Multi-Step Look-Ahead Trajectory Planning in SLAM: Possibility and Necessity

Shoudong Huang, N. M. Kwok, G. Dissanayake, Q.P. Ha

ARC Centre of Excellence for Autonomous Systems (CAS)
Faculty of Engineering
University of Technology, Sydney, Australia
{sdhuang,ngai.kwok,gdissa,quangha}@eng.uts.edu.au

Gu Fang

School of Engineering
University of Western Sydney, Australia
g.fang@uws.edu.au

Abstract— In this paper, the possibility and necessity of multi-step trajectory planning in Extended Kalman Filter (EKF) based SLAM is investigated. The objective of the trajectory planning here is to minimize the estimation error of the robot and landmark locations subject to a given time horizon. We show that the problem can be regarded as an optimization problem for a gradually identified model. A numerical method is proposed for trajectory planning using a variant of the nonlinear Model Predictive Control (MPC). The proposed method is optimal in the sense that the control action is computed using all the information available at the time of decision making. Simulation results are included to compare the results from the one-step look-ahead trajectory planning and the proposed multi-step look-ahead technique.

Index Terms— SLAM, trajectory planning, Extended Kalman Filter, model predictive control.

I. INTRODUCTION

Simultaneous localization and mapping (SLAM) is a fundamental requirement in robot exploration and has attracted a lot of research interest in recent years. Extended Kalman Filter (EKF) based approaches are frequently used in SLAM (e.g. [1]–[3]) and the results are mostly satisfactory.

Most of the SLAM algorithms did not take trajectory planning into account. That is, the robot trajectory is predetermined or randomly chosen. It is well known that a good robot trajectory is critical in robot exploration. Moreover, in most cases the environment is at least partially unknown and on-line trajectory planning is necessary.

When SLAM is regarded as only a part of a robot exploration task, several trajectory planning strategies have been proposed. For example, Makarenko et al. [4] considered the integrated exploration problem where the tasks of localization, mapping and motion control are combined together. Multiple utility functions are used to evaluate the utilities of the potential destinations and the destination with the highest total utility is selected as the next destination. In [5], a minimal time motion control technique in SLAM is developed when the destination is a prescribed location relative to the starting point. The robot is required to reach the destination within minimal time while keeping the estimation error of the robot and landmark locations below a prescribed level. Stachniss et. al [6] focused on the active loop closing for FastSLAM in robot exploration. They provided approaches on

how to identify opportunities for closing loop, how to close the loop and when to stop following the loop for the grid-based FastSLAM framework.

In the case when SLAM is the sole robot exploration objective, trajectory planning is also important in obtaining a more accurate estimation and/or using less time. In particular, when the sensor capability is limited (e.g. bearing only sensor), a properly planned trajectory will be highly desirable [7].

An adaptive motion control technique in SLAM is reported in [8]. The robot creates a map and localizes itself simultaneously while making local decisions on where to move next in order to maximize the information obtained from the observations. The inverse of the estimation error covariance is used as an optimization objective. The choice of the decision is based on a single-step look-ahead strategy (greedy method).

In this paper, we analyze the possibility and necessity of multi-step look-ahead trajectory planning in SLAM. The problem addressed here is similar to that of [8]. In particular, the objective of the trajectory planning is to minimize the estimation error spanning in a finite time horizon. The contributions of this paper include (1) the necessity of trajectory planning in SLAM is analyzed in term of the information gain/loss (2) it is shown that the multi-step look-ahead trajectory planning problem is an optimization problem for a gradually identified nonlinear model; (3) it is shown that multi-step look-ahead is possible when the current estimation error is small because it optimizes the mean value of the performance metric (4) a novel optimization strategy is proposed for this trajectory planning problem using a variant of the Model Predictive Control (MPC) [10] with the optimality proven.

The paper is organized as follows. In Section II, the SLAM algorithm using EKF is briefly reviewed. In Section III, the multi-step trajectory planning problem considered in this paper is formulated. In Section IV, we analyze the necessary assumptions used in the one-step and multi-step optimization. A novel optimization strategy is proposed and optimality is proved in Section V. In Section VI, simulation results are provided to compare the one-step and multi-step look-ahead optimization techniques. Finally, Section VII concludes the paper.

II. SLAM USING EXTENDED KALMAN FILTER

In this section, we briefly review the estimation-theoretic SLAM algorithm using the Extended Kalman Filter (EKF) (see, e.g. [2], [8]).

Let the robot's state be denoted by $\mathbf{x}_r = [x_r, y_r, \phi_r]^T$ and the dynamic model for the robot be given by

$$\mathbf{x}_{r_{k+1}} = \mathbf{f}(\mathbf{x}_{r_k}, \mathbf{u}_k, \mathbf{d_x}), \tag{1}$$

where $\mathbf{u}_k$ is the control input at time k, $\mathbf{d_x}$ is the Gaussian process noise with covariance Σ. The exact formula of function $\mathbf{f}$ depends on the types of the robot and the process noise.

The landmarks (features) are assumed to be stationary and we use $\mathbf{x}_f$ to denote the state of all the landmarks. Let the state vector $\mathbf{x} = [\mathbf{x}_r^T, \mathbf{x}_f^T]^T$ contain both the robot states $\mathbf{x}_r$ and the landmark states $\mathbf{x}_f$, then the process model can be written as

$$\mathbf{x}_{k+1} = \mathbf{F}(\mathbf{x}_k, \mathbf{u}_k, \mathbf{d_x}) = \begin{bmatrix} \mathbf{f}(\mathbf{x}_{r_k}, \mathbf{u}_k, \mathbf{d_x}) \\ \mathbf{x}_f \end{bmatrix}. \tag{2}$$

The observation model at time $k+1$ is

$$\mathbf{z}_{k+1} = \mathbf{h}_{k+1}(\mathbf{x}_{k+1}) + \mathbf{d}_{\mathbf{z}(k+1)}, \tag{3}$$

where $\mathbf{h}_{k+1}$ is the nonlinear observation vector function, $\mathbf{d}_{\mathbf{z}(k+1)}$ is the Gaussian observation noise vector with covariance $\mathbf{R}_{k+1}$. Notice that the formulation of the observation function and the dimension of the noise depend on the sensor model as well as the group of beacons that can be observed at time $k+1$ (e.g. field of view). Since the group of beacons that can be observed highly depends on the relative locations of the robot and landmarks at the time of observation, the observation function $\mathbf{h}_{k+1}$ and the noise $\mathbf{d}_{\mathbf{z}(k+1)}$ are both time dependent.

The estimation-theoretic SLAM algorithm is based on the work of Smith et al. [1]. It uses the EKF algorithm to optimally estimate the state vector $\mathbf{x}$ and the associate error covariance matrix

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_{rr} & \mathbf{P}_{rf} \\ \mathbf{P}_{rf}^T & \mathbf{P}_{ff} \end{bmatrix}. \tag{4}$$

Suppose at step k, the estimation of the states of the robot and landmarks is $\hat{\mathbf{x}}_{k|k}$ and the error covariance matrix is $\mathbf{P}_{k|k}$. The EKF algorithm proceeds recursively in two stages:

(1) Predict the current states of the robot and the landmarks $\hat{\mathbf{x}}_{k+1|k}$ using the process model, and compute the state estimate error covariance matrix $\mathbf{P}_{k+1|k}$:

$$\begin{aligned} \hat{\mathbf{x}}_{k+1|k} &= \mathbf{F}(\hat{\mathbf{x}}_{k|k}, \mathbf{u}_k, \mathbf{0}) \\ \mathbf{P}_{k+1|k} &= \mathbf{F}_{1k}\mathbf{P}_{k|k}\mathbf{F}_{1k}^T + \mathbf{F}_{2k}\Sigma\mathbf{F}_{2k}^T, \end{aligned} \tag{5}$$

where $\mathbf{F}_{1k}$ and $\mathbf{F}_{2k}$ are the Jacobians of $\mathbf{F}$ with respect to $\mathbf{x}$ and $\mathbf{d_x}$ evaluated at $(\hat{\mathbf{x}}_{k|k}, \mathbf{u}_k, \mathbf{0})$, respectively. That is,

$$\mathbf{F}_{1k} = \nabla_{\mathbf{x}}\mathbf{F}|_{(\hat{\mathbf{x}}_{k|k}, \mathbf{u}_k, \mathbf{0})}, \quad \mathbf{F}_{2k} = \nabla_{\mathbf{d_x}}\mathbf{F}|_{(\hat{\mathbf{x}}_{k|k}, \mathbf{u}_k, \mathbf{0})}. \tag{6}$$

(2) Update the estimation using the observation $\mathbf{z}_{k+1}$:

$$\begin{aligned} \hat{\mathbf{x}}_{k+1|k+1} &= \hat{\mathbf{x}}_{k+1|k} + \mathbf{K}_{k+1}(\mathbf{z}_{k+1} - \mathbf{h}_{k+1}(\hat{\mathbf{x}}_{k+1|k})), \\ \mathbf{P}_{k+1|k+1} &= \mathbf{P}_{k+1|k} - \mathbf{K}_{k+1}\mathbf{S}_{k+1}\mathbf{K}_{k+1}^T, \end{aligned} \tag{7}$$

where

$$\begin{aligned} \mathbf{K}_{k+1} &= \mathbf{P}_{k+1|k}\mathbf{H}_{1k}^T\mathbf{S}_{k+1}^{-1}, \\ \mathbf{S}_{k+1} &= \mathbf{H}_{1k}\mathbf{P}_{k+1|k}\mathbf{H}_{1k}^T + \mathbf{R}_{k+1}, \end{aligned} \tag{8}$$

and $\mathbf{H}_{1k}$ is the Jacobian of $\mathbf{h}_{k+1}$ evaluated at $\hat{\mathbf{x}}_{k+1|k}$. That is,

$$\mathbf{H}_{1k} = \nabla_{\mathbf{x}}\mathbf{h}_{k+1}|_{\hat{\mathbf{x}}_{k+1|k}}. \tag{9}$$

III. MULTI-STEPS TRAJECTORY PLANNING PROBLEM

A. The need for trajectory planning in SLAM

Trajectory planning is definitely very important in SLAM because the robot needs to explore new area in order to have a better knowledge of the environment. Suppose there is no particular destination or direction that the robot needs to follow, is trajectory planning still necessary to reduce the estimation error of the landmark and robot locations?

When trajectory planning is considered in SLAM, there is a trade-off between information gain obtained from the observations and information loss arising from the process noise. In the following, we would like to comment that changing the observing location is critical to obtain more information if the process noise is relatively small.

By the Extended Information Filter approach [9], [11], the information obtained from the observation $\mathbf{z}_{k+1}$ is a matrix

$$\mathbf{I}_{new}^{k+1} = \mathbf{H}_{1k}^T\mathbf{R}_{k+1}^{-1}\mathbf{H}_{1k}, \tag{10}$$

where $\mathbf{H}_{1k}$ is given in (9).

Suppose the information matrix before the observations is $\mathbf{I}_k$ ($\mathbf{I}_k$ is approximately $\mathbf{P}_{k+1|k}^{-1}$), then the information matrix after the observation is

$$\mathbf{I}_{k+1} = \mathbf{I}_k + \mathbf{I}_{new}^{k+1} = \mathbf{I}_k + \mathbf{H}_{1k}^T\mathbf{R}_{k+1}^{-1}\mathbf{H}_{1k}. \tag{11}$$

Note that the matrix $\mathbf{H}_{1k}$ is generally not full rank (if the robot observe m beacons with range and bearing sensor, then the rank of $\mathbf{H}_{1k}$ is at most $2m$, but the dimension of $\mathbf{I}_k$ is $3 + 2m_0$ where m_0 is the total number of the beacons in the state vector $\mathbf{x}$), hence the information matrix $\mathbf{I}_{new}^{k+1}$ is not full rank. If the robot is kept stationary while observing the same group of beacons, there will be an upper limit on the minimal eigenvalue of the total information matrix minimal eigenvalue (the minimal eigenvalue is inversely proportional to the length of the longest axis of the uncertainty ellipse). However, if the robot moves to a new observation position, a new matrix (different from $\mathbf{I}_{new}^{k+1}$) can be added to $\mathbf{I}_k$ and the upper limit on the minimal eigenvalue of the total information matrix may be further increased.

Let us use a simple example to illustrate this. Suppose there is only one beacon at position $(0, 0)$, and the current information on the robot and beacon locations $[x_r, y_r, \phi_r, x_f, y_f]^T$ is

$$\mathbf{I}_0 = diag[100, 100, 100, 1, 1]. \tag{12}$$

If the robot is stationary at the position $(5, 5)$ and keeps observing the beacon (range and bearing observation with observation covariance matrix $diag[0.2, 0.1]$), the minimal eigenvalue of the total information matrix can not exceed 2.9038. But if the robot moves to another position $(7, 0)$

later on and keeps observing the beacon at the new position, then the minimal eigenvalue of the total information matrix can be increased to 50.5 if there is no process noise (when process noise is $diag[0.2, 0.2, 0.2]$, the minimal eigenvalue is 3.4986, when process noise is $diag[0.5, 0.5, 0.5]$, the minimal eigenvalue is 1.9605 which is worse than keeping stationary).

When the observations are range only or bearing only observations, trajectory planning is more critical because the rank of $\mathbf{I}_{new}^{k+1}$ is even lower. Certainly, the necessity of trajectory planning in SLAM is highly depended on the process and observation model and it is formidable for a detailed analysis without the true models.

Another conclusion can be obtained by further analyzing equation (11): In order to find the best movement such that the total information at next step (e.g. the minimal eigenvalue of $\mathbf{I}_{k+1}$) is maximized, the whole current information matrix $\mathbf{I}_k$ is needed. If only a quantity measure of $\mathbf{I}_k$ is known (e.g. the minimal eigenvalue or the determinant), it is impossible to decide which position is the best to make further observations (a different observation position results in a different $\mathbf{H}_{1k}$). Hence, a trajectory planned by using any information quantity (e.g. entropy) is only sub-optimal.

Feder et.al [8] provided an adaptive motion control technique that using the whole error covariance matrix $\mathbf{P}_{k|k}$ to choose the control at time k. This is a one-step look ahead optimization method. A nature question to ask is whether the multi-step look-ahead optimization is needed or not. The purpose of this paper is to address this question.

B. The multi-step trajectory planning problem

The multi-step trajectory planning problem considered in this paper is the following.

Trajectory Planning Problem. Consider the finite time horizon $[0, N]$ where N is a given integer. Suppose at time 0, the estimation of the locations of the robot and landmarks is $\hat{\mathbf{x}}_{0|0}$ and the error covariance matrix is $\mathbf{P}_{0|0}$. Choose the control

$$\mathbf{u}_0, \mathbf{u}_1, \cdots, \mathbf{u}_{N-1},$$

such that the trace of the error covariance matrix at time N,

$$trace(\mathbf{P}_{N|N}),$$

is minimized, where $\mathbf{P}_{N|N}$ is given by (5)-(9).

We use $trace(\mathbf{P}_{N|N})$ as the quantity measure of the final estimation error (the approach proposed in this paper will be the same if another measure is used). Note that $trace(\mathbf{P}_{N|N})$ depends not only on $trac(\mathbf{P}_{0|0})$, it depends on the whole matrix $\mathbf{P}_{0|0}$ as well as some other information. So we can not only use $trac(\mathbf{P}_{0|0})$ to decide the optimal trajectory.

C. Optimization problem with a gradually identified model

Recall equations (5)-(9), the observation $\mathbf{z}_1$ is required to compute the state estimation $\hat{\mathbf{x}}_{1|1}$. So $\hat{\mathbf{x}}_{1|1}, \hat{\mathbf{x}}_{2|1}$ and $\mathbf{H}_{11}$ can not be decided for given $\mathbf{u}_0, \mathbf{u}_1$ until the real observation $\mathbf{z}_1$ is made at time 1. In other words, at time 0, we can not obtain a clear relationship between the variables (control $\mathbf{u}_0, \cdots, \mathbf{u}_{N-1}$) and the objective function $trace(\mathbf{P}_{N|N})$.

Hence at time 0, the EKF model (5)-(9) can not be uniquely determined. However, a more precise model can be obtained gradually. For example, we know $\mathbf{z}_1$ at time 1, we know $\mathbf{z}_2$ at time 2, and so on. Therefore, the Trajectory Planning Problem is an optimization problem with a gradually identified (nonlinear) model.

IV. ASSUMPTIONS ON ONE-STEP AND MULTI-STEP OPTIMIZATION

In this section, we will state clearly the basic assumptions required in the one-step and multi-step optimization for the Trajectory Planning Problem.

A. One-step optimization – Greedy method

At step 0, $\hat{\mathbf{x}}_{0|0}$ and $\mathbf{P}_{0|0}$ are known. By (6), $\mathbf{F}_{10}, \mathbf{F}_{20}$ are functions of the control input $\mathbf{u}_0$. Hence by (5), $\hat{\mathbf{x}}_{1|0}$ and $\mathbf{P}_{1|0}$ are functions of $\mathbf{u}_0$.

Notice that the observation model $\mathbf{h}_1$ and the observation noise $\mathbf{d}_{\mathbf{z}(1)}$ depend on the true position of the robot and the true positions of the beacons, that is, the true state $\mathbf{x}_1$. But in practice, the true state is not available. At time 0, we can only use the prediction $\hat{\mathbf{x}}_{1|0}$ to obtain an approximate model of $\mathbf{h}_1$ and $\mathbf{d}_{\mathbf{z}(1)}$.

Assumption I. For any possible control $\mathbf{u}_0$, the group of beacons that are predicted to be observed at time 1 are the same as those that will be really observed at time 1.

Assumption I implies that there will be no new beacon to be observed at time 1. Under Assumption I, the estimated model $\mathbf{h}_1$ is the same as the true model. Now we can say that $\mathbf{h}_1$ and hence $\mathbf{H}_{10}$ are functions of $\mathbf{u}_0$, so $\mathbf{S}_1, \mathbf{K}_1$ and $\mathbf{P}_{1|1}$ are also functions of $\mathbf{u}_0$ (Notice that $\mathbf{P}_{1|1}$ does not depend on the observation $\mathbf{z}_1$ though $\hat{\mathbf{x}}_{1|1}$ does). Thus we can perform the following one-step look-ahead optimization to find $\mathbf{u}_0$.

One-Step Optimization Problem. Given $\hat{\mathbf{x}}_{0|0}$ and $\mathbf{P}_{0|0}$, choose $\mathbf{u}_0$ to minimize $trace(\mathbf{P}_{1|1})$ where $\mathbf{P}_{1|1}$ is given by

$$
\begin{aligned}
\mathbf{P}_{1|0} &= \mathbf{F}_{10}\mathbf{P}_{0|0}\mathbf{F}_{10}^T + \mathbf{F}_{20}\Sigma\mathbf{F}_{20}^T, \\
\hat{\mathbf{x}}_{1|0} &= \mathbf{F}(\hat{\mathbf{x}}_{0|0}, \mathbf{u}_0, \mathbf{0}) \\
\mathbf{H}_{10} &= \nabla_{\mathbf{x}}\mathbf{h}_1|_{\hat{\mathbf{x}}_{1|0}} \\
\mathbf{S}_1 &= \mathbf{H}_{10}\mathbf{P}_{1|0}\mathbf{H}_{10}^T + \mathbf{R}_1, \\
\mathbf{K}_1 &= \mathbf{P}_{1|0}\mathbf{H}_{10}^T\mathbf{S}_1^{-1}, \\
\mathbf{P}_{1|1} &= \mathbf{P}_{1|0} - \mathbf{K}_1\mathbf{S}_1\mathbf{K}_1^T.
\end{aligned}
\tag{13}
$$

Similarly, at time 1, the observation $\mathbf{z}_1$ is available and can be used to update the new estimate $\hat{\mathbf{x}}_{1|1}$ and $\mathbf{P}_{1|1}$. The above one-step optimization method can be used again to compute $\mathbf{u}_1$. The procedure repeats until time $N-1$, where the set of control $\mathbf{u}_0, \mathbf{u}_1, \cdots, \mathbf{u}_{N-1}$ are all obtained.

B. Multi-step optimization

It is very easy to find out in the control and optimization literature, that one-step optimization solution may turn out to be far inferior from multi-step optimal solution.

Since the observation $\mathbf{z}_{k+1}(k = 0, \cdots, N-1)$ is not available at time 0, it seems impossible to perform a multi-step optimization. However, in the EKF implementation, we always assume that the distribution of the true locations $\mathbf{x}_{k+1}$

is Gaussian with mean $\hat{\mathbf{x}}_{k+1|k}$. By this we can say that at time 0, the innovations $\mathbf{z}_{k+1} - \mathbf{h}_{k+1}(\hat{\mathbf{x}}_{k+1|k})$ are all random variables with zero mean. So we can perform the multi-step optimization assuming that the innovations at all the steps are zero.

Assumption II. For $k = 0, \cdots, N-1$, the group of beacons that are predicted to be observed at time $k+1$ (the predictions are made using only $\hat{\mathbf{x}}_{0|0}$, $\mathbf{P}_{0|0}$ and the possible control law $\mathbf{u}_0, \cdots, \mathbf{u}_{N-1}$) are the same as those that will be really observed at time $k+1$.

Assumption III. For $k = 0, \cdots, N-1$, the innovations at any time $k+1$ are zero, i.e.

$$\mathbf{z}_{k+1} - \mathbf{h}_{k+1}(\hat{\mathbf{x}}_{k+1|k}) = 0 \qquad (14)$$

for all $k = 0, \cdots, N-1$.

Under Assumption II and Assumption III, by (7) we have

$$\hat{\mathbf{x}}_{k+1|k+1} = \hat{\mathbf{x}}_{k+1|k}, \qquad (15)$$

for all $k = 0, \cdots, N-1$. Hence Assumptions II and III imply that no new beacon will be observed and the estimation will not be updated after the observations.

Under Assumptions II and III, the multi-step Trajectory Planning Problem is equivalent to the following N-step optimization problem.

N-**Step Optimization Problem**. Given $\hat{\mathbf{x}}_{0|0}$ and $\mathbf{P}_{0|0}$, find $\mathbf{u}_0, \mathbf{u}_1, \cdots, \mathbf{u}_{N-1}$ such that $trace(\mathbf{P}_{N|N})$ is minimized, where $\mathbf{P}_{N|N}$ is given by the following equations:

$$
\begin{aligned}
\mathbf{P}_{1|0} &= \mathbf{F}_{10}\mathbf{P}_{0|0}\mathbf{F}_{10}^T + \mathbf{F}_{20}\Sigma\mathbf{F}_{20}^T, \\
\hat{\mathbf{x}}_{1|0} &= \mathbf{F}(\hat{\mathbf{x}}_{0|0}, \mathbf{u}_0, \mathbf{0}) \\
\mathbf{H}_{10} &= \nabla_{\mathbf{x}}\mathbf{h}_1|_{\hat{\mathbf{x}}_{1|0}} \\
\mathbf{S}_1 &= \mathbf{H}_{10}\mathbf{P}_{1|0}\mathbf{H}_{10}^T + \mathbf{R}_1, \\
\mathbf{K}_1 &= \mathbf{P}_{1|0}\mathbf{H}_{10}^T\mathbf{S}_1^{-1}, \\
\mathbf{P}_{1|1} &= \mathbf{P}_{1|0} - \mathbf{K}_1\mathbf{S}_1\mathbf{K}_1^T, \\
\hat{\mathbf{x}}_{1|1} &= \hat{\mathbf{x}}_{1|0}, \\
&\vdots \\
\mathbf{P}_{N|N-1} &= \mathbf{F}_{1(N-1)}\mathbf{P}_{N-1|N-1}\mathbf{F}_{1(N-1)}^T \\
&\quad + \mathbf{F}_{2(N-1)}\Sigma\mathbf{F}_{2(N-1)}^T, \\
\hat{\mathbf{x}}_{N|N-1} &= \mathbf{F}(\hat{\mathbf{x}}_{N-1|N-1}, \mathbf{u}_{N-1}, \mathbf{0}) \\
\mathbf{H}_{1(N-1)} &= \nabla_{\mathbf{x}}\mathbf{h}_N|_{\hat{\mathbf{x}}_{N|N-1}} \\
\mathbf{S}_N &= \mathbf{H}_{1(N-1)}\mathbf{P}_{N|N-1}\mathbf{H}_{1(N-1)}^T + \mathbf{R}_N, \\
\mathbf{K}_N &= \mathbf{P}_{N|N-1}\mathbf{H}_{1(N-1)}^T\mathbf{S}_N^{-1}, \\
\mathbf{P}_{N|N} &= \mathbf{P}_{N|N-1} - \mathbf{K}_N\mathbf{S}_N\mathbf{K}_N^T.
\end{aligned}
\qquad (16)
$$

V. A NOVEL OPTIMIZATION STRATEGY

A. Feasibility of the multi-step optimization

Assumptions II and III are the only possible assumptions we can make at time 0 if we want to look multi-steps ahead because we have no idea about where the estimated location is deviated from the true location. However, these two assumptions are very conservative. When they do not hold, the model (16) is different from the model (5)-(9) and it is a question whether the optimization results obtained by an inaccurate model is close to the real optimal solution or not.

Also note that as new observations are made, part of the true model (5)-(9) becomes available and the state vector is updated by adding the new observed beacons. So it is not wise to keep using the previously obtained control when the model has changed significantly.

In the next subsection, we will provide an optimization strategy taking into account this concern.

B. The new optimization strategy

The idea of our optimization strategy is to use all the information available at the time of decision making to achieve the goal (minimize $trace(\mathbf{P}_{N|N})$). The detail of the decision making is the following.

At time 0: Perform an N step look-ahead optimization, obtain $\mathbf{u}_0^0, \mathbf{u}_1^0, \cdots, \mathbf{u}_{N-1}^0$, but only apply $\mathbf{u}_0^0$.

At time 1: Use the observation $\mathbf{z}_1$ to update $\hat{\mathbf{x}}_{1|1}$ and $\mathbf{P}_{1|1}$ by (7), then use them to perform an $N-1$ step look-ahead optimization, obtain $\mathbf{u}_1^1, \mathbf{u}_2^1, \cdots, \mathbf{u}_{N-1}^1$, but only apply $\mathbf{u}_1^1$.

The above steps are repeated until

At time $N-1$: Use the updated estimation $\hat{\mathbf{x}}_{N-1|N-1}$ and $\mathbf{P}_{N-1|N-1}$ to perform a one-step look-ahead optimization (greedy), obtain $\mathbf{u}_{N-1}^{N-1}$, apply it to the robot.

In summary, the control actions applied at the time sequence $0, \cdots, N-1$ are respectively

$$\mathbf{u}_0^0, \mathbf{u}_1^1, \cdots, \mathbf{u}_{N-1}^{N-1}. \qquad (17)$$

The strategy is similar to the nonlinear Model Predictive Control (Receding-Horizon Optimal Control) [10], the difference is that in the nonlinear Model Predictive Control, the optimization horizon is fixed at any time step, but here the optimization horizon is changing.

C. Optimality

Now we address the issue of optimality for the proposed strategy.

Suppose we have obtained three controllers (i) greedy controller obtained by solving the one-step optimization problem at each time step, (ii) controller obtained by solving the N-step optimization problem, and (iii) controller obtained by the proposed new optimization strategy.

We use $\mathbf{P}_{N|N}^{greedy}, \mathbf{P}_{N|N}^{Nstep}$ and $\mathbf{P}_{N|N}^{new}$ to denote the final covariance matrix by applying the three controllers. Note that $\mathbf{P}_{N|N}^{greedy}, \mathbf{P}_{N|N}^{Nstep}$ and $\mathbf{P}_{N|N}^{new}$ depend on the true observations.

The following three lemmas provide some comparison among the three optimization strategies.[1]

Lemma 1. $trace(\mathbf{P}_{N|N}^{new}) \leq trace(\mathbf{P}_{N|N}^{Nstep})$.

Suppose $N = 2$, let $\mathbf{U}$ denote the set of all the possible control actions. Denote the one step greedy controller as $\mathbf{u}^{gre}$ and the two-step ($N = 2$) optimal control law as $\mathbf{u}_0^{opt}, \mathbf{u}_1^{opt}$, then the controller obtained by the proposed new strategy is $\mathbf{u}_0^{opt}, \mathbf{u}_1^{gre}$.

Proof of Lemma 1. For the proposed MPC method and the two-step optimization method, the control actions at time 0 are the same — $\mathbf{u}_0^{opt}$, so the models of the optimization problems

---

[1]We only provide a brief proof of these lemmas for the case $N = 2$. The general case can be proved by induction.

at time 1 (after the observation and update) are the same. Since the proposed MPC method uses the optimal controller $\mathbf{u}_1^{gre}$ at time 1, it is better than using $\mathbf{u}_1^{opt}$ (which is optimal for the old model but may not be optimal for the updated model).

Lemma 2. *Under Assumption II, if we regard the innovations as Gaussian random variables with zero mean and small variances, then $trace(\mathbf{P}_{N|N}^{Nstep})$ is also a random variable, its mean value is $trace(\bar{\mathbf{P}}_{N|N}^{Nstep})$, where $\bar{\mathbf{P}}_{N|N}^{Nstep}$ is the final covariance matrix obtained by (16) applying the N-step optimal controller.*

Proof. When a random variable y is Gaussian with mean y_0 and a small variance, any nonlinear function $g(y)$ can also be regarded as Gaussian with mean $g(y_0)$ provided y is close to y_0 (because Taylor Expansion approximation can be applied). Since the innovation $\mathbf{z}_1 - \mathbf{h}_1(\hat{\mathbf{x}}_{1|0})$ is a random variable with zero mean, $\hat{\mathbf{x}}_{1|1}$ is a random variable with mean $\hat{\mathbf{x}}_{1|0}$. Hence $\mathbf{H}_{11}$ in (9) is a random variable with mean being equal to the $\mathbf{H}_{11}$ in (16). Now we can conclude that $\mathbf{P}_{2|2}^{Nstep}$ is a random variable with mean being equal to the $\mathbf{P}_{2|2}$ in (16) and hence $trace(\mathbf{P}_{2|2}^{Nstep})$ is also a random variable with mean $trace(\bar{\mathbf{P}}_{2|2}^{Nstep})$.

Lemma 3. *Under Assumptions II and III, $trace(\mathbf{P}_{N|N}^{new}) = trace(\mathbf{P}_{N|N}^{Nstep}) \leq trace(\mathbf{P}_{N|N}^{greedy})$.*

Proof. It is easy to see when Assumptions II and III hold, $\mathbf{u}_1^{gre}$ is the same as $\mathbf{u}_1^{opt}$ (at least they result in the same objective function value). The greedy controller is just one possible controller, certainly it is not as good as the optimal one.

Lemma 1 says that the proposed MPC optimization strategy is better than the N-step optimization strategy. Lemma 2 says that when Assumption II holds and the innovations are small, the N-step optimization strategy is satisfactory because it optimizes the mean of the true objective function. Lemma 3 says under Assumptions II and III, the new MPC strategy results in the same objective as the N-step optimization strategy, they are both optimal and better than the greedy strategy.

In conclusion, under the assumptions in Lemma 2, at any time step, the proposed optimization strategy always make use of all the information available to optimize the mean value of the true objective in the finite horizon trajectory planning problem. From this point of view, it is optimal.

D. Scalability

Scalability is a key issue in multi-step optimization. The computational complexity increases exponentially when increasing the optimization horizon. Since the information obtained from observations at slightly different observation points are similar in general, it may be applicable to consider only finite control options in order to predict a longer horizon within the fixed computation capacity.

Another problem in the scalability is caused by the number of beacons in the map. When the number of beacons is large, the multi-step optimization procedure becomes time consuming and it is almost impossible to obtain the solution in real time. In this case, more effective SLAM algorithms are required (e.g. [11], [3], [12]).

When the time horizon N is too large, it is impossible to perform the N-step optimization directly. In this case, we suggest to select a suitable receding horizon N_0 according to the computation capacity and to perform the optimization as follows:

(1) From step 0 to $N - N_0$, perform N_0-step optimization and apply the first control action (typical nonlinear MPC)

(2) From step $N - N_0$ to $N - 1$, perform the variant of MPC optimization strategy.

Certainly, the optimality is not guaranteed in this case. But this is the best we can do based on the limited computation capacity.

VI. Simulation – (near) Bearing Only SLAM

A. The Model

The process model (rotate/translate with control $\mathbf{u} = [v, \gamma]^T$ and actuator noise $\mathbf{d_x} = [\delta v, \delta \gamma]^T$) is given by (2) where

$$\mathbf{f}(\mathbf{x}_r, \mathbf{u}, \mathbf{d_x}) = \begin{bmatrix} x_r + (v + \delta v)T \cos[\phi_r + (\gamma + \delta \gamma)T] \\ y_r + (v + \delta v)T \sin[\phi_r + (\gamma + \delta \gamma)T] \\ \phi_r + (\gamma + \delta \gamma)T \end{bmatrix}. \tag{18}$$

Assume that there are m beacons and the robot can observe all the beacons at any time step. Let the range and bearing observation for the ith beacon be r_i and θ_i with sensor noises δr_i and $\delta \theta_i$. We denote

$$\begin{aligned} \mathbf{z} &= [r_1, \theta_1, \cdots, r_m, \theta_m]^T, \\ \mathbf{d_z} &= [\delta r_1, \delta \theta_1, \cdots, \delta r_m, \delta \theta_m]^T, \end{aligned} \tag{19}$$

then in the observation model (3),

$$\mathbf{h}_{k+1}(\mathbf{x}) = \begin{bmatrix} \sqrt{(y_{f_1} - y_r)^2 + (x_{f_1} - x_r)^2} \\ \arctan\left(\frac{y_{f_1} - y_r}{x_{f_1} - x_r}\right) - \phi_r \\ \vdots \\ \sqrt{(y_{f_m} - y_r)^2 + (x_{f_m} - x_r)^2} \\ \arctan\left(\frac{y_{f_m} - y_r}{x_{f_m} - x_r}\right) - \phi_r \end{bmatrix}, \tag{20}$$

which is time independent. We also assume that δr_i is much larger than $\delta \theta_i$.

B. Simulation results

Simulations are conducted for the cases of 1) a circular path, 2) path planned by the greedy approach and, 3) the proposed MPC approach. Fig. 1(a) - 1(c) show the robot path and the landmark estimation uncertainty ellipses. It is observed that the paths followed by the robot are different for the greedy and the proposed MPC approaches.

The optimization objective values[2] are recorded in Table I with different approaches (Sim[3]), choices of the number of landmarks (Mrk) and control discretization (Ctr).

---

[2] results are calculated by $trace(P)/M$, M=no. of states

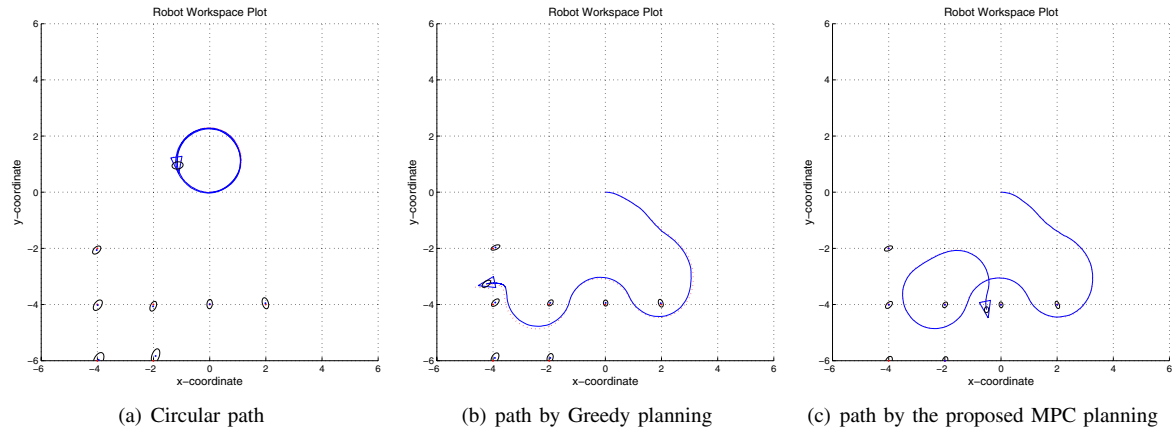[3] 0=circular path, 1=greedy planner, 2=proposed MPC planner

(a) Circular path	(b) path by Greedy planning	(c) path by the proposed MPC planning

Fig. 1.   Robot path and landmark uncertainties

<div align="center">

TABLE I

RESULTS FROM THE THREE APPROACHES

</div>

Sim	Loop	Mrk	Ctr	Result	Loop	Mrk	Ctr	Result
0	50	3	NA	0.0325	50	7	NA	0.0197
0	500	3	NA	0.0047	500	7	NA	0.0016
1	50	3	3	0.0232	50	7	7	0.0210
1	500	3	3	0.0024	500	7	7	0.0014
2	50	3	3	0.0225	50	7	7	0.0181
2	500	3	3	0.0023	500	7	7	0.0013

Simulated results indicate that path planning techniques (greedy or the proposed MPC) perform better than the circular path. Moreover, the proposed MPC trajectory planner slightly out-performs the greedy planner. However, the computational complexity involved may make the approach less attractive when comparing with the performance gained.

VII. CONCLUSION AND FURTHER WORK

In this paper, we consider the trajectory planning in Extended Kalman Filter (EKF) based SLAM. We suppose that the only objective of trajectory planning in SLAM is to minimize the estimation error in a fixed time horizon, and aim at answering the following two questions: (1) Is Multi-Step Look-Ahead Trajectory Planning possible in SLAM? (2) Is Multi-Step Look-Ahead Trajectory Planning really needed in SLAM?

According to the analysis (especially Lemma 2) and simulation results, the answer to the first question is: Yes, provided that (1) the current estimation error is small, (2) the probability of observing new beacons is low, and (3) the computation capability is high.

For the second question, the answer highly depends on the requirement of the map accuracy and the computation capability available. In most of our simulations, the greedy method performs satisfactorily for trajectory planning in SLAM. In general, the multi-steps optimization strategy is computationally expensive and it is probably not worth increasing the computational complexity without any guaranteed rewards. To find a metric to resolve completely the trade-off between the advantage of multi-step trajectory planning and the involved computational cost is an important future research topic.

REFERENCES

[1] R. Smith, M. Self, and P. Cheeseman, "Estimating uncertain spatial relationships in robotics", in *Automomous Robot Vehicles*, I.J. Cox and G.T. Wilfon, Eds, New York: Springer Verleg, 1990, pp. 167-193.
[2] J.A. Castellanos, J. Neira, J.D. Tardos, "Multisensor fusion for simultaneous localization and map building *IEEE Trans. on Robotics and Automation*, vol. 17, no.6, pp. 908-914, 2001.
[3] J.E. Guivant and E. M. Nebot, "Optimization of the simultaneous localization and map building (SLAM) algorithm for real time implementation", *IEEE Trans. on Robotics and Automation*, vol. 17, no.3, pp. 242-257, 2001.
[4] A. A. Makarenko, S.B. Williams, F. Bourgault, and H. F. Durrant-Whyte, "An Experiment in Integrated Exploration". *2002 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2-4 October 2002, Lausane, Switzerland.
[5] S. Huang, Z. Wang, and G. Dissanayake, "Time optimal robot motion control in simultaneous localization and map building (SLAM) problem", *Proceeding of 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems*. Sendai, Japan. pp. 3110-3115.
[6] C. Stachniss, D. Haehnel,and W. Burgard, "Exploration with Active Loop-Closing for FastSLAM", *Proceeding of 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems*. Sendai, Japan. pp. 1505-1510.
[7] A. J. Davison, "SLAM With a Single Camera", *SLAM/CML Workshop at ICRA 2002*. pp. 11-15, May, 2002.
[8] H. Feder, J. Leonard, and C. Smith, "Adaptive mobile robot navigation and mapping", *International J. of Robotics Research*, vol. 18, no. 7, pp. 650-668, 1999.
[9] P. Maybeck, "Stochastic Models, Estimation, and Control", Vol.1, Academic, New York, 1979.
[10] D. Q. Mayne, J.B.Rawlings, C.V. Rao, and P. O. M. Scokaert, "Constrained model predictive control: Stability and optimality", *Automatica*, vol. 36, pp. 789-814, 2000.
[11] S. Thrun, Y. Liu, D. Koller, A.Y. Ng, Z. Ghahramani and H. Durrant-Whyte, "Simultaneous Localization and Mapping with Sparse Extended Information Filters", *International J. of Robotics Research*, vol. 23, no. 7-8, pp. 693-716, 2004.
[12] G. Dissanayake, H. Durrant-Whyte, and T. Bailey, "A computationally efficient solution to the simultaneous localization and map building (SLAM) problem". *In Proceedings IEEE International Conference on Robotics and Automation*. 1009-1014, 2000.