Active SLAM using Model Predictive Control and Attractor based Exploration

Cindy Leung, Shoudong Huang, Gamini Dissanayake

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

{cleung, sdhuang, gdissa}@eng.uts.edu.au

Abstract **– Active SLAM poses the challenge for an autonomous robot to plan efficient paths simultaneous to the SLAM process. The uncertainties of the robot, map and sensor measurements, and the dynamic and motion constraints need to be considered in the planning process. In this paper, the active SLAM problem is formulated as an optimal trajectory planning problem. A novel technique is introduced that utilises an attractor combined with local planning strategies such as Model Predictive Control (a.k.a. Receding Horizon) to solve this problem. An attractor provides high level task intentions and incorporates global information about the environment for the local planner eliminating the need for costly global planning with longer horizons. It is demonstrated that trajectory planning with an attractor results in improved performance over systems with local planning alone.**

Index Terms – Path Planning, Simultaneous Planning Localization and Mapping (SPLAM), Optimization, Nonlinear Model Predictive Control (MPC), Extended Kalman Filter (EKF), Exploration

I. INTRODUCTION

Planning actions for SLAM (Simultaneous Localisation and Mapping) requires fast algorithms that can adapt to changes in the environment. Changes occur when new features or obstacles are detected and when the state estimates in the EKF (Extended Kalman Filter) are updated. Platforms such as UAVs (Unmanned Arial Vehicles) can not stop and wait for a planner to compute the next safe informative control action. Local planning strategies, such as Model Predictive Control (MPC), are well suited systems with hard real-time constraints or highly dynamic environments. MPC is simple and fast enabling continuous replanning to incorporate feedback and up to date knowledge of the system state and the environment.

MPC has been applied in many control applications due to its ability to incorporate constraints. In [7], nonlinear MPC is used to stabilise multiple helicopters and conduct planning for obstacle avoidance. In our previous work, MPC is applied to geolocation [4] and SLAM [3] for planning trajectories for information gathering. It has proven to perform well for these tasks. However, through this work, several shortcomings of this algorithm and other similar local planners are revealed. MPC suffers from a short-sighted outlook, it is unable to perceive beyond the set planning horizon. This limitation results in an optimised plan which ignores distant benefits even if they are known to the system. Intuitively, the more knowledge used in planning, the higher the rewards, but extending the planning horizon to achieve this exponentially increases computational cost. MPC is also incapable of proactively exploring as it is unable to predict new features. Thus a method to incorporate global knowledge in the planning process is desirable.

This paper extends our previous work by introducing a novel technique of using an attractor to facilitate the local MPC strategy in optimal trajectory planning. The attractor enables global information along with high level goals, to be incorporated into the planning process. This strategy is then applied to active SLAM, often referred to as SPLAM (Simultaneous Planning Localisation and Mapping). The main considerations in active SLAM are coverage, efficiency and map accuracy. The performance of our technique is analysed based on these criteria.

In Section II several related works are provided. Section III introduces the system models and the EKF for information gathering. Section IV revises the solution to the trajectory optimisation problem using MPC. Section V introduces the technique of the attractor based on a state machine. The simulation results are presented in Section VI followed by a discussion in Section VII. Section VIII concludes the paper.

II. RELATED WORK

Optimal trajectory planning for autonomous robots has long been studied. Rosenblatt [6], developed a reactive navigation system to incorporate multiple objectives. Each objective is considered a behaviour and their task is to weight a set of discrete motion controls based on its expected utility. The utilities from each behaviour are also weighted by a central arbiter depending on the current mission of the system. The control option with the highest expected utility is executed. This technique however, is only implemented for simple reactive behaviours.

Frew [2], used a receding time horizon strategy (similar to MPC) for optimal planning and control of UAVs as it requires limited computational resources and it offers an explicit mechanism for responding to dynamic environments with obstacles. Geolocation was performed incorporating dynamic constraints, though SLAM and coverage are not considered.

Several works have focused on the SPLAM problem. In [11], Stachniss conducted frontier based exploration with SLAM. Using two maps, occupancy grid and topological, active loop-closing is performed. However, maximisation of information gain along the path is not considered.

Feder [1], tackled the active SLAM problem. A greedy approach is applied where an action is chosen given the current knowledge to maximise the information gain in the next measurement. However, it can easily be shown that planning with a longer horizon can achieve improved results.

Makarenko [5], considered localisability during the SLAM exploration process as a form of utility, however the localisability and potential information gain are only considered at the destination. If these factors are considered at each step in the path then greater optimality may be achieved.

Sim, has done much work on active SLAM. In [8], coverage was encouraged by randomly placing dummy features in unexplored areas. A Voronoi graph is used for the path planning; perfect data association and unlimited sensor field of view are assumed. This strategy, however, is not effective for systems with short planning horizons and limited sensing as the dummy features would not influence the robot's decision if they are not visible within the planning horizon.

In [10], Sim conducted active SLAM with a grid based approach and states that optimisation using the trace of the covariance matrix from the EKF performs better than using the determinant. However it is assumed that the robot is quasi-holonomic with unlimited sensing and robot motion constraints are not considered in the planning process. In [9], Sim addresses the stability issue. Features that are too close to the robot which may cause filter instability are blocked by using a virtual minimum range sensor.

III. OPTIMAL TRAJECTORY PLANNING

The aim of this paper is to plan trajectories to enable an autonomous robot performing SLAM to completely cover an unexplored area of a given boundary and to maximise the accuracy of the built map within a prescribed terminal time. This is achieved in two stages. In the first stage the objective is to maximise coverage while maintaining a map at a certain level of accuracy. The second stage starts when the environment has been completely covered and the task is then to minimise the uncertainty of the map.

In the optimal trajectory planning strategy presented in this paper, MPC is implemented to conduct planning as this strategy is proven to perform well [3][7], especially in tasks such as obstacle avoidance, consideration of control constraints, determining the most informative path based on sensor models and local knowledge of the system state.

However MPC is principally a local planning strategy. Information beyond the planning horizon is generally ignored as evident in Fig. 2(d), where the robot is given an area of 20m^2 to explore using EKF-SLAM and MPC, the features detected by the robot are well localised due to the optimization of the map but even after 3000 time steps 6 features remain undetected. Methods to overcome this limitation such as extending the planning horizon may enable the robot to see longer term rewards but it is computationally expensive and exploration remains an outstanding issue. Weighting the objective function may provide incentives for exploration and to incorporate long term rewards; however it is difficult to express the value of long term rewards in short planning horizons and tuning weights are cumbersome.

A novel technique of using an attractor to facilitate MPC in the planning process is proposed. With the attractor, the robot is able to perform goal selection based on the current state of the system and determine instances for goal transition. It also enables global information, such as the direction of points or features of interest, to be incorporated. This strategy is applied to active SLAM using the EKF. The system is described herewith as a discrete time model where actions and observations are made at each time step *k*.

A. *Process Model*

The motion of the autonomous robot conducting active SLAM is modeled by

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

where **f** is a nonlinear function which depends on the dynamic model of the robot, $\mathbf{x}_\mathrm{r}$ is the pose of the robot, $\mathbf{u}(k)$ is the control input at time *k* and is assumed to be constant from *k* to *k* + 1, $\mathbf{d}_\mathbf{x}$ is the zero-mean Gaussian process noise with covariance Σ.

B. *Observations*

At each time step *k*, the robot takes an observation. Let *J*(*k* + 1) denote the set of indices of the features that the robot can sense at time *k* +1,

$$J(k+1) = \{j_1, \ldots, j_q\}, \tag{2}$$

where the integer *q* depends on the pose of the robot at time $k+1$, the range of the sensor and the feature distribution in the environment. The *q* features may contain both previously observed features and new features.

The observation model of the robot at time *k* + 1 is

$$\mathbf{z}(k+1) = [\mathbf{z}_{j1}(k+1), \ldots, \mathbf{z}_{jq}(k+1)] \tag{3}$$

where for each feature $j \in J(k+1)$,

$$\mathbf{z}_j(k+1) = h(\mathbf{x}_r(k+1), \mathbf{x}_{fj}) + \mathbf{d}_\mathbf{z}, \tag{4}$$

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

C. *Information Evolution*

At time k+1, the combination of the prediction and update of the EKF can be summarised as

$$\begin{aligned}\hat{\mathbf{x}}(k+1) &= \hat{\mathbf{F}}(\hat{\mathbf{x}}(k),\ \mathbf{P}(k),\ \mathbf{u}(k),\ J(k+1),\ \mathbf{z}(k+1)) \\ \mathbf{P}(k+1) &= \hat{\mathbf{G}}(\hat{\mathbf{x}}(k),\ \mathbf{P}(k),\ \mathbf{u}(k),\ J(k+1))\end{aligned} \tag{5}$$

where $\hat{\mathbf{x}}$ denotes the state vector estimate (the state vector includes both the robot pose and the features of interest), $\mathbf{P}$ is the associated covariance matrix. The functions $\hat{\mathbf{F}}$ and $\hat{\mathbf{G}}$ are determined by the process models and observation models and the prediction and update formulas in the EKF (for details see [4]).

IV. Model Predictive Control

A. Gradually Identified Model

Optimisation of information gathering requires knowledge of the sensor model, the current state of the world and how the world changes. The features in this system are assumed to be stationary. However in active SLAM, the state of the world is unknown. Knowledge of the world is gathered through taking observations. The state vector containing the features gradually grows as new features are detected. As further observations are taken, the estimated locations of the features also become increasingly accurate.

B. Multi-step Prediction

MPC is applied as this system is gradually identified. The principle of MPC is to "look ahead a few steps, but only perform one step". The planning is recursively computed at every step k, so the most recent estimated model can be incorporated into the planning, enabling continual feedback from the environment.

In the MPC strategy, at each time k, an optimal control problem with fixed planning horizon of N-steps is solved and a sequence of N control actions

$$\mathbf{u}(k),\ \mathbf{u}(k+1),\ \ldots,\ \mathbf{u}(k+N-1) \tag{6}$$

is obtained, with only the first control action, $\mathbf{u}(k)$, is applied.

During each N-step planning process, two assumptions are made: a) there are no new features, and b) the innovations are zero. Through making these assumptions, it is possible to predict multiple steps ahead so as to compute a reasonable plan quickly. Details and justification of these assumptions can be found in our previous work [3][4].

C. Objective Function

In the N-step optimal control problem at time k, the objective is to maximize the information gain from time k to time k+N.

In EKF a small covariance matrix equates to a large amount of information. Hence the task of the N-step optimal control is to *minimise* a particular quantitative measure of the covariance matrix P(k+N). In this paper, the chosen objective of the system is to minimise

$$trace(\mathbf{P}(k+N))\,. \tag{7}$$

Some other quantitative measure of P(k+N) (e.g. the maximal eigenvalue) can be used depends on the applications.

D. Search Technique

Searching for the N-step optimal control options is performed using an Exhaustive Expansion Tree Search (EETS). Basically, the information gains of a set of feasible control options are evaluated for N-steps. The best control option according to the objective function is selected. Further details can be found in our previous work [4].

E. Dynamic and Control Constraints

The constraints of the system include both control constraints and state constraints. Similar to the range-gating technique used by Sim [9], a safety bound is imposed around the features so that the robot does not obtain observations that are too close which may cause instability in the EKF update.

Physical motion constraints are incorporated in the EETS by only selecting from a set of discrete control options within the range of allowable motions (i.e. within the maximum turn-rate and velocity). The constraint of no-go-zones are enforced in the planning process by removing branches from the EETS that violate them.

New constraints such as the appearance of a new feature can easily be incorporated in the next step of the planning.

V. State Machine

A. Effective Exploration Strategy

Exploration of new area is important in active SLAM to ensure coverage. However, the robot is required to revisit known features to localise itself from time to time in order not to get lost. On the other hand, if robot revisit known features too often, the coverage time would increase.

Additionally, when new features are detected, further observations of these features will generally increase the accuracy of its estimation. Localisation of new features is generally more effective if the robot travels between the new features and well known features because the correlation between the good and poor features can then increases.

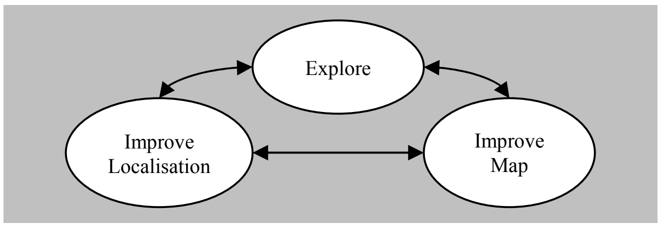


Fig. 1 Exploration States

To conduct active SLAM efficiently, a system is devised with three states: *explore*, *improve localisation* and *improve map*, as seen in Fig. 1. These states are devised for this system because they encapsulate the task intentions of our active SLAM strategy. In the remainder of this paper, these states shall be referred to as goals so as to not get confused with the estimation state.

B. State Transitions

The robot determines its current goal in the state machine based on the uncertainty of the state estimation. When the uncertainty is low (below a predefined threshold), the goal of the robot is to *explore*. If the robot's uncertainty exceeds a threshold then the robot should *improve localisation*. Otherwise the goal would be to *improve map*. Upon visiting a poor feature in the *improve map* goal, the robot either repeatedly switches between the *improve localisation* and *improve map* goals or choose to *explore* depending on the uncertainty of the system state.

If the current goal is *improve localisation, then it will not be changed until the robot uncertainty is reduced to a certain threshold. If the* current goal is one of the other two, it is maintained until either of the following two events occurs: a) The desired destination of the goal is reached, or b) the robot's uncertainty exceeds a predefined threshold triggering a improve localization goal.

These thresholds are determined based on experimental trials. For example, if the robot has a small maximum turn-rate it may need to turn a large circle to return to known features in which case the threshold to switch to the *improve localisation* goal may be a smaller. If the robot's sensor noise is large then the accuracy of the feature estimates will generally have a larger uncertainty and converge at a slower rate so the thresholds for selecting a good or poor feature may be larger.

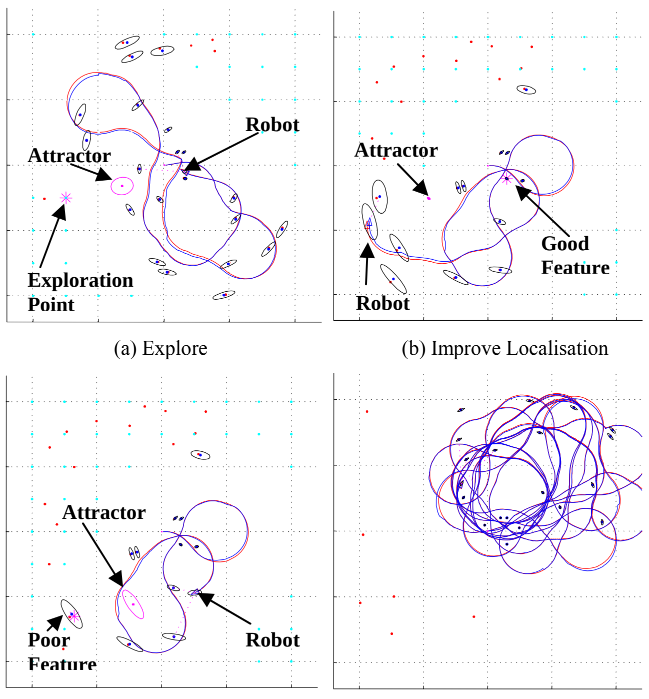
C. Attractor

The state machine is implemented through combining MPC with an attractor. The function of the attractor is to influence the gross motion of the robot to desired locations by influencing the information gain of particular control actions. For the SPLAM example, it is proposed to use the attractor in the form of an artificial feature because the objective function in the MPC strategy is predominately driven by the uncertainty of the features in the map. Through tactical placement of the attractor, the information gain of certain control actions will be increased and thus the motion of the robot will be directed towards the desired goal. In this approach, the local MPC strategy is made unaware of the high level decisions and goals. The goals are incorporated by the directedness of the incentive provided by the attractor.

In each of the goals a reference point is selected heuristically as the destination. When the goal is *explore* an exploration point is selected. When the goal is to *improve localisation* a good feature is selected and when the goal is to *improve map* a poor feature is selected. Selection of the particular exploration point, good feature or poor feature is based on a heuristic of minimum distance so it may be reached quickly.

At each time step, k, the attractor is placed in the direction of the reference point selected. The attractor needs to be placed in a location visible to the robot within the planning horizon in order to influence the robot's decision. The attractor should also be placed in a position further than the distance the robot can travel within the N-steps so the robot does not consider the attractor an obstacle or possibly move past the attractor in the initial steps. It is proposed to place the attractor at a range equivalent to the robot's sensor range, which meets the above two requirements. The optimal distance for the attractor to be placed from the robot is unresolved yet and need further investigation.

The position and covariance of the attractor not only depends on the current goal the robot is in but also on the current knowledge of the system, i.e. $\mathbf{P}(k)$ and $\hat{\mathbf{x}}(k)$.



(a) Explore (b) Improve Localisation

(c) Improve Map (d) MPC without attractor

Fig. 2 Robot Exploring with SLAM, MPC and attractor

D. State Definitions

1) Explore: For this goal the closest exploration point is selected from an exploration point list. Initially the entire search space is covered uniformly with exploration points, as indicated by light blue dots in Fig. 2(a-c). An exploration point represents an unexplored area and is deleted from the list once it is covered by the robot's sensor. These points are distributed with a distance proportional to the robot's sensor range. If the points are too sparse then certain areas may be left unexplored, whereas too many points will result in an unnecessary increase of computation.

The robot is given an artificial state, $\mathbf{x}_{artificial}$, and an artificial covariance, $\mathbf{P}_{artificial}$. The artificial state contains the attractor as a new feature initialised to a position at the robot's sensor range in the direction of the exploration point, i.e.

$$\mathbf{P}_{artifical} = diag(\mathbf{P}(k) \quad \mathbf{P}_{attractor})$$
$$\mathbf{x}_{artificial} = \left[\hat{\mathbf{x}}(k)^T \quad \mathbf{x}_{attractor}{}^T\right]^T . \tag{8}$$

The result of this is depicted in Fig. 2(a), where the attractor has a large uncertainty. The robot sees the attractor as a new feature and is moving to localise it. Once all the exploration points are covered this goal is no longer active.

2*) Improve Localisation*: For this goal, the nearest good feature is selected as the attractor. A good feature is any feature with an uncertainty below a set threshold. If there are no good features then the feature with the lowest uncertainty is selected, j_s represents the index of the feature selected.

Only the state vector, $\hat{\mathbf{x}}\,(k)$, is changed. The position of the good feature selected is altered to be at the robot's sensor range in the direction of the good feature. The covariance $\mathbf{P}(k)$ is left unchanged so the affects of the correlation by observing the feature is maintained, i.e.

$$\begin{aligned}\mathbf{P}_{artifical} &= \mathbf{P}(k)\\ \mathbf{x}_{artifical} &= \hat{\mathbf{x}}(k)\\ \mathbf{x}_{artificial}(j_s) &= \mathbf{x}_{attractor}\end{aligned} \tag{9}$$

Fig. 2(b), captures the robot in the improve localisation goal, where the robot's uncertainty is quite high. The attractor in this case has a low uncertainty equivalent to the uncertainty of the good feature selected. The robot sees the attractor as a good feature and moves towards it to localise.

3*) Improve Map*: For this goal the nearest poor feature is selected as the attractor. A poor feature is any feature with an uncertainty above a set threshold. If there are no poor features then the feature with the highest uncertainty is selected.

Similarly only the state vector, $\hat{\mathbf{x}}\,(k)$, is changed. The position of the poor feature is set to be at the robot's sensor range in the direction of the poor feature. The covariance $\mathbf{P}(k)$ is unchanged so that the covariance of the poor feature is maintained, as in (9).

It can be seen in Fig. 2(c), the uncertainty of the attractor is equivalent to the uncertainty of the poor feature. The robot sees the attractor as a poor feature and moves towards it to localise it. When all the exploration points have been visited, the robot may still localise poor features.

VI. Simulation Results

Table I displays the results of several trials. In each trial, a terminal time of 3000 time steps is applied and 22 features are randomly placed in a 20m^2 area. The uncertainty of the first feature observed becomes the minimum uncertainty of the entire system; hence the time of the observation of the first feature significantly affects the final result. To prevent the robot from moving too far before the first feature can be observed, three additional features are fixed in a position near the starting point of the robot. The sensor range of the robot is set to 5m with field of view of ±45 degrees. The task of the robot is to perform active SLAM.

A. MPC+Attractor vs. MPC alone

Results in Table I reveal that MPC+Attractor perform significantly better than MPC alone in terms of coverage. MPC+Attractor took an average of 1606 time steps to cover the entire exploration space. After 3000 time steps MPC alone only managed to cover an average of 88% of the exploration space. Conversely, MPC alone achieved on average a slightly lower uncertainty of 0.0045 compared to 0.0048 from MPC+ Attactor. Evidently, there is a tradeoff between coverage and information gain. In the MPC+Attractor strategy, the attractor often leads the robot to the edges of the environment to ensure coverage. The exploration points may be far from a cluster of known features and there may be few or no features to be detected at these points. If a new feature is detected at these points the uncertainty would be large because the robot would have traveled a long distance without making observations. When MPC is implemented without the attractor, the robot only traverses near known features to maximise information and new features are detected by chance. The new features detected would hence be reasonably close to known features. As a result the uncertainty does not grow excessively large.

TABLE I

SPLAM Simulation Results

Comparison of Strategies for Active SLAM			
Method	Coverage %	Coverage (time step, *k*)	Avg. Trace*
MPC (3 steps) + Attractor	100	1777	0.0035
	100	1124	0.0035
	100	1643	0.0043
	100	1644	0.0082
	100	1760	0.0039
	100	1685	0.0056
Average	**100**	**1606**	**0.0048**
MPC (3 steps) without Attractor	86	N/A	0.0037
	91	N/A	0.0046
	84	N/A	0.0040
	80	N/A	0.0036
	95	N/A	0.0065
	92	N/A	0.0044
Average	**88**	N/A	**0.0045**
Greedy Method + Attractor	100	2760	0.0081
	100	2460	0.0092
	100	2195	0.0066
	100	2621	0.0043
	100	2594	0.0072
	100	2370	0.0088
Average	**100**	**2500**	**0.0074**

*Trace(**P**) / (number of rows in **P**)

The performance of these strategies largely depends on the density of the features and the sensor range. The larger the sensor range, the higher the amount of features and the smaller the exploration space, the easier the exploration task would be. The differences in Table I are not affected by these factors as the number of features generated, the search space and the robot sensor range are kept constant. The variance of the result was mainly due to the random spread of the features.

B. Multi-step Planning vs. Greedy Method

Further simulations were conducted to observe the level of influence MPC has on the performance of the system when an attractor is present. It can be seen that a 3-step look-ahead performs significantly better than the greedy single-step look-ahead. On average the greedy method took 56% longer to cover the area and the final uncertainty of the system was 54% higher. A reason why 3-steps performs better than a single step is that the robot is able to predict further ahead and see more features hence the trajectory is optimised based on more local knowledge. This shows that even with the attractor providing global knowledge, the local optimisation MPC conducts still plays a major role in the system performance. The robot does not necessarily follow the attractor directly, e.g. if there are 3 features to the left of the robot and the attractor is directly ahead, the 3 features will have a stronger influence on the resulting path. The path towards the goal will hence deviate left to maximise information gain.

VII. Discussion

A. Coverage Improved

Even with the attractor present, there is still no guarantee of complete coverage. In the case where there are no features for a long distance, the robot may become very uncertain of its pose due to the process noise and will be forced to localise every time it attempts to explore in the distance. Having said this, there is significant improvement in terms of coverage when using the attractor as shown in Table I.

B. Obstacle Avoidance

In some cases (depending on the robot's velocity and maximum turn-rate and control options available) there are no feasible control options. This problem is encountered more frequently in the greedy method where the robot does not consider possible obstacles after one step and moves too close to no-go-zones. In the current implementation the robot is forced to stop then turn randomly on the spot. However if the robot is an UAV, it is not possible to stop in mid-flight. This is one of the situations where it is worth planning more than a single step.

C. Localizability

Even with the incentive provided by the attractor to localise when necessary, there is also no guarantee the robot will not become lost by making an incorrect data association when the robot is uncertain of its pose. In some cases, (depending on the robot's maximum turn-rate and control options available) the robot may move to the outskirts of the exploration space to observe new features and would require a long time to turn around to observe features again. If the robot is uncertain of its pose when returning to the exploration space, the first observation may be incorrectly associated.

An approach to improve localizability would be to use more robust methods for data association. Currently the nearest neighbour approach is implemented but other approaches such as joint compatibility test would reduce the occurrence of an incorrect association.

VIII. Conclusion and Future Work

This paper introduces a novel technique of using an attractor together with MPC to optimise information gathering in the task of active SLAM. Combining MPC with an attractor yields promising results. It is revealed that coverage is much improved when using the attractor and that MPC is effective in optimising information gain during the exploration process. Using an attractor with MPC for active SLAM is a good approach in terms of coverage, efficiency and accuracy. In addition, it is shown that planning with MPC using 3-step look-ahead performs better than the greedy method. However, the placement of the attractor requires further research.

This work raises a fundamental question in SLAM: what density of features is necessary for SLAM to be possible? If the features are too sparse, then there is no guarantee for the localization of the robot and thus no guarantee of the coverage. If the features are too dense, then data association is an issue.

Future work includes demonstration of this strategy for active SLAM on a physical platform, developing an explicit method for obstacle avoidance and creating a more informative map with extension beyond point features.

Acknowledgement

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.

References

[1] 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.

[2] E. W. Frew, "Receding Time Horizon Control Using Random Search for UAV Navigation with Passive, Non-cooperative Sensing", *AIAA Guidance, Navigation, and Control Conference*, CA, August 2005

[3] S. Huang, N.M. Kwok, G. Dissanayake, Q. P. Ha, G. Fang; "Multi-Step Look-Ahead Trajectory Planning in SLAM: Possibility and Necessity" *Proc. IEEE Int. Conf. on Robotics and Automation*. pp. 1091-1096, 18-22 April 2005

[4] C. Leung, S. Huang, N. M. Kowk, G. Dissanayake, "Planning under uncertainty using model predictive control for information gathering" *Robotics and Autonomous Systems*. 2006

[5] A. A. Makarenko, S. B. Williams, F. Bourgault, H. F. Durrant-Whyte, "An experiment in integrated exploration", *IEEE/RSJ Int. Conf. on Intelligent Robots and System,* Vol 1, pp. 534-539, 30 Sept.-5 Oct. 2002

[6] J. K. Rosenblatt, "Optimal selection of uncertain actions by maximizing expected utility". *Proc. 1999 IEEE Int. Symposium on Computational Intelligence in Robotics and Automation,* 8-9 pp. 95-100, Nov 1999.

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

[8] R. Sim, "Stable Exploration for Bearings-only SLAM", *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Barcelona, Spain, 6 pages. 2005.

[9] R. Sim, "Stabilizing information-driven exploration for bearings-only SLAM using range gating" *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems,* pp. 2745–2750, 02-06 Aug. 2005

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

[11] C. Stachniss, D. Hahnel, W. Burgard, “Exploration with active loop-closing for FastSLAM” Proc. 2004 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, Volume 2, pp.1505 – 1510, 28 Sept.-2 Oct. 2004