

상용 관성항법시스템용 CDU(Control Display Unit) 설계

Design of a Control Display Unit for Commercial Inertial Navigation Systems

김정원* 신대식** 황동환* 이상정* 박용운***
Kim, Jeong-Won Shin, Dae-Sik Hwang, Dong-Hwan Lee, Sang-Jeong Park, Yong-Woon

ABSTRACT

This paper proposes a design method of a CDU(Control Display Unit) for commercial INS(Inertial Navigation Systems). In order to guarantee reusability and extendability, the design method is based on the class programming of the Windows operating system. Since the CDU has abstracted functions and variables, it can be interfaced with any INS. It is also easy to extend the designed functions using inheritance and polymorphism of the class. In order to show usefulness of the CDU, it has been implemented for the H-726 INS.

주요기술용어(주제어) : CDU(Control Display Unit), Inertial Navigation System(관성 항법 시스템), Class(클래스), Windows operating system(윈도우즈 운영 체제)

1. 서론

관성항법시스템(INS : Inertial Navigation System)은 자이로스코프와 가속도계를 통해 항체의 각속도와 가속도를 측정하여 항체의 위치, 속도 및 자세 정보를 제공한다. 또한 항법 정보를 빠르게 제공하며 외부 환경에 강인한 장점을 가지고 있다. 관성항법시스템은 제2차 세계대전의 촉발로 군사적 요구에 의해 1940년대부터 개발 되었으며 독일의 V-2 로켓에 처음으로 적용된바 있다. 1950년대에 Gimbal형 관성 항법 시스템(GINS)이 소개되었고 1960년도에는

Strapdown형 관성 항법 시스템(SDINS)이 소개되어 오늘날까지 계속 발전이 이루어지고 있다. 관성 항법 시스템은 각종 선박, 잠수함, 항공기, 무인항공기, 우주 발사체, 유도무기, 지상차량 및 무인로봇에 이르기까지 여러 분야에서 이용되고 있다.

일반적으로 상용 관성 항법 시스템은 관성 센서와 항법 컴퓨터 등이 패키지 형태로 이루어져 있고 운용자는 CDU(Control Display Unit)라는 장비를 통하여 관성 항법 시스템의 동작을 제어하고 항법 데이터를 확인한다^[1]. CDU는 MMI(Man Machine Interface)의 일종으로 항법 시스템의 초기위치 값을 로드하는 기능, 항법 시스템 동작 모드 변경 명령, 항법 컴퓨터의 결과를 표시하는 기능 및 항법 시스템의 고장 진단(BIT, Built-In-Test)을 수행한 결과를 표시하는 기능을 수행한다^[1,2]. 일반적으로 CDU는 관성항법시스템을 제작한 회사에서 공급하며 대부분 사용자 입력장치인 키보드, 데이터를 표시하기 위한 모니터와

† 2006년 12월 11일 접수~2007년 3월 5일 게재승인

* 충남대학교(Chungnam National University)

** (주)한양네비콤(Hanyang Navicom Co., Ltd.)

*** 국방과학연구소(ADD)

주저자 이메일 : dhhwang@cnu.ac.kr

항법 시스템과의 통신을 위한 통신 포트가 결합된 휴대용 컴퓨터(portable computer) 형태를 가진다. 이러한 CDU는 일반적인 컴퓨터와 유사한 기능을 가지고 있지만 적은 수만 생산되므로 고가이다. 또한, 사용하는 관성 항법 시스템에 따라 데이터 종류, 길이, 형태 등이 달라지므로 관성 항법 시스템이 바뀌면 CDU도 교체하여야 한다. 항법 시스템이 바뀌에 따라 사용자가 CDU의 내부를 변경하려고 하여도 CDU 개발 회사에서는 탑재된 프로그램을 공개하거나 수정을 허락하지 않으므로 사용자가 자유롭게 변경하거나 기능을 추가하는 것이 쉽지 않다.

본 논문에서는 적은 비용으로 손쉽게 구현 가능하고 기능 확장 및 변경이 용이한 윈도우즈 운영체제 기반 상용 관성항법시스템용 CDU 설계 기법을 제안한다. 최근의 컴퓨터 기술 발달로 인하여 저가격의 고성능 컴퓨터를 쉽게 구입할 수 있고 CDU가 가져야 할 통신 기능도 확장 카드(Card) 등의 하드웨어를 이용하여 추가하는 것이 쉬우므로 일반적인 컴퓨터를 이용하여 CDU를 구현하는 것이 가능하다. CDU 소프트웨어는 재사용성 및 기능 확장을 위하여 클래스를 이용하여 설계하였다. 설계한 클래스는 기능 추상화를 통하여 데이터 종류, 길이, 형태 등에 관계 없이 사용 가능할 수 있도록 하였고 상속 및 함수 재정의(Overriding) 등의 특징을 통하여 쉽게 기능 확장 및 수정이 가능하게 하였다. 또한 GUI(Graphic User Interface)를 통하여 사용자가 손쉽게 사용할 수 있도록 하였다. 제안한 기법은 정밀 관성 항법 시스템의 하나인 H-726 MAPS(Modular Azimuth Position System)라는 시스템용 CDU를 구현하고 설계 내용을 확인하였다.

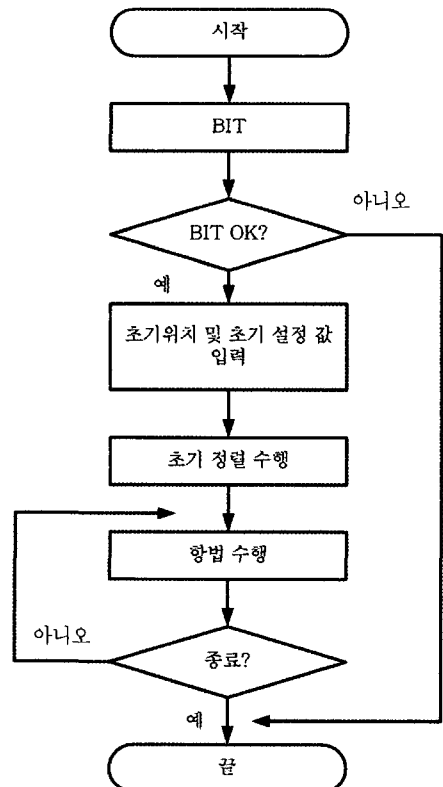
본 논문의 전체 구성은 다음과 같다. 2절에서는 CDU의 기능 추상화 과정 및 이를 이용한 클래스 설계에 대하여 서술하였다. 3절에서는 제안한 CDU 설계 기법을 검증하기 위하여 상용 INS인 H-726 MAPS용 CDU를 구현한 내용을 서술하였고 차량 실험 결과에 대하여 서술하였고 마지막으로 결론 및 추후 과제를 제시하였다.

2. 기능 추상화를 통한 상용 관성 항법 시스템용 CDU 클래스 설계

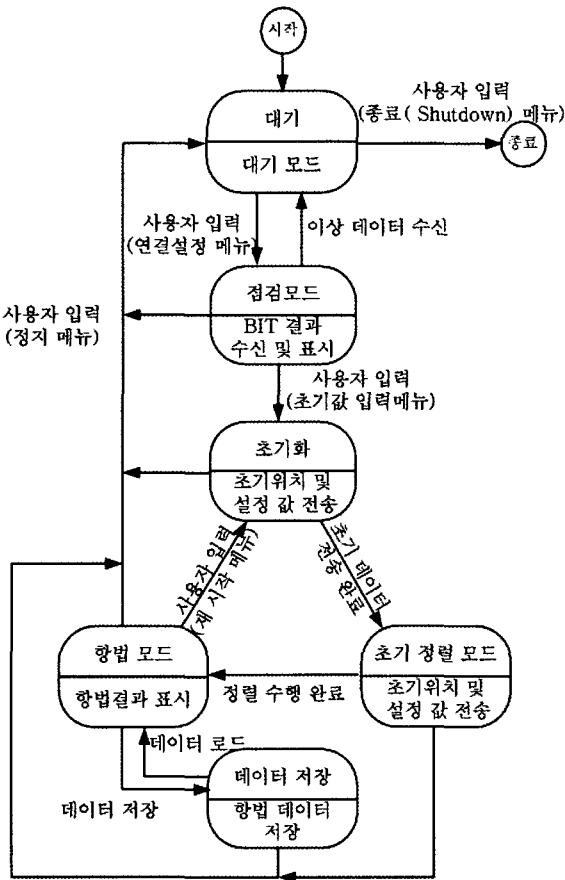
가. 관성항법시스템용 CDU의 기능

CDU는 관성항법시스템의 시작에서 종료까지 각 단계의 동작을 관장하고 관성항법시스템으로부터 수신한 데이터를 표시하는 기능을 수행한다. 이것은 관성 항법 시스템의 동작과 관계가 있으므로 관성항법 시스템의 기본적인 동작을 그림 1에 나타내었다.

관성 항법은 초기 정렬과 순수 항법으로 구성되며 먼저 하드웨어의 동작 점검을 위하여 BIT(Built-In-Test)기능을 수행한다. 관성 항법 시스템이 동작을 시작하면 자체 점검 결과를 출력하고 CDU는 이를 표시하여 사용자가 다음 운용 모드를 선택할 수 있게 한다. 초기 정렬 단계에서 CDU는 항법 시스템의 초기 위치, 정렬 시간 및 정렬 기법 등의 정보 및 명령을 전송하고 초기 정렬 결과와 남은 시간을 표시하여



[그림 1] 관성 항법 시스템 동작 순서

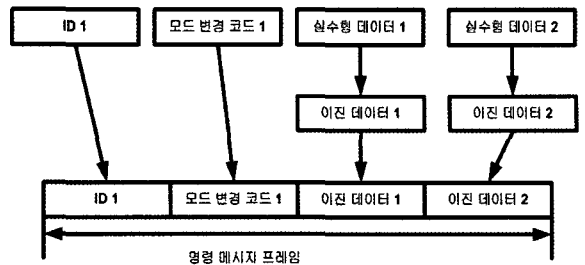


[그림 2] CDU 기본 기능

사용자가 인식할 수 있도록 한다. 초기 정렬 이후 항법 시스템은 항법을 수행하고 CDU는 출력되는 항법 결과를 사용자에게 표시한다. 항법 수행 중에도 시스템의 종료, 시스템 상태 정보 요구 등의 명령을 수행한다. 이와 같은 CDU의 기본 동작은 그림 2와 같이 나타내었다.

나. CDU 기능 추상화

전술한 바와 같이 CDU의 기능은 근본적으로는 유사하지만 관성 항법 시스템이 사용하는 데이터 종류 및 길이, 동작 순서등에 의하여 구체적인 기능은 달라질 수 있다. 본 논문에서는 CDU의 기본 기능을 추상화한 후 클래스로 설계하여 데이터 종류 등이 달라지더라도 전체를 변경할 수 없는 설계 기법을 서술한다.



[그림 3] 명령 메시지 프레임 생성 과정

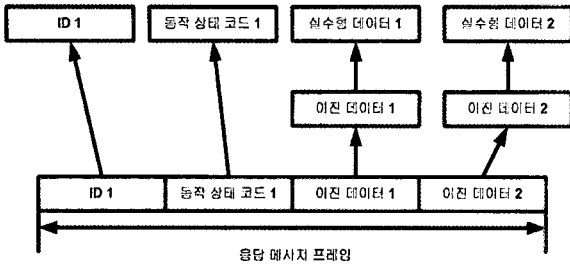
CDU의 기능은 크게 명령 메시지 생성과 응답 메시지 해석이다. 명령 메시지는 관성 항법 시스템의 동작을 제어, 초기값을 입력, 항법과 상태 정보를 요구하는 메시지도고 응답 메시지는 각각의 명령 메시지에 대한 응답으로 항법 정보, 상태 정보 등을 포함한다.

명령 메시지는 초기 위치 등 실수형의 데이터, 명령 메시지 ID(Identifier)와 모드 변경 코드 등과 같이 하나의 비트 또는 여러 비트가 의미를 가지는 데이터로 이루어진다. 초기 위치와 같은 실수형의 데이터는 CDU가 전송하기 전에 관성 항법 시스템이 요구하는 형태에 맞춰 이진 데이터로 변환하여야 하고 하나의 프레임으로 구성한 후 전송하여야 한다. 그림 3은 명령 메시지 생성 과정을 보여주고 있다.

명령 메시지 프레임은 그림 3과 같이 ID, 모드 변경 코드, 이진화된 실수형 데이터 등이 함께 포함될 수도 있고 이진 데이터 또는 실수형 데이터만 포함될 수도 있다. 이것의 종류, 길이, 형태(Format) 등은 관성 항법 시스템에 따라 달라진다. 명령 메시지 생성은 실수형 데이터를 이진 데이터로의 변환하는 것과 여러 데이터들을 순서대로 결합하여 메시지 프레임 생성하는 것으로 나누어진다.

응답 메시지는 항법 데이터와 동작 상태 정보 등이 하나의 프레임으로 구성된다. 응답 메시지 해석은 이러한 응답 메시지 프레임으로부터 원하는 데이터를 추출하는 것으로 명령 메시지 생성의 반대 과정이다. 그림 4는 응답 메시지 해석 과정을 나타내는 것이다.

응답 메시지 프레임도 명령 메시지와 마찬가지로 ID, 동작 상태 코드, 이진화된 실수형 데이터로 이루어지며 관성 항법 시스템에 따라 다른 형태를 가지게 된다.



[그림 4] 응답 메시지 해석 과정

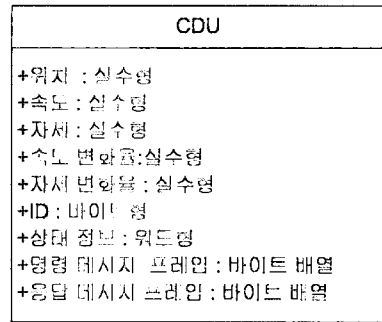
전술한 기능외에도 그래픽 인터페이스를 통한 데이터 표시, 메뉴등과 데이터의 송수신을 담당하는 통신 인터페이스 등도 CDU를 구성하는 중요한 기능이며 그래픽 인터페이스는 윈도우즈 운영 체제가 제공하는 기본 사용자 인터페이스 프레임인 SDI (Single Document Interface), MDI(Multi Document Interface) 등을 사용할 수 있고 통신 인터페이스는 상용 통신 하드웨어에 대한 디바이스 드라이버 및 API(Application Program Interface)를 사용할 수 있으므로 전술한 기능에 대한 클래스 설계를 통하여 CDU를 설계할 수 있다^[4,5].

다. 추상화된 기능을 통한 CDU 클래스 설계

특정 관성 항법 시스템에 국한하지 않고 사용할 수 있는 CDU를 설계하기 위해서 2.나 절의 기능을 클래스로 설계하였다. 먼저 CDU에서 사용하는 데이터를 저장하기 위한 클래스 멤버 변수(Member variable)를 선언한다. 관성 항법 시스템용 CDU는 기본적으로 운용시간, 위치, 속도, 자세 데이터를 사용하므로 이 데이터에 대한 변수를 실수형 멤버 변수로, ID, 모드 변경 코드 등은 1바이트의 변수로 선언하며 명령 메시지와 응답 메시지를 저장할 배열 변수를 선언한다. 그림 5는 정의하는 클래스 멤버 변수를 나타낸 것이다.

이외에 관성 항법 시스템에 따라 추가되는 데이터들은 클래스의 상속성을 이용하여 자식 클래스에서 추가적으로 선언하면 되므로 이미 설계한 클래스의 변경 없이 사용 가능하다^[6].

명령 메시지 생성 기능은 전술한 바와 같이 실수형 데이터를 이진 데이터로 변환하는 기능과 여러 데이터들을 순서대로 결합하여 메시지 프레임을 생성하는



[그림 5] CDU 클래스 멤버 변수 구성

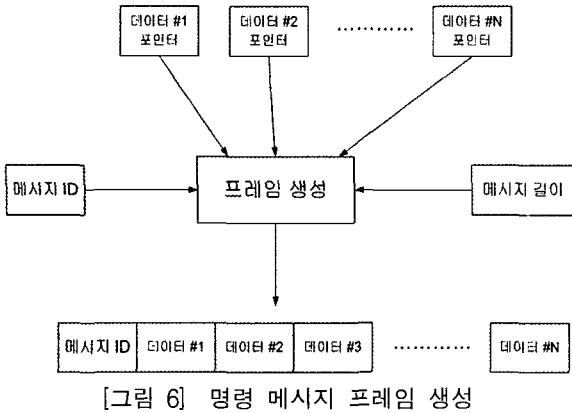
기능으로 나눌 수 있다. 실수형 데이터를 이진 형태로 나타내는 방법은 IEEE 745 표준에 따라 부호, 지수부, 가수부로 표시하는 것이 일반적이지만^[3] 시스템에 사용되는 CPU의 종류에 따라 표현 범위, 지수부와 가수부의 길이 등이 달라지므로 관성 항법 시스템과 CDU가 서로 다른 CPU를 사용하는 경우에는 데이터 오류가 발생할 수 있다. 이를 방지하기 위하여 메시지 생성 멤버 함수 중 실수형 데이터를 이진 데이터로 변환하는 기능은 분해능 정보를 이용하여 비트열로 변환하는 방법을 사용하여 설계한다. 실수형 데이터를 이진 형태로 변환하는 것은 아래와 같이 실수형 데이터와 분해능의 역수의 곱으로 수행한다.

$$\text{변환된 이진 데이터} = \text{실수 데이터} \times \frac{1}{\text{분해능}}$$

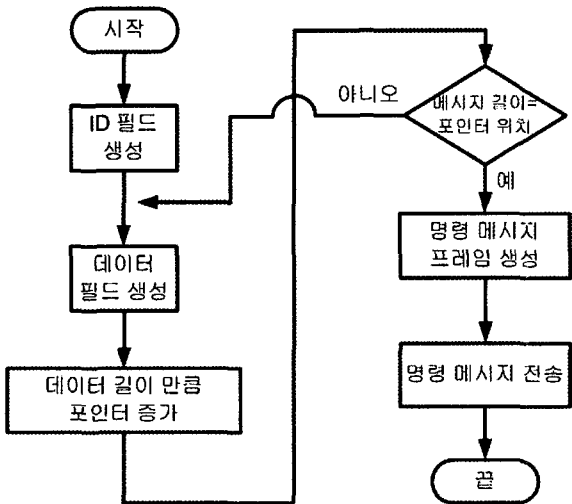
이와 같이 변환된 이진 데이터는 소수점이 없는 정수의 비트열이므로 CPU의 종류에 관계 없이 분해능만 결정되면 사용 가능하고 역으로 분해능을 곱하여 실수 데이터로 변환 가능하다.

프레임 생성 기능은 이진화된 실수형 데이터와 ID 같은 이진 데이터를 순서에 맞게 정렬하여 프레임을 만드는 것이다. 이 기능의 결과, 프레임을 구성하는 각 필드 데이터는 배열 형태의 메모리에 저장된다. 그림 6, 7은 명령 메시지 생성 기능 멤버 함수의 동작 개념과 순서를 나타낸 것이다.

응답 메시지 해석 기능은 응답 메시지 프레임으로부터 각 필드를 분리하고 이진화된 실수형 데이터는 다시 정해진 형태의 실수로 변환하고 상태 정보등의 이진 데이터는 정해진 길이로 분리하여 이진 데이터



[그림 6] 명령 메시지 프레임 생성

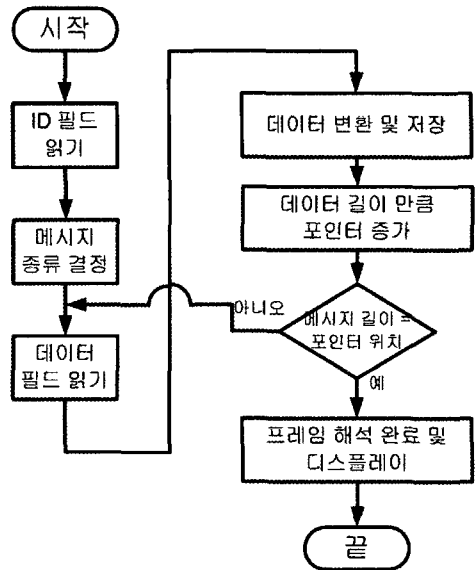


[그림 7] 명령 메시지 생성 순서도

가 의미하는 정보로 맵핑(Mapping)시키는 것이다. 이진화된 실수형 데이터의 변환 기능은 명령 메시지 생성과 반대의 과정을 거쳐 수행한다. 그림 8은 응답 메시지에서부터 데이터 필드를 분리하면서 해당 데이터로 변환하는 과정을 나타내고 있다.

이와 같이 설계한 명령 메시지 생성과 응답 메시지 생성 멤버 함수는 메시지의 종류에 관계 없이 ID, 메시지 시작 포인터, 길이, 분해능을 입력하면 사용할 수 있으므로 메시지마다 각각의 함수를 정의하여 사용할 필요가 없다.

설계한 CDU 클래스의 구조는 그림 9와 같다. 이와 같이 설계한 CDU 클래스는 관성 항법 시스템의 기본 기능을 포함하고 있고 각각의 멤버 함수들은 추



[그림 8] 응답 메시지 해석 순서도

CDU	
+위치 : 실수형	
+속도 : 실수형	
+자세 : 실수형	
+속도 변화율:실수형	
+자세 변화율: 실수형	
+ID : 바이트형	
+상태 정보 : 워드형	
+명령 메시지 프레임 : 바이트 배열	
+응답 메시지 프레임 : 바이트 배열	
+ 실수형 데이터 변환(실수 데이터, 분해능)	
+ 이진 데이터 실수 변환(이진 데이터, 분해능)	
+ ID 생성(ID 필드 포인터)	
+ ID 해석(ID 필드 포인터)	
+ 상태 코드 해석(상태 코드 필드 포인터)	
+ 명령 메시지 생성(메시지 길이, 데이터 포인터)	
+ 응답 메시지 해석(메시지 길이, 데이터 포인터)	

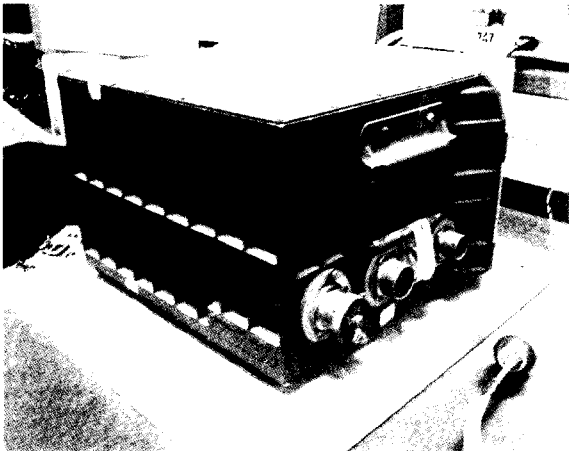
[그림 9] CDU 클래스 구조

상화된 기능을 통하여 관성 항법 시스템이 사용하는 데이터 형태, 길이, 종류에 관계 없이 사용할 수 있다. 이 외에 메시지의 오류 점검 기능, 시스템 점검 등 추가적인 기능이 필요한 경우에는 CDU 클래스를 상속하고 새로운 변수와 함수를 추가하여 확장할 수도 있다.

3. CDU 구현 및 검증

제안한 CDU 설계 기법을 검증하기 위하여 Honeywell사의 정밀 관성 항법 시스템인 H-726 MAPS(Modular Azimuth Positioning System)의 CDU를 설계한 클래스를 이용하여 구현하였다. 그림 10은 H-726 MAPS의 모습을 나타낸 것이다.

MAPS용 CDU는 노트북 PC 기반으로 마이크로소프트사의 윈도우즈용 소프트웨어 개발 도구인 Visual C++ 6.0을 사용하여 제안한 클래스 기반으로 구현하였다.



[그림 10] H-726 MAPS

가. H-726 MAPS의 메시지 구조 및 종류

H-726 MAPS는 SDCL(Synchronous Data Link Control)와 RS-485를 프로토콜로 사용하고 CDU의 모드 변경 명령, 상태 정보 또는 항법 정보 요구 명령에 대하여 응답하는 형태로 동작하고 하나의 명령 메시지에 대하여 하나의 응답 메시지를 송신하는 통신 방식(Half-duplex)을 취한다. 다음 그림 11은 MAPS의 명령과 응답 메시지 형태를 나타내고 있다.

그림 11에서 보듯이 명령 메시지와 응답 메시지는 유사한 구조를 가지고 있다. 각각의 메시지는 시작과 끝을 나타내는 플래그(Flag), 명령과 응답 메시지의 종류를 구분하는 ID(Identification)로서 명령과 응답 코드, 데이터, 오류 검사 코드로 이루어져 있고 응답 메시지의 MAPS의 동작 상태를 나타내는 상태 코드

Start Flag	Command Code	Data #1	Data #N	Error Check #1	Error Check #2	End Flag
------------	--------------	---------	-------	---------	----------------	----------------	----------

(a) 명령 메시지 포맷

Start Flag	Response Code	Status #1	Status #2	Data #1	Data #N	Error Check #1	Error Check #2	End Flag
------------	---------------	-----------	-----------	---------	-------	---------	----------------	----------------	----------

(b) 응답 메시지 포맷

[그림 11] H-726 MAPS 메시지 포맷

[표 1] 사용된 명령 및 응답 메시지 종류

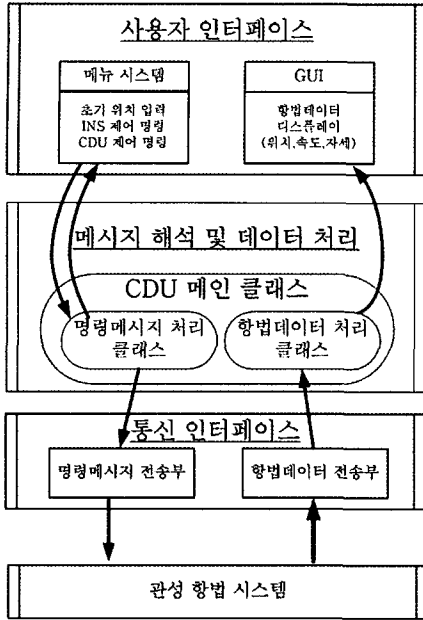
명령 메시지	응답 메시지
상태 정보 요구	상태 정보
점검 정보 요구	BIT 결과
초기 정보 입력	상태 정보
초기 설정값 입력	상태 정보, 입력한 설정값
항법 데이터 요구	상태 정보, 항법 데이터
재정렬 요구	상태 정보
재시작 요구	상태 정보
종료 명령	상태 정보

가 추가된다^[6]. 명령 메시지 각각에 하나의 응답 메시지가 대응하며 MAPS용 CDU를 구현하기 위하여 사용한 메시지의 종류는 표 1과 같다.

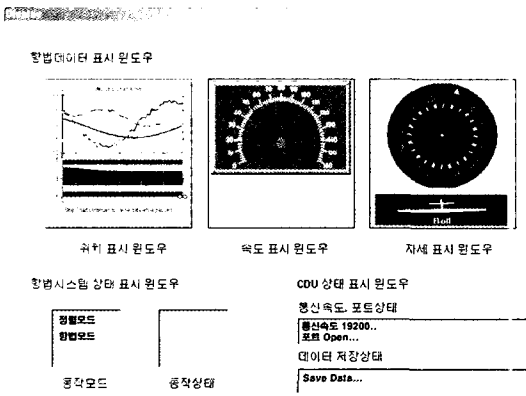
나. H-726 MAPS용 CDU 구현

MAPS CDU의 사용자 인터페이스부인 메뉴, 글자 상자, 그림 상자 등은 윈도우즈가 제공하는 것을 사용하였고 메뉴가 선택되었을 때 수행하는 기능과 글자 및 그림 상자에 데이터를 표시하는 기능은 설계한 클래스를 상속하여 구현하였다. 다음 그림 12는 본 논문에서 제안한 방법을 적용한 MAPS CDU의 구조를 나타내고 있다.

사용자 인터페이스 윈도우는 위치, 속도, 자세 데이터를 표시하는 항법 데이터 표시부, H-726의 동작 모드 및 상태 표시부, CDU의 통신 상태 및 데이터 저장 등의 동작 상태 표시부로 구성된다. 데이터 표시부는 속도계, 나침반 모양의 계기판 형태로 GUI를 구현하여 운용자가 항법 결과를 쉽게 인지할 수 있도록 하였고 메뉴를 선택하면 부 윈도우를 실행하여 저



[그림 12] 구현된 CDU 구조

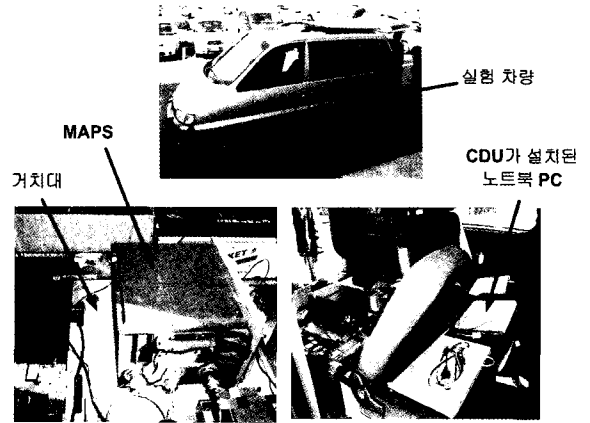


[그림 13] CDU 프로그램 실행 화면

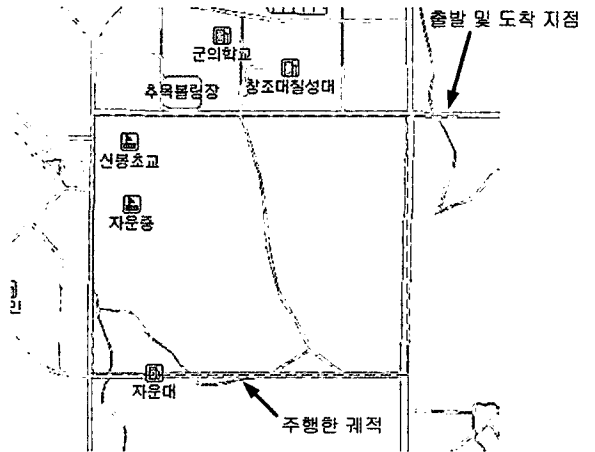
장된 이전 위치, 속도, 자세 등을 확인할 수 있도록 하였다. 그림 13은 CDU 프로그램의 실행 모습을 나타낸다.

다. 차량 실험을 통한 CDU 동작 검증

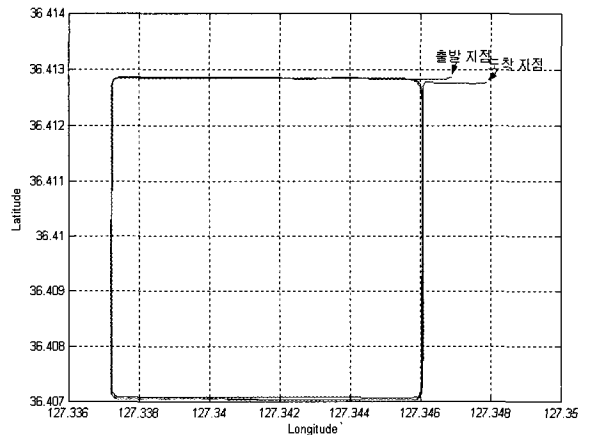
차량 실험을 통하여 CDU의 동작을 검증하였다. 구현한 CDU를 이용하여 초기 위치와 영속도 보정 여부 등의 설정값을 MAPS에 입력한 후 차량 주행 중



[그림 14] 차량 실험 장치 구성



[그림 15] 차량 실험 도로 지도

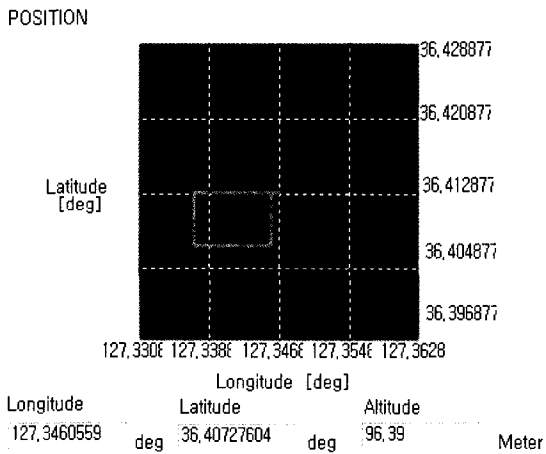


[그림 16] CDU를 이용하여 수집한 MAPS의 위치 궤적

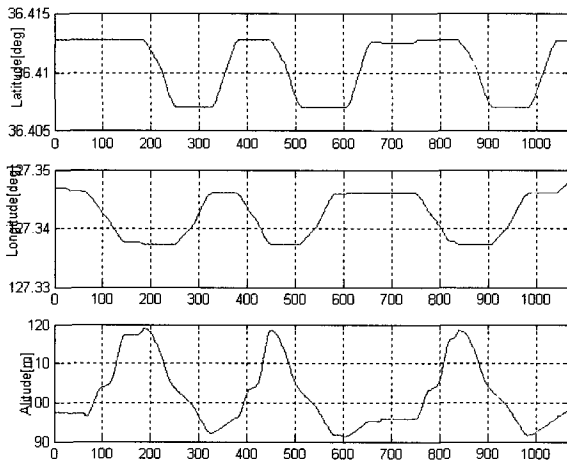
에 1초 간격으로 항법 데이터 요구 명령 메시지를 MAPS에 전송하여 수신한 항법 데이터를 확인하였다. 그림 14는 차량 실험 장치를 보여주고 있다.

H-726 MAPS와 거치대, 전원, CDU가 설치된 노트북 PC를 차량에 장착한 후 대전시 유성구에 위치한 자운대 내의 도로를 주행하였다. 실험 대상인 도로는 그림 15의 지도에서 보듯이 사각형의 형태를 가지고 있다.

그림 15의 도로를 3회 주행하고 저장된 위치, 속도, 자세의 궤적을 확인하여 CDU의 동작을 확인하였다.



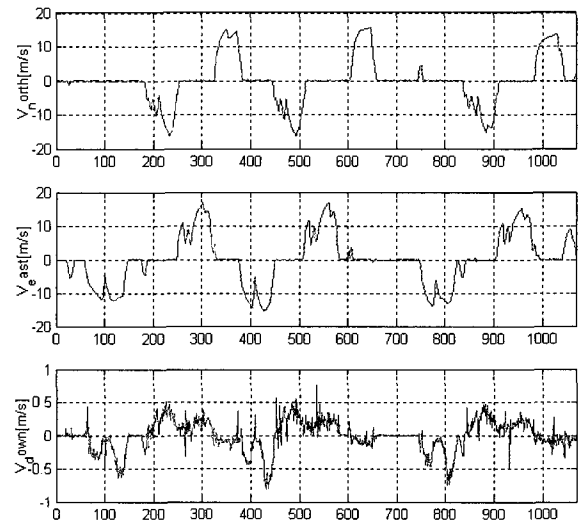
[그림 17] 위치 궤적 디스플레이 윈도우 실험 모습



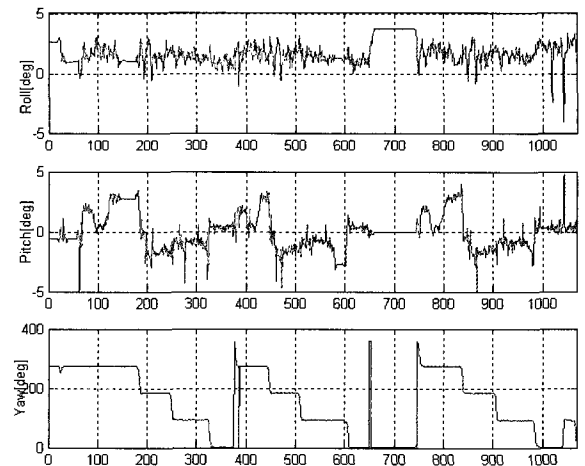
[그림 18] CDU를 이용하여 수집한 MAPS의 위치 데이터

그림 16은 수집한 데이터를 이용하여 표시한 위치 궤적이고 그림 17은 실시간으로 표시된 위치에 대한 CDU 부 윈도우를 나타낸 것이다.

실험 결과에서 CDU를 이용하여 저장한 위치 데이터의 궤적은 실험한 도로와 같은 직사각형의 형태를 가지는 것을 알 수 있고 데이터를 손실없이 잘 수신하는 것을 알 수 있다. 다음 그림 18~20은 위치, 속



[그림 19] CDU를 이용하여 수집한 MAPS의 속도 데이터



[그림 20] CDU를 이용하여 수집한 MAPS의 자세 데이터

도, 자세를 나타낸 것이다.

그림 18~20의 결과에서 볼 수 있듯이 CDU가 MAPS의 위치, 속도, 자세 데이터를 수신하여 오류 없이 처리한 것을 알 수 있고 이것으로부터 제안한 기법의 유효성을 확인할 수 있었다.

4. 결론

본 논문에서는 관성 항법 시스템용 CDU 설계 기법을 제안하였다. 제안하는 CDU 설계 기법은 객체 지향 기법에 기반한 것으로 관성 항법 시스템의 종류에 관계없이 재사용이 가능하고 기능 보완 및 추가가 용이한 구조를 가진다. CDU의 기능을 크게 명령 메시지 생성과 해석 기능으로 추상화하여 클래스로 설계하였으며 이것은 관성 항법 시스템이 사용하는 메시지의 형태나 길이에 제약을 받지 않고 사용할 수 있어 여러 가지 상용 관성 항법 시스템에 적용 가능하다. 자주포의 항법 및 자세 센서로 사용되는 H-726 MAPS에 제안한 기법을 적용하였으며 차량 실험을 통하여 제안한 기법의 기능 및 성능을 확인하였다.

추후에 통신 방식과 데이터 형식 등이 다른 관성

항법 시스템의 CDU 구현에 논문에서 제시한 기법을 적용할 예정이며 기능 추상화를 통한 설계외에 UML(Unified Modelling Language)과 같은 언어를 이용하여 CDU의 동작 순서, 상태 변환 등에 대한 모델링을 사용하여 CDU를 설계하는 기법을 제시할 것이다.

참 고 문 헌

- [1] G. M. Siouris, *Aerospace Avionics Systems*, Academic Press, San Diego, California, 1993.
- [2] C. R. Spitzer, *The Avionics Handbook*, Editor, CRC Press, Boca Raton, Florida, 2001.
- [3] Assembled by Martin Horvat, *Standard for binary floating decimal point ANSI/IEEE 745*, IEEE Standard, October, 2005.
- [4] 송호중 역, *C++ 프로그래밍의 이해*, 한빛 미디어, 서울, 2001.
- [5] 이상엽, *Visual C++ 프로그램 바이블*, 영진출판사, 서울, 1997.
- [6] *Modular Azimuth Position System Technical Description*, Honeywell, Military Avionics Division, Florida. 1989.