

자가 적응 무인 시스템의 임무수행 전략 검증

김상수⁰, 채정욱, 인 호
{sookim, bravejw, hoh_in}@korea.ac.kr
고려대학교 컴퓨터학과

Verification of Self-Adaptation Strategy for Unmanned Weapon Systems

Sang-Soo Kim⁰, Joung-Wook Chae, Hoh In
Department of Computer Science and Engineering, Korea University

요 약

자가 적응 시스템을 무인 시스템(UWS: Unmanned Weapon Systems)에 적용하기 위한 다양한 연구가 이루어지고 있다. 자가 적응 시스템은 임무중인 시스템이 다양한 주변 환경 및 시스템의 변화에 따라 능동적으로 시스템 또는 임무수행을 위한 전략을 수정해 항상 최상의 성능을 발휘할 수 있도록 하는 능력을 갖춘 시스템을 말한다. 자가 적응 시스템에서 능동적으로 변화시킨 시스템의 아키텍처나 임무수행 전략이 유효한 것인지에 관한 검증을 수행한 후 시스템에 적용해야 한다. 기존의 대부분의 자가 적응 시스템에 대한 연구결과에서는 능동적으로 변화된 시스템이 임무수행에 적합한지에 대한 검증 방법을 제시해 주고 있지 않다. 본 연구에서는 UWS의 자가 적응 시스템이 임무수행 중 변화 되었을 때 미래의 발생할 사건에 대해 적절하게 적용 가능한지를 검증하기 위하여 시간적인 사건의 완전성을 검증하기에 적합한 Computation Tree Logic(CTL) 모델체킹(Model Checking)을 적용하여 자가 적응 시스템의 적응결과를 검증하는 방법을 제시하였다.

1. 서 론

일반적인 mission critical system(군사용-Unmanned Weapon Systems, sensor to shooter system, network centric, 의료용, 관공, 통신시스템)은 특정임무를 수행하기 위해 적합하도록 고정되어 있다. 이러한 시스템은 일반적인 임무에는 적용 가능하지만 예기치 않은 임무환경의 변화와 급작스런 비정상상태에서는 시스템의 성능을 발휘하지 못한다. 따라서 이러한 문제점을 해결하기 위하여 자가 적응 시스템을 적용하고 있다.

자가 적응 시스템은 임무수행 중에 시스템을 중단하지 않고 새로운 환경에서도 최대의 성능을 발휘 할 수 있도록 자체적인 알고리즘의 변화 또는 새로운 임무수행 전략을 수립하여 적용한다. 이때 반드시 수행되어야 하는 것은 새로운 환경에 적용하기 위한 알고리즘의 변화와 임무수행 전략이 효과적인지를 검증하는 일이다. 불행하게도 지금까지 연구된 자가 적응 시스템의 경우 검증을 통하여 시스템의 완전성을 확보하기 위한 연구 성과는 없다.

본 연구에서는 UWS에 적용하기 위하여 자가 적응을 통해 얻어진 결과를 실 시스템에 적용되기 전에 검증하기 위한 방법으로 미래에 일어날 일의 완전성을 검증할 수 있는 Computation Tree Logic(CTL) 모델체킹(model checking)을 이용하는 방안을 제시 하였다. 본 논문에서 다루고 있는 자가 적응 시스템 아키텍처는 mission critical 실시간 시스템에 적용하기 위한 것이다. UWS의 자가 적응 시스템을 통해 개발된 새로운 환경에 대해 수립된 전략이 실제 임무를 수행하기에 적합함을 검증하기 위하여 임무수행전략을 Finite States Machine (FSM)으로 표현한 후 CTL 모델체킹으로 검증하였다.

이를 위해 본 논문의 2장에서는 자가 적응 시스템의

개념과 FSM, CTL 모델체킹, 등 연구의 이론적 배경에 대하여 설명하고, 3장에서는 UWS 도메인 분석 및 검증을 위한 프레임워크를 제시하고, 4장에서는 세부 실행방법을 제시하고 사례를 보였으며, 5장에 결론과 향후 연구방향에 대하여 언급하였다.

2. 배 경

2.1 자가 적응 시스템

자가 적응 시스템(Self-adaptive systems (software))은 운용 환경의 변화에 대응하기 위해 자기 자신의 행위를 스스로 개조하여 적응해 가는 시스템을 말한다.[1] 이때 운용환경을 뜻하는 것은 소프트웨어 시스템으로부터 관찰될 수 있는 사용자의 입력, 외부 하드웨어 장치 및 센서 또는 프로그램 계측을 통해서 얻어지는 것을 말한다.

Mission Critical 실시간 시스템의 운용 또는 작동환경은 유비쿼터스 환경화 되면서 복잡하고 역동적으로 변화하므로 다양한 정보가 생성되며 생성된 정보는 시스템에 영향을 준다. 다양한 변화에 대한 정밀한 정보를 시스템에 반영하기 위해서 바로 자가 적응형 시스템이 필요하다. 무인무기체계(UWS: Unmanned Weapon Systems)의 운용 환경에서 살펴보면 현대 및 미래 전의 전장 환경은 Dynamic하게 변화되는 형태이며, UWS의 알고리즘에 대응해서 UWS의 방어력을 무력화 시키는 무기체계가 현대 및 미래 전에서 사용될 것이다.

UWS가 이러한 위협에 대처하기 위해서는 적 또는 적 무기체계의 활동과 상태를 파악하여 탐색하고 추적하여 교전하는 전략을 적절하게 수립하고 변화 시키면서 임무를 수행해야 한다. 하지만 일반적인 무인무기체계의 경우

다음과 같은 문제점으로 인하여 임무에 실패 할 것이다:

- 시스템의 주변 상황 또는 환경에 맞는 전략을 수립하는 것, 알고리즘의 변화 즉 임무수행 전략의 변화는 통제소로부터의 인위적인 변화를 통해서서는 빠른 대처가 불가하다.
- 일반적인 무기체계는 주변상황의 변화에 따라 적절한 대응을 하기위하여 조건 표현 방법을 적용하고 있다. 하지만, 이 또한 현대 및 미래전에서의 위협을 제압하기에는 부적절하다.

따라서, 전장 환경의 변화를 자동으로 인식하여 평가하고 (situation awareness), 적절한 대응전략을 수립하여 (self-adaptation), 유효성을 검증 후 (Verification), 실시간적으로(real-time) 임무를 수행 할 수 있는 시스템이 필요하다. 본 연구에서는 이를 자가 적응 시스템이라 정의하고(자가적응 시스템에 대한 세부설명은 생략하였다), 이러한 자가 적응형 시스템이 적용된 UWS가 임무수행을 위해 수립한 제어전략의 유효성 검증 시 CTL 모델체킹을 이용하는 방법에 대하여 제시하였다.

2.2 Finite State Machine

Finite State machine (FSM) M은 다음과 같이 6개의 tuple로 정의 된다:

$$M=(I, O, S, \delta, \Upsilon, S^0)$$

- I: 입력
- O: 출력
- S: 상태(state)
- $\delta: I \times S \rightarrow S$ 는 상태전이함수
- $\Upsilon: I \times S \rightarrow O$ 는 출력매핑함수
- S^0 : 초기 상태

임무수행 전략을 표현하기 위한 FSM에 대하여, 상태는 임무수행 중인 시스템의 고정된 조건을 말하며, 입력은 물리적인 환경의 측정결과이고, 출력은 임무수행을 위한 명령을 뜻한다. δ 는 현재 상태와 입력에 따른 시스템의 다음단계를 뜻한다.

2.3 CTL Model Checking

시제 논리(Temporal Logic) 모델 검증에서는 상태 전이 시스템과 시제 논리로 된 프로퍼티가 주어지면 그 시스템이 프로퍼티를 만족하는지 결정하게 된다.

모델 검증의 주요한 두 가지 논리로는 Computation Tree Logic (CTL)과 Linear Temporal Logic (LTL)이 있다. CTL은 CTL*의 제한적인 부분집합으로 항상 경로 제한자 $q \in \{A(\text{for all computation paths}), E(\text{for some computation path})\}$ 뒤에 시제 연산자 $o \in \{X(\text{next time}), F(\text{in the future}), G(\text{globally}), U(\text{until}), R(\text{release})\}$ 이 나타나야 하는 논리이다. LTL은 f 가 경로 식일 때 Af 형식을 따라야하는 논리이다. 본 논문에서는 CTL을 사용하여 무인체계의 임무수행 알고리즘의 유효성을 검증한다. CTL 프로퍼티의 예로 다음과 같은 것이 있다:

- $AG(Req \rightarrow AF Ack)$: 언제나, request가 일어나면, 언젠가는 acknowledged가 일어난다.
- $AG(EF Restart)$: 어떠한 상태에서부터 Restart 상태로 가는 것이 가능하다.

3. 도메인 분석 및 검증 프레임워크

3.1 도메인 분석

일반적인 UWS는 지상 통제소나 위성을 통하여 임무 정보를 교신하면서 임무를 수행하게 될 것이다. 불행하게도 다음과 같은 문제가 발생하였을 때 임무수행에 차질을 가져올 수 있을 것이다:

- 임무수행 중 새로운 추가 임무 발생,
- 임무수행 전장 환경의 변화
- 위의 상황에서 임무수행 중 교신을 위한 통신수단의 문제점 발생

위와 같은 문제가 발생하였을 때 UWS에서 가장 중요한 시스템은 자가 적응 시스템이다. 자가 적응 시스템은 외부 전장 환경의 변화에 스스로 대처하기 위하여 새로운 임무수행 전략을 수립하여 적용하게 된다. 임무수행을 위한 전장은 편의상 격자 형태로 표현하였다. 이중 일부의 노드는 임무수행을 위해 반드시 지나가야 할 지점이며 일부 지역은 대공 또는 대지 위협으로 인하여 반드시 피해 가야할 접근 금지구역으로 표현한다.

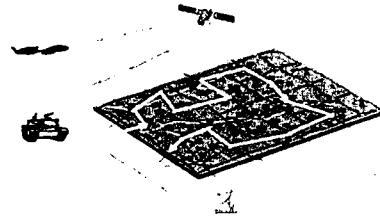


그림 3-1. 무인 무기체계 운용 환경

3.2 검증 프레임워크

CTL 모델체킹을 적용하여 임무수행 전략의 유효성을 검증하기 위한 기본적인 절차는 다음과 같다:

- ① 임무수행 전략을 수립하여 FSM을 작성한다.
- ② CTL 모델체커를 통하여 안전 property를 검사한다.
- ③ 안전하다면 새로운 전략으로 채택하고 그렇지 않다면 수립된 전략 FSM을 폐기하고 임무수행 전략을 재수립한다.

일반적으로 임무수행 전략을 수립하기 위해서는 Evolutionary Algorithm (EA) 등 다양한 방법을 적용할 수 있다. 수립된 전략은 FSM으로 표현하며 이것은 UWS의 임무수행 후보 전략(μ)이 된다. 후보 전략들은 CTL 모델체커를 통하여 임무수행 중에 접근금지 구역에 들어가지 않고 안전한지를 검사한다. 검사후 안전한 전략은 취하고 추후 새로운 전략 수립 시 활용하고 그렇지 못한 전략은 폐기하게 된다.

4. 유효성 검증

이 장은 새로이 수립된 (알고리즘)전략을 어떻게 검증하는지에 대한 세부적인 방법과 적용 결과를 제시한다.

4.1 검증 대상 문제

본 논문에서 제시하는 방법은 UWS 분야에 다양하게 적용할 수 있다. 그러나 Unmanned Air Vehicle (UAV)

무인항공기를 비롯한 UWS 분야는 상상하기 힘들 정도로 다양한 방법이 존재하기 때문에 검증 대상 문제를 선택하는 것이 무엇보다도 중요하다. 우리는 특별히 다음과 같은 두 가지의 주요 특성(property)을 고려하여 선택하였다:

- 첫째, 무인체계의 성공적인 임무수행 전략을 설명할 수 있어야 한다.
- 둘째, 관련 이해당사자들이 쉽게 이해할 수 있어야 한다.

특히 두 번째 문제의 경우의 고려하려 비전문가도 쉽게 이해하도록 하기 위하여 FSM을 통하여 검증해야할 대상을 표현하였다. 그림 4-1은 검증대상 문제에 대한 FSM의 예를 보여주고 있다.

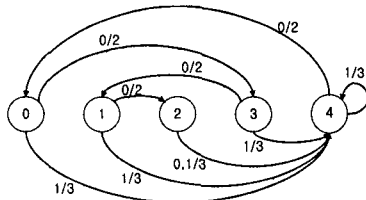


그림 4-1. UWS 임무수행 전략 FSM 예제

그림 4-1에서 입력은 UWS의 전장 환경정보를 나타내며, 출력은 UWS의 이동 및 임무수행 명령을 나타낸다. 어떠한 상태든지 초기 상태가 될 수 있다. 화살표 위의 숫자는 입력/출력(0: 임무수행 대상이 없음, 1: 임무수행 대상 있음/0: 임무중지, 1: 좌선회, 2: 우선회, 3: 계속 전진하여 임무수행)을 나타낸다. 원 안의 숫자는 상태를 나타낸다. 모든 형태의 임무수행 전략이 FSM에 완전히 반영 될 수 있다. 예를 들어, 현재 상태가 3이고 전방에 아무런 임무수행 대상이 없다면 UWS는 상태 1로 전이하고 우선회 한다. 그러나, 전방에 임무수행대상이 관측된다면 상태 4로 전이하고 UWS는 전방으로 이동하여 임무를 수행할 것이다. 이러한 통제 전략은 [2]에서 연구된 결과를 응용하였다. 본 논문에서는 임무수행 중 피해야할 지역 즉, 적의 대공포가 있는 지점을 접근금지 구역으로 지정하고 접근금지구역으로 진입 시에 UWS가 피격되며 임무는 실패하는 것으로 설정하였다.

4.2 CTL 모델 체크의 적용

접근 금지구역으로 가지 않는 것이 임무를 실패하지 않은 것으로 간주하고 이것을 안전 property S 로 설정하였다. CTL formula $AG(S)$ 는 S 가 계속해서 hold 된다는 뜻이다. 모든 상태를 검사하여 S 가 hold 되는지를 검사하여야 한다. CTL formula가 선형 시간동안 true인지 false인지 검사할 수 있다. 본 연구에서 제시한 임무수행 전략에 대한 formula S 는 다음과 같이 간단하게 표현할 수 있다:

$$S \Rightarrow \neg(\text{전방이동 임무수행} \wedge \text{접근금지구역})$$

또한, 어떠한 임무수행 전략은 다음과 같은 형식으로 안전함을 검증할 수 있다.

$$AG \left(\bigwedge_{j=1}^k s_j \right) = 1$$

S_j 가 j 번째 안전 property라고 할 때 $S_j = 1$ 은 property가 hold 된다는 것을 의미한다.

4.3 검증결과

본 연구에서의 검증 대상 전장 환경을 그림 4-2에서 보는 바와 같이 가상의 통제 결정점을 총 100개 (10 X 10 격자) 환경으로 하고 임무수행을 위해 반드시 도달해야할 지역을 8개, 접근금지 구역은 4개를 대상으로 수행한 검증을 수행하였다.

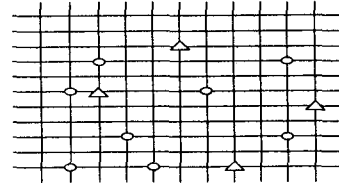


그림 4-2. 임무수행 위치(원)와 접근금지구역(삼각형)

초기 후보 전략의 population 수 μ 를 5개로 하여 FSM을 10개를 작성하여 수행하였다. 초기의 population은 random하게 FSM으로 생성하였으며, FSM은 5개-8개의 상태를 갖도록 하였다. 이 FSM들을 모델체커를 통하여 검증하였다. 검증의 time step은 50번 수행하여 결과를 확인하였다. 검증결과 초기 population(FSM)의 30% 정도가 안전하지 않은 후보 전략으로 확인되었다.

5. 결 론

본 논문에서는 자가 적응 시스템에 의해 수립된 임무수행전략의 유효성을 검증하기 위하여 CTL 모델체킹을 적용하는 방법에 대하여 제시하였다. UWS의 예를 통하여 자가 적응 시스템의 검증을 위한 방안으로서 CTL 모델 체킹이 대안이 될 수 있음을 알 수 있었다.

본 논문에서 제시한 접근법은 실험적인 성격이 강하며 실제 UWS에 적용하기에는 전략 수립 알고리즘의 유효성과, 제시된 방법론의 실시간 시스템에서의 구현 시 효율성 등에 대한 평가가 이루어진 후에야 비로로 그 가치를 인정받을 것이다. 이러한 문제를 해결하기 위하여 향후에는 자가 적응 시스템의 임무수행 전략 및 적응 방법에 관한 알고리즘에 대한 연구와 자가 적응 시스템의 소프트웨어 아키텍처 등에 관한 연구를 수행할 계획이다.

참고문헌

[1] Peyman Oreizy, Michael M. Gorlick, Richard N. Taylor, Dennis Heimbigner, Gregory Johnson, Nenad Medvidovic, Alex Quilici, David S. Rosenblum, and Alexander L. Wolf, "An Architecture-Based Approach to Self-Adaptive Software" IEEE INTELLIGENT SYSTEMS, 1999

[2] Sanchez, E., Perez-Urbe, A., and Mesot, B. (2001) Solving partially observable problems by evolution and learning of finite state machines. In Proceedings of ICES2001, LNCS 2001.