

ROS 기반의 자율주행 소프트웨어의 실시간 요구사항 분석

조수민⁰, 김진현*, 강인혜**

⁰서울시립대학교 기계정보공학과,

*국립경상대학교 지능형통신공학과,

**서울시립대학교 기계정보공학과

e-mail: jsm33582@uos.ac.kr⁰, jinkim.gnu@gmail.com*, inhye@uos.ac.kr**

Real-time requirement analysis of ROS-based autonomous driving software

Soo-Min Cho⁰, Jin-Hyeon Kim*, Inhye Kang**

⁰Dept. of Mechanical and Information Engineering, University of Seoul,

*Dept. of Intelligent Communication Engineering, Gyeongsang National University,

**Dept. of Mechanical and Information Engineering, University of Seoul

● 요약 ●

자율주행 소프트웨어의 안전성에 대한 요구가 높아짐에 따라 소프트웨어를 개발 시 정형 검증 기법을 적용하여 안전성을 확보하고자 하는 노력이 증가하고 있다. 본 논문에서는 ROS 기반의 자율주행 소프트웨어의 동작을 Timed Automata를 사용하여 모델링하고 그 정확성(실시간적 특성)을 모델체킹을 통해 정형 검증하는 방법을 제시한다. ROS 기반 응용 프로그램의 동작은 ROS가 제공하는 노드 간의 통신에 영향을 받기 때문에 자율주행 소프트웨어뿐만 아니라 그 동작에 영향을 끼치는 ROS의 큐를 통한 통신을 통합한 모델을 제시한다. 또한 통합 모델에서 큐 크기, 타임아웃, 프로세스의 주기를 매개 변수로 주어 다양한 조합에 따른 실시간 요구사항의 만족 여부를 검증함으로써 ROS에서 매개 변수 설정 기준을 제시한다.

키워드: 로봇 운영체제(ROS), 자율주행차(Autonomous Vehicle), 정형 검증(Formal Verification), UPPAAL, 모델체킹(Model Checking)

I. 서론

최근 시스템의 복잡성이 증가하고 개발 시간이 단축됨에 따라 설계 단계에서 테스트를 통한 오류를 찾아 수정하는 것은 한계가 있다. 소프트웨어 테스트는 소프트웨어 품질 보증 방법 중 하나이지만 오류가 없음을 보여줄 수 없으며 오류 존재만 표시할 수 있다.

특히 자율주행 소프트웨어는 탑승자의 안전을 보장하기 위해 반드시 발견하고 피해야 하는 정적, 동적 장애물 회피하는 복잡한 알고리즘들이 포함되어 안전을 보장하는 것은 매우 중요하다.

본 논문은 ROS 기반 응용 프로그램의 실시간 특성을 검증하기 위해 큐 크기, 타임아웃과 프로세스의 주기와 같은 구성 매개 변수의 조합을 찾고 Timed Automata(TA)를 사용하여 ROS 기반 애플리케이션의 실시간 속성을 모델링하는 접근 방식을 제안한다.

II. 관련 연구

본 논문에서의 정형 검증된 모델은 Timed Automata[1]로서 다양한 정형 검증 도구가 이 모델의 성질을 모델체킹한다. 본 논문에서 UPPAAL [2]의 TA를 모델링 및 정형 검증언어로 사용한다.

연구[3]에서는 ROS 기반 애플리케이션의 노드 간 통신에 초점을 맞추어 TA를 이용한 모델링을 보인다. 해당 연구에서는 복잡한 ROS 기반 kobuki 로봇의 소스 코드를 분석하여 해당 접근 방식을 통해 적절한 매개 변수 조합을 식별한다.

III. 통합 모델 및 검증 기법

자율주행 소프트웨어의 아키텍처는 Fig. 1.에 제시된 4가지 주요 컴포넌트들이 노드로 매핑한다. 이외에 Lidar, Odometry, Actuator 센서가 존재하며 각 노드는 특정 센서에 토픽으로 연결되어 경로 계획을 진행한다. 노드는 특정 주기를 반복하는데 선입 선처리 알고리즘

즘에 따라 스케줄링 된다. 토픽에 사용할 대기열의 길이, 노드 주기와 같이 ROS 매개 변수를 설정하여 다양한 매개 변수를 검증할 수 있게 한다.

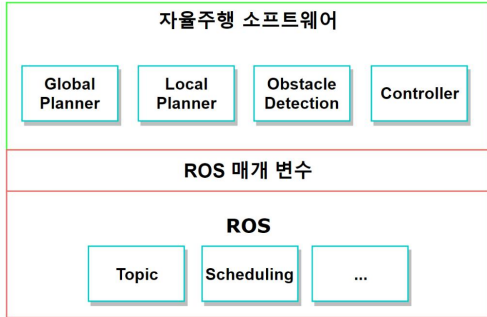


Fig. 1. ROS 기반 소프트웨어 통합 모델

Fig. 2는 자율주행 소프트웨어의 formal model의 노드를 보여준다. 노드는 subscribe queue의 데이터를 받는 상태, 경로를 계획하는 상태, 토픽으로 메시지를 보내는 상태, 주기를 기다리는 상태로 크게 4가지 상태로 나누어진다. CPU의 자원을 사용할 때는 특정 상태를 제외하고 insert_buf() 함수를 통해 대기 열에 삽입하여 CPU의 자원을 할당할 때까지 대기한다. 경로 계획이 끝나면 대기 열에서 제거한다.

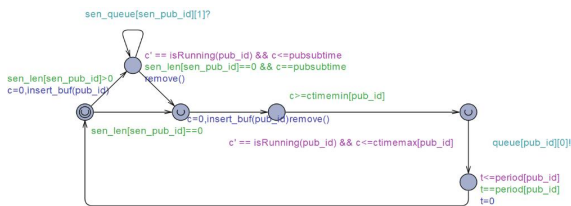


Fig. 2. Global Planner TA 모델

대기열의 오버플로를 검증하기 위해 subscriber queue를 모델링한다. Overflow 상태를 추가하여 대기 열의 길이가 최대일 때 메시지를 수신하면 Overflow 상태로 전이가 일어나게 구현하여 최종적으로 Overflow 상태에 도달하는 않는지 검증한다.

대기열 크기와 시간 제약조건에 따라 UPPAAL 모델 검사기를 사용하여 다음 속성을 검증한다. 속성(1)은 대기열이 오버플로 발생 여부, 싱글 프로세서 스케줄링 시나리오에서 속성(2)은 주기 내에 CPU 자원을 할당받지 못하는 스케줄 가능성 여부(deadlock 발생 여부로 확인)를 검증할 수 있다.

Table 1. ROS 매개 변수 (ms)

노드 주기	계산 시간	대기열 길이	콜백 시간	속성 (1)	속성 (2)
14	[1, 3]	5	[0, 1]	○	x
15	[1, 3]	5	[0, 1]	○	○

기존 자율주행 소프트웨어는 주기를 14ms로 동작하고 있었는데 Table 1.에서 보는 바와 같이 검증 결과는 데드락 가능성(속성 (2))이

있음을 확인할 수 있다. 하지만 주기를 15ms로 조정된 경우 두 특성을 위배하지 않고 잘 동작함을 보여준다. 이와 같이 통합 모델의 검증을 통해 잠재적인 결함을 확인할 수 있을 뿐 아니라 주어진 ROS 매개 변수의 조합이 원하는 특성을 만족함을 증명할 수 있다.

IV. 결론

본 논문은 자율주행 소프트웨어와 ROS의 동작을 통합하여 TA로 모델링하고 검증함으로써 소프트웨어의 안전성뿐 아니라 ROS 매개 변수에 따른 실시간성 만족 여부를 확인하는 방법을 제시하였다. 향후 연구로는 정형 검증된 설계 모델로부터 ROS 상에서 수행되는 자율주행 소프트웨어와 필수적인 안전성 속성을 실시간으로 모니터링 하는 코드를 자동 생성하는 방법을 개발하고자 한다.

REFERENCES

[1] R. Alur, "Timed automata," in International Conference on Computer Aided Verification, pp. 8-22, Springer, 1999.

[2] G. Behrmann, A. David, and K. G. Larsen, A Tutorial on Uppaal, pp. 200-236. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004.

[3] R. Halder, J. Proença, N. Macedo and A. Santos, "Formal Verification of ROS-Based Robotic Applications Using Timed-Automata," 2017 IEEE/ACM 5th International FME Workshop on Formal Methods in Software Engineering (FormaliSE), Buenos Aires, Argentina, 2017, pp. 44-50, doi: 10.1109/FormaliSE.2017.9.