論文 J. of The Korean Society for Aeronautical and Space Sciences 46(11), 961-968(2018)
DOI:https://doi.org/10.5139/JKSAS.2018.46.11.961
ISSN 1225-1348(print), 2287-6871(online)

항공기 시현계통 소프트웨어의 비행시험을 위한 분석도구

이용래*, 최으뜸**, 전용기***

An Analysis Tool for Flight Test of Airborne Display Software

Yong-Rae Lee*, Eu-Teum Choi** and Yong-Kee Jun***
Korea Aerospace Industries, LTD.*
Department of Informatics, Gyeongsang National University**,***
Department of Aerospace and Software Engineering, Gyeongsang National University***

ABSTRACT

Airborne display systems provide pilots with a variety of information needed to operate aircraft. Software faults in the display system can seriously affect the operation of the aircraft, because it can provide inaccurate information to the pilot. Therefore, the software faults are identified and eliminated through ground testing and flight testing. This paper presents an analysis tool called FDR (flight data replay) for flight test of airborne display software. This tool works in real time with the mission computer of aircraft. Also, the tool reproduces the functional error conditions that appear in the display systems by applying flight test data to the display software.

초 록

항공기 시현계통은 항공기 운영에 필요한 다양한 정보를 통합하여 조종사에게 시현한다. 시현계통에서 소프트웨어 결함이 발생되면 정확하지 않은 정보가 조종사에게 시현될 수 있기 때문에 항공기의 운영에 심각한 영향을 미칠 수 있다. 따라서 시현계통의 개발 중에 지상시험과 비행시험을 통해 발생 가능한 소프트웨어 결함을 식별하고 제거하려고 노력하고 있다. 본 연구는 비행시험 자료를 이용하여 시현계통의 소프트웨어 결함 분석을 지원하는 도구인 FDR(Flight Data Replay)을 제시한다. 이 도구는 항공기 임무컴퓨터와 연동되어 실시간으로 동작한다. 그리고 시현소프트웨어에 비행시험 자료를 적용하여 시현장치에 나타나는 기능적인 오류상황을 재현한다.

Key Words : Airborne Display Systems(항공기 시현계통), Software Faults(소프트웨어 결함), Flight Test(비행시험)

Ⅰ. 서 론

항공전자[1,2] 계통 중의 하나인 항공기 시현계통은 조종사에게 비행에 중요한 정보들을 제공하며 MFD(Multi Function Display)와 HUD(Head Up Display)가 대표적이다. MFD는 항공기 운용 중에 항법, 비행, 생존, 전자지도, 그리고 정비관련 정보를 입력받아 시현하는 장치이다[3]. HUD

† Received : August 7, 2018 Revised : October 18, 2018 Accepted : October 23, 2018
*** Corresponding author, E-mail : jun@gnu.ac.kr

는 조종사가 항공기 전방 캐노피의 바깥 상황을 보는 동시에 비행 정보를 식별할 수 있는 투명한 시현 장치이다[4].

항공기 소프트웨어는 항공기 구성품에 탑재되는 임베디드 소프트웨어이며 기술이 발전함에 따라 항공기에서 소프트웨어를 이용한 기능 구현이 증가하고 있다. 이러한 항공기 소프트웨어 규모의 증가는 소프트웨어 결함의 증가로 이어진다. 소프트웨어 규모가 2K 라인인 경우 1K 라인 당 0~25 결함이 존재하는데 반해 512K 라인 경우 1K 라인 당 4~1,000으로 결함이 증가한다[5-7].

소프트웨어 결함은 항공기 개발과정 중에 시험을 수행하여 식별되고 제거되고 있다[8,9]. 하지만 시험을 통해 소프트웨어 결함을 제거하려는 노력에도 불구하고 실 운용 중에 소프트웨어 결함에 의한 오류는 발생하고 있다. 따라서 실 운용 전인 비행시험 단계에서 비행데이터[10,11]를 통해 오류를 재현하여 소프트웨어 결함을 분석하고 제거할 필요가 있다.

비행데이터를 이용한 소프트웨어 결함 분석도구는 제어법칙 고장탐구 도구[12]와 F-16 전투기의 AACAS (Auto Air Collision Avoidance System) 시뮬레이션 도구[13]가 있다. 제어법칙 고장탐구 도구는 비행제어 계통에서만 사용할 수 있으며 AACAS 시뮬레이션 도구는 비행데이터를 통해 충돌알고리즘의 검증에만 사용할 수 있기 때문에 시현계통의 소프트웨어 결함을 분석하기 위해 사용할 수 없다. 그리고 비행데이터를 이용하여 항공기 시현계통의 소프트웨어 결함을 분석하는 연구 및 도구는 없다.

본 연구는 항공기 시현계통의 비행시험을 위한 분석도구인 FDR(Flight Data Replay)를 제시한다. FDR은 전투기의 시현소프트웨어가 탑재된 미션컴퓨터의 OFP(Operational Flight Program)와 시현 장치인 MFD, HUD에서 발생하는 오류 상황을 재현하는 도구이다. FDR은 MC(Mission Computer) Rack, SIC(Simulation Interface Console), FDR 워크스테이션으로 구성된다.

MC Rack은 시현 OFP가 탑재된 미션컴퓨터, MFD, HUD 등의 하드웨어가 장착되는 랙 장비이다. SIC는 비행데이터를 각 항공전자 장비들에게 전달하고 미션컴퓨터로 처리된 데이터를 실시간으로 전송한다. FDR 워크스테이션은 사용자에게 GUI(Graphic User Interface)를 제공한다. GUI를 통해 사용자는 비행데이터를 선택하고 소프트웨어 결함을 분석할 수 있다.

FDR의 기능 및 성능을 시험하기 위해 시스템 시험을 수행한다. 시스템 시험은 사용자 인터페이스, 비행데이터의 분석 및 변환, 비행데이터 선별 및 가공, 비행데이터의 재생, 시현장치의 운용환경, 품질 사양, 성능으로 정의된 시스템 사양을 기준으로 수행한다.

또한, 비행 중에 발생한 오류상황의 재현 성능을 평가하기 위하여 실제 비행데이터를 통해 오류상황을 재현하는지 시험하였다. 시험 내용은 공대지 무장 운용 문제, 공대공 무장 시현 문제, CEM(Cruise Engery Management) 알고리즘 문제, MFD 화면 깜빡임 문제이다. 이 중 실제 지원 사례도 포함되어있다.

II. 연구 배경

항공기 시현계통은 개발 중에 다양한 시험을 통해 소프트웨어 결함을 제거하기 위한 노력에도 불구하고 실 운용 중에 결함이 발생할 수 있다. 본 장은 항공기 시현계통의 소프트웨어에서 발생하는 오류를 제거하는 도구에 대해서 설명한다.

2.1 항공기 시현계통

항공기 시현계통은 항공전자[1,2] 계통에 속하며 비행에 중요한 정보들을 조종사에게 제공하며 MFD, HUD 등이 대표적이다. MFD는 항공기 운용 중에 항법, 비행, 생존, 전자지도, 그리고 정비 관련 정보를 입력받아 조종사에게 시현하는 장치이다[3]. HUD는 조종사가 항공기 전방 캐노피의 바깥 상황을 보는 동시에 비행 정보를 식별할 수 있는 투명한 시현 장치이다[4].

Figure 1은 F-16 전투기의 시현체계를 나타낸다. Fig. 1과 같이 시현계통은 미션컴퓨터의 응용 프로그램인 OFP(Operational Flight Program)가 수많은 항공전자 장비로부터 수집된 정보를 처리하여 조종사에게 필요한 정보를 텍스트와 심벌 등으로 표현한다. 이러한 시현계통은 조종사에게 비행에 중요한 정보들을 제공하기 때문에 고신뢰성이 보장되어야 한다[3,4,14,15].

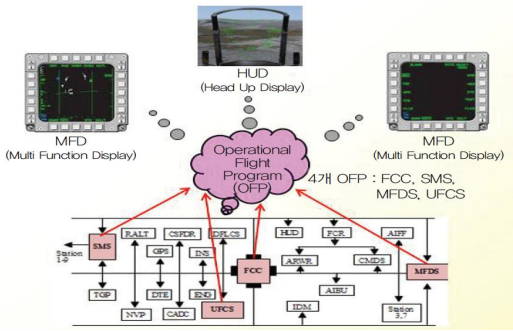

Fig. 1. Display Subsystem in F-16 Fighter

2.2 항공기 소프트웨어

항공기 소프트웨어는 항공기 구성품에 탑재되는 임베디드 소프트웨어로써 실시간 운영체제, 미들웨어, 응용소프트웨어, 사용자서비스 등으로 구분된다. 항공기에서 사용되는 실시간 운영체제는 소프트웨어를 운영하기 위한 환경을 제공하며 DO-178 인증을 받은 제품이어야 한다[6,7]. 대표적인 상용제품은 VxWork 653, LynxOS-178, INTEGRITY-178 등이 있다. 미들웨어는 항공기의 표준기능에 대한 지원, 항공 통신 및 응용소프트웨어와 연동을 위한 기능을 담당한다. 응용소프트웨어는 중앙통제, 시현, 자료입력, 무장관리 등의 소프트웨어를 말하며 OFP라고도 불린다. 마지막으로 전투정보/조종사 임무지원, 조종사 훈련체계 등의 사용자 서비스가 있다.

기술이 발전함에 따라 항공기에서 소프트웨어를 이용하여 기능을 구현하는 비중이 증가하고 있다. 미국 DoD(Department of Defense) 자료에 의하면 Fig. 2와 같이 항공기가 수행해야하는 기능 중에서 소프트웨어에 할당되는 비중이 F-4는 8%, F-14는 16%, F-35는 90%로 점차 증가함을 알 수 있다. 그리고 미국 VDC(Venture Development Corporation)의 2017년 자료에 의하면 군사 및 항공분야에서 소프트웨어 개발비용이 전체 개발비용의 51.4%로 5년 전의 39.7%보다 10% 이상 증가함을 알 수 있다[5,16].

항공기에서 소프트웨어의 비중이 높아짐에 따라 규모와 복잡도가 증가하여 시스템 내에 존재하는 잠재적인 오류도 증가하고 있다. 이러한 오류를 제거하기 위해 항공기 소프트웨어 개발과정에서 많은 시간을 소비하며 시험(Testing)을 수행한다. 하지만 개발과정에서 오류를 식별하고 제거하려는 노력에도 불구하고 비행시험이나 실운용 중에 소프트웨어 오류는 발생하고 있다. 따라서 개발과정 중에서 제거되지 못한 결함을 식별하고 제거하기 위해 실 운용 전인 비행시험단계에서 비행시험 결과를 이용하여 소프트웨어 결함을 분석할 수 있어야 한다[5-8].

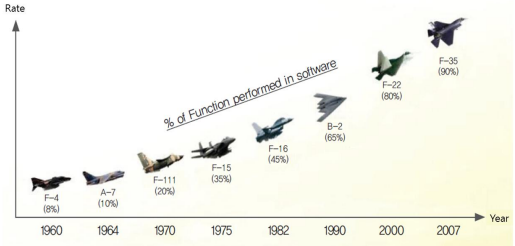

Fig. 2. Software Rates in Fighters

2.3 비행시험에서 소프트웨어 결함분석

비행데이터[12,13]는 가상 데이터와 달리 센서에 의해 항공기의 비행정보를 기록한 자료이다. 비행데이터는 항공기의 고유 특성이 기록되어 있기 때문에 항공기 분석을 위해 다양하게 사용하고 있으며 항공기 개발과정 중에서는 비행시험단계부터 획득이 가능하다. 비행데이터는 DTE (Data Transfer Equipment), 계측장비, 비행데이터 저장장치와 같은 항공전자 장비들을 통해 획득 할 수 있다[10,11]. 최근의 항공전자 장비들은 미션 디브리핑 (Mission Debriefing)과 사고조사 등의 용도로 비행데이터를 수집하고 있다. 항공기의 비행 중에 오류상황이 발생하면 오류와 관련된 정보들이 모두 비행데이터에 저장될 것이다. 따라서 비행데이터의 분석으로 오류의 원인이 되는 결함을 찾을 수 있다.

비행 중에 발생한 오류는 발생 원인이 복잡한 경우가 많아 재현이 어렵다. 따라서 항공기 개발과정에서 수행되는 시험과 동일한 절차로는 대응할 수 없다. 그리고 비행시험 중에 식별된 오류를 분석하여 결함을 제거하지 못하면 항공기의 운용 중에 발생되어 인명과 재산피해를 발생시키는 심각한 사고를 초래할 수도 있으므로 비행데이터를 이용하여 항공기 소프트웨어의 결함을 분석하는 것은 중요하다.

비행데이터를 이용한 소프트웨어 결함을 분석하기 위한 도구는 제어법칙 고장탐구 도구[12]와 F-16 전투기의 AACAS 알고리즘 검증도구[13]가 있다. 제어법칙 고장탐구 도구는 비행제어 컴퓨터의 비행제어 법칙의 고장을 탐구한다. 비행제어 소프트웨어의 개발언어는 어셈블리 언어로 대부분 구현하기 때문에 다른 계통과의 개발환경이 다르다. 그리고 비행제어 컴퓨터에 연결된 장비는 비행제어에 필요한 정보만을 다루기 때문에 다른 계통의 결함을 분석하기에 적합하지 않다. 따라서 제어법칙 고장탐구 도구는 비행제어 법칙의 고장탐구에만 사용할 수 있다.

AACAS 시뮬레이션 도구는 훈련용 장비인 ACMI(Autonomous Air Combat Maneuvering Instrumentation) Pod의 AACAS 시뮬레이션의 알고리즘을 검증 및 결함 탐지에 사용한다. 이 도구는 비행데이터를 통해 충돌알고리즘의 검증에만 사용할 수 있다. 따라서 이 도구들로 시현계통의 소프트웨어 결함을 분석하기 위해 사용할 수 없다. 또한 비행데이터를 통해 시현계통 소프트웨어의 결함을 분석하기 위한 연구는 없기 때문에 시현계통 소프트웨어의 결함 분석을 위한 분석도구의 개발이 필요하다.

III. 도구 개발

본 장은 비행시험에서 항공기 시현계통 소프트웨어의 결함을 분석하는 FDR의 요구사항을 정의한다. 그리고 요구사항을 바탕으로 도출된 도구에 대한 사양, 설계, 그리고 구현을 설명한다.

3.1 요구사항

비행데이터를 이용한 항공기 소프트웨어의 결함 분석에 필요한 요구사항은 다음과 같다.

- 분석도구의 사용자 인터페이스
- 비행시험 시 녹화된 비행데이터를 처리
- 사용자가 비행데이터를 선별하여 도구를 실행
- 비행 상황과 동일한 데이터를 재현
- 다른 항공전자장비로 시뮬레이션 및 스케줄링

3.2 시스템 사양

FDR은 전투기의 시현소프트웨어가 탑재된 미션컴퓨터의 OFP와 시현 장치인 MFD, HUD로 시현 할 때 발생하는 오류상황을 재현을 통한 오류 분석도구이다. FDR은 실제 비행환경에서 운용될 때와 동일한 시뮬레이션 환경을 제공할 수 있다. 본 절은 분석도구의 요구사항에 대해 시스템이 제공하는 기능 및 성능 사양을 설명한다.

3.2.1 사용자 인터페이스 (GUI)

사용자 인터페이스는 오류 상황을 재현하기 위해 다음과 같은 기능을 제공한다.

- File Open: 녹화된 비행데이터를 선택
- Signal Select: 비행데이터의 분석된 신호 목록
- Play List: 재현을 위해 선택된 신호 시현 창
- Play Control: 신호들을 재생 제어
- Operation Log: 상태, 오류, 경고 정보를 시현

3.2.2 비행데이터의 분석 및 변환

FDR은 DTRS(Data Transfer and Recording System)에 저장된 비행데이터를 사용한다. DTRS는 비행경로, 영상, 그리고 비행데이터를 기록하는 장치이다[17]. DTRS는 DTRU(Data Transfer and Recording Unit)와 RMM(Removable Memory Module)로 구성되며 각종 데이터를 임무컴퓨터에 저장한다. 저장된 데이터는 지상에서의 임무분석에 활용된다. GUI의 File Open 기능으로 DTRS에 저장된 비행데이터를 시간 순으로 획득할 수 있다.

3.2.3 비행데이터의 선별 및 가공

항공전자장비의 ICD(Interface Control Document)는 시스템 간 데이터를 주고받기 위한 데이터 정의와 프로토콜을 기술한 문서이다[18]. 비행데이터의 선별 및 가공을 위해 ICD를 통한 비행데이터를 분석이 수행된다. 사용자는 분석된 비행데이터의 신호 명을 Signal Select를 통해 선택할 수 있다. 선택된 신호들은 Play List에 등록되며 이 신호들은 Play Control에서 실행에 적합하도록 다시 한 번 변환 및 가공된다.

3.2.4 비행데이터의 재생

사용자가 Play Control를 통해 재생을 실행하면 시현 OFP는 녹화된 상황과 동일한 데이터를 수신하고 MFD와 HUD에 시현한다. 재생 시 비행데이터는 통신, 항법, 센서, 무장 등의 항공전자 장비로 시뮬레이션 된다. 그리고 시현 OFP가 탑재된 장치인 미션컴퓨터는 실제 장비들 간 실시간 통신하는 것과 동일한 데이터 주기로 실시간 비행데이터를 수신한다. 미션컴퓨터는 MFD와 HUD에 비행 정보를 시현하게 된다.

3.2.5 시현장치의 운용환경

전투기의 시현장치인 MFD와 HUD는 항공기 조종석에 배치되어 다양한 정보를 시현하여 조종사의 임무를 돕는다. 이러한 장치와 시현소프트웨어가 탑재되는 미션컴퓨터는 상호간의 물리적 인터페이스와 전기적 인터페이스를 포함하는 구동 환경 지원한다.

3.2.6 성능

분석도구에서 요구되는 MIL-STD-1553B 네트워크의 성능은 다음과 같다.

- 전송 주기: 20ms 이내
- 동시재생 신호개수: 1000개

항공기 시스템에서 가장 빠른 데이터 전송 주기는 20ms이다. 따라서 전송주기는 20ms 이내를 만족해야 한다. 그리고 알고리즘 테스트 시 필요한 시험 인자의 평균값인 1000을 동시재생 신호개수로 설정한다.

3.3 도구 설계

FDR의 전체 시스템 구조는 Fig. 3과 같다. FDR은 MC Rack, SIC, 그리고 FDR 워크스테이션으로 구성된다.

3.3.1 MC Rack

MC Rack은 시현 OFP가 탑재된 미션컴퓨터와 시현장치인 MFD, HUD 등의 하드웨어가 장착되는 랙 장비이다. 시현장치가 하드웨어적인 문제로 화면을 띄우지 못하는 경우를 대비하여 음성 및 영상 녹화 장치인 AVSR(Airborne Video Solid State Recorder)의 신호를 가져와 별도의 모니터로 시현할 수 있도록 설계한다.

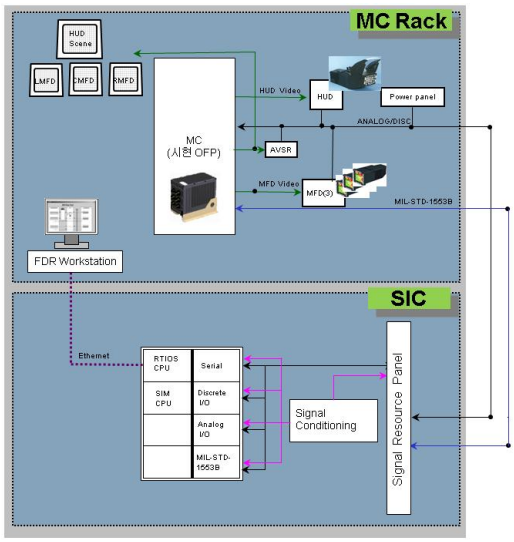

Fig. 3. The Overall Architecture of FDR

3.3.2 SIC (simulation interface console)

SIC는 FDR 소프트웨어의 실행 시 비행데이터를 실시간으로 처리하는 장비이다. 그리고 비행데이터가 각각 항공전자 장비들로 시뮬레이션 하는 기능을 가지고 있다. 시뮬레이션을 통해 처리된 비행데이터는 실시간으로 입출력 보드를 통해 출력 신호를 시현소프트웨어가 탑재된 장비인 미션컴퓨터로 전달한다.

3.3.3 FDR 워크스테이션

FDR 워크스테이션은 사용자가 항공기 소프트웨어 결함을 분석하기 위한 FDR 소프트웨어를 탑재하고 있다. 사용자는 FDR 소프트웨어를 통해 비행데이터를 선택하고 재생할 수 있다. 비행데이터를 재생하면 ADS2에서 제공하는 이더넷 통신을 통해 선택된 비행데이터를 SIC로 전송한다. 그 외에 상태에 대한 메시지를 SIC로부터 전달받아 Operation Log로 시현한다. Fig. 5는 FDR 소프트웨어의 흐름도(Flow Chart)를 보여

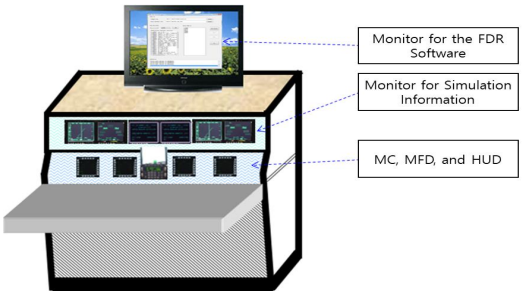

Fig. 4. The Design of MC Rack

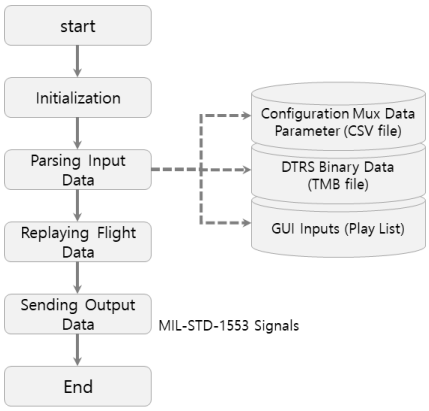

Fig. 5. The Flow Chart of FDR Software

준다. 각 항목은 클래스로 설계되고 구현된다.

사용자에 의해서 FDR 소프트웨어가 시작이 되면 신호체계와 환경설정의 초기화와 DTRS의 데이터 읽기 및 파싱을 위한 클래스를 초기화 한다. 초기화가 완료되면 입력데이터를 사용자가 정의한 설정 파라미터 데이터를 참고하여 비행데이터 파일의 구조를 분석하여 재생할 수 있는 데이터로 가공한다. 비행데이터 재생은 신뢰성 있는 통신 및 데이터 동기화를 위해 성능 요구사항보다 낮은 10ms/frame에 맞춰서 각 데이터들을 동기화 한다. 그 후 재생 목록에서 비행데이터를 선택하여 신호와 값 형태로 재생한다. 출력신호 처리는 SIC에서 재생된 신호와 값 데이터를 MIL-STD-1553B[21] 인터페이스로 출력한다. 따라서 저장된 비행데이터를 통해 각 시현체계에 재현하여 시현체계 소프트웨어에서 발생하는 오류현상을 분석할 수 있다.

3.4 도구 구현

MC Rack은 미션컴퓨터, MFD, HUD 등의 장치를 위한 물리적 운용 환경을 구성하는 랙 장비이다. MC Rack의 구현에 소요되는 시간과 비용상의 제약으로 항공전자 통합시험을 위해 개발된 SIL(System Integration Laboratory) 장비를 이용하였다.

SIC는 ADS2(Avionics Development System) 시스템을 기반으로 개발되었다. SIC의 구현을 위한 시간과 비용을 절약하기 위해 SIL에 적용된 SIC 장비를 사용하였다[19,20]. SIL의 SIC 기능 중 실장비와 연동되는 부분, FIBO(Fault Insertion & Breakout) 부분 등을 제외하여 FDR 구조를 만족하도록 변경하여 최적화 하였다.

FDR 워크스테이션은 윈도우 운영체제 환경에서 FDR 소프트웨어가 동작하도록 구축되었다.

Fig. 6. The FDR System

FDR 소프트웨어는 Python언어와 ADS2의 API를 이용하여 개발하였다. GUI는 Python GUI API 중 하나인 wxPython[22]을 사용하였다. 그리고 FDR 소프트웨어를 사용하기 위한 모니터, 마우스, 키보드 등의 장치들은 MC Rack에 연결된 별도의 탁자에 배치하였다.

FDR 시스템을 계획된 설계 수준으로 Fig. 6과 같이 구현하였다. 시스템의 주요 요소인 MC Rack, SIC, FDR 워크스테이션 간 연동을 위해서 필요한 물리적인 연결을 하였다. 그 외 FDR 시스템의 환경 구축에 에 필요한 하드웨어, 인터페이스 등은 SIL의 구성품을 활용하여 시간, 노력, 그리고 비용 등을 절약하였다.

Ⅳ. 실 험

본 장은 구현된 FDR 시스템이 정의된 시스템 사양을 만족하는지 검증한다. 그리고 도구의 재현 능력을 평가하기 위해서 총 24개의 종류로 시험을 수행하고 결과를 분석한다.

4.1 FDR 시스템 시험

FDR 시스템 시험은 3장에서 기술한 시스템 사양을 기준으로 수행하였다. 사용자 프로그램의 시험은 FDR 소프트웨어가 제공하는 이벤트 처리, 정보시현 등과 같은 사용자 인터페이스만 대상으로 하였다. DTRS의 비행데이터 분석은 컨테이너 클래스 형태를 명령어와 데이터 단어로 분리하여 설계하였으며 많은 수의 데이터도 문제없이 분석하였다. 변환 로직도 정상적인 동작을 확인하였다. 데이터 선별을 위해 필터링 옵션을 추가로 제공하여 사용자 편의를 제공하였으며 선별 및 재생을 위한 데이터 가공도 정상적으로 동작하였다.

데이터 재생 기능을 검증하기 위해 FDR 시스템이 정상적으로 초기화 되었는지 확인하였다. 또한, ADS2의 RTIO (Realtime Input/Output)를 이용하기 때문에 RTIO가 정상적으로 초기화 되었는지 확인하였다. 그리고 상용 실시간 운영체제인 VxWorks가 정상적으로 적재되고 MIL-STD-1553 Card와 ADS2 코어 소프트웨어가 모두 정상적으로 초기화 되고 준비가 됨을 확인하였다. ADS2의 세션관리 소프트웨어를 통해 FDR 시스템에 가동에 필요한 실시간 데이터베이스, IO 설정, 시뮬레이션 모델 등을 적재하고 정상적으로 동작함을 확인하였다.

FDR 소프트웨어를 실행하여 비행데이터 중 가장 빠른 주기로 변하는 데이터 중 하나인 항공기 3축의 속도 데이터를 재생하여 녹화하였다. 먼저 데이터의 주기를 보면 20ms가 지켜지고 있으며 각 3축의 데이터가 실제 비행데이터와 동일함을 확인하였다.

시험장치의 운용환경은 MC Rack의 미션컴퓨터, MFD, HUD를 포함한 모든 하드웨어들이 정상 동작하였다. 성능은 FDR 소프트웨어가 Python API를 이용하여 이더넷 통신을 통한 데이터 전송을 위해 10ms 정도 소요되었다. 이를 위해 RTIO의 글로벌 스케줄링을 위한 Main frame의 전송률을 20ms에서 10ms로 설정하였다. 변경 후 실시간 데이터 스케줄링을 통한 오차범위 20ms로 데이터 재생이 됨을 확인하였다.

FDR 소프트웨어에서 1000개의 신호를 선택하여 시스템의 성능 변화를 측정하였다. 실시간 처리시간이 7.5ms에서 8.3ms 정도 증가하였지만 시스템에는 크게 영향을 미치지 않는다. 실제로 2000개 실행에도 잘 동작함을 확인하였다. 시스템 시험을 통해 FDR의 기능과 성능이 만족함을 확인하였다.

4.2 FDR의 재현 능력 평가

본 절은 FDR 시스템의 신뢰성을 검증하기 위해 비행데이터로 오류상황을 재현하는 기능을 평가하였다. 재현 능력은 항법, 무장, 통신, 그리고 센서 장비를 대상으로 총 24개의 종류의 시험을 수행하였으며 이 중 실제 지원 사례도 포함되어 있다.

본 연구에서는 FDR 시스템을 통하여 Table 1과 같은 종류의 복잡성과 관계없이 오류 상황을 재현할 수 있었다. OFP의 수정 후 결함을 해결하는 과정에서 다시 한 번 FDR 시스템을 실행하여 동일한 현상이 발생하지 않음을 확인하는 용도로도 사용되었다. 이 과정을 통해 이 도구가 결함의 검증과정에서도 지원이 가능함을 확인하였다. 그리고 지상에서 OFP 수정 후 반복적인 시험을 통해 충분한 검증을 할 수 있기 때문에

Table 1. Replay Performance

Type	Class of Test	# Test	Failed	Succeed
Navigation	10	10	0	100
weapon	7	10	0	70
Communication	3	10	0	30
Sensor	4	10	0	40
Total	24	40	0	240

검증을 위한 불필요한 고비용의 비행시험을 줄일 수 있는 가능성도 발견하였다.

4.2.1 공대지 무장 운용 오류

공군에서 FA-50 전투기의 운용 중 CCIP (Continuously Computed Impact Point) 공대지 무장의 투하모드에서 폭탄의 탄착점인 원 형태의 pipper가 갑자기 위로 튀어 올랐다가 유지되거나 돌아오는 현상이 발생하였다. 만약 이와 같은 문제가 발생한다면 잘못된 지점에 무장 투하를 할 수 있다.

이 문제를 분석하기 위해 녹화된 DTRS 데이터를 공군으로 부터 입수하여 FDR 시스템을 적용하여 재현에 성공하였다. 문제가 발생한 상황의 출력 데이터를 녹화하여 데이터를 추출하였다. 녹화된 데이터를 분석하여 시스템 고도 값이 다른 고도 값과 동일한 값으로 출력되어야 하지만 두 시점에서 차이가 발생함을 확인하였다. 이 결과를 바탕으로 미션컴퓨터의 OFP를 수정하였으며 해당 결함을 해결하였다.

4.2.2 공대공 무장 (gun) 시현 오류

MFD에서 공대공 무장 중 근접전 운용에서 쓰이는 기관포의 남아있는 발수 표시에 오류가 발생하였다. MFD에는 50GUN으로 표기되어 있으나 실제로 조종사에게 남은 발수는 없었다. 항공기의 전투 중에 근접전은 조종사에게 매우 긴급한 상황이다. 따라서 이러한 오류는 자칫 조종사의 생명에 위험을 초래할 수도 있으며 주어진 임무를 완수 못할 수도 있다. FDR 시스템을 통해서 이 현상 또한 쉽게 재현할 수 있었다.

4.2.3 순항연료 관리 알고리즘 오류

T-50 항공기는 순항연료 관리기능이 있으며 조종사에게 비행지시 (Flight Director)를 통해 정보를 제공하고 있다. 순항연료 관리는 4가지가 있는데 이 중 TOS(Time-Over Steerpoint)에서 HUD에 시현되는 현재 G/S 기준의 예상도착시간(ETA)과 항공기 목표속도(RQD G/S)의 비정상적 시현, Optimum Airspeed Caret의 점핑 현상이 발견되었다.

이 문제는 각 정보를 산출하는 계산 알고리즘이 복잡하여 결함을 분석하기 위한 오류상황을 재현하기가 어렵다. 하지만 FDR을 이용하여 동일한 오류상황의 재현에 성공하였다. ETA가 10:31:44로 정상 값보다 4~5초 정도 차이가 나며, RQD G/S 또한 정상 값인 125knots가 아닌 2배의 속도의 값이 시현됨을 확인할 수 있었다. 마지막으로 Optimum Airspeed Caret이 가끔씩 튀는 현상도 동일하게 재현이 되었다.

V. 결 론

항공기의 시현계통은 다양하고 많은 정보를 통합하여 조종사에게 시현하고 있다. 시현계통에서 소프트웨어 결함이 발생되면 정확하지 않은 정보가 조종사에게 시현될 수 있기 때문에 항공기의 운영에 심각한 영향을 미칠 수 있다. 따라서 시현계통의 개발 중에 소프트웨어 시험과 비행시험을 통해 발생 가능한 소프트웨어 결함을 식별하고 제거하려고 노력하고 있다.

본 연구는 비행시험에서 비행데이터를 이용하여 항공기 시현계통의 결함을 분석하는 도구인 FDR을 제시하였다. FDR은 항공기 임무컴퓨터와 연동되어 실시간으로 동작한다. 그리고 비행시험 데이터를 시현소프트웨어에 적용하여 시현장치에 나타나는 오류상황을 재현할 수 있다.

FDR의 결함분석능력의 입증을 위해 항공기의 운용 상황에서 보고된 시현소프트웨어의 결함을 분석하였다. 오류상황이 녹화된 비행데이터를 적용하여 FDR로 비행 상황을 재현하여 결함을 발견 할 수 있었다. 따라서 본 연구를 통해 불필요한 비행시험 없이 시현계통 소프트웨어의 결함분석을 지상에서 수행함으로써 빠르게 결함을 대응할 수 있었으며 검증을 위한 비행시험 횟수 감소를 기대할 수 있다.

향후 연구는 비행데이터를 재생하기 전에 상세한 분석 정보를 제공하는 기능을 추가할 것이다. 그리고 비행 시뮬레이션 모델을 적용하여 비행데이터의 시각화 기능을 제공할 것이다. 마지막으로 재생 시 특정 비행데이터를 변경하거나 제거할 수 있는 편집 기능을 추가할 것이다.

후 기

이 연구는 2015년도 경상대학교 발전기금재단 재원으로 수행되었음.

References

1) Kim, J. Y., NA, W. K., Chin, C. S., Kim, B. G., and Park, H. S., "Technical Development Trend of Cockpit Display Equipment," *Proceeding of The Korean Society for Aeronautical and Space Sciences Conference,* Apr. 2014, pp. 1060-1066.

2) Chang, W. H., Park, J. S., Jo, Y. W., and Byun, J. K., "Verification of Hierarchically Structured Avionics System Utilizing Multi-Mode System Integration Laboratory," *Journal of The Korean Society for Aeronautical and Space Sciences*, vol. 45, no. 11, Nov. 2017, pp. 998-1005.

3) Kim, Y. M., Jang, J. J., and Jun, B. K., "A Study on Improvement about display of Multi Function Display for KUH," *Journal of The Korean Society for Aeronautical and Space Sciences*, vol. 42, no. 4, 2104, pp. 344-350.

4) Jeon, H. K., Lee, K. H., and Lim, Y. B., 2014, "A Study on the Development of Digital Head Up Display for Aircraft," *Proceeding of The Korean Society for Aeronautical and Space Sciences Fall Conference*, 2014, pp. 1082-1085.

5) Lee, S. N., "A Suggestion to Create Blue Ocean of the Weapon System Software," *Defense & Technology*, no. 433, Mar. 2015, pp. 76-97.

6) Han, S. H., "An Approval Process of Software in Aircraft Certification Process," *The Journal of Aerospace Industry*, vol. 68, pp. 38-63. 2006.

7) RTCA, *DO-178C : Software Considerations in Airborne Systems and Equipment Certification*, RTCA, Dec. 2011.

8) Boehm, B. W., "Software Engineering Economics," *TRW Defense Systems Group*, 1983.

9) Balaji, S., Murugaiyan, M.. S., "Waterfall vs V-Model vs Agile: A Comparative Study On SDLC," *International Journal of Information Technology and Business Management*, 2012, pp. 26-30.

10) Park, W. H., Baek, H. S., Lee, D. H., Jeon, M. U., and Lee, J. H., "GPS/INS Bomb Flight Test Instrument from FA-50 Flight Test," *Proceeding of The Korean Society for Aeronautical and Space Sciences Conference*, no. 11, 2012, pp. 1873-1877.

11) Ban, C. B., and Ko, S. Y., "The Development Status of Crash Protected Integrated Airborne Recorder," *Proceeding of The Korean Society for Aeronautical and Space Sciences Conference*, no. 4, 2011, pp. 672-675.

12) Kim, J. S., Jo, S. B., and Choi, K. Y., "Development of Control Law Debugging Data Plot and Simulation (Flight Replay, Plot, Simulation) Tool," *Proceeding of The Korean Society for Aeronautical and Space Sciences Conference*, no. 11, 2014, pp. 1410-1413.

13) Lehmann, R. C., Wadley, J. E., von Bose, D. R., and Turne, R. M., "Simulation Tools for the Development and Evaluation of an Automatic Air Collision Avoidance System," *AIAA Modeling and Simulation Technologies (MST) Conference*, 2013.

14) Kim, D. G., Lee, Y. J., Myung, H. S., Jang, Y. C., Kim, C. Y., and Jee, C. K., "UAV System Designed for AIS POD for ACMI Simulation Linkage," *Proceeding of The Korean Society for Aeronautical and Space Sciences Conference*, no. 11, 2013, pp. 1214-1217.

15) Kim, D. Y., "A Technique and Development Trend of Cockpit Displays and Controls of Modern Aircraft," *Defense & Technology*, no. 283, 2002, pp. 66-73.

16) Fancher, S., 2010, "Avionics cost and complexity," *Aviation Week & Space Technology,* February 2015.

17) Kim, K. I., Lee, C. J., and Park, K. C., "Design of Interface Component for DTRS," *KICS Conference*, no. 2, 2012, pp. 1-2.

18) Nam, G. R., "ICD Definition or Automatic Generation of Interface Code and Interface Code Auto-generating System," *KIISE Confenrece*, no. 38(2A), 2011, pp. 31-34.

19) TechSAT, "*Avionics Development System 2nd Generation/Release4 User Manual,*" TechSAT, 2017.

20) Kim, J. H., Lee, S. C., and Ryu, K. S., "Development of Avionics Hot Bench for Avionics System Integration Test," *Journal of The Korean Society for Aeronautical and Space Sciences*, no. 36, 2008, pp. 507-513.

21) DoD (Department of Defence), *Aircraft Internal Time Divion Command/Response Multiplex Data Bug,* USA DoD, 1989.

22) Tutorials Point, *wxPython GUI Toolkit*, Tutorials Point, 2015.