

상태감시컴포넌트를 사용한 OPRoS 프레임워크의 고장감내 기법

State-Monitoring Component-based Fault-tolerance Techniques for OPRoS Framework

안 희 준*, 안 상 철
(Heejune Ahn¹ and Sang Chul Ahn²)

¹Seoul National University of Technology

²Korea Institute of Science and Technology

Abstract: The OPRoS (Open Platform for Robotic Services) framework is proposed as an application runtime environment for service robot systems. For the successful deployment of the OPRoS framework, fault tolerance support is crucial on top of its basic functionalities of lifecycle, thread and connection management. In the previous work [1] on OPRoS fault tolerance supports, we presented a framework-based fault tolerance architecture. In this paper, we extend the architecture with component-based fault tolerance techniques, which can provide more simplicity and efficiency than the pure framework-based approach. This argument is especially true for fault detection, since most faults and failure can be defined when the system cannot meet the requirement of the application functions. Specifically, the paper applies two widely-used fault detection techniques to the OPRoS framework: 'bridge component' and 'process model' component techniques for fault detection. The application details and performance of the proposed techniques are demonstrated by the same application scenario in [1]. The combination of component-based techniques with the framework-based architecture would improve the reliability of robot systems using the OPRoS framework.

Keywords: OPRoS (Open Platform for Robotic Services), fault tolerance, fault detection, bridge pattern, process model

I. 서론

서비스 로봇의 활성화에 있어서 프레임워크 기술은 매우 중요하다. 이는 한정된 하드웨어와, 특화된 작업을 수행하는 산업용 로봇과는 달리, 서비스로봇은 다양한 하드웨어 환경에서 다양한 응용서비스의 지속적인 제공이 가능해야 하기 때문이다. 이를 위해서는 UCC 형태의 광범위한 응용 개발자의 참가가 가능하도록 하여야 하며, 따라서 기술적인 측면에서 응용개발과 응용결과들의 통합이 쉽고 효과적으로 할 수 있는 환경이 필요하다. 참고로 최근 모바일시장의 웹스토어의 성공은 Apple사의 iPhone이나 Google사의 Android가 제공하는 프레임워크기술이 없이는 기술적으로 불가능한 것이다.

이러한 필요성으로 인하여, 해외에서는 마이크로소프트의 MRDS (Microsoft Robotics Developer Studio)[2], OMG의 RTC (Robot Technology Component)[3], 미국 스탠포드 대학에서 개발한 ROS (Robot Operating System)[4] 등 다수의 프레임워크가 개발되어 있다. 국내에서도 로봇 플랫폼과 표준의 중요성을 인식하고 2005년부터 로봇관련 표준화 사업을 진행하고 있으며[5], 2008년부터는 이러한 내용이 통합되어, OPRoS (Open Platform for Robotic Service) 라는 이름으로 서비스 로봇의 플랫폼과 프레임워크 개발을 위한 국가적인 연구개발 사업을 진행 중이다[6].

현재 서비스 로봇 프레임워크는 재사용성 강화가 가장 중

요한 요소이기 때문에, OPRoS를 비롯한 거의 모든 프레임워크는 컴포넌트 기반 소프트웨어 공학(component-based software engineering)적 구조에 바탕을 두고 있다. OPRoS는 현재 동적 컴포넌트 관리, 실행 관리, 컴포넌트간 통신 방법 등 기본적인 기능들이 표준화되었고[7], 이에 따른 참고구현[6]도 공개되었다. 그러나 이러한 OPRoS 프레임워크가 시장에서 활용되기 위해서는 실시간성지원을 위한 QoS 기술, 시스템 보안기술, 그리고 신뢰성확보를 위한 테스트 및 실시간 고장안전 기술들의 개발이 필요하다.

특히 로봇의 고장 안전성에 대한 기술은 응용 개발의 용이성 측면과 더불어 간과될 수 없는 중요 기술요소이다. 2004년 개봉한 I, Robot 이라는 영화를 보면 서비스로봇이 가질 수 있는 위험성에대한 인간의 공포심이 잘 나타나 있다. 그러나 앞서 언급한 국내외의 프레임워크는 고장에 대한 고려는 거의 찾아볼 수 없으며, 있더라도 아주 기초적인 수준이다. 예를 들어 SOAP에 기반하여 설계된 MRDS의 경우 고장은 서비스를 요청한 컴포넌트에 특화된 리턴의 형태로 지원하는 것이 전부이다.

이에 본 연구자들은 연구[1]에서 OPRoS 프레임워크에서 고장의 유형과 고장감내(fault-tolerance)를 제공하기 위한 방안을 제시하였다. 그 연구에서는 주로 시스템 구조에 집중하여, 컴포넌트를 블랙박스로 보고 통합 시 프레임워크가 고장감내를 제공하기 위한 방안에 대하여 제시하였다. 그러나, 연구결과를 공개하고, 구체화된 실험을 적용을 진행하는 과정에서 고장판단 자체가 응용에 의존적이기 때문에, 응용에 종립적인 프레임워크에서는 지원이 어려운 부분이 존재한다는 점이 부각되었다[8]. 이에 본 논문에서는 이를 보완하고 개선하는 연구로 컴포넌트레벨에서 고장안전을 보장할 수 있는 방안, 특히 고장검출에 대한 방안에 대하여 제안한다.

* 책임저자(Corresponding Author)

논문접수: 2010. 4. 16., 수정: 2010. 6. 4., 채택확정: 2010. 6. 10.

안희준: 서울산업대학교 제어계측공학과(heejune@snut.ac.kr)

안상철: 한국과학기술연구원 영상미디어센터(asc@imrc.kist.re.kr)

※ 본 연구는 지식경제부 전라기술개발사업 지능형 로봇 개발을 위한 공통기반 플랫폼 기술개발 (OPRoS) No-10030826의 지원을 받았음.

※ 본 논문은 2010년도 ICROS 학술대회에서 초안이 발표되었습니다.

서론에 이어 제 II 장에서는 고장감내를 지원하기 위하여 프레임워크 기반 지원방법과 컴포넌트 기법이 가지는 차이점에 대하여 설명한다. 제 III 장과 제 IV 장에서는 브리지(bridge) 컴포넌트와 프로세스 모델 컴포넌트의 두 가지 컴포넌트 고장 검출기법을 제안한다. 제 V 장에서는 제안된 방식의 실효성을 검증하기 위해 실험로봇에 적용한 결과를 제시한다. 제 IV 장에서는 본 연구의 의미와 현재 진행중인 확장내용을 기술한다.

II. 컴포넌트기반 fault-tolerant 지원방법

‘고장’이란 시스템이 설계나 요구한 범위 안에서 동작하지 않는 상황을 말한다. 컴퓨터 및 제어시스템에서 사용되는 대표적인 고장처리 방식에 대한 자료는 참고문헌[9-11]에 잘 정리되어 있으며, 특히 Hammer는 [10]에서 고장처리 방법들을 패턴화하여 정리하고 있고, Isermann은 [11]에서 제어시스템과 같이 하드웨어적인 요소에서 발생하는 고장 처리 방법인 프로세스 모델에 대한 연구결과들을 정리하고 있다. 이러한 연구들을 바탕으로 연구 [1]은 서비스 로봇에서 발생할 수 있는 고장의 유형과 이에 따른 대표적인 처리방법을 자세히 정리하였다.

그림 1은 연구 [1]에서 제안한 고장 처리를 위한 시스템 골격이다. 고장검출기능은 포트와 실행기에서 담당하며, 고장분석과 고장처리는 고장관리자를 통하여 수행된다. 이때 고장관리자가 참고해야 하는 정보를 XML 기술자에 기술하여, 응용에 의존적인 부분을 해결하려고 하였다. 그러나 한편으로 실제로 고장을 검출하고 처리하는 데에는 해당 응용, 즉 OPRoS 상의 컴포넌트에 대한 구체적인 이해가 필요하다. 특히, 고장 처리의 3개의 단계인 고장검출, 고장분석, 고장대응 중에서 고장 검출의 경우에는 응용자체의 이해가 바탕이 되지 않고서는 검출이 매우 제한적이다.

본 연구에서는 컴포넌트의 고장 영향 분석과 고장 대응에 대해서는 기존연구에서의 방법을 그대로 유지하는 것으로 가정하였다. 본 연구에서는, 고장 검출을 위한 방법으로 대상 응용컴포넌트를 둘러싸는 브리지 컴포넌트의 사용과, 입출력 데이터를 바탕으로 고장 증상을 검출하도록 도와주는 비교

모델 컴포넌트 기법을 추가적으로 적용하였다.

이러한 컴포넌트기반 고장 검출기법은 다음과 같은 장점이 있을 것으로 판단된다. 컴포넌트 레벨에서 우선 프레임워크 보다 좀더 가까운 곳에서 지원을 하게 되므로 지원속도가 향상될 수 있다. 또한 컴포넌트마다 다른 형태의 코드를 생성할 수 있으므로 컴포넌트의 특성을 반영하기가 용이해진다. 프레임워크 기반의 경우에는 고장감내 기능은 XML에 기반한 기술자를 통하여 표현이 되고 있는데, 일반적인 프로 그래밍 언어로 표현하는 것에 비해서는 상당히 표현력이 제한적이다. 반면, 통합 개발자가 환경에 맞도록 코딩을 해줘야 하므로 사용하기가 불편하고, 컴포넌트와 고장감내 기법에 대한 상세한 지식이 필요하다. 또한 중복적이 코드가 생성되며, 이 브리지 컴포넌트 자체에 오류에 대한 고려가 불가능하다.

III. 브리지 컴포넌트 기반 고장 검출 방법

브리지기반 고장검출 기법을 OPRoS 프레임워크에 적용하기 위한 방안을 설명한다. 브리지 패턴은 ‘프락시(proxy) 기반 기법’이라고도 불리며 기존의 컴포넌트를 수정하지 않고 추가적으로 새로운 기능을 확장하기 위한 목적으로 사용된다 [11]. 브리지기반 고장감내 지원방법은 복합 컴포넌트작성 기법을 사용하여, 기존에 고장감내 기능이 지원되지 않는 컴포넌트를 확장하는 역할을 한다.

OPRoS 컴포넌트 표준은 컴포넌트 모델을 C++클래스 형태로 정의하고 있으며, 컴포넌트는 단위(atomic) 컴포넌트와 복합(composite) 컴포넌트로 구분된다. 복합 컴포넌트는 주로 하나이상의 단위 컴포넌트를 합쳐서 하나로 묶어서 외부에 단순한 인터페이스를 제공하는 역할을 한다. 또한 본 연구에서 브리지 방식은 컴포넌트의 입출력이 모델이 가능한지 여부에 따라서 규칙 기반 방식과 선출기반방식으로 나눈다.

1. 규칙기반 에러 검출 방법

규칙기반 에러 검출 방법은 입출력 값이 정상적이라고 정해진 규칙을 준수하는지 브리지 컴포넌트가 검사를 하는 방식이다. 그림 2(a)는 규칙기반 에러 검출 방법을 지원하는 복합 컴포넌트의 구성을 도시한다. 적용 가능한 예로서 최근 자동차 급발진의 원인으로 의심되는 현상을 예방하기 위하여, 자동차의 페달의 눌림 정도를 입력 받는 센서에서, 정상적인 상화에서 사람이 누를 수 있는 센서범위를 (예를 들어 +0.5 ~ 5 volt)를 정의하고, 이 범위 밖의 경우나 지나치게 갑작스러운 변화는 오류, 즉 단락이나 개방이나 노이즈가 발생한 것으로 판단한다.

구체적으로, 이 방식을 OPRoS에 적용하는 경우 대상 컴포넌트의 입출력을 브리지 컴포넌트를 통하여 전송되므로 브리지 컴포넌트의 onExecute() 나 onEvent()에서 해당포트 입출력 시에 원하는 규칙에 해당하는 다음과 같은 코드를 포함함으로 가능하다.

```

data = port.pop();
if (checkValid(data)){
    faultDetected(data);
}else{
    extPort.push(data);
}
    
```

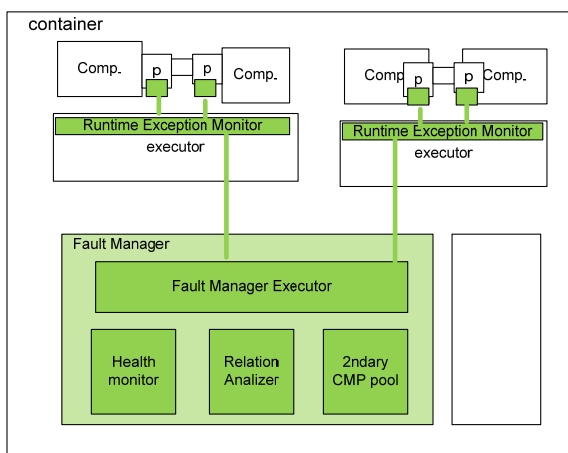


그림 1. 고장감내지원 OPRoS 프레임워크 구조[1].
 Fig. 1. The fault tolerance support OPRoS framework architecture [1].

프레임워크 기반 방식에서는 입출력 포트에 감시기능을 추가하고 허용 범위를 컴포넌트 상세서(component profile)에 기술하도록 하였다. 이 경우 간단한 수치 입력이고 간단한 최대치 최소치정도의 계산으로 물이 정해진다면 효과적일 수 있다. 반면 물이 복잡하고 입출력이 정수나 실수 등 수치 값이 아닌 경우에는 복합 컴포넌트를 구성하는 것이 보다 효과적이다. 또한 자체 컴포넌트가 입출력값을 직접 검사할 수도 있는데, 이 경우는 통합 시 마다 코드를 수정하거나, 설정이 가능하도록 구현하는 번거로움이 발생한다.

2. 무모델(modelless) 기법 지원

앞선 정의에서 고장이란 시스템이 설계나 요구한 범위 안에서 동작하지 않는 상황이라고 했으나, 고장감내 와 관련하여 시스템이나 입출력 신호의 모델자체가 불가능하거나 실용적이지 못한 경우가 존재한다[9, 10]. 이런 경우 전형적인 해결 방법은 다양성을 이용한 후 선출(voting)을 통하여 다수의 결과를 따르는 것이다. 이를 위해서는 각 단위 컴포넌트 A1, A2, ..., An이 입력되고 그 결과를 판단해서 출력하는 것이다. 이때, 컴포넌트의 수는 홀수로 해야 하며, 일반적으로 액추에이터에는 적용이 어렵고, 주로 센서나 소프트웨어 컴포넌트인 경우 적용이 가능하다.

그림 2(b)에는 두 개의 보조 컴포넌트를 사용한 선출알고리즘의 적용 예를 보이고 있다. 입력을 사용하는 경우는 다수의 컴포넌트로 분배되어야 하므로 분배 컴포넌트를 사용하거나, 연결설정을 다중으로 해야 한다. 출력선출컴포넌트는 입력 결과값들에 평균, 중간치 값 등을 사용하여 최종출력을 구한다. 일반적으로 중간치 값이 선호되며, 이때 특정 범위를 초과하는 컴포넌트는 고장으로 검출할 수도 있다. 다음 코드는 선거 컴포넌트를 구할 때 필요한 onExecute() 나 onEvent() 콜백함수에서의 예이다.

```

results[nResult]= internalPort.pop();
if(nResult++>= numVoters){
    finalResult=getMajorityValue(results);
    exTPort.push(finalResult);
}
    
```

IV. 프로세스 모델 기반 고장 검출

프로세스 모델 기반 기법은 정상이라고 생각되는 모델의 특성과 실제 시스템의 동작을 비교하는 방법으로 자동차, 항공기 등 제어 시스템의 고장안전 기술에서 많이 연구, 활용되고 있다[9]. 세부적인 방법으로 들어가면 다양한 형태가 있으나 기본시스템구조는 유사하여, 시스템 모델과 시스템 상태를 추정하는 부분, 시스템 모델의 출력 또는 시스템 상태와 실제 시스템을 비교하는 부분으로 구성된다.

그림 3은 프로세스 모델을 OPRoS의 컴포넌트 모델에 반영하여 구성한 것이다. 그림 3(a)에서 보는 바와 같이 대상 응용컴포넌트의 입력이 모델컴포넌트에 입력되어야 하며 그 출력은 고장 판단 알고리즘 컴포넌트로 입력되어야 한다. 고장판단컴포넌트는 이 두 입력을 비교하여 고장판단 시 고장관리자에 알리게 된다. 이러한 연결관계는 응용상세서(application profile)의 포트연결을 통하여 표현이 가능하다.

최근 발표되는 프로세스모델 논문들에는 추정 알고리즘 컴포넌트는 고급 시스템 추정 알고리즘으로 구성되며 판단 시스템으로는 인공지능 알고리즘이 사용되고 있다. 그러나 본 연구의 대상인 서비스로봇의 경우 일반적인 경우는 이처럼 복잡한 기능이 필요하지 않을 것으로 생각되며, 시스템의 난이도가 증가하면 개발자의 어려움도 따를 것으로 보인다. 따라서 간단한 시스템 입력과 확정적인 출력결과를 예상하는 시스템의 경우에는 그림 3(b)와 같이 단순화된 구조도 같이 제안한다. 단순화된 프로세스 모델의 경우에는 모델 컴포

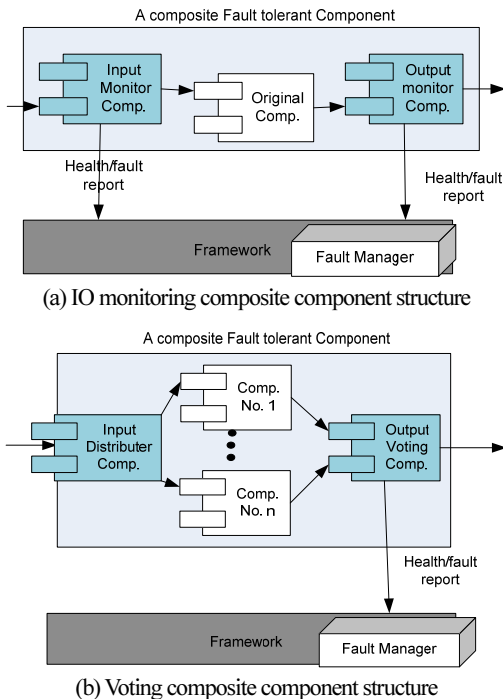


그림 2. 브리지기반 고장감내 확장 컴포넌트구조. Fig. 2. Bridge-based Fault-Tolerant extension.

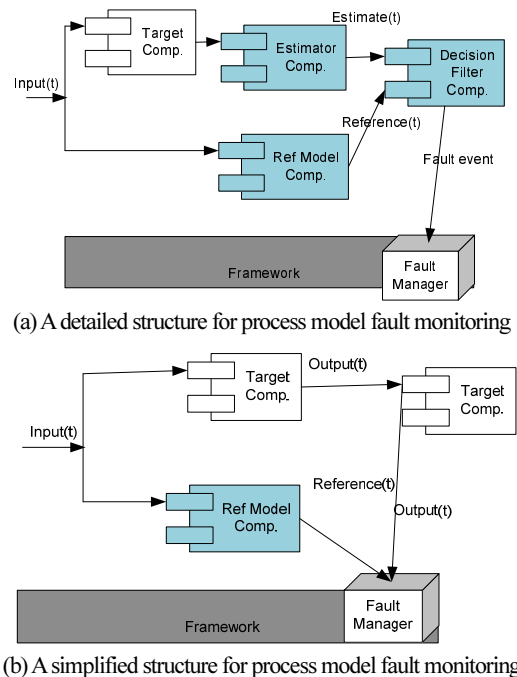


그림 3. 프로세스 모델 기반 고장 감지를 위해 제안된 구조. Fig. 3. The proposed architecture for process model fault detection.

넌트를 제공하고, 판단기능은 프레임워크에서 주어진 변수에 따라 지원하도록 한다. 변수는 출력 값의 평균치를 계산하는 측정 구간을 정의한 시간윈도우 T와 모델과 실제 센서 컴포넌트의 평균값의 차이가 임계치 D를 가지고 판단하도록 정의하였다. 이는 개발자에 따라 좀더 정교한 알고리즘으로 변경이 가능하다. T 값은 고장시의 응답속도를 D는 민감도를 나타내는 변수이므로, 이 두 값은 응용에 적합하도록 사용자가 설정할 수 있다.

V. 검증 시험

1. 검증 시험 환경

본 연구에서 제안된 기법들을 평가하기 위하여 [1]에서 사용한 응용시나리오를 기반으로 하였다(그림 4). 그러나 이 연구에 사용한 휴머노이드 타입의 로봇[13]은 로봇본체의 제어가 큰 명령(macro-command) 형태의 제어만이 가능하고, OPRoS가 탑재된 메인보드에서 모션보드로의 세부적인 동작 제어가 어려우며(즉, 단위 동작단위가 일보전진, 후진, 손들기 등 임), 센서 값의 수집 등이 어려운 점 등 본 연구에 적합하지 않아, 로봇의 본체를 자체적으로 새로 구성하였다. 그 규격은 표 1과 같다. 또한 하드웨어적인 고장을 모사하기 위하여 센서와 CPU간 및 모터 드라이버와 모터간의 점퍼 선을 통하여 연결되도록 하였으며, 고장 난 바퀴로 교환이 용이하도록 수정하였다.

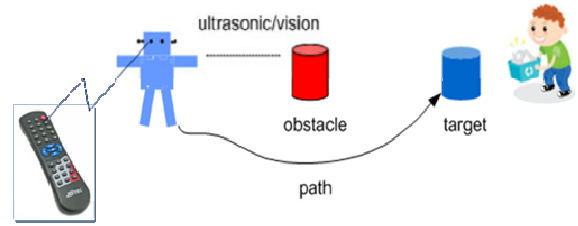
충돌방지 응용은 비전센서 (SensorVision) 컴포넌트와 물체를 검출하는 영상처리 컴포넌트 (AlgObjDetect), 그리고 이 데이터를 받아 방해물체들의 위치를 계산하고 이 결과를 경로계획 컴포넌트 (AlgPathPlan)로 전달하는 컴포넌트(AlgObsAvoidX/Y)로 구성하였다. 경로계획 컴포넌트는 비전센서 (SensorVision), 물체인식 (AlgObjDetect), 경로계획 (AlgPathPlan), 그리고 모터 구동 보드를 제어하는 4개의 바퀴에 해당하는 구동 컴포넌트 (ActDCmotor)로 구성하였다(그림 4(b)).

DC 모터 컴포넌트는 경로계획 컴포넌트로부터 명령/입력을 받아 시리얼 통신을 통해 Atmega128 CPU로 구성된 구동 모듈로 전달하며, 엔코더와 초음파 센서 값 수신한다. 4개의 바퀴를 각각 4개의 드라이버를 사용하여 PWM으로 제어한다. 이동거리는 역시 각각 엔코더 값을 이용하여 측정하며, 이동위치는 기본적으로 엔코더를 통하여 수행되지만, 초음파 센서에 의한 거리측정 변화치와 비전센서에 의한 변화 등도 최종 결과에 포함된다.

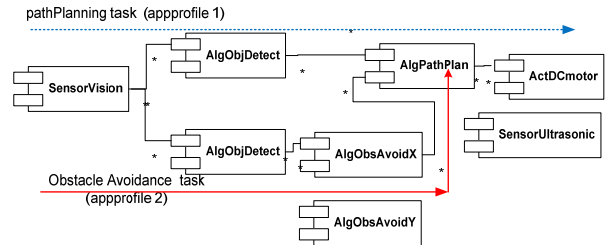
표 1. 실험에 사용한 로봇의 사양.

Table 1. System specification of test robot.

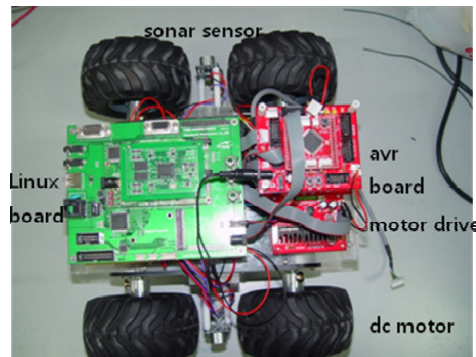
	구성	사양
메인 보드 [12]	CPU	PXA272-520MHz,
	메모리	16MB flash, 64MB dram
	OS	Linux 2.6.12 (WINCE 가능)
	카메라 입력	CCD, SAA7113H, FPGA 160x120 (up to NTSC)
로봇 바디	외형	1.3kg, 210x140x90mm
	서버모터	4개DC 모터, PWM 제어, Encoder
	CPU	Atmega128, C언어
	센서	4개의 초음파센서 (전후좌우)



(a) The scenario illustration



(b) Component and application task structure for robot navigation with obstacle avoidance.



(c) Test robot system

그림 4. 대표적인 테스트 시나리오.

Fig. 4. Test application scenario.

경로계획 컴포넌트로부터의 입력은(motion_type, parameter)로 구성된다. 여기서 각 원소의 의미는 각각, motion_type = 0 (정지), 1 (전진), 2 (후진), 3 (좌회전), 4 (우회전), parameter = 속도 (rpm) (전진/후진), 각도 (좌/우 회전)이다. 반대로 이동거리를 측정하기 위하여 엔코더에서 수신한 측정값 (enc_r1, enc_l1, enc_r2, enc_l2) 을 경로계획 컴포넌트로 출력한다. 여기서 'r/l'은 각각 right, left 를 1,2는 앞바퀴와 뒷바퀴를 의미한다.

2. 브리지 기반 고장검출 기법 시험

브리지 컴포넌트 기반 고장검출 방식의 구현 예를 보이고, 성능을 확인하기 위하여 물체검출 알고리즘 컴포넌트인 'AlgObjDetect'을 대상으로 시험하였다. 실제로 해당 영상처리 컴포넌트를 개발하면서 영상 처리시에 잡음이 심하거나 광원이 약해 물체를 찾지 못하는 경우, 물체의 위치의 위치를 화면의 경계 픽셀로 보고하는 오류가 있었다. 또한 물체가 시야를 벗어난 경우에도 결과 위치 값이 불규칙적으로 변하는 경우가 있었다. 이 예는 매우 단순한 예 같아 보이지만, 실제로 개발된 컴포넌트를 다른 환경에서 적용해보면 컴포넌트에서 흔히 발생하는 오류이다.

이러한 오류를 검출해내기 위하여 AlgObjDetect 컴포넌트의 입출력에 브리지 컴포넌트를 연결하고, 출력컴포넌트의 경우 입력 값이 실제 영상 영역인 (160, 120)에서 주변 값인

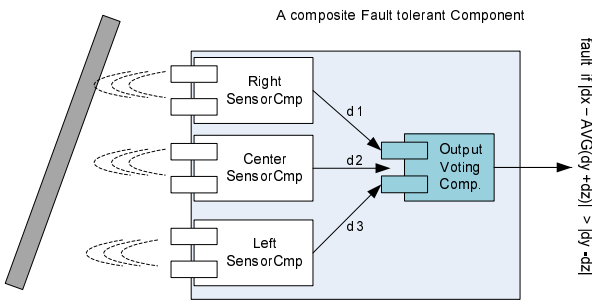


그림 5. 3중 초음파센서의 선출 복합 컴포넌트.
Fig. 5. A voting composite component with 3 sonar sensors.

(1~159, 1~119)을 넘어가는 경우는 데이터를 차단하였다. 또한 이전 값과 일정한 수치 이상 (예, 40 픽셀이상) 차이 나는 경우, 이 경우도 출력에서 제한하였다. 또한 일정기간 오류가 발생하는 경우 컴포넌트를 오류로 판단하였다. 실험 결과 오류가 있는 버전의 알고리즘을 사용한 컴포넌트의 경우에도 동작이 원활히 이루어짐을 확인하였다. 이러한 방식은 모터의 급가속을 제한하는 방으로 활용하게 되면 고장검출을 넘어서 시스템과 대인안전에도 도움이 될 것이다.

선출기능을 사용한 복합 컴포넌트방식은 현재 센서의 수가 제한적이고 시험에 적합한 형태로 실장이 되지 않은 관계로, 별도로 3개의 초음파센서를 자체의 앞면에 설치하고 시험하였다(그림 5). 3개의 초음파 센서는 차량 앞면의 좌측, 중앙, 우측에 장치하였다. 따라서 3개의 검출결과는 약간씩 차이가 있는 것이 자연스럽다. 실제 응용에서는 3개의 차이를 이용하여 물체가 어느 방향에 있는 지를 확인하는데 사용하였다. 이때 고장은 한 센서의 값이 다른 센서에서 측정된 값의 차보다 5배 이상 차이가 나면 오류로 판단하였다. 간이 테스트 결과 정상상태와 오류를 잘 확인하는 것을 확인하였다.

3. 모델 컴포넌트 기반 고장검출 기법 시험

모델 컴포넌트 방식의 고장검출 구현 예와 성능을 확인하기 위하여, ActDCmotor 컴포넌트의 입력과 출력을 모델의 입력과 출력으로 정의하고, 이에 해당하는 모델 컴포넌트를 구현하였다(그림 6). 구현의 편의상 플랫폼은 1차 시스템으로 가정하였다. 지연시간은 스텝응답실험결과 라이징(rising)시간으로 측정된 40 ms로 사용하였다. 현재 사용한 모델은 매우 간단하지만 많은 서비스로봇의 구동기에 대한 모델로도 충

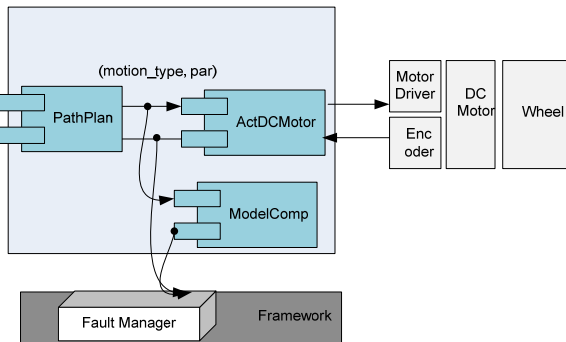


그림 6. DC 모터 엔코더 출력 비교를 통한 모델기반 고장감지 실험.
Fig. 6. Process model Test with DC motor Encoder measurement.

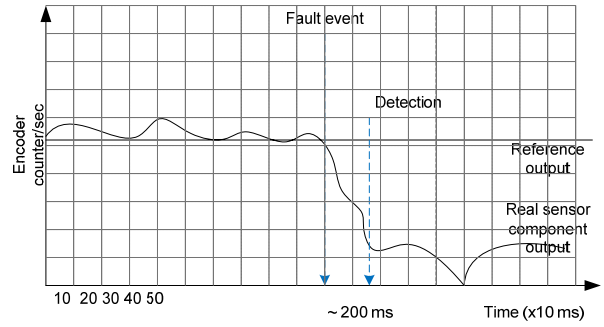


그림 7. 엔코더 출력 비교를 시험 결과.
Fig. 7. Experiment results from DC motor Encoder measurement.

분하다고 판단된다.

경로 계획 컴포넌트에서 50 rpm으로 입력을 한 상태에서, 회로연결 연결 점퍼선을 이용하여 바퀴의 구동부회로의 저항 값을 변경한 것을 보여준다. 점퍼연결 변경을 측정할 수 있도록 하였다. T = 200 ms, D = +/-50%을 사용한 경우 결과는 그림 7과 같이 약 200 ms (정확히는 180 ms) 에 감지되는 것을 확인하였다. 몇 가지 다른 실험 결과 고장 감지의 시간은 고장 발생 후 T시간 안에 검출이 가능한 것을 확인하였다.

마지막으로 이러한 고장검출내용은 프레임워크기반 물에 따른 고장영향분석과 대응에 따라 처리되도록 하였다. 즉, 컴포넌트 기반 고장 검출과 프레임워크기반 고장분석 및 처리가 함께 고장처리에 활용되었다.

VI. 결론

본 연구에서는 OPRoS 프레임워크의 고장감내를 제공하는 기존의 연구 [1]를 보완하는 접근 방법으로 브리지 패턴과 프로세스모델에 기반한 방법을 제안하고, 구체적인 적용 예에 대하여 설명 하였다. 이를 바탕으로 고장검출 컴포넌트를 구성하고, [1]에서 수행한 응용시나리오에서 고장 상황을 정의하여 실험하였다. 현재 많은 노력에도 불구하고, 사용자 입장에서 쓸만한 응용 컴포넌트 확보가 어려운 상황이다. 따라서, 본 연구의 결과의 응용 시험도 상당히 제한적이다. 앞으로 새로운 응용서비스에 적용된 결과들은 허용하는 한 공식 OPRoS 홈페이지 등에 공개를 할 계획이다.

제한된 방식을 사용하는 경우, 프레임워크기반 방식보다 지원속도가 향상될 수 있으며, 컴포넌트의 특성에 따른 코드를 생성할 수 있으므로 컴포넌트의 특성을 반영하기가 용이해진다. 반면 통합 개발자가 환경에 맞도록 코딩을 해주어야 하므로 사용하기가 불편하고, 컴포넌트와 고장감내기법에 대한 상세한 지식이 필요하다.

참고문헌

[1] H. J. Ahn, D. S. Lee, and S. C. Ahn, "OPRoS based fault tolerance support for reliability of service robots," *Journal of Institute of Control, Robotics and Systems*, vol. 16, no. 6, pp. 601-607, 2010.
 [2] Microsoft Robotics Developers Studio R2, [online] Available: <http://msdn.microsoft.com/en-us/robotics>
 [3] OMG, Robotic Technology Component Specification Version 1.0, April 2008.
 [4] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E.

- Berger, R. Wheeler, and A. Ng, "ROS: an open-source Robot Operating system," *Proc. of the International Conference on Robotics and Automation (ICRA)*, Kobe, Japan, May 2009.
- [5] 김현, 조영조, 오상록, "URC: 네트워크기반 서비스로봇," *정보과학회지*, 제24권 제3호, pp. 5-11, 4. 2006.
- [6] OPRoS project official site, <http://www.opros.or.kr/>
- [7] Korean Intelligent Robot Standard Forum, "OPRoS Component Spec," 2009.
- [8] H. Ahn and S. C. Ahn, "Component-based fault-tolerance extension techniques for OPRoS framework," *Proc. of ICROS Annual Conference 2010*, Chuncheon, Korea, pp. 142-143, May 2010.
- [9] I. Koren and C. M. Krishna, *Fault Tolerant System*, Morgan Kaufman Publisher, San Francisco, CA, 2007.
- [10] R. Hanmer, *Patterns for Fault Tolerant Software*, John Wiley and Sons, West Sussex, England, 2007.
- [11] R. Isermann, "Supervision, fault-detection and fault-diagnosis methods—An introduction," *Control Engineering Practice*, vol. 5, no. 5, pp. 639-652, May 1997.
- [12] H. Ahn, H.-J. Oh, and J. Hong, "Towards reliable OSGi operating framework and applications," *Journal of Information Science and Engineering*, vol. 23, no. 5, pp. 1379-1390, Sep. 2007.
- [13] 한백전자, HBE-Robonova-AI (Embedded Robot with Robonova Body), <http://www.hanback.co.kr/products/>



안희준

1993년 KAIST 전기및전자공학과(공학사). 1995년 KAIST 전기및전자공학과(공학석사). 2000년 KAIST 전기및전자공학과(공학박사). 1999년~2000년 독일 뉴른버그대학 박사 후 연구원. 1997년~2002년 LG 전자-정보통신(선임연구원).

2002년~2003년 Tmax Soft 연구소(책임연구원). 2004년~현재 서울산업대학교 제어계측공학과(부교수). 관심분야는 임베디드 소프트웨어, 영상 통신.



안상철

1988년 서울대학교 제어계측공학과(공학사). 1990년 서울대학교 제어계측공학과(공학석사). 1996년 서울대학교 제어계측공학과(공학박사). 1996년~1997년 미국 USC 방문연구원. 1997년~현재 KIST 영상미디어센터(책임연구원). 관심분야는 로봇공학, 비전 기반 HCI, 가상현실.

관심분야는 로봇공학, 비전 기반 HCI, 가상현실.