# 단일 플랫폼 기반의 다기종 항공기 시뮬레이터 개발에 관한 연구

A Study of Development for Various Aircraft Simulator based on Single Platform

최성환[1]*, 강형구[1], 김성원[1], 박지혁[1], 김병수[1]
경상대학교[1]

## 초 록

본 연구에서는 다양한 급의 항공기를 단일 플랫폼에서 모의할 수 있는 시뮬레이터를 개발하기 위해 표준 비행역학 모의모델의 구조 및 인터페이스를 설계하였다. 비행역학 모의모델은 다양한 급의 항공기를 모의하기 위해 공통적으로 적용할 수 있는 비행동역학 모델과 환경모델은 항공기 기종에 관계없이 동일하게 설계하였고, 엔진, 구동기, 비행제어, 랜딩기어 모델과 같이 항공기 기종에 따라 차이가 있는 모델은 사용자가 선택하여 사용할 수 있도록 설계하였다. 또한 복잡한 구조의 소프트웨어를 사용자가 쉽게 접근하여 수정/모의할 수 있도록 비행역학 모의모델의 입력파일을 텍스트 기반인 XML 형식으로 설계하였다.

## 1. 서 론

항공기 시뮬레이터는 크게 전임무급 비행훈련 시뮬레이터(FFS, Full Flight Simulator), 비행훈련 장치(FTD, Flight Traning Device), 조종석 절차 훈련장비(CPT, Cockpit Procedure Trainner)로 구분되며, 주로 조종사 훈련 및 교육, 비행제어법칙의 검증 또는 비행성 평가와 같은 연구용 등으로 활용되고 있다. 대부분의 항공기 시뮬레이터는 높은 충실도를 요구하기 때문에 항공기 기종에 따라 하드웨어와 소프트웨어를 구성하여 운용하기 때문에 다양한 기종을 운용하기 위해서는 큰 비용이 요구된다. 또한 민항기 시뮬레이터는 외국에서 큰 비용을 들여 도입하지만, 모델 및 소스코드가 공개되지 않아 항공기 개량에 따른 항공기 모델의 수정/모의가 불가능한 상황이다. 국방 분야에서는 이와 같은 문제점을 해결하고자 가변형 비행시뮬레이터 조종실을 개발하여 유사한 급의 항공기를 모의하는 연구를 수행하였다.[1]

본 논문에서는 민간 분야의 사용자가 다양한 급의 항공기를 단일 플랫폼에서 수정/모의할 수 있는 항공기 시뮬레이터를 개발하기 위해 비행역학 모의모델의 구조와 인터페이스를 설계하기 위한 연구를 수행하였다.

## 2. 연구내용

### 2.1 비행역학 모의모델 소프트웨어의 구조

다양한 급의 항공기를 모의하기 위해 표준비행역학 모의모델의 소프트웨어 형상품목을 Fig 1과 같이 설계하였다. 표준비행역학 모의모델의 구조는 국내 시뮬레이터 소프트웨어 체계와 유사하게 설계하였으며, 비행동역학, 추진계통, 비행제어계통, 환경모델, 랜딩기어로 구성되어 있다. 다양한 급의 항공기를 모의하기 위해서 공통적으로 적용할 수 있는 비행동역학 모델과 환경모델은 기종에 관계없이 동일하게 설계하였으며, 항공기 급에 따라 차이가 크게 나타나는 엔진모델은 피스톤 엔진과 터보제트 엔진 모델을 설계하였고, 구동기는 가역/비가역 조종방식의 항공기를 모두 모의할 수 있도록 구동기 동역학 모델과 힌지모멘트 모델을 설계하였다. 비행제어 모델은 항공기 기종에 따라 구성의 차이가 있기 때문에 제어기를 구성하는 요소를 함수형태의 유틸리티로 제공하여 사용자가 쉽게 구현할 수 있도록 설계하였다. 랜딩기어 모델은 강체모델과 완충형 모델로 나누어 구성하여 항공기 기종에 따라 선택할 수 있도록 설계하였다.

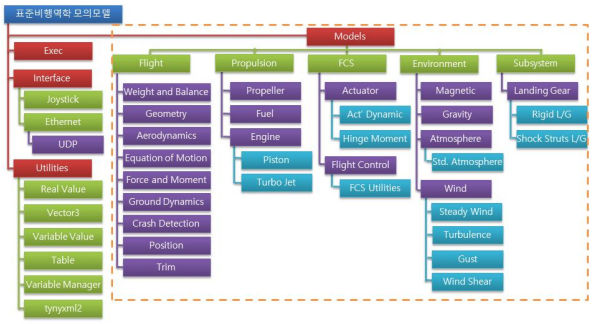


Fig 1. 표준비행역학 모의모델 소프트웨어 CSCI

2.1.1 비행동역학 모델

비행동역학 모델은 항공기의 6자유도 운동을 모사하기 위해 국내 시뮬레이터 소프트웨어 체계와 유사하게 중량 및 평형(Weight and Balance), 형상(Geometry), 공기역학(Aerodynamics), 운동방정식(Equation of Motion), 힘과 모멘트(Force and Moment), 지면반력(Ground Reaction), 항공기위치(Position), 트림(Trim) 소프트웨어 구성품(CSC, Computer Software Unit)으로 설계하였다.[2,3]

2.1.2 추진계통 모델

추진모델은 항공기의 추력 성능을 모사하기 위해 엔진(Engine), 연료(Fuel), 프로펠러(Propeller) CSC로 설계하였으며 항공기 기종에 따라 엔진의 종류를 선정하고 이에 따라 관련 데이터를 불러와 추력성능을 계산하는 과정은 Fig 2와 같은 순서에 따라 수행된다.

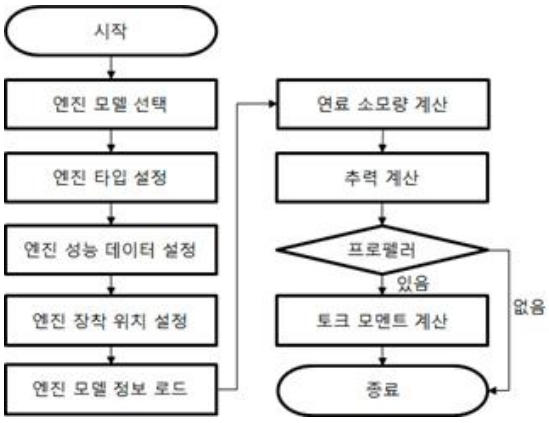


Fig 2. 추력성능 계산 소프트웨어 흐름도

2.1.2 비행제어계통 모델

비행제어계통 모델은 조종간 입력으로부터 조종면의 타각을 모사하기 위해 구동기(Actuator)와 비행제어(Flight Control) CSC로 설계하였다.

구동기 모델은 가역 비행조종방식을 갖는 기종의 경우 조종력 재현 시스템(CLS, Control Loading System)에서 조종면 타각에의 한 조종간 반력을 모사할 수 있도록 힌지모멘트를 계산하며[4] 비가역 비행조종방식을 갖는 기종은 유압 또는 전기 구동기의 동역학적 특성 모사할 수 있도록 1차 전달함수 형태로 모델링 하였다.

비행제어 모델은 Fig 3과 같이 종축, 횡축, 방향축 그리고 속도제어로 나누어 구성하였으며, 제어법칙 알고리즘은 PID 제어기법을 적용하여 설계하였다. 또한 항공기 기종에 따라 자동비행 및 조종 안정성 증대 장치(S/CAS, Stability and Control Augmentation System)의 적용여부가 다르므로, 스위치 로직을 이용하여 기종에 따라 자동비행 또는 S/CAS의 적용 여부를 결정할 수 있도록 설계하였다.

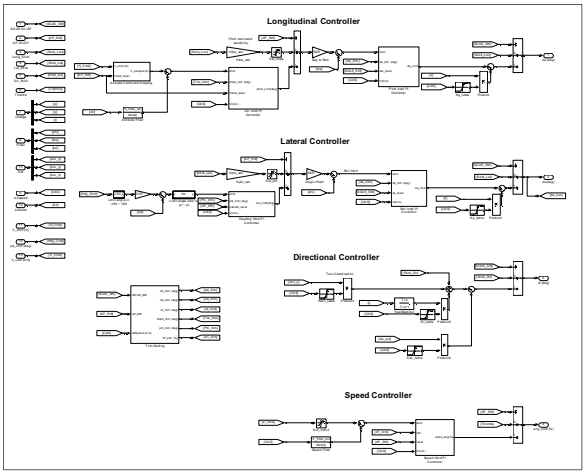


Fig 3. 제어법칙 알고리즘의 기본구조 블록다이어그램

2.1.2 환경모델

환경모델은 비행 중, 항공기 주변의 대기, 자기, 중력이 항공기에 미치는 영향을 모의하는 모델로서 대기(Atmosphere), 바람(Wind), 자기(Magnetic), 중력(Gravity) CSC로 설계하였다. 대기 모델은 65,000ft 까지 1000ft 단위로 고도에 대한 온도, 압력, 밀도, 음속 값을 테이블로 구

성하였으며, 40,000ft 까지는 국제민간항공기구(ICAO, International Civil Aviation Organization)에서 제공하는 표준대기 모델을 활용하여 설계하였으며, 40,000ft부터 65,000ft 까지는 미국항공우주국(NASA, National Aeronautics and Space Administration)에서 제공하는 고고도 대기모델(MSIS-E-90)을 활용하여 설계하였다.

바람모델은 정상풍(Steady Wind), 전단풍(Wind Shear), 난류(Gust) 모델을 MIL-STD-1797A를 활용하여 설계하였으며, 지자기 모델은 미국 국립지리정보국(NGA, National Geospatial-Intelligence Agency)에서 제공하는 WMM(World Magnetic Model)의 2015년 데이터를 이용하여 자기편차를 계산하도록 설계하였다.

## 2.2 비행역학 모의모델 소프트웨어 인터페이스

다양한 기종의 항공기에 대한 데이터를 단일 플랫폼 기반의 비행역학 모의모델에서 사용하기 위해서 입력파일의 표준화와 데이터베이스의 구조화를 수행하였다.

입력파일은 항공기 기종에 따라 시뮬레이터 소프트웨어의 수정/모의가 편리하도록 사용자 또는 개발자의 접근성과 확장성이 확보되어야 하기 때문에 XML(Extensible Markup Language)형식으로 설계하였다. XML 형식은 사람과 기계가 모두 해독할 수 있는 텍스트 기반의 부호화 규칙을 의미하며, 텍스트 기반이기 때문에 작성 내용을 사람이 직관적으로 판단할 수 있으며 데이터 표현이 구조화 되어 있기 때문에 항공기에 공통적으로 필요한 데이터 및 항공기 기종에 따라 추가되는 데이터의 표준화가 용이하다.

Fig 4는 입력파일에서 공력모델에 해당하는 부분으로서 <aerodynamic>과 </aerodynamic>사이에 있는 내용이 공력 모델과 관계된 내용임을 사용자가 직관적으로 판단할 수 있으며, 데이터 변수명 설정 및 차수 등에 대한 설정을 쉽게 수정할 수 있도록 설계되어 있다. Fig 5는 다양한 항공기 기종의 형상, 공력, 추력데이터 및 환경모델의 데이터베이스를 구조화하기 위해 설정한 Table의 차수에 따른 데이터 작성 표준양식이다.

```
<aerodynamics>
  <axis name="[LIFT || Drag || SIDE || ROLL || PITCH || YAW]">

    <coefficient name="[coefficient name]">
      <description>[coefficient description]</description>
      <product>
        <variable>[variable name]</variable>
        <value>[value]</value>
      </product>
    </coefficient>

    <coefficient name="[coefficient name]">
      <description>[coefficient description]</description>
      <product>
        <table name="[table name]" dimension="[1D || 2D || 3D]">
          <axisVariable axis="[row || column || sheet]">[variable name]</axisVariable>
          <tableData>
            [table data]
          </tableData>
        </table>
      </product>
    </coefficient>
    ····· other coefficient information
  </axis>
  ····· other axis information
</aerodynamics>
```

Fig 4. Aerodynamics 입력파일의 구조

[1D Table 양식]

```
<table name="[table name]" dimension="1D">
  <axisVariable>[variable name]</axisVariable>
  <tableData>
    row1   value1
    row2   value2
    row3   value3
    ···    ···
    rown   valuen
  </tableData>
</table>
```

[2D Table 양식]

```
<table name="[table name]" dimension="2D">
  <axisVariable axis="row">[variable name]</axisVariable>
  <axisVariable axis="column">[variable name]</axisVariable>
  <tableData>
            column_1  column_2  column3   ··· column_m
    row_1   value_11  value_12  value_13  ··· value_1m
    row_2   value_21  value_22  value_23  ··· value_2m
    row_3   value_31  value_32  value_33  ··· value_3m
    ···     ···       ···       ···       ···
    row_n   value_n1  value_n2  value_n3  ··· value_nm
  </tableData>
</table>
```

[3D Table 양식]

```
<table name="[table name]" dimension="3D">
  <axisVariable axis="row">[variable name]</axisVariable>
  <axisVariable axis="column">[variable name]</axisVariable>
  <axisVariable axis="sheet">[variable name]</axisVariable>
  <tableData breakpoint="[sheet_1]">
            column_1  column_2  column3   ··· column_m
    row_1   value_11  value_12  value_13  ··· value_1m
    row_2   value_21  value_22  value_23  ··· value_2m
    ···     ···       ···       ···       ···
    row_n   value_n1  value_n2  value_n3  ··· value_nm
  </tableData>
  <tableData breakpoint="[sheet_2]">
            column_1  column_2  column3   ··· column_m
    row_1   value_11  value_12  value_13  ··· value_1m
    row_2   value_21  value_22  value_23  ··· value_2m
    ···     ···       ···       ···       ···
    row_n   value_n1  value_n2  value_n3  ··· value_nm
  </tableData>
···
  <tableData breakpoint="[sheet_x]">
            column_1  column_2  column3   ··· column_m
    row_1   value_11  value_12  value_13  ··· value_1m
    row_2   value_21  value_22  value_23  ··· value_2m
    ···     ···       ···       ···       ···
    row_n   value_n1  value_n2  value_n3  ··· value_nm
  </tableData>
</table>
```

Fig 5. Table 차수에 따른 데이터 작성 표준양식

## 2.3 비행역학 모의모델 소프트웨어 설계문서

표준비행역학 모의모델 소프트웨어를 개발하기 전에 소프트웨어 공학에서 정의하는 소프트웨어 개발 단계에 맞춰 설계 문서를 작성하였다.[5]

본 연구에서 진행된 과정은 요구도 분석과 설계로서, 작성되어야 하는 설계문서는 Fig 6의 소프트웨어 요구사양서(SRS, Software Requirement Specification)와 소프트웨어 설계명세서(SDD, Software Design Document)이다.

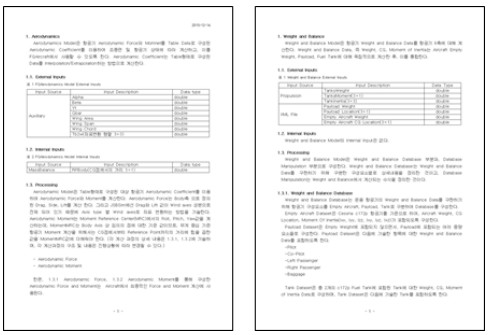

Fig 6. 소프트웨어 요구사양서

## 3. 향후 계획

본 연구를 통해 설계한 비행역학 모의모델을 바탕으로 C언어 기반의 실시간 시뮬레이션 소프트웨어의 코딩을 수행할 계획이다. 또한 비행역학 모의모델 소프트웨어의 검증을 위해 미국 연방항공청(FAA, Federal Aviation Administration)의 항공기 시뮬레이터 검증기준인 AC120-40C와 국토교통부에서 고시한 모의비행훈련장치 지정기준 및 검사요령을 활용하여 비행성능 검증기준에 해당하는 항목을 도출하여 지정검사교범(QTG, Qualification Test Guide)를 작성하고 절차에 맞게 수행할 계획이다. QTG를 수행하기 위해 경량항공기급 기종인 CTLS의 비행성능데이터 및 실제 비행시험을 통한 데이터 확보를 수행할 계획이며 중형여객기급 기종을 추가로 선정하여 QTG를 수행할 계획이다.

## 후 기

본 연구는 차세대기계항공 창의인재양성사업단의 후원과 산업통상자원부 및 한국산업기술평가관리원의 미래산업선도기술개발사업(신시장창출)의 일환으로 수행하였으며 지원에 감사드립니다. [10053273, 경험지식의 확장성을 갖는 몰입형 가상훈련시스템 개발]

## 참고문헌

[1] 양지연, 2011, 가변형 비행시뮬레이터 조종실 개발, 한국항공우주학회지, 39(7), pp. 660~665.
[2] 박건명, 박춘배, 1991.11, 연구용 비행시뮬레이터의 구성에 관한 연구, 한국항공우주학회 학술발표회 논문집, pp.239~245.
[3] 이상혁, V. Povstyanoy, 노갑성, 박주원, 김인규, 고상휘, 1999.4, 항공기 개발용 시뮬레이터 소프트웨어 설계에 관한 연구, 한국항공우주학회지 pp.100~104.
[4] 이찬석, 김병수, 2004, 비행 시뮬레이터용 조종력 재현 장치 설계 및 시험연구, 한국항공우주학회지, 32(4), pp.95~100.
[5] 김치수, 2015, 쉽게 배우는 소프트웨어 공학, 한빛아카데미, pp.8~30.