Planning Under Uncertainty using Model Predictive Control for Information Gathering

Cindy Leung, Shoudong Huang, Ngai Kwok, Gamini Dissanayake

ARC Centre of Excellence for Autonomous Systems (CAS)
Faculty of Engineering, University of Technology Sydney, Australia

Abstract

This paper considers trajectory planning problems for autonomous robots in information gathering tasks. The objective of the planning is to maximize the information gathered within a finite time horizon. It is assumed that either the Extended Kalman Filter (EKF) or the Extended Information Filter (EIF) is applied to estimate the features of interest and the information gathered is expressed by the covariance matrix or information matrix. It is shown that the planning process can be formulated as an optimal control problem for a nonlinear control system with a gradually identified model. This naturally leads to the Model Predictive Control (MPC) planning strategy, which uses the updated knowledge about the model to solve a finite horizon optimal control problem at each time step and only executes the first control action. The proposed MPC framework is demonstrated through solutions to two challenging information gathering tasks: 1) Simultaneous planning, localization, and map building (SPLAM) and 2) Multi-robot Geolocation. It is shown that MPC can effectively deal with dynamic constraints, multiple robots/features and a range of objective functions.

Key words: Nonlinear Model Predictive Control, Simultaneous localization and map building (SLAM), Target Localization, Extended Kalman Filter (EKF), Extended Information Filter (EIF), Optimization

1 Introduction

Information gathering via observations is fundamental for many autonomous robot tasks such as target localization, exploration, mapping, search and simultaneous localization and map building (SLAM). These processes are subject to uncertainty about the environment, uncertainty in the sensor measurement and uncertainty in the process model. To enable an autonomous robot to optimally exploit the sensor data, planning for information gathering needs to effectively mitigate the adverse effects of all these uncertainties.

Several techniques currently exist for planning with uncertainty. For example, the uncertainties can be viewed as random variables or a stochastic process. This leads to the field of stochastic control and popular expressions include Markov Decision Processes (MDPs) and their extensions, Partially Observable Markov Decision Processes (POMDPs) [12]. Uncertainties can also be treated as a deterministic signal (with a known hard bound or a known energy bound) which leads to nonlinear l^∞ bounded robust control [10] and nonlinear H^∞ control [9]. All these problems are theoretically solvable and optimal solutions can be obtained by computing the value functions defined for the belief states [12] or the information state [9, 10]. However, the computational cost involved is extremely high and it is difficult to apply these methods in real time planning for information gathering tasks in large scale environments.

Model Predictive Control (MPC), or Receding Horizon Control [1], is another technique that can be applied in the information gathering tasks. MPC has found many applications from its ability to incorporate constraints in the planning process, its limited computation requirements, and its capability of allowing feedback at each planning horizon. Recently, MPC has been applied in trajectory planning for multi-robot planning and control under input/state constraints for collision-avoidance [20], regulation of nonholonomic mobile robots [8], and UAV navigation with passive noisy sensing [5].

In this paper, we consider the case when the information gathering task is formulated by estimating a state vector containing features of interest in the environment. The estimation is performed by applying either the Extended Kalman Filter (EKF) or the Extended Information Filter (EIF). We show that the planning problem in this scenario can be regarded as an optimal control problem for a nonlinear system with a gradually identified model. Then we propose the MPC strategy, which is suitable for online decision making, changing models and dynamic constraints. Two scenarios, *Simultaneous Planning, Localization and Map building (SPLAM)* [4] and *Multi-robot Geolocation* [7] are used as case studies. The first scenario is single-robot SLAM using an EKF. The second scenario is the case where there are multiple robots localizing multiple targets using an EIF.

This paper builds on our recent work [11] and [15]. It is organised as follows. Section 2 formulates the planning problem for the information gathering task as an optimal control problem with a gradually identified model. In Section 3, the MPC strategy for information gathering is introduced. In Section 4, the effectiveness of the strategy is illustrated using simulation results of the two examples. Section 5 discusses issues regarding the MPC strategy along with some related work and possible extensions. Section 6 concludes the paper.

2 The Trajectory Planning Problem for Information Gathering

Suppose there are n robots performing an information gathering task. The features of interest in the environment are expressed by a state vector. The state vector may contain the locations of the features and/or some key properties of the features etc. depending on different applications. When the robots' poses (that is, their positions and orientations) are not known (such as in SLAM), they may also be included in the state vector.

2.1 The Process and Observation Model

Suppose the discrete-time process model of the ith robot is

$$\mathbf{x}_{r_i}(k+1) = \mathbf{f}_i(\mathbf{x}_{r_i}(k), \mathbf{u}_i(k), \mathbf{d}_{\mathbf{x}_i}), \tag{1}$$

where $\mathbf{f}_i$ is a nonlinear function which depends on the type and the dynamic model of heterogeneous robots, $\mathbf{x}_{r_i}$ is the pose of robot i, $\mathbf{u}_i(k)$ is the control input at time k (it may include velocity, turning rate, acceleration etc.) and is assumed to be constant during $[k, k+1)$, and $\mathbf{d}_{\mathbf{x}_i}$ is the zero-mean Gaussian process noise with covariance Σ.

The constraints of the robot's motion are to be incorporated into the planning process. The control constraints and state constraints can be expressed by

$$\mathbf{u}_i(k) \in U_i(k), \quad \mathbf{x}_{r_i}(t) \in X(k+1), \ \forall t \in [k, k+1], \tag{2}$$

where $U_i(k)$ is the set of admissible controls for robot i at time k and $X(k+1)$ describes the safe region for robots during time k to time $k+1$.

The features are assumed to be stationary. Dynamic features may be considered (e.g. in moving target tracking) by introducing the process model for such features, however stability will be an issue (see Section 5.1).

The robots are equipped with sensors that can observe the features. For each robot i, let $J_i(k+1)$ denote the set of the indices of the features that robot i can sense at time $k+1$, which is expressed by

$$J_i(k+1) = \{j_1, \cdots, j_l\}, \tag{3}$$

where the integer l depends on the pose of robot i at time $k+1$, the range of the sensor equipped on robot i, and the feature distribution in the environment. The l features may contain both previously observed features and new features. The the observation model of robot i at time $k+1$ is then

$$\mathbf{z}_i(k+1) = [\mathbf{z}_{ij_1}(k+1), \cdots, \mathbf{z}_{ij_l}(k+1)] \tag{4}$$

where for each feature, $j \in J_i(k+1)$, the observation model is

$$\mathbf{z}_{ij}(k+1) = \mathbf{h}_i(\mathbf{x}_{r_i}(k+1), \mathbf{x}_{f_j}) + \mathbf{d}_{\mathbf{z}_i}, \tag{5}$$

where $\mathbf{x}_{f_j}$ denotes the state of the j-th feature, $\mathbf{h}_i$ is a nonlinear function which depends on the model of the sensor equipped on the ith robot, and $\mathbf{d}_{\mathbf{z}_i}$ is a zero-mean Gaussian measurement noise with covariance matrix $\mathbf{R}_i$.

Furthermore, we denote

$$J(k+1) = [J_1(k+1), \cdots, J_n(k+1)] \tag{6}$$

as the indices of all the features that any robot can observe at time $k+1$, and

$$\mathbf{z}(k+1) = [\mathbf{z}_1(k+1), \cdots, \mathbf{z}_n(k+1)] \tag{7}$$

as all the observations made by the robots at time $k+1$.

2.2 Information Evolution

The information gain from observations can be represented in different forms. For example, in [14], particles are used to represent hypotheses of the current state, and the uncertainty is described by the density of the particles. When the distribution of the state vector can be approximated as a Gaussian distribution, either the Extended Kalman Filter (EKF) or the Extended Information Filter (EIF) can be efficiently used to update the knowledge.

In both the EKF and the EIF (e.g. [19]) formulations, the estimate of the state vector and the corresponding covariance matrix (or information matrix) are updated in a recursive manner. The EKF formula can be summarized as

$$\begin{aligned} \hat{\mathbf{x}}(k+1) &= \hat{\mathbf{F}}_P(\hat{\mathbf{x}}(k), \mathbf{P}(k), \mathbf{u}(k), J(k+1), \mathbf{z}(k+1)), \\ \mathbf{P}(k+1) &= \hat{\mathbf{G}}_P(\hat{\mathbf{x}}(k), \mathbf{P}(k), \mathbf{u}(k), J(k+1)) \end{aligned} \tag{8}$$

and the EIF formula (with the information vector update equation changed into an equivalent state vector update equation) can be summarized as

$$\begin{aligned} \hat{\mathbf{x}}(k+1) &= \hat{\mathbf{F}}_I(\hat{\mathbf{x}}(k), \mathbf{I}(k), \mathbf{u}(k), J(k+1), \mathbf{z}(k+1)) \\ \mathbf{I}(k+1) &= \hat{\mathbf{G}}_I(\hat{\mathbf{x}}(k), \mathbf{I}(k), \mathbf{u}(k), J(k+1)) \end{aligned} \tag{9}$$

where $\hat{\mathbf{x}}(k), \mathbf{P}(k), \mathbf{I}(k)$ and $\hat{\mathbf{x}}(k+1), \mathbf{P}(k+1), \mathbf{I}(k+1)$ denote the state vector estimate, covariance matrix, information matrix at time k and time $k+1$, respectively, $\mathbf{u}(k)$ is the control applied at time k, $J(k+1)$ and $\mathbf{z}(k+1)$

are defined in (6) and (7). The functions $\hat{\mathbf{F}}_P, \hat{\mathbf{G}}_P, \hat{\mathbf{F}}_I, \hat{\mathbf{G}}_I$ are determined by the process models and observation models and the prediction and update formulas in the EKF and/or the EIF.

In the following, we will use two examples, *SPLAM* and *Multi-robot Geolocation*, to illustrate the formulas (8) and (9).

2.2.1 Example 1: Extended Kalman Filter in SPLAM

We consider a single robot SLAM problem. The process model of the whole state vector containing the robot pose $\mathbf{x}_r$ and the locations of all the observed features $\mathbf{x}_f$ is

$$\mathbf{x}(k+1) = \mathbf{F}(\mathbf{x}(k), \mathbf{u}(k), \mathbf{d}_\mathbf{x}) = \begin{bmatrix} \mathbf{f}(\mathbf{x}_r(k), \mathbf{u}(k), \mathbf{d}_\mathbf{x}) \\ \mathbf{x}_f \end{bmatrix}. \tag{10}$$

In the EKF SLAM algorithm, the prediction step is

$$\begin{aligned} \hat{\mathbf{x}}^-(k+1) &= \mathbf{F}(\hat{\mathbf{x}}(k), \mathbf{u}(k), \mathbf{0}) \\ \mathbf{P}^-(k+1) &= \mathbf{F}_1(k)\mathbf{P}(k)\mathbf{F}_1(k)^T + \mathbf{F}_2(k)\Sigma\mathbf{F}_2(k)^T, \end{aligned} \tag{11}$$

where $\hat{\mathbf{x}}(k)$ and $\mathbf{P}(k)$ are the state estimate and covariance matrix at time k (after the update), $\hat{\mathbf{x}}^-(k+1)$ and $\mathbf{P}^-(k+1)$ are the predicted state estimate and covariance matrix at time $k+1$ (before the update), $\mathbf{F}_1(k)$ and $\mathbf{F}_2(k)$ are the Jacobians of $\mathbf{F}$ with respect to $\mathbf{x}$ and $\mathbf{d}_\mathbf{x}$ evaluated at $(\hat{\mathbf{x}}(k), \mathbf{u}(k), \mathbf{0})$, respectively.

Since there is only one robot, the observation at time $k+1$ is

$$\mathbf{z}(k+1) = \mathbf{h}_{k+1}(\mathbf{x}(k+1)) + \mathbf{d}_\mathbf{z} \tag{12}$$

where the function $\mathbf{h}_{k+1}$ and the dimension of the noise vector $\mathbf{d}_\mathbf{z}$ depend on $J(k+1)$.

The update step in EKF SLAM is:

$$\begin{aligned} \hat{\mathbf{x}}(k+1) &= \hat{\mathbf{x}}^-(k+1) + \mathbf{K}(k+1)(\mathbf{z}(k+1) - \mathbf{h}_{k+1}(\hat{\mathbf{x}}^-(k+1))), \\ \mathbf{P}(k+1) &= \mathbf{P}^-(k+1) - \mathbf{K}(k+1)\mathbf{S}(k+1)\mathbf{K}(k+1)^T, \end{aligned} \tag{13}$$

where

$$\begin{aligned} \mathbf{K}(k+1) &= \mathbf{P}^-(k+1)\mathbf{H}_1(k+1)^T\mathbf{S}(k+1)^{-1}, \\ \mathbf{S}(k+1) &= \mathbf{H}_1(k+1)\mathbf{P}^-(k+1)\mathbf{H}_1(k+1)^T + \mathbf{R}, \end{aligned} \tag{14}$$

and $\mathbf{H}_1(k+1)$ is the Jacobian of $\mathbf{h}_{k+1}$ evaluated at $\hat{\mathbf{x}}^-(k+1)$.

By combining equations (11) and (13), we can obtain the formula (8). Note that $\mathbf{H}_1(k+1)$ depends on $J(k+1)$ since $\mathbf{h}_{k+1}$ depends on $J(k+1)$. Also, when new features are observed, the dimension of $\hat{\mathbf{x}}(k+1)$ and $\mathbf{P}(k+1)$ will increase accordingly (for details see e.g. [4]).

2.2.2 Example 2: Extended Information Filter in Multi-robot Geolocation

In Multi-robot Geolocation [7], it is assumed that the poses of the robots can be obtained from an external source such as GPS. The task is to localise the features in the environment as quickly as possible from a sequence of observations. Here, each observation $\mathbf{z}_{ij}$ of feature j from robot i comprises of both bearing and elevation measurements and is given by (5).

Since the poses of robots are available and the features are assumed to be stationary, there is no prediction step. Thus, it is convenient to apply an EIF in the estimation process. Furthermore, because the information about the different features is independent, the information matrix $\mathbf{I}(k)$ is block-diagonal,

$$\mathbf{I}(k) = diag[\mathbf{I}_1(k), \cdots, \mathbf{I}_{m_k}(k)] \tag{15}$$

where m_k is the total number of features detected up to time k, $\mathbf{I}_j(k)$ is the information matrix of feature j at time k, $1 \leq j \leq m_k$.

Let the state estimate at time k be $\hat{\mathbf{x}}_{f_j}(k)$ for $j = 1, \cdots, m_k$. The formulas for updating the estimation using EIF are

$$\begin{aligned} \mathbf{I}_j(k+1) &= \mathbf{I}_j(k) + \sum_{i=1}^{n} \mathbf{H}_{ij}(k+1)^T \mathbf{R}_i^{-1} \mathbf{H}_{ij}(k+1), \\ \hat{\mathbf{x}}_{f_j}(k+1) &= \hat{\mathbf{x}}_{f_j}(k) + \mathbf{I}_j(k+1)^{-1} \sum_{i=1}^{n} \mathbf{i}_{ij}(k+1) \end{aligned} \tag{16}$$

for $j = 1, \cdots, m_k$, where

$$\begin{aligned} \mathbf{i}_{ij}(k+1) &= \begin{cases} \mathbf{H}_{ij}(k+1)^T \mathbf{R}_i^{-1} \mu_{ij}(k+1), & j \in J_i(k+1), \\ 0, & j \notin J_i(k+1), \end{cases} \\ \mu_{ij}(k+1) &= \mathbf{z}_{ij}(k+1) - \mathbf{h}_i(\mathbf{x}_{r_i}(k+1), \hat{\mathbf{x}}_{f_j}(k)), \quad j \in J_i(k+1) \\ \mathbf{H}_{ij}(k+1) &= \begin{cases} \nabla_{\mathbf{x}_{f_j}} \mathbf{h}_i|_{(\mathbf{x}_{r_i}(k+1), \hat{\mathbf{x}}_{f_j}(k))}, & j \in J_i(k+1), \\ 0, & j \notin J_i(k+1), \end{cases} \\ \mathbf{x}_{r_i}(k+1) &= \mathbf{f}_i(\mathbf{x}_{r_i}(k), \mathbf{u}_i(k)), \end{aligned} \tag{17}$$

for $i = 1 \cdots, n,\ j = 1, \cdots, m_k$. If a new feature f_{new} is observed at time

$k+1$, then $\hat{\mathbf{x}}_{f_{new}}(k+1)$ is initialized based on the observation and the information matrix can be computed by the first equation in (16) assuming that the information matrix at time k is zero (see e.g. [7]).

The above formulas have the same form as equation (9) where $\mathbf{I}(k)$ takes the form of (15) and $\hat{\mathbf{x}}(k)$ takes the form of $\hat{\mathbf{x}}(k) = [\hat{\mathbf{x}}_{f_1}(k), \cdots, \hat{\mathbf{x}}_{f_{m_k}}(k)]$.

2.3 Problem Statement

The objective of the planning is to maximise the total information gathered. In an EKF, the information gathered is expressed by the covariance matrix while in an EIF, it is expressed by the information matrix.

Problem Statement. Suppose the total time for the information gathering task is T_0. At time step 0, the robots poses are $\mathbf{x}_{r_i}(0),\ i = 1, \cdots, n$, the initial estimate of the m_0 features observed at time 0 are $\hat{\mathbf{x}}_{f_j}(0),\ j = 1, \cdots, m_0$ and the covariance matrix is $\mathbf{P}(0)$ (the information matrix is $\mathbf{I}(0)$). The task is to choose suitable control actions for the n robots during $[0, T_0)$,

$$\mathbf{u}_i(0), \mathbf{u}_i(1), \cdots, \mathbf{u}_i(T_0 - 1), \qquad i = 1, \cdots, n, \tag{18}$$

such that a quantitative measure of the final covariance matrix $\mathbf{P}(T_0)$ is minimised (or a quantitative measure of the information matrix $\mathbf{I}(T_0)$ is maximised).

Note that different quantitative measures of the covariance matrix (or the information matrix), such as the maximal/minimal eigenvalue, determinant, or trace of the matrix, may be used depending on the application. For example, in our SPLAM example we minimize the trace of the covariance matrix

$$trace(\mathbf{P}(T_0)) \tag{19}$$

while in our Multi-robot Geolocation example, we maximize the minimal eigenvalue of the information matrix

$$\min(\mathrm{eig}(\mathbf{I}(T_0))) = \min_{1 \le j \le m_{T_0}} \min(\mathrm{eig}(\mathbf{I}_j(T_0))). \tag{20}$$

2.4 An Optimal Control Problem for a Gradually Identified Model

The above planning problem can be regarded as a finite-time horizon optimal control problem for a nonlinear control system where the system dynamic is given by (8) (for the EKF) or (9) (for the EIF). The system state includes the estimate of the state vector $\hat{\mathbf{x}}(k)$ and the independent elements in the covariance matrix $\mathbf{P}(k)$ (for the EKF) or the information matrix $\mathbf{I}(k)$ (for the EIF). The objective of the control problem is to minimize $trace(\mathbf{P}(T_0))$ or maximize $\min(\mathrm{eig}(\mathbf{I}(T_0)))$, both of which are functions of the system state.

However, the systems given by (8) and (9) are not ordinary nonlinear discrete-time control systems. The key differences are:

(a) The dimension of the system state changes at various times. For example, as new features are detected in the environment, the dimensions of $\hat{\mathbf{x}}(k)$ and the associated covariance matrix $\mathbf{P}(k)$ (or information matrix $\mathbf{I}(k)$) are enlarged;

(b) The state constraints in (2) are not known *a priori*. For example, the obstacle may not be detected until robot is close enough; and

(c) The dynamics of the system depends on $J(k+1)$ and $\mathbf{z}(k+1)$. But both $J(k+1)$ and $\mathbf{z}(k+1)$ are not available until time step $k+1$. For example, it is difficult to predict the set of features $J(k+1)$ that can be observed due to the inaccurate feature position estimation, the inaccurate robot pose estimation, and the unknown new feature locations. In addition, the true observation $\mathbf{z}(k+1)$ contains sensor noise.

Equations (8) and (9) show that the information at time $k+1$ (either $\mathbf{P}(k+1)$ or $\mathbf{I}(k+1)$) does not depend on the true observation $\mathbf{z}(k+1)$; it only depends on the information at time k, the state vector estimation at time k, the control action taken at time k, and the set of features observed at time $k+1$. However, this is only true for one step prediction. Because the estimation $\hat{\mathbf{x}}(k)$ depends on the noisy observations $\mathbf{z}(1), \cdots, \mathbf{z}(k)$, when we predict $\mathbf{P}(k+1)$ or $\mathbf{I}(k+1)$ at time 0, we need to predict not only which features can be observed at each time step, $J(1), \cdots, J(k+1)$, but also the real observations $\mathbf{z}(1), \cdots, \mathbf{z}(k)$. In the special case when there are no new features and the observation noises are very small (thus the innovations are small and $\hat{\mathbf{x}}(i) \approx \hat{\mathbf{x}}(0), i = 1, \cdots, k$), it is possible to predict $\mathbf{P}(k+1)$ by only predicting $J(1), \cdots, J(k+1)$; this is the method used in [22].

In general, as the information gathering task proceeds, knowledge about the environment accumulates. Firstly, knowledge of the total number of features in the environment and the locations of the obstacles accumulates, so the control system dimension and the state constraints become clearer. Secondly, the estimate of the locations of the features become increasingly accurate, thus $J(k+1)$ and $\mathbf{z}(k+1)$ are increasingly predictable and hence the uncertainties involved in the dynamics of the control systems become smaller and smaller. Because of these, we call the control systems (8) and (9), systems with a *gradually identified model.*

3 Nonlinear Model Predictive Control for Information Gathering

In the information evolution equations (8) and (9), it may be possible to use a general representation such as MDP or POMDP to take into account all the possible dimensions of the state vector, all the possible $J(k+1)$ and $\mathbf{z}(k+1)$ based on our knowledge about the feature distribution in the environment and the sensor noise. However, the model will be extremely complex and it will be difficult to make the planning tractable with the current computational speed. In this paper, we propose to use Model Predictive Control (MPC), a simple but tractable strategy, in the planning process.

3.1 Model Predictive Control

The principle of Model Predictive Control (MPC) is to "look ahead a few steps, but only perform one step." To state clearly, at any time k, an optimal control problem of fixed planning horizon of N steps is solved and a sequence of N control actions

$$\mathbf{u}_i(k), \mathbf{u}_i(k+1), \cdots, \mathbf{u}_i(k+N-1), \qquad i = 1, \cdots, n, \tag{21}$$

is obtained for the n robots, but only the first control action, $\mathbf{u}_i(k), \quad i = 1, \cdots, n$, is applied. This strategy is repeated at each time step k.

As shown in Section 2.4, the model of the information gathering problem is gradually identified. Thus MPC is a natural strategy to solve it because the updated knowledge is used in the new plan at each time step. However, at each time step, a fixed control system model is required to compute the N-step optimal control actions. So we need a way to predict $J(k+1)$ and $\mathbf{z}(k+1)$ for N steps.

There are many different possible $J(k+1)$ and $\mathbf{z}(k+1)$ due to the uncertainty of the environments and the observations, so there are different ways to predict them. In order to plan within a limited computational capacity, we propose to make simple multi-step predictions at each time step — i.e. use the mean value of the current state estimate to predict $J(k+1)$ for N steps assuming no new features, and perfect process models; and use the predicted $J(k+1)$ to predict $\mathbf{z}(k+1)$ assuming zero observation innovations. In the N-step optimal control problem, we also assume the control constraints and the state constraints are not changing.

Since the possible future changes in the model are not considered in each plan, it is crucial to replan as soon as new estimates are available.

3.1.1 Example 1: Multi-Step Prediction for SPLAM

In order to plan multiple steps in SPLAM, we first assume no new features and no new obstacles will be detected.

Assumption I. For any possible control sequence $\mathbf{u}(0), \cdots, \mathbf{u}(N-1)$, the group of features that are predicted to be observed at time $k+1$ ($0 \leq k \leq N-1$) are the same as those that will be really observed at time $k+1$. Moreover, the safe region $X(k+1)$ contains the safe region $X(0)$.

In SPLAM, the real observations are required to update the state. Since the observations $\mathbf{z}(k+1)$ for $k = 0, \cdots, N-1$, are not available at time $k = 0$, we need to predict their values. In the EKF implementation, we always assume that the distribution of the true locations $\mathbf{x}_{k+1}$ is Gaussian with mean $\hat{\mathbf{x}}^-(k+1)$. The observation noises are also assumed to be Gaussian with zero mean. Thus we can say that at time $k = 0$, the innovations $\mathbf{z}(k+1) - \mathbf{h}_{k+1}(\hat{\mathbf{x}}^-(k+1))$ are all random variables with zero mean. So we predict the observations assuming that the innovations are zero at all steps.

Assumption II. The innovations at any time $k+1$ are zero, i.e.

$$\mathbf{z}(k+1) - \mathbf{h}_{k+1}(\hat{\mathbf{x}}^-(k+1)) = 0 \tag{22}$$

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

At time $k = 0$, under Assumptions I and II, the N-step optimal control problem for the gradually identified system becomes an optimal control problem for an ordinary deterministic control system. So we can perform the following multi-step look-ahead optimal control to find

$$\mathbf{u}(0), \cdots, \ \mathbf{u}(k+N-1). \tag{23}$$

***N*-step optimal control problem.** Given $\hat{\mathbf{x}}(0)$ and $\mathbf{P}(0)$, find (23) such that $trace(\mathbf{P}(N))$ is minimized, where $\mathbf{P}(N)$ is given by the following equations:

$$\begin{aligned}
\mathbf{P}^-(1) &= \mathbf{F}_1(0)\mathbf{P}(0)\mathbf{F}_1(0)^T + \mathbf{F}_2(0)\Sigma\mathbf{F}_2(0)^T, \\
\hat{\mathbf{x}}^-(1) &= \mathbf{F}(\hat{\mathbf{x}}(0), \mathbf{u}(0), \mathbf{0}), \\
\mathbf{H}_1(1) &= \nabla_{\mathbf{x}}\mathbf{h}_1|_{\hat{\mathbf{x}}^-(1)} \\
\mathbf{S}(1) &= \mathbf{H}_1(1)\mathbf{P}^-(1)\mathbf{H}_1(1)^T + \mathbf{R}(1), \\
\mathbf{K}(1) &= \mathbf{P}^-(1)\mathbf{H}_1(1)^T\mathbf{S}(1)^{-1}, \\
\mathbf{P}(1) &= \mathbf{P}^-(1) - \mathbf{K}(1)\mathbf{S}(1)\mathbf{K}(1)^T,
\end{aligned} \tag{24}$$

$$\begin{aligned}
\hat{\mathbf{x}}(1) &= \hat{\mathbf{x}}^{-}(1),\\
&\vdots\\
\mathbf{P}^{-}(N) &= \mathbf{F}_1(N-1)\mathbf{P}(N-1)\mathbf{F}_1(N-1)^T + \mathbf{F}_2(N-1)\Sigma\mathbf{F}_2(N-1)^T,\\
\hat{\mathbf{x}}^{-}(N) &= \mathbf{F}(\hat{\mathbf{x}}(N-1), \mathbf{u}(N-1), \mathbf{0}),\\
\mathbf{H}_1(N) &= \nabla_{\mathbf{x}}\mathbf{h}_N|_{\hat{\mathbf{x}}^{-}(N)}\\
\mathbf{S}(N) &= \mathbf{H}_1(N)\mathbf{P}^{-}(N)\mathbf{H}_1(N)^T + \mathbf{R}(N),\\
\mathbf{K}(N) &= \mathbf{P}^{-}(N)\mathbf{H}_1(N)^T\mathbf{S}(N)^{-1},\\
\mathbf{P}(N) &= \mathbf{P}^{-}(N) - \mathbf{K}(N)\mathbf{S}(N)\mathbf{K}(N)^T.
\end{aligned} \tag{25}$$

Following this, at time k, similar assumptions need to be made from time k to $k + N$ and the objective is to *minimise trace*$(\mathbf{P}(k+N))$.

3.1.2 Example 2: Multi-Step Prediction for Multi-robot Geolocation

By making similar assumptions as stated by Assumptions I and II, we can perform an N-step optimal control as follows.

N-step optimal control problem. Given $\mathbf{x}_{r_i}(0), \hat{\mathbf{x}}_{f_j}(0)$ and $\mathbf{I}_j(0)$, choose control (21) to maximize the measure $\min\limits_{1\le j\le m_N} \min(\text{eig}(\mathbf{I}_j(N)))$ where $I_j(N)$ is given by

$$\begin{aligned}
\mathbf{I}_j(1) &= \mathbf{I}_j(0) + \sum_{i=1}^{n}\mathbf{H}_{ij}(1)^T\mathbf{R}_i^{-1}\mathbf{H}_{ij}(1),\\
\mathbf{H}_{ij}(1) &= \begin{cases} \nabla_{\mathbf{x}_{f_j}}\mathbf{h}_i|_{(\mathbf{x}_{r_i}(1),\hat{\mathbf{x}}_{f_j}(0))}, & j \in J_i(1),\\ 0, & j \notin J_i(1),\end{cases}\\
\mathbf{x}_{r_i}(1) &= f_i(\mathbf{x}_{r_i}(0), \mathbf{u}_i(0))\\
&\vdots\\
\mathbf{I}_j(N) &= \mathbf{I}_j(N-1) + \sum_{i=1}^{n}\mathbf{H}_{ij}(N)^T\mathbf{R}_i^{-1}\mathbf{H}_{ij}(N),\\
\mathbf{H}_{ij}(N) &= \begin{cases} \nabla_{\mathbf{x}_{f_j}}\mathbf{h}_i|_{(\mathbf{x}_{r_i}(N),\hat{\mathbf{x}}_{f_j}(0))}, & j \in J_i(N),\\ 0, & j \notin J_i(N),\end{cases}\\
\mathbf{x}_{r_i}(N) &= f_i(\mathbf{x}_{r_i}(N-1), \mathbf{u}_i(N-1))
\end{aligned} \tag{26}$$

for $i = 1\cdots, n,\ j = 1, \cdots, m_N$.

Following this, at time k, similar assumptions need to be made from time k to $k + N$ and the objective is to *maximize* $\min\limits_{1\le j\le m_{k+N}} \min(\text{eig}(\mathbf{I}_j(k+N)))$.

3.2 Number of Prediction Steps in MPC

A question regarding the use of MPC is how long the planning horizon N should be. Intuitively, when the uncertainty involved is large, a short planning horizon with immediate rewards are desired. This is due to the high likelihood that the optimal plan will change significantly once new information is acquired. If the uncertainty is small, longer plans generally equate to obtaining higher benefits. However, the planning horizon N is limited by the computational capacity of the planner. So, the choice for the length of the planning horizon N is application specific.

3.3 Optimization Techniques

Systematic techniques are required to solve the N-step optimal control problem and determine the sequence of control actions. There are many ways to search for the optimal control sequence. The two strategies we used in our examples are the Exhaustive Expansion Tree Search (EETS) and Sequential Quadratic Programming (SQP).

3.3.1 Exhaustive Expansion Tree Search (EETS)

EETS conducts an exhaustive search among a limited number of control sequences. For each robot, the number of possible control options it can take at each time step is N_ω. Each robot i can move to N_ω different poses at time $k+1$ if they were to apply N_ω separate controls over the period k to $k+1$.

For the N-step optimal control problem, different control options may be chosen at different steps from step k to step $k+N-1$. Thus each robot would have $(N_\omega)^N$ different control sequence options. For the n robots, if a centralized optimal control is considered, there will be $(N_\omega)^{Nn}$ combinations of control options available. The information gain for the $(N_\omega)^{Nn}$ options need to be evaluated and compared; hence the computation cost is $O((N_\omega)^{Nn})$.

Constraints of no-go-zones can be enforced by an explicit condition based on predicted feature and robot locations to omit branches from the tree search.

Since no information is obtained when the features are out of the robot's sensor range, it is possible that robots cannot gain any information even after looking at all possible trajectories in N steps. This often occurs when the robot is too far away from the features. In this case, it is beneficial to predict further again if the computational capacity allows. However, if we simply increase the planning horizon N and plan again, the computation load will increase exponentially. In our examples, we choose to double the planning horizon N but reduce the number of control options by keeping a control option unchanged every two steps (e.g. $[k, k+2)$) instead of one step (e.g. $[k, k+1)$). Through this approach, the information gain is evaluated every 2 steps and the computational cost remains the same (the total planning time is doubled). Although this is a coarser search, it allows the robot to look further ahead without severely burdening the processor. If there is still no information gain by doubling the planning horizon and the computational capacity still allows for a replan, then the same strategy is repeated.

3.3.2 Sequential Quadratic Programming (SQP)

The benefit of an exhaustive search is that it finds the global optimum among the finite control options. However, it can be argued that a coarse exhaustive search would not obtain the optimal solution because only a few discrete options are available. We present an alternative method for the N-step optimization. This method is also used to evaluate our results from EETS. We combine EETS with SQP because unlike EETS, SQP is an efficient method for continuous optimization problems. In general feeding SQP with a random guess would not result in good performance because a local optimum is often found. Instead the system is fed the control sequence for each robot that was obtained from EETS. This way the SQP is given a good initial condition so that the coarseness of EETS can be refined.

4 Simulation Results

4.1 Example 1: SPLAM

Simulations for information gathering in SPLAM were conducted using a single Pioneer robot with multiple features in a 2D environment. Large sensor noise is given to the range measurements to create similar conditions to a bearing-only case. The control inputs available are velocity and turn-rate. In addition, the external constraints in the simulations are dynamic. There are safety boundaries around the features as they are considered to be obstacles. When new features are detected, new areas are defined to be no-go-zones.

4.1.1 Compare Paths from Different Strategies

In our first set of simulations, we compared information gathering using random control, fixed control, the greedy method [4] and planning with MPC.

For each of these examples, 20 features are scattered in a 20m^2 search space as can be seen in Figure 1. The tasks of the robot are to discover and localise these features while maintaining a good estimate of itself. A terminal time of $T_0 = 1000$ discrete time steps is allotted to the robot. Each time step from k to $k+1$ is set to 0.4 seconds. The sensor is given a limitation of ±22.5° field of view with a 5 meter range.

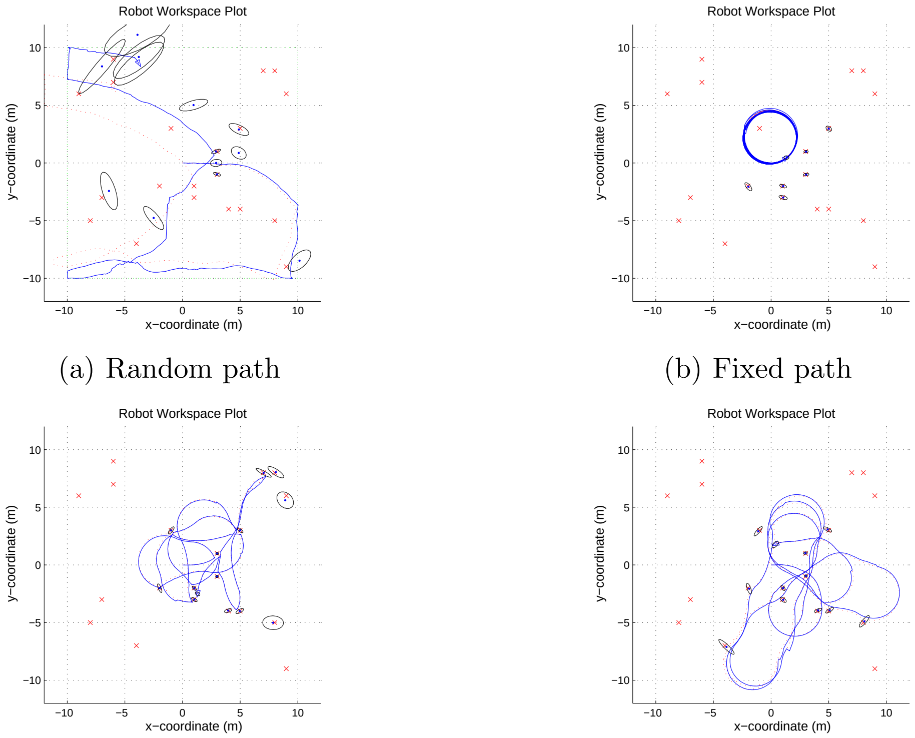
Figure 1(a) shows the path resulting from motions randomly selected from a set of feasible control options. The random path swerves from left to right heading in a predominantly straight line fashion. To prevent the robot from heading directly out of the exploration space, we added an extra constraint. If the robot reaches the edge of the exploration space, it will stop and turn randomly. This constraint is indicated by the green border. It can be seen that at the terminal time, much of the exploration space is covered but the robot is lost and the uncertainties are extremely high.

The result of maintaining a fixed control (constant velocity and turn-rate) can be seen in Figure 1(b). The robot was unable to cover much of the area due to the restricted path. However it was able to localise features along the path quite well because it continuously revisited previously observed features.

A method that is commonly used in information-based strategies is the greedy method [4]. It is a single step optimisation technique. Figure 1(c) shows the path of the greedy method with $N_\omega = 5$ control options. The short-sightedness of this method is apparent when the robot has detected a new feature but does not have the look-ahead to revisit the feature once it has moved past it. Two features with significantly large uncertainties can be seen as a result of the robot only visiting the nearby features that can be observed in the next step.

The path of MPC is shown in Figure 1(d). MPC is implemented using a planning horizon of $N = 5$ and $N_\omega = 5$ control options. The features detected are revisited to minimise the uncertainty of the known map. The detection of features with MPC and the greedy method are only through chance since no explicit exploration strategy is implemented. However there is still a trade-off between coverage and accuracy. Having said this, we compare the two cases of greedy and MPC where a similar amount of features have been identified; it can be seen the final uncertainty from the MPC strategy is smaller.

For the N-step optimal control problem, in Section 3.1, if a fixed number of control options are considered and an EETS is used to select the best option, then the performance of N-step optimization is definitely better than the performance of the one-step optimization (the greedy method). This is because the greedy solution is one of the options considered in the N-step optimization (N-step optimization achieves a better performance at the cost of higher computation load). Similarly, performing one-step optimization is definitely better



(a) Random path

(b) Fixed path

(c) Greedy path

(d) MPC path

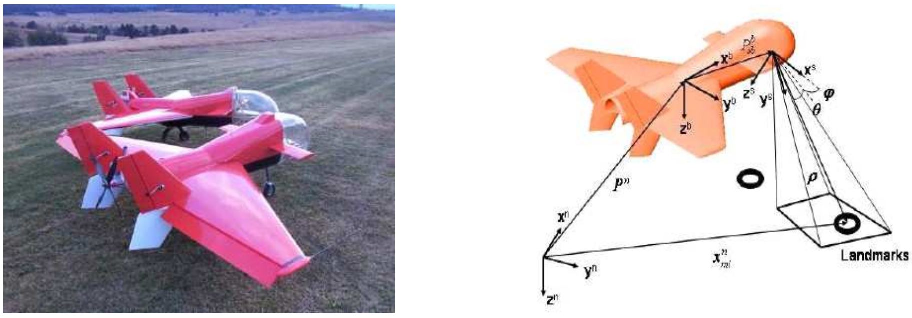
Fig. 1. Robot Path and Feature Uncertainties

than random control and fixed control because the fixed control solution and the random control solution are two options considered in the one-step optimization. However, since we have simplified the original planning problem by making Assumptions I and II and only look-ahead N-steps, there is no guarantee that the proposed MPC strategy is the best for the overall planning problem, though our simulations show that it outperforms the other strategies in most cases.

4.1.2 Increasing Process Noise

In further trials, we removed the sensor limitations to avoid large differences in the number of features detected. Ten trials were conducted each with $N_\omega = 3$ control options, 20 features and 1000 loops. Table 1 shows the average uncertainty of the ten trials and the ratio of the average uncertainties over the greedy method. As can be seen in Table 1, the benefits of planning with a longer lookahead decreases as the process noise increases. However the results in the table show that even at the highest noise level, a planning horizon longer than a single step still performs better than the greedy method.

The level of the benefit attainable by applying MPC also depends on many other factors, such as the environmental conditions (e.g. feature density) and



(a) Brumbies in Field (b) Camera View

Fig. 2. Brumbies at CAS

system constraints (such as field of view, maximum turn-rate).

Table 1
Results from Increasing Process Noise

σ^2 Vel noise	0.03		0.04		0.05		0.06	
σ^2 Turn noise	3°		4°		5°		6°	
N	Trc	R	Trc	R	Trc	R	Trc	R
0 (fixed ctrl)	0.0065	1.14	0.0084	1.17	0.0111	1.22	0.0153	1.31
1 (greedy)	0.0057	1.00	0.0072	1.00	0.0091	1.00	0.0117	1.00
3	0.0052	0.92	0.0067	0.93	0.0085	0.94	0.0112	0.96
5	0.0047	0.83	0.0061	0.85	0.0080	0.88	**0.106**	**0.91**
7	**0.0042**	**0.73**	**0.0055**	**0.77**	**0.0075**	**0.83**	0.0107	0.92

Trc = Average $Trace(\mathbf{P})/(number\ of\ rows\ in\ \mathbf{P})$ of 10 trials
R = Ratio of $Trc/Trc(greedy)$

4.2 Example 2: Multi-robot Geolocation

Simulations for information gathering in Multi-robot Geolocation were conducted using two UAVs (Unmanned Arial Vehicles) and three targets on the ground. The process model for the UAVs is obtained from [13]. This model is particular to the two UAVs (Brumbies) we have at CAS (Centre of Excellence for Autonomous Systems), as seen in Figure 2(a). We set each UAV to fly at different altitudes so inter-robot collision is not relevant in this case.

The control $\mathbf{u}(k)$ to be determined at each step is the roll. Each UAV has a maximum bank angle limiting the roll and turn-rate. The UAVs are equipped with a camera with ±15° field of view. It is mounted on the base of the UAVs so the direction of view of the camera depends on the current roll of the UAVs, as can be seen in Figure 2(b).

4.2.1 MPC vs. Fixed Control

Simulations were conducted using MPC and are compared with the robots moving in a fixed path (constant roll). It was found that the initial fixed path violated the state constraints and the targets were not visible to the robots. The initial poses of the robots were manually repositioned and checked to ensure that each target was visible to at least one robot along the fixed path as seen in Figure 3(b). The constraints of the no-go-zones were also checked to ensure they were not violated. As shown in Figure 3(c), the fixed path remained outside of the green circles.

Ten trials were conducted. The results of the fixed path ($N = 0$) are compared to those from MPC with one ($N = 1$) to five ($N = 5$) step planning horizons. Table 2 contains the results of the average information gain from the ten trials. Results for the fixed control, presented in row $N = 0$, shows that the information gain from MPC is much higher than that of the fixed path.

Furthermore, it can be seen that the information gain from the MPC strategy increased as the planning horizon N increased. After 50 steps, with a planning horizon of $N = 5$, MPC gained on average 426% of the information of the greedy method. The difference in information gain is not as large after 500 steps. This is a result of the planes flying at a fixed altitude high above the features. The eigenvalues of the information matrices in the vertical direction eventually became dominant and could not be increased much further due to the flight constraint. Figure 3(d) shows the vertical offset of the planes and the uncertainties of the features with greater uncertainty in the vertical direction.

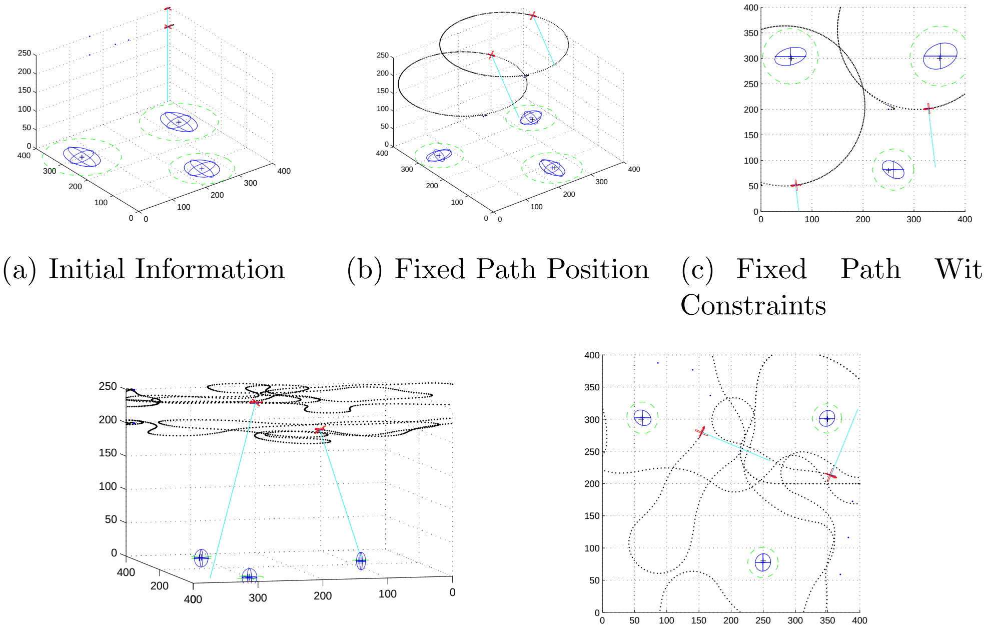
Table 2
Results from the three approaches

N	Loop	N_ω	Info	Ratio	N	Loop	N_ω	Info	Ratio
0 (fixed ctrl)	50	NA	0.0080	0.84	0	500	NA	0.0212	0.53
1 (greedy)	50	3	0.0095	1.00	1	500	3	0.0403	1.00
2	50	3	0.0179	1.88	2	500	3	0.0535	1.33
3	50	3	0.0219	2.31	3	500	3	0.0561	1.39
4	50	3	0.0333	3.51	4	500	3	0.0584	1.45
5	50	3	0.0405	4.26	5	500	3	0.0596	1.48

Info = $\min(eig(\mathbf{I}(T_0)))$, Ratio = Info/Info(greedy).

4.2.2 Dynamic Constraints

We enforced a no-go-zone constraint: the robots cannot fly within the maximal axis of the 95% confidence ellipse of any target in addition to a predefined safety distance. When the uncertainty of the target location is large, this no-go-zone constraint is larger. When the robots gain more information on the target's location, the no-go-zones around the targets reduce accordingly.



(a) Initial Information

(b) Fixed Path Position

(c) Fixed Path Within Constraints

(d) Path of MPC $N = 3$ with vertical offset

(e) Path of MPC Within Constraints

Fig. 3. Brumbies' Flight Path

Figure 3(a) shows the initial information of the target is small thus the initial uncertainty is large, hence the path of the UAV is restricted.

A benefit of using MPC is that varying constraints can be continuously incorporated into the planning process. When the no-go-zones reduce, MPC takes advantage of it and the UAVs move closer to the targets to gain more information. Figure 3(e) shows the top-view of the paths depicting that the constraints of no-go-zones are not violated. If new targets were discovered MPC would also take that into account and plan paths outside of the no-go-zones. Obviously, when using fixed control, dynamic constraints cannot be accommodated.

5 Discussion and Related Work

5.1 Stability of Nonlinear MPC (NMPC)

Linear MPC has established its reputation as a powerful and a broadly-applicable tool [18]. However, the real applications of nonlinear MPC (NMPC) are still limited; one of the major difficulties lies in the stability analysis of NMPC [1].

However, for the information gathering tasks, stability is not critical when the features are stationary. In general, for information gathering tasks, we are only concerned about the stability of robot motion and the convergence of the estimation algorithms. The stability of the robot motion can be guaranteed as long as the control actions are admissible and the obstacles are avoided. The convergence of the estimation algorithm is generally guaranteed for any robot motion. For example, Dissanayake et al. proved that the EKF SLAM algorithm converges as long as successive observations are made from the robot [3]. For the Multi-robot Geolocation problem, the information about the targets can only be increasing (resulting in smaller uncertainties) when observations are made. Thus the uncertainty of the target estimation is always bounded even when no robot trajectory planning is executed.

In many information gathering tasks, as long as the chosen control policy will not crash the robot, the worst case is only to “obtain no information from the environment,” which is often acceptable. This makes the MPC strategy applicable to many information gathering tasks without worrying about the stability issue. Certainly, if dynamic features in the environment are considered, the estimation algorithm may diverge due to poor planning (e.g. losing track of moving targets).

5.2 Computational Complexity

In the proposed MPC strategy, the computational complexity increases exponentially as the planning horizon N is extended. Computational complexity also increases with the number of possible control options at each step and the number of robots (for multi-robot information gathering). We have shown that in SLAM simulations, there is a limit to the benefits gained by increasing the planning horizon especially for systems with large uncertainties, where long-term rewards may not be realized. In our previous work [15] we have shown that for multi-robot geolocation, a coarse discretization of the control options is sufficient to provide a near optimal solution and that combining a coarse EETS with SQP negates the need to implement a fine discretization.

The computational cost also depends on the dimension of state vector and the covariance matrix/information matrix. Many techniques can be used to further reduce the computational complexity in the optimization process. One possible way is to compute the first control, which is implemented, exactly, while approximating the remaining controls, which are not implemented, as proposed by A. Zheng in [1]. Another possible way is to only use part of the nearby features and the corresponding covariance submatrix (information submatrix) in the planning process.

The greedy methods (one-step look ahead) have been proposed for different information gathering tasks. For example, Feder [4] et al. used the greedy method in their SPLAM algorithm. In [23], the greedy method is used in multi-robot mapping. A greedy strategy for information gathering in vision and image processing called the Next Best View (NBV) is used in [17] for the task of 3D object reconstruction. However, it can be easily proved that planning longer than a single step can obtain improved results. Our simulation on SPLAM and Multi-robot Geolocation (see Section 4) have confirmed this.

In contrast, the POMDP [12], nonlinear H^∞ control [9] and nonlinear l^∞ control [10] all aim at providing global optimal control policies. However, *the curse of dimensionality* and *curse of history* [2] are major concerns of these global planning strategies. Though some decomposition and approximation methods are available to obtain the solution (such as clustering [16], implementing a hierarchical POMDP [6] and using parameters to represent approximate continuous states [2]), it is still very difficult to make them tractable in real-time large-scale applications with the current computational speed.

The most powerful aspect of MDPs and POMDPs is that the results from an action can be probabilistic. However, the set of all the possible states of the environment needs to be known before the planning process. In information gathering tasks such as SLAM, the states of interest contain the location of features in the environment. When defining the state space in POMDP before SLAM is performed, all the possible numbers of the features and all the possible locations of features need to be taken into account and this will make the dimension of the state space extremely large. It is also difficult to model the POMDP problem when new state constraints may be detected. If not all the possible constraints are taken into account then the resultant control policy may be far from reliable (needless to say optimal). On the other hand, if all the possible constraints are accounted for then either (i) a lot of parameters need to be identified (which makes the system more complicated), or (ii) the states are only allowed to stay in a very limited region (which significantly limits the solution space and the result will be far from optimal).

In the proposed MPC strategy, it is not necessary to determine the (expected) reward for every possible state. Only a limited number of states within a short planning horizon are evaluated at each time step. When the environment is unknown or dynamic, it is beneficial to generate short term plans with continual feedback from the environment. Observations are used to update the system model to incorporate any changes in the environment. Although the *curse of dimensionality* is also a limitation of MPC, computational complexity may be reduced by maintaining a short planning horizon.

MPC has been applied in many autonomous robot applications such as trajectory planning for multi-robot collision-avoidance [20] and regulation of nonholonomic mobile robots [8]. Recently, Frew [5] proposed to use MPC in a few different problems such as UAV waypoint navigation, trajectory following, safe exploration of unknown environment and aircraft see-and-avoid. However, SLAM is not considered in [5] and the planning is for a single robot. Another difference is that they put constraints in the cost function with a large weight, whereas we put them as hard constraints when selecting the controls. Also, the optimization techniques they used is breath-first random search instead of EETS and SQP as adopted in our work. Regarding robot geolocation, Grocholsky [7] also considered multi-robot target localization using the EIF. But his work mainly focused on sub-optimal decentralized control.

The closest related work on EKF based SPLAM is [22], where an A-optimal global planning strategy in SLAM is proposed. Especially, it was pointed out in [22] that the trace of the covariance matrix is a better measure of map quality in SLAM than the determinant. The main difference between the global planning strategy suggested in [22] and our MPC strategy is that they made the assumption that the approximate locations of all the features are available at the beginning, thus replanning is not that critical. Another difference is that they discretized the environment into a grid and the planning did not take into account the robot dynamics. Furthermore, they approximately expressed the state of the EKF as the estimated position of the robot and the trace of the covariance matrix, which reduces the computational cost significantly and makes global planning possible. In [21], Sim presents an approach to information-driven exploration for SLAM. The robot is driven to a globally optimal position for maximizing information gain of the features. However, the work is for bearing-only SLAM and the main focus is to overcome the stability issues.

5.4 Possible Extensions

Changing objectives: At times, it is necessary to change the control objective in an information gathering task. For example, map accuracy and coverage are two of the most important requirements in the SLAM problem. When the map accuracy reaches a certain level, we may prefer the robot to explore new regions. MPC can easily accommodate the change of control objectives by switching to a different objective in the finite time horizon optimal control.

Including heuristic long-term rewards: In general, MPC works well when short-term actions are more important to our eventual total payoff than long-term actions are, or if there is a high chance that the world will rearrange itself significantly due to some unexpected events before we get to the end of our plan. This is probably true in the initial stage of an information gathering task but not the later stage of the task. Some ways to counteract the short-sightedness of the MPC strategy are possible. For example, a heuristic potential future rewards determined by the current knowledge about the environment (e.g. feature density) can be added in the objective function.

Using possible observation trees: In the prediction step of our current MPC strategy, no new features and zero observation innovations are assumed. Another improvement of the strategy may be to consider a few possible observations in the planning tree (together with the probabilities of their occurrence), and evaluate the average expected rewards. This will result in higher computational costs (or smaller number of steps look-ahead within the given computational capacity) because more branches are required to be evaluated. However, the planning quality may be enhanced significantly.

6 Conclusion and Future Work

In this paper, we proposed to use the Model Predictive Control (MPC) strategy in planning for information gathering tasks where the EKF/EIF is applied in the estimation process. The effectiveness of the strategy is illustrated by two examples; *Simultaneous planning, localization and map building (SPLAM)* and *Multi-robot Geolocation.*

In our opinion, the MPC strategy is suitable for planning in the information gathering tasks for the following reasons:

(a) Changes in the environment are managed by utilising updated models at each time step;
(b) Dynamic constraints which cannot be detected a priori are incorporated into the planning as they change;
(c) The prediction time horizon is flexible and can be made to meet the computational capacity limit;
(d) The stability of MPC is not an issue in general;
(e) The strategy provides an improved control policy compared with greedy methods; and
(f) The strategy is applicable when the objective changes at various times (such as subgoals in an information gathering task).

Although the MPC strategy is proposed for the planning, significant effort is still required to completely solve any specific information gathering task. Unresolved issues include determining the best prediction time horizon and identifying the best optimization techniques. Currently we are working on some possible extensions of the MPC strategy. Further work is needed to consider both stationary features and dynamic features in the environment and to study the related stability issues.

Acknowledgment

This work is supported by the ARC Centre of Excellence programme, funded by the Australian Research Council (ARC) and the New South Wales State Government. The authors would like to thank the anonymous reviewers for their valuable comments and pointing out a few related work.

References

[1] F. Allgower and A. Zheng, *Nonlinear Model Predictive Control*, Boston: Birkhauser, 2000.

[2] A. Brooks, A. Makarenko, S. Williams, H. Durrant-Whyte, *Planning in continuous state spaces with parametric POMDPs*, IJCAI Workshop Reasoning with Undertainty in Robotics, Edinburgh, Scotland, 30 July 2005.

[3] G. Dissanayake, P. Newman, S. Clark, H. Durrant-Whyte, and M. Csorba, *A solution to the simultaneous localization and map building (SLAM) problem*, IEEE Trans. on Robotics and Automation, vol. 17, pp. 229–241, 2001.

[4] H. Feder, J. Leonard, and C. Smith, *Adaptive mobile robot navigation and mapping*, Int. J. of Robotics Research, vol. 18, no. 7, pp. 650-668, 1999.

[5] E. Frew, *Receding horizon control using random search for UAV navigation with passive, non-cooperative sensing.* AIAA Guidance, Navigation, and Control Conference, San Francisco, CA, August 2005.

[6] A. Foka, P. Trahanias, *Real-time hierarchical POMDPs for autonomous robot navigation*, IJCAI Workshop Reasoning with Undertainty in Robotics, Edinburgh, Scotland, 30 July 2005.

[7] B. Grocholsky, *Information-Theoretic Control of Multiple Sensor Platforms*, PhD Thesis, The University of Sydney.

[8] D. B. Gu and H. S. Hu, *A stabilizing receding horizon regulator for nonholonomic mobile robots*, IEEE Trans. on Robotics, 21 (5): 1022-1028 Oct. 2005.

[9] J. W. Helton and M. R. James, *Extending H^∞ Control to Nonlinear Systems: Control of Systems to Achieve Performance Objectives*, Advances in Design and Control, SIAM, 1999.

[10] S. Huang and M. R. James, *l^∞-bounded robustness for nonlinear systems: analysis and synthesis*, IEEE Trans. on Automatic Control, 48(11), 1875-1891, 2003.

[11] S. Huang, N. M. Kwok, G. Dissanayake, Q. P. Ha and G. Fang, *Multi-step look-ahead trajectory planning in SLAM: possibility and necessity*, Proceedings of the 2005 IEEE International Conference on Robotics and Automation, Barcelona, Spain, pp. 1103-1108, April 2005.

[12] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, *Planning and acting in partially observable stochastic domains*, Artificial Intelligence, vol. 101, pp. 99-134, 1998.

[13] J. Kim, S. Ong, E. Nettleton and S. Sukkarieh, *Decentralised approach to unmanned aerial vehicle navigation: without the use of GPS and preloaded map*, Journal of Aerospace Engineering, Institute of Mechanical Engineering, Vol. 218, Part G, pp. 399-416, 2004.

[14] C. Kwok, D. Fox, M. Meila, *Adaptive real-time particle filters for robot localization*, Proceedings of the 2003 IEEE International Conference on Robotics and Automation, September 2003.

[15] C. Leung, S. Huang, G. Dissanayake and T. Furukawa, *Trajectory planning for multiple robots in bearing-only target localisation*, Proceedings of the 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, Edmonton, Canada, pp. 2312-2317, August 2005.

[16] X. Li, W. K. Cheung, J. Liu, *Towards solving large-scale POMDP problems via spatio-temporal belief state clustering*, IJCAI Workshop Reasoning with Undertainty in Robotics, Edinburgh, Scotland, 30 July 2005.

[17] Y. F. Li, Z. G. Liu, *Information entropy-based viewpoint planning for 3-D object recontruction*, IEEE Trans. on Robotics, Vol.21, No.3, pp. 324-337, June 2005

[18] J. M. Maciejowski, *Predictive Control with Constraints*, Prentice Hall, 2002.

[19] P. Maybeck, *Stochastic Models, Estimation, and Control*, Vol.1, Academic, New York, 1979.

[20] D. H. Shim, H. J. Kim, S. Sastry, *Decentralized nonlinear model predictive control of multiple flying robots.* Proceedings of 42nd IEEE Conference on Decision and Control, Maui, Hawaii USA, vol.4, pp. 3621- 3626, Dec. 2003.

[21] R. Sim, *Stablizing information-driven exploration for bearings-only SLAM using range gating*, Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Edmonton, AB, pp. 2745-2750, August 2005.

[22] R. Sim and N. Roy, *Global A-Optimal Robot Exploration in SLAM.* Proceedings of the 2005 IEEE International Conference on Robotics and Automation, Barcelona, Spain, pp. 673-678, April 2005

[23] A. W. Stroupe, T. Balch, *Value-based action selection for observation with robot teams using probabilistic techniques*, Robotics and Autonomous Systems, vol. 50, pp. 85-97, 2005.