

## 선박용 대형 디젤기관의 자동제어 기술 구현



정경열

(KIMM 환경설비연구부)

- '78 - '81 한국해양대학교 기관학(학사)
- '81 - '83 한국해양대학교 제어공학(석사)
- '83 - '86 승선근무(기관사)
- '94 - '00 한국해양대학교 제어공학(박사)
- '87 - 현재 한국기계연구원 선임연구원



오상훈

(KIMM 환경설비연구부)

- '92 - '98 대전산업대 기계설계공학과(학사)
- '98 - '00 충남대학교 기계공학과(석사)
- '99 - 현재 한국기계연구원 연구생

### 1. 서론

오늘날의 대부분의 자동화 시스템은 마이크로 프로세서 기술의 발달에 힘입어 시스템 구축의 유연성 및 사용자 편리성을 가진 고성능의 마이크로프로세서 기반 시스템으로 개발되고 있으며, 이로 인한 자동화 시스템의 개발환경 개선과 개발시간 단축은 실시간 제어하는 동시에 많은 기능을 수행하는 것이 수월하도록 하여 준다.

선박 분야에서는 가혹한 해상 환경에서의 근무여건 개선 및 선원인력 감소에 대비하기 위한 목적으로 자동화 시스템이 개발되고 있다. 특히, 운항 중인 선박의 자동화 시스템에 고장이 발생하면 승무원 자체의 기술력만으로 대처해야 하기 때문에 선박 디젤엔진 원격제어 시스템과 같이 선박운항에 직접적으로 관련있는 시스템의 고장은 대부분 인명과 선박자체의 손실로 직결된다. 그러므로 선박 자동화 시스템은 고성능·고신뢰성이 육상의 시스템보다 더욱 절실히 요구되고 있으며, 조선 선진국에서는 선박자동화 시스템을 마이크로 프로세서 기반 시스템으로 개발하여 이러한 요구에 대응해 나아가고 있다.

선박 디젤엔진 원격제어 시스템이란 기존의 제품화된 시스템을 두고 언급할 때 선박 디젤엔진을 기계실에서 멀리 떨어진 기관제어실(control room)이나, 브릿지(bridge)로부터 조작하는 시스템을 말한다. 디젤엔진 원격제어 시스템이 구성되는 것도 디젤엔진의 기종이나 선박의 대소에 따라서 약간의 차이는 있으나 디젤엔진의 기본적인 제어 즉, 디젤엔진의 시동 및 정지, 전·후진, 증·감속 등의 운전 조작을 원격

지로부터 할 수 있는 시스템을 말하며, 이외에도 디젤엔진의 이상운전시 자동감속 및 자동정지, 위험진동 대역의 자동회피 등 기타 안전장치가 부착되는 시스템으로 발전해 가고 있다.<sup>[1]~[3]</sup>

노르웨이, 일본, 덴마크 등에서는 이러한 선박 디젤엔진 원격제어 시스템을 마이크로프로세서 기반 시스템으로 개발하여 상품화하고 있으나<sup>[4]</sup>,<sup>[5]</sup> 각 나라의 시스템들은 회사별로 독특한 방식에 의하여 독점 설계·생산되어 있으며, 보안이 철저하게 이루어져 있어서 서로 호환성이 없고, 당연히 설계의 정형화된 형식도 없다. 또한 제어 알고리즘도 각기 달라서 통일된 규격이 없으며 각각의 알고리즘을 비교하여 최적의 기법을 찾는 것은 거의 불가능하다.

실제로 상품화되어 있는 선박 디젤엔진 원격제어 시스템의 동작기능을 살펴보면 NABCO사의 제품은 소형선박을 대상으로 하여 기관제어실에서만 조작 가능한 시스템으로 통신기능이 없는 단순구조로 이루어져 있으며, NORCON사나 LYNGSO MARINE<sup>[6]</sup>사의 경우에는 대형선박을 대상으로 하여 여러 곳에서 엔진 조작이 가능하도록 하거나 원격제어 기능과 엔진 안전 감시 기능을 분리하여 서로 다른 하드웨어 유니트에서 동작되도록 하는 분산시스템 형태로 구축하고 있다. 그러나 이들 제품에서 경제성, 신뢰성 및 사용자 편리성 등을 모두 고려한다면 시스템의 하드웨어를 보다 더 간결하게 하고 소프트웨어 기능을 강화시키되 실시간 성능이 떨어지지 않도록 하는 알고리즘화 기법이 필요하다.

본 연구에서는 경제성, 신뢰성 및 사용자 편리성을 모두 도모하기 위한 마이크로프로세서 기반형 선박 디젤엔진 원격제어 시스템을 분산 형태로 구축하는 방법을 제안하였다. 이를 위해 선박 디젤엔진 원격제어 시스템을 세 개의 유니트 즉, Remote Control System Unit(RCSU), Bridge Control System Unit(BCSU), Safety System Unit(SSU)로 분산하여 구축하고, 또한 각 유니트의 하드웨어는 경제성과 실행속도를 고려하여

Intel<sup>[7],[8]</sup> 계열의 8088CPU<sup>[9]~[12]</sup>를 이용하여 설계 제작하였다. 특히 소프트웨어를 개발함에 있어서 소프트웨어의 재사용, 하드웨어의 간결성, 실시간 성능 및 고 기능화를 목표로 하여 다음과 같은 방법을 도입하였다.

첫째, 태스크들은 프로그램 길이가 길기 때문에 단순하게 속도면을 중시한 어셈블리 언어로만 개발하는 것은 개발시간이 많이 소요될 뿐만 아니라 디버그 및 수정 등이 극히 곤란하다. 따라서 소프트웨어의 구조가 명확하고 시스템 프로그램에까지 접근가능하며 범용성이 있는 C 언어를 이용하였다.<sup>[13]~[15]</sup>

둘째, 이 태스크들은 상호 배타적인 실행, 시퀀스 실행, 비주기적인 실행 등 각기 실행 특성이 다르다. 따라서 이것들을 실시간내에 실행시키기 위해서는 각기 다른 실시간 운영기법을 포함시켜 프로그래밍하거나 아니면 실시간 운영기법을 별도의 시스템으로 프로그래밍하여 이 실시간 운영시스템이 상기 태스크들을 통합관리하면서 실행시키도록 프로그래밍 하여야 한다. 본 연구에서는 후자와 같은 실시간 멀티태스크 기법을 채택하여 이를 알고리즘화 하였다.

셋째, 통신방식을 구현함에 있어서 하드웨어를 간결하게 하기 위해 수신은 인터럽트 방식을 취하고 송신은 폴링방식을 취하도록 하는 통신프로토콜과 통신알고리즘을 제안하였다.

넷째, 시스템이 큰 메모리를 차지하는 것에 대비하여 메모리를 절약하는 방법으로 수신된 데이터를 처리하기 전에 임시로 저장해 놓는 버퍼영역을 가급적 적게 차지하도록 하는 가상 링버퍼를 사용하는 방법을 제안하고 이때에 발생할 수 있는 비정상적인 상태를 수정하기 위한 방법도 제시하였다.

다섯째, 디젤엔진의 원격제어 태스크들으로써 엔진진동 및 엔진정지, 디젤엔진의 속도명령제어 특히 임계 회전수 영역이 존재하는 경우의 속도명령제어, 비상후진(crash astern), 비상전진(emergency ahead), 제어위치 절환, 경보를 다단

계로 구별하여 표시하기 위한 플리커링(flickering) 알고리즘들을 제시하였다. 이 중 비상전진 알고리즘은 메이커와 선박기관사들의 요청에 의해 추가된 기능으로써, 지금까지 개발된 제품들에는 없는 기능이며, 이 기능은 필요에 따라 선택 가능하도록 하였다.

마지막으로 시스템을 구축한 후, 구축된 시스템의 동작확인을 위해 시뮬레이션을 실시하고, 쌍용중공업에서 제작한 선박용 디젤엔진<sup>[16],[17]</sup>(B&W 7L35MC)에 장착하여 시운전 테스트해 봄으로써 그 유효성을 입증하였다.

## 2. 시스템의 하드웨어 구성과 구동드라이버

선박 디젤엔진 원격제어 시스템은 기능상 엔진의 운전제어 기능과 안전관리 기능을 포함하고 있어야 하고, 브릿지와 기관제어실에 분산되어 설치될 수 있어야 하며, 신뢰성과 보수성을 높이는 방안도 고려해야 한다. 이를 충족시킬 수 있는 가장 좋은 방법은 원격제어 시스템을 서브유니트의 형태로 구성하는 것이며 3개의 서브유니트, 즉 BCSU, RCSU, SSU로 구별하여 구성함으로써 요구되는 조건을 충분히 만족시킬 수 있다.

따라서 선박 디젤엔진 원격제어 시스템을 BCSU, RCSU, SSU 등 3개의 유니트로 구성하고, 원격제어 시스템을 중심으로 이들을 연결하기 위한 통신라인, 부가적 유니트인 브릿지용 ETSU(Engine Telegraph System Unit), 기관제어실용 ETSU, 엔진부착용 ETSU, RCSU로부터 E/P convertor를 통해 연결되어 있는 DGSU(Digital Governor System Unit)로 구성하였다. 그리고 각 유니트의 하드웨어는 유지보수의 편리성을 위하여 Microprocessor board, I/O board, Signal processing board, RPM counter board, LED & 7-segment display board로 분리하여 구성하였으며, 이 중 Microprocessor board는 8088 CPU를 중심으로 RAM, ROM, 어드레

스 디코딩부, I/O 인터페이스와 인터럽트 제어부, 시리얼 통신부 등으로 구성하였다. I/O 인터페이스부는 8255소자와 7218소자로, 인터럽트 제어부는 8259소자와 8253소자로, 시리얼 통신부는 두 개의 8251소자로 구성하였다. 이때 각 소자를 구동하기 위한 구동드라이버 프로그램은 C언어를 이용하고, 인터럽트 처리부분만은 어셈블리어언어를 이용하였다.

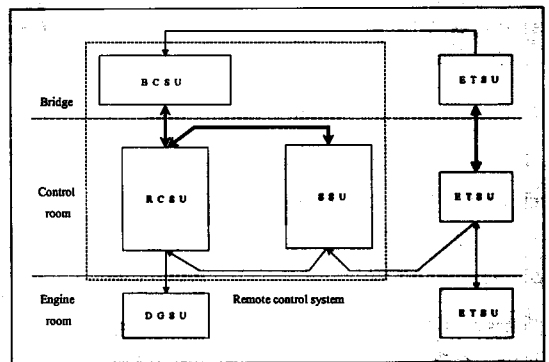


그림 1. 전체 시스템 구성

선박 디젤엔진 원격제어 시스템의 구동은 서브 유니트의 각 소자의 내부 제어 레지스터에 제어 워드를 저장함으로써 사용하고자 하는 목적에 따라 초기화하고, 기능을 수행하도록 하였으며, 또한 시스템이 다운된 경우 빠른 복구와 디폴트값, 상한값과 하한값 등을 보존해 두기 위하여 시스템에 EEPROM을 장착하여 어느 파라미터 값이 어느 번지에 저장되어 있는지 알 수 있도록 C 언어의 구조체로 표현함으로써 파라미터를 정형화하고, 구조체의 부분참조가 가능한 함수를 작성하여 액세스 속도를 향상시켰다.

## 3. 실시간 멀티태스킹

선박 디젤엔진 원격제어 시스템과 같이 대규모인 제어시스템은 각각의 실행 특성이 다르기 때문에 실시간내에 실행시키기 위해서 실시간 운영기법을 별도의 시스템으로 프로그래밍하여 이 실시간 운영시스템이 태스크들을 통합관리하면

서 실행시키도록 하여야 한다. 그리고 대규모의 실시간 운영시스템에서는 제어기능을 순차적으로 실행해 나가는 절차형 방식보다는 대상의 특성에 따라 여러 개의 태스크로 분할하여 필요한 기능만을 실행할 수 있는 실시간 멀티태스크 기능을 가진 효율적인 시스템의 구성이 요구된다. 또한 본 연구에서와 같이 선박 디젤엔진 원격제어 시스템을 Intel 8088CPU만으로 구성할 경우에는 인터럽트처리, 통신알고리즘, 제어알고리즘 등은 필수적으로 구성해야 하고, 필수 불가결한 최소한의 기능만을 부여하여 메모리를 적게 차지하도록 해야 하기 때문에 효율적인 프로그램을 구성하지 않으면 실시간 제어를 기대하기가 매우 어려워진다.

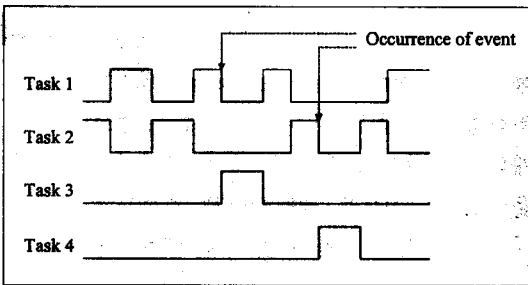


그림 2. 이벤트드리븐 방식과 타임슬라이스 방식의 태스크 실행 타임차트

따라서 본 연구에서는 멀티태스크 기능을 가지도록 타임슬라이스 방식과 이벤트드리븐 방식<sup>[18]~[22]</sup>을 결합할 수 있는 함수들을 정의하여 이용함으로써 RCSU, BCSU 및 SSU시스템을 쉽게 구현할 수 있도록 하였다. 즉 Tick timer 및 RPM 측정과 같은 정교한 실행을 요구하는 모듈들은 타임슬라이스 방식을 이용하여 실행시키도록 하였으며, 기타 시동, 정지, 비상후진, 속도제어, 통신, 디스플레이 및 파라미터 변환 등과 같은 모듈들은 그 모듈이 정지할 때까지 계속적으로 실행시켜야 하기 때문에, 이러한 모듈에는 이벤트드리븐 방식을 이용하여 비선점으로 처리되게 함으로써 동작의 신뢰성과 태스크의 전환시간을 최소화하도록 하였다.

반면에 제어의 목적에 사용되는 실시간 멀티태스크 시스템에서 각 태스크가 이벤트드리븐 방식을 이용하여 실행권을 획득하려면, 실행중이던 태스크를 일시적으로 정지시킨 후 다른 태스크에 실행권을 넘겨주어야 되므로 태스크 전환시에 발생하는 오버헤드시간이 포함되어 전체시스템의 실행시간을 길게 만드는 문제가 발생한다. 이러한 문제는 시스템의 하드웨어 속도 및 규모 등 여러 가지 복잡한 환경과 관련되지만, 빈번한 이벤트 발생에 의한 태스크의 잦은 전환으로 절대적인 오버헤드시간이 수반되어 실시간 제어 성능을 떨어뜨리는 요인이 된다. 또한 태스크내에 시간지연 요소를 포함하고 있는 경우에는 하나의 태스크에서 많은 시간을 소비하게 되므로 역시 실시간 성능이 떨어지게 된다.

이러한 이벤트드리븐 방식의 단점을 보완하기 위해 전체시스템에서 독립적으로 동작해도 무방한 기능들을 하나의 태스크로 만들어 이 태스크를 순차적으로 실행시키는 라운드로빈 방식을 도입하였다. 이 때 하나의 태스크에 대한 실행시간이 너무 길면 다른 태스크의 실시간 실행에 문제가 발생하므로 이에 대한 보완책을 마련해야 한다. 실행시간이 길게 되는 이유로는 태스크의 전체 실행시간이 긴 경우와 시간지연을 포함하는 경우가 있다. 이 중 태스크의 전체 실행시간이 긴 경우에 대한 보완책으로는 각 태스크를 다시 서브태스크로 나누어 이것들을 라운드로빈 방식으로 실행권을 이양받았을 때 한번에 하나의 서브태스크만을 실행시켜주는 방식을 도입하였으며, 시간지연을 포함하는 경우 하나의 태스크에 너무 많은 실행시간이 할당되어 다른 태스크의 실행에 지장을 초래하게 되므로, 이에 대한 보완책으로는 시간지연에 대한 계산을 인터럽트에 의한 타이머방식으로 처리해 주는 특권 타이머태스크를 만들어 이 태스크에서 지연시간을 계산하도록 하고 지연 시간이 완료되면 다시 각 태스크에 알려주도록 하는 방법을 도입하였다.

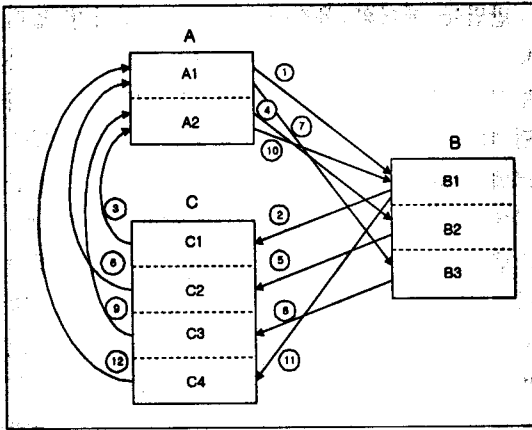


그림 3. 라운드로빈 방식의 태스크 실행순서

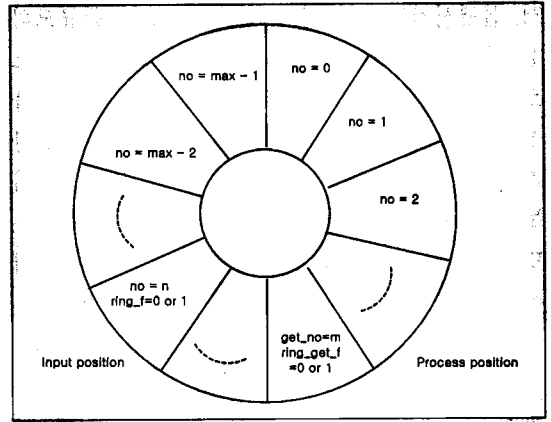


그림 5. 수신데이터를 위한 링버퍼

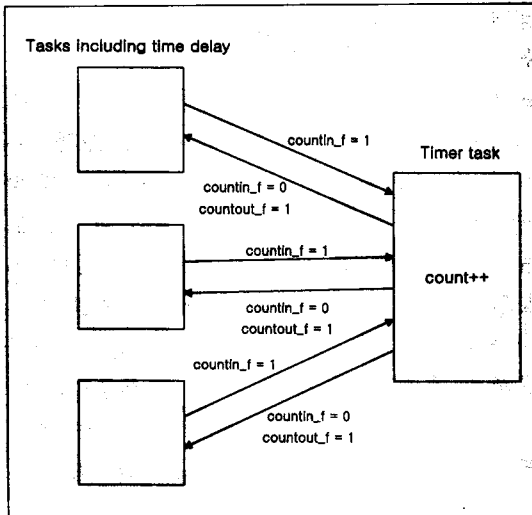


그림 4. 시간지연을 포함하는 태스크와 타이머 태스크 사이의 관계

위와 같이 기능별 태스크 형태로 구현함으로써 디젤엔진 원격제어 시스템의 수정과 프로그래밍이 간결하게 되었으며, 특히 타이머를 하나만 이용하기 위해 타이머 태스크를 별도로 두어 시간지연 계산을 요청하는 태스크들과 플래그 변수만을 이용하여 처리하는 기법을 도입함으로써 하드웨어를 간결하게 할 수 있었다.

#### 4. 시리얼 통신

본 연구에서 구현한 대형선박을 대상으로 한

디젤엔진 원격제어 시스템의 통신방식의 경우에는 통신기능이 RCSU와 BCSU, RCSU와 SSU 사이에서 요구되며, 이들간에는 일대일 통신이 가능하기 때문에 버스 방식보다는 RS-422<sup>[23],[24]</sup>를 이용한 점대점(point to point) 방식의 시리얼 통신방식을 채택하였다. 또한 주 구성요소인 세개의 서브유닛 사이에 수신은 인터럽트 방식을 취하고 송신은 폴링방식을 취하도록 함으로써, 인터럽트 콘트롤러 및 타이머 콘트롤러를 각 1개씩만 이용하고, 이를 위한 통신프로토콜과 통신알고리즘을 제안함으로써 하드웨어를 간결하게 할 뿐만아니라 효율적인 통신이 이루어지도록 구현하였다. 이러한 방식을 도입하는 경우 다른 유닛에서의 송신 타이밍과 유닛 자신의 수신 타이밍을 일치시키기가 대단히 어려우므로 송신을 완료한 후에는 즉시 수신대기상태를 유지하도록 해야 하며, 이 때문에 수신버퍼에는 많은 무효데이터가 저장되므로 별도의 처리루틴에서 버퍼내의 내용을 검사하여 필요한 데이터만을 수신하도록 해야 할 필요가 있다. 더불어 원격제어 알고리즘이 대단히 많은 메모리를 차지할 경우를 대비하여 메모리를 절약하기 위해 수신된 데이터를 처리하기 전에 임시로 저장해 놓는 영역을 가급적 적게 차지하도록 가상 링버퍼를 사용하는 방법을 제안하였고 이때에 발생할 수 있는 비정상적인 상태를 수정하기 위한 방법

도 제시하였다.

에러 검출 기법으로는 통신패킷의 형식에 맞추어 0xCC, 0xDD, 0xEE의 플래그값이 예상되는 수신 위치의 값을 조사하여 이와 동일하면 통신 데이터에 에러가 없다고 판단하고 그렇지 않을 경우에는 해당 패킷에 에러가 있다고 간주하여 그 패킷을 포기하는 방법을 도입하였다.

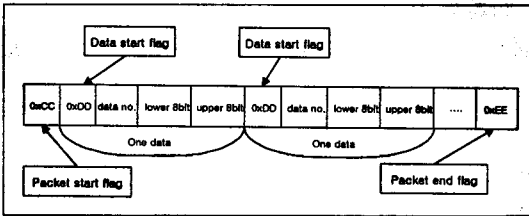


그림 6. 통신패킷 형식

송수신 알고리즘은 RCSU측에서 먼저 BCSU측에 송신을 시작하고 BCSU측에서는 이 데이터를 인터럽트방식으로 수신하여 링버퍼에 저장함과 동시에 멀티태스킹 루틴에서 이 버퍼의 내용을 처리하도록 하였으며, BCSU에서는 수신이 완료되면 RCSU측에 송신을 시작하고, RCSU에서는 BCSU와 같은 방법으로 수신을 시작한다. 이렇게 해서 RCSU와 BCSU의 송수신이 완료되면, 이번에는 RCSU와 SSU간에 송수신을 수행하도록 하였다.

또한 통신을 수행할 때 어느 한쪽이 고장 등의 원인으로 송수신이 불가능한 경우 다른 쪽에서는 무한정으로 대기할 수 없기 때문에 통신실패 알고리즘에 의해 상대 시스템의 통신 불능상태를 확인하도록 하였다.

## 5. 시스템 구축

디젤엔진 원격제어 시스템에서 각 서브유니트의 기능을 살펴보면 BCSU에서는 RPM 커맨드, 제어위치선택, 각종 스위치조작들이 가능하고 이 값들을 RCSU에 전송해 주며, 또한 디젤엔진 상태값<sup>[25]</sup> 및 안전관련 값들을 전송받아 이것들을 패널에 디스플레이 해주고 있다. RCSU에서는

디젤엔진의 원격제어기능과 함께, RPM 검출, 제어위치, 디젤엔진 상태표시, 알람표시 등을 해주며, 이 값들을 BCSU와 SSU에 전송해 준다. SSU에서는 디젤엔진을 감시보호하기 위해 RPM 검출, Overspeed 검출, Shut down, Slow down, Emergency Stop 등의 기능을 수행해 주고, 이 값들을 RCSU에 전송해 준다.

위와 같이 서브유니트간 기능수행을 위해서 본 연구에서는 선박 디젤엔진 원격제어 시스템의 태스크를 디젤엔진의 상태에 따라 Stop과 Run상태, Run상태는 Ahead와 Astern상태로 세분하고, 디젤엔진의 동작을 Stop상태의 보존동작, Run상태의 보존동작, Stop상태에서 Run상태로의 전환동작, Run상태에서 Stop상태로의 전환동작, Ahead Run상태에서 Astern Run상태로의 전환동작, Astern Run상태에서 Ahead Run상태로의 전환동작으로 분류하였다. 그리고 각각의 상태에 대하여 다시 서브태스크로 나누어 프로그램을 작성하도록 하여 디젤엔진의 각 상태의 보존동작과 상태간의 전환동작을 구현하였다.

여기에서 Stop 상태의 보존동작은 안전조건을 감시하고 조속기 출력(Governor output)값을 항상 0으로 유지해 주면 되고, Run상태에서 Stop상태로의 전환동작은 Stop Command, Engine Tripped, Wrong Rotation의 조건에 의해 기동되며 조속기 출력값을 0으로 유지해 주면 된다. 또한 Stop상태에서 Run상태로의 전환동작은 시간 지연 요소 등을 포함하고 있어서 많은 실행시간을 요구하므로 다시 서브태스크로 나누어 실행시킨다.

Run상태의 보존동작은 BCSU로부터의 RPM 커맨드에 대응하는 조속기 출력값을 출력해 주는 동작을 수행하면 된다. 그러나 RPM 커맨드는 순간적으로 변화되기 때문에, 여기에 맞추어 조속기 출력값을 즉시 변화시키면 엔진에 무리한 속도변화를 초래하여 엔진 고장의 원인이 된다. 따라서 RPM 커맨드에 대해 일정시간 간격으로 조속기 출력값을 변화시켜야만 한다. 그러나 대

부분의 디젤엔진의 경우 임계영역을 포함하고 있기 때문에 이 영역을 빨리 회피할 수 있도록 조속기 출력값을 출력해 주어야 한다. 이를 위해 RPM이 임계 영역내에 들어간 경우 임계영역 직전의 RPM에 해당하는 조속기 출력값을 출력해 주다가 RPM이 임계 영역의 중간지점을 넘어서는 시점에서 임계영역을 벗어나는 RPM에 해당하는 조속기 출력값을 출력해 주도록 하였다.

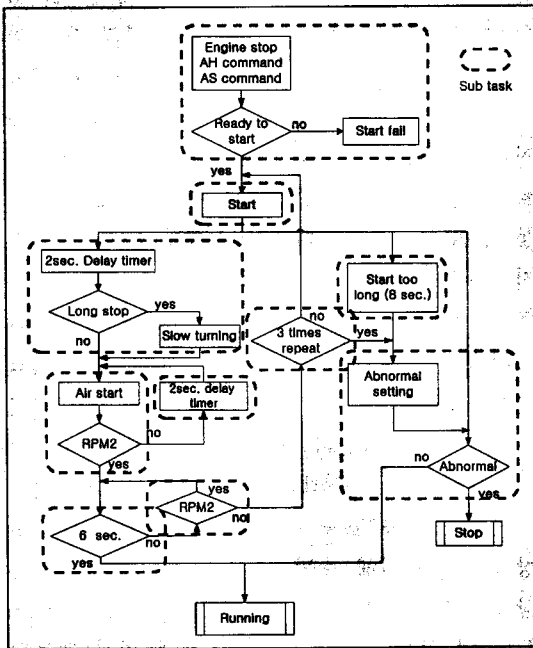


그림 7. 시동 태스크

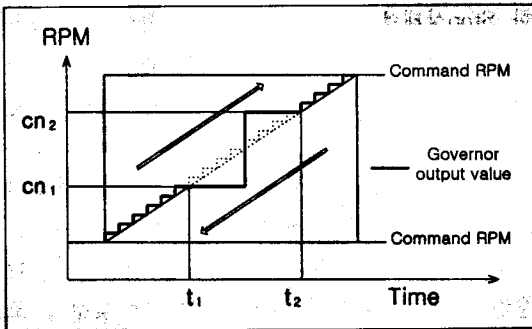


그림 8. 속도제어 방법

디젤엔진 원격제어 시스템에서는 BCSU와 RCSU가 RS-422 시리얼 라인을 통해 서로 통신

을 수행하고 있어서 어느 곳에서라도 원격제어 할 수 있으며, 이때 나머지 하나는 단지 모니터 링만을 수행한다. 실제 제어위치를 변경시키기 위해서는 RCSU측에 있는 절환 스위치(change over switch)를 이용하지만, 그 이전에 어디에서 원격제어를 수행할 것인가를 결정해 둘 필요가 있다.

따라서 두 시스템간에 제어위치를 절환하기 위해서는 두 개의 버튼을 이용하여 서로 약속된 신호에 따라 절환되도록 하여야 한다. 이를 위해 제어위치의 절환을 위해 플래그 변수를 이용하고 있으며, RCSU에서 BCSU로 송신할 때에는 BCSU와 RCSU용의 플리커링을 위한 플래그 값과 절환 스위치의 값을 넘겨주고, BCSU에서 RCSU로 송신할 때에는 BCSU와 RCSU용의 플리커링을 위한 플래그 값만을 넘겨주도록 하였다.

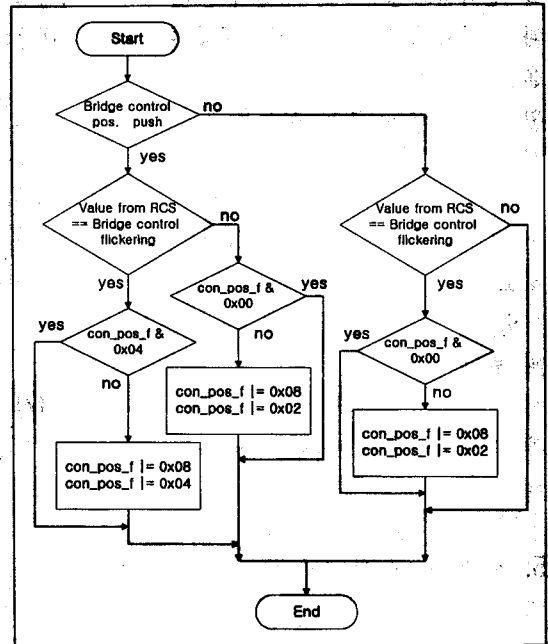


그림 9. 브리타지에서의 제어위치 절환 알고리즘

대규모시스템을 마이크로프로세서 기반형 시스템으로 구축하는 경우 가장 고려해야 하는 것 중에 하나가 메모리의 관리문제이다. 즉 많은 데이터들을 저장해야 하기 때문에 효율적인 메모

리 관리가 필요하다. 따라서 디지털 값을 가지는 데이터들은 관련이 있는 여러개를 모아서 하나의 변수로 관리할 필요가 있으며, 이 때는 변수의 각 비트가 서로 다른 데이터를 가리키게 된다. 본 연구에서는 디젤엔진에 이용되고 있는 입출력 데이터들을 입력 및 출력데이터로 나누고, 논리적으로 연관성이 있는 데이터들을 모아서 동일 변수에 저장하도록 하였다. 또한 각 데이터들의 상태를 모니터링하기 위해서는 이들 값을 LED등을 이용하여 표시해 주어야 한다. LED에 표시하는 방법으로는 ON 또는 OFF가 주로 이용되고 있지만, 경우에 따라서는 경보를 나타내기 위하여 LED를 일정간격으로 ON/OFF하는 플리커링 방법이 필요하게 된다. 본 연구에서는 플리커링을 필요로 하는 데이터들을 별도로 관리하기 위해서 논리적으로 구성되어 있는 데이터들로부터 Steady LED용 데이터와 Flicker LED용 데이터로 분류하고, Flicker LED용 데이터들은 플래그 변수를 이용하여 플리커링 신호를 감지하여 플리커링 하도록 하였다.

시스템에서 이용하고 있는 데이터로는 디지털 입력(DI), 디지털 출력(DO), 아날로그 입력(AI), 아날로그 출력(AO)이 있으며, 특별히 RPM측정을 위해 펄스 카운터에 의한 아날로그 입력이 있다. 이 중에서 RPM측정은 신뢰성과 정밀성이 요구되는 데이터로서 별도의 하드웨어로 구성될 필요가 있으며, DO 데이터들은 5V의 전원을 출력해 주면 되기 때문에 8255 소자의 구동 방법을 이용하여 쉽게 구현할 수 있다. 또한, 선박의 경우 24V가 주류를 이루고 있으므로 전압값을 이용하고 있는 장치의 경우에는 5V에서 24V로 변환해 주는 터미널 보드가 필요하며, 시스템의 안전을 위해 두 전압사이에는 릴레이 등을 이용하여 구성함으로써 절연시켜야 한다.

그러나 나머지 데이터들, 즉 DI, AI, AO용 데이터들은 변환 장치가 별도로 필요하다. 따라서 이들 데이터들은 시스템 내부에서 동일 버스를 이용하여 동일 방식으로 측정하고 출력하는 것

은 소프트웨어를 구성하는데 있어서 대단히 편리할 뿐만 아니라, 하드웨어 구성면에서도 경제적이라고 할 수 있다. 이를 위해 여기에서는 입출력단에 센서별로 어댑터 카드를 장착하고, 시스템 측에서는 하나의 12비트 DAC만을 이용하는 방법을 도입하였다. 즉 AO인 경우에는 DAC를 이용하여 출력하고자 하는 아날로그값에 상응하는 값(0 - 4095)을 해당 어댑터 카드에 출력하도록 하고, DI 및 AI의 경우에는 어댑터 카드에 비교값(0 - 4095)을 내보내어 센서값과 일치될 때 디지털 신호를 발하도록 하여 이 신호를 8255소자로 받아들이도록 구성함으로써 입력값을 판별하도록 하였다.

## 6. 엔진 작동 시물레이션 및 시운전 테스트

세 개의 서브 유니트인 BCSU, RCSU, SSU로 구성된 선박 디젤기관 원격제어 시스템의 제어 기능과 안전운전 기능들은 정밀한 시퀀스 제어에 의해 해당 전자밸브를 작동시키게 된다. 따라서 이러한 실시간 시스템이 완성되면 정확한 동작을 수행하는지를 확인하는 작업은 디젤엔진을 보호하는 의미에서 대단히 중요한 단계 중의 하나라고 할 수 있다.

따라서 선박 디젤엔진 원격제어 시스템의 고기능화를 위해서 하드웨어를 8088CPU로 구성한 시스템의 동작확인을 위한 타임차트 시뮬레이터를 구성<sup>[26]</sup>하였다. 시뮬레이터는 각종 전자밸브들의 ON/OFF 상태와 같은 디지털 값을 실시간으로 표시해 주기 위한 기능과 기관 회전수와 같은 아날로그 값을 실시간으로 표시해 주기 위한 기능으로 이루어져 있다. 시물레이션을 통하여 디젤엔진 원격제어 시스템의 동작상태와 각 전자밸브의 작동시기를 비교하여 본 결과 정상적으로 동작된다는 것을 알수 있었다.

또한 선박 디젤엔진 원격제어 시스템 장치를 실제 엔진(쌍용중공업 B&W 7L35MC, 5320마



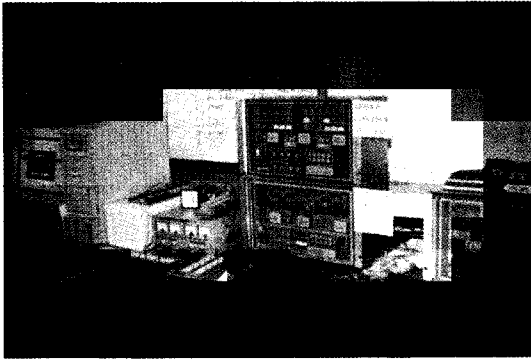


그림 10. 시제품 사진

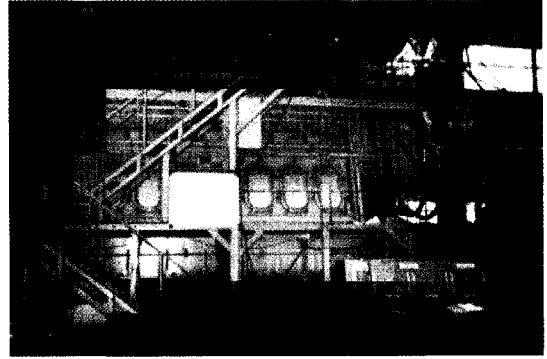


그림 11. 엔진테스트 장면

력)에 장착하여 시험하였다. 본 시험에서는 선내에서의 전원과 부하를 걸기 위하여 각각 24VDC 와 동력계를 사용하였으며 시동, 긴급 후진, 속도 제어의 3가지로 크게 나누어 성능시험을 실시하였다.

시동시험으로는 정상시동과 Start failure로서 Start too long과 3 fail start를 테스트하여 보았으며, 시퀀스 순서가 시뮬레이션시의 테스트 결과와 일치하였다.

긴급 후진 시험에서는 엔진의 회전속도가 RPM4 이상이고 회전상태가 전진방향일 때 제어 레버를 급히 후진방향으로 이동시킨 결과 엔진의 회전수가 RPM2 이하로 떨어진 후 후진방향으로 시동되어 계속 엔진이 운전되는 것을 확인할 수 있었다. 그러나 이 시험은 실제 선박에서는 전진방향의 관성력이 존재하기 때문에 더욱 열악한 환경에서 많은 시험이 있어야 할 것이다.

속도 제어시험에서는 Critical limit에 대하여 위험 개소를 2군데 설정해 두고 실험하였으며 Load up은 30분 동안, Load down은 15분 동안에 목표치 달성 여부를 실험하였다.

## 7. 결론

본 연구에서는 마이크로프로세서를 이용한 선박 디젤엔진 원격제어 시스템의 구현을 위하여 전체 시스템을 RCSU, BCSU, SSU로 나누어 분

산형태로 구성하였다. 또한, 시스템의 동작 구현을 위해 실시간 멀티태스크 기법, 시리얼통신 방식, 각 디바이스의 구동 방법, 아날로그 및 디지털 센서의 입출력 방법, 디젤엔진의 전후진 및 속도제어 방법, 비상후진 및 비상전진 기법, 서로 다른 유니트간의 원격제어위치의 절환 방법 및 비정상상태의 표시를 위한 LED 플리커링 기법 등을 소프트웨어적으로 구성하는 방법에 관하여 논하였다. 시스템을 구성함에 있어서 경제성, 신뢰성, 사용자 편의성을 목표로 하였으며, 다음과 같은 특징을 얻을 수 있었다.

첫째, 실시간 멀티태스크 소프트웨어를 구축함으로써, 기존의 절차형 프로그램 작성법과 달리 대규모 시스템에 적합하며 기능의 추가 및 삭제가 용이하여 새로운 환경변화에 대한 확장성이 뛰어난 시스템을 구현할 수 있었다.

둘째, 태스크의 기능에 따라 타임슬라이스 방식과 이벤트드리븐 방식을 혼용하였으며, 이로 인해 불필요한 태스크의 절환시간을 줄임으로써 전체 시스템의 실시간 성능을 향상시킬 수 있었다. 특히 타이머 태스크를 별도로 두어 시간지연 계산이 필요한 태스크들과의 사이에 플래그변수 즉, 공용변수를 이용하여 지연시간의 시작과 종료를 전달하도록 함으로써 태스크의 독립성 즉, 간결성을 높일 수 있었다. 이러한 기법은 한개의 타이머 태스크로 여러 개의 태스크의 지연시간을 계산해 줄 수 있으므로 타이머 소자도 한개만 있

으면 충분하므로 하드웨어도 간결하게 된다.

셋째, 통신 방식에 있어서 수신만을 인터럽트로 처리하는 방법을 도입하고, 무효 데이터의 처리기법, 가상 링버퍼 개념의 도입 및 링버퍼 수정법 등을 제시함으로써, 타이머 소자의 절약, 통신선 수의 절약, 메모리 절약 등을 통한 하드웨어의 간결화와 데이터의 신뢰성 확보를 추구할 수 있었다.

넷째, 시뮬레이션과 엔진장착 테스트를 통하여 개발된 선박 디젤엔진 원격제어 시스템이 정확한 동작을 하고 있음을 확인할 수 있었다.

마지막으로, 본 연구를 통하여 개발된 디젤엔진 원격제어 알고리즘 및 실시간 멀티태스킹 알고리즘은 실시간 운전이 필요한 플랜트 장치의 마이크로프로세서 기반형 자동화 시스템 구축에 대한 초석이 될 것으로 사료된다.

### 참 고 문 헌

- [1] 中川司, “舶用コンピュータシステムと國際動向”, 日本舶用機關學會誌, 第32卷, 第2號, pp.79-85, 1997.
- [2] 細川成通, “主機關の自動化の現狀と將來動向”, 日韓學術交流講演會, pp.2-11, 1993.
- [3] 渡部文男 外3人, “內航船の近代化とネットワークシステム”, 日本舶用機械學會誌, Vol.32, No.2, pp.120-124, 1997.
- [4] M-800B Bridge Maneuvering System Manual, NABCO, 1995.
- [5] Auto-chief 4 Main Engine Remote Control System Manual, NORCON, 1996.
- [6] DMS 900 Diesel Maneuvering System Manual, Lyngso Marine, 1995.
- [7] Microprocessor System Component Handbook, Vol. I, Vol. II, Vol III, Intel Handbook, 1986.
- [8] Development Systems Handbook, Intel Handbook, 1986.
- [9] 那須川德博, “BootStrap “8086ボ-ドコンピュータの設計&製作”, BootStrap No.1, CQ出版社, pp.6-109, 1992.
- [10] LSIC-86ユ-ザ-スマニュアル, LSI JAPAN CO.LTD, 1991.
- [11] 조순복, 김광희, 8086/8088 마이크로프로세서와 그 주변소자들, 집문당, 1991.
- [12] John Uffenbeck, THE 8086/8088 FAMILY, Prentice-Hall International Inc., 1987.
- [13] 横山直陸, C言語による制御實習入門, 哲學出版, 1992.
- [14] 風間悦夫 外2人, C言語による8086系制御用マイコン使い方, 日刊工業新聞社, 1989.
- [15] 麥田憲司 外6人, “Cによる組みこみ機器用プログラミング”, トランジスタ技術 SPECIAL No.24, CQ出版社, 2-171, 1990.
- [16] L35MC, Maneuvering System Manual, 쌍용중공업, 1995.
- [17] D.C.wakman, BP's Performance Monitoring System for Marine Diesel Engine, IMF Trans, 1983.
- [18] 최진태·김종식, “실시간운영체제와 로봇제어기에의 적용”, 대한기계학회지, 제36권, 제3호, pp.231-242, 1996.
- [19] 佐竹秀己, “リアルタイムシステムのOSとデバイスドライバ, 인터-フェ-ス, pp.136-151, 1996.
- [20] 大原, 田, Peter Petrov, 實踐リアルタイムプログラミング技法, オ-ム社, 1991.
- [21] 宮崎久則, 森大資, “マルチタスク・プログラミング入門”, 트라技コンピュータ, CQ出版社, pp.20-79, 1992.
- [22] <http://cis1.snu.ac.kr/rtarc/reall.html>
- [23] 宮崎誠一ら, “標準シリアルインターフェ-ス”, 트ランジスタ技術 Special No.8, CQ出版社, pp.12-35, 1992.
- [24] 石川裕次, “シリアル 轉送用LSIの使い方”, 트ランジスタ技術 Special No.8, CQ出版社,

pp.50-75, 1992.

[25] Robert P. Benedict, Fundamentals of Temperature, Pressure and Flow Measurements, John Wiloy & Sons N.Y., 1983

[26] 정경열, 류길수, "선박 디젤기관 원격제어 시스템을 위한 시뮬레이터 개발에 관한연구", 한국해양대학교 산업기술연구소논문집, 제16편, pp.1-9, 1998.