2018 IEEE International Conference on Robotics and Automation (ICRA)
May 21-25, 2018, Brisbane, Australia

Efficient Active SLAM based on Submap Joining, Graph Topology and Convex Optimization

Yongbo Chen[1,2], Shoudong Huang[1], Robert Fitch[1], and Jianqiao Yu[2]

Abstract— **The active SLAM problem considered in this paper aims to plan a robot trajectory for simultaneous localization and mapping (SLAM) as well as for an area coverage task with robot pose uncertainty. Based on a model predictive control (MPC) framework, these two problems are solved respectively by different methods. For the uncertainty minimization MPC problem, based on the graphical structure of the 2D feature-based SLAM, a non-convex constrained least-squares problem is presented to approximate the original problem. Then, using variable substitutions, it is further transformed into a convex problem, and then solved by a convex optimization method. For the coverage task considering robot pose uncertainty, it is formulated and solved by the MPC framework and the sequential quadratic programming (SQP) method. In the whole process, considering the computation complexity, we use linear SLAM, which is a submap joining approach, to reduce the time for planning and estimation. Finally, various simulations are presented to validate the effectiveness of the proposed approach.**

I. INTRODUCTION

Active SLAM is a decision making problem where a robot's trajectory is chosen both to improve mapping and localization results, and at the same time, to perform other tasks such as coverage or exploration. The active SLAM problem presents a well-known dilemma for the activity of the robot in an unknown space [1]; it is difficult to strike a balance between visiting new places and revisiting known areas to reduce map uncertainty.

There are several known approaches for active SLAM, such as the MPC framework [2] and the partially observably Markov decision process formalism [3]. These methods all attempt to select the best future action from a set of alternatives. The two main questions in active SLAM, then, are: (i) how to evaluate the effect of a future action and (ii) how to select the best action.

The direct idea of evaluating actions in terms of estimation accuracy is to obtain the information matrices of the estimate vector after executing the future actions and then to compare a certain metric of the information matrices. The Theory of Optimal Experimental Design (TOED) [4], including *A-opt*, *D-opt*, and *E-opt*, is used commonly in active SLAM. A comparison of these optimality criteria is presented in [5]. In [6], the authors show that only *D-opt* retains monotonicity during the exploration phase of an active SLAM algorithm for the linearized framework. Thus, in most situations, the *D-opt* metric, which is to maximize the determinant of the information matrix, is the best criterion to use.

Obtaining future information matrices accurately in the case of unknown future measurements is difficult. In [7], the future information matrix is obtained by zero-innovation extended Kalman filter (EKF) and extended Information Filter (EIF) prediction. In [8], expectation-maximization (EM) and a Gauss-Newton (GN) approach are applied together to solve this problem. In [9], a conservative sparse information space is used to reduce the computational complexity of the *D-opt* objective function of the candidate action.

Selecting the best future action is an optimal control problem, whose global optimal solution is usually hard to obtain. Considering the large size of the search space, some researchers choose future waypoints from a small subset of locations. This approach leads the problem into the discrete optimization domain; an example is frontier-based exploration [10]. Other approaches can find locally optimal policies in continuous-space planning under uncertainty [11]. Active SLAM has also been addressed using path planning algorithms such as RRT* [12], D* [13], and potential information fields [14]. Finding an optimal solution for active SLAM is still an open problem in general.

In this paper, the MPC method is applied to greatly reduce the problem difficulty and the planning time. The objective functions employed include the popular *D-opt* optimality criterion as well as area coverage, taking into account the uncertainties involved. For the *D-opt* MPC problem, because the prediction of the future information matrix is a difficult and time-consuming process, the graphical structure of the 2D feature-based SLAM is analyzed. Then, the original problem can be approximated as a non-convex constrained least-squares problem. By using new variables, the problem can be further transformed into a convex problem. For the coverage problem with uncertainty, we use the same MPC framework and the SQP method considering its highly-nonlinear, highly-nonconvex properties. In order to improve run-time performance, the linear submap joining approach [15] is used in the SLAM process. Finally, simulations are presented to validate the algorithm.

The main contributions of this work are as follows:

- The application of submap joining (linear SLAM) in active SLAM (Section V), which greatly improves the algorithm's real-time ability. Results in Section VI-B.1 show that using submaps is efficient for planning.
- Using the incidence matrices of the 2D feature-based

[1]Yongbo Chen Yongbo.Chen@student.uts.edu.au, Shoudong Huang Shoudong.Huang@uts.edu.au and Robert Fitch Robert.Fitch@uts.edu.au are with Faculty of Engineering and Information Technology, University of Technology, Sydney, Ultimo, NSW, Australia

[2]Yongbo Chen and Jianqiao Yu jianqiao@bit.edu.cn are with School of Aerospace Engineering, Beijing Institute of Technology, Beijing, China

SLAM graph as well as the strong positive correlation between the weighted number of spanning trees and the *D-opt* criterion, we simplify the original objective function, shown in Section IV. Compared with the original objective function, the new problem is much easier to solve with a good approximation. The compared results are shown in Section VI-B.2.

- A relaxation approach and convex method are applied to obtain a solution to the simplified problem (Sections IV-D and IV-E). This method finds sub-optimal solutions quickly when the look-ahead step is large. Simulation results are given in Section VI-B.2.
- A new formulation of area coverage considering the robot pose uncertainty, especially when submap joining based SLAM is used, is shown in Section III-B and V.

These contributions make it possible for efficiently solving robot coverage problems in unknown environments, which have many potential applications such as surveillance and search & rescue.

II. MODEL AND PROBLEM STATEMENT

A. Vehicle model

The kinematic equations of the robot moving at a constant velocity V in a 2D environment can be written as:

$$\begin{aligned} \begin{bmatrix} V\Delta T \\ 0 \end{bmatrix} &= R_k^T(\boldsymbol{x}_{k+1}^v - \boldsymbol{x}_k^v) + \begin{bmatrix} \delta x_k \\ \delta y_k \end{bmatrix}, \\ \omega_k &= \frac{\theta_{k+1}-\theta_k}{\Delta T} + \delta\omega_k \end{aligned} \tag{1}$$

where ΔT is a discrete time interval, $(\boldsymbol{x}_k^{vT}, \theta_k) = (x_k, y_k, \theta_k)$ is the pose of the robot at the k-th step, $R_k = \begin{pmatrix} \cos(\theta_k) & -\sin(\theta_k) \\ \sin(\theta_k) & \cos(\theta_k) \end{pmatrix} \in SO(2)$ is the rotation matrix of the k-th pose of the robot, ω_k is the angular velocity of the robot at the k-th step, δx_k, δy_k and $\delta\omega_k$ are the discrete time noises of the coordinates and angular velocity and are assumed to be zero-mean Gaussian.

B. Sensor model

There are many types of sensors used in SLAM tasks. In this paper, a simplified sensor model is used:

$$\boldsymbol{Z}_k^{fi} = R_k^T(\boldsymbol{x}^{fi} - \boldsymbol{x}_k^v) + \begin{bmatrix} w_x^k \\ w_y^k \end{bmatrix}, \tag{2}$$

where $\boldsymbol{Z}_k^{fi}$ is the observed value of the i-th feature at k-th step, $\boldsymbol{x}^{fi}$ is the coordinate of the i-th feature, w_x^k, w_y^k are the noises of the sensor in the x and y axes and are assumed to be zero-mean Gaussian.

C. Problem statement

The goal of this paper is to present a solution framework that allows a robot to select optimal/sub-optimal control inputs to perform two specific tasks (uncertainty minimization and coverage). The environment is assumed to contain a number of point features with unknown locations. The robot can observe features within its sensor range. The boundary of the area of interest to be covered and mapped is known. The problem is to cover the area of interest as quickly as possible, while performing 2D point feature based SLAM continuously with accurate localization and mapping results.

III. MPC FRAMEWORK FOR UNCERTAINTY MINIMIZATION TASK AND COVERAGE TASK

The uncertainty minimization task and the coverage task both need to be performed. It is likely that, in most situations, these two tasks will lead to conflicting results. The uncertainty minimization task will lead the robot to visit known space to obtain more measurements. On the contrary, if we only consider the coverage task, the robot will try its best to avoid loop-closure, which leads to large uncertainty in the map and localization (and the robot will get lost). Hence, we need to consider these tasks jointly.

A. MPC framework for uncertainty minimization task

The problem is considered in the framework of MPC by an L-step look-ahead. The objective function is minimized over the time horizon $[k, k+L]$. Similar to [8], the objective function of the uncertainty minimization task is based on the generalized belief at the L-th planning step:

$$\begin{aligned} &J_k(u_{k:k+L-1}) = f_J(gb(\boldsymbol{X}_{k+L})) = -log\left(det(I_{k+L}(\boldsymbol{X}_{k+L}^{opt}))\right) \\ &\boldsymbol{X}_{k+L}^{opt} = \underset{\boldsymbol{X}_{k+L}}{argmin} \ -log(p(\boldsymbol{X}_{k+L}|\boldsymbol{B}(k,L), \boldsymbol{Z}_{k+1:k+L}) \\ &\boldsymbol{B}(k,L) = \boldsymbol{Z}_{1:k}, u_{0:k-1}, u_{k:k+L-1} \end{aligned}, \tag{3}$$

where $u_{0:k+L-1}$ denotes the $k+L$ control inputs $u_j = \omega_j,\ j = 0,\ \cdots,\ k+L-1$, $\boldsymbol{X}_{k+L}$ is the real state vector including the $k+L-1$ poses and the coordinates of the features, $gb(\boldsymbol{X}_{k+L})$ is the Gaussian belief of $\boldsymbol{X}_{k+L}$, $gb(\boldsymbol{X}_{k+L}) \approx \mathcal{N}(\boldsymbol{X}_{k+L}^{opt}, I_{k+L}(\boldsymbol{X}_{k+L}^{opt})^{-1})$, $\boldsymbol{Z}_{1:k}$ and $\boldsymbol{Z}_{k+1:k+L}$ denote the observed values from step 1 to k and step $k+1$ to $k+L$, respectively.

Using Bayes' theorem, we have:

$$\begin{aligned} &p(\boldsymbol{X}_{k+L}|\boldsymbol{Z}_{1:k}, u_{0:k-1}, \boldsymbol{Z}_{k+1:k+L}, u_{k:k+L-1}) \\ =&\frac{p(\boldsymbol{Z}_{k+1:k+L}|\boldsymbol{X}_{k+L}, \boldsymbol{B}(k,L))p(\boldsymbol{X}_{k+L}|\boldsymbol{B}(k,L))}{p(\boldsymbol{Z}_{k+1:k+L}|\boldsymbol{B}(k,L))}. \end{aligned} \tag{4}$$

We now take the assumption that the prior $p(\boldsymbol{Z}_{k+1:k+L}|\boldsymbol{B}(k,L))$ is uninformative. This assumption is fairly standard in inference, see [8]. The above equation (5) can thus be rewritten as:

$$\begin{aligned} &p(\boldsymbol{X}_{k+L}|\boldsymbol{Z}_{1:k}, u_{0:k-1}, \boldsymbol{Z}_{k+1:k+L}, u_{k:k+L-1}) \\ \propto&p(\boldsymbol{Z}_{k+1:k+L}|\boldsymbol{X}_{k+L}, \boldsymbol{B}(k,L))p(\boldsymbol{X}_{k+L}|\boldsymbol{B}(k,L)) \end{aligned}. \tag{5}$$

The future measurement $\boldsymbol{Z}_{k+1:k+L}$ is unknown. It is a probabilistic event. If the j-th feature is outside the sensor range from the actual position of the robot, it will not be observed. In order to solve this problem, we assume the measurement $\boldsymbol{Z}_{k+1:k+L}$ is perfect (zero-innovation). Using the Markov property, we obtain the predicted state vector $\boldsymbol{X}_{k+L}^{opt}$:

$$\begin{aligned} \boldsymbol{X}_{k+L}^{opt} =& \underset{\boldsymbol{X}_{k+L}}{argmin} \\ -log(p(\boldsymbol{X}_k|\boldsymbol{Z}_{1:k}, u_{0:k-1})&p(\boldsymbol{X}_{k+1:k+L}|\boldsymbol{X}_k, u_{k:k+L-1})) \end{aligned}. \tag{6}$$

Equation (6) includes two parts. The first part is a classical SLAM problem, and the second is a prediction process with zero-innovation probabilistic measurement $\boldsymbol{Z}_{k+1:k+L}$. Assuming that the SLAM result is $\boldsymbol{X}_k^{opt}$, the predicted future pose $\boldsymbol{X}_{k+L}^{opt}$ will be:

$$\boldsymbol{X}_{k+L}^{opt} = \begin{bmatrix} \boldsymbol{X}_k^{opt} \\ \boldsymbol{x}_{k+1}^{opt} \\ \boldsymbol{x}_{k+2}^{opt} \\ \vdots \\ \boldsymbol{x}_{k+L}^{opt} \end{bmatrix} = \begin{bmatrix} \boldsymbol{X}_k^{opt} \\ f_v(\boldsymbol{x}_k^{opt}, u_k) \\ f_v(\boldsymbol{x}_{k+1}^{opt}, u_{k+1}) \\ \vdots \\ f_v(\boldsymbol{x}_{k+L-1}^{opt}, u_{k+L-1}) \end{bmatrix}, \quad (7)$$

where $f_v(\star)$ is the motion equation shown in (1) and $\boldsymbol{x}_i^{opt}, i = 1,2,...,k+L$ is the predicted pose at step i.

Using the *D-opt* optimality criterion, the optimization problem for uncertainty minimization will be:

$$\begin{aligned} &\min_{u_{k:k+L-1}} \quad f_a(u_{k:k+L-1}) = -log(det(I_{k+L}(\boldsymbol{X}_{k+L}^{opt}))) \\ &s.t. \boldsymbol{X}_{k+L}^{opt} = (\boldsymbol{X}_k^{opt}, \cdots, f_v(\boldsymbol{x}_{k+L-1}^{opt}, u_{k+L-1}))^T \\ &\quad |u_{k+i}| \le C, \quad i = 0, \cdots, L-1 \end{aligned} \quad . \quad (8)$$

where C is the control limitation.

B. Coverage task under uncertainty based on MPC

As the robot moves, the area covered will increase based on its actual trajectory and sensing range. However, only an estimated position (with uncertainty) is available from the SLAM result, so firstly we need to formulate an expression for area covered under uncertainty.

At the k-th step, the major and minor axes of the 95% confidence ellipse $S_{95\%}$ of a robot position $\hat{\boldsymbol{x}}_k^{opt} \sim \mathcal{N}(\bar{\boldsymbol{x}}_k^{opt}, I^{-1})$ is $2\sqrt{5.991}\lambda_1(I^{-1})$ and $2\sqrt{5.991}\lambda_2(I^{-1})$, where $\lambda_1(I^{-1})$ and $\lambda_2(I^{-1})$ are respectively the major and secondary eigenvalues of its covariance matrix. The orientations of the axes of the ellipse are the eigenvectors of the covariance matrix. The range of the sensor is set as R_s. If the discrete coordinates of the points in the confidence ellipse $S_{95\%}$ are $(x_i^s, y_i^s), i = 1, ...,$ the coordinates $(x_i^c, y_i^c), i = 1, ...$ of the bound of the covered area S_i^c at the i-th step in the worst case will be (Fig. 1):

$$\begin{aligned} x_i^c &= \frac{l_s - R_s}{l_s}\bar{x}_k^{opt} + \frac{R_s}{l_s}x_i^s \\ y_i^c &= \frac{l_s - R_s}{l_s}\bar{y}_k^{opt} + \frac{R_s}{l_s}y_i^s \\ l_s &= \sqrt{(\bar{x}_k^{opt} - x_i^s)^2 + (\bar{y}_k^{opt} - y_i^s)^2} \end{aligned} \quad , \quad (9)$$

where $\bar{\boldsymbol{x}}_k^{opt} = (\bar{x}_k^{opt}, \bar{y}_k^{opt})^T$.

The coverage task is also built based on the MPC framework. The objective function is defined as:

$$\begin{aligned} f_c(u_{k:k+L_c-1}) &= \begin{cases} \frac{1}{f_A(A_{add})} & A_{add} \neq \varnothing \\ \min\limits_{i=1,...,N_r}(l_i^c) & A_{add} = \varnothing \end{cases} \\ A_{add} &= \left(\bigcup_{i=1}^{k+L_c} S_i^c - \bigcup_{i=1}^{k} S_i^c \right) \bigcap Space \\ l_i^c &= \sum_{j=1}^{L_c} \sqrt{(\bar{x}_{k+j}^{opt} - x_i^r)^2 + (\bar{y}_{k+j}^{opt} - y_i^r)^2} \end{aligned} \quad , \quad (10)$$

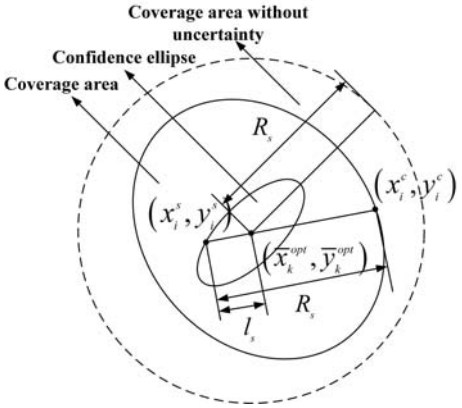

Fig. 1. The coverage area under uncertainty

where $f_A(\star)$ is a function that computes the area of $\star$, L_c is the number of the look-ahead steps, (x_i^r, y_i^r) is the centroid of the i-th uncovered area, N_r is the number of remaining uncovered areas, *Space* means the whole planning space, A_{add} is the new covered space executing L_c look-ahead control inputs, and l_i^c represents the minimum distance between the estimated position of the robot and the centroids of the uncovered areas.

We now formulate the MPC problem for the coverage task:

$$\begin{aligned} &\min_{u_{k:k+L_c-1}} \quad f_c(u_{k:k+L_c-1}) \\ &s.t. \boldsymbol{X}_{k+L_c}^{opt} = (\boldsymbol{X}_k^{opt}, \cdots, f_v(x_{k+L-1}^{opt}, u_{k+L_c-1}))^T. \\ &\quad |u_{k+i}| \le C, \quad i = 0, \cdots, L_c - 1 \end{aligned} \quad (11)$$

C. Solution framework for two MPC problems

Because the coverage MPC problem is a highly-nonlinear problem with a non-analytical formula, it is difficult to compute its optimal solution. We propose to use the SQP approach to compute a sub-optimal result. Then, for the uncertainty minimization MPC problem, we first approximate it by the graph topology and solve it using convex optimization.

D. Decision making for control switching

Because the physical meanings and the units of the two objective functions are significantly different, it is difficult to find good weights for combining the two objective functions. Thus we proposed to use a switching mechanism. We use a threshold on the SLAM uncertainty to decide which strategy to implement, either minimizing uncertainty or maximizing coverage. To avoid frequent switching between the two strategies when the uncertainty level is close to the threshold, we propose to use two thresholds instead of one, and the control action will be combined when the uncertainty level is between the two thresholds. This scheme is defined as:

$$\begin{aligned} u_r &= \begin{cases} u_c & Index_1 \le C_1^{index} \\ c_a u_c + c_c u_a & Index_1 \in (C_1^{index}, C_2^{index}) \\ u_a & Index_1 \ge C_2^{index} \end{cases} \\ Index_1 &= \sum_{i=1}^{N_f} (\lambda_i^{fx} + \lambda_i^{fy}) \end{aligned} \quad , \quad (12)$$

where N_f is the feature number, u_c is the first control input of the coverage task, u_a is the first control input of the *D-opt*

MPC solution, and c_a and c_c are weights. C_1^{index} and C_2^{index} are two switching indexes, λ_i^{fx} and λ_i^{fy} are the eigenvalues of the covariance matrices of the features at the x and y axes based on the SLAM result.

IV. SOLUTION FOR THE UNCERTAINTY MINIMIZATION PROBLEM

In this section, we use the graphic structure of 2D feature-based SLAM to estimate the effect of the future perfect $\boldsymbol{Z}_{k+1:k+L}$ measurements on the *D-opt* objective function.

A. Graphic structure of 2D feature-based SLAM

In [16], results for 2D pose-based SLAM described by the incidence matrix of the SLAM graph are derived. We derive similar results for the 2D feature-based SLAM problem.

Assumption 1: Isotropic Noise. We assume that the noises in (1) and (2) satisfy:

$$\begin{array}{c} \left[\begin{array}{c} \delta x_k \\ \delta y_k \end{array}\right] \sim \mathcal{N}\left(\mathbf{0}, \delta_v(k)^2 I_2\right) \\ \left[\begin{array}{c} w_x^k \\ w_y^k \end{array}\right] \sim \mathcal{N}\left(\mathbf{0}, \delta_f(k)^2 I_2\right) \end{array}. \tag{13}$$

It is known that the feature-based SLAM graph can be viewed as a weighted directed graph with nodes (pose and features) and the edges (odometry and observations). The weighted value means the precision (inverse of variance) of the corresponding measurement.

For feature-based SLAM, two incidence matrices and their reduced matrices need to be defined: One is that of the pose-graph (a pose-chain), and the other is for the whole graph. The reduced incidence matrices corresponding to the pose graph and the whole graph are defined as A_p and A_g, respectively.

The state vector is rewritten as $\boldsymbol{X} = \begin{array}{cc}(\boldsymbol{p}^T & \boldsymbol{\theta}^T)^T\end{array}$ where $\boldsymbol{p}$ are the coordinates of the robot and the features, and $\boldsymbol{\theta}$ are the orientation angles of the robot. The stacked vector of translational and rotational measurements is $\boldsymbol{z} = \begin{array}{cc}(\boldsymbol{z}_p^T & \boldsymbol{z}_\theta^T)^T\end{array}$.

Based on (1) and (2), the measurement model can be expressed as,

$$\left[\begin{array}{c} \boldsymbol{z}_p \\ \boldsymbol{z}_\theta \end{array}\right] = \left[\begin{array}{cc} R^T(A_g \otimes I_2)^T & 0 \\ 0 & A_p^T \end{array}\right] \left[\begin{array}{c} \boldsymbol{p} \\ \boldsymbol{\theta} \end{array}\right] + \boldsymbol{\delta} \quad , \tag{14}$$

where $\boldsymbol{\delta} \sim \mathcal{N}(\mathbf{0},\ \Sigma)$, here $\otimes$ is the Kronecker product, R is a block-diagonal matrix which consists of $\{R_i\}_{i=1}^m$ in which m is the number of measurements, R_i is the rotation matrix corresponding to the robot orientation making the i-th observation; i.e.,

$$R \triangleq diag(R_1, R_2, ..., R_m), \tag{15}$$

where the covariance matrix Σ can be written as $\Sigma = diag(\Sigma_p; \Sigma_\theta)$, where

$$\Sigma_p = diag(\delta v(1)^2, \cdots, \delta v(k)^2, \delta_f(1)^2, \cdots, \delta_f(m)^2) \otimes I_2, \tag{16}$$

$$\Sigma_\theta = diag\left(\begin{array}{ccc} \delta\omega_k{}^2, \cdots, \delta\omega_k{}^2 \end{array}\right). \tag{17}$$

Then, similar to [16], we can get the Fisher information matrix (FIM):

$$I(\boldsymbol{X}) = \left[\begin{array}{cc} L_{w_p}^g \otimes I_2 & A_{w_p}^g \otimes I_2 \Gamma \triangle_{w_p} \\ (A_{w_p}^g \otimes I_2 \Gamma \triangle_{w_p})^T & \triangle_{w_p}^T \triangle_{w_p} + L_{w_\theta}^p \end{array}\right], \tag{18}$$

where $L_{w_p}^g = A_g \Sigma_p A_g^T$ is the reduced weighted Laplacian matrix of the whole graph. $A_{w_p}^g = A_g \Sigma_p^{\frac{1}{2}}$ is the reduced weighted incidence matrix, Γ is given by, $\Gamma \triangleq I_m \otimes \left[\begin{array}{cc} 0 & 1 \\ -1 & 0 \end{array}\right]$, $L_{w_\theta}^p$ is the reduced weighted Laplacian matrix for pose graph, $A_{w_\theta}^p = A_p \Sigma_\theta^{\frac{1}{2}}$, $\triangle_{w_p} = \triangle \Sigma_p^{\frac{1}{2}}$, $\triangle \in R^{2m \times n}$ is a special structure associate with $\boldsymbol{p'} = (A_g \otimes I_2)^T \boldsymbol{p}$. Suppose in the k-th measurement, the i_kth node has observed the j_kth node. For a 2×1 block in $\triangle$,

$$\triangle_{2k-1:2k,ik} = f(\boldsymbol{p'}) = \boldsymbol{p'}_{2k-1:2k} = \boldsymbol{p}_{jk} - \boldsymbol{p}_{ik}. \tag{19}$$

The remaining elements in $\triangle$ are all zero.

B. Uncertainty bounds for 2D feature-based SLAM

The upper and lower bounds on estimation error covariance of the maximum likelihood estimator in 2D pose SLAM has been given by the weighted number of the spanning trees of the weighted directed graph [16]. Similarly, we can give uncertainty bounds for 2D feature-based SLAM using the D-optimality criterion as:

$$\begin{array}{c} \mathscr{LB} \leq log(det(I(\boldsymbol{X}))) \leq \mathscr{UB} \\ \mathscr{LB} = 2log(det(L_{w_p}^g)) + log(det(L_{w_\theta}^p)) \\ \mathscr{UB} = 2log(det(L_{w_p}^g)) + \sum_{i=1}^{n} log(\lambda_i(L_{w_\theta}^p) + \lambda) \end{array}, \tag{20}$$

where λ is $\left\|\Delta_{w_p}^T \Delta_{w_p}\right\|_\infty$, which is the maximum absolute value of all elements in the matrix, and $\lambda_i(L_{w_p}^p)$ is the eigenvalue of $L_{w_p}^p$.

C. The discussion of the D-opt objective function

Based on Eq. (20), we can bound the determinant of the information matrix for 2D feature-based SLAM. In fact, the lower bound is an important index which can be used to replace the original objective function. If the lower bound of the uncertainty can meet the requirements of the task, the original objective function will meet it unquestionably. Further, in most situations the real value $log(det(I(\boldsymbol{X})))$ approaches the lower bound $\mathscr{LB}$. So, in short, we can firstly replace the original objective function by $\mathscr{LB} = 2log(det(L_{w_p}^g)) + log(det(L_{w_\theta}^p))$. To validate this result numerically, we performed an experiment using one of our simulation datasets. Fig. 2 shows that the real objective function is very close to its lower bound.

Based on [16], $det(L_{w_p}^g)$ and $det(L_{w_\theta}^p)$ are equal to the weighted numbers of the spanning tree of the whole graph and the pose graph, respectively. However, the weighted numbers of the spanning trees of a graph is directly based on the weighted numbers of the cycles in this graph. The whole graph including pose graph and measurements is much more

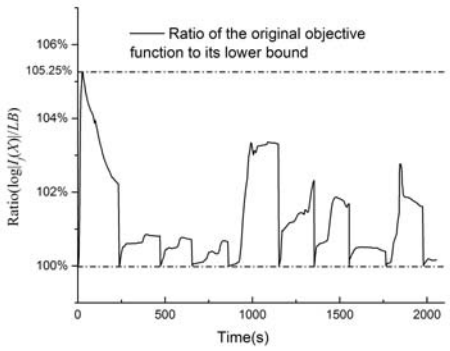

Fig. 2. The real value gets closer to its lower bound

important than the single pose graph, because there are no cycles in the pose graph. So the part $det(L^p_{w_\theta})$ is ignored.

Based on the above discussion, we conclude that the growth in the weighted number of the spanning trees caused by future probabilistic measurements $\boldsymbol{Z}_{k+1:k+L}$ is the most important index. Hence we decide the following new objective function to replace the lower bound $\mathscr{LB}$ and further replace the original objective function $-log(det(I_{k+L}(\boldsymbol{X}^{opt}_{k+L})))$. For the omnidirectional range sensors, we use a new non-convex non-linear constrained least-squares function:

$$c_j = \frac{\sum_{i=1}^{L}\sum_{j=1}^{m} c_j \|\widehat{\boldsymbol{x}}^{opt}_{k+i} - \boldsymbol{x}^{opt}_{fj}\|}{log(det(\widehat{L}^g_{w_p}(j))) - log(det(L^g_{w_p}))} \Big/ \sum_{j=1}^{m}(log(det(\widehat{L}^g_{w_p}(j))) - log(det(L^g_{w_p}))), \tag{21}$$

where $\widehat{\boldsymbol{x}}^{opt}_{k+i}$ is the estimated coordinate of the robot position at the $k+i$-th step and $\boldsymbol{x}^{opt}_{fj}$ is the estimated coordinate of the observed features. $\widehat{L}^g_{w_p}(j)$ is the new weighted laplacian matrix of the whole graph of $\boldsymbol{X}^{opt}_{k+L}$ by adding a virtual future measurement between the j-th observed feature and the predicted future pose $\boldsymbol{x}^{opt}_{k+1}$, where

$$\begin{aligned} \widehat{L}^g_{w_p}(j) &= A_g(j)\Sigma^*_p A_g(j)^T \\ A_g(j) &= \left[\begin{array}{cc} A_g & \boldsymbol{a}_j \end{array}\right] \end{aligned}, \tag{22}$$

here $\boldsymbol{a}_j$ is a column vector that has only two non-zero elements, respectively -1 and 1. -1 is located at the k-th dimension and 1 is located at $k+j$-th dimension, $j = 1,...,N_f$, $\Sigma^*_p(j)$ is the corresponding augmented Σ_p.

There are two important reasons why (21) is suitable for replacing the original objective function:

(1) $\|\widehat{\boldsymbol{x}}^{opt}_{k+i} - \boldsymbol{x}^{opt}_{fj}\|$: In this section, we have shown that the future probabilistic $\boldsymbol{Z}_{k+1:k+L}$ is the key point to improve the value of the objective function. Because we know nothing about the non-detected features, the only way to improve this probability is to make the future poses of the robot get close to the detected features and make their distance smaller than the range of the sensor. So the distances between the future positions of the robot and the estimated positions of the old features need to be minimized as an objective function.

(2) c_j: Even though every new measurement for the observed feature will result in at least one new cycle in the future graph, these cycles obtained by different old features

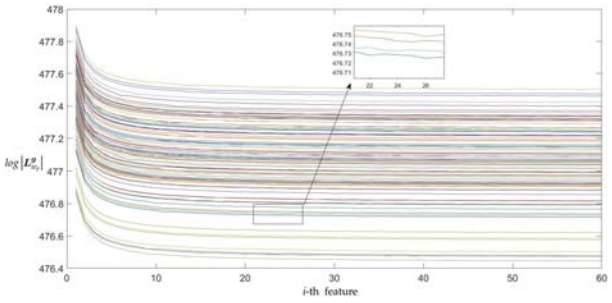

Fig. 3. Monte Carlo simulation for different measurements (The information changes by adding an observation to one feature, assuming the i-th feature has already been observed by i times ($i = 1,\cdots,60$). Different lines mean the different original graph A_g. The total number of measurements remain identical.)

are greatly different. In general, for the features which have been observed for many times, adding a new measurement will not make a big contribution towards decreasing the uncertainty. We perform a Monte Carlo simulation for a SLAM problem with 101 poses and 60 features, and assume that the i-th feature has been observed by random poses for i times. There is no measurement from the last pose. The other settings are chosen randomly. Then one new measurement is added between the last pose and one feature. We get Fig. 3. Every line represents the changing $log(det(\widehat{L}^g_{w_p}(j)))$ values of the new graph when this measurement changes from the 1-st feature to the 60-th feature.

In Fig. 3, we see that the objective function becomes small in general with the order of the feature growing. But it is not monotonic which means that it cannot be replaced by the measurement number and needs to be computed exactly. So the different observed features have different importance degrees. This weight c_j is necessary and reasonable.

D. Transformation into a convex optimization problem

In this section, we will perform variable substitutions to produce a convex problem. The new MPC problem is:

$$\begin{aligned} \underset{u_{k:k+L-1}}{min} \quad & \sum_{i=1}^{L}\sum_{j=1}^{m} c_j \left\| \left[\begin{array}{c} x^{opt}_{k+i} \\ y^{opt}_{k+i} \end{array}\right] - \left[\begin{array}{c} x^{opt}_{fj} \\ y^{opt}_{fj} \end{array}\right] \right\| \\ s.t. \quad & x^{opt}_{k+i+1} = x^{opt}_{k+i} + V\cos(\theta^{opt}_{k+i})\Delta T \\ & y^{opt}_{k+i+1} = y^{opt}_{k+i} + V\sin(\theta^{opt}_{k+i})\Delta T \\ & \theta^{opt}_{k+i+1} = \theta^{opt}_{k+i} + u_{k+i}\Delta T \\ & |u_{k+i}| \le C, i = 0,...,L-1 \end{aligned}, \tag{23}$$

where $\boldsymbol{x}^{opt}_{k+i} = (x^{opt}_{k+i}, y^{opt}_{k+i}, \theta^{opt}_{k+i})^T$, $\boldsymbol{x}^{opt}_{fj} = (x^{opt}_{fj}, y^{opt}_{fj})^T$.

This problem is not convex because of the sine and cosine function of the control inputs. So we need to substitute variables. The new control inputs are defined as:

$$\begin{aligned} u^*_{i,1} &= \cos(\theta^{opt}_{k+i}) \\ u^*_{i,2} &= \sin(\theta^{opt}_{k+i}) \end{aligned}. \tag{24}$$

And they need to satisfy:

$$u^{*\,2}_{i,1} + u^{*\,2}_{i,2} = 1. \tag{25}$$

However, this new constraint (25) is non-convex. Therefore we relax this constraint to:

$$u^{*\,2}_{i,1} + u^{*\,2}_{i,2} \le 1. \tag{26}$$

The constraints $\theta^{opt}_{k+i+1} = \theta^{opt}_{k+i} + u_{k+i}\Delta T$ and $|u_{k+i}| \le C$ can be written as a unified form:

$$|\theta^{opt}_{k+i+1} - \theta^{opt}_{k+i}| \le C\Delta T,\ i = 0, ..., L-1. \tag{27}$$

Using the mean value theorem, we introduce new constraints to relax and replace the constraint:

$$\begin{aligned} |u^*_{i+1,1} - u^*_{i,1}| &\le |\sin(\theta^*)| C\Delta T \le C\Delta T \\ |u^*_{i+1,2} - u^*_{i,2}| &\le |\cos(\theta^*)| C\Delta T \le C\Delta T \end{aligned}, \tag{28}$$

where $\theta^* \in (\theta^{opt}_{k+i+1}, \theta^{opt}_{k+i})$ if $\theta^{opt}_{k+i} > \theta^{opt}_{k+i}$ or $\theta^* \in (\theta^{opt}_{k+i}, \theta^{opt}_{k+i+1})$ if $\theta^{opt}_{k+i} < \theta^{opt}_{k+i}$.

The new convex problem is then:

$$\begin{aligned} \underset{u_{k:k+L-1}}{min} \quad & \sum_{i=1}^{L}\sum_{j=1}^{m} c_j \left\| \begin{bmatrix} x^{opt}_{k+i} \\ y^{opt}_{k+i} \end{bmatrix} - \begin{bmatrix} x^{opt}_{fj} \\ y^{opt}_{fj} \end{bmatrix} \right\| \\ s.t. \quad & x^{opt}_{k+i+1} = x^{opt}_{k+i} + V u^*_{i,1}\Delta T \\ & y^{opt}_{k+i+1} = y^{opt}_{k+i} + V u^*_{i,2}\Delta T \\ & |u^*_{i+1,1} - u^*_{i,1}| \le C\Delta T \\ & |u^*_{i+1,2} - u^*_{i,2}| \le C\Delta T \\ & {u^*_{i,1}}^2 + {u^*_{i,2}}^2 \le 1,\ i = 0, ..., L-1 \end{aligned}. \tag{29}$$

This convex problem can be easily solved using convex toolboxes such as CVX and CVXGEN. Because of its convexity, its result will be globally optimal.

E. Candidate solution to original least-squares problem

Because of the relaxations in the section above, the solution to (29) may not be a feasible solution to the original non-convex least-squares problem. In this section, the way to generate a candidate solution is addressed. Suppose the solution of the new convex problem (29) is:

$$\begin{aligned} \boldsymbol{V}^*_1 &= (\widehat{u}_{0,1}, \widehat{u}_{1,1}, ..., \widehat{u}_{L-1,1})^T \\ \boldsymbol{V}^*_2 &= (\widehat{u}_{0,2}, \widehat{u}_{1,2}, ..., \widehat{u}_{L-1,2})^T \end{aligned}. \tag{30}$$

We can also get the control inputs of the original non-convex non-linear constrainted least-squares problem by:

$$\begin{aligned} u_{k+i} &= f_{round}\left(\theta^{opt}_{k+i+1} - \theta^{opt}_{k+i}\right) \\ &= f_{round}\left(atan2(\widehat{u}_{i+1,2}, \widehat{u}_{i+1,1}) - atan2(\widehat{u}_{i,2}, \widehat{u}_{i,1})\right) \end{aligned}, \tag{31}$$

where $f_{round}(\star)$ is a piecewise function:

$$f_{round}(\star) = \begin{cases} -C & \star/\Delta T < -C \\ \star/\Delta T & -C \le \star/\Delta T \le C \\ C & \star/\Delta T > C \end{cases}. \tag{32}$$

Finally, we get the solution $(u_k, u_{k+1}, .., u_{k+L-1})^T$ for the uncertainty minimization problem.

V. SLAM BASED ON LINEAR SUBMAP JOINING

Early work on active SLAM has been based on EKF SLAM framework [2]. However, the quality of EKF SLAM is questionable especially when the orientation error is not very small, which is the case for most of the active SLAM scenarios. In this paper, we use the optimization based SLAM as shown in (3) to improve the quality of SLAM. The GN method is used to solve the SLAM problem. It is known that the real-time ability of the GN-based SLAM will decrease if the size of the problem becomes large. Submap joining [15] is a way to solve this problem. Using submap joining, we only need to perform SLAM, uncertainty minimization and coverage optimization within each local map.

Even though the submap joining can greatly reduce the estimation and planning time, several new issues are introduced in the problem solving process.

The first problem is how to finish the coverage task based on the submap. It is easy to know that the covered area is obtained based on the global coordinates of the robot. Because of the submap joining method, we cannot obtain the accurate global coordinates of the robot without the joining process. However, the joining process is time-consuming if this process is used at every step. So we need to estimate the pose and its uncertainty without performing the joining. We assume that at step k the robot is located at N^{sub}_i-th submap and sub_j-th step of the submap. The estimated coordinate $\boldsymbol{x}^{sub_j}_i,\ i \in \{1, 2, ..., N^{sub}_i\}$ and its corresponding 2×2 covariance matrix $Cov^{sub_j}_i$ in i-th submap coordinate system can be calculated by:

$$\begin{aligned} \boldsymbol{x}^{sub_j}_{N^{sub}_i - 1} &= \bar{R}_{N^{sub}_i - 1}\boldsymbol{x}^{sub_j}_{N^{sub}_i} + \bar{\boldsymbol{x}}_{N^{sub}_i - 1} \\ &\vdots \\ \boldsymbol{x}^{sub_j}_1 &= \bar{R}_1\boldsymbol{x}^{sub_j}_2 + \bar{\boldsymbol{x}}_1 \\ \boldsymbol{x}^{opt}_k &= \boldsymbol{x}^{sub_j}_1 \\ Cov^{sub_j}_{N^{sub}_i - 1} &= \bar{R}_{N^{sub}_i - 1} Cov^{sub_j}_{N^{sub}_i} \bar{R}^{-1}_{N^{sub}_i - 1} + \overline{Cov}^{sub_j}_{N^{sub}_i - 1} \\ &\vdots \\ Cov^{sub_j}_1 &= \bar{R}_1 Cov^{sub_j}_2 \bar{R}^{-1}_1 + \overline{Cov}^{sub_j}_{\mathbf{1}} \\ Cov^{opt}_k &= Cov^{sub_j}_1 \end{aligned}, \tag{33}$$

where $\bar{R}_i$ are respectively the rotation matrices of the last pose of the i-th submap, $\bar{\boldsymbol{x}}_i$ is the xy coordinate of the last pose of the i-th submap, $\overline{Cov}^{sub_j}_i$ is the corresponding 2×2 covariance matrix. So with the current pose in the first local frame $\boldsymbol{x}^{opt}_k$ and its covariance matrix Cov^{opt}_k, we can get the covered area of the robots based on submap without joining at every step.

Beside this problem, because our active SLAM can be performed in every submap independently, all the features detected in the new submap will be regarded as new features in the SLAM until the map joining process. This will lead to the robot continuously visiting some old features which have been mapped in the other submap as the unknown features.

For this problem, after opening a new submap, we need to judge whether the feature has been detected and whether its uncertainty has been reduced to an acceptable level in the old submaps or not. If a feature meets this condition, it will not be used in the objective function of the uncertainty minimization task.

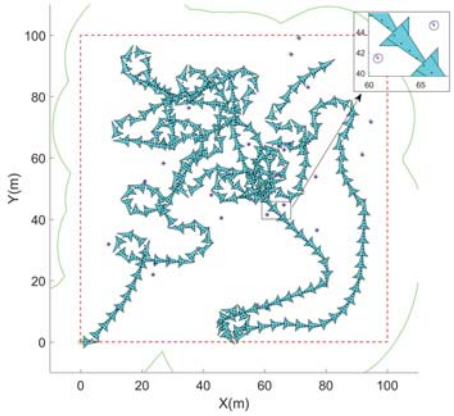

Fig. 4. Final results for dataset 1 after using submap joning

VI. Simulations

In this section, we will show simulation results to validate the effectiveness of the presented algorithm.

A. Active SLAM and coverage task using proposed method

The simulation environment is created using MATLAB. The robot, using a 20 m limited range omnidirectional sensor, moves in an 100×100 m non-obstacle square-shaped space with the known bounds and 30 unknown static features. Specifically, it moves at 1m/s and its control input ω_k is limited in $[-0.3, 0.3]$ rad/s. Synthetic errors, with a Gaussian distribution, are generated for the odometry model of the robot ($\delta_v(k) = 0.1$ m and $\delta_\omega(k) = 0.1$ rad) and the sensor measurements ($\delta_f(k) = 0.3$ m) is assumed. 10 datasets with different feature positions name as dataset 1-10. The other parameters are set as: $c_a = 0.85$, $c_c = 0.15$, $C_1^{index} = 0.06N_f$ m, $C_2^{index} = 0.1N_f$ m, the size of submap is 50. The results of proposed method using dataset 1 is shown in Fig. 4.

In Fig. 4, the blue vehicles are the real trajectory of the robot at each 5 steps. The black points show the estimated trajectory and features obtained by SLAM. The red star points are the real positions of the features. The blue circles are the 3σ covariance ellipses of the features. Fig. 4 shows that the robot finishes the SLAM task with good accuracy. The area of interest is covered by the sensor.

B. Comparisons

1) The effect of the linear submap joining method: The linear SLAM method can limit the size of the SLAM problem with an acceptable information lost. If we do not use this method, the time used in the SLAM part will become longer and longer. Here we compare the performance when different submap size is used. Except the submap size, the features and the stop conditions, the other settings are the same as the settings of the simulation described in SectionVI-A. When the whole area is covered more than 97% percent and all features are detected, the simulation will stop. The stop condition of the simulation without using the submap is that the simulation time reaches 2000s. Then, the comparison results are presented in Fig. 5.

Because of the different number of poses involved, we only compare the information matrix of the features by the

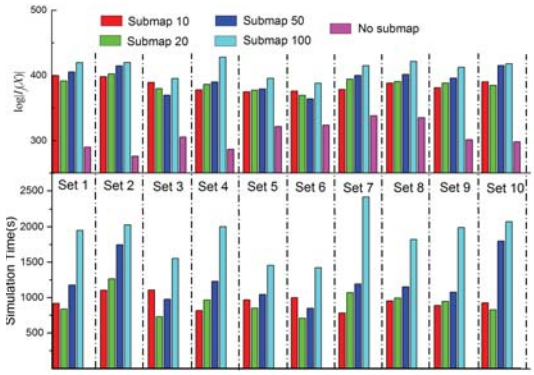

Fig. 5. Comparison results using different sizes of submaps (In 2000s, the simulation without using submap does not finish the coverage task. the covered percents from set 1 to 10 are respectively 35.2%, 32.6%, 29.3%, 46.2%, 29.8%, 37.3%, 25.2%, 39.3%, 35.2% and 31.1%)

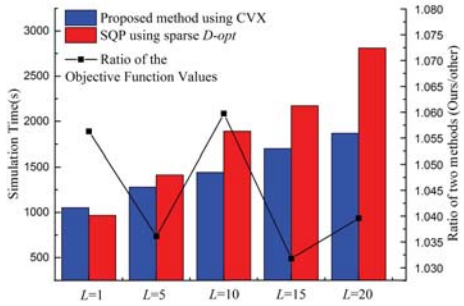

Fig. 6. Comparison results using graphical approximation and convex method (Histogram shows the time and line graph shows the ratio)

D-opt optimality criteria $log(det(I_f(X)))$. All simulations using submaps finish the coverage task and detect all features. For the non-submap simulation, because of the slow SLAM process, its total number of measurements is less than the ones using submap, which leads to the bad objective functions.

2) The effect of the graphical approximation and the convex method: We use the graphical method to approximate the original *D-opt* objective function and solve the problem by the convex method. In this part, we solve the uncertainty minimization by the SQP method with the original objective function. The results based on dataset 10 with different look-ahead steps are shown in Fig. 6.

Because the original objective function is a non-convex complex function whose solving difficulty grows with the look-ahead step L, we can see the estimation and planning time grows quickly. Because of the convexity of new objective function, the problem can be solved in polynomial time. Meanwhile, many local minimums were obtained when using the SQP method. So its final solution is worse than the one of our proposed method.

C. The effect of two control inputs

In this part, we will discuss the real active control input u_r obtained by two different MPC problems. Fig. 7 shows the result using dataset 10 at the 80-th step, including the real robot trajectory and covered area. The green dotted line indicates the optimization results of the rounding convex method in 15 look-ahead steps. The black dotted line indicates the optimization results of the coverage task in 5 look-ahead

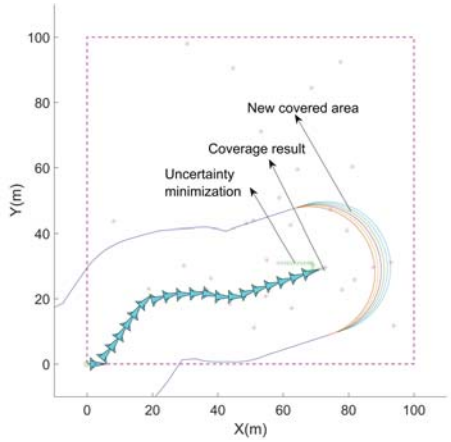

Fig. 7. Trajectory and optimization outputs at 80-th step

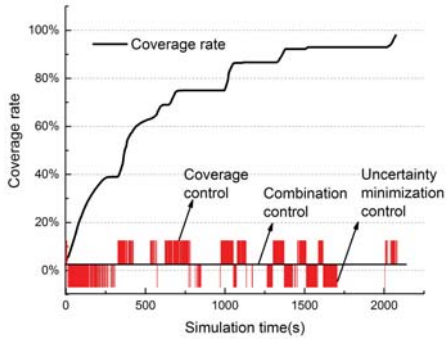

Fig. 8. Coverage rate and real active control inputs

steps. The new covered area using the control inputs of the coverage task is shown. We can see that the uncertainty minimization results lead the robot to loop-closure and the coverage results make the robot visit the unknown space.

Fig. 8 shows that the real active control inputs and its coverage rate changes with the simulation time, where 1, 0 and -1 respectively mean that the final acting control u_r is the coverage control, combination control or uncertainty minimization control. We can see that the parameters of the switching mechanism are suitable in the simulation. Three different control inputs are used rather than the single input.

VII. CONCLUSIONS

In this work we have formulated the *D-opt* uncertainty minimization problem and the coverage problem with pose uncertainty in the submaps by the MPC framework. The linear SLAM method is used to reduce the estimation and planning time. We have presented a new non-convex constrainted least square problem to replace the traditional *D-opt* uncertainty minimization problem by using the distance function and the lower bound of the D-optimality criterion, which is given by the weighted number of the spanning trees of the SLAM graph and its pose sub-graph. We also have proposed a convex planning approach for this new problem. We test our approach in realistic simulation scenarios; experimental results show that the approach is able to deal with the coverage tasks in an unknown environment with a good SLAM result. Compared with the SQP-based method, the approach produces better and faster results.

In the future, we would like to solve the 2D active SLAM in the complex obstacle environment based on new directional sensor with multiple constraints, such as fuel constraint and complex dynamic constraints. We also plan to extend this algorithm into 3D active SLAM. We will try to improve real-time ability of this method and perform experiments by a real unmanned aerial vehicle (UAV) platform.

REFERENCES

[1] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1309–1332, 2016.

[2] S. Huang, N. Kwok, G. Dissanayake, Q. Ha, and G. Fang, "Multi-step look-ahead trajectory planning in SLAM: Possibility and necessity," in *2005 IEEE International Conference on Robotics and Automation (ICRA)*, 2005, pp. 1091–1096.

[3] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and acting in partially observable stochastic domains," *Artificial Intelligence*, vol. 101, no. 1, pp. 99–134, 1998.

[4] A. Pazman., *Foundations of Optimum Experimental Design*. Springer, 1986.

[5] H. Carrillo, I. Reid, and J. A. Castellanos, "On the comparison of uncertainty criteria for active SLAM," in *2012 IEEE International Conference on Robotics and Automation (ICRA)*, 2012, pp. 2080–2087.

[6] H. Carrillo, Y. Latif, M. L. Rodriguez-Arevalo, J. Neira, and J. A. Castellanos, "On the Monotonicity of Optimality Criteria during Exploration in Active SLAM," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 1476–1483.

[7] C. Leung, S. Huang, N. Kwok, and G. Dissanayake, "Planning under uncertainty using model predictive control for information gathering," *Robotics and Autonomous Systems*, vol. 54, no. 11, pp. 898–190, 2006.

[8] V. Indelman, L. Carlone, and F. Dellaert, "Planning in the continuous domain: A generalized belief space approach for autonomous navigation in unknown environments," *The International Journal of Robotics Research*, vol. 34, no. 7, pp. 849–882, 2015.

[9] V. Indelman, "No Correlations Involved: Decision Making Under Uncertainty in a Conservative Sparse Information Space," *IEEE Robotics and Automation Letters*, vol. 1, no. 1, pp. 407–414, 2016.

[10] M. Keidar and G. A. Kaminka, "Efficient frontier detection for robot exploration," *The International Journal of Robotics Research*, vol. 33, no. 2, pp. 215–236, 2014.

[11] V. D. Berg, S. P. Jur, and R. Alterovitz, "Motion planning under uncertainty using iterative local optimization in belief space," *The International Journal of Robotics Research*, vol. 31, no. 11, pp. 1263–1278, 2012.

[12] J. Vallvé and J. A. Cetto, "Active pose SLAM with RRT*," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 1050–4729.

[13] I. Maurović, M. Seder, K. Lenac, and I. Petrović, "Path Planning for Active SLAM Based on the D* Algorithm With Negative Edge Weights," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 99, no. 1, pp. 1–11, 2017.

[14] J. Vallvé and J. A. Cetto, "Potential information fields for mobile robot exploration," *Robotics and Autonomous Systems*, vol. 69, pp. 68–79, 2015.

[15] L. Zhao, S. Huang, and G. Dissanayake, "Linear SLAM: A linear solution to the feature-based and pose graph SLAM based on submap joining," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2013)*, 2013, pp. 24–30.

[16] K. Khosoussi, S. Huang, and G. Dissanayake, "Novel insights into the impact of graph structure on SLAM," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2014, pp. 2707–2714.