

무인기 자율임무를 위한 BDI 아키텍처 기반 자율행동 소프트웨어 개발

Development of Autonomous Behavior Software based on BDI Architecture for UAV Autonomous Mission

양승구* · 엄태원 · 김경태
LIGNEX원 항공드론연구소

Seung-Gu Yang* · Taewon Uhm · Gyeong-Tae Kim

Fixed Wing Drone System Lab, LIGNEX1, Daejeon 34115, Korea

[요 약]

현재 대한민국은 인구절벽 현상에 의한 병역자원의 감소 문제에 직면해 있으며 이에 대응하기 위해 군 구조 개편 및 전력 구조 변화를 추구하고 있다. 육군은 이런 상황 변화에 대한 대응으로 미래 전장을 주도할 드론봇 전투체계의 전력화를 추진하고 있다. 미래의 전장은 지휘통제, 감시정찰, 정밀타격이 결합된 통합 전장 개념으로 전환될 것이며 이러한 변화에 따라 드론봇을 포함한 무인 전투체계가 실전 배치되어, 위험도가 높고 인간이 수행하기 어려운 전투상황에 더욱 광범위하게 적용될 것이다. 본 논문에서는 이러한 변화에 대한 대응 방안의 하나로 BDI 아키텍처 기반의 의사 결정 시스템을 가진 자율행동 소프트웨어를 개발하였다. 자율행동 소프트웨어는 다 기종에 대한 적용성 확보를 위해 프레임워크 구조를 적용하였고 대상 무인기를 대대급 감시정찰 무인기로 가정하여 PC기반 환경에서 기능을 검증하였다.

[Abstract]

Currently, the Republic of Korea is facing the problem of a decrease in military service resources due to the demographic cliff, and is pursuing military restructuring and changes in the military force structure in order to respond to this. In this situation, the Army is pushing forward the deployment of a drone-bot combat system that will lead the future battlefield. The battlefield of the future will be changed into an integrated battlefield concept that combines command and control, surveillance and reconnaissance, and precision strike. According to these changes, unmanned combat system, including dronebots, will be widely applied to combat situations that are high risk and difficult for humans to perform in actual combat. In this paper, as one of the countermeasures to these changes, autonomous behavior software with a BDI architecture-based decision-making system was developed. The autonomous behavior software applied a framework structure to improve applicability to multiple models. Its function was verified in a PC-based environment by assuming that the target UAV is a battalion-level surveillance and reconnaissance UAV.

Key word : Autonomy, Autonomous Behavior, BDI Architecture, Decision Making, Intelligence Surveillance Reconnaissance, Unmanned Aerial Vehicle.

<https://doi.org/10.12673/jant.2022.26.5.312>



This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Received 16 September 2022; Revised 5 October 2022

Accepted (Publication) 19 October 2022 (30 October 2022)

*Corresponding Author; Seung-Gu Yang

Tel: +82-042-723-2487

E-mail: Seunggu.yang2@lignex1.com

1. 서론

대한민국의 국방 여건은 인구절벽 현상에 따른 병역자원의 감소문제와 직면해 있어, 이에 대응하여 정예화된 부대 및 전력구조로의 개편이 필요하다. 또한 군 구조개편 및 전력구조 변화에 대응하기 위해 육군은 미래 전장을 주도할 드론봇 전투체계의 전력화를 추진하고 있다[1].

미래의 전장은 지휘통제, 감시정찰, 정밀타격이 결합된 통합 전장 개념으로 전환될 것이며 이러한 변화에 따라 드론봇을 포함한 무인 전투체계가 실전 배치되어, 위험도가 높고 인간이 수행하기 어려운 전투상황에 더욱 광범위하게 적용될 것이다[2].

본 논문에서는 대한민국 국방 여건 및 미래 전장 패러다임 변화에 대응하는 하나의 기술로 다양한 플랫폼에 적용 가능한 프레임워크 구조의 무인기 자율임무를 위한 자율행동 소프트웨어를 제안한다.

본 논문의 구성은 다음과 같다. 2-1 절에서는 본 연구의 의사결정을 위한 기본 기술인 BDI 아키텍처에 대해서 설명한다, 2-2 절에서는 본 연구의 프레임워크화된 자율행동 소프트웨어의 구조를 설명하고, 2-3 절에서는 자율행동 프레임워크에 탑재될 의사결정 모듈의 실행 구조에 대해 설명하였다. 이어지는 하부 장절에서는 의사결정 모듈 및 시나리오 기반 시험 결과에 대해서 제시하였다. 마지막으로 IV 장에서는 본 연구의 결론을 제시한다.

II. 본론

자율행동 소프트웨어란 환경의 변화에 따른 사용자의 다양한 요구를 만족할 수 있어야 한다. 이러한 자율행동 소프트웨어를 구현하기 위해서는 무인기 임무에 대한 도메인 지식, 상황 인지, 의사결정 등이 필요하다.

본 논문에서는 자율행동을 위한 의사결정 기능을 구현하기 위하여 BDI(Belief/Desire/Intention) 아키텍처 기반의 절차적 추론 시스템(Procedural Reasoning System)의 구현체 중 하나인 JAM(Java Agent Module)을 적용하였다[3], [4].

2-1 BDI 아키텍처

본 논문에서 제시하는 자율행동 소프트웨어 내 의사결정 시스템은 BDI 아키텍처에 기반을 두고 있다. BDI 아키텍처는 절차적 추론 시스템(Procedural Reasoning System)의 이론적 배경으로 동적 환경에서 복잡한 작업을 수행할 수 있도록 실시간 추론을 제공하고, 주어진 상황 및 지식 정보를 통해 현재 수행 가능한 작업 계획을 판단하고 수행할 수 있도록 한다[5].

BDI 아키텍처는 믿음(Belief), 목표(Desire), 의도(Intention)를 통해 인간의 생각과 행동 과정을 모방하는 소프트웨어 모델이다. 믿음은 환경 모델로서 인간이 사실로 인식한 데이터

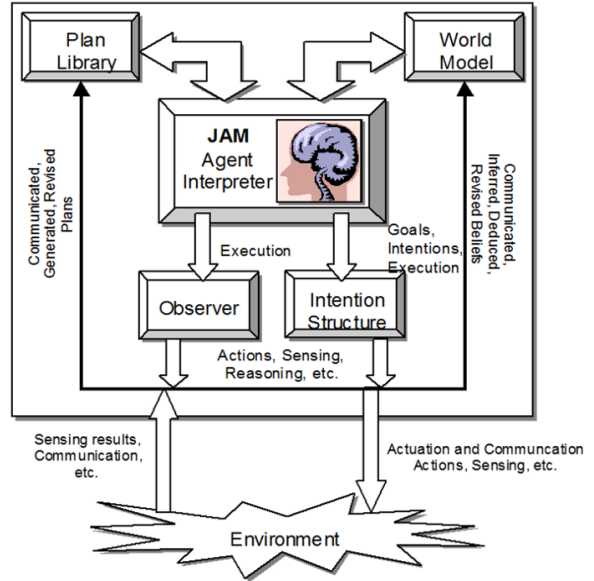


그림 1. JAM 지능형 에이전트 아키텍처
Fig. 1. JAM intelligent agent architecture.

를 나타내며 이는 거짓일 가능성도 내포하고 있다. 목표는 인간이 바라는 이상적인 환경 상태로 미래에 성취되기를 바라는 것을 나타낸다. 의도는 목표를 달성하기 위해 현재 수행할 수 있는 다양한 행동들을 말한다.

BDI 아키텍처는 믿음, 목표, 의도의 3가지 요소를 통해 목표를 이루기 위해 순간 행해야 할 적합한 행위를 결정한다. 그림 1은 BDI 아키텍처 기반 구현체 중 하나인 JAM의 아키텍처를 보여준다. JAM(Java agent Module)에서 BDI 아키텍처의 Belief/ Desire/Intention은 World Model/Goal/Intention(Plan)으로 대응된다[6].

JAM은 외부 환경의 변화에 대해 Observer를 통해 인식하여 World Model에 갱신한다. Agent Interpreter는 World Model의 상황 인지 정보에 따라 수행 가능한 Plan을 Plan Library에서 검색하여 Intention Structure에 저장하여 목표를 이루기 위한 행동을 수행할 수 있도록 한다.

JAM은 Goal 달성을 위한 Plan을 정의하기 위해 스크립트 언어 형식을 사용한다. 그림 2는 JAM Plan의 구조를 보여준

```
Plan [GoalType] [goal specification]
{
    NAME: [string]
    BODY: [procedure]
    <DOCUMENTATION: [string]>
    <PRECONDITION: [expression]>
    <CONTEXT: [expression]>
    <UTILITY: [numeric expression]>
    <FAILURE: [non-subgoaling procedure]>
    <EFFECTS: [non-subgoaling procedure]>
    <ATTRIBUTES: [string]>
}
```

그림 2. JAM plan 구조
Fig. 2. JAM plan structure.

다. JAM Plan은 Goal의 형식으로 Perform, Achieve 및 Maintain을 가진다. Perform은 단순 수행, Achieve는 Goal 달성 시까지 반복 수행하며 Maintain은 Goal이 제거될 때까지 반복 수행하는 Goal 형식이다. Plan 구조 중 Name은 각 Plan을 구분하는 식별자 역할이고 Body는 Plan에서 수행될 절차를 작성하는 부분이며 두 요소는 필수 요소이다. 그 외 ‘<’로 구분된 항목은 선택적 요소(위에서부터 : Plan 설명, Plan 실행 유지 조건, Plan 사전조건, Plan 유용도, Plan 실패 시 실행 절차, Plan 성공시 실행 절차, Plan의 특성 지정 요소)이다. JAM은 절차적 추론을 통해 Plan을 선택하고 실행하여 Goal을 달성하는 구조를 가진다[6].

2-2 자율행동 소프트웨어

그림 3은 자율행동 소프트웨어의 연동 구조를 나타낸다. 자율행동 소프트웨어는 기존의 무인기에서 직접 연동되던 데이터링크와 비행제어컴퓨터 사이에 삽입하여 지상의 운용자 대신 무인기 유도명령을 생성하여 비행제어컴퓨터로 제공하여 무인기의 자율화를 이루도록 하였다.

자율행동 소프트웨어는 외부 정보 수신 및 비행체 제어 정보 송신을 위한 어답터 CSC(Computer Software Component)와 JAM 기반의 자율행동 의사결정 및 비행/영상센서 제어 명령 생성을 위한 자율행동 프레임워크 CSC, 임무 정보 및 비행체 정보, 의사결정 모델을 포함하는 자율행동 모델 CSC로 구분된다[8].

어답터 CSC는 비행제어 컴퓨터(FCC; Flight Control Computer)와 연동하는 비행제어컴퓨터 어답터 CSU(Computer Software Unit), 영상센서와 연동하는 영상센서 어답터 CSU, 지상연동을 위한 데이터링크와 연동되는 데이

터링크 어답터 CSU로 구성하였다. 비행제어컴퓨터 어답터 CSU는 비행제어컴퓨터와 연동되며 자율행동 프레임워크 CSC에서 사용되는 데이터 형식과 비행제어컴퓨터 사용 데이터 형식 간 변환과 데이터 연동 기능을 수행한다. 영상센서 어답터 CSU는 영상센서로부터 수신한 영상센서 상태 정보 및 표적/위협 인식 정보를 수신하여 자율행동 프레임워크 CSC로 전송하고 자율행동 프레임워크 CSC로부터의 영상센서 제어 명령을 영상센서로 전송하는 기능을 수행한다. 데이터링크 어답터 CSU는 데이터링크를 통해 수신된 지상 운용자의 명령을 자율행동 프레임워크 CSC로 전송하고 자율행동 프레임워크 CSC 및 비행제어컴퓨터, 영상센서의 상태 정보 등의 지상으로 송부해야 할 데이터를 데이터링크 장비로 송신하는 기능을 수행한다.

자율행동 프레임워크 CSC는 현재 비행 상태 정보 및 영상센서 상태, 표적/위협 정보 등을 기반으로 자율행동 의사결정 수행 후 의사 결정에 따른 비행 제어 명령 및 영상센서 제어 명령을 생성한다. JAM CSU는 BDI 아키텍처 기반으로 의사결정 모델에 따라 자율행동 의사결정을 수행한다. 비행 제어 CSU는 JAM CSU의 자율행동 의사결정 및 비행 상태에 따라 비행 제어 명령을 생성한다. 영상센서 제어 CSU는 JAM CSU의 자율행동 의사결정 및 표적/위협 정보를 바탕으로 영상센서 제어 명령을 생성한다. 비행 계획 CSU는 사전 정의된 비행 계획을 관리하는 기능을 수행한다. 데이터 관리 CSU는 자율행동 프레임워크 CSC 내에서 사용하는 공유 데이터를 관리하는 기능을 수행한다.

자율행동 모델 CSU는 자율행동 프레임워크를 동작 시키기 위해 임무 및 비행체 특성 등의 정보와 자율행동을 위해 JAM CSU에서 해석되어 동작하는 의사결정 모델을 포함한다.

2-3 자율행동 의사결정 흐름

자율행동 소프트웨어는 초기화 시 자율행동 프레임워크 CSC의 자율행동 모델 로드 기능을 통해 자율행동 모델 CSC의 속성 파라미터와 의사결정 모델을 로드한다. 속성 파라미터는 비행체의 특성 및 임무 중속적인 정보를 포함하며 자율행동 프레임워크는 속성 파라미터를 통해 비행 제어 및 영상센서 제어를 위한 기초 자료로 활용한다. 의사결정 모델은 JAM CSU의 Plan 및 Goal로 구성된 스크립트 언어로 임무 수행의 기본 흐름 및 변화된 상황에 대한 대처 방법 등이 기술된다.

자율행동 소프트웨어는 외부 환경의 변화를 인식하기 위해 어답터 CSC를 사용한다. 그림 4는 외부 환경 변화에 JAM CSU의 의사결정 흐름을 나타낸다. 어답터 CSC는 비행제어 컴퓨터에서 수신되는 비행체의 위치, 자세, 속도 및 가속도 정보 등의 비행 상태 정보를 수신하고 영상센서를 통해 표적/위협 탐지 여부와 탐지된 표적/위협의 위치 정보 및 영상센서 상태 정보를 수신하여 자율행동 프레임워크로 송신하여 데이

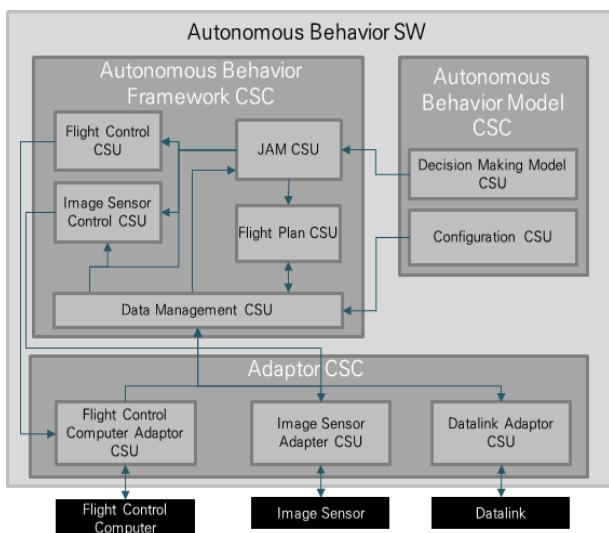


그림 3. 자율 행동 소프트웨어 연동 구조
 Fig. 3. Autonomous behavior SW interfaces.

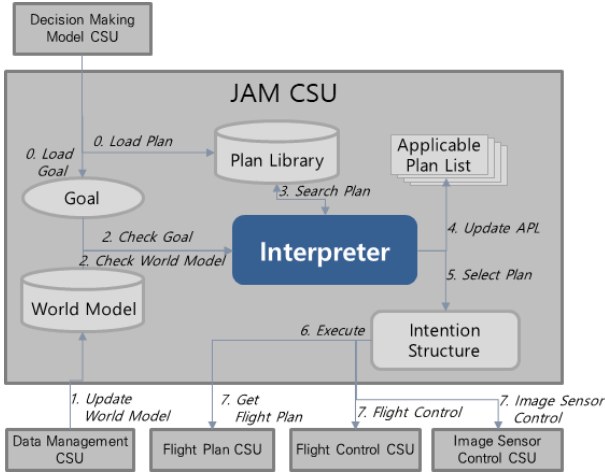


그림 4. JAM CSU의 의사결정 흐름
 Fig. 4. Flow diagram of decision making on JAM CSU.

터 관리 CSU를 통해 JAM CSU의 World Model에 저장한다. JAM CSU는 World Model의 변화가 발생하면 보유한 Plan을 검색하여 사전 조건에 따른 실행 가능한 Plan 리스트를 갱신한다. 갱신된 실행가능한 Plan 리스트 중 유용도(Utilty)가 가장 높은 Plan 이 선택되어 Intention Structure에 포함된다. Intention Structure는 선택된 Plan을 실행하여 Plan에 정의된 작업을 수행하여 비행 제어 CSU 및 영상 센서 제어 CSU를 실행시키고 필요에 따라 비행 계획 CSU로부터 항로점 정보를 수신한다.

2-4 임무 정의

본 논문에서는 자율행동 소프트웨어 시험을 위한 시험 환경으로 대상 무인기, 임무 범위 및 임무를 아래 표 1과 같이 가정하였다. 시험 대상 무인기는 소형 복합형 무인기로 수직 이착륙이 가능한 정찰용 무인기로 육군 대대에서 사용될 것으로 가정하였다. 임무 영역은 미래 부대의 작전 범위에 따라 6km x 8km로 가정하였다[1].

임무란 운용자가 무인기 운용을 통해 달성하고자 하는 목표이다. 자율행동 소프트웨어의 임무는 정찰용 대대급 무인기가 사전 정의된 비행 계획에 따라 비행하며 표적 탐지 시 표적 추적, 위협 탐지 시 위협 회피 비행을 수행하여 비행 계획

표 1. 시험 가정 사항
 Table 1. Assumptions for test.

Item	Assumption
Target UAV	small combined unmanned aerial vehicle
Classification of UAV	VTOL(Vertical Take-Off and Landing aircraft)
Scale	for battalion
Size of Mission Area	6km x 8km
Types of Mission	ISR(Intelligence, Surveillance, Reconnaissance)

을 완수하고 기지로 귀환하는 것으로 정의하였다[7].

2-5 의사결정 모델

군의 드론봇 전투체계 발전방향 연구보고서 및 국방 드론봇 통합관제 체계 발전방향 연구 논문에 따르면 정찰용 대대급 무인기는 6km x 8km의 임무 영역을 가지며 평시에는 전장 상황 파악을 위한 운용을 수행하며 국지전 시에는 중심지역 내 운용을 통해 적 주요활동에 대한 조기 정보체계 구축하고 전면전 시에는 전투원 근접이 제한되는 위험 지역에서의 정보활동 수행을 목적으로 두고 있다[7],[8].

자율행동 소프트웨어는 항로 추종, 표적 추적, 위협 회피, 항로 복귀, 기지 귀환, 데이터링크 복구의 총 6가지 행동에 대해 의사결정 모델의 절차적 추론을 통해 선택 및 수행된다.

의사결정 모델은 앞서 설명한 바와 같이 JAM CSU에서 해석 가능한 스크립트 언어의 형태로 Plan을 구현된다. JAM CSU는 World Model의 상태 변화를 확인하여 변화 발생 시 반응적으로 의사결정하는 구조를 가지고 있어 기본 흐름과 상황 변화에 대한 정의를 통해 의사결정을 수행한다. 자율행동 소프트웨어의 의사결정 모델은 사전 정의된 비행 계획에 따른 비행을 수행하는 항로 추종을 기본 흐름으로 설정하여 표적/위협 탐지, 경로 이탈, 마지막 항로점 도착 및 데이터링크 두절의 상황 발생 시 해당 상황에 맞는 의사결정을 수행한다. 표 2는 6개의 의사결정 항목에 대한 사전 조건 및 의사결정 유용도를 나타내었다.

항로 추종은 사전 정의된 비행 계획에 따라 비행하는 의사결정 항목으로 상황변화가 없는 상황에서 선택되어 진다. 항로 추종 중에는 다음 항로점에 도착 시 목표 항로점을 갱신하고 비행 중 영상센서에 전방 주시 명령을 통해 무인기 전방에 표적/위협을 탐지 할 수 있도록 구현하였다.

항로 복귀는 표적 추적 및 데이터링크 복구 등의 의사결정에 따른 자율행동 후 기존의 항로로 복귀하기 위한 의사결정 항목이다. 항로 복귀 의사결정 시에는 항로 복귀여부를 지속적으로 확인하며 영상센서는 정방 주시 명령을 통해 무인기 전방의 표적/위협을 탐지할 수 있도록 하였고 항로 복귀 완료

표 2. 의사결정 항목 별 사전 조건 및 의사결정 유용도
 Table 2. Precondition and utility of decisions.

Decision	Precondition	Utility
Follow Plan	the next waypoint remains	10
Return to Route	out of the route	11
Track Target	recognized the target to track	12
Evade Threat	recognized the threat to evade	13
Recover Datalink Loss	datalink loss	14
Return to Base	last waypoint arrived not enough battery	15

후 항로 추종에 들어갈 수 있도록 항로 추종 다음으로 낮은 유용도를 설정하여 구현하였다.

표적 추적은 영상센서를 통해 표적이 탐지 되었음을 수신한 순간 선택되어 진다. 표적 추적 시 무인기는 표적과의 이격 거리를 유지하며 선회 비행하도록 하였다. 또한 영상센서는 표적의 이동을 관찰하기 위해 표적을 지향하도록 지속적인 제어 명령을 생성하도록 하였으며, 표적 추적 후 일정 시간이 지나면 표적 추적을 중단하도록 구현하였다.

위협 회피는 영상센서를 통해 위협이 탐지 되었음을 수신한 순간 선택되어 진다. 위협 회피 시 무인기는 위협과의 이격 거리를 유지하며 우회 비행하도록 하였다. 또한 영상센서는 위협의 이동을 관찰하기 위해 위협을 지향하도록 지속적인 제어 명령을 생성하도록 하였으며, 위협 회피가 종료되는 순간은 우회 비행 중 위협 회피 진입 시점의 반대편 항로에 도착하는 순간으로 구현하였다.

데이터링크 회복은 데이터링크가 끊어져 지상과의 연동이 되지 않는 순간 선택되어 진다. 데이터링크 회복 시 사전 정의된 데이터링크 회복 지점으로 이동하여 데이터링크 회복을 대기하도록 하였고 데이터링크가 다시 연결되는 순간 데이터링크 회복 의사결정을 종료하도록 구현하였다.

기지 귀환은 비행 계획의 마지막 항로점에 도착하거나 배터리 부족으로 비행 계획에 따른 임무 수행이 불가능한 경우 선택되어 진다. 기지 귀환 시 의사결정 모델은 기지로 무인기를 이동하도록 하고 영상센서는 전방을 주시하도록 구현하였다.

III. 시험 결과

본 논문에서 제안한 자율행동 소프트웨어는 비행상태 정보, 영상센서의 표적/위협 탐지 및 위치 정보, 지상 제어 명령으로 외부 상황을 인식 한다. 개발 결과를 확인하기 위한 시험은 자율행동 소프트웨어의 기능 및 성능 시험의 편의성을 위해 외부 환경을 모의하는 대신 외부 상황을 직접 인가하는 방식을 사용하였다. 자율행동 소프트웨어를 탑재할 임베디드 보드로는 NVIDIA 사의 TX2i 보드를 사용하였다. 시험 환경은 그림5와 같다.

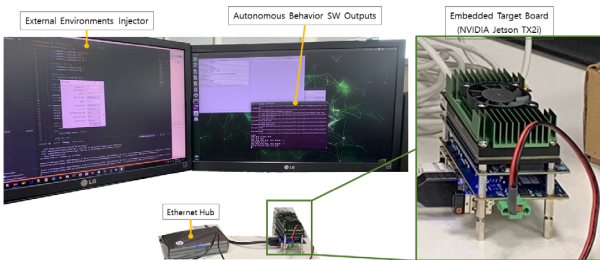


그림 5. 시험 환경
Fig. 5. Test environment.

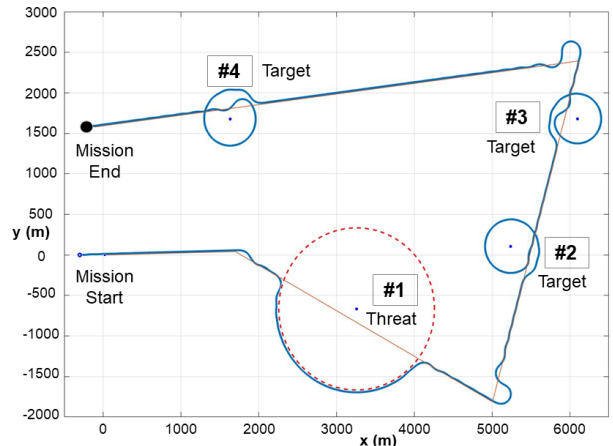


그림 6. 시험 시나리오 1의 결과
Fig. 6. Result of first test scenario.

자율행동 소프트웨어의 검증을 위한 시험을 위해 시험 시나리오를 정의하였다. 첫 번째 시나리오는 사전 정의된 비행 계획에 따라 비행하며 위협 회피 및 표적 추적을 수행하는 지 확인하기 위한 시나리오이다. 첫 번째 시나리오에서 출현하는 객체는 총 4개로 회피 대상 위협 1개와 추적 대상 표적 3개를 가지며 시작/종료 항로점을 포함하여 5개의 항로점을 가진다.

두 번째 시나리오는 첫 번째 시나리오와 유사하나 비행 중 두 번째 표적을 추적 중 데이터링크 두절 상황이 발생하는 경우 데이터링크 회복 행동을 수행하는지 확인하기 위한 시나리오이다.

3-1 시나리오 1 시험 결과

그림 6은 첫 번째 시나리오에 대한 시험결과이다. 그림의

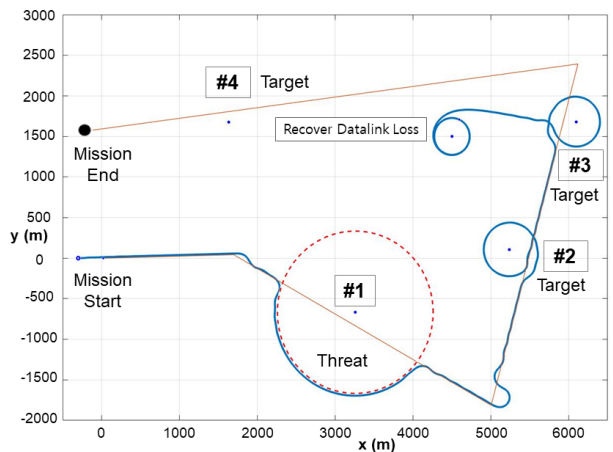


그림 7. 시험 시나리오 2의 결과
Fig. 7. Result of second test scenario.

황색 선은 사전 비행 계획이며 파란 선이 시나리오에 따라 무인기가 이동한 항적이다. 무인기는 임무 시작 후 첫 번째 항로 점을 지나 위협(#1)을 탐지하게 되고 1km의 이격거리를 두고 회피 비행을 수행하였다. 회피 완료 후 다시 항로로 복귀되어 다음 비행 계획에 따라 다음 항로점을 향해 비행하며 3개(#2, #3, #4)의 추적 표적을 탐지하여 표적 추적 비행을 수행한 후 항로로 복귀하여 잔여 비행 계획을 수행하는 모습을 볼 수 있다.

3-2 시나리오 2 시험 결과

그림 7은 두 번째 시나리오에 대한 시험 결과이다. 무인기는 임무 시작 후 첫 번째 항로점을 지나 위협(#1)을 탐지하여 회피 비행 후 첫 번째 표적(#2)에 대해 추적을 완료하였다. 그 후 두 번째 표적(#3)에 대한 추적 진행 중 데이터링크 두절이 발생하여 사전에 지정된 데이터링크 회복 가능 지점으로 이동하여 선회하며 데이터링크 회복을 대기하는 비행을 수행하였다.

IV. 결 론

본 연구는 BDI 아키텍처 기반의 의사결정 시스템을 활용하여 플랫폼의 종류 및 다양한 임무에 영향 없이 공통적으로 사용 가능한 프레임워크 구조의 자율행동 소프트웨어를 제안하였다. 자율행동 소프트웨어는 지상의 운용자를 대신하여 의사결정 시스템을 통해 결정된 행동에 대해 비행체 유도 명령 및 센서 제어 명령을 생성하여 자율적으로 임무를 수행하게 하였다. 시뮬레이션 결과는 제안한 자율행동 소프트웨어가 사전 정의된 임무를 수행하며 임무 수행 중 발생하는 상황(표적/위협 탐지, 데이터링크 두절 등)에 대해 자율적인 대처가 가능함을 검증하였다.

본 연구에서의 자율행동 소프트웨어는 의사결정 및 무인기 제어에 대한 부분을 프레임워크로 구현하였다. 무인기의 자율화 성능 향상을 위해서는 표적 및 위협 식별을 위한 객체 인식(Object Recognition) 기술과 현재 전장 상황 인식(Situation Awareness) 향상을 상황 추론(Context Reasoning) 기술을 자율행동 소프트웨어에 융합하는 것이 필요하다. 현재는 본 연구에서의 결과물을 확장하여 다수의 표적 및 위협을 동시에 조우한 상황과 비행금지 구역 임무 영역 및 항공기의 고장 발생 상황등 임무 수행 중 발생 가능한 다양한 상황에 대해 능동적으로 대처할 수 있도록 객체 인식 및 상황추론 기술을 융합한 프레임워크 및 모델 개발 기술을 타 과제를 통해 연구

중이다.

Acknowledgments

본 연구는 2021년도 LIG 넥스원 자체투자과제(“자율 행동 S/W 프레임워크”)의 일환으로 연구되었음.

References

- [1] S. Yoon, Y. Choi, “Development of Defence Dronebot Integrated Control System: From battalion operations to Dronebot control and integrated control,” *Journal of Advances in Military Studies*, Vol. 3, No. 3, pp. 33-50, 2020.
- [2] J. Choi, “The Growth of UAV and Development Trend,” *Agency for Defence Technical Investigation Paper*, pp. 104-113, 2014.
- [3] M. Beetz, L. Mösenlechner, and M. Tenorth, “CRAM—A Cognitive Robot Abstract Machine for everyday manipulation in human environments,” *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Taipei, pp. 1012-1017, 2010.
- [4] B. Kwak and J. Lee, “Task Management Framework for Intelligent Robots,” *Journal of the Institute of Information and Technology*, pp. 21-24, 2006.
- [5] B. Choi, I. You and J. Lee, “A Policy-Based Meta-Planning For General Task Management for Multi-Domain Services,” *Journal of Korea Information Processing Society Trans. Software and Data Engineering*, Vol. 8, No. 12, pp. 499-506, Dec. 2019.
- [6] M. J. Marcus, “JAM: A BDI-theoretic mobile agent architecture,” *Proceedings of the Third Annual Conference on Autonomous Agents*, Seattle, pp. 236-243, 1999.
- [7] T. Uhm, J. Lee, G. Kim, S. Yang, J. Kim, J. Kim, S. Kim, “Decision-Making System of UAV for ISR Mission Level Autonomy,” *Journal of the Korean Society for Aeronautical and Space Sciences*, Vol. 49, No. 10, pp. 829-839, 2021.
- [8] S. Yang, J. Shin, G. Kim, “Method of Autonomous Behavior Software Design based on BDI Architecture for UAV,” *Fall Conference of the Korean Society for Aeronautical and Space Sciences*, Jeju, pp.146, Nov. 2021.



양 승 구 (Seung-Gu Yang)

2006년 2월 : 중앙대학교 전자전기공학부 (공학사)
2008년 2월 : 중앙대학교 전자전기공학부 (공학석사)
2008년 1월 ~ 2011년 10월 : 코원시스템 MD 사업부 대리
2011년 11월 ~ 현재 : LIG넥스원 항공드론연구소 수석연구원
※ 관심분야 : 항공전자시스템, 내장형 소프트웨어, 자율화 무인기



엄 태 원 (Taewon Uhm)

2007년 8월 : 연세대학교 전기전자공학과 (공학사)
2009년 8월 : 연세대학교 전기전자공학과 (공학석사)
2009년 7월 ~ 현재 : LIG넥스원 항공드론연구소 수석연구원
2018년 3월 ~ 현재 : 충남대학교 항공우주공학과 (전공: 구조 및 제어, 박사수료)
※ 관심분야 : 지능 제어, 강화 학습, 센서 데이터 융합



김 경 태 (Gyeong-Tae Kim)

2019년 8월 : 서울시립대학교 컴퓨터과학부 (공학사)
2019년 8월 ~ 현재 : LIG넥스원 항공드론연구소 선임연구원
※ 관심분야 : 항공전자시스템, 내장형 소프트웨어, 자율화 무인기