재구성 가능한 차량 임베디드 시스템의 유용도 분석

강민구 °*, 박기진*, 박수용**, 김성수**

*아주대학교 공과대학 산업정보시스템공학부
**아주대학교 정보통신전문대학원
e-mail : {ozige°, kiejin, cslab03, sskim}@ajou.ac.kr

Utility Analysis for Reconfigurable Vehicle Embedded Systems

Minkoo Kang°*, Kiejin Park*, Suyong Park**, Sungsoo Kim**

*Division of Industrial & Information Systems Engineering, Ajou University
**Graduate School of Information and Communication, Ajou University

요 약

임베디드 시스템의 신인도(Dependability)를 높이기 위해 기존 컴퓨터시스템에서 주로 사용되는 결함허용(Fault-tolerant) 기술을 그대로 적용시키는 것은 임베디드 시스템의 엄격한 비용 제약과 설치 공간의 부족 등의 이유로 적합하지 않다. 본 논문에서는 여분(Redundancy)을 최소 한도로 사용하는 차량 임베디드 시스템에 적합한 소프트웨어 결함허용 기법을 연구하였으며, 임베디드 시스템의 신인도를 반영할 수 있는 척도인 유용도(Utility) 척도를 정의하고, 임베디드 시스템의 결함허용을 위해 고려해야 할 각각의 재구성 조합에 대한 유용도 평가를 수행하였다. 이를 통해 차량 임베디드 시스템의 일부 부품이 결함 시, 가능한 최대의 유용도를 제공하는 구성조합으로 재구성 작업을 가능하게 하였다.

1. 서론

최근 컴퓨터시스템 사용자들은 점점 원하는 시간 내에 원하는 서비스를 받을 수 있는 가능성이 보장되는 고신인도(High Dependability: Reliability, Availability, Safety 등) 시스템을 요구하고 있으며, 이러한 요구는 기존 컴퓨터시스템은 물론 최근 각광 받고 있는 임베디드 시스템(Embedded System)에도 확산 적용되고 있다. 임베디드 시스템이란 미리 정해진 특정한 기능을 수행하기 위하여 하드웨어와 소프트웨어를 조합한 컴퓨터 제어 시스템을 말하며, 프로세서, 센서, 액츄에이터 및 소프트웨어 등으로 구성되어 있고, 분산성(Distributed), 실시간성(Real-time)과 같은 성격을 지니고 있다[1].

시스템의 고장(Failure)은 시스템에 내재하는 결함(Fault)으로부터 비롯되며, 미처 제거되지 못한 결함이 발현하여 오류(Error)를 야기시킨다. 오류가 지속적으로 발생할 경우, 결국에는 시스템 서비스 중지 상태에 이르게 되므로, 이러한 시스템 고장을 막기 위해서는, 근원적으로 결함을 다루어야 한다. 기존 컴퓨터시스템에서는 아래와 같은 4 가지의 대표적인 결함처리 기술이 사용되고 있다. 1) 결함방지(Prevention): 결함의 발생을 예방하는 기술을 말함. 2) 결함허용(Tolerance): 결함이 존재함에도 불구하고 올바른 서비스를 제공하게 하는 기술을 말함. 3) 결함제거(Removal): 결함을 줄이는 기술을 말함. 4) 결함예측(Forecasting): 시스템의 현재 결함 수 또는 앞으로 발생할 결함의 빈도를 예측하는 기술을 말함. 이들 중에서도 결함허용 기술은 오류를 찾고(Detection), 그것을 복구(Recovery)하는 방법을 사용하며, 앞서 말한 4 가지 기술 중에서 가장 활발히 연구/응용되는 분야이다[2].

기존의 컴퓨터시스템에 성공적으로 적용되었던 대표적인 결함허용 기법들은 하드웨어/소프트웨어적으로 여분(Redundancy)을 사용하는 것이었으나, 이들을 차량 임베디드 시스템에 그대로 적용시키기에는 여분의 하드웨어 및 소프트웨어를 설치할 공간, 비용 및 전원이 부족하다. 따라서 차량 임베디드 시스템의 결함허용 기술은 기존의 컴퓨터시스템에서 요구되었던 완전한 동작(Full Operation)으로의 복귀보다는 어느 정도의 성능 감퇴(Degraded Operation)를 허용하는 방향으로 설정해야 할 필요성이 있다[1].

여분 사용을 최소화하면서 시스템의 운영을 유지하기 위해서는, 결함이 발생한 시스템의 재구성(Reconfiguration) 작업이 필수적이며, 이 때 시스템의 가능한 모든 구성조합(Configuration)을 고려해야만 한다. 예를 들어 N개의 부품으로 이루어진 시스템의 가능한 구성조합의 수는 2^N개로서, 일정 시간 내에 결함복귀를 위한 특정조합을 찾는 것은 어려운 문제이다. 본 논문에서는 차량 임베디드 시스템을 구성하는 하드웨어/소프트웨어 부품들을 소프트웨어 중심의 Subset들로 나누고, 부품들의 고장에 따른 시스템 재구성을 위해 고려해야 할 축약된 구성조합의 수를 계산하였으며, 각 구성조합별 유용도를 분석하였다.

2. 관련 연구

여분 사용이 최소화된 임베디드 시스템의 하드웨어/소프트웨어 요소들을 재구성함으로써, 결함이 발생한 시스템의 성능을 우아하게 감퇴(Graceful Degradation)시킬 수 있다는 연구가 수행되었으며[3], 우아한 성능감퇴 개념은 1) 결함을 감지하는 작업, 2) 결함이 감지된 부품이 시스템에 미치는 영향을 제거(Isolation)하는 작업, 3) 나머지 부품들이 올바른 서비스를 지속적으로 제공할 수 있도록, 시스템을 재구성하는 작업이 동시에 이루어질 때 가능하다. 부품 결함으로 인한 시스템 재구성 작업은 적절한 하드웨어/소프트웨어 요소를 선택하여, 배정(Allocation)하는 것을 뜻하며, 이러한 문제는 NP-Complete 이기 때문에 발견적 기법(Heuristic)을 고안하여 해결해야 한다.

시스템의 일부 부품에 고장이 발생한 경우, 나머지 부품들이 고장이 발생한 부품의 기능을 어느 정도 대신할 수 있다는

- 본 연구는 ㈜ NGV의 " 2005년도 차세대 자동차 선행기술" 과제의 연구비 지원으로 수행되었음

기능적 대안(Functional Alternative) 개념을 제시하여, 임베디드 시스템의 신인도를 향상 시키는 연구에서는[4], 전체 시스템을 Feature Subset으로 나누어 묶은 다음, 각 Feature Subset별로 발생 가능한 구성조합의 수를 계산함으로써, 그 수를 현저하게 줄이는 방법($2^N \rightarrow m*2^k$, N은 전체 부품의 수, m은 Feature Subset의 수, k는 한 Feature Subset에서의 최대 부품 수)을 제시하였다. 또한 임베디드 시스템의 신인도와 성능을 동시에 확보하기 위해, 이들을 모두 고려할 수 있는 유용도 (Utility)를 정의하고, 각 구성조합 별로 유용도를 산출하였다. 하지만 이론적으로는 Feature Subset 개념이 가능할지라도 이를 실제 임베디드 시스템에 구현 시에는 각각의 Feature Subset 내의 부품들의 고장만을 고려할 수 있는 소프트웨어 요소가 반드시 필요하다. 게다가 크고 복잡한 시스템일수록 요구되는 기능이 증가함에 따라 Feature Subset의 수도 매우 증가하게 되므로 구성조합의 수 또한 지수적으로 증가될 가능성이 있다. 기능적 대안 기법이 실제로 적용되기 위해서는 각 시스템에서 결정론적(Deterministic)으로 Subset의 수가 정해질 필요성이 존재한다.

[5]에서는 임베디드 시스템의 신뢰도 최적화를 위해, 어떠한 부품의 여분을 얼마나 두어야 하는 가에 대한 방법이 제안되었고, 이를 위해 비용 제약하에서 시스템의 신뢰도를 최적화하는 4개의 모델과 신뢰도 제약하에서 비용을 최적화하는 1개의 모델을 설계하여 실험하였다. 각 모델 별로 시스템을 구성하는 하드웨어와 소프트웨어의 고장 확률을 이용하여 전체 시스템의 신뢰도를 산출하였으며, 신뢰도를 최적화하기 위해 적절한 하드웨어와 소프트웨어를 선택하는 방법으로 Simulated Annealing 알고리즘을 이용하였다.

실제 임베디드 시스템에 적용하는 데 있어서, [4]에서 제시한 Feature Subset 개념보다는, 본 논문에서 제시하는 소프트웨어 중심의 Subset 개념을 사용하는 것이 임베디드 시스템의 재구성 작업을 구현하기에 적합하다고 판단된다. 이에 본 논문에서는 시스템을 소프트웨어 중심의 Subset으로 나눈 다음, 시스템의 운영 시 발생 가능한 부품 고장에 대해 시스템 구성조합을 모두 파악하고, 각 구성조합을 대상으로 시스템을 재구성할 수 있는 소프트웨어 결함허용 기법을 연구하였으며, 시스템의 신인도를 나타낼 수 있는 척도로서, 각 구성조합에 대해 시스템 전체의 유용도 평가를 수행하였다. 본 논문의 3장에서는 시스템 모델을 정의하고, 모노레일 제어시스템을 통해 임베디드 시스템의 결함허용 방법을 설명하였으며, 4장에서는 샘플을 직접 제작하여, 제안된 방법의 성능 평가를 수행하였고, 5장에서는 결론을 내렸다.

3. 시스템 모델

본 논문에서 다루고자 하는 임베디드 시스템은 프로세서, 센서, 액츄에이터 및 소프트웨어들로 구성되어 있으며, 결함 발생에 대비한 하드웨어 및 소프트웨어의 여분은 없다고 가정하였다. 또한 임베디드 시스템을 구성하는 각 부품의 결함 발생에 대해서는 Fail-fast, Fail-silent하며, 부품 간 네트워크의 고장은 배제하고, 각 부품의 고장 발생 여부는 즉각적으로 감지된다고 가정하였다.

재구성 가능한 임베디드 시스템의 소프트웨어 결함허용 기법을 개발하기 위해, 1) 임베디드 시스템을 구성하는 각 부품을 정의하고, 2) 부품 간의 데이터 흐름을 파악하여 소프트웨어를 중심으로 한 Subset 를 분할 설계한 뒤, 3) Subset 별로 부품의 고장 여부에 따라 발생 가능한 구성조합을 모두 파악하였으며, 4) 각 구성조합 별로 시스템 유용도를 보장할 수 있는 재구성 시나리오를 설계하였다.

3.1 임베디드 시스템 모델링

임베디드 시스템을 구성하는 각 부품은 요구되는 목적에 따라 조금씩 차이가 있지만, 대체로 프로세서, 센서, 액츄에이터 및 소프트웨어로 구성된다(e.g., 차량 Anti-lock Breaking System을 구성하는 ECU, 휠 속도 센서, 액츄에이터). 정의된 각 부품의 데이터 흐름을 파악하여 Subset을 설계하기 위해, 임베디드 시스템을 구성하는 각 부품을 시스템 내에서 목표하는 소프트웨어 기능별로 분류하는 작업이 필요하다.

본 논문에서는 소프트웨어 중심의 Subset으로 분류된 부품의 고장 여부에 따라 Subset별로 발생 가능한 각 구성조합의 수를 산출하였으며, Subset으로 분류하기 이전보다 그 수가 줄어듦을 확인하였고, 이를 실제 수식으로 표현하면, Subset i 에서 고려해야 할 부품의 수가 k_i개이기 때문에, 해당 Subset에서 고려해야 할 구성조합의 수는 2^{k_i} 개이기 때문에, 전체 시스템의 구성조합의 수는 식 (1)과 같이 구할 수 있다.

$$\sum_{i=1}^{m}(2^{k_i}-1)+1=\sum_{i=1}^{m}2^{k_i}-m+1, \text{ m은 Subset의 수} \quad (1)$$

한편 Subset 이 목표하는 기능을 위해 반드시 동작해야만 하는 부품들이 고장일 경우에는 곧바로 Subset 이 목표하는 기능을 제공할 수 없게 되므로, 소프트웨어 결함허용 기법을 적용한다는 것이 사실상 무의미하다. 때문에 각 Subset 별로 고려해야 할 부품의 수, k_i 는 더욱 작아지게 되므로 전체 시스템의 구성조합의 수는 현저히 줄어드는 것을 확인할 수 있다.

3.2 유용도 함수

임베디드 시스템을 운영하는 도중, 일부 부품에 고장이 발생하게 되면, 시스템은 동작 가능한 부품들의 목록을 바탕으로 가능한 최대의 유용도를 가지는 구성조합으로 재구성 작업이 실시된다. 때문에 가능한 모든 구성조합 별로 유용도 평가가 선행되어야 하며, 이를 위해 다음에서 설명하는 3 단계를 거쳐 유용도 평가가 이루어진다. 1) 각 부품의 유용도 평가: 동작 여부에 따라 해당 부품은 0 또는 1 의 유용도 값을 가짐. 2) Subset 의 유용도 평가: 해당 Subset 이 목표하는 기능을 위해 반드시 동작해야 하는 부품들이 모두 동작한다면, 나머지 부가적인 부품들의 유용도 값($U_{Component\_1}$, $U_{Component\_2}$, … , $U_{Component\_k}$)과 Subset 의 유용도 함수(f_{Subset})에 의해 0 에서 1 사이의 값으로 평가됨. 3) 전체 시스템의 유용도 평가: 각 Subset 의 유용도 값(U_{Subset_1}, U_{Subset_2}, … , U_{Subset_m})과 시스템 유용도 함수(U_{System})에 의해 0 에서 1 사이의 값으로 결정됨.

3.3 모노레일 제어시스템 분석사례

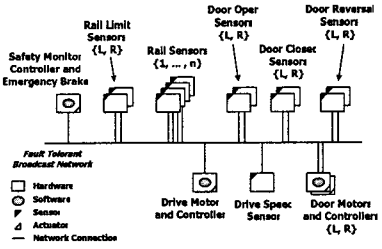

그림 2. 모노레일 제어 시스템의 구조도

본 절에서는 임베디드 시스템 모델링과 유용도 함수의 정확한 이해를 돕기 위해 모노레일 제어시스템을 사례로 제시하였다. 다루고자 하는 모노레일 제어 시스템의 구조는 그림 2에서 보는 바와 같이 프로세서, 센서, 액츄에이터 및 소프트웨어로 이루어져 있으며, 모노레일은 왕복 운행하고 있다. 본 시스템을 구성하는 전체 부품의 수는 모두 17+n(n은 Rail Sensor의 수)개이므로, 발생 가능한 모든 구성조합의 수는2^{17+n}이다. 하지만 일부 부품의 결함 발생 시에 모든 구성조합에 대해 재구성 알고리즘을 수행하는 것은 비현실적이다.

시스템을 구성하는 전체 부품들 중에서 시스템의 기본적인 목적을 수행하기 위해 필요한 최소한의 부품들(e.g., Drive

Controller, Emergency Brake Actuator)이 고장이 발생할 경우에는 전체 시스템이 곧바로 고장 상태에 도달하게 되므로, 이들의 고장을 소프트웨어 결함허용 기법의 고려대상에서 제외하면, 구성조합의 수는 대폭 감소된다.

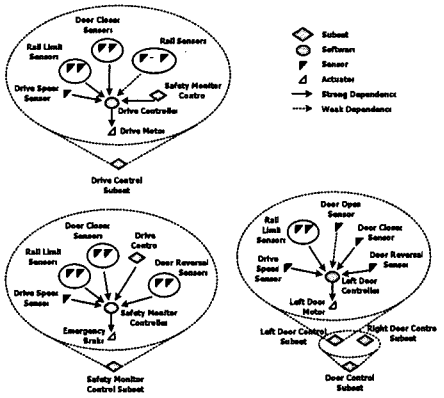

그림 3. 모노레일 제어시스템의 소프트웨어 중심 Subset 구조

한편 모노레일 제어시스템의 Drive Controller 소프트웨어는 모노레일의 원활한 주행을 위한 속도 데이터(Drive Speed Sensor)와 위치 데이터(Rail Sensors) 및 탑승자의 안전을 위한 데이터(Door Closed Sensors, Safety Control Subset)를 필요로 하므로, Drive Control Subset으로 분류할 수 있었으며, 마찬가지로 Safety Monitor Subset과 Door Control Subset을 도출할 수 있었다(그림 3). 이 과정에서 Subset이 목표하는 기능을 위해 반드시 동작해야 하는 부품들의 데이터 흐름은 Strong Dependence라고 표현하였고, 그렇지 않은 부품들의 데이터 흐름은 Weak Dependence로 표현하였다.

각 Subset 내에서 Weak Dependence 데이터 흐름을 발생시키는 부품들은 소프트웨어 결함허용 기법의 적용 대상이 되며, 각 Subset 별로 구성조합의 수를 계산하면 표 1에서 보는 바와 같이 그 수가 현저히 줄어듦을 확인할 수 있다. 예를 들어 Rail Sensors의 수를 4 라고 결정하면, 총 부품의 수는 21개로 고려해야 할 구성조합의 수는 2^{21}개에서 18개로 줄어든다.

표 1. Subset 별 발생 가능한 구성조합의 수

Subset	부품의 수	구성조합의 수	Subset의 복제 수	Total
Drive control	n	2^n-1	1	2^n-1
Door control {L, R}	1	2^1-1	2	2
Safety monitor	0	0	1	0
Total	n+1		4	$2+2^n$

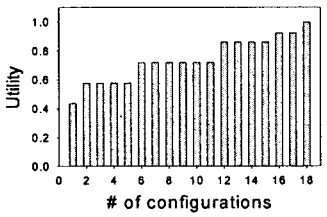

그림 4. 모노레일 제어시스템의 각 구성조합별 유용도

그림 4는 Rail Sensors의 수를 4 라고 가정했을 때, 앞 절에서 설명한 유용도 평가를 수행하여 얻은 그래프이다. 모든 Rail Sensors가 고장이 발생한 경우가 1번 구성조합이며, 18번 구성조합은 모든 부품이 정상 동작할 경우이고, 나머지는 Rail Sensors와 Door Open Sensor가 부분적으로 고장이 발생한 경우를 나타낸다.

4. 실험 및 성능 분석

임베디드 시스템의 결함 발생 시 재구성 작업을 통한 시스템의 유용도 확보를 실험하기 위해, Java 개발 환경에서 레고 마인드스톰(Lego Mindstorms)을 이용한 AGV(Automatic Guided Vehicle)를 제작하였다. 이 AGV에는 빛 센서 2개, 구동 모터 2개 및 RCX 2.0이 사용되었다(그림 5). 실험에서는 구동 모터의 초기 속력을 좌, 우 모두 2m/s로 설정하였으며, 빛 센서 2개가 모두 정상 동작 시의 평균 속력과 임의의 빛 센서 1개를 분리시켰을 때의 평균 속력을 비교하였다. 표 2에서 보는 바와 같이 AGV의 빛 센서 1개가 고장 시에도 다소 성능은 감퇴되더라도 목표하는 기능을 제공하는 것을 확인하였다.

표 2. 정상 동작하는 빛 센서의 수에 따른 AGV의 성능 비교

구분	빛 센서 2개	빛 센서 1개
평균 속력(m/s)	1.04	0.61
트랙 이탈율(%)	0.0	0.0

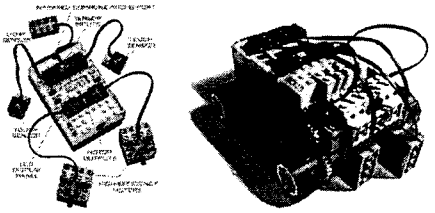

그림 5. 레고 마인드스톰을 이용하여 제작한 AGV

5. 결론

본 연구에서는 여분을 사용하지 않는 임베디드 시스템의 신인도를 향상시키기 위해, 소프트웨어 결함허용 기법을 이용하여 재구성 가능한 차량 임베디드 시스템의 유용도(Utility) 분석을 수행하였다. 이를 통해 소프트웨어 결함허용 기법의 적용 가능성을 확인하였으며, 추후에는 임베디드 시스템의 평균 유용도에 영향을 미치는 다양한 요인을 분석하여, 임베디드 시스템의 평균 유용도 최적화에 대한 연구를 수행할 예정이다.

참고 문헌

[1] 박기진, 김광섭, 최석호, " 고신뢰성 차량 임베디드 컴퓨팅 시스템의 백업 최소화 방안," 한국신뢰성학회 2005 학술발표대회 논문집, pp 295-301, June 2005.

[2] A. Avizienis, et al., " Fundamental Concepts of Dependability," Research Report N01145, LAAS-CNRS, Apr. 2001.

[3] W. Nace and P. Koopman, " A Graceful Degradation Framework for Distributed Embedded Systems," Workshop on Reliability in Embedded Systems, Oct. 2001.

[4] C. Shelton and P. Koopman, " Improving System Dependability with Functional Alternatives," 2004 International Conference on Dependability Systems and Networks, pp. 295-304, July 2004.

[5] N. Wattanapongsakorn and S. Levitan, " Reliability Optimization Models for Embedded Systems With Multiple Applications," IEEE Transactions on Reliability, Vol. 53, No. 3, pp. 406-416, Sep. 2004.