PhD Dissertation

해양 탐사를 위한 새로운 AUV 플랫폼의 설계 및 제어

A Design and Control of a New AUV Platform for Ocean Exploration

Supervisor: Professor Hyeung-Sik Choi

December 2014

Graduate School of Korea Maritime and Ocean University
Department of Mechanical Engineering

TRAN NGOC HUY

PhD Dissertation

해양 탐사를 위한 새로운 AUV 플랫폼의 설계 및 제어

A Design and Control of a New AUV Platform for Ocean Exploration

Supervisor

Prof. Hyeung-Sik Choi

by

Tran Ngoc Huy

A dissertation submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

At the

Department of Mechanical Engineering

Graduate School of Korea Maritime and Ocean University

December 2014

We certify that we have read this dissertation and that, in our opinion, it is satisfactory in scope and quality as a dissertation for the degree of doctor of philosophy in Mechanical Engineering, submitted by **TRAN NGOC HUY**.

DISSERTATION COMMITTEE

Chairperson: Professor Sam-Sang You
Department of Refrigeration and Air-Conditioning Engineering, Korea Maritime and Ocean University

____

Supervisor: Professor Hyeung-Sik Choi
Department of Mechanical Engineering, Korea Maritime and Ocean University

____

Reviewer: Professor Jong-Rae Cho
Department of Mechanical Engineering, Korea Maritime and Ocean University

____

Reviewer: Professor Kwang-Il Hwang
Department of Refrigeration and Air-Conditioning Engineering, Korea Maritime and Ocean University

____

Reviewer: Professor Yong-Seong Cho
Department of Electronics Engineering, Dong-A University

____

December, 2014

Department of Mechanical Engineering

Graduate School of Korea Maritime and Ocean University

본 논문을 TRAN NGOC HUY 의 공학박사 학위논문으로 인준함.

위원장: 공학박사 유삼상 (인)

위 원: 공학박사 최형식 (인)

위 원: 공학박사 조종래 (인)

위 원: 공학박사 황광일 (인)

위 원: 공학박사 조용성 (인)

2014 년 12 월 28 일

한국해양대학교 대학원

Acknowledgement

First and foremost, I would like to express my sincere thanks to my advisor, Professor Hyeung-Sik Choi, for his invaluable advice, guidance and motivation throughout the course of this research. Furthermore, he always gives me the best of mental and material conditions to help me complete my study and research.

I would like to show my gratitude to members of my dissertation committee: Prof. Sam-Sang You, Prof. Jong-Rae Cho, Prof. Kwang-Il Hwang, and Prof. Yong-Seong Cho for their many helpful comments and suggestions to complete this dissertation.

I would like to say thank you to all members of KIAL Laboratory and Vietnamese students in KMOU University for all scientific discussions and helps when I studied in KIAL Lab.

Last but not least there are various thanks to my fathers, my mothers, my sisters, my brothers and especially my wife and my son for their love, encouragement and sympathy for me not only in this dissertation time but also in the whole of my life.

Tran Ngoc Huy

KMOU, Busan, South Korea

A Design and Control of a New AUV Platform for Ocean Exploration

Tran Ngoc Huy

Department of Mechanical Engineering

Graduate School of Korea Maritime and Ocean University

Abstract

This dissertation presents a design, control, and implementation of a new autonomous underwater vehicle (AUV) platform for ocean exploration, which is designed as a torpedo shape with small and light enough to be handled easily by one person. According to a unique ducted propeller and rudder located at the aft, the AUV can perform horizontal motion. It can also control pitch angle and depth motion by an inside mass shifter mechanism (MSM) which changes the vehicle center of gravity. In addition, hardware and software architectures of the control system are addressed and the functions of all parts are described.

Navigation system design is a critical step in determining position, course, and distance traveled of a vehicle. Hence, in this dissertation, a navigation algorithm based on discrete Kalman filter is developed to estimate the states and to compensate accumulative errors. Moreover, experiments were carried out to verify the performance of the developed algorithm using the developed navigation system of AHRS, GPS-INS, and DVL-INS.

Based on the nonlinear six degree of freedom AUV model and multivariable sliding mode control method, the controllers of heading and depth are designed. After that, with the developed navigation system and suggested controllers, a number of experiments were performed and their results are compared with the simulation outputs.

In practical operation, the AUV needs to avoid obstacles such as reefs and seawalls in the littoral environment. For this, an algorithm of obstacle avoidance

was devised. Also, a modeling of the sonar sensor that gathers information of the ocean environments and the forward obstacle for safe navigation was established.

So this dissertation focuses on examining the behavior and control system required for the AUV to maneuver over obstacles in the vertical and horizontal planes. Through the developed obstacle avoidance algorithm, the AUV has ability to use range and bearing data received from sonar modeling to determine if that return constitutes a threat along its desired path and further navigate around the threat before regaining its original path. In addition, it is difficult and unsafe to maneuver AUV to avoid obstacles with constant surge velocity, the velocity controller with output of the propeller thrust is also considered.

Finally, to validate the developed obstacle avoidance algorithm and controllers, simulations in vertical and horizontal planes were performed and their results were described.

KEY WORDS: Autonomous underwater vehicle, navigation system, discrete Kalman filter, sliding mode control, obstacle avoidance.

Contents

Nomenclature

m	vehicle mass
p	roll rate (body-fixed reference frame)
q	pitch rate (body-fixed reference frame)
r	yaw rate (body-fixed reference frame)
u	surge velocity (body-fixed reference frame)
v	sway velocity (body-fixed reference frame)
w	heave velocity (body-fixed reference frame)
x_g	the body-fixed coordinate of the vehicle center of gravity on the surge axis
y_g	the body-fixed coordinate of the vehicle center of gravity on the sway axis
z_g	the body-fixed coordinate of the vehicle center of gravity on the heave axis
x	the x-component inertial coordinate of the vehicle
y	the y-component inertial coordinate of the vehicle
z	the z-component inertial coordinate of the vehicle
x_m	the body-fixed coordinate of the movable mass center of gravity on the surge axis.
x_{m_d}	the desired position of the movable mass
ϕ	roll angle (inertial reference frame)
θ	pitch angle (inertial reference frame)
ψ	yaw angle (inertial reference frame)
W	vehicle weight
B	vehicle buoyancy

List of Tables

List of Figures

Chapter 1: Introduction

1.1 Background

The ocean covers about two-thirds of the earth and has a great effect on the future existence of all human beings. The ocean is generally overlooked as we focus our attention on land and atmospheric issues. That is why we have not been able to explore the full depths of the ocean, and its abundant living and non-living resources. The underwater robots can help us better understand marine and other environmental issues, protect the ocean resources of the Earth from pollution, and efficiently utilize them for human welfare, in Fig. 1.1.

Fig. 1.1 Application of underwater robots (www.google.com)

There are many various kinds of the underwater vehicles, as shown in Fig. 1.2. Most of the early commercial scientific work under the oceans was done by manned underwater submersibles. The production of these reached their peak in the late 1960s when several of the defense contractors such as General Dynamics, Rockwell and Westinghouse built these systems [1]. Safety concerns and technology advances led to the idea of unmanned submersibles and the remotely operated vehicles (ROV) were developed. In the early 1980s, many of initial technical problems in ROV design were worked out. At this time, ROVs began to take over many of the tasks originally assigned to manned submersibles and are currently the workhorse of the oceans underwater vehicles [2]. Woods Hole Oceanographic Institution’s ROV Argo, for instance, was a primary tool used in discovering the wreck of the Titanic [3]. The tethered submersible enabled 24-hour reconnaissance of the wreck site at depths of almost 2.5 miles while the research teams remained above the surface. More recently, the Deepwater Horizon (British Petroleum) Oil Spill in the gulf of Mexico, employed teams of ROVs in several efforts to stem the flow from the leaking pipe. These ROVs featured fully maneuverable actuator arms and were operating in depths of nearly 1 mile [4] which is beyond the limit of human physiology [5]. However, extensive use of manned submersibles and ROVs are currently limited to a few applications because of very high operational costs, operator fatigue and safety issues.

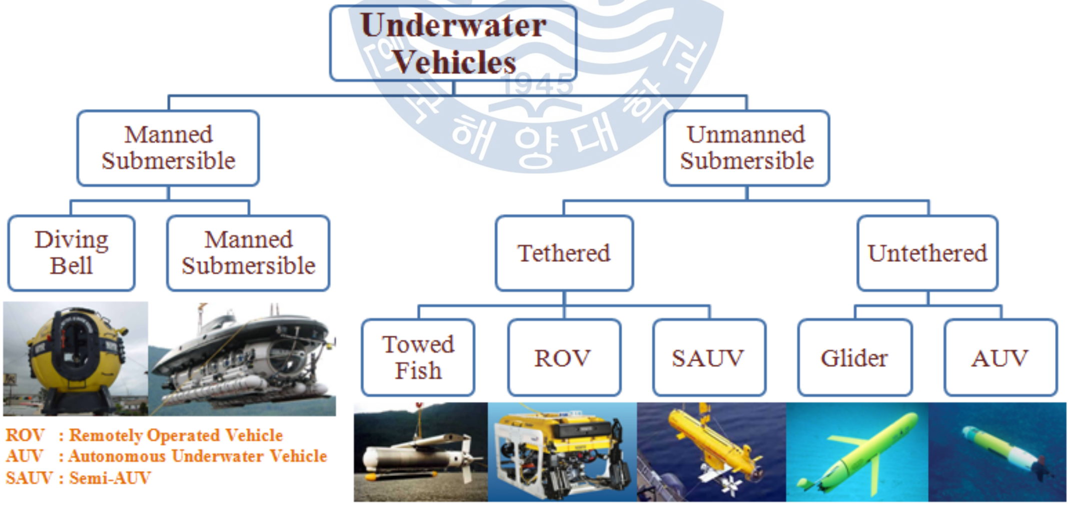

Fig. 1.2 Classification of underwater robots

With the limitations on ROVs and manned submersibles, the untethered underwater vehicle (UUV) or autonomous underwater vehicle (AUV) started to come on the scene in the 1970s. Moreover, AUVs are capable of operating in numerous underwater environments, including littorals and even under polar icecaps. They have the capability to search, detect and classify objects using its sonar and video camera and they can also measure oceanographic data such temperature, salinity and current. These capabilities are necessary for both oceanographic research and military operations such as mine hunting and harbor reconnaissance. With advances in computer processors and memory, the concept of autonomy underwater became apparent and real. In recent years, various research efforts have increased autonomy, reliability of the vehicle and minimized the need for the presence of human operators. Universities and institutes have been researching these autonomous underwater vehicles such as the Remus [6] shown in Fig. 1.3 at the Woods Hole Oceanographic Institution (WHOI), the Bluefin [7] at Massachusetts Institute of Technology (MIT), the R2D4 [8] at the University of Tokyo, and several others. Government and military research laboratories expanded in the untethered arena with AUVs of the Naval Postgraduate School, the Naval Undersea Warfare Center and so on.

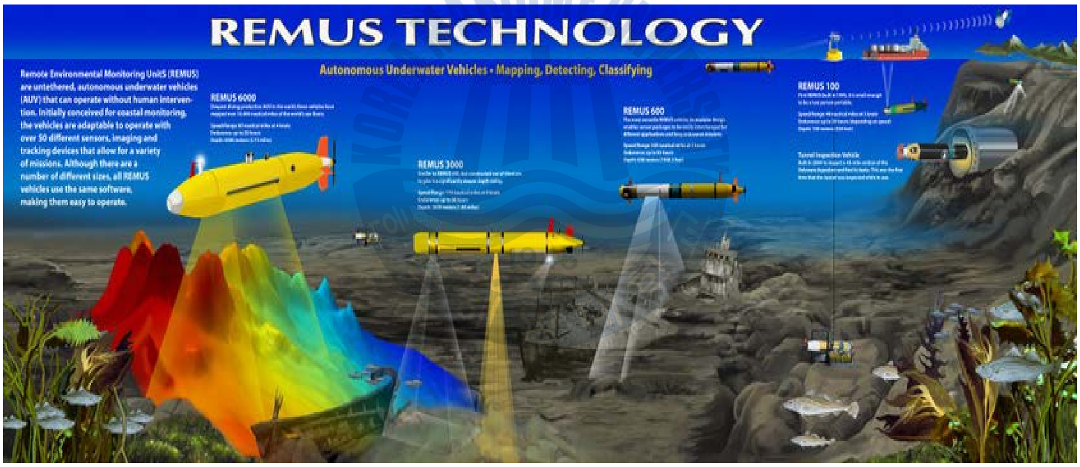

Fig. 1.3 An underwater illustration of six REMUS AUV designed by the WHOI (http://www.whoi.edu/osl/remus-auv)

1.2 Motivation

Despite the considerable improvement in AUV performance, AUV technologies are still attractive to scientists and engineers as a challenging field. For example, multiple AUVs, underwater docking, and obstacle avoidance are recent challenging issues in the field of AUV technologies [9], [10], [11]. Or challenges in design such as the torpedo-AUVs are designed with a fixed propeller at the aft and use pairs of fins for steering and diving [12]. However, the drawback of these designs is that they could not work effectively at low speeds. A more advanced architecture of AUV without any fins was devised for Bluefin [13]. Its propeller that is in a double gimbal arrangement could be actuated in the horizontal and vertical directions, worked as both a rudder and elevator. In this configuration, the vehicle is more clean-limbed, and the control system is smaller. However, both rudder and elevator of this vehicle are tied with the single propeller, so interoperability between them may be pretty much limited, for example, in case of a sharp turn with nonzero roll which results in an undesired change in depth. This issue needs a coupled control system to mix rudder and elevator commands to minimize disturbances in depth as mentioned in [14]. Therefore, a unique propeller and the inside mass shifter mechanism changing the vehicle center of gravity was designed. With this structure, the AUV can perform horizontal and vertical motions independently.

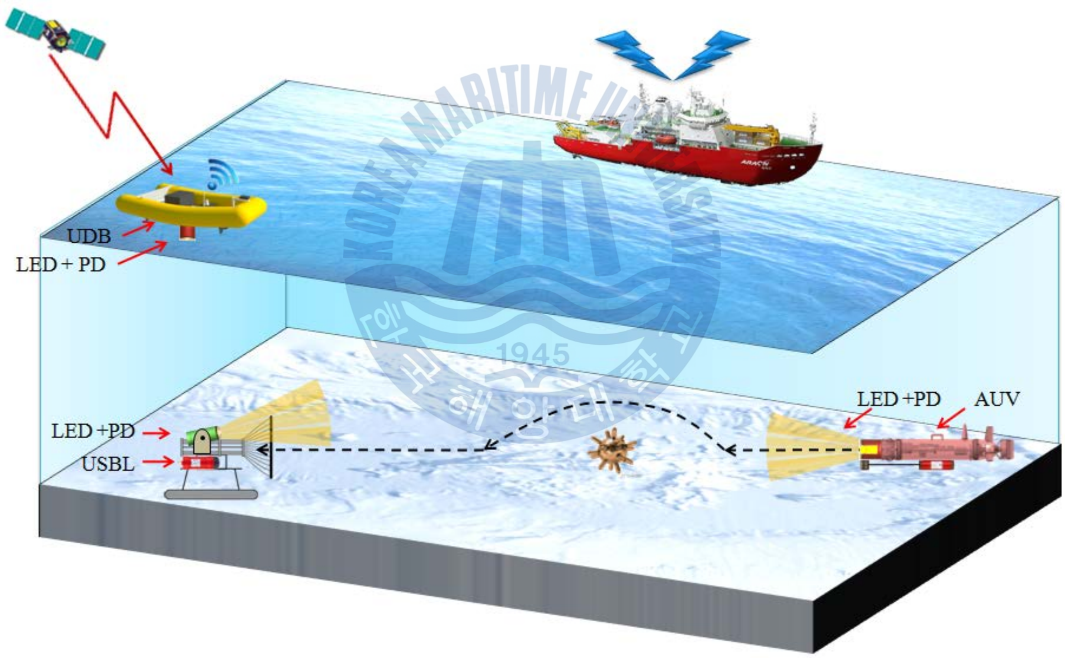

Fig. 1.4 Obstacle avoidance of the AUV

At this time, most AUVs travel on a fixed path through the waters where a certain level of knowledge of the seafloor is known. A problem exists when AUVs travel into unknown waters or where the local bathymetry is not predetermined. On this occasion, AUVs is likely to encounter underwater obstacles such as coral reefs, sea walls and shipwrecks. Obstacle avoidance technology is the creation of computer algorithms that will determine maneuvering options for an AUV confronting an obstacle as shown in Fig. 1.4, are still in development. In a chaotic and treacherous underwater environment, it can't be hoped to be able to plan for every contingency, such as in military operations on hostile littorals. Therefore, it is necessary to develop an obstacle avoidance algorithm within the AUV that will allow it to recognize an obstacle and make a correct maneuver to avoid the obstacle and then return to its mission path as soon as possible.

1.3 Contribution

In this dissertation, the design and control of the new AUV platform were presented. With the unique propeller and the inside mass shifter mechanism changing the vehicle center of gravity, the AUV can perform horizontal and vertical motions independently.

The control system including hardware and software architectures were designed, and navigation system composed of the GPS-INS-DVL sensors and the Kalman filter algorithm was also developed. The INS system based on the measured acceleration information of the vehicle can estimate the position, velocity, and attitude of the AUV. But these values have accumulative errors due to inherent drift of the dead-reckoning velocities and integration of the accelerations. Hence the Kalman filter algorithm combined with a number of sensors such as GPS, DVL, and pressure sensor was applied to compensate these errors.

The developed obstacle avoidance algorithm for the AUV to maneuver over obstacles in the vertical and horizontal planes was proposed. Hence the AUV has ability to use range and bearing data received from sonar modeling to determine if that return constitutes a threat along its desired path and further navigate around the threat before regaining its original path. In addition, it is difficult and unsafe to maneuver AUV to avoid obstacles with constant surge velocity, the velocity controller with output of the propeller thrust is also considered.

Chapter 2: Architecture of the Developed AUV

2.1 Mechanical Design

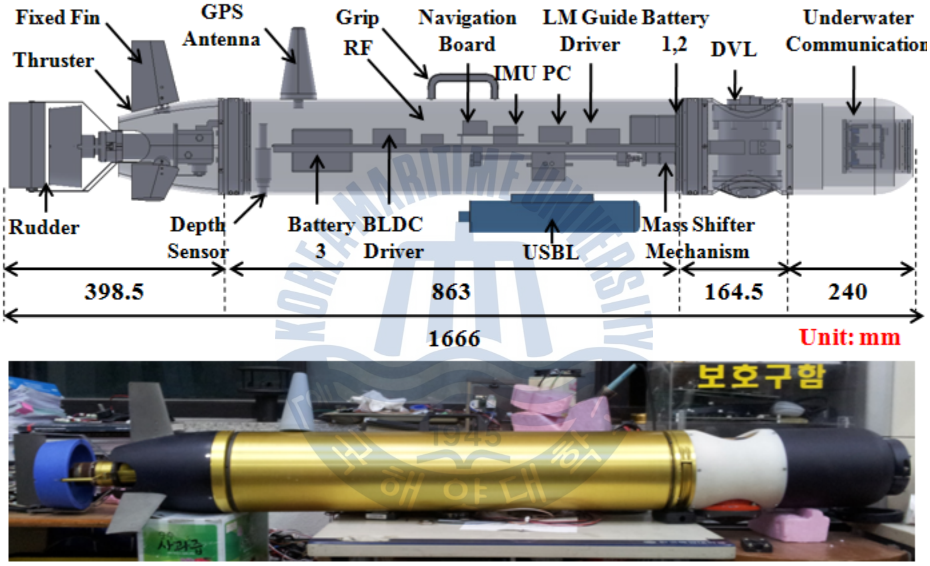

Fig. 2.1 The virtual design and developed AUV

The AUV has an approximative length of 1.67 m, a maximum diameter of 0.18 m, and weight of 31.5 kg as shown in Fig. 2.1. It is composed of a pressure hull, propulsion system, mass shifter mechanism, underwater communication system, electronic control system, and etc. The vehicle is designed with a buoyancy is slightly greater than its weigh, so that the AUV will eventually float to the surface in the event of a computer or power failure. To promote the stability of the vehicle, three fixed fins are mounted on the perimeter of the circular aft with intervals of 120 degrees. The hull of the AUV is divided into three sections: bow, aft, and main housing. The bow and aft shapes are determined by the Myring hull profile equations [15] to have the minimum drag force. The bow and aft are made of NC-nylon, and the main housing is made of aluminum. They are all connected together and waterproof-sealed by o-rings.

2.1.1 Rudder and Propulsion Structure

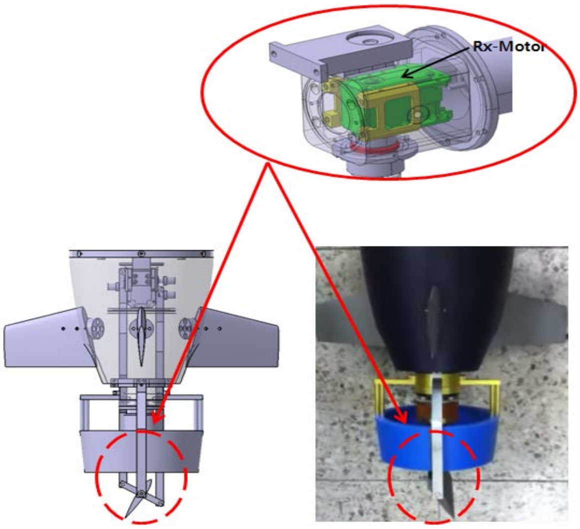

Fig. 2.2 Virtual design and manufacture of the rudder

With a unique ducted propulsion and rudder located at the aft, the AUV can perform horizontal motion. As shown in Fig. 2.2, the rudder is connected with RX28 Dynamixel actuator by links. Hence, under control of the rudder angle, the fluid flow will be changed. And then, the AUV can control the turning direction. The conventional thrusters which can cease to function at depths of greater than 10 feet due to the pressure of the water breaking the seals of the thruster, and entering the motor. One solution to this problem is magnetic coupler that uses the ability of magnets to exert force through objects to separate the motor housing from water. Therefore, the AUV propulsion system is designed using Maxon Motor 200 W and magnetic coupler to rotate propeller. The torque and ampere of the developed propulsion system were measured and shown in Fig 2.4. The maximum torque is 3.7 kgf at the maximum ampere of 11 A.

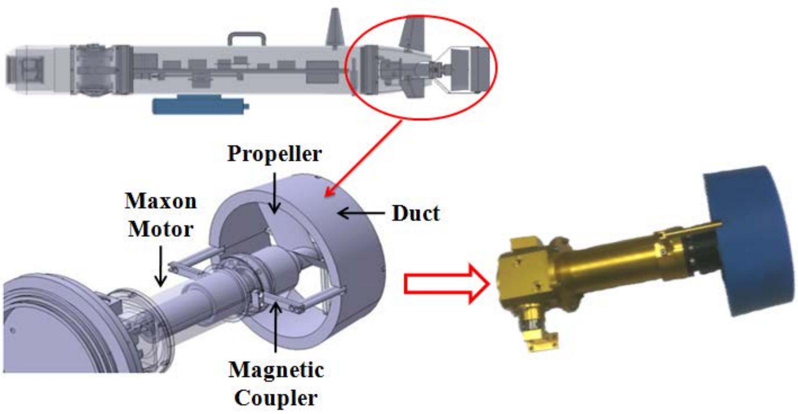

Fig. 2.3 Virtual design and manufacture of the propulsion system

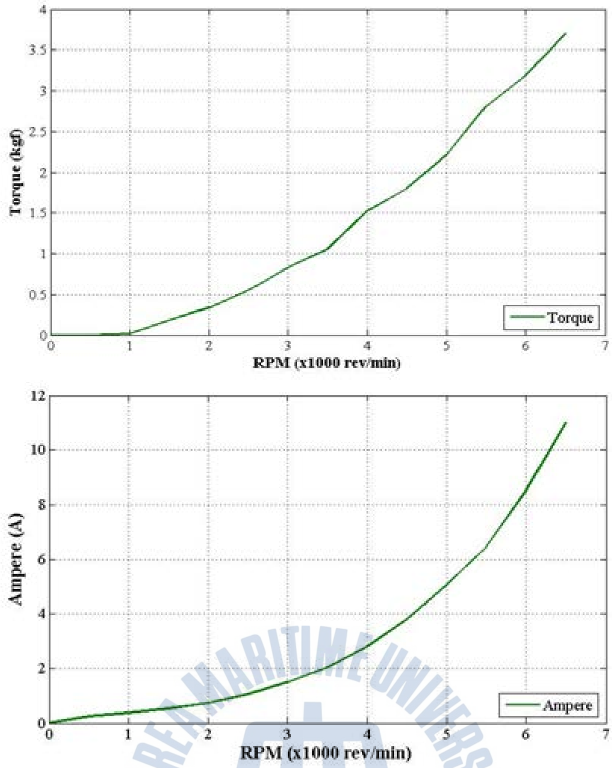

Fig. 2.4 Torque and Ampere of the propulsion system

2.1.2 *Mass Shifter Mechanism (MSM)*

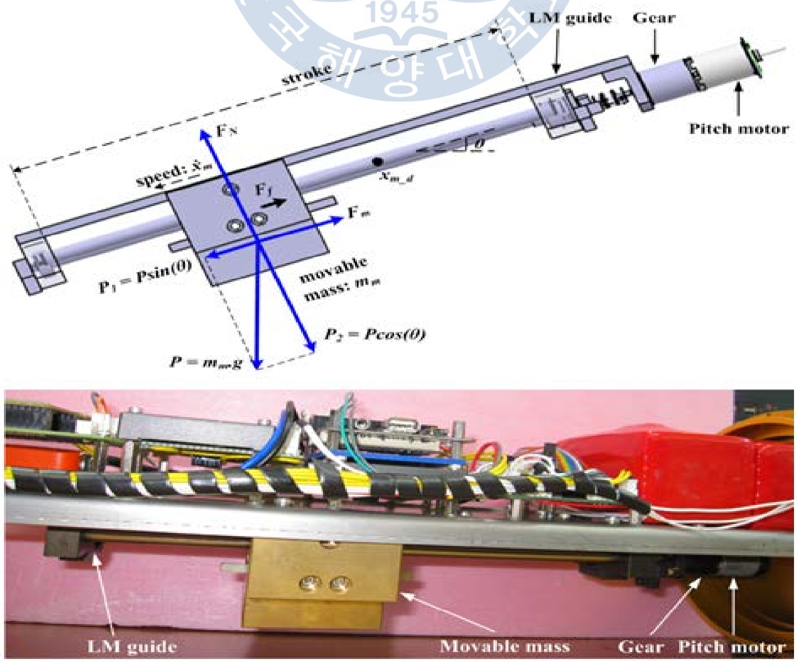

Fig. 2.5 Mass shifter mechanism

With an inside mass shifter mechanism, the AUV can control pitch angle and perform depth motion by changing the center of gravity. This mechanism is designed with a linear motor (LM) guide actuator, and a movable mass ($m_m = 2.5\,kg$) fixed onto the LM block, as shown in Fig. 2.5. When the movable mass goes forwards or backwards, the center of gravity of the whole vehicle also shifts forwards or backwards, respectively. These shifts make the vehicle pitch down or up; and under propulsion of the thruster, the vehicle will be propelled downward or upward respectively.

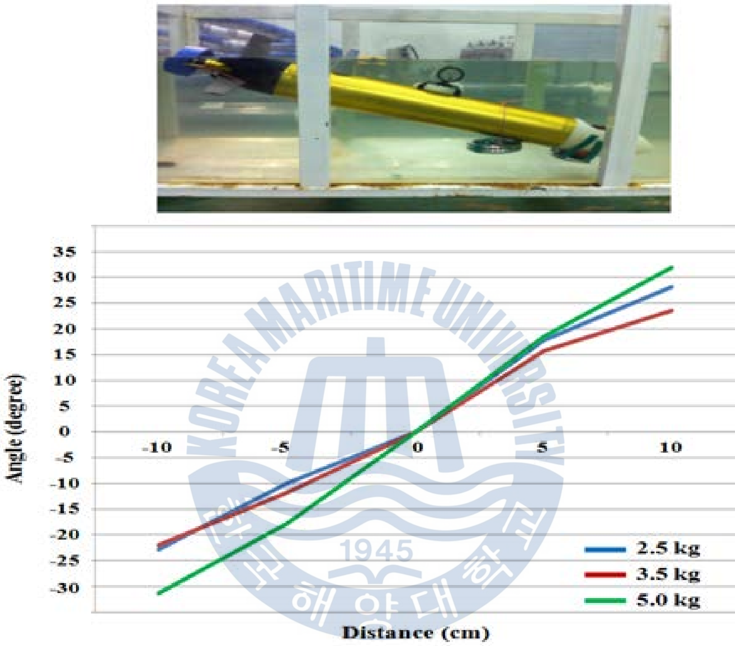

Fig. 2.6 Experiment results for each different weight of the movable mass

The length of stroke, the weight of movable mass, and the installation location of the LM guide are designed in the limited space inside the vehicle such that the pitch angle can promptly change within an appropriate range for control purposes. To determine value of the movable mass, experiments were carried out with the mass of 2.5 kg, 3.5 kg, and 5 kg respectively were placed outside of the AUV. As shown in Fig. 2.6, there is not much difference of the pitch angle at each weight value. Hence, based on trial and error method, the weight of movable mass is selected of 2.5 kg, and the length of stroke is [-0.1 m, 0.1 m]. After manufacturing the mass shifter mechanism, the experiments were performed again at the movable mass's position of 10 cm, and -10 cm respectively. The results are shown in Fig. 2.7.

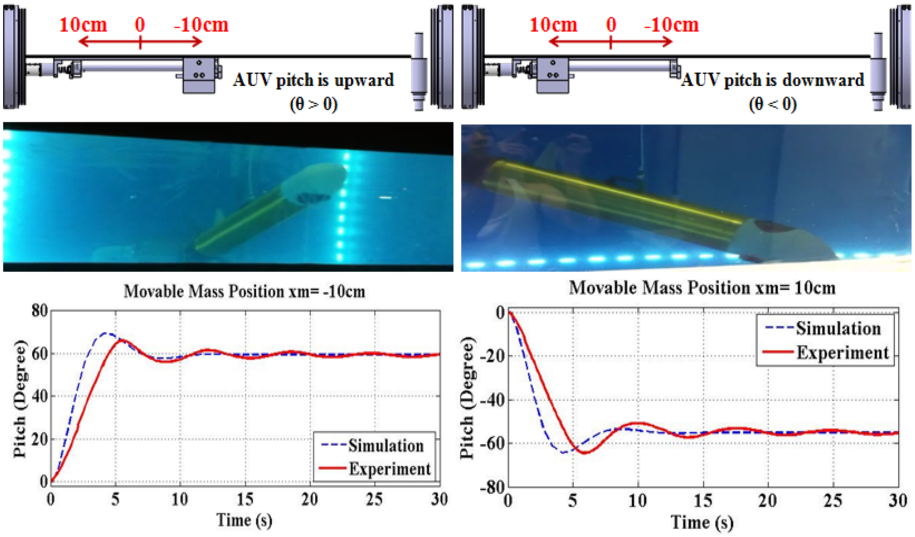

Fig. 2.7 AUV pitch angles with xm = -10cm and 10cm

2.2 Control Hardware Design

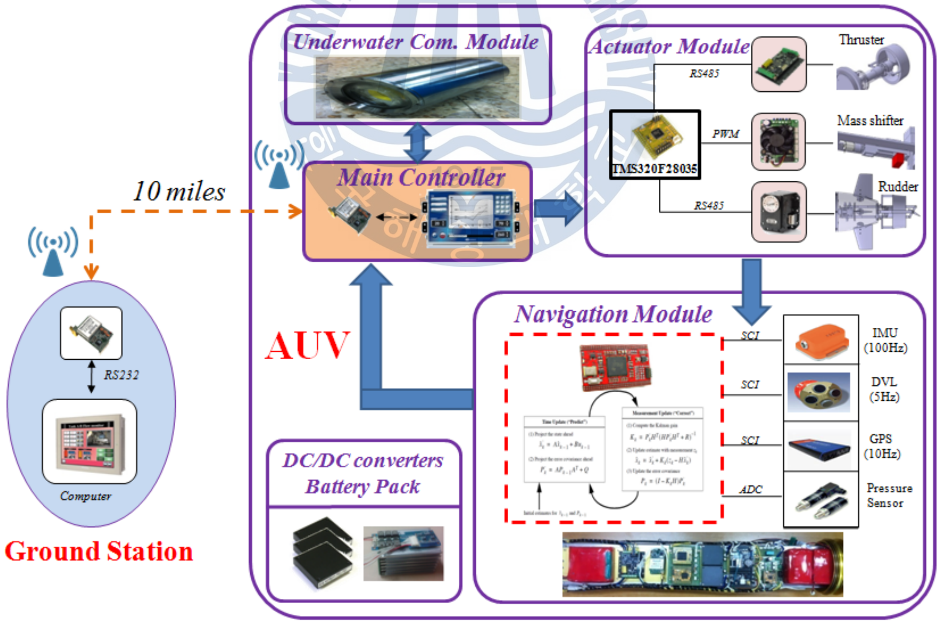

Fig. 2.8 Control system architecture block diagram

The control system shown in Fig. 2.8 includes one main controller board using IEC-667 embedded system and three sub-boards (actuator, navigation, and underwater wireless communication boards) using digital signal processors (DSP) TMS320F28346 and TMS320F28035 respectively. The navigation board is connected to sensors including IMU, compass, depth sensor, DVL, and GPS receiver. The specifications of these sensors are shown in table 1. After receiving data from sensors, the navigation board will run the algorithm programs of the INS and the Kalman filter to estimate the velocities, attitude, and position of the vehicle and then transfers these calculation results to the main controller board for controlling the AUV. A detailed description of the navigation algorithm and experiments will be presented in chapter 4. The actuator board receives instructions from the main board to control the mass shifter actuator thruster, and rudder. When the vehicle is on the surface, the RF transceiver receives commands from the ground station and transfers them to the main board. Using the data obtained from the navigation board and RF transceiver, the embedded computer directly gives out instructions to the actuator board to control the mass shifter equipment, thruster, and rudder. Finally, the underwater wireless communication is not only used as an underwater lighting fixture, but also used as an underwater communication device by controlling the distance between light source and lens when communication is needed.

Table 1. Specifications of electronic components used in AUV control system

Embedded Controller – IEC667 Lite	
System	▪ CPU: 32Bit RISC ARM1179JZF - 667MHz, RAM: 256MB, Flash:128MB ▪ SD Memory support, Supply Voltage: 5V
Communication	▪ RS232: 2 channels, RS485: 1 channel, TTL: 1 channel ▪ USB Device: 1xUSB1.1, 1xUSB2.0 ▪ Ethernet: 10Mbps
Microcontroller – TMS320F28335	
System	CPU: 32 bit, Floating-point Unit, 150 MIPS, RAM: 34K, Flash: 256K
Peripheral	6/2x CAP/QEP, 18xPWM, 3xTimer, 12 bit ADC resolution, 3xSCI, 2xCAN, 1xSPI, 88xI/O, Watchdog…
Sensors	
IMU	Static accuracy (roll, pitch): < 0.50, Static accuracy (yaw): < 10
GPS	AsteRx1: Position accuracy 1.7m x 3m, Velocity accuracy 2cm/s x 4cm/s
DVL	NavQuest600M, Maximum velocity: ± 20 knots, Velocity accuracy: 1% ± 1mm/s, Digital interface: Rs-232 v Rs-422
Depth sensor	PSHD0002KBPJ, Accuracy: ±0.15%, Operating voltage: 11V~28V,

	Output: 0~5V
Electrical Components	
RF Module	24XStream, 2.4 GHz, Indoor up to 180m, Outdoor up to 16km, supply voltage 5VDC, Serial data interface UART
BLDC driver	Power: 300W, Nominal supply voltage: 24V, Continuous current: 46A, PWM frequency 20~83KHz, RS232 9600~115200bps, CAN speed up to 1Mbps
DC driver	Power: 300W, Supply voltage: 10~24V, Current: 20A
Batteries	Lithium-polymer, 2x(26V, 10.6Ah), 2x(26V, 5.3Ah)

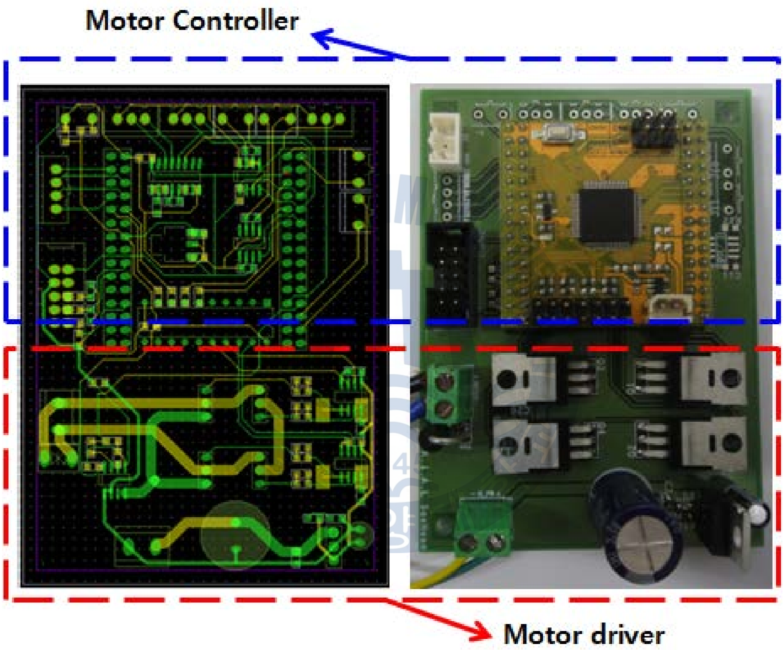

Fig. 2.9 Motor Driver & Controller for LM Guide actuator

To control position and velocity of the movable mass, a developed DC motor driver and controller were designed to drive the LM Guide actuator, as shown in Fig. 2.9. The controller uses DSP micro-processor to receive encoder signal, and then apply PID algorithm to generate PWM signal and driving direction for the power amplifier section. The power amplifier section is structured by H-Bridge as shown in Fig. 2.10. The H-Bridge is built with four switches. When the switches S1 and S4 are closed (S2 and S3 are open) a positive voltage will be applied across the motor. And by opening S1 and S4 switches and closing S2 and S3 switches, this voltage is reversed, allowing reverse operation of the motor.

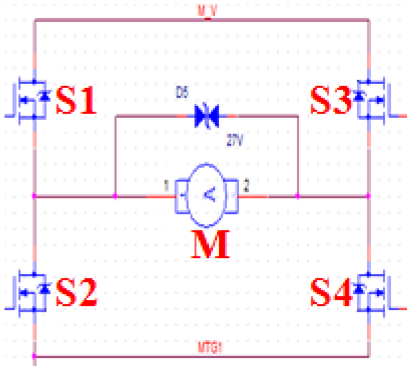

Fig. 2.10The schematic of H-Bridge

The battery system is composed by four 26 V rechargeable lithium-polymer packs. Each pack has capacity of 5.3 Ah and its specifications are shown in table 2. To drop the voltage to 24 V, 15 V, 5 V for supplying to sensors, thruster, communication, and control systems, DC/DC converters are used, as shown in Fig. 2.11.

Table 2. Specifications of Lithium Polymer battery

Rated Capacity		Typ. 5,300mAh
Nominal Voltage		3.7V
End Of Discharge		3.0V
Max. Charge Voltage		4.2V±0.03V
Max. Conti. Charge Current		10.6A
Max. Conti. Discharge Current		10.6A
Operation Temperature Range	charge	0 ~ 45^{0}C
	Discharge	-20 ~ 55^{0}C
Storage Temperature Range	1 Year	-20 ~ 25^{0}C
	3 Month	25 ~ 40^{0}C
	1 Week	40 ~ 60^{0}C
Weight		102g
Cell Dimension	Length	Max.97.5mm
	Width	Max.64.5mm
	Thickness	Max.7.8mm

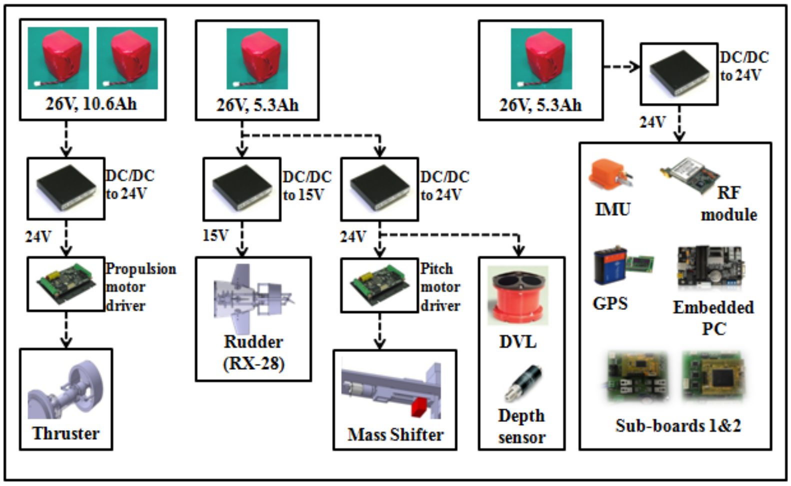

Fig. 2.11 Power distribution diagram

2.3 Control Software Design

The control software architecture of the AUV system is composed of two elements: monitoring system software programmed for the ground center station, and control system software used in the onboard AUV module. The programming language used in both of them is Visual Studio 2008.

2.3.1 Ground Monitoring System Software

Fig. 2.12 The visual operating interface in the ground monitoring system

To monitor the states and to communicate with the AUV, the GUI (Graphic User Interface) control software of the ground system as shown in Fig. 2.12 was programmed. It is not only prompt and flexible to set control modes and parameters for the AUV's initialization. But also quite accurate to transmit a pre-scheduled trajectory following commands, and receive state feedback from the vehicle. The algorithm diagram of the ground monitoring system shown in Fig. 2.13 will clarify the above statement. After starting the GUI program, the user will be asked to select control modes of remote or automatic control. For each control mode, there will be a different parameter initialization. For example, in remote control mode, the axis and buttons on the joystick should be declared to correspond with the axes (surge, sway and yaw) of the body-fixed co-ordinate put on the AUV, maximum and minimum speeds of the vehicle when adjusting the joystick potentiometer, and sensitivity (loop time) for reading signal from the joystick, etc. With automatic control mode, first, the user should select control algorithms, such as heading control, depth control, waypoint tracking, and path tracking, etc. Next, sensors selection (IMU, GPS, DVL, depth sensor) used in the navigation system, desired trajectories, and control gains should be chosen, to complete declaration for the parameter initialization. After that, the monitor of the GUI program starts operating. With a question loop every 100 ms (10 Hz), the ground monitoring system can quickly update, exactly log, and intuitively display the states of the AUV through real-time visual graphs. Moreover, a special feature of this ground monitoring program is its synchronous nature. If errors, such as tracking error, navigation error, out of battery, or damaged equipment (sensors, thrusters, control boards), are detected from the processing of feedback signals, the user can immediately interfere in the onboard control system software, to solve these problems.

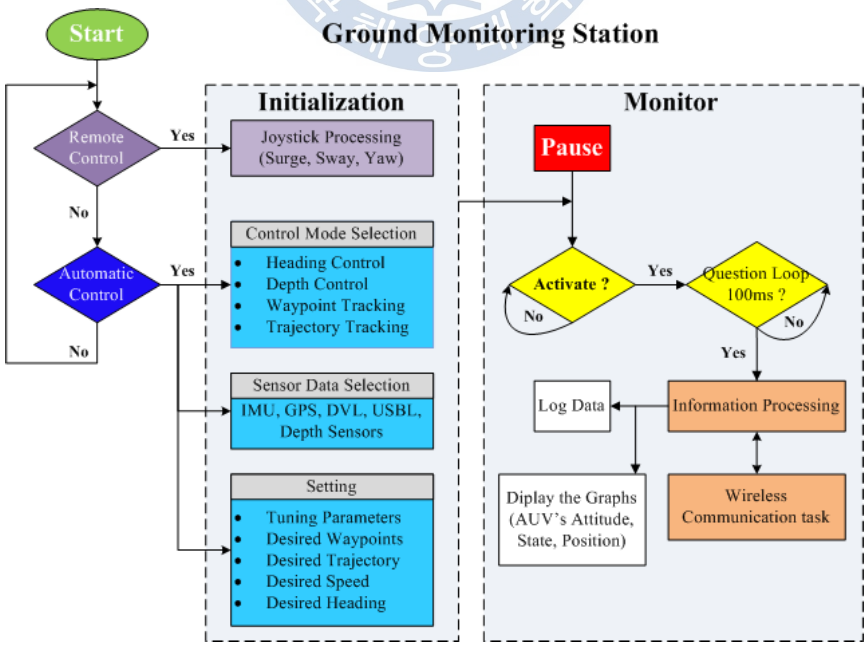

Fig. 2.13 An algorithm diagram of the ground monitoring system

2.3.2 *Onboard Control System Software*

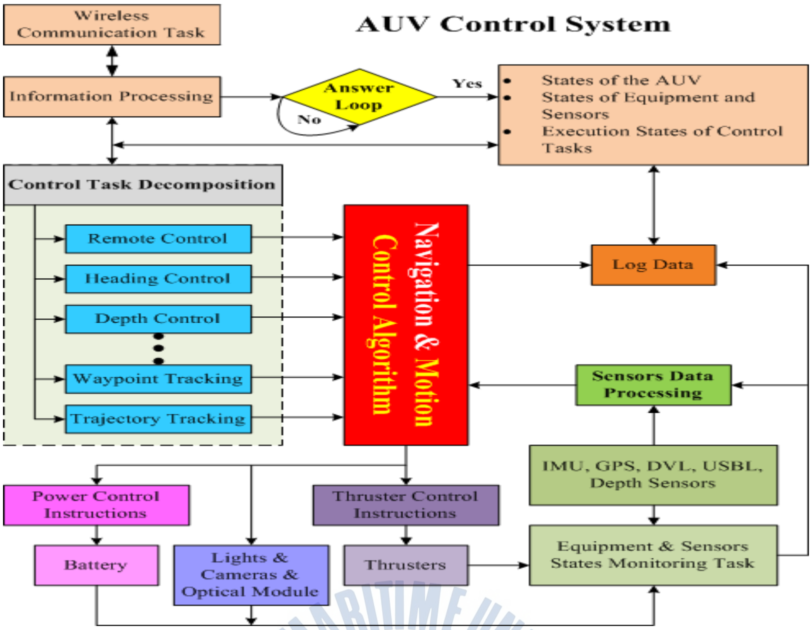

Fig. 2.14 An algorithm diagram of the onboard control system

The software architecture of the onboard control system is presented in Fig. 2.14. In this, the mission of the information processing block is to receive commands from the ground monitoring station, and to sort them according to separated missions already shown in the ground monitoring system software, such as control modes, control algorithms, and etc. Then, these delivered missions will be completely performed by the next control blocks. The control task decomposition block is used to initialize control parameters for each different control. These parameters will be sent to the navigation and motion control algorithm block that is the main control block, to process the control missions, such as remote control, depth control, waypoint tracking, etc. Besides that, navigation values (attitude, position, velocity) are also estimated in this block, through data received from the sensors. The navigation algorithm will be particularly described in the control strategy design section. According to situations of the control tasks and navigation estimation, control instructions are processed by the embedded computer, and sent to the actuator module to operate the AUV. In addition, the states of the AUV, as well as onboard equipment and sensor information, will be summarized to save in logbook memory, and also to answer the question loop (10 Hz), received from the ground monitoring station when AUV is operating on the surface.

Chapter 3: Mathematical Model of the AUV

3.1 Mass Shifter Mechanism

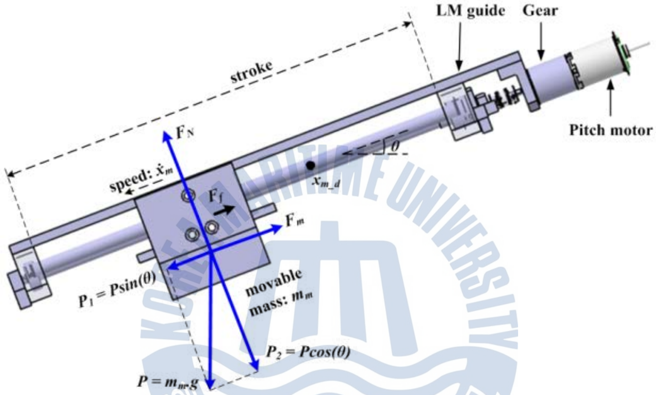

Fig. 3.1 Mass Shifter Mechanism of the AUV

Fig. 3.1 shows all the forces acting on the movable mass, they are the weight of the movable mass P, normal force F_N, total frictional force F_f, and force of the pitch motor F_m. In this figure, it is assumed that the movable mass is moving towards the stern of the vehicle with speed $\dot{x}_m$ and the vehicle pitches up from horizontal at a pitch angle θ. Because friction always acts in a direction opposite the speed, the direction of total frictional force F_f is towards the bow of the vehicle. The weight of the movable mass could be resolved into two components at right angles to each other, P_1 and P_2. P_1 is parallel to the screw shaft and P_2 is perpendicular to P_1. P_2 is balanced by the normal force F_N which is exerted on the movable mass by the screw shaft.

The differential equation describing the dynamics of the mass shifter mechanism is as follows:

$$m_m \ddot{x}_m = P_1 + F_f + F_m \tag{1}$$

where $\ddot{x}_m$ denotes the acceleration of the movable mass

From Eq. (1), with assumption F_f small compared to the other forces, we can simply to:

$$\ddot{x}_m = f + u_m \tag{2}$$

where $u_m = F_m / m_m$ and $f = g\sin\theta$

With Eq. (2), to drive the movable mass to the desired position x_{m_d}, the appropriate force u_m need to be calculated by a position controller.

3.2 Vehicle Kinematics

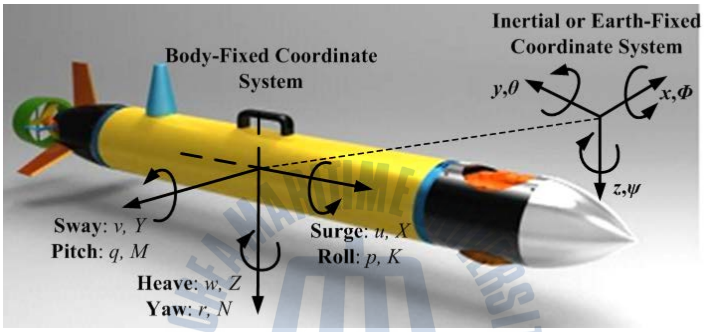

Fig. 3.2 Body-fixed and inertial coordinate systems

As shown in Fig. 3.2, the general motion of the AUV six degrees of freedom (DOF) can be described by the following vectors:

$$\eta = [\eta_1^T, \eta_2^T]^T = [x, y, z, \phi, \theta, \psi]^T \tag{3}$$

$$\upsilon = [\upsilon_1^T, \upsilon_2^T]^T = [u, v, w, p, q, r]^T \tag{4}$$

$$\tau = [\tau_1^T, \tau_2^T]^T = [X, Y, Z, K, M, N]^T \tag{5}$$

where η denotes the position and orientation vector with coordinates in the earth fixed frame, υ 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.

The following coordinate transform relates translational velocities between the body-fixed and inertial coordinates:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = J_1(\eta_2) \begin{bmatrix} u \\ v \\ w \end{bmatrix} \tag{6}$$

where $J_1(\eta_2) = [J_1^1(\eta_2), J_1^2(\eta_2), J_1^3(\eta_2)]$

$$J_1^1(\eta_2) = \begin{bmatrix} \cos\psi\cos\theta \\ \sin\psi\cos\theta \\ -\sin\theta \end{bmatrix} J_1^2(\eta_2) = \begin{bmatrix} -\sin\psi\cos\phi + \cos\psi\sin\theta\sin\phi \\ \cos\psi\cos\phi + \sin\psi\sin\theta\sin\phi \\ \cos\theta\sin\phi \end{bmatrix}$$

$$J_1^3(\eta_2) = \begin{bmatrix} \sin\psi\sin\phi + \cos\psi\sin\theta\cos\phi \\ -\cos\psi\sin\phi + \sin\psi\sin\theta\cos\phi \\ \cos\theta\cos\phi \end{bmatrix}$$

The second coordinate transform relates rotational velocities between the body-fixed and inertial coordinates:

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = J_2(\eta_2)\begin{bmatrix} p \\ q \\ r \end{bmatrix}, J_2(\eta_2) = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi/\cos\theta & \cos\phi/\cos\theta \end{bmatrix} \tag{7}$$

Note that $J_2(\eta_2)$ is not defined for pitch angle θ = ±90°. This is not a problem, as the vehicle motion does not ordinarily approach this singularity.

3.3 Vehicle Rigid-Body Dynamics

Given that the origin of the body-fixed coordinate system is located at the center of buoyancy, the following represents the full equations of motion for a six degree-of-freedom rigid body in body-fixed coordinates [16]:

$$\begin{cases} m[\dot{u} - vr + wq - x_g(q^2 + r^2) + y_g(pq - \dot{r}) + z_g(pr + \dot{q})] = \sum X \\ m[\dot{v} - wp + ur - y_g(r^2 + p^2) + z_g(qr - \dot{p}) + x_g(qp + \dot{r})] = \sum Y \\ m[\dot{w} - uq + vp - z_g(p^2 + q^2) + x_g(rp - \dot{q}) + y_g(rq + \dot{p})] = \sum Z \\ I_{xx}\dot{p} + (I_{zz} - I_{yy})qr - (\dot{r} + pq)I_{xz} + (r^2 - q^2)I_{yz} + (pr - \dot{q})I_{xy} \\ \qquad + m[y_g(\dot{w} - uq + vp) - z_g(\dot{v} - wp + ur)] = \sum K \\ I_{yy}\dot{q} + (I_{xx} - I_{zz})rp - (\dot{p} + qr)I_{xy} + (p^2 - r^2)I_{xz} + (qp - \dot{r})I_{yz} \\ \qquad + m[z_g(\dot{u} - vr + wq) - x_g(\dot{w} - uq + vp)] = \sum M \\ I_{zz}\dot{r} + (I_{yy} - I_{xx})pq - (\dot{q} + rp)I_{yz} + (q^2 - p^2)I_{xy} + (rq - \dot{p})I_{xz} \\ \qquad + m[x_g(\dot{v} - wp + ur) - y_g(\dot{u} - vr + wq)] = \sum N \end{cases} \tag{8}$$

where

– u, v, w: surge, sway, heave velocities respectively

– p, q, r: roll, pitch, yaw rates

– X, Y, Z: external forces

– K, M, N: external moments

– x_g, y_g, z_g: center of gravity wrt origin at center of buoyancy

– I_{ab}: moments of inertia wrt origin at center of buoyancy (a and b symbolize x or y or z)

– m: vehicle mass

The sum of the forces and moments on the AUV can be expressed as:

$$\begin{cases} \sum X_{ext} = X_{HS} + X_{u|u|}u|u| + X_{\dot{u}}\dot{u} + X_{wq}wq + X_{qq}qq + X_{vr}vr \\ +X_{rr}rr + X_{prop} \\ \sum Y_{ext} = Y_{HS} + Y_{v|v|}v|v| + Y_{r|r|}r|r| + Y_{\dot{v}}\dot{v} + Y_{\dot{r}}\dot{r} + Y_{ur}ur \\ +Y_{wp}wp + Y_{pq}pq + Y_{uv}uv + Y_{uu\delta_r}u^2\delta_r \\ \sum Z_{ext} = Z_{HS} + Z_{w|w|}w|w| + Z_{q|q|}q|q| + Z_{\dot{w}}\dot{w} + Z_{\dot{q}}\dot{q} + Z_{uq}uq \\ +Z_{vp}vp + Z_{rp}rp + Z_{uw}uw \\ \sum K_{ext} = K_{HS} + K_{p|p|}p|p| + K_{\dot{p}}\dot{p} + K_{prop} \\ \sum M_{ext} = M_{HS} + M_{w|w|}w|w| + M_{q|q|}q|q| + M_{\dot{w}}\dot{w} + M_{\dot{q}}\dot{q} \\ +M_{uq}uq + M_{vp}vp + M_{rp}rp + M_{uw}uw \\ \sum N_{ext} = N_{HS} + N_{v|v|}v|v| + N_{r|r|}r|r| + N_{\dot{v}}\dot{v} + N_{\dot{r}}\dot{r} \\ +N_{ur}ur + N_{wp}wp + N_{pq}pq + N_{uv}uv + N_{uu\delta_r}u^2\delta_r \end{cases} \tag{9}$$

with the formulas of hydrostatic forces and moments:

$$\begin{cases} X_{HS} = -(W-B)\sin\theta \\ Y_{HS} = (W-B)\cos\theta\sin\phi \\ Z_{HS} = (W-B)\cos\theta\cos\phi \\ K_{HS} = (y_gW - y_bB)\cos\theta\cos\phi - (z_gW - z_bB)\cos\theta\sin\phi \\ M_{HS} = -(z_gW - z_bB)\sin\theta - (x_gW - x_bB)\cos\theta\cos\phi \\ N_{HS} = (x_gW - x_bB)\cos\theta\sin\phi + (y_gW - y_bB)\sin\theta \end{cases} \tag{10}$$

here

– X_{prop} : the thrusts of the thrusters projected on the corresponding axes

– K_{prop} : the steering moments made by the thrusters

– *W, B:* weight and buoyancy of the vehicle respectively

– The remaining factors are other nonlinear maneuvering coefficients of forces and moments [16].

3.4 Linearized Equations of Motion

3.4.1 Linearized Diving System Dynamics

The control system is decoupled in the vertical and horizontal directions, meaning that the diving motion is controlled independently with the steering motion. So, to design a diving controller, only the equations of vertical-plane motion need to be considered. The following assumptions can be made:

– $v = p = r = \dot{v} = \dot{p} = \dot{q} = 0$

– Surge velocity u is constant $u = U = 1.543\, m/s$

– Heave velocity w is small compared to the other terms.

– Moments of inertia are affected very little by the position change of the movable mass.

From Eq. (8), the lineared diving dynamics equations of motion are therefore [17]

$$\begin{cases} I_y \dot{q} + z_g W \theta = M_q q + M_{\dot{q}} \dot{q} - W x_g \\ \dot{\theta} = q \\ \dot{z} = -U\theta \end{cases} \tag{11}$$

where:

– W denotes the weight of the vehicle

– I_y moment of inertia about y-axis

– x_g vehicle center of gravity

The remaining factors are other nonlinear maneuvering coefficient of forces and moments. From theory of solid mechanics, it is easy to get the formula for calculating the vehicle center of gravity as:

$$x_g = \frac{m_m . x_m + m_o . x_o}{m} = \frac{m_m . x_m}{m} \tag{12}$$

where x_m denotes the position of the movable mass, x_0 is the position of the center of gravity of the vehicle excluding the movable mass (assuming that is zero). m_0 is the vehicle mass excluding m_m.

Finally, substituting Eq. (12) into Eq. (11), the lineared diving dynamics equations of motion are expressed as follows:

$$\begin{bmatrix} \dot{q} \\ \dot{\theta} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} M_q/(I_y - M_{\dot{q}}) & -Wz_g/(I_y - M_{\dot{q}}) & 0 \\ 1 & 0 & 0 \\ 0 & -U & 0 \end{bmatrix} \begin{bmatrix} q \\ \theta \\ z \end{bmatrix} + \begin{bmatrix} M_{x_m}/(I_y - M_{\dot{q}}) \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} x_m \end{bmatrix} \tag{13}$$

where $M_{x_m} = -W.m_m/m$

which we is simply expressed as:

$$\dot{x}_1 = A_1 x_1 + B_1 u_1 \tag{14}$$

3.4.2 Linearized Steering System Dynamics

Similarly, some state vectors related the diving-plane are considered as null for design a steering controller. The following assumptions can be made:

- $w = p = q = z = \phi = \theta = 0$
- Surge velocity u is constant $u = U = 1.543\, m/s$
- The vehicle is symmetric in its inertial properties.

From Eq. (8), the lineared steering dynamics equations of motion have been formed as follows [17]:

$$\begin{bmatrix} m - Y_{\dot{v}} & -Y_{\dot{r}} & 0 \\ -N_{\dot{v}} & I_z - N_{\dot{r}} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{v} \\ \dot{r} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} Y_v & Y_r - mU & 0 \\ N_v & N_r & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} v \\ r \\ \psi \end{bmatrix} + \begin{bmatrix} Y_\delta \\ N_\delta \\ 0 \end{bmatrix} \delta_r(t) \tag{15}$$

where $\delta_r(t)$ is the rudder angle of the AUV

which we is simply expressed as:

$$\dot{x}_2 = A_2 x_2 + B_2 u_2 \tag{16}$$

Chapter 4: Developed Navigation System of the AUV

4.1 Theoretical Background

4.1.1 Coordinate Frames

Three right-handed orthogonal coordinate frames were defined: a navigation frame (n-fixed), body frame (b-fixed), and earth frame (e-fixed). The navigation frame was defined as the local horizontal coordinates, with the origin as the center of mass and the other three axes facing north, east, and the vertical lower side. The D-axis is perpendicular to the earth's ellipsoid, the E-axis points directly east, and the N-axis points directly north, as shown in Fig. 4.1.

Frame symbol	Description
I	Inertial
E	Earth-fixed
N	North-East-Down (NED)
B	Body-fixed (as N, but not North-aligned)

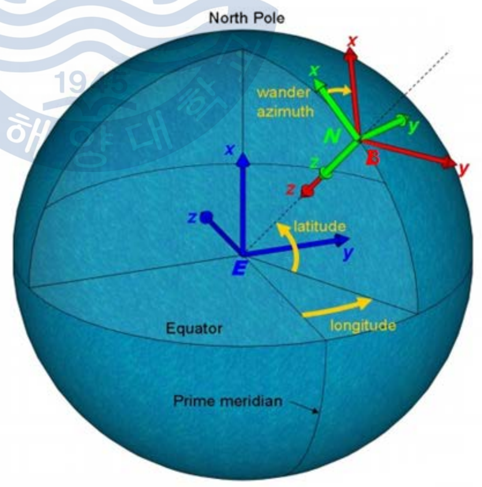

Fig. 4.1 Navigation Frame of UUVs

A body frame is a coordinate system in which the center resides at the center of mass in the AUV, and it is defined such that its X-axis pointing forward is the roll, its Y-axis pointing to the right is the pitch, and its Z-axis pointing downward is the yaw of the AUV, as shown in Fig. 4.2.

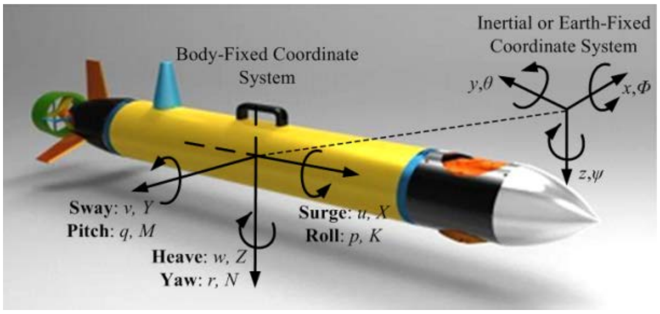

Fig. 4.2 Body Frame of AUV

Various mathematical representations can be used to define the tilt and heading of a body frame with respect to a navigation frame such as Euler angles and quaternions.

According to Euler's theorem, we can specify the orientation of the body frame relative to the navigation frame using three angles (ϕ,θ,ψ), known as Euler angles which can be obtained using three successive rotations about different axes, known as the Euler angle sequence. These three rotations may be expressed mathematically using the following three separate direction cosine matrices, respectively [18]:

$$C_z = \begin{pmatrix} c_\psi & s_\psi & 0 \\ -s_\psi & c_\psi & 0 \\ 0 & 0 & 1 \end{pmatrix}, C_y = \begin{pmatrix} c_\theta & 0 & -s_\theta \\ 0 & 1 & 0 \\ s_\theta & 0 & c_\theta \end{pmatrix}, C_x = \begin{pmatrix} 1 & 0 & 0 \\ 0 & c_\phi & s_\phi \\ 0 & -s_\phi & c_\phi \end{pmatrix} \tag{17}$$

Thus, the transformation from the navigation to body axes may be expressed as the product of these three separate transformations as follows:

$$\begin{aligned} C_n^b &= C_x C_y C_z \\ &= \begin{pmatrix} c_\psi c_\theta & s_\psi c_\theta & -s_\theta \\ -s_\psi c_\phi + c_\psi s_\theta s_\phi & c_\psi c_\phi + s_\psi s_\theta s_\phi & c_\theta s_\phi \\ s_\psi s_\phi + c_\psi s_\theta c_\phi & s_\psi s_\theta c_\phi - c_\psi s_\phi & c_\theta c_\phi \end{pmatrix} \end{aligned} \tag{18}$$

Similarly, the inverse transformation from the body to navigation axes is given by:

$$C_b^n = (C_n^b)^T = (C_x)^T (C_y)^T (C_z)^T \tag{19}$$

According to the quaternion theorem, we can specify a four-dimensional complex number (q_0, q_1, q_2, q_3) that can be used to represent the orientation of a rigid body or coordinate frame in three-dimensional space. The transformation between quaternions and Euler angles is shown below [18]:

$$\begin{pmatrix} \phi \\ \theta \\ \psi \end{pmatrix} = \begin{pmatrix} a\tan 2(2(q_0q_1+q_2q_3),1-2(q_1^2+q_2^2)) \\ a\sin(2(q_0q_2-q_3q_1)) \\ a\tan 2(2(q_0q_3+q_1q_2),1-2(q_2^2+q_3^2)) \end{pmatrix} \tag{20}$$

$$\begin{aligned} q &= \begin{pmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{pmatrix} \\ &= \begin{pmatrix} \cos(\phi/2)\cos(\theta/2)\cos(\psi/2)+\sin(\phi/2)\sin(\theta/2)\sin(\psi/2) \\ \sin(\phi/2)\cos(\theta/2)\cos(\psi/2)-\cos(\phi/2)\sin(\theta/2)\sin(\psi/2) \\ \cos(\phi/2)\sin(\theta/2)\cos(\psi/2)+\sin(\phi/2)\cos(\theta/2)\sin(\psi/2) \\ \cos(\phi/2)\cos(\theta/2)\sin(\psi/2)-\sin(\phi/2)\sin(\theta/2)\cos(\psi/2) \end{pmatrix} \end{aligned} \tag{21}$$

4.1.2 Initial Alignment Algorithm

Because the using three-axis angular velocity sensor was not equipped with a function to measure the angular velocity of the earth's rotation, an initial alignment using the geomagnetism sensor and acceleration sensor was required. In this dissertation, the roll and pitch angles were calculated by using the output of the accelerometer, which included gravitational acceleration information acquired during the initial alignment. Then, the yaw angle was calculated using the 3-axis geomagnetic sensor data, which included the calculated roll, pitch angles, and vector value of geomagnetism.

a/ Tilt determination using accelerometer

Because the accelerometer in this application is fixed to the body, it provides a measurement of the specific force in the b-frame, denoted as $f_b = \begin{bmatrix} f_{xb} & f_{yb} & f_{zb} \end{bmatrix}^T$. Thus, we have [19]

$$f_b = C_n^b(a_n - g^n) \tag{22}$$

where $a_n = \begin{bmatrix} a_{xn} & a_{yn} & a_{zn} \end{bmatrix}^T$ is the acceleration of the body in the n-frame, and $g^n = \begin{bmatrix} 0 & 0 & g \end{bmatrix}^T$ is the mass attraction navigation vector, with g=9.81 m/s2. If we denote x as the third column of C_n^b, rearranging Eq.(22), we have:

$$f_b = C_n^b a_n - xg \tag{23}$$

To calculate the roll and pitch angles, assuming that the UUV is not moving, $a_n = 0$, we can write:

$$f_b = -xg \tag{24}$$

From Eq.(24), the tilt angles are calculated as:

$$\begin{cases} \theta = \sin^{-1}(\dfrac{f_{xb}}{g}) \\ \phi = \sin^{-1}(\dfrac{-f_{yb}}{g\cos\theta}) \end{cases} \tag{25}$$

It should be noted that the tilt determination using Eq.(25) is true only if the body is not accelerating. When the body is accelerating or vibrating, the tilt angles calculated from the accelerometer measurements are erroneous.

b/ Heading Determination Using Magnetometer

As shown in Eq. (22), multiplication of the vector $(a_n - g^n)$ by transformation matrix C_n^b causes the disappearance of the yaw angle values. For this reason, another sensor other than the one for the acceleration is required. To solve this problem, an initial alignment using the geomagnetism sensor and acceleration sensor was executed.

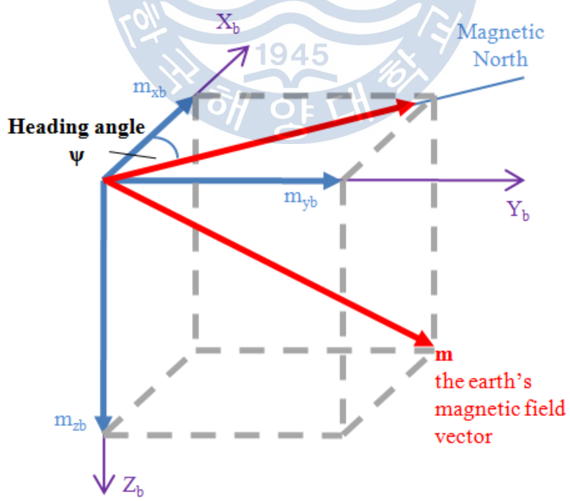

Fig. 4.3 Identify heading angle by measuring components of the earth's magnetic field vector

To determine the heading angle of the system from the earth's magnetic field vector measurement, we have to know the components of the vector in the n-frame, denoted as $m_n = \begin{bmatrix} m_{xn} & m_{yn} & m_{zn} \end{bmatrix}^T$ as illustrated in Fig. 4.3. However, in our application, the magnetometer is fixed to the UUV's body. It provides a measurement of the earth's magnetic field vector in the b-frame, denoted as $m_b = \begin{bmatrix} m_{xb} & m_{yb} & m_{zb} \end{bmatrix}^T$. Hence, the measured vector has to be transformed from the b-frame into the n-frame by applying the rotational equation shown below [19]:

$$\begin{pmatrix} m_{xb} \\ m_{yb} \\ m_{zb} \end{pmatrix} = C_n^b \begin{pmatrix} m_{xn} \\ m_{yn} \\ m_{zn} \end{pmatrix} \tag{26}$$

For the convenience of the calculation to obtain the yaw angle, C_n^b was divided into the C_1 and C_2 matrices, which included the roll, pitch and yaw angles respectively. Thus Eq. (26) becomes:

$$m_b = C_1 C_2 m_n \tag{27}$$

with:

$$C_1 = \begin{pmatrix} c_\theta & 0 & -s_\theta \\ s_\phi s_\theta & c_\phi & s_\phi c_\theta \\ c_\phi s_\theta & -s_\phi & c_\phi c_\theta \end{pmatrix}, C_2 = \begin{pmatrix} c_\psi & s_\psi & 0 \\ -s_\psi & c_\psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{28}$$

Eq. (27) can be expressed in another way as below:

$$C_1^{-1} \begin{pmatrix} m_{xb} \\ m_{yb} \\ m_{zb} \end{pmatrix} = C_2 \begin{pmatrix} m_{xn} \\ m_{yn} \\ m_{zn} \end{pmatrix} \tag{29}$$

According to Eq.(29), we have

$$\begin{cases} c_\psi m_{xn} + s_\psi m_{yn} = c_\theta m_{xb} + s_\phi s_\theta m_{yb} + c_\phi s_\theta m_{zb} \\ -s_\psi m_{xn} + c_\psi m_{yn} = c_\phi m_{yb} - s_\phi m_{zb} \end{cases} \tag{30}$$

For convenience of the calculation, if the substitution is made as $c_\theta m_{xb} + s_\phi s_\theta m_{yb} + c_\phi s_\theta m_{zb} = X$, $c_\phi m_{yb} - s_\phi m_{zb} = Y$, Eq. (30) is simplified to:

$$\begin{pmatrix} m_{xn} & m_{yn} \\ m_{yn} & -m_{xn} \end{pmatrix} \begin{pmatrix} c_\psi \\ s_\psi \end{pmatrix} = \begin{pmatrix} X \\ Y \end{pmatrix} \tag{31}$$

Using Eq. (31), the yaw angle can be expressed as a closed-form as:

$$\tan(\psi) = \frac{s_\psi}{c_\psi} = \frac{-m_{yn} X + m_{xn} Y}{-m_{xn} X - m_{yn} Y} \tag{32}$$

$$
\begin{aligned}
\psi &= \tan^{-1}\left(\frac{-m_{yn}X + m_{xn}Y}{-m_{xn}X - m_{yn}Y}\right) \\
&= \tan^{-1}\left(\frac{-m_{yn}(c_\theta m_{xb} + s_\phi s_\theta m_{yb} + c_\phi s_\theta m_{zb}) + m_{xn}(c_\phi m_{yb} - s_\phi m_{zb})}{-m_{xn}(c_\theta m_{xb} + s_\phi s_\theta m_{yb} + c_\phi s_\theta m_{zb}) - m_{yn}(c_\phi m_{yb} - s_\phi m_{zb})}\right) \\
&= \tan^{-1}\left(\frac{-m_{yn}c_\theta m_{xb} - m_{yn}s_\phi s_\theta m_{yb} - m_{yn}c_\phi s_\theta m_{zb} + m_{xn}c_\phi m_{yb} - m_{xn}s_\phi m_{zb}}{-m_{xn}c_\theta m_{xb} - m_{xn}s_\phi s_\theta m_{yb} - m_{xn}c_\phi s_\theta m_{zb} - m_{yn}c_\phi m_{yb} + m_{yn}s_\phi m_{zb}}\right)
\end{aligned}
\tag{33}
$$

As mentioned above, this method has good performance when the body is static but becomes very erroneous when the body is accelerating or vibrating. An error in the tilt angles measurement would result in tilt compensation errors, and thus, heading angle error. Another problem with determining the heading angle using magnetometer data is the existence of magnetic interferences. Compensating for the error caused by magnetic interference is not a trivial task.

4.1.3 The Discrete Kalman Filter Algorithm

The Kalman filter estimates a process by using a form of feedback control: the filter estimates the process state at some time and then obtains feedback in the form of (noisy) measurements [20], [21]. As such, the equations for the Kalman filter fall into two groups: time update equations and measurement update equations. The time update equations are responsible for projecting forward the current state and error covariance estimates to obtain the a priori estimates for the next time step. The measurement update equations are responsible for the feedback. That mean it incorporates a new measurement into the a priori estimate to obtain an improved a posteriori estimate. The time update equations can also be thought of as predictor equations, while the measurement update equations can be thought of as corrector equations. Indeed the final estimation algorithm resembles that of a predictor-corrector algorithm for solving numerical problems.

The specific equations for the time and measurement updates are presented in Fig.4.4. The time update equations project the state and covariance estimates forward from time step (k-1) to step k. The nxn matrix A in the difference equation (1) of time update equations relates the state at the previous time step k-1 to the state at the current step k. The nxl matrix B relates the optional control input $u \in R^l$ to the state x. And Q is the process noise covariance that changes with each time step or measurement. In the measurement update equations, the first task is to compute the Kalman gain K_k. The next step is to actually measure the process to obtain z_k and then to generate an a posteriori state estimate $\hat{x}_k$ by incorporating the measurement as in equation (2). The final step is to obtain an a posteriori error covariance estimate P_k via equation (3).

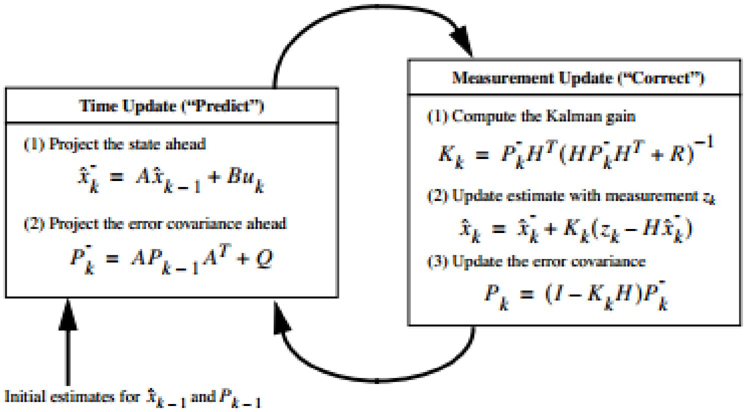

Fig. 4.4 The Discrete Kalman Filter Algorithm

After each time and measurement update pair, the process is repeated with the previous a posteriori estimates used to project or predict the new a priori estimates.

4.2 Development of AHRS System

4.2.1 AHRS System Design

For the performance test of the developed AHRS system, a three-axis gimbal device shown in Fig. 4.5 was set up, with the developed AHRS installed to compare with the state outputs of the MTi product (pitch, roll, and yaw angles).

Fig. 4.5 Estimation system on three-axes gimbal structure

A low-cost hardware system using a TMS320F28335 microprocessor and an ADIS16045 (IMU sensor) was constructed as shown in Fig. 4.6. Here, with a high processing speed reaching up 150 MHz and a floating point calculating ability, the

TMS320F28335 could easily reach a sample frequency of 100 Hz (quite similar to MTi's frequency). The ADIS1605 provided a simple, cost effective method for integrating accurate, multi-axis inertial sensing into automatic systems that includes a tri-axis gyroscope, tri-axis accelerometer, and tri-axis magnetometer. The specifications for the ADIS16405 and MTi are indicated in Table 3. The signal processing sensed from the IMU sensor is estimated in real time using the Kalman filter algorithm. Then it will be synthesized with MTi's data for transmission to a monitoring interface programmed by using Visual Studio 2008. The interface includes the functions such as real-time graph indicate the speed, altitude, heading, turn rate, etc.

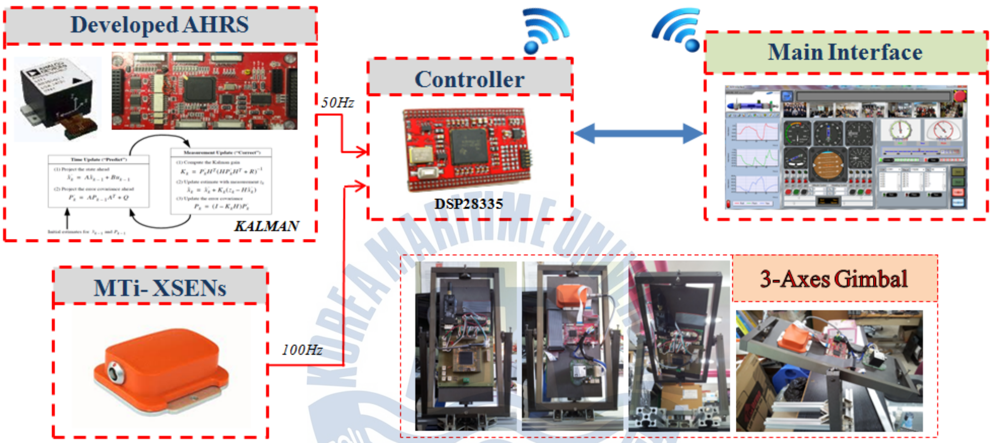

Fig. 4.6 Hardware architecture of developed AHRS using Kalman filter

Table 3. ADIS16405 & MTi specifications

<table>
<tr><th colspan="3">ADIS16405</th><th colspan="2">MTi</th></tr>
<tr><td rowspan="3">Output</td><td colspan="2">3 axis angular velocity</td><td colspan="2">Roll, Pitch, Yaw</td></tr>
<tr><td colspan="2">3 axis acceleration</td><td colspan="2" rowspan="2">3 axis angular velocity, acceleration, geomagnetism</td></tr>
<tr><td colspan="2">3 axis geomagnetism</td></tr>
<tr><td rowspan="5">Specifications</td><td>Angular velocity noise (3D)</td><td>0.9°</td><td>Roll/Pitch (static)</td><td>0.5°</td></tr>
<tr><td>Acceleration noise (3D)</td><td>9 mg</td><td>Yaw (static)</td><td>1°</td></tr>
<tr><td>Geomagnetism noise (3D)</td><td>1.25 mgauss</td><td>Dynamic accuracy</td><td>5°</td></tr>
<tr><td>Sample frequency</td><td>100Hz</td><td>Sample frequency</td><td>100Hz</td></tr>
<tr><td>Communication</td><td>SPI</td><td>Communication</td><td>Serial</td></tr>
</table>

The Fig. 4.7 represents the algorithm flow diagram of developed AHRS system. The Kalman filter needs the initial estimated input which are as accuracy as possible. Hence in first 10 seconds, just calculate the roll, pitch, yaw angles by using Low-pass filter and then send to the Quaternion from Euler angles block. After that, apply Kalman filter that includes two steps: predictor equations and corrector equations to get the real attitude values. Finally, convert these values from Quaternion to Euler angles.

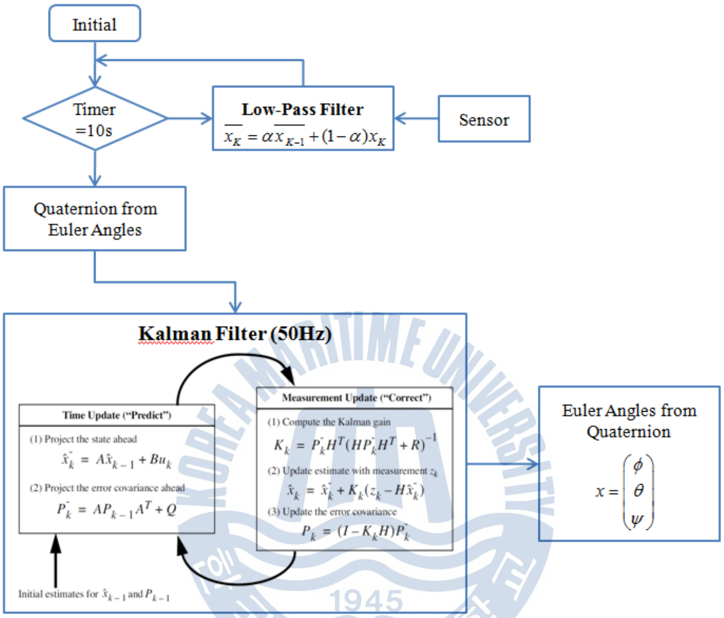

Fig. 4.7 Algorithm flow diagram of AHRS system

4.2.2 Simulation and Experiments

4.2.2.1 Simulation

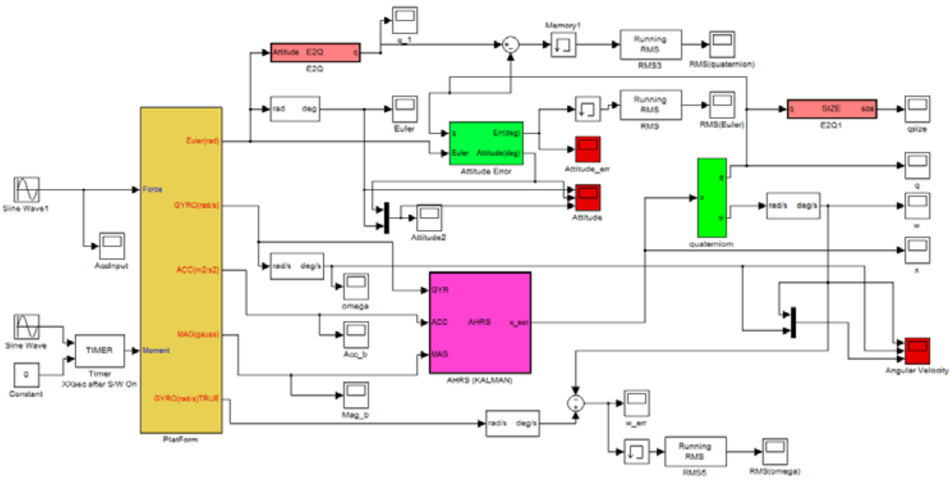

Fig. 4.8 Matlab Simulink of AHRS

For the performance test of the AHRS algorithm, a simulink program was designed in Matlab as shown in Fig. 4.8. For the simulation, input values were used for the simulation program: 10 N for the x, y, and z-axes and the force as well as the moment force, which has a sine wave form with 0.69 Nm toward the x-axis, 0.66 Nm toward the y-axis, and 0.36 Nm toward the z-axis were used. For the angular velocity, acceleration, and geomagnetism sensors, the modeled sensors had an average noise level of 0 and noise covariance values of 2.54 E-4, 8.0 E1, 1.56 E0, respectively. Fig. 4.9 shows the simulation results, where the top graph represents the actual values, and the middle graph indicates the values estimated using the Kalman filter. The lowest graph shows the simultaneous expression of the estimated and actual values. As shown in the third graph, the values estimated using Kalman filter were quite close to the actual values, with the error values shown in Fig. 4.10. This AHRS has good performance when the body is static. However, there are small errors due to translational acceleration.

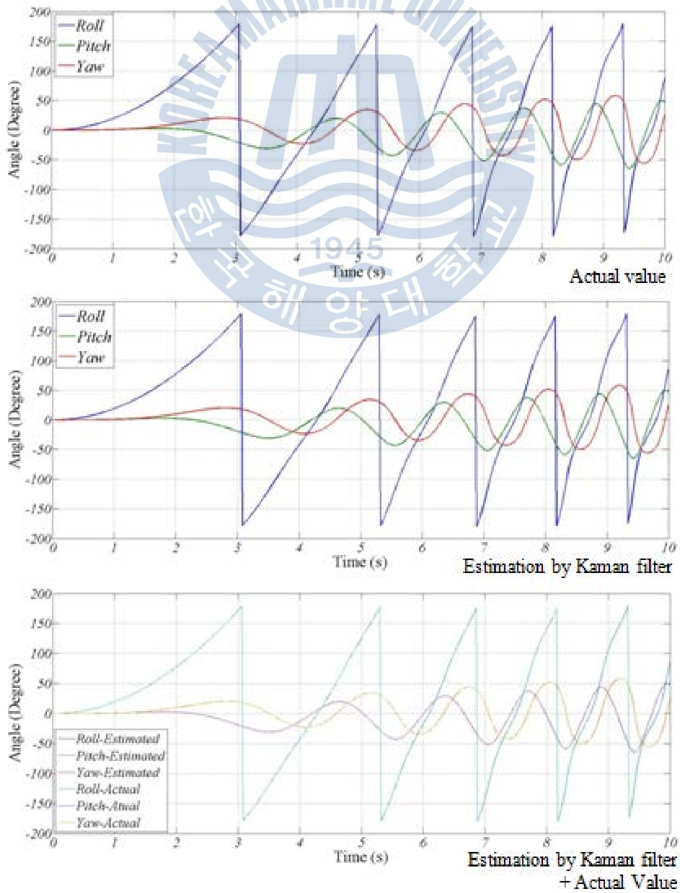

Fig. 4.9 Simulation results for attitude & heading using Kalman filter

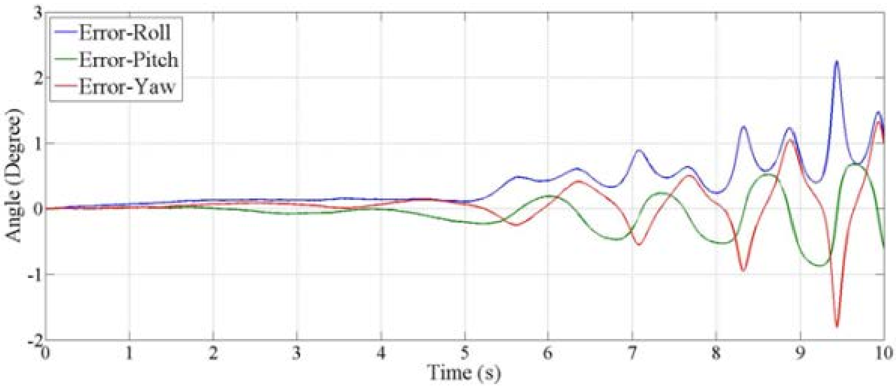

Fig. 4.10 Simulation results for attitude & heading error

4.2.2.2 *Experiments*

Two performance tests were carried out and compared with state data of the MTi product [22]. The first one tested the estimation performance of the developed AHRS for the roll, pitch, and yaw angles under stationary conditions. The second one tested the estimation performance for the states under rotational motion around x-axis and y-axis.

a/ Stationary test

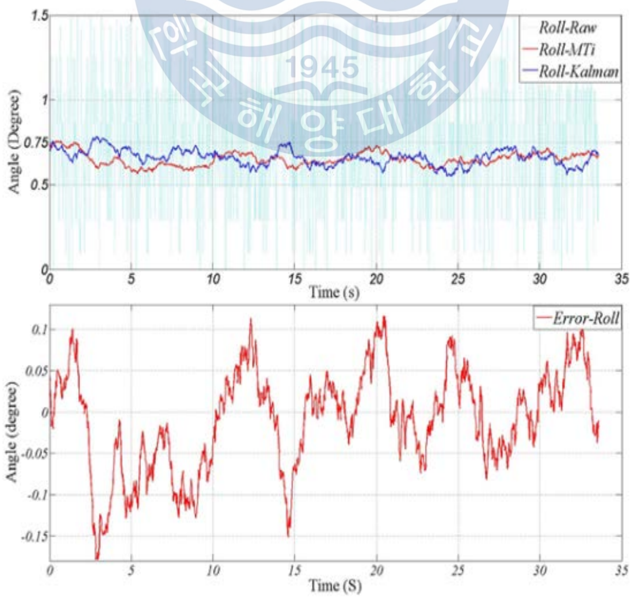

Fig. 4.11 Roll and error values when stationary

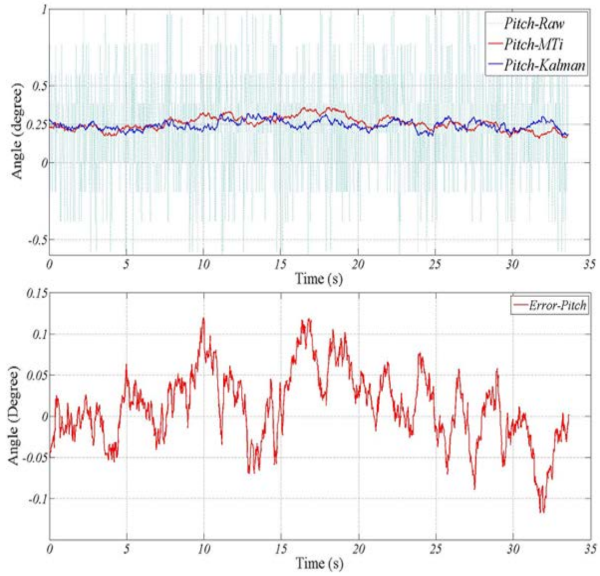

Fig. 4.12 Pitch and error values when stationary

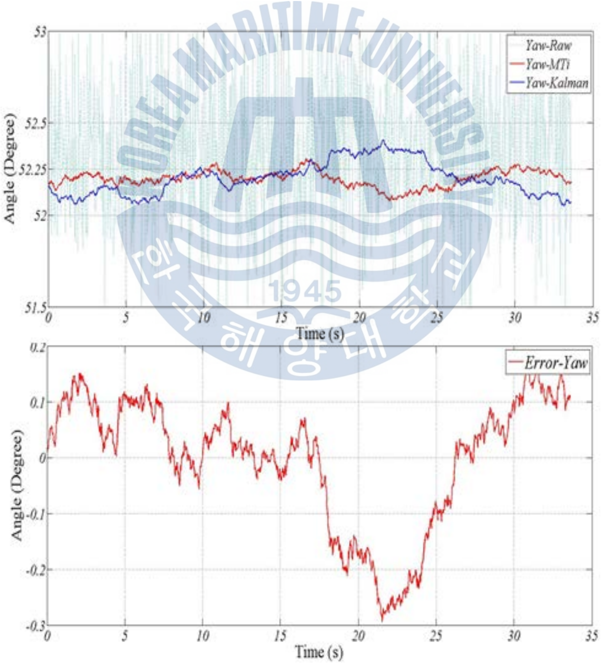

Fig. 4.13 Yaw and error values when stationary

Figs.4.11~4.13 show the test results for the three angles (Roll, Pitch, Yaw) estimated by using the Kalman filter, the three angles read from the MTi AHRS, and the three angles calculated directly from the raw signals received from the IMU sensor without using Kalman filter. From these figures, it is shown that the Kalman filter filters out noises such that the developed AHRS estimates the output states similarly with those of the MTi AHRS. The roll angle estimates have an error (compared with reference) range of 0.25 degrees, the pitch angle has an error range of 0.20 degrees, and the yaw error is 0.50 degrees. It means that the obtained tilt and heading information is reliable over long period of time.

b/ Rotation test

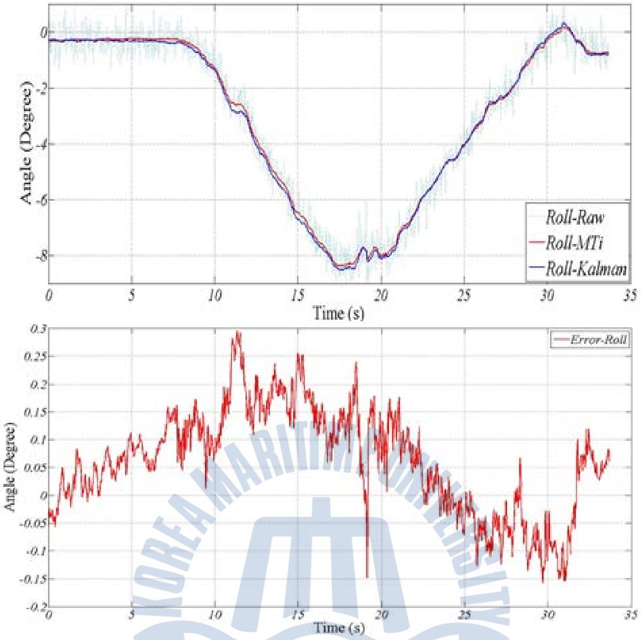

Fig. 4.14 Roll and error values when rotating

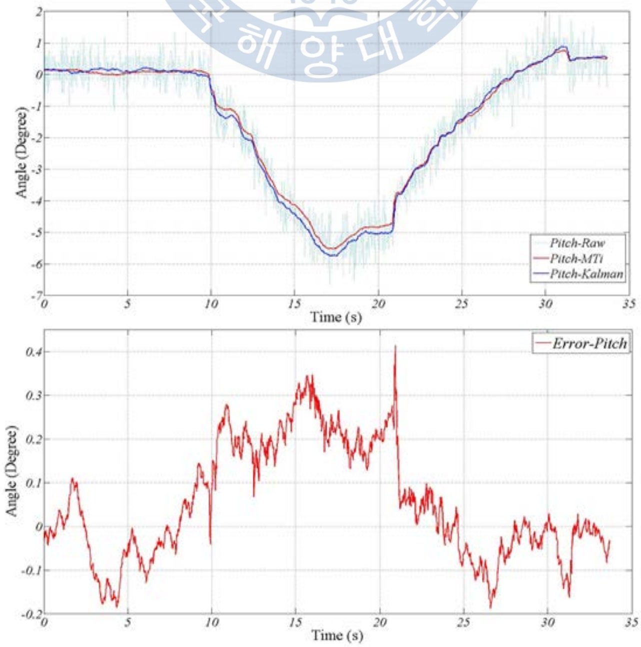

Fig. 4.15 Pitch and error values when rotating

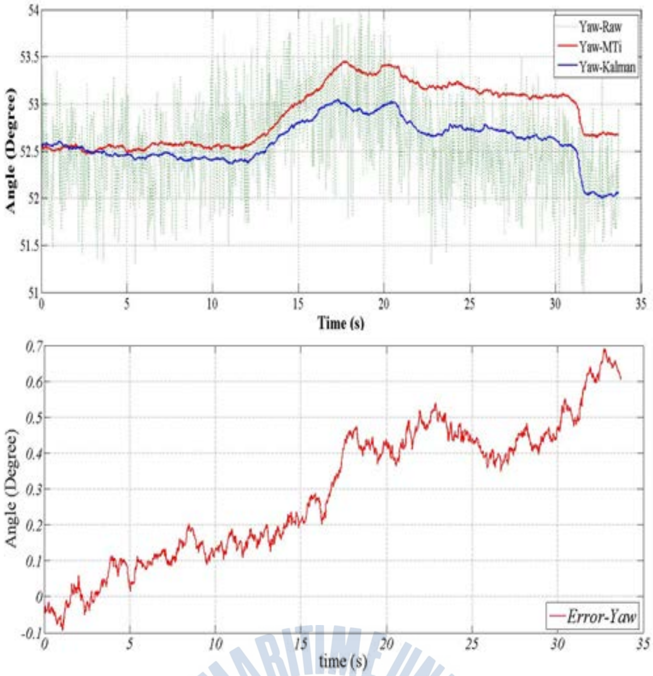

Fig. 4.16 Yaw and error values when rotating

Figs.4.14~4.16 show the test results for the three angles estimated by using the Kalman filter, the three angles read from the MTi AHRS, and the three angles calculated directly from the raw signals received from the IMU sensor without using the Kalman filter. The roll angle estimates have an error (compared with reference) range of 0.450 degrees, the pitch angle has an error range of 0.60 degrees, and the yaw error is 0.80 degrees. These errors are larger than the previous ones for the stationary test. This is because of the accumulative attitude errors when translational acceleration exists. The reason was mentioned in a previous section. However, these errors are small and acceptable compared with MTi's specifications, as shown in Table 3.

4.3 Development of Navigation System for the AUV

4.3.1 Sensor System Design for Navigation

In navigation of the AUV, a sensor system composed of a number of sensors, states estimation algorithm, and control algorithm are required. As one of the important elements, the inertial navigation system (INS) based on the IMU sensor estimates the position, velocity, and attitude of the vehicle, from the measured acceleration information of the vehicle. To compensate for accumulative errors of the INS resulting from inherent drift of dead-reckoning velocities and integration of acceleration, a Kalman filter algorithm, combined with sensors (GPS, DVL, pressure sensor), is developed.

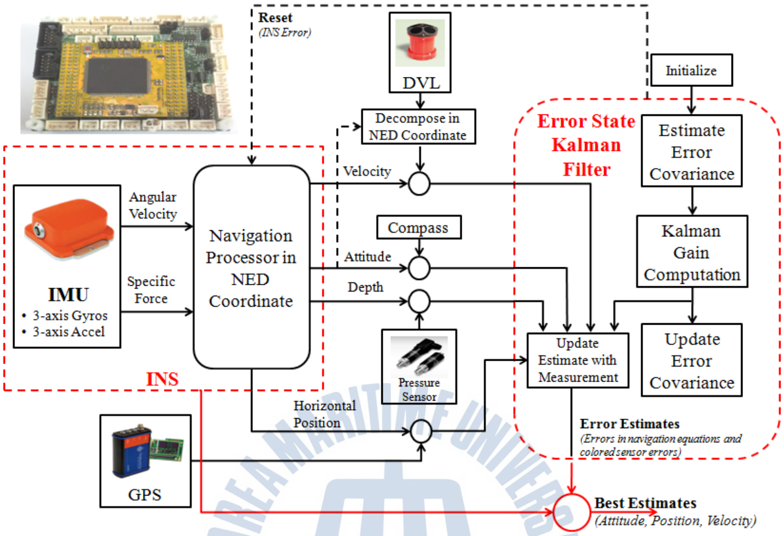

Fig. 4.17 Block diagram of the developed navigation algorithm

The Kalman filter algorithm is composed of four blocks: error covariance, Kalman gain computation, update error covariance, and update estimate with measurement. The block diagram of the developed INS algorithm for the AUV control system including Kalman filter is presented in Fig. 4.17. The DVL with accuracy ±1 mm/s is applied to compensate the position error growth in long submerged operations of the vehicle. The GPS, which has static accuracy of less than 3 meter, provides the position update of the AUV, when it floats on the surface. In addition, the attitude and depth information are provided by the pressure and compass sensors. To verify the developed navigation algorithm, experiments using GPS-INS and DVL-INS were performed on ground and in water tank. The results are presented in below section.

4.3.2 Navigation Experiments for the AUV

a/ GPS-INS Experiment

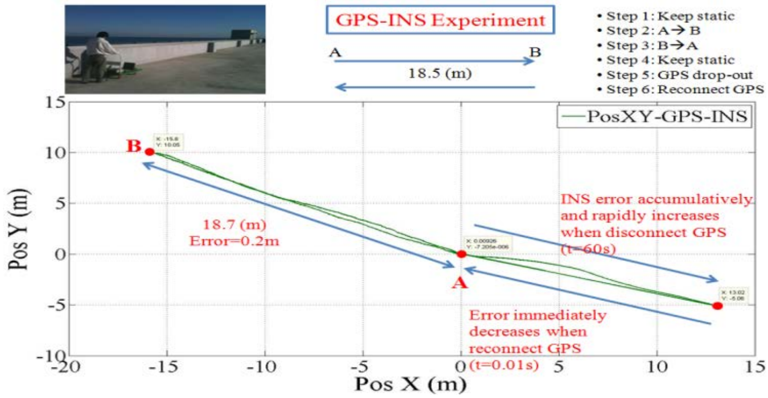

Fig. 4.18 Waypoint tracking experiment result on the XY plane

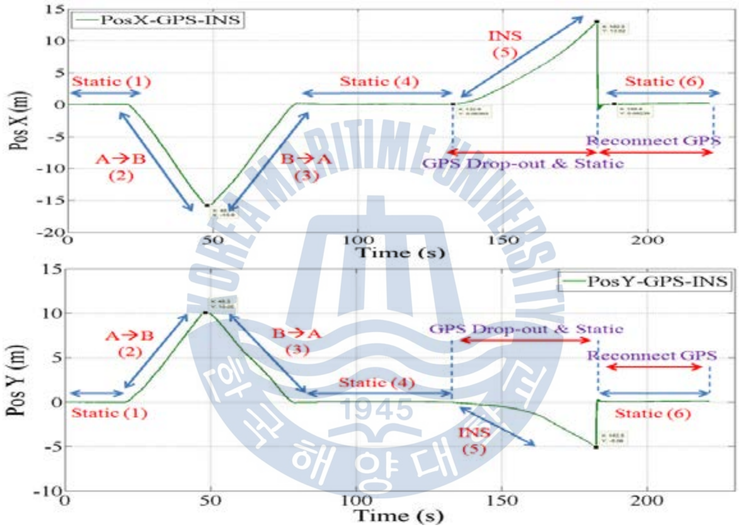

Fig. 4.19 X-position and Y-position values of the waypoint tracking experiment

The GPS-INS system is composed of the IMU, compass, and GPS sensors. Its position is estimated by using the INS and Kalman filter algorithm. The experiments include static positioning and dynamic tracking performances that are the main effects of applied sensors on the positioning accuracy of the vehicle.

The experiments of sensor system for developed navigation algorithm are performed on the ground before the sea experiment. A man moves a cart equipped with the developed GPS-INS system along two waypoints (A and B), already known on the ground. The AB distance is approximately 18.5 m. As a result, when the AUV performs motions (2nd, 3rd steps in Fig. 4.18 and Fig. 4.19), the maximum dynamic position error is 0.2m, which is relatively small in the motion of the vehicle. When the AUV is kept in static states (1st, 4th, 6th steps) for a long-term period, the position errors are nearly zero. Hence, from the two above performances, although the used GPS has low position accuracy (the horizontal error is 1.7 m and vertical error is 3 m), we can easily recognize that the position errors decrease a lot. But if the GPS sensor is off (the IMU is applied alone), as in the 5th step, the position error of the INS accumulatively and rapidly increases. However, as the INS has high frequency, and notwithstanding the accumulative errors mentioned above, it can be used as a short-term (around a few seconds, less than 5 s) fallback, while the GPS signals are unavailable, due to bad weather or obstruction. At the 6th step, it shows that the position errors immediately decrease a lot after the system received GPS signals, and quickly stabilize at small position errors. According to the experimental results, it is concluded that the position errors are bounded in the permitted range, when both GPS and IMU are utilized through the developed INS and Kalman filter algorithms.

b/ DVL-INS Experiment

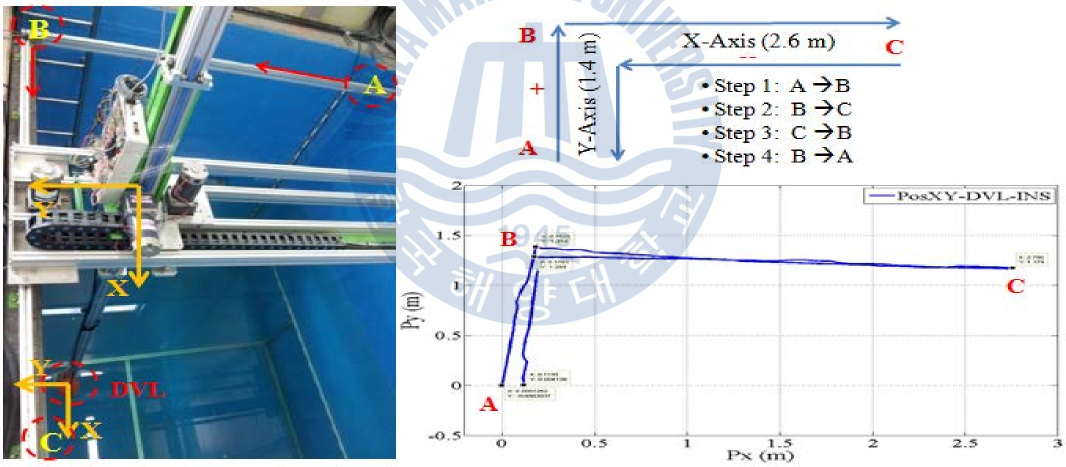

Fig. 4.20 DVL-INS experiment in a water tank

Importantly, in long-term submerged operation of the AUV, the GPS signals are unavailable in sea water. So the navigation of the vehicle depends entirely on the sensor accuracies (DVL, IMU, pressure sensor), and especially the Kalman filter described in above section. To verify the developed navigation algorithm, a DVL-INS experiment was performed in a water tank. The DVL-INS system is composed of the IMU, compass, and DVL sensors. The coordinates of the DVL and IMU are aligned to coincide with directions of a 3-axis motion system which can move along X, Y, and Z axis, as shown in Fig. 4.20. The DVL-INS system was moved along three waypoints (A, B, and C) previously known in the water tank.

The distance between point A and B is 1.4 m and the distance between point B and C is 2.6 m. Four steps of experiments were performed with starting from waypoint A to B, C, and finally return A again. With using the developed navigation algorithm combined with the DVL signals, the accumulative errors of the INS are compensated. In 1st and 2nd steps, from waypoint A (0, 0) to B (0, 1.4), C (2.6, 1.4), respectively the estimated positions are (0.0001, 0.0002), (0.182, 1.38), and (2.756, 1.174). After that, in 3rd and 4th steps, the system was moved in return direction. The estimated positions from C to B and A respectively are (2.756, 1.174), (0.176, 1.289), and (0.119, 0.004). In general, these values are quite accurate with small position estimation errors of about 0.2 m.

Chapter 5: Control Algorithm Design

5.1 Sliding Mode Controller Design

An underwater vehicle operates with six degrees of freedom, highly nonlinear, and must respond to influences of hydrostatic and hydrodynamic forces from the ever changing environment of the ocean. Additionally, an AUV must avoid obstacles in the ocean environment such as the changes in the sea floor depth, mines, shipwreck, reefs, etc. In this reason, feedback controllers are required with

AUV to provide autopilot functions that mean it can properly maintain depth, tracking, and obstacle avoidance during an AUV's mission. Feedback controllers must be robust enough to account for changes in ocean current and changes in ocean floor depth. To control the highly responsive AUV, a sliding mode controller (SMC) has been developed and applied. The SMC is a robust state feedback controller for uncertain dynamic systems and disturbances from the underwater environment. It allows n^{th} order system to be effectively replaced by a (n-1) order system and makes system states stay in a switching surface on which the system remains insensitive to internal parameter variations and extraneous disturbances [23].

Generally, the multivariable sliding mode control methods are used with predominantly linear system models as opposed to the SMC methods used for nonlinear systems [24]. So the lineared model of AUV for diving-plane and steering-plane can be written as single-input multi-output model as:

$$\dot{x} = Ax + Bu + d \tag{34}$$

Now, sliding surface can be defined in the error state space form as follows

$$S_s = S^T \tilde{x} = [s_1 \dots s_n]\tilde{x} = 0 \tag{35}$$

where $\tilde{x} = x - x_d$ are state errors, x_d is desired tracking state.

The positive definite Lyapunov function candidate is

$$V(S_s) = \frac{1}{2} S_s^T S_s = \frac{1}{2} \tilde{x}^T S S^T \tilde{x} \tag{36}$$

$V(S_s)$ guarantees that the error state converges to sliding surface exists if following condition is satisfied

$$\dot{V}(S_s) = S_s \dot{S}_s < 0 \tag{37}$$

If we select

$$\dot{S}_s = -\eta sign(S_s) \quad \eta > 0 \tag{38}$$

Differentiating the sliding surface in Eq. 35

$$\dot{S}_s = S^T \dot{\tilde{x}} = S^T (Ax + Bu + d - \dot{x}_d) = -\eta sign(S_s) \tag{39}$$

Extracting control input u from Eq. 39

$$u = -K^T x + (S^T B)^{-1} [S^T \dot{x}_d - S^T \hat{d} - \eta sign(S_s)] \tag{40}$$

with $K^T = (S^T B)^{-1} S^T A$

From Eq. 39, it can be rewritten as follows

$$\dot{S}_s = S^T A_c x - \eta sign(S_s) + S^T (\hat{d} - d) \tag{41}$$

where $A_c = A - BK^T$

Choosing S as the eigenvector of A_c^T for eigenvalue $\lambda = 0$ that is

$$\dot{S}_s = -\eta sign(S_s) + S^T (\hat{d} - d) \tag{42}$$

Differentiation of Lyapunov function candidate is

$$\begin{aligned} \dot{V}(S_s) &= -\eta S_s sign(S_s) + S_s S^T (\hat{d} - d) \\ &= -\eta |S_s| + S_s S^T (\hat{d} - d) \end{aligned} \tag{43}$$

Error state converging to sliding surface exists if η is chosen as follows

$$\eta > \| S^T \| \| \hat{d} - d \| \tag{44}$$

To prevent chattering because of the discontinuity in control law with switching function $sign(S_s)$, the sign function could be replaced by the continuous functions, where ϕ is the sliding surface boundary layer thickness.

5.2 Heading Control & Waypoint Tracking

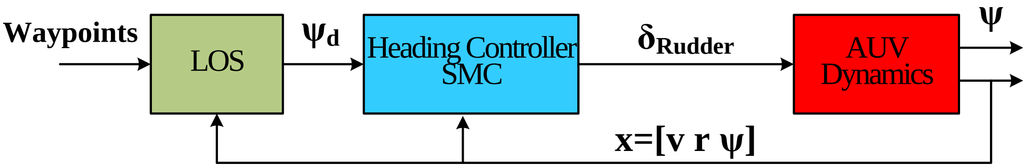

Fig. 5.1 Block diagram of waypoints tracking with heading control

In Fig. 5.1, a block diagram of waypoints tracking using line of sight (LOS) algorithm and heading controller is described. The desired heading ψ_d is calculated by LOS algorithm using a set of waypoints and feedback values of the AUV states $x = [v\ r\ \psi]$. And then, with the input value of ψ_d, the heading controller will estimate a rudder angle δ_{rudder} that is input into AUV nonlinear dynamics to control the AUV.

Substituting the hydrodynamic coefficients [15] of the AUV into Eq. (16), the sliding poles of the steering control system selected arbitrarily at (-1.4, -1.55, 0), and then using the Eq. 40 and Eq. 44, we get the steering control law as:

$$\begin{cases} S_{s2} = [0.056 \quad 0.477 \quad 0.877]\tilde{x}_2 \\ u_2 = \delta_R = -0.8v + 0.682r + 1.868\eta_2 \tanh(S_{s2}/\phi_2) \\ -\pi/6 \le u_2 \le \pi/6 \\ \eta_2 = 0.5, \quad \phi_2 = 0.1 \end{cases} \tag{45}$$

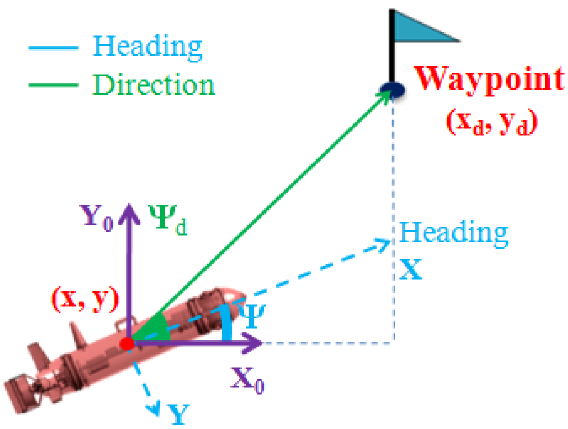

Fig. 5.2 Coordinate systems for waypoints tracking

For the AUV to follow a certain destination by way of desired points, a tracking algorithm is necessary. In this dissertation, a well known LOS algorithm is utilized. If the AUV mission is given to pass through a set of waypoints, the waypoints are defined as $[x_d(k), y_d(k)]$ for (k=1…N). If the desired heading angle is only changed at each waypoint, some overshoot will be observed, when changing waypoint. One of the algorithms to generate a smooth reference trajectory is LOS [16]. In the LOS algorithm, the desired heading angle shown in Fig. 5.2 is defined as

$$\psi_d(t) = tan^{-1}\left(\frac{y_d(k) - y(t)}{x_d(k) - x(t)}\right) \quad (46)$$

After the quadrant check is performed with the desired heading angle, the next waypoint can be selected on the basis of vehicle's residence within a circle of acceptance with radius ρ_0 around the waypoint $[x_d(k), y_d(k)]$. If the vehicle location $[x(t), y(t)]$ at the time t satisfies the condition:

$$[x_d(k) - x(t)]^2 + [y_d(k) - y(t)]^2 \le \rho_0^2 \quad (47)$$

then the next waypoint $[x_d(k+1), y_d(k+1)]$ is selected. A guideline can choose ρ_0 equal to vehicle length.

5.3 Depth Control

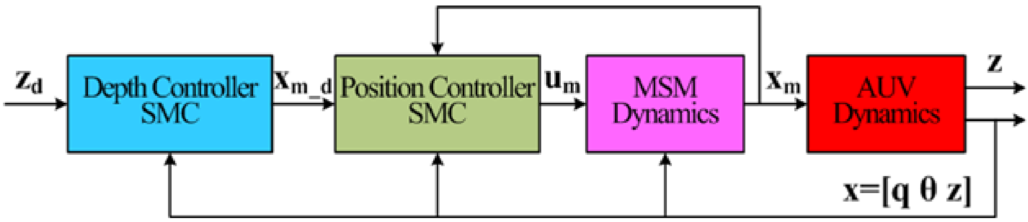

Fig. 5.3 Block diagram of diving control system

A diving control law is proposed for the AUV as in Fig. 5.3. All the controllers are designed by sliding mode control. First, the state errors calculated from desired set-points and feedback values of the vehicle states $x = [q\ \theta\ z]$ are the input of the depth controller block whose output is the desired movable mass position x_{m_d}. Then, the movable mass position error is calculated from the desired and feedback values of the movable mass position and becomes the input of the position controller block. The output of the position controller block is the appropriate force u_m used to drive the movable mass from position x_m to x_{m_d}. And finally, with this diving control law, the vehicle constructed by nonlinear equations will be controlled to the desired depth.

Substituting the coefficients [15] into Eq. (14) and the sliding poles of the diving control system selected arbitrarily at (-0.6, -0.5, 0), we get the depth controller as

$$\begin{cases} S_{s1} = [0.667 \quad 0.7337 \quad -0.1297]\tilde{x}_1 \\ u_1 = x_{m\_d} = 0.067q - 0.098\theta + 0.3637\eta_1 \tanh(S_{s1}/\phi_1) \\ -0.1 \le u_1 \le 0.1 \\ \eta_1 = 0.3, \quad \phi_1 = 0.1 \end{cases} \tag{48}$$

As expressed in SMC theory [23], the position controller for movable mass is designed from Eq. (2) as follows

$$\begin{cases} S_{sm} = \dot{\tilde{x}}_m + \lambda \tilde{x}_m \\ u_m = -f - \lambda \dot{\tilde{x}}_m - \eta_m \tanh(S_{sm}/\phi_m) \\ \lambda = 4.3, \quad \eta_m = 0.7, \quad \phi_m = 0.1 \end{cases} \tag{49}$$

5.4 Obstacle Avoidance

For obstacle avoidance, the travel of the AUV is considered in two dimensional environments. The environments consist of the vertical plane for tracking of the AUV's depth, and the horizontal plane for tracking the AUV's heading and position. During a typical mission, the sliding mode controllers mentioned previous will be applied to maintain commanded AUV's states. As the obstacle detected, feedback from forward looking sonar will create an error signal with the new commanded state which is built by a developed obstacle avoidance algorithm. And based on SMC controller, this error will be corrected by actuation of the rudder and mass shifter mechanism.

5.4.1 Obstacle Avoidance Using Heading Controller

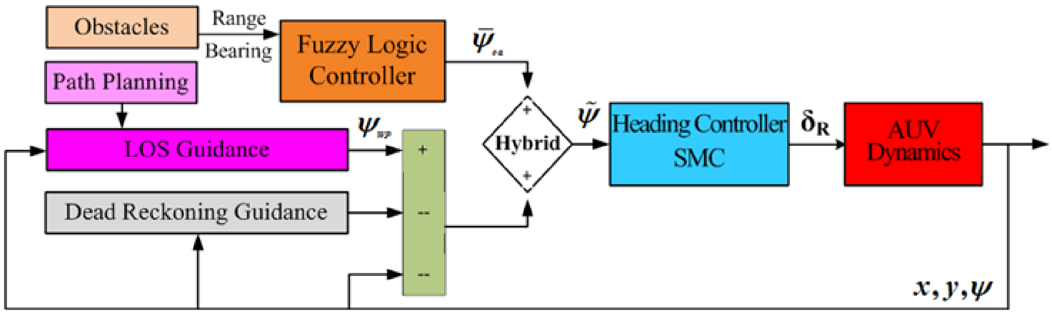

Fig. 5.4 Block diagram of the obstacle avoidance system

As shown in Fig. 5.4, the obstacle avoidance control in the horizontal plan is designed by using the heading controller described previously. The controller forces the vehicle to head in the direction of the current waypoint by reducing the heading error $\tilde{\psi}(t)$ to zero. Based on obstacle detected or not, the hybrid block will decide value of the heading error $\tilde{\psi}(t)$ that is calculated by LOS guidance, dead reckoning guidance, and fuzzy logic controller measuring the range and bearing of obstacles.

With LOS guidance, the heading error $\tilde{\psi}(t)$ as the difference between the commanded line of sight $\psi_{WP}(t)$ and the actual heading $\psi(t)$ can be expressed as:

$$\tilde{\psi}(t) = \psi_{WP}(t) - \psi(t) \tag{50}$$

where $\psi_{WP}(t)$ is defined in Eq. 46

The AUV adds a dead-reckoning point on the track toward the next waypoint forward of the vehicle position. The distance ρ to this point is incorporated into the heading error as follow:

$$\tilde{\psi}(t) = \psi_{WP}(t) - \psi(t) - \arctan(\varepsilon(t)/\rho) \tag{51}$$

where $\varepsilon(t)$ is the cross track error between the actual vehicle position and the desired track.

A two-dimensional forward looking sonar model with a 120^0 horizontal scan and a 50 m radial range is modeled for obstacle detection, as seen in Fig. 5.5 below. The probability of detection is based on a cookie-cutter approach in which the probability of detection is unity within the scan area and zero anywhere else.

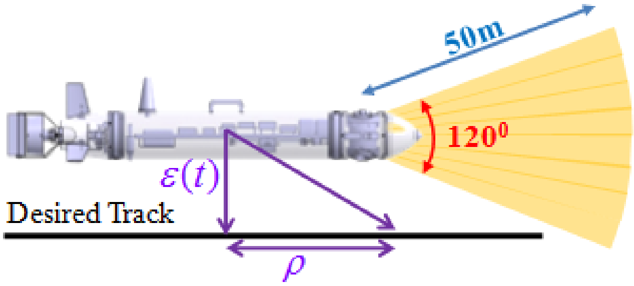

Fig. 5.5 Forward-looking sonar in horizontal plane

The obstacle avoidance model developed in this dissertation is based on the product of bearing and range weighting functions that form the gain factor for a dynamic obstacle avoidance behavior [25]. The weighting functions are Matlab membership functions from the fuzzy logic toolbox with the parameters selected to maximize obstacle avoidance behavior. The bearing weighting function is a Gaussian curve function of the form:

$$w_1 = e^{\frac{-(x-c)^2}{(2\sigma^2)}} \tag{52}$$

where the parameters $x, c,$ and σ are position, center, and shape respectively. The bearing weighting function can be seen in Fig. 5.6 below with values selected for these parameters are -90:90, 0, 20 respectively. With this selection, the bearing weight will be approximately in bound of [0 1] for anything locating in 120 degree horizontal scan.

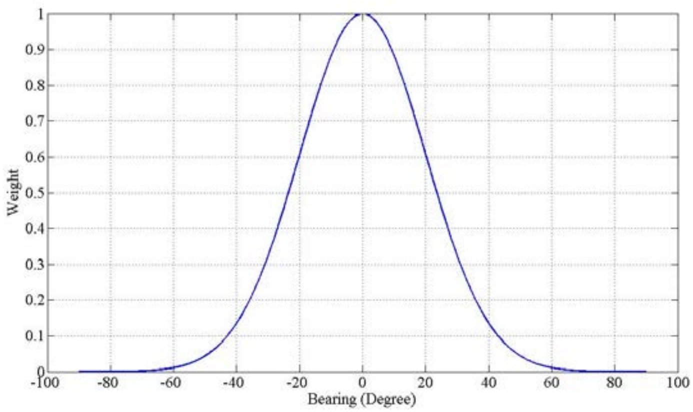

Fig. 5.6 Bearing weighting function

The function for range is an asymmetrical polynomial spline-based curve that is formulized as:

$$w_2 = zmf(x,[a\ b]) \tag{53}$$

where a and b are parameters that locate the extremes of the sloped portions of the curve shown in Fig. 5.7 with values selected for these parameters are 15, 50 respectively. With this selection, the range weight is approximately unity for anything closer than 15 m and zero for anything farther than 40 m from the AUV.

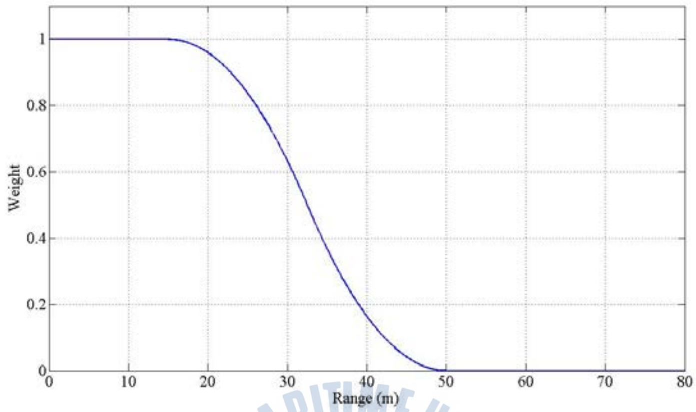

Fig. 5.7 Range weighting function

Following an evaluation of each obstacle at every time step, a final obstacle avoidance heading term is determined from the sum of the obstacle avoidance heading of each individual object within a bearing and range from the AUV, as expressed below:

$$\bar{\psi}_{oa}(t) = \frac{\sum_{1}^{i=n} \frac{\pi}{6} w_1(t,i) w_2(t,i)}{n} \tag{54}$$

where i is the obstacle being evaluated, n is the counter used to determine how many obstacles fall into horizontal scan. The counter is used to normalize this overall obstacle avoidance term to an average for all of the obstacles within the range above. In order to fall into the horizontal scan, the gain factor must be equal to or exceed a value of $w_1 w_2 = 0.15$. And the maximum heading for each individual object is not over $\pi / 6$ rad.

Finally, from Eq. 51 and Eq.54, the heading error $\tilde{\psi}(t)$ can be rewritten as:

$$\tilde{\psi}(t) = \psi_{WP}(t) - \psi(t) - \arctan(\varepsilon(t)/\rho) + \bar{\psi}_{oa}(t) \tag{55}$$

Through the designed heading controller in Eq.45, this heading error $\tilde{\psi}(t)$ will control the rudder to maneuver the vehicle avoid detected objects on the track path.

5.4.2 *Obstacle Avoidance Using Heading and Surge Velocity Controllers*

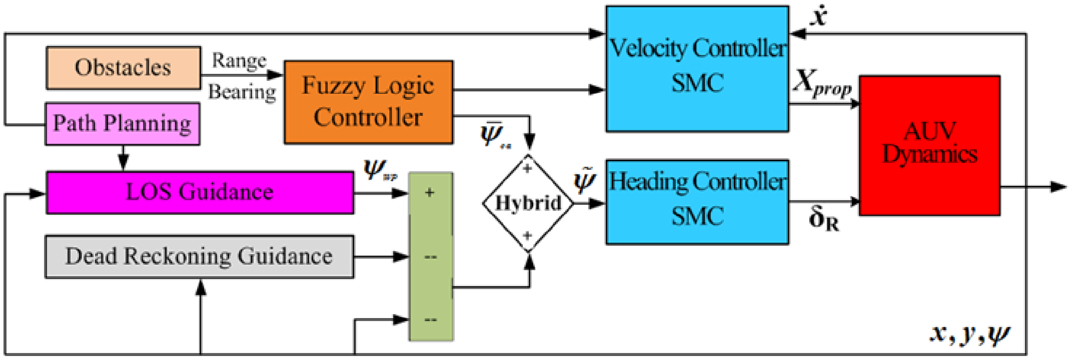

Fig. 5.8 Block diagram of the obstacle avoidance and velocity control

The previous section just considers the obstacle avoidance with constant velocity of 3 knots. As increasingly close to obstacle, it is difficult and unsafe to maneuver AUV to avoid the detected objects. So the velocity controller with output of the propeller thrust X_{prop} should be conducted in obstacle avoidance, as shown in Fig. 5.8.

Fig. 5.9 AUV arriving at a waypoint in commanded time

The AUV's mission is to move on the specified distance (AB) during a commanded time (Time_com), as shown in Fig. 5.9. Until an object is detected in a zone of the sonar sensor, obstacle avoidance algorithm mentioned previously will be applied and velocity also changed depend on the bearing and range weighting functions ($w_1(t), w_2(t)$). With assumption that the hydrodynamic forces in surge direction are not generated by lift and are dominated by drag and added mass effects. The surge equation modeling the longitudinal dynamics of the vehicle can be simplified as:

$$(m - X_{\dot{u}})\dot{u} = mvr + X_{u|u|}u|u| + X_{prop} \tag{56}$$

Assuming the lateral velocity v is negligible. Using hydrodynamic coefficients from [15], Eq. (56) can be rewritten as:

$$\begin{aligned}\dot{u} &= \frac{1}{(m - X_{\dot{u}})}[X_{u|u|}u|u| + X_{prop}] \\ &= -0.052u|u| + 0.032X_{prop}\end{aligned} \tag{57}$$

The sliding surface can be defined in the error state space form as follows

$$S_v = u(t) - u_{COM}(t) \tag{58}$$

To guarantee the error state converging to sliding surface as Eq. (37) we select

$$\dot{S}_v = -\eta_v \tanh(S_v / \phi_v) \quad \eta_v > 0 \tag{59}$$

Hence the velocity control law in terms of the command for X_{prop} is defined as:

$$X_{prop} = 1.625u|u| - 31.25\eta_v \tanh(S_v / \phi_v) \tag{60}$$

with $\eta_v = 0.4$ and $\phi_v = 0.3$

The velocity command $u_{COM}(t)$ can be determined when no obstacle detected:

$$\begin{cases} u_{COM}(t) = s(t) / Time\_remaining \\ Time\_remaining = Time\_com - Time\_spent \\ 0\ knot \le u_{COM}(t) \le 5\ knots \end{cases} \tag{61}$$

where $s(t)$ is the distance to the specified waypoint

Or when obstacle is detected:

$$\begin{cases} u_{COM}(t) = \dfrac{s(t)}{Time\_remaining}\left(1 - \dfrac{\sum_{1}^{i=n} w_1(t,i)w_2(t,i)}{n}\right) \\ 0\ knot \le u_{COM}(t) \le 3\ knots \end{cases} \tag{62}$$

5.4.3 *Obstacle Avoidance Using Depth Controller*

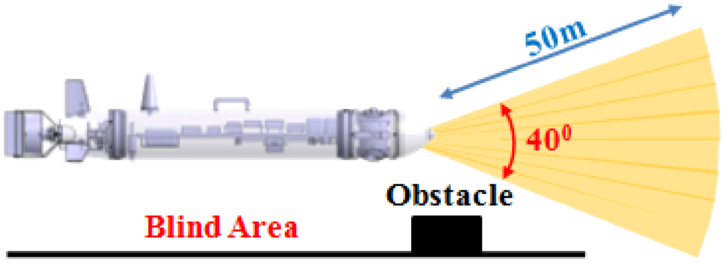

Fig. 5.10 Forward looking sonar model

To perform obstacle avoidance in the vertical plane, a forward looking sonar is required to install on the AUV. With this sonar, coral reefs and other such obstacles can be detected at a distance adequate enough to allow for a gradual ascent over the obstacle. For simulation, the range of selected forward looking sonar is 50 m with a 40^0 vertical scan, as shown in Fig. 5.10. An existing problem due to the abilities of the forward looking sonar that is a blind spot located below the scan of the sonar. Clearly, the sonar will detect the object as it moves forward, but this area can cause problems with the controller when trying to determine how to maneuver over the object and when the object has safety passed. This will be seen in the below simulation using a basic control method.

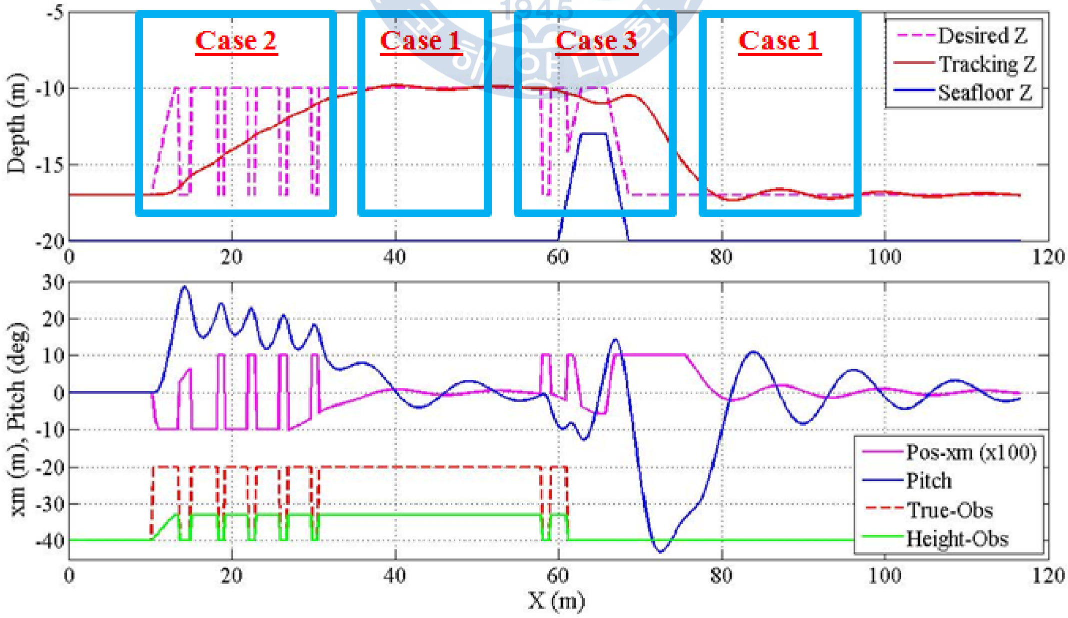

Fig. 5.11 Obstacle avoidance problems using depth command

The final obstacle avoidance algorithm was not developed immediately. It was a process of trial and error, eliminating concepts and developing new ones based on old results. Initially, a basic control method used to avoid an obstacle is depth command. When an obstacle is detected within the zone of the sonar sweep (True = 1), the obstacle avoidance algorithm will create a new depth command that increases linearly and based on the height and range of the detected obstacle. In the otherwise, the AUV will attempt to return to its original depth when obstacle is removed from the sonar's field of view (True = 0). A basic simulation using depth control method for obstacle avoidance was performed, as shown in Fig. 5.11. And there are three cases of the problem in this method need to be solved. As the results in case 1 of this simulation showed that the depth command and the pitch angle battled each other. When commanded depth was obtained, the pitch didn't get steady-state. The result was a difficult to predict flight path of the AUV. And this created a sinusoidal flight path that was both inefficient and unsafe for obstacle avoidance. So a solution to the battle between the depth and pitch command was developed by creating a new depth command in support of the pitch command. That means the pitch angle was kept at 0 degree when the AUV got a desired depth. As shown in Fig. 5.12, the case 1 of the problem was solved but the other cases still occurred.

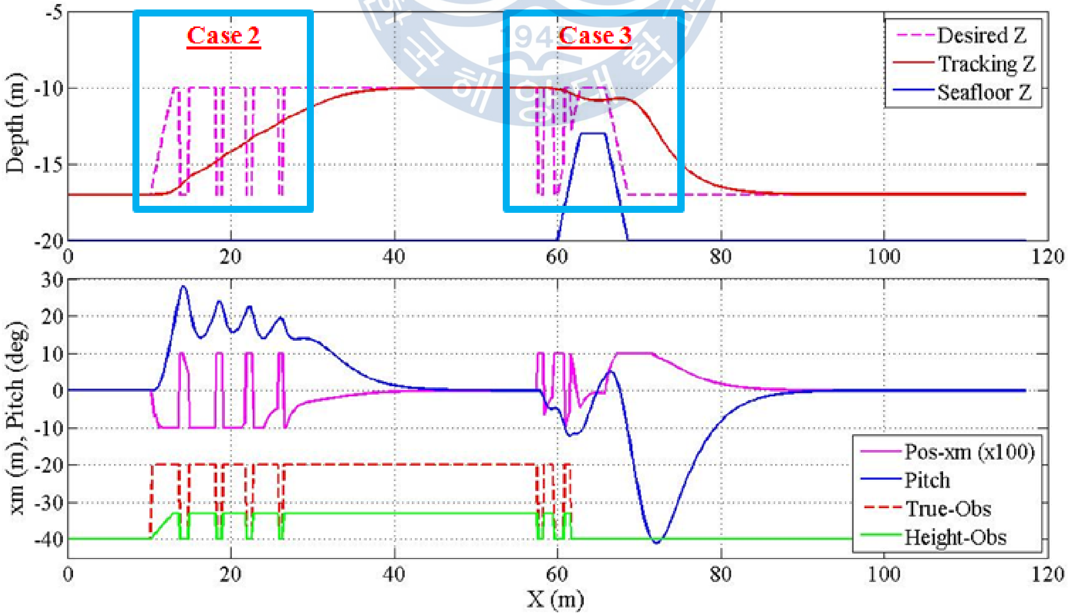

Fig. 5.12 Obstacle avoidance results using pitch and depth command

Moreover, results in case 2 of Fig. 5.11 and Fig. 5.12 displayed that the pitch angles varied too much when the AUV was trying to reach a safe altitude compared with obstacle. So the obstacle no longer was detected by forward looking sonar, and the red dashed line of True variable was also changed continuously. Each time the obstacle is removed from the sonar's field of view the AUV will attempt to return to its original depth. This will create a sinusoidal type flight path which both wastes energy and put the AUV in a danger of collision. To smooth out the flight path and improve predictability of the AUV when the obstacle first detected, a sloping depth command based on a linear equation was used as below:

$$y = ax + b \tag{56}$$

where y is the desired depth command, b is a safe height from obstacle, and a is the angle factor determined from the quotient of the measured height of the obstacle and the range to the obstacle. These variables are illustrated as a geometry expressed in Fig. 5.13.

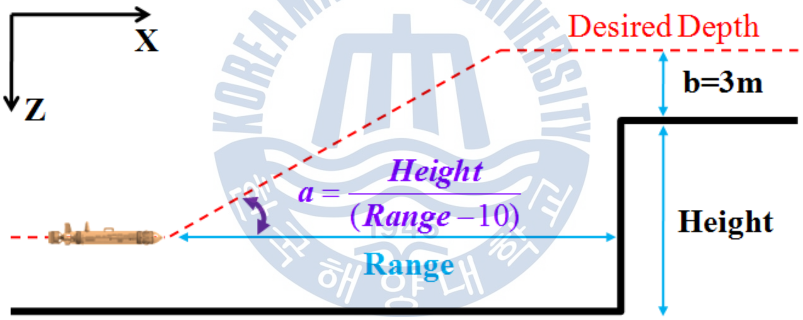

Fig. 5.13 Sloping depth command generator for obstacle avoidance

The simulation results using pitch and sloping depth commands are shown in Fig. 5.14. The AUV successfully avoids an obstacle in the vertical plane with the problems in case 1 and 2 solved completely. By setting the pitch command to zero at all operating time, the obstacle is always within the zone of the forward looking sonar (True =1) and therefore a smooth flight path is created by the sloping depth command. However, the obstacle may still enter the blind spot of the sonar, in case 3 of the problem. And then, the AUV will mistakenly think that the obstacle has already passed safety and it will also pitch down to descend quickly to its original altitude of 3 meters. For this reason, an algorithm using a dead-reckoning approach making an estimated position of the obstacle may allow for the AUV to blindly ascend until the obstacle has passed. This algorithm combined with pitch and sloping depth command will make a final solution for the obstacle avoidance system of the AUV, as shown in Fig. 5.15. And the final simulation results with different obstacles will be described in next chapter.

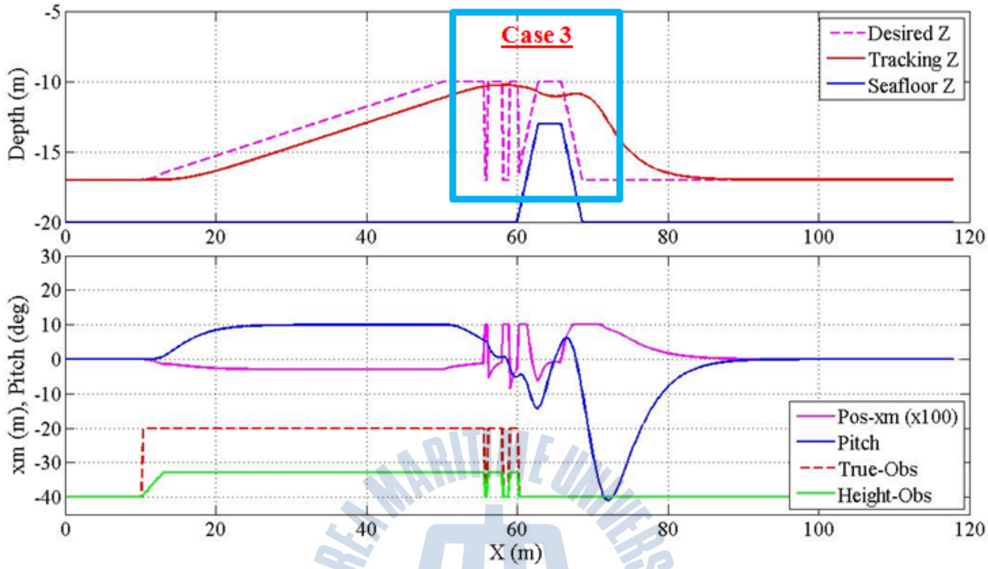

Fig. 5.14 Obstacle avoidance results using pitch and sloping depth command

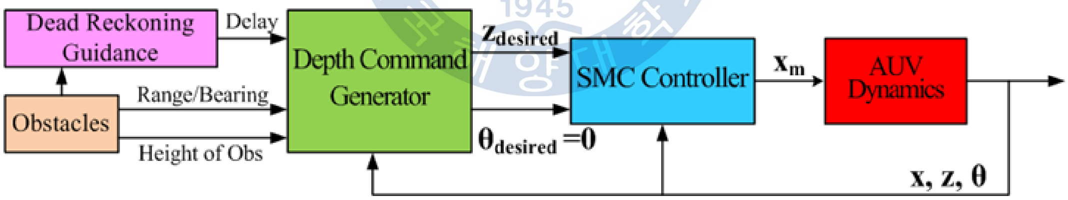

Fig. 5.15 Block diagram of the obstacle avoidance system

Chapter 6: Simulation and Experiments

For performing simulation and experiments of the control algorithms proposed in chapter 5, the nonlinear hydrodynamic coefficients of forces and moments [15] were adapted to the developed AUV 6 DOF nonlinear dynamics model. The weight of movable mass was selected of 2.5 kg, and the length of stroke was manufactured of [-0.1 m, 0.1 m]. Experiments showed that the pitch angle should be less than 30 degrees to have good efficiency [25], so it was limited to ±30 degrees.

6.1 Heading Control

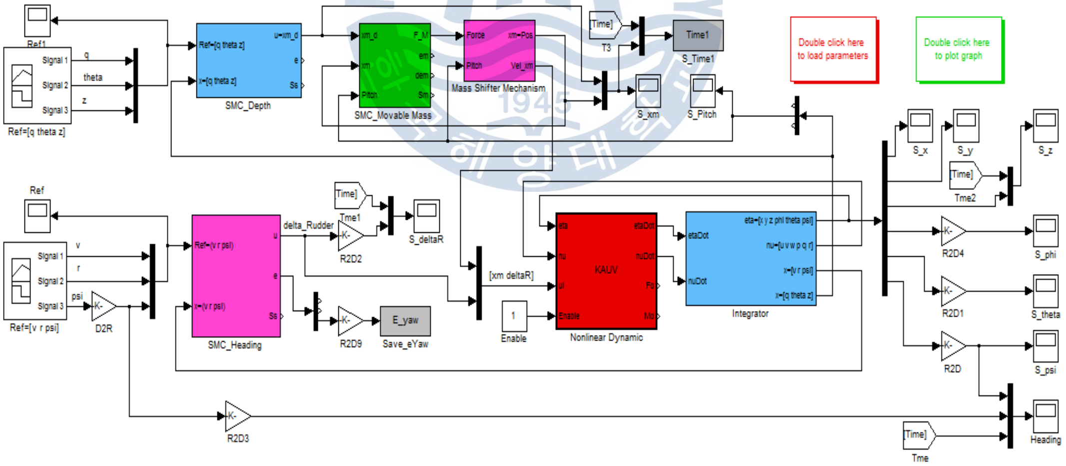

Fig. 6.1 Matlab simulink of the heading control

From Eq. 45, the Matlab/Simulink of the heading control was performed as shown in Fig. 6.1. To stabilize the dynamic system in heading control, the depth control using Eq. 48-49 was also implemented. For this reason, the AUV maintains a zero-degree pitch angle and a very small value of the depth control as seen in Fig. 6.4. The surge velocity was always kept constant at 1.5 m/s (3 knots) and the initial roll angle was set at -5 degrees. The initial heading angle of 100 degrees was kept static in 5s as shown in Fig. 6.2. After that, the AUV was commanded to track two desired heading angles of 125 degrees and 100 degrees respectively in 20 seconds for each turn. The experimental results showed that the overshoot angle was in the range of 5 degrees. The settling time of the experimental results was longer than that of the simulation ones. In general, both the experimental and simulated results show good convergence with the desired heading angles. Finally, the Fig. 6.3 expresses the rudder angle and the movable mass position curves when the AUV performs heading control. The rudder angle is not only saturated to ±30 degrees but also the movable mass position is well-tracked to the desired one. Moreover, the chattering of control inputs were eliminated by the continuous function $\tanh(S_{s2}/\phi_2)$ with $\phi_2 = 0.1$.

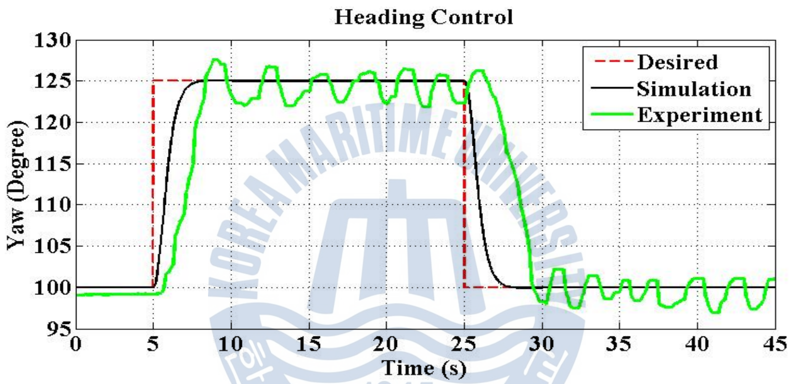

Fig. 6.2 Heading control results of the AUV

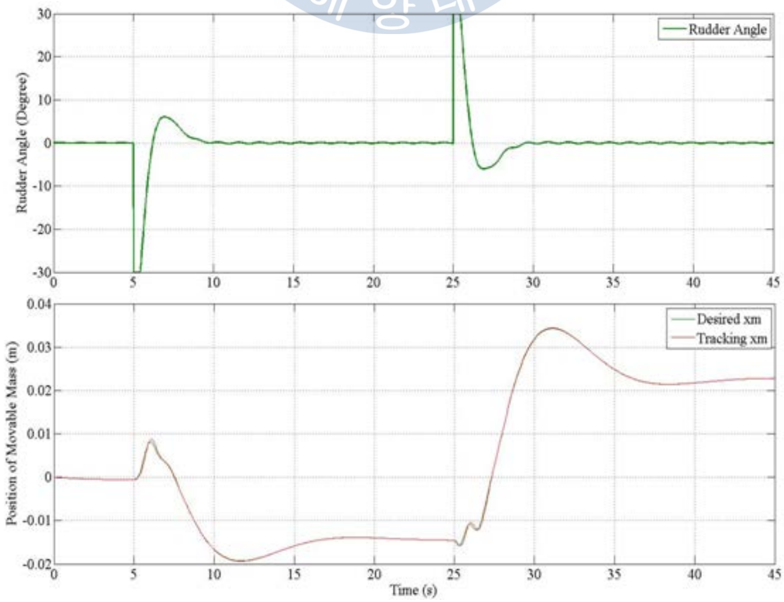

Fig. 6.3 Roll, pitch, and depth values of the AUV

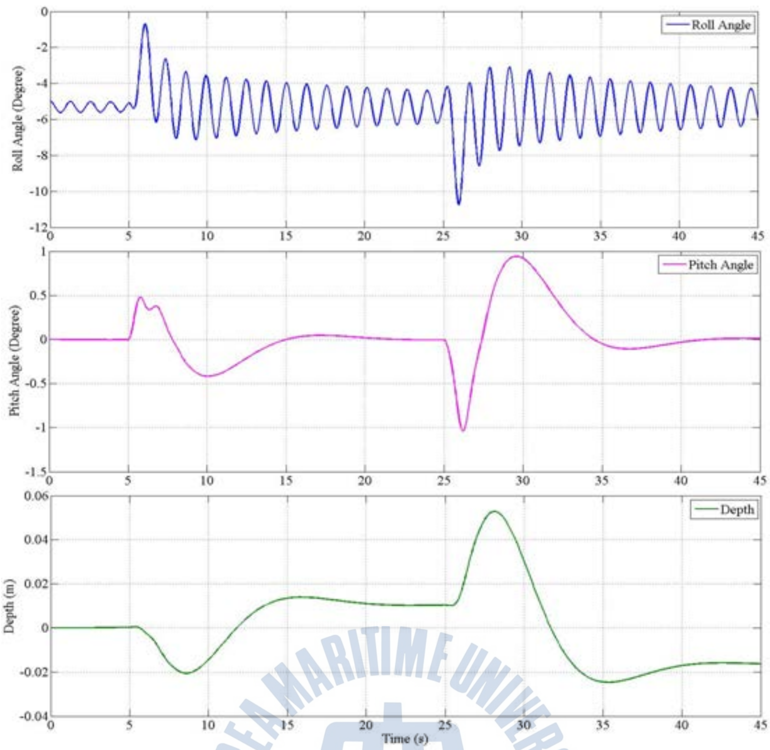

Fig. 6.4 Rudder angle and position of the movable mass

6.2 Waypoint Control

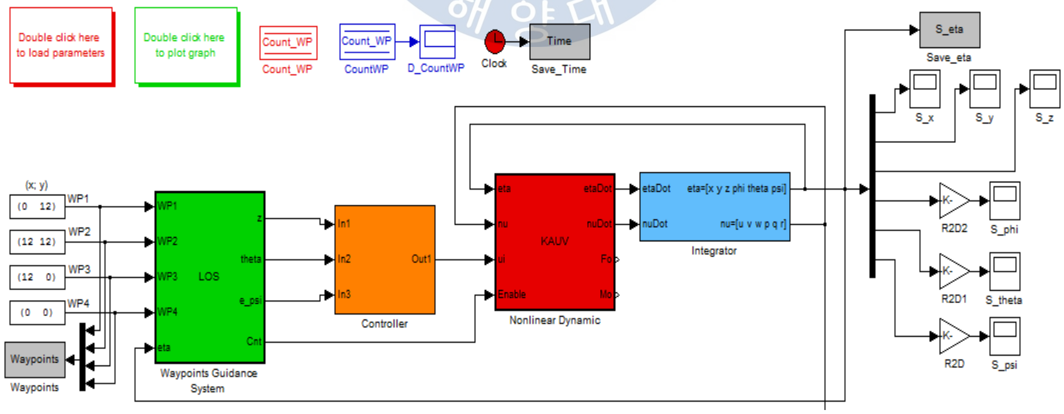

Fig. 6.5 Matlab simulink of the waypoint tracking control

In experiment and simulation of the waypoint tracking, the AUV was commanded to track four predefined waypoints as (0, 0), (0, 12), (12, 12), (12, 0) to validate the LOS algorithm and heading control strategy described in chapter 6. The experiment was carried out on a sea, but assuming it was performed in a sea. So positions of the vehicle were measured by DVL-INS system replacing GPS-INS system. The threshold value ρ_0 in Eq. 47 was selected to 1 m. The results of experiment and simulation are shown in Fig. 6.6 with four desired waypoints achieved successfully and no big difference between both performances.

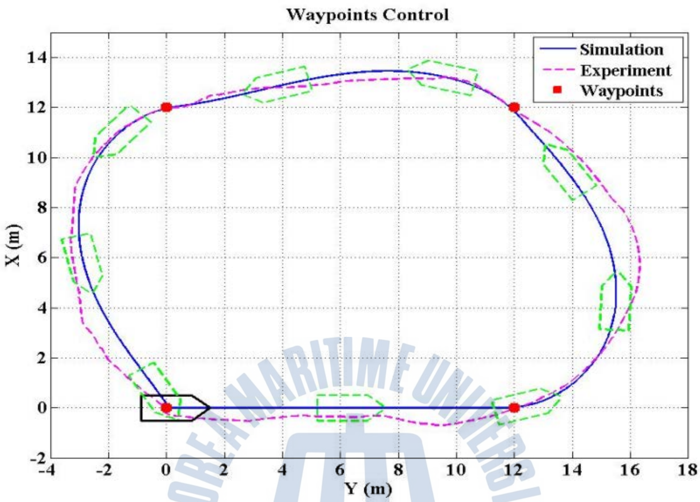

Fig. 6.6 Results of the waypoint tracking control

6.3 Depth Control

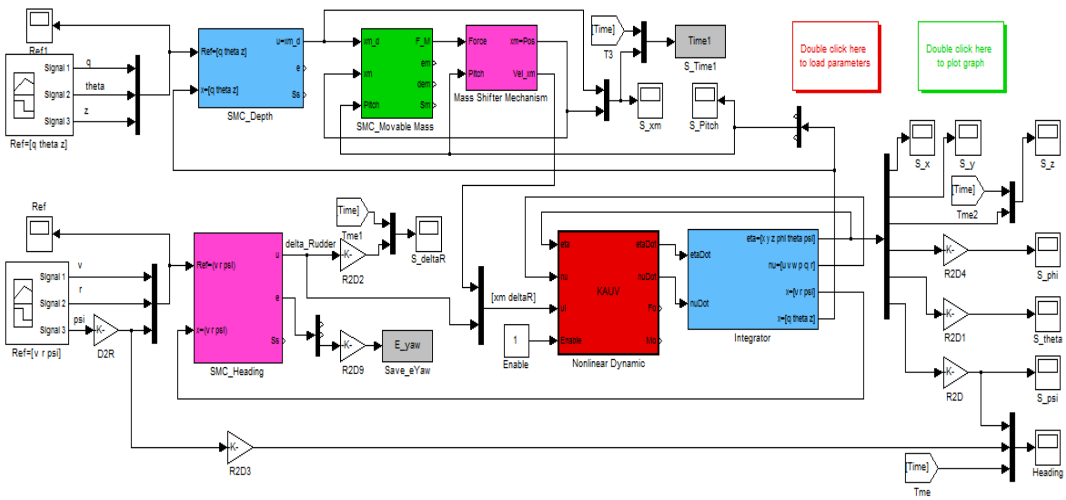

Fig. 6.7 Matlab/simulink of the depth control

For simulating the diving control of the developed AUV using the proposed depth control strategy in Eq. 48-49, the Matlab/Simulink was performed as shown in Fig. 6.7. The surge velocity was always kept constant at 1.5 m/s (3 knots) and the initial roll angle was set at -5 degrees. The results of experiment and simulation are plotted in Fig. 6.8. In this performance, the initial depth was 0m, and the reference depth was 2.5 m. The mass shifter mechanism was controlled in bound of stroke's length [-10 cm, 10 cm] to shift the center gravity of the whole vehicle for diving and floating. Both the experimental and simulated results show good convergence with the desired depth. The settling time of the experiment results was longer than that of the simulation ones. It is supposed that the discrepancies in the responses are due to the errors of the numerical and real model. A better numerical model will be estimated based on the experimental data using the system identification method in future works.

Based on the sway/yaw control loop architecture derived from the work [20], a heading controller as Eq. 45 was also implemented to steer the AUV such that it maintains a zero-degree heading error and a constant y component coordinate position. The results shown in Fig. 6.9 demonstrate that the AUV states will stabilize at steady states after each change in depth of the vehicle.

Finally, the Fig. 6.10 expresses the rudder angle and the movable mass position curves when the AUV performs the heading and depth controls. Easy to recognize, these curves have a direct correlation with each turns of these controls. The rudder angle is commanded to hold heading value at 100 degrees. And position of the movable mass is always controlled in bound of [-10 cm, 10 cm] when the AUV performs the depth control.

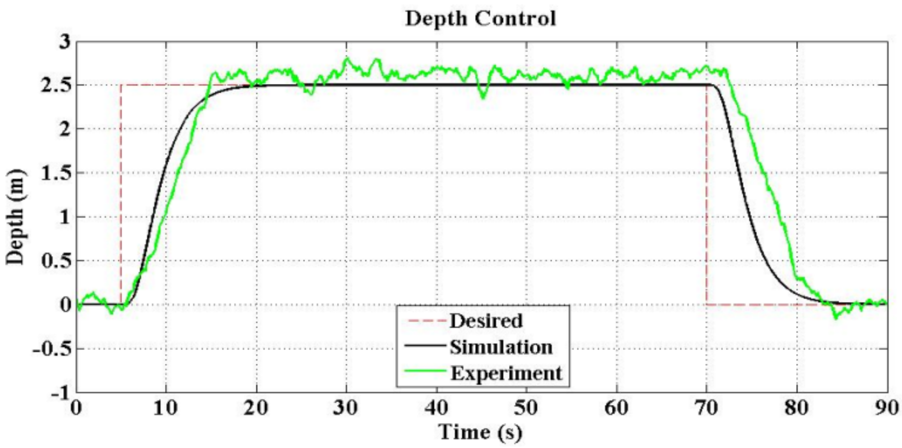

Fig. 6.8 Diving control results of the AUV

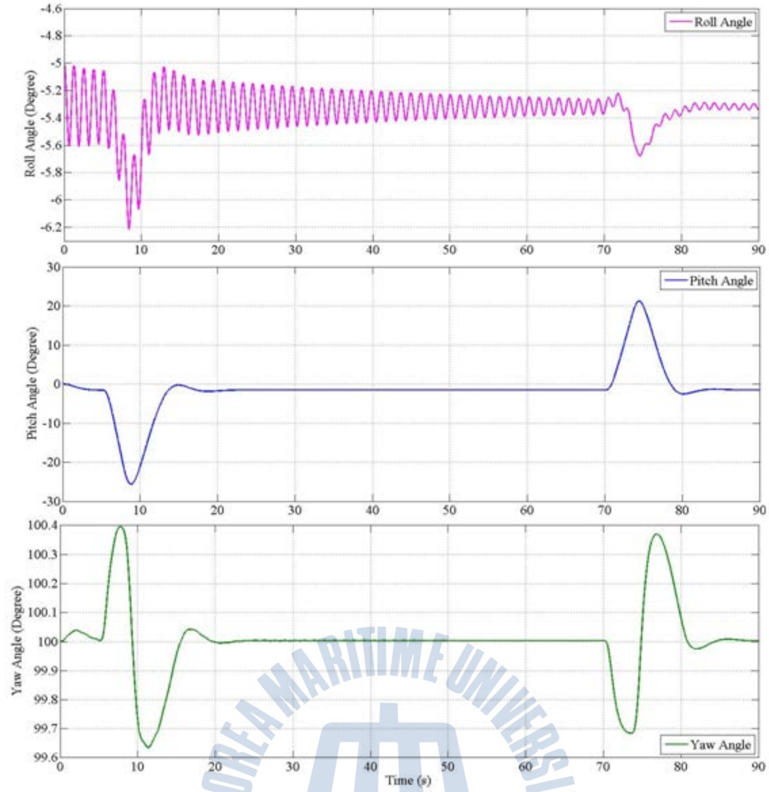

Fig. 6.9 Roll, pitch, and yaw values of the AUV

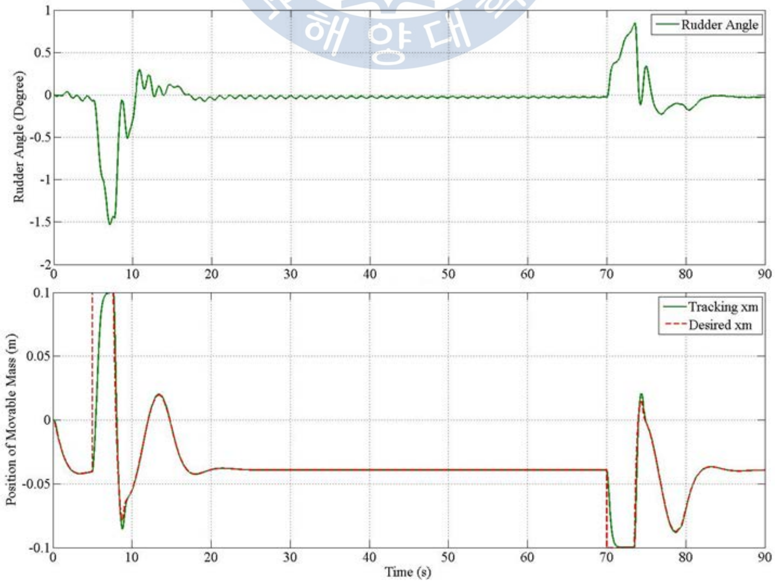

Fig. 6.10 Rudder angle and position of the movable mass

6.4 Combination of Heading and Depth Controls

This section performs a combined simulation of heading and depth controls. As shown in Fig. 6.11, the initial heading value of 100 degrees is kept static in 25 seconds. After that, the AUV is commanded to track two desired heading angles of 125 degrees in 20 seconds and 100 degrees in 45 seconds respectively. Besides the heading angle ordered, the depth values are also commanded as seen in Fig. 6.11. The simulation results show that there is an associated depth control action with each turns of the heading control and vice versa. Through the previous controllers designed, operation of the AUV dynamic system is stable. And the states shown in Fig. 6.12 are too. Moreover, the rudder angle and the movable mass position curves are smooth when the AUV operating, as expressed in Fig. 6.13. The rudder angle is saturated to ±30 degrees and the movable mass position is also well-tracked to desired position with the limited bound of [-10 cm, 10 cm].

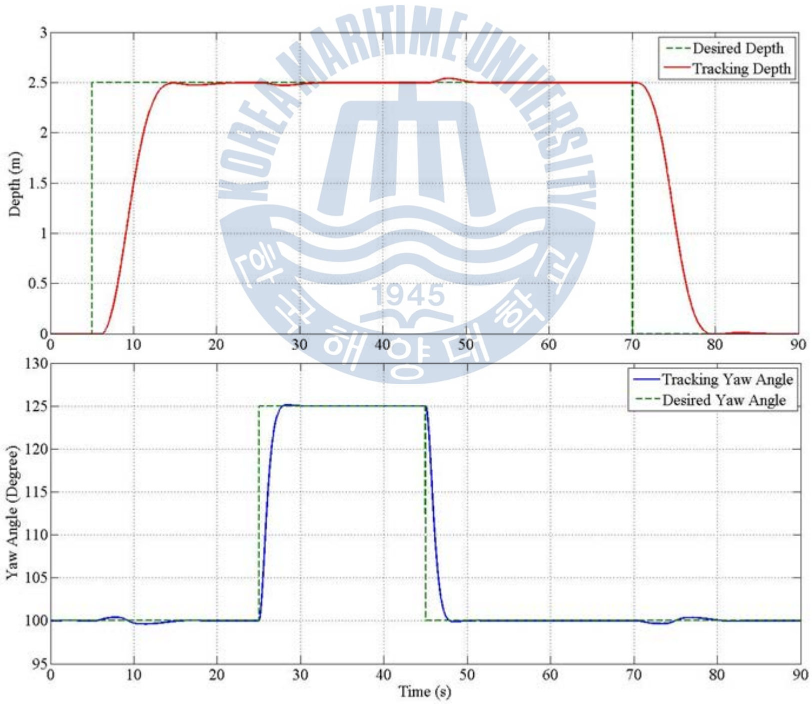

Fig. 6.11 Combined control results of the heading and depth control

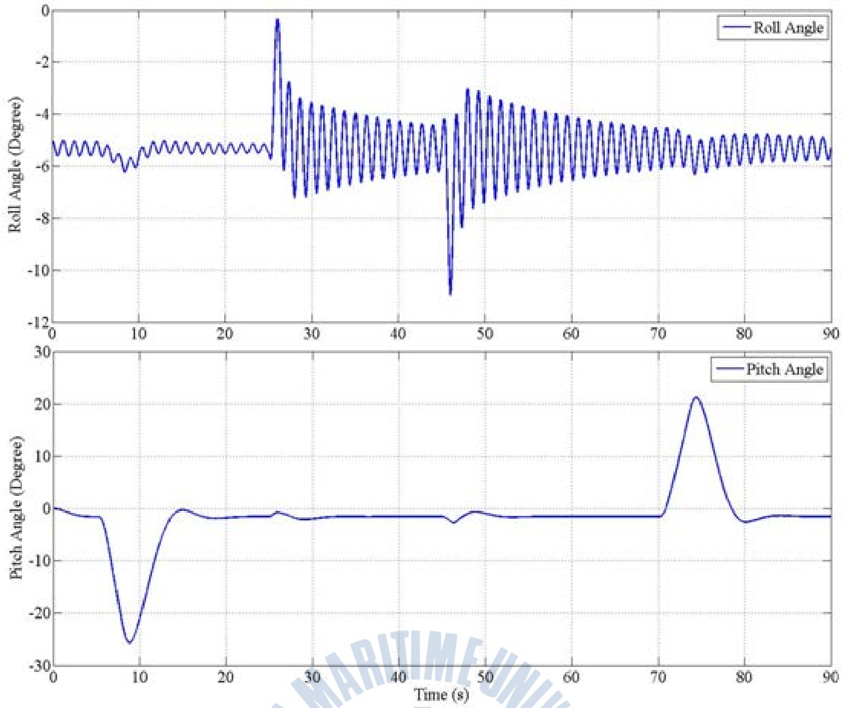

Fig. 6.12 Roll and pitch values of the AUV

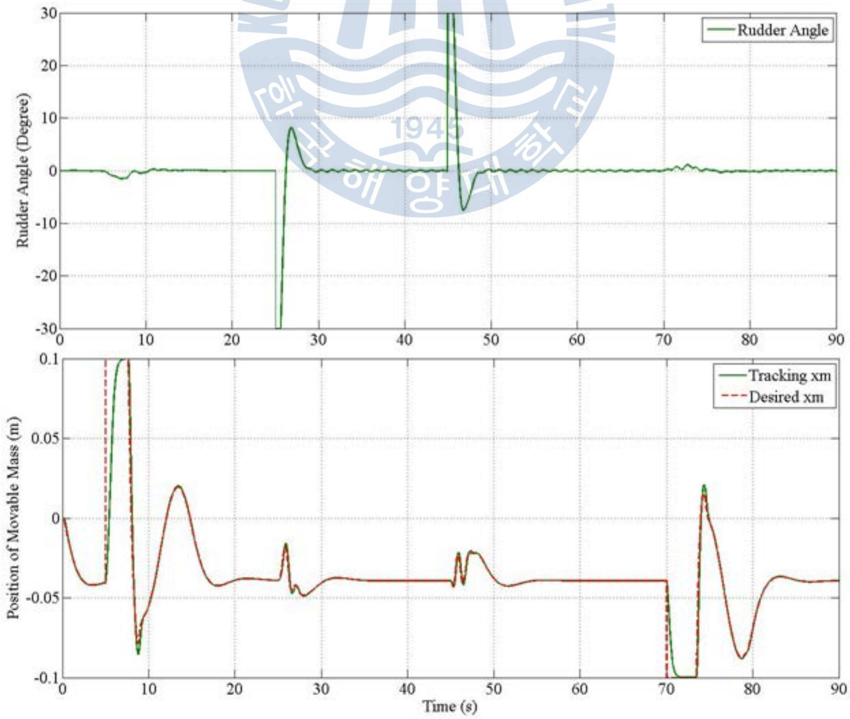

Fig. 6.13 Rudder angle and position of the movable mass

6.5 Obstacle Avoidance

6.5.1 Obstacle Avoidance with Constant Surge Velocity in Horizontal Plane

A computer simulation applying the proposed control block diagram as Fig. 5.4 was performed by Matlab/Simulink. In this control system, the vehicle guidance is built from the heading error $\tilde{\psi}(t)$ of Eq. 55. Using this error, the heading controller as Eq. 45 is applied to drive the AUV tracking the desired path. Depend on search area and target detection analysis, this path may be varied. In this dissertation, two common types of paths as square and zigzag paths were used to maneuver the developed AUV.

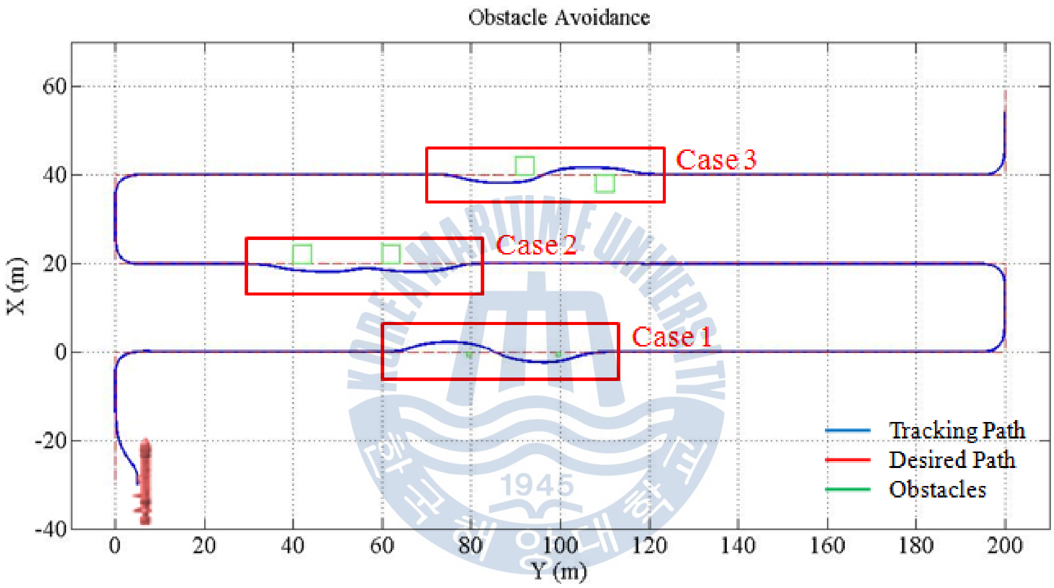

Fig. 6.14 Obstacle avoidance with tracking the square path

The first desired path as shown in Fig. 6.14 is square type. The length of square path is 200 m with 20 m of separated width. To verify the obstacle avoidance ability of the AUV, some square obstacles with different sizes of 1 m^2 and 16 m^2 are disposed on the desired path. There are three disposition cases for this simulation: multiple obstacles on the left side of the path, multiple obstacles on the right side, and multiple obstacles on either side of the path. In the initial condition of this simulation, the AUV starts at position of (5 m, -30 m), not on the desired path. The surge velocity is kept constant at 1.5 m/s, the initial yaw angle of 0 degree, and ρ of 6 m. Hence, to track the path from the started position, the AUV needs 10 seconds as shown in Fig. 6.16. In this figure, the rudder action has a direct correlation with the vehicle heading. The large angle motions of the heading are the

90 degrees turns built to track the ordered AUV path. There is an associated rudder action with each of these turns as seen by the corresponding rudder curve. These rudder curves show that the maximum programmable rudder deflection is 30^0. For all dynamic behaviors, whether associated with a turn or obstacle avoidance maneuver, the absolute rudder angle for turning is always less than this maximum value. Because the desired path is square, the desired headings are only 0 degree or ±90 degrees. These commanded angle values are tracked well before obstacle detected. When case 1, 2, 3 happen, the controller will maneuver the AUV heading to avoid obstacle with heading error $\tilde{\psi}(t)$ depend on the bearing and range weighting functions of the obstacle detection. As seen in Fig. 6.15, the zoom of obstacle avoidance maneuver in case 1, 2, 3, the AUV successfully avoids multiple obstacles and always maneuvered at a safe distance. In addition, rudder dynamics are minimal during all avoidance maneuvers for an efficient response.

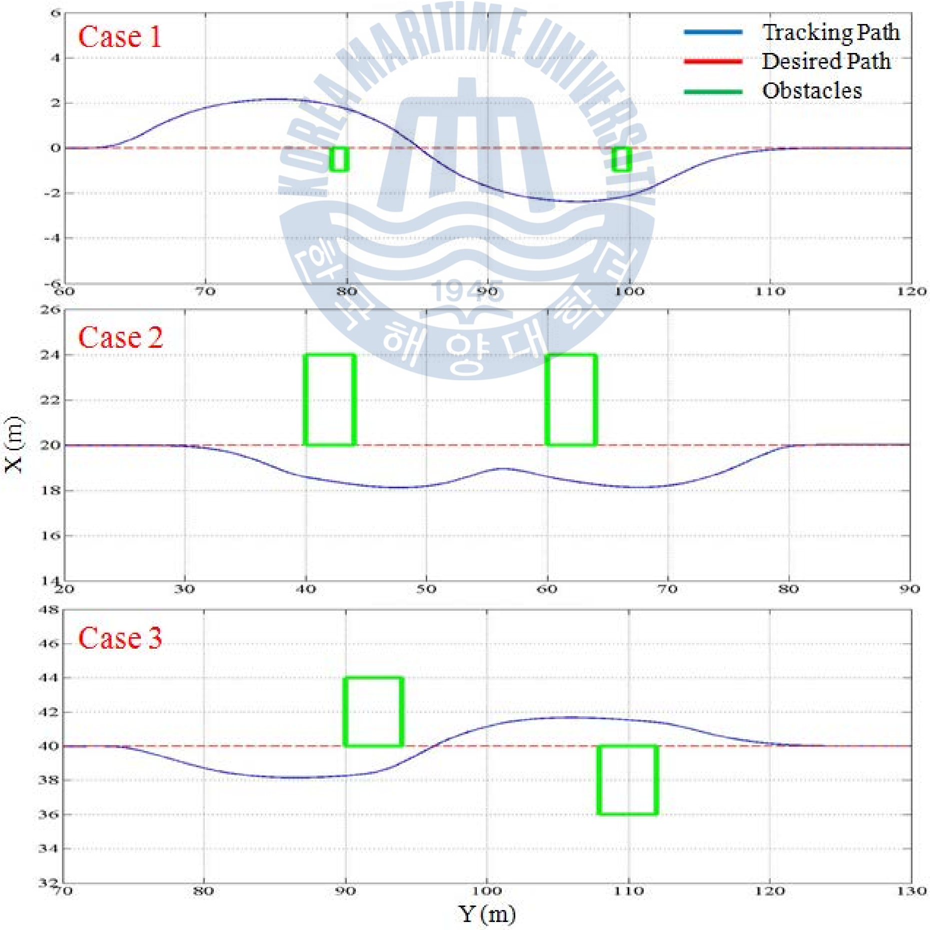

Fig. 6.15 Zoom of obstacle avoidance in case 1, 2, 3

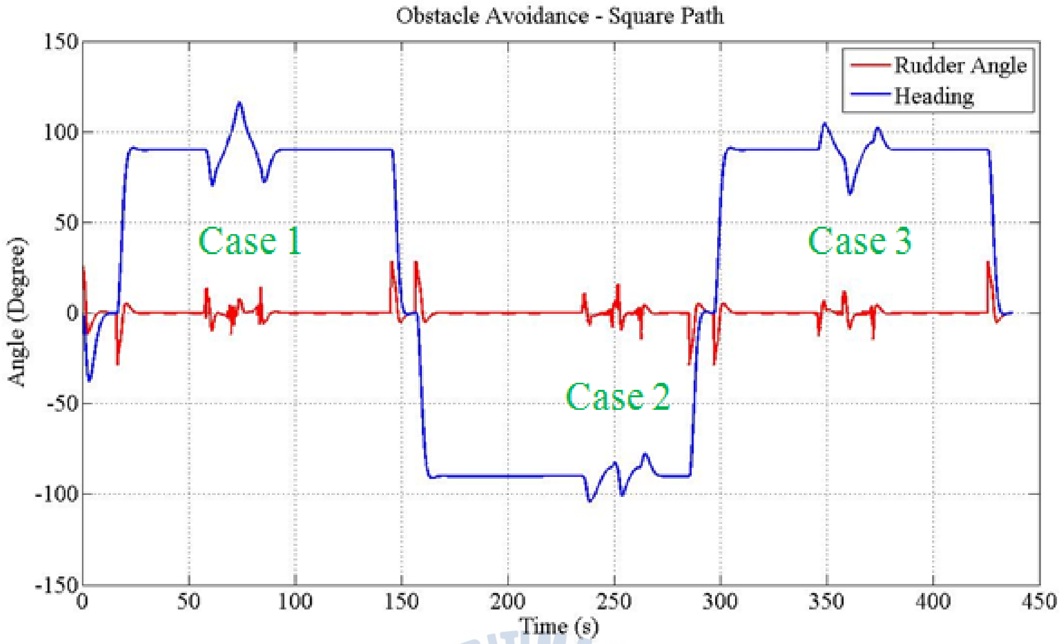

Fig. 6.16 Rudder and yaw angles of the AUV

To check the adaptability of the developed obstacle avoidance algorithm for different desired paths, the other zigzag path with length of 100 m and maximum separated width of 50 m was used to maneuver the developed AUV. The single and multiple obstacles with size of 16 m^2 are disposed on the desired path. For this simulation, the dispositions are on the left side, and on the right side of the path respectively, as shown in Fig. 6.17. In the initial condition of this simulation, the AUV starts at position of (-50 m, 5 m), not on the desired path. The initial yaw angle is 50 degrees, and another setting parameters similar the previous simulation. The dynamic obstacle avoidance behaviors shown in Fig. 6.17 and zoomed in Fig. 6.18 demonstrate that the AUV successfully avoids multiple obstacles and always maneuvered at a safe distance. As expressed in Fig. 6.19, the rudder curves show that the absolute rudder angle for turning and obstacle avoidance maneuver is always less than 30^0. In addition, rudder dynamics are minimal during all avoidance maneuvers for an efficient response.

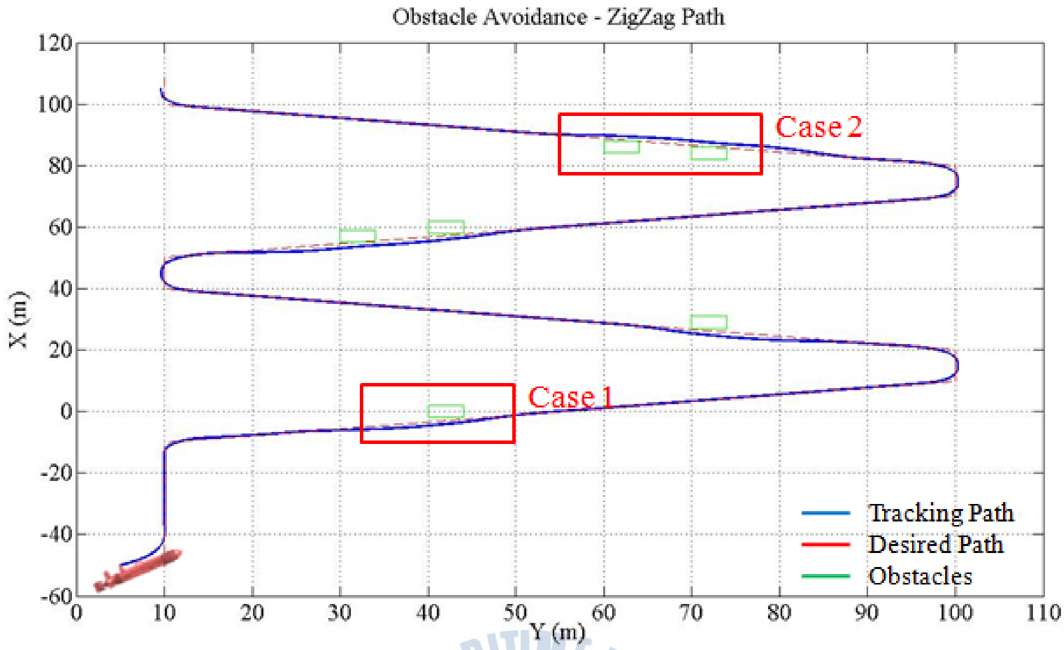

Fig. 6.17 Obstacle avoidance with tracking the zigzag path

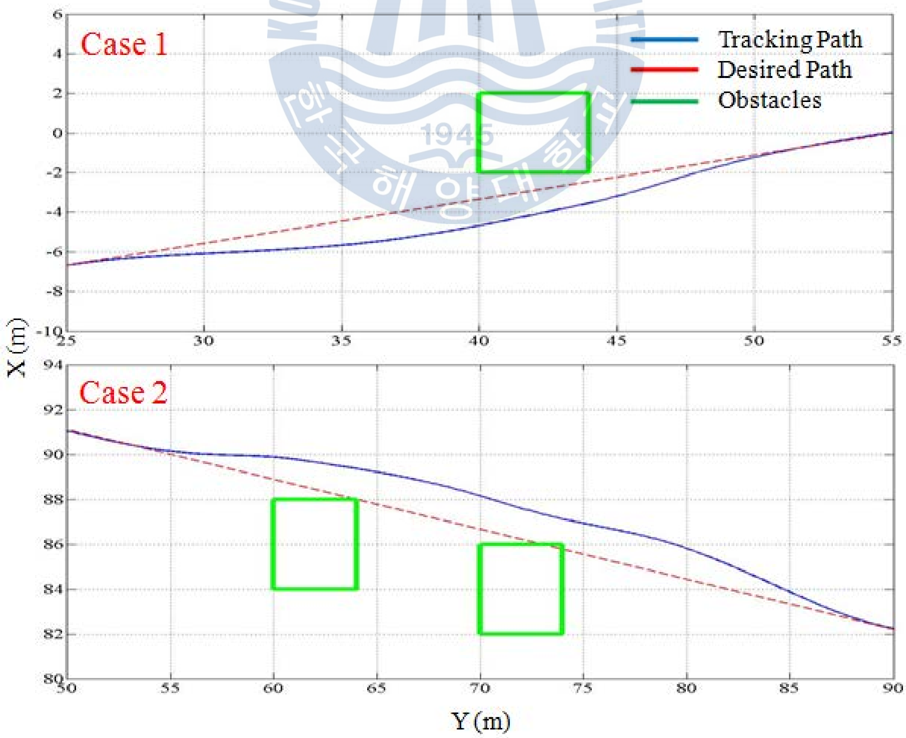

Fig. 6.18 Zoom of obstacle avoidance in case 1, 2

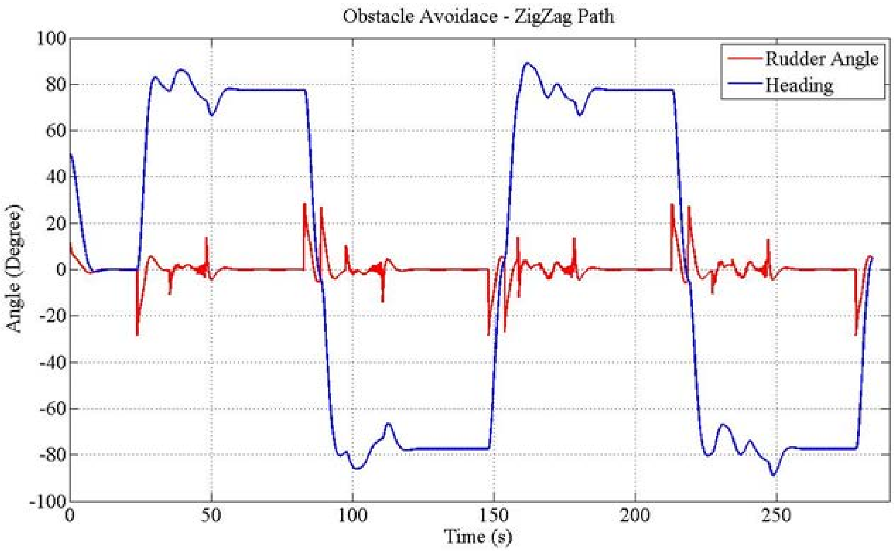

Fig. 6.19 Rudder and yaw angles of the AUV

6.5.2 *Obstacle Avoidance with Changing Surge Velocity and Limited Time*

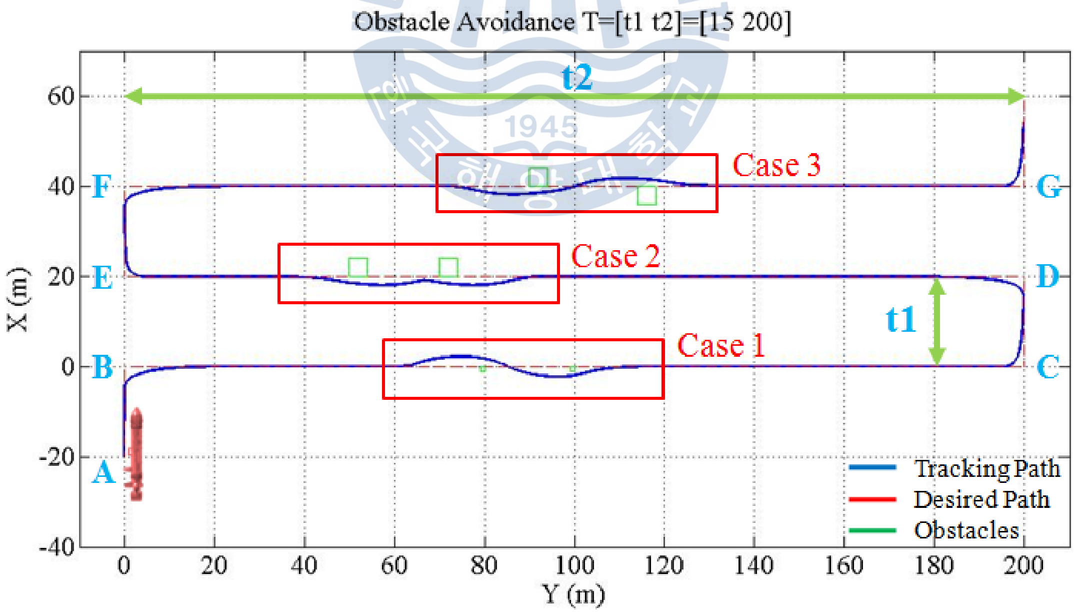

Fig. 6.20 Obstacle avoidance with T = [15 200]

Different with previous section 6.5.1, the surge velocity and travelling time will be considered in this section. A simulation applying the proposed control block diagram as Fig. 5.8 was performed. Using sliding mode control, the propeller thrust X_{prop} was calculated by Eq. 60 with commanded velocity as Eq.61 and Eq.62. Based on trial and error method, the control parameters were selected as $\eta_v = 0.4$, $\phi_v = 0.3$, and $\rho = 6\,m$.

In Fig. 6.20, the AUV is commanded to track a square path from A to G with travelling intervals of $t1=t_{AB}=t_{CD}=t_{EF}=15$ seconds and $t2=t_{BC}=t_{DE}=t_{FG}=200$ seconds respectively. In the initial condition of this simulation, the AUV starts at position of (0 m, -20 m), the surge velocity of 1.03 m/s (2 knots), yaw angle of 0 degree, and propeller thrust $X_{prop} = 0N$. This simulation result shows that the AUV can track the desired path well with small errors and avoid multiple obstacles in case 1, 2, 3 successfully. The rudder angle and heading angle of the AUV are shown in Fig. 6.21. The rudder curve shows that the maximum programmable rudder deflection is 30^0. According to the direction of waypoints, the heading curve is maneuvered to get desired values of 0 degree or ±90 degrees. And when having the obstacle detection, based on Eq. 55 of heading error $\tilde{\psi}(t)$, the heading curve will be changed so that the AUV can over obstacles. The pink dashed line describes the obstacle detectability of the sonar modeling. This line is set True/False when an obstacle is in/out the zone of the sonar sweep.

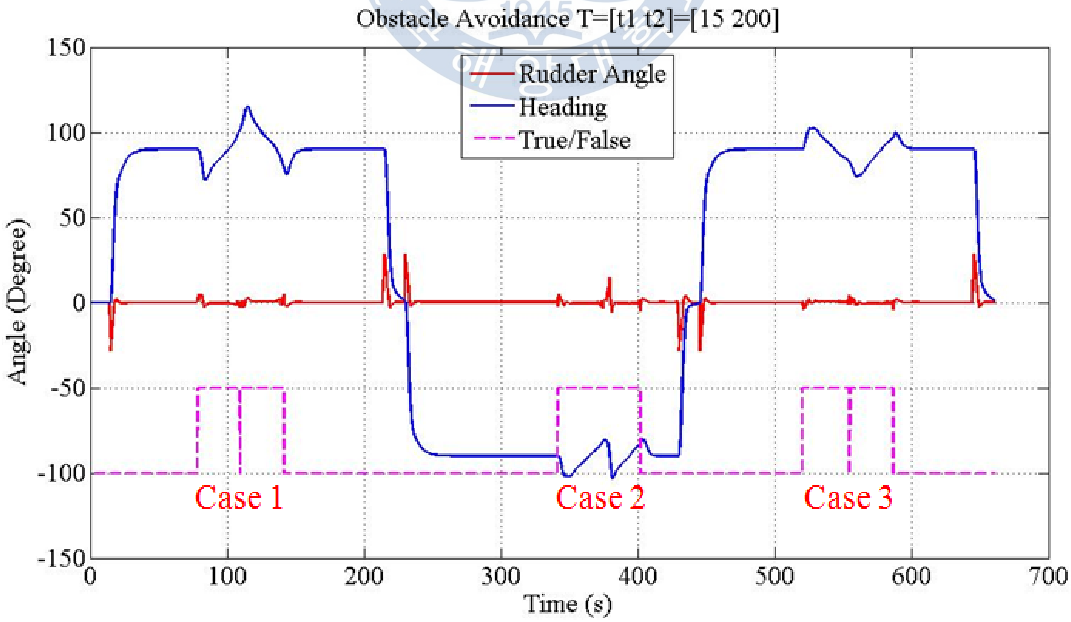

Fig. 6.21 Rudder and yaw angles of the AUV with T = [15 200]

The surge velocity and travelling time of the AUV motion are shown in Fig. 6.22. The travelling time for each segment of the square path satisfies the required time, T1=T3=T5=t1=15 seconds and T2=T4=T6=t2=200 seconds. So the total time of the AUV moving from waypoint A to waypoint G is 645 seconds. In segment BC with travelling time T2, the AUV velocity is 0.98m/s during 63 seconds. When having obstacles on the path, the surge velocity will be reduced to ensure safety for obstacle avoidance. This reduction is proportional to the propeller thrust $X_{prop} = [0, 6]N$, as shown in Fig. 6.23. Due to obstacle avoidance with surge velocity smaller than 0.98m/s, after the AUV maneuvered over obstacles, the surge velocity will be increase to 1.28 m/s in remaining time of T2. The other segments of the square path are also similar behavior of the segment BC.

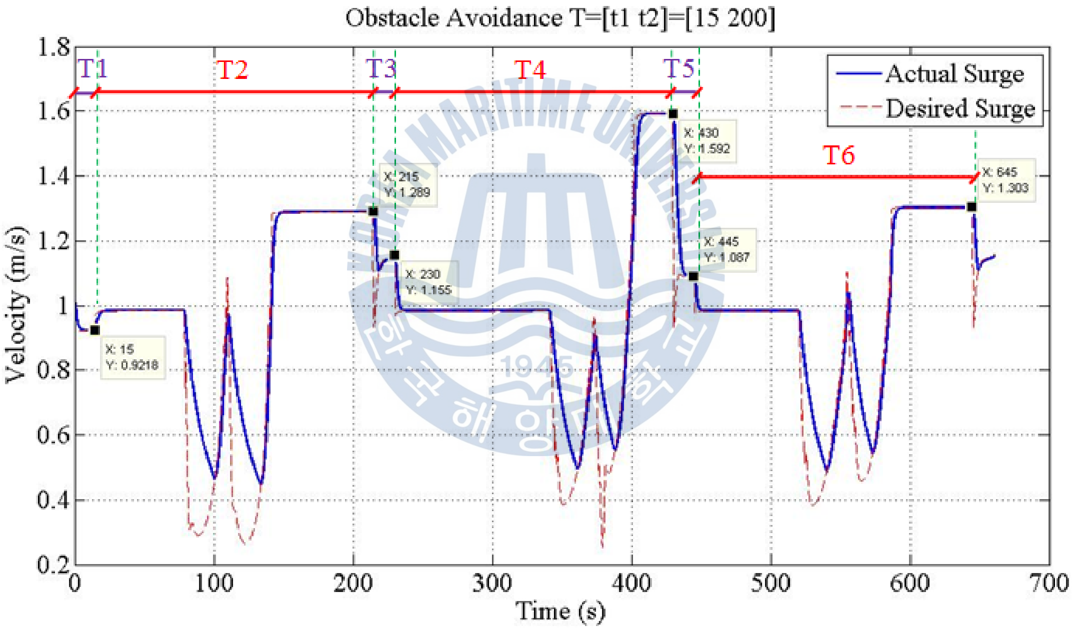

Fig. 6.22 Surge velocity of the AUV with T = [15 200]

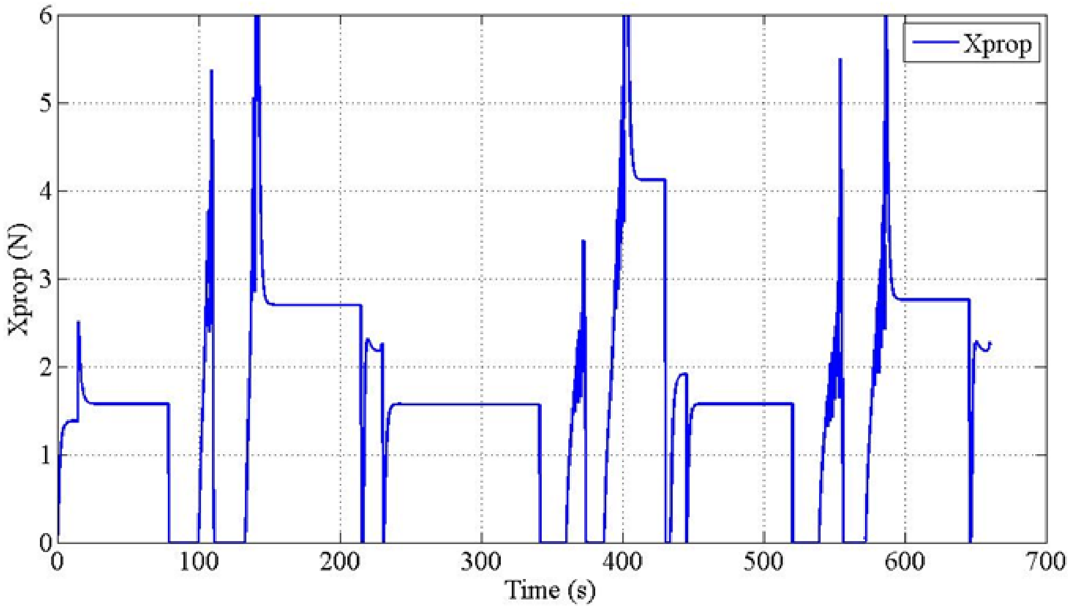

Fig. 6.23 Propeller thrust X_{prop} with T = [15 200]

6.5.3 Obstacle Avoidance in Vertical Plane

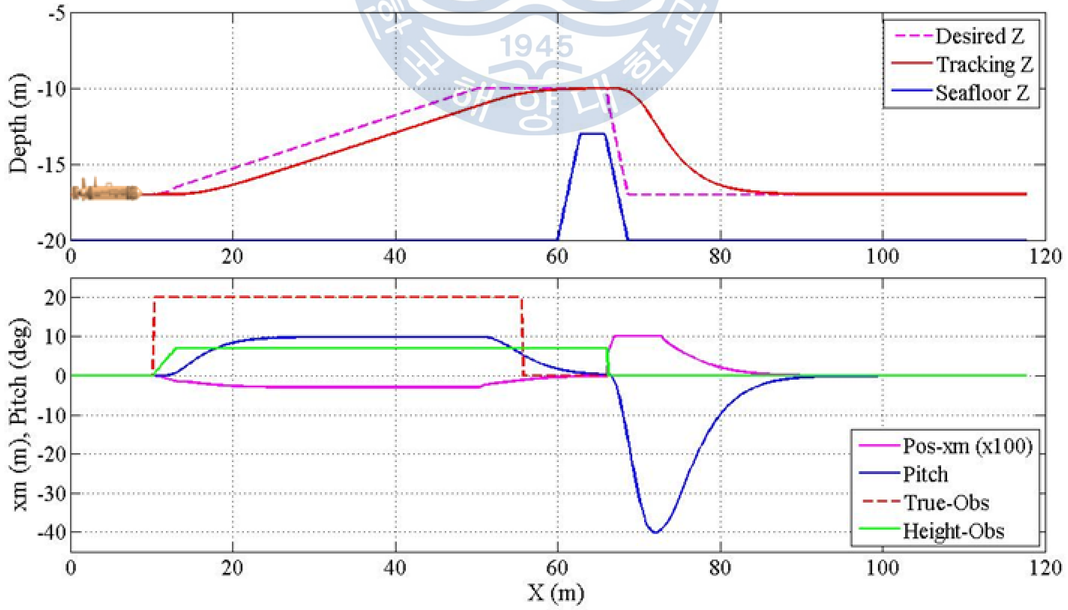

Fig. 6.24 Obstacle avoidance results for single obstacle A

In Fig. 6.28, the final obstacle avoidance algorithm described in chapter 5 was simulated to avoid the single obstacle A. The AUV started at X position of 0 m, depth of 17 m, and initial pitch angle of 0 degree. The surge velocity was kept constant at 1.5 m/s (3 knots). At first, when the obstacle was outside the zone of the sonar's vertical scan, the vehicle was maneuvered to move straight along X-axis and to maintain the altitude of 3 m. Only until the obstacle was detected within the sonar's field of view (True = 1), the obstacle avoidance algorithm would create a new sloping depth command that increased linearly and based on the height of the detected obstacle. The pitch command was also set to zero at all operating time so that the obstacle was always within the zone of the forward looking sonar. According to these commands, the smooth flight path, red solid line of the top graph in Fig. 6.28, was created, and energy was used effectively with gradual changes of the pitch angle and the movable mass position. In addition, at X position of 55 m, due to blind spot of the sonar, the True variable was returned to zero but the AUV was still maneuvered to reach a safe altitude and then successfully to avoid the obstacle. With the multiple obstacles A, the simulation results shown in Fig. 6.29 were also good performance.

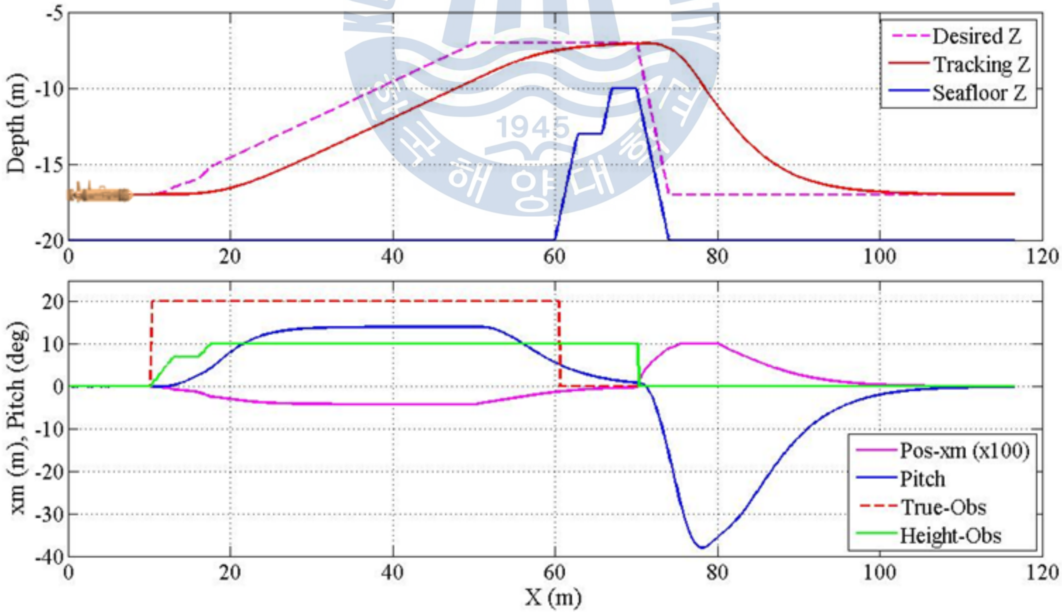

Fig. 6.25 Obstacle avoidance results for multiple obstacles A

To check the adaptability of the obstacle avoidance algorithm with changes in seafloor topography, the other single and multiple obstacles B were simulated respectively in Fig. 6.30 and Fig. 6.31. As a result, the missions were completed, and the characteristics performed for obstacle A were still exhibited in obstacle B.

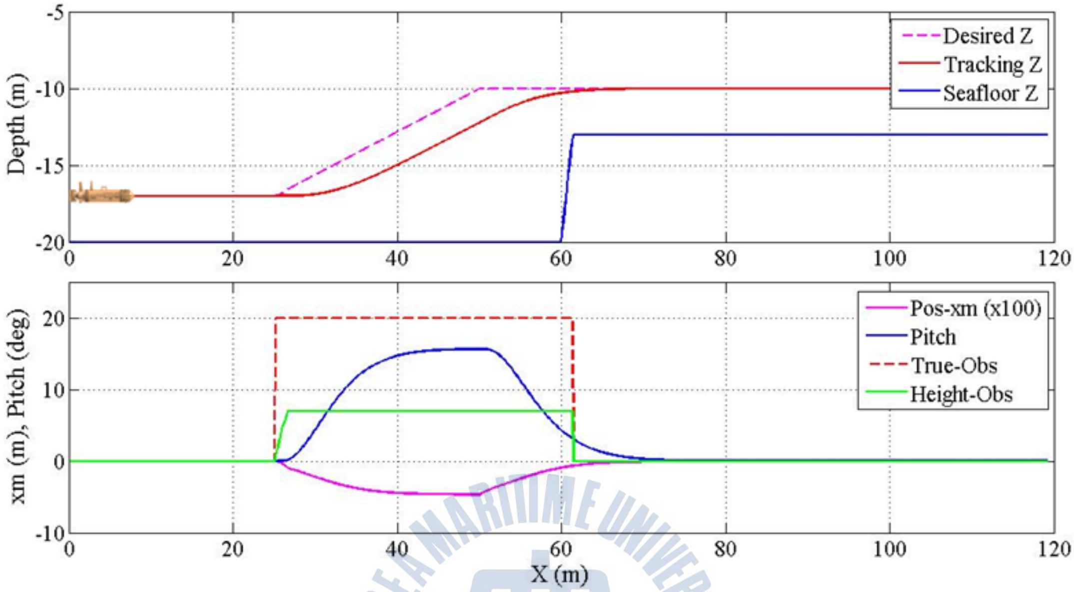

Fig. 6.26 Obstacle avoidance results for single obstacle B

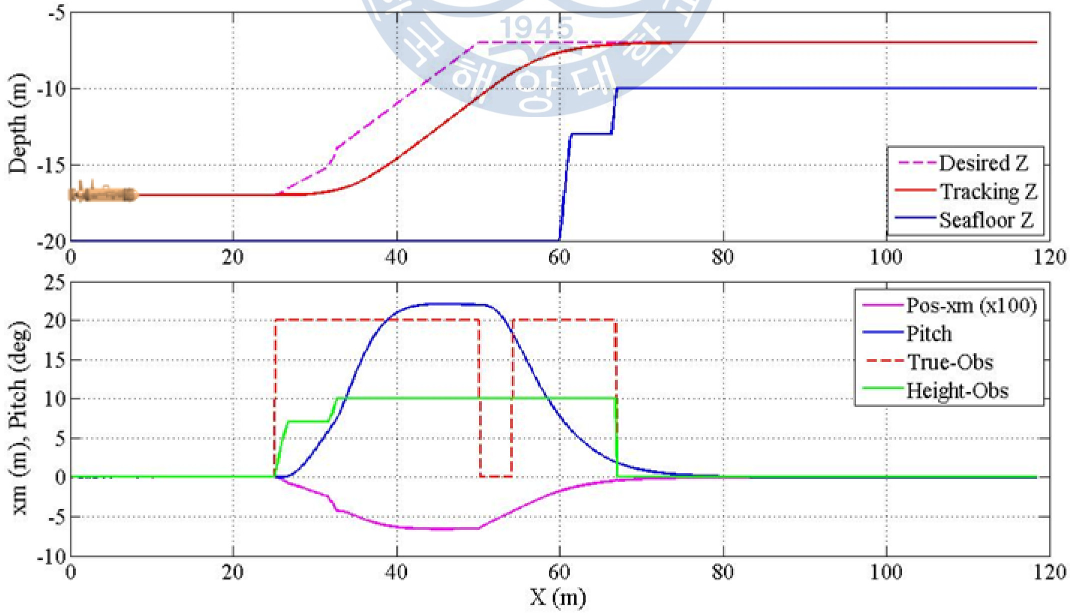

Fig. 6.27 Obstacle avoidance results for multiple obstacles B

Chapter 7: Conclusion

In this dissertation, the design, control, and implementation of the new AUV platform were presented. With the unique propeller and the inside mass shifter mechanism changing the vehicle center of gravity, the AUV can perform horizontal and vertical motions independently. The AUV propulsion system was designed using Maxon Motor 200W and magnetic coupler to rotate propeller and have maximum torque of 3.7 kgf at the ampere of 11 A. The effect of mass shifter dynamics to the pitching motion of AUV was studied through experiments and simulation. According to the result of simulation similar with experiment, the behavior of mass shifter can be predictable through simulation. And also, the weight of movable mass was selected of 2.5 kg with the length of stroke in [-0.1 m, 0.1 m].

The control system including hardware and software architectures were designed, and navigation system composed of the GPS-INS-DVL sensors and the Kalman filter algorithm was also developed. The INS system based on the measured acceleration information of the vehicle can estimate the position, velocity, and attitude of the AUV. But these values have accumulative errors due to inherent drift of the dead-reckoning velocities and integration of the accelerations. Hence the Kalman filter algorithm combined with a number of sensors such as GPS, DVL, and pressure sensor was applied to compensate these errors. Moreover, to verify the developed navigation algorithm, the static positioning and dynamic tracking experiments of GPS-INS and DVL-INS systems were undertaken on the ground before the sea experiment. The navigation results showed the maximum dynamic position error is 0.2 m in motion performance. And when the AUV is kept in static states for a long-term period, the position errors are nearly error.

The nonlinear AUV model with six degrees of freedom was presented and performed linearization to establish depth-plane and heading-plane models. Based on these equations, the multivariable sliding mode control method was used to design heading and depth controllers that were applied to the nonlinear AUV model. In addition, the developed navigation system and designed controllers were applied to heading, waypoint, and depth controls of the AUV through sea experiments and their results were compared with the simulation outputs. There is a little discrepancy between the simulation and experiment results considering the noises and disturbances such as sea currents, waves, etc which were not counted in the simulation. In heading control, the AUV was commanded to track two desired heading angles of 125 degrees and 100 degrees respectively in 20 seconds for each turn. The experimental results showed the overshoot angle was in the range of 5 degrees with rudder angle saturated in ±30 degrees. Applying the designed heading controller and LOS algorithm, the AUV was ordered to track four predefined waypoints as (0, 0), (0, 12), (12, 12), (12, 0) with threshold radius ρ_0 of 1 m. The result of this experiment was quite good and no big difference with simulation result of the waypoints tracking. Finally, using the mass shifter mechanism in bound of stroke's length [-10 cm, 10 cm], the depth control was performed to get the desired depth of 2.5 m and the surge velocity kept constant at 1.5 m/s. Both the experimental and simulated results showed good convergence with a maximum error of 0.3 m.

Obstacle avoidance for underwater vehicles is widely studied for a variety of applications. This dissertation just focuses on a particular application for the developed AUV in vertical and horizontal planes. One of the most critical factors in obstacle avoidance behavior is the ability to discern how a vehicle will react to its environment. So, it is necessary to model the sonar sensor that gathers sufficient environmental data for safe navigation of the vehicle. Through the developed obstacle avoidance algorithm previously mentioned, the AUV has ability to use range and bearing data received from sonar modeling to determine if that return constitutes a threat along its desired path and further navigate around the threat before regaining its original path. Moreover, as increasingly close to obstacle, it is difficult and unsafe to maneuver AUV to avoid the detected objects with constant surge velocity. So the velocity controller with output of the propeller thrust is also considered. The surge velocity is inversely proportional to the bearing and range weighting function of obstacle detection. With developed obstacle avoidance algorithm and designed heading and depth controllers, the simulation results in vertical (obstacles A, B) and horizontal (multiple obstacles on square and zigzag paths) planes shown that the AUV performed the missions of obstacle avoidance successfully. The surge velocity of the AUV was also controlled and the time commanded for desired path was guaranteed.

References

[1] Jones, L.B., “The State of the Technology: Manned Submersibles,” Marine Technology Society (1996).

[2] Juh, J., West, M., (2001), Underwater Robotics, Advanced Robotics, Taylor & Francis.

[3] Woods Hole Oceanographic Institution (WHOI), “Ships and Vehicles Used in 1985 Discovery of Titanic,” 1 March 2004.

[4] Canadian Broadcasting Corporation (CBC News), “Robot Subs Trying to Stop Gulf Oil Leak,” 25 April 2010.

[5] Commander, Naval Sea Systems Command, “U.S. Navy Diving Manual,” U.S. Government Printing Office, Washington, D.C., USA, SS521-AG-PRO-010 Revision 6, 15 April 2008.

[6] http://www.whoi.edu

[7] http://www.bluefinrobotics.com

[8] http://underwater.iis.u-tokyo.ac.jp

[9] Fiorelli, E., Leonard, N.E., Bhatta, P., et al., (2004), “Multi-AUV control and adaptive sampling in Moterey Bay,” Workshop on Multiple AUV Operations (AUV04), pp. 134–147.

[10] Stokey, R., Allen, B., Austin, T., Goldsborough, R., et al., (2001), “Enabling technologies for REMUS docking: an integral component of an autonomous ocean-sampling network,” IEEE Journal of Oceanic Engineering 26 (4), 487–497.

[11] Saravanakumar, S., Thomas G., Asokan T., (2012), Trends in Intelligent Robotics, Automation and Manufacturing, Springer Berlin Heidelberg.

[12] Kukuly, A., et al., (2010), “Under-ice operations with a REMUS-100 AUV in the Arctic,” Proc. AUV 2010 IEEE Conference, Monterey, CA, USA (2010), pp. 1-8.

[13] McEwen, R., (2006), Modeling and control of a variable-length AUV, Technical Report.
[14] Panish, R., (2006), "Dynamic control capabilities and developments of the Bluefin robotics AUV fleet," Proc. 16th International Symposium on Unmanned Untethered Submersible Technology, Durham, NH, USA.
[15] Prestero, T., (2001), Verification of a six-degree of freedom simulation model for the REMUS autonomous underwater vehicle, M.S. Thesis, Massachusetts Institute of Technology, USA.
[16] Fossen, T.I., (1994), Guidance and Control of Ocean Vehicles, John Wiley & Sons Ltd., New York.
[17] Healey, A.J., Lienard, D., (1993), "Multivariable Sliding Mode Control for Autonomous Diving and Steering of Unmanned Underwater Vehicles," IEEE Journal of Oceanic Engineering 18 (3), 327–339.
[18] Titterton, D.H., Weston, J.L., Strapdown inertial navigation technology, 2nd edition, American Institute of Aeronautics and Astronautics, 2004.
[19] Ho-Sung Kim, Hyeung-Sik Choi, "Study on AHRS Sensor for Unmanned Underwater Vehicle," International Journal of Ocean System Engineering (KSOE), Vol. 1, No. 3, pp. 165~170.
[20] Greg Welch, Gary Bishop, "An Introduction to the Kalman Filter," University of North Carolina, Department of Computer Science, 2001.
[21] Zarchan, P., Musoff, H., "Fundamentals of Kalman filtering: A practical approach," 2nd edition, American Institute of Aeronautics and Astronautics, 2005.
[22] Tran, N.H., Choi, H.S., Kim, J.Y., Lee, M.H., (2012), "Development of Rotational Motion Estimation System for a UUV/USV based on TMS320F28335 microprocessor," Int. J. of Ocean System Engineering, 2(4), pp. 223-232.
[23] Slotine, J.J., Li, W., (1991), Applied Nonlinear Control, Prentice Hall.
[24] Healey, A. J., Marco, D. B., "Slow Speed Flight Control of Autonomous Underwater Vehicles: Experimental Results with NPS AUV II," Proceedings of the 2nd International Offshore and Polar Engineering Conference, San Francisco, July 14-19 1992.

[25] Fodrea L.R., Healey, A.J., “Obstacle Avoidance Considerations for the REMUS Autonomous Underwater Vehicle,” ASME 2003 22nd International Conference on Offshore Mechanics and Arctic Engineering, Mexico, June 8–13, 2003

[26] Tran, N.H., Choi, H.S., et al., (2014), “Navigation and Depth Control of a Small-Sized AUV with Mass Shifter Mechanism,” Proceeding of the 24th International Ocean and Polar Engineering Conference (ISOPE), Busan, Korea, June 15-20, 2014.

[27] Tran, N.H., Choi, H.S., et al., (2014), “Design, Control, and Implementation of a new AUV platform with a weight balance,” Proceeding of the International Symposium on Green Manufacturing and Application (ISGMA), Busan, Korea, June 24-28, 2014.

[28] Kanayama Y., and Hartmann B.I., “Smooth Local Path Planning for Autonomous Vehicles” Autonomous Robot Vehicles, Springer-Verlag, ISBN 000-387-97240-4, pp. 62-68, 1990.

[29] Allen B., Stokey R., Austin T., et al., “REMUS: A Small Low Cost AUV: System Description, Field Trials, Performance Results,” Proceedings IEEE Oceans 97, pp. 994-1000, 1997.

[30] Moitie, R. and Seube, N., “Guidance Algorithms for UUVs Obstacle Avoidance Systems”, OCEANS 2000, Brest, France.

[31] Park J.Y., Jun B.H., et al., “Development of Test-Bed AUV ‘ISiMI’ and Underwater Experiments on Free Running and Vision Guide Docking,” Underwater Vehicle, INTECH Open Access Publisher, pp. 582, 2009.

[32] Lane, David M., Petillot, Yvan, and Ruiz, Ioseba Tena, “Underwater Vehicle Obstacle Avoidance and Path Planning Using a Multi-Beam Forward Looking Sonar,” IEEE Journal of Oceanic Engineering, Vol. 26, Apr 2001.

[33] Horner, D.P., Healey, A.J., Kragelund, S.P., “AUV experiments in obstacle avoidance,” Proceedings of MTS/IEEE, OCEANS 2005, Washington, DC, 17-23 Sept. 2005.