Long-Term Accuracy in Sea Navigation without using GNSS Systems

Mårten Lager*, Elin A. Topp, Jacek Malec†

May 8, 2017

Abstract

Many ships today rely on Global Navigation Satellite System (GNSS), for their navigation, where GPS (Global Positioning System) is the most well known. Unfortunately, the GNSS systems make the ships dependent on external systems, which can be malfunctioning, be jammed or be spoofed.

There are today some proposed techniques where, e.g. bottom depth measurements are compared with known maps using Bayesian calculations, which results in a position estimation. Both maps and navigational sensor equipment are used in these techniques , most often relying on high accuracy maps, with the accuracy of the navigational sensors being less important.

Instead of relying on high accuracy maps and low accuracy navigation sensors, this paper presents an idea of the opposite, namely using low accuracy maps, but compensating this by using high accuracy navigational sensors and fusing data from both bottom depth measurements and magnetic field measurements.

1 Introduction

Finding the way over great seas has been important for thousands of years. The compass was invented almost a thousand years ago, and the first nautical sea charts were used in Italy in the 13th century. With a compass and a nautical sea chart, it is possible to perform *dead reckoning* to estimate the current location based on the previous location and the compass direction. In this technique, it is also possible to compensate for the vessel's drift and sea current. But if one is not able to estimate the drift and sea current accurately, the position error starts increasing, as each estimation of the position is relative to the previous one, which means that the position error is accumulated over time. This deficiency can be overcome in different ways. By regularly determining the position compared to known landmarks, the accumulation of error is reset. But if landmarks cannot be found because one is on open water, either there is a need for increasing the accuracy of the dead reckoning by using better equipment (e.g. compass, logs (for speed), gyro, accelerometers, inertial sensors), or there is a need to use information about the environment that can be seen out on open waters. During the 18th century the celestial navigation was invented, which uses angle measurements to the sun, moon and stars to greatly improve the long-term accuracy of navigation. Nowadays the celestial navigation has almost completely been abandoned, because GNSS can determine the position accurately and efficiently. The most common and oldest GNSS system is GPS, but there are also other systems, e.g. Galileo and Glonass.

The GNSS systems have made it very simple to determine a vessels position with good accuracy, but there are still some disadvantages. One important disadvantage is that the ship needs to rely on external information from the GNSS satellites which is sent to the GNSS receiver onboard. It is quite simple to jam the radio reception from the GNSS satellites, which results in that it is not possible to determine the position any more. Even worse, it is possible to spoof the GNSS transmission information with advanced equipment, resulting in that an incorrect position is provided [8].

*M. Lager is an industrial Ph.D student with Saab Kockums in Malmö and with the Department of Computer Science, Lund University, Sweden.

†E.A. Topp and J. Malec are with the Department of Computer Science, Lund University, Sweden.

If a dependency on the GNSS system is not desired, and a more modern and a less time consuming technique than the celestial technology from the 18th century is wanted, there might be some alternatives. Some alternatives can be found in the following research papers.

Reference [2] describes how systems on an airplane measure the elevation and compare it to a known digital elevation map with a *Particle Filter* (PF) algorithm. By doing so, the algorithm's estimation of the position eventually converges to the correct position. A similar technique is used in [3] for surface and underwater navigation, where the bottom depth is compared to a high accuracy bottom map with a PF algorithm. The same paper also describes how almost the same PF algorithm can be used to estimate the position by comparing measured distances to the surrounding shore line with a map of the same area. There is other information which can be used by particle filters for positioning. The earth magnetic field surrounds the earth, and is disturbed by ferromagnetic elements. In an indoor environment, these disturbances are normally bigger than the earth magnetic field itself [5], and both [4] and [5] suggest how to estimate a position in an indoor environment with a PF comparing magnetometer measurements to a known magnetic map.

This paper presents an idea of how to perform sensor fusion based on various types of PF calculations in order to estimate a ship's position. By using various types of measurements for the PF calculations, and by relying on high accuracy navigation sensors, the probability of being able to obtain the current position without having to rely on high accuracy maps or GNSS data is high.

This paper is organized as follows: In Section 2, first a brief discussion is given about Bayesian calculations and how these can be used to estimate the position of a ship. Then Particle Filters (PF) are explained more in detail, and it is described how e.g. bottom depth and magnetic fields can be used for the Correction Step of the PF. Based on the current available research, limitations and opportunities of this research is described in Section 3. In Section 4 an idea of how to implement an algorithm that correct the limitations is given. Further, what has been implemented so far and what is left of the implementation is presented. In Section 5, concluding remarks are given.

2 Probabilistic position estimation

2.1 Bayesian position estimation

The key problem we have is that we would like to estimate the position, but are not able to measure the position directly. Instead we can measure other information, such as how the ship is moving, bottom depth and magnetic field data. This can be modeled as a *Hidden Markov Model* (HMM), where the state (i.e. the position, attitude, velocity and acceleration) influences the data which can be measured. Figure 1 illustrates how the state x(t) (i.e. position) influences the data which can be measured, denoted by y(t).

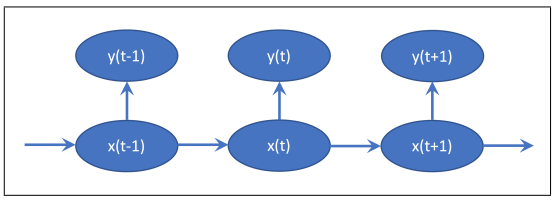

Figure 1: Each state (x(t)) in an HMM influences the data which can be measured (y(t)). The position is not possible to measure directly, but the position will influence which bottom depth and magnetic field vector that is measured. The state in time t (x(t)), contains the state position, attitude, and acceleration. The measured quantities (y(t)) which are influenced by the state, are attitude measurements, acceleration measurements, bottom depth measurements, magnetic field measurements, etc.

To get the best possible estimation of the state, not only the present measurement is to be analyzed, but all previous data. The equations for calculating the probability of being in one state and going to another state given the measurements at time t, are given as follows:

$$p(x_{t+1} \mid \mathbb{Y}_t) = \int_{\mathbb{R}^n} p(x_{t+1} \mid x_t) p(x_t \mid \mathbb{Y}_t) dx_t \quad (1)$$

$$p(x_t \mid \mathbb{Y}_t) = \frac{p(y_t \mid x_t) p(x_t \mid \mathbb{Y}_{t-1}) dx_t}{p(y_t \mid \mathbb{Y}_{t-1})} \quad (2)$$

These equations are not analytically solvable, and therefore a filter will instead be used for estimation of the position. If the measurements and transition functions would have been linear and the measurement and process noise Gaussian, a Kalman Filter would have been the optimal choice to compute the position [6]. In our case the transition functions are non-linear and the measurements have no Gaussian distribution, but instead a highly multi-modal distribution. There are some non-optimal extensions to Kalman Filters to handle the issues with non-linearity and not having a Gaussian distribution [1]. However, PF are more flexible and have a built-in capability to handle multi-modal distributions. Therefore, the PF algorithm will be used in this paper.

2.2 Particle Filters for estimation of the position

A *Particle Filter* (PF) is a Bayesian sequential Monte Carlo method (SMC). It keeps track of an object through a *Probability Density Function* (PDF), which may be non-Gaussian and even multi-modal [1, 7].

The objective of the PF is to evaluate $p(\mathbb{X}_t \mid \mathbb{Y}_{0:t})$, where $\mathbb{X}_t$ is the vector of all available states in the time t, and $\mathbb{Y}_{0:t}$ are all measurements up to the time t. Instead of directly calculating $p(\mathbb{X}_t \mid \mathbb{Y}_{0:t})$, what has happened before time t is modeled into a large set of particles, where the number of particles in each location and their weights estimate how likely the position is. For each new time step, the $p(x_t \mid \mathbb{Y}_t)$ is calculated for each particle.

Initially, the PF algorithm starts with a large number of random samples (particles), where each particle is given a weight that characterizes its quality. At the beginning, each particle has the same weight. The estimated state is given by calculating the weighted sum of all particles. There are three important steps in the PF algorithm:

- Prediction
- Correction (Filtering)
- Re-sampling

During the *Prediction Step*, each particle is moved according to a random value of the state model including the modeled noise. In our case, we have a good idea of how the ship is moving because of the navigational sensor equipment, which measure the ship's Reference Data (RD), estimating both the position via dead reckoning calculation and the orientation. We also know the noise that these sensors have. This will give us a PDF of to where the state has moved, and for each particle we then pick a random value from that distribution.

During the *Correction step* the weight of each particle is regenerated according to the sensor readings that can validate the probability of each state. In our case, we use the bottom depth and/or magnetic field vector compared to maps, to estimate how likely it is that each particle is in the correct position. This step is also known as the filtering step.

During the *Re-sampling step* new particles are resampled randomly according to the PDF (including weights) of the old particles. By this step, there will be many new particles in states where the *Correction step* has judged the probabilities for the particles to be high, and few where the probabilities were low. The old particles from the previous step will not be used any more, and can now be discarded.

One cycle with the *Prediction step*, *Correction step* and *Re-sampling step* is now complete, and the algorithm continues with iterations for the newly sampled particles, see figure 2.

The complete algorithm looks as follows:

1. **Initialization**
   - t = 0
   - Generate N initial samples with an initial distribution of the position.
2. **Prediction**:
   Predict how the particles are moving to the next position regards to the RD.
3. **Correction (Filtering):**
   Compute the weights for each particle and normalize the weights, i.e. compute how likely it is that the particle is positioned where it is, regards to measured bottom-depth.
4. **Re-sampling**:
   Generate a new set of N particles according to how the previous particles are distributed including their weights.
5. Increase t, and **iterate** to step 2.

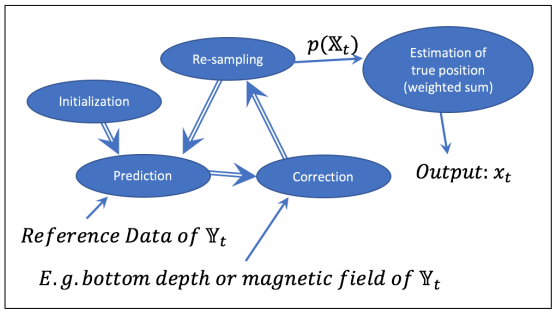

Figure 2: A block diagram of a Particle Filter. After initialization, the particle filter iterates between *Prediction*, *Correction* and *Re-sampling*. After the *Re-sampling Step*, a probability of each state $p(\mathbb{X}_t)$ is estimated from the particles. From this, a weighted sum of all particles can be used as an estimation of the true position.

2.3 Depth data in the particle filter

Reference [2] describes how elevation can be used in the PF for navigation of an airplane. In [3] almost the same technique is used for the domain of naval ships, where a PF is used for estimating the position based on how the bottom depth measurements varies when moving in a trajectory.

The state of the system is denoted as x_t. The state can contain different variables depending on what sensors are available and how complex we want the algorithm to be. [3] suggests using the Cartesian position (X, Y) and the crab angle δ for x_t (the crab angle δ is the angle between the direction the ship is pointing towards and the direction of the velocity of the ship).

$$x_t = (X_t \quad Y_t \quad \delta_t)^T \tag{3}$$

The following equation with discrete time with the sample time Δ, model the state:

$$x_{t+1} = f(x_t, u_t, w_t) = \begin{pmatrix} X_t + v_t \Delta sin(\varphi_t - \delta_t) \\ Y_t + v_t \Delta cos(\varphi_t - \delta_t) \\ \delta_t \end{pmatrix} + w_t \tag{4}$$

In this equation $u_t = (v_t \quad \varphi_t \quad \theta_t \quad \phi_t)^T$ is the input signal, which consists of the speed v_t, compass angle φ_t, the sensor azimuth angle ϕ_t relative to the stern of the ship and the sensor elevation relative to vessel θ_t. The w_t is the process noise. The range to the sea floor is measured in the direction from the sonar sensor. The measurement relation is given by the following equation [3]:

$$y_t = h(x_t, u_t) + e_t = r(x_t, \phi_t + \varphi_t + \theta_t) + e_t \tag{5}$$

where $r(x_t, \phi_t + \varphi_t + \theta_t)$ is the range measured from the position of x_t with the azimuth angle $\phi_t + \varphi_t$ and elevation angle θ_t. A sonar sensor that measures the range to the bottom typically has a fixed elevation and azimuth, where the sensor normally is pointing straight downwards.

We now have the model for how to go from one state to the next state in function (4) and we have the model for how the measured value y_t depends on the state in function (5).

If they would have been analytically solvable, the Bayesian calculations in function (1) and function (2) would have been used for calculating the position given all history. Now that they are not, PF instead is used, according to the algorithm in section 2.2.

2.4 Magnetic data in the particle filter

The earth is surrounded by a magnetic field, a phenomenon, which has been used by compasses for many decades. The compass needle points towards the magnetic north, which might give the user the idea that the magnetic field is horizontal to the surface of the earth. The magnetic field is in fact more accurately represented by its *declination* and *inclination* [4]. The declination describes the horizontal deviation of the magnetic field, and it is this field which is measured by the compass. The inclination describes the vertical deviation of the field, and this is more or less neglected in the compass by, e.g., arranging the compass needle on a floating device on a water bed. The magnetic field also varies depending on the time of the day, but the fluctuations are relatively small with fluctuations between 10 nT and 30 nT, which is less than 0.1% of the average magnitude of 48.19 μT [4].

Each ferromagnetic element disturbs this magnetic field, and these disturbances can for indoor environments be even greater than the natural magnetic field of the earth [5]. For ships, the local ferromagnetic elements onboard the ship disturb the compass. When navigating, bigger ships

often have had a binnacle, which has two movable compensating magnets trying to compensate for the magnetic disturbances of the ship. There are also techniques for manually correcting the compass course depending on which direction the compass points towards. For a long time, the compass direction has been the desired sensor measurement, and the disturbances the thing that is to be minimized.

In [5] and [4], the disturbances are instead considered as a signal rather than as noise. For indoor environments, the many ferromagnetic elements create a complex magnetic environment where the magnetic vector varies greatly depending on where the sensor is located. The magnetic field is also quite stable if no major furniture or iron walls are moved. In [5] and [4] all three dimensions of the magnetic field vector are considered, i.e. not only the magnetic intensity. This information is compared to a magnetic map with a PF, and in conjunction with some sort of odometry, such as wheel encoders or inertial sensors, it has in [4] been possible to precisely localize a human or robot. In [5] only cheap smartphone sensors are used, where the 3-axis magnetic field and acceleration are used for determining the position of the user.

In [4], the following PF algorithm is presented.

1. **Initialization**
   - Generate N particles and give them a random starting position, heading and drift rate.
2. **Prediction** - For each particle:
   - Increase/decrease the drift rate, and update the heading according to the drift.
   - Update the heading according to measured heading changes.
   - Update position according to traveled distance and the heading.
3. **Correction**
   - Check if each particle is within the mapped area, and if so, calculate the weight of each particle. The weight is calculated by a likelihood function that compares the difference between the magnetic field in the map, and the measured magnetic field.
   - Normalize the particle weights to sum 1.
4. **Re-sampling**
   - Resample the particles
5. Iterate to step 2.

In [4], there are three different alternatives for the likelihood functions calculating the weights. The simplest function only measures and compares the magnetic intensity, the second measures and compares the horizontal and vertical magnetic field component, and the third measures and compares the full 3-dimensional magnetic vector. As can be expected, [4] shows that the third algorithm performs better than the second one, and the second algorithm performs better than the first one. Especially the robustness and the time for filter convergence have improved when going for higher dimensions.

Although [4] and [5] have explored indoor environments, the same algorithms are applicable for outdoor environments. The magnetic field does not fluctuate as fast as in indoor environments, but on the other hand it is more stable, because no furniture or building parts are moved around as in the indoor environments. There are satellite maps available covering the entire magnetic field of the earth, and in some areas of the world, accurate magnetic field maps have been created. Therefore, the magnetic field is a good candidate to be used for the PF algorithm when estimating a ship position, at least as a complement to the bottom depth.

2.5 Using other data in the particle filter

The bottom depth and the magnetic field are good candidates to use for the PF algorithm when estimating the position, but there are other alternatives. In addition to the bottom depth, [3] also uses range measurements to land objects in another PF algorithm. This range is measured by a radar, and is compared to a sea chart database.

It is also possible to not only use the depth directly to the bottom. If the ship is equipped with a sonar system, it is also possible to use multiple bottom depth measurements covering a large area at once. This will increase the performance of the PF even further, as it will be possible to evaluate if

the bottom readings matches the map with better precision.

The strength of the PF algorithm is that it is very flexible when it comes to which measurements to use. The important thing is that the measurements shall vary enough when changing position, and that it shall have varied in the same way (or in a predicted way) when the map was created, and when doing the PF measurement. Other candidates which could be used for the PF algorithm are:

- Celestial navigation items such as star positions, where a star either is present in a proposed direction, or is not.
- Gravitation, which vary depending on where the ship is located on the earth.
- Various types of available bearing measurements, depending on which sensors the ship is equipped with. For instance, radio and radar sources with known map locations can be used, if the ship's sensors are able to estimate the bearing to that kind of sources.

3 Limitations with current research

The papers referenced from this paper show that it is possible to do accurate position estimations if having sensors measuring data which has previously been mapped into accurate maps. Many of the references also evaluate how accurate the position estimation can become, when already having accurate maps available. However, there are some limitations with the current research.

The algorithms proposed in the studied research papers require that there are highly accurate maps. This is not the case out on open water, and not even in most coastal areas. The reality is that different areas have been mapped with various accuracy, where highly trafficked areas more often have better accuracy than less trafficked areas. The algorithm for positioning in e.g. [4] assumes that it can get the true bottom depth in any position of the map, but from a normal sea chart it is more likely that it is possible to compute some sort of likelihood distribution of the bottom depth for each position. In figure 3, a 1000 m wide part of a sea chart is presented as an example with the position of interest marked with an X. In figure 4 an example of the bottom depth likelihood distribution is presented, which gives the algorithm an estimation of the bottom depth, when no accurate bottom depth is available. It should also be noted, that when creating a sea chart, the most important thing is that there is no shallower area in the map than what is presented. If there are indications that there are bottom depths of 15 m, 18 m and 22 m in an area, it is quite unlikely that there are any depths of 10 m in the middle of these indications, but there could be bottom depths of 30 m.

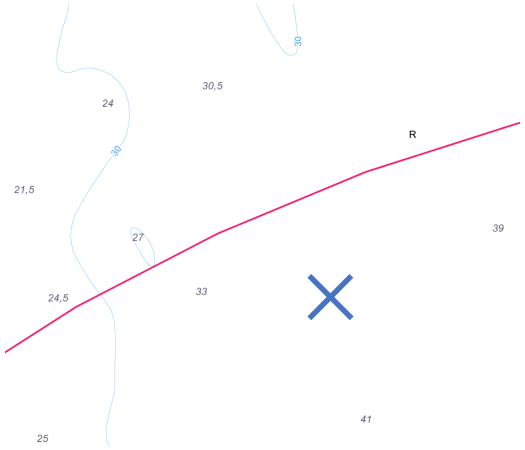

Figure 3: An example of a sea chart and the current position of interest marked with a blue X. (The red line marks the area surrounding a lighthouse in the sea chart.)

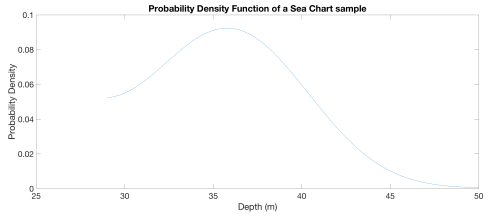

Figure 4: An example of how a bottom depth likelihood distribution could look like for the position of interest in figure 3.

The user platforms that are most likely to have a need for a system for accurate position estimation techniques which eliminates the need for GNSS

systems, are not a cheap platform with moderate navigation systems. The most probable platform is instead an advanced platform with accurate and expensive navigation sensors, where the RD, bottom depth and speed can be measured with high accuracy. The extra expense for buying and integrating a magnetometer sensor for measuring also the magnetic field vectors are foreseen to be a minor investment on platforms like this.

The current research in this field has mainly focused on achieving good performance of the positioning systems when having limited performance of the sensor suite, but nearly unlimited accuracy of the map. In this paper, the goal is to investigate if it is possible to do it the other way around. The main research question is therefore: **Is it possible to navigate accurately enough without GNSS systems, only relying on high performance navigation sensors and normal sea chart and magnetic charts?** In some areas, the sea charts are accurate, and in other areas the magnetic charts can be more accurate. It would therefore be a good feature to be able to estimate the position both with magnetic measurements and bottom depth measurements at the same time, and weighting the fusion between the two, depending on which data that has the highest confidence level. When both maps are not accurate enough, dead reckoning can instead be used for a while until going into areas with enough accuracy in the maps.

4 Proposal of a new implementation

4.1 Proposed Implementation

The future goal is to implement something in line with the following algorithm:

1. **Initialization** - Generate N particles and give them a random starting pose around a manual estimation of the starting position.
2. **Prediction Step** - Move each particle according to the ship's total navigation sensor suite measurements including their probability distributions.
3. **Correction Step** - Calculate the weight for each particle given all available sensors and maps.
4. **Re-sampling Step** - Re-sample the particles.
5. Iterate to step 2).

The most interesting step in the algorithm is the *Correction Step*, which will need the following parts to work:

4.1.1 Map data

To make the PF algorithm operable, there must be data supporting the likelihood calculations, which estimates how likely it is that the current measurement has been performed at each location. If the goal is that the PF should be possible to use on most places, it is important not to require high accuracy maps. The best is if it is possible to use the best available information in each area, i.e. high accuracy maps where available and normal sea charts for bottom depth information if only those are available. To support the PF algorithm, a function is needed to create bottom depth values from a sea chart database, including confidence estimations. If the algorithm e.g. can read surrounding bottom depth coordinates and surrounding bottom depth curves (see figure 3), an estimation of the bottom depth and confidence estimations can be calculated. In the simplest form, only a single bottom depth estimation is given from the function, and almost as simple as this would be to let the sea chart give an interval of valid bottom depths, e.g. 20-30 meters. The best support for the PF algorithm would be to give an accurate PDF estimation (see figure 4) based on knowledge of how sea charts are created. The same type of algorithm would be needed for magnetic data.

4.1.2 Fusion of sensor data

On an advanced ship with high precision navigation sensors, it can be acceptable to use the navigation sensors for dead reckoning without using global positioning techniques for some time. It will take a long time before the drift has become large enough for resulting in a completely inaccurate position. When using the PF algorithm to correct the position, it is therefore important to not spoil the advantages of the already well working navigation system, by lean to much against the estimation of

the position from the PF compared to the dead reckoning algorithm. If e.g. there is uncertain evidence that estimates a particle is in the wrong position, it is better to let the particle remain, than removing it. The worst thing that could happen is if all particles at the correct position eventually is discarded, which could happen if the local measurements have not been accurate enough, the maps are not accurate enough, or the map measurements have changed, e.g. due to some external effect. To meet this challenge, we propose dividing the particles into subsets in the beginning of each Correction Step. Then the particles in each subsets are corrected according to the correction rule in the particular subsets. There are different alternatives of how to divide the particles into sub-groups, when using magnetic and bottom depth data for the PF algorithm. We propose the following alternatives:

1. Divide the particles into three subsets, where one subset of particles will be weighted according to a PF algorithm working with bottom depth, one subset of particles will be weighted according to a PF algorithm working with magnetic fields, and the last subset will have equal weights, where only the dead reckoned position from the RD matters. The size of each subset can then be determined by the quality of the bottom depth and magnetic maps/measurements compared to RD accuracy. The advantage is that e.g. bad magnetic measurements or maps not will damage the subsets where magnetism is not taken into consideration. The drawback is that it will take longer time before the PF converges to the correct position.

2. Divide the particles into two subsets, where one subset of particles will be weighted according to a PF algorithm working with both bottom depth and magnetic fields at the same time, and the other subset will have equal weights, where only the RD matters. The size of each subset including its internal subsets can be determined by the quality of the bottom depth and magnetic maps/measurements. By combining both magnetic fields and bottom depth into $\mathbb{Y}$, the PF will be able to calculate the probability density $p(x_t \mid \mathbb{Y}_t)$ very efficiently. The drawback is that particles can be discarded incorrectly if any measurements or the maps are inaccurate.

3. By combining 1) and 2) and having four sets of particles, the advantages can be taken from each solution. If bottom depth is better than magnetic fields in one area, the particles can e.g. be divided according to table 1.

Table 1: Example distribution of particles

Subset	**Nbr of particles**
Depth and magn. field	50%
Only bottom depth	20%
Only magnetic field	10%
No PF (only RD)	20%

In this way, the strength in combining data to support the PF are used by half of the particles. The other half of the particles are more carefully used, so that some particles will survive even if local measurement errors occur or maps are inaccurate.

4.2 Present situation of the implementation

To start the investigation of the possible solutions, a Python program has been created to explore the possibilities with using depth measurements for the PF, see figure 5 (left). In this program, a sea chart is digitized into 10x6 squares by manually setting a lower and upper boundary of the bottom depth in each square. In the initialization of the program, the ship is placed in an (for the algorithm) unknown position, marked with a green dot. The program then iterates through the following algorithm:

1. **Initialization** - Generate 1000 particles and put them into a random square of the 60 available squares. This can be seen in figure 5 (left), where some small blue dots can be seen in the middle of each square, where the size of the dots indicate how many particles are located in each square.

2. **Prediction Step** - The user then moves the ship by pressing some direction and speed in

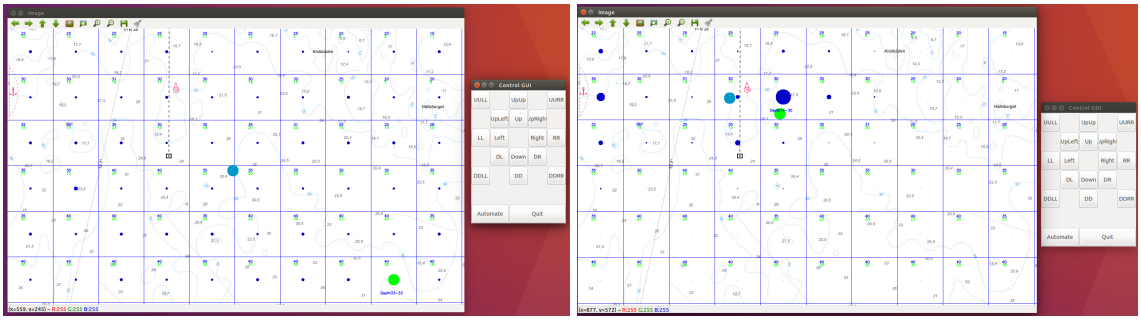

Figure 5: (left) In the initial state, the particles (indicated by the size of the blue dots) are randomly distributed between all 60 squares. The light blue dot in the middle of the sea chart indicates the weighted sum of all particles, and is the current estimation of the true position of the ship. The green dot to the lower right, indicates where the ship is located. The ship can then be moved by clicking the GUI. (right) After 8 random steps, the square with the highest number of particles is the same square as the ship. The weighted sum of all particles (the estimation of the position), is still in the wrong square.

the GUI (or alternatively the Automate button for a random movement). A random error is added to both the direction and speed, and then the ship is moved according to those values. Each particle is also moved according to an estimation of the random movement. (If the ship or particles hit the boarder of the map, they stay on the border.)

3. **Correction Step** - When the ship has moved, it will measure 10 random bottom depths in the square, and removing the biggest and smallest values in order to increase the resistance against error measurements. Then it starts comparing each particle's square's minimum and maximum depth to the measured values by the ship. If the measurements are within the interval, 10 is given as a weight for the particle. If not in the interval, 2 is given. Misplaced particles will then for each iteration decline in number, in favor for well-placed particles.

4. **Re-sampling Step** - Next, 1000 new particles are re-sampled according to the weighted sums in each square.

5. Iterate to step 2).

After an example run of the program, about eight moves from an initial location in south-east, mainly in the direction north-west, the square with the most particles is the same square as the square where the ship is located, which can be seen in figure 5 (right). After 10 moves, the weighted sum of all particles is located in the same square as the ship, which is shown in figure 6.

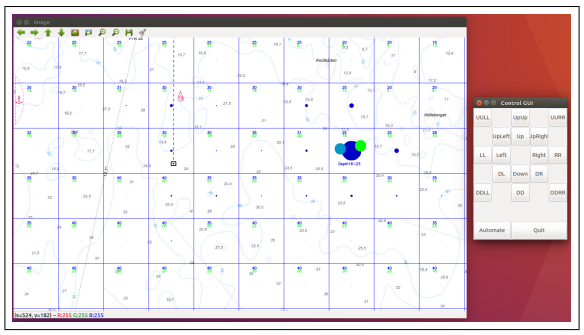

Figure 6: After 10 random steps, both the square with the highest number of particles and the weighted sum of all particles are in the square where the ship is located.

4.3 Further development of the implementation

There are several ways of refining the program. Some of the most important features are as follows:

- Adding support also for magnetic field maps.

- Using data from a real sea chart database, and creating a function for estimating bottom depths from that data set.
- Evolving the Correction Step by implementing the features in table 1 in section 3.
- Evolve the principles of how to be weighting the different subsets to increase the stability and the robustness. This can possibly be done on-line using machine learning, by feeding the "true position" from the GPS.

After finishing the implementation, a comparison needs to be performed on data originating from a real ship. The navigation performance can then be compared to the GPS location of the same ship.

5 Conclusion

It has already been shown that PF algorithms can be used for estimating positions [1, 2, 4, 6], at least for other domains than for naval ships. In this paper, an idea of how to use this knowledge in an algorithm more suitable for real world scenarios has been presented. A brief explanation of how to do the implementation has been discussed, and the present implementation has been presented along with proposed future upgrades.

If it is possible to navigate accurately enough without GNSS systems, only relying on high performance navigation sensors and normal sea chart and magnetic charts remains unclear. Further implementation and testing with real ship data is first needed.

ACKNOWLEDGMENT

This work was partially supported by the Wallenberg Autonomous Systems and Software Program (WASP).

References

[1] Frank Daellaert, Dieter Fox, Wolfram Burgard, Sebastian Thrun "Monte Carlo Localization for Mobile Robots", Proc. IEEE International Conference on Robotics and Automation (ICRA-99), 1999

[2] Fredrik Gustafsson, Fredrik Gunnarsson, Niclas Bergman, Urban Forssell, Jonas Jansson, Rickard Karlsson, and Per-Johan Nordlund "Particle Filters for Positioning, Navigation, and Tracking", IEEE Transactions on signal processing, vol. 50, No. 2, 2002

[3] Rickard Karlsson and Fredrik Gustafsson "Bayesian Surface and Underwater Navigation", IEEE Transactions on Signal Processing, vol. 54, No. 11, 2006

[4] Martin Frassl, Michael Angermann, Michael Lichstenstern, Patrick Robertson, Brian J. Julian, Marek Doniec "Magnetic Maps of Indoor Environments for Precise Localization of Legged and Non-legged Locomotion", IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2013

[5] Etienne Le Grand and Sebastian Thrun "3-Axis Magnetic Field Mapping and Fusion for Indoor Localization", 2012 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI), 2012

[6] Michael Angermann, Patrick Robertson "Inertial-Based Joint Mapping and Positioning for Pedestrian Navigation", Proc. ION GNSS, 2009

[7] Lawrence A. Klein "Sensor and Data Fusion - A Tool for Information Assessment and Decision Making", Publisher: SPIE Press, 2012

[8] Todd E. Humphreys, Brent M. Ledvina, Mark L. Psiaki, Brady W. O'Hanlon, and Paul M. Kintner, Jr., Cornell "Assessing the Spoofing Threat: Development of a Portable GPS Civilian Spoofer", ION GNSS Conference, 2008