

체계공학 접근방법을 통한 제어시스템 설계에 관한 연구

A Study on the Control System Design through Systems Engineering Approach

안 장 근*

Ahn, Jang-Keun

ABSTRACT

There are several kinds of error factors in control system design. All error factors must be analysed before designing the control system. Therefore, each error factor must be compensated and eliminated completely. Systems Engineering can solve these error factors. In this paper, systems engineering approach on control system design are studied under model based systems engineering with RDD-100, Matlab-Simulink. Systems Engineering shall be used in defense development from control system design to system development.

주요기술용어 : Systems Engineering(체계공학), Requirement Analysis(요구사항 분석), Function Analysis(기능분석), Synthesis(조합), ECU(Electronic Control Unit)

1. 머리말

최근 시스템의 복잡성이 증가하고 요소 요소들간의 관계가 기하급수적으로 증가하여, 최초 설계안 대로 동작하지 않는 비효율적인 제어시스템 개발을 반복하고 있다. 시스템이 복잡해질수록 시스템 개발의 성패를 좌우하는 제어시스템의 설계는 그 무엇보다 중요시되고 있다. 즉 시스템 개발의 핵심부분으로 인식되고 있는 것이 제어시스템 설계라는 뜻이다. 하지만 제

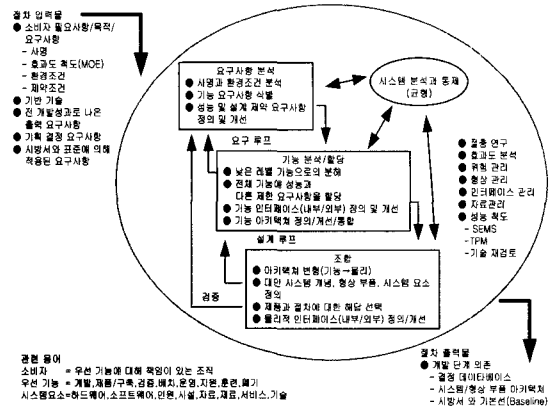
어시스템을 설계하기 위한 제반요소와 적용방법들이 혼재되어 있어 명확한 해결방법을 제시하지 못하고 있는 실정이다. 시스템의 달성코자하는 목적과 요구사항을 최초 설계부터 반영치 못하고 개발자의 경험과 지식에 의존하여 제어시스템을 설계하는 것은 이제는 지양해야 한다.

본 논문에서는 자동차 엔진시스템의 핵심인 ECU(Electronic Control Unit)에 대한 제어시스템 설계를 체계공학 접근방식에 의거, 우선 요구사항 분석부터 기능분석/할당, 조합(아키텍처 설계) 및 인터페이스 설계까지 구축하는 과정을 보여주고, 이를 기반으로

* 국방과학연구소 연구원

ECU에 대한 논리구조 및 연산구조에 대한 모델링과 시뮬레이션(Modeling & Simulation)을 수행하여 검증한다. 이 과정에서 국제시스템공학회(INCOSE : International Council On Systems Engineering)에 SE 전산지원도구로 등재되어 있는 RDD-100과 Matlab을 사용하였다.^[1]

ECU의 최적화된 공연비(air-fuel ratio)를 구현하는 과정에서 체계공학의 중요성과 역할을 보여주고자 한다.



[그림 1] MIL-STD-499B의 체계공학 과정

2. 체계공학 접근방식 (Systems Engineering Approach)

체계공학 접근방식이란 간단하게 표현하자면 체계공학 과정(SEP : Systems Engineering Process) 및 M&S(Modeling & Simulation)를 적용하는 것이다. 체계공학 과정은 MIL-STD-499B(Draft) 표준에 의거^[2] (1) 요구사항 분석, (2) 기능 분석/할당, (3) 조합, (4) 시스템 분석과 통제에 대한 반복적인 과정을 가지고 있다. 이 과정은 상부단계(시스템 레벨)에서 시작하여 하부단계(구성품 레벨)까지 파급되어 결국에는 시스템 해답(system solution)을 이끌어 낸다. 설계에 필요한 결정을 내리는데 필수적인 최하부 단계의 설계 반복이 이 과정을 통해 이루어져야 한다. 각각의 반복동안, 많은 개념 대안들이 절충연구(trade-off study)동안에 도출되고, 분석되며, 평가되어 진다. 체계공학 과정상의 진행단계에 대한 전체적인 그림은 아래와 같다.

M&S는 체계공학 과정을 통해 얻어진 시스템 규격의 검증을 실시하는 부분으로 개발코자 하는 시스템의 개발 목적이 충분히 시스템 구성요소에 반영되었

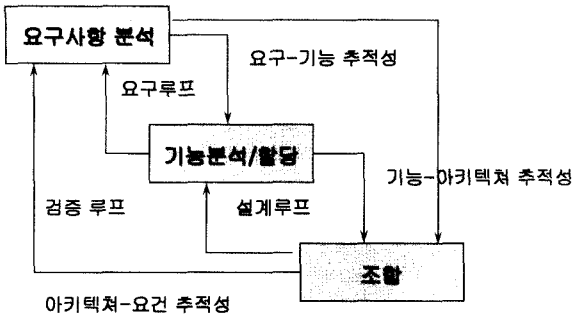
는지를 확인하는 과정이다. 이 과정에서 문제점이 발생시 문제 해결을 위해 최초 요구사항 분석부터 다시 반복하여 실시하여야 한다.

이 논문에서는 시스템공학에서 중요시 여기는 체계공학 과정(SEP)와 M&S를 통해 제어시스템 설계에 접근하였다.

3. 제어 시스템 설계

이 논문에서는 자동차 엔진 제어에 사용되는 ECU 설계에 있어서의 최적화된 공연비(air-fuel ratio)로 알려져 있는 14.6:1^[3]을 얻기 위해 요구사항과 기능간의 상호 관계를 정의하는 설계 과정을 체계공학 접근방식을 활용하였다. ECU 제어시스템에 대한 요구사항 분석으로부터 기능분석/할당, 아키텍처 설계, 인터페이스 설계 등은 SE 전산지원도구인 RDD-100을 이용하였고, 최종 제어시스템의 모델링과 시뮬레이션을 위해서 Matlab의 Simulink를 활용하여 최적의 공연비값을 얻고자 했다.

이 제어시스템의 설계에 적용한 체계공학 엔진^[4]



[그림 2] 체계공학 엔진(Engine)

은 MIL-STD-499B에 근간을 둔 것으로 그림 2와 같다.

체계공학 엔진을 테일러링(tailoring)하여 ECU 제어 설계에 적용한 과정(process)은 다음과 같다.

- 1) 연료분사시스템의 요구사항(requirement)을 도출한 다음 추적성의 용이를 위해 구체적인 부분까지 분해한다.
- 2) 추적성을 분명하고 명확하게 하기 위해 각각의 기능들이 어느 요구사항에 의해서 정의되었는지 결정한다.
- 3) 도출된 요구사항들을 거동(behavioral)한 부분과 비거동(non-behavioral)부분으로 목록화한다.
- 4) 거동(behavioral)에 대한 요구사항들을 가지고 거동 모델링을 구축한 후 DVF(Dynamic Verification Facility)를 통하여 최종 검증한다
- 5) 검증된 기능분석 결과를 토대로 PBS(Product Breakdown Structure)를 구축한다.
- 6) 시스템 계층구조 상의 구성품 간 인터페이스를 정의한다.
- 7) 정의된 시스템 계층구조와 인터페이스를 기반으로 모델링과 시물레이션을 구현, 검증한다.
- 8) 검증된 결과를 바탕으로 전 과정을 재검토한다.

3.1 요구사항 분석

ECU 요구사항 분석은 절차의 첫 번째 목적인 요구사항을 설계로 변환시키는 것이기 때문에 체계공학 과정상에서 우선적인 초점대상이 된다. 요구사항 분석 과정을 통해 제약조건 내에서 ECU 제어시스템의 요구사항을 개발한 후, ECU 설계가 결과적으로 요구사항과 제약조건이 반영되었는지 검증한다.

요구사항 분석 단계에서 계획된 사용자 사용, 환경, 시스템 기능에 대한 요구사항을 결정하고, 식별된 시스템 요구사항을 통해 사용자 필요와 목적을 정의한다.

요구사항 분석은 식별된 기능에 대해 성능 요구사항을 최적화 하기 위해서, 그리고 조합된 해답이 사용자 요구사항을 만족할 수 있는지에 대한 검증을 하기 위해서 기능분석을 통해 반복적으로 실시한다.

ECU 제어시스템의 요구사항 분석의 목적은 다음과 같다.

- 사용자 목적과 요구사항을 개선한다.
- 초기 성능 목적을 정의하고, 성능 목적을 요구사항에 들어갈 수 있도록 개선한다.
- 해답을 제한하는 제약조건을 식별, 정의한다.
- 효과도 척도(사용자에 의해 제공)에 기반을 둔 기능 및 성능 요구사항을 정의한다.

ECU 제어시스템의 요구사항 분석을 통해 다음 항목에 대한 확실한 이해의 결과를 만들고자 하였다.

- 기능 : 시스템이 달성해야하는 목표(what)
- 성능 : 기능이 얼마만큼 잘 이루어져야 하는가 (how well)
- 인터페이스 : 시스템이 이루어져야하는 환경 조건

○ 다른 요구사항과 제약조건

요구사항 분석으로부터 도출된 이해의 폭은 ECU 기능 및 물리 설계에 대한 기초를 만들어준다. 좋은 요구사항 분석은 성공적인 설계 정의에 있어서 기반이 된다.

본 논문에서 개발코자 하는 ECU 제어시스템의 요구사항 항목은 Dennis M. Buede의 분류법^[5]에 의거하여 분류하였다.

- 1) 입출력 요구사항
- 2) 기술 및 시스템 요구사항
- 3) 시스템 품질 요구사항

상기 3가지 분류법에 따라 요구사항들을 분류하여 요약하여 보았다. 이 요구사항들을 최초 요구사항(OR : Original Requirement)이라 지칭한다. RDD-100에 사용된 연료분사시스템의 최초 요구사항의 원문은 다음과 같다.

이 최초 요구사항(OR)은 요구사항 분석과정을 통하여 신규 요구사항의 도출, 기존 요구사항의 변경, 세분화되어 진다. 세분화된 요구사항은 시스템의 요구사항을 구체적으로 분석하는데 도움을 준다.

[표 1] 입출력 요구사항

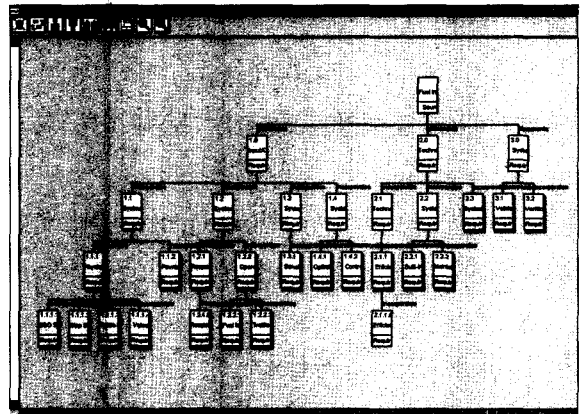
- 1.1 System Input Requirements
 - 1.1.1 Physical Signal
 - 1.1.2 Power Input
- 1.2 System Output Requirements
 - 1.2.1 Responsibility
 - 1.2.2 Operation
- 1.3 System Exterior Boundary Restraint Requirements
 - 1.3.1 Sensitivity
- 1.4 System Functional Requirement
 - 1.4.1 Optimization
 - 1.4.2 Control Function

[표 2] 기술 및 시스템 요구사항

- 2.1 Technology Requirements
 - 2.1.1 Efficiency
- 2.2 System Adequacy
 - 2.2.1 Built-in-Test
 - 2.2.2 Maintainability
- 2.3 System Life Cycle

[표 3] 시스템 품질 요구사항

- 3.1 Validation Requirements
- 3.2 Verification Requirements

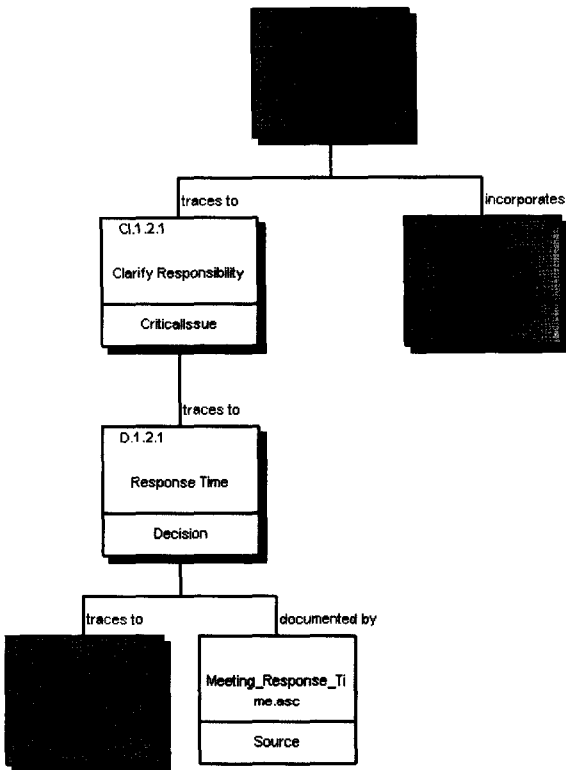


[그림 3] 요구사항 계층구조

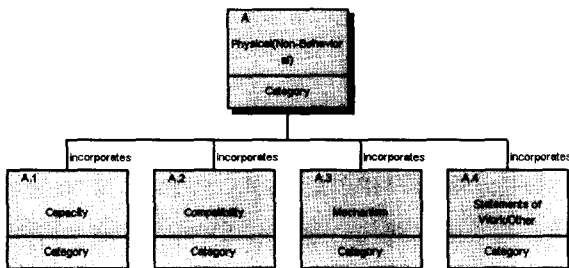
세분화 또는 신규 요구사항을 도출하는 과정에서 애매모호한 요구사항은 주요 검토사항(critical issue)으로 설정한 후, 시스템에 적합한 요구사항으로 변경하는 과정을 거친다.

상기 작업은 최초 요구사항 중 1.2.1 항목의 응답성(responsibility)이다. 응답성이 빨라야 한다는 모호한 표현이 주요 검토사항을 선정되어 정량화하는 과정을 보여주고 있다. 이 과정을 통해 최초 요구사항을 변경한다. 요구사항의 변동사항에 대한 추적성을 확보해 준다는 점에서 큰 의미가 있다.

요구사항 분석이 끝나면 요구사항을 목록화한다. 이



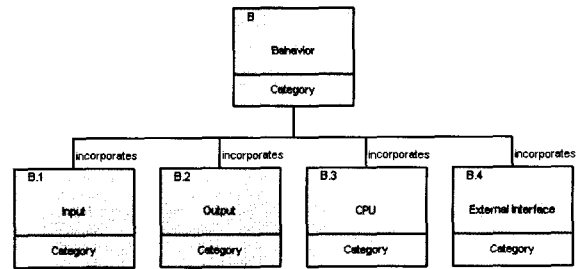
[그림 4] 주요 검토사항(Critical Issue)의 해결



[그림 5] 물리적 부분의 목록화

과정은 요구사항 분류에 의해 이루어지는 계층구조를 다른 측면에서 검토할 수 있도록 한다. 여기서는 물리적 부분과 거동적 부분으로 목록화하였다.

물리적 부분은 용량, 호환성, 메카니즘, 작업설명서 등으로 세분화하였다.



[그림 6] 거동적 부분의 목록화

거동적 부분은 입력, 출력, CPU(ECU), 외부 인터페이스 등으로 세분화하였다.

세분화된 목록에 각각의 요구사항과 연결하여 추적성을 확보한다.

3.2 기능 분석/할당

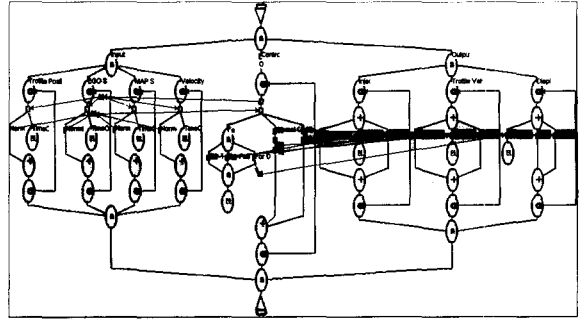
ECU 제어시스템의 기능 분석/할당의 목적은 식별된 요구사항 등을 다음에 따라오는 “조합”에 사용될 수 있는 시스템 기능으로 변환하기 위함이다. 설계자는 시스템이 달성해야 하는 것(what)과 성능(how well)을 알 필요가 있으며, 또한 설계 유통성을 제한하는 제약조건이 무엇인지에 대해서도 알아야 한다.

기능 분석/할당은 기능을 논리적인 순서로 배열, 높은 단계 기능을 낮은 단계 기능으로 분해, 높은 단계 부터 낮은 단계 기능에 성능을 할당함으로써 이루어진다. 본 논문에서는 거동분석(behavior analysis)을 사용하였다.

제품은 기능 아키텍처, 시스템 묘사(물리적 묘사보다는 기능과 성능 매개 변수를 표현)의 구현 실체이다. 기능 분석/할당은 요구사항에서 설계 조합의 산출물인 해답 묘사(solution description)까지의 추적성을 확보해 준다.

ECU 제어시스템의 기능분석/할당은 다음과 같이 진행하였다.

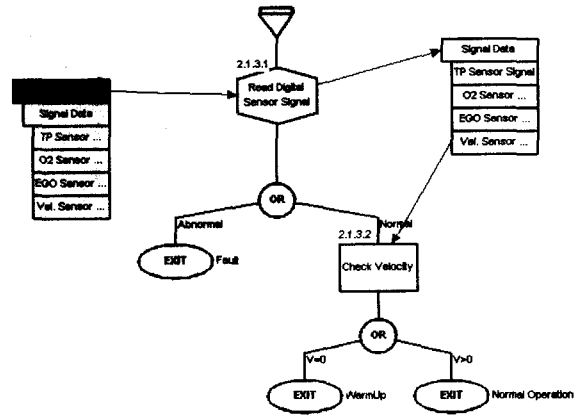
- 1) ECU 제어시스템 거동을 중심으로 거동 모델링을 구축한다.
- 2) 구축된 거동 모델링의 기능은 ECU 제어시스템 요구사항과 연결한다.
- 3) 요구사항(requirement)을 충분히 분해하여 기능(function)과 1:1의 관계로 연결되도록 한다.
- 4) 기능은 ECU 제어시스템 아키텍처의 구성품에 할당한다.



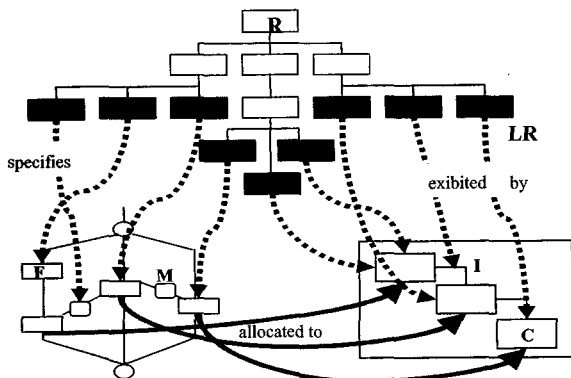
[그림 8] 거동 분석

거동 분석은 구축하고자 하는 ECU 제어시스템을 잘 이해하기 위해 사용하였다. 거동 분석을 통해 ECU 제어시스템 설계에 대한 많은 부분에서 문제점을 파악해 낼 수 있으며 쉽게 모델링 단계에서 이를 수정할 수 있다. ECU 제어시스템 완성 후 문제점을 해결할 때보다 적은 비용과 시간이 걸린다. 기능분석을 위해서는 반드시 거동분석을 하여야 한다.

여기서는 정적 거동 분석과 동적 거동 분석을 모두 수행하였다. 정적 거동 분석은 거동 모델링(behavior modeling)을 통하여 수행하였다. RDD-100에서의 거동 분석은 시나리오 분석을 통해 이루어진다. 시스템의 거동을 분석함으로써 시스템의 기능을 쉽게 파악할 수 있다.



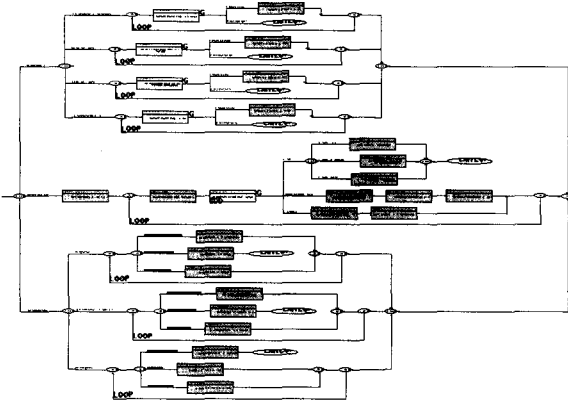
[그림 9] 데이터 모델링(Data Modeling)



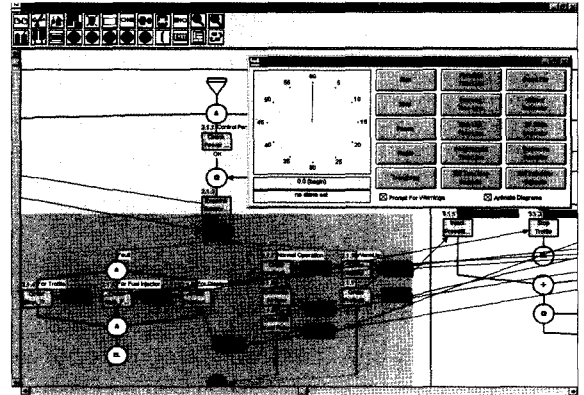
[그림 10] 요구사항과 기능분석/할당과의 관계

또한 시스템 내부의 거동 분석 후 각 프로세스 사이에 오고 가는 메시지에 대한 데이터 모델링(data modeling)을 실시하였다. 데이터 모델링을 통해 상세 설계시 필요한 데이터 형식의 정의가 여기서 이루어진다. 데이터 모델링은 기능의 내부적 거동 흐름을 보여주고, 결정사항에 대한 흐름의 알고리즘을 분석할 수 있다. 정의된 데이터의 형식은 모델링시 구체적인 기초자료로 활용하였다.

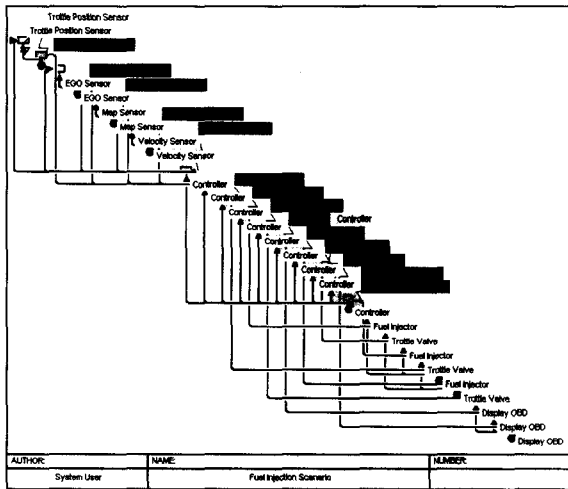
거동분석이 끝나면 FFBD(Functional Flow Block Diagram), IDEF0(Integration Definition for Function Modeling), N2 Diagram 분석의 측면에서 다시 기능분석을 실시하였다.



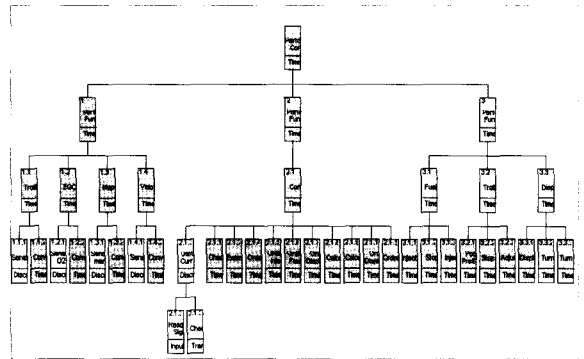
[그림 10] FFBD 분석



[그림 12] 거동 시뮬레이션 : DVF 분석



[그림 11] IDEF0 분석



[그림 13] 기능 아키텍처

다른 측면에서의 기능분석을 통해 누락되거나 오 설정한 기능을 바로 잡을 수 있다. FFBD 분석은 기능간의 순서, 상호관계를 파악할 수 있고, IDEF0는 데이터 흐름, 시스템 통제등을 한눈에 파악할 수 있다는 장점을 가지고 있다.^[6]

그림 12는 정적 모델링한 결과를 가지고 동적 거동분석인 거동 시뮬레이션을 실시한 결과이다. RDD-100에서 제공하는 DVF(Dynamic Verification Facility)를 통해 시간에 따른 기능측면, 프로세스 측

면, 메시지 링크측면 등에 대한 거동 시뮬레이션을 실시하여 논리적 오류를 바로잡았다.

정적 거동분석과 동적 거동분석을 통해 기능 아키텍처를 구현한 결과가 그림 13이다.

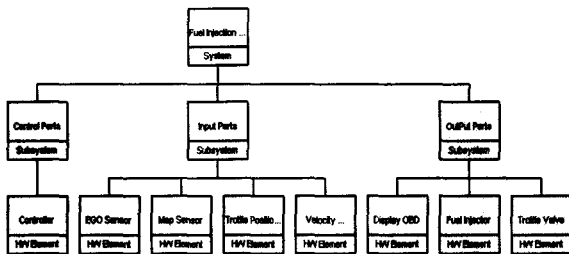
3.3 조합(아키텍처 설계)

ECU 제어시스템의 조합(아키텍처 설계)은 기능 분석과 할당을 통해 나온 기능 아키텍처를 기반으로 ECU 아키텍처 설계를 수행하는 과정이다. 아키텍처 설계는 물리적 아키텍처를 개발하는 창조적 활동이다. 기능 및 성능 요구사항을 만족하는 H/W, S/W 아키텍처를 개발한 후에 절충연구를 통해 대안 아키텍처

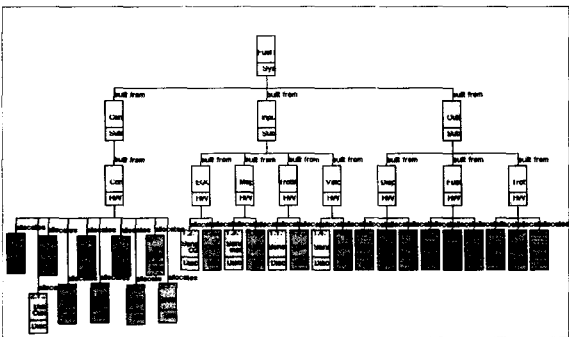
택처를 선택한다. ECU 제어시스템의 아키텍처 설계의 목적은 설계 해답을 얻기 위한 구성품의 아키텍처를 구축하는 것이다. 구축된 ECU 제어시스템의 물리적 아키텍처는 설계 정의 문서화(예, 사양서, 기본계획, WBS)의 기본이 된다. ECU 제어시스템 관련 거동분석을 통해 제품 분해구조도(PBS : Product Breakdown Structure)를 동시에 구축한다.

구축된 제품 구조도와 전 단계에서 분석된 기능들을 서로 결합하여 구성품과 기능간의 추적성을 확보한다. 회색부분이 구성품을 나타내는 제품 분해구조도이고 각 구성품과 연결된 노란색과 청색 색깔을 지니고 있는 부분이 바로 구성품의 기능들이다.

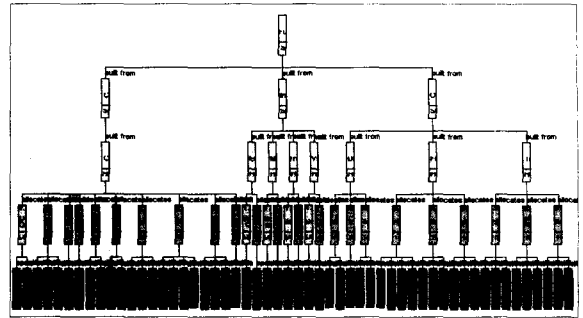
제품 분해구조도의 구성품과 기능들을 연결 한 다음, 각 기능들에 대하여 요구사항 들을 다시 연결하여, 구성품, 기능, 요구사항들의 추적성을 확보한다.



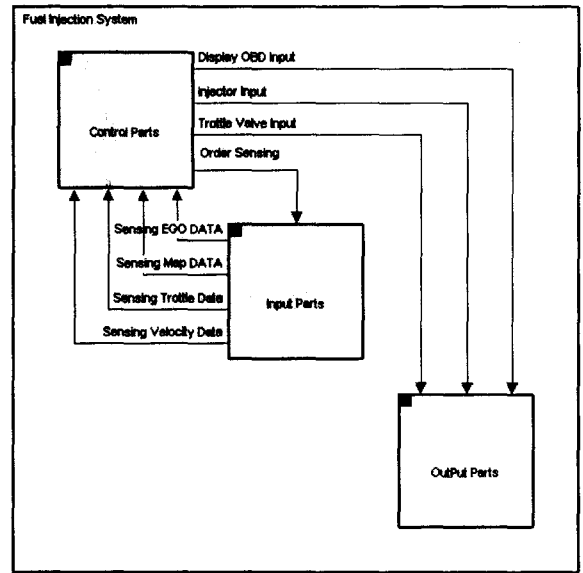
[그림 14] 제품 분해구조도



[그림 15] 제품 분해구조도와 기능간의 추적성



[그림 16] PBS, 기능, 요구사항의 추적성



[그림 17] 인터페이스 설계

그림 16에서 녹색부분이 바로 기능과 연결된 요구사항이다.

이 추적성 확보는 추후 구성품 변경 또는 업그레이드시 최초 개발시 요구되었던 조건 들을 파악할 수 있으므로 시스템 개발 목적에 맞게 설계 변경을 할 수 있다는 것을 알 수 있다.

추적성을 확보하는 단계 후 구성품의 인터페이스를 설계한다. 인터페이스 설계는 개발된 시스템내부의 구성품간 물리적 환경을 표현하고, 연료분사시스템의 운

영뿐만 아니라 실제 환경에서 작동하기 위한 외부적 요인들의 집합적 의미를 담고 있다.

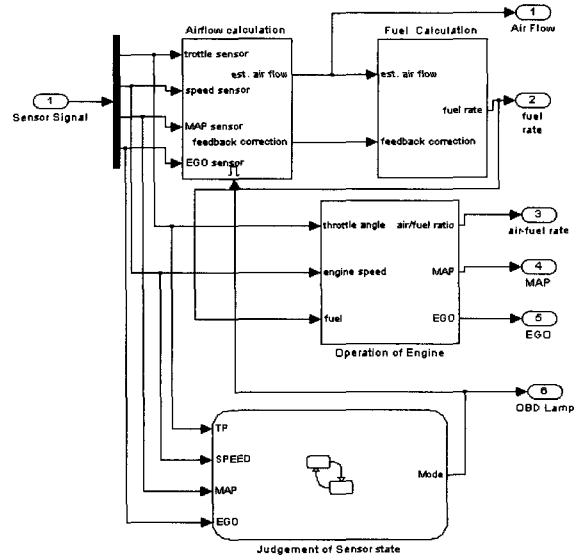
3.4 모델링과 시뮬레이션

RDD-100을 통하여 얻어진 시스템 규격을 가지고 모델링과 시뮬레이션을 행한다. 시스템 개념 형성을 통해 연료분사시스템의 주요 구성품은 입력 부분, ECU 부분, 출력 부분으로 크게 세분화되며 이들의 세부 인터페이스는 전 단계에서 규명되어 졌다. ECU 제어시스템의 내적, 외적 인터페이스가 이 단계에서 정의된다. 이를 바탕으로 Matlab의 Simulink^[7]를 사용하여 세부 모델링을 실시하였다.

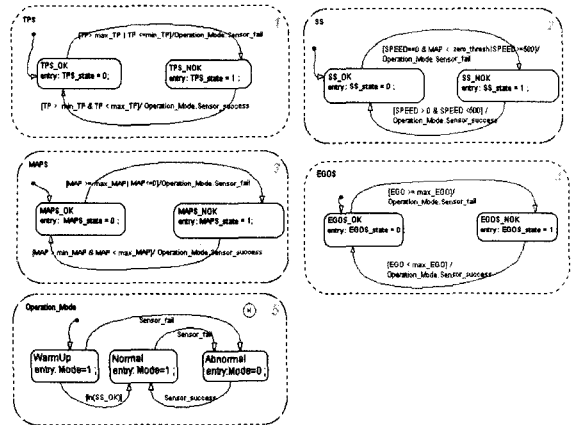
상기 그림은 상세설계의 상부 단계의 설계이다. 소프트웨어 부분의 입력부분을 2가지 이상 변화시키고, 각각의 센서 정보등을 입력시켰을 때 공연비의 결과치가 일정하게 유지되는 지 확인했다.

그림 19는 ECU 내부 연산 알고리즘과 외부 조건 판단 알고리즘을 설계한 것이다.

외부조건 판단 알고리즘은 각각에 대한 외부 입력 조건(Trottle Position, MAP Sensor, Speed Sensor,



[그림 19] ECU 설계(Level 2)

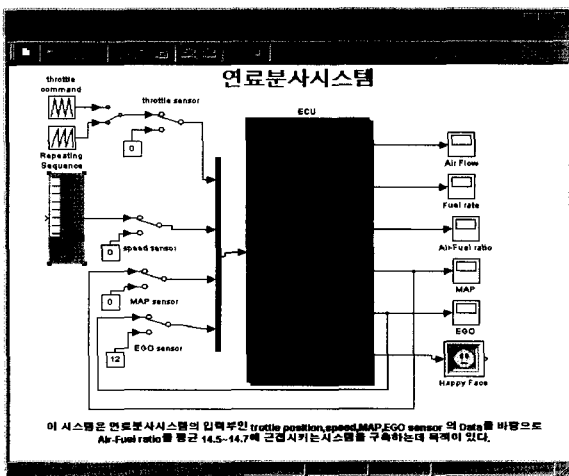


[그림 20] ECU 논리 판단구조(Level 3)

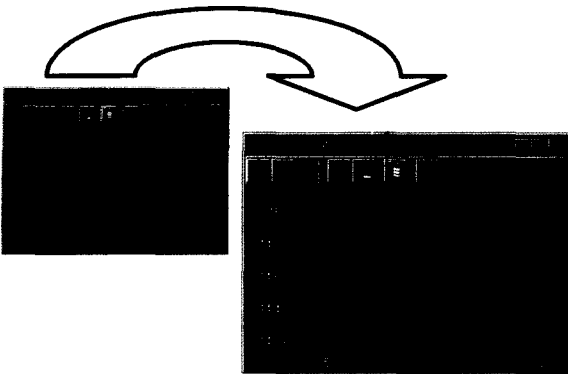
EGO Sensor)에 대한 판단을 내리고, 이를 시스템에 적합성 여부 판단(위밍업, 정상, 비정상)를 표시하도록 하였다.

그림 20은 Simulink의 Stateflow를 이용하여 ECU의 논리구조를 구축한 예이다.

그림 21은 구축된 모델링에 대하여 시뮬레이션을 실시했을 때, 가변 입력치에 대하여 최적화된 공연비



[그림 18] 연료분사시스템의 모델링(Level 1)



[그림 21] 공연비(Air-Fuel Ratio) 결과

14.6:1에 근접하여 ECU가 처리하였다는 것을 보여주는 결과 그림이다.

ECU 제어시스템을 통한 최적의 공연비 14.6:1이라는 시스템 요구사항은 검증하였으며, 이는 철저하게 시스템 요구사항, 기능, 아키텍처등의 체계공학적 접근방법을 통해 체계적으로 접근한 결과이다. 체계공학을 적용하여 시스템의 요구사항, 기능분석, 조합, M&S 등을 시스템 개발 전에 실시한 경우 수많은 설계변경 및 실패요인을 줄일 수 있다는 것을 이 논문을 통해 알 수 있다.

4. 맺은말

이 논문에서, 수많은 요구사항을 충족해야 하는 제어시스템의 설계를 하는데 있어 체계공학 접근방식을 이용하여 원하는 결과를 도출할 수 있다는 것을 보여주었다.

시스템 규격 형성을 위한 요구사항 분석, 기능분석/할당, 조합(아키텍처 설계) 및 인터페이스 정의를 통하여 모델링과 시뮬레이션을 위한 기초를 제공했다.

시스템 규격 형성 과정에서 얻어진 정보를 토대로 모델링과 시뮬레이션을 진행하여 원하는 결과치를 획득, 검증하였다. 이 과정에서 SE 전산지원도구인 RDD-100과 Matlab의 Simulink를 이용하여 제어시스템을 설계하였다.

체계공학을 적용한 제어시스템 설계를 통해 얻어진 교훈은 다음과 같이 요약할 수 있다.

첫째, 요구사항 분석을 통해 요구되는 시스템 목적을 분명하게 규명할 수 있다. 또한 애매모호하게 표현된 요구사항과 이로부터 도출된 새로운 요구사항간의 이력을 파악할 수 있었다.

둘째, 기능분석/할당은 시스템 기능에 대한 정의와 기능의 논리적 결함을 사전에 파악 가능함에 따라 최적화된 시스템 기능을 할당할 수 있었다.

셋째, 시스템 아키텍처 개발을 통해 개발코자 하는 시스템의 전반적 개념 형성이 가능케 한다.

넷째, 정의된 각 구성품간 인터페이스를 정의함으로써 인해 모델링시 시스템 구성이 용이하게 이루어진다.

다섯째, 요구사항, 기능, 아키텍처간 추적성이 가능하게 된다. 요구사항, 기능, 구성품간에 서로 어떠한 영향을 주는지 파악 가능함에 따라 이를 기초로 설계변경시 도움을 줄 수 있다.

여섯째, 모델링 & 시뮬레이션을 통해 단기간내, 적은 비용으로 최적화된 결과치를 획득, 검증할 수 있다.

일곱째, 최초 계획된 시스템 목적을 설계 결과물에 반영 가능하다.

여덟째, SE 전산지원도구는 체계공학의 시스템 개발 적용을 용이토록 해준다.

결론적으로, 제어시스템 설계에 체계공학 접근방식을 이용한 연구개발 수행이 가능하며, 이를 통해 개발

시 빈번하게 발생하는 설계 변경을 줄일 수가 있으며, 추후 성능개량 및 제어시스템 수정시 단기간내에 시스템 파악이 가능함에 따라 효율적인 업무수행에도 도움을 줄 수 있다는 것을 알 수 있었다.

앞으로 국방 무기체계의 체계개발뿐만 아니라 체계에 적용되는 제어시스템 개발에도 폭넓게 체계공학이 적용되어야 할 것이다.

참 고 문 헌

[1] INCOSE, www.incose.org.

[2] AMSC, Systems Engineering, 1974.

[3] 김현규·은정표, 자동차 전자제어(가솔린 분사장치 실무편), 중원사.

[4] James N.Martin, Systems Engineering Guidebook, 1997.

[5] Dennis M.Buede, The Engineering Design of Systems, 2000.

[6] DSMC, Systems Engineering Fundamentals, 2000.

[7] Matlab, Demos(sensor failure detection), www.mathworks.com.