^\star) \leq d_{\max}(\mathbf{x}_t^\star)$, as required in guarantee (5). Averaging over numerous runs yields an estimate p_{MC} of $p_{\min}$ for which the deviation guarantee in the real system holds. If $p_{\text{MC}} < p_{\min}$, one can use arbitrary heuristics to place additional landmarks. For example, one could run our algorithm for increased values of $p_{\min}$ or decreased values of $d_{\max}$.

Algorithm 1 Landmark Placement for the Linearized System

Input: trajectory $\mathbf{x}_{0:T}^\star$, space of landmark locations $\mathcal{L}$
Output: landmark configuration $\mathcal{A}$

```
A ← ∅
t ← 0
while t < T do
    l* ← argmax_{l∈L} t_max(A ∪ {l}, x*_{0:T})
    t* ← t_max(A ∪ {l*}, x*_{0:T})
    if t* = t then
        l* ← argmin_{l∈L} a_{t_max}(A ∪ {l})
    end if
    A ← A ∪ {l*}
    t ← t*
end while
return A
```

VI. Experimental Results

We evaluated our landmark placement algorithm and compared it to other landmark placement approaches in extensive experiments both in simulation and with a real robot.

A. Setup of the Simulation Experiments

In the simulation experiments, we considered a wheeled robot with a differential drive. For self-localization we simulated three different kinds of sensors detecting uniquely identifiable landmarks: a *range-only* sensor, measuring only the distance to the landmarks, a *bearing-only* sensor, measuring only the relative angle between the robot and the landmarks, and a *range-and-bearing* sensor, measuring both. In this setup, the differential drive motion model and all sensor models have non-linear components. For all sensors, we assumed a circular field of view around the robot with radius $2m$. We evaluated our algorithm on five navigation tasks T1-T5 for all three sensor models, resulting in 15 experiments. Fig. 2 shows the landmarks our algorithm computed for the three sensor models in the first task T1, together with a-priori and a-posteriori distributions. Fig. 3 depicts the landmark configurations and a-priori distributions for the other four tasks T2-T5 for a range-only sensor. For all trajectories, we set $p_{\min} = 99\%$ and $d_{\max} = 0.5m$. For the pick-and-place task T5, we changed $d_{\max}$ to $0.2m$ in the pick up zone and in the deposit zone (gray rectangles). Because of the high accuracy necessary to fulfill this task, we simulated a more precise robotic system than for the other tasks, i.e., we scaled down the noise values.

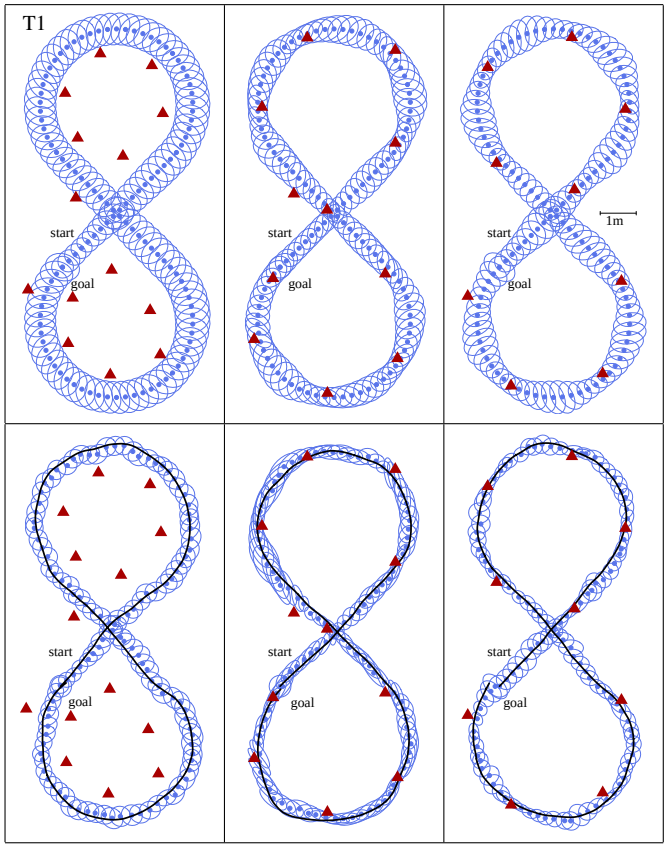

Fig. 2. The Landmark configurations (red triangles) our algorithm computed for a figure eight trajectory T1 for three different sensor models: range-only (left), bearing-only (middle) and range-and-bearing (right). The blue points and ellipses in the upper row correspond to the means and 99% covariance ellipses of the a-priori distributions, and in the lower row to the a-posteriori distributions of simulated sample runs. The true positions of the robot in the sample runs are depicted as black lines.

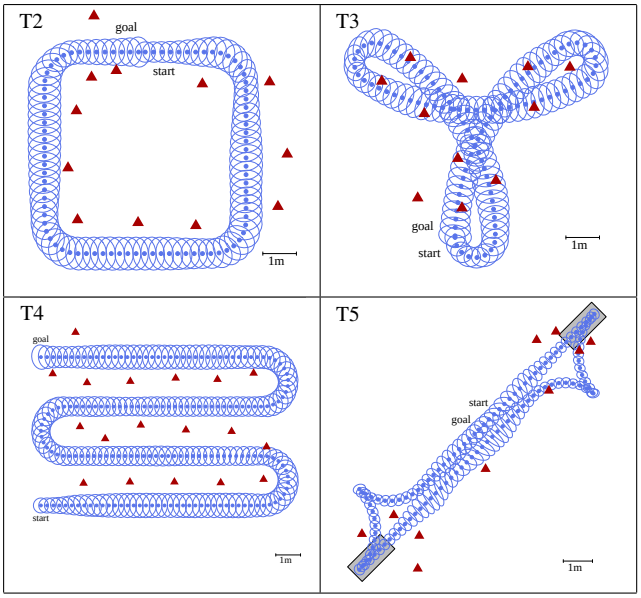

Fig. 3. The Landmark configurations (red triangles) our algorithm computed for four sample trajectories T2-T5 using a range-only sensor. T2 is a square, T3 a curved shape, T4 a sweeping trajectory, and T5 a *pick-and-place* task. The blue points and ellipses correspond to the means and 99% covariance ellipses of the a-priori distributions. In T5, the pick up zone and the deposit zone are marked as gray rectangles.

B. Restriction of the Search Space

To implement the argmax and argmin operators in Algorithm 1, we used an insight gained from empirical evaluations. In the experiments, $l^\star$ typically lay inside the field of view of $\mathbf{x}_t^\star$. Therefore, we restricted the search space for the optimizations to this area. As $t_{\max}$ and $a_{t_{\max}}$ usually had several local optima inside the field of view of $\mathbf{x}_t^\star$, we did a search on a discrete grid covering this region followed by a fine search with Powell's method [10] around the optimum on the grid.

To evaluate this strategy, we also ran grid searches on the full environment of the robot followed by Powell optimization steps. In all 15 experiments, Algorithm 1 with the full searches did not select less landmarks than our implementation with the restricted search spaces. On an Intel® Core™ i7 2.8GHz with 12GB RAM, the execution time of our algorithm averaged over the 15 experiments was 138 *sec* utilizing the restricted search spaces and 617 *sec* using the full searches.

C. Influence of the Sensor Model

As can be seen in Fig. 2, the amount of landmarks our algorithm computes and their configuration strongly depend on the chosen sensor model. For the range-only sensor, the landmarks tend to be further away from the trajectory than for the other two sensor models. The numbers of landmarks needed are stated in the first row of Table I. Also the results of the Monte Carlo simulations in our algorithm varied strongly for the different sensor models. In every Monte Carlo simulation, we performed 1000 simulated runs of the robot to get an estimate p_{MC} of $p_{\min}$. For all trajectories and all sensor models, the values of p_{MC} for the landmark sets our approach computed are stated in the fifth row of Table I. For the range-only sensor, p_{MC} is considerably above the intended value of 99% in all tasks. For the range-and-bearing sensor, p_{MC} is slightly below 99% in the pick-and-place task and for the bearing-only sensor, p_{MC} is below 99% in three of the five tasks. These results indicate that the non-linear components of the range measurements are less critical for landmark placement than the ones of the bearing measurements.

D. Comparison to other Landmark Placement Strategies

For comparison, we evaluated three other landmark placement methods. Each method starts with a minimum number of landmarks and successively increases the number (or density) of landmarks until it finds a set for which the guarantee in the linearized system holds.

- *On trajectory* places landmarks equidistant on the desired trajectory.
- *On grid* places a landmark in the center of each cell of a regular grid. Starting with one cell covering the whole environment, the cell size is decreased at every iteration until the deviation guarantee holds. For efficiency, landmark positions which are outside the field of view of all states $\mathbf{x}_{0:T}^\star$ on the desired trajectory are not used.
- *Random* successively places landmarks at randomly chosen positions observable from the desired trajectory.

The numbers of selected landmarks and the values of p_{MC} for all landmark placement strategies are stated in Table I.

TABLE I

NUMBERS OF SELECTED LANDMARKS AND RESULTS OF MONTE CARLO SIMULATIONS

	Range-only sensor					Bearing-only sensor					Range-and-bearing sensor				
	T1	T2	T3	T4	T5	T1	T2	T3	T4	T5	T1	T2	T3	T4	T5
Number of landmarks															
Our approach	14	12	11	18	10	11	9	7	16	7	9	8	6	13	5
On trajectory	–	–	25	–	58	41	–	12	–	23	12	9	10	17	7
On grid	48	32	38	56	23	26	19	17	30	18	20	20	17	25	16
Random	108	63	62	138	62	75	66	51	88	31	38	29	38	37	15
p_{MC}															
Our approach	0.999	0.998	0.999	0.999	0.999	0.979	0.978	0.991	0.994	0.826	0.999	0.997	0.998	0.994	0.986
On trajectory	–	–	0.996	–	0.962	0.353	–	0.955	–	0.773	0.996	0.999	0.983	0.999	0.980
On grid	0.999	0.999	0.999	0.999	0.998	0.997	0.981	0.995	0.999	0.931	0.996	0.999	0.995	0.999	0.999
Random	0.999	0.999	0.999	0.999	0.998	0.996	0.996	0.999	0.999	0.999	0.999	0.999	0.996	0.999	0.999

Dashes in the table indicate that no valid landmark configuration could be found. For all experiments, our approach placed less landmarks than the other approaches. The *on trajectory* method is the best method after ours for the *range-and-bearing* sensor, measured in the number of landmarks placed. However, for the other two sensor models, the *on trajectory* method was not always able to find a landmark configuration which satisfied the guarantee in the linearized system. For this method, especially the non-linearities in the *bearing-only* sensor model resulted in low values for p_{MC}.

E. Experiments with a Real Robot

To further validate the simulation experiments, we evaluated a landmark set our algorithm generated also on the real e-puck robot [9] depicted in Fig. 1. As a range-and-bearing sensor, we utilized a webcam pointing upwards detecting uniquely identifiable ARToolkit markers [5] attached to the ceiling. We considered the navigation task T1 scaled down to suit the miniature size of our robot (diameter $75mm$) and the lower ceiling. Scaling the task by the factor 0.08 yields $d_{\max} = 0.04m$. To evaluate the deviation $d^{\text{pos}}(\mathbf{x}_t, \mathbf{x}_t^\star)$ of the e-puck robot from its desired trajectory, we obtained the reference positions $\mathbf{x}_t$ from a MotionAnalysis motion capture system with four digital Raptor-E cameras. During 20 autonomous runs, $d^{\text{pos}}(\mathbf{x}_t, \mathbf{x}_t^\star)$ was below $d_{\max}$ in 99.7% of the time steps.

VII. CONCLUSIONS

In this paper, we presented a landmark placement method guaranteeing a bound on the maximum deviation of the robot from its trajectory with high confidence. During a placement stage, our approach approximates the real motion model and sensor model by their linearizations to efficiently evaluate the guarantee. In a subsequent validation stage, we apply a Monte Carlo simulation using the real system dynamics to check if the selected landmark set satisfies the deviation guarantee also for the possibly non-linear models. In contrast to other approaches, our algorithm is customizable to specific robotic systems and navigation tasks and inherently chooses the number of landmarks needed. In extensive experiments, we demonstrated that our method outperforms other approaches.

ACKNOWLEDGMENTS

This work has partly been supported by the German Research Foundation (DFG) within the Research Training Group 1103. The authors would like to thank Axel Rottmann for his implementation of the Powell algorithm.

REFERENCES

[1] M. Beinhofer, J. Müller, and W. Burgard. Near-optimal landmark selection for mobile robot navigation. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2011.

[2] J. van den Berg, P. Abbeel, and K. Goldberg. LQG-MP: Optimized path planning for robots with motion uncertainty and imperfect state information. In *Proc. of Robotics: Science and Systems (RSS)*, 2010.

[3] J. van den Berg, S. Patil, R. Alterovitz, P. Abbeel, and K. Goldberg. LQG-based planning, sensing, and control of steerable needles. In *Proc. of the Int. Workshop on the Algorithmic Foundations of Robotics (WAFR)*, 2010.

[4] D. Bertsekas. *Dynamic Programming and Optimal Control*. Athena Scientific, 2nd edition, 2000.

[5] M. Billinghurst and H. Kato. Collaborative augmented reality. *Communications of the ACM*, 45(7):64–70, 2002.

[6] H.F. Durrant-Whyte, D. Pagac, B. Rogers, M. Stevens, and G. Nelmes. Field and service applications - an autonomous straddle carrier for movement of shipping containers. *IEEE Robotics & Automation Magazine*, 14:14–23, 2007.

[7] J.-S. Gutmann, E. Eade, P. Fong, and M. Munich. A constant-time algorithm for vector field SLAM using an exactly sparse extended information filter. In *Proc. of Robotics: Science and Systems (RSS)*, 2010.

[8] D.B. Jourdan and N. Roy. Optimal sensor placement for agent localization. *ACM Trans. Sen. Netw.*, 4(3):1–40, 2008.

[9] F. Mondada, M. Bonani, X. Raemy, J. Pugh, C. Cianci, A. Klaptocz, S. Magnenat, J.-C. Zufferey, D. Floreano, and A. Martinoli. The e-puck, a Robot Designed for Education in Engineering. In *Proc. of the Conf. on Autonomous Robot Systems and Competitions*, 2009.

[10] M. Powell. An efficient method for finding the minimum of a function of several variables without calculating derivatives. *The Computer Journal*, 7(2):155–162, 1964.

[11] C. Pradalier and S. Sekhavat. "Localization Space": a framework for localization and planning, for systems using a sensor/landmarks module. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2002.

[12] P. Sala, R. Sim, A. Shokoufandeh, and S. Dickinson. Landmark selection for vision-based navigation. *IEEE Transactions on Robotics and Automation*, 22(2):334–349, 2006.

[13] H. Strasdat, C. Stachniss, and W. Burgard. Which landmark is useful? Learning selection policies for navigation in unknown environments. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2009.

[14] S. Thrun. Finding landmarks for mobile robot navigation. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 1998.

[15] M. Vitus and C. Tomlin. Sensor placement for improved robotic navigation. In *Proc. of Robotics: Science and Systems (RSS)*, 2010.

[16] G. Welch and G. Bishop. An introduction to the Kalman filter. Technical report, 1995.

[17] S. Zhang, L. Xie, and M.D. Adams. Entropy based feature selection scheme for real time simultaneous localization and map building. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2005.