TERRAIN AIDED UNDERWATER NAVIGATION – A DEEPER INSIGHT INTO GENERIC MONTE CARLO LOCALIZATION

Alexander Bachmann, Stefan B. Williams
ARC Centre of Excellence for Autonomous Systems
School of Aerospace, Mechanical and Mechatronic Engineering
University of Sydney, Sydney NSW
Australia
Email: a.bachmann@acfr.usyd.edu.au

Abstract

This paper proposes a method for terrain-aided navigation in underwater environments based on the simple means of a digitized seafloor map and sonar measurements. Since the depth information is highly non-linear and non-Gaussian, a family of probabilistic algorithms known as Monte Carlo Localization (MCL) is used for the estimation process. To increase the performance of MCL under limited computational power the sample size of the classical particle filter is adapted on-the-fly. Within this adaptation process the number of samples is increased if the state uncertainty is high and decreased if the density distribution is focused on a small area. This approach is based on a method known as KLD-sampling. As the performance of the filter- and therefore the accuracy of the position estimation- can be limited by the noise levels of the system, a method is proposed to adapt the noise level over time. This adaptation is made with reference to the terrain excitation in the area of the observation. The performance of this method is presented in terms of the entropy propagation of the particle distribution.

I. Introduction

In underwater navigation the imperfections of traditional navigation systems, based on inertial navigation and velocity measurements, begin to severely impact the navigation results over time. This paper investigates the possibility of supporting traditional navigation systems with position estimates based on information from the terrain beneath the vehicle using recursive Monte Carlo Methods.

A particular, recursive implementation of Monte Carlo based statistical signal processing is known as particle filter. The key idea of this technique is to represent probability densities by sets of discrete samples. This makes the filter very robust and allows it to represent a wide range of density distributions. With the unstructured and very complex nature of a seabed, a particle filter is an ideal technique for incorporating altitude information into the tracking process of underwater vehicles. The ability to use natural features will allow a submersible body to be deployed in a large range of environments without the need to introduce artificial beacons - a very significant innovation in this area of work.

This remainder of this paper is organized as follows. Section II outlines the basics of particle filters and their extensions. Furthermore a technique to reduce the complexity of the estimation process, called Rao-Blackwellization, is introduced. Section III presents the results of the simulations of the deployed underwater vehicle in different environments. Conclusions are drawn and future work is suggested in Section IV.

II. Particle Filters for Bayesian Estimation

When implementing Bayesian filters for localizing mobile robots, one needs to know three distributions: the initial belief $\text{Bel}(x_0)$, the next state probability $p(x_t \mid x_{t-1}, u_{t-1})$ [also called posterior density] and the perceptual likelihood $p(y_t \mid x_t)$. The main idea in Bayesian estimation is to compute the posterior density x_t based on the observation y_t made at time t. The sequential Monte Carlo methods simulate this distribution with a finite number of samples, or "particles". The more particles, the better the approximation gets. Because of the concept of particles, these filters are often referred to as particle filters. In the underwater vehicle tracking case we have the following non-linear time discrete system

$$x_{t+1} = f(x_t) + v_t \qquad (1)$$
$$y_t = h(x_t) + e_t \qquad (2)$$

, where x_t is the state of the system, for instance the position, the heading and the course, y_t are measurements, v_t and e_t are process noise and measurement noise respectively. The particle filter approximates $p(x_t | y_t)$ with a set of N random samples $\{x_t^i\}_1^N$, where each particle x_i is assigned a weight w_t

$$S_t = \left\{ \langle x_t(i), w_t(i) \rangle \right\} \qquad (3)$$

The weight of each particle should in some way reflect the probability that the properties of this particle are the correct ones. Those particles with the highest weights are propagated in time and natural selection is performed. It should be noted that the number of particles, N, has to be chosen large enough to accurately represent the underlying probability density function.

A. Classic Particle Filter

The basic form of the particle filter updates the belief of a state x according to a recursive sampling sequence that incorporates observations like depth measurements into the estimation:

1. *initializing the filter: a sample set of N samples/ particles is generated based on the initial belief.*
2. *update the likelihood/ importance weights by incorporating the observations*
3. *resample a new sample set on the basis of the importance weights. The resampling step is necessary in order to avoid having the importance weights becoming to degenerated.*
4. *predict the new state of the system with the system model and the controls.*
5. *iterate to item 2.*

The resampling step is often implemented using Sampling Importance Resampling (SIR) [8]. The SIR algorithm resamples the particles at each iteration cycle. This could be quite computationally expensive and the dependency between the samples becomes unnecessarily high. Therefore an algorithm was implemented which updates the weights recursively and only resamples when the weights are too degenerated. While running latter algorithm, it was found out that the efficiency of the resampling process was highly restricted by the process noise (in the prediction phase of the filter process) when reaching areas of increasing terrain excitation. The broad distribution of the samples at each iteration cycle limited the convergence and therefore the

efficiency of the filter drastically. In Section B.ii a method is presented to work against this effect by adjusting the process noise to lower levels and therefore forcing the system to a less distributed sample set.

Another big issue of making particle filters efficient is the number of deployed samples. Classical filters use a fixed sample number during the entire iteration process – independent of the confidence level of the estimation process. The desire to create a highly efficient particle filter led to a new approach of introducing a generic particle filter that changes the number of samples over time based on the confidence level of the current estimation.

B. Generic Particle Filter

Obviously, the number of samples has a significant influence on the computational burden of the system. While a higher number of samples might be needed to represent the belief during early stages of the localization and tracking process, a smaller number of samples should be sufficient once the estimate has converged to a compact distribution about the true robot position. The approach used here was first introduced by [5] and is called KLD-sampling.

i. KLD-Sampling

The key idea of this method is to determine the number of samples at each iteration cycle of the particle filter such that the error ε between the true density distribution and the approximated sample-based representation of the filter is less than a certain bound. To derive this bound, we assume that the true distribution is given by a discrete, multinomial distribution – depicted by a number of bins k. Since the error between the filter-estimate and the true distribution is measured by the Kullback-Leibler distance, the approach is denoted as KLD-sampling algorithm. The term "distance" here should not be misunderstood as a metric measure. The KL-distance describes the distance of two probability densities as a kind of matching ratio of the distributions.

Using the so called Wilson-Hilferty transformation [5], a relation connecting the likelihood ratio to the KL-distance is presented. This key relation allows a minimum number of particles required to limit the error bound ε to be found. For fixed ε, Equation (4) computes the required number of samples N as a function of k.

$$N_{threshold} = \frac{k-1}{2\varepsilon} \left\{ 1 - \frac{2}{9(k-1)} + \sqrt{\frac{2}{9(k-1)}} z_{1-\delta} \right\}^3 \qquad (4)$$

K corresponds to the number of bins with support. Here, a bin has support if the probability of the multinomial, true distribution is above a certain

threshold. $z_{1-\delta}$ is the quantile of the standard normal distribution.

Equation (4) shows that the required number of samples is inversely proportional to the error bound ε and proportional to the number of bins k. In the KLD-sampling process the bins k are estimated by the number of grid cells that are at least occupied by one particle.

To approximate the multinomial distribution, a fixed, three-dimensional grid is used. During the prediction step of the particle filter the algorithm checks if the newly generated particle falls into an empty cell of the grid or not. If the grid cell is empty, the number of bins k is incremented and the cell is marked as not-empty. By doing so, after each sample the number of required samples $N_{threshold}$ is updated using Equation (4). This process is repeated till no more empty grid cells are filled and k stops increasing. Consequently $N_{threshold}$ stabilizes. The grid is reset after every filter update.

In the case of the terrain-aided localization process this approach selects small sample numbers for regions with unique terrain signature where the samples focus on a small area in state space and large sample numbers for regions of low or symmetric terrain information where the samples start to spread out.

ii. Adaptive Noise Level

To further increase the performance of the estimation process, another approach, addressing the relation between the process noise, the estimation quality and the level of terrain excitation, is presented.

The performance of the localization process is highly dependant on the noise levels applied to the system. High process noise spreads the sample set within the resampling phase over a wide area. Though it makes the filter robust, it has severe impact on the estimation accuracy. An approach is presented that uses the importance weight formation of the estimation process and the existing terrain composition to adapt the noise levels.

The idea is to keep track of the non-normalized importance weights in every iteration cycle. As the importance weights represent the matching rate of sample set and respective observation, the sum of the single weights provide a meaningful measure [8].

$$w_{sum,t} = \sum_{i=1}^{N} w_t^i \qquad (5)$$

If the sample set is well supported by the sensor readings, $w_{sum,t}$ is large. Is the matching rate low, the filter degenerates and the number will be small.

If the vehicle is deployed to uneven terrain the sensors can extract a high level of information for the estimation process.

A measure of the information content of the environment in a particular region is given by the gradient field of the terrain map f

$$\nabla h(f \mid x_t) = \left[\frac{\partial f}{\partial x} + \frac{\partial f}{\partial y} + \frac{\partial f}{\partial z} \right]_{f(x_t)} \qquad (6)$$

∇h represents the gradient at the position x_t of the actual observation. The bigger the gradient field in the area around the vehicle, the better the localization filter should perform.

The estimator uncertainty is supposed to be high in regions of flat terrain. As a result of the insignificant elevation data, the importance weights are uniformly distributed with low values. For higher excitation levels the filter algorithm should perform increasingly well as the observations contain more information about the actual position. Therefore the importance weights should also increase noticeably. If this is not the case it can be assumed that the process noise is spreading the particles too wide. As this limits the gain of accuracy within every iteration cycle - and valuable computational resources are lost - an attempt has been made to reduce this effect. The approach introduced here calculates the importance weight sum $w_{sum,t}$ for each iteration cycle. At the same time the terrain excitation is checked for the proposed position of the vehicle. If the excitation level is rising above a certain threshold, while $w_{sum,t}$ is not increasing distinctly, the noise levels are reduced constantly with each iteration cycle. As a result the newly predicted sample set gets distributed over a smaller area and the particles can concentrate effectively around the observation-based positions. The noise levels are limited to a lower value to avoid the sample set to get too concentrated. For regions with a low gradient field the uncertainty of the localization process grows and the process noise is adapted accordingly to higher values.

C. Rao-Blackwellization

In this section, the prediction step of the filter process itself is considered. The particle filter solves the non-linear filtering problem in a numerical way.

The complexity of the estimation process increases with the number of states to be estimated. Therefore, with many states the number of particles needs to be very high in order to ensure that all modes of the system are excited so that the filter performs well. An approach known as Rao-Blackwellization estimates linear portions of the problem analytically with a Kalman filter [10].

In the underwater application of the submarine one candidate state to be estimated using Rao-Blackwellization would be the heading. By using the measurements of an integrated compass and a

gyroscope, the heading can be estimated analytically.

The resulting lower complexity of the system should make the particle filter more effective. Rao-Blackwellization reduces the variance of the estimate error as well as the number of particles resulting in a reduction of the computational burden.

III. Simulations

In the simulations the vehicle was assumed to be equipped with an altimeter to scan the terrain below the vehicle and a depth sensor. The vehicle was supposed to travel in a constant depth. Terrain information was provided by a-priori seafloor maps. In addition to simulated terrain maps, the algorithms have also been verified using a simulated vehicle travelling over a real terrain-elevation map representing the newly forming Hawaiian island of Loihi. Figure *1* shows one of the simulated terrain maps. This map provided a good source of information as the symmetrys in the centre region around the hump have severe impact to the estimation process.
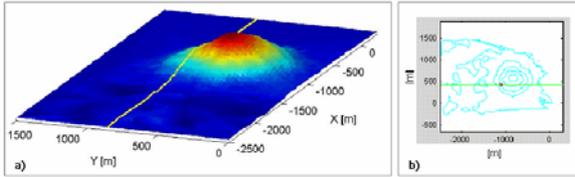

Figure 1*: Terrain the sub was deployed during simulations; a) Seafloor map with symmetrical hump, the yellow line shows the path of the sub; b) Contour plot of terrain, the symmetries can be seen in form of rings (same heights of hump), here the sub path is a green line*

In order to recursively estimate the location of the vehicle, an initial density distribution was needed. Therefore, the algorithm had to be provided with an approximate initial state $x0$, along with a measure of the accuracy assessment of that initial estimate. The initial state used in this simulation was a deliberate starting position of the vehicle in Cartesian coordinates with a Gaussian initial uncertainty distribution. The vehicle was expected to receive measurements of velocity, depth, orientation and altitude every 20ms. *Table 1* lists the simulation parameters in detail.

Item	Description	Value
N	Number of particles	900
v_{sub}	Vehicle speed	5 m/s
t	Deployment time	150 sec
t_s	Scan cycle	0.2 sec
σ_{init}	Gaussian Initial distribution	20m
σ_{noise}	Initial noise level	20m

Table 1: simulation parameters

Since the distributions are fairly uni-modal in this application, the probability distributions in the simulation runs were determined by taking the mean and standard deviation of the particle cloud. The mean of the system states was interpreted as the estimated position and heading of the vehicle. As a measure of the uncertainty of the estimation the standard deviation of the sample set was used.

The different approaches were evaluated by providing the vehicle with the observations of a predefined path within the terrain maps. For each iteration cycle the deviation of predefined path to estimated mean was computed. In this work it is referred to as error of the filter. The vehicle was supposed to travel from north to south in a straight trajectory. Figure *2* shows a model of the vehicle equipped with a sonar that is scanning the seabed perpendicular to its horizontal.
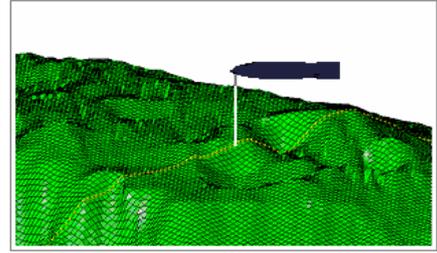

Figure 2: Terrain-aided navigation using information available from on-board sensors and an a-priori map; The sonar beam scans the terrain continuously

A. Running the generic particle filter

To have a measure of the performance of the generic particle filter a reference density distribution was used. The sample sets generated by the generic particle filter were compared with the respective precompiled reference distributions by measuring the KL-distance of the two distributions. The reference distributions were allocated by using a particle filter with a fixed number of particles and Gaussian distribution. The distributions produced by this particle filter were available for every position of the predefined vehicle path.

As the resulting density distributions of the filter are discrete a kernel density estimation method proposed in [4], [10] was used to obtain the underlying probability density. The Epanechnikov-/ and Gaussian kernel were determined to provide the best results during the simulations. After each iteration cycle the KL-distance between the estimated and the precompiled sample set was calculated with this kernel density estimator.

For the results presented in this paper, the vehicle was deployed at $x = -2000m$. Throughout the simulation runs a constant threshold ε (KL-distance) of 0,25 was used. The bin size of the three-dimensional grid was fixed to $0,5m \times 0.5m \times 0.5m$.

4

Figure *3* shows that the effective number of particles slowly approaches the deployed number of particles. About 15 seconds after the initialization of the filter the two numbers are converging noticeably. The execution time per iteration cycle is decreasing proportionally to the number of samples used.
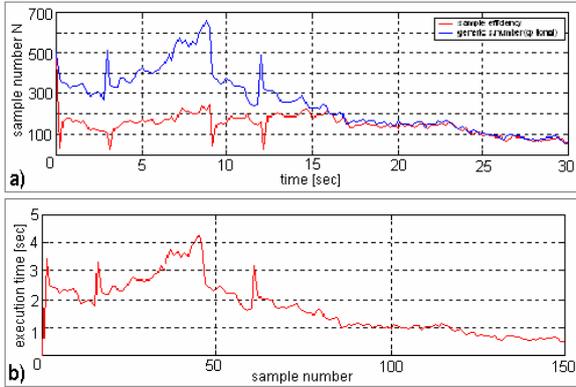

a)

b)

Figure 3*: a) The effective (red) and the adapted (blue) sample size for a typical filter run over 150m. The filter algorithm first increases the particle number N as result of the symmetries in the map before entering unique terrain and decreasing N constantly; b) Execution times of the single iteration cycles*

Figure *4* shows the error of the mean state estimate and the true location together with the 2-σ uncertainty bounds of the particle distribution. The filter seems to perform reasonably well as the estimation error never exceeds the uncertainty bounds. It can be noted that the uncertainty bounds drop significantly after receiving unique terrain information at about 8 seconds.
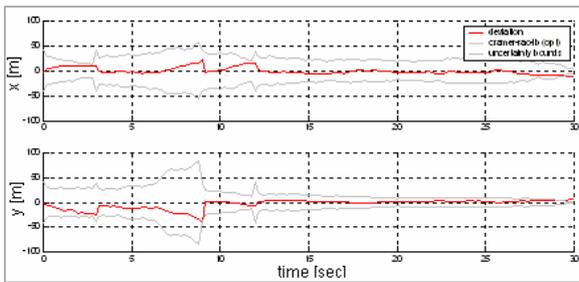

Figure 4: Error (red) between the mean estimate and the predefined, given path together with the 2-σ bounds (grey) of a typical filter run: after first increasing variances the values drop significant when reaching unique terrain after about 8 seconds

The increasing number of particles after about 25 cycles or 5 seconds after initializing the filter can be explained by having a closer look to the Kullback-Leibler distance in Figure *5*. As the terrain map shows a highly symmetrical appearance in the area close to the initialization of the vehicle, the filter starts to run "off" the expected track. The KL-distance increases drastically without being noticed by the filter. While located in symmetrical terrain and therefore higher localization uncertainty, the filter

algorithm continuously adds particles to the system. At sample number 45 the vehicle reaches terrain with unique signature and the filter is able to recover. The observations in this area lead to a decreasing number of particles with an increasing efficiency of the estimation process.
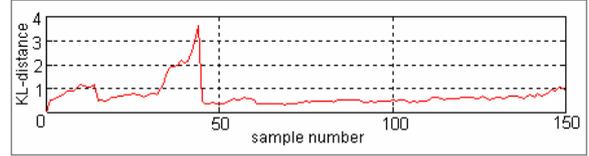

Figure 5: Kullback-Leibler distance of a filter run over 150m; after 35 iteration cycles the filter looses track due to symmetries in the terrain map, at around cycle number 45 the terrain map reveals significant character and the filter recovers abruptly.

B. Adaptive Noise Level

The process noise of the system was initialized with 20m respectively. To avoid the sample set to get to concentrated a limitation of the process noise levels to a minimum of 10m was defined. At this value the filter was still robust and did not run the risk of degeneration.

During the simulations the process noise levels were adapted continuously on the basis of the gradient field of the vehicle position and the importance weight propagation. The gradient field for the single vehicle positions was determined by calculating the values along the predefined vehicle path. In Figure *6* the normalized seafloor excitation along the expected vehicle path is shown. After the filter process was started, the sum of the importance weights w_{sum} was calculated for each iteration cycle. The terrain excitation at position x_t was obtained from a terrain excitation map based on the used terrain map.
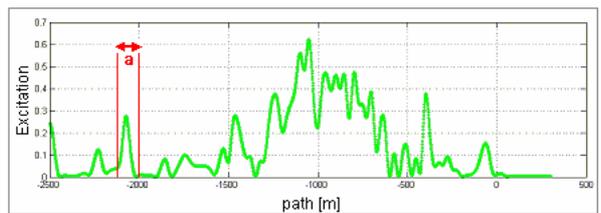

Figure 6: Normalized excitation level along the predefined path the vehicle should follow in the simulation run; the region, the sub was deployed to is marked with a

If the normalized terrain excitation level exceeded 0.1 and $w_{sum,t}$ did not increase its value noticeable to $w_{sum,t-1}$, it was assumed that the particle distribution was too broad and as a result the matching rate of observation and estimation – represented by the importance weights – did not rise. Consequently the process noise was decreased by 5% of its current value. By doing so, the filter propagates a less

broadly distributed sample set. If the terrain in the area in which the vehicle was located had a high information level, the filter should converge faster around the distinctive positions of the observation.

The entropy propagation of the estimated density distribution was used as a measure for the convergence of the filter. To determine the entropy the kernel density estimation methods of Section III.A were used.

Decreasing entropy signifies higher certainty of the system state and therefore better information about the position of the vehicle. The vehicle was initialized in flat terrain with a low excitation. Figure 7-a) shows that the accuracy of the postition estimation first suffers from a lack of terrain information. An increasing particle distribution, displayed by the 2-σ uncertainty bounds, can be noted. As soon as some unique terrain features are present, such as hills and rock edges, the probability distribution converge to a more compact estimate. Consequently, the sum of the importance weights should increase. As can be seen in Figure 7a), the large spread in particles and the high process noise appear to keep the resampling weights low in the area of high terrain information. Therefore the mean state estimate does not converge to the true value.

In Figure 7-b) the process noise is adapted to lower values when the vehicle is going over higher excited terrain. After continuously decreasing the noise levels the sample set is restricted to a smaller area. Therefore more samples are concentrated around the area of the actual observation and the mean state estimation is more accurate.
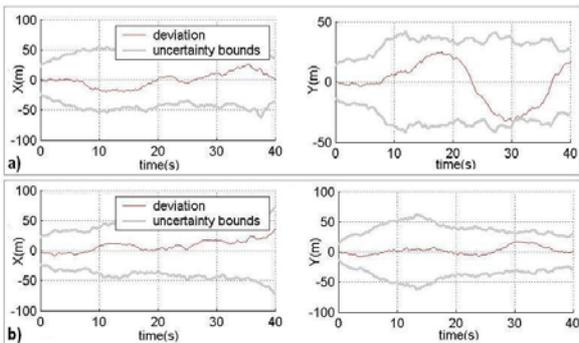

Figure 7: a) Error (red) between the mean estimate and the predefined, given path together with the 2-σ bounds (grey) of a typical filter run; b) Error and 2-σ bounds using the filter with adaptive noise

In Figure 8 three consecutive density distributions of a filter run over 150 meters are presented. The plain represents the x/y-area, the vehicle is deployed to. In the vertical axis, the continuous density estimate of the current particle distribution is drawn. The first image on the left shows the density distribution of the initial postion with a variance of 20m, respectively for each direction. The entropy of the system is set to 1, representing the uncertainty of the estimation at the time of the initialization. In a typical simulation run with the same initial settings as in Section III.A, the variances first increase due to the terrain symmetry. But then they decrease constantly until the end. At the end of the process the entropy has deceased by about 20% with respect to the initial state.
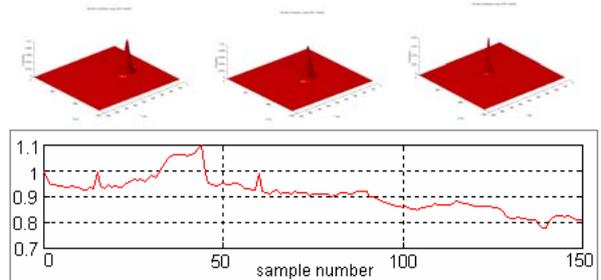

Figure 8: Top: The propagation of the density distribution shown as a two-dimensional plot using the Gaussian kernel density estimation method; Bottom: Entropy propagation of a filter run over 150 m

IV. Conclusion

In this paper an adaptive particle filter approach was introduced. The efficiency of the different filter concepts was investigated by doing simulations. Improvements in terms of computational burden compared to a commonly used "static" particle filter were presented and discussed. As the sample number was adapted on-the-fly, the performance of the estimation process increased drastically to about 90% of effectively deployed samples. The proposed adaptation of the noise level also increased the accuracy of the estimation process when limited by the process noise.

A more detailed investigation of the Cramer-Rao lower bound is planned. This bound is referred to in several publications as a measure of the uncertainty limitation in navigation applications based on terrain maps. The use of an extended sonar beam that scans the terrain in two dimensions promises a higher level of extractable information from the map. Based on this, one of the next steps will be the implementation of an active filter that uses the terrain information to minimize the estimation uncertainty. This can be achieved by guiding the vehicle to terrain with high excitation.

The restricted availability of digitized underwater terrain maps makes it difficult to verify the results of the simulations with experimental measurements in a real environment. After receiving a set of digitized terrain maps of the Sydney harbour area it is now planned to implement the algorithms during deployment of our Unmanned Underwater Vehicle.

Acknowledgement

References

[1]. S.B.Williams, G.Dissanayake and H.F.Durrant-Whyte; "Towards terrain-aided navigation for underwater robotics"; 15(5)533 550, 2001

[2]. S.B.Williams; "A Terrain Aided Tracking Algorithm for Marine Systems"; International Conference on Field and Service Robotics 2003

[3]. R.Karlsson, F.Gustafsson and T.Karlsson; "Particle filtering and Cramer-Rao lower bound for underwater navigation"; Linkoeping,Sweden LiTH-ISY-R-247 4 2002

[4]. B.W.Siverman; "Density Estimation for Statistics and Data Analysis"; ISBN 0 412 246201; Chapman & Hall

[5]. D. Fox ; "KLD-sampling: Adaptive Particle Filters"; Advanced in Neural Information Processing Systems 14, NIPS-2001, MIT press

[6]. F.Gustafsson, J.Jansson, U.Forssell and P.-J. Nordlund; "Particle Filters for Positioning, Navigation and Tracking"; Linkoeping,Sweden LiTH-ISY-R-2333 2001, IEEE Trans. Of Signal Processing

[7]. S.Thrun, D.Fox, W.Burgard and F.Dellaert; "Robust Monte Carlo localization for mobile robots"; Intelligence-2000

[8]. N.Bergman ; "On the Cramer-Rao bound for Terrain-Aided Navigation"; Linkoeping,Sweden LiTH-ISY-R-1970

[9]. S.Arulampalam, S.Maskell and N.Gordon ; "A Tutorial on Particle Filters for on-line non-linear/non-Gaussian Beyesian tracking"; DSTO 2001, IEEE2001

[10]. H.D. Whyte; "Notes on Monte-Carlo and Kernel Methods"; Australian Centre for Field Robotics; 3-2003

[11]. P.J. Nordlund; "Recursive state estimation of nonlinear systems with applications to integrated navigation"; R-No: LiTH-ISY-R-2321; Department of Engineering University Linkoeping, Sweden