Simultaneous Localization and Sampled Environment Mapping

Rongchuan Sun, Shugen Ma, Member, IEEE, Bin Li, Yuechao Wang

Abstract—Simultaneous localization and map building is a key issue to ensure the mobile robot move in an unknown environment autonomously. A hot topic of SLAM is how to build a map describing the complex environment. This paper presents a new SLAM algorithm using the sampled environment map, which describes the environment in detail, rather than represent the environment with a small number of geometric parameters. The proposed method segments measurements into primitive objects and fits them with implicit polynomials. Algebraic distances or orthogonal distances are then considered as new measurements, which are used to update the whole state. A method considering geometric constraints is presented to remove redundant environment samples from the SEM. The algorithm’s main merits are its compactness and adaptability. Simulation and experimental results demonstrate the efficiency of our algorithm.

I. INTRODUCTION

Simultaneous localization and map building (SLAM) is the problem that a robot explores the environment and builds a map about it, while localizing itself relatively to this map simultaneously [1], [2]. It plays an important role in robotic application and is a challenging research topic.

Over the past two decades, many approaches have been proposed to solve this problem [3], [4], [5], [6], [7]. One of the famous approaches is the extended Kalman filter (EKF) based SLAM. Dissanayake et al. [8] proposed three theorems to prove that the standard EKF-SLAM algorithm is convergent in the case of linear motion and observation model with white Gaussian noise. Montemerlo et al. [9], [10] proposed FastSLAM 1.0 and FastSLAM 2.0, which use the particle filter instead of EKF, aiming to avoid the problem introduced by nonlinear models and non-Gaussian noise.

Early research on SLAM has focused on simplistic environment, in which the obstacles can be described by a small number of geometric parameters. Typical examples are trees in a park [11], corners and walls in office rooms [12]. However, mobile robots usually work in complex environments, in which they do not know the geometric parameters of the obstacle in advance, or the obstacle’s shapes cannot be represented by parameters explicitly. This makes the SLAM algorithm working in complex environments a problem urgent to be solved.

Up to present, there are only a few SLAM algorithms that can describe the environment in detail. One of them uses Rao-Blackwellized particle filter to estimate the mobile robot’s trajectory firstly, then builds a grid map for each particle based on the corresponding trajectory. This kind of method is referred as GridSLAM [13] and GMapping [14]. A similar algorithm is the DP-SLAM, which uses the distributed particle (DP) map instead of the grid map [15]. The difference between the grid map and the DP map is that each single grid in the grid map is a value denoting the probability of occupancy, while it is an ancestry tree in the DP map. Another algorithm is the Scan-SLAM [16], which follows the conventional EKF-SLAM building a landmark map. For each landmark in the map, there is a landmark description template created by extracting the laser measurements corresponding to the object. The process of estimating landmarks’ position together with robot’s pose is the conventional EKF-SLAM, while landmark description template describes the environment in detail additionally. All of the algorithms mentioned above do not build the rich map directly. Instead, the maps are built based on an estimation of the robot’s trajectory or current robot’s pose.

In this paper we present a new approach to localize the robot, while building a rich map at the same time. We adopt the sampled environment map to describe the complex environment. Because this map cannot be updated directly like traditional landmark-based SLAM, we have developed a new observation model to update the state and a method to eliminate the redundant sample points. The sampled environment map used in our algorithm is more compact and adaptable compared with the grid map. In addition, our algorithm updates the robot’s pose and map concurrently in one filter, which enables it a more accurate result potentially. The computational complexity of the algorithm is quadratic on the number of sample points \( N \). However, the proposed approach has the same structure as traditional EKF-SLAM. This makes it feasible to reduce the computational complexity from \( O(N^2) \) to \( O(N) \) [17].

The rest of the paper is organized as follows. Section II describes the sampled environment map. We present a new observation model in Section III and propose the algorithm of simultaneous localization and sampled environment mapping in Section IV. Section V presents a method to remove the redundant sample points from the map. Finally, we give the results of the proposed algorithm with simulation and experiment in Section VI and conclusions in section VII.
II. SAMPLED ENVIRONMENT MAP

Leal introduced the sampled environment map (SEM) [18] to represent the complex environments. This method approximates the obstacle’s contour to a set of discrete, infinitesimal points, referred to environment samples. All the environment samples form the sampled environment map. As shown in Fig. 1, the environment, which is a wall actually, is approximated by a SEM composed of 20 sample points.

Note that, there is no one-to-one correspondence between the sample points in the map and the points on the contour. All what we know is that the sample points lie on the contour, and is considered as a geometrical constraint in Section V.

![Fig. 1. (a) An environment containing an arciform wall with a groove. (b) A SEM of the environment](image)

As declared by Leal, SEM is more compact and adaptable compared with grid map [18]. SEM can allocate more points to an interested contour. There is no need to allocate space to store the area without obstacles. In contrast, grid map has to allocate the storage for the entire domain a priori, and is not easy to increase the resolution for the interested area individually. Extending the SEM representation in detail only needs to add sample points to a list. Conversely, reducing detail of the representation only needs to remove sample points from a list.

In [18], the SEM was built under the assumption of knowing robot’s pose and was updated through a sample-based Bayesian data-fusion technique. When we want to realize SLAM algorithm by using the SEM, the most difficult problem is that there is no one-to-one correspondence between the sample points in the map and the points in measurements to calculate the innovation. Moreover, the robot’s pose and sample points cannot be estimated at the same time with EKF because of the lack of correlation between them.

In the following sections, we will first give a new observation model, then present our SLAM algorithm based on the model. Finally we will give a method to eliminate the redundant sample points in SEM.

III. A NEW OBSERVATION MODEL

To update the state, the standard Kalman filter needs to calculate the innovation between the predicted value and the observed one. This requires the measurement to be an explicit function of the state. When we use the SEM in SLAM, however, the measurement is not a function of the robot’s pose and environment samples directly, because there is no one-to-one correspondence between the sample points in the map and the points in the measurement. Thus we propose another observation model as below.

Suppose that the measurement, usually collected by a scanning laser range finder, is a set of range-bearing points, denoted as:

\[
\mathbf{z} = \begin{bmatrix} r_1 & r_2 & \cdots & r_N \\ \theta_1 & \theta_2 & \cdots & \theta_N \end{bmatrix}
\]  

(1)

The raw measurement \( \mathbf{z} \) is translated into Cartesian coordinate, and expressed as:

\[
\mathbf{z}_x = \begin{bmatrix} x_1 & x_2 & \cdots & x_N \\ y_1 & y_2 & \cdots & y_N \end{bmatrix}
\]  

(2)

The points set \( \mathbf{z}_x \), denoted by equation (2), is clustered and segmented into primitive objects [19], which are then fitted with implicit polynomials (IP) of degree \( n \), \( p(x, y) = 0 \), as:

\[
p_{00} + p_{10}x + p_{01}y + p_{20}x^2 + p_{11}xy + p_{02}y^2 + p_{30}x^3 + p_{21}x^2y + p_{12}xy^2 + p_{03}y^3 + \cdots + p_{n0}x^n + p_{n-1,1}x^{n-1}y + \cdots + p_{0n}y^n = 0
\]  

(3)

To make the polynomial unique, we add an additional coefficients constraint:

\[
p_{00}^2 + p_{10}^2 + p_{01}^2 + p_{20}^2 + \cdots + p_{0n}^2 = 1
\]  

(4)

For indoor environments, we find that the first and second order polynomial are good enough to fit most of the obstacles. The objects that cannot be fitted well by a low order polynomial are again divided into smaller objects. In this way, all of the objects can be fitted well with a first or second order polynomial.

Fitting a set of points with a first order polynomial can be solved by minimizing mean square error (MSE) produced by \( y = kx + b \) or \( x = ly + c \). Considering the numerical singularity, we fit the data with both of the two forms at the same time. Then we select a better one and translate it into the standard IP model (3).

Fitting a set of points with a second order polynomial can be solved by 3L-Fitting method quickly [20], [21].

When all the objects are fitted with an IP successfully, they are used to generate new observations of sample points.

Suppose at time instant \( k + 1 \), we have divided the measurements and fitted them with IPs \( \{p_1, p_2, \cdots, p_m\} \), then we consider the algebraic distance and orthogonal distance as a kind of measurement of the sample points. The sample points will converge to the true contour if the algebraic distance or orthogonal distance approaches zero.

Given a sample point \( [x \ y]^T \), its algebraic distance and orthogonal distance are calculated as:

1. Algebraic Distance:
   
   The algebraic distance is the polynomial’s value calculated at point \( [x \ y]^T \)
   
   \[
d_{alg} = p(x, y)
\]  

(5)

2. Orthogonal Distance
Orthogonal distance is the Euclidean distance from point \([x y]^T\) to the curve \(p\) and is approximately calculated as:

\[
d^\text{oth} = |p(x, y)|/ \| \nabla p_{xy} \|
\]  
(6)

The Jacobians of \(d^\text{oth}\) is not continuous in all the defined intervals. So we use the signed function value instead of the absolute one. Thus, the distance is:

\[
d^\text{oth} = p(x, y)/ \| \nabla p_{xy} \|
\]  
(7)

For each sample points, the measurement is supposed to be zero, and the predicted value is calculated by (5) or (7).

In the rest of the paper, the distance function is written in a general form as:

\[
d = g(x, y)
\]  
(8)

We take \(g(x, y)\) as the new observation model used in our algorithm.

IV. OUR ALGORITHM — SLASEM

The proposed algorithm, namely simultaneous localization and sampled environment mapping (SLASEM), holds the same structure as conventional EKF-SLAM, which is an iteration of motion prediction and measurement update.

Denote the state at time instant \(k\) as \(X(k)\), which is a combination of the robot’s configuration and sample points’ position

\[
X(k) = [X_r^T(k) \quad X_{s1}^T \quad \ldots \quad X_{sn}^T]^T
\]  
(9)

where \(X_r\) describes the robot’s configuration, which is composed of position and orientation, and \(X_{si}(i = 1 \ldots n)\) is the sample point’s position, which is assumed to be lying on the obstacle’s contour. The mean and covariance of the state \(X(k)\) are denoted by \(\hat{X}(k|k)\) and \(P(k|k)\), and described as:

\[
\hat{X}^T(k|k) = [\hat{X}_r^T(k|k) \quad \hat{X}_{s1}^T \quad \ldots \quad \hat{X}_{sn}^T]^T
\]  
(10)

\[
P(k|k) = 
\begin{bmatrix}
P_{rr}(k|k) & P_{r1}(k|k) & \ldots & P_{rn}(k|k) \\
P_{1r}(k|k) & P_{11}(k|k) & \ldots & P_{1n}(k|k) \\
\vdots & \vdots & \ddots & \vdots \\
P_{nr}(k|k) & P_{n1}(k|k) & \ldots & P_{nn}(k|k)
\end{bmatrix}
\]  
(11)

At time instant \(k + 1\), the robot moves to \(X(k + 1)\) and gets observations \(z(k + 1)\). The motion model is

\[
X(k + 1) = f(X(k), u(k), w_k)
\]  
(12)

where \(u(k)\) is the control input and \(w_k\) is assumed to be zero mean uncorrelated Gaussian distributed noise with covariance \(Q_k\).

The robot gets measurement \(z(k + 1)\) at point \(X(k + 1)\). Following the method described in Section III, we form a new observation function \(h(X(k + 1), V_{k+1})\), which is a function vector composed by distance functions for each observed sample point.

\[
h(X(k + 1), V_{k+1}^d) = 
\begin{bmatrix}
g(x_{i1}^p, y_{i1}^p, v_{d1}^d) \\
g(x_{i2}^p, y_{i2}^p, v_{d2}^d) \\
\vdots \\
g(x_{im}^p, y_{im}^p, v_{dm}^d)
\end{bmatrix}
\]  
(13)

where, \([x_{ij}^p, y_{ij}^p]^T(j = 1 \ldots m)\) is the Cartesian coordinate of the \(i, j\)th sample point in robot-centric Cartesian space.

\(V_{k+1}^d\) is the measurement noise, which is assumed to be zero mean uncorrelated Gaussian distributed, with covariance matrix \(R_{k+1}^d\):

\[
R_{k+1}^d = 
\begin{bmatrix}
r_{11} & 0 & \cdots & 0 \\
0 & r_{22} & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & r_{mm}
\end{bmatrix}
\]  
(14)

\(r_{jj} = R_0/n_j\) \((j = 1, 2, \ldots m)\)  
(15)

Here, \(R_0\) is the basic covariance, whose value is determined by the sensor used, \(n_j\) is the number of measurement points in the neighborhood of \([x_{ij}^p, y_{ij}^p]^T\).

\[
\sum_{j=1}^n n_j = N
\]  
(16)

\(N\) is the total number of the measurement points.

We take \(h(X(k + 1), V_{k+1}^d)\) as the measurement model, and the measurement at time instant \(k + 1\) is denoted as:

\[
z_{d}(k + 1) = h(X(k + 1), V_{k+1}^d)
\]  
(17)

Here, \(z_{d}(k + 1)\) is assumed to be zero in the procedure of measurement update.

When we get the new measurement model, we can use it to update the state in the standard EKF. A single step of the EKF is described as below:

A. Motion Prediction

This step projects the state at time instant \(k\) to the state at time instant \(k + 1\).

\[
\hat{X}(k + 1|k) = f(\hat{X}(k|k), u(k), 0)
\]  
(18)

\[
P(k + 1|k) = FP(k|k)F^T + GQ_kG^T
\]  
(19)

where,

\[
F = \frac{\partial f}{\partial X}\bigg|_{X(k|k)} \quad \text{and} \quad G = \frac{\partial f}{\partial u}\bigg|_{u(k)}
\]

are the Jacobian matrices of the motion model with respect to the state \(X(k)\) and the input \(u\).

B. Measurement Update

This step updates the state \(X(k + 1)\) using measurement \(z_{d}(k + 1)\), rather than \(z(k + 1)\).

\[
\hat{X}(k + 1|k + 1) = \hat{X}(k + 1|k) + W(k + 1)\nu(k + 1)
\]  
(20)

\[
P(k + 1|k + 1) = P(k + 1|k) - W(k + 1)S(k + 1)W^T(k + 1)
\]  
(21)

where,

\[
\nu(k + 1) = z_{d}(k + 1) - h(\hat{X}(k + 1|k), 0)
\]  
(22)

\[
W(k + 1) = P(k + 1|k)H^T S^{-1}(k + 1)
\]  
(23)

\[
S(k + 1) = HP(k + 1|k)H^T + R_{k+1}
\]  
(24)
Here,
\[ H = \frac{\partial h}{\partial X} \hat{X}(k+1|k) \]
is the Jacobian matrix of the observation model with respect to the state \( X(k+1) \).

V. STATE ELIMINATION BASED ON EQUATION
CONSTRAINED KALMAN FILTER

A significant difference between SLASEM and conventional EKF-SLAM is that states in SLASEM are internal geometrically constrained, while no constraint exists in EKF-SLAM.

If the coordinate of a sample point could satisfy the polynomial that describes the contour, we say that the sample point satisfies the geometrical constraint of the contour. When removing redundant sample points, we should consider the geometrical constraint to make sure that the contour formed by the rest points has little changes compared with the previous contour. If the geometrical constraint is not considered before removing redundant sample points, there will be great differences between the shapes represented by the remaining points and the shapes before removing the redundant points. This means that the accuracy of the SEM is decreased.

![Fig. 2. Black pentagrams are the points describing a contour. Green cycles are the points updated from the black pentagrams after performing ECKF. Thin red solid line is fitted by using \{S1, S2, S3, S4, S5\}. Thick blue dashed line is fitted by using \{S2, S3, S4\}.](image)

We give an example here. Suppose sample points \{S1, S2, S3, S4, S5\} lie on a same wall as shown in Fig. 2. The polynomial that describes the contour is denoted by thin red solid line, which is fitted by using all the 5 sample points.

As shown in Fig. 2, if we eliminate points S1 and S5, the thick blue dashed line that is fitted by using points \{S2, S3, S4\} will have great changes compared to the curve fitted by using \{S1, S2, S3, S4, S5\}. It illustrates that if we have not consider the geometrical constraint when eliminating sample points, the curve fitted by the rest of the sample points would cause a wrong result of the contour. In this case, the rest of the sample points are not able to describe the contour effectively.

However, redundant sample points are desired to be eliminated to reduce the computational cost. And what we expect is that no matter which sample point is removed, the shape described by the rest sample points should not change too much.

In this section, we propose a method to eliminate sample points considering geometrical constraint. Before eliminating sample points, we firstly translate the constrained sample points set denoted by its mean \( \bar{X}(k+1|k) \) and covariance \( P(k+1|k+1) \) into free sample points set denoted by its mean \( \bar{X}^f(k+1|k+1) \) and covariance \( P^f(k+1|k+1) \) through equation constrained Kalman filter (ECKF) [22]. After that, we can eliminate the redundant sample points by removing their counterparts in \( \bar{X}^f(k+1|k+1) \) and \( P^f(k+1|k+1) \).

Suppose that there are too many sample points on a small segment of a contour, and we want to remove some of them to make the map more compact. First, we fit an IP \( p(x, y) = 0 \) using the sample points on a small segment. Then we linearize the IP at each sample point to get the linear form of the geometrical constraints as:

\[ DX(k+1) = d \] (25)

Note that fitting an IP here do not use the sample points on the whole contour, but the ones on a small segment. The reason is that the contour may hold a very complex shape, which cannot be fitted well with a low order IP. However, a small segment of the contour can be fitted well enough with a low order IP.

Using the following steps, the state with geometrical constraints is transformed into state without geometrical constraints.

\[ \hat{d}_k = D \bar{X}(k|k) \] (26)
\[ P^{dd}(k|k) = DP(k|k)D^T \] (27)
\[ P^{rd}(k|k) = P(k|k)D^T \] (28)
\[ K^P = P^{rd}(k|k)(P^{dd}(k|k))^{-1} \] (29)
\[ \bar{X}^P(k|k) = \bar{X}(k|k) + K^P_k(d - \bar{d}) \] (30)
\[ P^P(k|k) = P(k|k) - K_k^P P^{dd}(k|k) K_k^{PT} \] (31)

After the constrained state \( \bar{X}(k|k) \) and its covariance \( P(k|k) \) are transformed to unconstrained state \( \bar{X}^P(k|k) \) and its covariance \( P^P(k|k) \), we can eliminate the redundant sample points by removing their counterparts in \( \bar{X}^P(k|k) \) and \( P^P(k|k) \). Finally, we will use \( \bar{X}^P(k|k) \) and \( P^P(k|k) \) instead of \( \bar{X}(k|k) \) and \( P(k|k) \) for the next Kalman filter’s iteration.

VI. SIMULATION AND EXPERIMENTAL RESULTS

To validate the efficiency of the approach proposed in this paper, we have carried out a simulation and an indoor experiment.

A. Simulation Result

In the simulation, the mobile robot moves along the dark curve (nearly overlapped with the blue curve), while keeping observing the obstacle, as shown in Fig. 3 (a). The control noises are assumed to be Gaussian distributed ones with zero mean and covariance \( \text{diag}(v + 30\%)^2, (3^\circ)^2) \), where \( v \) denotes the current speed of the vehicle. The measurement
Fig. 3. Simulation Results. (a) The simulated environment and the map built by the proposed algorithm. Grey closed area denotes the obstacles. Red stars are the SEM built by our algorithm. Black curve denotes the estimated trajectory. The blue curve and the black one are nearly overlapped with each other. Green dashed line shows the odometry data. (b) The eight curves of the mean distance from the samples to their corresponding contour.

In this paper, we presented a SLAM algorithm, namely simultaneous localization and sampled environment mapping, which is applied in complex environment. This algorithm holds the same structure as conventional EKF-SLAM, except that it uses a SEM instead of a landmark map. In order to update the SEM, we have developed a new observation model by calculating the signed distance from the predicted sample points to the curve that is fitted with the measurements. To remove the redundant sample points, we have applied ECKF to translate the states with geometrical constraints into states without the constraints, then remove the redundant states as they are unconstrained. Finally, we have validated the efficiency of the proposed algorithm by a simulation and an indoor experiment.
Currently, the proposed algorithm cannot prohibit the inconsistency as shown in Fig. 5. In our future studies, we will apply corner constraint to the algorithm to improve its performance. We will also consider using the least samples to represent the map. The fewer the number of samples, the faster the algorithm will carry on.

REFERENCES

[1] H. Durrant-Whyte and T. Bailey, "Simultaneous Localization and Mapping: Part I," IEEE Robot. Autom. Mag., vol.13, no.2, pp.99-110, 2006.
[2] T. Bailey and H. Durrant-Whyte, "Simultaneous Localization and Mapping (SLAM): Part II," IEEE Robot. Autom. Mag., vol.13, no.3, pp.108-117, 2006.
[3] S. Thrun et al., "Simultaneous Localization and Mapping with Sparse Extended Information Filters," Int. J. Robotics Research, vol.23, no.7-8, pp.693, 2004.
[4] P. M. Newman, "On the Structure and Solution of the Simultaneous Localization and Mapping Problem," Ph.D. dissertation, Australian Centre for Field Robotics, University of Sydney, 1999
[5] K. Murphy, "Bayesian Map Learning in Dynamic Environments," Neural Info. Proc. Systems, vol.12, pp.1015-1021, 1999
[6] J. Nieto, J. Guivant, and E. Nebot, "DenseSLAM: Simultaneous Localization and Dense Mapping," Int. J. Robotics Research, vol.25, no.8, pp.711-744, 2006.
[7] J. A. Castellanos et al., "The SMap: A Probabilistic Framework for Simultaneous Localization and Map Building," IEEE Trans. Robotics, Automation, vol.15, no.5, pp.948-952, 1999.
[8] M. Dissanayake et al., "A Solution to the Simultaneous Localization and Map Building (SLAM) Problem," IEEE Trans. Robotics, Automation, vol.17, no.3, pp.229-241, 2001.
[9] M. Montemerlo et al., "FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem," in Proc. National Conf. Artificial Intelligence, 2002, pp.593-598.
[10] M. Montemerlo et al., "FastSLAM 2.0: An Improved Particle Filtering Algorithm for Simultaneous Localization and Mapping that Provably Converges," Int. Joint Conf. Artificial Intelligence, 2003, pp.1151-1165.
[11] Guivant and E. Nebot, "Optimization of the Simultaneous Localization and Map-Building Algorithm for Real-Time Implementation," IEEE Trans. Robotics, Automation, vol.17, no.3, pp.242-257, 2001.
[12] A. Martinelli et al., "A Relative Map Approach to SLAM Based on Shift and Rotation Invariants," Robotics and Autonomous Systems, vol.55, no.1, pp.50-61, 2007.
[13] D. Haehnel et al., "A Highly Efficient FastSLAM Algorithm for Generating Cyclic Maps of Large-Scale Environments from Raw Laser Range Measurements," in Proc. IEEE/RSJ Int. Conf. Intelligent Robots, Systems, 2003, pp.27-31.
[14] G. Grisetti, C. Stachniss, and W. Burgard, "Improved Techniques for Grid Mapping with Rao-Blackwellized Particle Filters," IEEE Trans. Robotics, vol.23, no.1, pp.34-46, 2007.
[15] A. Elazar and R. Patt, "DP-SLAM: Fast, Robust Simultaneous Localization And Mapping Without Predetermined Landmarks," in Proc. Int. Conf. Artificial Intelligence, 2003, pp.1135-1142.
[16] J. Nieto, T. Bailey, and E. Nebot, "Scan-SLAM: Combining EKF-SLAM and Scan Correlation," Int. Conf. Field and Service Robotics, 2005, pp.129-140.
[17] L. M. Paz, J. D. Tardos, and J. Neira, "Divide and Conquer: EKF SLAM in O(n)," IEEE Trans. Robotics, vol.24, no.5, pp.1107-1120, 2008.
[18] J. Leal, "Stochastic Environment Representation," Ph.D. dissertation, Australian Centre for Field Robotics, University of Sydney, 2003.
[19] W. Wan and J. A. Ventura, "Segmentation of Planar Curves into Straight-Line Segments and Elliptical Arcs," Graphical Models and Image Processing, vol.59, no.6, pp.484-494, 1997.
[20] M. M. Blane et al., "The 3L Algorithm for Fitting Implicit Polynomial Curves and Surfaces to Data," IEEE Trans. Pattern Analysis, Machine Intelligence, vol.22, pp.298-313, 2000.
[21] T. Sahin and M. Unel, "Globally Stabilized 3L Curve Fitting," Lecture Notes in Computer Science, vol.495, pp.502, 2004.
[22] B.O.S. Teixeira et al., "State Estimation for Equality-Constrained Linear Systems," in Proc. IEEE Conf. Decision, Control, 2007, pp.6220-6225.
[23] J. S. Julier and K. Uhlmann, "A Counter Example to the Theory of Simultaneous Localization and Map Building," in Proc. IEEE Int. Conf. Robotics, Automation, 2001, pp.4238-4243.
[24] http://www.mathworks.com/matlabcentral/fileexchange/12627