

# 네트워크 기반 Steer-by-wire 시스템을 위한 지능형 액츄에이터 제어

이 동 익<sup>†</sup>

## Control of smart actuators for networked steer-by-wire system

Dongik Lee<sup>†</sup>

### Abstract

This paper presents a control algorithm for smart actuators that can be used in a 'steer-by-wire' system for next-generation intelligent automobiles. 'Steer-by-wire' is a network-based mechatronic steering system in which mechanical linkages and hydraulics are replaced by electric motors and a digital communication network. The algorithm is designed not only to achieve the capabilities of following the desired steering angles while minimizing the discrepancy between two steering wheels, but also to tolerate an actuator with faults. The proposed algorithm is based on a simple proportional control so that it can be implemented in real-time with a low-cost embedded microcontroller. The effectiveness of the control algorithm is examined using a hardware-in-the-loop simulator consisting of two DC motors and a CAN bus network.

**Key Words** : smart actuator, intelligent vehicle, steer-by-wire, automatic control, networked control

## 1. 서 론

1990년대 이후 지능형 센서, 액츄에이터 및 필드버스(fieldbus)를 이용한 네트워크 기반 임베디드 제어 기술이 산업계 전반에 걸쳐 폭넓게 적용되고 있다. 최근에는 고속철도차량<sup>1,2)</sup> 등 고도의 안전성과 신뢰도가 요구되는 안전중요 시스템(safety-critical systems) 분야에서도 네트워크 기반 제어기술을 적용하기 위한 연구와 노력이 급속히 진행되고 있다. 대표적인 사례로써, 차세대 지능형 자동차를 위한 'x-by-wire' 시스템을 들 수 있다. 'X-by-wire' 시스템이란 기존의 조향, 제동, 변속 등 차량제어에 필요한 기계적 연결이나 유압장치를 전기 모터와 필드버스로 대체한 네트워크 기반 차량 제어 시스템을 통칭하는 말이며, 이 가운데 네트워크 기반 조향장치를 'steer-by-wire'라고 부른다<sup>3-5)</sup>. 이러한 필드버스와 지능형 부품을 이용한 차량제어 시스템은 개발, 통합, 유지보수, 수정변경, 안전성, 연료효율, 조종성 등에 있어서 획기적인 장점을 얻을 수 있다.

본 논문에서는 steer-by-wire 시스템을 위한 지능형

액츄에이터 제어 알고리즘을 제안한다. 지능형 액츄에이터란, 물리적 힘을 제공하는 구동기(예: 모터, 유압실린더 등), 마이크로컨트롤러, 신호변환, 부가적인 센서, 그리고 필드버스 인터페이스가 하나의 모듈로 제작되어 자체적으로 진단/보상/제어를 수행하고 양방향 통신을 지원할 수 있는 액츄에이터 시스템을 가리킨다<sup>6)</sup>. 특히 지능형 액츄에이터를 이용하면 제어시스템의 안전성과 신뢰도 요구수준을 효과적으로 개선시킬 수 있다<sup>3,7)</sup>.

그러나 기존의 중앙집중식 제어시스템과 달리, 네트워크 기반 제어시스템은 불규칙한 통신지연과 분산처리로 인하여 시스템 설계 및 분석에 있어서 여러 가지 어려움이 따른다. 특히 통신지연 문제를 해결하기 위한 연구가 많이 수행되었는데, 관측기를 이용한 방법, 메시지 큐(queue)를 이용한 방법 등이 있다<sup>8-10)</sup>. 그러나 이러한 접근법을 이용하여 설계한 제어기들은 일반적으로 복잡하고 모델링 오차에도 매우 민감하다. 따라서 지능형 액츄에이터와 같은 저비용 임베디드 시스템에 적용할 경우, 실시간 구현 등에서 많은 어려움이 따른다. 그런데 최근 자동차 산업계를 중심으로 이루어진 연구결과에 따르면, 시간분할(time-triggered) 방식으로 시스템을 구현할 경우 불규칙한 통신지연과 분산처리 및 노드들 사이의 시간동기(clock synchronization) 문

경북대학교 전자전기컴퓨터학부 (School of Electrical Engineering and Computer Science, Kyungpook National University)

<sup>†</sup>Corresponding author: dilee@ee.knu.ac.kr

(Received : July 21, 2006, Accepted : August 21, 2006)

제점들을 효과적으로 극복할 수 있을 뿐 아니라, 시스템 설계분석과 성능평가에 있어서 매우 큰 장점을 기대할 수 있다<sup>[11,12]</sup>. 그 결과로써, 이미 자동차 산업계는 반드시 시간분할 방식을 적용하여 x-by-wire 시스템을 구현하도록 요구하고 있으며, 이를 위해서 TTP/C<sup>[13]</sup>, TTCAN<sup>[14]</sup>, FlexRay<sup>[15]</sup> 등 시간분할 방식의 통신 프로토콜이 개발되었다. 따라서 본 논문에서도 시간분할 방식을 통해서 통신 지연과 시간 동기 문제가 해소되었다고 가정하고 제어 알고리즘을 설계한다.

본 논문에서는 먼저 지능형 액츄에이터를 위한 제어 시스템의 주요 요구사항 및 제한사항으로써 다음과 같은 네 가지 항목을 도출하였다.

- i) 각 조향바퀴의 명령각 추종성능
- ii) 두 조향바퀴 사이의 동기(synchronized motion) 유지
- iii) 고장 대처 능력(fault-tolerance)
- iv) 저비용 마이크로컨트롤러를 이용한 실시간 구현.

이어서, 시간분할 방식으로 시스템이 구현되었다는 가정 하에서, 네 가지 요구사항을 만족시킬 수 있도록 비례제어(proportional control) 기법을 바탕으로 매우 간단하고 효율적이며 고장상황에도 대처할 수 있는 제어 알고리즘을 설계한다. 끝으로 DC모터, 16비트 마이크로컨트롤러, CAN버스<sup>[16]</sup> 등으로 구성된 실험장치를 이용하여 제한한 알고리즘의 성능과 효율성을 검증한다.

## 2. Steer-by-wire 시스템 구조

Steer-by-wire 시스템의 구조는 그림 1과 같이 크게 두 가지로 분류할 수 있다. 그림 1(a)에 제시한 구조는 네트워크 기반 차량 제어시스템의 잇점을 극대화한 것으로써, 설치, 공간활용, 정비보수 등에서 유리하다. 단점으로는, 두 개의 독립적인 액츄에이터 및 제어장치(ECU)가 통신을 통해 제어해야 하므로, 제어알고리즘 설계가 어렵고, 고장대처를 위한 하드웨어 여유도를 확보하는 것도 상대적으로 어렵다. 반면에 그림 1(b)에 제시한 구조는, 기계적인 연결이나 유압장치를 최소화하려는 steer-by-wire 시스템의 장점이 반감되는 측면이 있으나, 단일 제어기에 의해서 동작하므로 제어 알고리즘 설계는 비교적 용이하고 액츄에이터 고장에 대처하기 위한 하드웨어 여유도 확보도 효율적이다. 그러나 고도의 안전성 요구사항을 만족시킬 수 있는 제어 시스템 설계의 어려움 등으로 인하여 산업계에서조차 실험실 수준의 연구만을 진행하고 있을 뿐, steer-by-wire 시스템은 아직까지 상용화된 사례가 없으며, 두

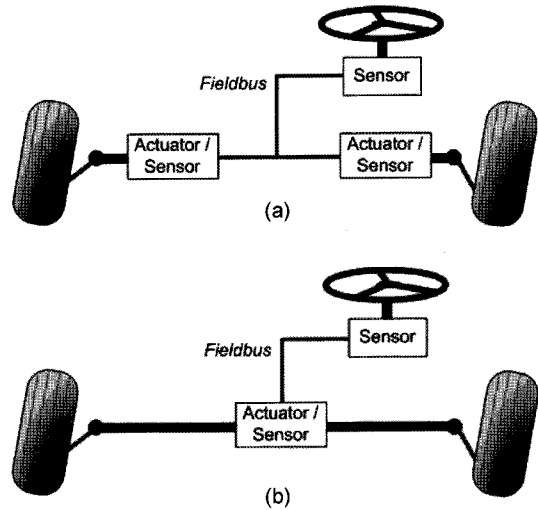


그림 1. Steer-by-wire 시스템 구현 개념  
Fig. 1. Implementation concepts for steer-by-wire systems.

가지 구조에 대한 연구가 지금도 진행 중에 있다. 본 연구에서는 제어기 설계가 어렵지만 보다 이상적인 steer-by-wire 시스템인 그림 1(a) 구조를 가정하였으며, 지능형 액츄에이터와 제한한 제어 알고리즘을 적용함으로써 하드웨어 여유도가 없어도 부분적인 고장대처가 가능함을 보인다.

## 3. 지능형 액츄에이터 제어 시스템 설계

### 3.1. 제어시스템 요구사항 분석

Steer-by-wire 시스템을 위한 액츄에이터 제어장치를 설계하기 위해서 먼저 다음과 같은 제어목표 및 제한사항을 설정하였다:

- i) 각 조향바퀴의 각도는 조향핸들로부터 입력된 명령각을 추종해야 함
- ii) 두 개의 조향바퀴 각도는 언제나 서로 동일한 값을 가져야 함
- iii) 고장 발생시 안전한 상태로 전환되거나 제한적인 주행이 가능해야 함
- iv) 저비용 마이크로컨트롤러를 이용하여 실시간 구현이 가능해야 함

항목 i)은 일반적인 제어시스템의 요구사항으로써, 조향명령각과 각 바퀴의 실제 조향각 사이의 정상상태 오차 허용범위로 표현할 수 있다. 반면에 항목 ii)는 과도상태를 포함하여 임의의 시점에서 두 조향바퀴각은 동일해야 함을 의미하며, 이 조건은 안전한 주행을 위해서 반드시 필요하다. 두 조향바퀴가 서로 다른 방향

을 향할 경우 차량에 진동이 발생하거나 사고로 이어질 수 있기 때문이다. 항목 iii)의 경우 허용 고장의 종류와 고장 정도에 따라서 매우 어렵고 복잡한 문제가 될 수 있으나, 본 논문에서는 액츄에이터의 동적특성이 일부 저하되는 고장에 대해서만 다룰 예정이다. 마지막으로 항목 iv)는 대량 생산 임베디드 제어 시스템 개발 시 공통적으로 적용되는 제약사항이라고 볼 수 있는데, steer-by-wire 제어시스템 역시 연산속도, 메모리, 통신 대역폭 등에 있어서 매우 엄격한 제한조건 하에서 시스템을 구현하게 된다. 그러므로 본 논문에서는 세 가지 요구사항을 만족하되, 자동차에 많이 적용되고 있는 16비트 마이크로컨트롤러와 CAN 버스를 이용하여 구현이 가능하도록 매우 간결하고 효율적인 제어 알고리즘을 설계한다.

위에서 제시한 요구사항 i)과 ii)는 다음과 같이 식 (1)로 나타낼 수 있다.

$$\theta_{e1} = |\theta_d - \theta_1| < R \tag{1a}$$

$$\theta_{e2} = |\theta_d - \theta_2| < R \tag{1b}$$

$$\delta\theta = |\theta_1 - \theta_2| < \Delta \tag{1c}$$

여기서  $\theta_1$ 과  $\theta_2$ 는 그림 2에 나타낸 것과 같이 각 조향바퀴의 조향각을 가리키며,  $R$ 은 조향명령각과 실제 조향각 사이의 정상상태오차 허용범위,  $\Delta$ 는 두 조향바퀴각 사이의 최대 허용편차를 나타낸다. 그런데 조향바퀴각은 조향바퀴를 구동하는 두 액츄에이터의 스트로크(stroke) 길이에 의해서 결정되므로, 편의상 본 논문에서는 핸들에서 입력된 조향명령각을 스트로크 길이로 변환한 뒤 액츄에이터 제어기로 전송하고, 액츄에이터 제어기에서는 스트로크 길이 측정값만을 사용하여 제어하기로 한다.

두 개의 액츄에이터가 동기를 유지하면서 각각의 스트로크를 제어하기 위해서는 두 액츄에이터 스트로크의 길이( $x_1, x_2$ )를 주기적으로 비교할 필요가 있다. 그런데 두 액츄에이터 스트로크는 항상 반대 방향으로 움직이므로  $x_1, x_2$ 의 길이를 직접 비교할 수 없다. 따라서  $x_2$ 의 길이를  $x_1$ 과 동일한 방향으로 운동할 때의 값으로 환산한 보조변수  $x_2^*$ 를 다음과 같이 정의한다:

$$x_2^* = x_2 + (x_1^0 + x_2^0) \tag{2}$$

여기서  $x_1^0, x_2^0$ 는 각 액츄에이터 스트로크의 초기값을 나타낸다.  $x_1 = x_2^*$ 가 만족되면 두 개의 조향바퀴각이 동일함을 알 수 있다. 따라서 식(2)에 나타낸 제어 요구사항은 다음과 같이 표현할 수 있다:

$$e_{p1} = |x_d - x_1| < r \tag{3a}$$

$$e_{p2} = |x_d - x_2^*| < r \tag{3b}$$

$$\delta x = |x_1 - x_2^*| < D \tag{3c}$$

여기서  $r$ 은 각 액츄에이터 스트로크의 정상상태 허용오차이며,  $D$ 는 두 액츄에이터 스트로크 사이의 최대 허용편차를 나타내는 설계인자이다.

### 3.2. 제어시스템 하드웨어 구조

본 논문에서 다루는 steer-by-wire 시스템을 비롯한 네트워크 기반 제어시스템의 경우, 불규칙한 통신 지연과 분산처리로 인해서 제어시스템 설계, 분석, 평가시 많은 어려움이 따른다. 즉 일반적으로 피드백 제어시스템에서는 주기적으로 센서측정값을 읽고, 제어알고리즘을 처리하며, 그 결과를 액츄에이터에 인가하는데, 만약 공간적으로 분리되어 위치한 연산노드(제어기)들 사이에 시간동기가 이루어지지 않았을 경우, 서로 다른 시점에 센서출력을 샘플링하고 이를 제어량 계산에 이용하게 된다. 이러한 문제는 통신지연과 함께 제어시스템의 성능을 저하시키거나, 심지어 시스템을 불안정하게 만들 수도 있다. 또한 시변(time-varying) 시스템 특성을 갖게 됨으로써 제어시스템 설계 및 분석이 매우 어렵게 된다<sup>11)</sup>. 이를 해결하기 위해서 자동차 산업계를 중심으로 시간분할 기술을 적용하여 시스템을 구현하는 기술에 대하여 많은 연구가 진행되었으며, 그 결과로써 TTCAN, TTP/C, FlexRay 등과 같은 시간분할 방식의 고신뢰성 필드버스 프로토콜이 개발되었다. 현재 자동차 산업계에서는 steer-by-wire를 포함하여 x-by-wire 시스템에 대해서 반드시 시간분할 방식으로 시스템을 구현할 것을 요구하고 있다. 이처럼 시간분할 방식으로 시스템을 구현하고 통신을 수행할 경우, 앞서 언급한 전송지연 및 시간동기 관련 문제들을 효과적으로 극복할 수 있게 된다. 따라서 본 논문에서는 시간분할 방식으로 통신 프로토콜과 시스템 구현이 이루어진 steer-by-wire 시스템을 가정하고 지능형 액츄에이터의 제어알고리즘을 설계한다.

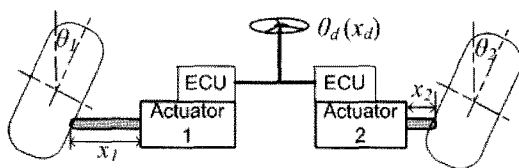


그림 2. Steer-by-wire 시스템을 위한 주요 변수들  
Fig. 2. Key variables of steer-by-wire system.

### 3.3. 제안한 액츄에이터 제어 알고리즘

액츄에이터 스트로크의 길이로 표현된 조향명령각을

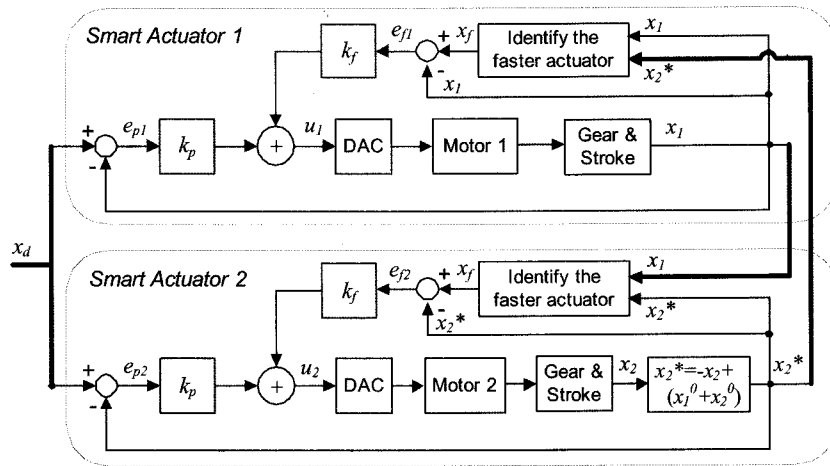


그림 3. 액츄에이터 제어기 블록선도. 굵게 표시한 선은 CAN 버스를 통해서 전송된 메시지를 가리킴  
 Fig. 3. Block diagram of the actuator controller. The bold lines indicate the messages received through a CAN bus network.

추종하면서, 동시에 두 개의 스트로크가 허용오차  $D$  이내를 유지하기 위한 각 액츄에이터의 제어법칙을 다음과 같이 설계하고, 그림 3에 나타내었다.

액츄에이터-1에서 수행되는 제어알고리즘:

$$u_1 = \begin{cases} 0 & \text{if } x_1 = x_f \text{ and } |e_{s_1}| > D \\ k_p e_{p_1} + k_f e_{f_1} & \text{otherwise} \end{cases} \quad (4)$$

여기서,

$$e_{p_1} = x_d - x_1 \quad (5a)$$

$$e_{f_1} = x_f - x_1 \quad (5b)$$

$$e_{s_1} = x_s - x_1 \quad (5c)$$

액츄에이터-2에서 수행되는 제어알고리즘:

$$u_2 = \begin{cases} 0 & \text{if } x_2^* = x_f \text{ and } |e_{s_2}| > D \\ k_p e_{p_2} + k_f e_{f_2} & \text{otherwise} \end{cases} \quad (6)$$

여기서,

$$e_{p_2} = x_d - x_2^* \quad (7a)$$

$$e_{f_2} = x_f - x_2^* \quad (7b)$$

$$e_{s_2} = x_s - x_2^* \quad (7c)$$

$$x_f = \begin{cases} x_1 & \text{if } |e_{p_1}| \leq |e_{p_2}| \\ x_2^* & \text{otherwise} \end{cases} \quad (8)$$

$$x_s = \begin{cases} x_1 & \text{if } |e_{p_1}| \geq |e_{p_2}| \\ x_2^* & \text{otherwise} \end{cases} \quad (9)$$

$x_f$ 와  $x_s$ 는 각각 상대적으로 빠르게 움직이는 액츄에

이터와 느리게 움직이는 액츄에이터의 스트로크 길이를 나타내며, 여기서 빠르게 움직이는 액츄에이터란 명령값에 보다 근접한 것을 의미한다.  $k_f e_f$ 항을  $k_f e_f^2$ 로 변경할 경우 제어특성은 허용오차  $D$ 에 보다 민감하게 됨으로써 좀더 동기화된 제어특성을 기대할 수 있다.

### 3.4. 액츄에이터 특성 추정

Steer-by-wire 시스템은 전형적인 안전중요 시스템으로써, 고장 발생시 치명적인 사고로 연결될 수 있다. 따라서 주행 중 발생하는 액츄에이터 고장에 적절히 대처할 수 있는 기능이 반드시 필요하다. 액츄에이터 동작이 완전히 멈추는 등 심각한 고장이 발생할 경우, 여유 액츄에이터가 없으면 고장에 대처할 수 없으며 이러한 상황은 본 논문에서는 고려하지 않는다. 그러나 치명적인 고장이 아닐 경우, 다소 성능이 저하되거나 제한적인 상황 하에서라도 주행을 계속하거나 가까운 정비소로 안전하게 이동할 수 있는 기능이 반드시 필요하다. 예를 들면, 액츄에이터를 장기적으로 사용하거나 베어링 마모 등 여러 가지 요인으로 인해서 액츄에이터의 반응속도가 느려지는 경우를 생각할 수 있다. 위에서 설계한 비례제어기의 경우 두 액츄에이터 사이의 속도차를 보정하는 항이 포함되어 있으나, 액츄에이터의 특성 차이가 커질 경우에는 차량에 진동을 유발하거나 심지어 사고로 연결될 위험성도 있다. 이러한 상황에서는 조향각을 신속하게 추종하는 기능에 비해서, 두 조향바퀴의 조향각을 일치시키는 능력이 운전자의 안전에 있어서 더욱 중요하다. 따라서 본 논문에서는 이러한 경우에 대처할 수 있도록 각 액츄에이터의

동특성을 나타내는 시상수를 온라인으로 추정하고, 이를 이용하여 보다 느린 액츄에이터의 특성에 맞추어 제어이득( $k_p, k_f$ )을 수정함으로써, 명령값 추종능력은 다소 저하되더라도 두 조향바퀴각의 동기는 유지될 수 있도록 설계하였다.

임의의 액츄에이터가 가용범위 안에서 동작할 경우, 요구값( $u_d$ )와 실제값( $u_a$ ) 사이의 폐루프 전달특성( $G_o$ )은 다음과 같은 일차함수로 표현할 수 있다<sup>6)</sup>:

$$G_o(s) = \frac{U_a(s)}{U_d(s)} = \frac{K_o}{1 + T_o s} \quad (10)$$

시상수  $T_o$ 는 액츄에이터의 동특성을 결정하는 파라미터로써, 고장으로 인해 느려질 경우  $T_o$ 값이 커지게 된다. 시상수  $T_o$ 는 RLS(Recursive Least Squares) 알고리즘을 이용하여 온라인으로 추정한다<sup>18)</sup>.

### 4. 실험 및 고찰

#### 4.1. 실험장비 구성

네트워크 기반 제어 시스템에서는 연산노드들이 공간적으로 분리되어 있으며, 각 노드는 필드버스를 통해서 서로 데이터를 교환하며, 이 때 불규칙한 전송지연 또는 데이터 분실 등이 일어날 수 있다. 이러한 문제들은 제어시스템의 성능과 실시간 특성에 심각한 영향을 미칠 수 있지만, 전통적인 컴퓨터 시뮬레이션 기법으로는 충분한 성능검증에 어려움이 따른다. 이에 대한 효과적인 해결책으로써 HiLS(Hardware-in-the-Loop Simulation) 기법이 많이 이용되고 있다<sup>19)</sup>. HiLS는 일종의 실시간 시뮬레이션 기법으로써, 시뮬레이터의 입출력 신호가 부분적으로 실제 시스템과 동일한 실시간 특성을 재현할 수 있으므로 다양한 실제 조건 및 환경 하에서 실제 임베디드 제어시스템을 시험할 수 있는 기회를 제공한다. 특히 통신지연 또는 메시지 분실 등으로 인한 안정성이나 제어성능을 분석할 수 있을 뿐만 아니라, 샘플링 효과, 측정잡음, 액츄에이터 비선형 특성 등 전체 시스템의 소프트웨어 및 하드웨어를 통합적으로 평가할 수 있다는 장점이 있다. 그리고 다양한 고장상황을 재현함으로써 고장대처 장치의 성능을 반복적으로 정확하게 평가할 수 있는 장점도 제공한다.

본 논문에서는 그림 4 및 그림 5에 나타난 것과 같이 DRTS Ltd사에서 제작한 X-Test<sup>®</sup>를 이용하여 HiLS를 구성하였다. X-Test는 Infineon C167CR 16비트 마이크로컨트롤러와 CAN버스 기반으로 제작된 교육용 장비로써, 포텐서미터(Potentiometer)가 내장된 DC 모터 기반 선형 액츄에이터를 실시간으로 제어할 수 있

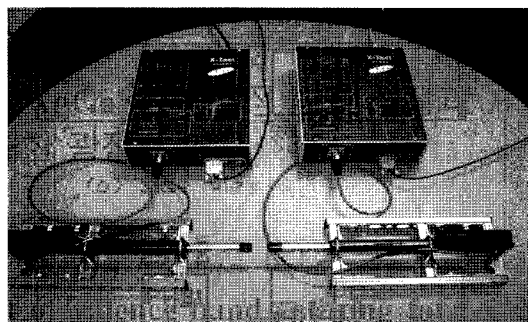


그림 4. X-Test를 이용한 두 개의 지능형 액츄에이터  
Fig. 4. Two smart actuators using the X-Test units.

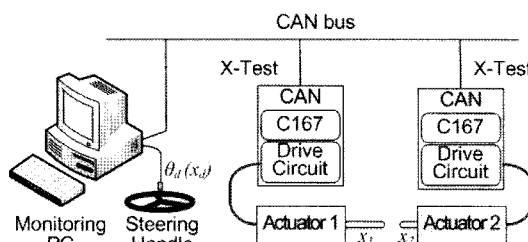


그림 5. 실험 구성도  
Fig. 5. Experimental setup.

는 기능과 여러 가지 입출력 포트를 제공하는 교육용 실험실습 기기이다. 모니터링 PC에서는 조향장치 역할을 수행하는 조이스틱으로부터 조향명령각( $\theta_d$ )을 읽어서 액츄에이터 스트로크 길이명령값( $x_d$ )으로 변환한 뒤, CAN 버스를 통해 2대의 X-Test로 전송한다. X-Test에 내장된 C167CR 프로세서는 제한된 제어 알고리즘을 실시간으로 연산하여 액츄에이터의 스트로크 길이를 제어한다. 제어 및 액츄에이터와 관련된 모든 데이터는 X-Test에 의해서 모니터링 PC로 전송된다.

#### 4.2. 액츄에이터 제어성능

먼저 두 개의 액츄에이터가 정상적인 동적특성을 갖는 경우에 대해서 제안된 알고리즘의 제어성능을 시험하였다. 액츄에이터 스트로크 명령값으로 0~100 mm 범위를 갖는 사인과 신호를 사용하였다. 액츄에이터 제어기의 샘플링 주파수는 25 Hz이며, 두 액츄에이터의 최대허용 편차  $D=5$  mm를 적용하였다. 그림 6(a)는 각 액츄에이터가 명령값을 정밀하게 추종함과 동시에 두 개의 스트로크가 모든 시간에 걸쳐서 동기가 잘 이루어지고 있음을 보여준다. 그림 6(b)는 두 액츄에이터 스트로크의 동기오차가  $\delta_x=1$  mm <  $D$  이내로써 설계사양을 잘 만족하고 있음을 확인할 수 있다. 액츄에이터

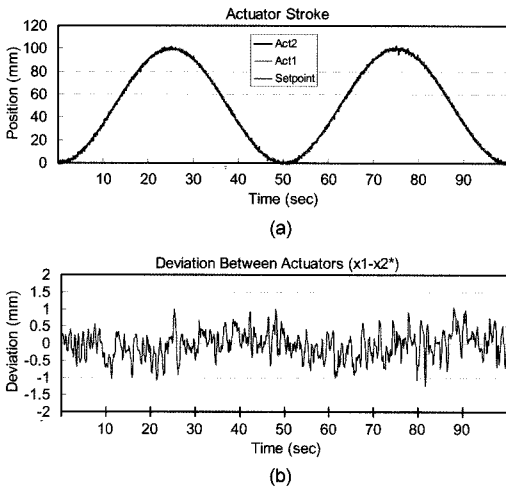


그림 6. 정상적인 조건하에서 제어성능. 명령값 신호  $x_d=50\sin(0.125t)+50$ . (a) 액추에이터 스트로크  $x_1, x_2^*$ , (b) 두 액추에이터 스트로크 사이 편차  $\delta_x$

Fig. 6. Control performance of the actuators in normal condition. Desired position  $x_d=50\sin(0.125t)+50$ . (a) Actuator stroke trajectories,  $x_1, x_2^*$ , (b) Deviation between two strokes,  $\delta_x$ .

에 내장된 포텐서미터의 측정잡음을 충분히 제거하면 보다 나은 제어성능을 얻을 수 있다.

4.3. 액추에이터 지연 대처

액추에이터를 장기적으로 사용하거나 일부 부품의 결합으로 인해서 액추에이터의 입출력 특성이 저하될 수 있으며, 백러쉬, 동작범위 감소 등과 같은 비선형성이나, 반응속도 저하등과 같은 현상을 예로 들 수 있다. 본 논문에서는 두 액추에이터의 반응속도가 현격한 차이를 보이는 상황을 가정하고 제안된 제어알고리즘의 성능을 시험하였다. 고장상황은 X-Test에서 제공하는 모터 구동전압 변경 기능을 이용하여 재연하였다. 즉  $t=100\text{ sec}$ 에서 액추에이터-2의 반응속도가 느려지도록 인위적으로 고장을 인가하였으며, 이때 액추에이터의 시상수는 3.4절에서 언급한 RLS 알고리즘을 이용하여 추정하고 그 결과를 그림 7에 나타내었다. 액추에이터의 시상수는 정상적인 경우 약  $T_d=32\text{ sec}$ 이며, 고장인가 후에는  $T_d=66\text{ sec}$ 로써 약 2배로 증가되었다.

그림 8(a) 및 (b)에서 보는 것처럼 시간  $t=100\text{ sec}$ 에서 액추에이터-2에 고장이 인가됐으나 액추에이터 제어이득을 수정하지 않은 경우, 두 액추에이터 사이의 오차가 최대 10 mm 이상으로 크게 나타났으며 이는 설계사양  $D=5\text{ mm}$ 을 초과하는 값이다. 이에 반해서, 그림 9(a) 및 (b)는 온라인으로 추정된 시상수 값을 이

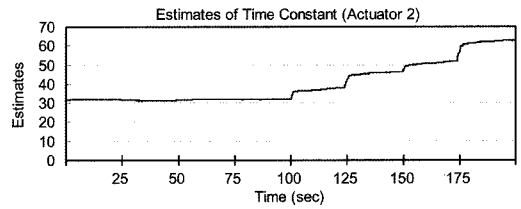


그림 7. 액추에이터-2의 시상수 추정  
Fig. 7. Estimates of the time constant of actuator-2.

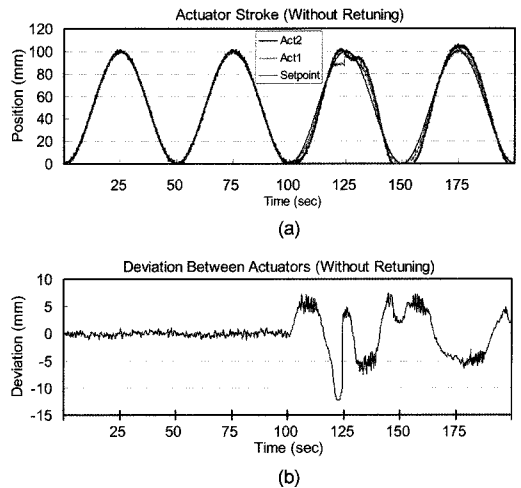


그림 8. 시간  $t=100$ 초에서 액추에이터-2의 고장발생시 제어성능 (제어이득 수정 없음). (a) 액추에이터 스트로크  $x_1, x_2^*$ , (b) 두 액추에이터 스트로크 사이 편차  $\delta_x$

Fig. 8. Control performance without retuning in the presence of fault with actuator-2 at  $t=100\text{ sec}$ . (a) Actuator stroke trajectories,  $x_1, x_2^*$ , (b) Deviation between two strokes,  $\delta_x$ .

용하여 제어이득( $k_p, k_f$ )을 수정한 경우에는 오차가 5 mm 이내에서 유지됨을 보여준다. 새로운 제어이득 결정은 경험적인 방법을 이용하였다. 즉 느려진 액추에이터 특성으로 인한 오버슈터 및 속도포화 상태가 발생하는 것을 완화시키기 위해  $k_p$ 값을 정상적인 이득의 1/10로 줄였으며, 두 액추에이터 사이의 편차를 줄이기 위해서  $k_f$ 값은 10배로 증가시켰다. 그 결과 그림 9(a) 및 (b)에서 보는 것처럼, 스트로크 명령값 추정속도는 다소 느려졌으나 두 액추에이터 사이의 편차는 허용범위 이내로 유지되었음을 알 수 있다. 실제 차량 주행 중에 이러한 상황이 발생할 경우 제어이득 수정을 하지 않은 경우에는 사고로 연결될 가능성이 큰 반면에, 제안된 제어방법을 적용할 경우 주행속도를 감소시키고 핸들조작을 부드럽게 함으로써 계속 주행이 가능할

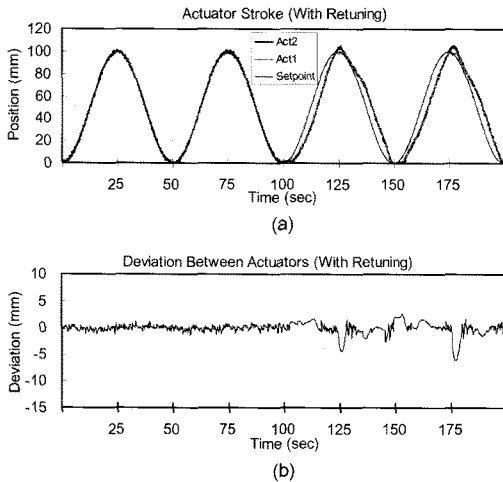


그림 9. 시간  $t=100$ 초에서 액츄에이터-2의 고장발생시 제어 성능 (제어이득 수정). (a) 액츄에이터 스트로크  $x_1, x_2^*$ , (b) 두 액츄에이터 스트로크 사이 편차  $\delta_x$   
 Fig. 9. Control performance with retuning in the presence of fault with actuator-2 at  $t=100$  sec. (a) Actuator stroke trajectories,  $x_1, x_2^*$ , (b) Deviation between two strokes,  $\delta_x$ .

것으로 판단된다.

## 5. 결 론

Steer-by-wire 시스템은 전형적인 네트워크 기반 안전중요 임베디드 시스템으로써, 필드버스로 연결된 2개의 지능형 액츄에이터에 의해서 구동되므로 불규칙한 통신지연과 분산처리의 어려움이 따른다. 본 논문에서는 steer-by-wire 시스템을 위한 지능형 액츄에이터 제어 알고리즘을 제안하고 그 효용성을 시험하였다. 먼저 시간분할 방식의 시스템 구조를 가정함으로써 통신지연과 시간동기 문제를 배제하였다. 그리고 이러한 시간분할 시스템 조건 하에서 동작할 수 있는 매우 간단하고 효율적이며 고장상황에도 대처할 수 있는 지능형 액츄에이터 제어 알고리즘을 제시하였다. 참고로 시간분할 시스템에 관한 가정은 이미 자동차 산업계에서 채택하고 있는 시간분할 방식의 통신 프로토콜을 사용함으로써 쉽게 만족시킬 수 있다.

제안한 알고리즘의 효용성을 분석하기 위해서 2개의 DC모터 기반 선형 액츄에이터, 제어 알고리즘을 처리하는 16비트 마이크로컨트롤러, 그리고 자동차 산업계의 표준 필드버스인 CAN을 이용하여 실험을 수행하였다. 실험을 통해서 두 개의 액츄에이터가 엄격한 동기를 유지하면서 조향명령값을 적절히 추종함을 확인

하였다. 이와 더불어 하나의 액츄에이터가 고장으로 인해 성능이 저하된 상황에서도 액츄에이터의 시상수 추정을 이용한 제어이득 수정을 통해 하드웨어 여유 액츄에이터 없이도 제한적으로 조향장치를 운용할 수 있는 가능성을 확인하였다.

제한한 제어 알고리즘을 개선하기 위해서는 다중 센서 및 액츄에이터가 포함된 시스템에서 중대한 고장이 발생하였을 때 여유 부분을 이용하여 적절히 시스템을 재구성함으로써 고장에 대처할 수 있는 기법과, PID 제어 이득값을 온라인으로 조절하는 방법에 관한 연구가 필요하다.

## Nomenclature

$D$	Maximum allowed jitter of $\delta_x$
$e_{fi}$	Position error of actuator- $i$ from the fast actuator
$e_{pi}$	Position error of actuator- $i$ from the demand position
$e_{si}$	Position error of actuator- $i$ from the slow actuator
$k_p$	Control parameter for position error $e_{pi}$
$k_f$	Control parameter for position error $e_{fi}$
$x_1$	Stroke position of actuator-1
$x_2$	Stroke position of actuator-2
$x_2^*$	Converted stroke position of actuator-2
$x_1^0$	Initial stroke position of actuator-1
$x_2^0$	Initial stroke position of actuator-2
$x_d$	Desired stroke position of actuators
$x_f$	Stroke position of the fast actuator
$x_s$	Stroke position of the slow actuator
$\theta_1$	Actual angle of steering wheel-1
$\theta_2$	Actual angle of steering wheel-2
$\theta_d$	Desired angle of steering wheels
$\delta_x$	Distance between the strokes of actuator-1 and -2

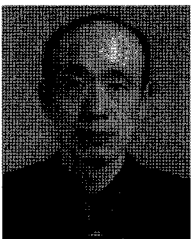
## 감사의 글

이 논문은 2005년도 경북대학교 학술진흥연구비에 의하여 연구되었음.

## 참고 문헌

- [1] 김석원, 김영국, 박태원, "한국형 고속전철의 디스크

- 제동력 측정방법에 관한 연구”, *센서학회지*, vol. 13, no. 3, pp. 244-251, 2004.
- [2] 김석원, 한영래, 최강윤, 김영국, 김종영, 박태원, “걸인제동 계측시스템 개발”, *센서학회지*, vol. 13, no. 4, pp. 115-121, 2004.
- [3] R. Isermann, R. Schwarz, and S. Stoelzl, “Fault-tolerant drive-by-wire systems”, *IEEE Control Systems Magazine*, pp. 64-81, October 2002.
- [4] G. Leen and D. Heffernan, “Expanding automotive electronic systems”, *IEEE Computers*, vol. 35, no. 1, pp. 88-93, Jan. 2002.
- [5] R. Isermann, “Mechatronic systems-innovative products with embedded control”, *Proc. 16th IFAC World Congress*, pp. 52-68, Prague, Czech Republic, July 3-8, 2005.
- [6] R. Isermann and U. Raab, “Intelligent actuators ways to autonomous actuating systems”, *Automatica*, vol. 19, no. 5, pp. 1315-1331, 1993.
- [7] D. Lee, J. Allan, H. A. Thompson, and S. Bennett “PID control for a distributed system with a smart actuator”, *Control Engineering Practice*, vol. 9, pp. 1235-1244, 2001.
- [8] R. Luck and A. Ray, “An observer-based compensator for distributed delays”, *Automatica*, vol. 26, no. 5, pp. 903-908, 1990.
- [9] H. Chan and U. Ozguner, “Closed-loop control of systems over a communications network with queues”, *International Journal of Control*, vol. 62, no.3, pp. 493-510, 1995.
- [10] T. C. Yang, “Networked control system: A brief survey”, *IEE Proc. Control Theory Appl.*, vol. 153, no. 4, pp. 403-412, July 2006.
- [11] H. Kopetz, *Real-time systems: design principles for distributed embedded applications*, Kluwer, 1997.
- [12] J. Rushby, “A comparison of bus architecture for safety-critical embedded systems”, *NASA/CR, Tech. Rep. NASA/CR-2003-212161*, Mar. 2003.
- [13] TTTech Computertechnik GmbH. *Time-triggered protocol TTP/C, High-Level Specification Document*, Available: <http://www.tttech.com>
- [14] *Road Vehicles-Controller Area Network (CAN)-Part 4: Time-Triggered Communication*, ISO 11898-4, 2000.
- [15] *FlexRay Requirement Specification*, ver 2.0.2, April 2002, Available: <http://www.flexray.com>
- [16] *CAN specification (ver. 2.0)*, Robert Bosch GmbH, Postfach 50, D-7000 Stuttgart 1, Germany, 1991.
- [17] J. Nilsson, “Real-time control systems with delays”, *PhD Thesis, Lund Institute of Technology*, Sweden, 1998.
- [18] J. Ljung and T. Soderstrom, *Theory and practice of recursive identification*. MIT Press, Cambridge, Massachusetts, USA, 1983.
- [19] R. Isermann, J. Schaffnit, and S. Sinsel, “Hardware in the loop simulation for the design and testing of engine control systems”, *Control Engineering Practice*, vol. 7, pp. 643-653, 1999.



## 이 동 익

- 1987년 8월 경북대학교 전자공학과(공학사)
- 1990년 2월 경북대학교 대학원 전자공학과 (공학석사)
- 1990년 3월~1997년 8월 국방과학연구소 연구원
- 1997년 9월~2002년 4월 영국 셰필드대학교 자동차제어시스템공학과(공학박사)
- 2002년 1월~2005년 3월 영국 DRTS Ltd 공동설립 및 CTO
- 2005년 4월~현재 경북대학교 전자전기 컴퓨터학부 전임강사