

논문 2014-09-20

MOOS-IvP를 이용한 무인잠수정 제어기 개발의 효용성

(The Effectiveness of MOOS-IvP based Design of Control System for Unmanned Underwater Vehicles)

김지연, 이동익*
(Jiyeon Kim, Dongik Lee)

Abstract : This paper demonstrates the benefit of using MOOS-IvP in the development of control system for Unmanned Underwater Vehicles(UUV). The demand for autonomy in UUVs has significantly increased due to the complexity in missions to be performed. Furthermore, the increased number of sensors and actuators that are interconnected through a network has introduced a need for a middleware platform for UUVs. In this context, MOOS-IvP, which is an open source software architecture, has been developed by several researchers from MIT, Oxford University, and NUWC. The MOOS software is a communication middleware based on the publish-subscribe architecture allowing each application to communicate through a MOOS database. The IvP Helm, which is one of the MOOS modules, publishes vehicle commands using multi-objective optimization in order to implement autonomous decision making. This paper explores the benefit of MOOS-IvP in the development of control software for UUVs by using a case study with an auto depth control system based on self-organizing fuzzy logic control. The simulation results show that the design and verification of UUV control software based on MOOS-IvP can be carried out quickly and efficiently thanks to the reuse of source codes, modular-based architecture, and the high level of scalability.

Keywords : Unmanned underwater vehicle(UUV), MOOS-IvP, Depth control, MOOS application

I. 서론

1950년대 미국을 중심으로 개발되기 시작한 무인잠수정(Unmanned Underwater Vehicle, UUV)은 1980년대부터 수중 센서/액추에이터 및 마이크로프로세서 기술의 발전에 힘입어 급속히 발전되고 있다. 최근에는 무인잠수정이 수행해야 할 임무의 복잡성이 증가함에 따라 자율성(autonomy)을 갖는 자율무인잠수정(Autonomous Underwater Vehicle, AUV)의 개발에까지 이르렀다[1]. AUV는 통신이 어려운 수중 환경에서 다양한 임무를 수행해야 하

므로 경로계획, 장애물 탐지 및 충돌회피, 고장복구, 임무제할당 등의 과정이 자율적으로 처리되어야 한다. 그 결과, 탑재되는 센서와 액추에이터 개수가 늘어나고 네트워크를 통해 데이터를 공유하게 됨으로써, 무인잠수정의 소프트웨어를 보다 손쉽게 개발하고 검증하기 위한 미들웨어가 도입되고 있다[2]. MOOS-IvP(Mission Oriented Operating Suite-Interval Programming)는 이러한 요구에 부응하기 위해 개발된 C++ 기반의 오픈소스 소프트웨어 플랫폼으로, 이미 Bluefin 21-inch UUV, Hydroid REMUS-100, REMUS-600, Ocean Server Iver2 등 실제 무인잠수정에 탑재되어 운용되고 있다[3].

MOOS-IvP 아키텍처는 크게 무인잠수정의 각 모듈간 통신 인터페이스를 지원하는 MOOS, 그리고 항해와 관련된 자율 시스템을 담당하는 IvP로 구성된다[4]. MOOS 부분은 MIT와 옥스퍼드대학교 연구진에 의해 공동 개발되었으며, 무인잠수정이나 무인 지상로봇 어플리케이션의 유비쿼터스 환경을 구

*Corresponding Author (dilee@ee.knu.ac.kr)

Received: 29 Nov. 2013, Revised: 7 Jan. 2014,

Accepted: 3 Feb. 2014.

J. Kim, D. Lee: Kyungpook National University

※ 이 논문은 2012학년도 경북대학교 학술연구비에 의하여 연구되었음.

축할 수 있는 미들웨어 기능을 제공한다. 여기에 덧붙여 NUWC(Naval Undersea Warfare Center)에서 개발한 IvP Helm이라는 어플리케이션이 추가되어 MOOS-IvP 아키텍처를 이루고 있다. IvP Helm 어플리케이션에 포함된 IvP 알고리즘은 행위 (behavior) 기반의 아키텍처에서 각 행위 간 중재를 위해 다목적 최적화 기법을 적용한다. MOOS-IvP는 데이터베이스를 중심으로 각 MOOS 모듈들이 독립적으로 연결되어 있으므로 개발자가 제공하는 기본 모듈 외에 사용자가 직접 개발한 다양한 어플리케이션을 추가할 수 있다. 기본 모듈의 소스코드를 재사용할 수 있다는 점도 중요한 장점이다.

본 논문에서는 사례연구를 통해 MOOS-IvP 아키텍처의 모듈 확장성과 개발기간 단축 및 재사용 편의성을 확인함으로써 MOOS-IvP에 대한 국내 연구자들의 관심을 높이는데 주목적 있다. 따라서 본 논문에서는 사례연구로서 기 개발된 자기구성 퍼지제어(self-organizing fuzzy control) 기반 자동심도제어기[5]를 하나의 독립적인 MOOS 어플리케이션으로 구현하고, 이를 표준 MOOS 아키텍처에 적용하여 시뮬레이션을 수행한다. 즉, 표준 MOOS-IvP에 포함된 무인잠수정의 수학모델을 전혀 모르는 상태에서 자동심도제어기를 모듈로 설계하고, 이를 MOOS-IvP 아키텍처에 통합하여 제어성능을 검증한다. 제어기를 제외하고, 시뮬레이션에 필요한 무인잠수정의 운동방정식, 센서, 액추에이터 등의 모델 및 자율 알고리즘은 표준 MOOS에서 제공되는 모듈을 재사용하므로 제어기 설계검증 및 시뮬레이션 환경 구축에 필요한 시간과 노력을 획기적으로 줄일 수 있음을 확인한다.

II. MOOS-IvP 개요

1. MOOS-IvP의 구조

MOOS는 그림 1과 같이 MOOSDB라는 하나의 데이터베이스를 중심으로 다양한 기능의 어플리케이션(MOOSApp)들이 스타 토폴로지 구조로 연결되어 있다. 각각의 MOOS 어플리케이션은 publish-scribe 방식으로 MOOSDB에게 데이터를 전달하며, 다른 어플리케이션에서 제공하는 정보가 필요한 경우 MOOSDB를 통해서만 제공받을 수 있다. 각 MOOS 어플리케이션은 다른 어플리케이션과는 별개로 독립적인 내부 주기를 가지고 MOOSDB와 통신하기 때문에 추가적인 어플리케이션의 확장이 용이하다.

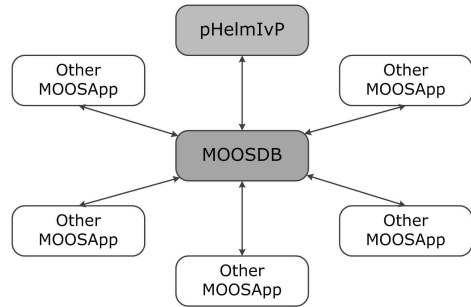


그림 1. MOOSDB와 MOOS 어플리케이션
Fig. 1 MOOSDB and MOOS application

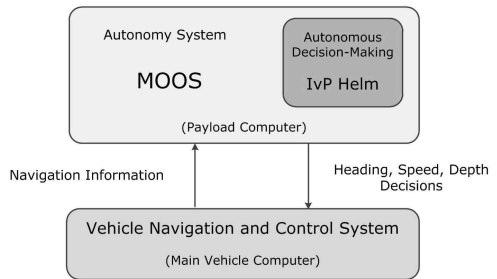


그림 2. Backseat driver 구조
Fig. 2 Backseat driver paradigm

전체적인 MOOS-IvP 아키텍처는 그림 2와 같이 항법제어(control and navigation) 모듈과 자율제어 (autonomy system) 모듈이 분리된 ‘backseat driver’ 구조로 이루어져 있다. 자율제어 모듈은 ‘frontseat driver’인 항법제어 모듈로부터 무인잠수정의 현재 상태정보를 획득하여 속도, 방향, 심도 명령값을 계산한다.

2. IvP 자율화 어플리케이션

MOOSDB와 연결된 하나의 어플리케이션인 행위 기반의 IvP 모듈은 pHElmIvP라고도 하며 무인잠수정의 방향, 속도, 심도를 결정하는 자율항해와 관련되어 있다. 행위 기반의 제어 구조는 정보를 입출력 처리하는 단위인 행위 모듈의 집합으로 이루어져 있으며, 각 모듈이 독립적으로 동작하기 때문에 응답속도가 빠르고 확장성이 우수하다. IvP 알고리즘은 ‘interval programming’이라는 선형 프로그래밍 모델을 사용하여, 미리 선언되어 있는 행위들로부터 생성된 목적함수(objective function)를 통

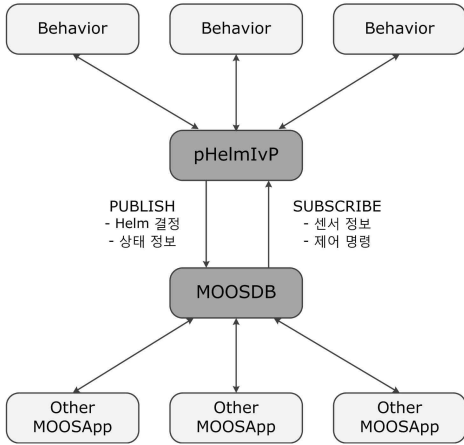


그림 3. pHelmIvP와 MOOS 어플리케이션
Fig. 3 The pHelmIvP and MOOS application

해 무인잠수정의 다목적 최적화(multi-objective optimization) 문제를 해결한다[6]. IvP Helm도 MOOS 구조와 마찬가지로 그림 3과 같이 다수의 행위 모듈과 스타 토폴로지 구조로 연결되어 있다. pHelmIvP와 MOOSDB 간에는 현재 무인잠수정의 위치와 방향, 주변 환경 및 장애물에 대한 센서 정보, 제어 명령 등을 publish-subscribe 방식으로 주고 받는다.

IvP Helm 내부에는 그림 4와 같이 사전에 선언된 미션모드와 각각의 IvP 행위, 그리고 IvP Solver가 포함되어 있다. MOOSDB로부터 무인잠수정의 센서 정보와 제어 명령값을 받아 IvP Helm의 버퍼에 저장한다. 이 단계에서 행위 파일에 선언된 미션 모드가 있을 경우, 이를 분석하여 미션에 포함되어 있는 IvP 행위를 순차적으로 구분한다. 이 때 IvP 행위의 구분 순서는 고려하지 않는다. 각각의 IvP 행위 모듈에서 IvP 함수와 각 행위에 대한 출력값(variable-value)을 계산하고, IvP Solver에서 IvP 함수들에 대한 가중치를 최적화하여 하나의 출력값을 계산하게 된다. n 개의 decision 변수(x_1, \dots, x_n)로 이루어진 i 번째 목적함수를 $f_i(\vec{x})$, 각 함수에 대한 가중치를 ω_i 라고 할 때, k 개의 구간선형(piecewise linear) 함수에 대한 IvP 알고리즘은 다음과 같다[6].

$$\vec{x}^* = \arg\max_x \sum_{i=0}^{k-1} \omega_i f_i(\vec{x}) \quad (1)$$

IvP 알고리즘에 의해 계산된 명령값은 DESIRED_HEADING, DESIRED_SPEED, DESIRED_DEPTH

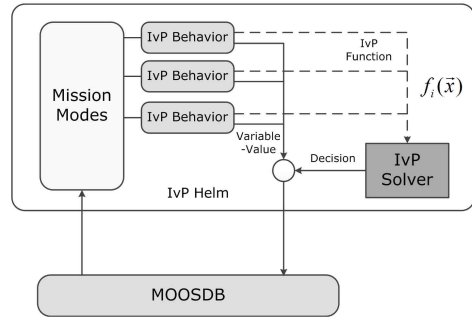


그림 4. IvP Helm 모듈의 내부 동작 루프
Fig. 4 Iterate loop of IvP Helm

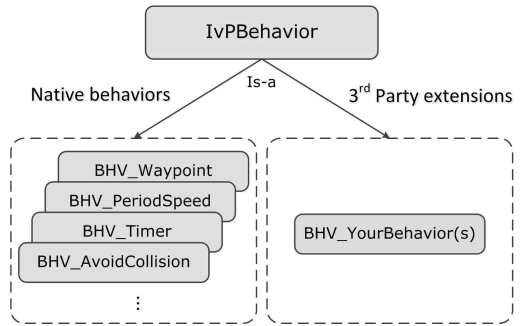


그림 5. 행위 클래스의 상속
Fig. 5 Behavior inheritance

의 형태로 MOOSDB에게 publish된다.

개발자가 제공하는 표준 MOOS-IvP 소프트웨어는 한 대의 무인잠수정에 관련된 동작 뿐 아니라, 여러 대의 무인잠수정의 협업 제어와 관련된 동작 등 30개의 행위 클래스를 포함한다. C++ 기반의 모든 행위 클래스는 그림 5와 같이 IvPBehavior라는 슈퍼클래스와 IS-A 상속 관계를 가지고 있으며, 사용자가 직접 구현하고 싶은 행위는 기존의 행위 클래스의 소스코드를 재사용하여 IvP Helm 내에 쉽게 추가할 수 있다.

III. 사례연구-자기구성퍼지 심도제어기

1. 자동심도제어기 모듈

앞서 소개한 MOOS-IvP 구조의 모듈 확장성을 확인하기 위한 사례연구로써 무인잠수정의 심도제어기를 설계한다. 즉 참고문헌[5]에서 제안한 자기구성 퍼지제어(Self-Organizing fuzzy logic Control, SOC) 기반의 자동심도제어기를 설계하여

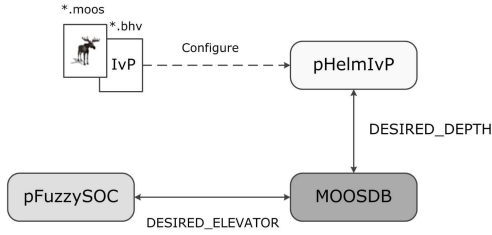


그림 6. pHelmIvP와 pFuzzySOC 어플리케이션
Fig. 6 The pHelmIvP and pFuzzySOC application

‘pFuzzySOC’ 라는 하나의 MOOS 어플리케이션으로 모듈화하고, 이를 표준 MOOS-IvP에 포함된 PID 기반의 기존 어플리케이션 모듈 ‘pMarinePID’을 대신하여 적용한다.

그림 6과 같이 MOOS 파일(*.moos)에 지정된 특정 미션이 실행됨과 동시에 사용자가 행위 파일(behavior file, *.bhv)에 설정한 각 행위들의 세부 명령이 pHelmIvP 모듈에 전달되고, pHelmIvP에서는 행위 파일에 지정된 심도값을 DESIRED_DEPTH라는 키와 함께 MOOSDB로 publish한다. MOOSDB에서는 현재 무인잠수정의 심도 및 방향 정보와 함께 명령값을 pFuzzySOC 모듈로 전달하고, 자기구성 퍼지제어기에서는 제어판(elevator) 명령값을 계산하여 DESIRED_ELEVATOR라는 키와 함께 다시 MOOSDB로 publish한다.

2. 자기구성 퍼지제어 기반 자동심도제어기

퍼지논리 기반의 제어 기법은 전문가의 경험적 지식이나 제어 대상의 특성을 바탕으로 적절한 제어 규칙을 구축하여 제어하는 방법으로서 정확한 수학적 모델을 얻기 어려운 비선형 시스템에서 우수한 성능을 보인다[7-9]. 하지만 전문가의 제어 지식과 경험을 이용하여 정밀한 제어 규칙을 생성해야 하므로 많은 시간과 비용이 소모된다. 이러한 문제점을 해결하기 위해 1979년 Procyk와 Mamdani에 의해 자기구성 퍼지제어기가 제안되었다[10].

자기구성 퍼지제어기는 프로세스의 제어성능을 평가하여 제어기 스스로 제어규칙을 조정하는 기법으로서, 의료과학[11-13], 자동차 제조[14], 전력 기술[15] 등 많은 산업 분야에서 성공적으로 적용되고 있다. 그림 7의 구성도에서 볼 수 있듯이, 자기구성 퍼지제어기는 퍼지추론기(fuzzy inference engine)와 퍼지규칙기반(fuzzy rule base)으로 구성된 일반적인 퍼지 제어기에 스스로 제어규칙을 변경할 수 있는 자기구성단계(self organizing level)

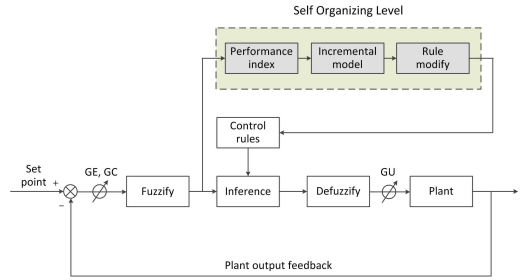


그림 7. 자기구성 퍼지제어기 시스템의 구조
Fig. 7 Structure of the SOC system

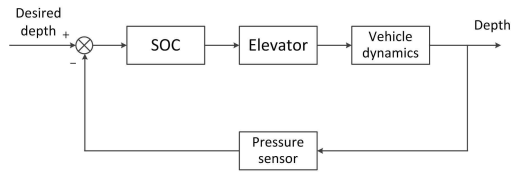


그림 8. 자동심도제어시스템의 구조
Fig. 8 Structure of auto depth control system

표 1. 자동심도제어기의 퍼지 성능평가표
Table 1. Fuzzy rule-based performance evaluation table for auto depth controller (e=error, ce=change of error)

e \ ce	PB	PM	ZO	NM	NB
PB	PB	PB	PB	PM	ZO
PM	PB	PB	PM	ZO	NM
ZO	PB	PM	ZO	NM	NB
NM	PM	ZO	NM	NB	NB
NB	ZO	NM	NB	NB	NB

가 추가된 형태를 가진다.

그림 8에 나타난 것처럼, 압력센서를 통해 측정된 무인잠수정의 실제 심도값과 목표 심도값의 오차가 자기구성 퍼지제어기의 입력으로 인가되며, 이를 기반으로 제어판 명령값이 계산된다. 본 논문에서는 표 1과 같이 Procyk와 Mamdani[10]가 제시한 성능평가표를 그대로 사용하였으며, 소속함수는 그림 9에 나타내었다.

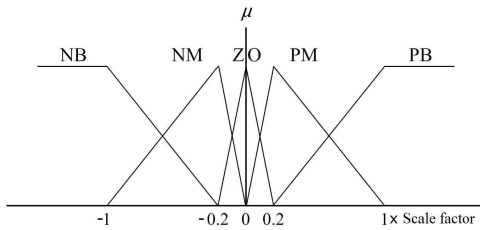


그림 9. 자동심도제어기의 소속함수
Fig. 9 Membership functions for auto depth controller

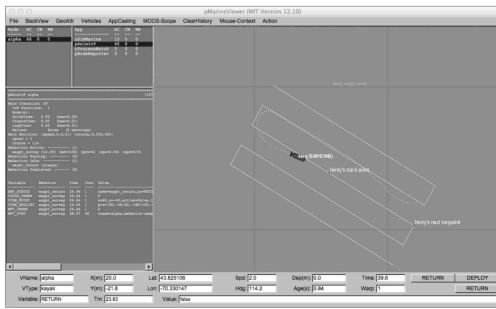


그림 10. pMarineViewer의 임무 실행 화면
Fig. 10 Screen shot of mission running in pMarineViewer

IV. 시뮬레이션 및 고찰

MOOS-IvP에는 그림 10과 같이 MOOSDB를 통해 subscribe된 무인잠수정의 위치, 방향, 속도 등에 대한 정보를 실시간으로 렌더링하여 시뮬레이션 과정을 확인할 수 있는 pMarineViewer 어플리케이션이 포함되어 있다. pMarineViewer는 FLTK/OpenGl 기반의 GUI로서 NODE_REPORT라는 키와 함께 실제 수중 환경에서의 무인잠수정의 정보를 받아 사용자에게 이차원 화면으로 제공한다. NODE_REPORT에는 GPS나 마그네틱컴퍼스, 로컬 무인잠수정의 시뮬레이터를 통해 전달된 무인잠수정의 이름과 종류, 렌더링한 시간, X-Y 좌표, 속도, 심도 등의 정보가 포함되어 있다.

본 논문에서는 MOOS-IvP 소프트웨어에서 제공하는 ‘Golf 미션’을 이용하여 자동심도제어기 모듈을 검증한다. Golf 미션은 무인잠수정이 초기 지점에서 시작하여 지그재그 형태의 8개의 경로점 (waypoint)를 따라가며 사용자가 설정한 행위 파일의 목표심도를 유지하는 미션이다. 본 논문에서는

표 2. Golf 미션에 사용된 행위 파일과 MOOS 어플리케이션

Table 2. Behaviors and MOOS applications of Golf mission

Mission	“Golf” in moos-ivp/missions/s7_golf
Behaviors	BHV_Loiter, BHV_Waypoint, BHV_PeriodicSurface, BHV_ConstantDepth
MOOS Apps	pHelmIvP, pMarineViewer, pLogger, pNodeReporter, uTimerScript, uSimMarine, pMarinePID(or pFuzzySOC)

표 3. 자기구성 퍼지제어기의 입출력 변수 및 환산계수

Table 3. Input/output variables and scaling factors of self-organizing fuzzy controller

Variable		Scaling factor
Input	Error [m]	2.0
	Change in error[m/sec]	0.5
Output	Elevator command [deg]	4.0

120초 간격으로 0m-30m-15m- 30m로 목표 심도 값을 변경하였다. 해당 미션에서 사용한 행위 파일과 MOOS 어플리케이션은 표 2와 같다.

동일한 미션에 대해서 pFuzzySOC 어플리케이션의 성능과 표준 MOOS-IvP에 내장된 PID 제어기의 성능을 비교하였다. 시뮬레이션에서 사용한 PID 제어기의 계수값은 비례항 $K_p = 0.5$, 적분항 $K_i = 1.0$, 미분항 $K_d = 0.0$ 이며, 자기구성 퍼지제어기의 입출력 변수와 환산계수는 표 3에 제시하였다.

그림 11에서 보는 것처럼, 자기구성 퍼지 기반 자동심도제어기를 pFuzzySOC 어플리케이션으로 구현하여 하나의 MOOS 모듈로 구성한 경우와, 기존의 PID 제어기 모듈을 이용한 경우 모두 원하는 심도값을 잘 추종함을 확인할 수 있다. 다만 pFuzzySOC 어플리케이션의 경우 첫 번째 목표심

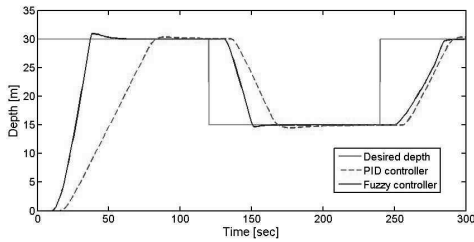


그림 11. 각 제어기에 대한 시뮬레이션 결과

Fig. 11 Simulation results of two controllers

도인 30m를 따라가는 과정에서 상대적으로 큰 오버슈트가 발생하였는데, 이는 퍼지제어 규칙의 수정이 완료되지 않은 것에 기인한다. 이후의 목표심도에 대해서는 자기구성을 통해 제어규칙을 수정하므로서 오버슈트가 점차 줄어드는 것을 볼 수 있다.

V. 결론

최근 미국을 중심으로 무인잠수정을 위한 소프트웨어 아키텍처인 MOOS-IvP를 적용한 자율 무인잠수정이 활발히 개발되고 있다. 본 논문에서는 MOOS-IvP에 대한 국내 연구자의 관심을 증대시키고자, 기 개발된 자기구성 퍼지 기반 자동심도제어기의 설계 및 시뮬레이션 과정을 사례연구로 수행하여 MOOS-IvP의 모듈 확장성과 재사용성을 고찰하였다. 먼저 자기구성 퍼지 기반 자동심도제어기를 설계하여 pFuzzySOC라는 하나의 MOOS 어플리케이션으로 구성하고, 이를 표준 MOOS에 포함된 PID 기반 제어기인 pMarinePID 모듈을 대체하였다. 심도제어기의 성능 검증에 위한 시뮬레이션에 필요한 무인잠수정의 운동방정식, 항법, 장애물회피, 센서/액추에이터 모듈 등은 표준 MOOS-IvP에 포함된 라이브러리 및 소스코드를 그대로 재사용하였다. 또한 실시간으로 시뮬레이션 과정을 확인할 수 있는 pMarine Viewer를 통해 자동심도제어기 모듈의 작동 상태를 손쉽게 분석하였다. 본 연구를 통해 사용자가 향후 MOOS-IvP를 이용할 경우, 직접 개발한 어플리케이션을 MOOS-IvP 아키텍처에 손쉽게 추가할 수 있으므로 소프트웨어 재사용성 향상 및 개발기간 단축 효과를 기대할 수 있음을 확인하였다.

References

- [1] X.Q. Chen, Y.Q. Chen, J.G. Chase, Mobile Robots - State of the Art in Land, Sea, and Collaborative Missions, I-Tech Education and Publishing, Austria, 2009.
- [2] Y. Song, S. Byun, B. Choi, H. Kim, J. Seo, D. Kim, "Architecture design of ASW UUV simulator based on MOOS-IvP," Proceedings of Fall Workshop on Korea Unmanned Underwater Vehicle, pp.12-15, 2012 (in Korean).
- [3] M.R. Benjamin, H. Schmidt, P.M. Newman, J.J. Leonard, "Nested autonomy for unmanned marine vehicles with MOOS-IvP," Journal of Field Robotics, Vol. 27, No. 6, pp.834-875, 2010.
- [4] M.R. Benjamin, H. Schmidt, P.M. Newman, J.J. Leonard, "An overview of MOOS-IvP and a users guide to the IvP Helm," MIT CSAIL Technical Report, 2013.
- [5] D. Lee, D. Kwak, J. Choi, "An intelligent control system design for autonomous underwater vehicle," Journal of Control, Automation and System Engineering, Vol. 3, No. 3, pp.227-237, 1997 (in Korean).
- [6] M.R. Benjamin, "The interval programming model for multi-objective decision making," Technical Report AIM-2004-021, Computer Science and Artificial Intelligence Laboratory, MIT, Cambridge, 2004.
- [7] S. Yasunobu, T. Hasegawa, "Evaluation of an automatic container crane operation system based on predictive fuzzy control," Control Theory and Advanced Technology, Vol. 2, No. 3, pp.419-432, 1986.
- [8] M.C.M. Teixeira and S. H. Zak, "Stabilizing controller design for uncertain nonlinear systems using fuzzy models," IEEE Transactions on Fuzzy Systems, Vol. 7, No. 2, pp.133-142, 1999.
- [9] T. Tagaki, M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," IEEE Transactions on Systems, Man, and Cybernetics Society, Vol. 15, No. 1,

pp.116-132, 1985.

- [10] T.J. Procyk, E.H. Mamdani, "A linguistic self-organising process controller," Automatica, Vol. 15, pp.15-30, 1979.
- [11] J. Shieh, M. Fu, S. Huang, M. Kao, "Comparison of the applicability of rule-based and self-organizing fuzzy logic controllers for sedation control of intracranial pressure pattern in a neurosurgical intensive care unit," IEEE Transaction on Biomedical Engineering, Vol. 53, No. 8, pp.1700-1705, 2006.
- [12] H. Ying, F. Lin, R.D. MacArthur, J.A. Cohn, D.C. Barth-Jones, H. Ye, L.R. Crane, "A self-learning fuzzy discrete event system for HIV/AIDS treatment regimen selection," IEEE Transactions on Systems, Man, and Cybernetics, Vol. 37, No. 4, pp.966-979, 2007.
- [13] D.A. Linkens, S.B. Hasnain, "Self-organising fuzzy logic control and application to muscle relaxant anaesthesia," IEE Proceedings-D, Vol. 138, No. 3, pp.274-284, 1991.
- [14] C. Lin, C. Hsu, "Self-learning fuzzy sliding-mode control for antilock braking systems," IEEE Transactions on Control Systems Technology, Vol. 11, No. 2, pp.273-278, 2003.
- [15] N. Khaehintung, C. Kangsajian, P. Sirisuk, and A. Kunakorn, "Grid-connected photovoltaic system with maximum power point tracking using self-organizing fuzzy logic controller," Power Electronics and Drives Systems, pp.517-521, 2005.

저 자 소 개

김 지 연



2010년, 경북대학교 전자
전기컴퓨터학부 학사.
2012년, 경북대학교 대학원
전자공학부 석사.
현재, 경북대학교 대학원
전자공학부 박사과정

관심분야: 고장대처, 군집제어, 무인잠수정
Email: jiykim@knu.ac.kr

이 동 익



1987년, 경북대학교 전자
공학과(이학사).
1990년, 경북대학교 전자
공학과(공학석사).
1990년~1997년, 국방과학
연구소 연구원

2002년, 영국 셰필드대학교 자동제어시스템
공학과(공학박사).
2002년 1월~2005년 3월, 영국 DRTS Ltd
공동설립 및 CTO.
2005년~현재, 경북대학교 IT대학 전자공학부
부교수.
관심분야: 고장진단, 고장대처, 시스템 안전,
산업용 네트워크, 풍력발전기, 지능형 자동차,
무인잠수정
Email: dilee@ee.knu.ac.kr