

Orientation and Depth Control in Rippling Water for an Autonomous Underwater Robot

R. Dóczy^{1,2}, B. Takács², B. Sütő^{1,2}, T. Haidegger², M. Kozlovsky¹, József K. Tar²

¹ BioTech Research Center, Obuda University, Budapest, Hungary

`roland.doczi@irob.uni-obuda.hu,`

`suto.balazs@biotech.uni-obuda.hu,`

`kozlovsky.miklos@nik.uni-obuda.hu`

² Antal Bejczy Center for Intelligent Robotics, Obuda University, Budapest, Hungary

`{bence.takacs, haidegger}@irob.uni-obuda.hu,`

`tar.jozsef@nik.uni-obuda.hu`

Abstract. This paper describes an algorithmic solution for simple and efficient underwater orientation and depth control. Maintaining a position with an underwater robot is a difficult task. In the case of an Autonomous Underwater Vehicles (AUVs), not only the underwater conditions, but also the environmental effects off the surface need to be considered. There are a large number of algorithms have been designed by researchers based on computer vision, sensor fusion, etc. to estimate the location precisely, yet most of them are specific for the given hardware. Our solution employs a multi-sensor fusion based algorithm, where the data is taken from magnetic and pressure sensors. A PID controller was designed and implemented to ensure proper orientation keeping and depth control in rippling water. The solution has been tested in various environments, and successfully used during the marine challenges of the euRathlon 2015 competition.

Keywords: autonomous underwater vehicle, AUV depth control, response robotics, underwater navigation

1 Introduction

One of the main challenges of (underwater) robotics is how an autonomous robot can adapt to the environment, identify and localize themselves. For example, an underwater current or drift can easily relocate an AUV in the water. To design a solution for position keeping problems, one should reckon with not only the underwater conditions, but also the interactions above the water. Potentially useful sensory input includes any measurable data indicating the stream, the wind speed or drifts. With a proper sensor-fusion algorithm and adjoin process control, an AUV is able to keep its position and depth. Our aim was to design a controller that can stabilize a specific AUV, even under extreme conditions. There are numerous types of sensors, such as gyroscope, camera, accelerometer, pressure sensor etc. that can be employed for underwater applications.

2 Control problems

There are several ways to control an underwater vehicle [1]. During a field mission, an AUV would encounter interactions, such as drifts, buoyancy forces or streams. Many of these are unknown, with non-linear [2] effects, so they cannot be included directly in the motion equations. However, these forces take effect on the AUV, and influence its motion. Magnetic sensors are susceptible to external magnetic fields, such as the one generated by the thrusters. Furthermore, these magnetic fields can interfere with the communication link between other electrical components. To avoid this, the electrical parts—which are able to disrupt the communication during a mission—must be insulated from other parts. AUVs are specifically crowded devices (regarding their interiors), so electrical measurement systems should be shielded and designed properly to keep the measurement error between known intervals and minimize them as much as possible.

Measurements, simulations or approximation methods should be used in order to add the unknown forces with non-linear effects to the motions equations. One of the most popular controller systems used by industry and academia is the historical PID controller, which can be adequate for this solution as well, where a proper depth and orientation keeping is presented with error minimization regarding the measurement errors of the sensors.

2.1 PID control

The PID controller is a closed loop controller. It consists of three main elements: P (proportional), I (integral) and D (derivative). A PID controller deals with cumulated errors, which are derived from the difference of the system's measured output value from a predefined setpoint. It is a linear controller; however, the underwater systems are non-linear because of the non-linear effects that can influence them. For example, the buoyancy effect is always acting back to the AUV, but on the other hand, there could be different kind of underwater streams which are not always linear. In order to design a linear controller for a non-linear system, a feedback linearization approach can be used. In this case, the input or the output of the controller system should be transformed to make it linear. A common approach to solve this is feedback linearization, or the use of adaptive process controllers.

All three PID parameters have a significant influence on the system. With a high P value, the control system will be fast, but on the other hand, an extremely high parameter selection will lead to instability or oscillation. With an I term added, a faster error elimination can be reached, but in the meanwhile, there will be larger overshoots. A larger D parameter will decrease the overshoot, otherwise slows the response of the system.

Both underwater depth and orientation keeping should be described as a linear system.

2.2 Adaptive process control

During an underwater mission in rippling water, an AUV can become under- or over actuated. If the parameters of a control system are configured well for one particular environment, it may not work properly in a different one. This means that if a depth control problem is solved in calm water, there is only a small chance that it will work in rippling water as well. Furthermore, the salinity of the freshwater and sea water is different, which can cause larger resistance and a different buoyancy effect to the AUV. In order to implement a controller which is able to deal with the rippling water, an adaptive process controller or a feedback linearization method can be a suitable approach. The *adaptivity* means that the P , I or D parameters are changing relatively to the error during runtime. In case of the depth keeping the mentioned value is the output of the pressure sensor, and as for the orientation keeping, that is the output of the compass sensor.

The adaptive process control is a good approach to control a non-linear system with a linear PID controller. It keeps changing the parameters of the controller relatively to a function. This function can be a much more complex method, such as a quadratic- or a cube root function, or a fuzzy system. A simple approach is to use a quadratic ramp function for adjusting the control parameters. The input of the process controller should be separated into smaller sections after the measuring, so a single value can represent a group of values. With a quadratic function a process controlling can be achieved, wherein the output will be larger if the system is far from the set-point, otherwise it will be close to zero. This can minimize the disturbing effect of the streams, which can influence the position keeping of an AUV and occurs randomly.

2.3 Underwater position keeping

Position keeping is a process, which is used to enable an underwater vehicle to keep orientation (separated to roll, pitch, yaw) and depth, based on sensors and actuators. The position keeping can be separated into orientation and depth keeping. Both of them are supposed to be kept relative to a baseline. The base value can be the magnetic North (for orientation) and the output of the pressure sensor (for depth), or an initiated value which is set when the system starts, such as an Inertial Measurement Unit (IMU). When employing an IMU, the system must reckon with the inaccuracy of the measurement, because usually these values should be filtered.

A compass sensor should be used in order to solve the orientation keeping problem in one degree of freedom, furthermore, a gyroscope module is needed in order to compensate the error during the measurement. On the other hand, in the case of the depth keeping, a pressure sensor can be used—combined with another sensor—to increase its effectiveness. A variety of algorithms can be used to filter the acquired signals for more precise measurement, such as an adaptive, median filter, extended Kalman filter [8].

3 AUV system components

Nowadays, modularity is in the focus of robot control solutions. This means that the source code of a program is not only one large code or a state machine. In a modular software, functions which deal with a different problem (for example, during collecting data, usually a filter need to be applied that needs several functions) should formulate a different component. For example, measuring, controlling, or filtering functions should be in different software components.

3.1 Software

The Robot Operating System (ROS) [4] is a widely known, modular, open source framework, with a message passing interface. The main elements of the ROS are the nodes. A node is a program, which communicates through a ROS master component with other nodes. The ROS master gives the parameter server, the naming and the registration services to the nodes. The nodes are able to communicate in two ways; through a topic or through a service. A topic is a broadcast message (open communication for nodes that are in same ROS network), the service is a point-to-point connection (secured communication for alarm, and secured data exchanging). A node needs to be subscribed for a topic in order to listen to its communications. If a node would like to publish to a topic, it should not be subscribed to it. For example if a node needs pressure sensor data, it should subscribe for a topic that provides this data by a publisher node. The publisher node releases the measured values, and other parameters, such as a time stamp, ID etc., to the topic. With this solution, separated communication interfaces can be defined. However, there is no security solutions implemented for the publishing or subscribing process. This means that each node can subscribe or publish to a topic without any authentication or authorization process.

One of the key features of ROS is that it supports several programming languages (Java, C++, Python, LUA, etc.), and because of the message passing interface, different language codes can smoothly communicate with each other. There are available tools for testing, or visualizing measurement values, for example the *rqt_plot* for 2D data visualization, or the *rViz* for 3D visualization [7]. The ROS proved to be a proper framework to implement and test codes, furthermore, the available hardware uses ROS interface too.

3.2 Hardware

The hardware that was available for testing these functions was a Sparus II AUV [3] which was developed by the University of Girona Center for Maritime Research (CMRE). It is a torpedo-shaped AUV, with a mass of 52 kg and a length of 160 cm. Because of its shape, it has proper hydrodynamic characteristic, and is ideal for distant, underwater operations. There are 3 thrusters mounted on it: one of them is on the center of gravitation of the hull (can be used for depth control), and two others at the end of the hull lengthwise (for movement). Because of its length and the two lengthwise thrusters, the Sparus II is not able to turn in one place that could be a prob-

lem during operation in narrow places. To support the on-board computations, there the Sparus II is equipped with an Intel i7 processor, 4 GB of RAM and a 250 GB SSD storage, and is equipped with a 1.5 kWh battery (providing up to 8 hours autonomous navigation). Ubuntu 14.04 Trusty Tahr is installed on the device with ROS hydro distribution. There is another software element developed by the manufacturers and installed on the Sparus II; the COLA2 (Component Oriented Layer-based Architecture for Autonomy) framework, which is developed as a part of ROS, and provides implemented functions to get data from the built-on sensors and manipulate the actuators or configure them.

Due to the location of the thrusters, the Sparus II cannot be controlled in pitch and roll axes. The passive ballast is located at the bottom of the AUV, not far from the center of gravity. This means that it is not allowed to control the roll direction. The manual balancing of the hull was needed before launching, so its effect on the pitch could be mitigated.

4 Structure of the solution

To solve the orientation and depth keeping problem, the built-in compass and pressure sensors were used. These sensors are validated by the manufacturer of the Sparus II AUV. There are predefined COLA2-based topics in the ROS environment, where data from these sensors are published. The orientation and the depth keeping can be separated and behaving independently from each other, and each can be handled as 1-1 degree of freedom (DOF). To tune the P , I and D controller parameters properly, the Ziegler–Nichols method was used. This is a classical, empirical PID design method where the step response of the system is not necessary to be known. The steps are the following: convert the PID controller into a simple P controller by eliminating the I and D parameters, or setting them to 0, then turn the K_c (P) parameter up until the response of the system is in a sustained periodic oscillation in the output. Store the K_u (gain) value, the P_u (period time of the oscillation). Then adjust the I and D parameters as the Ziegler–Nichols tuning chart recommends (Table 1).

P	I	D
$K_u / 1.7$	$P_u / 2$	$P_u / 8$

Table 1: Ziegler–Nichols Tuning Chart

We tested this controller in a computer simulation environment, with satisfying outcome, as presented in Figure 2.,

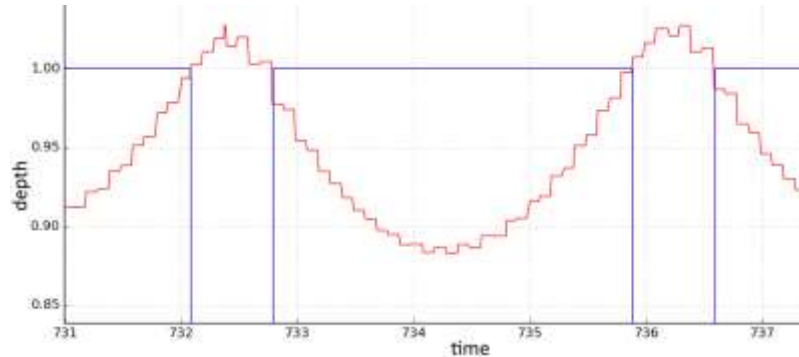


Figure 2. Measuring the K_u and P_u values in the oscillated system. The red lines show the actual depth of the Sparus II AUV, and the blue represents the control value. The horizontal axis is the elapsed time in seconds, and the vertical axis is the actual depth in meters.

To solve for an adaptive controller, the I and D parameters should be changed. With modifying the I and D values, the controller can be fine-tuned. The windup can be a problem during tuning the control parameters. This issue occurs when the output of the control system cannot influence the controlled characteristic. This can occur when there is a large positive change in the setpoint. To avoid this, the setpoint should be increased along a ramp function, or the integral term should be disabled until the actual state of the system is not in a controllable region.

4.1 Orientation keeping

To keep the orientation, the Sparus' built-in compass sensor combined with a PID controller was used. The input of the controller was the measured orientation value from the compass sensor, with regard to the magnetic North. The output of the control system should be a value between -1 and 1, because the COLA2 provides a topic where the control can be solved, and it accepts only a value between the aforementioned intervals. Depending on the lateral direction of the target regarding to the AUV, the thrusters can be controlled with opposite signs of the control system output. This means that if the setpoint is defined as -90° , so it is to the right from the hull, the output will be sent to the left thruster, and the negated value of the output will be the input of the right thruster.

4.2 Depth keeping

In the case of depth keeping, the built-in pressure sensor and an adaptive PID controller were used. The input of the controller is the depth value, which is calculated from the pressure sensor and provided by the COLA2 framework through a ROS topic. The output of the control system is a value between -1 and 1 as it was mentioned in section 4.1.

5 Experiments

The methods have been tested in four different test environments. Initial experiments were done in software simulation (provided by the ROS and COLA2), then the hardware was tested in a pool (of limited size), followed by a freshwater lake test, while the final testing was conducted on the two marine sections of the euRathlon 2015 competition (www.eurathlon.eu).

5.1 Simulator

The simulation environment [5] was the UWsim (Underwater Simulator) scenario, provided by the COLA2 framework, and developed for testing AUV solutions and implementations. In this environment the parameters of the controller can be reconfigured, and tested before a field mission. The environment has adjustable hydrodynamic parameters. This means that for example, if an AUV has a movement speed and wants to do a 90° right turn, the AUV will drift during the turning process. The density, the drift or the stream of the water can all be configured before starting the simulator.

During the simulation, one of the built-in functions of ROS has been used to record measurement data for the offline process (rosvbag). In Figure 1, the adaptive PID control solution can be seen. In this case, the setpoint of the control system is 1.0.

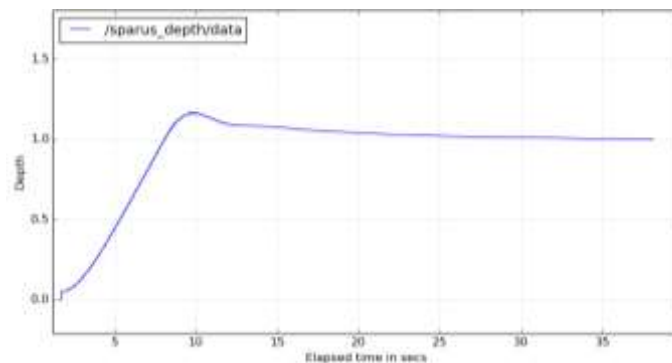


Figure 2. The adaptive PID depth control solution plotted in UWSIM with `rqt_plot`, when the setpoint was 1 m depth. The vertical axis is depth in meters and the horizontal axis is time in seconds.

5.2 Test scenario 1: Pool (freshwater)

A pool with rigid walls has been set for testing experiments. It has dimensions of 4 x 2 x 2 meters, so only the depth keeping could be tested. The pool was filled up with fresh water.

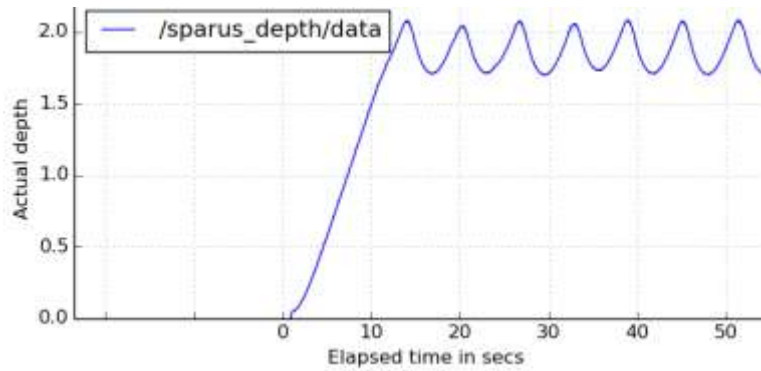


Figure 3. The adaptive PID controller for depth control in a pool, plotted in UWSIM with rqt_plot. The vertical axis is depth in meters and the horizontal axis is time in seconds.

5.3 Test scenario 2: Lake (freshwater)

The first field tests were conducted at a fresh water lake. During these, the measured speed of the wind was 2-3 km/h which means that there were 5-10 cm waves, creating drifts at the surface of the water. The experiments with the depth keeping solution were similar to the pool tests.

5.4 Test scenario 3: Seawater

The ultimate environment was at the marine section of the euRathlon 2015 robotic competition, which was organized in shallow sea water. The parameters of the controller had to be reconfigured because of the density of the sea, and it was accounted by the adaptive PID controller [6], as presented in Figure 4.

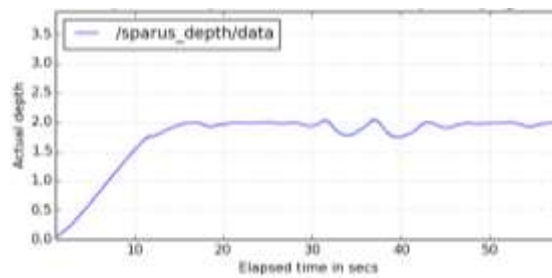


Figure 4. Adaptive PID based depth control solution measured in euRathlon 2015, plotted in UWSIM with rqt_plot.

6 Conclusions

This paper described a practical and low-cost solution for the problem of orientation- and depth keeping of an autonomous underwater vehicle. With an adaptive PID controller, the orientation- and depth keeping problems could be solved in a cheap and efficient way, under limited conditions. These solutions do not provide robustness, and are only meant for limited experiments. During the euRathlon competition entered with a Sparus II (Figure 5), we did not deal with the position keeping, because of the shape of the hull and the placement of the thrusters. The Sparus II is not able to turn in one place, so the Y direction (sideway) errors could not be corrected in a simple way.



Figure 5. Participating teams at the euRathlon 2015 competition with their Sparus II AUV units.

7 Acknowledgement

Authors would like to thank NATO Centre for Maritime Research and Organization (CMRE) for the opportunity to access and use the Sparus II AUV during the euRathlon competition, and also the friendly support of NIST and University of Girona (UdG). Financial support for this work was provided by the University Research and Innovation Center (EKIK) of Óbuda University.

References

1. ZHAO, Side; YUH, Junku. Experimental study on advanced underwater robot control. *Robotics, IEEE Transactions on*, 2005, 21(4), pp. 695–703.

2. SANTHAKUMAR, M.; ASOKAN, T. Non-linear adaptive control system for an underactuated autonomous underwater vehicle using dynamic state feedback. *Int. J. Recent Trend Eng*, 2009, 2, pp. 384–389.
3. CARRERAS, Marc, et al. Sparus II, design of a lightweight hovering AUV. In: *Martech 5th International Workshop on Marine Technology*. SARTI, 2013.
4. QUIGLEY, Morgan, et al. ROS: an open-source Robot Operating System. In: *ICRA workshop on open source software*. 2009. p. 5.
5. PRATS, Mario, et al. An open source tool for simulation and supervision of underwater intervention missions. In: *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on*. IEEE, 2012. pp. 2577–2582.
6. 3–30, 2016.
B. TAKACS, R. Doczi, B. Suto, J. Kallo, T. Varkonyi, T. Haidegger and M. Kozlovsky, "Extending AUV Response Robot Capabilities to Solve Standardized Test Methods," *Acta Polytechnica Hungarica,s* 13(1), pp. 157–170, 2016.
7. Available tools in Robot Operating System:
<http://wiki.ros.org/Tools>
8. Ahn, Kyoung Kwan, and Dinh Quang Truong. "Online tuning fuzzy PID controller using robust extended Kalman filter." *Journal of Process Control* 19(6) (2009): 1011–1023.