

풍력발전시스템에서 제어기를 위한 고 가용성의 FTSM 설계

김영환* **, 손재기*, 황태호*, 함경선*, 홍지만**

*전자부품연구원 RFID/USN 융합연구센터

**송실대학교 컴퓨터공학부

e-mail: yhkim93@keti.re.kr

A Design on FTSM for supporting High Availability in Wind Turbine Controller

Young-Hwan Kim* **, Jae-Gi Son*, Tae-Ho Hwang*, Kyoung-Sun Ham*, Jiman Hong**

*RFID/USN Conversions Research Center, Korea Electronics Technology Institute

**School of Computing, Soongsil University

요 약

IT 기술의 발전에 따른 전력 사용량 증가로 인해서 친환경 재생 에너지에 대한 관심이 집중되고 있는 상황으로 다양한 재생 에너지자원 가운데 풍력에너지 개발이 전 세계적으로 급속히 증가하고 있다. 그러나 풍력발전기를 제어하는 제어기의 소프트웨어-하드웨어적인 문제로 인해 불규칙한 에너지 생산과 유지보수를 위한 비용이 증가함에 따라 최근에는 풍력제어기의 고장분석 및 고장 감내 제어에 대한 연구가 활발히 진행되고 있는 상황이다. 본 논문에서는 이와 같은 풍력발전기가 지속적으로 일정한 에너지 생산과 유지보수 절감이 가능하도록 하기 위해 풍력제어시스템 차원에서의 고장에 대해서 감내 가능한 고 가용성의 미들웨어에 대한 구조를 제안한다. 풍력제어시스템의 고 가용성 미들웨어는 가상 운영체제를 기반으로 이중화 구조를 가지고 있으며, 주 운영체제에서 소프트웨어적인 고장이 발생 시 다음 우선순위의 가상 운영체제에서 지속적인 서비스를 제공할 수 있도록 각각의 가상 운영체제 환경을 관리한다.

1. 서론

IT 기술의 발전은 현대사회에서 전기에 의해 운용되는 장치의 전력 사용량을 폭발적으로¹⁾ 증가 시켜왔다. 이로 인해 화석연료에 대한 수요증가를 초래하게 되었고 전력 사용에 대한 비용 또한 증가하게 되었다. 그러나 전력에 대한 수요증가에 대응하기 위해 한정된 화석연료와 대기 오염의 주범인 탄소배출에 대한 환경적인 문제로 인해 친환경 재생 에너지에 대한 관심으로 집중되었다. 재생 에너지 가운데 풍력에너지는 전 세계적으로 급속히 증가하고 있고, 풍력발전 용량은 과거 10년 동안 연평균 30%이상 성장해 왔다. 이와 같은 성장은 저비용, 무한정, 청결, 친환경 등과 같은 많은 장점으로 인해 가능하게 되었다.

현재 국내뿐만 아니라 국외에서도 많은 풍력발전기가 바람이 많은 해안가, 산중에 설치되고 있는데 진동, 전·자기적 노이즈와 같은 환경적인 요인과 풍력발전기를 제어하는 제어기의 소프트웨어-하드웨어적인 문제로 인해 유지보수에 많은 비용이 지출되고 있는 상황으로 최근에는 풍력제어기의 고장분석(Fault Diagnosis) 및 고장 감내 제어(Fault Tolerant Control)에 대한 연구가 많이 진행되고 있다. 이와 같은 기술은 예상치 못한 풍력발전기의 동작으로 인해 발생할 수 있는 전력 생산 중단과 유지보수 증가 측면에 있어

일정한 전력생산과 비용절감을 위해 매우 중요한 기술이다.

기존에 연구되고 있는 고장분석 및 고장 감내 기술은 고장 감내 제어시스템(Fault-Tolerant Control System)을 기반으로 수동적 고장 감내 제어시스템(Passive Fault Tolerant Control System), 능동적 고장 감내 제어시스템(Active Fault Tolerant Control System)과 고장 분석(Fault Diagnosis)과 같이 구분할 수 있는데 고장 감내 제어시스템은 풍력제어기의 시스템 차원에서 발생한 문제로부터 전체 풍력발전기를 이루고 있는 각 구성요소에 대한 고장을 방지를 위한 기술을 의미한다. 이와 같은 고장 감내 제어시스템에는 주로 이중화 구조의 동일한 제어기 상에서 하나의 제어기(Master)에 고장이 발생했을 경우 또 다른 제어기(Slave)가 기존에 수행하던 시스템 차원의 일을 계속 수행하는 수동적 고장 감내 제어시스템이 있는데 강인 시스템(Robust System)은 아니지만 고 가용성의 제어시스템이라 할 수 있다. 그리고 수동적 고장 감내 제어시스템과 같이 이중화 구조이지만 서로 상이한 성능의 제어기로 구성되어 풍력발전기 전체를 제어·관리하는 감독관(Supervisor)차원에서 고장분석 알고리즘을 통해 고장을 분석함으로써 상황에 따라 능동적으로 제어기를 선택하여 해당 고장에 대해 감내할 수 있는 능동적 고장 감내 제어시스템이 있다. 마지막으로 고장 분석기술은 풍력제어기에서의 특정 고장에 대한 발견(Detection), 고립(Isolation), 예측(Estimation)이 가능한 알고리즘을 포함하는 기술로

지식경제부 신재생에너지기술개발의 “풍력발전시스템용 제어기술 및 기기 개발(2010T100200257)” 지원을 받아 수행된 연구임

앞서 설명한 능동적 고장 감내 제어시스템에 사용된다[1].

본 논문에서는 풍력제어기에서의 발생할 수 있는 다양한 고장에 대해 감내할 수 있는 풍력제어시스템에서의 FTSM(Fault-Tolerant Service Middleware)에 관한 것으로 앞서 설명한 수동적 고장 감내 시스템에서 활용될 수 있는 기술이다. 또한 실제 풍력제어기로 개발된 제어시스템을 동일한 제어기로 하드웨어적 이중화 구조를 구축한 상태에서 고 가용성 미들웨어를 통해 소프트웨어 차원에서도 이중화 구조로 가집으로서 매우 강인한 고 가용성의 고장 감내 제어시스템이라 할 수 있다. 제어기에서의 고 가용성 미들웨어는 단일 풍력제어시스템 상에서 호스트(Host) 운영체제를 기반으로 사용자 레벨에서의 다중 가상 운영체제 환경을 제공하게 되고 각 가상 운영체제에서는 개별적으로 응용 프로그램이 구동 된다. 그리고 사용자 레벨에서의 가상 운영체제는 각각 고립된 상태이기 때문에 소프트웨어적인 고장으로 인해 전체 풍력제어시스템의 고장으로 확대되지 않는다. 이와 같은 다중 가상 운영체제 구조를 기반으로 각각의 가상 운영체제에 우선순위를 제공함으로써 단일 제어시스템 상에서는 여러 가상 운영체제가 구동되지만 실제 풍력제어기에서 하나의 주 가상 운영체제(Master)에서 주요 프로그램을 수행하고, 주 가상 운영체제에서 고장 발생 시 다음 우선순위의 가상 운영체제가 지속적으로 해당 프로그램을 수행할 수 있는 고 가용성의 이중화 구조로 설계되었다.

본 논문의 2장에서는 풍력제어기에서의 고장 감내 제어기술과 관련된 기존 기술에 관해 분석하고, 3장에서는 본 논문에서 설계한 풍력발전기에서 제어기를 위한 고 가용성의 미들웨어에 대한 구조, 4장에서는 결론과 함께 향후 작업에 대한 설명으로 구성된다.

2. 기존 연구

풍력제어기에서의 고장 감내 기술은 주로 센서 고장(Sensor Faults), 액추에이터 고장(Actuator Faults), 풍력발전기의 구성요소 고장(Component Fault)에 따른 감내 기능에 대한 것이 대부분이며, 최근에는 풍력발전기의 상태 모니터링(CM: Condition Monitoring) 위한 DAQ(Data Acquisition) 시스템을 통해 풍력발전기의 상태를 관리하고 고장에 대한 감내 기능을 지원하는 기술에 대한 연구가 진행되고 있다. 그러나 CM 과 같은 기술은 풍력발전기에 있어 매우 중요한 기술임에도 실제 적용을 위한 검증이 필요한 상황이다. 이와 같이 다양한 풍력발전기에서의 고장 발견, 고장 감내, 상태모니터링에 관한 연구가 많이 진행되고 있지만 아직 풍력발전기 내의 제어시스템 자체에 대한 고장 감내 연구는 부진한 상황이다. 본 논문은 기존 풍력발전기의 고장 감내 기술과는 달리 풍력 제어기 자체의 하드웨어 이중화 구조 지원뿐만 아니라 호스트 운영체제를 기반으로 고 가용성 미들웨어를 통해 소프트웨어적으로 이중화 구조를 지원하도록 설계했다. 자세한 풍력제어기의 고 가용성 미들웨어에 대한 설명은 3장에 하도록 한다.

2.1 고장 감내 제어 기술

[2]에서 제안하는 고장 감내 기술은 능동적 고장 감내 제어 시스템에 관한 것으로 두 개의 분리된 제어기 구조를 갖는다. 풍력제어시스템이 정상적인 동작을 유지하고 있을 때 첫 번째 제어기에서는 아무런 값을 출력하지 않고 두 번째 제어기는 첫 번째 제어기와 동등하게 정상적인 동작을 유지한다. 이와 같은 상황에서 첫 번째 제어기에서 고장이 발생할 경우 제어시스템 내의 첫 번째 제어기에서는 '0' 이상의 값을 출력하게 되고 두 번째 제어기가 첫 번째 제어기의 역할을 대신한다.

2.2 고장 감지 및 분석 기술

[3]은 풍력발전기의 고장 감지에 대한 연구에 관한 것으로 주로 제어시스템 내에 내장되어 동작한다. 풍력 발전기의 상태 모니터링을 위해 반드시 필요한 파라미터를 구분하여 데이터를 수집 및 분석함으로써 SCADA 시스템으로 전송하게 된다. 모든 풍력발전기는 정상적인 동작을 유지하기 위해 다양한 센서, 액추에이터와 풍력제어기간에 제어/상태 관련 데이터를 송수신하게 된다. 풍력제어기를 통해 지원되어야 하는 주요 데이터들은 풍력발전기로 부터의 아날로그 신호, 풍력발전기로의 설정값, 풍력발전기로 부터의 이진 신호/상태 데이터, 발전기로의 이진 제어명령어, 알람, 이벤트, 카운터, 타이머, 텍스트 메시지, 파일등이 있다.

2.3 실시간 고장 감내성 미들웨어

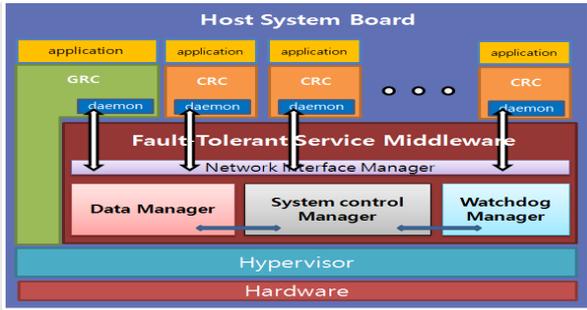
[4]는 풍력제어기와 유사한 구조를 가지고 있는 차량용 제어기(ECU: Electronic Control Unit) 상에서 고장 감내를 위한 미들웨어 설계에 관한 것이다. 차량용 제어기에서 처리되는 데이터의 경우도 풍력제어기와 마찬가지로 대부분이 실시간성을 필요로 하는데 데이터 전송의 실시간성, 고장 감내성, 자동차용 네트워크에 효과적으로 구현 가능한 통신 모델, Global time base 지원, 경량성과 같은 요구조건을 만족해야 한다.

3. 풍력제어기를 위한 고 가용성 FTSM 구조

풍력제어기에서의 고 가용성 미들웨어는 고장 감내를 지원하기 위해 이중화 및 응용 프로그램의 실행환경에 대한 고립(Isolation)기법을 통한 접근방식을 사용한다. 그림 2는 본 논문에서 풍력제어시스템을 위해 설계한 고 가용성 미들웨어 구조로 단일 하드웨어 시스템상에서 다중 독립적 계층화(MILS: Multiple Independent Layered Security) 환경을 지원하고 호스트 운영체제 환경을 포함하는 Hypervisor에 Data Manager, System Control Manager, Watchdog Manager로 구성된 고장 감내 기능을 지원하는 미들웨어 구조로 설계했다.

이중화 컴포넌트(RC: Redundancy Component)는 실제 풍력제어시스템에서 실행되는 소프트웨어에 대한 이중화 컴포넌트로 FTSM에 의해 관리된다. 이중화 컴포넌트는 General Redundancy Component(GRC)와 Critical Redundancy Component(CRC)로 나뉘는데 GRC는 순간적이며, 데이터에 대한 고립이 필요 없는 작업을 수행하고, CRC는 지속적이고

데이터에 대한 고립이 필요한 작업을 수행한다. 그 이유는 풍력제어시스템 특성 상 중요한 데이터를 처리하던 중에 고장 발생시 Roll Back을 통해 이전 환경으로 복구하기 위해서이다.



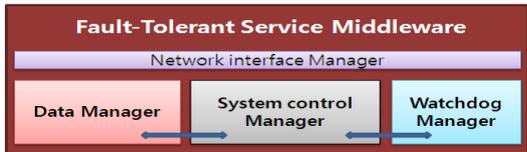
(그림 1) 고 가용성을 위한 미들웨어 구조

풍력제어시스템에서 이중화 구조는 Watchdog Manager가 Heartbeat 프로토콜을 통해 주기적으로 가상 실행환경(GRC, CRC)의 상태를 검사하게 되고, 하나의 가상 실행환경이 고장 상태로 진입하게 되면, 동일한 여분의 가상 실행환경의 컴포넌트를 통해 작업을 지속적으로 진행함으로써 고장 감내 기능을 구현한다.

고립기능은 단일 하드웨어기반의 제어시스템으로 수신되는 데이터를 FTSM 계층에서 각 GRC, CRC의 데이터에 대해 고립하는 기법으로, GRC, CRC 내에서 발생하는 고장으로 부터 데이터를 보호하기 위한 방법이다. GRC, CRC에서는 언제든지 예기치 않은 고장 상황으로 빠질 수 있으므로 고장 감내를 위해 Data를 별도로 관리할 수 있는 방법이 필요하다. Hypervisor는 풍력제어시스템에서 RC를 지원하기 위한 플랫폼으로 하드웨어 자원에 대한 관리와 RC의 자원할당을 관리한다.

3.1 Fault-Tolerant Service Middleware

FTSM은 RC와 Hypervisor간의 인터페이스 역할을 하며, 데이터 고립을 유지하기 위한 미들웨어이다. 각 미들웨어를 구성하는 Manager들은 서로 IPC를 통해 데이터 전달을 한다.



(그림 2) 고장 감내 서비스 미들웨어(FTSM)

1) System Control Manager

System Control Manager(SCM)은 RC에 대한 실행을 제어한다. 다수의 RC에 대해 우선순위를 지정하며, 최고 우선순위를 갖는 RC에게 GRC로써의 역할을 수행하도록 한다. 그리고 각각의 RC와 FTSM간의 데이터 통신을 위해 RC에게 동적으로 IP를 할당하여 테이블로 관리한다.

2) Data Manager

Data Manager(DM)는 데이터 고립을 구현하여 관리한다. DM은 RC에서 Hypervisor에게 요구하는 모든 데이터를 Repository에 관리하고 해당 Repository는 빠른 데이터 탐색과 저장을 위한 마이크로 데이터베이스를 사용한다.



(그림 3) Data Manager 구조

마이크로 데이터베이스에 저장되는 정보는 데이터를 요구한 RC의 number, 데이터 요구 시간, 데이터 Type, 데이터로 구성된다.

<표 2> Data Manager 저장 규칙

RC#	time	type	Data Size	Data
-----	------	------	-----------	------

3) Watchdog Manager

Watchdog Manager(WM)은 여러 RC에 대한 상태를 모니터링하며, 해당 RC가 정상적으로 동작 중인지 체크한다. 또한 RC는 주기적으로 Watchdog Manager에게 TCP/IP 통신으로 Heartbeat, 상태정보, 수행 중인 응용 프로그램 정보들을 전송한다. WM은 Heartbeat를 체크하여 고장 감지를 수행하며, RC에서의 고장 발생시 SCM에게 해당 RC에 대한 정보를 전달한다.

<표 3> RC 상태 관리 테이블

RC#	IP	Status	Application List
RC1	192.168.0.2	live	app1, app2, app3

- RC#: 동작 중인 RC 번호
- IP: RC에게 할당된 IP 주소
- Status: RC의 동작 상태
- Application: RC에서 실행 중인 응용 프로그램 이름

또한, WM은 확장성을 위해 Wind Farm 또는 Bottom Level의 모니터링 시스템과 연결되어 현재 실행 중인 RC의 응용 프로그램에 대한 모니터링을 수행할 수 있도록 한다. 이러한 외부 모니터링은 WM에서 제공하는 인터페이스를 사용하여 수행되며, 해당 모니터링을 위한 응용프로그램 상태 관리 테이블을 통해 지원한다.

RC#	Application	CPU%	MEM%	Stat
RC1	app1	0.3	1.4	S

- Application: RC에서 동작 중인 응용프로그램 이름
- CPU%: 동작 중인 응용프로그램의 CPU 점유율
- MEM%: 동작 중인 응용프로그램의 메모리 점유율
- Stat: 응용프로그램의 프로세스 동작 상태

4) Network Interface Manager

Network Interface Manager(NIM)은 RC와 Manager들간의 데이터 전송을 위한 인터페이스이다. RC에서 TCP/IP 통신으로 NIM에게 패킷을 전달하면, NIM은 패킷을 분석하여 해당 Manager에게 RC에서의 요청을 전달한다.

<표 3> 패킷 구조

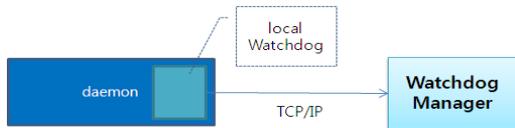
RC#	Manager	Request
-----	---------	---------

- RC#: 패킷을 보내는 RC에 대한 number
- Manager: RC 요청 대상이 어떤 Manager인지를 나타냄
- Request: Manager가 제공하는 기능 중 어떤 것을 요청하는 것인지 명시

3.2 이중화 컴포넌트(Redundancy Component)

RC는 풍력제어시스템에서 수행되는 프로그램을 실행하기 위해 기반이 되는 컴포넌트로 응용프로그램 또는 RC 자체의 고장 발생으로 인한 비정상적으로 동작 시 FTSM에 의해 복구되도록 이중화 구조를 구현하기 위한 개체이다. RC는 내부적으로 Heartbeat 프로토콜 및 RC의 상태를 검사하기 위한 데몬이 실행되고 있으며, 해당 데몬을 통해 FTSM과 상호작용한다.

데몬은 RC 내부에서 서비스 형태로 동작하는 프로그램으로 FTSM과 상호작용할 수 있는 인터페이스를 제공한다. 또한, WM와 주기적인 통신으로 고장 감지가 가능하도록 하며, RC의 동작에 대한 모니터링을 지원하기 위해 WM에게 상태 정보를 전달한다.



(그림 4) 이중화 컴포넌트의 데몬과 WM간의 정보 전달

4. 풍력제어기에서의 RC 고장시 FTSM 동작 시나리오

RC에서의 응용프로그램은 풍력발전기에서의 발생한 데이터를 기반으로 다양한 연산 작업을 수행하고 주기적으로 동작 상태에 대한 로그를 Daemon에게 전달한다. 그리고 RC 또는 응용프로그램 고장 시 Recovery 작업을 수행하기 위해 모든 사용자 응용프로그램에서는 복구 루틴을 별도로 만들어 주어야 한다. 이러한 데이터 복구 작업은 DM에게 이전에 사용하였던 데이터를 요구하여 처리된다. 응용프로그램은 필요한 데이터 항목에 대해서 DM에게 요청하고, DM은 마이크로 DB를 통해 저장된 최근의 데이터를 응용프로그램에게 전달한다.

4.1 고장 감내 흐름도

풍력제어시스템에서 고장 감내 기능을 구성하기 위해서는 일반적으로 이중화 구조로 구성되어 하며, 4단계 과정을 통해 고장 발생 시 이를 대처하게 된다.

1) Fault Detection

고장 감지는 고장의 발생을 인지하는 과정으로 고장 감내를 하기위한 시발점이 된다. 각 RC에 존재하는 데몬이 WM에게 보내는 Heartbeat를 통해 고장 발생 여부를 확인할 수 있다.

2) Fault Location

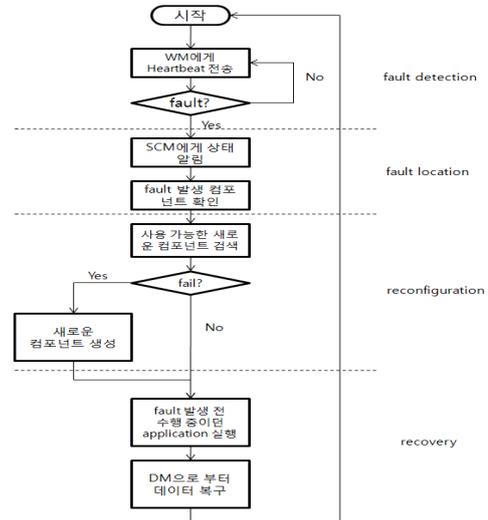
WM은 테이블로 RC에 대한 목록을 관리하는데, RC로부터 Heartbeat가 전송되지 않으면, 오류가 발생한 RC 번호를 테이블에서 검색하여 어떤 RC가 고장 상황에 처하게 되었는지 확인하고 해당 RC의 IP주소를 SCM에게 알린다.

3) Reconfiguration

SCM은 WM으로부터 전달 받은 고장 발생 RC의 정보를 통해 해당 RC가 사용하던 자원들을 반납시키고 여분의 RC에 자원을 할당하여 실행 가능한 상태로 만든다. 가용한 RC가 없다면, SCM은 새로운 RC를 생성하여 IP를 할당하고 실행 가능한 상태로 만든다.

4) Recovery

SCM은 WM로부터 전달 받은 해당 CM에서의 응용프로그램 수행 목록을 통해 고장 발생 전에 수행 중이던 응용 프로그램을 새로운 RC에서 다시 수행 가능하도록 동작시키고 응용프로그램은 본래의 작업을 수행하기 전에 데이터 복원 루틴을 수행하여 DM을 통해 이전에 사용했던 데이터들을 복구시킨다. 그림 5는 풍력제어시스템에서의 고장 감내 과정에 관한 전반적인 흐름도를 나타낸다.



(그림 5) 고장 감내 흐름도

4. 향후 작업

본 논문에서는 풍력발전기의 내·외부의 환경적 요인과 제어시스템 자체의 소프트웨어, 하드웨어적인 고장 가운데 소프트웨어적인 고장으로 인해 제어시스템 전체의 동작 중단을 막기 위해 소프트웨어적으로 이중화 구조를 지원하기 위한 고 가용성의 미들웨어에 관한 것이다. 현재 부분적으로 실제 개발이 진행 중에 있기 때문에 설계한 내용은 일부 수정될 부분이 있을 것으로 예상된다. 또한 개발 완료 시 고 가용성 미들웨어에 대한 성능평가는 향후에 수행해야 할 작업으로 예상된다.

참고문헌

[1] T. Esbensen and C. Sloth, "Fault Diagnosis and Fault-Tolerant Control of Wind Turbines", Master's Thesis, Aalborg University, 2009

[2] H. Niemann and J. Stroustrup, "Integration of control and fault detection: Nominal and robust design," in Proc. Safeprocess 97, Hull, U.K., 1997, vol. 1, pp. 341 - 346.

[3] C. Bjerge, P. Christiansen, A. Hedevang, A. Johnsson, N. Raben, J. Svensson, "Functional requirements on communication system for wind turbine applications". preliminary version, Elforsk report, 31. March 2001

[4] 박지용, 김세화, 최동호, 홍성수, "자동차용 소프트웨어를 위한 실시간 고장 감내 미들웨어의 설계," 자동차공학회 춘계 학술대회논문집, Vol.3, pp.1361-1366, Jun 2006