

네트워크 기반 고장대처 제어 시스템

고장은 시스템의 하나 이상의 구성 요소에서 비정상적인 출력이 나타나는 현상으로 인명 손실, 임무 실패 등의 비극적인 결과를 야기하게 된다. 따라서 고장대처 문제는 시스템 안전성 향상을 위한 매우 중요한 주제이다. 본 논문에서는 고장대처 제어 기법 관련 기술현황과 문제점을 살펴보고, 이어서 네트워크 기반 시스템의 고장대처 제어 기법 적용을 통해 현재의 문제점을 해결할 수 있는 가능성을 검토한다. 아울러 고장대처 제어 기법이 전통적인 항공우주 분야뿐만 아니라 공장자동화 등 일반적으로 시스템 고장이 인명손상으로 연결되지 않는 것으로 인식되는 산업분야에서도 시스템의 신뢰도, 유지보수 편의성, 제품의 품질 개선 등의 측면에서 매우 유용하게 적용될 수 있음을 소개한다.

■ 양인석*, 김동길, 이동익
(경북대학교)

I. 서론

고장은 정상적인 시스템의 구성 요소 중 하나 이상에서 발생하는 비이상적인 출력을 의미한다[1]. 고장 발생은 결과적으로 임무 실패, 비용 손실, 나아가 인명 피해와 같은 중대한 사태를 초래할 수 있다. 일례로 항공기의 고장은 막대한 인명 손실을 초래할 수 있으며, 산업용 생산기기의 고장은 제품의 품질저하를 초래할 수 있다. 본 논문에서는 먼저 지난 30여년에 걸쳐 다양하게 연구되고 있는 고장대처 제어(fault-tolerant control, FTC) 기법 현황과 문제점을 살펴보고, 이어서 현재 기법들의 공통적인 문제점을 해결할 수 있는 하나의 대안으로써 네트워크 기반 고장대처 제어 구조를 소개한다. 특히 제안된 네트워크 기반 고장대처 제어는 다중 하드웨어가 지원되지 않거나 PID 제어로 구현된 다양한 산업분야에도 적용 가능하며 이를 통해 신뢰도와 유지보수 비용 등을 개선할 수 있을 것으로 기대된다.

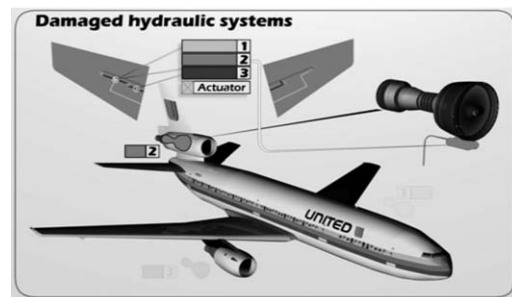
II. 고장대처 제어 기법 고찰

1. 고장대처의 필요성

항공우주 시스템은 안전문제가 가장 민감한 산업분야로 인식된다. 이는 항공시스템의 특성상 사소한 고장이 대형 참사로 연결될 위험성이 크기 때문이다. 예를 들어, UA232편의 사고 사

례를 통해 고장대처 제어의 필요성을 알아보자.

1989년 7월 19일 United Airlines의 UA232편이 Denver에서



(a) The damages of United Airlines flight UA232.



(b) Trajectory of United Airlines flight UA232.

그림 1. United Airlines의 UA232 항공 사고 [2].
Fig. 1. The accident of United Airlines flight UA232 [2].

Chicago로 비행하였다. 사고기종은 Mc-Donnell Douglass의 DC-10기로 승무원 11명을 포함해 총 296명이 탑승하고 있었다. DC-10기는 세 개의 엔진(좌우 날개(1, 3번 엔진)와 수직 꼬리 날개(2번 엔진)에 각각 한 개씩)를 장착하고 있으며, 제어면은 세 개의 독립적인 유압 튜브에 의해 제어된다. UA232는 Alta 상공의 교차점에서 항로를 변경하던 중, 2번 엔진에서 고장이 발생하여 엔진의 팬디스크(fan disk)가 파열되었다. 파열된 팬디스크의 파편들은 항공기의 유압 튜브를 절단하였으며, 결과적으로 그림 1(a)에 나타난 것처럼, 조종사는 항공기의 제어면을 통제할 수 없게 되었다. 조종사는 자신이 통제할 수 있는 양 날개의 두 개 엔진을 편향적으로 제어하여 항공기를 Sioux Gateway 공항까지 접근하였다(그림 1(b)). 하지만 불행하게도 착륙 속도를 제어하지 못하고 추락하여 111명이 사망하는 인명 피해가 발생하였다[2].

그림 2는 1993년부터 2007년까지의 CAA-NL (Civil Aviation Authority of the Netherlands)에서 발행한 항공 사고에 대한 통계 조사 결과를 보여준다. 그림 2에서 UA232 사고의 원인에 해당하는 'loss of control - inflight'는 2번째로 높은 사고 비중을 나타내고 있으며, 발생 빈도는 줄어들지 않고 있는 추세이다. 만약 항공기에 고장대처 제어 시스템이 탑재되어 있다면 16% 정도의 항공기 사고를 줄일 수 있을 것으로 많은 보고서들은 내다보고 있다. 과거에 발생한 사고는 당시의 기술로는 불가항력적인 사건일 수도 있다. 하지만 컴퓨터 및 통신 기술의 비약적인 발

전을 거듭한 현재는 항공기를 비롯한 다양한 시스템의 안전성을 향상시키기 위한 시도가 이루어지고 있다.

한편, 고장대처 제어는 항공우주 분야 외에도 와인딩 머신(winding machine)[23]등 일반 산업용 기기에도 다양하게 적용되고 있는 추세이다. 산업용 기기의 예상치 못한 고장 발생은 생산제품의 품질을 저하시킬 뿐 아니라, 생산라인의 중단 등으로 인해 막대한 경제적 손실을 불러오기 때문이다. 따라서 고장대처 제어는 사고로부터의 안전성 확보뿐 만 아니라 시스템의 신뢰도, 유지보수 편의성, 제품의 품질 개선 등을 위하여 산업 전반에 걸쳐 폭넓게 적용될 수 있다.

2. 고장대처 기법 동향

시스템의 안전 문제는 시스템을 개발하는 단계에서부터 고려되어야 하는 중요한 주제이다. 현재 시스템의 안전성을 향상시키기 위해 널리 쓰이는 방법은 동일한 기능을 수행하는 다중의 백업 시스템을 추가로 설계하는 하드웨어 다중화(hardware redundancy)방법이다. 또한 다중의 하드웨어를 탑재하여 안전성을 높이는 방법과는 별도로 최근 30년동안 제어 이론을 기반으로 고장을 대처하는 고장대처 제어 기법에 대한 연구가 진행되어 왔다. 본 장에서는 하드웨어 다중화 기법 및 고장대처 제어 기법에 대한 동향 및 그 문제점을 제시 한다.

하드웨어 다중화 기법 : 시스템에서 발생하는 고장을 대처하는 방법 중 현재 가장 널리 사용되고 있는 방법은 다중의 하

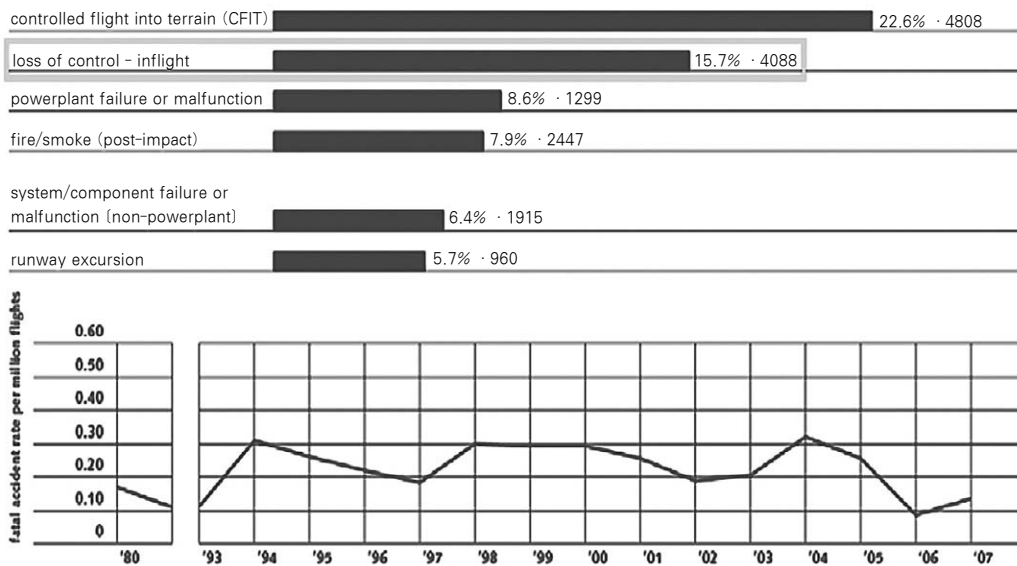


그림 2. 항공기 사고 통계[3].
Fig. 2. Accident statistics[3].

드웨어를 구성하는 하드웨어 다중화 방법이다. 하드웨어 다중화는 설계 단계에서 다중의 하드웨어를 구성하는 방법으로, 만약 시스템 내에 고장이 발생한다면, 기 설계된 다중의 하드웨어를 이용하여 고장을 대처하게 된다. UA232 사고 사례에서 본 DC-10 항공기에는 각 제어면에 두 개의 제어면이 추가적으로 탑재되어 있으며, 각 제어면은 서로 독립적인 3개의 유압 라인으로 연결되어 있다(그림 1(a)). 따라서 DC-10 항공기의 경우, 제어면에 고장이 발생하더라도 여분의 제어면을 이용하여 비행이 가능할 뿐 아니라 만약 유압 라인에 고장이 발생하더라도 다른 유압 라인과 연결되어 있는 제어면을 이용하여 고장을 대처할 수 있게 설계되어 있다. 이는 DC-10기를 설계하는 과정에서 항공기에 다수의 하드웨어를 설계한 예로 볼 수 있으며, 고장이 대형 참사로 이어지는 항공기의 안전성을 높이기 위해 항공기 제조사에서 필수적으로 채택하고 있는 방법이다.

고장대처 제어 기법 : 하드웨어 다중화 기법이 시스템의 설계 단계에서 다중의 하드웨어를 구성하여 안전성을 높인 방법이라면, 고장대처 제어 기법은 고장으로 인한 성능 저하를 제어 이론을 기반으로 하여 대처하는 방법을 말한다. 따라서 고장대처 제어 기법은 추가적인 하드웨어의 구성을 필요로 하지 않으며, 단지 제어 알고리즘에 영향을 받게 된다. 이러한 이유로 고장대처 제어 기법을 하드웨어 다중화 기법과 구분하여 소프트웨어 다중화(software redundancy) 기법으로 분류한다.

고장이 발생한 시스템을 다음과 같이 표현할 수 있다.

$$\dot{x}_f = f(x_f, u, f_a, f_s, f_p) \quad (1)$$

여기서, x_f 와 u 는 각각 고장으로 인하여 영향을 받은 상태벡터와 입력이다. 또한 f_a, f_s 와 f_p 는 각각 액추에이터, 센서 그리고 플랜트 고장이라고 가정하자. 그리고 만약 고장대처 제어 입력 u , 이 인가된 시스템을 다음과 같다고 가정하자.

$$\dot{x}_r = f(x_r, u_r, f_a, f_s, f_p) \quad (2)$$

여기서, x_r 은 고장대처 제어 입력이 인가된 시스템의 상태벡터를 의미한다. 그러면 고장대처 제어는 (3)을 만족하는 u_r 을 찾는 방법을 의미한다. $\varepsilon = [\varepsilon_1, \varepsilon_2, \dots, \varepsilon_n]$, $\varepsilon_i > 0$ 에 대하여,

$$\|x - x_r\| < \|\varepsilon\| \quad \text{또는} \quad |x_i - x_{r,i}| < |\varepsilon_i|, \text{ for } i = 1, 2, \dots, n \quad (3)$$

여기서, x_i 는 i 번째 상태변수를 나타낸다.

고장대처 제어 기법은 고장대처 제어 입력 u_r 을 찾는 방법에 따라 크게 수동적인(passive) 방법과 능동적인(active) 방법으로 나누어진다[4].

수동적인 고장대처 제어(passive fault-tolerant control, PFTC) 방법은 그림 3과 같이, 시스템의 제어기를 충분히 강인하도록 설계하여 고장으로 인한 성능의 저하에 대처 가능하도록 하는 방법이다[5-6]. 즉, PFTC 방법은 발생 가능한 고장의 집합 F 와 외란의 집합 Δ 에 대하여, 원하는 목표를 만족시키는 강인제어기 K 를 설계하는 방법으로 볼 수 있다. 따라서 고장대처 제어 관점에서 대부분의 강인 제어기는 수동적인 고장대처 제어기로 분류할 수 있다.

반면 능동적인 고장대처 제어 기법(active fault-tolerant control, AFTC)은 고장에 대하여 정확하게 진단하고 고장 특성을 고려하여 제어기를 재구성(reconfiguration)하는 제어 기법이다. 즉, 그림 4와 같이, 액추에이터, 플랜트 또는 센서에 고장이 발생하면, FDD (Fault Detection and Diagnosis) 모듈에서 고장 발생 유무의 탐지 및 고장에 대한 진단을 수행하고, 수행된 결과를 바탕으로 제어기 재구성 모듈에서 제어기를 재구성하게 된다. 따라서 AFTC는 다음과 같이 표현할 수 있다: 발생 가능한 고장의 집합 F 와 외란의 집합 Δ 에 대하여, FDD 알고리즘을 이용한 고장 $f_i (i = a, s, p)$ 에 대한 추정 결과 \hat{f}_i 를 바탕으로, 원하는 목표를 만족

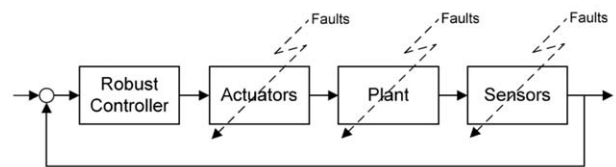


그림 3. 수동적인 고장대처 기법 구조 [4].
Fig. 3. The structure of passive fault-tolerant control [4].

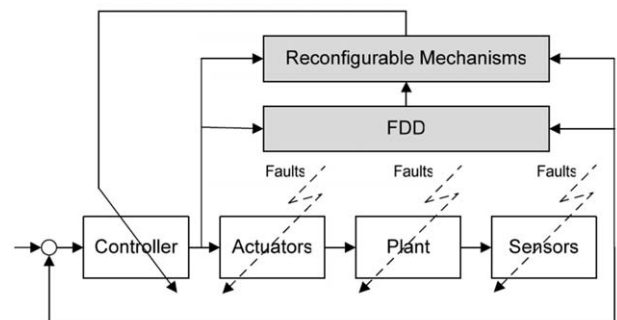


그림 4. 능동적인 고장대처 기법 구조 [4].
Fig. 4. The structure of active fault-tolerant control [4].

시키는 제어기 $K(\hat{f}_i)$ 를 재설계하는 방법이다. 여기서, AFTC는 고장 특성에 따라 기 제어기를 다시 설계해 주기 때문에 재구성으로 정의된다. PFTC 기법과는 달리 AFTC 기법은 시스템에 발생한 고장에 대하여 적절한 제어기를 설계하게 되어, 고장에 적극적으로 대처할 수 있는 장점이 있다.

앞에서 살펴본 것과 같이, PFTC는 강인 제어기를 설계하여 고장을 대처하는 방법으로, 고장으로 인한 성능저하를 외란으로 간주하여 원하는 성능을 유지시키는 방법이다. 하지만 PFTC의 경우, 고장으로 인하여 시스템의 성능 저하가 설계 단계에서 설정된 허용치를 경우 대처가 불가능한 한계를 가지고 있다. 즉, 대처 가능한 고장의 범주가 매우 협소하다. 따라서 현재 고장대처 제어 기법에 대한 연구는 AFTC에 대한 연구에 초점이 맞추어져 있다.

그림 5는 PFTC와 AFTC를 중심으로 현재까지 연구된 기법들을 분류한 것이다[7]. 그림 5로부터, AFTC 기법은 크게 극점 위치 조절(pole assignment) 방법[8-11], 다중 모델(multiple model) 방법[12] 그리고 제어면 재구성(control allocation) 방법[13-14]의 세 가지로 분류될 수 있다. 극점 위치 조절 방법은 정상적인 모델과 고장이 발생한 시스템의 오차를 최소화하도록 제어기를 재구성하는 방법으로, pseudo-inverse[8], eigenstructure assignment[9], model reference adaptive control[10], sliding mode control[11] 등이 있다. 다중 모델 방법은 제어기 설계 단계에서 예상 가능한 고장의 모델을 미리 정의하고 각 고장에 따른 제어기를 사전 설계한 후, 시스템에 고장이 발생하면 기 정의된 고장 모델에 가장 적합한 제어기를 스위칭 시켜 고장을 대처하는 방법이다[12]. 마지막으로 제어면 재분배 기법은 액추에이

터의 수가 상태변수의 수보다 많은 시스템(over-actuated system)에서 사용 가능한 방법으로, 여분의 액추에이터를 관리하는 방법이다[13-14]. 여분의 액추에이터는 시스템의 성능 향상을 위해 추가되는 경향이 있으나 고장에 대한 대처 가능성을 높여주는 역할을 하기도 한다. 제어면 재분배 기법은 증가된 고장 허용 여유도를 이용하여 고장을 대처하는 방법으로 액추에이터 고장으로 인하여 저하된 성능을 여분의 액추에이터를 이용하여 보상하는 방법이다.

3. 고장대처 기법의 문제점

앞 절에서 살펴본 고장대처 제어 기법을 통해 시스템의 안전성이 향상되는 효과를 얻을 수 있음은 분명한 사실이다. 하지만 산업현장의 다양한 실제 시스템에 적용하고자 할 경우 다음과 같은 공통적 문제점에 직면하게 있다.

하드웨어 다중화 기법의 문제점 : 앞 절에서 소개한 고장대처 제어는 일반적으로 다중의 하드웨어를 가정하고 있다. 그러나 다중 하드웨어는 크기, 무게, 비용, 복잡성 등 많은 제약이 따르며 시스템의 유지보수 비용 증가를 초래하므로, 백업 하드웨어는 최소한으로 적용되어야 한다. 반면에, 다중 하드웨어의 적절한 활용을 위해서는 고장대처제어 등 다양한 소프트웨어 기법이 동시에 적용될 필요가 있다. UA232 사고기인 DC-10은 이미 하드웨어 다중화를 통해 안전성을 높인 항공기이다. 하지만 사고 사례에서와 같이, 다중의 백업 시스템들은 사고의 순간에 적절히 동작하지 않았다. 만약 UA232에 고장대처제어 기법으로 편향추력 제어 장치가 탑재되어 있었다면 인명 손실을 크게 줄일 수 있었을 것이다.

한편, 고장대처기법 적용이 항공우주, 원자력 등 하드웨어 다중화가 잘 갖추어진 시스템을 넘어서 다양한 산업분야로 확대되고 있는 추세를 고려할 때, 다중 하드웨어에 의존하지 않는 고장대처 기법에 대한 연구도 절실한 상황이다.

제어기법의 문제점 : 앞 절에서 살펴본 바와 같이, AFTC는 고장이 발생한 시스템의 극점을 옮겨주거나 기 설계된 고장 모델을 이용/제어면을 재분배하여 고장을 대처하게 된다. 하지만 대부분의 산업 현장에서는 PID를 기반으로 한 제어 시스템이 주를 이루고 있다. 따라서 기존의 고장대처 기법을 다양한 산업 현장에 적용하기 위해서는 PID 기반 제어기를 그림 5에서 제시된 제어기 형태로 다시 설계할 필요가 있는데, 이는 현실적으로 매우 어렵다. 따라서 산업용 시스템에 적용하기 위해서는 기 개발된 PID 제어기에 적합한 PID기반의 고장대처 제어 연구가 요구된다. PID 기반의 고장대처 제어 기법으로는

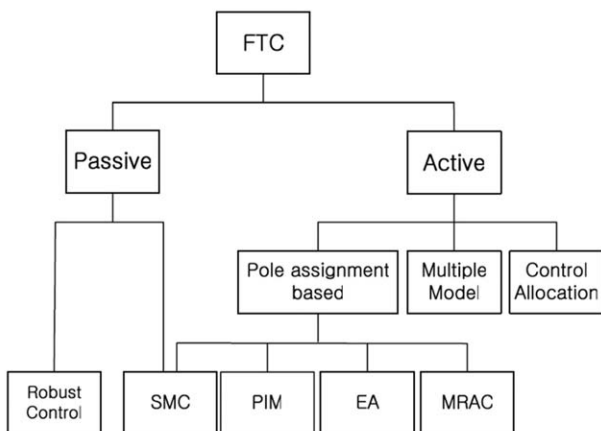


그림 5. 고장대처 기법 분류(7).
Fig. 5. The classifications of FTC(7).

Kim *et al.*이 제시한 방법이 있으나[24], 추가적인 연구가 필요한 상태이다.

고장진단 기법의 문제점 : AFTC를 산업 현장에 적용할 때 직면하는 또다른 어려움은 FDD와 재구성 메커니즘의 효과적 통합이다. AFTC는 고장에 대한 정확한 진단을 기반으로 제어를 재구성하여 고장을 적극적으로 대처하는 방법이다. 따라서 AFTC를 적용하기 위해서는 신속하고 정밀한 고장 진단 결과가 필요하지만, 이는 매우 어려운 문제이다[4]. 불행하게도 지난 30여 년간 고장대처 제어 기법과 고장 탐지 및 진단 기법은 서로 독립적으로 연구가 진행되고 있다[15]. AFTC는 완벽한 FDD 결과를 가정하고 다양한 고장대처 제어기법을 연구한 반면에, FDD 연구는 주로 고장대처제어와 무관한 시스템 모니터링(*monitoring*)에 초점을 맞추어 진행되고 있다[15]. 결과적으로 항공분야를 제외하면 AFTC가 실제 시스템에 적용된 사례를 찾아보기 어려운 근본 원인 가운데 하나가 AFTC에서 요구되는 고장진단 정보를 실제 제공받기 어렵다는 사실이다. 이 문제를 해결하기 위해 FDD와 AFTC를 결합하는 연구가 최근에 일부 시도되고 있으나, 현재 주를 이루는 전역 FDD (*global FDD*) 기반 연구결과는 여러 가지 문제점으로 인해 AFTC에 적용하는데 큰 제약이 따른다. 즉 그림4에서 나타낸 것처럼, 전역 FDD는 센서, 프로세스, 구동장치의 고장유무를 식별하기 어렵고, 시스템 모델링 오차에 따른 진단결과의 정확성이 부족하며, 시스템 규모에 따라 진단 소프트웨어가 복잡해지고 지연시간이 급격히 증가한다. 따라서 전역 FDD와 달리, 각 센서와 구동장치에 내장된 컴퓨터와 센서를 이용하여 국부적으로 고장을 진단하는 지역 FDD (*local FDD*) 기술과 이를 위한 네트워크 기반 제어구조의 적용이 대안으로 인식되고 있다.

Ⅲ. 네트워크 기반 고장대처제어시스템

최근 전자 및 통신 기술의 발전에 힘입어 기존의 기계적으로 구성된 시스템을 전기식 부품과 디지털 통신으로 대체하는 경향이 나타나고 있다. 대표적으로 항공기 산업의 Fly-by-Wire (FBW) 및 자동차 산업의 X-by-Wire (XBW) 기술을 예로 들 수 있다[13]. 이와 같이 기존의 기계적인 구동 장치로 구성된 제어 시스템을 전기모터와 양방향 통신네트워크로 대체한 시스템을 통칭하여 네트워크 기반 제어시스템(*networked control system; NCS*)이라 부른다[16]. 그림 6은 일반적인 NCS의 구조를 보여준다. NCS에서는 제어기와 액추에이터, 센서와 제어기 사이를 연결하는 필드버스(*fieldbus*)를 통해 제어에 필요한 정보를 공유한

다. 따라서 NCS는 기존의 제어시스템에 비해 부피 및 무게가 감소되는 효과를 얻을 수 있으며, 유지보수가 편하다는 장점을 얻게 된다[18]. 또한 네트워크의 유연성과 확장성으로 인해 시스템의 초기 설계 단계에서 벗어난 새로운 상태에 대하여 유연한 대처가 가능하며, 이는 고장에 대처할 수 있는 유연한 시스템 설계가 용이함을 의미한다[13].

NCS에서는 네트워크를 통한 분산 구조 형태로 설계가 가능하며, 지역 FDD의 구현이 가능하다는 장점이 있다. 즉 각 노드에서 자가 FDD를 수행하여 자신의 상태에 대하여 진단이 가능하며, AFTC에서 필요로 하는 고장 진단 정보를 빠르고 정확하게 전달할 수 있게 된다. 아울러 지능형 센서와 구동장치의 자체연산 기능을 활용함으로써 제어기 재구성 메커니즘과 구조를 단순화할 수 있다. 그 결과 기존의 AFTCS와는 달리, PID 제어 기반의 고장대처제어와 다중 하드웨어의존도를 경감시킬 수 있는 고장대처제어기 설계가 가능하다. 이러한 장점들로 인해서 NCS 기반 고장대처제어 기법(*networked fault-tolerant control system, NFTCS*)에 대한 연구가 점차 확산되고 있다.

1. 네트워크 기반 고장대처 제어시스템

현재까지 제시된 주요 NFTCS 연구를 살펴보면, Autonomous Control and Supervision System(ACSS)[19], Fast on-Line Actuator Recovery Enhancement(FLARE)[20], self-diagnostic smart actuator based NFTCS[13] 등이 있다. 그러나 이 연구들은 기능적인 측면에서 거의 동일한 구조를 가진다. 궁극적으로 고장대처는 고장에 대한 정보를 로컬 영역(액추에이터, 센서 등의 하위 단계)에서의 자가 고장 진단 결과를 바탕으로 재구성되는 방법이다.

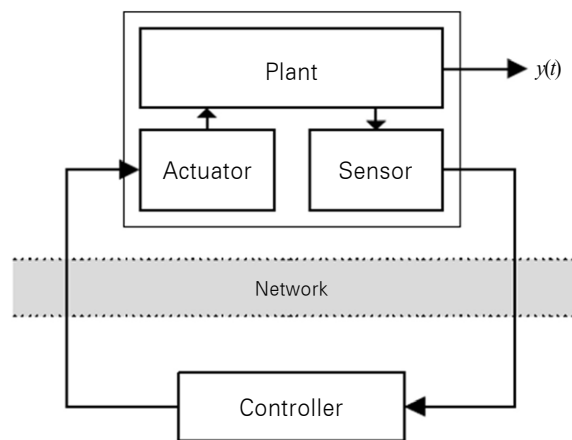


그림 6. 네트워크 제어 시스템 구조[17].
Fig. 6. The structure of NCS[17].

즉, 그림 7은 Yang *et al.*이 제안한 NFTCS 구조로 하위 단계인 액추에이터가 자가 고장을 진단하여, 고장이 발생할 경우 액추에이터의 상태 정보를 생성하여 상위 단계인 supervisor에게 전달하게 된다. 액추에이터로부터 상태 정보를 전달받으면, supervisor는 상태 정보를 바탕으로 제어기를 재구성하여 고장을 대처 한다. 여기서 자가 고장 진단 및 네트워크 통신이 가능한 액추에이터를 통칭하여 지능형 액추에이터로 정의하였으며, 지능형 액추에이터는 기설계된 FDD 모듈을 액추에이터에 탑재하여 구현할 수 있으며(ACSS, FLARE), 또한 그림 8과 같이, 고장대처에 필요한 정량적인 상태 변수(0~1사이의 정량적인 값)를 제공한다[21].

2. NFTCS 적용사례

현재 NFTCS를 구현한 상용화된 시스템은 보고된 사례가 많지 않지만, 항공 산업을 중심으로 연구가 진행되고 있으며 NASA Dryden Flight Research Center에서 연구한 propulsion control aircraft (PCA) 시스템이 대표적이다[22].

PCA 시스템은 UA232와 같이 항공기의 제어면의 통제가 불가능한 상황에서 추력만을 제어하여 항공기를 착륙시키는 시스템이다[22]. 1998년까지 NASA는 F-15, MD-11, B-747-400 등 7종의 상용 항공기에 PCA 시스템을 시험/시뮬레이션을 하였으며, 결과적으로 활주로에 안전하게 착륙할 수 있다는 결론을 얻었다.

그림 9는 PCA 시스템의 구조를 보여준다. 그림 9에서 비행경로와 방향각은 조종사에 의해 직접 입력되는 값이다. 만약 PCA가 동작하는 상태에서 조종사가 비행경로와 방향각을 입력했다면, 비행 제어 컴퓨터는 현재 비행 상태를 바탕으로 오차를 계산하여 좌우 엔진에 편향적인 추력이 나타나도록 하여, 결과적으로 추력만을 이용하여 항공기를 제어하게 된다. 여기서 PCA는 항공기의 하드웨어의 변화 없이 단지 소프트웨어를 변경함으로써 구현된다.

그림 10은 MD-11 항공기에 PCA 적용한 실험 결과를 보여준다. 본 실험에서 PCA 시스템은 비행 제어 컴퓨터의 알고리즘으로 삽입되어 구현되었으며, ARINC 429 데이터 버스를 이용하여 FADEC에 전송되었다. 실험에 적용된 착륙 접근 속도는 175 knot, 플랩 28에 제어면은 동작하지 않았다. 그림 10에서, 항공기는 4 ft/sec의 속도로 하강하여 활주로의 중앙선에 착륙하였으며, 착륙 과정에서 제어면의 각도가 움직이지 않았음을 확인할 수 있다. 또한 좌우 엔진의 EPR이 편향적으로 동작하고 있음을 볼 수 있다.

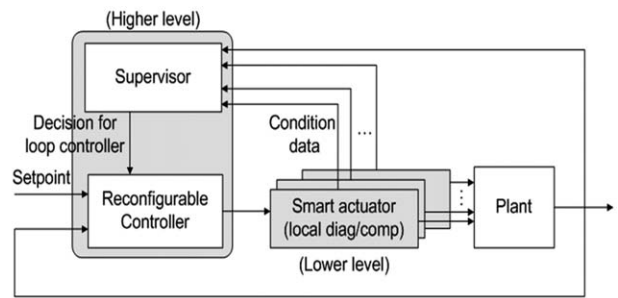


그림 7. 지능형 액추에이터 기반의 NFTCS 구조[13].
Fig. 7. The structure of smart actuator based NFTCS[13].

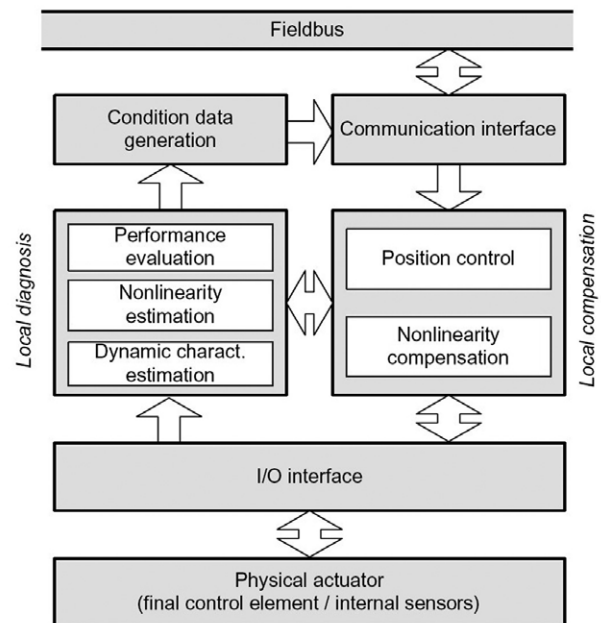


그림 8. 지능형 액추에이터의 구조[21].
Fig. 8. The structure of smart actuator[21].

IV. 결론

본 논문에서는 시스템의 고장발생시 안전성을 향상 시키기 위한 다양한 고장대처 기법들을 소개하고, 이들을 실제 시스템에 적용시 고려할 문제점을 살펴보았다. 아울러 기존 방법의 문제점을 개선하기 위한 대안으로써 네트워크 기반 고장대처 제어의 기본 개념을 소개하였다. 네트워크 기반 고장대처 제어법은 분산 구조를 갖는 제어시스템 구현을 통해서 기존의 중앙 집중식 고장대처제어 기법이 공통적으로 직면하는 문제점, 즉 다중 하드웨어 및 적절한 고장진단 정보 확보의 어려움, PID 기반 제어시스템 등의 한계를 효과적으로 극복할 수 있을 것으로

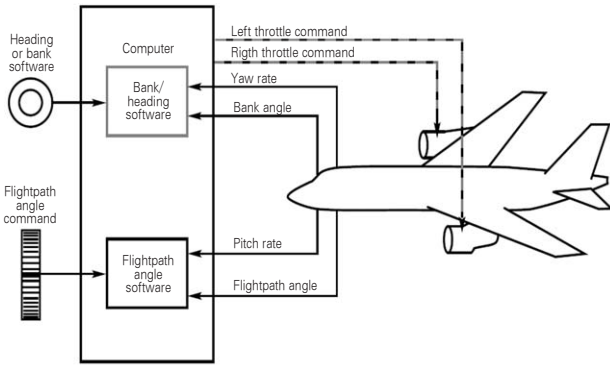


그림 9. PCA 시스템의 구조[22].
Fig. 9. The structure of a PCA system[22].

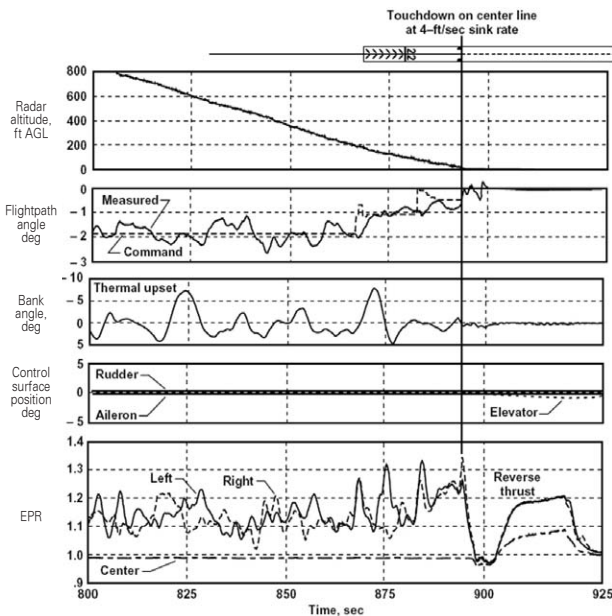


그림 10. PCA 착륙 결과[22].
Fig. 10. The results of a PCA landing[22].

기대된다. 이를 통해서 전통적으로 고장대처제어가 집중적으로 적용되던 항공우주 산업 뿐 아니라, 자동차, 로봇, 공장자동화 등 다양한 산업 분야에 걸쳐 시스템 안전성 및 신뢰도 개선, 제품의 품질 개선, 유지보수 비용 절감 등의 효과가 기대되므로, 이에 대한 지속적인 연구가 필요하다.

참고문헌

- [1] R. Isermann, and P. Ballé, "Trends in the application of model-based fault detection and diagnosis of technical processes," *Control Engineering Practice*, vol. 5, no. 5, pp. 709-719, 1997.
- [2] M. Job, *Air Disaster*, Aerospace Publications Pty Ltd., vol. 2, 1996.
- [3] *Civil aviation safety data 1993-2007*. Technical report, Civil Aviation Authority of the Netherlands(CAA-NL), 2008.
- [4] Y. Zhang, and J. Jiang, "Bibliographical review on reconfigurable fault-tolerant control systems," In Proc. of the 5th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes 2003, Washington, D. C., USA, pp. 265-276, 2003.
- [5] F. Liao, J. L. Wang, and G. -H. Yang, "Reliable robust flight tracking control: an LMI approach," *IEEE Transactions on Control Systems Technology*, vol. 10, no. 1, pp. 76-89, 2002.
- [6] Q. Zhao, and J. Jiang, "Reliable state feedback control systems design against actuator failures," *Automatica*, vol. 34, no. 10, pp. 1267-1272, 1998.
- [7] I. Yang, "Networked fault-tolerant control using robust nonlinear dynamic inversion and control allocation with aerospace application," PhD Thesis, Kyungpook National University, 2011.
- [8] Z. Gao, and P. J. Antsaklis, "Stability of the pseudo-inverse method for reconfigurable control systems," *Int. J. of Control*, vol. 53, no. 3, pp. 717-729, 1991.
- [9] J. Jiang, "Design of reconfigurable control system using eigenstructure assignment," *International Journal of Control*, vol. 59, pp. 395-410, 1994.
- [10] K. J. Åström, and B. Wittenmark, *Adaptive Control, 2nd Ed.* Addison-Wesley Publishing Company, Inc., 1995.
- [11] D. Kim, and Y. Kim, "Robust Variable Structure Controller Design for Fault Tolerant Flight Control," *Journal of Guidance, Control, and Dynamics*, vol. 23, no. 3, pp. 430-437, 2000.
- [12] J. D. Bošković, and R. K. Mehra, "A Multiple Model-Based Reconfigurable Flight Control System Design," *Proc. the 37th IEEE Conference on Decision & Control*, pp. 4503-4508, Dec. 1998.
- [13] I. Yang, D. Kim, and D. Lee, "Fault-tolerant control strategy based on control allocation using smart actuators," in *Proc. Conf. Control and Fault-Tolerant System, Nice*, pp. 377-381, 2010.
- [14] J. J. Burken, P. Lu, Z. Wu, and C. Bahm, "Two reconfigurable

flight-control design methods: robust servomechanism and control allocation," *Journal of Guidance, Control, and Dynamics*, vol. 24, no. 3, pp. 482-493, 2001.

[15] C. Dailly, "Fault monitoring and diagnosis," *Computing and Control Engineering Journal*, vol. 1, no. 2, pp. 57-62, 1990.

[16] W. Zhang, M. S. Branicky, and S. M. Phillips, "Stability of networked control systems," *IEEE Cont. Syst. Mag.*, vol. 21, no. 1, pp. 84-99, Feb., 2001.

[17] I. Yang, S.-Y. Kang, and D. Lee, "Compensating transmission delay and packet loss in networked control system for unmanned underwater vehicle," *Journal of IEMEK*, vol. 6, no. 3, pp. 149-156, Jun., 2011.

[18] Y. Tipsuwan, and M. -Y. Chow, "Control methodologies in networked control systems," *Control Engineering Practices*, vol. 11, pp. 1099-1111, 2003.

[19] C. Kambhampati, R. J. Patton, and F. J. Uppal, "Reconfiguration in networked control systems: fault tolerant control and plug-and-play," *6th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes, Tsinghua University, P. R. China*, pp. 151-156, 2006.

[20] J. D. Bošković, S. E. Bergstrom, and R. K. Mehra, "Retrofit reconfigurable flight control in the presence of control effector damage," *2005 American Control Conference Portland, OR, USA*, pp. 2652-2657, June 2005.

[21] D. Lee, "Distributed real-time fault-tolerant control using smart actuators and time-triggered communication," *PhD Thesis, Dept. Automatic Control & Systems Eng., Sheffield University, UK*, 2002.

[22] F. W. Burcham, Jr., T. A. Maine, and J. Bull, "Using engine thrust for emergency flight control: MD-11 and B-747 Results," *NASA/TM-1998-206552*, May, 1998.

[23] J. C. Ponsart, D. Sauter, and D. Theilliol, "Control and fault

diagnosis of winding machine based on a LTV model," *In Proc. of the 2005 IEEE Conference on Control Applications, Toronto, Canada*, pp.1642-1647, 2005.

[24] D. Kim, K. Kang, and D. Lee, "Fault-tolerant synchronized motion control for crane system using wireless sensor network," *In Proc. of the 7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes Barcelona, Spain*, pp.989-994, 2009.

● 저 자 약 력



양인석

- 2001년 2월 경북대학교 수학과(이학사).
- 2005년 2월 경북대학교 대학원 산업응용수학과(이학석사).
- 2011년 8월 경북대학교 대학원 산업응용수학과(이학박사).
- 2011년 10월~현재 경북대학교 IT대학 국방수중통신/탐지특화연구센터 박사후연수연구원.

· 관심분야 : 항공제어, 고장진단, 고장대처제어.



김동길

- 2006년 2월 경북대학교 전자전기컴퓨터학부(공학사).
- 2008년 2월 경북대학교 대학원 전자공학과(공학석사).
- 2008년 3월~현재 경북대학교 대학원 전자전기컴퓨터학부 박사과정 재학중.

· 관심분야 : 고장대처제어, 네트워크 기반 제어.



이동익

- 1987년 8월 경북대학교 전자공학과(이학사).
- 1990년 2월 경북대학교 대학원 전자공학과(공학석사).
- 1990년 3월~1997년 8월 국방과학연구소 연구원.
- 1997년 9월~2002년 4월 영국 셰필드대학교 자동제어시스템공학과(공학박사).

· 2002년 1월~2005년 3월 영국 DRTS Ltd 공동설립 및 CTO.
· 2005년~현재 경북대학교 전자공학부 조교수.