

© [2008] IEEE. Reprinted, with permission, from [Cindy Leung, Shoudong Huang, and Gamini Dissanayake, Active SLAM in Structured Environments, 2008 IEEE International Conference on Robotics and Automation Pasadena, CA, USA, May 19-23, 2008]. This material is posted here with permission of the IEEE. Such permission of the IEEE does not in any way imply IEEE endorsement of any of the University of Technology, Sydney's products or services. Internal or personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution must be obtained from the IEEE by writing to [pubs-permissions@ieee.org](mailto:pubs-permissions@ieee.org). By choosing to view this document, you agree to all provisions of the copyright laws protecting it

# Active SLAM in Structured Environments

Cindy Leung, Shoudong Huang, and Gamini Dissanayake

**Abstract**— This paper considers the trajectory planning problem for line-feature based SLAM in structured indoor environments. The robot poses and line features are estimated using Smooth and Mapping (SAM) which is found to provide more consistent estimates than the Extended Kalman Filter (EKF). The objective of trajectory planning is to minimise the uncertainty of the estimates and to maximise coverage. Trajectory planning is performed using Model Predictive Control (MPC) with an attractor incorporating long term goals. This planning is demonstrated both in simulation and in a real-time experiment with a Pioneer2DX robot.

## I. INTRODUCTION

FOR a robot to map an unknown environment efficiently, simultaneous planning, localization, and mapping is required to be implemented. This problem is called SPLAM or Active SLAM [1]. Lines are common features in indoor environments and can be extracted easily from observations acquired by a laser range finder. Representing the environment with line features instead of point features results in a compact map with a reduced number of parameters. A map based on lines may also be more informative than point features due to the use of more measurements from the laser range finder.

SLAM using line features has been previously achieved [2][3]. Garulli [2] used the EKF as the basis of the estimation and determined the covariance of the observation noise from the statistical properties of the laser range and bearing measurements. Yuen [3] used Sequential Monte Carlo (SMC) SLAM to perform the estimation and demonstrated that the performance is better than that of EKF. Several variations for performing SLAM with line features have been developed [4][5]. The symmetries and perturbations map (SPMap) [4] is one such method, where the robot pose and each feature are treated as geometric entities represented by a quadruple that includes a location vector, a perturbation vector, a covariance matrix and a self-binding matrix. A hierarchical multi-scale strategy [5]

Manuscript received September 14, 2007. This work was supported in part by the ARC Centre of Excellence programme, funded by the Australian Research Council (ARC) and the NSW State Government.

C. Leung is with Faculty of Engineering, the University of Technology, Sydney (corresponding author phone: 61-02-95143140; fax: 61-02-95142655; e-mail: c.leung@cas.edu.au).

S. Huang is with Faculty of Engineering, the University of Technology, Sydney (e-mail: s.huang@cas.edu.au).

G. Dissanayake is with Faculty of Engineering, the University of Technology, Sydney (e-mail: g.dissanayake@cas.edu.au).

has also been developed in which features are grouped into blocks to reduce computation time.

To the authors' knowledge, Active SLAM using line feature estimation is yet to be studied and is the topic of this paper. The traditional implementation of the EKF has however, been shown recently to produce inconsistent estimates when used for line feature SLAM in large-scale structured environments [6]. Our EKF implementation also confirmed the inconsistency. Therefore, another estimator is required before the trajectory planning can be performed.

A Smoothing and Mapping (SAM) technique developed by Dellaert and Kaess in [7] was presented as an accurate and fast method to perform SLAM. In the SAM state vector, the entire robot trajectory is retained thereby the linearization errors embedded into the robot pose are not accumulated. Compared against the EKF, results showed SAM to be faster for large-scale problems while providing consistent estimates. The method was only implemented for point-feature SLAM in [7] but it is viewed as a generic strategy that can also be applied to line feature SLAM.

In this paper, incremental SAM (iSAM) [7] is applied as the underlying estimator of robot poses and line features. Based on the estimates obtained from iSAM, Model predictive control (MPC) based planning strategies, which has shown to be effective for planning in point feature based SLAM [11], are developed for efficient mapping in structured environments. The strategy uses MPC with an attractor to optimise the information gain, aid exploration and to incorporate long term planning. Simulation and experiment results show that the proposed strategy is effective for SLAM in structured environments.

## II. TRAJECTORY PLANNING PROBLEM FOR LINE FEATURE SLAM

Consider a mobile robot placed in an unknown structured environment, which is assumed to contain lines as features. The robot is required to explore this area and build a map.

Observations are taken at each time step  $i$ , where line features are detected and their locations are estimated during the mapping process along with the robot poses.

### A. Representation of a line

A line may be represented in many different ways. In this paper, the two parameters that are used in the SLAM state vector to represent a line are  $\alpha$  and  $d$ , where  $\alpha$  is the angle of the perpendicular to the line from the reference origin and  $d$  is the distance from the reference origin to the line (similar to [8] and [2]). Whereby the equation of the line is

$$x \cos(\alpha) + y \sin(\alpha) = d. \quad (1)$$

There may be many line segments associated with a single line feature. All the individual line segments are recorded separately using the coordinates of the endpoints in an array. This array is recorded and updated but it is not included in the state vector of the SLAM estimator.

The  $q$ th row of the array corresponds to the  $q$ th line segment and is with the form.

$$\Omega_q = [j \quad x_{pt1}^q \quad y_{pt1}^q \quad x_{pt2}^q \quad y_{pt2}^q]. \quad (2)$$

where  $j$  is the associated feature index,  $x_{pt1}^q, y_{pt1}^q$ , and  $x_{pt2}^q, y_{pt2}^q$ , are the coordinates of the two endpoints of the line segment.

### B. Observation model

Lines are extracted from each 2-D laser scan consisting of range and bearing measurements. The line extraction follows the method described in [8]. The algorithm provides parameters of a line given by,  $\alpha_k^{local}$  and  $d_k^{local}$ . These line parameters constitute observation  $\mathbf{z}_k$ . Let the general observation model be

$$\mathbf{z}_k = h_k(\mathbf{x}_i, \boldsymbol{\lambda}_j) + \mathbf{v}_k, \quad (3)$$

where  $\mathbf{x}_i = [x, y, \phi]^T$  is the robot pose where the observation is made,  $\boldsymbol{\lambda}_j = [\alpha_j^{global} \quad d_j^{global}]^T$  is the  $j$ th line feature,  $h_k$  is the relationship between  $\alpha_k^{local}$  and  $d_k^{local}$  to  $\alpha_j^{global}$  and  $d_j^{global}$ , and  $\mathbf{v}_k \sim N(0, \boldsymbol{\Sigma}_k)$  is the zero mean Gaussian noise with covariance  $\boldsymbol{\Sigma}_k$ . This covariance can be calculated using knowledge of the distribution of the sensor noise contained in each laser measurement that constitutes the line observation.

As shown in [2], there are two sets of equations which apply for the observation function depending on the location of the robot with respect to the observed line and the origin of the map.

#### Case 1: Line is not between the robot and origin

$$\begin{aligned} \mathbf{z}_k &= h_k^{case1}(\mathbf{x}, \boldsymbol{\lambda}) + \mathbf{v}_k \\ &= \begin{bmatrix} \alpha_k^{local} \\ d_k^{local} \end{bmatrix} = \begin{bmatrix} \alpha_k^{global} - \phi \\ d_k^{global} - x \cos(\alpha_k^{global}) - y \sin(\alpha_k^{global}) \end{bmatrix} + \mathbf{v}_k. \end{aligned} \quad (4)$$

#### Case 2: Line is between the robot and origin

$$\begin{aligned} \mathbf{z}_k &= h_k^{case2}(\mathbf{x}, \boldsymbol{\lambda}) + \mathbf{v}_k \\ &= \begin{bmatrix} \alpha_k^{local} \\ d_k^{local} \end{bmatrix} = \begin{bmatrix} \alpha_k^{global} - \phi + \pi \\ -d_k^{global} + x \cos(\alpha_k^{global}) + y \sin(\alpha_k^{global}) \end{bmatrix} + \mathbf{v}_k. \end{aligned} \quad (5)$$

### C. Process model

For a small time step, the discrete-time process model is

$$\begin{aligned} \mathbf{x}_{\tau+\Delta t} &= f_\tau(\mathbf{x}_{\tau-1}, \mathbf{v}_\tau, \mathbf{w}_\tau, \Delta t) \\ \mathbf{w}_\tau &\sim N(0, \mathbf{P}_w) \end{aligned} \quad (6)$$

where  $\Delta t$  is the time elapsed between steps. The noises come through the velocity,  $\mathbf{v}_\tau$ , and turn-rate,  $\mathbf{w}_\tau$ , i.e.  $\mathbf{v}_\tau = \mathbf{v}_{true} + \mathbf{v}_{noise}$  and  $\mathbf{w}_\tau = \mathbf{w}_{true} + \mathbf{w}_{noise}$ , and the covariance of the control noise is described by  $\mathbf{P}_w = \text{diag}([\sigma_v^2 \quad \sigma_\omega^2])$ .

To accommodate process noise which has not been modelled, the covariance of a small stabilising noise is added to the covariance of the robot pose at each prediction step, defined as  $\mathbf{Q} = \text{diag}([1e^{-6} \quad 1e^{-6} \quad 1e^{-6}])\Delta t$ .

### D. The Active SLAM problem

The Active SLAM problem considered in this paper is to plan a trajectory for the robot such that the line features in the environment can be estimated accurately and efficiently. To this end, an estimation algorithm is required to properly estimate the line features and robot poses using the observation information and process information.

## III. INCREMENTAL SAM

In attempting to apply the EKF to line feature SLAM it was found that the estimates soon became inconsistent in many simulations. Incremental Smoothing and Mapping (iSAM) [7] is then implemented and proves to perform much better in terms of maintaining consistent estimates. The following describe the iSAM algorithm as adapted to line features, with modifications included to improve the efficiency.

#### A. Relative Pose between Observation Points

One key difference between EKF-SLAM and iSAM is that in iSAM all robot poses are maintained in the state vector. To minimise computation and to avoid recording all the positions between updates, only robot poses where observations are made are included in the state vector. This is achieved by computing the relative position,  $\mathbf{x}_i^{rel}$ , between the observation steps, i.e.

$$\mathbf{x}_i^{rel} = \gamma(\mathbf{x}_{i-1}, \mathbf{x}_i) + \boldsymbol{\eta}_i. \quad (7)$$

Here  $\gamma(\mathbf{x}_{i-1}, \mathbf{x}_i)$  is the function describing the relation of the two poses  $\mathbf{x}_{i-1}$  and  $\mathbf{x}_i$ . The noise term  $\boldsymbol{\eta}_i$  is defined to be Gaussian with zero mean and covariance  $\boldsymbol{\Lambda}_i$ , i.e.  $\boldsymbol{\eta}_i \sim N(0, \boldsymbol{\Lambda}_i)$ . Both  $\mathbf{x}_i^{rel}$  and  $\boldsymbol{\Lambda}_i$  can be obtained by applying the process model (6) for a number of steps using typical EKF prediction formula.

#### B. SLAM as a least squares problem

At time  $i$ , the entire state to be estimated in iSAM includes all robot poses from time  $m=1 \dots i$ , combined with all the landmarks from  $j=1 \dots J$  and is defined by

$$\boldsymbol{\Theta} = [\mathbf{x}_1 \quad \dots \quad \mathbf{x}_i \quad \boldsymbol{\lambda}_1 \quad \dots \quad \boldsymbol{\lambda}_J]^T. \quad (8)$$

In iSAM, the SLAM problem is treated as a least squares problem as follows

$$\Theta^* \triangleq \underset{\Theta}{\operatorname{argmin}} \left\{ \sum_{m=1}^i \left\| \gamma(\mathbf{x}_{m-1}, \mathbf{x}_m) - \mathbf{x}_m^{rel} \right\|_{\Lambda_m}^2 + \sum_{k=1}^K \left\| h_k(\mathbf{x}_{m_k}, \boldsymbol{\lambda}_{j_k}) - \mathbf{z}_k \right\|_{\Sigma_k}^2 \right\}. \quad (9)$$

The first term in the equation (11) is the process innovation, which can be linearised as

$$\begin{aligned} \gamma(\mathbf{x}_{m-1}, \mathbf{x}_m) - \mathbf{x}_m^{rel} &\approx \left\{ \gamma(\mathbf{x}_{m-1}^0, \mathbf{x}_m^0) + \mathbf{F}_m^{m-1} \delta \mathbf{x}_{m-1} + \mathbf{G}_m^m \delta \mathbf{x}_m \right\} - \mathbf{x}_m^{rel} \\ &= \left\{ \mathbf{F}_m^{m-1} \delta \mathbf{x}_{m-1} + \mathbf{G}_m^m \delta \mathbf{x}_m \right\} - \boldsymbol{\varepsilon}_m \end{aligned} \quad (10)$$

where  $\mathbf{F}_m^{m-1}$  and  $\mathbf{G}_m^m$  are the Jacobians of (7), and  $\boldsymbol{\varepsilon}_m$  is the odometry prediction error defined by

$$\boldsymbol{\varepsilon}_m = \mathbf{x}_m^{rel} - \gamma(\mathbf{x}_{m-1}^0, \mathbf{x}_m^0) \quad (11)$$

where  $\mathbf{x}_m^0$  and  $\mathbf{x}_{m-1}^0$  are the initial guesses of the robot poses obtained either from the state vector or through prediction.

The second term in (9) is the observation innovation and can be linearized as

$$\begin{aligned} h_k(\mathbf{x}_{m_k}, \boldsymbol{\lambda}_{j_k}) - \mathbf{z}_k &\approx \left\{ h_k(\mathbf{x}_{m_k}^0, \boldsymbol{\lambda}_{j_k}^0) + \mathbf{H}_k^m \delta \mathbf{x}_{m_k} + \mathbf{J}_k^j \delta \boldsymbol{\lambda}_{j_k} \right\} - \mathbf{z}_k \\ &= \left\{ \mathbf{H}_k^m \delta \mathbf{x}_{m_k} + \mathbf{J}_k^j \delta \boldsymbol{\lambda}_{j_k} \right\} - \boldsymbol{\mu}_k \end{aligned} \quad (12)$$

where  $\boldsymbol{\lambda}_{j_k}^0$  is the initial guess of the feature observed and  $\mathbf{H}_k^m$  and  $\mathbf{J}_k^j$  are Jacobians of (3), and  $\boldsymbol{\mu}_k$  is the measurement prediction error defined by

$$\boldsymbol{\mu}_k = \mathbf{z}_k - h_k(\mathbf{x}_{m_k}^0, \boldsymbol{\lambda}_{j_k}^0). \quad (13)$$

Now the linearized problem is

$$\delta^* = \underset{\delta}{\operatorname{argmin}} \left\{ \sum_{m=1}^i \left\| \mathbf{F}_m^{m-1} \delta \mathbf{x}_{m-1} + \mathbf{G}_m^m \delta \mathbf{x}_m - \boldsymbol{\varepsilon}_m \right\|_{\Lambda_m}^2 + \sum_{k=1}^K \left\| \mathbf{H}_k^m \delta \mathbf{x}_{m_k} + \mathbf{J}_k^j \delta \boldsymbol{\lambda}_{j_k} - \boldsymbol{\mu}_k \right\|_{\Sigma_k}^2 \right\} \quad (14)$$

and the noise covariances  $\Lambda_m$  and  $\Sigma_k$  can be incorporated into the norm using the following property:

$$\left\| e \right\|_{\Sigma}^2 \triangleq e^T \Sigma^{-1} e = (\Sigma^{-T/2} e)^T (\Sigma^{-T/2} e) = \left\| \Sigma^{-T/2} e \right\|_2^2. \quad (15)$$

By collecting the Jacobians into a big matrix  $\mathbf{A}$  and the innovations into a vector  $b$ , the linearised least-squares problem (16) can be solved by

$$\delta^* = \underset{\delta}{\operatorname{argmin}} \left\| \mathbf{A} \delta - b \right\|_2^2. \quad (16)$$

This process is repeated until the solution has converged, i.e.  $\delta$  is smaller than a threshold.

### C. Data association

To use the iSAM estimates for data association, the EKF state and covariance is extracted from the iSAM state vector  $\Theta$  and matrix  $\mathbf{A}$ . First, the information matrix is computed by

$$\mathbf{I} = \mathbf{A}^T \mathbf{A}, \quad (17)$$

then the covariance  $\mathbf{P}$  is simply the inverse of the information matrix, i.e.  $\mathbf{P} = \mathbf{I}^{-1}$ .

From this point, the EKF states and its covariance matrix,  $\mathbf{P}_{EKF}$ , can be extracted by selecting relevant values.

Namely, the current robot pose and all the features. Using these values, the standard nearest neighbour method [9] is applied for the data association.

## IV. TRAJECTORY PLANNING

The primary objective of trajectory planning in feature based SLAM is to minimise the uncertainty of the estimate. Due to the uncertainties involved in the estimation process, the trajectory planning problem is not an optimal control problem with a fixed model. Similar to the arguments in [10], the system is gradually identified, thus Model Predictive Control plus attractor [11] is a good strategy for the planning as the optimisation is considered along the path and not only at the destination point. This strategy has been demonstrated to encourage localisation, mapping and exploration for point feature SLAM and is therefore used again for trajectory planning in line feature SLAM. A number of modifications are applied to adapt the trajectory planning to structured environments and are as follows.

### A. Model Predictive Control

The implementation of MPC for line-feature SLAM requires not only the current state estimate to be considered, but also the array of line segments,  $\boldsymbol{\Omega}_q$ , to determine the robot trajectory. Also, although the underlying estimator for SLAM is iSAM, the EKF continues to be used in the MPC strategy as it is efficient and the estimates remain consistent for short periods. The MPC strategy uses the current estimate from iSAM but the multi-step control optimisation is based on the result from the EKF algorithm. Thus at each planning time step, the updated estimates from iSAM are converted to an EKF form for the planning.

#### 1) Obstacle avoidance

Modifications are made to the obstacle avoidance previously developed in [11]. As not all objects present in an environment are line features, a method is required to avoid all possible obstacles. Given that the sensor range is much greater than the planning horizon of the robot, the current laser scan can be used to determine whether a path is obstacle-free, which is similar to the idea in [12].

#### 2) Prediction of line feature observation

The line observations need to be predicted to determine the information gain for each control option. By use of the

predicted robot pose and the line segments, a prediction of sensor measurements may be made. If the robot is predicted to observe an adequate number of sensor measurements to an estimated line feature then that line is deemed to be observed. Observation noises of range measurements are simulated and all the measurements associated to the line feature estimate are then used to determine the covariance of the predicted line observation.

### B. Attractor Strategy

Since MPC alone is still a local planning algorithm, the Attractor strategy [11] is used to encourage long term goals for the robot. In this strategy, a virtual feature is placed within the scope of the planning horizon to guide the robot towards different goals (reference points).

#### 1) Attractor as a Virtual Point Feature

The attractor in [11] is a virtual point feature. The same form is maintained for line-feature Active SLAM. Defining the attractor to be a line segment requires many parameters to be set and greater processing. The aim here is to maintain simplicity and to minimise computation. By maintaining the attractor as a point feature, it is also clear which direction the robot is encouraged to move and this direction is determined by the reference point.

#### 2) Addition of Occupancy Grid Map

In [11], the reference point for the attractor was determined based on minimum distance and the attractor was placed at the maximum sensor range in the line of sight to the reference point. In a structured environment the minimum distance may not be the minimum traversable distance and there is no guarantee of visibility of the attractor placed at the maximum sensor range. To facilitate exploration, localisation and mapping in structured environments, an occupancy grid map is built in conjunction with the iSAM line-feature map. The occupancy grid map records information for explored and unexplored areas as well as occupied and obstacle-free cells. It is used to determine frontiers for exploration and traversable areas.

#### 3) Reference Point for Exploration

In order to obtain a reference point for the *explore* mode in the state machine (see [11]), the frontier points are extracted from the occupancy grid map. Frontier points are cells on the grid map that are unexplored and adjacent to an empty cell. The frontier points are grouped together into regions and the centres of each frontier region are used as possible reference points. The frontier region is selected based on the minimum absolute bearing to the robot. This encourages the robot to continue exploring in its current direction and minimises turning.

#### 4) Reference Point for Localisation and Mapping

For the *improve localisation* mode and the *improve map* mode [11], a candidate set of well-defined or poorly-defined features are selected as features of interest for the respective modes based on a threshold of their covariance. Determining the reference point from the candidate set of features is not as simple as in point feature SLAM. The

observability of the line segments belonging to the line feature need to be considered. Thus, the robot poses that have previously observed the line feature are used as the potential reference points. These poses can easily be retrieved as all previous data associations are recorded for the iSAM process.

Once a group of potential reference points are obtained, the distance transform [13] is computed for the occupancy grid map and the reference point is selected based on minimum traversable distance.

#### 5) Placement of Attractor

To lead the robot towards the reference point, the attractor is placed within the sensor range. In structured environments the attractor may be obscured by walls, thus its placement needs to be adapted to the environment. To determine this placement, a cell path is planned using the result from the distance transform. Starting from the reference point, the location of the first cell along the path that is visible to the robot is selected as the location of the attractor; refer Figure 1 for an illustration.

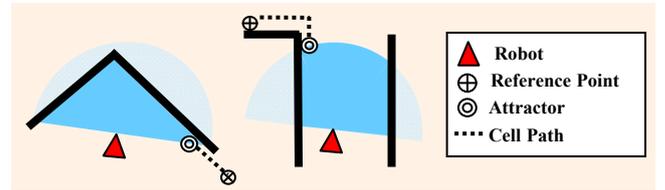


Figure 1 Placement of Attractor

## V. SIMULATION RESULTS

The proposed planning strategy using Attractor aided MPC is demonstrated for line-feature SLAM using iSAM. The simulation results for MPC are also presented so as to provide a framework for discussion and to highlight its distinguishing features. The task of the robot is to map an enclosed indoor environment within a prescribed number of time steps. Three control options,  $N_o=3$ , at each time step are available and a planning horizon of  $D=3$  is used.

To illustrate the result where no planning is conducted, a “random select” method is implemented. The robot also has three available control options and a random selection is made from the options which are free from obstacles.

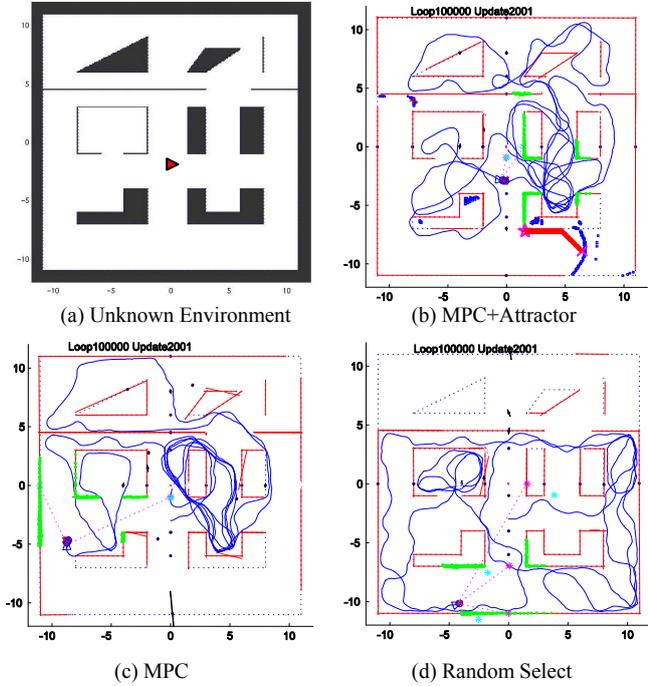
### A. Simulation environment

The robot has an initial pose  $\mathbf{x}_0 = [0 \ -3 \ 0]^T$  as denoted by the red triangle in Figure 2(a). The velocity and turn-rate are recorded each step with  $\Delta t = 0.055$  s and the laser takes a reading every 50 steps. The maximum velocity of the robot is set to be  $0.5\text{ms}^{-1}$  and the maximum turn-rate is  $\pi/36\text{rads}^{-1}$ . The environment contains a wall dividing the area into two separate rooms, as in Figure 2(a). The simulations are run for 100,000 steps.

### B. Map and Path

The resultant map and path for the three methods are

displayed in Figure 2(b)-(d). It can be seen in these maps that there are several false line segments. These false line segments are initialised when the robot turns quickly. The robot, however, is still able to maintain a consistent pose estimate as it is able to correctly associate the observations to the line features in the subsequent steps. The robot pose estimate is accurate as the current laser scan (green) is aligned with the true line segments.



**Figure 2 Map and Path of Simulated Environment**

The behaviour of the robot is quite different for the different strategies. For the MPC path, the robot remained near the centre of the map. There are more line features in the centre of the map and these features are well known as they are close to the starting point of the robot. Observing features near the starting point minimises the robot uncertainty and staying in areas with more features attributes to higher information gain. Hence the robot rarely expands the path to the unknown areas.

For the attractor aided MPC strategy, the path also remains near the starting point of the robot as a result of maximising the information gain of the robot, however in this case, the robot does venture out further due to the attractor pulling the robot towards the frontiers. Thus the attractor once again improved the result of MPC, encouraging the robot to explore. With knowledge of the line segments and the map, unlike the random select which relies on chance, the robot was able to plan paths in and out of the rooms created by the divider.

### C. Final coverage and Uncertainty

The final values for coverage and uncertainty of the state estimate are displayed in Table I. It can be seen that the MPC strategy alone had the lowest uncertainty and the

MPC+Attractor strategy with higher coverage had larger uncertainty as expected.

The random select method has an unusually large uncertainty. This is due to the robot observing features inside the smaller room from the entryway but did not go through as it did not optimise for information gain.

MPC+Attractor		MPC		Random Select	
C	$T_P$	C	$T_P$	C	$T_P$
92.97	0.0010	78.91	0.0009	82.13	0.0016

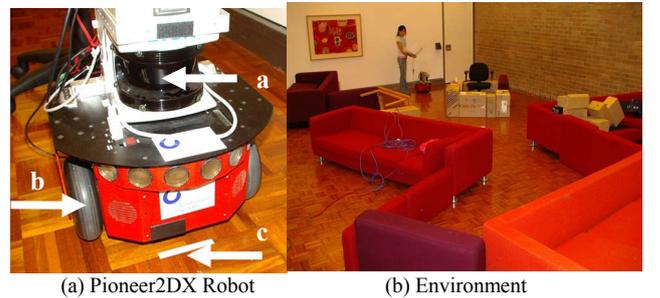
**Table I Average Coverage and Uncertainty Obtained from Ten Trials Mapping Simulated Environment where C=Coverage(%) and  $T_P$ =Trace(P)/(Number of rows in P)**

## VI. EXPERIMENTAL RESULTS

The proposed attractor aided MPC strategy is demonstrated in real-time in a practical experiment.

### A. Experimental setup

The experiment setup is constructed from 4 sofas, several cardboard boxes, cushions and a table as seen in Figure 3(b). A Pioneer2DX robot, pictured in Figure 3(a) is used. This robot is equipped with a SICK laser range finder that allows scans to be taken up to 10Hz. The wheel encoders on the robot are used to obtain robot velocity and turn-rate and runs by differential drive. The robot is set to move at  $0.15\text{ms}^{-1}$  with a maximum turn-rate of  $\pi/22.5\text{rads}^{-1}$  (8deg/s). The processor is 1.7GHz with 512MB of RAM with the software coded in a combination of Matlab and C++.



**Figure 3 Experiment Setup - a=laser sensor, b=wheel encoders, c=marker for start pose**

### B. Extensions for practical implementations

The simulations apply the assumption that the computation time for the algorithms is instantaneous. In practice, the computation time required to process the observations, update the estimations and plan the trajectories is a major issue. Thus a prediction of the computation time between the times of data acquisition and the execution of new controls must be made.

In the experiment, the control measurements between observation time steps are recorded and used to obtain initial value for each robot pose for the iSAM estimation. As the information and planning requirements evolve gradually, the duration of the previous steps are similar to that of the current. Hence the computation time of the

previous step is used to approximate the pose of the robot where the next planned control action is to be applied.

### C. Map and path

Figure 4 displays the resultant map and path from the practical experiment using the MPC+Attractor strategy. The robot trajectory appears quite jagged in the diagrams as only the robot poses at the update steps are displayed.

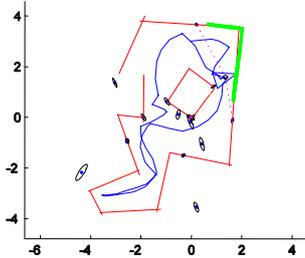


Figure 4 Map and Path from Practical Experiment (unit: m)

### D. Loop and time

The computation time for each loop is displayed in Table II. It can be seen that the computation time for a single loop is significantly larger than that between control measurements. This demonstrates the necessity for the relative pose between updates to be computed to reduce the number of robot poses in the state vector. In parallel to the main thread that performs SLAM and trajectory planning, a control measurement thread is implemented to effectively reduce the linearization error from the process model between update steps. Control measurements are generally taken every 0.05 to 0.085 seconds.

Total Number of Control Measurement Steps	6483
Total number of Observation Steps	150
Total Experimental Time (s)	374
Average Loop Time (s)	2.4784
Average Time between Control Measurements (s)	0.0656
Trace( $\mathbf{P}_T$ ) / number of rows in $\mathbf{P}_T$	0.0022
Number of Features	15

Table II Experimental Results for Attractor aided MPC where  $\mathbf{P}_T$  is the covariance matrix at the Terminal Time

It should be noted that a shorter computation time allows a greater frequency of map updates and hence a more accurate map. The majority of the computation in the loop time is consumed by the MPC+Attractor planning. However despite the computation time required for the trajectory planning, it is demonstrated that the algorithm can work in real-time to perform active SLAM.

## VII. CONCLUSION AND DISCUSSION

The strategy of using MPC and the attractor is proposed for trajectory planning of line feature SLAM. There are many differences in optimising for information gain between SLAM with point features and SLAM with line features, thus a number of modifications are introduced.

The active SLAM algorithm is shown to be effective in

both simulation and experiment using a Pioneer2DX robot. Trajectory planning with MPC+Attractor was successful in mapping the environment in real-time, with the robot moving at low speeds.

Although the iSAM algorithm provided consistent estimates, computation time remains an issue for long experiments. When the number of poses grows large in the iSAM state vector, the computation time increases and the trajectory planning eventually becomes ineffective. As there are generally only a few dominant lines in a structured environment, reducing the poses in the state vector would significantly reduce computation.

Rodriguez-Losada et al. [6], achieved real-time SLAM for large indoor environments using the EKF with ideas adapted from the SPMAP [4]. Trajectory planning with this approach as the underlying estimator is also possible.

## REFERENCES

- [1] R. Sim, and N. Roy. "Global A-Optimal Robot Exploration in SLAM". Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2005), Barcelona, April 2005.
- [2] A. Garulli, A. Giannitrapani, A. Rossi, and A. Vicino, "Mobile robot SLAM for line-based environment representation," 44th IEEE Conf. on Decision and Control, and 2005 European Control Conf.
- [3] D. C. K. Yuen and B. A. MacDonald, "Line-based SMC SLAM Method in Environment with Polygonal Obstacles," Australasian Conference on Robotics & Automation, Brisbane, Australia.
- [4] J. A. Castellanos, J. M. M. Montiel, J. Neira, and J. D. Tardos, 1999, "The SPMAP: a probabilistic framework for simultaneous localization and map building," IEEE Transactions on Robotics and Automation, vol. 15, pp. 948-952.
- [5] S. T. Pfister and J. W. Burdick, 2006, "Multi-scale point and line range data algorithms for mapping and localization," Proceedings 2006 IEEE International Conference on Robotics and Automation.
- [6] D. Rodriguez-Losada, F. Matia, A. Jimenez, and R. Galan, "Consistency improvement for SLAM-EKF for indoor environments". Proceedings of the IEEE Conference on Robotics and Automation, Orlando, Florida, May 2006, pp. 418-423.
- [7] F. Dellaert and M. Kaess, "Square Root SAM: Simultaneous Localization and Mapping via Square Root Information Smoothing," The Int. Journal of Robotics Research, vol. 25, pp. 1181-1203.
- [8] A. Alempijevic, "High-Speed Feature Extraction in Sensor Coordinates for Laser Rangefinders," Australasian Conference on Robotics and Automation, Canberra, Australia
- [9] M. W. M. G. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte, M. Csorba, "A solution to the simultaneous localization and map building (SLAM) problem" IEEE Transactions on Robotics and Automation, June 2001, vol. 17, no. 3, pp. 229-241
- [10] C. Leung, S. Huang, N. M. Kwok and G. Dissanayake, "Planning under Uncertainty using Model Predictive Control for Information Gathering", Robotics and Autonomous Systems, Volume 54, Issue 11, November 2006, pp. 898-910.
- [11] C. Leung, S. Huang, G. Dissanayake, "Active SLAM using Model Predictive Control and Attractor Based Exploration", Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, Beijing, China, pp. 5026-5031, October 2006.
- [12] H. H. Gonzalez-Banos and J.-C. Latombe, "Navigation Strategies for Exploring Indoor Environments," The International Journal of Robotics Research, 2002, vol. 21, pp. 829-848.
- [13] R. A. Jarvis, "Collision-free trajectory planning using the distance transforms", Mechanical Engineering Trans. of the Institution of Engineers, ME10(3), September 1985.