

지능형 서비스 로봇의 자가 적응 방법에 관한 연구

박정민(Jeong-Min Park)*, 전인걸(Ingeol Chun)**

ya23ma@dongyang.ac.kr, igchun@etri.re.kr

초 록

현재 산업에서 주목 받고 있는 지능형 서비스 로봇은 사람과 가장 가까운 곳에서 동작하는 중대한 시스템(Critical system)이기 때문에 발생한 오류에 대한 자가 적응 능력이 요구되고 있다. 본 연구에서는 로봇의 강건성(Robustness)을 높이기 위한 자가 적응 방법을 제안한다. 자가 적응 방법론은 총 8 단계로 구성되어 있으며 크게 1) 자가 적응을 위한 오류 정의 및 설계 단계와 2) 설계한 자가 적응 방법의 실행 단계로 나뉜다. 자가 적응 방법론을 적용한 시스템의 내부 구조는 목표시스템의 소프트웨어와 독립적으로 동작하며 목표시스템의 오류를 감지, 진단, 계획하여 적응전략을 실행하는 구조로 구성된다. 본 연구의 유효성 평가를 위해 제안 방법론을 지능형 서비스 로봇에 적용하고 오류주입 기법과 목표트리를 통하여 유효성을 검증한다.

1. 서론

지능형 서비스 로봇 산업은 스스로 인지할 수 있는 지능과 작업 능력을 가지며 인간과 시공간적으로 공존하여 상호작용을 목적으로 하는 차세대 국가 유망 산업으로 자리 잡아 가고 있다[1]. 지식경제부의 제 1 차 지능형 로봇 기본 계획[2] 이후 국내에서도 지능형 서비스 로봇의 상용화를 위한 많은 노력이 계속되고 있으며 2018 년까지 20 조원의 거대한 시장이 형성될 것으로 예상되고 있다.

이러한 정부의 계획들을 통해 점점 성장하고 있는 지능형 서비스 로봇들은 다양한 환경들에 의해서 다양화되고, 복잡해짐에 따라 로봇의 이상 상태 발생 시 문제를 감지, 진단, 해결하는데 쉽지 않은 문제점을 가진다.

IBM에서 발표한 보고서[3]에 의하면 시스템의 복잡성으로 초래하는 문제를 해결하기 위해서는 관리자에 의해 설정된 목표에 맞추어 시스템이 스스로 문제 환경에 적응하는 자율 컴퓨팅(Autonomic computing) 환경의 필요성을 강조하고 있다.

자율 컴퓨팅의 속성에는 시스템의 오류를 조사, 발견, 진단하여 정상동작을 가능하게 하는 속성을 가지는데 이를 자가 치유(self-healing) 또는 자가 적응(Self-adaptation)이라고 한다.

이러한 속성을 바탕으로 본 연구에서는 지능형 서비스 로봇의 강건성(Robustness)을 높이기 위한 방법으로 ‘지능형 서비스 로봇을 위한 자가 적응 방법’을 제안 한다. 제안 방법론은 총 8단계로 구성된다. 1) 1 단계에서는 오류 모델링(Fault Modeling) 단계로 목표 시스템의 정상동작을 분석, 발

본 연구는 한국전자통신연구원(ETRI)의 ‘고신뢰 자율제어 SW를 위한 CPS기술’ 연구비 지원으로 이루어졌음.

* 한국전자거래학회 학술이사

* 동양미래대학 전산정보학부 전임강사

** 한국전자통신연구원(ETRI) 책임연구원

생 가능한 오류의 분석, 오류 내용을 정의하고 2) 2단계에서는 오류 모델 분석(Fault Model Analysis)단계로 오류 발생 이벤트 추출 및 인과관계 정의를 정의한다. 3) 3단계에서는 오류 이벤트 분석(Fault Event Analysis) 오류들의 인과관계에 의한 트리를 모델링하고 4) 4단계에서는 오류 상태 해석(Faulty State Interpretation)단계로 오류상태를 해석할 수 있는 조건을 정의한다. 5) 5단계에서는 적응 행위자 분석(Actor Analysis) 적응 전략 및 행위자를 정의하고, 6) 6단계에서는 오류 감지(Fault Detection) 단계로 제약 조건에 따라 오류를 감지 한다. 7) 7단계에서는 오류 원인 분석(Faulty Cause Analysis)단계로 감지된 오류의 평가 및 진단을 수행하며 8) 마지막 8단계에서는 실행(Execution)단계로 평가된 오류의 적응전략을 통해 재구성을 수행한다.

제안 방법론의 사례 연구로 우리의 이전 연구인 ‘네트워크 기반의 지능형 서비스 로봇’[4]을 적용하고, 평가를 위해 오류 주입(Fault Injection)기법[5]과 목표 트리[6]를 이용한다. 이를 통해 로봇의 오류를 최대한 감내하여 일반 사용자들은 로봇의 서비스를 제공받는 것에 있어 불편을 최소화하는 것이 가능하다.

본 논문의 구성은 다음과 같다. 2장에서는 관련연구를 설명하고, 3장에서는 제안 사항을 서술한다. 4장에서는 자가 적응 시스템의 구조를 살펴보고, 5장에서는 시나리오를 설명하고 6장에서는 결론을 맺는다.

2. 관련연구

인간과 로봇의 상호작용은 크게 인간의 명령이나 의도를 인식하기 위해 로봇이 처리하는 일종의 절차와, 그에 대한 응답으로 로봇이 인간에게 보이는 출력으로 구분할 수 있다. 입력의 경우에는 시각, 청각, 촉각,

후각, 미각의 오감을 기반으로 연구가 진행중이고, 현재에는 많은 연구들이 시각, 청각, 촉각을 중심으로 진행되고 있다 [7][8][9][10].

그러나 ‘지능형 서비스’ 로봇의 시장이 확대됨에 따라 증가하는 복잡성은 ‘관리의 어려움’을 야기하고 있다.

따라서 본 절에서는 ‘지능형 서비스 로봇’의 강건성을 위한 연구 분야 중 ‘자가 적응 방법’에 필요한 관련 연구들을 소개하고 부분적으로 연구에 적용된 기술을 설명한다.

2.1 오류 이벤트 모델을 이용한 모델 기반 진단 기법

진단 시스템들의 중요한 기능 중에는 하나인 오류(Fault)와 증상(Symptom) 사이의 원인 관계를 추론하는 것이다. “오류 이벤트 모델을 이용한 모델 기반 진단 기법 [11]”은 컴포넌트의 정상상태(normal state)에서 오류상태(faulty state)로 이르는 상태 전이를 오류 이벤트(fault event)라 정의하고, 오류 이벤트 모델(fault event model)기반의 진단 기법을 제안하였다. 이 기법은 오류 이벤트의 원인 관계를 제공하여 컴포넌트가 오류 상태에 이르는 이유와 방법을 설명하고, 오류 이벤트 모델을 구성하는 추론 시스템의 구조를 표현한다.

본 논문에서는 위의 기법을 확장하여 오류 이벤트 모델과 오류 트리를 이용하여 ‘지능형 서비스 로봇’의 오류의 원인을 추론하고 결과를 진단하는 오류 평가에 적용한다.

2.2 상태 기반의 자가 적응 기법

상태 기반의 자가 적응 기법[12]은 컴포넌트와 커넥터로 이루어진 소프트웨어 아키텍처

텍처 스타일에서의 상태 이상 탐지를 위한 방법을 제안하였다. 이 기법은 소프트웨어 아키텍처 내에 있는 비정상적인 동작을 컴포넌트 모니터와 시스템 모니터를 통해 감지한다. 컴포넌트 모니터는 컴포넌트 내에 있는 객체들을 감시하고, 시스템 모니터는 컴포넌트 사이에 있는 메시지 커뮤니케이션을 감시한다. 각 모니터들은 모니터링 되어야 할 객체의 명세서인 '상태모델'을 가지고 있다.

본 논문에서는 관련 연구에서 제안된 시스템 모니터와 컴포넌트 모니터를 부분적으로 확장하여 로봇의 동작을 감시하도록 했다. 이를 통해 오류를 감지하고, 오류의 위치를 인식할 수 있도록 적용한다.

2.3 컴포넌트의 신뢰성 측정 기법

컴포넌트 신뢰성 측정 방법론[5]은 오류 주입(fault Injection) 기법을 통해 시스템의 신뢰성을 측정하는 방법을 제안하였다.

오류 주입이란 소프트웨어의 신뢰성과 강건성을 테스트하는 방법으로, 발생 가능한 오류를 주입하여 소프트웨어가 오류에 대해 대처하는 방법을 분석하기 위해 사용되는 기법이다. 분석된 내용을 통해 소프트웨어에 존재하는 문제를 제거하고, 결함을 감내할 수 있는 다양한 방법들을 추가, 보완하여 강건한 소프트웨어를 개발하는 방법이다. 본 논문에서는 오류 주입 기법을 평가를 위해 적용한다.

3. 자가 적응 방법론

우리는 제안 사항을 적용하기 위해 선행 연구로 지능형 서비스 로봇을 제안했다[4]. 선행 연구로 제작된 로봇은 <그림 1>과 같이 EZ-Robomaster와 Lego Brick을 이용하여 사용자 애플리케이션과 이동 로봇, 웹

카메라, RFID 시스템으로 구성되어 1) 격자 형태로 구역화 된 RFID Tag를 이용하여 로봇의 위치를 인식하고, 2) 컴퍼스 센서를 이용하여 로봇의 방향을 인식한다. 3) 전방 초음파 센서를 이용하여 장애물 인식하여, 4) 사용자 애플리케이션을 통해 로봇을 제어한다.



<그림 1>시스템 구성도

본 절에서는 지능형 서비스 로봇의 오동작을 최소화하기 위해 여덟 단계로 구성된 자가 적응 방법론을 제안한다. 구체적인 각 단계의 목표는 다음과 같다.

3.1 <1단계> 오류 모델링

오류 모델링 단계는 목표 시스템의 정상 동작을 분석하여 발생 가능한 모든 오류를 추출하고 정의하여 자가 적응을 위해 요구되는 자료로 이용한다. 자가 적응 시스템 개발자는 경험에 의해 오류를 예측해야 하므로 시스템 개발에 앞서 적용할 목표 시스템의 정상 동작을 알아야 한다. 오류 분석 단계를 통해 추출된 오류들은 발생위치에 따라 구분되어 오류 모델(Fault Model)로 작성된다. <표 1>은 로봇에서 추출된 오류로

작성된 오류 모델의 일부분이다.

<표1> 오류 모델(Fault Model)

구분	오류	오류 상태
센서 노드	잘못된 이동	싱크 노드로부터 전송 받은 좌표에 도착하지 못함
	경로이탈 (스텝모터 비정상)	스텝 모터의 비정상적인 행동으로 직선 주행이 불가능
	이동불가 (RFID 비정상)	RFID Tag 가 고장 난 지역으로 해당 좌표로 이동 불가능
	전방 초음파 센서 비정상	초음파 센서가 정상적으로 동작하지 않아 전방에 장애물을 인식 못함
	잘못된 회전	컴퍼스 센서의 잘못된 값으로 인해 목표 방향으로 회전을 하지 못함
컴퍼스 센서 비정상	자기장 또는 센서 파손으로 인해 컴퍼스 센서가 정상적으로 동작하지 않음	
⋮	⋮	⋮

<표1>은 오류가 발생할 수 있는 위치, 즉 로봇(센서노드)과 사용자 애플리케이션(싱크노드)에 따라 구분 지었으며, 오류 상태를 추출하고 오류를 정의한다.

예를 들면, 센서 노드에서 발생할 수 있는 오류 중 ‘싱크 노드로부터 전송 받은 좌표에 도착하지 못하는 상태’는 ‘잘못된 이동(arrived location == abnormal)’으로 정의 할 수 있다.

3.2 <2단계> 오류 모델 분석

시스템은 어떤 오류상태에서 특정행동을 수행 하는 경우 다른 오류상태로 전이하게 된다. 따라서 오류 모델 분석 단계에서는 1단계에서 작성된 오류 모델을 기반으로 발생 원인이 되는 오류(Causative Faulty States)와 결과가 되는 오류(Resultant Faulty

State)로 관계를 정의한다.

이렇게 원인 오류상태가 결과 오류상태로 전이 되게 하는 특정한 행동을 오류 이벤트(Fault Event)으로 정의 하고, 최종적으로 오류 이벤트 모델(Fault Event Model)을 참정의 한다. 다음 <표2>은 오류 이벤트 모델을 정의해 놓은 것이다.

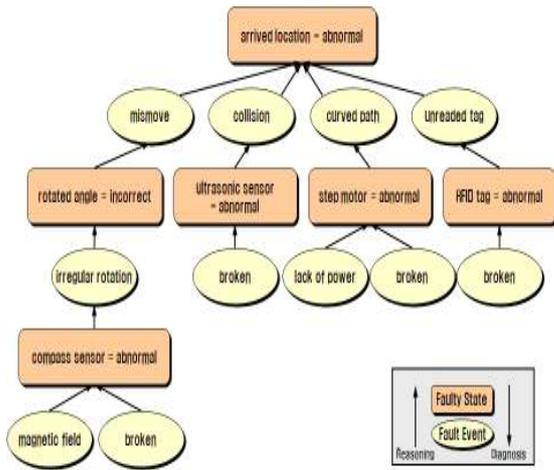
<표2> 오류 이벤트 모델

Fault Event	Causative States	Process	Resultant States
mismove	rotated angle == incorrect	move()	arrived location == abnormal
collision	front ultrasonic sensor == abnormal	move()	arrived location == abnormal
curved path	step motor == abnormal	move()	arrived location == abnormal
unread tag	RFID tag == abnormal	move()	arrived location == abnormal
irregular rotation	compass sensor == abnormal	turn()	rotated angle == incorrect
lack of power	-	move()	step motor == abnormal
magnetic field	-	turn()	compass sensor == abnormal
broken	-	move()	step motor == abnormal
		turn()	compass sensor == abnormal
⋮	⋮	⋮	⋮

3.3 <3단계> 오류 이벤트 분석

오류 이벤트 분석 단계에서는 오류 이벤트 모델을 참조하여 오류 상태(Faulty State)의 원인과 결과 상태에 따라 각 오류 상태들을 관계시킨다(부모 노드는 결과 상태이고 하위 노드는 원인 상태가 되도록 구성). 관계

된 오류 상태들은 하나의 트리와 같은 형태를 가진다. 오류 상태를 트리의 형태로 구성하는 이유는 오류 발생 원인을 파악하는 진단 과정을 수행할 수 있고, 어떤 오류가 발생할 수 있는지를 추론할 수 있기 때문이다. <그림 1>은 오류 이벤트 분석을 통해 수행된 오류 트리이다.



<그림 1> 오류 트리

예를 들어, 잘못된 회전(rotated angle == incorrect)오류가 발생한 경우 이것은 컴퍼스 센서로 인한 문제(compass sensor == abnormal)이며 이로 인해 잘못된 이동(arrived location == abnormal)을 유발하게 된다. 진단과 추론의 정확성은 2 단계에서 수행된 오류에 대한 분석이 세밀할수록 높아진다.

3.4 <4 단계> 오류 상태 해석

오류 상태 해석 단계에서는 로봇이 오류상태를 해석할 수 있는 조건을 정의하여 다음 <표 3>과 같이 지식 테이블(Knowledge Table)로 작성한다.

<표 3> 지식 테이블

Faulty State	Constraint
arrived location == abnormal	destination ≠ current_position
rotated angle == incorrect	(after turn) right_ultra_sonic ≠ left_ultra_sonic
front ultrasonic sensor == abnormal	changed_distance == 0 OR changed_distance > ULTRA_AVG
step motor == abnormal	(moving) before_angle ≠ current_angle
RFID tag == abnormal	destination ≠ current_position
compass sensor == abnormal	changed_angle > COMPASS_AVG
:	:

지식 테이블(Knowledge Table)은 3단계의 결과물인 오류 트리를 인식 할 수 있도록 정의해야 한다. 이를 위해서는 각 오류가 정상 상태에서 비정상 상태로 전이 되는 시점을 제약(Constraint)으로 정의해야 한다.

예를 들면, 목적 시스템의 이동 경로 중 목적지 좌표(Destination)와 현재 도착 목적 좌표(current_position)가 다를 경우 잘못된 이동(arrived location == abnormal)이 발생할 수 있다.

3.5 <5단계> 적응 행위자 분석

적응 행위자 분석 단계에서는 진단된 증상에 대한 적응행위자 및 전략을 정의하여 다음 <표4>와 같이 전략 테이블(Strategy Table)을 작성한다.

<표 4> 전략 테이블

오류 상태	적용 전략	행위자
arrived location == abnormal	UI 로 경로 재연산 요청	센서노드
rotated angle == incorrect	측면 초음파 센서로 벽과의 평행거리를 유지 후 이동	센서노드
front ultrasonic sensor == abnormal	전방 초음파 센서의 대체 센서로 채널 변경	센서노드
RFID tag == abnormal	이동불가 지역(Tag 고장)을 UI 로 좌표 전송 후 이동 경로 제외	싱크노드
compass sensor == abnormal	이동불가 지역을 UI 로 좌표 전송 후 이동 경로 제외	센서노드
:	:	:

4단계의 오류 상태 해석 단계의 결과물인 지식 테이블을 통해 목적시스템이 오류의 종류를 인식하게 되면, 발생한 오류를 해결하기 위해 적응 전략이 필요하게 된다. 이러한 오류 상태들의 적응전략을 정의한 것이 전략 테이블(Strategy Table)이다. 전략 테이블은 목적 시스템이 각 오류 상태에 따른 적응 전략과 적응의 행위자를 정의해야 한다. 예를 들면, 잘못된 이동(arrived location == abnormal)이 발생 할 경우 센서노드(로봇)가 “UI로 경로 재 연산 요청”을 한다는 적응 전략을 정의 할 수 있다.

3.6 <6단계> 오류 감지(Fault Detection)

1-5단계의 산출물인 테이블 및 오류 트리를 기반으로 시스템의 오류를 감지한다.

3.7 <7단계> 오류 원인 분석 및 감지

모니터링 단계에서 감지한 오류들에 대해 지식테이블을 참조하여 원인을 추론하고,

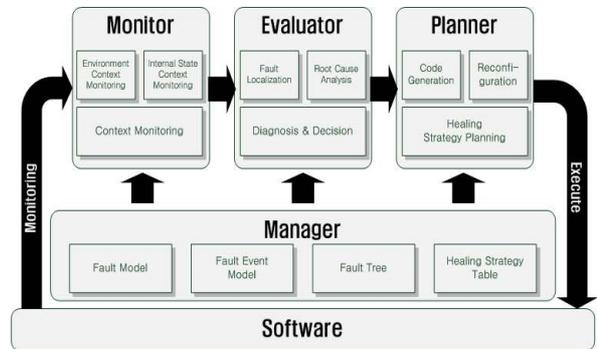
증상을 진단한다.

3.8 <8 단계> 자가 적응 실행(Execution)

원인에 따른 적응 행위자를 결정하고 적응 전략을 계획하여 실행한다.

4. 자가 적응 시스템 구조

제안 방법론을 수행하기 위한 로봇 내부에 내장된 자가 적응 시스템은 <그림 2>과 같이 구성된다.



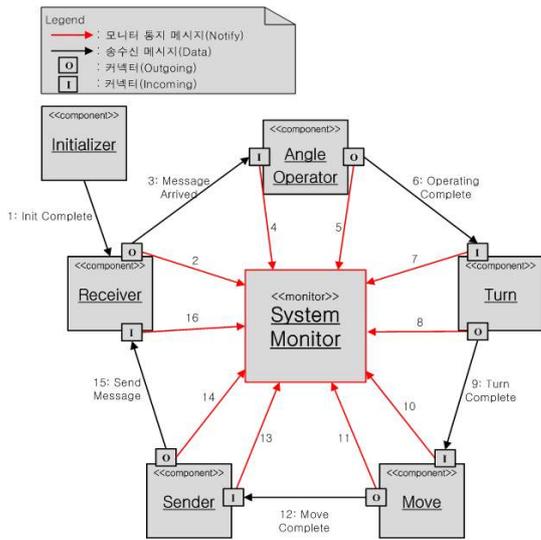
<그림 2> 내장된 자가 적응 시스템 구조

4.1 관리기(Manager)

관리기는 소프트웨어와 자가 적응 시스템 사이에 존재하고, 3.1절의 1-5단계에서 로봇으로부터 도출한 정상동작, 오류 테이블(Fault Table) 등을 저장하고 있다.

4.2 모니터(Monitor)

모니터는 관리기가 가지고 있는 오류 발생 조건 테이블(Fault Event Table)을 통해 감지하고, 평가기로 감지된 오류정보를 전송함으로써 6단계를 수행한다. 감지기(monitor)는 시스템 모니터와 컴포넌트 모니터로 구성된다. <그림 3>은 컴포넌트와 커백터를 이용한 시스템 모니터를 나타낸다.

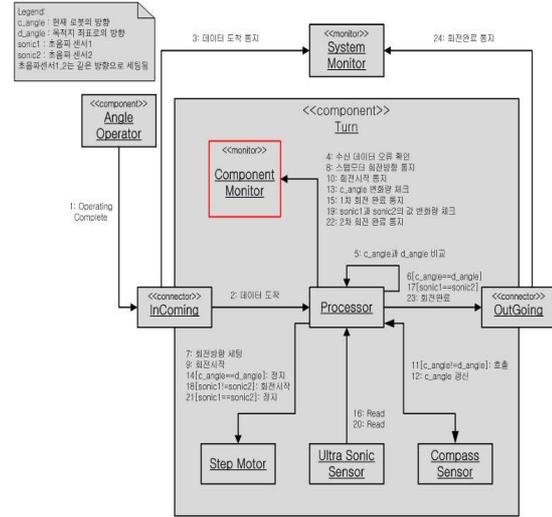


<그림 3> 시스템 모니터

시스템 모니터는 컴포넌트 간 상호작용 내의 이상은 컴포넌트 간 커넥터로부터 오는 통지 메시지를 이용하여 시스템 모니터에 의해 탐지된다. 송신 커넥터는 시스템 모니터에게 메시지가 전송되었음을 통지하고 수신 커넥터는 시스템 모니터에게 메시지가 수신되었음을 통지한다. 이를 통해 시스템 모니터는 수신된 통지 메시지들을 지식 테이블의 제약 조건과 비교하여 시스템의 비정상동작을 감지한다.

<그림 4>는 컴포넌트 내부 상태 감지를 위한 컴포넌트 모니터의 행동을 나타내고 있다. 컴포넌트 내부에는 태스크와 수동적 객체들로 구성된다. 컴포넌트 모니터는 각각의 컴포넌트 내부에 존재하여 태스크 및 수동적 객체들의 통신을 감시한다.

컴포넌트 모니터들은 로봇의 행동을 기술한 상태차트를 캡슐화 하여 보관하고 있으며, 제한시간 내에 상태의 전이가 정상적으로 이루어지지 않은 경우 비정상동작으로 인식하게 된다.



<그림 4> 컴포넌트 모니터

4.3 평가기(Evaluator)

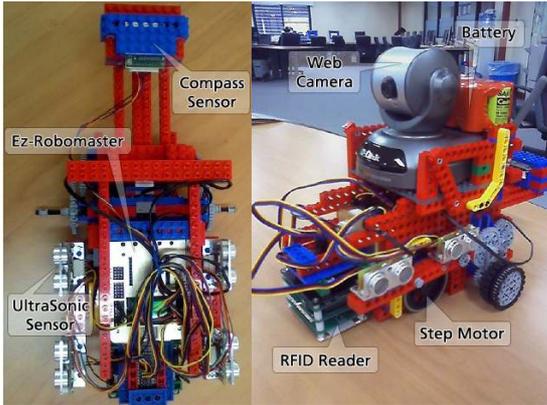
평가기는 감지기로부터 수신한 오류정보를 오류 테이블 및 오류 트리에 정의된 정보와 비교하여 오류의 종류를 분석, 진단하고 적응 실행 여부를 결정한다. 적응이 필요하다고 판단된 경우 계획자로 오류정보를 전송함으로써 7단계를 수행한다.

4.4 계획자(Planner)

계획자는 평가기로부터 수신한 오류정보를 적응전략 테이블(Healing Strategy Table)에 기반을 두어 적응전략을 결정하고, 코드를 생성하여 소프트웨어를 재구성 또는 사용자 애플리케이션으로 알린다. 이는 8단계에 해당한다.

5. 시나리오

테스트를 위해 우리는 로봇과 상호작용하는 사용자 애플리케이션을 구현하였다. <그림 5>는 사용자 애플리케이션을 나타내고 있으며, Visual Basic 6.0을 이용해 구현했다.



<그림 5> 로봇과 사용자 애플리케이션

구체적인 사용자 애플리케이션의 기능은 다음과 같다

1) Map: 마우스의 드래그나 클릭으로 로봇의 이동경로 설정할 수 있으며, 로봇의 현재 위치를 이미지로 표시한다. 또한, 장애물이나 자기장 영역 등 이동 제한구역을 표시하여 사용자에게 알려준다.

2) Monitor: 인터넷으로 웹 카메라에 접속하여 영상을 보여주고, 카메라의 줌 또는 방향을 제어한다.

3) Command: 십자버튼으로 로봇을 수동 조작할 수 있고, 로봇의 코드를 재인스톨하는 등 각종 명령 버튼이 있다.

4) Log: 로봇의 이동경로, 시작/종료 시간, 오류의 종류 및 발생 시각, 치유 결과 등을 기록하고, 저장하여 관리한다.

5) State Monitor: 로봇 센서들의 상태를 오류의 심각한 정도에 따라 색상으로 표시한다. 센서를 클릭하면 해당 센서의 오류 발생 정보를 확인할 수 있다.

5.1 전방 초음파 센서 비정상

정상 이동 중인 로봇에 고의적으로 초음파 센서가 정상적인 값을 읽을 수 없도록 하여, 장애물을 인식하지 못하도록 한다. 로봇의 이동 중 초음파센서가 읽어 들이는 값의 변화량을 측정 중인 모니터가 ‘초음파센서 비정상’ 오류를 인식하여 평가기로 오류 정보를 전송하고, 사용자 애플리케이션에게 오류의 발생을 알린다. 평가기는 오류 트리를 통해 오류의 발생 원인을 ‘파손’으로 진단하고, 계획자는 ‘파손’에 따른 치유 전략인 ‘대체 센서 사용’을 실행기로 전송한다. 실행기는 대기 중이던 초음파 센서로 채널을 변경하고 로봇은 정상적으로 장애물을 탐지할 수 있다. 대체 초음파 센서가 전방의 장애물을 정상적으로 인식하는 것을 확인하면 사용자 애플리케이션으로 적응의 성공을 알리고, 사용자 애플리케이션은 이를 상태 모니터에 표시하며, 로그 창에 치유 정보를 기록한다.

5.2 경로 이탈(스텝모터 비정상)

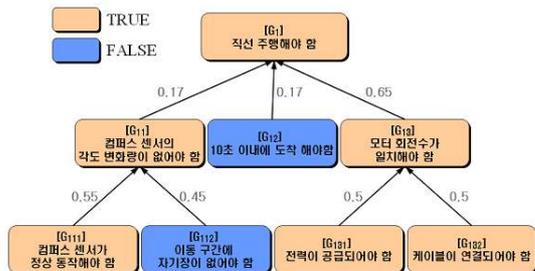
직진 이동 중인 로봇의 양쪽 모터 중 한 쪽 모터의 회전 수를 고의적으로 감소시켜 정상적인 경로에서 이탈 시킨다. 이동 중 컴퍼스 센서로 각도의 변화량을 측정하고

있던 모니터는 양쪽 모터의 회전수가 달라 경로가 휘어지게 되면 ‘스텝모터 비정상’으로 오류를 인식한다.

오류 정보를 평가기로 전송하여 사용자 애플리케이션으로 오류 발생 정보와 스텝 모터의 비정상 상태를 전송하게 된다. 평가기가 오류 정보를 전송 받으면 오류 트리를 통해 해당 오류가 발생한 원인을 ‘전력이 낮음’으로 진단하게 되고, 원인과 치유해야 할 오류의 정보를 계획자로 전송한다. 계획자는 오류의 원인에 따른 치유 전략을 ‘모터 회전 수 조절’로 결정하게 되고 이를 실행기로 전송하여 로봇이 직진할 때까지 모터의 속도를 조절하며 소프트웨어를 재구성하게 된다. 로봇이 직진 이동하게 되면 치유 전략의 성공을 사용자 애플리케이션으로 알리게 되고, 사용자 애플리케이션은 이를 로그 창에 기록하며 상태 모니터에 실시간으로 나타낸다.

5.3 목표 달성률 측정

목표 달성률은 ‘지능형 서비스 로봇’의 임무를 얼마나 달성했는지를 판단하는 확률이다. 이를 위해 본 연구에서는 <그림 6>과 같이 처럼 목표트리를 이용했다. 이것은 목표 시스템의 오류 중 ‘스텝모터 비정상 (step motor == abnormal)’ 이 적용되어 정상 상태가 됨을 증명할 수 있는 목표들이다.



<그림 6> 스텝모터 비정상에 대한 목표트리

예를 들어, 최종 목표 [G1]은 달성해야 하는 세부 목표 G[11], G[12], G[13]을 가진다. 링크에는 상위노드에 대한 하위노드의 기여도를 결정한다. 이것은 테스터가 판단할 수 있거나 시스템 자가 평가할 수 있는 목표들로 작성할 수 있다. 목표 달성률의 측정은 우리의 선행 연구에서 사용된 수식을 이용한다[13].

$$(1) GA(R) = \sum_n \{DR(L_n, R) \times GA(L_n)\}$$

(GA: Goal Achievement-rate)

(L : Leaf node)

$$(2) DR(G_x, G_y) = G_x' \text{'s DP} \times G_{\lfloor \frac{x}{10} \rfloor}' \text{'s DP} \times G_{\lfloor \frac{x}{100} \rfloor}' \text{'s DP} \times \dots \times G_y' \text{'s DP}$$

(DR: Dependency on Root)

(DP: Dependency on Parent)

만약 [G12], [G112] 두 목표가 실패 했다면 다음과 같이 목표 달성률이 측정되고 이것의 의미는 74% 만큼 로봇이 임무 수행을 했음을 나타낸다.

$$\begin{aligned} DR(L_1, R) &= DR(G_{111}, G_1) = 0.55 \times 0.17 \times 1 = 0.0935 \\ DR(L_2, R) &= DR(G_{112}, G_1) = 0.45 \times 0.17 \times 1 = 0.0765 \\ DR(L_3, R) &= DR(G_{12}, G_1) = 0.17 \times 1 = 0.17 \\ DR(L_4, R) &= DR(G_{131}, G_1) = 0.5 \times 0.65 \times 1 = 0.325 \\ DR(L_5, R) &= DR(G_{132}, G_1) = 0.5 \times 0.65 \times 1 = 0.325 \\ GA(L_1) &= GA(G_{111}) = 1 \\ GA(L_2) &= GA(G_{112}) = 0 \\ GA(L_3) &= GA(G_{12}) = 0 \\ GA(L_4) &= GA(G_{131}) = 1 \\ GA(L_5) &= GA(G_{132}) = 1 \\ \therefore GA(R) &= (0.0935 \times 1) + (0.0765 \times 0) + (0.17 \times 0) + (0.325 \times 1) + (0.325 \times 1) = 0.7435 \end{aligned}$$

$$\therefore \text{목표달성률} = 0.7435 \times 100 \approx 74(\%)$$

6. 결론

본 연구는 지능형 서비스 로봇의 강건성을 높이기 위한 자가 적응 방법론을 제안하였다. 제안 방법들을 통해 1) 오류 발생 위치를 분석할 수 있었고, 2) 오류 발생의 원인을 추론할 수 있었다. 또한 3) '지능형 서비스 로봇'의 목표 달성률을 측정하는 것이 가능했다. 그러나 오류 분석 및 치유 전략 정의 과정을 개발자가 경험에 의해 수행해야 하는 부하가 매우 크며 예측하지 못한 오류를 감내하지 못하는 문제점이 있다.

따라서 목표 시스템의 정상동작과 비교하여 비정상적인 동작을 감지하고 이를 오류 트리로 자동 확장 하는 방안 에 대해서는 향후 연구로 나타내며, 오류정보 및 테스트 결과 등 기록된 오류정보 데이터들을 통해 진단 가능한 자가 치유 방법으로 확장할 계획이다.

참고문헌

- [1] 김영욱, 이종배, 백준기, “지능 로봇틱스 영상처리 기술동향”, 정보과학회지 제26권 제4호, pp.5-11, 2008.
- [2] 지식경제부, “제1차 지능형로봇 기본계획”, <http://www.mke.go.kr>, 2009.
- [3] IBM Thomas J. Watson Research Center, "The Vision of Autonomic Computing, IEEE Computer, vol 36, number 1, pp. 41-50, Jan. 2003.
- [4] 고재현, 박정민 “Ez-Robomaster를 이용한 네트워크 기반의 서비스 로봇 시스템의 설계와 구현” 한국정보과학회 제 35회 추계학술대회 논문집 2008.
- [5] Leslie Cheung, “Early Prediction of Software Component Reliability”, In Procs. of ICSE 08, 2008.
- [6] 김재선, 박수용 “자기적응형 소프트웨어를 위한 목표 기반의 외부상황 평가 기법”, 정보과학회논문지, 제 33권 제 3호, pp.316-334, 2006.
- [7] 임수정, 홍진혁, 조성배 “대화를 기반으로 한 인간-로봇 상호작용”, 정보과학회지 제26권 제4호, pp.30-35, 2008.
- [8] S.D. Lowe, et al., “Vision based global localization and mapping for mobile robots, IEEE Trans. Robotics, vol.21, No.3, pp.364-375, 2005.
- [9] J.H. Hong, et al., “Mixed-initiative human-robot interaction using hierarchical Bayesian networks”, IEEE Trans. System, Man and Cybernetics, vol. 37, no.6, pp.1158-1164, 2007.
- [10] H.Yang, et al., “Gesture sotting and recognition for human-robot interaction”, IEEE Trans. Robotics, vol.34, pp.256-270, 2007.
- [11] Y. Kitamura, "A Model-based Diagnosis with Fault Event Models", Procs. of Pacific Asian Conference on Expert Systems, pp.322-329, 1997.
- [12] Michel E. Shin, "Detection of anomalies in software architecture with connectors" Journal of Science of Computer Programming, vol2, pp.233-442, 2006.
- [13] 민동욱, 박정민 “목표 트리 기반의 자가 치유 시스템 평가 기법” 한국정보과학회 제37회 추계학술대회 논문집 2009.