Belief Space Planning using Landmark Density Information

Nordlöf Jonas, Hendeby Gustaf and Axehill Daniel

Belief Space Planning using Landmark Density Information

Jonas Nordlöf*[†], Gustaf Hendeby[†], and Daniel Axehill[†]
*Div. of C4ISR, Swedish Defence Research Agency (FOI), Linköping, Sweden
Email: jonas.nordlof@foi.se
[†] Dept. of Electrical Engineering, Linköping University, Linköping, Sweden
e-mail: {firstname.lastname}@liu.se

***Abstract*—An approach for belief space planning is presented, where knowledge about the landmark density is used as prior, instead of explicit landmark positions. Having detailed maps of landmark positions in a previously unvisited environment is considered unlikely in practice. Instead, it is argued that landmark densities should be used, as they could be estimated from other sources, such as ordinary maps or aerial imagery. It is shown that it is possible to use virtual landmarks to approximate the landmark density to solve the presented problem. This approximation is also shown to give small errors during evaluation. The approach is tested in a simulated environment, in conjunction with an *extended information filter* (EIF), where the computed path is shown to be superior compared to other alternative paths used as benchmarks.**

***Index Terms*—Belief Space Planning, Extended Information Filter (EIF), Virtual landmarks**

I. Introduction

High positioning accuracy is often required in order for complex tasks in autonomous systems to be completed successfully. In structured environments, this is often accomplished with artificial positioning infrastructure, such as *Global Navigation Satellite System* (GNSS), *radio-frequency* (RF) transmitters, WiFi, or artificial landmarks. The problem of attaining good positioning accuracy without access to such support systems is much more difficult.

Estimation techniques such as *simultaneous localization and mapping* (SLAM) for solving these problems have in recent years reached a good maturity [1]. However, they can still fall short in some scenarios, *e.g.*, due to absence of informative landmarks or due to low sensor visibility [2]. For instance, if an autonomous platform is to follow a path to a position and the path moves the platform through a smokey, dark or an otherwise featureless area, the positioning accuracy will be reduced, regardless of how well the SLAM algorithms perform.

One way to limit this problem is to take the positioning uncertainty into consideration in the planning phase of a task [3]. The general problem of incorporating the positioning uncertainty in planning is often referred to as *belief space planning* or, in a SLAM setting, *active SLAM* [4].

Traditionally, planning algorithms usually aim to minimize the length of the travelled path or the amount of energy consumed. In belief space planning, algorithms are instead often designed to minimize some measure of uncertainty. One of the most well known approaches for belief space planning is the *belief roadmap* (BRM) [5]. The drawback of this approach is that it assumes a known map of all detectable landmarks to be given, and is thereby not applicable for SLAM problems. Furthermore, detailed maps of landmark positions are unlikely to exist for previously unvisited or dynamic environments. Different ways of addressing this problem have been proposed. In [6], the authors present an approach where the map created by a Pose SLAM algorithm [7] is used to update the estimated uncertainty in belief space planning. The drawback of this approach is that it requires all nodes to be visited beforehand, which is also impractical in some situations.

The problem of not having prior knowledge of landmark positions has also been acknowledged in the field of active SLAM. There, several *receding horizon controller* (RHC) approaches [8, 9] have been proposed. These utilize already detected landmarks in order to improve positioning performance by replanning and finding loop-closures in an existing path. However, these approaches require the environment to be at least partially explored before any improvements on the original path can be made.

This paper introduces a planning strategy which aims to go beyond the current limitations of having prior knowledge of the environment in belief space planning by using landmark densities rather than the explicit position of landmarks. These densities are represented by virtual landmarks which seamlessly fit into SLAM frameworks. More explicitly, the introduction of virtual landmarks removes the requirement of prior knowledge of landmarks positions which exists in some belief space planning algorithms, such as the belief road map, thereby making them more suitable for SLAM applications.

The contribution of this paper is three-fold:

- a new framework for path planning under uncertainty utilizing landmark densities;
- an approximate problem formulation using virtual landmarks, which is practically solvable; and
- a numerical evaluation of, first, the density approximation itself and, second, an algorithm which uses the presented approximation to solve the path planning problem.

The justification of using landmark densities is based on the presumption that the landmark density in an area can be

This paper was supported by research projects at the Swedish Defence Research Agency (FOI) funded by the Swedish Armed Forces.

remotely estimated more easily than the position of landmarks, without requiring the area to be explored in detail. One way of obtaining these densities could be semantic segmentation of aerial images [10, 11] or online maps. Although it is possible to detect landmarks in aerial images, these do not necessarily correspond to landmarks detectable from a ground platform. Instead, this information is more suitable for estimating the landmark density in an area. Furthermore, in a dynamic environment the actual locations of landmarks might get obstructed or even change, *e.g.* due to seasonal changes such as snow, which further argues for the use of landmark densities.

The outline of this paper is as follows: The problem is formulated in Section II. The *extended information filter* (EIF) for a SLAM setting (EIF-SLAM) is introduced in Section III, this section also introduces the virtual landmarks which is the foundation for the proposed solution. Section IV describes how the A^*-algorithm can be adapted to use this approach for path planning and solving the optimization problem while Section V includes an evaluation of the approach.

II. Problem formulation

This paper considers the problem of planning a path from a starting position to a goal position while minimizing position uncertainty along the path and at the goal position, without a map of the area. That is, there are no known landmarks available, but the densities of landmarks are assumed known. This problem is formulated as an optimization problem as outlined below.

A. Motion and measurement model

Consider an autonomous platform with the state vector $x_t = [(x_t^p)^\top, m_t^\top]^\top \in \mathcal{X}$, where x_t^p describes the pose of the platform and m_t is the map at time t and $\mathcal{X}$ is the state space. Let $\mathcal{I}_t$ denote the information matrix of this state vector. The autonomous platform is assumed to move according to

$$x_{t+1} = f(x_t, u_t, w_t), \quad w_t \sim \mathcal{N}(0, Q_t), \quad u_t \in \mathcal{U}(x_t), \tag{1}$$

where $f(\cdot,\cdot,\cdot)$ is the system dynamics function, u_t is the control input, w_t is the process noise and $\mathcal{U}(x_t)$ is the set of feasible control inputs at state x_t.

The autonomous platform is also assumed to have a sensor with limited range, *e.g.* a LiDAR, used to detect landmarks. If the landmark m^i is within range of the sensor at time t, a measurement y_t^i of the landmark is obtained as

$$y_t^i = h(x_t) + \epsilon_t^i, \quad \epsilon_t^i \sim \mathcal{N}(0, R_t), \tag{2}$$

where $h(\cdot)$ is the measurement function and ϵ_t^i is measurement noise.

B. Landmark densities

In the presented problem, the number of landmarks and their positions are unknown. However, the landmarks are assumed to be distributed with density $\rho(m)$, where m is a position in the area of interest.

C. Performance measure

The goal of the considered problem is to generate a path which optimizes a performance measure, which defines the success criteria of the platform. This performance measure typically includes a part associated with the uncertainty of the platform, another with the execution of the planned path. Denote this performance measure $J(\cdot,\cdot,\cdot)$ where the first part, $J_\mathcal{I}(\cdot)$, relates to the position uncertainty, and measures how well the system knows its state. The second part, $J_p(\cdot,\cdot)$, is a state-dependent penalty related to the planned position of the platform and its movement. These two parts are then weighted together with the parameters α and β, giving

$$J(\mathcal{I}_{1:T}^{-1}, x_{1:T}^p, u_{1:T}) = \alpha J_\mathcal{I}(\mathcal{I}_{1:T}^{-1}) + \beta J_p(x_{1:T}^p, u_{1:T}), \tag{3}$$

where $\bullet_{1:T}$ represent the set $\{\bullet_t\}_{t=1}^T$.

Several different choices of $J_\mathcal{I}(\cdot)$ have been proposed [12, 13]. Some of the more popular ones are the sum of all determinants, $\det(\cdot)$, or the trace, $\text{tr}(\cdot)$, along the path, while the travelled distance is usually used for $J_p(\cdot,\cdot)$.

D. Optimization problem

Given a starting position x_S^p, a goal position x_G^p, and a set of landmarks $\{m^i\}_{i=1}^M$, the problem of belief space planning can be solved by finding a policy π that minimizes the following optimization problem:

$$\begin{aligned}
\underset{\pi, T}{\text{minimize}} \quad & \mathbb{E}\Big(J(\mathcal{I}_{1:T}^{-1}, x_{1:T}^p, u_{1:T})\Big) \\
\text{subject to} \quad & x_t = f(x_{t-1}, u_{t-1}, w_{t-1}), \\
& y_t^i = h(x_t) + \epsilon_t^i, \quad \forall i \in \{1, ..., M_t\}, \\
& u_t = \pi(x_t^p) \\
& \mathcal{I}_t = \Lambda(\mathcal{I}_{t-1}, y_t^{1:M_t}, x_t^p, u_t) \\
& x_0^p = x_S^p, \quad x_T^p = x_G^p, \\
& \epsilon_t^i \sim \mathcal{N}(0, R_t), \\
& w_t \sim \mathcal{N}(0, Q_t), \\
& x_t \in \mathcal{X}, \quad u_t \in \mathcal{U}(x_t)
\end{aligned} \tag{4}$$

where $y_t^{1:M_t}$ are measurements of the M_t landmarks observed at time t, and $\Lambda(\cdot,\cdot,\cdot)$ describes how the position accuracy develops over time given measurements y_t^i and input u_t and is explained in more detail in Section III-A.

The expectation is evaluated with respect to the measurement noise, process noise, and landmark observations as these are not given during planning but have a large effect on the information gained. Calculating the expectation is in practice very difficult. Maximum likelihood assumptions on motions and measurements are often used to limit computations in regards to the measurement and process noise [14, 15]. However, there are no practical ways in the literature of dealing with all possible variations of landmark positions. This is the problem that the proposed approach aims to solve.

III. State estimation and SLAM

The following sections describe the algorithm used for state estimation, *i.e.* $\Lambda(\cdot,\cdot,\cdot,\cdot)$ in (4), and how only knowing the density of landmarks can be handled. For notational convenience the EIF-SLAM algorithm is presented, however the EIF and EKF variants can be used interchangeably.

A. EIF-SLAM

The presented SLAM algorithm uses an extended information filter based on [16]. A full derivation of the EIF-SLAM algorithm [17] is outside the scope of this paper, but for clarity, the algorithm is outlined below. For an introduction of the SLAM problem, the reader is referred to [18, 19].

The EIF-SLAM algorithm uses a state-space vector $\hat{x}_{t|t} = [(\hat{x}^p_{t|t})^\top, \hat{m}^\top_{t|t}]^\top$, where $\hat{x}^p_{t|t}$ and $\hat{m}_{t|t}$ are the estimated pose and map, respectively, and an information matrix $\mathcal{I}_{t|t}$ to estimate the position of the platform and detected landmarks.

The time update of the EIF can then be formulated as

$$\hat{x}_{t+1|t} = f(\hat{x}^p_{t|t}, u_t, 0) \tag{5a}$$

$$\mathcal{I}_{t+1|t} = \left(F_t \mathcal{I}^{-1}_{t|t} F^\top_t + G_t Q_t G^\top_t\right)^{-1}, \tag{5b}$$

where F_t and G_t are the Jacobians of $f(\cdot,\cdot,\cdot)$ with respect to x_t and w_t, respectively, evaluated in $x_t = \hat{x}_{t-1|t-1}$ and $w_t = 0$. The measurement update can be formulated as

$$\hat{\epsilon}^i_t = y^i_t - h(\hat{x}_{t|t-1}) + H^i_t \hat{x}_{t|t-1} \tag{6a}$$

$$\mathcal{I}^i_t = \left(H^i_t\right)^\top R^{-1}_t H^i_t \tag{6b}$$

$$\mathcal{I}_{t|t} = \mathcal{I}_{t|t-1} + \sum_{i=1}^{M} \mathcal{I}^i_t \tag{6c}$$

$$\hat{x}_{t|t} = \mathcal{I}^{-1}_{t|t}\left(\mathcal{I}_{t|t-1}\hat{x}_{t|t-1} + \sum_{i=1}^{M}\left(H^i_t\right)^\top R^{-1}_t \hat{\epsilon}^i_t\right), \tag{6d}$$

where H^i_t is the Jacobian of the measurement function $h(\cdot)$ for the ith landmark evaluated in $x_t = \hat{x}_{t|t-1}$.

B. Information approximation

The EIF-SLAM algorithm assumes distinct landmarks that can be observed. However, when predicting the future state during planning, new landmarks do not become available. Therefore, to take future landmarks into consideration, the density of landmarks is used to predict the expected information gain of the future state.

For clarity, let $H_t(x^p_t, m)$ denote the measurement matrix H_t calculated from the sensor position x^p_t to a landmark at position $m = [x_m, y_m]^\top$. The information gain of a single landmark at m calculated from a sensor position x^p_t is

$$\mathcal{I}(x^p_t, m) = \left(H_t(x^p_t, m)\right)^\top R^{-1}_t H_t(x^p_t, m). \tag{7}$$

Now, assume the region of interest is populated with infinitely dense landmarks with information gain

$$d\mathcal{I}(x^p_t, m) = \rho(m)\mathcal{I}(x^p_t, m)dm, \tag{8}$$

where the information is compensated for the density of the landmarks by $\rho(\cdot)$. The resulting information from an area Ω then becomes

$$\mathcal{I}_\Omega(x^p_t) = \int_{m\in\Omega} \rho(m)\mathcal{I}(x^p_t, m)dm. \tag{9}$$

If the density is a Dirac delta function, this converges to (7) in the limit.

Due to the lack of knowledge of future landmarks, the expression (9) will be used to indicate information gains. The integral is, however, not realistic to compute in practice, hence an approximation is considered. First Ω is divided into K non-overlapping subregions Ω_k:

$$\begin{aligned}\mathcal{I}_\Omega(x^p_t) &= \int_{m\in\Omega} \rho(m)\mathcal{I}(x^p_t, m)dm \\ &= \sum_{k=1}^{K} \int_{m\in\Omega_k} \rho(m)\mathcal{I}(x^p_t, m)dm \end{aligned} \tag{10a}$$

The integral over the subregions Ω_k in (10a) could now be computed with numerical integration. However, this would require (10a) to be recomputed each time the sensor position x^p_t changes, which is computationally expensive. Instead, if the subregions Ω_k are small then $\mathcal{I}(x^p_t, m)$ will be almost constant over Ω_k, for a given x^p_t, and can be approximated by $\mathcal{I}(x^p_t, m^\star_k)$ for an $m^\star_k$ in the region. For simplicity, the centre point of Ω_k will be used as $m^\star_k$. Approximating the region with a single point allows the information gained to be represented in a convenient way in the EIF-SLAM algorithm. Note that, it is the measurement model, and its underlying geometry, that decides how large the regions Ω_k can be with reasonable approximation errors. Using this approximation we get:

$$\begin{aligned}\mathcal{I}_\Omega(x^p_t) &\approx \sum_{k=1}^{K}\left(\int_{m\in\Omega_k}\rho(m)dm\right)\mathcal{I}(x^p_t, m^\star_k) \\ &= \sum_{k=1}^{K}\Phi_k \mathcal{I}(x^p_t, m^\star_k) = \hat{\mathcal{I}}_\Omega \end{aligned} \tag{11}$$

where

$$\Phi_k = \int_{m\in\Omega_k}\rho(m)dm \tag{12}$$

is determined through, *e.g.*, numerical integration. As the subregion Ω_k becomes smaller, the approximation improves and in the limit the left hand side and right hand side of (11) coincide.

The density $\rho(\cdot)$ is assumed to be constant in time and known a priori, Φ_k can thereby be pre-computed for all regions. On the other hand, if we assume that $\rho(\cdot)$ is constant over Ω_k, the numerical integration of Φ_k in (12) could be replaced with a single point at, say, $m^\star_k$. However, this assumption does not necessarily hold as the density $\rho(\cdot)$ may be non-continuous or include large gradients. Since Φ_k can be pre-computed, the computational gain of this point approximation is also limited. Note that, accuracy improvements could possibly be attained by aligning subregions with transitions in $\rho(\cdot)$

since these transitions can be large. Additional information can possibly also be extracted by utilizing these transitions. This is, however, not done here.

Given a set of K virtual landmarks with corresponding approximate cumulative densities Φ_k, $k \in \{1, 2, ...K\}$, the modified measurement update of $\mathcal{I}_{t|t}$, in (6c), which incorporates these approximations is

$$\mathcal{I}_{t|t} = \mathcal{I}_{t|t-1} + \sum_{k=1}^{K} \Phi_k \mathcal{I}(x_t^p, m_k^\star). \tag{13}$$

This is more computationally efficient than replacing (6c) with (9) directly.

IV. Motion planning

The following sections describe how a motion planning algorithm is adjusted to solve the presented belief space planning problem.

A. Belief space planning

In belief space planning, the mean and covariance are used to represent the belief, or the distribution, of the current state at any given time, while the true state is considered known in ordinary motion planning. In order to solve (4), the distribution is predicted given maximum likelihood time and measurement updates. Furthermore, it is proposed in this paper that any observations of real landmarks are also exchanged with observations of virtual landmarks, as presented in Section III-B. Note that, the optimization problem in (4) does not need to be reformulated when approximating the landmark densities with virtual landmarks. Solving (4) is non-trivial even with the above approximation. To further simplify the problem, the state space is discretized allowing only a subset of the otherwise continuous set of positions for the platform. Several different algorithms can then be employed to, with minor modifications, solve the resulting graph-search problem, among others the well-known A*-algorithm [20] which is used in the following section.

B. A*-algorithm

After discretization of the state space, each possible position of the platform is represented by a node in a search graph, in which an index value j_t and a goal value g_t used in the A*-algorithm are stored. Given the performance measure $J(\cdot,\cdot,\cdot)$ in (4), the goal value, or *cost-to-come*, g_t can be computed as

$$g_t = J(\mathcal{I}_{1:t}^{-1}, x_{1:t}^p, u_{1:t}). \tag{14}$$

Note that, when $J(\cdot,\cdot,\cdot)$ is separable in time, *e.g.*

$$J(\mathcal{I}_{1:t}^{-1}, x_{1:t}^p, u_{1:t}) = J(\mathcal{I}_{1:t-1}^{-1}, x_{1:t-1}^p, u_{1:t-1}) + l(\mathcal{I}_t^{-1}, x_t^p, u_t), \tag{15}$$

then g_t can be recursively computed as

$$g_t = g_{t-1} + l(\mathcal{I}_t^{-1}, x_t^p, u_t), \tag{16}$$

where $l(\cdot,\cdot,\cdot)$ is the cost of u_t with the state x_t^p and uncertainty $\mathcal{I}_t^{-1}$. Given g_t, the index value j_t is often computed as

$$j_t = g_t + c_t(\mathcal{I}_t^{-1}, x_t^p, u_t), \tag{17}$$

where $c_t(\cdot,\cdot,\cdot)$ is a heuristic function. In order to preserve *admissibility*, $c_t(\cdot,\cdot,\cdot)$ should be a lower bound of the true *cost-to-go*, $c_t^*(\cdot,\cdot,\cdot)$, *i.e.*

$$\begin{aligned} c_t(\mathcal{I}_t^{-1}, x_t^p, u_t) &\leq c_t^*(\mathcal{I}_t^{-1}, x_t^p, u_t) \\ &= J(\mathcal{I}_{t+1:T}^{-1}, x_{t+1:T}^p, u_{t+1:T}). \end{aligned} \tag{18}$$

Care must also be taken when choosing $J_\mathcal{I}(\cdot)$ such that

$$J_\mathcal{I}(\mathcal{I}_{1:t}^{-1}) \geq J_\mathcal{I}(\mathcal{I}_{1:t-1}^{-1}), \tag{19}$$

otherwise optimality, given the linearizations in (5) and (6) and the approximation in (11), cannot be guaranteed [21]. Examples of measures that fulfil this requirement are the sum of the trace of $\mathcal{I}_t^{-1}$ along the path,

$$\sum_{t=1}^{T} \text{tr}(\mathcal{I}_t^{-1}), \tag{20}$$

or the maximum of the trace of $\mathcal{I}_t^{-1}$ along the path,

$$\max_t \text{tr}(\mathcal{I}_t^{-1}). \tag{21}$$

Obtaining a good lower bound of $J_\mathcal{I}(\cdot)$ is difficult, hence $0 \leq J_\mathcal{I}(\cdot)$ is used, yielding

$$c_t(\mathcal{I}_t^{-1}, x_t^p, u_t) = \beta \hat{J}_p(x_t^p, u_t), \tag{22}$$

where $\hat{J}_p(x_t^p, u_t)$ is a lower bound on $J_p(x_{t+1:T}^p, u_{t+1:T})$.

A difference between the standard A*-algorithm and the proposed algorithm is that instead of only propagating and saving the position (and its associated j_t and g_t values), the information matrix $\mathcal{I}_t$ must also be computed using the EIF-SLAM algorithm. For each step between a parent node and its child, the input signal u_t is first determined. The matrix $\mathcal{I}_t$ is then propagated to the child with the EIF-algorithm using the introduced virtual landmarks. The information matrix $\mathcal{I}_{t|t}$, calculated in (6c) of the EIF measurement update, is then used in the performance measure $J(\cdot,\cdot,\cdot)$. The complete modified A*-algorithm can be found in Algorithm 1.

V. Simulation study

The proposed approach is evaluated using simulations in two different experiments. The purpose of the first experiment is to verify the accuracy of (11), while the second experiment evaluates the performance of the proposed approach.

A. Simulation setup

The presented problem is currently being studied in a 2D environment. The platform is described by the state $x_t^p = [x_t^{p,x}, x_t^{p,y}, x_t^{p,\theta}]^\top$ at time t, where $x_t^{p,\theta}$ is the orientation of the platform and $x_t^{p,x}$ and $x_t^{p,y}$ are the x- and y-positions, respectively.

The movement of the platform follows a unicycle model which consists of a rotation (u_t^θ) followed by a translation (u_t^t), as given by

$$\begin{aligned} x_{t+1}^p &= f(x_t^p, u_t, w_t) \\ &= x_t^p + \begin{bmatrix} (u_t^t + w_t^t)\cos(x_t^\theta + u_t^\theta + w_t^\theta) \\ (u_t^t + w_t^t)\sin(x_t^\theta + u_t^\theta + w_t^\theta) \\ u_t^\theta + w_t^\theta \end{bmatrix}, \end{aligned} \tag{23}$$

Algorithm 1: The A*-algorithm modified for use with virtual landmark

```
Let $\mathcal{U}(\cdot)$ be the set of motions $u_t$ that connects a
node $n$ to all neighbouring nodes in the search graph.
$closedSet = \emptyset$
$openSet = \emptyset$
$openSet$.push($n_{init}$)
while $openSet \neq \emptyset$ do
    $n \leftarrow openSet$.pop()
    $closedSet$.push($x$)
    if $n = n_{goal}$ then
        return back_track($x$)
    end
    for $u_t \in \mathcal{U}(n.x^p_{t-1})$ do
        $x^p_t \leftarrow f(n.x^p_{t-1}, u_t, 0)$
        if $x^p_t \in closedSet$ then
            continue
        end
        $\mathcal{I}_t \leftarrow \Lambda(n.\mathcal{I}_{t-1}, y_t, x^p_t, u_t)$
        $x^p_{1:t} \leftarrow [n.x^p_{1:t-1}\ x^p_t]$
        $\mathcal{I}^{-1}_{1:t} \leftarrow [n.\mathcal{I}^{-1}_{1:t-1}\ \mathcal{I}^{-1}_t]$
        $u_{1:t} \leftarrow [n.u_{1:t-1}\ u_t]$
        $g \leftarrow J(\mathcal{I}^{-1}_{1:t}, x^p_{1:t}, u_{1:t})$
        $j = g + c_t(\mathcal{I}^{-1}_t, x^p_t, u_t)$
        $n_{child}.g \leftarrow g$
        $n_{child}.j \leftarrow j$
        $n_{child}.x^p_{t-1} \leftarrow x^p_t$
        $n_{child}.x^p_{1:t-1} \leftarrow x^p_{1:t}$
        $n_{child}.u_{1:t-1} \leftarrow u_{1:t}$
        $n_{child}.\mathcal{I}_{t-1} \leftarrow \mathcal{I}_t$
        $n_{child}.\mathcal{I}^{-1}_{1:t-1} \leftarrow \mathcal{I}^{-1}_{1:t}$
        if $x^p_t \notin openSet$ then
            $openSet$.push($n_{child}$)
        else
            if $n_{child}.g > g$ then
                $openSet$.push($n_{child}$)
            end
        end
    end
end
```

where the input $u_t = [u^\theta_t, u^t_t]^\top$ is affected by process noise $w_t = [w^\theta_t, w^t_t]^\top$ in each element and

$$w_t \sim \mathcal{N}\left(0, \begin{bmatrix} (2\pi/180)^2 & 0 \\ 0 & 5^2 \end{bmatrix}\right). \tag{24}$$

It is assumed that the simulated autonomous platform is using a LiDAR-sensor with range-bearing output and a limited range of 100 m. The obtained measurements $y^i_t = [y^{i,r}_t, y^{i,\theta}_t]^\top$ to any landmarks $m^i = [m^{i,x}, m^{i,y}]^\top$ within range are the relative range ($y^{i,r}_t$) and bearing ($y^{i,\theta}_t$), where $m^{i,x}$ and $m^{i,y}$ are the x- and y-coordinates of the landmark. This gives the measurement model

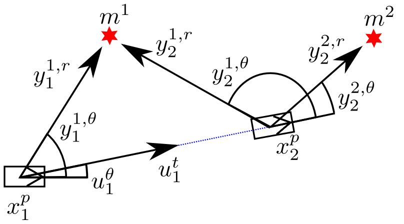

Fig. 1. The figure shows an example of how the positions of the platform (x^p_t), control signals ($u_t = [u^\theta_t, u^t_t]^\top$), landmarks ($m^i$) and measurements ($y^i_t = [y^{i,r}_t, y^{i,\theta}_t]^\top$) relate to each other.

$$\begin{aligned} y^i_t &= h(x^p_t, m^i) + \epsilon^i_t \\ &= \begin{bmatrix} \|[x^{p,x}_t, x^{p,y}_t]^\top - m^i\| \\ \text{atan2}(m^{i,y} - x^{p,y}_t, m^{i,x} - x^{p,x}_t) - x^{p,\theta}_t \end{bmatrix} + \epsilon^i_t, \end{aligned} \tag{25}$$

where atan2$(\cdot,\cdot)$ is the four-quadrant arctangent function and $\epsilon^i_t = [\epsilon^{i,r}_t, \epsilon^{i,\theta}_t]^\top$ is a vector containing the range noise $\epsilon^{i,r}_t$ and bearing noise $\epsilon^{i,\theta}_t$, where

$$\epsilon^i_t \sim \mathcal{N}\left(0, \begin{bmatrix} 0.2^2 & 0 \\ 0 & (\pi/180)^2 \end{bmatrix}\right). \tag{26}$$

The setup is illustrated in Fig 1.

For simplicity, the subregions Ω_k, used for approximating the landmark density, are equally sized square regions, but, in general, any space discretization or tessellation should be applicable. Furthermore, the variable Φ_k, in (12), is determined through numerical integration, *i.e.* Gauss quadrature formula [22], on the form

$$\Phi_k = \int_{m \in \Omega_k} \rho(m) dm \approx \sum_{n=1}^{N} \gamma_n \rho(s_n), \tag{27}$$

where the abscissas s_n and weights γ_n are tabulated values and $N = 400$.

B. Verifying the Information Approximation

In this section the information approximation introduced in (11) is evaluated, *i.e.* whether the centre point $m^\star_k$ of each subregion approximately gives the same information as the entire subregion Ω_k. The information gains from three different sets of virtual landmarks for an area Ω_T are compared with the integral (9). The area Ω_T is $40 \times 40\,\text{m}$ and is approximated by sets of 1×1, 2×2 and 3×3 virtual landmarks, with their information gain denoted $\hat{\mathcal{I}}_{1\times1}$, $\hat{\mathcal{I}}_{2\times2}$ and $\hat{\mathcal{I}}_{3\times3}$ respectively. An illustration of this area and the different subregions can be seen in Fig. 2.

Two different comparisons are performed. The first setup is designed to highlight the significance of the distance between the observer and the landmark region. The quality of the approximation is computed for distances between 21 and 60 m from the centre of the landmark area with the area right in front of the sensor. The second setup is designed to evaluate if the angle from the sensor platform to the landmark region affects the quality of the approximation. This is done with sensor locations on an arc 35 m from the centre of the landmark area.

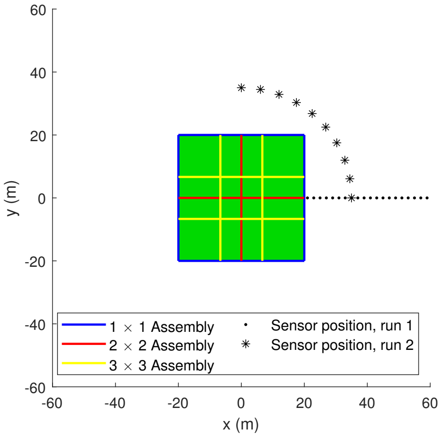

Fig. 2. Sensor positions for model verifications. The green square represents the area with a landmark density that is approximated with the three different virtual landmark sets.

This distance is chosen as it is considered to be on the limit of reasonable approximation errors. In both of these comparisons, the sensor is stationary and all sensor placements are evaluated independently.

For all positions, the relative error between the information matrix for the integral (9) and the three landmark sets are evaluated with the Frobenious norm, *i.e.*,

$$\frac{\|\mathcal{I}-\hat{\mathcal{I}}_{j\times j}\|_F}{\|\mathcal{I}\|_F}, \quad j=1,2,3, \tag{28}$$

where $\mathcal{I}$ is the integral (9). These norms can be seen in Fig. 3 and 4. The information gain $\mathcal{I}$ is computed using numerical integration in Wolfram Mathematica [23].

The assumption that $\mathcal{I}(x_t^p, m_k) \approx \mathcal{I}(x_t^p, m_k^\star)$ in the entire subregion Ω_k is expected to be less accurate when the sensor is closer to Ω_T. In these cases, $H_t(x_t^p, m_k)$ changes considerably within the region Ω_k, hence the approximation $H(x_t^p, m_k) \approx H(x_t^p, m_k^\star)$ is poor. As seen in Fig. 3, the errors for the 2×2 and 3×3 virtual landmark sets are small for all sensor placements and are considered to be reasonably low when the sensor platform is 25 m, or more, from the centre. The error of the 1×1 set is considered reasonably low when the sensor platform is placed at around 35 m, or more, from the virtual landmark.

In the second comparison, seen in Fig. 4, the error marginally increases when the sensor is close to the diagonal of the area. However, the approximation accuracy is considered to be reasonably low for all positions around the landmark area. That is, the accuracy is largely independent of the angle from the sensor platform to the landmark area.

C. Path planning performance

In this section, a reference path created with the proposed approach, here referred to as the *optimal path*, is compared to two other reference paths in a small environment. For evaluation, a simulated autonomous platform is executing each reference path for 100 different sets of landmarks, sampled from the underlying density. During these realizations, the position of the platform is tracked by the standard EIF-SLAM algorithm, presented in Section III-A, with measurements of sampled landmarks.

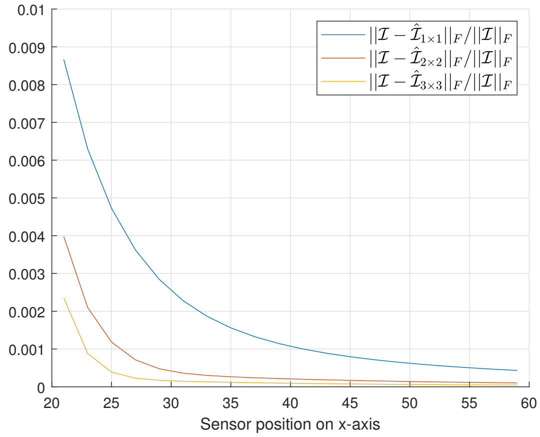

Fig. 3. Frobenious norm of the difference between $\mathcal{I}$ and the information gain of the different landmark sets normalized by the norm of $\mathcal{I}$ with different distances to the sensor position. $\mathcal{I}$ represents the information gain computed through numerical integration, $\hat{\mathcal{I}}_{1\times1}$, $\hat{\mathcal{I}}_{2\times2}$ and $\hat{\mathcal{I}}_{3\times3}$ represents the approximate information gain computed using the 1×1, 2×2 and 3×3 virtual landmark sets, respectively.

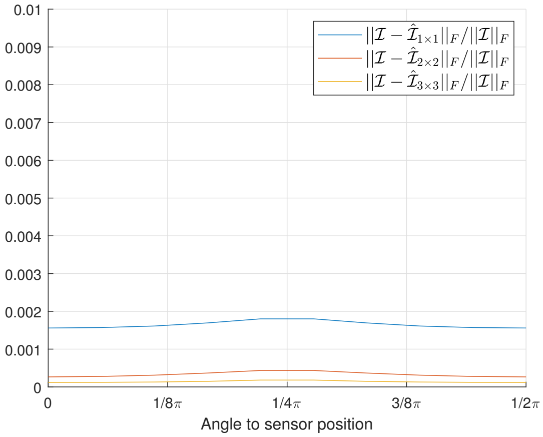

Fig. 4. Frobenious norm of the difference between $\mathcal{I}$ and the information gain and the different landmark sets normalized by the norm of $\mathcal{I}$ at different positions around the target area. $\mathcal{I}$ represents the information gain computed through numerical integration, $\hat{\mathcal{I}}_{1\times1}$, $\hat{\mathcal{I}}_{2\times2}$ and $\hat{\mathcal{I}}_{3\times3}$ represents the approximate information gain computed using the 1×1, 2×2 and 3×3 virtual landmark sets, respectively.

The environment contains two regions with different landmark densities, indicated by the light and dark green areas in Fig. 5, where the light green area has a landmark density of 0.0024 landmarks/m^2 and the dark green has 0.0002 landmarks/m^2. The size of each subregion Ω_k is $40\times 40\,\mathrm{m}$ and are illustrated in Fig. 5 together with the weights Φ_k. In order to avoid bias in the comparison, the area is assumed to be free of any obstacles.

During planning of the optimal path, each node in the search graph represents one point in a uniform grid, where each point is 20 m apart along the x and y axis, and the input space $\mathcal{U}(\cdot)$ is the set of input signals that moves the platform to the

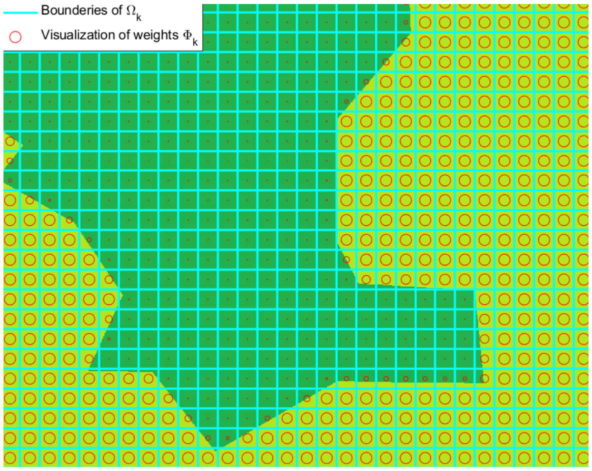

Fig. 5. The figure shows the simulated environment with the subregions Ω_k the weights Φ_k, where a large value of Φ_k is represented with a larger circle.

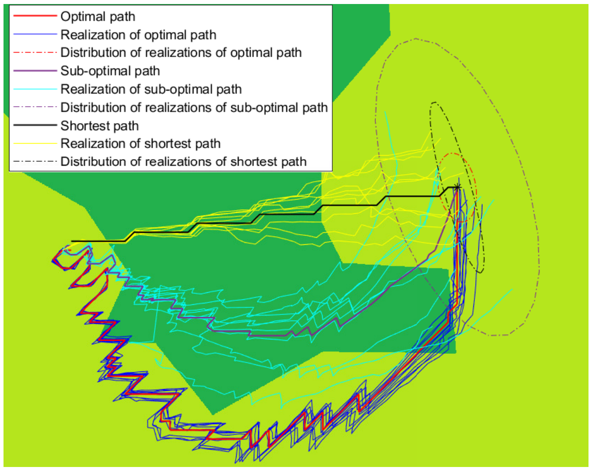

Fig. 6. The figure shows the three different paths used in the comparison. The path created with the proposed approach (red), the shortest path (black) and a mixture of these (purple). Ten realizations are shown for each path. The ellipses at the end of the path represents the goal point distribution of these.

surrounding eight points. The performance measure $J(\cdot,\cdot,\cdot)$ is chosen as the trace of the covariance of the pose estimate of the autonomous platform, that is $\alpha = 1$, $\beta = 0$ and

$$J_{\mathcal{I}}(\mathcal{I}_{1:t}^{-1}) = \max_t \operatorname{tr}\left(\left[\mathcal{I}_{t|t}^{-1}\right]^p\right), \tag{29}$$

where $\left[\mathcal{I}_{t|t}^{-1}\right]^p$ is the 3×3 sub-matrix in the uncertainty matrix $\mathcal{I}_{t|t}^{-1}$ related to x_t^p. In order to reduce the risk of violating the linearization constraints introduced by the EIF [24], the minimum measurement range of the LiDAR-sensor is set to

$$\max(10, 1.96 \times \lambda_{\max}^p), \tag{30}$$

where $\lambda_{\max}^p$ is the largest eigenvalue of the uncertainty matrix $\left[\mathcal{I}_{t|t-1}^{-1}\right]^p$.

The first of the two reference paths is the *shortest path* between the start and goal position. The second path is a mixture between the shortest and the optimal path, and is referred to as the *sub-optimal path*. Each point $\pi_t^{so} = [\pi_t^x, \pi_t^y]$ in the sub-optimal path is calculated as

$$\pi_t^{so} = \zeta\pi_t^o + (1-\zeta)\pi_\tau^s, \tag{31}$$

where π_t^o and π_τ^s denote points along the optimal, π^o, and shortest path, π^s, and τ is an interpolated time index such that π^o and π^s are of equal length. Different values of ζ has been tested, but in the following analysis ζ is equal to 0.5.

The first ten realizations of each of the three paths are illustrated in Fig. 6. These trajectories represent the ground truth position of the simulated platform while following the path. The proposed algorithm has found a reference path which avoids the area with the low landmark density, reaching the maximum value of (29) in the end node. As a consequence, the realizations of the optimal path has a much smaller spread, *i.e.* deviations from the planned path, than both the shortest path and the sub-optimal path.

It can also be seen that the proposed approach produces a jagged path. Our interpretations of these jagged motions in the optimal solution is that they increase the chances of revisiting previously seen landmarks, thereby facilitating possibilities of local loop-closures.

However, as the platform progresses along the path, the path becomes less jagged. One possible explanation is that, due to an increased uncertainty of the platform, $\lambda_{\max}^p$ is the dominating term in (30) which increases the minimum measurement range. Hence, fewer landmarks are observed in each step which makes local loop-closures less beneficial as they cannot cancel the process noise added in each extra step. At the end of the path the minimum measurement range is $39.77\,\mathrm{m}$. Furthermore, errors in the beginning of the path will be propagated forward in all consecutive steps, thereby having larger impact on the final results.

During path execution, the performance measure is considerably lower for all realizations of the optimal path than both the sub-optimal path and the shortest path, as seen in Fig. 7. With varying values of ζ, $0 \leq \zeta \leq 1$, the same results are attained, although with smaller differences as ζ goes to 1. Note that, due to scaling, the figure shows the maximum square root of the trace and not the maximum trace.

The goal point error, *i.e.* the distance between the estimated position and the ground truth position at the goal point, can be seen in Fig. 8. In general, apart from a few outliers, the errors for the optimal path is lower than both the sub-optimal and the shortest path.

VI. Conclusion

Current belief space planning problem formulations often rely on known landmarks, which are in many cases not available in practice. Therefore, this paper proposed a problem formulation where these landmarks are replaced by landmark densities. To solve this new problem, a virtual landmark approximation was derived. The accuracy of this approximation was evaluated in a simulation study, where the virtual landmark approximation was shown to provide results with small differences compared to solving the exact problem.

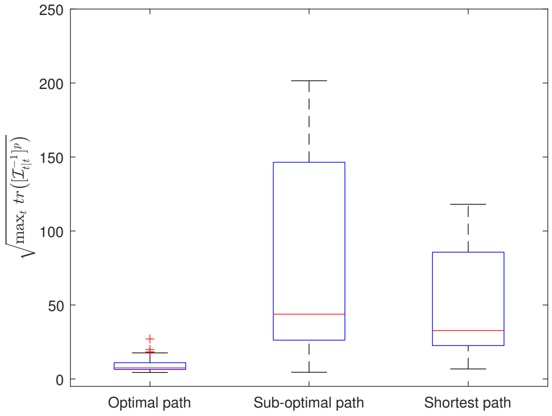

Fig. 7. Boxplots over the performance measure during 100 path realizations for the three different paths. It can be seen that the performance measure of the path created with the proposed approach is considerably lower, and with lower variance, than both the sub-optimal path and the shortest path.

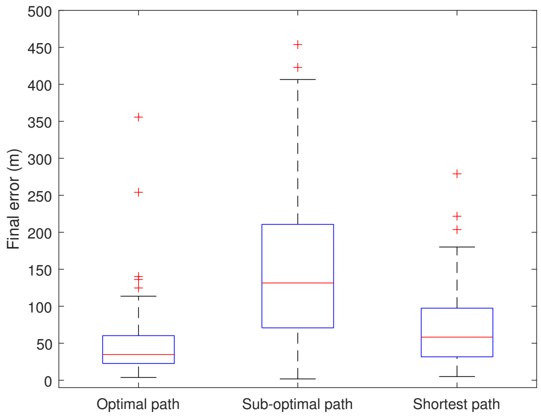

Fig. 8. Boxplots over the goal point error during 100 path realizations of the three different paths. It can be seen that the error for the path created with the proposed approach is lower than both the sub-optimal and the shortest path.

The proposed virtual landmark concept was also integrated in a path planning algorithm aimed to minimize a measure of uncertainty. This algorithm was tested in simulations and gave from an application perspective a reasonable result and outperformed the alternative paths in terms of the objective function value.

It was shown in this paper that subregions close to the platform need to be small to give small approximation errors, but at large distances the benefit of using small subregions is diminishable. In order to increase the computational performance and reduce the approximation error an adaptive grid method for the approximation will be developed in the future. A benefit of the proposed approach is that, when observed, real landmarks can seamlessly replace virtual landmarks in the EIF-SLAM algorithm. In future work, we will exploit this property during replanning in a receding horizon approach.

REFERENCES

[1] M. R. U. Saputra, A. Markham, and N. Trigoni, "Visual SLAM and structure from motion in dynamic environments: A survey," *ACM Computing Surveys*, vol. 51, no. 2, pp. 1–36, 2018.

[2] J. M. Santos, M. S. Couceiro, D. Portugal, and R. P. Rocha, "A sensor fusion layer to cope with reduced visibility in SLAM," *J. Intell. Robot. Syst.*, vol. 80, no. 3-4, pp. 401–422, Jan. 2015.

[3] F. Celeste, F. Dambreville, and J.-P. Le Cadre, "Optimal path planning using cross-entropy method," in *Proc. 9th Int. Conf. Inf. Fusion*. Florence, Italy: IEEE, Jul. 2006.

[4] A. Kim and R. M. Eustice, "Active visual SLAM for robotic area coverage: Theory and experiment," *Int. J. Robot. Res.*, vol. 34, no. 4-5, pp. 457–475, Nov. 2014.

[5] S. Prentice and N. Roy, "The belief roadmap: Efficient planning in belief space by factoring the covariance," *Int. J. Robot. Res.*, vol. 28, no. 11-12, pp. 1448–1465, 2009.

[6] R. Valencia and J. Andrade-Cetto, "Path planning in belief space with pose SLAM," in *Mapping, Planning and Exploration with Pose SLAM*. Springer, 2018, pp. 53–87.

[7] V. Ila, J. M. Porta, and J. Andrade-Cetto, "Information-based compact pose SLAM," *IEEE Trans. Robot.*, vol. 26, no. 1, pp. 78–93, 2009.

[8] C. Leung, S. Huang, and G. Dissanayake, "Active SLAM using model predictive control and attractor based exploration," in *Proc. IEEE/RSJ Int. Conf. on Intell. Robots Syst.*, Beijing, China, Nov. 2006, pp. 5026–5031.

[9] V. Indelman, L. Carlone, and F. Dellaert, "Planning in the continuous domain: A generalized belief space approach for autonomous navigation in unknown environments," *Int. J. Robot. Res.*, vol. 34, no. 7, pp. 849–882, 2015.

[10] D. Marmanis, J. D. Wegner, S. Galliani, K. Schindler, M. Datcu, and U. Stilla, "Semantic segmentation of aerial images with an ensemble of CNSS," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences,*, vol. 3, pp. 473–480, 2016.

[11] K. Chen, K. Fu, M. Yan, X. Gao, X. Sun, and X. Wei, "Semantic segmentation of aerial images with shuffling convolutional neural networks," *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 2, pp. 173–177, 2018.

[12] M. L. Rodríguez-Arévalo, J. Neira, and J. A. Castellanos, "On the importance of uncertainty representation in active SLAM," *IEEE Trans. Robot.*, vol. 34, no. 3, pp. 829–834, 2018.

[13] C. Yang, L. Kaplan, and E. Blasch, "Performance measures of covariance and information matrices in resource management for target state estimation," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 48, no. 3, pp. 2594–2613, 2012.

[14] S. Miller, Z. Harris, and E. Chong, "Coordinated guidance of autonomous UAVs via nominal belief-state optimization," in *Proc. Am. Control Conf.*, St. Louis, USA, Jul. 2009, pp. 2811 – 2818.

[15] R. Platt, R. Tedrake, L. Kaelbling, and T. Lozano-Perez, "Belief space planning assuming maximum likelihood observations," in *Robotics: Science and Systems VI*. Robotics: Science and Systems Foundation, Jun. 2010.

[16] T. Kailath, A. H. Sayed, and B. Hassibi, *Linear Estimation*. New Jersey: Prentice Hall, 2000.

[17] S. Thrun, Y. Liu, D. Koller, A. Y. Ng, Z. Ghahramani, and H. Durrant-Whyte, "Simultaneous localization and mapping with sparse extended information filters," *Int. J. Robot. Res.*, vol. 23, no. 7-8, pp. 693–716, Aug. 2004.

[18] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: part i," *IEEE Robot. Autom. Mag.*, vol. 13, no. 2, pp. 99–110, Jun. 2006.

[19] T. Bailey and H. Durrant-Whyte, "Simultaneous localization and mapping (SLAM): part II," *IEEE Robot. Autom. Mag.*, vol. 13, no. 3, pp. 108–117, Sep. 2006.

[20] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Trans. Sys. Sci. and Cybern*, vol. 4, no. 2, pp. 100–107, 1968.

[21] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*. Prentice Hall, 2003, vol. 82.

[22] A. Ghizzetti and A. Ossicini, *Quadrature formulae*. Akademie Verlag, 1970.

[23] Wolfram Research, Inc., "Mathematica, Version 12.0," Champaign, IL, 2019.

[24] R. Sim, "Stabilizing information-driven exploration for bearings-only SLAM using range gating," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Edmonton, Canada, Aug. 2005.