

소프트웨어공학 : 프로젝트 계획서

기술개발 과제	모바일 로봇 컨트롤러의 애드온 시스템 (ADD-ON system for Mobile Robot Controller)				
과제팀 이름	오동동		지도교수		이병정
개발기간	2023년 10월 ~ 2023년11월 (총 2개월)				
개발소요비용	총액	6560 (천원)	학교부담금	6560천원	
			과제팀부담금	0천원	
과제팀 구성원	이름	신동운	한동민		
	사진				
	학번	2019550013	2019550028		
	연락처	010-2713-9515	010-5636-3381		

소프트웨어공학 프로젝트를 성실히 수행하고자 계획서를 제출합니다.

2023년 10월 16일

과제 수행자1 : 신 동 운 (인)
과제 수행자2 : 한 동 민 (인)

서울시립대학교 컴퓨터과학부 귀중

1. 개발 과제의 개요

가. 개발 과제 요약

SIM은 사람이 수행하기 힘든 임무를 무인으로 수행하기 위해 오퍼레이터가 원격으로 로봇을 조종하는 시스템이다. 그러나 재난 지역의 정보가 제한적일 때는 운용이 상대적으로 부담스러워, ADD-ON 시스템을 통해 재난 지역의 정보를 받고 알맞게 SIM을 지원하는 방식으로 운용의 부담을 덜고자 한다. 이를 통해 로봇의 활동에 능동성을 부여할 수 있다는 이점을 기대한다.

나. 개발 과제의 배경 및 효과

화재, 지진의 발생, 폭발, 독성 물질 유출 등과 같이 여러 종류의 재난 상황이 발생한 경우, 체계적인 상황 해제를 위해 다양한 정보를 파악할 필요가 있다. 재난 지역 내의 위험 지점, 구조 대상의 위치 등이 대표적일 것이다.

재난 상황에서는 구조를 위한 탈출 경로 작성 등의 체계적인 계획이 필요하며, 사람이 활동하기 힘든 재난 지역의 경우 사람 대신 작업을 수행해 주는 로봇이 필요할 것이다.

이와 같은 필요 하에서, 로봇의 동작을 시뮬레이션 하는 Mobile Robot Simulator(이하 SIM) 시스템이 만들어졌다. SIM을 통해 오퍼레이터가 원격으로 로봇의 행동을 조종하여 임무를 수행할 수 있다. 하지만 구조적으로 SIM은 재난 지역의 정보가 완전함을 가정하여 로봇의 움직임을 수동으로 제어하는 시스템이기에, 재난 지역이 방대하거나 정보가 부족할 때 운용이 어렵다.

일반적인 경우, 재난 지역에 대한 정보가 불완전한 한계가 있으므로 SIM의 한계가 명확히 드러난다. 이를 고려하여 로봇의 동작이 자동으로 제어되는 Additional Mobile Robot Controller(이하 ADD-ON) 시스템을 추가하고자 한다. ADD-ON 시스템은 SIM과 상호작용하여 재난 지역의 정보를 주고 받고, 재난 지역을 최신화한다. 이 기능을 통해 로봇이 스스로 이동 경로 및 다음 동작 등을 계획하는 효과를 기대할 수 있다.

다. 개발 과제의 목표와 내용

주어진 과제의 목표는, ADD-ON이 아래 기술될 기능들을 수행하도록 구현하여 로봇의 효율적인 임무 수행을 돕는 것이다.

ADD-ON은 사용자로부터 재난 지역에 대한 정보(Area Size, Start Location, Spot, Hazard)를

입력받아, 로봇이 활동할 재난 지역을 모델링한다. 모델링된 재난 지역은 로봇이 반드시 방문해야 하는 중요 지점, 방문해서는 안되는 위험 지점을 포함한 여러 지점들로 구성된다.

ADD-ON은 이 재난 지역(Operation Area)으로부터 로봇이 위험 지점을 방문하지 않으면서, 모든 중요 지점을 방문하도록 하는 경로를 계산한다. 이 때, 계산된 경로대로 로봇이 이동하도록 Robot Movement Interface를 조종하여 로봇을 움직인다.

[로봇의 이동 경로 결정 로직]

- (1) Operation Area의 남은 중요 지점과 로봇의 현재 위치를 모은다.
- (2) 이들 사이의 최단 거리를 계산하여 저장한다.
- (3) (2)에서 계산한 최단 거리 정보를 이용하여 로봇의 현재 위치로부터 로봇이 순차적으로 방문해야 할 중요 지점들을 줄세운다. 구현의 효율성을 위하여, Nearest Neighbor Algorithm으로 Open Revisitable TSP Problem의 유효한 근사해를 구한다.
- (4) (3)에서 결정된 순서대로 최단경로를 계산하고 경로들을 이어붙여 최종 경로를 결정한다.

ADD-ON은 SIM의 세 센서들

Hazard Sensor : 로봇의 전방 1칸이 위험 지점인지 판별한다.

Color Blob Sensor : 로봇의 전/후/좌/우 1칸이 각각 중요 지점인지 판별한다.

Position Sensor : 로봇의 지도상 좌표를 나타낸다.

로봇이 인접한 위치의 지점들에 대한 정보를 얻는다. 입력으로 준 Operation Area는 불완전하여, 센서들의 정확한 정보에 의해 덧씌워 질 수 있다.

Hazard Sensor, Color Blob Sensor에 의해 재난 지역의 정보가 변경될 경우, 로봇의 이동 경로가 다시 계산된다.

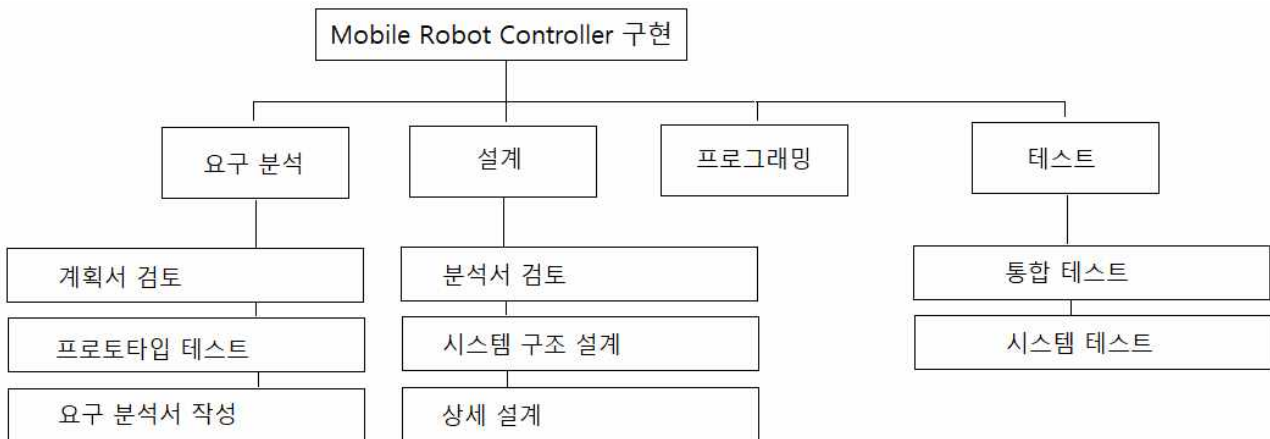
Position Sensor를 통해 로봇의 현재 위치를 받아, 계산된 경로를 정확히 따라가는지를 검사한다. 오작동 시, 로봇의 위치를 보정한다.

최종적으로 ADD-ON 시스템은 로봇이 모든 중요 지점을 방문할 수 있도록 돕는다.

2. 완료작품의 평가방법

평 가 항 목	평 가 방 법	적용기준	개 발 목표치	비중 (%)
1. 작동 여부	20회의 테스트에서 로봇이 요구사항을 만족하며 이동하는지 확인	로봇이 지역 이탈 없이 모든 중요지점을 방문, Hazard는 미방문 시 성공	20	60
2. 알고리즘 성능 테스트	20회의 테스트에서 알고리즘의 이동 거리가 실제 최단거리와 얼마나 차이어나나 확인	[알고리즘 이동 거리] / [실제 최단 거리] 값을 성능으로 정량화하여 표현	0.8	40

3. WBS(Work Breakdown Structure)



4. 개발 일정 및 추진 체계

가. 개발 일정

WBS 번호	작업 제목	작업 소유자	시작일	마감일	1단계		2단계		3단계			4단계
					1주	2주	3주	4주	5주	6주	7주	8주
1	요구분석 단계											
1.1	계획서 검토	신동운, 한동민	23년 10월 9일	23년 10월 10일								
1.2.1	프로토타입 구현(한동민)	한동민	23년 10월 11일	23년 10월 13일								
1.2.2	프로토타입 구현(신동운)	신동운	23년 10월 11일	23년 10월 13일								
1.3	프로토타입 테스트	신동운, 한동민	23년 10월 16일	23년 10월 17일								
1.4	요구 분석서 작성	신동운, 한동민	23년 10월 18일	23년 10월 20일								
2	설계 단계											
2.1	분석서 검토	신동운, 한동민	23년 10월 23일	23년 10월 25일								
2.2	시스템 구조 설계(한동민)	한동민	23년 10월 26일	23년 10월 27일								
2.2	시스템 구조 설계(신동운)	신동운	23년 10월 26일	23년 10월 27일								
2.3	상세 설계(한동민)	한동민	23년 10월 30일	23년 11월 3일								
2.3	상세 설계(신동운)	신동운	23년 10월 30일	23년 11월 3일								
3	프로그래밍 단계											
3.1.1	GUI 구현	한동민	23년 11월 6일	23년 11월 17일								
3.2.1	음식인식(센서) 구현	한동민	23년 11월 20일	23년 11월 24일								
3.3.1	맵 구현	신동운	23년 11월 6일	23년 11월 10일								
3.3.2	SIM 객체 구현	신동운	23년 11월 6일	23년 11월 10일								
3.3.3	알고리즘 구현	신동운	23년 11월 13일	23년 11월 17일								
3.3.4	이벤트 리스너 구현	신동운	23년 11월 20일	23년 11월 24일								
4	테스트 단계											
4.1	통합 테스트	신동운, 한동민	23년 11월 27일	23년 11월 28일								
4.2	시스템 테스트	신동운, 한동민	23년 11월 29일	23년 12월 1일								

나. 구성원 및 추진체계

구성원

1. 한동민 (팀장)
2. 신동운 (팀원)

추진 체계

1. 요구 분석

- 1.1 계획서 검토
- 1.2 프로토타입 구현 (한동민)
- 1.2 프로토타입 구현 (신동운)
- 1.3 프로토타입 테스트
- 1.4 요구 분석서 작성

2. 설계

- 2.1 분석서 검토
- 2.2 시스템 구조 설계 (한동민)
- 2.2 시스템 구조 설계 (신동운)
- 2.3 상세 설계 (한동민)
- 2.3 상세 설계 (신동운)

3. 프로그래밍

- 3.1 프론트엔드 구현 (한동민)
 - 3.1.1 GUI 구현
- 3.2 음성인식 구현 (한동민)
- 3.3 백엔드 구현 (신동운)
 - 3.3.1 Operation Area 구현
 - 3.3.2 SIM 객체 구현
 - 3.3.3 알고리즘 구현
 - 3.3.4 이벤트 리스너 구현

4. 테스트

4.1 통합 테스트

4.2 시스템 테스트

5. 위험분석

위험	심각성	해결 방법
1. 질병 등에 의한 회의 불참	가벼움	매주 예비 회의날을 약속하여 회의 진행
2. 시험 기간	보통	합의된 시험 기간 내에서는 회의 미진행
3. 기술적 한계 도달	심각함	Google, ChatGPT, 개발자 커뮤니티 등에서 정보를 찾아 돌파구 마련
4. 특정 인원의 개발 지연	보통	지연된 부분은 함께 해결하며 연기된 기간만큼 개발 일정 조정

6. 개발 환경

가. SW 명세

1. Visual Studio Code : 프로그래밍 코드 작성 도구
2. ChatGPT : 기술적 도움을 줄 기술 자문
3. Flask : Python 기반의 서버 프레임워크 툴
4. React : GUI 제작을 위한 JavaScript 라이브러리

나. HW 명세

개인 PC 2대

7. 개발사업비 산정내역서

(단위 : 천원)

항 목 (품명, 규격)		수 량	단 가	금 액	비 고
직 접 개 발 비	UI/UX 디자이너 인건비	1	2160	2160	주 당 10시간, 8주 시급 27,000원
	시스템 SW개발자 인건비	1	2400	2400	주 당 10시간, 8주 시급 30,000원
	작업용 개인 pc	2	1000	2000	
	합 계			6560	

8. 보고 및 감시 체계

회의 일정

회의는 **매주 수요일 20 : 00 ~ 21 : 00 [1시간]** 온라인으로 진행한다.
단, 매주 **토요일 20 : 00 ~ 21 : 00 [1시간]** 을 예비 회의로 규정하여, 수요일 회의를 진행하지 못할 경우에만 진행한다.

또한, 추가적인 회의가 필요하다고 생각될 경우, 상호 협의 하에 금요일을 제외한 18 : 00 ~ 22 : 00 사이 추가 회의를 진행할 수 있다.

정보 공유 방법

1. 계획서 : Google Docs 공유 문서
2. 산출물(요구분석서, 소스코드) 및 UML 파일 : Github