DIVING IN DENSITY: CONTROLLING THE DEPTH OF A PROFILING FLOAT IN COASTAL WATERS

Eemeli Aro, Mika Vainio, Zhongliang Hu and Aarne Halme
Finnish Centre of Excellence in Generic Intelligent Machines Research
Aalto University Department of Automation and Systems Technology
Espoo, Finland
email: firstname.lastname@tkk.fi

ABSTRACT
Controlling a profiling float as it dives in coastal waters is a challenging task, requiring efficient and accurate movements while in a constantly changing environment that the float can't sense remotely. With the algorithms presented in this paper, we are able to control a SWARM float's vertical position with an accuracy of ±1m safely, energy-efficiently and without prior knowledge of the specific conditions of the water in which it is deployed. The only instruments used are the on-board pressure and temperature sensors along with a downward-facing echosounder for sea-bottom avoidance. The algorithms have been verified using simulated floats in a realistically modelled Baltic Sea environment.

KEY WORDS
UUV control, adaptive control, and profiling float control.

1. Introduction

A profiling float is a freely drifting oceanographic measurement platform with buoyancy control. In slightly different terms, it is an autonomous robot moving in a three dimensional fluid with only one actuator that lets it indirectly control its depth. Most floats are a part of the Argo array, surveying the world's oceans to provide real-time information from depths of 1000-2000m [1]. The Autonomous Underwater Multi-probe System for Coastal Area/Shallow Water Monitoring (SWARM) project, on the other hand, requires its floats to operate in the brackish Baltic Sea which has an average depth of just 50 meters. This means that the depth control requirements of a SWARM float are of an entirely different class compared to most other floats.

A float moves by changing its volume using an oil bladder and pump or a mechanical piston. This leads to a change in the float's overall density, and provokes a vertical movement as the float finds the matching density in its water column. The resulting depth cannot be absolutely determined before it is reached, as the density of the water varies with the environmental conditions, primarily pressure, temperature and salinity. In order to accurately predict the density at a desired depth, very accurate temperature and salinity predictions are therefore required. In large oceans and at depths of 100s or 1000s of meters, these variables change rather slowly. In the Baltic Sea, on the other hand, conditions are prone to changing unpredictably.

The paper first presents the environmental and hardware considerations particular to the SWARM float and its mission. Next, the initialization of the float's environmental estimates is presented before walking through the float's diving control and estimate update methods. Finally, a sea simulator is presented and the models and methods are tested with it.

2. Physical Realities

The value of services provided by the coastal seas, including estuaries, to the human welfare is estimated to be higher than those of terrestrial or open ocean systems. Coastal waters supply food via fisheries, renewable and non-renewable resources like sand and hydrocarbons, sites for recreation, and sites for waste disposal, and especially for effective nutrient cycling [2]. The relevant spatial and seasonal scales for biological variability are often related to hydrophysical events; these are mostly unpredictable and practically impossible to cover by traditional monitoring with sparse sampling.

SWARM was an EU-funded (FP5, 2003-2005) project aiming to design, implement and test a multi-robot system that could measure local and transient biological and physical variability in the Baltic Sea and similar areas at the scale relevant for single events. See [3] for a generic presentation of the project. Based on the initial ideas and hardware designed and built during the SWARM project, the Finnish Centre of Excellence in Generic Intelligent Machines has continued the related research.

2.1. The Baltic Sea

The surface salinity of the Baltic Sea varies from 1-2 PSU (practical salinity units) to 20 PSU, while the salinity of most other seas stays in the range 33-36 PSU from the surface to the bottom. In the Baltic Sea, salinity tends to increase with depth, with a significant change (halocline) at a depth of 40-80m separating the surface and bottom waters. In the Gulf of Finland, the salinity may vary from 6 PSU on the surface to 10 PSU at the bottom (120m at the deepest). Temperature in the Baltic Sea also varies greatly with depth, along with a significant annual variation. In the summer, a warmer wind-mixed surface layer of 10-25m develops and is separated from the deeper waters by a thermocline where the temperature may drop 10°C within a few meters. In the winter, a large part of the Baltic is frozen.

2.3. The SWARM Float

A SWARM float is a part of a system of multiple homogeneous, robust and easy to use underwater robots (autonomous intelligent profiling floats) that can perform autonomous missions of up to two weeks. These floats may communicate with the control station via Iridium satellite communication and use inter-robot acoustic ranging and communication for localization and data exchange. See Figure 1 (block diagram) and Figure 2 (the SWARM float). Controlled movement is achieved using a motor-driven piston that allows the float to accurately control its volume and therefore its density, inducing the float to move vertically to a matching depth.

2.2. The Depth-Density Relationship

The relationship between water density and depth is of high importance to a float, as its mode of movement depends on the monotonic increase in density with respect to depth. Density may be approximated by the international equation of state for seawater [4], which links it together with pressure, temperature and salinity. The range of densities experienced in the Baltic Sea is rather limited, ranging from about 1003 kg/m³ on the surface to 1008 kg/m³ at 100m. This means that the change in a float's volume that is required to move to a different depth is relatively small and therefore has a low cost in terms of energy.

Of the environmental measurements required for estimating and predicting water density, pressure and temperature may be directly measured with instruments that have a high precision. Salinity, on the other hand, is commonly determined by measuring water conductivity, from which (together with pressure and temperature measurements) salinity may be calculated. Unfortunately, the conductivity sensor presently used on SWARM floats is an experimental prototype with an expected accuracy of ±20%, which makes its online use inadvisable. Primarily due to this limitation, the developed depth control method only uses the pressure and temperature sensors, along with the known density of the float itself. When the float is vertically stable, the water density matches the float's own density, and the water salinity may be determined using the equation of state [4].

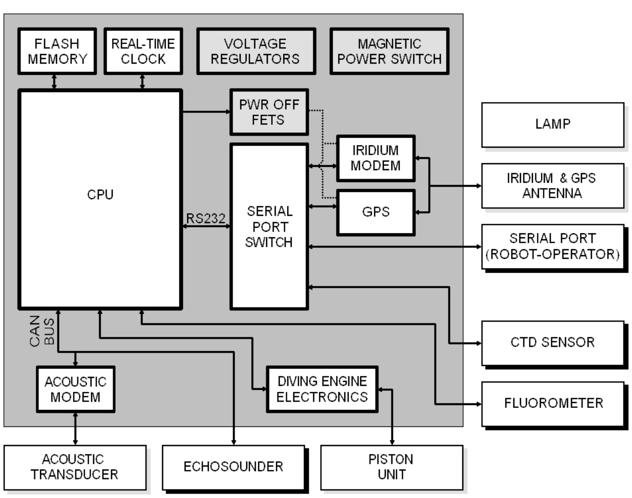

Figure 1: Block diagram of SWARM float subsystems

3. Initialization

For ease of deployment, the SWARM float requires no prior information or assumptions about the environment in which it is deployed. In order to gather the required data, an initialization dive is performed.

In normal operation, the float is required to dive to a specified depth by adjusting its density, but to start with it has no knowledge of the depth-density relationship. The depth-density map is built during the initialization dive, during which the float dives down in steps until reaching the sea-bottom (detected using the echosounder) or its maximum operational depth, and then dives back up to the surface.

As the sea-bottom depth and other environmental properties of the water may be unknown before initialization, the time it takes the float to dive down is also unknown. In order to make the initialization

Figure 2: Three SWARM floats, ready for testing

practical, it is given a maximum time of two hours. Upon reaching the bottom of its dive, the float determines how long on average each of its steps down has taken, and from time left until the end of initialization calculates the number of steps it may make on the way up. As the surface density is also unknown, the first three steps are assigned to densities between the estimated surface density and the next density down. The rest of the steps up are distributed evenly in the remaining gaps between steps by filling them from the top down. Depending on the environmental conditions, this should allow the float to stop at 6-10 different depths during the initialization dive.

If the initialization time runs out before the float has surfaced, the initialization is terminated and the float returned to the surface. Initialization is seen to end when the float surfaces, which in most cases happens when the float is attempting to reach a density just below the estimated surface density. Once back at the surface, the tables of estimates for temperature and salinity are initialized from the gathered data, providing the required mapping between depth and density.

In addition to determining the environmental conditions, this initialization also acts as a verification of the float's systems; if the float has a faulty sensor or some other error is detected, the float's mission may be aborted before it even properly begins.

4. Diving Control

A SWARM float is expected to have a repeating operational cycle of at most a few hours, during which it may stay at one or more depths. Based on this, the diving control needs to take into account three conflicting requirements: diving to a given depth needs to be rapid, accurate and energy-efficient. Of particular concern is the energy-efficiency, as the energy required for movement is one of the most significant factors in limiting the maximum mission time of the float—the float's batteries can't be recharged during a mission.

In order to achieve the desired motion, our solution for float diving control attempts to minimize the number of piston movements that are required, as each movement needs to overcome significant static friction in the motor and the piston. In other words, the float will estimate the density at the target depth and adjust its density to match.

Once a piston movement has been made, the float will eventually settle at some depth in the water. If this depth is within the required tolerance of the target depth, no further action is required and the temperature and salinity tables of estimates are updated. If the target depth has not been reached, another piston movement is required. Method 1 below gives an overview of these steps.

As the water density varies rather little and the drag force experienced by the float is relatively large, the vertical velocity of the float may for the most part be left uncontrolled; the length of a dive will not be sufficiently long to accelerate the float to a high velocity, and the overshoot experienced when settling to a depth will be at most in the tens of centimeters. This is in rather strong contrast to deep-sea floats which dive between much greater depths and therefore need to limit their rate of ascent or descent in order to take measurements at their required depth resolution.

Due to the above-described environmental conditions, SWARM float depth control is more of a density prediction problem rather than a control problem. Further, given the complex relationship between density and pressure, temperature and salinity as well as the independence of these variables from each other, it makes sense to track and predict each separately. Keeping the variables separate not only allows for their estimates to be updated separately, but also allows other properties of the seawater to be determined from the same data, such as the speed of sound.

Method 1: Diving to a new depth

1. Estimate density at the goal depth.
2. Store the current depth and density.
3. Move the piston to match the goal depth density.
4. While diving, measure pressure and temperature continuously.
5. Wait for vertical motion to end.
6. Check divergence between current and goal depths, and if too far, re-estimate goal depth density and continue from step 2 (see Section 4.2).
7. Update the table of estimates for temperature (see Section 4.3).
8. Using the new temperature estimates and the stored depths and densities, update the table of estimates for salinity (see Section 4.4).
9. Done; arrived at goal depth.

An earlier version of this method was presented in [5].

4.1. Determining Depth from Pressure

Gauge pressure (absolute pressure minus atmospheric pressure) and depth have an almost linear relationship, so with a pressure reading and knowledge of the atmospheric pressure at sea level we can determine with high accuracy the depth of a float. Using the method of Fofonoff and Millard [4] with corrective terms for the Baltic Sea by Leroy and Parthiot gives a pressure-to-depth conversion accuracy of ±0.1m without needing to take into account the local salinity and temperature [6].

The pressure sensor of the SWARM float has a listed accuracy of ±80mbar, giving a depth resolution of ±0.8m. In order to improve on this and to take into account the drag-induced settling of a float as it dives to a new depth, we make use of a piecewise linear online segmentation of the depth data, which gives us a filtered estimate of the current depth and rate of ascent or descent. This filter also provides us with the length backwards in time for which the estimate is valid, as well as the variance of the actual measurements from the estimate.

The developed segmentation algorithm and its use in other applications [7] will be presented in detail in a later paper.

4.2. Density Corrections

It is unavoidable that the float will at times initially settle to a depth that diverges from the target depth. In this situation, the density of the target depth will need to be re-estimated, taking into account all new data gathered during the preceding movement or movements. Given the complex relationship between depth, density, temperature and salinity, this means that the temperature and salinity estimates that are based on earlier data are to some degree incorrect, and that their use in density estimation is questionable. In such a case, it may be more beneficial to base the estimate directly on the most recent depth-density vector of measurements.

During a dive, the float keeps track of each measured temperature and density. When a density adjustment needs to be made, new tables of estimates for temperature and salinity are calculated using this data (see sections 4.3 and 4.4), and an updated estimate of the target density is made using the equation of state for seawater [4].

If density measurements exist for depths above and/or below the target depth, they are also used to improve the estimate. If the divergence between the measured and estimated densities at these depths is small enough, a linear interpolation is made of that divergence and applied to the target density estimate. Otherwise, the estimated density is discarded altogether, and the target density taken as a linear interpolation of the densities above and below it. Algorithm 1 presents this method in more detail.

Algorithm 1: Re-estimating goal density

For each depth z, with:

$\boldsymbol{d}$ a vector of density measurements (d_i, z_i) sorted by increasing depth and

$\tilde{\mathrm{d}}(c)$ an estimate of the density at depth c based on the temperature and salinity models,

We can find $\mathrm{d}'(z)$, the density estimate for depth z using $n = \mathrm{len}(\boldsymbol{d}) - 1$ as the index of the lowest measurement in $\boldsymbol{v}$:

if $(z < z_0)$:

$$\mathrm{d}'(z) = d_0 - \begin{cases} \tilde{\mathrm{d}}(z_0) - \tilde{\mathrm{d}}(z), & \tilde{\mathrm{d}}(z) < d_0 \\ (z_0 - z) \cdot \dfrac{d_1 - d_0}{z_1 - z_0}, & \text{else} \end{cases}$$

else if $(z > z_n)$:

$$\mathrm{d}'(z) = d_n + \begin{cases} \tilde{\mathrm{d}}(z) - \tilde{\mathrm{d}}(z_n), & \tilde{\mathrm{d}}(z) > d_n \\ (z - z_n) \cdot \dfrac{d_n - d_{n-1}}{z_n - z_{n-1}}, & \text{else} \end{cases}$$

else:

with (d_a, z_a) the nearest measurement in $\boldsymbol{d}$ above z and $p = \dfrac{z - z_a}{z_{a+1} - z_a}$ as a measure of the position of z in $[z_a, z_{a+1}]$, along with $\mathrm{e}(k) = d_k - \tilde{\mathrm{d}}(z_k)$ as a measure of the divergence of measurement k from its expected value:

if $\begin{pmatrix} d_a \leqslant \tilde{\mathrm{d}}(z) \leqslant d_{a+1} \quad \text{and} \\ \max(|\mathrm{e}(a)|, |\mathrm{e}(a+1)|) \leqslant (d_{a+1} - d_a)/2 \end{pmatrix}$:

$$\mathrm{d}'(z) = \tilde{\mathrm{d}}(z) + [(1-p) \cdot \mathrm{e}(a) + p \cdot \mathrm{e}(a+1)]$$

else :

$$\mathrm{d}'(z) = (1-p) \cdot d_a + p \cdot d_{a+1}$$

Once a float has settled at its target depth, it may still move vertically due to environmental changes. If this movement is greater than the prescribed tolerance, the float will begin a new adjustment cycle in order to reach the target depth.

4.3. Tracking Temperature

The float maintains an internal table of estimates for the temperature at all depths throughout the water column, with a discretization of 1m. This table is updated every time the float successfully dives to a new depth, as that is the only time when synchronous measurements from more than one depth can be gathered. The temperature sensor of the SWARM float has an accuracy of ±0.05°C and a time constant of less than 5s.

The measurements are resampled to the table's resolution of 1m by taking the weighted mean of values in the nearest ±1m for each depth. Missing values or those with too few measurements are interpolated from the surrounding values. Values outside the range covered by the current dive are untouched except for those within a few meters of the dive's minimum and maximum depth, for which a decreasing offset is calculated to let the new and old data gradate continuously.

Algorithm 2 describes the generic estimate update method, while Algorithm 3a specifies the particular interpolation method used for temperature.

Algorithm 2: Updating a table of estimates

For each depth $z \in \{1\text{m}, 2\text{m}, 3\text{m}, \ldots\}$, with:

- $\boldsymbol{v}$ a vector of measurements (e.g. temperature, salinity) (v_i, z_i) sorted by increasing depth,
- $\mathrm{v}(z)$ the previous estimate of the value for depth z,
- $\tilde{\mathrm{v}}(z)$ the value at depth z, based only on $\boldsymbol{v}$,
- $\mathrm{v}'(z)$ the new estimate for depth z,
- r_z the range of the depth adjustment and
- r_{grad} the range of the gradation,

We can find $\mathrm{v}'(z)$, the new estimate for the value at depth z using the following measures of the divergence in depth $\delta\mathrm{z}(z)$ and in value $\delta\mathrm{v}(z)$ between $\mathrm{v}(z)$ and $\tilde{\mathrm{v}}(z)$:

$$\delta\mathrm{z}(z) = \operatorname{sign}(z_d - z)\cdot\min\left(r_z, |z_d - z|\right)\Big|_{\mathrm{v}(z_d)\equiv\tilde{\mathrm{v}}(z)}$$

$$\delta\mathrm{v}(z) = \mathrm{v}(z+\delta\mathrm{z}(z)) - \tilde{\mathrm{v}}(z)$$

and the index $n = \operatorname{len}(\boldsymbol{v}) - 1$ of the lowest measurement in $\boldsymbol{v}$:

if $(z_0 \leqslant z \leqslant z_n)$:

$\quad \mathrm{v}'(z) = \tilde{\mathrm{v}}(z)$

else if $\left((z \leqslant z_0 - r_{\text{grad}}) \text{ or } (z_n + r_{\text{grad}} \leqslant z)\right)$:

$\quad \mathrm{v}'(z) = \mathrm{v}(z)$

else:

$$i = \begin{cases} 0, & z < z_0 \\ n, & \text{else} \end{cases}, \text{ the nearest end of } \boldsymbol{v}$$

$$g = 1 - \left|\frac{z - z_i}{r_{\text{grad}}}\right|, \text{ the level of gradation } g \in (0,1)$$

$$\mathrm{v}'(z) = (1-g)\cdot\mathrm{v}(z) + g\cdot\left[\mathrm{v}\left(z + \delta\mathrm{z}(z_i)\right) + \delta\mathrm{v}(z_i)\right].$$

Algorithm 3a: Interpolating temperature

For temperature measurements we may determine $\tilde{\mathrm{v}}_{\text{temp}}(z)$ from $\boldsymbol{v}_{\textbf{temp}}$ of (t_i, z_i) as follows, using $w_i(z) = 1 - \min(1, |z - z_i|)$ as the relative weight given to measurement i:

$$\tilde{\mathrm{v}}_{\text{temp}}(c) = \frac{\sum t_i \cdot w_i(c)}{\sum w_i(c)}\bigg|_{c \in \mathbb{N}}$$

$$\tilde{\mathrm{v}}_{\text{temp}}(z) = (\operatorname{ceil}(z) - z)\cdot\tilde{\mathrm{v}}_{\text{temp}}(\operatorname{floor}(z)) + (z - \operatorname{floor}(z))\cdot\tilde{\mathrm{v}}_{\text{temp}}(\operatorname{ceil}(z)) .$$

4.4. Tracking Salinity

Due to the large measurement uncertainty of the SWARM float's conductivity sensor, in this implementation salinity is estimated using the equation of state for seawater [4] together with measurements for density, temperature and gauge pressure. As the water density may only be determined (from the float's own density) when the float is vertically stable, this approach gives only one new measurement for each vertical movement. As a move from one depth to another may consist of more than one piston movement, each such move will provide a new measurement vector of at least two data points—except when diving to or from the surface, when only one data point may be available due to the inability to exactly measure the water density at the surface.

Similarly to temperature, the float maintains an internal table of estimates for salinity with a discretization of 1m. This table is updated using Algorithms 2 and 3b by finding a minimum distance in depth or salinity from the previous estimate to both of these end points, and adjusting the table of estimates to match the new measurements. Points above and below this data pair are untouched, except for a similar gradation as with the temperature estimates.

Algorithm 3b: Interpolating salinity

For salinity measurements we may determine $\tilde{\mathrm{v}}_{\text{salt}}(z)$ from $\boldsymbol{v}_{\textbf{salt}}$ of (s_i, z_i) as follows:

With (s_a, z_a) the nearest measurement in $\boldsymbol{v}_{\textbf{salt}}$ above z and $p = \dfrac{z - z_a}{z_{a+1} - z_a}$ as a measure of the position of z in $[z_a, z_{a+1}]$, the divergences δz and δv between $\mathrm{v}_{\text{salt}}(z)$ and $\tilde{\mathrm{v}}_{\text{salt}}(z)$ can be calculated:

$$\delta z = (1-p)\cdot\delta\mathrm{z}(z_a) + p\cdot\delta\mathrm{z}(z_{a+1})$$

$$\delta v = (1-p)\cdot\delta\mathrm{v}(z_a) + p\cdot\delta\mathrm{v}(z_{a+1})$$

and therefore $\tilde{\mathrm{v}}_{\text{salt}}(z) = \mathrm{v}_{\text{salt}}(z + \delta z) + \delta v$.

4.5. Escaping the Surface

Movement to and from the surface needs to be handled as a separate case. When intentionally surfacing, the float will need to keep its antenna above the surface in order to make and maintain a satellite connection, so the piston is moved further out to stabilise the depth and to take into account the reduction in the displaced water volume.

Due to the float at least partly surfacing with any density less than that of the water at the surface, measurement of the actual surface water density is difficult. This becomes a significant factor when the float attempts to reach a depth very close to the surface (0-10m below the surface), in which case the error in the density prediction will lead to the float surfacing. The float keeps track of the maximum density at which the surface has been reached as well as the number of adjustments made while remaining on the surface, and adjusts the density estimate of a sub-surface depth to account for these factors.

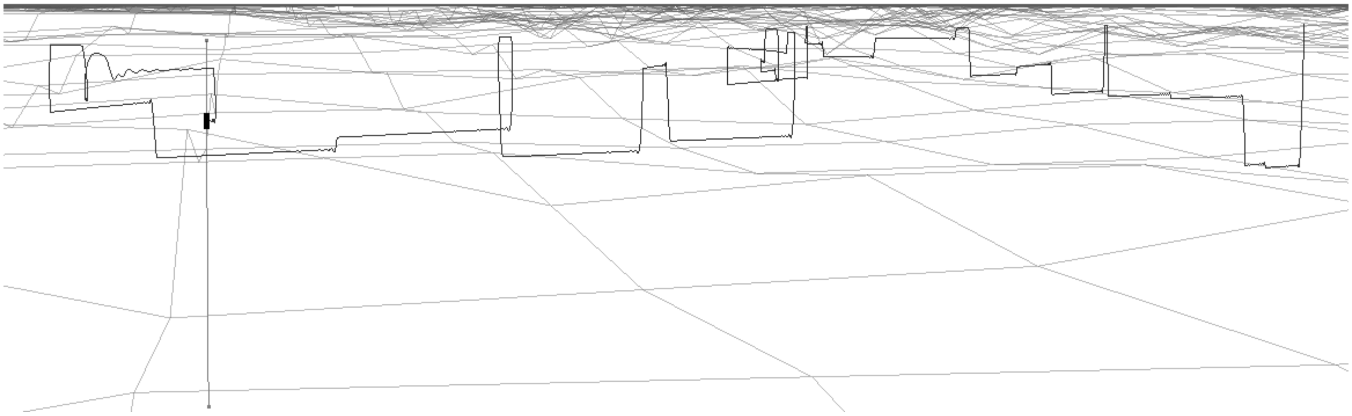

Figure 3: A screenshot of the simulation GUI, showing the position and history of a float over approximately 48 hours. The float's vertical position is indicated by a line connecting the surface to the sea-bottom.

4.6. Avoiding the Sea-bottom

A SWARM float is equipped with a downward-facing echosounder which it uses to prevent potentially terminal collisions with the sea-bottom. The echosounder has a reliable range of at most 8 meters, which is sufficient given the low velocities of the float. The distance to the sea-bottom is combined with pressure sensor information to give an estimate of the bottom depth (z_b), which in turn is used to provide a maximum depth limit that may be attempted ($z_b - z_{safe}$). Additionally the projected path of the float is checked against a closer limit ($z_b - z_{panic}$), and a corrective piston movement is made if that limit is expected to be breached within the following 60 seconds.

Testing with the simulator would indicate that values of 3.0m for z_{panic} and 6.0m for z_{safe} are sufficient to prevent collisions, but these will need to be adjusted upwards to take into account the greater variance of the actual sea-bottom.

5. Simulator Tests

The diving procedures and environmental variable predictors described above have been developed and verified with a simulation of the Gulf of Finland, using a simulator platform developed for this project. The simulator is a platform for testing and developing autonomous floats. It uses a server-client architecture, with each float connecting to the environment as a client and provided with the same interfaces as the actual float has to its sensors and actuators. After a float is initialized, its movement in the water is controlled by the simulator according to a model taking into account its buoyancy and drag forces, as well as the three-dimensional water current vector. Horizontal and vertical positions and velocities are not discretized, and time advances using a fourth-order Runge-Kutta integration with a variable step size of 2.5-3 seconds.

The sea simulator is based on data from the Finnish Marine Research Institute's BalEco ecosystem model [8], which has a variable depth resolution ranging from 3 meters near the surface to 30 meters at a depth of 150 meters. Horizontal resolution is roughly 11km by 11km, and temporal resolution is 6 hours. The bathymetric data has a resolution of about 1.8km x 1.8km. Continuous values are interpolated from the 8 nearest points in four dimensions. The particular dataset used in this paper comes from August 2008. The simulator is in no way limited to modelling the Baltic Sea, as long as the requisite data is available.

The development of this novel simulator was necessary as no other platform was found with the required capabilities of realistically modelled fluid dynamics and support for simultaneous modelling of multiple floats (tens or even hundreds at a time) as well as hardware-in-the-loop testing.

5.1. Test Setup

An intentionally difficult but realistic test scenario was designed to verify the developed algorithms. The float is set to start at a random surface position with a water depth of at least 20 meters within the available data. From that point, the float will first do an initialization dive, followed by a five day mission of diving at 120-minute intervals to random depths, using a tolerance of ±1m, and surface for 20 minutes every 380 minutes. Atmospheric pressure at sea level is kept constant, but its effect on depth measurements should be negligible.

5.2. Test Results

A total of seven runs of ten floats each was made, starting at a random time. This resulted in data for 61 floats after discarding floats that beached in shallow water or drifted beyond the simulation boundaries. In total these floats made 4728 successful dives, of which 1152 were to the surface. As all of the dives to the surface were successful with just one piston move, these are not included in the following. One float made one unsuccessful dive, i.e. was unable to reach its target depth within two hours. 699 adjustments were made by floats due to changes in environmental conditions pushing them outside their goal depth tolerances.

With respect to dive depth, the data shows clearly that diving to shallower depths is significantly more difficult (see Figure 4) due to the faster rate of change and otherwise more complex behaviour of the top water layers. For depths in the range [8m, 60m] the average number of piston moves required to reach a goal depth was 1.134.

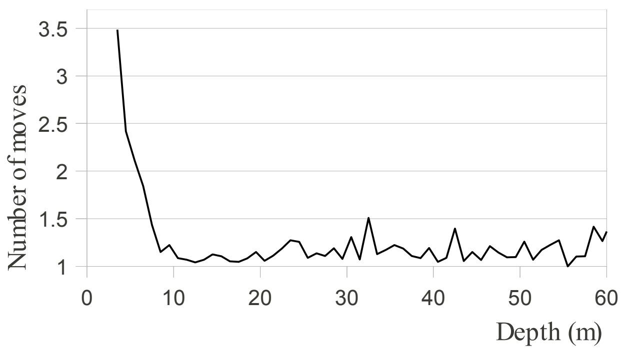

Figure 4: Mean number of piston moves vs. goal depth

Once the initialization dive has been made, the number of piston moves required appears to be independent of time, indicating that the two-hour initialization dive is able to map the environmental state, and that further dives will not improve it radically. For comparison, Figure 5 also shows the required number of piston moves with respect to time if the initialization dive is not performed and instead the float's salinity and temperature estimates are initialized with only a rough estimate of the expected values for the whole of the Gulf of Finland and the time of year. In that case, it may take over 24 hours for the float's environmental estimates to reach a plateau in terms of accuracy.

The qualitative difference in environmental conditions may also been seen in Figure 6, which plots the cumulative probability of having reached the goal depth in a given number of piston moves.

The quality of the temperature and salinity estimates was also directly tracked by periodically comparing the estimates for temperature and salinity at specific depths against the actual values. The error in depth that would result if the float were to dive to those depths was also determined by calculating an estimate for the density at each depth and finding the matching depth.

As Figures 6 and 7 below show, the temperature and salinity estimates appear to be sufficiently accurate, providing estimates which allow the float to reach its goal efficiently. However, these estimates may need to be improved to take into account the curvature of their relationship with depth, which introduces a small bias that may be seen in Figure 7c.

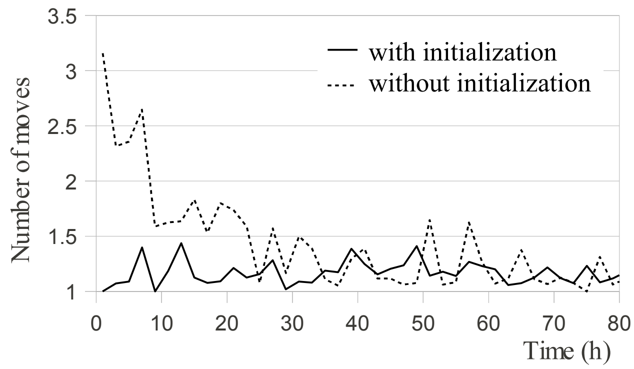

Figure 5: Mean number of piston moves vs. time for depths of at least 8m, with time discretized to bins of 2h

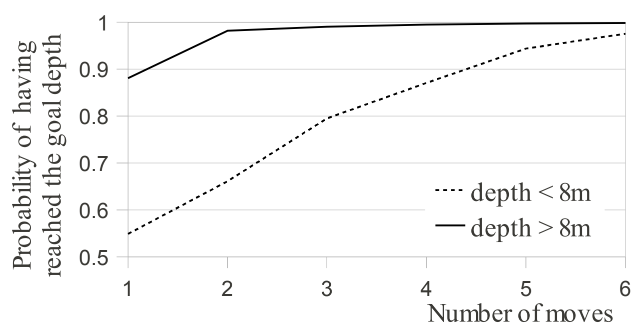

Figure 6: Probability of requiring at most N piston moves to reach a goal depth. For depths greater than 8m, in 98.2% of cases at most one correction is needed.

6. Conclusion

A SWARM float has to operate in a difficult environment using very limited sensors and actuators. With the methods presented here, the groundwork is laid for much more complex operations and missions; the ability to reliably go where you intend to go is a prerequisite for most things that an autonomous robot might be asked to do.

Regarding the methods presented, work remains primarily on two fronts: First, at-sea tests need to be carried out to verify the behaviour of the simulator as well as to test the diving control directly. Second, the methods need to be expanded to take into account external sources of data, such as other floats sharing the same mission. In particular, the initialization dive will need to take into account the simultaneous availability of more than one float.

The diving methods presented here rely on very accurate and precise control of the float's volume, which may not be possible with a different volume control method such as an oil pump system. On the other hand, diving in shallow waters as such requires very accurate control of volume.

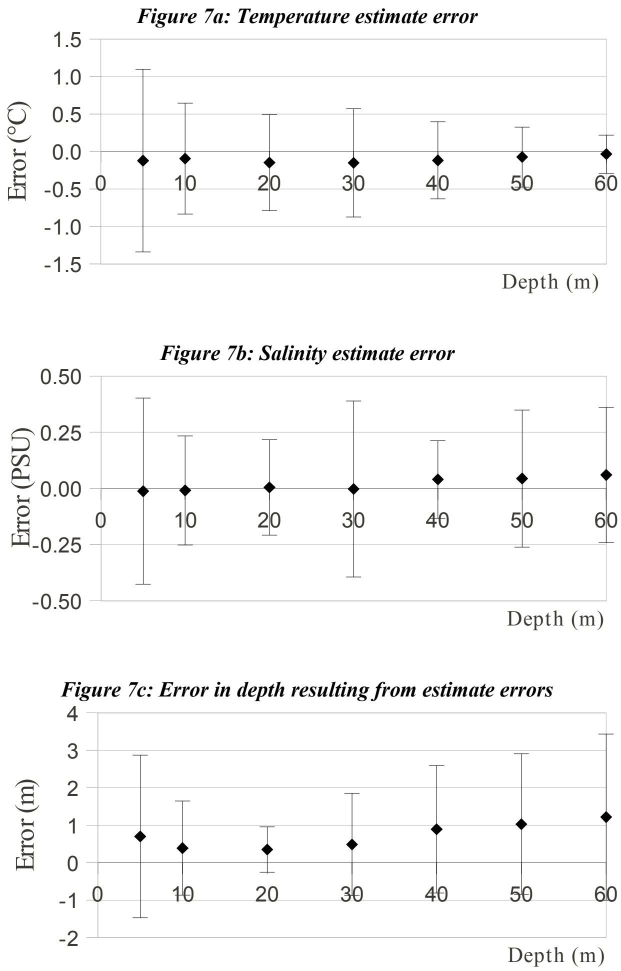

Figure 7: Errors in (a) temperature and (b) salinity estimates and the errors' standard deviations, measured when reaching a goal depth. The resulting error (c) in terms of depth and its standard deviation is also shown. These errors indicate the divergence between then estimated and actual values.

Acknowledgements

We would like to thank the Academy of Finland for funding the Finnish Centre of Excellence in Generic Intelligent Machines, the SWARM consortium, and Tapani Stipa and Antti Westerlund at the Finnish Meteorological Institute.

References

[1] Argo Science Team, *On the Design and Implementation of Argo* (Melbourne, Australia: GODAE International Project Office, 1998).

[2] R. Costanza, R. d'Arge, R. de Groot, S. Farberk, M. Grasso, B. Hannon, K. Limburg, S. Naeem, R.V. O'Neill, J. Paruelo, R.G. Raskin, P. Sutton & M. van den Belt, The value of the world's ecosystem services and natural capital, *Ecological Economics, 25*(1), 1998, 3–15.

[3] M. Vainio, A. Halme, I. Troshin, T. Stipa, J. Seppälä, F. Pollehne, E. Bauerfeind, H. Haardt, P. Brault, N. Seube, A. Smerdon, S. Caine, B. Swale & A. Hakala, Autonomous underwater multiprobe system for coastal area/shallow water monitoring (SWARM), *Proc. EurOCEAN 2004*, Galway, Ireland, 2004.

[4] N.P. Fofonoff and R.C. Millard Jr., Algorithms for computation of fundamental properties of seawater, *UNESCO Technical papers in marine science, 44*, 1983.

[5] Zengcai Qu, *Intelligent diving control of Lagrangian type of underwater robot* (Espoo, Finland: Helsinki University of Technology, 2009).

[6] Clause C. Leroy & François Parthiot, Depth-pressure relationships in the oceans and seas, *J. Acoust. Soc. Am. 103*(3), 1998, 1346-1352.

[7] Teppo Pirttioja, *Applying agent technology to constructing flexible monitoring systems in process automation* (Espoo, Finland: Helsinki University of Technology, 2008).

[8] Tapani Stipa (ed.), Short-term effects of nutrient reductions in the North Sea and the Baltic Sea as seen by an ensemble of numerical models, *Meri 49*, 2003.