

# 상업용 자율무인잠수정의 안전운행을 위한 모듈러 구조 제어기 개발

지상훈\* · 정연수\* · 김지민\* · 이범희\* · 우종식\*\* · 주영석\*\*

\*서울대학교 전기컴퓨터공학부

\*\*DSME유택(주)

## Development of a Modular Structured Controller for a Commercial AUV

Sang-Hoon Ji\*, Yeon-Soo Jung\*, Ji-Min Kim\*, Beom-Hee Lee\*, Jong-Sik Woo\*\*, AND Young-Seok Joo\*\*

\*School of Electrical Engineering and Computer Sciences, Seoul National University, Seoul, Korea

\*\*DSME UTech Co. Ltd., Daejeon, Korea

**KEY WORDS:** Autonomous underwater vehicle 자율무인잠수정, Modular structure 모듈러 구조, Fault tolerance 오류 강인성

**Abstract:** This paper deals with the ways how the AUV is composed of sub-modules to be fault-tolerant during the mission. The emergency situations can be classified into several parts according to the zones where the situations take place - sensor zone, actuator zone, interpreter zone, and communication zone. This paper explains how each element of emergency situation is detected, contained and treated, and as a result, introduce the modular structured AUV controller for the autonomous cruising.

### 1. 서론

해양은 인류의 새로운 활동 공간 및 고갈되어 가고 있는 육지 자원을 대체할 새로운 자원의 보고로서 그 중요성이 더해가고 있다. 일반적으로 해양에서의 작업은 인간의 잠수 한계를 넘어서거나 인간이 직접 할 수 없는 작업들을 필요로 한다. 이러한 해양에서의 작업 과정에서 자율무인잠수정(Autonomous Underwater Vehicle, 이하 AUV)가 작업 전용기에 비해 유리한 점은 임무 수행의 변경에 대한 유연성과 행동의 자유로움이다. 그러나 AUV는 일단 운용에 들어가면 운용자와 분리된 상태에서 독립적으로 임무를 수행하게 되므로 AUV 설계 시 가장 우선적으로 고려할 점이 자체적으로 위험 상황 및 손상 정도를 판단 분석하여 이에 능동적으로 대처하며 임무를 수행하는 기능을 갖게 하는 것이다. 이를 위하여 AUV는 동작 환경으로부터 발생하는 외부 위험과 시스템 손상으로부터 발생하는 내부 위험 상황의 극복에 필요한 복구 프로그램을 능동적으로 호출할 수 있어야 한다. 그러나 현실적으로는 AUV의 복잡한 제어기 구조로 인하여 임무 수행 도중에 소프트웨어의 구조적인 변경이 용이하지 않다.

본 논문에서는 AUV의 현실적인 문제를 해결하기 위해 소프트웨어의 동적 재결합이 가능한 모듈러 AUV를 제시한다. 이를 위하여 AUV의 임무와 구성 요소에 따라 AUV 제어기의 구성 모듈을 정의한다. 그리고 위험 요인들을 원인별, 결과별 특성에 따라 분류하고 각 요인에 대한 판단 방법 및 모듈 재구성 방법에 대하여 제시하였다. 본 연구에서 제시된 모듈러 AUV는

시스템의 제어기 구조가 간단하여 시스템의 유지 및 업그레이드가 쉽고, 자동화된 결합 시스템을 가지고 있어서 위험 상황에서의 능동적인 대처가 가능하다.

### 2. AUV 제어기

#### 2.1 제어기 하드웨어

본 논문에서는 DSME(주)에서 제작한 6,000m급 AUV인 옥포 6000을 바탕으로 AUV 제어기의 테스트베드를 개발하였다. 그림 1에서와 같이 본 논문의 AUV는 임무 해석 및 동작 계획을 담당하는 AUV 제어 PC(펜티엄4 3.2GHz), 추진기의 위치 제어를 담당하는 모션 제어기 및 모터 드라이버, 주변 기기(소나 센서 모듈, 온도/염도 센서 모듈, 자이로 및 방위 센서 모듈 포함)를 대신하는 주변 기기 제어용 PC와 누수 센서 및 추진기 과부하 신호의 입력을 담당하는 비트 제어기로 구성되었다. 그리고 선상의 오퍼레이터용 PC로서 GUI용 PC를 설치하였다.

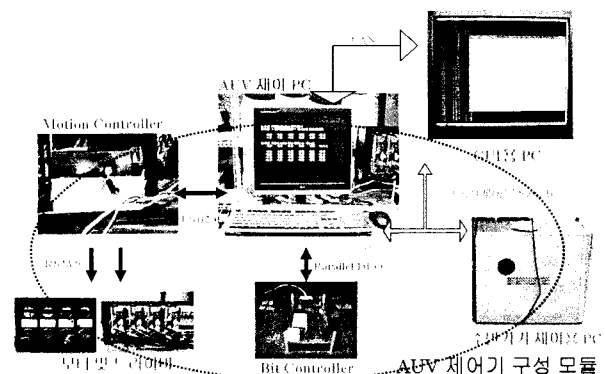


Fig. 1 Hardware of AUV Controller

교신저자 지상훈: 서울시 관악구 신림동 산56-1번지

02-880-6485 robot91@snu.ac.kr

GUI용 PC와 AUV 제어 PC는 TCP/IP 기반 통신을 통하여 임무 및 원격 명령의 전달과 AUV 상태 정보의 전달이 이루어지고 있다. AUV제어 PC와 모션 제어기는 USB2.0을 통하여 제어 신호 및 상태 정보의 교환이 이루어지고 있다. USB2.0은 통신 속도가 빠르고(480MHz) 값이 싸서 보편화된 통신 프로토콜이지만 CAN이 가지고 있는 데이터 전송의 실시간성이 없다. 이를 위하여 본 논문에서의 AUV 제어PC에 상용 실시간 OS인 QNX를 사용하였고 속도나 가속도에 비하여 제어 주기가 긴 위치 데이터를 지령 명령으로 사용하였다. 모션 제어기는 주CPU로서 TMS320LF3207를 사용하고 전용 모션 제어 칩인 PCL6045를 4개를 사용하여 최대 16축까지 제어가 가능하다.

## 2.2 제어기 소프트웨어

AUV 제어 소프트웨어의 복잡도 문제를 해결하기 위하여 실시간 시스템 설계 기법인 DARTS(Design Approach for Real Time Systems)를 이용한다. 이 기법은 시스템의 기능을 기준으로 요소를 나눈 후, 이 요소들을 요소들 간의 상관 관계, 즉 높은 긴밀도(high cohesion)과 낮은 상관관계(low coupling)을 기준으로 태스크라는 기능 단위로 구분한다. 요소 간 상관 관계로는 의존 I/O, 실시간 요구 여부, 요구 계산 량, 주기적인 실행 여부, 기능적인 동질성, 동일 이벤트에 의한 구동 여부 등이 정의되었다.

본 논문의 제어기는 그림2와 같이 3개의 계층으로 구성되었다. Manager 계층은 사용자와의 인터페이스(GUI용 PC의 프로그램과의 네트워크 통신 및 터미널을 통한 상태 Display)와 시스템 변수 관리 및 긴급 상황 처리 일을 담당하는 모듈들로 구성된다. Executor 계층은 미션 수행과 관련된 모듈들로 구성되어 있다. 특히 Real Time Module(이하 RTM)들은 기능 모듈의 상관 관계에 따라 표1과 같이 구분되었고, 이들은 자체 타이머에 의하여 구동된다. 미션 해석 모듈과 미션 프로그램의 생성, 임무 시작 및 종료, 그리고 소멸은 시스템 관리 모듈에 의하여 결정된다. Muscular 계층은 추진기나 센서와 같은 실제 하드웨어들로 이루어진 계층이다.

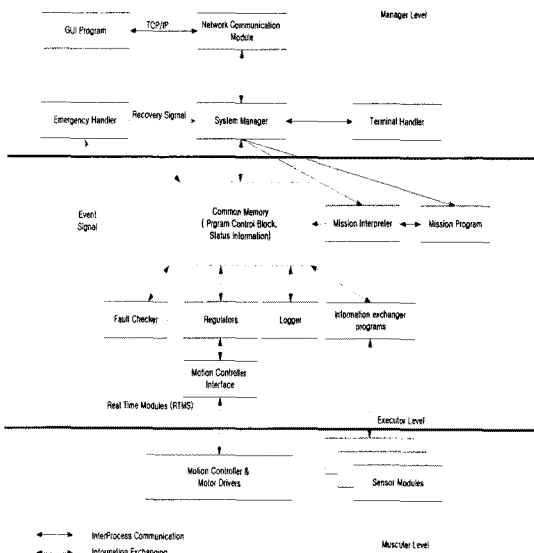


Fig. 1 Software Architecture of AUV Controller

Table 1 Real time modules

구성모듈	수행임무	우선순위
OAS Handling	충돌회피용 소나 센서	15
100ms Counter	100msec 프로그램 타이머	15
Navigation Sensors Handling	속도 및 가속도 센서	14
Regulator	속도(위치) 지령 생성	14
Motors Control	모션제어기 인터페이스	13
Bits Controller	Digital I/O 인터페이스	13
Emergency System	위험 요소 감지	13
Logger	데이터 기록	12
SSS Handling	Side Scan 소나	12
Control Computer	주변 기기 인터페이스	12
Information Print	데이터 저장	11
RC Interpreter	원격통신	10

한 모듈의 오동작이 다른 프로그램 모듈로 이전되지 않도록 하기 위하여 각 모듈들을 별도의 프로그램(프로세스)로 구현하였고 각 모듈들은 모듈의 초기화 시 자신의 이름을 QNX가 지원하는 "Name Space"에 등록하도록 하였다. 이 "Name Space"는 QNX의 커널이 프로세스 간 통신 시 사용하는 통신 채널(Channerl)과 연결점(Connection)을 관리하는 방법이다. 이 방법을 사용할 경우 상위 계층의 프로그램 모듈은 하위 프로그램 모듈의 이름을 알 경우 하위 계층 프로그램 모듈이 동작하고 있는지를 확인할 수 있고, 언제든지 데이터 전송을 할 수 있다. 특히 하위 계층에 두 개 이상의 프로그램 모듈이 동일 기능을 하도록 한 경우에 이름을 동일하게 사용하고 전송 메시지 규약을 통일하도록 할 경우 상위 계층 프로그램 모듈은 프로그램의 실행 중에 언제든지 시스템의 재구성을 할 수 있다. 그리고 각 모듈 사이의 데이터 전송 수단으로 InterProcess Commnucation(이하 IPC) 외에 공유메모리(Shared Memory)를 이용하도록 하였다. 공유메모리는 데이터 영역을 공유하지 못하는 프로세스 간 대량의 데이터 전달 방식이다. 그리고 각각의 프로그램에는 실시간 프로그램 타이머를 설정해 놓았다. 각 프로그램 타이머들은 일정 주기마다 할당된 프로그램 모듈에 비동기 이벤트인 펄스 신호를 발생한다.

시스템의 초기에는 RTM중에서는 100ms Counter만 구동된다. 조작자가 임무 수행을 요구하면 시스템 관리 모듈이 현재 구동 중인 100ms Counter RTM을 중단한 후 임무 수행에 필요한 모든 RTM들(GUI를 통하여 사용 여부가 설정된)을 순차적으로 생성하고, 마지막으로 미션 프로그램과 미션 핸들러 프로그램을 생성한다. 이 프로그램들을 생성시킨 후에 시스템 관리 프로그램은 서비스 대기 상태로 존재한다. 미션은 미션 프로그램이 수행할 임무를 미션 핸들러로 전송한다. 미션 핸들러는 수행할 임무가 도착하면 이를 해석한 후, 공유메모리(Program Control Block, 이하 PCB)에 수행 임무의 종류와 목표점들을 갱신하고, 임무 수행이 완료되었을 경우 미션 프로그램에 임무 수행 결과를 통지한다. 임무 완료를 통보 받은 미션 프로그램은 다음 임무의 수행을 재차 요구한다. 그림2와 같이 RTM의 위험 요소 발견 모듈(Fault Checker)이 위험을 발견하면 위험 처리자

(Emergency Handler)에게 위험 요소에 대한 처리를 통보하고 위험의 감내 한계를 확인한 후 임무의 지속 여부를 결정한다.

Fig. 2 Mechanism of Emergency Management

### 2.3 수행 임무

조작자는 AUV가 수행할 임무를 표2에서 정의된 임무들을 가지고 작성한다. 사용자가 임무 파일을 작성한 후(C문법 이용)한 후 GCC 컴파일러를 이용하여 실행 이미지를 만든다. 만들어진 프로그램들을 AUV를 잠수시키기 전에 미리 AUV 제어 PC의 특정 디렉토리들에 저장한다. 각 임무 프로그램들의 이름을 Mission\_01, Mission\_02"와 같이 설정하여, 원격에서 미션 프로그램의 번호 지령에 따라 현재 수행하던 프로그램을 중단하고 새로운 임무를 수행할 수 있게 하였다.

Table 2 Missions

임무	설명
EOM	임무 프로그램 종료
DIVING	특정 깊이 또는 고도까지 잠수
AUTO_NAVIGATION	특정 고도 또는 깊이를 유지한 상태에서의 이동 요구
MAN_FORCE	특정 방향 이동 및 회전 요구
VIDEO	Video On/Off 요구
SSS	Side Scan Sonar On/Off 요구
REQ_SONAR	소나 데이터 요구
REQ_NAVIGATION	항주 상태 정보(위치, 깊이, 고도, 속도, 가속도), 환경 정보(온도, 염도)요구
LOG	특정 데이터 저장 요구

## 3. 위험 요소

### 3.1 위험 요소의 분류

AUV의 임무 수행 중 발생할 수 있는 위험 요소는 크게 내부적 요소와 외부적 요소로 분류된다. 위험 요소들 중 주요 내부적 요소는 다음과 같다.

- ① 전력 공급 이상
- ② 운용 장비 및 센서 이상
- ③ 누수
- ④ 추진기 과부하 및 전력 공급 이상
- ⑤ 프로그램의 오작동
- ⑥ 통신 오류

위험 요소들 중 주요 외부적 요소는 다음과 같다.

- ① 조류에 의한 추진기 과부하
- ② 급격한 지형 변화
- ③ 염도 변화에 따른 부력 변화

### 3.2 위험 요소의 판단

위험 요소를 판단 방법에 따라 다음과 같이 구분하였고, 별도의 모듈로 구성하였다.

#### ① Digital I/O 및 통신에 의한 판단

누수 여부 및 추진기의 과부하 여부와 Digital I/O로 확인 가능한 오류와 주변 기기와의 통신 오류를 함께 측정한다.

#### ② A/D 변환 데이터에 의한 판단

전력 공급 이상은 A/D 변환 데이터의 +24V, +12V, +5V 및 접지 전압 등으로 이상여부를 확인한다. 각 전압마다 경고 전압과 이상 전압을 각각 설정하여 경고 전압의 범위를 초과한 경우에는 시스템 이상 발생을 기록만 하고 임무 수행을 지속하게 하고 설정된 이상 전압의 범위를 임계 시간 동안 초과할 경우 수행 중인 임무를 포기하고 AUV는 부상하도록 한다.

#### ③ 센서 데이터에 의한 판단

동작 깊이가 임계 동작 깊이를 초과할 경우, 잠수 시간 초과 및 소나 센서 데이터의 임계치 초과한 깊이 센서, 소나 센서 및 타이머 데이터를 기준으로 판단한다.

#### ④ 환경 및 항주 상태의 추정에 의한 판단

추종 오차(깊이, 방위각, 속도)는 깊이 센서 및 하방 소나 센서를 이용한 깊이 또는 고도 데이터와 자이로 센서 및 속도 센서를 이용한 항주 상태를 기준으로 판단한다. 그리고 급격한 지형 변화는 계측된 소나 센서 데이터들과 이전 데이터들을 기준으로 추정되었던 동작 환경의 비교를 통하여 판단한다.

#### ⑤ Watch-dog 타이머

주요 센서 인터페이스 모듈과 동작 제어 모듈들에는 각자의 Watch-dog 타이머가 할당되어 있다. 각 모듈들은 자신들에게 주어진 프로그램들은 Watch-dog 타이머가 fired 되지 않도록 타이머의 값들을 재설정하도록 되어있다. 따라서 특정 모듈이 오작동하거나 특정 시간동안 실행되지 못하였을 경우 해당 모듈을 복구한 후 임무 수행을 재개한다.

### 3.3 위험 요소의 처리

이상과 같이 AUV가 위험을 발견하면 우선 각 위험 요소가 자신이 감내할 수 있는지에 대하여 판단하여야 한다. 이의 판단 기준은 크게 위험 요소가 AUV의 안전에 심각한 위협을 초래할 수 있는가와 임무 수행이 가능한가에 대한 여부이다.

AUV의 안전에 심각한 영향을 주는 요소로는 다음과 누수와 압력 용기의 파괴가 있다. 특히 누수의 경우에는 AUV의 누수량의 총합이 부상용 ballast의 무게보다 클 경우 AUV가 부상할 수 없게 된다. 임무 수행 불능 요인으로는 대체 장비가 없는 계측 장비(예를 들어 항해를 위한 소나 등)의 고장과 추진기 고장, 항로 유지 불능과 배터리 용량 부족 등이 있다. 위의 두 가지 위험 요소 그룹에 속한 요인들이 발생할 경우에는 주어진 임무를 포기하여야 한다. 이 외에 대체 장비가 있는 계측 장비의 고장

이나 부가 장치의 고장이 발생할 경우에는 특정 임무를 포기하거나 새로운 임무를 지령 받도록 한다. 그림 3에서와 같이 위험 요소에 대한 처리 과정은 위험 요소의 탐지 및 감내 한계 판단, 임무 지속 여부의 판단, 조치(임무의 포기 또 대체 임무 수행)으로 구성된다.

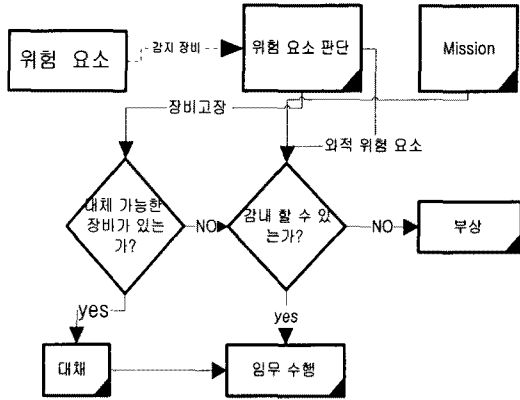


Fig. 3 Procedure of Emergency Management during AUV's Operation

### 3.3 모듈의 재구성

위험 처리자가 위험 요소들을 처리하도록 하는 모듈 재구성 과정은 그림 4와 같다. Fault Checker는 각 RTM들이 설정한 위험 발생 신호(NTU 이벤트)와 감내 한계를 고려하여 실행 중이던 임무 프로그램(mph)의 수행을 정지하고 위험 처리자를 생성한다. 생성된 위험 처리자는 NTU 이벤트에 따라 위험 처리 과정을 수행하고, 처리 완료 이벤트를 설정(NTU←NULL)한다. 위험 요소에 대한 처리가 완료된 후, 임무가 지속될 수 없는지를 위험 요소 판단 모듈이 판단하여 임무 종료 요구를 시스템 관리자에게 요구하던가 또는 저장되었던 임무 프로그램의 PCB를 환원하여 임무 수행을 재개시킨다.

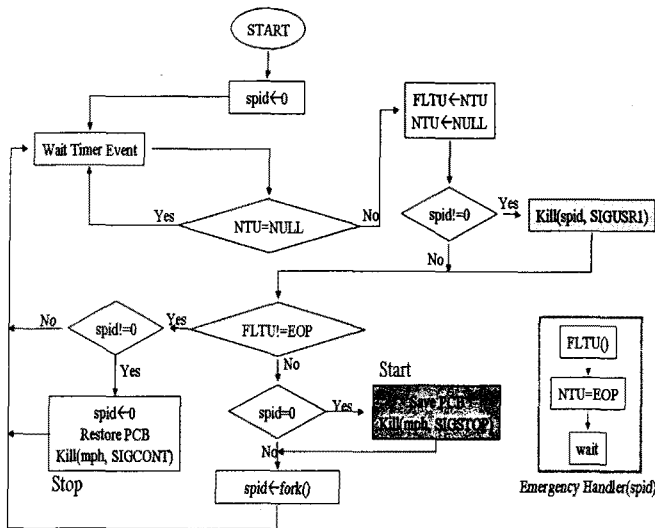


Fig. 4 Reconfiguration of RTM's for treating Emergency

시스템 관리자가 임무 종료 요구를 받았을 경우 동작 중이던 모든 RTM들과 미션 프로그램 및 미션 핸들러를 종료시키고 부상(浮上)한다. 이 후 사용자의 요구가 발생되기 전까지 100ms Counter RTM만 동작한다.

정상적으로 미션이 완료되면(미션 프로그램의 EOM 명령이 수행 되었을 경우거나 또는 미션 프로그램이 종료된 경우)에는 미션 핸들러 프로그램은 시스템 관리자 모듈에게 미션 종료 상황을 통보한다. 이 이벤트에 대하여 시스템 관리자 모듈은 수행되고 있던 모든 RTM들을 종료시키고 100ms Counter RTM만 동작하고 사용자의 요구가 발생될 때까지 대기한다.

## 4. 결 론

본 논문에서는 운항 중에 발생하는 위험요소들을 자체적으로 판단 분석하여 이에 능동적으로 대처하며 안전하게 임무를 수행하는 소프트웨어의 동적 재결합이 가능한 모듈러 AUV를 제시하였다. 이를 위하여 AUV의 임무와 제어기의 구성 모듈을 정의하였고 위험 요인들을 원인 및 결과별 특성에 따라 분류하고 각 요인에 대한 판단 방법 및 모듈 재구성 방법에 대하여 제시하였다. 추후 다양한 회사의 상업용 하드웨어들을 이용하여 AUV의 업그레이드 및 손쉬운 재구성이 가능한 개방형 구조의 AUV에 대한 연구가 필요하다.

## 참고 문헌

- 우종식 (1999). "무인잠수정(AUV) OKPO-6000 개발", 대우조선해양 연구과제보고서
- Juan Carlos Fraile, Christiaan J.J. Paredis, and Cheng Hua Wang (1999), "Agent-Based Planning and Control of a Multi-Manipulator Assembly System", Proc. of the 1999 IEEE International Conference on Robotics & Automation, pp 1219-1225
- H. Gomma (1984). "A Software Design Method for Real-Time Systems", Communication of the ACM, Vol 27, No 9, pp 938-949.
- D.B.Marco, A.J.Healey, and R.B.McGHee (1996). "Autonomous Underwater Vehicles: Hybrid Control of Mission and Motion", Autonomous Robots 3, pp 169-186.
- Rob Krten (2001). "Getting Started with QNX Neutrino 2", PARSE Software Devices.
- Howard H. Wang and Stephen M. Rock (1996). "OTTER: The Design and Development of an Intelligent Underwater Robot", Autonomous Robots 3, pp 297-320.
- J.Z. Pan and R.V. Patel (2004). "Reconfigurable distributed real-time processing for multi-robot control: Design, implementation and experimentation", Robotica, Vol 22, pp 661-679.