ROV AUTONOMIZATION – YAW IDENTIFICATION AND *AUTOMARINE* MODULE ARCHITECTURE

Marin Stipanov[1], Nikola Miskovic[2], Zoran Vukic[2] and Matko Barisic[2]
[1] Brodarski Institut, Ave. V. Holjevca 20, 10020 Zagreb, Croatia
[2] Faculty of Electrical Engineering and Computing, University of Zagreb, Unska 3, 10000 Zagreb, Croatia
E-mail: nikola.miskovic@fer.hr

Abstract: The Automarine Module is a simple cost effective method for transforming underwater remotely operated vehicles (ROVs) into autonomous underwater vehicles (AUVs), with minimum development time, and with no ROV circuit alteration. This paper presents architecture of autonomization of the VideoRay Pro II ROV together with its technical specifications. The paper also presents a procedure for open-loop identification of the nonlinear yaw model. Analytical expressions that are used for model identification are also provided.

Keywords: underwater vehicle, AUV, architecture, identification

1. INTRODUCTION

Underwater vehicles are characterized by complex 6 DOF which are coupled and highly nonlinear. This makes them difficult to control. Remotely operated vehicles (ROVs) are usually operated via a tether which serves as energy as well as a communication link with surface. The tether, however, presents a great disturbance for the vehicle, especially when the vehicle is operated at greater depths. This is one of the main reasons why autonomous underwater vehicles (AUVs) are slowly replacing tether-controlled ROVs. However, AUV systems require not only low-level control algorithms to be robust and fault-tolerant, but that trajectory and mission planning should be carefully designed in order to avoid catastrophic situations.

This paper presents the *Automarine Module* developed to transform the *VideoRay Pro* II ROV into an AUV, with minimal development cost and time, for underwater system control research purposes at the University of Zagreb, Laboratory for Underwater Systems and Technologies (LUST). The *Automarine Module* is a system enclosed in a waterproof hull which can be attached to the bottom of the submersible. It controls and powers the submersible over its standard communication/power socket (Fig. 1.). This way, simple switching between the autonomous and remote mode of operation is achieved, and, most importantly, the interior of the original submersible stays intact.

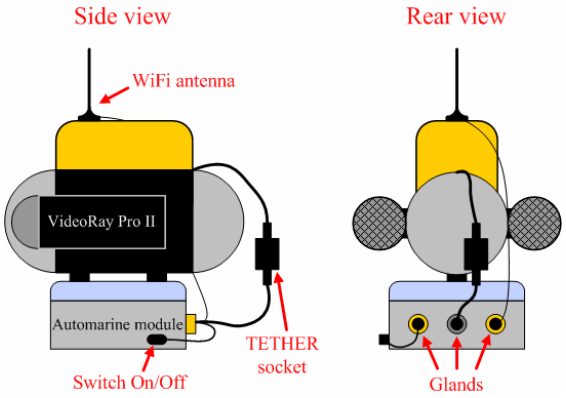

Fig. 1. The *Automarine Module* installation

The *Automarine Module* hardware setup and the nonlinear yaw dynamics identification procedure for the resulting AUV are presented in this paper. Caccia, M. et al. (1998), (2000) and Carreras, M. et al. (2003) describe underwater and surface marine systems and some methods for identifying nonlinear dynamics. An interesting approach for linear model identification of a low-speed underwater vehicle is given by Ridao, P. (2004). Identification of coupled model dynamics for an underwater vehicle is given by Miskovic et al. (2007). This paper will give an open-loop identification procedure which differs in determining the system's yaw inertia.

The paper is organized as follows: Section 2 describes the *Automarine Module* architecture, i.e. hardware components and communication with the surface. Section 3 gives the results of the vehicle's thruster mapping experiments and describes a procedure for the development of the nonlinear yaw dynamic model. Section 4 presents validation results for the identified model, and Section 5 concludes the paper.

2. AUTOMARINE MODULE ARCHITECTURE

2.1. Hardware

Hardware choice for the autonomization module is focused on upgradeability, modularity and ease of reprogramming. These demands arise from the underwater system research purpose.

For the module to be completely modular in design, it is necessary to use already developed components available in the market. This way, development time and cost is greatly reduced. For the module to be easily reprogrammable it is best to use an embedded computer which supports one of popular computer operating systems. This way, the use of popular programming languages (for example NI *LabVIEW* like in this occasion) is made possible in an easy and cost effective way, instead of using specially designed integrated circuits which would require a lot more development time and costly equipment. Since Universal Serial Bus (USB) is a fast serial bus and there is a vast variety of products with support for it, one of the embedded computers with support for USB is the best solution for the task. This kind of design will result in inferior power efficiency and bigger size of the designed module, in comparison with an integrated circuit version. This is a small price to pay since the resulting module will endorse about 1.5*h* of autonomy for the resulting AUV, which is enough for laboratory use.

A two wire differential Controller Area Network (CAN) is used for communication between the *VideoRay Pro II* submersible and its console. Therefore it is best to use a CAN to RS-232 converter, since an RS-232 port is available on the

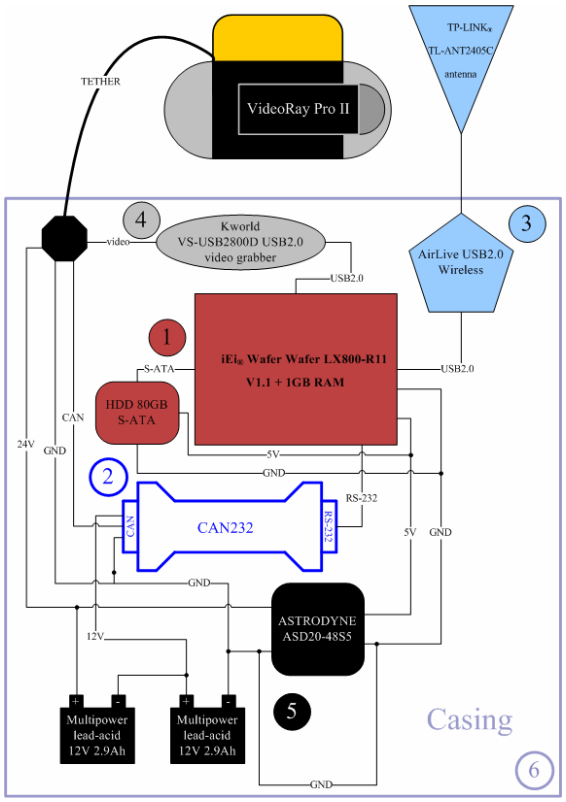

Fig. 4. *Automarine* Module wiring

chosen embedded computer. The CAN232 converter supports CAN bit rates up to 1Mbit/s, which is more than enough since the CAN bus installed on the *VideoRay Pro II* operates at a very unusual speed of 138.24kbit/s.

The chosen embedded computer is the Wafer LX-800 single board computer powered by a 500MHz low power processor which does not require active cooling. This is essential for the given purpose. Wafer LX-800 also supports a variety of standard busses and interfaces (USB, RS-232, LPT, PC-104, IDE…). The computer is equipped with 1GB of Random Access Memory (RAM) and 80GB Hard Disk Drive (HDD) to satisfy any given operation. Large memory and powerful processor support even most demanding operating systems which simplify software development and component communication.

An USB2.0 video grabber is used for digitalizing the video signal from the submersible (Phase
Alternating Line (PAL) format, see Fig. 3). The digitalized video is stored on the 80GB HDD.

The *VideoRay Pro II* submersible requires 48V for operation. This voltage is only used for lights and thrusters. The circuitry in the submersible is powered over a DC/DC switching power supply (input range 18~75V DC) integrated in the submersible, which lowers the input voltage to 5V. Therefore any input voltage above 18V keeps the submersible operational. Knowing that, there is no need to secure a 48V source in the *Automarine Module*. To obtain a 48V power source, it would take four 12V batteries, but instead it

is enough to secure 24V for the submersible (only two 12V batteries). This configuration uses much less space at the cost of submersible's thrust and lights intensity being 4 times weaker. That does not present a problem since the module is designed for laboratory testing. This way, the resulting *Automarine Module* is much smaller, since lead-acid 12V 2.9Ah batteries are used which are pretty big in size. These batteries were chosen due to their market availability. In an outdoor version of the *Automarine Module*, different batteries would be used, with a much better capacity/size ratio (like Li-Ion batteries). A separate DC/DC switching power supply with a 5V output is used in the *Automarine Module* for powering the installed hardware. Details on hardware components can be found in Stipanov, M. (2007) and references within.

2.2. Communication

As seen in Fig. 1, the user communicates with the *Automarine Module* over a Wireless Local Area Network (WLAN). For this purpose, a USB2.0 Wireless LAN adapter is installed on the chosen embedded computer. The USB2.0 WLAN adapter uses an external dipole antenna which is put on top of the submersible for communication.
Wireless communication is possible only when the AUV is surfaced. A server application continuously runs on the AUV itself, while a client application can be run from any surface computer. When the wireless connection to the server is detected, clients immediately start receiving data and sending control signals.

3. MODEL IDENTIFICATION

3.1. Thruster mapping

According to Fossen, T.I. (1994), the force exerted by a thruster can be modelled by a bilinear model

$$\tau = b_1 |n| n - b_2 |n| v \qquad (1)$$

where b_1 and b_2 are positive constants, and v is the vessels forward speed. Thruster model can be simplified if an affine model which has a form of $\tau = b|n|n$ is used. In this case, forward speed influence is neglected. This model is more appropriate in practical situations especially if submersibles are moving at low speed.
These thruster models are derived under the assumption that a thruster exerts exactly the same thrust while rotating in both directions. In most applications, this is not the case. Having this in mind, the following thruster model for small forwad speeds is proposed:

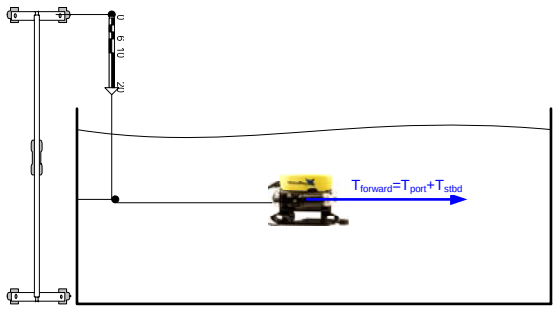

Fig. 5. Thruster mapping experiment.

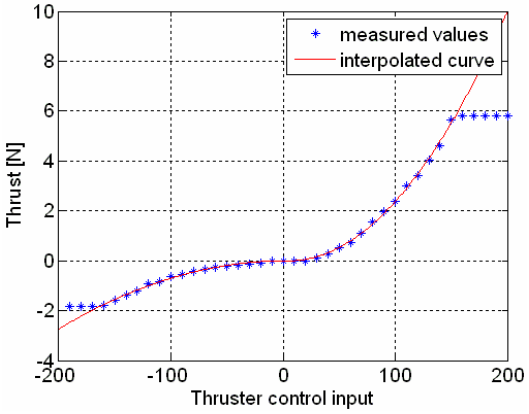

Fig. 6. Thruster mapping results.

$$\tau^i = \begin{cases} a_f |n| n, & n > 0 \\ a_b |n| n, & n < 0 \end{cases}, \qquad (2)$$

where subscripts f and b denote 'forward' and 'backward' respectively and superscript $i = \{port, stbd\}$.

The experiments for determining nonlinear static characteristic of the thrusters were conducted in the *Laboratory for Underwater Systems and Technologies* at the University of Zagreb, Faculty of Electrical Engineering and Computing, Department of Control and Computer Engineering. Both thrusters were excited with the same control signal, causing the submersible to perform heave motion. Dynamometers were connected to the vehicle, as shown in Fig. 5, in order to measure the overall exerted thrust. These experiments resulted in values obtained for overall heave thrust, as shown in Fig. 6. Dots present measured values, while the line shows the approximation of the measured values, using affine thruster model (2). Equation (3) is the model of both thrusters (performing heave motion) of the VideoRay Pro® micro ROV obtained by experiments.

$$\tau_X = \begin{cases} 2a_f |n| n = 5.233 \cdot 10^{-4} |n| n, & n > 0 \\ 2a_b |n| n = 1.328 \cdot 10^{-4} |n| n, & n < 0 \end{cases} \qquad (3)$$

These results can be used to determine the model of a single thruster, with an assumption that both thrusters exert the same thrust. In practice, it is rather difficult to measure individual thruster's exerted thrust, but the assumption of thruster identity enables us to perform this by using simple pool tests described above. The calculated individual thruster model can be described with (2) using identified model (3).

3.2. Identifying the yaw model

A general differtial equatin that describes the AUV yaw model can be written in a general form

$$I_r\dot{r} + D(r)r = \tau \qquad (4)$$

where I_r is the moment of inertia, and $D(r)$ is a general form of yaw drag.

Determining yaw drag. Usually, the term $D(r)$ is described by using three models:

- Model 1, $D(r) = k_r$ is used when a vehicle is moving at low speed, i.e. when hidrodynamic effects are negligable
- Model 2, $D(r) = k_{r|r|}|r| + k_r$ is a general form of presenting drag which is rarely used in practice due to its complexity
- Model 3, $D(r) = k_{r|r|}|r|$ is most common in practice since it describes the nonlinear effects in underwater vehicle motion due to hydrodynamic effects, and on the other hand it is rather simple, (Fossen, 1994 and Vukic et al., 2003).

The hydrodynamic drag can be determined by a series of experiments in steady state. The AUV is excited with normalized yaw moment in the range of [-1.2, 1.2] with a step of 0.1. For each of these moments, the heading response is recorded. Due to astatic characteristic of the system, the responses will be constantly rising.
In this experiment, special attention should be paid to whirlpool effects during the experiment. Namely, if the vehicle is rotating in place, after a full circle (or even before) the thrusters can cause turbulent circular rotation of water. This water movement will increase the vehicle's yaw speed hence giving false experimental results. The advice is to take into account only the responses of the first half of the circle movement.
For every yaw moment a couple of experiments were run. The data were used to interpolate the three proposed models. Table 1. gives model parameters values, along with the sum of square errors of the fitted curves.

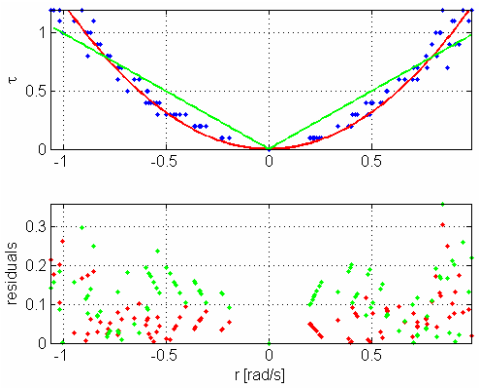

Fig. 7. Yaw drag identification results

As it can be seen, model 1 does not describe the given data in a satisfactory way. The difference between models 2 and 3 is very small therefore a simpler model 3 will be used.

Table 1 Identified yaw models' parameters

Model	*1*	*2*	*3*
$k_{r\|r\|}$	-	1.259	1.257
k_r	0.9961	0.02753	-
SSE	1.8908	0.8019	0.8353

The experimentally obtained static characteristic values are shown in Fig. 7 with blue dots. Green line presents the interpolated model 1, and blue dots give residuals for that model. Red lines and dots present interpolated model 3 and residuals respectively. It is clear that quadratic model gives better results.

Determining yaw inertia. The usual way of determining yaw inertia is performing zig-zag manoeuvres. This method is quite popular, but it is based on determining yaw inertia of the linearized model, when the system is excited in the vicinity of some static point.
The idea we are proposing in this paper uses the same experiments that were performed for determining yaw drag, but in this case transitional effects will be taken into account. The system is again excited with a constant moment, and due to dynamic properties of the system, the heading response will not be linearly increasing immediately, but will have a form shown in Fig. 8. ε_V from Fig 8. is the velocity error and can be defined as

$$\varepsilon_V = \lim_{t\to\infty}\left[\psi(t) - \dot{\psi}(t)t\right]. \qquad (5)$$

If the AUV yaw dynamic is assumed to be linear, equation $I_r\ddot{\psi} + k_r\dot{\psi} = \tau$ has the solution (6) with the assumption that the initial heading is ψ_0, and initial yaw speed $\dot{\psi}_0 = 0$, (Vukic et al. 2005).

$$\psi(t) = \tau\frac{I_r}{k_r^2}e^{-\frac{k_r}{I_r}t} + \frac{\tau}{k_r}t + \psi_0 - \frac{I_r}{k_r^2}\tau \qquad (6)$$

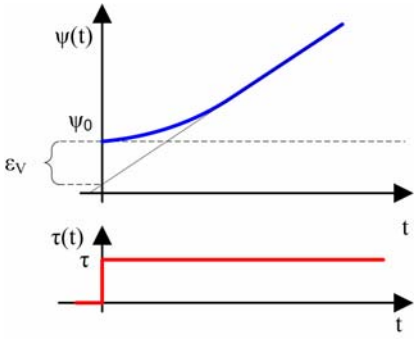

Fig. 8. Open-loop response with velocity error

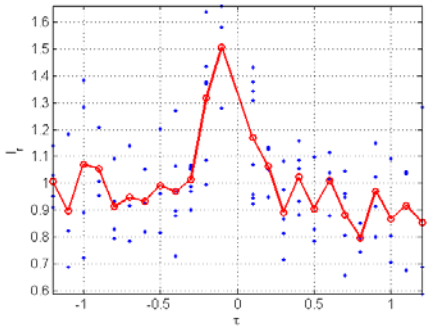

Fig. 9. Yaw inertia identification results.

Table 2 Identified yaw models' parameters

Model	*1*	*3*
$I_{\dot{r}}$	0.7058	1.018

It is clear that in steady state the first term disappears and the slope of the resulting response is $\frac{\tau}{k_r}$. The velocity error is then given with

$$\varepsilon_V = \lim_{t\to+\infty}\left[\psi - \frac{\tau}{k_r}t\right] = \psi_0 - \frac{I_{\dot{r}}}{k_r^2}\tau\,. \qquad (7)$$

In other words, by eliminating the initial heading, the heading response will be shifted from the line with slope $\frac{1}{k_r}$ by $\varepsilon_V = -\frac{I_r}{k_r^2}\tau$. Since applied yaw moment τ and yaw drag k_r are known, the system's yaw inertia can be determined.

If nonlinear dynamics are presumed, the situation is a bit more complex. The differential equation describing yaw dynamics is of the form

$$\begin{aligned} I_{\dot{r}}\dot{r} + k_{r|r|}r|r| &= \tau \\ \dot{\psi} &= r \end{aligned} \qquad (8)$$

where r is yaw speed. A somewhat more complex calculation gives the following velocity error

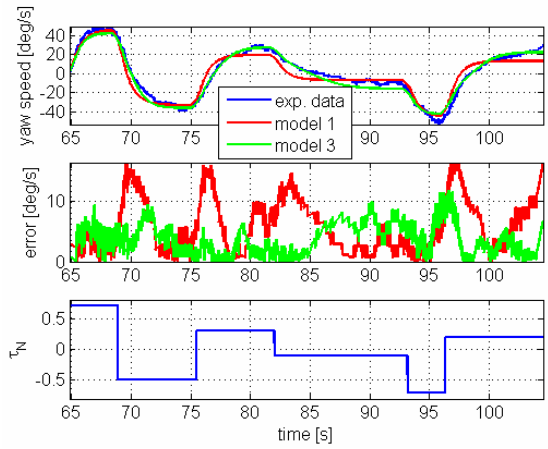

Fig. 10. Validation results

$$\begin{aligned} \varepsilon_v &= \lim_{t\to+\infty}\left(\psi - \operatorname{sgn}(\tau)\frac{I_{\dot{r}}}{k_{r|r|}}t\frac{\sqrt{k_{r|r|}|\tau|}}{I_{\dot{r}}}\right) = \\ &= -\operatorname{sgn}(\tau)\frac{I_{\dot{r}}}{k_{r|r|}}\ln 2 \end{aligned} \qquad (9)$$

It is obvious that equation (9) substantially differs from (7). Now, from the determined velocity error and yaw drag, the inertia moment can be determined for this nonlinear system. Since in was already concluded that linear drag describes the system better that the constant drag, equation (9) will be used for determining system's yaw inertia.

Fig. 9. gives results of the yaw inertia, using (9), for each experiment in blue dots, while red circles present mean values for each applied yaw moment. The overall average of the yaw inertia for the constant and linear drag case are shown in table 2.

3.3. Validation

The system's model was validated using a yaw moment sequence given in the third graph of Fig. 10. The figure also gives comparison between responses of model 1 (linear model) in red, and model 3 (nonlinear model) in green. Data obtained from the experiment are shown in blue colour in the first graph. It is obvious that simulation errors (second graph in Fig 10.) are much smaller for the case of model 3. It is also clear that yaw inertia identified for the nonlinear model gives much more accurate transient responses.

4. CONCLUSION

This paper demonstrated the procedure for the development of an autonomous underwater vehicle using some commercially available hardware components and a remotely operated vehicle. The

AUV is built like a server and communicates with a surface client via wireless connection. The main disadvantage of this system is its autonomy – this issue will be taken care of in future work.

In order to achieve proper vehicle control, identification of the vehicle dynamics should be performed. We used a classical open-loop method to perform numerous experiments, based on which the yaw drag was determined. Yaw inertia was determined using the same open-loop experiments, but transient responses were taken into account. The analytical derivation of the method for determining yaw inertia is given. Linear model (constant drag) was also identified, for the purpose of comparing it with the nonlinear one (linear drag). The validation proved that nonlinear model describes the system better, and that the method for determining yaw inertia can be applied in practice. The main drawback of open-loop identification methods is that significant number of experiments should be performed in order to obtain reliable results.
The future work will be based on using the same methods to determine mathematical models in surge and heave directions. In addition to that, nonlinear controllers for all three motions will be developed on the basis of the identified models.

ACKNOWLEDGMENT

The work was carried out in the framework of the research project "*RoboMarSec* – Underwater robotics in sub-sea protection and maritime security" supported by the Ministry of Science, Education and Sport of the Republic of Croatia (Project No.036-0362975-2999).

REFERENCES

Caccia, M., Casalino, G., Cristi, R. and Veruggio, G. (1998). Acoustic motion estimation and control for an unmanned underwater vehicle in a structured environment, *Control Engineering Practice* **6**, 661-670.

Caccia, M., Indiveri, G. and Veruggio, G. (2000). Modelling and identification of open-frame variable configuration underwater vehicles, *IEEE Journal of Ocean Engineering* **25**(2), 227-240.

Carreras, M., Ridao, P., Garcia R. and Nicosevici, T. (2003). Vision-based localization of an underwater robot in a structured environment, *IEEE International conference on robotics and automation, ICRA'03*, Taipei, Taiwan

Fossen, T.I. (1994). *Guidance and Control of Ocean Vehicles*.

Ljung, L. (1999). *System Identification – Theory for the user*, 2nd ed., Prentice Hall.

Miskovic, N., Vukic, Z., Barisic, M. and Tovornik, B. (2006). Autotuning autopilots for micro-ROVs, *Proceedings of the 14th Mediterranean Conference on Control and Applications*, Ancona, Italy.

Miskovic, N., Vukic, Z. and Barisic, M. (2007). Identification of coupled mathematical models for underwater vehicles, *Proceedings of the OCEANS'07 Conference*, Aberdeen, Scotland.

Ridao, P., Tiano, A., El-Fakdi, A., Carreras, M., and Zirilli, A. (2004). On the identification of non-linear models of unmanned underwater vehicles, *Control Engineering Practice* **12**, 1483-1499.

Stipanov, M. (2007). *Autonomization of the VideoRay Pro II remotely operated submersible*, master thesis, University of Zagreb, Faculty of Electrical Engineering and Computing, (in Croatian).

Vukic, Z., Kuljaca, Lj., Donlagic, D. and Tesnjak, S. (2003). *Nonlinear control systems*. Marcel Dekker. New York.

Vukic, Z. and Kuljaca, Lj. (2005). *Automatic control – analysis of linear systems*. Kigen. Zagreb (in Croatian)