SEA TRIAL VERIFICATION FOR A TERRAIN NAVIGATION METHOD

Mr. Filip Traugott
Defence Material Administration

SE-115 88 Stockholm
filip.traugott@fmv.se

Abstract

It is often important for an autonomous underwater vehicle (AUV) to have a good position fix. The basic idea of this method is to take a footprint of the bottom and correlate it with a bottom map. This terrain navigation method has the advantage of not needing to surface to get the position fix.

The method gives a very good position fix with only a few pings using a multibeam sonar. This has been confirmed after evaluation from a sea trial in October 2002. It has shown that the method is also very robust. Under the sea trial a multibeam sonar from Reson (Reson 8101) has been used.

Under the year 2002 has a hardware correlator been developed to show the possibility of fasten up the calculations and widen the data input. In an ordinary personal computer a calculation to get the result takes about 30 seconds, the hardware correlator decreased the time to about a half second.

The hardware correlator is designed in a FPGA using VHDL coding. When using a FPGA parallelism can be used, this means that the FPGA can make more than one calculation at the time.

The paper will discuss how robust the method is in different types of terrains, what kind of accuracy this method can deliver.

It will also discuss the FPGA technique, what are the advantages and disadvantages. What has been and can be done with this technique.

I. Introduction

This paper will describe the terrain correlation method and what the Swedish Defence Material Administration has been working with this last year in this issue.

In section II the background of the terrain navigation and the AUV project is described.

In section III the terrain navigation method is described, advantages and disadvantages is discussed.

In section IV the navigation correlation method implemented in hardware is discussed.

In section V the seal trial in October 2002 is presented.

II. Background

The Swedish Defence Materiel Administration has developed an Autonomous Underwater Vehicle (AUV) for demonstrations regarding underwater applications. This AUV is called AUV 62F and is based on prototype parts from the development of the Swedish torpedo 62 (export version torpedo 2000).

Figure 1: The AUV 62F with the pole up.

The vehicle is designed to handle different types of payloads. To achieve this it is constructed with interchangeable modules.

To make it possible to communicate wireless and to get a GPS position it has a pole with a radio transducer and a GPS receiver as seen in Figure 1.

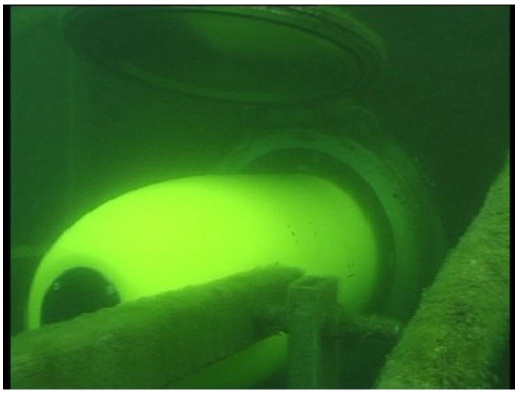

Figure 2: The AUV 62F launched from a stationary tube.

The AUV 62F is designed to swim out from a submarine and can be launched from an underwater tube. In Figure 2 the AUV 62F is just about to swim out from a underwater tube in the lake Vättern in Sweden.

Some demonstrations that are planed in the near future are terrain navigation update, bottom mapping, mine detection, swimming out from a submarine and docking with a submarine.

In many of the future military operations with AUV:s it will be a fundamental necessity for the AUV to have the ability to stay covert and still be able to navigate with high precision. This makes the GPS unusable because it requires the vehicle to operate near the surface.

There are some different methods to get a position fix without increasing the detection risk.

The demands for this type of system are:

- Low probability of revealing.
- Long time between sending out energy in the water.
- Ability to get a position fix with a stationary vehicle.

Because the vehicle want to stay covert it is very important to make the probability of revelation as low as possible. One way of achieving the second demand is to have a INS with low error growth so the vehicle can operate covert without sending out energy in the water. The last demand is mainly for submarine use, where the submarine sometimes has to stay still and quiet for a long time.

The traditional way to get an accurate position with terrain navigation is to update the navigation system with sonar measurements of the distance to the bottom continuously, see Figure 3.

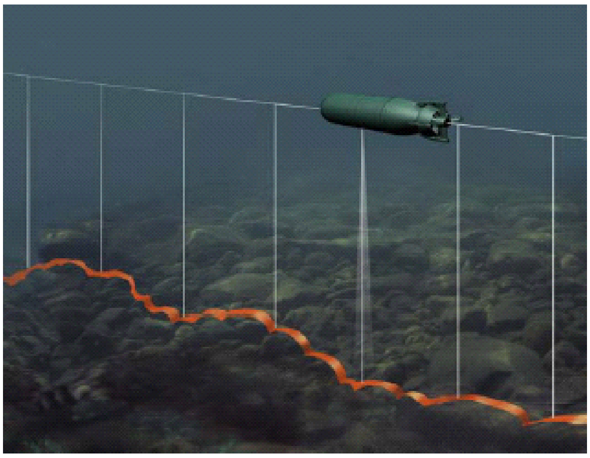

Figure 3: Navigation with traditionally recursive calculations.

The problem with this method is in quiet operations the sonar sends out to much energy in the water. The method also requires maps of the whole operation area.

Another method, in this paper called "the correlation method", uses small areas (navigation cells) within the operation area to update the position, see Figure 5. Therefore maps are needed just for those areas. This method is very calculation demanding, which makes an original computer to lame. To solve this, other techniques are looked into.

III. The terrain correlation method

The vehicle is equipped with a sonar for topographic measurements of the bottom. The measurements from the sonar can be put together to a footprint of the bottom as shown in Figure 4.

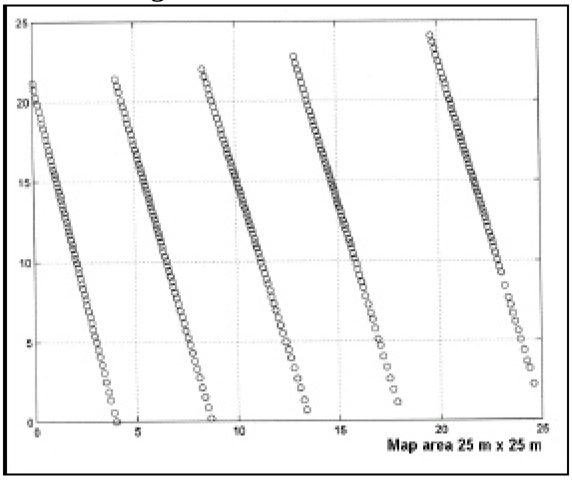

Figure 4: The footprint of the sonar data from five pings.

This method is based on comparison between a bottom map and footprint data from the bottom under the vehicle.

When the vehicle has been operating for some hours the INS has an insufficient accuracy and has to be updated. The vehicle moves to a positioning update cell were there are high-resolution topographical maps of the bottom. Depending of the positioning error in the INS the bottom map has to cover a sufficient area around the estimated position from the INS. Then the vehicle makes a footprint of the bottom under it with multibeam sonar.

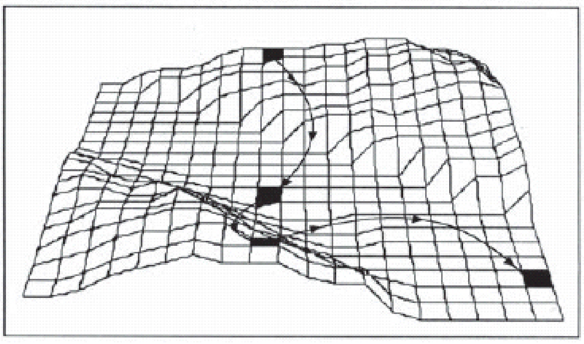

Figure 5: Bottom map that shows the navigations cells.

This footprint is then griddled in the same grid and orientation as the bottom map. The footprint matrix is compared allover the map with equal sized matrix and the result matrix shows were you are. For a simple use of this method the equation is:

$$e = \frac{1}{M \cdot N} \sum_{m=1}^{M} \sum_{n=1}^{N} b_{m,n} \cdot \left| e_{m,n} \right| =$$

$$\frac{1}{M \cdot N} \sum_{m=1}^{M} \sum_{n=1}^{N} b_{m,n} \cdot \left| \left(x_{m,n} - \bar{x} \right) - \left(y_{m,n} - \bar{y} \right) \right|$$

where $\{x\}$ is the sonar matrix and $\bar{x}$ is the mean value of it. $\{y\}$ is a part from the map and $\bar{y}$ is the mean value of it.

To compute the correlation, the absolute sum of the differences between the sonar matrix and the corresponding sized part of the map is calculated. The sonar matrix should glide over the whole map where the absolute sum is calculated, see Figure 6. The area with the lowest sum then becomes the likely position.

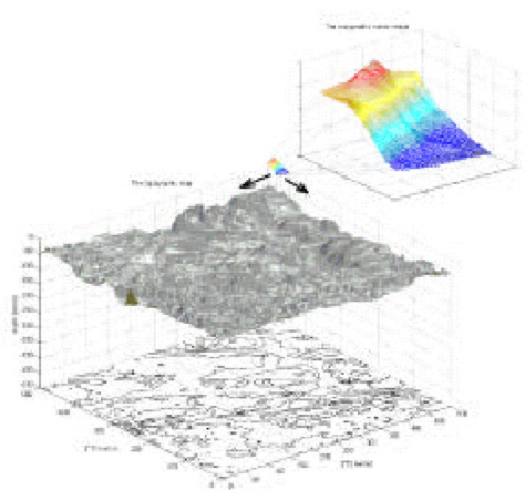

Figure 6: The principle of the correlation equation.

If the data is prepared before the calculation by subtracting the mean value of the calculating part the method gets depth independent. This means that it dos not matter what the sea level is for the moment. It is only the depth from the vehicle to the bottom that is important; the distance to the surface is not relevant provided that the sonar can get a sufficient footprint.

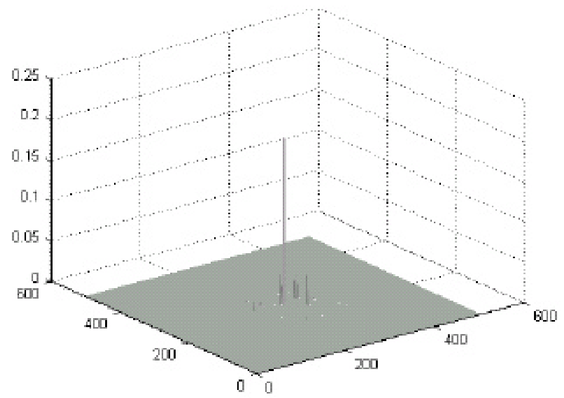

Figure 7: A typical result from the terrain correlation method.

When the result should be presented it is better to invert the result, which will make the best position shown in Figure 7 as a spike.

As mentioned in the introduction the calculation time for this process takes quite a long time. A typical calculation example is with a bottom map 501 times 501 points and a bottom footprint that has 10 times 10 points. The footprint should glide over the entire map and every time it moves one point a new correlation is made. One correlation sums 100 subtractions. This makes over 24 million subtractions to get the result. That amount of calculations takes to much time in a regular PC, at 700MHz laptop, it took about half a minute to do the calculations. One other problem with a standard computer is that it takes different time every time depending on what the computer is doing for the moment.

This problem can be solved using digital hardware technique.

IV. The hardware design

To get the robustness and reduce of the calculation time digital technique is recommended. This test board shown in Figure 9 that the Swedish Defence Materiel Administration has been developed is based on so called FPGA technique. A FPGA is a programmable digital circuit. It is reprogrammable so it is useful for developing new products because trial and error testing can be done simple and rapidly.

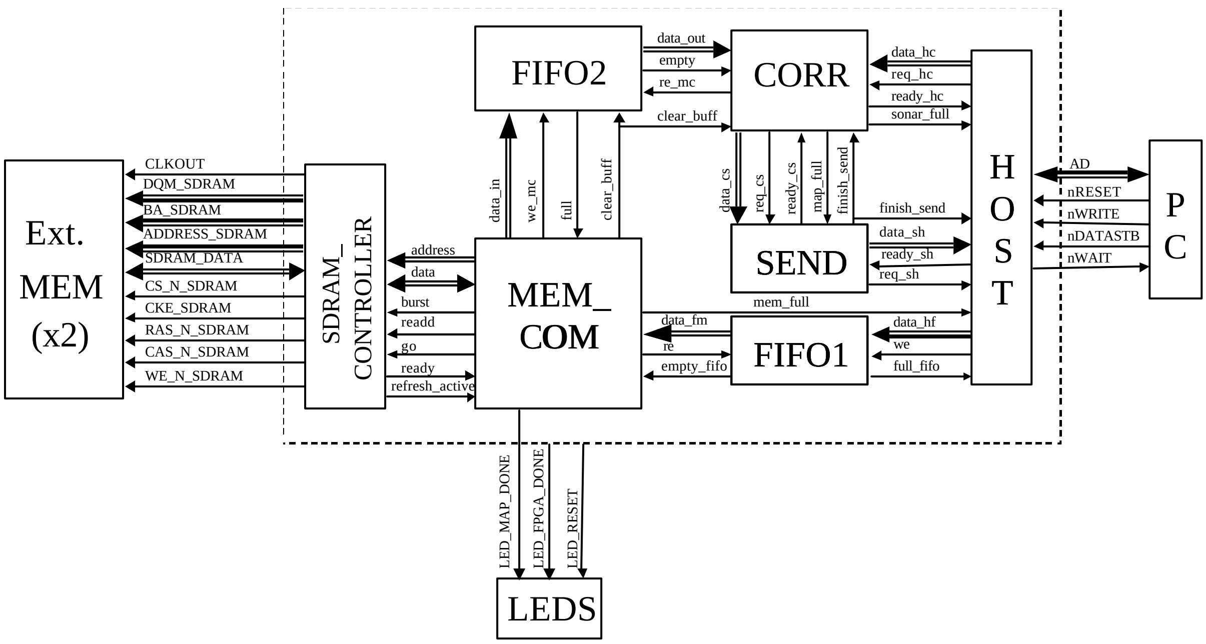

Figure 8: Block schedule.

The FPGA is synchronised with an external clock and it is possible to predict the exact time for every operation, which is very useful for this application.

The result for this application is calculated over fifty times faster than a regular PC. The FPGA is working with several tasks parallel which the computer is unable to do (it works sequential). This makes the FPGA very fast and for this application it can make several correlations at the same time.

The hardware programming is written in the language VHDL which is the second lowest abstract level when programming digital circuits.

The FPGA program is built-up in blocks that executes when it is trigged to do so. It does not have to wait for another block to be finish because of the parallelism.

Figure 8 shows the built-up blocks of the FPGA and the adjacent external blocks.

Figure 9: The circuit board with the FPGA on it.

The data that must be stored on the test board is the map with the size of 501 times 501 data pixels and the mean value data for the map that is 492 times 492 data pixels. In this version the data is stored in a special way so it takes shorter time to upload a new correlation area to the FPGA. All data is expressed with 2 bytes. This makes the data amount to over 5,3 Mbyte. The FPGA has internal memory but because of the big amount data that is stored an external memory is used for that and the internal memory is used for other applications in the program as shown in Figure 8 as FIFO1 and FIFO2.

The basics for the function is that the FPGA gets bottom topographic map data from a map data base and sends it direct to the external memory were the data is stored. Now the FPGA is waiting for the external PC to give a commando to correlate a footprint. The footprint is downloaded to the FPGA and in the block “CORR” the calculations is made. The result is sent back to the computer that can present the result shown in Figure 10.

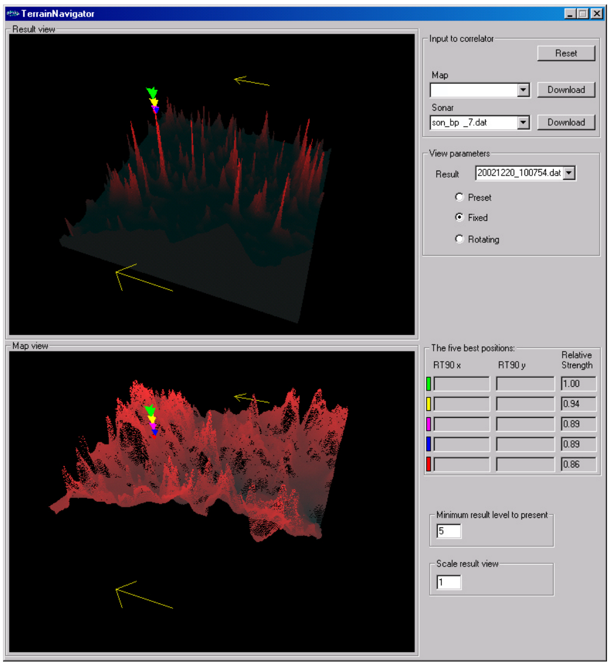

Figure 10: A print of the result GUI.

This test board is developed to test the functionality of the correlation method. Therefore a standard parallel communication has been used which is the biggest bottleneck in this system.

This test board can for the moment only take one size of the bottom map (501 times 501) and the footprint (10 times 10). It takes the FPGA 55 clock cycles to do one correlation area. This means that if the clock is on 50 MHz (an standard clock speed) the correlation will take 1.1 *m* s. The calculations for a whole map take 266ms, which is quite sufficient. For another application the time can be greatly reduce with newer FPGAs and modifications of the program structure.

V. The Sea Trial

In October 2002 the Swedish Defence Materiel Administration made a sea trial to evaluate the terrain correlation method.

The sea trial took place in the Swedish archipelago near by the island Utö. The trial platform was the twin keel ship Ocean Surveyor shown in Figure 11.

Figure 11: The platform ship "Ocean Surveyor".

The sea trial rote was planned as a possible AUV mission with position update areas (navigation cells) at appropriate distances in areas with different types of bottom structures, this to make comparison of what results the different types of terrain structure gives.

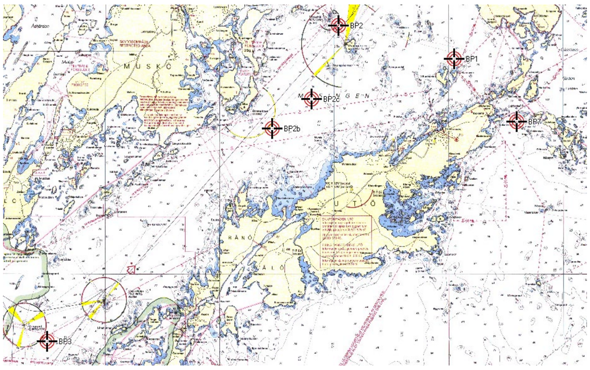

Figure 12: Sea chart over with the waypoints market.

The sonar used under the sea trials was the bathymetric sonar Reson 8101 shown in Figure 13.

The sonar was placed on a five-meter long pole and mounted in the water three meter below the surface in a moon pool on the ship. The trial was made while sailing with a speed of 4 knots over the waypoints five times per waypoint to get some statistics.

Figure 13: The bathymetric sonar Reson 8101.

The result from the sea trial was very encouraging and it was found out that even where the bottom structure is flat it is possible to get accurate results.

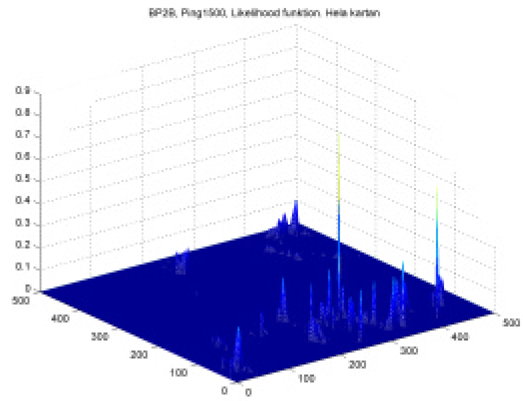

Figure 14: Correlation result with a spike in the correct possition.

In most of the update areas the error between the terrain correlation method and the DGPS were less than 10 meters. The error is when using maps with 10-meter grid, by interpolating the map down to 1 meter the error could be reduced sufficient.

In Figure 14 a result of a correct result is shown. There are some small spikes but one is sufficiently higher than the other. In Figure 15 the correct result is zoomed in.

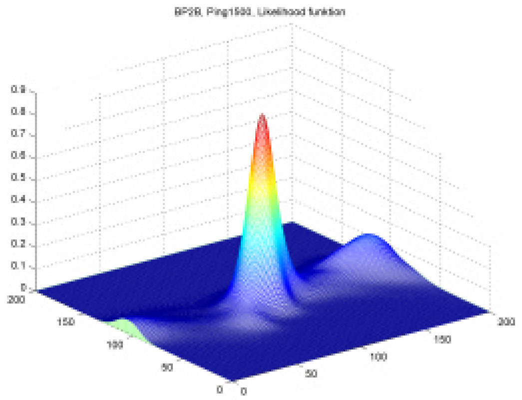

Figure 15: Zoomed correlation result from Figure 14.

In only one update area the terrain correlation method presented the wrong position shown in Figure 16, the error were about 1000 meter. This where probably a combination of very flat bottom and wave forming errors in the original bottom map. This could be caused by over steered beams from the sonar in the map-making process.

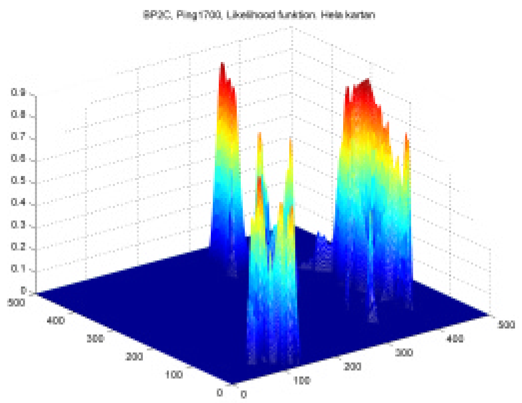

Figure 16: Correlation result with the higest top 1000 meter from the correct position.

VI. Conclusion

It has been shown that it is possible to make correlations of sonar measurements and topographic maps for terrain navigation applications with sufficient speed using a FPGA circuit. This enables the possibility to update the position of an AUV in real time in a similar fashion as with a GPS but with the big difference that is not necessary to surface. The prerequisites are however that the AUV is equipped with a multibeam sonar capable of measuring the bottom topography and that there are topographic maps with sufficient resolution (10 m grid or better) available.

VII. References

[1] Ingemar Nygren, A method for Terrain Positioning of an AUV, Conference proceedings, UDT 2003.

[2] Kim Andersson and Filip Traugott, Implementation of a terrain navigation correlator for an underwater vehicle in a FPGA, Västerås 2002, MDH-IEL-EX-2002:511-E.