A Simulation Environment for Autonomous Underwater Vehicles

Hans-Ulrich Kobialka, Thorsten Wunderlich
Fraunhofer Institut Intelligente Analyse- und Informations Systeme (IAIS)
Schloss Birtlinghoven, D-53754 Sankt Augustin, Germany

Abstract: **This paper gives an overview of the *AUV-Framework*, a simulation environment for the evaluation of concepts regarding AUVs. An infrastructure of functionalities (maps, interface to kinematics, communication between multiple AUVs) supports the development of autonomous control.**

I. INTRODUCTION

The development of Autonomous Underwater Vehicles (AUVs) has to consider the interplay between the vehicle (its kinematics and the sonars used) and the environment. This interplay constrains the abilities of an AUV regarding, for example, obstacle avoidance or exploration tasks. Unfortunately there is a vast amount of situations which the AUV may face in different environments. So, because experiments with real AUVs are expensive and time-consuming, technical concepts for AUVs have to be evaluated using simulators.

As part of a long termed research activity funded by the Bundeswehr Technical Centre for Ships and Naval Weapons (WTD 71), Fraunhofer IAIS has developed a simulation environment, called *AUV-Framework*, which can be used for the following purposes.

- Development of autonomous behaviours, in particular, obstacle avoidance, exploration of unknown areas, and cooperation between multiple AUVs.
- Investigation of how the characteristics of hardware (i.e. the kinematics of the vehicle, the aperture of the sonars) influence the performance of an AUV.
- Simulation of a particular mission right before it should be performed by a real AUV. Simulation helps to anticipate possible outcomes and to adjust mission parameters.

The *AUV Framework* targets these requirements by

1. Realistic 3-dimensional environments generated from S-57 sea charts
2. Simulation of side scan and multi-beam sonars.
3. Filtering and aggregation of sensor information in maps.
4. Interfaces to incorporate models of the kinematics and dynamics of a particular vehicle.
5. Means for simulation of multiple AUVs exploring an environment and communicating with each other.

Based on these functionalities, a particular behaviour repertoire has been developed: an implementation of VFH* obstacle avoidance approach [2], and a concept for generating partial plans for exploration of unknown areas [1].

II. GENERATION OF 3-DIMENSIONAL ENVIRONMENTS FROM S-57 SEA CHARTS

S-57 is the electronic data transfer standard prepared by the International Hydrographic Organization (IHO) committee. It defines the vector format of Electronic Navigational Charts (ENC). S-57 ENC charts are available for most maritime areas.

The *AUV-Framework* provides means for generating 3-dimensional environments from S-57 sea charts. There is a tool chain which first transforms an S-57 chart into a TIN file. TIN stands for “Triangulated Irregular Network” and is a format which models 3-dimensional surfaces by connected triangles. The generated TIN file is transformed into OBJ format where the depth of a triangle is color-coded.

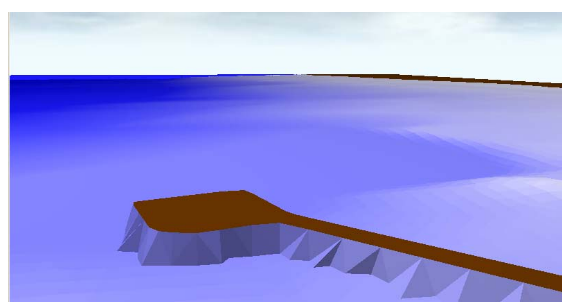

Figure 1: Environment automatically generated from an S-57 sea chart.

It is possible to modify the model (e.g. in case the resolution of a particular sea chart is too coarse) and to introduce additional objects (e.g. ships, mines). A model (OBJ format) can also be created from scratch using 3D tools like, e.g. Milkshape (http://chumbalum.swissquake.ch).

Figure 2: Another environment generated from an S-57 sea chart.

III. Specification of Currents

An AUV has to compensate the effects of currents. The *AUV-Framework* allows the specification of a current map, i.e. velocity of current for each location.

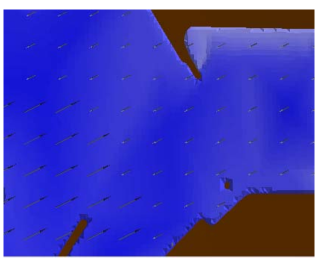

Figure 2: Currents at different locations

IV. Simulation of Sonars

An AUV typically perceives its environment using a forward looking sonar (for detecting obstacles in front of the AUV), a side scan sonar (for exploration of the sea floor), and an echo sounder.

A forward looking sonar is a multi-beam sonar, i.e., a sonar array sending beams at certain polar angles. Multi-beam sonars can also be used looking downwards for scanning the bathymetry of the sea floor.

A side scan sonar emits sonar pulses and records the intensity and the time-of-flight of the responses. There is no angular information about a response, but the resolution of the signal is larger compared to a multi-beam sonar.

The functionality of multi-beam sonars, side scan sonars and echo sounders can be simulated within the *AUV-Framework*. It uses the efficient collision detection of the JBullet game engine (http://jbullet.advel.cz/).

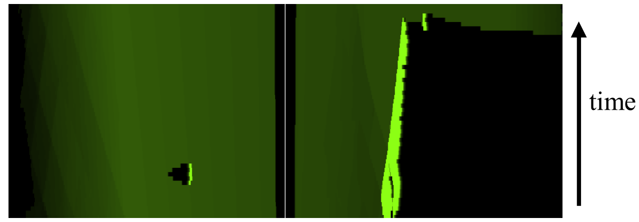

Figure 3: 'Waterfall' display of simulated side scan sonar data

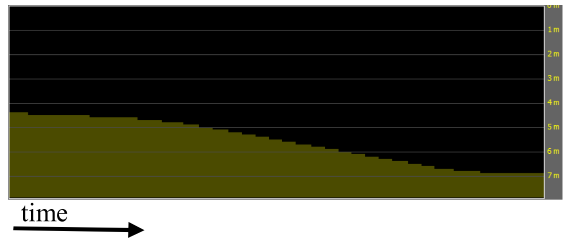

Figure 4: Output of a simulated echo sounder

V. Aggregation of Sensor Information in Maps

Sonar information is aggregated in two maps, the occupancy grid map, and the 'explored area' map.

The *occupancy grid map* collects the measurements of the forward looking sonar and the echo sounder. It is used for obstacle avoidance.

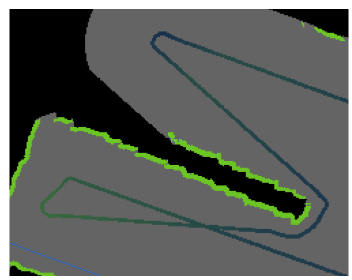

Figure 5: Occupancy grid. Objects are color-coded according to their depth. Areas, where the forward looking sonar didn't detect any obstacles, are grey.

The '*explored area' map* takes the information of the side scan sonar, and projects it to the sea floor. This determines the area of the sea floor which has been explored by the sonar. Sonar data is only considered while the AUV is driving straight (otherwise the data is not that useful). This map is used for planning trajectories which should explore unknown areas.

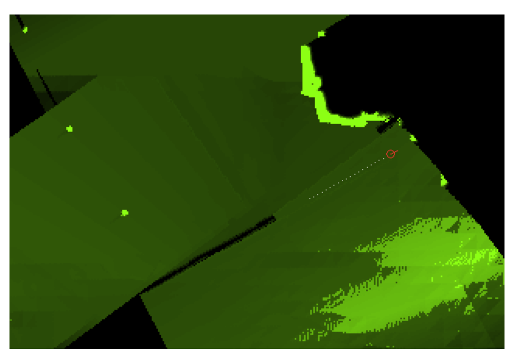

Figure 6: Map of area explored by a side scan sonar

Instead of a side scan sonar, a multi beam sonar can be used for exploring the sea floor.

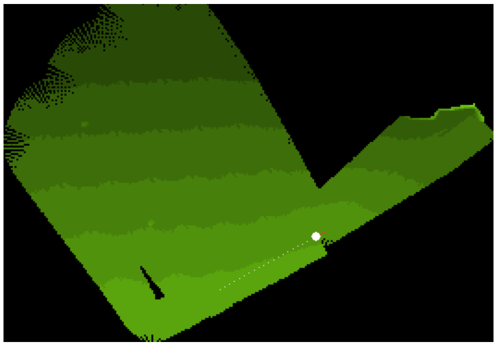

Figure 7: Map of area explored by a multi beam sonar

VI. Models of Vehicle Kinematics and Dynamics

For each simulated AUV, vehicle kinematics and dynamics can be defined. As many vehicle models are defined in Matlab/Simulink, the *AUV-Framework* offers an interface to Matlab.

VII. Simulation of Multiple AUVs

The *AUV-Framework* has a distributed client-server architecture where multiple AUVs (the clients) can simultaneously operate in an environment (provided by the server).

AUVs can communicate with each other within a limited range and exchange information, for instance, about how to cooperatively explore an area.

For instance, AUVs may broadcast the area explored by them. This way, an AUV can avoid exploring areas already explored by others (figure 8). Only area information is sent, not the sonar data, because of the low communication bandwidth in water. Also the reach of underwater communication is limited. Therefore the *AUV-Framework* permits communication between AUVs only if they are within a certain range.

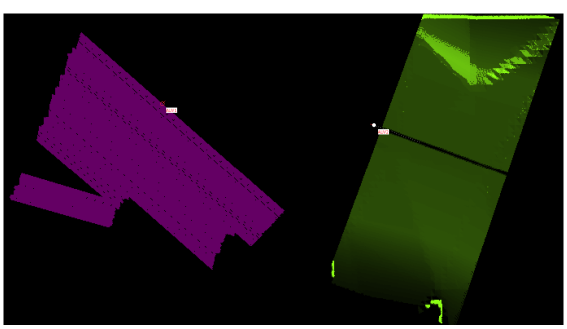

Figure 8: Cooperating AUVs communicate the area which they explore. Areas explored by remote AUVs are displayed as violet.

VIII. Autonomous Control

A. *Obstacle Avoidance*

In order to cope with unexpected obstacles, an implementation of VFH* obstacle avoidance approach [2] is provided. This is a reactive obstacle avoidance mechanism located in the 'Reactive Control' level in figure 9. Of course, any other obstacle avoidance algorithm can be integrated instead.

B. *Missions planned offline*

The *AUV-Framework* offers to execute missions defined using either SeeTrack© or the Maridan Operation Interface (MOI). These missions are defined as a list of waypoints which should be approached at certain speeds and depths.

C. *Autonomous exploration of unknown areas*

For the exploration of (partially) unknown areas, the use of pre-defined waypoints may not be flexible enough. Thus, instead of defining a list of waypoints, the user just defines the area to be explored. The mission is executed by a generative planner which generates *partial plans*.

A partial plan is generated using the knowledge available at the current point of time. This knowledge about the environment is often incomplete. Also the planning could not be exhaustive because plans should be generated in real-time. Therefore heuristics are used to produce partial plans being so called *good-enough solutions*.

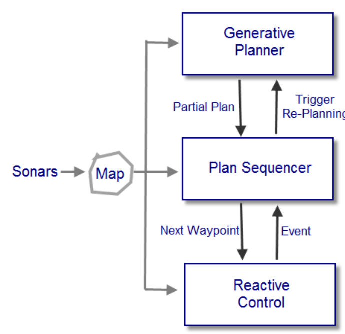

Figure 9: Control architecture implementing autonomy

The generative planner uses multiple heuristics implemented by modules called *experts*. There is an expert which generates trajectories parallel to previously driven ones. Exploring an area by driving parallel straight trajectories is a well-known heuristic used for AUVs.

In case obstacles appear, or parallel trajectories would not explore any significant unknown areas, another expert module is asked to propose random trajectories. These trajectories are not completely random as they consider obstacles and the area explored so far.

A-priori knowledge or offline plans can be incorporated by an expert module proposing trajectories defined by the user at the beginning of the mission.

A cost function is used to evaluate the partial plans proposed by the experts. The cost function balances the distance to be travelled and the area which could be explored by performing the particular partial plan. Then, the partial plan having lowest costs is chosen and executed [1].

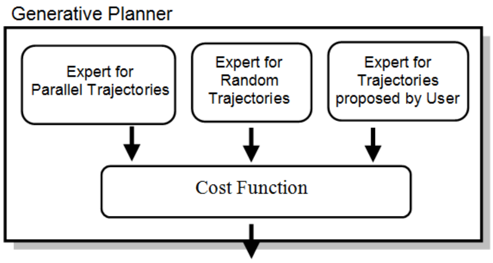

Figure 10: Partial plans are generated by different expert modules

The plan sequencer executes the partial plan either until it is completed, or until it is canceled, e.g., because of obstacles. Then the generative planner is triggered to produce a new partial plan (see figure 9).

This is only an example of how the *AUV-Framework* can be used to develop autonomous concepts. Other approaches could and should be implemented and evaluated using the *AUV-Framework* in the future.

IX. Conclusions

This paper gives an overview of the *AUV-Framework*, a simulation environment for the evaluation of concepts regarding AUVs. An infrastructure of functionalities (maps, interface to kinematics, communication between AUVs) supports the development of autonomous control.

Currently we are developing mission planning and coordination strategies for multiple AUVs jointly exploring large areas. These concepts will be validated using the *AUV-Framework*.

Acknowledgment

As part of a long termed research activity, the development of the *AUV-Framework* has been funded by the Bundeswehr Technical Centre for Ships and Naval Weapons (WTD 71).

References

[1] Kobialka, H-U., and W. Nowak (2007): High-Level Autonomy for Autonomous Underwater Vehicles; Vukic, Zoran (Ed.) et al.: CAMS '07: IFAC Conference on Control Applications in Marine Systems, September 19-21, 2007.

[2] Ulrich, I., and J. Borenstein (2000): VFH: Local Obstacle Avoidance with Look-Ahead Verification; 2000 IEEE International Conference on Robotics and Automation, San Francisco, CA, April 24-28, 2000, pp. 2505-2511*