Sensors & Transducers, Vol. 153, Issue 6, June 2013, pp. 45-52

Sensors & Transducers

Development of a Remotely Operated Vehicle Test-bed

Biao WANG, Chao WU, Tong GE
State Key Laboratory of Ocean Engineering, Shanghai Jiao Tong University,
Shanghai, 200240, China
Tel.: +86 02134207187, fax: +86 02134207187
E-mail: biaowang@sjtu.edu.cn

Received: 19 April 2013 /Accepted: 14 June 2013 /Published: 25 June 2013

Abstract: This paper presents the development of a remotely operated vehicle (ROV), designed to serve as a convenient, cost-effective platform for research and experimental validation of hardware, sensors and control algorithms. Both of the mechanical and control system design are introduced. The vehicle with a dimension 0.65 m long, 0.45 m wide has been designed to have a frame structure for modification of mounted devices and thruster allocation. For control system, STM32 based MCU boards specially designed for this project, are used as core processing boards. And an open source, modular, flexible software is developed. Experiment results demonstrate the effectiveness of the test-bed. *Copyright © 2013 IFSA.*

Keywords: ROV, Test-bed, Small-scale, STM32, uC/OS-II.

1. Introduction

A large portion of the earth's surface is covered by the ocean. As the ocean attracts great attention on environmental issues and resources as well as scientific and military tasks, the need for and use of underwater vehicles has become more apparent [1-3]. Many underwater operations that were once carried out by divers can now be carried out more quickly, more efficiently and in a more repeatable fashion using underwater vehicles. Further, underwater vehicles are not subject to the biological and environmental limitations of humans and marine mammals, such as depth, cold water duration, warm water duration, currents, pollutants and viscosity of medium.

Several unmanned underwater vehicles (UUVs) have been developed in our laboratory to satisfy various demands such as survey, science mission and nuclear inspection. A lot of research and experiments must be done on vehicle subsystems before the vehicle can act as a complete system. Independent components or modules can be developed and tested on another platform. At the same time it takes a lot of time and resources to do field test on large scale vehicles. Consequently a ROV test-bed is proposed. The goal is to enhance our ability to develop new underwater vehicles in the laboratory and rapidly test the components or modules. The design criteria of the vehicle were defined as follows:

- Small and light enough to handle experiments easily with a few researchers;
- Suited with typical on-board sensors;
- The low level control of vehicle is carried out with microcontroller;
- Easily re-programmed from outside;
- The frame structure is adopted for modification of mounted devices.

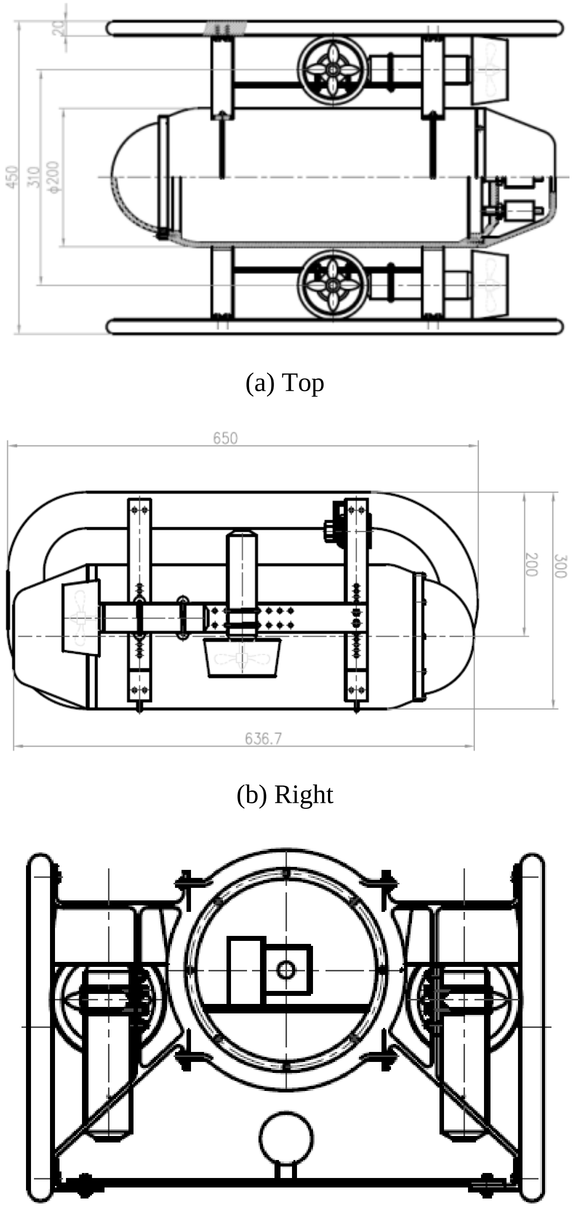
2. Vehicle Description

HIPPO is an observation class ROV, originally designed as a small, smart, inexpensive test bed for research in underwater technology. It can well perform normal inspection and observation tasks and can be transported, deployed, operated and recovered by a few researchers. Fig. 1 shows a photo of the developed vehicle. HIPPO has an overall length of 0.65 m and weight 22 kg with a maximum depth rating 100 m. The robot is designed to be passively stable in roll and pitch [4]. Specifications of the vehicle are shown in Table 1.

Fig. 1. A picture of the vehicle.

Table 1. Specifications of the ROV.

Parameter	Specification
Dimensions	(650×450×300) mm
Depth Rating	100 m
Weight	22 kg
Maximal Speed	1.5 knot
Power Supply	220 VAC, 600 W
Thrusters	4 total, 2 horizontals and 2 verticals
Sensors	Digital Compass Angular Velocity Sensor Pressure Sensor (Depth Sensor)
Video Cameras	1/4'Sony Exview CCD
Lighting	2x800 lumen LED lights, variable intensity



(a) Top

(b) Right

(c) Front

Fig. 2. Schematic design of the vehicle.

3. Mechanical Design

Fig. 2 shows the schematic design of HIPPO. The frame structure is adopted for modification of mounted devices and thruster configuration. A pressure hull provides the essential buoyancy and the watertight compartment for on-board electronics and sensors. The pressure hull is AA6061-T6 aluminum, designed to dive 100 meters depth using following equation [5].

$$P_k = E[\pi^4 / n^4 (n^2 - 1)(r/l)^4 + (n^2 - 1)/12(1 - v^2)(t/r)^2](t/r), \quad (1)$$

$$r = R + t/2, \quad (2)$$

The parameter P_k (=3 MPa, the safety factor is 3) is pressure of elastic bucking, E ($=6.93\times 10^4$) is the Young's coefficient, v (=0.33) is the Poisson's ratio, t is the thickness of hulls, l is the length of pressure hulls, r is obtained from equation (2), and R is radius of hulls. The length of pressure hulls becomes 0.42 m, the diameter 0.2 m and thickness 6.0 mm. Two caps are designed to complete the hull, and are attached to each end of the hull such that they reliably seal the hull. Sealing is achieved with commercially available o-rings. The front cap is designed as a fairing while the rear cap has six watertight connectors for electric connection. The caps also allow access to the interior for easy repair and maintenance in the field.

The propulsion system consists of four ducted propellers developed for this project. The propellers are driven by 24 VDC motors. Two horizontal thrusters are mounted aft of the vehicle to provide both forward and backward movement. Yaw is provided by operating the thrusters in opposite directions. Two vertical thrusters are mounted on left and right sides symmetrically of the vehicle. They make the vehicle's motion in heave and roll possible. The general arrangement of the vehicle will be given later.

4. Hardware Architecture

The overall architecture of the ROV control system is illustrated in Fig. 3.

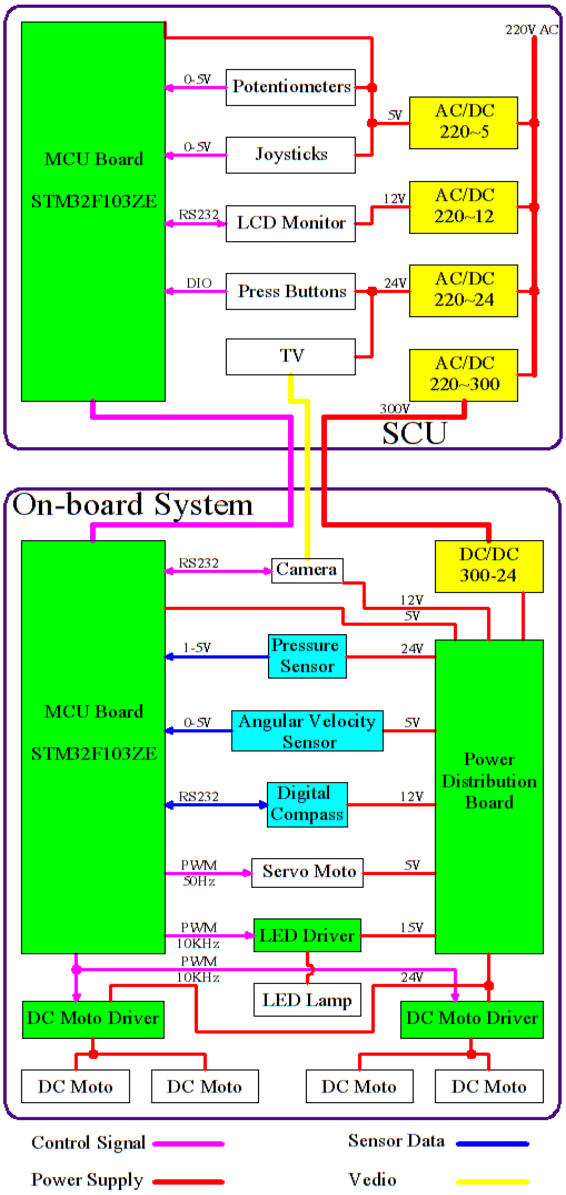


Fig. 3. Overall architecture of ROV control system.

External power supply for the whole system is 220 VAC, which makes it easy to power up the vehicle. AC/DC adapters are used to obtain the required DC voltage. Video signal is transmitted directly to the surface monitor through coaxial cable. Two MCU boards are used to deal with data acquisition, information processing and output calculating. They communicate with each other through Controller Area Network (CAN). The MCU board based on STM32F103ZE microcontroller has 8 12-bit ADC channels, 32 DIO channels, 8 PWM channels, 4 RS-232C interfaces and 1 CAN interface. It's schematic is illustrated in Fig. 4.

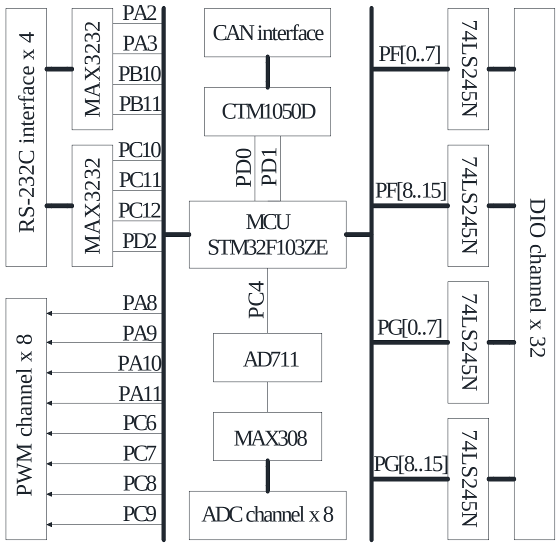
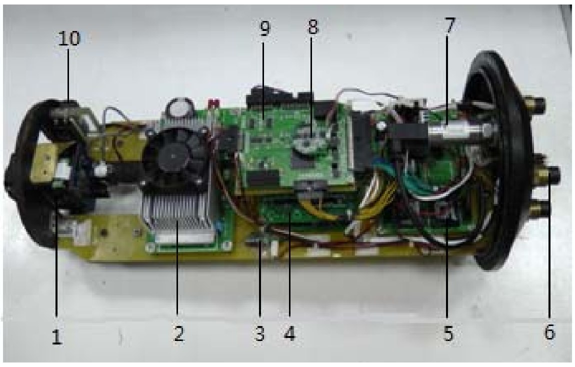


Fig. 4. Schematic of the MCU board.

The STM32 F1 is a series of mainstream MCUs covering the needs of a large variety of applications in the industrial, medical and consumer markets. High performance with first-class peripherals and low-power, low-voltage operation is paired with a high level of integration at accessible prices with a simple architecture and easy-to-use tools. The series consists of five product lines which are pin-to-pin, peripheral and software compatible. STM32F103ZE is a high-density performance line ARM-based 32-bit MCU with 256 to 512 KB Flash, USB, CAN, 11 timers, 3 ADCs, 13 communication interfaces. It works at 3.3 VDC, 72 MHz (maximum main frequency). The CTM1050 is a high speed (up to 1 Mbaud) CAN transceiver compatible with 3.3 V and 5 V devices. The device provides differential transmit capability to the bus and differential receive capability to the CAN controller, at least 110 nodes can be connected. It has 2500 V isolation voltage from pins 1-4 to 5-8 and allows for easy application by customer without external components.

4.1. Underwater System

Fig. 5 shows the arrangement of mounted devices. They are installed in the pressure hull. The motion sensors include a digital compass, an angular velocity sensor and a pressure sensor. Digital compass, LP3200, is mainly used for measuring heading. It can provide accurate heading (0.1º) measurement over a 0-360° tilt range. Its working voltage is 5 VDC, sampling frequency is 12 Hz, and a RS232 serial data port is used for connection to the MCU board. The angular velocity sensor, IDG300, produced by InvenSense, obtains angular velocity of roll and yaw, with 5 VDC power supply, 140 Hz update rate, ±500° sensitivity, 0-5 V analog output. With a range of 0-500 KPa, a working voltage of 18-26 VDC, the pressure sensor can output 4-20 mA current signal. All these sensor data are acquired by the underwater MCU.



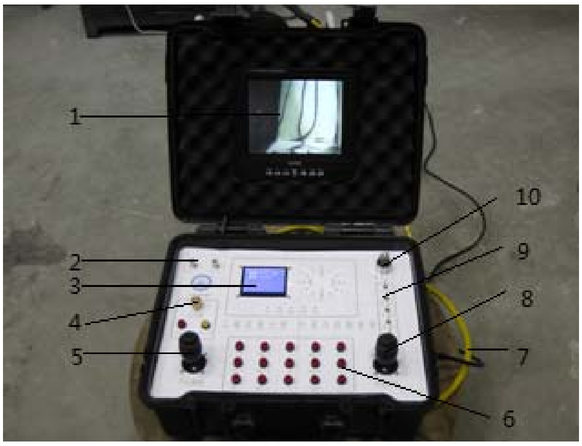
1-CCD camera; 2-DC-DC; 3-Angular velocity sensor; 4-DC motor driver; 5-Power distribution board; 6-Watertight connector; 7-Depth sensor; 8-Digital compass; 9-MCU board; 10-Servo motor.

Fig. 5. Underwater system.

One CCD camera is attached to a servo motor, allowing it to rotate in vertical plane. The camera is focused through RS232-C interface by MCU board. It can provide a good vision of underwater environment for the operator. LED lamps and thrusters specially designed for this project are attached to the vehicle frame as shown in Fig. 1. They are all driven by PWM signal at 10 kHz. The maximum output of a thruster is 30 W with a supply voltage of 24 V. Each DC motor driver has 2 PWM channels with total output power of 300 W. All of these actuators update their status and action according to surface commands.

4.2. Surface Control Unit

The SCU is responsible for operation of the whole system as shown in Fig. 6. Sensor data and underwater images are displayed on it in real time. Control commands are generated by joysticks and press buttons. Several buttons are spare for additions. A neutral tether connects SCU to the vehicle while providing data, video and power link between them.



1-Video monitor; 2-Video output; 3-LCD monitor; 4-LED tune; 5-Joystick(Vertical); 6-Press buttons; 7-Neutral tether; 8-Joystick(Horizontal); 9-Spare; 10-Power switch.

Fig. 6. Surface control unit.

5. Software Architecture

The software architecture has a modular, flexible structure for any modification or additions [6]. With the open architecture, it is possible to use any vendor's hardware. Each module can be controlled with minimum interface with others. Thus, almost all of the software modules can be easily rebuilt or replaced for upgrading or testing like hardware components. Each module follows its own specification to maintain the specific software/control architecture. The software architecture consists of three layers, device layer, user layer and application layer, as shown in Fig. 7.

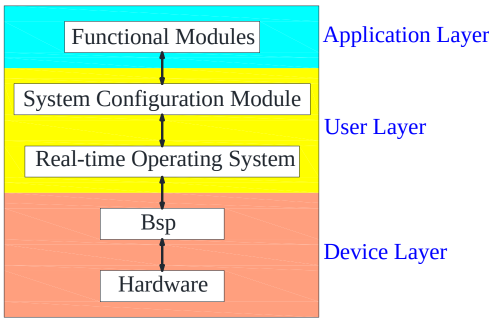


Fig. 7. Hierarchical control architecture for software.

The device layer is the only hardware dependent part. The board support package (BSP) implements specific support code for a given board that conforms to a given operating system. It is commonly built with a bootloader that contains the minimal device support to load the operating system and device drivers for all the devices on the board.

The user layer consists of a real-time operating system and a system configuration module. The main role of a system configuration module is providing a hardware independent interface for the application layer. A block diagram of the system configuration module is shown in Fig. 8. MainEntry is the entrance of the whole program. TargetInit finishes configuration of hardware resources such as DIO, ADC, CAN etc. ParaInit sets initial value of global variables used for data sharing between tasks. UsrTaskInit creats user tasks. WDInit sets software watchdog on.

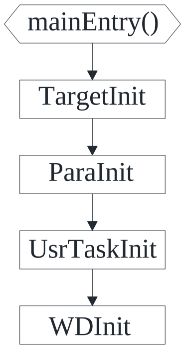


Fig. 8. Block diagram of the system configuration module.

The application layer is user-oriented and can be transplanted to other control systems without modification. It has no relation with hardware and operating system. The actual functional modules such as sensor data sampling, equipment control, actuator control etc. are embedded in different tasks of the application layer. The functional modules are different with respect to the specific application.

For implementation of the software architecture, uC/OS-II real-time operating system is embedded in the STM32 MCU. uC/OS-II is a portable, ROMable, scalable, preemptive, real-time deterministic multitasking kernel for microprocessors, microcontrollers and DSPs. Offering unprecedented ease-of-use, uC/OS-II is delivered with complete 100 % ANSI C source code and in-depth documentation [7]. Due to the consistency of hardware, the surface system and underwater system have the same device layer. Although there exist some differences according to actual resources utilized in each sub system, they have a similar user layer of the architecture as shown in Fig. 8. The differences in application layer between surface system and underwater system are evident. Fig. 9 shows the detailed task diagram of each sub system.

There are five tasks for underwater software and three for surface. The Communication task implements communication protocol and exchange data between surface and underwater. The Vehicle Operation task updates user operation and generates control commands. The Sensor Sampling task carries out sensor data acquisition. Control strategy is introduced in the Motion Control task. Decisions are made based on the sensor data, the given command and information stored in the database. Other devices such as lights, camera etc. are operated in the Peripheral Control task. The Reserved task is null for additions. This architecture is excellent in controllability, flexibility and stability.

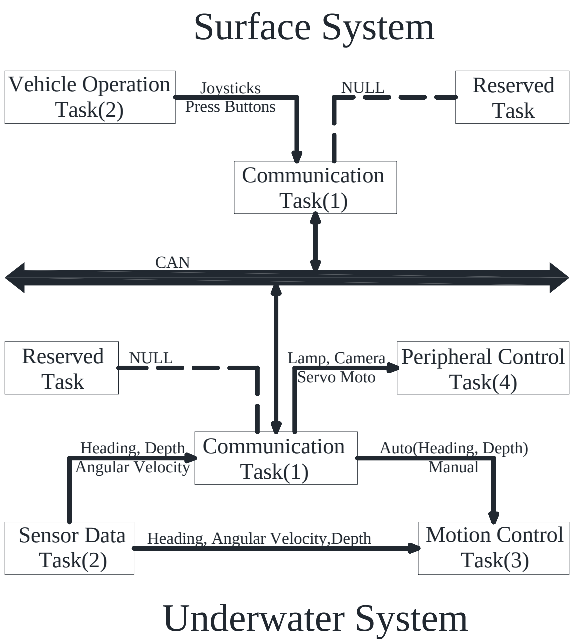


Fig. 9. Detailed task diagram (number in the block means priority).

6. Communication Protocol and Control Strategy

6.1. Communication Protocol

The surface system and underwater system communicate with each other through CAN bus. CAN implementations are designed according to BOSCH CAN specification 2.0. Herein standard message format is used and the baud rate is set to 500 bps. The communication data from surface system to underwater system is shown in Table 2.

In case of transmission '0x23 0x00 0xF0 0x09 0x94 0x00 0x00 0xB0', the sum of byte 1 to byte 7 is 0x1B0, therefore byte 8 is 0xB0.

The communication data from underwater system to surface is shown is Table 3.

The sensor data are converted into signed integer form by multiplying a coefficient and expressed using complement. In case of compass data transmission '273.5, -3.6, 18.4', we first multiply the data by 10.

2735, -36, 184

Table 2. Data format from surface to underwater.

Format	Byte	Description
1	1	0x23(format id)
	2	Operation mode
	3	Surge control
	4	Yaw control
	5	Heave control
	6	Pitch control
	7	NULL
	8	check sum from byte1 to byte7
2	1	0x24(format id)
	2	Camera control
	3	Pan&titl control
	4	Lamp switch
	5	Lamp tune
	6	NULL
	7	NULL
	8	Check sum from byte1 to byte7

Table 3. Data format from underwater to surface.

Format	Byte	Description
1	1	0x3C(format id)
	2	Head(low)
	3	Head(high)
	4	Pitch(low)
	5	Pitch(high)
	6	Roll(low)
	7	Roll(high)
	8	Check sum from byte1 to byte7
2	1	0x3E(format id)
	2	Depth(low)
	3	Depth(high)
	4	Angular velocity x-axis(low)
	5	Angular velocity x-axis(high)
	6	Angular velocity z-axis(low)
	7	Angular velocity z-axis(high)
	8	check sum from byte1 to byte7

Their complements are

0x0AAF, 0xFFDC, 0x00B8

Then

Head (high)=0x0A Head (low)=0xAF
Pitch (high)=0xFF Pitch (low)=0xDC
Roll (high)=0x00 Roll (low)=0xB8

Finally the data format of compass is

0x3C 0xAF 0x0A 0xDC 0xFF 0xB8 0x00 0x88

6.2. Control Strategy

The controller is the vehicle's brain and controls the vehicle's movements. Various advanced underwater vehicle control system have been proposed in the literature, such as sliding control [8], learning control [9] and adaptive control [10]. For the initial stage, we concentrate on reliability and effectiveness of the vehicle. Therefore the study about controllers design has been focused on the low level controllers design.

Classic PID approach is adopted for autopilot design and Ziegler-Nichols tuning rules are used to produce good values for the PID parameters [11, 12]. Because existing sensors can't measure velocity of heave, only PI controller is available for depth control. Fig. 10 shows the block diagram for the PID controller.

Here η denotes the position and orientation vector with coordinates in the earth fixed frame, ν denotes the linear and angular velocity vector with coordinates in the body-fixed frame and τ is used to describe the forces and moments acting on the vehicle in the body-fixed frame. u is the vector of control inputs. V_m is motor input voltage. The subscript d denotes the desired value. A simplified relationship between u and τ can be expressed through the linear mapping:

$$u = B\tau$$

B is a matrix known as the Thruster Control Matrix (TCM). Hippo has the following TCM:

$$B = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & l_y & -l_y \\ l_x & -l_x & 0 & 0 \end{bmatrix},$$

where l_i is the moment arms.

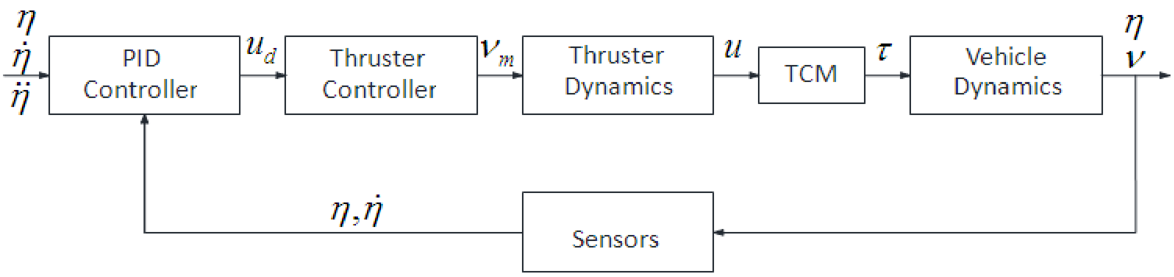


Fig. 10. Block diagram for the PID controller.

The dynamical model of underwater ROVs in body-fixed frame can be derived as shown in the following equation. The lengthy derivation procedure is omitted but more details can be found in [13]. It should be noted that we employed the simplified model without consideration of DOF of pitch and sway, as the thruster configuration don't allow for 6 DOF.

$$M\dot{\nu} + C(\nu)\nu + D(\nu)\nu + g(\eta) = \tau \,, \qquad (3)$$

$$\dot{\eta} = J(\eta)\nu \,, \qquad (4)$$

where M is the inertia matrix including added mass; C is the matrix of Coriolis and Centrifugal terms including added mass; D is the hydrodynamic damping matrix; $J(\eta)$ is a transformation matrix and the kinematic transformation (4) relates the body-fixed velocities to the time derivative of the positions in the local geographical frame; $g(\eta)$ denotes the vector of gravitational and buoyant forces and moments.

7. Experiment

To verify the performance of this control system experiments were conducted in a tank as shown in Fig. 11. Located in Institute of Underwater Engineering, East Jiangchuan Road, China, the tank has a size of 4m (length) by 2m (width) by 3m (maximum depth).

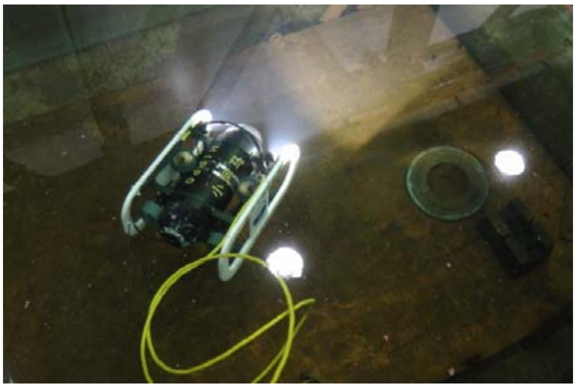


Fig. 11. A picture of tank test.

Fig. 12 shows the auto-heading result where the desired heading angle was set to 75 degree. The auto-depth experimental data was plotted in Fig. 13 where the target depth was 2.5 m.

8. Conclusion

In this paper a ROV test-bed has been developed. Experiments have been carried out to verify the effectiveness of the whole system. Future work will be focused on improvement of the vehicle. Another two thrusters will be mounted to make the vehicle motion in all six degree of freedom possible. A compact PC motherboard will replace the surface MCU board and a graphical user interface (GUI) will be developed. High performance STM32F4 series MCU will be adopted for MCU boards. Finally it is hope that this will form a basic platform that can be easily transplanted to other small-scale vehicles [14, 15].

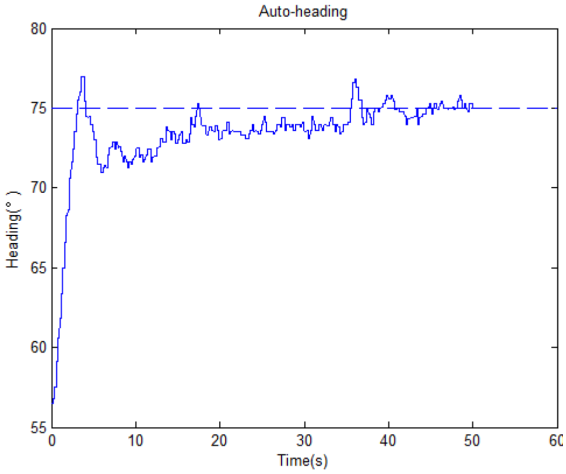


Fig. 12. Auto-heading in tank test.

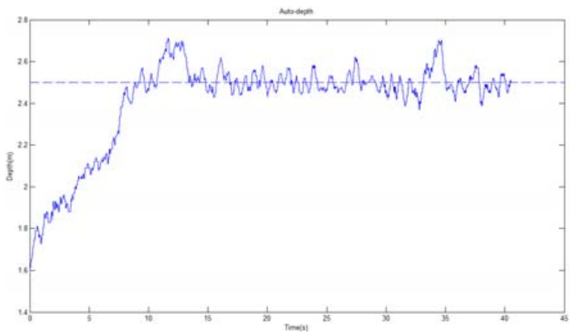


Fig. 13. Auto-depth in tank test.

References

[1]. Won-Shik Chu, Kyung-Tae Lee, Review of Biomimetic Underwater Robots Using Smart Actuators, *International Journal of Precision Engineering and Manufacturing*, Vol. 13, Issue 7, 2012, pp. 1281-1292.

[2]. Bovio E., Cecchi D., Baralli F., Autonomous Underwater Vehicles for Scientific and Naval Operations, *Annual Reviews in Control*, Vol. 30, Issue 2, 2006, pp. 117-130.

[3]. Hudson I. R., Jones D. O. B., Wigham D. B., A Review of the Uses of Work-Class ROVs for the Benefits of Science: Lessons Learned from the SERPENT Project, *The International Journal of the Society for Underwater*, Vol. 26, Issue 3, 2005, pp. 83-88.

[4]. Xinsong Jiang, Xisheng Feng, Ditang Wang, Unmanned Underwater Vehicles, Hong Liu, *Liaoning Science and Technology Publishing House*, 2005.

[5]. Satomi Ohata, Kazuo Ishii, Development of an Autonomous Underwater Vehicle for Observation of Underwater Structures, in *Proceedings of the OCEANS 2005 MTS/IEEE*, Washington, USA, 18-23 September 2005, pp. 1928-1933.

[6]. T. W. Kim, J. Yuh, Development of a real-time control architecture for a semi-autonomous underwater vehicle for intervention missions, *Control Engineering Practice*, Vol. 12, Issue 12, 2003, pp. 1521-1530.

[7]. Jean J. Labrosse, MicroC/OS-II The Real-Time Kernel, Second Edition, Ying Wang, *Beihang University Press*, 2003.

[8]. A. J. Healey, D. Lienard, Multivariable sliding mode control for autonomous diving and steering of unmanned underwater vehicles, *IEEE Journal of Oceanic Engineering*, Vol. 18, Issue 3, 1993, pp. 327-339.

[9]. J. Yuh, Learning control for underwater robotic vehicles, *IEEE Control System*, Vol. 14, Issue 2, 1994, pp. 39-46.

[10]. T. I. Fossen, S. I. Sagatun, Adaptive control of nonlinear underwater robotic systems, in *Proceedings of the IEEE Conference on Robotics and Automation*, Carlifornia, USA, 9-11 April 1991, pp. 1687-1694.

[11]. Thor I. Fossen, Guidance and Control of Ocean Vehicles, *John Wiley & Sons*, 1994.

[12]. K. J. Åström, T. Hägglunda, C. C. Hangb, W. K. Hob, Automatic Tuning and Adaptation for PID Controllers-a Survey, *Control Engineering Practice*, Vol. 1, Issue 4, 1993, pp. 699-714.

[13]. J. Yuh, Modeling and control of underwater robotic vehicles, *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 20, Issue 6, 1990, pp. 1475-1483.

[14]. Blair Thornton, Toshihiro Maki, HAL-urabo: A kit AUV for Competition and Outreach, in *Proceedings of the 6th International Symposium on Underwater Technology*, Wuxi, China, 21-24 April 2009, pp. 57-61.

[15]. J. V. Klump, R. W. Padock, I. G. Babb, The evolution and development of the small ROV as an essential experimental tool in limnological and coastal marine research, in *Proceedings of the OCEANS 2001 MTS/IEEE*, Hawaiian, USA, 5-8 November 2001, pp. 826-832.