

분산 실시간 제어 시스템을 위한 TTP 네트워크 시스템의 구현

Implementation of TTP Network System for Distributed Real-time Control Systems

김 만 호, 손 병 점, 이 석*, 이 경 창

(Man Ho Kim, Byeong Jeom Son, Suk Lee, and Kyung Chang Lee)

Abstract : Recently, many ECUs(Electronic Control Units) have been used to enhance the vehicle safety, which leads to a distributed real-time control system. The distributed real-time control system requires to reduce the network delay for dependable real-time performance. There are two different paradigms by which a network protocol operates: event-triggered and time-triggered. This paper focuses on implementation of a time-triggered protocol, i.e. TTP/C(Time-Triggered Protocol/class C). This paper presents a design method of TTP control network and performance evaluation of distributed real-time control system using TTP protocol.

Keywords : time-triggered protocol, event-triggered protocol, Time-Triggered Protocol/C(TTP/C), distributed control system, rapid prototyping, x-by-wire system

1. 서론

최근, 자동차의 안전과 편의에 대한 고객의 요구가 증가함에 따라, 지능형 자동차에 관심이 급속히 증대되고 있다. 고객의 요구를 만족시키기 위해 자동차 회사는 다양한 센서와 제어 알고리즘이 포함된 전자 제어식 안전성 프로그램(Electronic Stability Program, ESP)이나 순항 제어 시스템(Advance Cruise Control, ACC)과 같은 첨단 안전 시스템을 지능형 자동차에 분산하여 적용하고 있다. 특히, 지능형 자동차에 적용되는 첨단 안전 시스템이 증가함에 따라, 각각의 전자 제어 유닛(Electronic Control Unit, ECU)이 독립적으로 태스크(task)를 수행하고 네트워크를 통하여 그 정보를 공유하는 방식인 분산 제어 시스템에 대한 연구가 활발히 진행되고 있다[1,2].

그림 1은 자동차 주행에 필요한 채시(chassis) 시스템을 네트워크를 기반으로 하여 실시간 분산 제어를 수행할 수 있는 분산 통합 채시 제어 시스템의 구조를 나타내고 있다. 그림에서, 채시 제어를 위하여 필요한 주행, 제동 및 조향 기능을 담당하는 각각의 ECU는 주어진 태스크에 따라 센서 및 액추에이터를 제어하며, 그 정보는 네트워크를 통해 다른 ECU와 공유되고 있다. 특히, 지능형 자동차에 많은 첨단 시스템이 적용되면서 채시 제어 시스템은 바디 제어 시스템 및 정보 제어 시스템과 게이트웨이를 통해 더 많은 차량 정보를 공유하게 된다. 그러나, 채시 시스템과 같은

실시간 제어 시스템에서는 제한된 시간 내에 주어진 태스크가 수행될 수 있어야 하고 네트워크에 의한 전송 지연이

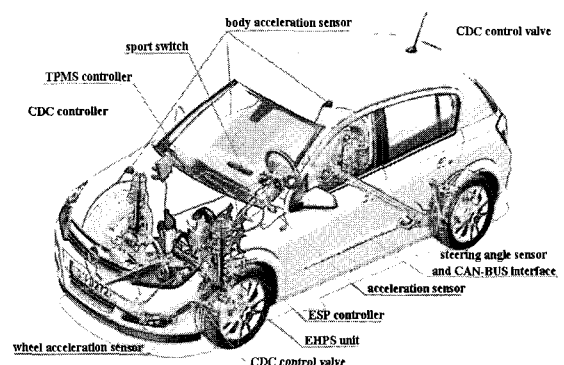


그림 1. 분산 통합 채시 제어 시스템의 개념도.

Fig. 1. Schematic diagram of distributed integrated chassis control system.

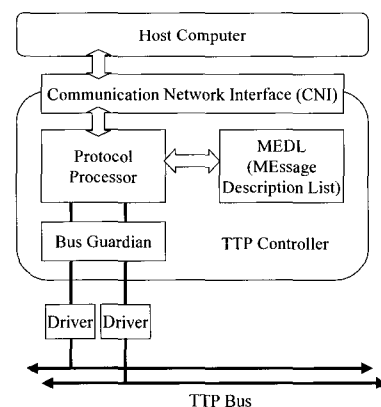


그림 2. TTP 노드의 구조.

Fig. 2. Schematic diagram of TTP node.

* 책임저자(Corresponding Author)

논문접수 : 2007. 1. 8., 채택확정 : 2007. 3. 6.

김만호 : 대구경북과학기술연구원 메카트로닉스연구부 지능형 자동차 연구팀(mhkim@dgist.ac.kr)

손병점, 이석 : 부산대학교 기계공학부

(maligy@pnu.edu/slee@pnu.edu)

이경창 : 부경대학교 제어자동화공학과(gclee@pknu.ac.kr)

※ 본 논문은 산업자원부에서 시행하는 자동차부품기반기술개발사업(사업명: 네트워크 기반 분산형 실시간 제어 시스템 설계 기반 기술 개발)으로 지원받아 수행하였음.

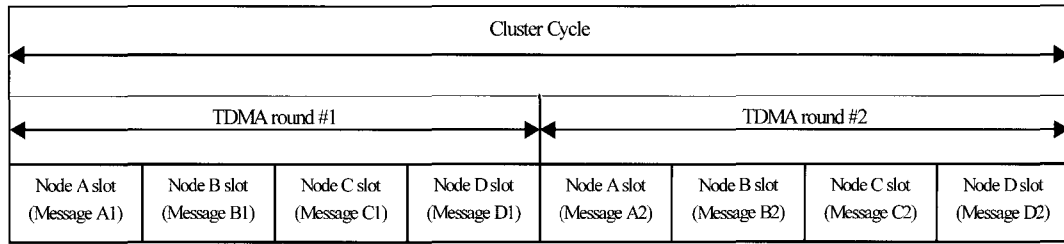


그림 3. TTP 프로토콜의 클러스터 사이클의 예.
Fig. 3. Example of cluster cycle of TTP protocol.

일정한 시간 이내가 되어야 한다[3,4].

일반적으로 제어 시스템에 적용되는 네트워크는 동작 방법에 따라 event-triggered 방식과 time-triggered 방식으로 구분된다. event-triggered 방식은 사건이 발생할 때 시스템이 작동하는 방식으로써, 식별자(identifier)를 부여하여 메시지를 구별하고 우선순위를 정하는 내용 기반 주소 지정 방식(content based addressing) 방식을 사용하는 CAN(Controller Area Network)이 대표적인 프로토콜이다. 이에 반하여, time-triggered 방식은 사건의 발생에 상관없이 정해진 시간이 되면 태스크가 작동하는 방식으로, 일정한 시간 주기(time slot)에 따라 메시지를 전송하는 TDMA(Time Division Multiple Access) 방식을 사용한다. Time-triggered 방식에는 TTP/C(Time Triggered Protocol/Class C), TTCAN(Time Triggered Controller Area Network), FlexRay가 있으며 지능형 자동차의 x-by-wire 시스템 기술을 위하여 경쟁적으로 개발되었거나 개발되고 있다[5,6].

특히, Kopetz에 의해 개발되고 TTTech사를 통해 상업화된 TTP는 새시 시스템의 표준 프로토콜이 되기 위하여 서로 경쟁하고 있는 TTCAN과 FlexRay에 비하여 앞서 개발되었을 뿐만 아니라, 항공기 산업을 비롯하여 자동차 산업에 일부 적용되고 있다[7].

본 논문에서는 time-triggered 방식의 TTP 프로토콜을 이용하여 분산 실시간 제어 시스템을 설계하는 네트워크 설계 방법을 제시한다. 또한, TTP 기반 모터 제어 시스템을 설계함으로써, 분산 실시간 제어 시스템에서 TTP 프로토콜의 적용 가능성을 검토한다.

본 논문은 서론을 포함하여 5장으로 구성되어 있다. II장에서는 TTP 프로토콜의 특성에 대하여 간단히 설명하고, III장에서는 TTP 프로토콜 설계 방법에 대하여 소개한다. IV장에서는 TTP 프로토콜 설계 방법을 이용하여 모터 제어 시스템을 설계하고, 정상적으로 동작하는 지를 확인한다. 마지막으로, V장에서는 결론과 향후 과제를 제시한다.

II. TTP 개요

TTP는 오스트리아의 빈 공과대학에 의해 1994년에 개발되었으며, TTTech사에 의해 하드웨어 및 소프트웨어가 개발되었다. TTP Group에는 Audi, PAS, Renault, NEG, Delphi, Visteon 등의 기업이 참여하고 있다. TTP는 미국 자동차 공학회(SAE)에서 규정한 실시간 요구 조건(real-time requirements) class C를 만족하고 있으며, 가장 빠르게 상업화되어

일부 차종에 적용되고 있다. 현재 TTP는 자동차뿐만 아니라, 에어버스 A380의 객실 압력 제어 시스템, F-16 전투기의 디지털 엔진 제어기와 스위스의 철도 신호 시스템에 적용되고 있다[8,9].

TTP는 저가의 자동화용으로 개발된 TTP/A와 고장 허용(fault tolerant) 기능을 가지고 있는 자동차용 TTP/C로 구분될 수 있으나, 일반적으로 TTP라고 하면 TTP/C를 의미한다. 또한, TTP는 자동차 분야에서 요구하는 분산 제어 구조를 지원하고, 뛰어난 고장 허용 기능을 지원한다. TTP는 사용하는 물리 계층에 따라 5Mbps에서 25Mbps의 다양한 전송 속도를 지원하고, 중복 노드와 중복 채널을 지원한다. 또한, 메시지 프레임의 오버헤드 사이즈가 작으며, 강인한 에러 검출 및 회복 기능을 가지고 있다.

TTP 노드는 그림 2와 같이 호스트 컴퓨터(host computer), TTP 컨트롤러(TTP controller)와 CNI(Communication Network Interface)로 구성되며, 고장 허용을 위한 중복 네트워크 구조를 지원한다. 그림에서 CNI는 호스트 컴퓨터의 CPU와 TTP 컨트롤러 사이의 정보 교환을 위한 인터페이스를 담당하고 TTP controller에 포함된다. 또한, TTP 컨트롤러는 버스 가디언(bus guardian), MEDL(MESSAGE DESCRIPT LIST)과 프로토콜 프로세서(protocol processor)로 구성된다. 여기서, 버스 가디언은 주어진 통신 구간에만 통신을 할 수 있도록 송수신을 조절하는 기능을 가지고 있으며, MEDL은 사전에 정해진 전송 주기에 따라 주기적으로 전송하는 TDMA에 관한 통신 스케줄을 저장하고 있다.

TTP의 전송 메커니즘은 그림 3과 같은 TDMA 방식을 이용한다. TDMA 전송 방식은 사전에 할당 받은 전송 주기에 따라 각 노드가 전송을 하는 방식이다. 그림에서, 하나의 전송 주기를 TDMA 라운드(round)라고 하며 TDMA 라운드의 배열을 클러스터 사이클(cluster cycle)이라고 한다. TDMA 라운드 길이는 모두 동일하며 각각의 노드 슬롯에서 보내는 프레임의 길이와 내용은 노드의 특성에 따라 다르게 설정할 수 있다. 즉, 각 노드가 전송할 수 있는 구간의 크기는 네트워크 설계 과정에서 자유롭게 설정할 수 있으며, 설정된 이후에 수정이나 추가적인 노드의 참여는 불가능한 구조를 가지고 있다.

그리고, TTP는 모든 메시지가 예상 가능한 범위 내의 지연을 가지며 알 수 없는 네트워크의 지터(jitter)의 발생이 상대적으로 적은 편이다. 또한, 고장을 허용하기 위한 방법으로 중복 노드와 중복 네트워크를 지원한다. 따라서 한 노

드에 고장이 발생하여도 다른 노드가 고장에 영향을 받지 않고 정상적으로 동작을 수행할 수 있다. 뿐만 아니라, 작은 프로토콜 오버헤드로 네트워크 용량의 효율이 뛰어나고 강력한 에러 검출 및 회복 기능을 가지고 있다. 마지막으로, 멤버십 서비스(membership service)를 통해 각 노드가 자신을 제외한 다른 노드들의 상태를 항상 알 수 있는 기능을 제공하기 때문에 분산 시스템에서 노드의 고장과 불일치성을 감지할 수 있다.

III. TTP 네트워크 설계

TTP 네트워크 설계 방법은 크게 두 단계로 구성되며 TTTech사에서 개발한 TTP cluster를 이용하여 구현할 수 있다[10]. 그림 4는 두 단계로 구성된 TTP 네트워크 설계 방법에 대한 구성을 나타내고 있다. 그림에서, TTP 네트워크 설계 방법은 클러스터 단계(cluster level)와 노드 단계(node level)로 구분된다. 클러스터 단계는 시스템을 구성하는 클러스터 구조와 서브시스템(subsystem)의 정의 및 배치를 설계하며 각각의 서브시스템의 인터페이스 절차를 설계하는 단계이다. 노드 단계는 클러스터 단계 설계를 바탕으로 각각 노드의 서브시스템 기능을 정의하는 단계이다. 즉, 클러스터 단계를 통해 전체 시스템의 배치를 결정하고 노드 단계에서 각 노드의 서브시스템 기능을 정의하는 단계를 가지고 설계한다.

1. 클러스터 단계 설계

클러스터 단계는 다수의 서브시스템으로 구성되어 있는

노드를 포함한 TTP 클러스터를 설계하는 과정이다. 클러스터 단계에서는 하나의 클러스터를 구성하는 노드의 수와 각 노드에서 전송되는 메시지들을 정의한다. 시스템 단계는 TTTech에서 개발된 TTP Plan을 이용하여 단계별로 구성하여 설계할 수 있다.

가장 먼저, TTP 네트워크의 TDMA 라운드 길이를 정의한다. 여기에서, TDMA 라운드 길이는 TTP 클러스터의 각 노드에서 전송되는 메시지와 시스템에서 요구되는 전송 지연 특성을 고려하여 설계해야 한다. 이는 시스템에서 요구되는 전송 지연 특성을 고려하여 TDMA 라운드를 설계하여만 신뢰성 있는 네트워크가 구성될 수 있기 때문이다. 또

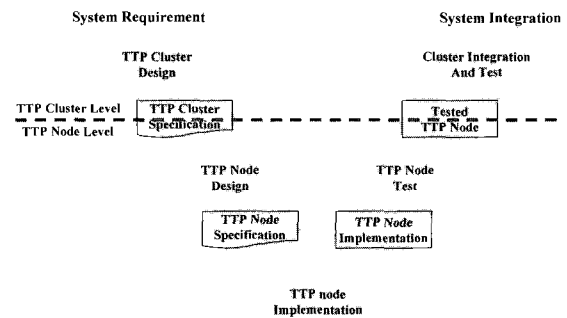


그림 4. V 모델에서 TTP 프로토콜 개발의 2단계 접근 방법.

Fig. 4. Two-level design approach in the V model for TTP protocol.

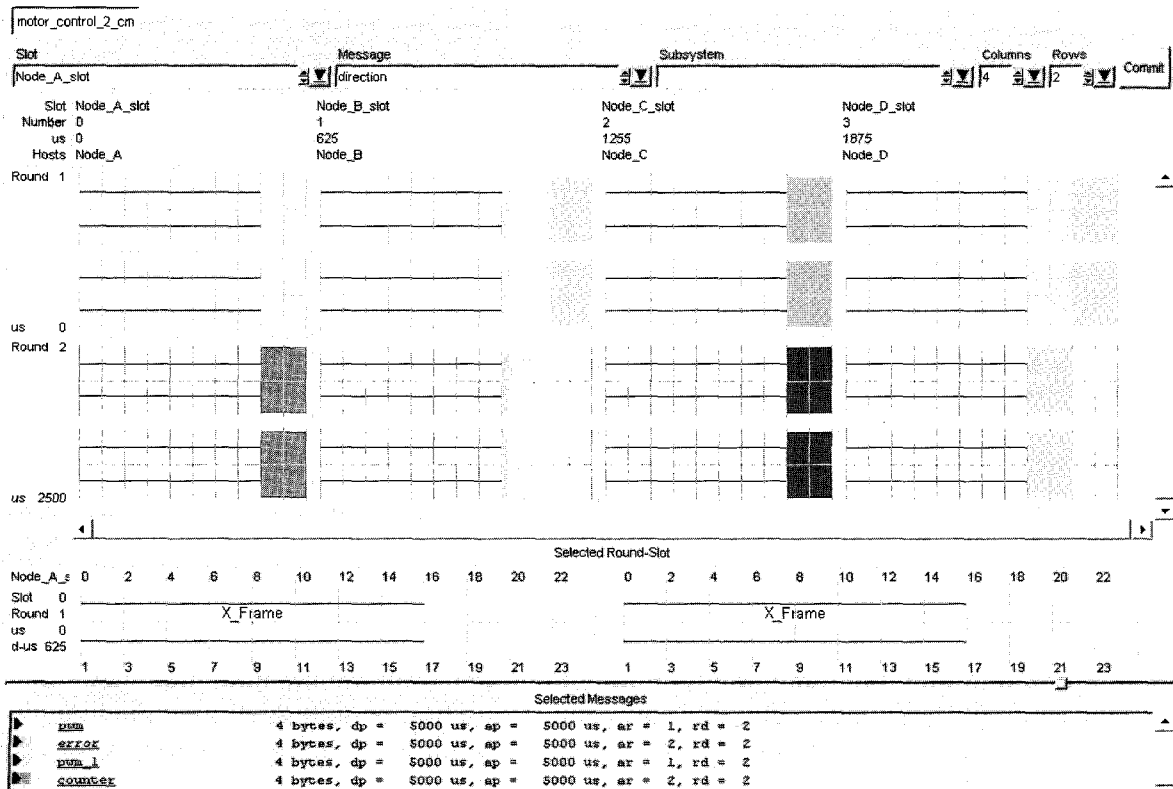


그림 5. TTP Plan의 그래픽 스케줄 에디터의 예.

Fig. 5. Example of graphic schedule editor of TTP Plan.

한, TTP 클러스터를 구성하는 각 노드는 TDMA 라운드에서 슬롯이 할당된 시간에만 메시지 전송을 할 수 있기 때문에 각각의 노드에서 발생하는 메시지를 고려하여 TDMA 라운드를 설계해야 한다.

두 번째 단계에서는 TTP 네트워크 클러스터를 구성하고 있는 각각의 노드를 정의한다. 네트워크 클러스터를 구성하는 각각의 노드 이름을 정의하고, 시리얼 번호를 부여하여 TDMA 라운드에서 메시지를 전송하는 순서를 정의한다. 또한, 각 노드를 구성하고 있는 서브시스템의 기능을 정의한다. 여기서, 서브시스템은 분산 제어 시스템에서 필요한 특정한 기능을 수행하는 기능을 의미한다.

세 번째 단계에서는 각 노드의 메시지를 정의한다. 즉, 정의된 각각의 메시지를 노드의 서브시스템과 연결시키는 작업을 수행하는 단계이다. TTP 클러스터의 각 노드에 메시지를 연결시키면 TTP 네트워크에서 메시지를 정의하는 과정은 완료가 된다.

네 번째 단계에서는 각 단계에서 발생할 수 있는 오류를 검사한다. 오류가 발생하면 오류가 발생된 부분을 다시 설계하고 오류를 검사를 반복하는 과정을 수행한다.

마지막으로, 오류가 없으면 자동 스케줄 생성 방법(auto scheduling generation)을 이용하여 스케줄을 생성한다. 각 단계의 정보를 이용하여 TTP Plan으로 스케줄을 생성하고, 그래픽 스케줄 에디터로 스케줄을 확인한다. 예로, 그림 5는 TDMA 자동 스케줄 생성 방법을 통하여 완성된 그래픽 스케줄 에디터를 나타내고 있다. 그림에서는, 정의된 4개의 노드가 각각 2개의 메시지를 가지고 하나의 네트워크 클러스터를 구성하도록 설계된 클러스터 단계 설계를 나타내고 있다.

2. 노드 단계 설계

노드 단계 설계는 시스템 단계 설계에서 생성한 클러스터 스케줄을 바탕으로 각 노드의 서브시스템을 설계하는 과정이다. 노드 단계는 TTTech사에서 개발한 TTP Build를 이용하여 단계별로 구성하여 설계할 수 있다.

먼저, 미리 설계된 TTP 클러스터의 데이터베이스를 불러와서 노드를 선택한다. 이는 TTP 네트워크에서 각 노드는 네트워크 클러스터와 동기화되어야 정상적으로 동작을 수행할 수 있기 때문이다. 클러스터 단계에서 생성된 클러스터 정보를 이용하여 클러스터에 포함된 각각의 노드의 서브시스템을 정의한다.

두 번째 단계에서는 각 노드의 서브시스템에서 동작을 수행하는 태스크를 생성한다. 서브시스템에서 실행되는 태스크를 생성하고 각각의 태스크에서 송신되는 메시지와 수신되는 메시지를 연결한다.

세 번째 단계에서는 각 노드의 I/O(Input/Output) 메시지를 생성한다. 여기에서, I/O 메시지는 센서 및 액추에이터의 입출력 값을 가지고 있는 메시지를 의미한다. 서브시스템의 태스크를 수행하는 I/O 메시지들을 연결하면 태스크 설계 과정은 완료된다.

마지막으로, 노드 설계에서 에러가 없는지를 검사를 하고, 서브시스템의 태스크 스케줄을 자동 스케줄 생성 방법을 통하여 생성한다. 그리고 TTP Build에 포함되어 있는

TTP OS가 자동으로 스케줄링 정보를 코드로 생성하게 되면 노드 단계 설계를 마무리한다.

IV. TTP 네트워크를 이용한 제어시스템 설계 및 성능 평가

본 논문에서는 time-triggered 방식의 TTP 네트워크를 설계하고 그 성능을 평가하기 위하여 그림 6과 같은 TTP 기반 분산 실시간 제어 시스템을 구성하였다. TTP 프로토콜은 TTTech사에서 개발한 TTP development cluster를 이용하여 클러스터 단계와 노드 단계를 구분하여 개발하였다. 실험 모델에서 TTP 프로토콜 기반 분산 실시간 제어 시스템은 TTP development cluster에서 제공하는 TTP 네트워크를 지원하는 네 개의 파워 노드(power node)로 구성되어 있으며, 각각의 노드는 호스트 CPU로 Freescales사의 MPC555를 사용하고 있다.

TTP 기반 제어 시스템에서 제어 대상이 되는 플랜트로 RC 서보 모터와 DC 서보 모터가 사용되었다. 여기에서, 모터의 방향 전환 제어 성능을 평가하기 위하여 RC 서보 모터를 사용하였으며, 속도 제어 성능을 평가하기 위해 DC 서보 모터를 사용하였다. 또한, 제어 입력은 산업용 조이스틱(joystick)을 사용하였으며, TTP 네트워크에 접속되어 있는 제어 입력 노드(control input node)를 통하여 제어 정보를 송신할 수 있도록 설계하였다. 또한, 중복 제어 노드(redundant control node)는 제어 입력 노드로부터 수신되는 제어 정보와 제어 출력 노드(control output node)로부터 수신되는 현재 상태 정보를 이용하여 제어 출력을 계산한 후 제어 출력 노드로 송신할 수 있도록 하였으며, 고장을 허용할 수 있는 중복 구조로 설계되었다. 마지막으로, 제어 출력 노드는 중복 제어 노드로부터 수신되는 제어 출력에 따라 PWM 출력을 조절하고 현재의 모터 속도 정보를 중복 제어 노드에 송신하도록 설계하였다.

플랜트로 사용된 모터의 전달 함수는 스텝 입력에 대한 응답을 이용하여 실험적으로 결정하였고 (1)과 같다[11].

$$G(s) = \frac{1}{0.1428s^2 + 0.00594s} \tag{1}$$

특히, 본 논문에서는 DC 서보 모터의 속도 제어 성능을

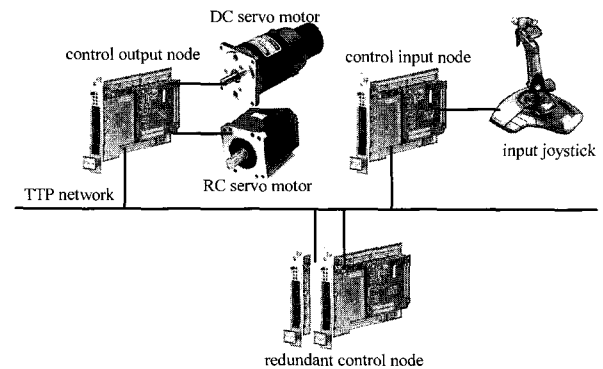


그림 6. TTP 기반 분산 실시간 제어 시스템의 테스트베드.
Fig. 6. Testbed of TTP based distributed real-time control system.

평가하기 위하여, DC 서보 모터 제어 시스템의 설계 사양을 최대 오버슈트는 20% 이하, 2% 정착시간은 100ms 이하, 감쇠 비는 0.5 이하로 설정하고 근궤적법(root locus)을 이용하여 PI 제어기를 (2)와 같다[12-14].

$$G_{PD}(s) = K_p + \frac{K_i}{s} = \frac{0.01s + 0.021}{s} \quad (2)$$

1. TTP 기반 제어 시스템 설계

TTP 기반 제어 시스템에서 속도 제어 시스템은 TTP Matlink를 이용하여 설계하였다. TTP Matlink는 TTTech사의 TTP Cluster에서 제공하는 소프트웨어로서 MATLAB의 Simulink, Real-Time Workshop Embedded Coder, Stateflow와 통합하여 rapid prototyping platform으로 TTP 응용 시스템을 개발할 수 있는 환경을 제공해 준다.

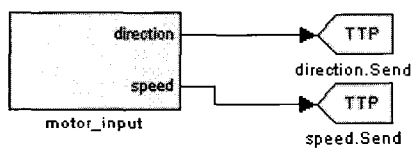
TTP 기반의 모터 속도 제어 시스템은 2500µs의 TDMA 라운드를 가지고 전송 속도는 500Kbps로 설정하였다.

1.1 제어 입력 노드 설계

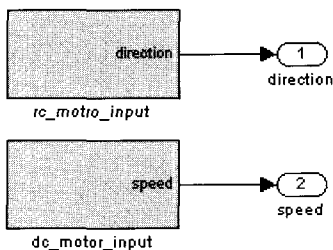
제어 입력 노드는 하나의 서브시스템(motor_input)과 두 개의 태스크(rc_motor_input, dc_motor_input)로 구성되어 있다. 그림 7은 모터 제어 입력 노드의 Matlink 함수 블록을 나타내고 있다. 정의된 두 개의 태스크는 조이스틱의 아날로그 변위 값을 ADC 채널을 이용하여 디지털 값으로 연산하여 그 입력값을 TTP 네트워크를 통해 중복 제어 노드로 전송하는 기능으로 정의되었다. 여기서, rc_motor_input 태스크는 조이스틱의 좌우 정보를 통해 RC 서보 모터의 회전 방향을 제어하기 위한 방향 메시지(direction.send)를 송신하고, dc_motor_input 태스크는 조이스틱의 전후 정보를 통해 DC 서보 모터의 회전 속도를 제어하기 위한 속도 메시지(speed.send)를 송신한다.

1.2 중복 제어 노드 설계

중복 제어 노드는 하나의 서브시스템(motor_control_unit)에 두 개의 태스크(rc_motor, dc_motor)로 구성되어 있다. 그림 8은 중복 제어 노드의 Matlink 함수 블록을 나타내고 있



(a) Subsystem function block

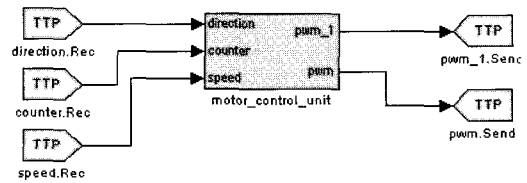


(b) Task function block

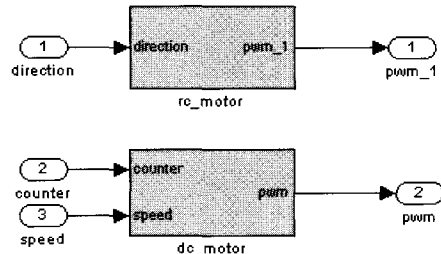
그림 7. 제어 입력 노드에서 함수 블록.

Fig. 7. Function block in control input node.

다. 정의된 두 개의 태스크는 제어 입력 노드로부터 수신된 입력 정보와 제어 출력 노드로부터 수신된 모터의 상태 정보를 바탕으로 제어 출력 값을 TTP 네트워크를 통해 제어 출력 노드에 전송하는 기능으로 정의되었다. 여기에서, rc_motor 태스크는 제어 입력 노드로부터 받은 방향 메시지 정보를 통해 방향 제어 메시지(pwm_1.send)를 제어 출력 노드에 송신하고 dc_motor 태스크는 제어 입력 노드로부터 받은 속도 메시지 정보와 제어 출력 노드로부터 받은 카운터 메시지 정보를 이용하여 PWM 출력 값을 연산한 후 속도 제어 메시지(pwm.send)를 제어 출력 노드에 송신한다.



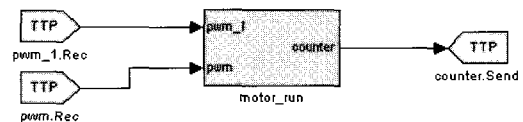
(a) Subsystem function block



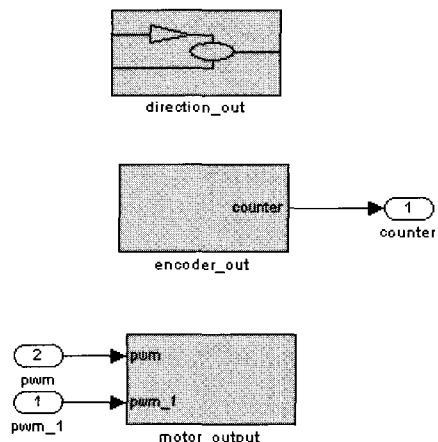
(b) Task function block

그림 8. 중복 제어 노드의 함수 블록.

Fig. 8. Function block in redundant control node.



(a) Subsystem function block



(b) Task function block

그림 9. 제어 출력 노드의 함수 블록.

Fig. 9. Function block in control output node.

또한, 중복 제어 노드는 동일한 기능을 수행하는 노드를 다수 설계하는 중복 구조를 가지고 있어 하나의 노드에서 고장이 발생하여도 정상적으로 동작될 수 있도록 설계하였다.

1.3 제어 출력 노드 설계

제어 출력 노드는 하나의 서브시스템(motor_run)과 세 개의 태스크(direction_out, counter_out, motor_output)로 구성하였다. 그림 9는 제어 출력 노드의 Matlink 함수 블록을 나타내고 있다. 정의된 태스크는 제어 모듈로부터 수신된 제어 정보에 따라 플랜트 출력을 발생시키는 기능과 DC 서보 모터의 엔코더 값을 연산하는 기능으로 정의하였다. 여기에서, direction_out 태스크는 DC 서보 모터의 회전 방향을 결정하며 실험에서는 편의상 한 방향으로 회전하도록 설정하였다. counter_out 태스크는 DC 서보 모터의 엔코더 출력 값을 나타내고 10ms 주기로 수신 받은 엔코더 카운터 수를 연산하여 카운터 메시지(counter.Send)를 중복 제어 노드에 송신한다. motor_output 태스크는 중복 제어 노드로부터 받은 pwm 값에 따라 모터를 제어하는 PWM 값을 DC 모터에 제공하여 모터를 제어하는 기능을 가진다.

2. TTP 네트워크를 이용한 제어시스템 성능 평가

TTP 기반 모터 제어 시스템의 성능을 평가하기 위하여, 기준 속도 입력에 따른 DC 서보 모터의 응답 성능 실험을 수행하였다. 기준 속도 입력은 산업용 조이스틱을 사용하였으며, 조이스틱의 좌우 입력은 RC 서보 모터의 방향을 제어하고 상하 입력은 DC 서보 모터의 속도를 제어하도록 설계하였다. 즉, 조이스틱을 상하로 조절하면 DC 서보 모터의 속도가 증감되게 되고 좌우로 조절하면 AC 서보 모터의 방향이 바뀌게 되도록 구성하였다.

그림 10은 TTTech사에서 제공하는 TTP cluster 모니터링 프로그램인 TTP view를 이용하여 전송되는 메시지를 모니터링 한 결과를 나타내고 있다. 그림에서, 모터의 카운터 값을 수신한 후, 이를 rpm으로 바꾸는 모듈을 추가함으로써 모터의 속도를 쉽게 확인할 수 있도록 하였으며, 기준 속도 입력 메시지가 전송되면 기준 입력과 같은 카운터 정보를 확인할 수 있도록 하였다. 그리고 멤버십 기능은 네트워크에서 동작하는 노드의 상태를 모니터링하기 위하여 사용하는 것으로서, 노드 4개 중 하나가 검정색으로 표시된 것은 노드 한 개가 동작하지 않는다는 것을 의미한다. 그림에서, 검정색으로 표시된 노드는 중복으로 설계된 제어 노드를 나타내고 있으며 중복 노드에 고장이 발생하여도 정상

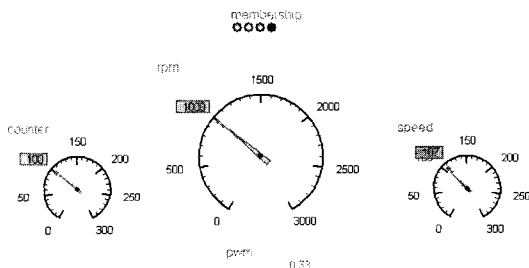


그림 10. TTP View를 이용한 제어 성능의 예.
Fig. 10. Example of control performance using TTP View.

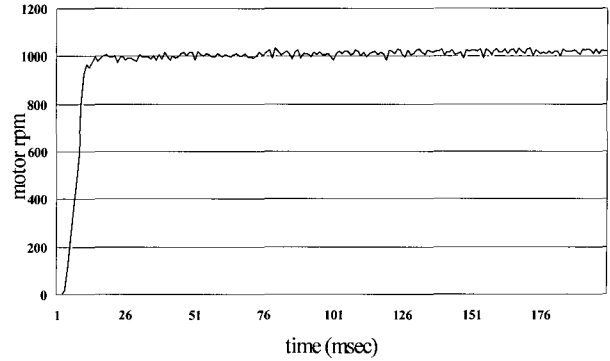


그림 11. TTP 기반 모터 제어 시스템의 스텝 응답.
Fig. 11. Step response of TTP based motor control system.

적으로 동작하고 있음을 알 수 있다. 즉, 제어 노드를 중복으로 설계하였기 때문에 하나의 노드에 고장이 발생하여도 시스템은 정상적으로 동작할 수 있음을 확인하였다.

그림 11은 TTP 기반 모터 제어 시스템의 스텝 응답을 나타내고 있다. 여기에서, 모터 속도 제어를 위한 기준 속도 입력으로는 1,000rpm으로 설정하였으며, 10ms 주기로 제어를 수행하였다. 그림에서 TTP 기반 모터 제어 시스템은 안정적으로 동작함을 알 수 있었다.

V. 결론

본 논문에서는 네트워크 기반 실시간 분산 제어 시스템을 위하여 time triggered 프로토콜의 한 종류인 TTP 네트워크 설계 방법을 제시하였다. 또한, TTP 네트워크 설계 방법을 이용하여 모터 제어 시스템을 설계하였으며, 모터 제어 시스템이 정상적으로 동작됨을 확인하였다.

이상과 같은 실험 결과로부터, 본 논문에서는 분산 실시간 제어 시스템을 위하여, time-triggered 방식인 TTP 네트워크가 상당히 유용한 프로토콜임을 확인하였다. 특히, 분산 실시간 제어 시스템의 구현 과정은 매우 많은 시간이 요구되는 작업이지만, TTP 네트워크 설계 방법을 이용하는 경우 효과적으로 분산 실시간 제어 시스템의 구현이 가능함을 확인하였다. 뿐만 아니라, 자동차와 같이 고장에 민감한 시스템에 중복 구조 설계 방법을 도입하는 경우, 고장에 대한 허용 성능이 우수함을 알 수 있었다.

그러나 본 논문에서는 실험실 수준에서 4개의 TTP 노드를 이용하여 제어 시스템을 구성하였다. 따라서, 향후 보다 많은 노드를 이용하여 분산 실시간 제어 시스템을 구성하여 성능을 평가하는 연구가 필요하다. 또한, 분산 통합 제어 시스템에서 생성되는 메시지를 이용하여 TTP 네트워크의 성능을 평가할 필요가 있다. 뿐만 아니라, 최근 들어 가장 많은 관심을 받고 있는 FlexRay 기반 분산 실시간 제어 시스템 설계에 대한 연구가 필요하며, TTP와 FlexRay의 성능에 대한 비교가 필요하다.

참고문헌

[1] 연제명, 선우명호, 이우택, "LIN 프로토콜 시간 모델링 및 메시지 응답 시간 해석에 관한 연구," 한국자동차

공학회 논문집 제 13 권, 제 6 호, pp. 48-55, 2005.

[2] 임창휘, 김만호, 이경창, 이석, "Steer-By-Wire system을 위한 피드백 제어 시스템 구축," 2005년 춘계 학술대회, pp. 501-506, 2005.

[3] 이장명, 최호식, 최효식, "자동차 네트워크 시스템의 효율적 관리를 위한 CAN의 동적 선행대기 열 기법," 제어자동화 시스템공학 논문집, 제 12 권, 제 6 호, pp. 614-620, 2006.

[4] 주영훈, "스마트 스페이스 구축을 위한 지능형 디지털 분산 제어 시스템 개발," 제어 · 자동화 · 시스템공학 논문집, 제 12 권, 제 1 호, pp. 54-59, 2006.

[5] E. Dilger, T. Fuhrer, B. Muller, and S. Poledna, "The X-By-Wire concept : time-triggered information exchange and fail silence support by new system services," *SAE Technical Paper*, 1998.

[6] H. Kopetz, "A comparison of CAN and TTP," *Annual Reviews in Control*, vol. 24, no. 1, pp. 177-188, 2000.

[7] Razafindralambo T. and Auge-Blum I. "Formal evaluation and comparison of real time embedded automotive networks," *Industrial Technology*, vol. 2, pp. 898-903, 2004.

[8] H. Kopetz and G. Grunsteidl, "TTP-a protocol for fault-tolerant real-time systems," *Computer*, vol. 27, Issue 1, pp. 14-23, 1994.

[9] *Time-Triggered Protocol TTP/C High-Level Specification Document Protocol Version 1.1*, TTTech, 2003.

[10] Time-Triggered Architecture and FlexRay: <http://www.ttautomotive.com>.

[11] 김만호, 하경남, 이경창, 홍금식, 이석, "CAN 네트워크를 이용한 자동차 크레인 시스템의 구현," 한국항해항만학회지, 제 29 권, 제 6 호, pp. 555-560, 2005.

[12] 이경창, 김태준, 이석, "실시간 산업용 통신을 위한 Switched Ethernet의 성능 평가," 제어 · 자동화 · 시스템공학 논문집, 제 9 권, 제 1 호, pp. 90-98, 2003.

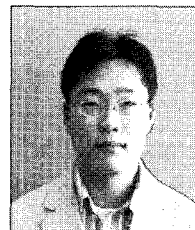
[13] 이경창, 이석, "Profibus-DP를 이용한 네트워크 기반 제어 시스템의 원격 피지 제어," 제어 · 자동화 · 시스템공학 논문집, 제 8 권, 제 4 호, pp. 281-287, 2002.

[14] M. Araki, and H. Taguchi, "Two degree of freedom PID controllers," *International Journal of Control, Automation, and Systems*, vol. 1 no. 4, pp. 90-98, 2003.



김 만 호

1977년 4월 16일생. 2001년 동아대학교 기계공학부 졸업. 2003년~2006년 부산대학교 지능기계공학과 석 · 박사 통합과정. 2006년 기계공학연구정보센터 전임연구원. 2006년~현재 대구경북과학기술연구원 지능형 자동차 연구팀 연구원. 관심분야는 차량용 네트워크 시스템, 임베디드 시스템, 고장 허용을 위한 중복 시스템, 운전자 행동 분석 및 운전 부하 분석.



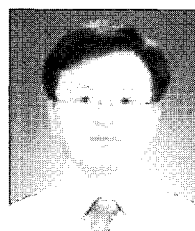
손 병 점

2006년 부산대학교 기계공학부 졸업. 2006년~현재 부산대학교 지능기계공학과 석 · 박사 통합과정 재학중. 관심분야는 차량용 네트워크 및 임베디드 시스템.



이 석

1984년 서울대학교 기계공학과 졸업. 1985년 펜실바니아 주립대학교 석사. 1990년 동 대학원 박사. 1990년~1993년 신시내티 대학교 기계공학과 조교수. 1993년~현재 부산대학교 기계공학부 교수. 관심분야는 필드버스, 차량용 네트워크, 홈 네트워크.



이 경 창

1971년 5월 1일 생. 1996년 부산대학교 생산기계공학과 졸업. 1998년 동 대학원 석사. 2003년 동 대학원 박사. 1998년~2003년 기계공학연구정보센터 전임연구원. 2003년~2005년 울산대학교 네트워크기반 자동차 연구센터 전임연구원. 2005년~현재 부경대학교 제어자동화공학과 조교수. 관심분야는 필드버스, 산업용 이더넷, 차량용 네트워크, 홈 네트워크.