

공 학 석 사 학 위 논 문

AUV의 개념설계를 위한 운동해석



2011년 2월

부 경 대 학 교 대 학 원

메 카 트 로 닉 스 공 학 과

현 지 민

공 학 석 사 학 위 논 문

AUV의 개념설계를 위한 운동해석



지도교수 백운경

이 논문을 공학석사학위논문으로 제출함.

2011년 2월

부경대학교 대학원

메카트로닉스공학과

현지민

현지민의 공학석사 학위논문을
인준함

2010년 12월 27일



주 심 공학박사 안 두 성 ㉠

위 원 공학박사 강 인 필 ㉠

위 원 공학박사 백 운 경 ㉠

목 차

Abstract	1
1. 서론	2
1.1 연구 배경	2
1.2 연구 목적 및 범위	3
2. 자율무인잠수정(AUV)의 개념설계.....	5
2.1 AUV의 요구 운동 성능.....	5
2.2 AUV의 선체 및 구조설계.....	6
2.3 AUV의 추진기 및 방향타 설계	7
2.4 AUV의 제어 및 통신 시스템	13
2.5 AUV의 센서 시스템	14
2.6 AUV의 전력 및 전원 시스템	15
2.7 AUV의 밸런싱 설계	16
3. AUV의 동역학 모델.....	21
3.1 AUV의 6자유도 동역학 모델	21
3.2 AUV의 6자유도 운동방정식	29
3.3 AUV의 X-Y 평면 운동방정식	30
3.4 AUV의 X-Z 평면 운동방정식	33
4. AUV의 운동 성능 해석	37
4.1 AUV의 운동시뮬레이션 프로그램.....	37
4.2 AUV의 평면 운동 시뮬레이션.....	39
4.3 AUV의 수직 운동 시뮬레이션	42
4.4 AUV의 Way-point 추적 시뮬레이션.....	45
5. 운동 시뮬레이션 결과 고찰	48
6. 요약 및 결론	49
부록: AUV의 운동성능 해석용 Matlab Code.....	50
참고문헌	59

List of Tables and Figures

Table 1	Dimension of AUV	7
Table 2	Power system of AUV	15
Table 3	Twelve state variables of an AUV with 6 degrees of free	22
Table 4	Non-Linear Maneuvering Coefficients : Forces	27
Table 4	Non-Linear Maneuvering Coefficients : Moments	28
Table 6	Six state variables of an AUV with X-Y plane	31
Table 7	Six state variables of an AUV with X-Z plane	34
Fig. 1	옥포6000, SAUV, 해미래, 해누리	3
Fig. 2	Concept design of the AUV	6
Fig. 3	직진 운동 중 횡단면	7
Fig. 4	선회 운동 중 횡단면	8
Fig. 5	Thruster model	9
Fig. 6	추진계수 그래프	10
Fig. 7	30° 선회중의 방향타의 단면적	11
Fig. 8	방향타의 축에 발생하는 토크	12
Fig. 9	방향타 제어 서보 모터	12
Fig. 10	제어 및 통신시스템 구성도	13
Fig. 11	IMU	14
Fig. 12	DVL	14

Fig. 13 수압센서	14
Fig. 14 누수감지센서	14
Fig. 15 리튬 폴리머 배터리	16
Fig. 16 AUV의 경심	18
Fig. 17 AUV의 잠수 운동	18
Fig. 18 AUV의 잠수 운동시의 방향타에 발생하는 외부 모멘트	20
Fig. 19 Body-fixed coordinate system	21
Fig. 20 X-Y Planar motion of the AUV	31
Fig. 21 X-Z Planar motion of the AUV	34
Fig. 22 AUV motion simulation block diagram	38
Fig. 23 Way-point simulation flow chart	38
Fig. 24 X-Y turing simulation results of the X-Y position	39
Fig. 25 X-Y turing simulation results of the X-Z position	40
Fig. 26 X-Y turing simulation results of the angular velocity $r(\text{deg/s})$ ·	41
Fig. 27 X-Y turing simulation results of the angle $\psi(\text{deg})$	41
Fig. 28 X-Z depth simulation results of the X-Z position	42
Fig. 29 X-Z depth simulation results of the X-Y position	43
Fig. 30 X-Z depth simulation results of the angular velocity $q(\text{deg/s})$ ·	44
Fig. 31 X-Z depth simulation results of the angle $\theta(\text{deg})$	44
Fig. 32 Way-point Simulation Result 1	46
Fig. 33 Way-point Simulation Result 2	46
Fig. 34 Way-point Simulation Result 3	47

Fig. 35 Verify the Way-point Simulation 48



Motion Analysis for Concept Design of the AUV

Ji-Min Hyun

Mechatronics Engineering, The Graduate School.

Pukyong National University

Abstract

AUVs(Autonomous Underwater Vehicles) are used for various scientific researches, commercial and military applications in the recent years. The structure of the AUVs consists of various parts (such as main body, thruster, rudder, power pack, camera, sensor, etc.) The body of the AUV must resist against the underwater hydraulic pressure. The thruster at the stern of the AUV must push the main body at an appropriate speed. The rudders controls the direction of the AUV. The battery must support a sufficient power for the operation of the AUV.

In this thesis, the basic concept for the motion performance of AUVs is described for the initial design stage. This thesis also describes the mathematical model of the dynamic motion of the AUV. Finally, based on the mathematical model, the AUV's dynamic motions were simulated using Matlab code to verify the initial design concept.

Keywords : AUV(Autonomous Underwater Vehicle), Basic Design, Motion Performance

제 1 장 서론

1.1 연구 배경

만물의 근원이라는 바다는 지구의 71%를 차지한다. 하지만 의외로 바다에 대해 밝혀진 것은 그리 많지 않다. 바다를 탐사하기 위해서는 많은 기술이 요구되기 때문이다. 바다는 10m 아래로 내려갈 때마다 1기압씩 높아져, 인간이 들어갈 수 있는 깊이는 보통 30m, 훈련된 사람의 경우 70m 정도에 불과하다. 사람은 지상 1만m 가까이 올라갈 수 있으나, 해저 70m조차도 내려갈 수 없다.

바다 탐사를 위한 과학 기술이 발전함에 따라 무인 잠수정인 ROV(Remotely Operated Vehicle)와 AUV(Autonomous Underwater Vehicle)를 개발되기에 이르렀다¹⁾. ROV는 오퍼레이터가 잠수정에 부착된 수중 카메라의 정보를 이용하여 직접 조종하는 잠수체로 수중에서의 작업, 항만이나 연안호수 등 수심이 얇은 곳의 검사에 주로 쓰인다. AUV는 자신의 위치와 자세 그리고 주위 환경 변화를 탑재된 센서를 통하여 스스로 인지한 후 주행경로를 자율적으로 설정하고 주어진 임무를 수행하는 자율성을 지니는 무인 잠수정을 말한다^{3~11)}.

무인잠수정은 해저 자원 탐사, 군사 목적, 자원 개발 등 여러 가지 목적을 가지고 개발되었으며 우리나라에서도 1986년 250m급 유인탐사정 '해양250' 개발을 시작으로 해양연구원은 1993년 해저탐사를 위한 수중로봇 '씨로브300'을 개발했다. 이후 1996년 대우조선이 자율항해무인잠수정 '옥포6000'을 개발했다.

이는 1997년 한국해양연구원의 시험용 자율항해무인잠수정 보람호, 2003년 민군겸용 무인잠수정(SAUV ; Semi-Autonomous Underwater Vehicle) 개발 및 2007년 6000m급 심해 잠수정 해미래/해누리 개발로 이어졌다^{12~15)}.



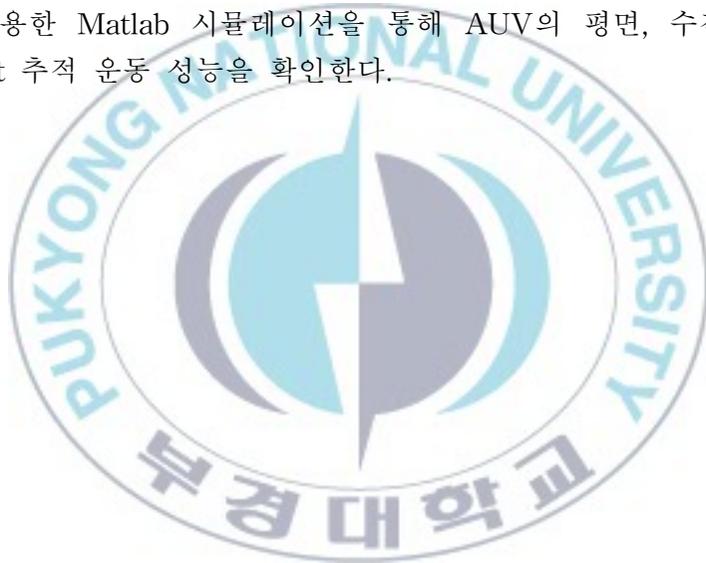
Fig. 1 옥포6000, SAUV, 해미라, 해누비

1.2 연구 목적 및 범위

AUV의 설계에서 최우선적으로 결정되어야 할 사항은 AUV에 부여되는 임무(Mission)이다. 우선 임무를 정한 후 해당 임무에 적합한 형상 및 제어방식, 적재 중량, 최대 잠수 깊이, 운항 속도, 작동시간 등을 설정하고 구체적인 세부사항을 결정해 나가는 과정으로 설계가 이루어진다.

적재 중량은 AUV 내부에 구성되는 컨트롤러, 배터리 및 각종 센서 등의 중량을 모두 합친 중량이다. 최대 잠수 깊이는 AUV의 수밀 능력에 좌우되고 운항속도는 AUV의 항력 및 추력의 추정을 통해 결정된다. 본 연구에서는 AUV 개념설계를 통해 AUV를 구성하는 각 하드웨어들을 설계 및 선정, AUV System을 구성하고 AUV의 운동성능을 해석을 통해 검증하는 것을 목적으로 한다.

본 연구에서는 AUV의 임무를 성공적으로 수행할 수 있는 운동성능 구현을 목표로 각각의 선체, 방향타, 추진기, 전원(배터리) 등의 개념설계를 수행하였다. 또한 AUV의 제어 및 통신시스템을 구상하였고 AUV의 부력 및 복원력을 고려하여 밸런싱 설계를 수행하였다. 그리고 AUV의 운동방정식을 이용한 Matlab 시뮬레이션을 통해 AUV의 평면, 수직 운동 및 Way-point 추적 운동 성능을 확인한다.



제 2 장 자율무인잠수정(AUV)의 개념설계

2.1 AUV의 요구운동성능

본 논문에서 설계하는 AUV의 요구운동성능은 다음과 같다.

첫째, 추가적인 센서나 장비의 탑재를 위한 10%의 여유부력을 두고 설계 인양이 가능하도록 공기 중 무게 35 kgf 정도의 범위 내에서 설계/제작되도록 한다.

둘째, 약 5 knot의 속도로 2시간 이상의 운항이 가능하여야 하고 수심 300 m의 깊이에서도 견딜 수 있도록 한다.

셋째, 선체 외형이 수학적모델의 추정이 용이하고 추진 효율을 높일 수 있도록 현재 실험적으로 알려진 선형을 이용하여 유체저항 계수가 적고 충분한 탑재 공간을 확보할 수 있도록 설계하도록 한다.

넷째, 탑재되는 주 처리 컴퓨터는 임무수행을 위한 충분한 계산능력을 갖도록 능력을 갖추도록 한다.

다섯째, 탑재장비나 부 시스템의 고장과 오동작의 빈도를 최소화하고 신뢰성을 높이기 위해 성능이 검증된 장비가 선정되어야 하고 지속적이고 안정적으로 공급되는 장비들을 선정하여 고장발생시 즉각적인 대처가 가능하도록 한다.

여섯째, 기본 기능에 포함되어 있지 않은 기능을 추가로 확장할 경우 기계적으로나 전기적인 기본 설계의 변경을 최소화 하면서 기능의 확장이

가능하도록 한다.

2.2 선체 구조 설계

AUV는 Fig.2 와 같이 선수부, 중앙부, 선미부로 나누어지고, 중앙부는 원통형 실린더이며, 선수부는 용력을 최소화할 수 있는 반구 형태로 제작된다.

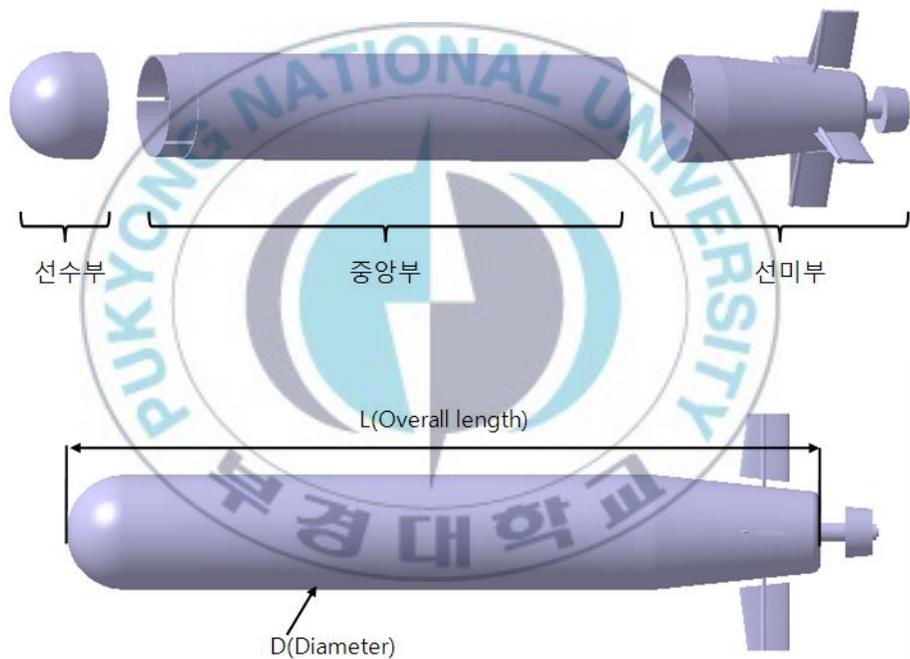


Fig. 2 Concept design of the AUV

AUV Concept design은 CATIA V5로 modeling 했으며 선체의 재질을 알루미늄으로 선정하였다. 이때 선체 길이(L)과 선체의 직경(D), 선체 부피 및 중량, 부력은 Table.1과 같다.

Table.1 Dimension of AUV

Parameter	Value	Unit	Description	Remarks
L	1485	mm	Overall length	
D	250	mm	Diameter	
V	0.0771	m^3	Volume	
W	34.263	kg	Weight	
B	781.39	N	Buoyancy	$\rho=1025$

2.3 AUV의 추진기 및 방향타 설계

본 연구에서는 AUV의 운동부는 선미부에 부착된 추진기와 방향타 4개로 구성되어 있다. 이러한 AUV의 횡단면을 Fig. 3, Fig. 4와 같이 나타내었다. Fig. 3은 직진 운동을 할 경우 횡단면의 단면적 이고 Fig. 4는 방향타가 최대한 회전했을 경우인 30° 선회했을 경우의 단면적이다. Fig.3 및 Fig.4의 단면적 계산 값은 식(1), 식(2)와 같다.



Fig. 3 직진 운동 중 횡단면

- 선체 단면적 = $0.25 \times 0.25 \times \pi = 0.049m^2$
 - 방향타 단면적 = $4 \times 0.11 \times 0.01 = 0.0044m^2$
 - 전체 단면적 = $0.049 + 0.0036 = 0.0526m^2$
- (1)

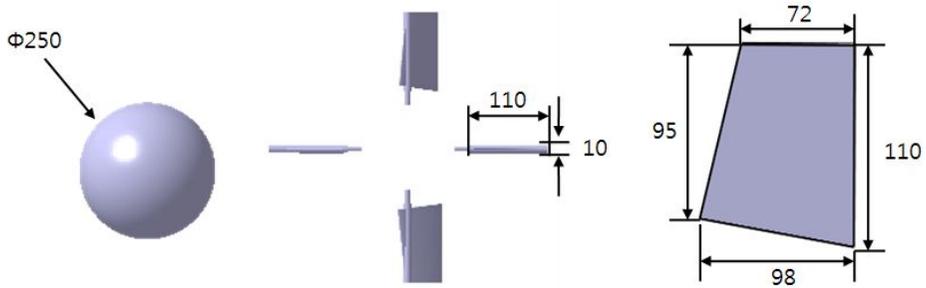


Fig. 4 선회 운동 중 횡단면

- 선체 단면적 = $0.25 \times 0.25 \times \pi = 0.049m^2$ (2)
- 방향타 단면적 = $2 \times 0.11 \times 0.01 = 0.0022m^2$
- 방향타 단면적(30°)
 $= 2 \times \{0.5 \times (0.11 + 0.095) \times 0.072 + 0.5 \times 0.095 \times 0.026\} \cos 30^\circ$
 $= 0.01492m^2$
- 전체 단면적 = $0.049 + 0.0022 + 0.01492 = 0.06612m^2$

본 연구에서 AUV의 형상은 REMUS의 형상과 매우 유사하게 설계하였다. 본 연구의 AUV의 유체저항계수를 도출하기 위해선 실제 제작 후 시험을 통해 도출해야 하나 그 기간이 길고 어렵기 때문에 비슷한 형상의 REMUS의 시험 측정 유체저항계수인 $c_d=0.3$ 을 적용하였다. 바다물의 밀도를 1025 kg/m^3 , AUV의 목표속도를 $5 \text{ knot}(2.55\text{m/s})$ 로 하면 AUV에 걸리는 저항 값은 (3)식과 같이 계산할 수 있다.

$$F_{AUV} = 0.5 \times 0.3 \times 0.06612 \times 1025 \times 2.55^2 = 52.59N \quad (3)$$

방향타를 30° 회전을 할 경우 AUV의 최고 목표 속도는 $3 \text{ knot}(1.53\text{m/s})$ 이며 이때 발생하는 저항 값은 (4)식과 같이 계산할 수 있다.

$$F_{AUV} = 0.5 \times 0.3 \times 0.06612 \times 1025 \times 1.53^2 = 23.80N \quad (4)$$

식 (3), (4) 계산 결과를 보면 알 수 있는 것과 같이 AUV의 최고 속도인 5 knot 로 운동을 할 때 발생하는 유체저항 값이 3 knot 로 회전할 때 발생

하는 유체저항 값보다 더 크기 때문에 5 knot에서 발생하는 유체저항 값을 기준으로 추진기를 선정한다.

AUV의 소요 동력은 AUV의 추진력 F_{AUV} 와 AUV의 속도 V 를 곱하여 계산 할 수 있다.

$$P_{AUV} = 52.59 \times 2.55 = 134.1 W \quad (5)$$

식(5) 로 계산한 소요 동력은 AUV가 5 knot(2.55m/s)로 움직일 수 있는 최소값이다.

추진기는 AUV의 추진기의 적합한 추진 시스템을 구성하기 위해 350 W 급의 출력을 가진 모델을 설계하며, 사용되는 전동기는 BLDC 모터이며 내압방수 구조를 갖는다. 공기 중의 중량은 약 1.5 kgf이며 수중 중량은 0.9 kgf 이다. 속도제어 명령은 ± 5 VDC 이며, 사용심도는 300미터 급으로 한다.

AUV의 소요 동력인 134.1W 이상의 출력을 만족하기 위해 Fig. 5와 같은 추진기를 검토해 보았다. 배터리로 구동하기 위하여 DC 전원으로 선정하였으며 정격은 DC 24-330V/ 325-350W 이다.



Fig. 5 Thruster model

Fig. 5의 추진기의 직경과 피치는 각각 $D=76, P=51 \sim 53mm$ 이고 P/D 는 약 0.7 이다. 이 때에 Fig.6의 그래프를 이용하여 $K_T, 10K_Q$ 를 구할 수 있다. (J_a 는 기준 값인 0.5를 사용)

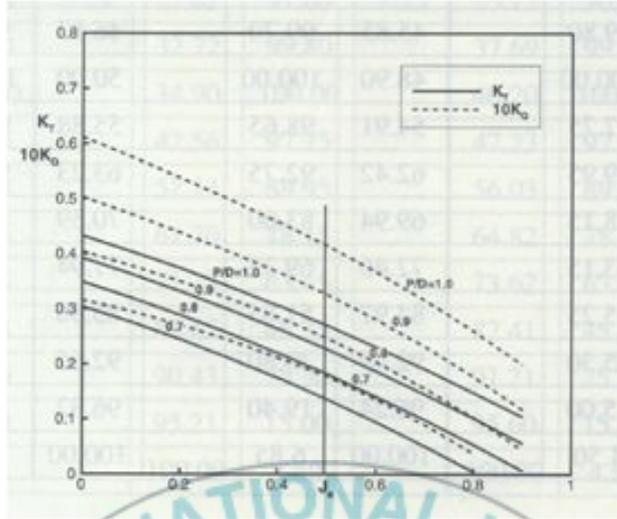


Fig. 6 추진계수 그래프

구해진 $K_T, 10K_Q$ 의 값은 각각 0.17 및 0.2 이다. 이 계수를 이용하여 추진기의 단독 효율을 계산하면 식 (6)과 같다.

$$\eta = \frac{J_a \times K_T}{2\pi \times K_Q} = \frac{0.5 \times 0.17}{2\pi \times 0.02} = 0.6764 \quad (6)$$

선정된 추진기는 식(6)로 계산한 결과 약 67.6%의 효율을 가지는 것을 알 수 있다. 선정된 추진기의 동력이 350 W이므로 실제 추진기에서 발생하는 동력은 (7)식과 같다.

$$P = 350 \times 0.676 = 236.6 W \quad (7)$$

식(7)를 보면 실제 추진기에서 발생하는 동력은 236.6 W 이며 AUV가 5 knot(2.55m/s)로 운동할 때 발생하는 저항력을 극복할 수 있는 동력 값인 134.1W 보다 크므로 충분한 성능을 가지는 것을 확인 할 수 있다.

AUV의 방향타 회전시의 소요 동력은 방향타에 걸리는 저항 값 F_{rudder} 와 AUV의 속도 V 를 곱하여 계산 할 수 있다. Fig.7과 같이 방향타가 단독으로 30° 회전할 경우 방향타의 단면적은 식(2)를 통해 알 수 있고 AUV의 최대 속도인 5 knot(2.55m/s)로 운동중이라고 가정하면 방향타에는 (8)식과 같은 유체 저항이 발생한다.

$$F_{rudder} = 0.5 \times 0.3 \times 0.00746 \times 1025 \times 2.55^2 = 7.46N \quad (8)$$

(8)식으로 계산한 유체 저항 값은 방향타가 30° 회전할 경우 AUV 최고 속도가 5 knot(2.55m/s)로 움직일 경우 발생하는 저항 값이다.



Fig. 7 30° 선회중의 방향타의 단면적

방향타의 유체저항은 방향타에서 토크를 발생시킨다. Fig.8과 같이 유체 저항력 F_{rudder} 가 방향타 길이 중심에 작용한다고 가정하면 방향타 길이 중심과 유체저항으로 인해 식(9)와 같이 토크가 발생한다.

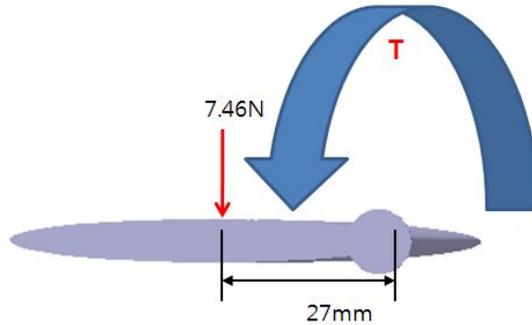


Fig. 8 방향타의 축에 발생하는 토크

$$T = 7.46 \times 27 = 201.42 \text{ Nmm} \quad (9)$$

AUV의 자세 제어용 Rudder, elevator를 동작시키는데 사용되는 서보모터는 정밀한 각도로 제어가능하며 강한 토크를 내야한다. 일반적으로 서보모터는 Open Loop 제어가 가능하며(속도, 위치제어), 정지토크가 우수하다. step motor 드라이브는 저 전력으로 구동될 수 있도록 설계한다.

방향타 제어용 서보모터는 Fig. 9를 검토하였다. 이 서보모터는 크기가 작으며 식(9)에서 계산된 토크보다 큰 토크를 발생하므로 방향타를 회전하는데 적합하다고 판단된다.



HDS - 2288		
Motor type	Coreless motor	
Type of Gear	Metal gear	
Operating Voltage	4.8 V ~ 7.4 V	
Neutral pulse	1.5 .. (Adjustable)	
Travel angle	40~45° at ±400.. (Adjustable)	
Dead Band	1 ~ 2 .. (Adjustable)	
Speed	(sec/60° at 4.8V)	0.17
	(sec/60° at 6.0V)	0.15
	(sec/60° at 7.4V)	0.12
Torque	(kg/cm at 4.8V)	16.0
	(kg/cm at 6.0V)	20.0
	(kg/cm at 7.4V)	25.0
Weight (g)	47	
Dimensions (mm)	40x20x34	
Ball bearing	Basic(2BB)	

Fig. 9 방향타 제어 서보 모터

2.4 제어 및 통신 시스템

시스템제어기의 출력에는 하나의 추진모터부와 4개의 Rudder모터부가 연결되어 있고, 입력부에는 RF모뎀과 GPS신호 수신기, DVL센서가 연결된다. 이 모든 요소의 동작 전원 레벨이 다르므로 Battery의 전원을 각각의 요소의 동작전원 레벨에 맞추는 DC/DC변환기로 구성된다. 또한 추진모터부는 속도센서 일체형으로 구성되고, Rudder 모터부는 각도센서 일체형으로 구성한다.

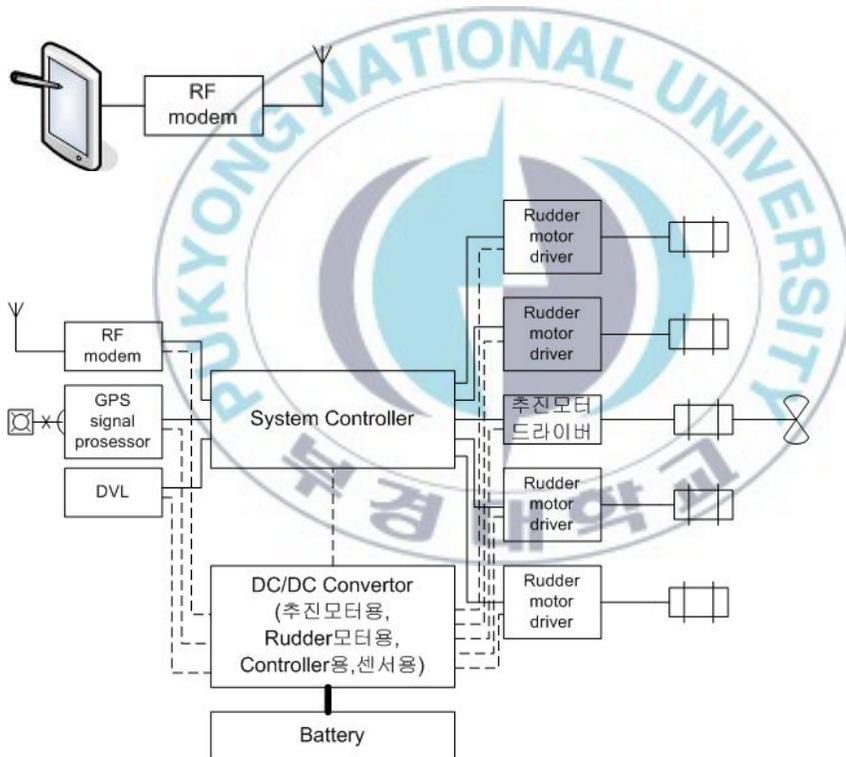


Fig. 10 제어 및 통신시스템 구성도

2.5 센서 시스템

AUV에 장착되는 센서는 자세를 측정하는 IMU와 3차원 속도를 계측하는 도플러 속도계(DVL), 그리고 수심을 측정하는 압력센서와 누수를 감지하는 누수감시 센서등이 있다. 각각의 형상은 Fig.11-14와 같다.



Fig. 11 IMU



Fig. 12 DVL



Fig. 13 수압센서



Fig. 14 누수감지센서

2.6 전력 및 전원시스템

AUV는 모선과 독립적으로 운항하기 때문에 모선으로부터 테더 케이블을 통해 전력이 공급되는 ROV와는 달리 동력원을 독립적으로 보유하고 있어야 한다. 따라서 AUV의 운용에 있어서 가장 중요한 문제 중의 하나로 자체 동력원인 배터리설계가 중요한 문제로 부각된다. 왜냐하면 사용되는 배터리의 용량에 따라서 AUV의 운항 범위 및 시간이 결정되기 때문이다.

본 연구에서는 AUV의 동력원으로 재충전이 가능한 리튬폴리머전지를 사용한다. 리튬폴리머전지는 에너지 밀도가 높고 자기 방전율이 낮으며 동일 체적, 무게도 용량도 크다. 그리고 메모리 효과가 없어서 사용상 편리하고 수명이 길다.

AUV 전체 시스템이 최대 출력 기준(5 knot) 2시간 이상 작동하기 위해서 충분한 용량을 가진 배터리 선정이 중요하다. AUV 전체 시스템은 Table.x와 같이 구성되며 각 Part에서 소모되는 총 전력은 403.42W 이다. 이 경우 AUV의 요구 전력량은 식 (10)과 같이 계산할 수 있다.

$$Wh = P \times h = 403.42 \times 2 = 806.84 Wh \quad (10)$$

Table.2 Power system of AUV

No.	장치명	전력(W)	전압(V)	전류(A)
1	DC Brushless Thrusters (Model 300)	350	24	14.6
2	Servomotor(HDS-2288) x 4	20	5	4
3	RF(Probee-ZS10)	1	5	0.2
4	Wince(IEC667-07B)	12	24	0.5
5	ATXmega	0.66	3.3	0.2
6	TMS320F28335	0.66	3.3	0.2
7	IMU (Mti)	8.4	24	0.35
8	DVL(NavGuest600M)	4.8	24	0.2
9	GPS (AseRx1)	1.2	5	0.24
10	DepthSensor(미정)	1.5	5	0.3
11	누수감지센서(미정)x2	3.0	5	0.6
Total		403.42	-	-

식 (10)과 같이 AUV 전체 시스템에 소요되는 전력량을 계산하면 약 806.84 Wh가 된다. 이 값을 기준으로 배터리를 Fig. 15과 같이 선정하였다. 배터리 사양은 18.5 V, 6 Ah 이며 총 8개를 Set으로 구성하였다. 이때 배터리가 충전할 수 있는 전력량은 식 (11)와 같다.

$$Wh = 18.5 \times 6 \times 8 = 888 Wh \quad (11)$$

이는 배터리의 성능이 설계된 806.84 Wh보다 더 크므로 충분한 성능을 가진다.



Fig. 15 리튬 폴리머 배터리

2.7 AUV의 밸런싱 설계

AUV의 부력은 식 (12) 와 같이 AUV의 전체 체적(V)에 물의 밀도(ρ)와 중력가속도(g)를 곱하여 구할 수 있다.

$$B = \rho g V \quad (12)$$

Fig. 2와 같은 3D 모델의 AUV의 전체 체적은 $0.07771m^3$ 이고 AUV 전체의 무게는 34.263Kg이다. 여기에 바닷물의 밀도값($1025Kg/m^3$) 및 중력

가속도($9.81m/s^2$)를 적용하면 식 (13)과 같이 계산할 수 있다.

$$1025 \times 9.81 \times 0.07771 = 781.39N \quad (13)$$

식 (13)에서 계산된 부력을 기준으로 10% 여유 부력을 가질 수 있도록 식(14)와 같이 계산할 수 있다.

$$(781.39 \times 0.9) - (34.263 \times 9.81) \div 9.81 = 37.42Kg \quad (14)$$

식 (14)와 같이 AUV의 내부에 탑재 가능한 무게는 37.42Kgf 로 계산된다.

AUV가 물 위에 떠있을 때 살짝 기울어지면 다시 제자리로 돌아오게 된다. 이 때의 회전중심이 경심이다. 경심의 위치는 AUV의 안정성에 큰 영향을 주는 요소이다. Fig.16에서 AUV가 해면에 똑바로 떠 있을 때 무게중심(G)과 부력중심(B)은 동일 연직선상에 있다. AUV가 기울었을 때 해면 아래에 있는 부분은 모양이 변하므로 부력중심이 이동(B1)한다. 이 때 선체에 작용하는 힘은 수직 하향 중력과 상향의 부력에서 작용점이 다르기 때문에, 짝힘으로 작용한다. 이것이 AUV에 대한 복원력이 되기 위해서는 반드시 기울기의 중심이 무게중심보다 위에 있어야 한다. 기울기 중심의 높이(h)가 높을수록 복원력이 커서 기운 AUV가 빨리 원위치로 돌아오지만 기울기 중심 높이가 낮으면 복원력이 작기 때문에 원위치로 돌아오는 속도가 늦다. $B=mg=W$ 일 경우 식(15),(16)의 2차 미분방정식을 보면 알 수 있는 것과 같이 θ 가 미소각일 경우 기울기 중심 높이는 AUV의 롤링(rolling) 주기와도 관계가 있는데, 그것이 작을수록 롤링 주기가 커서 AUV의 흔들림이 작다. 반대로 그것이 큰 경우에는 주기가 작기 때문에 AUV가 심하게 움직이고 흔들린다.

$$\ddot{\theta} = -(B \sin \theta)h \quad (15)$$

$$\ddot{\theta} + Wh \sin \theta = 0 \Rightarrow \ddot{\theta} + Wh \theta = 0 \quad (16)$$

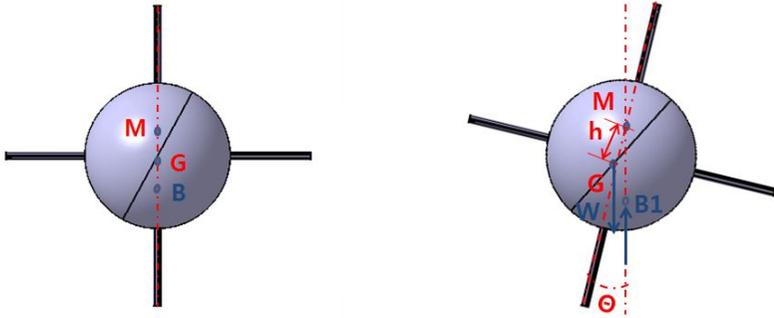


Fig.16 AUV의 경심

AUV가 물 속으로 잠수 하기 위해서는 AUV의 복원력보다 큰 모멘트가 발생하여야 한다. Fig. 17에서 내부 복원 모멘트(RM)는 식(17)과 같다.

$$RM = W \times GZ = GZ \times F_B \quad (17)$$

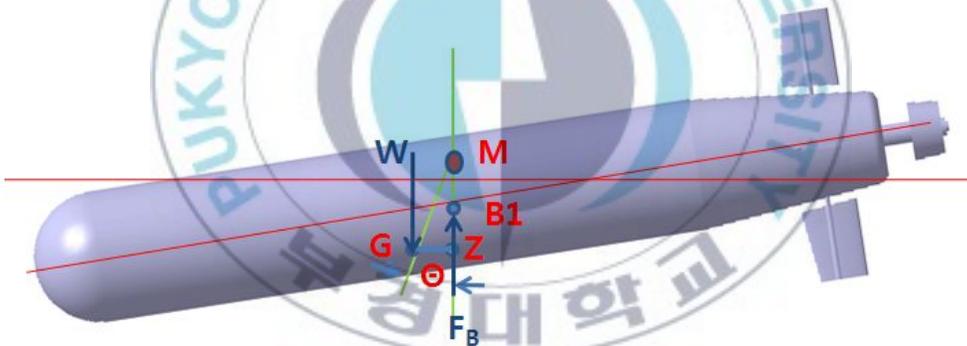


Fig. 17 AUV의 잠수 운동

AUV에서 내부 복원 모멘트(RM)보다 큰 모멘트를 발생하기 위하여 방향타를 이용한다. Fig.17에서 AUV가 최고 속도로 이동할 경우 식(18)과 같이 AUV에서 발생하는 F_{lift} 가 방향타에 집중되며 부심을 통과하는 수평선과 방향타 중심의 거리(L)에서 발생하는 외부 모멘트(M)가 발생된다.

$$M = L \times F_{lift} \quad (18)$$

내부 복원 모멘트 보다 내부 모멘트가 클 경우 AUV는 문제없이 수중으

로 잠수 운동을 할 수 있다.

본 연구에서 계산한 것과 같이 AUV의 부력이 780N이고 GZ가 20mm, θ 가 10° 일 경우 내부 복원 모멘트는 식(19)과 같이 계산 가능하다.

$$RM = 780 \times 0.02 \times \sin 10^\circ = 2.709 Nm \quad (19)$$

그리고 외부 모멘트는 Fig.18에 나타난 것과 같이 방향타에 유체저항과 더불어 Lift force도 발생하게 되는데 이것을 이용하여 계산할 수 있다. 먼저 잠수 운동의 경우 방향타에서 발생하는 유체저항은 REMUS의 값의 유체저항계수 C_D 를 적용하여 식(20)와 같이 계산 할 수 있다.

$$A = 0.008615 m^2 \times \cos 60^\circ = 0.00431 m^2 \quad (20)$$
$$drag\ force = 0.5 \times 1025 \times 0.558 \times 0.00431 \times 2.55^2 = 8.015 N$$

Lift Force를 구하기 위해 유체저항계수(C_L)를 계산에 의해 3.15로 적용이 가능하며 식(21)과 같이 계산할 수 있다.

$$Lift\ force = 0.5 \times 1025 \times 3.15 \times 0.00431 \times 2.55^2 = 45.24 N \quad (21)$$

식(21)에서 구한 Lift force가 2개의 방향타에 작용하기 때문에 외부 모멘트를 식(22)와 같이 계산할 수 있다.

$$M = L \times F_{lift} = 0.769 \times 45.24 \times 2 = 69.56 Nm \quad (22)$$

식(19), 식(22)를 비교하면 외부 모멘트가 내부 모멘트보다 크기 때문에 잠수운동을 원활하게 할 수 있는 것을 확인 할 수 있다.

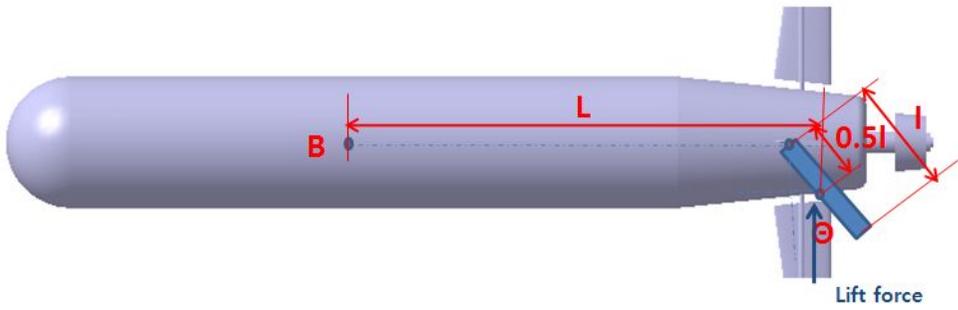


Fig. 18 AUV의 잠수 운동시의 방향타에 발생하는 외부 모멘트



제3장 AUV의 동역학 모델

3.1 AUV의 6자유도 동역학 모델

수중 운동체의 위치와 자세를 나타내기 위해서 먼저 기준 좌표계 (Reference frame)과 물체 고정 좌표계(Body fixed frame)를 정의하여야 한다. 기준 좌표계는 수중 운동체가 출발하는 지점에 대한 해수면 상의 한 점을 원점으로 하여 수중운동체가 출발하는 방향이 해수면에 투영된 방향을 X축, 해수면에 수직인 아래쪽 방향을 Z축, 그리고 오른손 법칙을 사용하여 이미 구해진 두 축으로부터 결정되는 방향을 Y축으로 정한다. 물체 고정 좌표계는 Fig.19와 같이 정의한다.

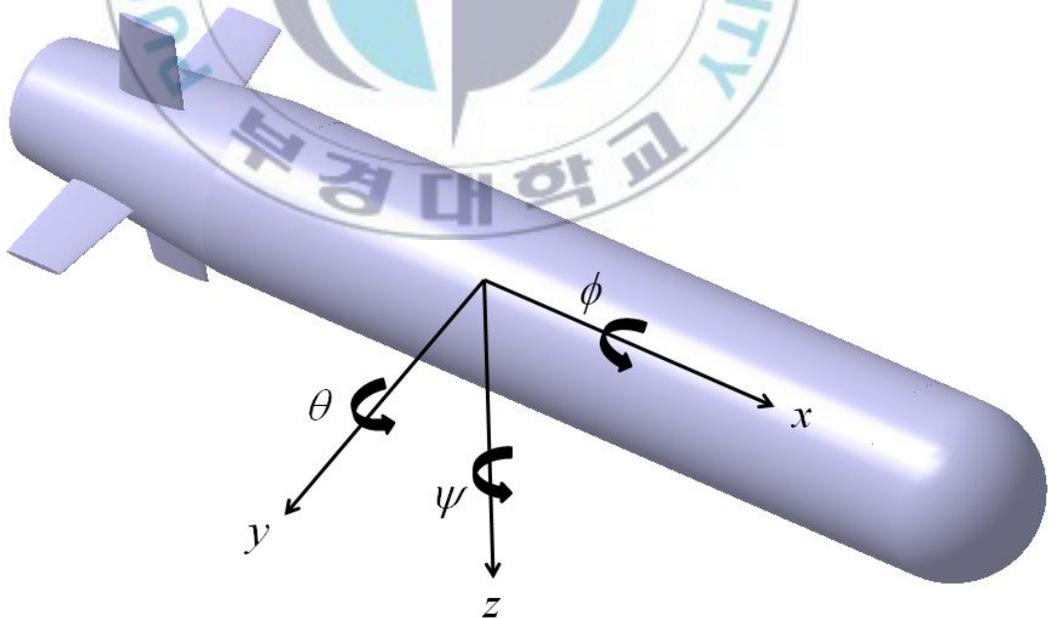


Fig. 19 Body-fixed coordinate system

위와 같이 기준 좌표계와 물체 고정 좌표계를 설정하게 되면 기준 좌표계에 대한 항체의 물체 고정 좌표계의 자세는 오일러 각(Euler angle)으로 표시한다. 오일러 각은 기준 좌표계를 각 축에 대하여 3번 회전시켜 동체 고정 좌표계에 일치시킬 때 얻어지는 각으로서, 회전은 ψ (yaw angle), θ (pitch angle), ϕ (roll angle)의 순서로 정한다. 오일러 각의 경우 항체의 자세가 수직에 가까워지면 자세를 표현할 수 없는 한계를 가지고 있으나 수중 항주체의 경우 자세가 수직에 가까운 운동이 거의 일어나지 않으므로 사용하여도 무방하다.

본 논문의 대상 물체인 무인잠수정은 수중을 운항하는 항체이다. 일반적으로 6개의 자유도를 가지는 수중 운동체는 Table.3 과 같은 12개의 상태 변수를 사용하여 식(23) 과 같은 비선형 미분방정식으로 표시할 수 있다. (Fossen, 2002)

Table.3 Twelve state variables of an AUV with 6 degrees of freedom

DOF		Linear and angular velocity	Position and Euler angles
1	Motion in the x-direction (surge)	u	x
2	Motion in the y-direction (sway)	v	y
3	Motion in the z-direction (heave)	w	z
4	Rotate about the x-axis (roll)	p	ϕ
5	Rotate about the y-axis (pitch)	q	θ
6	Rotate about the z-axis (yaw)	r	ψ

$$m[(\dot{u}-vr+wq)-x_G(q^2+r^2)+y_G(pq-\dot{r})+z_G(pr+\dot{q})] = X$$

$$m[(\dot{v}-wp+ur)-y_G(r^2+p^2)+z_G(qr-\dot{p})+x_G(qp+\dot{r})] = Y$$

$$m[(\dot{w}-uq+pv)-z_G(p^2+q^2)+x_G(rp-\dot{q})+y_G(rq+\dot{p})] = Z$$

$$I_x \dot{p} + (I_z - I_y)qr - (\dot{r} + pq)I_{xz} + (r^2 - p^2)I_{yz} + (pr - \dot{q})I_{xy} \\ + m[y_G(\dot{w} + pv - qu) - z_G(\dot{v} + ru - pw)] = K$$

$$I_y \dot{q} + (I_x - I_z)rp - (\dot{p} + qr)I_{xy} + (p^2 - z^2)I_{yz} + (qp - \dot{r})I_{xy} \\ + m[z_G(\dot{u} + wq - vr) - x_G(\dot{w} + pv - uq)] = M$$

$$I_z \dot{r} + (I_y - I_x)pq - (\dot{q} + rp)I_{yz} + (q^2 - p^2)I_{xy} + (rp - \dot{p})I_{zx} \\ + m[x_G(\dot{v} - wq + ur) - y_G(\dot{u} - vr + wq)] = N$$

(23)

식(23)을 벡터식 형태로 변경하면 식(24)와 같다.

$$M_{RB}v + C_{RB}(v)v = \tau_{RB}$$

$$v = [u, v, w, p, q, r]^T$$

$$\tau_{RB} = [X, Y, Z, K, M, N]^T$$

M_{RB} : 관성 매트릭스

C_{RB} : 코리올리 항에 의해 생기는 매트릭스

(24)

$$M_{RB} = \begin{bmatrix} mI_{3 \times 3} & -mS(r_G) \\ mS(r_G) & I_0 \end{bmatrix} = \begin{bmatrix} m & 0 & 0 & 0 & mz_G & -my_G \\ 0 & m & 0 & -mz_G & 0 & mx_G \\ 0 & 0 & m & my_G & -mx_G & 0 \\ 0 & -mz_G & my_G & I_x & -I_{xy} & -I_{xz} \\ mz_G & 0 & -mx_G & -I_{yx} & I_y & -I_{yz} \\ -my_G & mx_G & 0 & -I_{zx} & -I_{zy} & I_z \end{bmatrix}$$

$$C_{RB}(v) = \begin{bmatrix} 0_{3 \times 3} & -mS(v_1) - mS(S(v_2)r_G) \\ -mS(v_1) - mS(S(v_2)r_G) & mS(S(v_1)r_G) - S(I_0v_2) \end{bmatrix}$$

$$C_{RB}(v) = \begin{bmatrix} 0 & 0 & 0 & m(y_G q + z_G r) & -m(x_G q - w) & -m(x_G r + v) \\ 0 & 0 & 0 & -m(y_G p + w) & m(z_G r + x_G p) & -m(y_G r - u) \\ 0 & 0 & 0 & -m(z_G p - r) & -m(z_G q + u) & m(x_G p + y_G q) \\ -m(y_G q + z_G r) & m(y_G p + w) & m(z_G p - r) & 0 & -I_{yz}q - I_{xz} + I_z r & I_{yz}r + I_{xy}p - I_y q \\ m(x_G q - w) & -m(z_G r + x_G p) & m(z_G q + u) & I_{yz}q + I_{xz}p - I_z r & 0 & -I_{xz}r - I_{xy}q + I_x p \\ m(x_G r + v) & m(y_G r - u) & -m(x_G p + y_G q) & -I_{yz}r - I_{xy}p + I_y q & I_{xz}r + I_{xy}q + I_x p & 0 \end{bmatrix}$$

식(24)의 M_{RB} 와 C_{RB} 를 대입하여 식(23)을 간략화 시키면 다음과 같다.

$$\begin{cases} m[(\dot{u} - vr + wq) - x_G(\dot{q}^2 + r^2) + y_G(pq - \dot{r}) + z_G(pr + \dot{q})] = X \\ m[(\dot{v} - wp + ur) - y_G(r^2 + p^2) + z_G(qr - \dot{p}) + x_G(qp + \dot{r})] = Y \\ m[(\dot{w} - uq + pv) - z_G(p^2 + q^2) + x_G(rp - \dot{q}) + y_G(rq + \dot{p})] = Z \\ I_x \dot{p} + (I_z - I_y)qr + m[y_G(\dot{w} + pv - qu) - z_G(\dot{v} + ru - pw)] = K \\ I_y \dot{q} + (I_x - I_z)rp + m[z_G(\dot{u} + wq - vr) - x_G(\dot{w} + pv - uq)] = M \\ I_z \dot{r} + (I_y - I_x)pq + m[x_G(\dot{v} + ur - pw) - y_G(\dot{u} + qw - vr)] = N \end{cases} \quad (25)$$

식(25)의 우측의 항들은 항체에 작용하는 힘 벡터 F_0 및 모멘트 벡터 M_0 의 각 방향 성분을 나타낸다. 유체력과 모멘트 모델에는 Gertler(1967), Abcowitz(1969)등 잘 알려진 모델이 있다. 그 중에서 AUV와 같은 저속 운항체에 적용하기에 적합하다고 알려진 Cross Flow를 고려한 모델은 식(26)~(31)과 같고 이 식들에 표기된 비선형 계수들은 Table.4 와 Table.5 에 표기하였다.

$$\begin{aligned}
\sum X = & \frac{\rho}{2} l^4 [X_{qq} \dot{q}^2 + X_{rr} \dot{r}^2 + X_{rp} r \dot{p}] + \frac{\rho}{2} l^3 [X_u \dot{u} + X_{vr} vr + X_{wq} w q] \\
& + \frac{\rho}{2} l^2 [X_{uu} u^2 + X_{vv} v^2 + X_{ww} w^2] + \frac{\rho}{2} l^2 u^2 F_{xp}(\eta) - (W - B) \sin \theta \\
& + \frac{\rho}{2} l^2 [X_{\delta r \delta r} \delta r^2 + X_{\delta s \delta s} \delta s^2 + X_{\delta b \delta b} \delta b^2]
\end{aligned} \tag{26}$$

$$\begin{aligned}
\sum Y = & \frac{\rho}{2} l^4 [Y_r \dot{r} + Y_p \dot{p} Y_{|p|} |p| + Y_{pq} p q + Y_{qr} q r] \\
& + \frac{\rho}{2} l^3 [Y_v \dot{v} + Y_{wp} w p + Y_r u r + Y_p u p Y_{|v|} \frac{v}{|v|} \sqrt{v^2 + w^2} |r|] \\
& + \frac{\rho}{2} l^2 [Y_v u w + Y_{v|v|} v \sqrt{v^2 + w^2} + (W - B) \cos \theta \sin \phi] \\
& + \frac{\rho}{2} l^2 u^2 [Y_{\delta r} + Y_{\delta r \eta} (\eta - 1)] \delta r + Y_{cross} + Y_{vortex}
\end{aligned} \tag{27}$$

$$\begin{aligned}
\sum Z = & \frac{\rho}{2} l^4 Z_q \dot{q} + \frac{\rho}{2} l^3 [Z_w \dot{w} + Z_{vp} v p + Z_q u q + Z_{w|q|} \frac{w}{|w|} \sqrt{v^2 + w^2} |q| + Z_{rp} r p] \\
& + \frac{\rho}{2} l^3 [Z_w u w + Z_{w|w|} w \sqrt{v^2 + w^2} + (W - B) \cos \theta \cos \phi] \\
& + \frac{\rho}{2} l^2 u^2 [Z_{\delta s} + Z_{\delta s \eta} (\eta - 1)] \delta s + \frac{\rho}{2} l^2 u^2 Z_{\delta b} \delta b + Z_{cross} + Z_{vortex}
\end{aligned} \tag{28}$$

$$\begin{aligned}
\sum K = & \frac{\rho}{2} l^5 [M_p \dot{p} + M_r \dot{r} + K_q q r + K_{pq} p q] \\
& + \frac{\rho}{2} l^4 [K_p u p + K_r u r + K_v \dot{v} + K_{vq} v q + K_{wp} w p + K_{wr} w r] \\
& + \frac{\rho}{2} l^3 [K_v u w + K_{vw} v w] + (y_G W - y_B B) \cos \theta \cos \phi \\
& - (z_G W - z_B B) \cos \theta \sin \phi + K_{vortex}
\end{aligned} \tag{29}$$

$$\begin{aligned}
\sum M = & \frac{\rho}{2} l^5 [M_q \dot{q} + M_{rp} rp] \\
& + \frac{\rho}{2} l^4 [M_w \dot{w} + M_{vr} vr + M_{vp} vp + M_q uq + M_{|w|} w \sqrt{v^2 + w^2} q] \\
& + \frac{\rho}{2} l^3 [M^2 + M_w uw + M_{|w|} w \sqrt{v^2 + w^2}] \\
& - (x_G W - x_B B) \cos \theta \cos \phi - (z_G W - z_B B) \sin \theta \\
& + \frac{\rho}{2} l^3 u^2 [M_{\delta s} + M_{\delta s \eta} (\eta - 1)] \delta s + \frac{2}{\rho} l^3 u^2 M_{\delta b} \delta b + M_{cross} + M_{vortex}
\end{aligned} \tag{30}$$

$$\begin{aligned}
\sum N = & \frac{\rho}{2} l^5 [N_r \dot{r} + N_p p + N_{pq} pq + N_{qr} qr] \\
& + \frac{\rho}{2} l^4 [N_v \dot{v} + N_{wp} wp + N_{vq} vq + N_p up + N_r ur + N_{|v|} v \sqrt{v^2 + w^2} r] \\
& + \frac{\rho}{2} l^3 [N_v uv + N_{|v|} v \sqrt{v^2 + w^2}] \\
& - (x_G W - x_B B) \cos \theta \sin \phi - (y_G W - y_B B) \sin \theta \\
& + \frac{\rho}{2} l^3 u^2 [N_{\delta r} + N_{\delta r \eta} (\eta - 1)] \delta r + N_{cross} + N_{vortex}
\end{aligned} \tag{31}$$

Table.4 Non-Linear Maneuvering Coefficients : Forces

Parameter	Units	Description
X_{uu}	kg/m	Cross-flow Drag
$X\dot{u}$	kg	Added Mass
X_{wq}	kg/rad	Added Mass Cross-term
X_{qq}	kgm/rad	Added Mass Cross-term
X_{vr}	kg/rad	Added Mass Cross-term
X_{rr}	kgm/rad	Added Mass Cross-term
X_{prop}	N	Propeller Thrust
Y_{vv}	kg/m	Cross-flow Drag
Y_{rr}	kgm/rad ²	Cross-flow Drag
Y_{uv}	kg/m	Body Lift Force and Fin Lift
$Y\dot{v}$	kg	Added Mass
$Y\dot{\gamma}$	kgm/rad	Added Mass
Y_{ur}	kg/rad	Added Mass Cross Term and Fin Lift
Y_{wp}	kg/rad	Added Mass Cross-term
Y_{pq}	kg. m/rad	Added Mass Cross-term
Y_{uudr}	kg/(mrad)	Fin Lift Force
Z_{ww}	kg/m	Cross- flow Drag
Z_{qq}	kgm/rad ²	Cross-flow Drag
Z_{uv}	kg/m	Body Lift Force and Fin Lift
$Z\dot{w}$	kg	Added Mass
$Z\dot{q}$	kgm/rad	Added Mass
Z_{uq}	kg/rad	Added Mass Cross-term and Fin Lift
Z_{vp}	kg/rad	Added Mass Cross-term
Z_{rp}	kg/rad	Added Mass Cross-term
Z_{uuds}	kg/(mrad)	Fin Lift Force

Table.5 Non-Linear Maneuvering Coefficients : Moments

Parameter	Units	Description
K_{pp}	$\text{kgm}^2/\text{rad}^2$	Rolling Resistance
$K_{\dot{p}}$	kgm^2/rad	Added Mass
K_{prop}	Nm	Propeller Torque
M_{ww}	kg	Cross-flow Drag
M_{qq}	$\text{kgm}^2/\text{rad}^2$	Cross-flow Drag
M_{uw}	kg	Body and Fin Lift and Munk Moment
$M_{\dot{w}}$	kgm	Added Mass
$M_{\dot{q}}$	kgm^2/rad	Added Mass
M_{uq}	kgm/rad	Added Mass Cross Term and Fin Lift
M_{vp}	kgm/rad	Added Mass Cross Term
M_{rp}	$\text{kgm}^2/\text{rad}^2$	Added Mass Cross-term
M_{uuds}	kg/rad	Fin Lift Moment
N_{vv}	kg	Cross-flow Drag
N_{rr}	$\text{kgm}^2/\text{rad}^2$	Cross-flow Drag
N_{uv}	kg	Body and Fin Lift and Munk Moment
$N_{\dot{v}}$	kgm	Added Mass
$N_{\dot{r}}$	kgm^2/rad	Added Mass
N_{ur}	kgm/rad	Added Mass Cross Term and Fin Lift
N_{wp}	kgm/rad	Added Mass Cross Term
N_{pq}	$\text{kgm}^2/\text{rad}^2$	Added Mass Cross-term
N_{uudr}	kg/rad	Fin Lift Moment
Z_{uq}	kg/rad	Added Mass Cross-term and Fin Lift
Z_{vp}	kg/rad	Added Mass Cross-term
Z_{rp}	kg/rad	Added Mass Cross-term
Z_{uuds}	$\text{kg}/(\text{mrad})$	Fin Lift Force

3.2 AUV의 6자유도 운동방정식

본 연구에서 사용할 AUV의 6자유도 운동방정식을 간략화를 위하여 설계한 AUV의 부력중심에 원점을 두었으며 상하좌우 대칭이라 가정하였다.

$$m[\dot{u} - vr + wq + z_G(pr + \dot{q})] = \sum X \quad (32)$$

$$m[\dot{v} + ur - wp + z_G(qr - \dot{p})] = \sum Y \quad (33)$$

$$m[\dot{w} - uq + vp - z_G(p^2 + q^2)] = \sum Z \quad (34)$$

$$I_{xx}\dot{p} + (I_{xx} - I_{yy})qr - (\dot{r} + pq)I_{zx} + (r^2 + q^2)I_{yz} + (pr - \dot{q})I_{xy} - mz_G(\dot{v} + ur - wp) = \sum K \quad (35)$$

$$I_{yy}\dot{q} + (I_{xx} - I_{zz})pr - (\dot{p} + qr)I_{xy} + (p^2 - r^2)I_{zx} + (qp - \dot{r})I_{yz} + mz_G(\dot{u} - vr + wq) = \sum M \quad (36)$$

$$I_{zz}\dot{r} + (I_{yy} - I_{xx})pq - (\dot{q} + rp)I_{yz} + (q^2 - p^2)I_{xy} + (rq - \dot{p})I_{zx} = \sum N \quad (37)$$

식(32)~식(37)에서 우측의 항들은 항체에 작용하는 힘과 모멘트를 나타낸다. 본 논문에서는 선형 관성력, 선형 감쇠, 비선형 감쇠와 관련된 유체력 계수를 고려하여 식(38~43)과 같이 유체력 모델을 구성하였다.

$$\sum X = X_u\dot{u} + X_{rr}r^2 + X_{uu}u^2 + X_{vv}v^2 + X_{ww}w^2 + X_{th} \quad (38)$$

$$\sum Y = Y_v\dot{v} + Y_r\dot{r} + Y_p p + Y_r r + Y_v v + Y_{|v|} |v| \quad (39)$$

$$\sum Z = Z_w\dot{w} + Z_q\dot{q} + Z_{uq}uq + Z_q q + Z_w w + Z_{|w|} |w| + Z_{\delta s} \delta s \quad (40)$$

$$\sum K = K_p\dot{p} + K_r\dot{r} + K_v\dot{v} + K_{vq}vq + K_{qr}qr + K_{wr}wr + K_{vw}vw + K_p p + K_{th} \quad (41)$$

$$\sum M = M_w \dot{w} + M_q \dot{q} + M_u \dot{u} + M_r \dot{r} + M_{uw} uw + M_w w + M_{|w|} |w| |w| + M_{qs} \delta s \quad (42)$$

$$\sum N = N_v \dot{v} + N_r \dot{r} + N_{wp} wp + N_{pq} pq + N_{uv} uv + N_v v + N_{|v|} |v| |v| + N_{ur} ur + N_r r + N_{th} \quad (43)$$

또한, 무인잠수정의 자세와 위치는 오일러 각을 이용하여 식(44)~식(49)으로 표현된다.

$$\dot{\phi} = p + q \sin \phi \tan \theta + r \cos \phi \tan \theta \quad (44)$$

$$\dot{\theta} = q \cos \phi - r \sin \phi \quad (45)$$

$$\dot{\psi} = (q \sin \phi + r \cos \phi) / \cos \theta \quad (46)$$

$$\dot{X} = u \cos \psi \cos \theta + v (\cos \psi \sin \theta \sin \phi - \sin \psi \cos \phi) + w (\cos \psi \sin \theta \sin \phi + \sin \psi \sin \phi) \quad (47)$$

$$\dot{Y} = u \sin \psi \cos \theta + v (\sin \psi \sin \theta \sin \phi + \cos \psi \cos \phi) + w (\sin \psi \sin \theta \cos \phi - \cos \psi \sin \phi) \quad (48)$$

$$\dot{Z} = -u \sin \theta + v \cos \theta \sin \phi + w \cos \theta \sin \phi \quad (49)$$

3.3 AUV의 X-Y평면 운동방정식

XY평면에서만 움직이는 AUV의 운동은 Fig.20과 같이 X축 방향으로의 Translational Motion(Surge), Y축 방향으로의 Translational Motion(sway)와 Z축 방향으로의 Rotational Motion(yaw)운동만을 고려하게 된다.

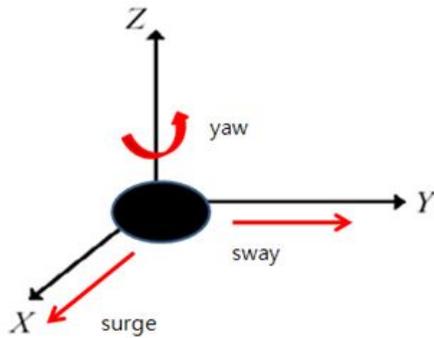


Fig. 20 X-Y Planar motion of the AUV

그러므로 6자유도 운동과 달리 평면운동에서의 상태변수는 아래의 Table.6과 같이 6개로 줄어들게 된다.

Table.6 Six state variables of an AUV with X-Y plane

DOF		Linear and angular velocity	Position and Euler angles
1	Motion in the x-direction (surge)	u	x
2	Motion in the y-direction (sway)	v	y
3	Motion in the z-direction (heave)	w	z
4	Rotate about the x-axis (roll)	p	Φ
5	Rotate about the y-axis (pitch)	q	Θ
6	Rotate about the z-axis (yaw)	r	Ψ

Surge, sway, yaw운동만 있으므로 나머지 요소 Heave, roll, pitch요소는 무시할 수 있다. 또 XZ-Plane이 좌우대칭이고 좌표계가 AUV의 무게중심에 있다고 가정하면 식(50)과 같이 세 가지 항에 0을 적용할 수 있다.

$$\begin{aligned} w = p = q = 0 \\ I_{xy} = I_{yz} = 0 \\ y_G = 0 \end{aligned} \quad (50)$$

위의 세 가지 항이 0이므로 앞 절에서 구한 6자유도 운동방정식을 더욱 간략화 할 수 있다.

앞에서 구한 관성 매트릭스 M_{RB} 와 C_{RB} 를 간략화하면 다음과 같다.

$$M_{RB} = \begin{bmatrix} m & 0 & 0 \\ 0 & m & mx_G \\ 0 & mx_G & I_z \end{bmatrix} \quad (51)$$

$$C_{RB}(v) = \begin{bmatrix} 0 & 0 & -m(x_G r + v) \\ 0 & 0 & mu \\ m(x_G r + v) & -mu & 0 \end{bmatrix} \quad (52)$$

여기서 Added mass 항과 added inertia항, added Coriolis centripetal항을 구하면 다음과 같다.

$$M_A = \begin{bmatrix} -X_u & 0 & 0 \\ 0 & -Y_v & -Y_r \\ 0 & -Y_r & -N_r \end{bmatrix} \quad (53)$$

$$C_A(v) = \begin{bmatrix} 0 & 0 & Y_v v + Y_r r \\ 0 & 0 & -X_u u \\ -Y_v v - Y_r r & X_u u & 0 \end{bmatrix} \quad (54)$$

위의 두 식을 종합하면 다음과 같다.

$$M = M_{RB} + M_A \quad (55)$$

$$C(v) = C_{RB}(v) + C_A(v) \quad (56)$$

$$M = \begin{bmatrix} m - X_u & 0 & 0 \\ 0 & m - Y_v & mx_G - Y_r \\ 0 & mx_G - Y_r & I_z - N_r \end{bmatrix} \quad (57)$$

$$C(v) = \begin{bmatrix} 0 & 0 & -(m - Y_v)v - (mx_G - Y_r)r \\ 0 & 0 & (m - X_u)u \\ (m - Y_v) + (mx_G - Y_r)r - (m - X_u)u & 0 & 0 \end{bmatrix} \quad (58)$$

위의 식들을 벡터 형태로 나타내면 식(59)과 같이 표현할 수 있다.

$$M\dot{v} + C(v)v + Dv = Bu \quad (59)$$

B : control matrix describing the thruster configuration

u : control vector

3.4 AUV의 X-Z 평면 운동방정식

XZ평면에서만 움직이는 AUV의 운동은 Fig.21과 같이 X축 방향으로의 Translational Motion(Surge), Z축 방향으로의 Translational Motion(heave)와 Y축 방향으로의 Rotational Motion(pitch)운동만을 고려하게 된다.

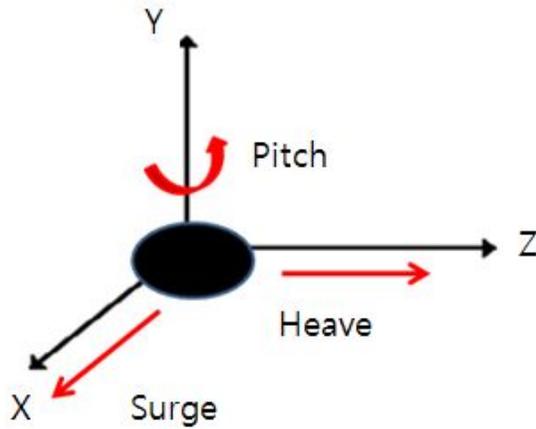


Fig. 21 X-Z Planar motion of the AUV

그러므로 6자유도 운동과 달리 평면운동에서의 상태변수는 아래의 Table.7과 같이 6개로 줄어들게 된다.

Table.7 Six state variables of an AUV with X-Z plane

DOF		Linear and angular velocity	Position and Euler angles
1	Motion in the x-direction (surge)	u	x
2	Motion in the y-direction (sway)	v	y
3	Motion in the z-direction (heave)	w	z
4	Rotate about the x-axis (roll)	p	ϕ
5	Rotate about the y-axis (pitch)	q	θ
6	Rotate about the z-axis (yaw)	r	ψ

Surge, heave, pitch 운동만 있으므로 나머지 요소 sway, roll, yaw요소는 무시할 수 있다. 또 XZ-Plane이 좌우대칭이고 좌표계가 AUV의 무게 중심에 있다고 가정하면 식(60)과 같이 세 가지 항에 0을 적용할 수 있다.

$$\begin{aligned} v = p = r &= 0 \\ I_{xz} = I_{zy} &= 0 \\ z_G &= 0 \end{aligned} \quad (60)$$

위의 세 가지 항이 0이므로 앞 절에서 구한 6자유도 운동방정식을 더욱 간략화 할 수 있다.

앞에서 구한 관성 매트릭스 M_{RB} 와 C_{RB} 를 간략화하면 다음과 같다.

$$M_{RB} = \begin{bmatrix} m & 0 & 0 \\ 0 & m & -mx_G \\ 0 & -mx_G & I_y \end{bmatrix} \quad (61)$$

$$C_{RB}(v) = \begin{bmatrix} 0 & 0 & -m(x_G q - w) \\ 0 & 0 & -m u \\ m(x_G q - w) & m u & 0 \end{bmatrix} \quad (62)$$

여기서 Added mass 항과 added inertia항, added Coriolis centripetal항을 구하면 다음과 같다.

$$M_A = \begin{bmatrix} X_u & 0 & 0 \\ 0 & -Z_w & -Z_q \\ 0 & -M_w & -M_q \end{bmatrix} \quad (63)$$

$$C_A(v) = \begin{bmatrix} 0 & 0 & -Z_w w - Z_q q \\ 0 & 0 & X_u u \\ Z_w w + Z_q q - X_u u & 0 & 0 \end{bmatrix} \quad (64)$$

위의 두 식을 종합하면 다음과 같다.

$$M = M_{RB} + M_A \quad (65)$$

$$C(v) = C_{RB}(v) + C_A(v) \quad (66)$$

$$M = \begin{bmatrix} m - X_u & 0 & 0 \\ 0 & m - Z_w & -mx_G - Z_q \\ 0 & -mx_G - M_w & I_y - M_q \end{bmatrix} \quad (77)$$

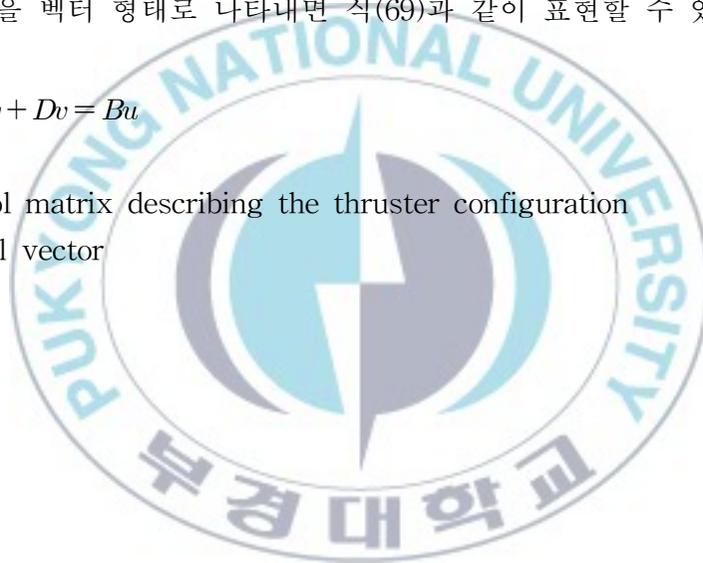
$$C(v) = \begin{bmatrix} 0 & 0 & -m(x_G q - w) - Z_w w - Z_q q \\ 0 & 0 & -\mu + X_u u \\ m(x_G q - w) + Z_w w + Z_q q & m u - X_u u & 0 \end{bmatrix} \quad (68)$$

위의 식들을 벡터 형태로 나타내면 식(69)과 같이 표현할 수 있다.

$$\dot{M}v + C(v)v + Dv = Bu \quad (69)$$

B : control matrix describing the thruster configuration

u : control vector



제4장 AUV의 운동 성능 해석

4.1 AUV 운동시뮬레이션 프로그램

3장에서 구한 AVU의 운동 방정식을 기반으로 AUV의 운동을 Matlab을 활용하여 시뮬레이션 할 수 있다. 하지만 AUV의 운동방정식의 Force 와 Moment는 Table.4와 Table.5 과 같이 비선형 데이터가 다수 존재하며 비선형 데이터는 실제 AUV를 제작한 후에 실험을 통해서 그 값을 측정해야 알 수 있다. 본 연구에서 설계한 AUV를 실제 제작 후 실험을 통해 비선형 데이터를 얻는다는 것은 많은 시간과 비용이 발생하며 AUV의 개념설계 단계에서 정확한 값의 데이터를 적용하는 것은 어렵다. 그러나 본 연구에서는 AUV의 개념설계를 통해 설계된 AUV의 운동을 실제 제작 전에 시뮬레이션을 통해 성능 검증을 하는 것에 목적이 있다. 따라서 기간 및 비용단축을 위해 AUV의 형상과 유사한 미군에 기 배치된 REMUS의 비선형 데이터 값들을 AUV 운동 방정식에 대입하여 시뮬레이션을 수행하였다.

AUV의 운동시뮬레이션 프로그램은 Euler Parameter를 이용하여 별도의 제어기가 없는 상태에서 AUV의 운동성능을 확인할 수 있는 X-Y 평면운동 시뮬레이션 및 X-Z 수직운동 시뮬레이션을 수행하여 AUV의 위치 및 각 위치의 각도 및 각속도를 확인할 수 있고 또한 Way-point 추적 시뮬레이션을 수행하여 간단한 PD 제어를 통해 AUV가 방향타의 각도를 제어하면서 목표 지점으로 운동을 어떻게 하는지 확인할 수 있다.

AUV의 운동 시뮬레이션은 AUV의 6자유도 운동방정식을 기반으로 하고 있으며 X-Y 평면운동 시뮬레이션 및 X-Z 수직운동 시뮬레이션의 경우 Table.4 및 Table.5의 Force, Moment의 비선형 데이터를 미리 알고 있고 그 값과 AUV 운동 설정 초기 값 및 방향타의 각도를 6자유도 운동 방정식에 대입하여 AUV의 전체 운동을 계산하는 방식으로 프로그램이 되어있다. AUV의 Way-point 추적 시뮬레이션의 경우는 앞서 언급했던 3가지 프로그램과 유사하나 목표 지점의 좌표값과 AUV의 운동 방향을 계산하여 오차를 보상하는 PD 제어기법을 적용하여 목표하는 지점을 통과할 수 있도록 프로그램이 되어있다. 이러한 평면, 수직 시뮬레이션의 Block Diagram 및

Way-point 추적 시뮬레이션의 Flow chart는 Fig.22, Fig.23와 같다.

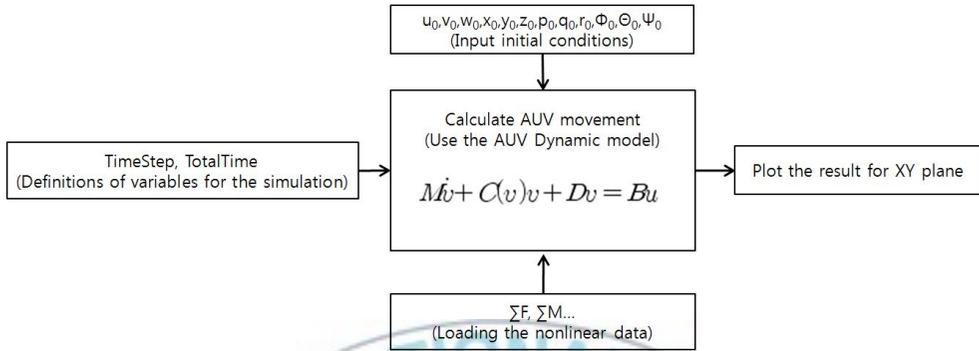


Fig. 22 AUV motion simulation block diagram

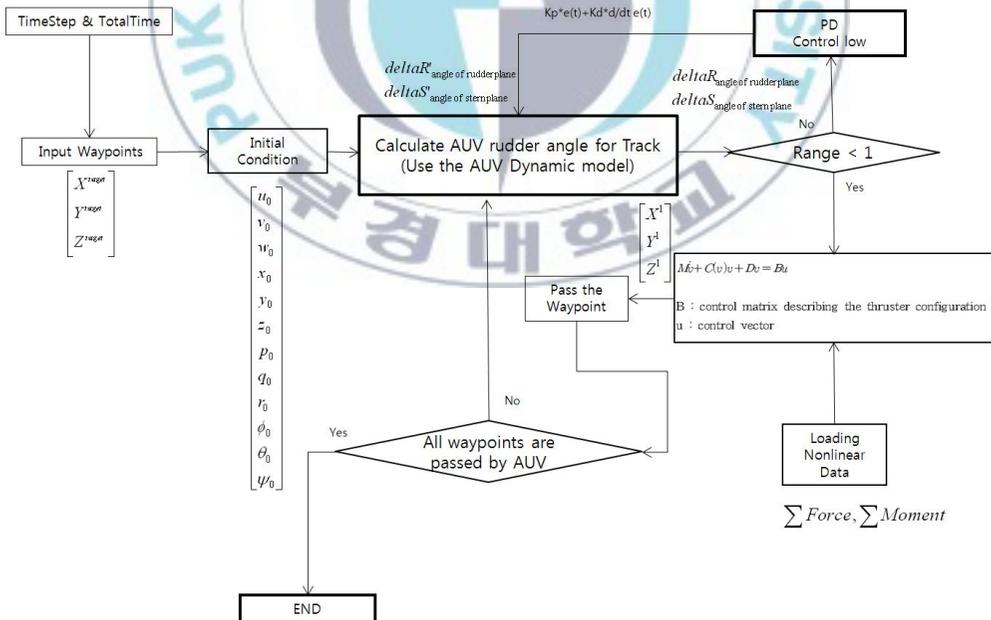


Fig. 23 Way-point simulation flow chart

4.2 AUV의 평면 운동 시뮬레이션

3장에서 구한 AUV의 운동 방정식을 기반으로 AUV의 X-Y 평면운동 시뮬레이션 수행하였다. X-Y 평면운동 시뮬레이션을 수행함으로써 별도의 제어기 없이 AUV의 초기 운동 상태를 정의하고(100s동안 2.55m/s로 추진 운동)일정 시간을 두고 AUV의 운동을 확인하였다. 이러한 작업을 통해 설계된 AUV 모델링의 운동 성능을 확인할 수 있고 모델링의 타당성을 확인할 수 있기 때문에 가장 기본적인 과정이라고 할 수 있다.

Table.1에 나와 있는 AUV의 설계 데이터 외에 4.1절에서 언급한 것처럼 설계된 AUV의 모델링 상에 존재하는 여러 비선형 데이터는 기 입수된 REMUS의 비선형 데이터를 시뮬레이션에 적용하였다.

AUV의 수직 방향타를 10도, 20도, 30도로 하여 시뮬레이션을 수행한 결과는 Fig.24 와 같다. Fig.24 시뮬레이션 결과에서 나타나듯이 일정한 속도로 운동 중에는 방향타의 각도가 커질수록 회전반경도 작아지는 것을 확인할 수 있다.

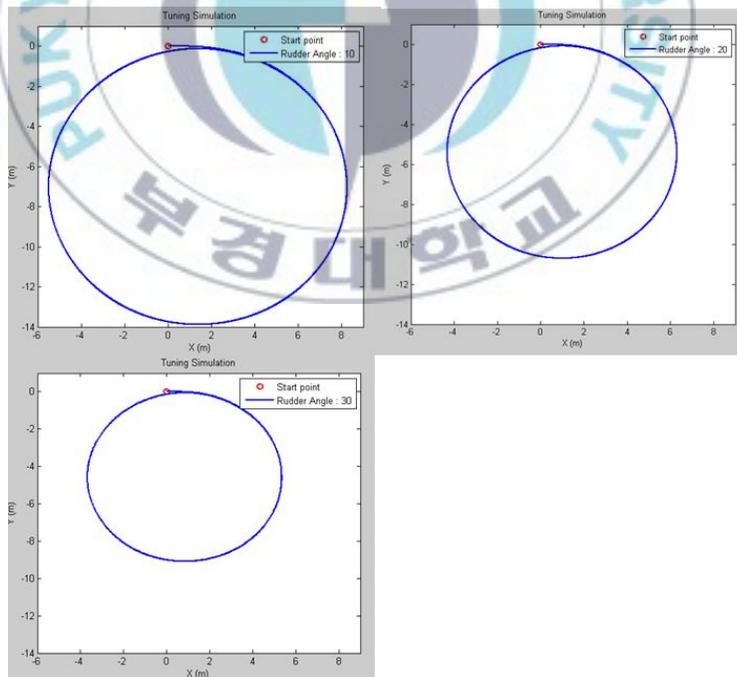


Fig. 24 X-Y tuning simulation results of the X-Y position

Fig. 24과 같이 회전 운동을 할 경우 X-Z상의 AUV의 위치를 Fig.25와 같이 나타내었다. Fig.25를 보면 확인할 수 있는 것 처럼 회전 운동의 경우 Z축의 위치가 동일한 것을 확인 할 수 있다. 이는 AUV가 회전운동을 하면서 Z방향으로 움직이는 않은 것을 확인 할 수 있다. 또한 방향타의 각도가 커질수록 X축의 변위가 점점 작아지는 것을 확인 할 수 있는데 이는 Fig.24에서 확인한 바와 같이 방향타 각도가 커지면 회전반경이 줄어드는 현상 때문이라고 할 수 있다. Fig.26, Fig.27에서 나타난 것과 같이 X-Y 평면 운동 중에 발생하는 Z축을 중심으로 회전하는 각속도 $r(\text{deg/s})$ 가 한바퀴 회전이 끝나면 일정하게 수렴하는 것과 각 $\psi(\text{deg})$ 가 일정하게 감소하는 것을 확인할 수 있다.

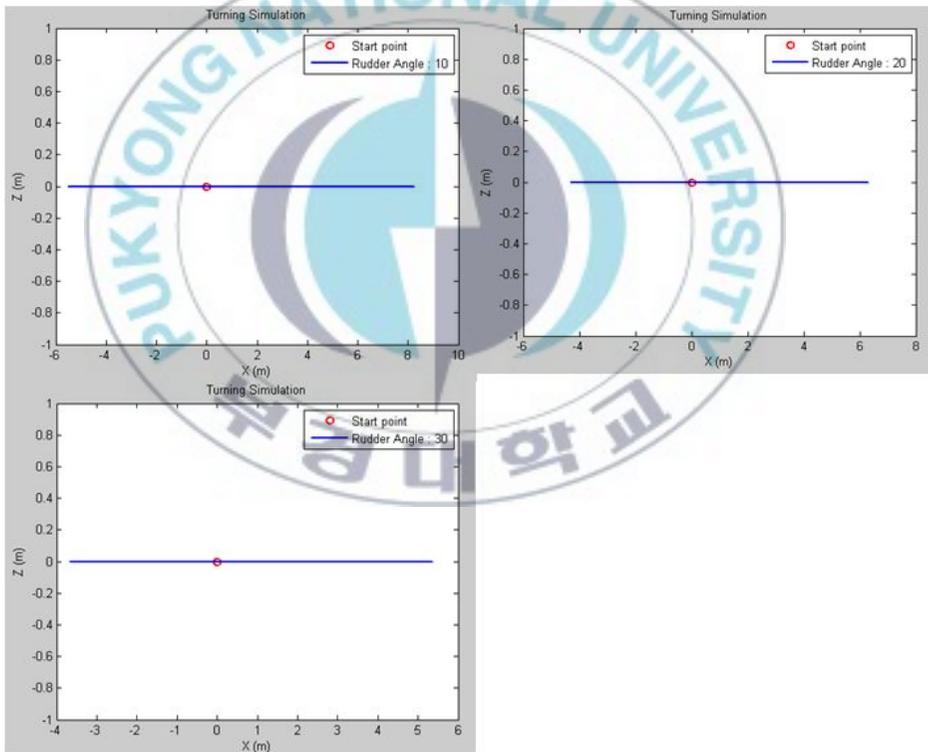


Fig. 25 X-Y turing simulation results of the X-Z position

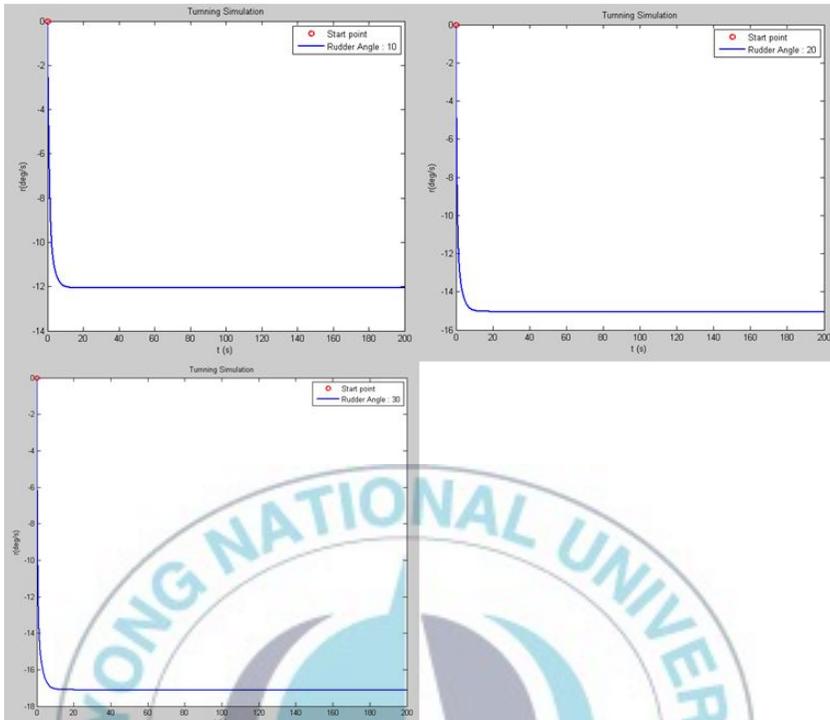


Fig. 26 X-Y turing simulation results of the angular velocity r (deg/s)

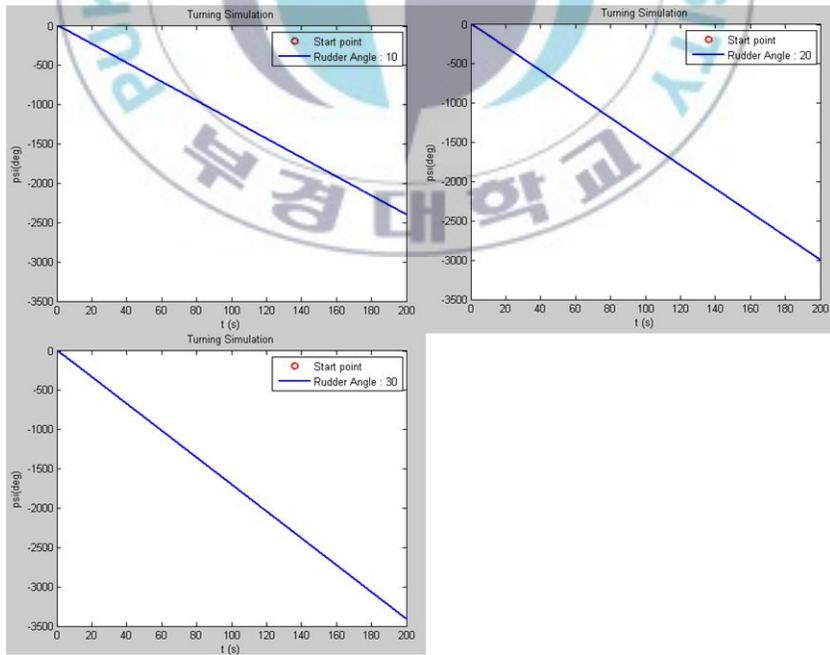


Fig. 27 X-Y turing simulation results of the angle ψ (deg)

4.3 AUV의 수직 운동 시뮬레이션

AUV의 운동방정식을 기반으로 AUV의 X-Z 수직 운동 시뮬레이션을 수행하였다. X-Z 수직 운동 시뮬레이션을 수행함으로써 일정 속도(2.55m/s)로 100s 동안 운동중에 AUV가 방향타의 각도에 의해 AUV의 운동이 어떻게 되는지 확인 할 수 있다. X-Z 수직 운동 시뮬레이션도 동일하게 Table.1의 데이터를 제외한 비선형 데이터들을 기 입수된 REMUS의 설계 데이터를 모델링에 적용하였다.

AUV의 수평 방향타를 10도, 20도, 30도로 하여 시뮬레이션을 수행한 결과는 Fig.28 와 같다. Fig.28 시뮬레이션 결과에서 나타난 바와 같이 방향타 각도가 10도, 20도, 30도로 커질수록 회전반경이 줄어드는 것을 확인할 수 있다.

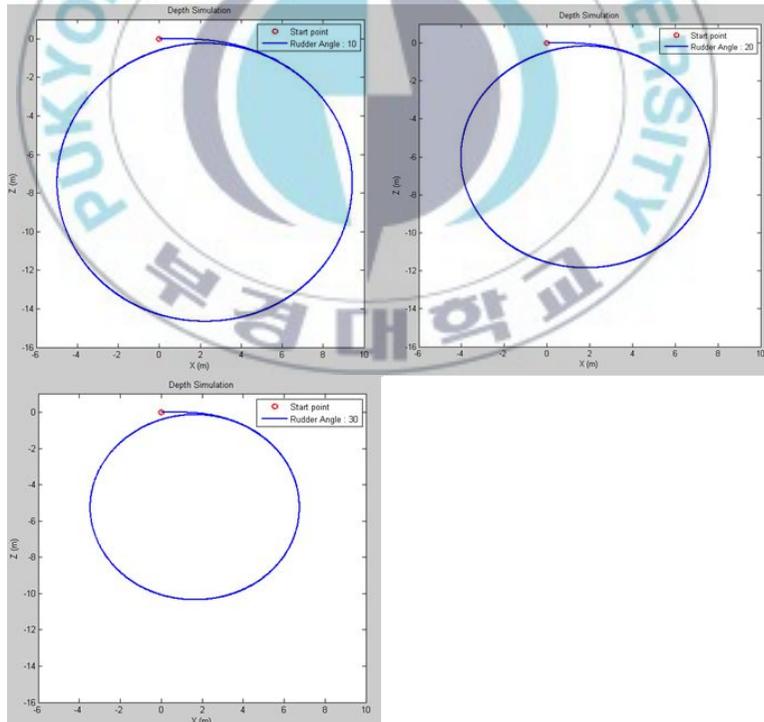


Fig.28 X-Z depth simulation results of X-Z position

Fig.28과 같이 X-Z로 수직 운동을 할 경우 X-Y상의 AUV의 위치를 Fig.29와 같이 나타내었다. Fig.29를 보면 확인할 수 있는 것처럼 수직 운동의 경우 Y축의 위치가 동일한 것을 확인할 수 있다. 이는 AUV가 수직 운동을 하면서 Y방향으로 움직이지 않는 것을 확인할 수 있다. 또한 방향타의 각도가 커질수록 X축의 변위가 점점 더 줄어드는 것을 확인할 수 있는데 이는 Fig.28에서 확인한 바와 같이 방향타 각도가 커지면 회전반경이 줄어드는 현상 때문이라고 할 수 있다. 그리고 Fig.30, Fig.31에서 나타난 것과 같이 X-Z 평면 운동 중에 발생하는 Y축을 중심으로 회전하는 각속도 $q(\text{deg/s})$ 가 한바퀴 회전이 끝나면 일정하게 수렴하는 것과 각 $\theta(\text{deg})$ 가 일정하게 커지는 것을 확인할 수 있다.

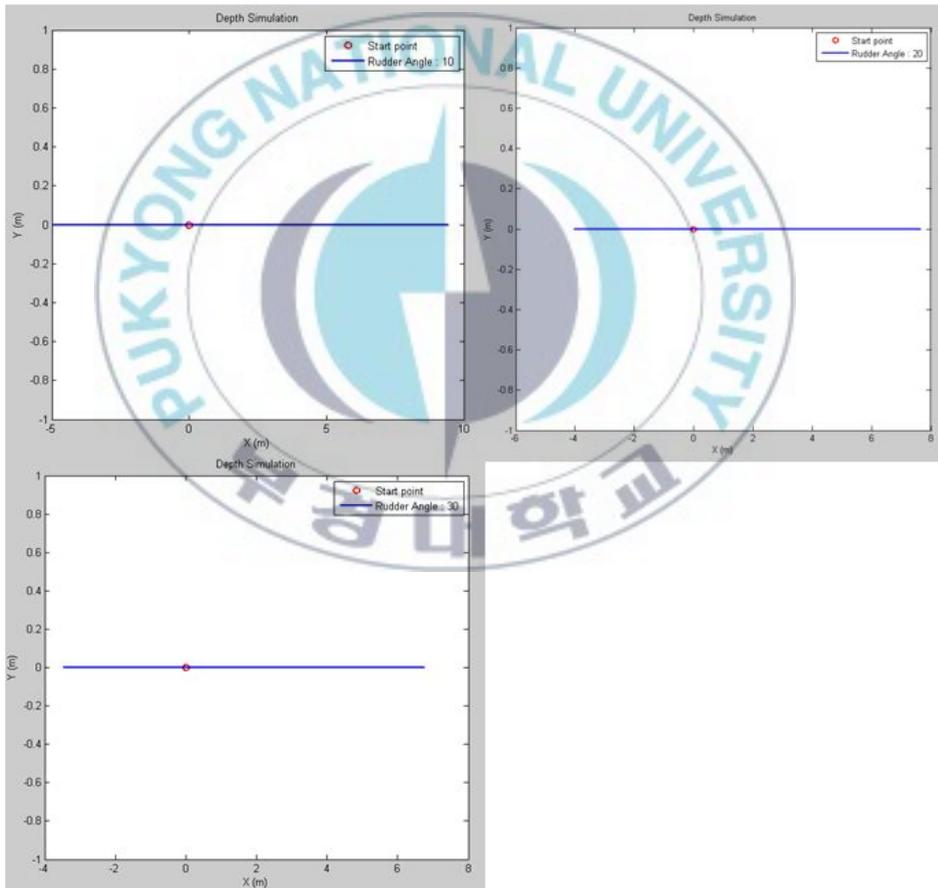


Fig.29 X-Z depth simulation results of the X-Y position

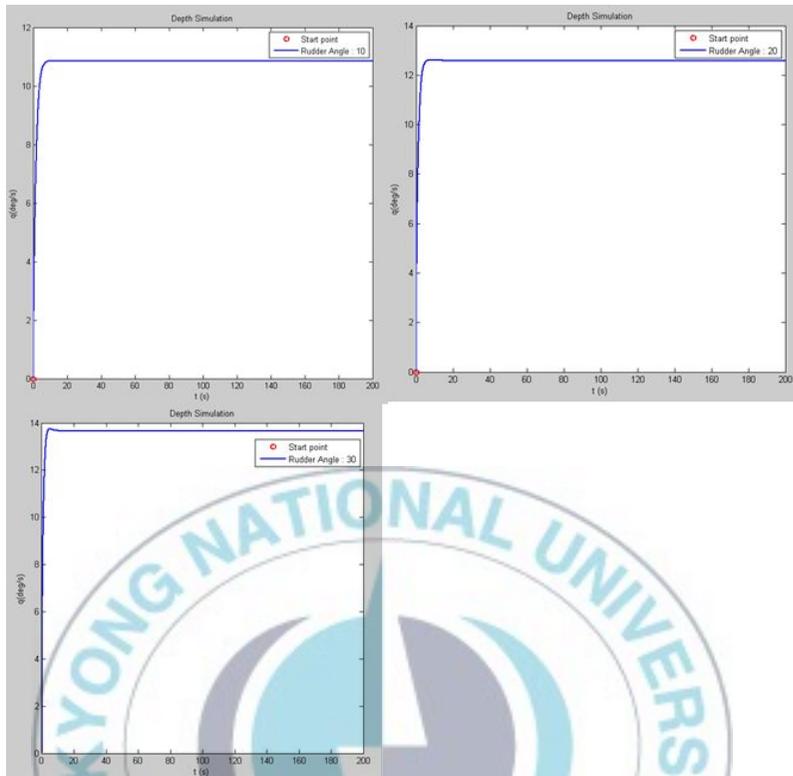


Fig.30 X-Z depth simulation results of the angular velocity q (deg/s)

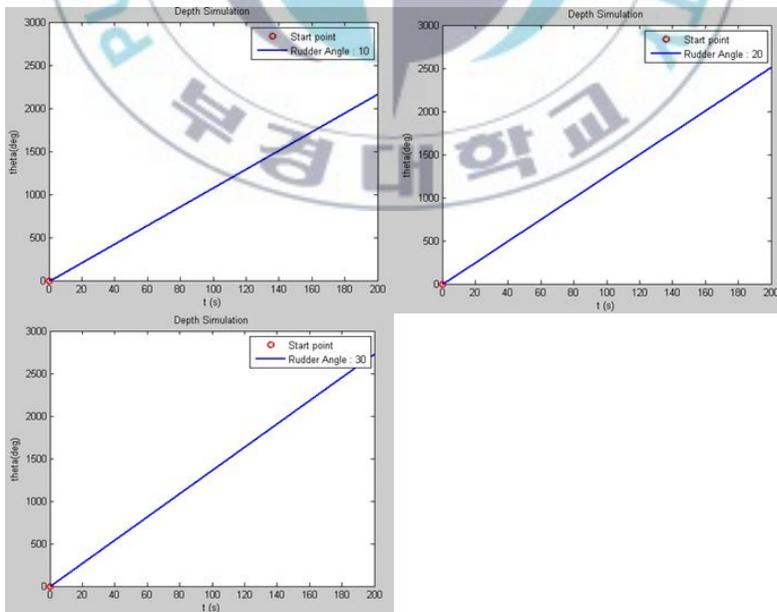


Fig.31 X-Z depth simulation results of the angle θ (deg)

4.4 AUV의 Way-point 추적 시뮬레이션

앞서 구현한 AUV의 수평 시뮬레이션 및 수직 시뮬레이션은 설계된 AUV의 운동을 AUV의 초기상태를 가정하여 AUV의 운동을 확인하는데 목적이 있다. 따라서 별도의 제어기능이 필요하지 않기 때문에 Fig. 22에서 확인 가능한 바와 같이 시뮬레이션에 제어가 포함되어 있지 않다. 하지만 AUV는 User가 원하는 곳으로 운동이 가능해야 성공적으로 그 임무를 수행할 수 있다. 그에 따라 AUV에 통신 시스템을 장착하여 RF안테나와 GPS를 통해 현재 위치를 파악하고 방향타 각도 조절을 통해 목표위치까지 운동할 수 있어야 한다. 이와 같은 목표 위치로 AUV가 운동을 할 수 있도록 하기 위해 간단한 Way-point 운동 시뮬레이션을 Fig.23과 같은 Flow chart를 통해 수행하였다. AUV의 초기 위치를 기준으로 AUV 방향타의 각도를 User가 원하는 Target Position을 통과하여 운동할 수 있도록 PD 제어를 통해 보상하는 시뮬레이션을 수행하였다.

현재까지 학문이 발달함에 따라 많은 제어이론이 존재하고 많은 제품에 다양한 제어가 활용되고 있다. 본 연구에서 사용한 PD 제어기의 경우 제어기 구성이 매우 간단하고 제어 이득 조절이 쉬우며 빠른 응답성과 안정성으로 로봇제어에 많이 활용되고 있다. 그에 따라 Fig.23의 Flow chart와 같이 방향타의 각도를 AUV의 위치를 이용해 각도값을 계산하여 범위를 설정하여 오차범위 밖이면 방향타 각도를 보상하여 오차범위 내에 들어오도록 제어기를 구성하였다. PD제어기의 비례계수(Kp) 및 미분계수(Kd)는 제어모델이 주어지기 때문에 주파수 영역 설계법, 근궤적법, 과도응답법을 사용하거나 반복 작업으로 좋은 응답을 보이는 계수 값을 추적하는 방법도 있다.

본 연구에서는 (0,0), (20,40), (40,40), (40,20), (20,0)와 같이 수평면에 5개의 Way-point를 설정하여 시뮬레이션을 수행하였다. Fig.32의 경우 비례계수(Kp) 및 미분계수(Kd)를 각각 -0.21, -0.21로 설정하였고 Fig.33의 경우 -0.50, -0.50으로 설정 마지막으로 Fig.33의 경우 -0.90, -0.90으로 설정하여 시뮬레이션을 수행하였다. 본 연구에서 비례계수(Kp) 및 미분계수(Kd)는 반복 작업을 통해 좋은 응답을 보이는 계수 값을 추적하는 방법을 사용하였

으며 그 결과 Fig.34과 같이 좋은 응답을 결과로 나타내었다.

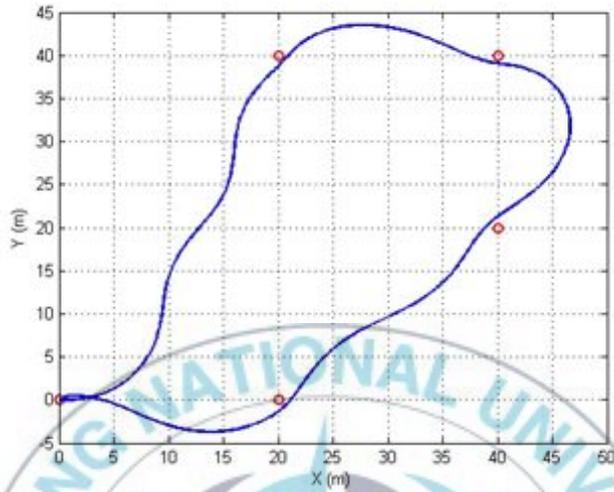


Fig. 32 Way-point Simulation Result 1

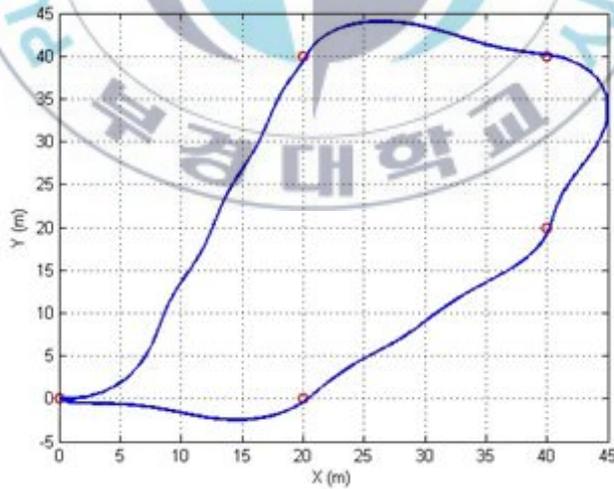


Fig. 33 Way-point Simulation Result 2

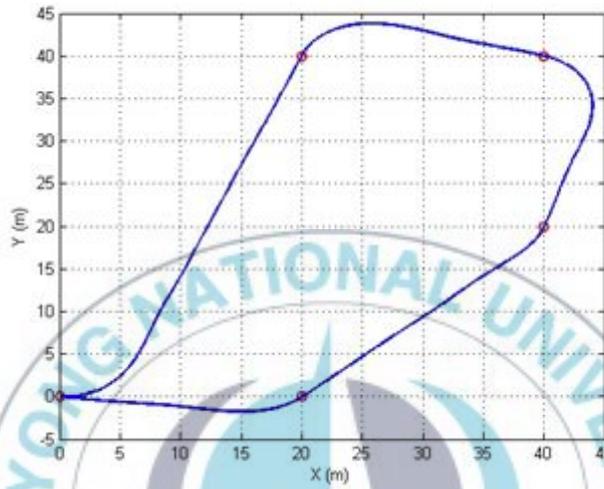


Fig. 34 Way-point Simulation Result 3

제5장 AUV의 운동시뮬레이션 결과 고찰

앞서 4장에서 수행한 AUV의 수평, 수직 운동 시뮬레이션은 AUV의 초기 값을 가정하여 AUV의 운동을 방향타의 각도별 X-Y, X-Z 위치를 그래프로 표현하였다. Fig.24 및 Fig.27를 보면 AUV의 X-Y 수평, X-Z 수직 운동이 수행되는 것을 확인할 수 있다. 그리고 Fig.25 및 Fig.28을 보면 X-Y 수평, X-Z 수직 운동이 수행될 때 AUV의 운동이 완벽하게 수평, 수직으로 표현되는 것을 알 수 있다. X-Y 수평 운동 중에는 AUV가 운동 중에 Z방향으로 정확하게 움직이고, X-Z 수직 운동 중에는 Y방향으로 정확하게 움직이는데 이러한 결과의 원인은 평면 운동 시뮬레이션 과정에서 2차원 운동에 연관되지 않는 비선형 데이터 텀($\sum F, \sum M$)을 모두 0로 두었기 때문인 것으로 판단된다. 하지만 2차원 운동 시뮬레이션상의 AUV의 X-Y 수평운동, X-Z 수직운동 시뮬레이션 결과는 AUV의 운동결과와 예상한 바와 같아 타당하다고 판단할 수 있다.

AUV의 Way-point 추적 시뮬레이션의 경우 4.4장에 기술한 제자리 복귀 Way-point 외에 다른 Way-point를 (0,0), (20,0), (40,20), (60,40), (80,40), (100,20)로 설정하고 시뮬레이션을 수행하였을 때 Fig. 35과 같이 정확하게 Way-point를 추적하는 것을 확인할 수 있다. 이는 시뮬레이션에 사용된 제어기가 AUV의 운동경로를 바르게 추적하는 것을 확인할 수 있다.

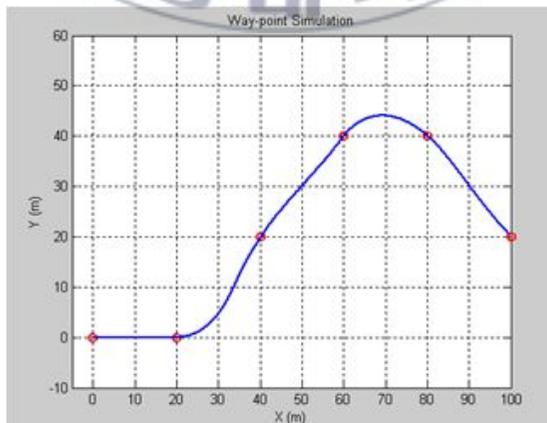


Fig. 35 Verify the Way-point Simulation

제6장 요약 및 결론

본 연구에서는 AUV의 개념설계를 수행하면서 AUV의 임무를 성공적으로 수행할 수 있는 운동성능을 시뮬레이션으로 구현을 목표로 연구를 진행하였다. 먼저 선체는 수심 300m에서도 충분히 견딜 수 있도록 구조적 강도를 가지도록 설계하였고 선체 내에는 수밀구조가 될 수 있도록 설계하였다. 추진기는 목표 성능을 만족하는 것으로 설계 및 선정 하였고 추진기 결합구조를 수밀구조가 될 수 있도록 하였다. 방향타는 회전이 가능하며 방향타 결합부 수밀 구조를 구현하였으며 방향타 축에 발생하는 토크를 계산하여 축 강도를 고려한 설계를 수행하였고 통신 및 전원 시스템은 목표성능이 구현 가능하도록 설계 및 선정 하였으며 AUV에 발생하는 부력 계산을 통해 복원력을 검토하는 밸런싱 설계를 수행하였다.

또한 AUV의 6자유도 운동방정식을 수립하고 수중에서 AUV의 운동에 관계되는 비선형 힘요소에 관하여 기술하였다. 이 6자유도 운동방정식을 활용하여 AUV의 수중운동 시뮬레이션을 위한 동역학 모델을 Matlab 프로그램으로 구현하고 AUV의 X-Y 수평 운동 시뮬레이션 및 X-Z 수직 운동 시뮬레이션, Way-point 추적 시뮬레이션을 수행하면서 설계된 AUV의 운동성능을 검증하였다.

본 연구에서는 AUV의 기본적인 운동성능 설계를 위해 필요한 여러 가지 요소들을 검토하였으므로 AUV 개발을 위한 기본적인 설계지침서가 될 수 있으리라고 생각된다. 향후 AUV의 운동제어를 위한 제어기를 개발하기 위해서는 운동방정식에 대한 추가적인 연구와 적절한 제어기법의 개발이 필요하다고 사료된다.

부 록

AUV의 운동성능 해석용 Matlab Code

1. Matlab Code for AUV Motion Control

```
close all;
clear all;
clc;

disp(sprintf('\nT.Horizontal test\nStraightly forward & slightly downward'));
disp('Simulation start...');
tic;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Definitions of variables for the simulation
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

D2R = pi/180;
R2D = 180/pi;
Ft2M= 0.3048;
tTimeStep = 0.1;      % for 4th order Runge-Kutta
tTotalTime = 500;
tt=0;
tHorizon = tTotalTime/tTimeStep;

disp(sprintf('Time step is %f sec.\nSimulation time is %f sec.', tTimeStep, tTotalTime));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Track [X Y Z]
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
nTr= 6;
indexTr = 1;
Tr = [0, 0, 0;
      20, 0, 0;
      40, 20, 0;
      40, 40, 0;
      20, 40, 0;
      0, 0, 0];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% initial conditions
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% x = [ u v w p q r x y z phi theta psi ]';
x = zeros(12, 1);
x(1) = 2.55;      % initial forward velocity: 2.55 m/s
x(3) = 0;        % downward velocity: 0.25 m/s
```

```

x(10) = 0*D2R;           % 5 deg ==> rad
x(11) = 0*D2R;
x(12) = 0*D2R;
forces = zeros(6, 1);

deltaR = 5*D2R;         % angle of rudder plane [rad]
deltaS = 0*D2R;         % angle of stern plane [rad]
np= 700;                 % rpm of main propeller [rpm]

Eta_FlightHeading = 1.0; %Control tuning parameter
Phi_FlightHeading = 0.5; %Control tuning parameter
r_com=0;

U = sqrt(x(1)*x(1) + x(2)*x(2) + x(3)*x(3)); % speed [m/s]

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Variables for numerical integration
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
k1 = zeros(12, 1);
k2 = zeros(12, 1);
k3 = zeros(12, 1);
k4 = zeros(12, 1);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Variables for data output
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
out = zeros(tHorizon+1, 19);
out2 = zeros(tHorizon+1, 3);
outDot = zeros(tHorizon+1, 12);
out(1, :) = [x', U, forces'];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% time marching
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for ii=1:1:tTotalTime/tTimeStep
    tt=tt+1;
    % angles of control fins to ui
    range = sqrt( ( x(7)-Tr(indexTr,1) )^2 + ( x(8)-Tr(indexTr,2) )^2 );
    if (range < 1 )
        indexTr=indexTr+1;
        if (indexTr==nTr+1)
            break;
        end
    end
end

c_psi = atan2( Tr(indexTr,2)-x(8) , Tr(indexTr,1)-x(7) ) - x(12);

while(abs(c_psi) > pi)
    c_psi = c_psi - sign(c_psi)*2.0*pi;
end;
while(abs(x(12)) > 2*pi)
    x(12) = x(12) - sign(x(12))*2.0*pi;

```

```

end;

% LOS PD control
deltaR = - 0.9*c_psi - 0.9*(r_com - x(6)) ;
ui= [0 deltaR np];

% Integration by Explicit Euler
[k1, U, forces] = REMUS(x, ui);
[k2, U, forces] = REMUS(x + k1*tTimeStep, ui);
x = x + tTimeStep*(k1 + k2)/2;
k1 = k2;

out(ii+1, :) = [x', U, forces'];
out2(ii+1, :) = ui;
outDot(ii+1, :) = k1';

end

toc;
disp(sprintf('Simulation End...\nNow... data plot\n'));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% data loading
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% x = [u v w p q r xpos ypos zpos phi theta psi]'
u = out(:, 1) ; % surge speed [m/s]
v = out(:, 2) ; % sway speed [m/s]
w = out(:, 3) ; % heave speed [m/s]
p = out(:, 4)*R2D; % rate of roll [deg/s]
q = out(:, 5)*R2D; % rate of pitch [deg/s]
r = out(:, 6)*R2D; % rate of yaw [deg/s]
x = out(:, 7) ; % x-position in local geographic [m]
y = out(:, 8) ; % y-position in local geographic [m]
z = out(:, 9) ; % z-position in local geographic [m]
phi = out(:, 10)*R2D; % angle of roll [deg]
theta = out(:, 11)*R2D; % angle of theta [deg]
psi = out(:, 12)*R2D; % angle of psi [deg]
X = out(:, 14); % x-dir. force in b-frame [N]
Y = out(:, 15); % y-dir. force in b-frame [N]
Z = out(:, 16); % z-dir. force in b-frame [N]
K = out(:, 17); % x-dir. moment in b-frame [N*m]
M = out(:, 18); % y-dir. moment in b-frame [N*m]
N = out(:, 19); % z-dir. moment in b-frame [N*m]

tt=0:tTimeStep:size(u)*tTimeStep;
plot(Tr(:,1), Tr(:,2), 'o', 'color', 'red', 'linewidth',2);
grid on;
hold;
plot(x,y, 'linewidth',2);
xlabel('X (m)'), ylabel('Y (m)');

```

2. Matlab Code for AUV Model Data

```

function [xDot, U, forces] = REMUS(x, ui)
% [xDot, U, forces] = REMUS(x, ui) returns the time derivative of the state vector:
% x = [ u v w p q r x y z phi theta psi ]', the speed U in m/s (optionally) and
% hydrodynamic forces for an Autonomous Underwater Vehicle (AUV), REMUS at the WHOI.
%
% Author: Gyungnam Jo (Ocean Engineering Lab., Seoul National Univ.)
% Revised by: Dong C. Seo (Ocean Engineering Lab., Seoul National Univ.)
%
% Ver.: 1.01
% Last update: Jan 28, 2009
%
% version history
% 0.1: Test for T. Prestero's original code
% 0.2: changing output format
% 0.3: changing mass matrix
% 0.5: Some bugs are fixed.
% 0.9: Some bugs are fixed.
% 0.99: Some bugs are fixed. (RTM version)
% 1.01: Some bugs are fixed.
%
% - Reference -
% 1. T. Prestero (2001), "Verification of a Six-Degree of Freedom Simulation Model for
% the REMUS Autonomous Underwater Vehicle", Master thesis of MIT
% 2. T. I. Fossen (2002), Marine Control Systems, Marine cybernetics,
% Trondheim, Norway (http://www.marinecybernetics.com)
%
% The length of the AUV is L = 1.33 m and the forward speed is u = 1.54 m/s, while the
% state vector is defined as:
%
% TERMS
% -----
% Input vectors = state vector, control input
% Control input:
% ui = [deltaS deltaR]'
%
% Control Fin Angles:
% deltaS = angle of stern planes [rad]
% deltaR = angle of rudder planes [rad]
%
% State vector:
% x = [u v w p q r xpos ypos zpos phi theta psi]'
%
% Body-referenced coordinates:
% u = Surge velocity [m/sec]
% v = Sway velocity [m/sec]
% w = Heave velocity [m/sec]
% p = Roll rate [rad/sec]
% q = Pitch rate [rad/sec]
% r = Yaw rate [rad/sec]
%
% Earth-fixed coordinates (geographic frame, NOT geodetic frame):
% xpos = Position in x-direction [m]
% ypos = Position in y-direction [m]
% zpos = Position in z-direction [m]
% phi = Roll angle [rad]
% theta = Pitch angle [rad]
% psi = Yaw angle [rad]

```

```

%
% Output vectors = derivatives of
% [xDot, U, forces]

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Initialize global variables
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Standard REMUS Hull Parameters
% -----
rho      = 1.03E+03; % Seawater Density [kg/m^3]
A_f      = 2.85E-02; % Hull Frontal Area [m^2]
A_p      = 2.26E-01; % Hull Projected Area (XZ-plane) [m^2]
S_w      = 7.09E-01; % Hull Wetted Surface Area [m^2]
Vol      = 3.15E-02; % Estimated Hull Volume [m^3]
W        = 2.99E+02; % Measured Vehicle Weight [N]
B        = 2.99E+02; % Measured Vehicle Weight (horizontal) [N]
% B      = 2.968E+02; % Measured Vehicle Weight (Pitch Up) [N]
% B      = 3.0577E+02; % Measured Vehicle Weight (Pitch down) [N]
% B      = 3.08E+02; % Measured Vehicle Buoyancy [N]
B_est    = 3.17E+02; % Estimated Hull Buoyancy [N]
x_cb     = 5.54E-03; % Estimated Long. Center of Buoyancy [m]
C_d      = 3.00E-01; % REMUS Axial Drag Coeff. [n/a]
C_dc     = 1.10E+00; % Cylinder Cross-flow Drag Coeff. [n/a]
c_yd     = 1.20E+00; % Hoerner Body Lift Coeff. [n/a]
x_cp     = -3.21E-01; % Center of Pressure [n/a]
alpha    = 3.59E-02; % Ellipsoid Added Mass Coeff. [n/a]

% Center of Buoyancy w.r.t. Vehicle Nose
% -----
% xB     = -6.11E-01; % Center of Buoyancy: X-dir [m]
% yB     = 0.00E+00; % Center of Buoyancy: Y-dir [m]
% zB     = 0.00E+00; % Center of Buoyancy: Z-dir [m]

% Center of Buoyancy w.r.t. Origin(CB)
% -----
xB       = 0.00E+00; % Center of Buoyancy: X-dir [m]
yB       = 0.00E+00; % Center of Buoyancy: Y-dir [m]
zB       = 0.00E+00; % Center of Buoyancy: Z-dir [m]

% Center of Gravity w.r.t. Origin at CB
% -----
xG       = 0.00E+00; % Center of Gravity: X-dir [m]
yG       = 0.00E+00; % Center of Gravity: Y-dir [m]
zG       = 1.96E-02; % Center of Gravity: Z-dir [m]

% Moments of Inertia w.r.t. Origin at CB
% -----
Ixx      = 1.77E-01; % Moment of Inertia [kg·m^2]
Iyy      = 3.45E+00; % Moment of Inertia [kg·m^2]
Izz      = 3.45E+00; % Moment of Inertia [kg·m^2]

% Non-linear Force Coefficients
% -----
Xuu      = -3.90E+00; % Cross-flow Drag [kg/m]
Xudot    = -9.30E-01; % Added Mass [kg]
Xwq      = -3.55E+01; % Added Mass Cross-term [kg/rad]
Xqq      = -1.93E+00; % Added Mass Cross-term [kg·m/rad]

```

Xvr	=	3.55E+01;	% Added Mass Cross-term	[kg/rad]
Xrr	=	-1.93E+00;	% Added Mass Cross-term	[kg·m/rad]
Xprop	=	9.25E+00;	% Propeller Thrust	[N]
Yvv	=	-1.31E+03;	% Cross-flow Drag	[kg/m]
Yrr	=	6.32E-01;	% Cross-flow Drag	[kg·m/rad ²]
Yuv	=	-2.86E+01;	% Body Lift Force and Fin Lift	[kg/m]
Yvdot	=	-3.55E+01;	% Added Mass	[kg]
Yrdot	=	1.93E+00;	% Added Mass	[kg·m/rad]
Yur	=	5.22E+00;	% Added Mass Cross Term and Fin Lift	[kg/rad]
Ywp	=	3.55E+01;	% Added Mass Cross-term	[kg/rad]
Ypq	=	1.93E+00;	% Added Mass Cross-term	[kg·m/rad]
YuudeltaR	=	9.64E+00;	% Fin Lift Force	[kg/(m·rad)]
Zww	=	-1.31E+02;	% Cross-flow Drag	[kg/m]
Zqq	=	-6.32E-01;	% Cross-flow Drag	[kg·m/rad ²]
Zuw	=	-2.86E+01;	% Body Lift Force and Fin Lift	[kg/m]
Zwdot	=	-3.55E+01;	% Added Mass	[kg]
Zqdot	=	-1.93E+00;	% Added Mass	[kg·m/rad]
Zuq	=	-5.22E+00;	% Added Mass Cross-term and Fin Lift	[kg/rad]
Zvp	=	-3.55E+01;	% Added Mass Cross-term	[kg/rad]
Zrp	=	1.93E+00;	% Added Mass Cross-term	[kg/rad]
ZuudeltaS	=	-9.64E+00;	% Fin Lift Force	[kg/(m·rad)]

% Non-linear Moment Coefficients

Kpp	=	-1.30E-01;	% Rolling Resistance	[kg·m ² /rad ²]
Kpdot	=	-7.04E-02;	% Added Mass	[kg·m ² /rad]
Kprop	=	-5.43E-01;	% Propeller Torque	[N·m]
Mww	=	3.18E+00;	% Cross-flow Drag	[kg]
Mqq	=	-1.88E+02;	% Cross-flow Drag	[kg·m ² /rad ²]
Muw	=	2.40E+01;	% Body and Fin Lift and Munk Moment	[kg]
Mwdot	=	-1.93E+00;	% Added Mass	[kg·m]
Mqdot	=	-4.88E+00;	% Added Mass	[kg·m ² /rad]
Muq	=	-2.00E+00;	% Added Mass Cross Term and Fin Lift	[kg·m/rad]
Mvp	=	-1.93E+00;	% Added Mass Cross Term	[kg·m/rad]
Mrp	=	4.86E+00;	% Added Mass Cross Term	[kg·m ² /rad ²]
MuudeltaS	=	-6.15E+00;	% Fin Lift Moment	[kg/rad]
Nvv	=	-3.18E+00;	% Cross-flow Drag	[kg]
Nrr	=	-9.40E+01;	% Cross-flow Drag	[kg·m ² /rad ²]
Nuv	=	-2.40E+01;	% Body and Fin Lift and Munk Moment	[kg]
Nvdot	=	1.93E+00;	% Added Mass	[kg·m]
Nrdot	=	-4.88E+00;	% Added Mass	[kg·m ² /rad]
Nur	=	-2.00E+00;	% Added Mass Cross Term and Fin Lift	[kg·m/rad]
Nwp	=	-1.93E+00;	% Added Mass Cross Term	[kg·m/rad]
Npq	=	-4.86E+00;	% Added Mass Cross Term	[kg·m ² /rad ²]
NuudeltaR	=	-6.15E+00;	% Fin Lift Moment	[kg/rad]

% Auxiliary variables

delta_max	=	1.36E+01;	% Maximum Fin Angle	[deg]
D2R	=	pi/180;	% type conversion: deg to rad	
R2D	=	180/pi;	% type conversion: rad to deg	

% Inertia matrix

m	=	3.048E+01;	% weight/G = 2.99E+2/9.81	[kg]
% m11	=	m - Xudot;		
% m15	=	m*zG;		
% m16	=	-m*yG;		
% m22	=	m - Yvdot;		
% m24	=	-m*zG;		
% m26	=	m*xG - Yrdot;		

```

% m33 = m - Zwdot;
% m34 = m*yG;
% m35 = -m*xG - Zqdot;
% m42 = -m*zG;
% m43 = m*yG;
% m44 = Ixx - Kpdot;
% m51 = m*zG;
% m53 = -m*xG - Mwdot;
% m55 = Iyy - Mqdot;
% m61 = -m*yG;
% m62 = m*xG - Nvdot;
% m66 = Izz - Nrdot;
%
% Inertia = zeros(6, 6);
% Inertia(1, 1) = m11;   Inertia(1, 5) = m15;   Inertia(1, 6) = m16;
% Inertia(2, 2) = m22;   Inertia(2, 4) = m24;   Inertia(2, 6) = m26;
% Inertia(3, 3) = m33;   Inertia(3, 4) = m34;   Inertia(3, 5) = m35;
% Inertia(4, 2) = m42;   Inertia(4, 3) = m43;   Inertia(4, 4) = m44;
% Inertia(5, 1) = m51;   Inertia(5, 3) = m53;   Inertia(5, 5) = m55;
% Inertia(6, 1) = m61;   Inertia(6, 2) = m62;   Inertia(6, 6) = m66;
% Minv = inv(Inertia);

```

```

% Inertia matrix by external code
Minv = zeros(6, 6);
Minv(1, 1) = 3.188077818202990e-02;
Minv(1, 3) = 6.733692984755734e-05;
Minv(1, 5) = -2.302015871161572e-03;
Minv(2, 2) = 1.560299856712500e-02;
Minv(2, 4) = 3.767726826187958e-02;
Minv(2, 6) = 3.615100508349490e-03;
Minv(3, 1) = 6.733692984755734e-05;
Minv(3, 3) = 1.525966866733359e-02;
Minv(3, 5) = -3.540382730917190e-03;
Minv(4, 2) = 3.767726826187958e-02;
Minv(4, 4) = 4.133018195140634e+00;
Minv(4, 6) = 8.729547148310637e-03;
Minv(5, 1) = -2.302015871161572e-03;
Minv(5, 3) = -3.540382730917190e-03;
Minv(5, 5) = 1.210333951222364e-01;
Minv(6, 2) = 3.615100508349490e-03;
Minv(6, 4) = 8.729547148310637e-03;
Minv(6, 6) = 1.208856115223427e-01;

```

```

% Get and check state variables and control inputs

```

```

% Get state variables

```

```

% -----
u = x(1);      v = x(2);      w = x(3);
p = x(4);      q = x(5);      r = x(6);
phi = x(10);   theta = x(11);  psi = x(12);
U = sqrt(u*u + v*v + w*w);

```

```

% Get control inputs

```

```

% -----
deltaS = ui(1); deltaR = ui(2);

```

```

% Trigonometric functions:

```

```

% -----
cPhi = cos(phi);
cThe = cos(theta);
cPsi = cos(psi);
sPhi = sin(phi);

```

```

sThe = sin(theta);
sPsi = sin(psi);
tThe = tan(theta);

% Check control inputs (useful later)
% -----
if deltaS > delta_max*D2R
    deltaS = sign(deltaS)*delta_max*D2R;
end

if deltaR > delta_max*D2R
    deltaR = sign(deltaR)*delta_max*D2R;
end

% Hydrostatic forces:
% -----
% T. Prestero
Xhs = -(W-B)*sThe;
Yhs = (W-B)*sPhi*cThe;
Zhs = (W-B)*cPhi*cThe;
Khs = (yG*W-yB*B)*cPhi*cThe - (zG*W-zB*B)*sPhi*cThe; % correction by gnjo
Mhs = -(zG*W-zB*B)*sThe - (xG*W-xB*B)*cPhi*cThe;
Nhs = (xG*W-xB*B)*sPhi*cThe + (yG*W-yB*B)*sThe; % correction by gnjo
% Khs = -(yG*W-yB*B)*cPhi*cThe - (zG*W-zB*B)*sPhi*cThe; % in thesis ==> incorrect
% Nhs = -(xG*W-xB*B)*sPhi*cThe - (yG*W-yB*B)*sThe; % in thesis ==> incorrect

% Static propeller Thrust:
% under 1500 RPM at sea water, the thrust matches the vehicle axial drag ==> 1.54 m/s
% -----
% Xprop1 = -Xuu*u*abs(u) = -2.37*Xuu;

% Static propeller Torque:
% under 1500 RPM at sea water, the torque matches the static roll moment. (-5.3 deg)
% -----
% Kprop = 0.995*(yG*W - yB*B) - 0.093*(zG*W - zB*B);
% Khs = -Kprop;

% Set total forces from equations of motion
% -----
X = Xhs + Xuu*u*abs(u) + (Xwq-m)*w*q + (Xqq + m*xG)*q*q + (Xvr+m)*v*r ...
    + (Xrr + m*xG)*r*r - m*yG*p*q - m*zG*p*r + Xprop;

% corrected by gnjo
Y = Yhs + Yvv*v*abs(v) + Yrr*r*abs(r) + Yuv*u*v + (Ywp+m)*w*p + (Yur-m)*u*r ...
    - (m*zG)*q*r + (Ypq - m*xG)*p*q + m*yG*(r*r + p*p) + YuudeltaR*u*deltaR;
% incorrect in thesis
% Y = Yhs + Yvv*v*abs(v) + Yrr*r*abs(r) + Yuv*u*v + (Ywp+m)*w*p + ...
%     (Yur-m)*u*r - (m*zG)*q*r + (Ypq - m*xG)*p*q + YuudeltaR*u*deltaR;

% corrected by gnjo
Z = Zhs + Zww*w*abs(w) + Zqq*q*abs(q) + Zuw*u*w + (Zuq+m)*u*q + (Zvp-m)*v*p + ...
    (m*zG)*p*p + (m*zG)*q*q + (Zrp - m*xG)*r*p - m*yG*r*q + ZuudeltaS*u*deltaS;
% incorrect in thesis
% Z = Zhs + Zww*w*abs(w) + Zuw*u*w + (Zuq+m)*u*q + (Zvp-m)*v*p + ...
%     (m*zG)*p*p + (m*zG)*q*q + (Zrp - m*xG)*r*p + ZuudeltaS*u*deltaS;

K = Khs + Kpp*p*abs(p) - (Izz-Iyy)*q*r - (m*zG)*w*p + (m*zG)*u*r ...
    + m*yG*u*q - m*yG*v*p + Kprop;

M = Mhs + Mww*w*abs(w) + Mqq*q*abs(q) + (Mrp - (Ixx-Izz))*r*p + (m*zG)*v*r ...
    - (m*zG)*w*q + (Muq - m*xG)*u*q + Muw*u*w + (Mvp + m*xG)*v*p ...

```

```

+ MuudeltaS*u*u*deltaS;

% corrected by gnjo
N = Nhs + Nvv*v*abs(v) + Nrr*r*abs(r) + Nuv*u*v + (Npq - (Iyy-Ixx))*p*q ...
    + (Nwp + m*xG)*w*p + (Nur - m*xG)*u*r - m*yG*v*r + m*yG*w*q + NuudeltaR*u*u*deltaR;
% incorrect in thesis
% N = Nhs + Nvv*v*abs(v) + Nrr*r*abs(r) + Nuv*u*v + (Npq - (Iyy-Ixx))*p*q ...
%     + (Nwp - m*xG)*w*p + (Nur + m*xG)*u*r + NuudeltaR*u*u*deltaR;

forces = [X Y Z K M N]';
xDot = ...
[Minv(1,1)*X          + Minv(1,3)*Z          + Minv(1,5)*M;
  + Minv(2,2)*Y          + Minv(2,4)*K          + Minv(2,6)*N;
Minv(3,1)*X          + Minv(3,3)*Z          + Minv(3,5)*M;
  + Minv(4,2)*Y          + Minv(4,4)*K          + Minv(4,6)*N;
Minv(5,1)*X          + Minv(5,3)*Z          + Minv(5,5)*M;
  + Minv(6,2)*Y          + Minv(6,4)*K          + Minv(6,6)*N;
cPsi*cThe*u + (cPsi*sThe*sPhi - sPsi*cPhi)*v + (sPsi*sPhi + cPsi*cPhi*sThe)*w;
sPsi*cThe*u + (cPhi*cPsi + sPhi*sThe*sPsi)*v + (cPhi*sThe*sPsi - cPsi*sPhi)*w;
  -sThe*u          + cThe*sPhi*v          + cPhi*cThe*w;
  p          + sPhi*tThe*q          + cPhi*tThe*r;
          cPhi*q          - sPhi*r;
          (sPhi/cThe)*q          + (cPhi/cThe)*r];

```



참 고 문 헌

- 1) "Guidance and Control of Ocean Vehicle", "Thor I.Fossen", 1999
- 2) "Modeling, Identification, and Analysis of Limit-cycling Pitch and Heave Dynamics in an ROV", "Sergio M. Savaresi, Fabio Previdi, Alessandro Dester", 2004 IEEE
- 3) "Preliminary Experiments in the Model-Based Dynamic Control of marine Thruster", "Louis L. Whitcomb, Dana R. Yoerger", 1996 IEEE
- 4) "AUV/ROV propulsion thruster", "Bert Dewijs", 2000 IEEE
- 5) "Navigation and control for a Test Bed AUV-SNUUV I", "Kihun Kim, Hang S. Choi", 2004 IEEE
- 6) "Estimation of AUV dynamics for sensor fusion", "Kjell Magne Fauske, Fredrik Gustafsson, Øyvind Hegrenæs",
- 7) "Sliding mode control of remotely operated vehicles for horizontal plane motions" "N.Olgac, B.E.Platin, J.M.Chang", IEE Vol138.No.5 1991
- 8) "Modified S-Plane Control of AUV Based on Motion Compensation", "Zhang Lei; Pang Yong-Jie; Wan Lei; Qin Zai-Bai", ISA 2009. International Workshop on , vol., no., pp.1-5, 23-24 May 2009
- 9) "Coupled, non-linear control system design for autonomous underwater vehicle (AUV)", "Santhakumar, M.; Asokan, T", ICARCV 2008. 10th International Conference on , vol., no., pp.2309-2313, 17-20 Dec. 2008

- 10) "The synthesis of multi-channel adaptive variable structure system for the control of AUV," "Lebedev, A.V.; Filaretov, V.F", IROS 2008. I EEE/RSJ International Conference on , vol., no., pp.2834-2839, 22-26 Sept. 2008
- 11) "A semi-physical virtual simulation system for AUV," "Wang Hong-jian; Shi Xiao-cheng; Zhao Jie; Li Juan" , 04. MTTs/IEEE TECHNO-OCEAN '04 , vol.3, no., pp.1560-1563 Vol.3, 9-12 Nov. 2004
- 12) "ROV (원격조종 수중로봇)의 소개와 ROV해미래의 운동 제어 및 신호 처리", "최현택, 김기훈, 이판목, 이종무, 전봉환", 한국정밀공학회지 제 26권 5호(2009년 5월)
- 13) "근해 조사용 무인잠수정의 개발 ", "김경기, 최형식, 강형석, 정구락, 권경엽", 대한기계학회 논문집 A권, 제 32권 제11호
- 14) "ROV (원격조종 수중로봇)의 소개와 ROV해미래의 운동 제어 및 신호 처리", "최현택, 김기훈, 이판목, 이종무, 전봉환", 한국정밀공학회지 제 26권 5호.
- 15) "확장칼만필터를 이용한 무인잠수정의 3차원 위치평가", "임종환, 강철웅", 한국정밀기계공학회지 제 21권 20 7호 (2004년 7월)

후 기

우선 지난 2년 6개월 동안 부족한 저를 아낌없는 가르침과 격려를 베풀어 주신 백운경 교수님께 깊은 감사를 드립니다. 그동안 제가 잘 못한 부분도 많았고 부족한 부분도 많았지만 저를 믿어주셔서 이 자리에 올 수 있었던 것 같습니다.

또한 바쁘신 와중에도 시간을 내주시어 논문 심사를 해주신 안두성 교수님과 강인필 교수님께도 감사드립니다.

그리고 저에게 많은 조언과 가르침을 주셨던 종석선배와 영기선배께도 감사드립니다. 그리고 지금 열심히 공부중인 명욱이, 창민이, 그리고 내년 에 들어올 상태, 남은 대학원 기간 중에 열심히 하면 좋은 결과가 있을 겁니다. 그리고 우리 타지에서 고생하고 있는 동기 지웅이, 그리고 S&T 대우 근무 중에 많은 도움을 줬던 태호 선배, 성원 선배, 동규 선배께도 감사드립니다.

그리고 2년 6개월 동안 제가 열심히 할 수 있도록 응원해 주었던 내 아내와 내 딸 서원이 그리고 자식을 믿어주신 부모님과 사위를 믿어준 장인 장모님께도 사랑한다고 전하고 싶습니다.

지난 2년 6개월 동안에 부족하지만 제가 배우고 익힌 것을 활용하여 이 논문을 완성하면서 많은 것을 느꼈습니다. 좀 더 노력을 해서 앞으로는 더욱 멋진 남편, 자랑스러운 아버지, 남부럽지 않은 아들과 믿음이 가는 사위, 든직한 선후배가 되도록 열심히 살겠습니다.

-2010년 12월 논문작성을 마치고... 현지민