



J. Korean Soc. Aeronaut. Space Sci. 49(6), 505-513(2021)

DOI:https://doi.org/10.5139/JKSAS.2021.49.6.505

ISSN 1225-1348(print), 2287-6871(online)

## 무인기 자율 임무관리 소프트웨어 설계 및 검증 기법

장우혁<sup>1</sup>, 이승규<sup>2</sup>, 김윤근<sup>3</sup>, 오태근<sup>4</sup>

### Autonomous Mission Management Software Design and Verification Technique for Unmanned Aerial Vehicles

Woohyuk Chang<sup>1</sup>, Seung-Gyu Lee<sup>2</sup>, Yun-Geun Kim<sup>3</sup> and Taegeun Oh<sup>4</sup>

Aerospace Technology Research Institute, Agency for Defense Development

#### ABSTRACT

We propose an autonomous mission management software design and verification technique for unmanned aerial vehicles to autonomously mitigate dynamic situation changes occurred in the inside and outside of an aircraft in compliance with the mitigation priority order. The proposed autonomous mission management software is designed in a modular architecture that consists of concurrently executing multiple threads. To verify it, we suggest three verification steps: 1) software integration by checking the expected request/response messages between the threads for all possible dynamic situation changes; 2) integration test to verify the software functionality; 3) performance test to verify the quantitative software performance. Especially, the software integration test environment is built and utilized to carry out the integration and performance tests.

#### 초 록

본 논문에서는 비행 중 비행체 내·외부에서 발생하는 동적상황 변화에 대해 대응 우선순위에 따라 무인기가 자율적으로 대처할 수 있는 무인기 자율 임무관리 소프트웨어의 설계 및 검증 기법을 제안한다. 제안된 무인기 자율 임무관리 소프트웨어는 다중 스레드 기반의 동시 수행 가능한 모듈화 구조로 하위 소프트웨어들을 설계하였으며, 이를 검증하기 위해 1) 모든 동적상황 변화에 대한 스레드들 간의 요청/응답 메시지를 확인하는 소프트웨어 통합 기법과 2) 소프트웨어 통합검증환경을 활용하여 소프트웨어의 기능 검증을 수행하는 통합시험 및 3) 정량적 성능 검증을 수행하는 성능시험 기법을 제시한다. 특히, 통합시험과 성능시험을 수행하기 위해 소프트웨어 통합검증환경을 제작하여 활용한다.

**Key Words** : Unmanned Aerial Vehicle(무인기), Autonomous Mission Management(자율 임무관리), Software Design(소프트웨어 설계), Software Verification(소프트웨어 검증)

#### 1. 서 론

최근 국방 분야뿐만 아니라 민수 분야에서도 무인

기를 활용한 응용 영역이 급속도로 확장됨에 따라 보다 높은 자율화 수준의 무인기 개발에 대한 수요가 증가하고 있다. 기존의 무인기들이 지상 조종사의

† Received : February 10, 2021 Revised : March 29, 2021 Accepted : March 30, 2021

<sup>1</sup> Senior Researcher, <sup>2</sup> 1<sup>st</sup> Lieutenant, <sup>3</sup> Senior Researcher, <sup>4</sup> Senior Researcher

<sup>4</sup> Corresponding author, E-mail : tgoh84@gmail.com, ORCID 0000-0002-4938-4717

© 2021 The Korean Society for Aeronautical and Space Sciences

수동 조종 혹은 사전에 계획된 임무계획에 따라 자동 비행하는 수준이라면, 최근의 무인기들은 지상에서 사전에 인지하지 못한 상황이 비행 중 비행체 내·외부에서 동적으로 발생하더라도 자율적으로 이에 대처하며 주어진 임무를 계속 수행할 수 있도록 보다 높은 자율화 수준이 요구되고 있다. 본 논문에서는 무인기가 임무수행 중 발생하는 동적상황 변화에 자율적으로 대처 가능한 자율 임무관리 소프트웨어의 설계 기법과 그 검증 방법론을 제안한다.

일반적으로 지상·해양·항공 무인 로봇의 자율 임무관리 소프트웨어를 체계적으로 설계하기 위해 지금까지 다양한 소프트웨어 구조가 제안되었다. 국방 분야를 먼저 살펴보면 미 해군에서는 MIT에서 개발한 ADEPT (All-Domain Execution and Planning Technology)[1]을 기반으로 확장 가능한 모듈화 구조의 소프트웨어 프레임워크인 MOAA (Maritime Open Autonomy Architecture)를 Draper 연구소와 함께 개발하였다[2]. 미 공군에서는 공군 연구소(Air Force Research Laboratory)에서 자체개발 통신 프로토콜인 LMCP (Lightweight Message Construction Protocol)을 기반으로 확장 가능한 모듈화 구조의 UxAS (Unmanned Systems Autonomy Services) 프레임워크를 개발하였으며[3], 개방형 버전인 OpenUxAS를 공개하였다[4]. 미 육군에서는 개방형 소프트웨어 프레임워크인 ROS (Robot Operating System)를 기반으로 국방용 ROS-M 프레임워크를 개발하였다[5]. 민수 분야에서는 [6]에서 1980년대 후반 카네기 멜론 대학교에서 개발한 지능형 에이전트(Intelligent Agent)인 Soar[7]를 기반으로 무인기 자율 임무관리 소프트웨어 구조를 설계하였으며, [8]에서는 임무 형식화를 위한 도구로서 Petri nets[9]을 도입하여 임무를 Petri nets으로 기술하면 무인기가 동적상황 변화에 따라 이를 자율적으로 해석하여 임무 수행이 지속 가능한 소프트웨어 구조를 설계하였다.

앞서 소개한 자율 임무관리 소프트웨어 구조들은 개방형 소프트웨어 프레임워크를 기반으로 매우 일반적인 지상·해양·항공 무인 로봇의 임무에 대해 적용 가능하도록 설계되었다. 하지만 복잡한 지형 상에서 운용되는 지상 로봇이나 임무수행 시간이 매우 긴 해양 로봇의 임무 환경과는 다르게 무인기는 3D 공간 상에서 빠르게 이동하며 사전에 인지하지 못한 동적상황에 즉각적으로 실시간 대처하는 기능이 보다 중요하므로, 무인기의 임무 환경에 특화된 기능은 심층 구현할 필요가 있다. 또한 무엇보다도 해외에서 제작된 자율 임무관리 소프트웨어들은 국내 기준에 따른 신뢰성시험 등을 통한 소프트웨어 검증을 수행하기 어려운 측면이 있다.

본 논문에서는 무인기 임무 환경을 고려하여 동적상황 변화에 대해 대응 우선순위를 부여하여 무인기가 해당 상황발생영역에 진입하기까지 충분한 시간

여유가 있는 상황, 즉각적인 대응이 필요한 상황, 무인기에 치명적인 해가 될 수 있는 매우 위급한 상황으로 분류하여 2가지 이상의 상황이 순차적으로 동시에 발생할 경우에 대해 우선순위에 따라 자율적으로 대처할 수 있도록 무인기 자율 임무관리 소프트웨어를 설계하였다. 또한 하위 소프트웨어들을 각각 독립적인 스레드(thread)로 생성하여 동작하도록 함으로써 다중 코어나 다중 프로세스에 활용이 용이하도록 하였으며, 스레드들 간 소프트웨어 통합 및 소프트웨어의 정량적 성능 검증을 위한 단계적인 검증 방법론을 제시하였다.

마지막으로, 제안된 무인기 자율 임무관리 소프트웨어는 사전 계획된 임무계획(항로점 집합)을 따라 자동 비행하며 임무를 수행하는 기존 무인기의 비행운용프로그램(Operational Flight Program)과 연동하여 동작하므로 기존 무인기 하드웨어와 소프트웨어의 개조를 최소화할 수 있다.

## II. 본 론

### 2.1 자율 임무관리 소프트웨어 설계

#### 2.1.1 자율 임무관리 소프트웨어 구조

자율 임무관리 소프트웨어는 Fig. 1과 같이 기존의 임무 컴퓨터에 탑재된 소프트웨어인 비행운용프로그램과 연동되어 현재 비행체 상태(위치, 속도, 자세, 고장 등)와 센서(레이다 위협, 공중충돌 위협, 적란운 등) 정보 등을 비행운용프로그램으로부터 수신하고 수정된 임무계획과 BIT (Built-In-Test) 정보 등을 비행운용프로그램으로 송신한다. 여기서 BIT 정보는 자율 임무관리 소프트웨어와 비행운용프로그램이 각각 별도의 하드웨어에 탑재된 경우 자율 임무관리 소프트웨어가 탑재된 하드웨어의 BIT 결과로서 비행운용프로그램을 거쳐 지상체로 전달된다.

자율 임무관리 소프트웨어는 현재 비행체 상태 및 센서 정보를 바탕으로 임무계획 수정 및 즉각적인 반사적 대응 여부를 판단하여 필요 시 이를 수행하며, 전원종료 및 BIT 수행과 같이 임무 자율화 시스템을 관리하고 비행운용프로그램과의 데이터 통신을 수행한다. 이러한 기능들을 효과적으로 구현하기 위



Fig. 1. Inter-communication between Autonomous Mission Management and Operational Flight Program (OFF)



Fig. 2. Autonomous Mission Management software architecture

해 자율 임무관리 소프트웨어는 Fig. 2와 같이 기능별로 상황인식, 의사결정, 임무계획, 위협대응, 치명적 위협대응, 시스템관리, 외부통신관리 소프트웨어들로 모듈화하여 설계되었다.

자율 임무관리 소프트웨어를 구성하는 하위 소프트웨어들은 각각 독립적인 스레드로 생성되어 동작하며, 스레드 간의 통신은 Fig. 3과 같이 공유 메모리 상의 요청/응답 메시지를 통해서 이루어진다. 예를 들면 의사결정 소프트웨어와 임무계획 소프트웨어 간의 통신은 의사결정 소프트웨어가 공유 메모리 상에서 임무계획 요청 메시지(Mission Plan Request)를 갱신하면 이를 주기적으로 확인하는 임무계획 소프트웨어가 요청에 따라 임무계획을 수정하고 그 결과를 공유 메모리 상에 임무계획(Mission Plan) 데이터로 저장 후 임무계획 응답 메시지(Mission Plan Response)를 갱신하며, 이를 주기적으로 확인하는 의사결정 소프트웨어가 응답에 따라 임무계획 수정 완료 여부를 판단한다.

자율 임무관리 소프트웨어를 구성하는 하위 소프트웨어들은 다음과 같이 동작한다. 상황인식 소프트웨어는 비행운용프로그램으로부터 실시간 수신한 비행체 상태 및 센서 정보를 확인하여 고장 혹은 위협(레이다, 공중충돌, 적란운) 발생 메시지를 생성하고

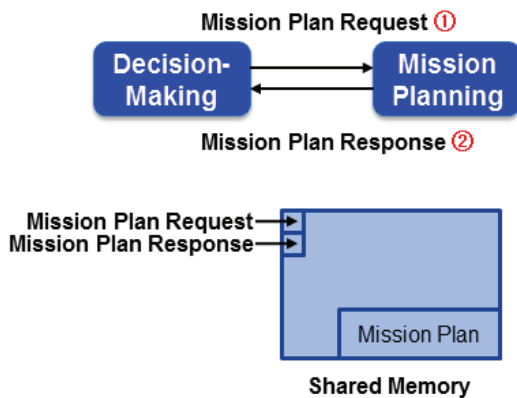


Fig. 3. Communications between Decision-Making and Mission Planning softwares

위험노출영역을 계산하여 3차원 임무 공간 데이터(Mission Space Data) 상에 반영하여 위협상황정보를 관리한다. 의사결정 소프트웨어는 고장 혹은 위협 발생, 임무수행 지연 정도에 따라 임무계획 수정 및 즉각적인 반사적 대응 여부를 판단하여 필요 시 이를 요청한다. 임무계획 소프트웨어는 임무계획 수정 요청에 따라 주어진 임무(임무점 집합)를 효율적으로 수행할 수 있는 임무 수행(임무점 방문) 순서를 계산하며 이를 위해 임무점들 간 비행경로계획을 생성하여 경로비용(비행시간, 연료소모량)을 계산한다. 위협대응 소프트웨어와 치명적 위협대응 소프트웨어는 반사적 대응 요청에 따라 위협 회피(반사적 대응)을 위한 비행경로계획을 생성한다. 시스템관리 소프트웨어는 전원종료 관리, 자율 임무관리 소프트웨어가 탑재된 하드웨어 관련 BIT 관리 등을 수행한다. 외부통신관리 소프트웨어는 비행운용프로그램과의 데이터 통신 및 로깅 데이터 전송을 수행한다.

2.1.2 상황인식 소프트웨어

상황인식 소프트웨어는 최초 3차원 임무 공간 상에 이미 알려진 초기 위협을 반영하여 위협상황정보를 생성 후 초기상황정보생성 완료 메시지를 송신한다. 이후 Fig. 4의 운용 개념에 도시된 바와 같이, 주기적으로 비행운용프로그램으로부터 비행체 내·외부 상황(고장 혹은 위협 발생)을 수신하고, 이로부터 고장 혹은 위협(레이다, 공중충돌, 적란운)이 인식되면 위협 인식 정보(Threat Recognition Information)를 의사결정 소프트웨어로 전달한다.

이후 임무계획 소프트웨어로부터 임무 공간 갱신 요청 메시지를 수신하면 신규 레이더, 적란운과 같이 동적으로 발생된 위협을 식별하고 이를 바탕으로 레이더 엄폐 공간 분석 결과에 따른 레이더위협 영역 혹은 적란운 영역 정보를 3차원 임무 공간 데이터 상의 위험노출영역에 반영하여 위협상황정보 갱신 후 임무 공간 갱신 응답 메시지를 송신한다.

2.1.3 의사결정 소프트웨어

의사결정 소프트웨어는 최초 모니터 상태에서 시작한다. 모니터 상태에서 임무수행 지연에 따른 임무계획 미추종, 위협, 고장 발생 상황 등이 발생하면 각각의 상황에 따라 임무계획 상태, 위협대응 상태, 치명적 위협대응 상태로 천이한다. Table 1은 의사결정 소프트웨어의 각 상태와 해당 상태로 천이하기 위한 동적 상황 조건과 그 예시를 나타내며, Fig. 5의 상태 천이 다이어그램에서는 상태들 간의 천이 관계를 나타낸다.

본 논문에서 다루는 위협은 임무계획 수정이 필요한 위협(Distant Threat)과 즉각적인 반사적 대응이 필요한 위협(Emergent Threat)의 2가지 위협으로 크게 구분되며, 반사적 대응이 필요한 위협은 다시 일반적 위협(Threat)과 치명적 위협(Fatal Threat)으로

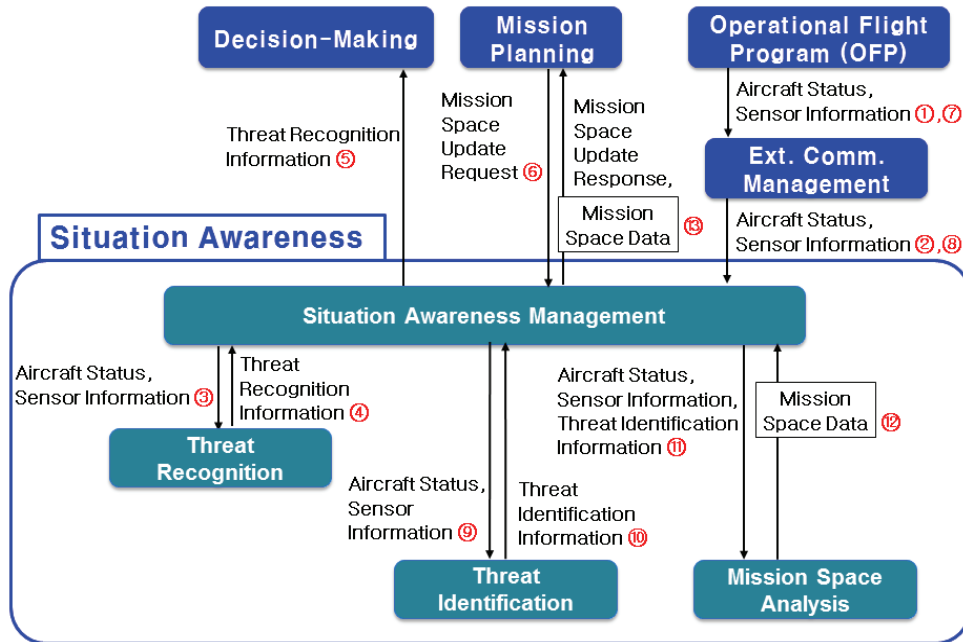


Fig. 4. Operational concept of Situation Awareness

Table 1. States according to dynamic situations in Decision-Making

States	Dynamic Situations	Examples of Dynamic Situations
Mission Planning	Distant Threat Appearance	- cumulus - radar threat far from the current location
	Mission Plan Mismatch	- mismatch between expected arrival time and actual arrival time
	Mission Equipment Failure	- malfunctions of EO/IR or SAR - data link disconnection
Threat Reaction	Threat Appearance	Threat and Fatal Threat are classified by the assigned priority to the following Emergent Threats.
Fatal Threat Reaction	Fatal Threat Appearance	- radar threat nearby the current location - threat to react collision avoidance alarm

구분된다. 임무계획 수정이 필요한 위협은 무인기가 위협노출영역으로부터 멀리 떨어져 있어서 위협노출영역에 진입하기까지 충분한 시간 여유가 있는 위협으로 3차원 임무 공간 상에 위협노출영역을 반영하여 이를 회피하도록 임무계획 수정이 가능한 위협을 가리키며, 적란운 위협 혹은 무인기가 미리 탐지한

신규 레이다 위협(레이다가 무인기 탐지 전) 등이 해당된다. 반사적 대응이 필요한 위협은 위협을 탐지한 즉시 위협을 회피하기 위한 즉각적인 기동이 필요한 위협을 가리키며, 탐지/추적 레이다 위협 혹은 공중 충돌 위협 등이 해당된다. 이때 반사적 대응이 필요한 위협은 동시 발생 시 위협 정도에 따라 대응 우선순위를 부여하여 우선순위가 높은 위협은 치명적 위협으로 정의한다.

의사결정 소프트웨어의 각 상태에서 운용 개념은 다음과 같다. 모니터 상태에서 임무계획 미추종, 임무계획 수정이 필요한 위협 발생(미리 탐지한 신규 레이다, 적란운 발생), 고장(임무장비, 주요 탑재장비 고장)과 같은 상황이 발생하면, 임무계획 상태로 천이하고 임무계획 소프트웨어로 임무계획을 요청한다.

모니터 상태에서 반사적 대응이 일반적 위협 혹은 치명적 위협 상황이 발생하면, 위협대응 상태 혹은 치명적 위협대응 상태로 각각 천이하고 위협대응 혹은 치명적 위협대응 소프트웨어로 위협대응 혹은 치명적 위협대응을 요청한다.

임무계획 상태에서는 임무계획 소프트웨어로부터 임무계획 응답이 수신되면 모니터 상태로 천이한다. 임무계획 상태에서 반사적 대응이 필요한 일반적 위협 혹은 치명적 위협 상황이 발생하면, 임무계획을 취소하고 위협대응 상태 혹은 치명적 위협대응 상태로 각각 천이한다. 이때 취소된 위협은 위협관리 큐(queue)에 저장하여 반사적 위협대응이 완료되어 위협 상황이 해소되면 임무계획 요청을 재수행할 수 있도록 한다.

위협대응 상태에서는 위협대응 소프트웨어로부터 위협대응 응답이 수신되면 모니터 상태로 천이한다.

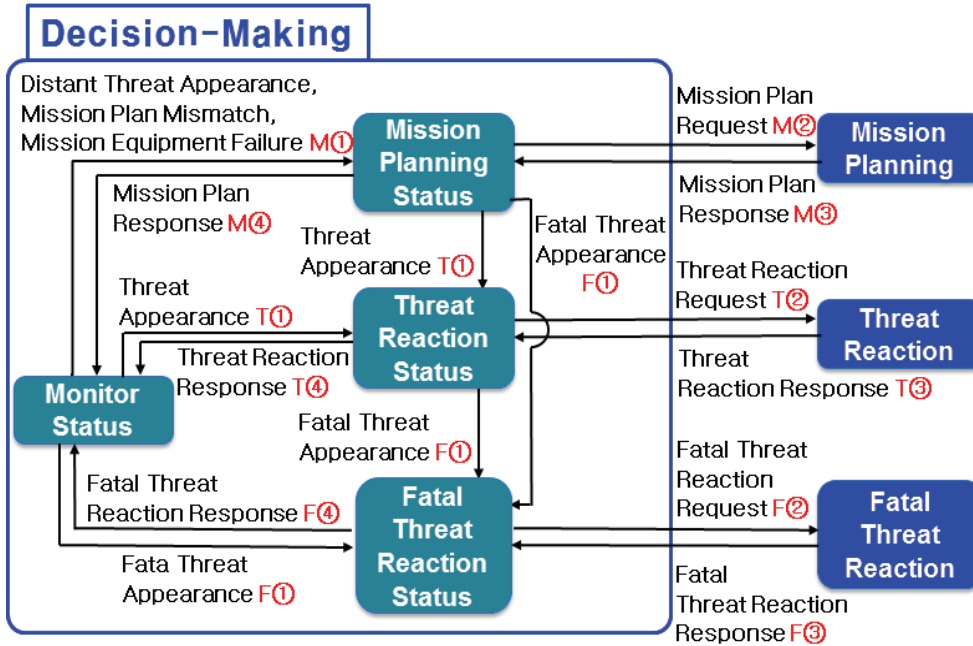


Fig. 5. State-transition diagram of Decision-Making

위협대응 상태에서 치명적 위협이 발생하면 위협대응 요청을 취소하고 치명적 위협대응 상태로 천이한다. 이때 취소된 일반적 위협은 치명적 위협대응이 완료되어 치명적 위협 상황이 해소된 후에도 상황인식 소프트웨어에서 여전히 일반적 위협으로 감지되면 위협대응 상태로 재진입하여 위협대응을 재요청하게 된다.

치명적 위협대응 상태에서는 치명적 위협대응 소프트웨어로부터 치명적 위협대응 응답이 수신되면 모니터 상태로 천이한다.

### 2.1.4 임무계획 소프트웨어

의사결정 소프트웨어가 임무계획 미추종, 임무계획

수정이 필요한 위협 발생(미리 탐지한 신규 레이다, 적란운 발생), 고장(임무장비, 주요 탑재장비 고장)과 같은 상황 발생을 인지하면 임무계획 소프트웨어로 임무계획을 요청한다.

임무계획 소프트웨어는 Fig. 6의 운용 개념에 도시된 바와 같이 임무계획 요청에 따라 동적으로 발생한 위협에 대해 3차원 임무 공간 갱신을 상황인식 소프트웨어에 요청한다. 상황인식 소프트웨어로부터 임무 공간 갱신 응답을 수신하면 경로 계산 모듈은 임무 공간 상에서 모든 임무점들 간에 위협노출영역을 회피하는 비행경로를 생성하고 이에 대한 소요 경로비용(비행시간, 연료소모량)을 계산하여 그 결과

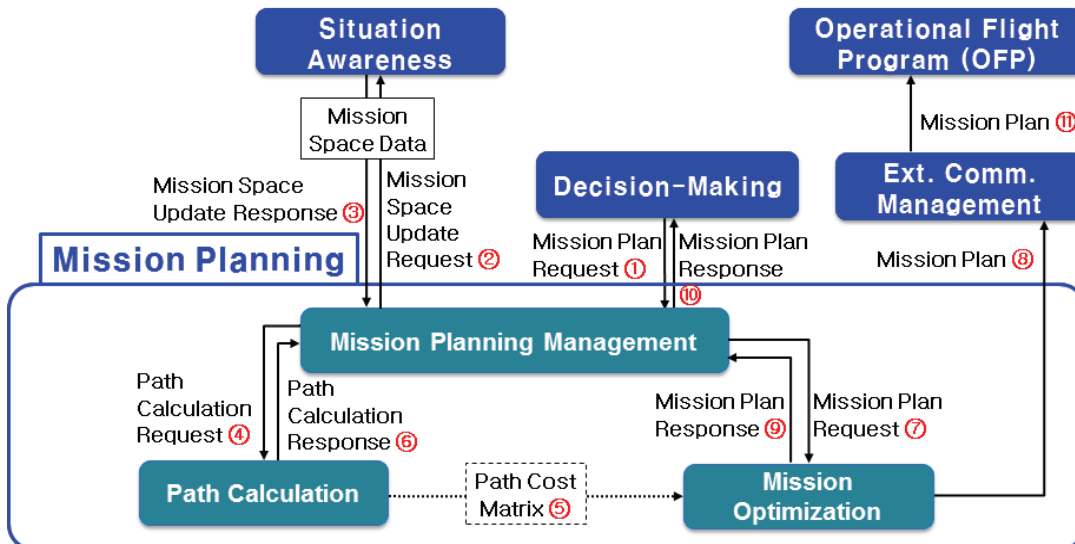


Fig. 6. Operational concept of Mission Planning

를 경로비용행렬(Path Cost Matrix) 형태로 공유 메모리 상에 저장하고 경로 계산 응답을 생성한다. 경로 계산 응답이 확인되면 임무계획 요청에 따라 임무 최적화 모듈은 공유 메모리 상의 경로비용행렬을 이용하여 주어진 임무(임무점 집합)에 대해 자율화 정책(임무시간 최소화 혹은 임무달성률 최대화 등)을 만족하는 임무 수행 순서(임무점 방문 순서)를 계산한다. 이때 혼합정수선형계획법(MILP, Mixed-Integer Linear Programming)과 같은 최적화 문제로 형식화 후 Gurobi 혹은 CPLEX 등과 같은 상용 최적화 solver[10]로 최적해를 구할 수 있으나 일반적으로 실시간 최적해 도출은 계산량이 매우 많이 요구되므로 실시간 연산이 가능한 부 최적해 도출 알고리즘이 사용된다. 대표적인 부 최적해 도출 알고리즘으로는 유전 알고리즘(genetic algorithm), 타부 검색(tabu search), 담금질 기법(simulated annealing)과 같은 메타 휴리스틱 기법이 있다.

임무계획 소프트웨어는 임무 수행 순서가 정해지면 비행 동역학을 고려한 최종 비행경로를 계산하고 이를 바탕으로 임무계획(항로점 집합)을 생성하여 외부통신관리 소프트웨어로 송신한다.

**2.1.5 (치명적) 위협 대응 소프트웨어**

의사결정 소프트웨어가 비행운용프로그램으로부터 즉각적인 반사적 대응이 필요한 위협 상황을 상황인식을 통해 수신하면 위협대응 혹은 치명적 위협대응 소프트웨어로 위협대응 혹은 치명적 위협대응을 요청한다. 위협대응 혹은 치명적 위협대응 소프트웨어는 Fig. 7의 운용 개념에 도시된 바와 같이 위협 회피를 위한 비행경로를 계산하고 이를 바탕으로 임무계획(항로점 집합)을 생성하여 외부통신관리 소프트웨어

로 송신한다. 비행운용프로그램으로 전달된 임무계획에 따라 무인기가 비행하여 위협 회피에 성공하면 비행운용프로그램은 위협 상황(위협 해소)을 갱신하여 외부통신관리 소프트웨어 및 상황인식을 통해 전달하고, 이를 위협대응 혹은 치명적 위협대응 소프트웨어가 확인하여 의사결정 소프트웨어로 위협대응 혹은 치명적 위협대응 응답을 송신함으로써 반사적 대응이 필요한 위협 상황에 대한 대응 절차가 종료된다.

**2.1.6 시스템 관리 소프트웨어**

시스템관리 소프트웨어는 초기 전원이 개시되면 자율 임무관리 소프트웨어가 탑재된 하드웨어 관련 PBIT (Power-On Built-In-Test)를 수행하고 CBIT (Continuous Built-In-Test) 초기화 및 독립적으로 개별 동작하는 스테드들을 초기화한다. 이후 시스템관리 소프트웨어는 CBIT 수행 명령을 주기적으로 펌웨어(Firmware)로 송신한다. 시스템관리 소프트웨어는 비행운용프로그램으로부터 외부통신관리 소프트웨어를 통해 IBIT (Initiated Built-In-Test) 명령이나 전원종료 명령을 수신하면 펌웨어에 IBIT 명령이나 전원종료 명령을 송신한다. 펌웨어에서는 시스템관리 소프트웨어로 CBIT 결과를 주기적으로 송신하며, 비주기적으로 IBIT 명령이나 전원종료명령을 수신하면 IBIT 결과와 전원종료준비상태를 시스템관리 소프트웨어로 송신한다. 시스템관리 소프트웨어는 펌웨어로부터 PBIT/CBIT/IBIT 결과 혹은 전원종료준비상태를 수신하면 외부통신관리 소프트웨어를 통해 비행운용프로그램으로 전달한다.

**2.1.7 외부통신관리 소프트웨어**

외부통신관리 소프트웨어는 전역변수 영역에 데이터를 읽고 쓰는 기능을 관리한다. 외부통신관리 소프

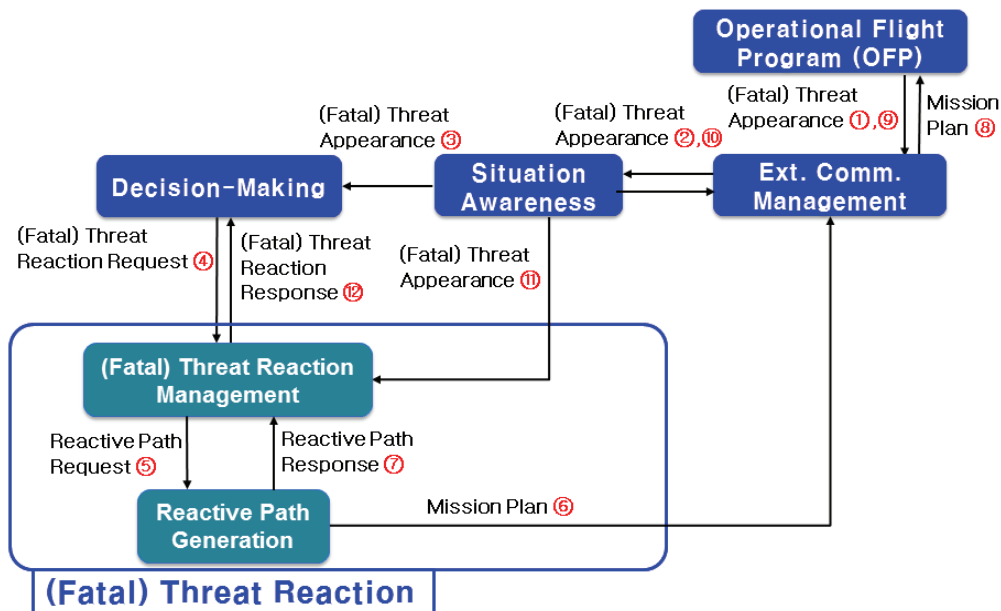


Fig. 7. Operational concept of (Fatal) Threat Reaction

트웨어는 PCI Express, Ethernet 통신 등을 통해 비행운용프로그램으로부터 현재 비행체 상태(위치, 속도, 자세, 고장 등)와 센서(레이다 위협, 공중충돌 위협, 적란운 등) 정보, IBIT 명령, 전원종료명령 등을 수신하며, 임무계획, PBIT/CBIT/IBIT 결과, 전원종료 준비상태 등을 송신한다. 또한 외부통신관리 소프트웨어는 소프트웨어 검증을 위해 로깅 데이터를 소프트웨어 통합검증환경으로 송신한다.

## 2.2 자율 임무관리 소프트웨어 검증 방안

### 2.2.1 소프트웨어 통합

자율 임무관리 소프트웨어 통합은 독립적인 개별 스테드들로 동작하는 하위 소프트웨어들인 상황인식, 의사결정, 임무계획, 위협대응, 치명적 위협대응, 시스템관리, 외부통신관리 소프트웨어들 간의 연동 검증을 통해 이루어지며, 각각의 동적상황 발생 시 관련 소프트웨어 스테드들 간의 요청/응답 메시지를 확인하여 연동을 검증한다. 대표적인 예로서, 임무계획 수정이 필요한 위협(적란운 위협 혹은 무인기가 미리 탐지한 신규 레이다 위협)이 발생한 경우 해당 위협 발생 정보를 수동으로 입력 후 관련된 요청/응답 메시지를 다음과 같이 확인한다.

- 외부통신관리 → 상황인식 : 위협 발생 정보 수동 입력
- 상황인식 → 의사결정 : Distant Threat Appearance (위협 발생) 확인
- 의사결정 → 임무계획 : Mission Plan Request 메시지 확인
- 임무계획 → 상황인식 : Mission Space Update Request 메시지 확인
- 상황인식 → 임무계획 : Mission Space Update Response 메시지 확인
- 임무계획 → 의사결정 : Mission Plan Response 메시지 확인
- 임무계획 → 외부통신관리 : Mission Plan 갱신 여부 확인

다른 대표적인 예로서 반사적 대응이 필요한 치명적 위협이 발생한 경우 해당 위협 발생 정보를 수동으로 입력 후 관련된 요청/응답 메시지를 다음과 같이 확인한다.

- 외부통신관리 → 상황인식 : 위협 발생 정보 수동 입력
- 상황인식 → 의사결정 : Fatal Threat Appearance (위협 발생) 확인
- 의사결정 → 치명적 위협대응 : Fatal Threat Reaction Request 메시지 확인
- 치명적 위협대응 → 외부통신관리 : Mission Plan 갱신 여부 확인
- 외부통신관리 → 치명적 위협대응 : Fatal Threat Appearance(위협 해소) 확인

- 치명적 위협대응 → 의사결정 : Fatal Threat Reaction Response 메시지 확인

또한 소프트웨어 통합 시 2가지 이상의 위협이 순차적으로 동시에 발생하였을 경우 위협대응 우선순위에 따라 우선순위가 낮은 위협대응은 취소하고 우선순위가 높은 위협대응 수행 여부를 확인한다. 따라서 자율 임무관리 소프트웨어 통합은 발생 가능한 모든 동적상황들을 시험항목으로 식별하고 관련된 요청/응답 메시지를 확인함으로써 요구되는 연동 검증 수행이 이루어진다.

### 2.2.2 통합시험

자율 임무관리 소프트웨어 통합시험은 소프트웨어가 탑재된 임무컴퓨터가 소프트웨어 통합검증환경과 연동된 상태에서 모의 비행시험을 통해 동적상황들에 대한 소프트웨어 출력 결과를 확인하여 소프트웨어 기능을 검증한다. 소프트웨어 통합검증환경은 Fig. 8과 같이 비행체 탑재장비 모의, 지상제 모의, 비행 모의, 임무/비행환경(기상, 위협 등) 모의, 운용통제(모의 환경 제어, 모의 시나리오 관리 등), 사전 임무계획, 임무상황전시 등으로 구성되며, 임무컴퓨터와 연동하여 사용자가 사전에 작성한 모의 시나리오에 의해 발생하는 동적상황에 따라 자율 임무관리 소프트웨어의 검증이 가능하도록 설계되었다.

대표적인 예로서 임무계획 수정이 필요한 위협(적란운 위협 혹은 무인기가 미리 탐지한 신규 레이다 위협)이 발생한 경우 해당 위협 발생 정보를 소프트웨어 통합검증환경의 모의 시나리오 상에 작성 후 모의 비행시험 시 위협을 회피하는 임무계획 생성 여부를 다음과 같이 확인한다.

- 통합검증환경 - 운용통제 : 임무 시작 후 특정 시간, 장소에 위협이 발생하도록 시나리오 작성
- 통합검증환경 - 임무/비행환경 모의, 임무상황전시 : 임무 시작 후 특정 시간, 장소에 위협 발생 확인
- 통합검증환경 - 임무상황전시 : 수정된 임무계획에 의해 위협 회피 가능여부 확인
- 통합검증환경 - 비행 모의 : 수정된 임무계획에 의한 위협 회피 비행으로 위협 해소 확인

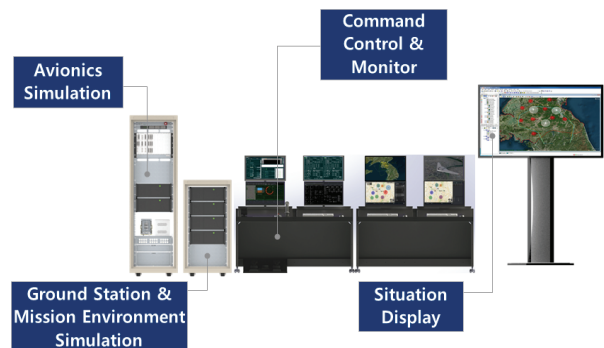


Fig. 8. Software Integration Test Environment

따라서 자율 임무관리 소프트웨어 통합시험은 발생 가능한 모든 동적상황들을 모의 시나리오 상에 작성하여 시험항목으로 식별하고 모의비행시험을 통해 해당 동적상황들에 대한 소프트웨어 출력 결과를 통합검증환경 상에서 확인함으로써 요구되는 소프트웨어 기능 검증 수행이 이루어진다.

### 2.2.3 성능시험

자율 임무관리 소프트웨어 성능시험은 소프트웨어 통합검증환경을 이용한 모의 비행시험을 통해 획득한 로깅 데이터를 확인하여 탑재 자율 임무관리 소프트웨어의 정량적 성능을 검증한다. 대표적인 예로서, 임무계획 소프트웨어가 도출한 부 최적해의 최적성 및 실시간성을 검증하는 경우 해당 위협 발생 정보를 소프트웨어 통합검증환경의 모의 시나리오 상에 반영하여 모의 비행시험을 수행한 후 로깅 데이터를 다음과 같이 확인한다.

- 로깅 데이터 : 임무 시작 후 특정 시간, 장소에 위협 발생 이벤트 확인
- 로깅 데이터 : 임무계획 소프트웨어가 Mission Plan Request 메시지를 수신한 시각과 Mission Plan Response 메시지를 송신한 시각 간의 차이를 계산하여 임무계획 소프트웨어의 연산 시간을 측정함으로써 실시간성 확인
- 로깅 데이터 : 임무계획 소프트웨어가 생성한 임무계획(Mission Plan)을 확인하여 적용된 자율화 정책(임무시간 최소화 혹은 임무달성률 최대화 등)의 목적함수(임무시간 혹은 임무달성률) 값을 계산하고, 동일한 상황에 대해 혼합정수선형계획법 등으로 형식화된 최적화 문제로부터 도출한 최적해의 목적함수 값과 비교하여 임무계획 소프트웨어가 도출한 해의 최적성 확인

따라서 자율 임무관리 소프트웨어 성능시험은 발생 가능한 모든 동적상황들을 통합검증환경의 모의 시나리오 상에 작성하여 시험항목으로 식별하고 모의비행시험을 통해 수집된 로깅 데이터로부터 탑재 소프트웨어가 도출한 값을 단독 분석하거나 성능 기준 소프트웨어의 도출 값과 비교 분석함으로써 요구되는 소프트웨어 정량적 성능 검증이 이루어진다.

### 2.3 적용 사례

본 논문에서 제안한 자율 임무관리 소프트웨어 구조를 기반으로 확장하여 실제 무인기 탑재용 임베디드 컴퓨터 환경에 자율 임무관리 소프트웨어를 구현하였다. 자율 임무관리 소프트웨어를 위한 CPU (Intel Core i7-3612QE 2.1GHz) 보드를 추가하여 기존의 무인기 탑재용 임무 컴퓨터와 PCI Express 및 Ethernet으로 연동하였다. 임무계획 수정 혹은 반사적 대응이 필요한 동적 상황은 다음과 같다.

- 임무계획 수정이 필요한 상황

- 임무계획 미추종 : 항로점 도착시각 혹은 잔여연료량 오차 기준치 이상 → 현재 시각 및 잔여연료량 기준으로 임무계획 수정
  - 적란운 발생 → 적란운 발생 영역 회피하도록 임무계획 수정
  - 신규 지대공 레이다(원거리에서 미리 탐지한 레이다) 위협 발생 → 레이다 위협 영역 회피하도록 임무계획 수정
  - 주요탑재장비(데이터링크 장비, 공중충돌방지장비) 고장 → 남은 임무점 포기 후 복귀
  - 임무장비(EO/IR, SAR) 고장 → 해당 장비 이용 촬영 임무점 포기하도록 임무계획 수정
  - 반사적 대응이 필요한 상황
    - 일반적 위협 : 이미 자항공기를 탐지한 지대공 레이다 위협 → 레이다 탐지 반대방향으로 항공기 기동
    - 치명적 위협 : 공중충돌위협 → 공중충돌방지장비의 지시에 따른 항공기 기동
- 임무계획 수정 시 다음과 같이 자율화 정책[11] 및 임무 제약사항을 적용하였다.

- 연료 소모량 우선 자율화 정책
  - 잔여 연료량으로 남은 모든 임무수행 가능
  - 목적함수 : 연료 소모량 최소화
- 임무달성률 우선 자율화 정책
  - 잔여 연료량으로 모든 임무수행 불가능
  - 목적함수 : 임무달성률(임무 가중치가 곱해진 방문 임무점 개수) 최대화
- 임무 제약사항
  - 임무점 간 임무 수행 의존성, 특정 임무 시간 내 임무점 방문, 일정시간 후 임무점 재방문, 임무점에서의 비행패턴(서쪽 → 동쪽, 동쪽 → 서쪽)에 따른 비행 옵션 등

임무 최적화 알고리즘은 유전 알고리즘[12]을 기반으로 다중 스테드를 활용하여 다수 부최적 해를 동시에 도출하여 가장 좋은 해를 구하는 방식으로 개발하였다. 개발된 자율 임무관리 소프트웨어 검증을 위해 소프트웨어 통합검증환경을 개발하고 이를 활용하여 39개 시험항목에 대한 통합시험과 12개 시험항목에 대한 성능시험을 수행하였다. 성능시험 시 로깅 데이터로부터 결과 분석을 자동화하기 위하여 Fig. 9와 같이 별도의 성능평가도구를 개발하여 시험의 효율성을 향상시켰다.

성능시험 시 50개의 가상 임무 시나리오를 생성하여 임무계획 수정이 필요한 상황을 총 299회 발생시켜 유전 알고리즘 기반의 임무 최적화 알고리즘이 도출한 부 최적해와 혼합정수계획법으로 형식화된 임무 최적화 문제에 대해 Gurobi로 도출한 최적해를 비교하여 Table 2와 시험 결과를 획득하였으며, 약 92.0%의 확률로 개발된 알고리즘이 최적해를 도출함을 알 수 있다.





Fig. 9. Software Performance Test Evaluation Tool

Table 2. Performance test results for the developed mission optimization algorithm

Policy	No. of replanning	Computation time (seconds)	Probability that the solution is optimal
Minimize the fuel consumption	107	Average : 0.33 Max : 0.46	99.1%
Maximize the mission achievement	192	Average : 1.72 Max : 3.17	88.0%
Total	299	Average : 1.23	92.0%

### III. 결 론

본 논문에서는 비행 중 비행체 내·외부에서 발생하는 동적상황 변화에 대해 대응 우선순위에 따라 무인기가 자율적으로 대처할 수 있도록 무인기 자율 임무관리 소프트웨어를 다중 스레드 기반의 동시 수행 가능한 모듈화 구조로 설계하였으며, 이를 검증하기 위한 단계적인 검증 방법론을 제시하였다. 제안한 무인기 자율 임무관리 소프트웨어 설계 및 검증 기법은 단일 무인기에 대해 서술하고 있으나 다수 무인기로 확장 가능하며, 이때 임무계획 소프트웨어에는 중앙집중형 혹은 분산형 임무할당 알고리즘이 추가 구현되어야 하며, 소프트웨어 통합 검증 시 다수 무인기를 고려한 보다 다양한 동적상황(일부 무인기 소실, 무인기간 통신 두절 등)이 고려되어야 한다. 다수 무인기 대상의 무인기 자율 임무관리 소프트웨어에 대한 보다 심층적인 연구는 후속 연구로서 추후 수행될 필요가 있다.

### References

1) Ricard, M. and Kolitz, S., "The ADEPT

Framework for Intelligent Autonomy," VKI Lecture Series on Intelligent Systems for Aeronautics, May 2002.

2) Gonzales, D. and Harting, S., *Designing Unmanned Systems with Greater Autonomy: Using a Federated, Partially Open Systems Architecture Approach*, RAND Corporation, Santa Monica, 2014.

3) Rasmussen, S., Kingston, D. and Humphrey, L., "A Brief Introduction to Unmanned Systems Autonomy Services (UxAS)," *Proceeding of International Conference on Unmanned Aircraft Systems*, June 2018, pp. 257~268.

4) English, J. and Wilhelm, J. P., "Collision Avoidance in OpenUxAS," *Proceeding of AIAA SciTech Forum*, January 2020, pp. 1~9.

5) Towler, J. and Bries, M., "ROS-Military: Progress and Promise," *Proceeding of NDIA Ground Vehicle Systems Engineering and Technology Symposium*, August 2018, pp. 1~10.

6) Gunetti, P., Dodd, T. and Thompson, H., "A Software Architecture for Autonomous UAV Mission Management and Control," *Proceeding of AIAA Infotech*, April 2010, pp. 1~11.

7) Laird, J. E., *The Soar Cognitive Architecture*, The MIT Press, Cambridge, 2012.

8) Barbier, M. and Chantry, E., "Autonomous Mission Management for Unmanned Aerial Vehicles," *Aerospace Science and Technology*, Vol. 8, No. 4, June 2004, pp. 359~368.

9) Murata, T., "Petri Nets: Properties, Analysis and Applications," *Proceedings of IEEE*, Vol. 77, No. 4, April 1989, pp. 541~580.

10) Hvattum, L. M., Løkketangen, A. and Glover, F., "Comparisons of Commercial MIP Solvers and an Adaptive Memory (Tabu Search) Procedure for a Class of 0 - 1 Integer Programming Problems," *Algorithmic Operations Research*, Vol. 7, No. 1, 2012, pp. 13~20.

11) Chang, W., Lee, D. H. and Kim, S.-H., "A Unified Framework for UAV Mission Planning," *Proceeding of The Korean Society for Aeronautical and Space Sciences Fall Conference*, November 2019, pp. 667~668.

12) Park, J.-H., Min, C.-O., Lee, D.-W. and Chang, W., "Multi-Mission Scheduling Optimization of UAV Using Genetic Algorithm," *Journal of The Korean Society for Aviation and Aeronautics*, Vol. 26, No. 2, June 2018, pp. 54~60.