

## 다중 백업 기능 제어를 갖는 네트워크형 분산 제어

최 군 호<sup>†</sup>

<sup>†</sup>경성대학교 전기공학과

### Network Type Distributed Control with Multi-Backup Function Controller

Goon Ho Choi<sup>†</sup>

<sup>†</sup>Dept. of Electrical Engineering, Kyung Sung Univ.

#### Abstract

Stable running is a very important factor when we are building a control system. But, sometimes malfunction of the system controller is caused by several reasons. To solve these problems, some systems have prepared a backup controller. In the generality of cases, making a backup controller is no easy work. Especially, it becomes more difficult when many backup controllers have to be embodied in one system. In this paper, the multi-backup function controller which was implemented in the form of the network type distributed control system will be suggested. Also, using Arago Disk System that has a structure for inner loop control, the validity of the proposed methods will be verified.

**Key Words** : Backup Controller, Network type Distributed Control System, Arago Disk System

#### 1. 서 론

제어 시스템을 구축하는 데 있어서 시스템 전체의 안정적인 운영은 매우 중요한 요소이다. 따라서 전체 시스템을 제어하는 제어기는 여러 방법을 통해 안정적인 동작이 가능하도록 만들어야 한다. 그러나 제어기는 외부 또는 내부적인 요인으로 고장이 발생할 수 있고 이러한 경우 이 제어기의 관리 상태에 있는 장비나 설비 또는 공장 전체가 고장 상태에 놓일 수 있다. 이 경우 보통 비상 정지 등의 방법을 통해 고장으로 인하여 발생한 문제가 확산되는 것을 우선 막고 그 후 교체 또는 고장 수리 등을 통하여 원래 상태로의 복구를 하는 것이 일반적인 처리 방식이다. 문제는 이러한 방식이 가능하지 않은 경우, 즉 동작 정지로 문제 해결이 불가능한 경우이다. 이러한 경우 시스템에 상당한 손실이 불가피한 경우가 많기 때문에 보통 이 경우에 백업 시스템(또는 고장 시스템)을 두어 문제를 최소화 하고자 한다.

백업 제어기는 보통 실제 제어기와 동일한 제어기를 하나 더 구축하고 외부에 고장 탐지 시스템을 운영하면서 고장이 발생할 때 빠른 시점에 운영 선로를 전환하여 고장 제어기를 시스템에서 탈락 시키고 백업 제어기가 동작되도록 하는 것이 일반적인 방식이다. 문제는 이러한 경우 추가 제어기와 고장 탐지를 위한 비용 증가이다. 이 비용에는 단순히 추가 장비에 관련된 비용뿐만 아니라 백업 동작을 위한 부가 선로 구축 및 운용 관리 등의 부대적인 비용도 추가적으로 포함되는 경우가 많다. 또한 백업 관리를 해야 할 제어기가 하나가 아닌 경우 각각 독자적인 백업 시스템을 구축해야만 한다는 것도 문제점이 될 수 있다.

본 논문에서는 이러한 경우에 대한 해결을 위해 유용한 백업 제어기 형태를 구현하여 보고자 한다. 즉 보다 효과적인 방법으로 구조적으로 간단하며 하나 이상의 제어기를 백업할 수 있는 형태를 제안하여 보고자 한다. 또한 이 제안된 형태를 실제 시스템에 적용하여 실험한 결과를 살펴 보고 그 효용성을 판단하고자 한다.

<sup>†</sup>E-mail : goonho@ks.ac.kr

## 2. 네트워크형 백업 제어기

### 2.1. 기존의 백업 제어기 구조

시스템의 안정적인 운영을 위해서는 백업 시스템이 중요하다. 이 중에서도 시스템의 운용, 관리의 역할을 담당하는 제어기의 경우 시스템의 다른 부 시스템에 비해 안정된 동작 성능이 더욱 중요하다. 때문에 중요한 시스템의 경우 제어기를 2중으로 두어 그 안정성을 보장받으려 하는 경우가 있다.

Fig. 1은 일반적인 시스템에서의 백업 제어기의 구조와 동작에 대한 개념도이다. 그림에서 보는 바와 같이 정상 운용 상태일 때는 주 제어기를 중심으로 시스템이 동작한다. 이 과정에서 Fault Detector는 제어기의 동작에 이상이 발생하지 않는지 지속적으로 확인하고 있다. 이상이 발생하면 미리 준비된 알고리즘에 따라 백업 제어기로 신호를 전환하여 시스템의 전체적인 동작에 문제가 발생하지 않도록 한다.

이러한 구조는 개념적으로 명확하다는 장점이 있지만, 실제 구현을 위해서는 주 제어기와 동일한 동작을 하는 제어기와 현재 제어기의 상태를 모니터링하고 필요한 경우 백업 제어기를 동작하도록 운용해야 하는 부가적인 시스템(Fault Detector) 등이 필요하다. 이러한 부분은 비용적인 증가를 가져오고 주 제어기의 고장 빈도가 낮은 경우 기회 비용 등의 문제가 추가적인 고려 대상이 될 수 있다. 또한 Fig. 1과 같이 관리해야 할 제어기가 하나인 경우와 달리 두 개 이상인 경우에는 비용의 증가는 물론 실제 시스템을 구현할 경우 신호(배선)의 복잡도가 보다 증가하는 문제가 발생할 가능성이 많다.

### 2.2. 네트워크형 분산 제어 시스템

앞서의 백업 제어기를 구현할 때 문제점 중의 하나인 신호의 복잡성 문제는 네트워크형 분산 제어 시스템(Network Type Distributed Control Systems)을 이용

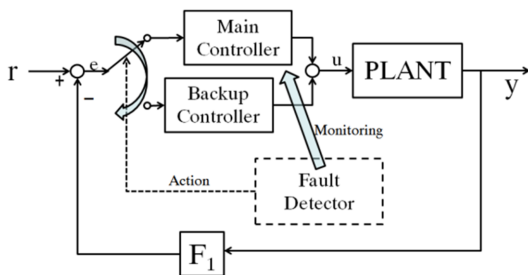


Fig. 1. Typical backup structure for a control system.

하여 해결하고자 한다. 일반적으로 분산 제어 시스템은 기존의 중앙 집중형 제어시스템이 가지는 제어 구조와 다른 개념으로 고안된 것으로, 플랜트 내의 각각의 기능을 부분별로 나누어서 처리하도록 함으로써 주 제어기의 부하를 각 부 시스템(Sub-System)으로 분산하여 중앙 집중형 구조보다 제어 성능을 향상시킬 수 있다. 또한 부 시스템의 일부 기능이 문제가 생기더라도 다른 시스템으로 대체하거나 보완하도록 하는 것이 가능하므로 전체 시스템의 신뢰도를 높일 수 있다[1,2]. 또한 이러한 분산 제어 구조에 네트워크 형태의 통신 선로를 이용한 제어 구조를 네트워크형 분산 제어 방식이라고 하는데, 이 경우 각 서버 시스템 사이의 통신 방식을 일반적으로 BUS로 불리는 단일 통신 선로를 이용함으로써 시스템 내의 각 서버 시스템 사이에 혼재되어 있는 통신 선로를 단일화 하여 실제 시스템을 구현할 때 통신 및 기타 선로 작업의 복잡성을 획기적으로 줄일 수 있도록 한 것이다[3].

Fig. 2는 일반적인 분산 제어 구조와 다른 네트워크형 분산 제어의 구조를 보여주고 있다. 그림에서 보는 바와 같이 네트워크형으로 시스템을 구현할 경우 각 서버 시스템 사이의 연결관계가 간단한 동일 구조를 가지게 됨으로써 시스템의 구현이 보다 쉬워진다. 이때 각 서버 시스템간의 연결관계는 각 서버 시스템의 내부 Firmware에서 설정할 수 있도록 한다. 이 방식은 추후 전체 시스템에서 각 서버 시스템 사이의 연결관계가 변화하더라도 외부적인 신호 변화 없이 내부의 S/W적인 변화만으로 그 구조를 바꿀 수 있다는 것을 의미하므로 시스템의 사후 유지 보수 또한 매우 간단해 진다고 할 수 있다[3].

이러한 네트워크형 분산 제어 시스템은 BUS라는 단일 선로를 사용하기 때문에 통신 과정에서 발생하는 외부 잡음이나 데이터 충돌 문제가 중요하다. 이러한

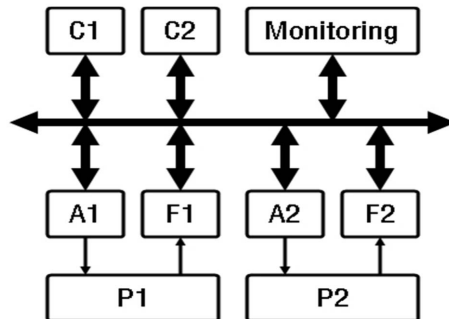


Fig. 2. Network type control systems (C:Controller, A: Actuator, F:Feedback, P:Plant).

문제해결을 위해 여러 분산제어용 필드 버스들이 다양한 방식을 사용하고 있는데, 이 중에서 전기적으로 가장 안정되어 있다고 알려진 CAN 프로토콜을 이용하고자 한다. CAN 프로토콜은 1980년대 초 독일의 BOSCH사에서 개발한 차량용 제어 네트워크 프로토콜로, 차량내에서라는 환경적인 특성상, 잡음 내력 등, 신호의 안정성과 신뢰도에 가장 큰 의미를 두고 있다. 비교적 근거리에서 짧은 신호의 송수신에 대해서는 가장 우수한 통신 방식으로 알려져 있으며, 이후 FiledBus의 한 형태인 DeviceNet의 기본 프로토콜로 확장되어 현재 가장 주목 받는 제어용 네트워크 프로토콜의 하나가 되었다[4].

**2.3. 내부 궤환 제어기 구조를 갖는 네트워크형 분산 제어 시스템**

일반적으로 내부 궤환 제어기는 시스템의 제어 대상 플랜트를 여러 개의 서브 플랜트로 나눌 수 있고 그 중에서 특정한 일부분의 서브 플랜트의 안정도를 만족하도록 하거나 성능 개선을 위하여 해당 서브 플랜트에만 적용하는 특정한 제어기를 말한다(Fig.3)[3].

이러한 형태의 제어 구조는 시스템 전체를 고려하여 제어 안정도와 성능 개선을 구현하는 제어기를 설계하는 것에 비하여 시스템에서 요구하는 제어 성능 중에서 중요한 부분이나 또는 제어가 어려운 부분만을 먼저 고려할 수 있다는 장점이 있다. 반면에 제어기 설계가 보다 어려워지고 또한 이와 같은 시스템을 실제로 구현할 경우 전체 제어 구조가 복잡해짐으로써 회로를 구성하거나 회로 선로를 연결하는 데에 복잡도가 증가하는 등의 문제점을 가지게 된다[3,5-7].

그러나 이 시스템에 네트워크형 분산 제어 구조를 적용하여 보면, 각 서브 시스템 사이의 정보의 취사 선택 여부는 각 네트워크 제어기의 내부 운용 프로그램에 따라 좌우 되므로 Fig. 4와 같은 구현이 가능하다. Fig. 3과 비교하여 보면 쉽게 그 차이를 알 수 있다. 이와 같이 네트워크형 분산 제어 시스템을 설치, 유지,

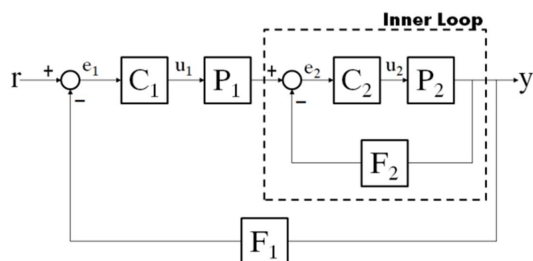


Fig 3. A block diagram of a control system with inner loop control structure.

및 보수의 편리성과 기존 시스템에 비해 추가 요구 자원이 적어 수정 보완하여 사용하는 데에도 크게 문제가 발생하지 않는다는 등의 장점이 있다[3,8,9].

**2.4. 네트워크형 백업 제어기**

본 논문에서 제시하는 백업 제어기 알고리즘의 유효성 확인을 위한 적용 대상으로, 시스템의 제어기가 2개 이상이고 네트워크형 분산 제어로 구현이 가능함을 확인한 아라고 원판 시스템을 이용한다[3,10]. 기존의 아라고 원판 시스템은 전체 시스템을 관장하는 주 제어기와 전동기 속도 제어를 위한 내부 제어기로 나뉘어져 있다. 이 구조를 Fig. 5와 같이 백업 제어기를 가지는 구조로 변경하기 위해 CAN 네트워크에 추가 노드를 구성한다.

추가된 백업 제어기는 형태상으로는 단일 제어기 형태로 보이지만 실제 내부에는 주 제어기와 내부 제어기의 기능을 모두 가지고 있도록 한다. 이러한 형태의

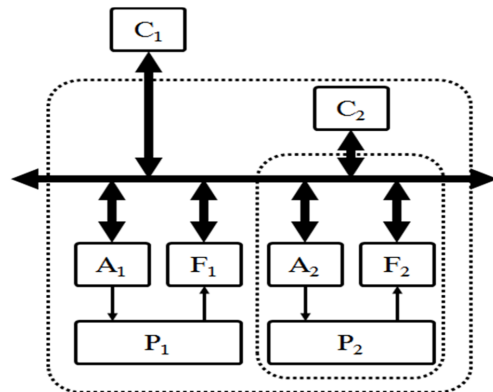


Fig. 4. Network type distributed control, with the inner loop control structure.

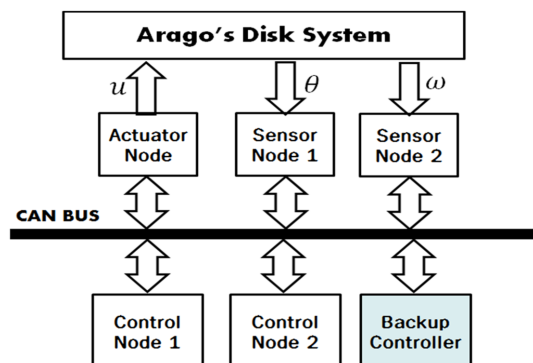


Fig. 5. Including the backup controller in the network type Arago Disk Control System.

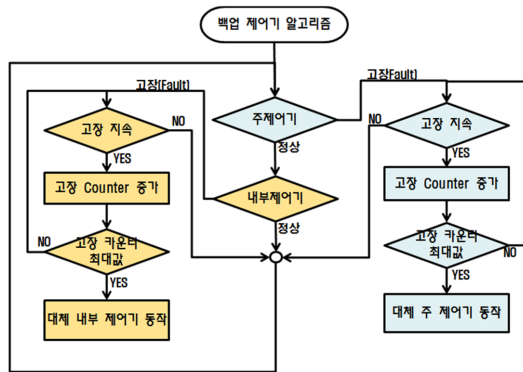


Fig. 6. Suggested algorithm for the backup controller.



Fig. 7. Experimental System.

구현이 가능한 것은 첫 째, 시스템 내의 두 제어기가 동시에 고장이 날 가능성이 적다는 것과 둘째, 네트워크형 분산 제어 구조에서는 두 제어기의 고장 탐지가 동시에 가능하다는 점에 근거한다. 이 중 두 번째의 내용은 기존의 중앙 집중형 제어 구조에서는 각각의 제어기의 상태를 확인하기 위해 별도의 고장 탐지부를 두어야 하지만 네트워크형 분산 제어 구조에서는 버스의 데이터를 분석하면 각 노드의 운용 상태를 확인할 수 있기 때문이다. 이에 대한 구현 알고리즘은 아래의 Fig. 6과 같다. 그림에서 보는 바와 같이 백업 제어기는 계속적으로 주제어기와 내부 제어기의 고장 상태를 버스를 통해서 확인을 하고 있다가 고장이 발생시 이에 대응하도록 되어 있다. 이 때 순간적(일시적)인 고장 상황일 경우에 대비하여 일정 시간(또는 일정 제어 주기) 동안 고장 반복 회수를 확인하도록 하며, 이 때 정해진 고장 지속 시간 이상이 되면 백업 제어기가 동작하도록 하고 있다. 고장 지속 시간은 시스템의 중요도나 고장 파급 시간 등을 고려하여 선택적으로 사용할 수 있으며 본 논문에 제시된 구조에서는 백업 제어기의 내부 Firmware 에서 S/W 적인 설정으로 변경이 가능하도록 구성되어 있다.

### 3. 실험 및 결과

#### 3.1. 실험 시스템 및 내부 Firmware 구현

Fig. 7은 제작된 아라고 원판 시스템이다. 기본적으로 직류 전동기의 회전 속도에 따라 막대의 각도가 변화하도록 구성되어 있으며, 직류 전동기 속도와 막대의 각도는 각각 7V/1000 rpm의 출력을 갖는 Tacho Generator와 막대와 직결로 연결된 Copal사의 J40S 포텐서 미터를 이용하여 측정한다. 모두 6개의 CAN 노드로,

기존의 5개 노드[3]에 추가로 백업 제어기 노드를 구현하였다. 각 노드들은 AT90CAN128 기반의 ATMEL사의 DVK90CAN 개발 보드를 이용하여 하나의 보드에서 CAN 노드와 내부 제어 프로그램(Firmware)을 통합하여 구성할 수 있도록 하였으며, 모든 노드는 하나의 CANBUS와 연결된 단일 통신 선로를 사용하도록 구성하였다.

주 제어기와 내부 제어기는 통신을 위한 패킷 데이터의 길이를 분석하여 각각 20 msec와 5 msec 제어 주기를 가지면서 동작하도록 하였다. 아울러 이 주기는 각 노드 컨트롤러의 내부 클럭에 동기하도록 함으로써 각 제어기의 제어 타이밍을 일정하게 유지할 수 있도록 하였다.

각 제어기의 동작 상태를 확인하기 위하여 각 제어기의 통신 패킷의 끝에 현재 상태를 나타내는 데이터를 추가로 송신하도록 하고 이 데이터를 백업 제어기가 확인하여 현재의 상태를 알 수 있도록 하였다. 주 제어기와 내부 제어기의 백업 기능은 백업 제어기 내부에 동시에 구현되어 있으며 각 제어기의 상태에 따라 선택적으로 동작이 가능하도록 Firmware를 구성하였다. 예를 들어 고장의 발생시 각각 5주기 동안 제어기가 동작을 하지 않았을 경우 내부 Firmware에서 백업 제어기가 동작하도록 구현하였다면, 이것은 주 제어기와 내부 제어기가 각각 100 msec와 25 msec 동안 동작을 멈추었을 경우에 백업 제어기가 동작함을 의미한다.

#### 3.2. 동작 실험 및 결과

우선 Fig. 8은 시스템에 고장이 없이 정상적으로 시스템이 운영될 때의 결과 파형이다. 그림에서 보는 바와 같이 약 1초 정도의 시간이 지난 후에 막대의 각도가 안정화 되는 것을 볼 수 있다.

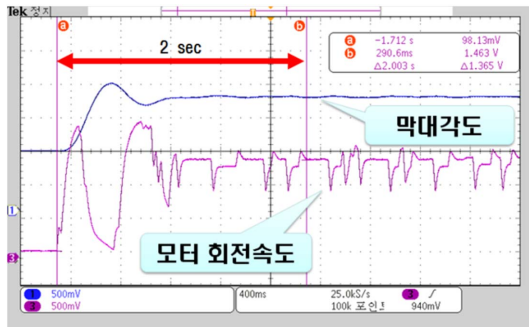


Fig. 8. A angle control result of the Arago Disk System using the experimental system

이 결과를 바탕으로 동작 개시 후 2초정도의 시간이 지난 후 각각의 제어기의 전원을 차단하여 강제로 고장 상태를 만들어 실험하였다. Fig. 9와 Fig.10의 결과는 각각 내부 제어기와 주 제어기가 고장 발생 후 500msec 정도의 시간 동안 고장이 복구 되지 않을 경우에 내부 Firmware에서 백업 제어기가 동작하도록 구현하고 실험한 결과이다.

Fig. 9는 ▼ 부근에서 내부 제어기의 고장을 발생시켰을 때, 일정 시간 불안정한 모습을 보이지만 이후 백업 제어기가 동작하면서 다시 정상상태로 돌아오는 것을 보여주고 있다. 다만 고장이 발생할 당시의 시스템의 상태에 따라 제어기가 고장 상태인 동안의 동작이 다른 데, Fig. 9의 위 두 결과는 직류 전동기의 속도가 목표치 부근에서 정속 운전을 하고 있을 때의 고장이 발생한 경우이고 아래의 두 결과는 각각 감속과 가속을 하고 있을 때 고장이 발생한 경우의 결과를 보여준다.

Fig. 10은 같은 방식으로 주 제어기에 고장이 발생하고, 이후 백업 제어기를 통해 다시 안정화되는 모습을 보여 주고 있다. 역시 이 경우에도 고장이 발생할 당시의 시스템의 상태에 따라 고장 상황에서의 동작은 다소 차이가 있음을 알 수 있다. Fig. 9, 10은 비교 과형이어서 범례는 Fig. 8에서의 동일함 것임을 밝히고자 한다.

#### 4. 결 론

기존의 백업 제어기는 단순히 기존 제어기에 병렬로 동일한 기능의 제어기를 추가해 두고 비상시 전환 사용될 수 있도록 하는 것이 일반적이었다. 이 경우 발생할 수 있는 여러 문제점을 해결하기 위하여 시스템의 구성 방식을 네트워크형 분산 제어 구조로 바꾸고 여기에 S/W 적인 추가 작업을 통해 하나의 백업 제어기

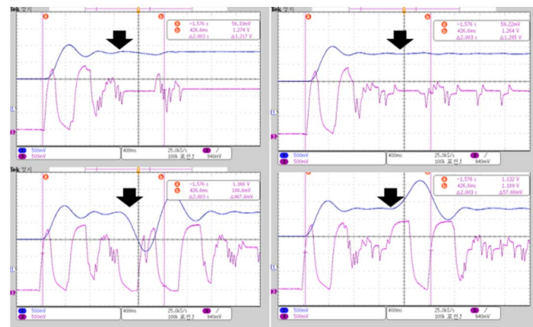


Fig. 9. A angle control result when a fault occurred on the inner controller

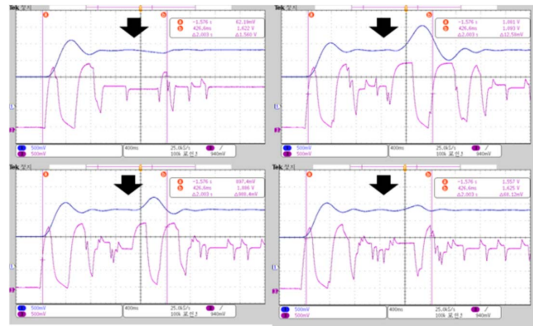


Fig. 10. A angle control result when a fault occurred on the main controller

로 다른 여러 개의 제어기를 백업 하는 것이 가능하도록 구현하였다. 실제로 본 논문에서는 하나의 시스템에서 두 개의 제어기가 필요한 아라고 원판 시스템을 이용하여 이 두 제어기가 각각 고장을 일으켰을 때 대응이 가능함을 보였다. 실험 결과는 고장 시간 500msec 후 동작에 대한 것이지만, 실제 실험은 이 보다 짧은 고장 상태에 대한 실험도 진행하였다. 다만 이 경우 시스템이 고장 상태에 빠졌다는 모습이 너무 미미하게 나타나서 결과로 언급하지 못하였다. 이는 역설적으로 짧은 고장 시간에 대해서는 시스템에 거의 영향을 미치지 않고 복구가 가능하다는 것을 의미한다. 제안된 방식은 이론상 시스템의 자원이 허용하는 한도 내에서 무한 개의 제어기를 백업하는 것이 가능하며, 구현 또한 용이한 편이다. 때문에 특정 시스템이 비상 백업 시스템을 운용하는 데 있어서 존재하는 비용적이나 시간적인 문제들의 상당 부분을 해결할 수 있으리라 생각되며, 이에 따라 현재 산업 현장의 각종 자동화 설비나 시스템에 응용 및 활용도를 가질 수 있는 방식이라고 생각된다.

### 참고문헌

1. G. H. Choi, "The Control of Multiple Plants using the CAN Protocol," J. of Semiconductor & Display Equipment Technology, Vol. 8, No. 2, 2009.
2. Feng-Li Lian, James Moyne, and Dawn Tilbury, "Network Design Consideration for Distributed Control Systems," IEEE Trans. Control sys. Tech., vol.10, No.2, pp.297-307, Mar. 2002
3. G. H. Choi, "Network Type Distributed Control of a System with Inner Loop Control Structure," J. of the KIIIEE, Vol.28, No.2, 2014.
4. BOSCH GmbH, CAN Specification Part A & Part B, 1991.
5. H. J. Kim, J. W. Kim, and K. W. Lee, "Flight control of a small unmanned aerial vehicle using a dynamic compensator," The J. of Korea Navigation Institute, Vol.16, No.4, 2012..
6. B. K. Kim, and W. K. Chung, "Design of Robust Motion Controllers with Internal-Loop Compensation," The J. of KSME, Vol.25, No.10, 2001.
7. H. J. Yeo, "Design of a Robust Controller Using Disturbance Rejection Controller," The J. of KAIS, Vol.7, No.2, 2006
8. J. H. Jung, J. H. Lee, T. D. Park, and K. H. Park, "Development of a Network-Based Distributed Control System," CASS 2008 , pp.777-782, 2008.
9. J. H. Jung, S. Y. Choi, and K. H. Park, "A Study on the Power System Control and Monitoring Technique Using CAN," The Transactions of the Korean Institute of Electrical Engineers, Vol.52D, No.5. pp.268-276, May 2003.
10. G. H. Choi, A Study on the Position Control of Arago's Disk Systems, Master Thesis, SungKyunKwan Univ., 1994.

---

접수일: 2014년 11월 28일, 심사일: 2014년 12월 11일,  
 게재확정일: 2014년 12월 22일