

## 技術論文

J. of The Korean Society for Aeronautical and Space Sciences 42(3), 263-269(2014)

DOI: <http://dx.doi.org/10.5139/JKSAS.2014.42.3.263>

## 비행시험 자료 분석 및 가시화 프로그램 개발

박영근\*, 이성진, 이기두, 임상수, 이인원

## Flight data analysis and visualization program development

Young-Keun Park\*, Sung-Jin Lee, Gi-Doo Lee, Sang-Soo Lim and In-Won Lee

Agency for Defense Development

## ABSTRACT

Flight test data visualization functions can improve an understanding of flight test results, test procedures, and the performance of a flight vehicle after flight tests. FlyingView was developed for researchers to analyse flight test data in a 3D virtual environment. It also can display X-Y plots using flight test data. It was developed and applied to flight tests of an air-to-ground weapon system of ADD. This paper describes the capabilities of FlyingView.

## 초 록

비행시험 자료 가시화 기능은 비행시험 이후 비행체의 시험 절차 및 현상을 이해하는데 많은 도움이 된다. FlyingView는 비행시험 자료를 활용하여 3차원 가상환경 기반으로 비행체의 비행화면을 전시할 수 있으며, 동시에 자료 분석을 위한 X-Y 그래프를 전시할 수 있도록 개발되었다. 본 프로그램은 국방과학연구소의 공대지 무장 비행시험에 적용하여 활용하였다. 본 논문에서는 FlyingView의 기능과 향후 개발계획에 대해서 기술하고 있다.

**Key Words** : Flight Test(비행시험), Data Analysis and Visualization(자료 분석 및 가시화), Air Weapon System(항공무장)

## 1. 서 론

공학 분야에서 해석 및 시험 결과에 대한 가시화는 공학자들이 시험결과에 대한 물리적 현상을 보다 빠르고 쉽게 이해할 수 있는 효과적인 방안으로 연구되어왔다. 특히 비행 시험의 경우 많은 계측 센서로부터 방대한 양의 디지털 및 아날로그 자료가 생성되어 자료 처리 및 분석을 통해 의미 있는 현상을 파악하는데 많은 시간과 노력이 소요된다.

미국 NASA Langley Research Center에서는

LiveView3D S/W[1]를 개발하여 실시간 풍동시험 자료 가시화에 적용하였으며, SimGraph S/W[2]를 개발하여 비행모의 가시화에 활용하였다. 미 해군연구소에서는 다양한 시뮬레이션 및 시험, 훈련용 어플리케이션에 공통으로 사용할 수 있는 가시화 프로그램인 SIMDIS를 개발하였다. SIMDIS는 가시화 측면에서 2D, 3D모드를 지원하며 많은 군 관련 기관에서 사용하고 있다[3].

국내의 경우 항공기에 대한 비행데이터 가시화 프로그램이 개발되어 KT-1 및 T-50 개발 비행시험 분야에 적용되었다[4][5]. 특히 방대한

† Received: August 9, 2013 Accepted: February 17, 2014

\* Corresponding author, E-mail : [pyk@add.re.kr](mailto:pyk@add.re.kr)

<http://journal.ksas.or.kr/>

pISSN 1225-1348 / eISSN 2287-6871

비행시험 데이터에 대한 빠른 처리를 위하여 리눅스 기반의 클러스터를 이용한 데이터 및 그래픽 처리 시스템에 대한 연구가 진행되어 왔다. [6][7] 최근에는 국내의 우주 발사체에 대한 연구가 활발히 진행됨에 따라 항공분야 뿐만 아니라 우주분야에도 시험 분석 및 가시화에 대한 연구가 진행되고 있다[8][9]. 그러나 기존에 국내에서 개발된 비행시험 가시화 프로그램의 경우 적용될 체계 목적에 맞춰 개발됨으로써 다른 체계 사업에 적용될 때 많은 어려움이 발생하였다. 따라서 이러한 문제점을 해결하기 위하여 가시화 프로그램 개발 시 확장성을 고려한 모듈화 설계 및 데이터 처리에 대한 표준화 필요성이 제기되었다.

비행시험 분석 및 가시화 프로그램(FlyingView)은 국방과학연구소에서 개발한 “중거리 GPS 유도폭탄(KGGB: Korean GPS Guided Bomb)”의 비행시험 결과 분석을 목적으로 확장성을 고려하여 개발되었다. FlyingView는 비행시험 중 측정된 계측 자료를 활용하여 비행시험 상황을 현실감 있게 재현함으로써 비행시험 시 발생한 물리적 현상을 보다 빠르고 쉽게 이해할 수 있도록 한다. FlyingView는 시험 결과 분석을 위해 다양한 그래픽 창을 제공하고 있는데, 일반적인 2D 그래픽 창 및 스트립차트, 고해상도의 3차원 비행 상황 가시화창을 지원한다.

비행시험 가시화 프로그램은 비행시험 이전에 비행시험 모의 결과를 분석하기 위한 목적과 비행시험 후 실제 비행시험 결과 분석을 위해 계측 자료에 대한 가시화를 하기 위한 2가지 목적으로 활용하고 있으며, 향후에는 비행시험에 대한 실시간 가시화 기능까지 개발할 예정이다.

## II. 본 론

### 2.1 시스템 개요

#### 2.1.1 개발 환경

FlyingView는 Windows XP 운영체계의 Visual Studio 2008 및 .Net Framework 3.5기반으로 개발언어는 C#을 활용하여 개발하였다.

#### 2.1.2 프로그램 구성

FlyingView는 크게 계측자료 처리 모듈, 가시화 지원 모듈로 구분할 수 있다. 계측자료 처리 모듈은 RawData 처리모듈, TextData 처리모듈, TransData 처리모듈로 구성되며, 가시화 모듈은 3차원모델 비행형상 가시화 서브모듈, 2차원 상황정보 가시화 서브모듈, 자료 분석 스트립차트

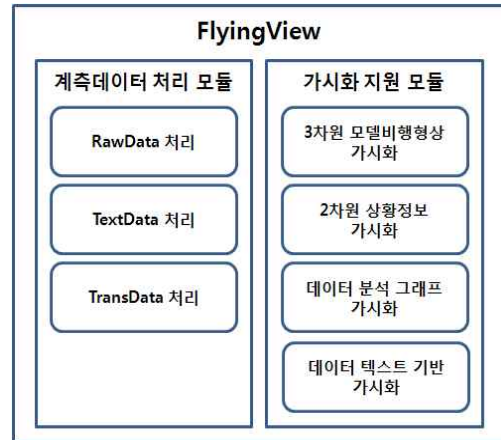


Fig. 1. Functions of FlyingView

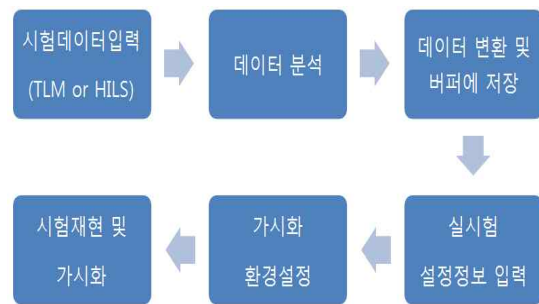


Fig. 2. Procedures to execute FlyingView

서브모듈, 시험자료 텍스트 가시화 서브모듈로 구성된다.

FlyingView는 비행시험 계측자료(Telemetry, HILS 등의 RawData)가 입력되면 자료의 종류 및 내용을 분석하여 프로그램에서 처리할 수 있는 형태(TransData)로 변환한 후 자료 관리 버퍼에 저장한다. 이렇게 변환된 형태의 계측 자료는 계측자료 처리 모듈이 버퍼에서 순차적으로 읽어서 broadcasting 하게 된다. 자료 분석을 목적으로 설계된 GUI 기반의 가시화 모듈은 broadcasting 자료가 입력되면 설정에 맞게 분석 자료를 전시한다. 이때 시험분석을 용이하게 하기 위해서 실제 비행시험에서 사용하는 시나리오, 예상궤적, 안전영역 등의 정보를 함께 전시할 수 있도록 고려했으며, 사용자에게 맞게 각 가시화 창의 설정을 변경할 수 있도록 설계했다.

### 2.2 시스템 개발

계측자료 처리 모듈은 계측 자료의 형식이 서로 다른 경우의 호환성을 고려하여 XML 및 템플릿 언어를 적용하였으며, 가시화 지원 모듈은 기본적으로 2차원, 3차원 각각에 대한 가시화 기능을 지원하고 있으며 보다 효과적인 분석을 고

려하여 2차원과 3차원 가시화 화면을 동시에 전 시할 수 있는 기능도 제공하고 있다.

### 2.2.1 RawData 처리

RawData는 Telemetry 자료 및 HILS 수행 시 생성되는 비행 궤적 및 자세 등에 대한 Binary 데이터를 통칭한다. 이것은 장비별 설계 데이터형식에 따라 다르므로 해석기능의 호환성을 고려하려면 해석을 지원하기 위한 별도의 파일이 필요하다. FlyingView의 초기버전에서는 RawData 해석 모듈을 호환성 고려 없이 개발하였다. 그 결과 RawData가 변경될 때마다 FlyingView의 기능을 수정해야했고 컴파일 및 테스트를 거쳐서 다시 정상적으로 운용하기까지 많은 시간과 비용을 소모해야하는 문제가 있었다. 이러한 이유로 XML 및 템플릿 언어를 활용하여 자료 처리에 대한 규칙을 정의할 수 있도록 함으로서 자료처리 호환성을 확보하였다.

RawData의 구조를 사람이 먼저 XML로 작성하면 이를 컴퓨터가 템플릿도구를 이용하여 자동으로 컴퓨터코드를 생성하는 방식으로 자료 처리의 호환성을 고려하고 있다. 새로운 형식의 RawData를 위해 작성 완료된 XML은 RawData에 대한 모든 정보(Endian 방식, 데이터타입, 데이터크기 등)를 담고 있으므로 이를 Text\_Templating\_File\_Generator라는 C# 도구와 연계하면 "RDManager.cs" 라는 C# 기반의 RawData 처리 클래스파일을 자동 생성할 수 있다. 이렇게 자동 생성한 "RDManager.cs" 파일을 기존파일과 교체하여 컴파일하면 변경된 형식의 자료를 처리할 수 있게 된다.

Text Templating File Generator는 VisualStudio

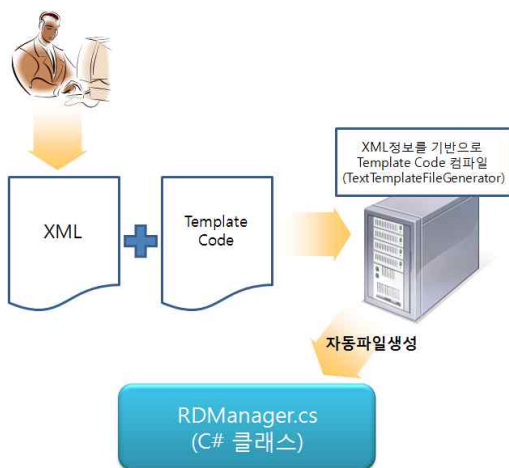


Fig. 3. Automated RawData managing class creation

```

<?xml language="C#" ?>
<?assembly name="System.Xml.dll" ?>
<?import namespace="System.Xml" ?>

<?
  XmlDocument doc = new XmlDocument();
  doc.Load(@"D:\Projects\K9gb_sim\src_file\K9GBSimulator\StateModel.xml");
  XmlElement xnModel = (XmlElement)doc.SelectSingleNode("/StateModel");
  string name = xnModel.GetAttribute("name");
  string ns = xnModel.GetAttribute("namespace");
  XmlNodeList states = xnModel.SelectNodes("descendant::State");
  XmlNodeList inputs = xnModel.SelectNodes("descendant::Input");
  XmlNodeList trns = xnModel.SelectNodes("descendant::Transition");
  ?>

using System;
using System.Collections.Generic;

namespace <#-name#>
{
  public class <#-name#>StateModelException : Exception
  {
    public <#-name#>StateModelException(string message)
    {
      base(message);
    }
  }

  public enum <#-name#>States : int
  {
    <#
    string sep = "";
    foreach(XmlElement s in states)
    {
      WriteLine(sep + s.GetAttribute("name"));
      sep = ", ";
    }
    <#
  } // end enum <#-name#>States
}
    
```

Fig. 4. Regulation definition using Text Templating File Generator

C# 컴파일러에서 지원하는 템플릿 도구로서 RawData정보를 담고 있는 XML의 정보를 어떻게 활용할 것 인지에 대한 규칙을 템플릿으로 구현해두면 XML 내용에 맞춰 자동으로 C# 클래스를 생성한다. 이러한 방식으로 자동 생성된 클래스는 작성오류가 없으므로 추가적인 기능 테스트 없이도 신뢰할 수 있다.

### 2.2.2 TextData 처리

TextData는 텍스트파일 형태로 이미 변환된 자료이므로 RawData의 처리에 비해서 비교적 간단하게 처리할 수 있다. "RDManager.cs"는 기본적으로 FlyingView의 자체 텍스트파일 포맷인 "\*.gbd" 파일을 해석할 수 있도록 설계되었다. 따라서 TextData를 GBD 포맷에 맞게 각 행별 데이터를 편집하면 FlyingView가 이를 해석할 수 있도록 고려하였다. 향후에는 GBD 파일생성을 위한 변환모듈을 추가 개발하여 보다 쉽게 TextData를 편집할 수 있도록 할 예정이다.

### 2.2.3 TransData 처리

TransData는 "RDManager.cs"가 RawData 또는 TextData의 각 세부 자료항목들을 인스턴스화하여 FlyingView에서 직접 활용할 수 있는 형태로 변환한 자료를 의미하며 FlyingView의 구동에 필요한 모든 정보를 포함하고 있다. 가변적이고 방대한 RawData로부터 TransData를 추출하기 위해서는 "RDManager.cs" 생성을 위해 사용하는 XML에 필요자료 항목을 정의해야 한다.

각종 가시화 모듈 및 분석 모듈이 시간적인 오차 없이 연동되기 위해서는 TransData를

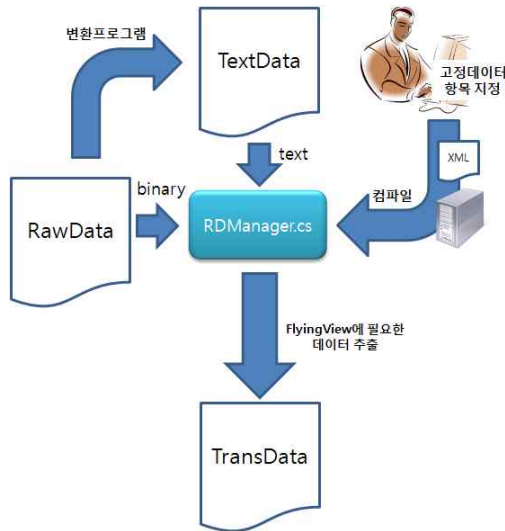


Fig. 5. Design of test data handling

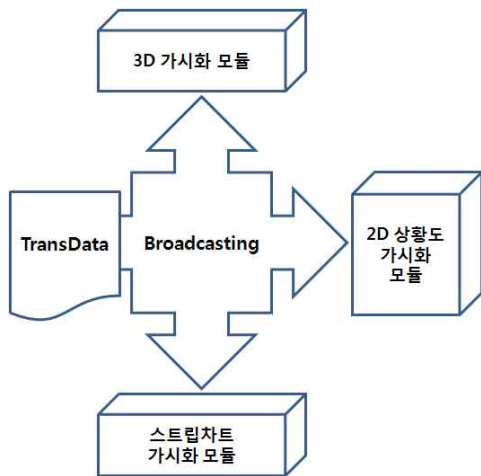


Fig. 6. Multi-thread based TransData broadcasting

broadcasting한 후 이를 가시화하는 방식으로 구현하는 것이 효과적이다. 따라서 Multi-thread 방식으로 가시화 모듈과 자료 처리 모듈의 thread를 분리하여 자료 처리 및 가시화 기능을 구현하였다.

2.2.4 3차원 모델 비행형상 가시화

3차원 모델 비행형상 가시화 모듈은 비행체의 미세한 자세변화 및 주변 환경들에 대한 영향을 효과적으로 가시화 하는 것이 주된 요구사항이다. 이를 위해 기본적으로 WGS-84 좌표체계 기반의 3차원 지형을 활용하였고 비행시험과 관련된 주요지물 및 정밀지형은 별도로 모델링하여 배치하였다. 3차원 지형으로는 pixel당 최대해상도 15m인 Landset 7 과 SRTM(Shuttle Radar Topography Mission) 데이터를 활용하여 대한민

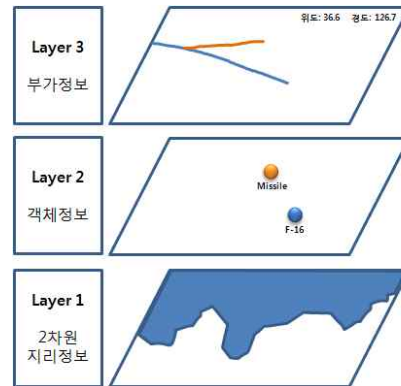


Fig. 7. Three layers of 2D view

국 지형 및 인근해상을 표현하였고 기타영역을 pixel당 1km급 Blue Marble 이미지를 이용하여 가시화하였다.

비행형상은 TransData의 6DOF 정보를 활용하여 3차원 공간상의 위치와 비행자세를 가시화하였다. 이때 TransData의 수신주기가 너무 빠르면 가시화에 대한 부하가 너무 커질 수 있으므로 10~30HZ 범위에서 하드웨어 환경에 맞게 자동 설정되도록 구현하였다.

2.2.5 2차원 비행상황 가시화

2차원 비행 상황 가시화 모듈은 비행시험상황을 전체적으로 확인 및 판단할 수 있도록 지원하는 것을 목적으로 한다. 본 모듈은 크게 3가지의 Layer로 구성되어 있다. 1단계 Layer는 ESRI 표준 WGS-84 기반의 벡터 지도 데이터를 활용하는 2차원 지리 정보를 표현하기 위한 Layer이고 2단계 Layer는 좌표정보로 표현되는 객체들을 표현하기 위한 Layer이며 3단계 Layer는 객체별 비행궤적, 기타 보조 정보 등의 부가정보를 표현하기 위한 Layer이다.

2.3 시스템 운용

Windows 계열의 XP 이상 운영체제와 NVIDIA 계열 그래픽카드를 고려했으며 동시에 여러 개의 화면을 볼 수 있는 시스템 특성상 화면해상도가 클수록 운용하기 편리하다.

2.3.1 프로그램 기능

구현된 프로그램은 Fig. 8과 같이 중앙에 2차원 종합 상황도를 기준으로 우측 상부에 3차원 비행 가시화 창과 좌·우측 측면의 계측 변수 창으로 구성된다. 2차원 종합 상황도의 경우 상면 창에는 위·경도를 기준으로 2차원 지도에 비행체의 비행경로가 시현되며, 하면 창에는 경도 및 고도를 기준 축으로 비행경로가 시현된다. 2차원 지도는 벡터지도, 래스터지도, 벡터 및 래스터 지



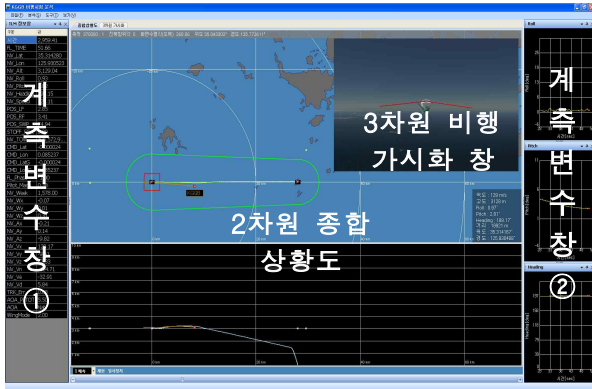


Fig. 8. Layout of FlyingView

도를 동시에 볼 수 있는 3가지 유형으로 사용자가 선택할 수 있고, 기본으로 WGS-84 및 BESSEL 좌표계가 제공된다. 3차원 비행 가시화 창은 3차원 지형 모델을 기반으로 시험 비행체의 비행 진행 현황을 입체적으로 시현하는 창으로서 필요에 따라 창의 크기 조절이 가능하며, 비행체의 관찰 방향을 사용자가 쉽게 변경할 수 있다. 또한 사용자의 편의를 고려하여 2차원 종합 상황도와 3차원 비행 가시화 창의 전환도 가능하며, 비행시험 재현속도는 0.1 배속부터 32배속까지 조절할 수 있도록 설계하였다. 좌측의 계측 변수 창은 사전에 정의된 계측 변수 값을 비행시간에 따라 Text형태의 수치 값으로 표시하며, 우측의 계측 변수 창은 시간에 따른 비행체의 3축 자세 (Pitch, Roll, Heading)를 스트립차트로 전시한다. 사용자는 필요에 따라 좌측 창에 정의된 계측 변수 값에 대하여 스트립차트를 별도로 추가 생성하여 전시할 수 있다.

2.4 시스템 적용

국방과학연구소를 비롯해 여러 기관에서 항공 및 우주와 관련한 각종 비행시험이 진행 중에 있다. 비행시험 후 비행 결과를 세부적으로 분석하기 위해서 전문 분야로 많은 인력과 시간, 전문가들이 요구된다. 그러나 비행시험 직후 설계자 및 시험 관련자들이 모여 비행시험 결과에 대한 전반적인 이해를 도모하는 것은 비행시험에 대한 세부적인 분석 방향을 수립하는데 매우 도움이 된다. 따라서 다양한 분야의 엔지니어들이 빠르고 쉽게 비행시험 결과를 이해할 수 있도록 지원할 수 있는 틀은 매우 필요하다.

FlyingView는 이러한 문제를 해결하기 위하여 개발되었으며 Fig. 9와 같이 국방과학연구소에서 개발 중인 KGGB의 비행시험 결과 분석에 적용되었다. KGGB는 국내 최초로 개발된 공대



Fig. 9. Mounted KGGB for flight test

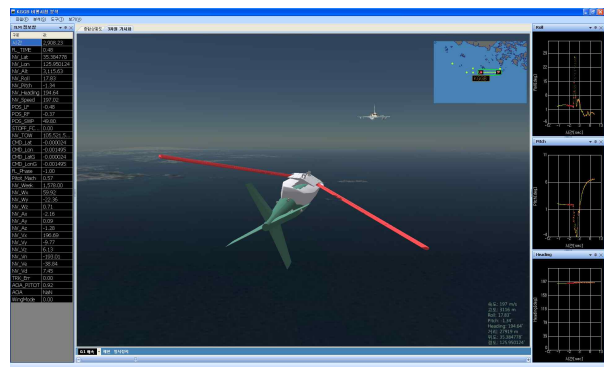


Fig. 10. 3D view mode

지 유도무기이며, 일반목적폭탄에 유도키트를 부착하여 사거리 및 정확도를 획기적으로 향상할 수 있는 무기체계로 국내에서 운용중인 다양한 기종의 전투기에 장착되어 운용될 예정이다.

KGGB 비행시험의 경우 FlyingView 운용을 위해 입력자료 처리에 있어 TextData 입력방식을 활용하고 있다. 비행시험 중 계측된 자료는 저장 장치에 저장되며 별도의 변환 프로그램을 거친 후 분석에 필요한 변수로 구성된 Text 파일로 저장된다. 이 파일에는 최소한 분석에 필요한 비행 시간 및 위치, 자세 등 필수 변수가 포함되어야 한다. 생성된 파일을 입력 자료로 FlyingView가 구동되며, 프로그램에서 별도로 지상온도의 입력을 요구하는데, 이때 비행시험 시점의 지상 대기 온도를 입력하면 표준 대기 모델을 기준으로 고도별 음속이 계산된다.

Figure 10은 구현된 기능 적용 예로 투하초기 2차원 종합 상황도와 3차원 비행 가시화 창을 전환하였을 때 화면 구성을 보여 주고 있다.

FlyingView에서 비행시간에 따른 비행체 자세를 스트립차트형태로 기본으로 제공하고 있으나 사용자의 분석 목적에 따라 다양한 스트립차트를 새롭게 생성할 수 있다. Fig. 11 및 Fig. 12는 종말 단계 비행 중 계측 변수 창의 기능 구현의 예로써 계측된 조종면의 각도를 스트립차트로 추가



Fig. 11. Configuration window for data analysis plots

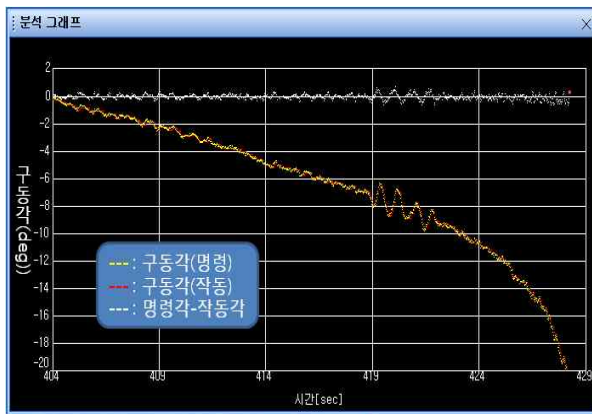


Fig. 12. Data analysis plot

생성하는 과정 및 결과를 보여 주고 있다. Fig. 11은 데이터 설정 창으로 x축으로 비행시간을 y축으로 3개의 변수를 지정하였는데 유도제어기로부터 생성된 좌측 조종면의 구동 명령(각도)과 실제 구동기에서 측정된 구동기 작동각도를 각각 변수로 설정하였으며, 변수 연산 기능을 활용하여 구동명령과 구동기 작동 각도 차이를 y축 변수로 추가 설정하였다. Fig. 12는 데이터 설정 결과로써 생성된 변수 스트립차트로 노란색은 조종면 구동명령 값, 적색은 조종면 구동각도를 보여주며, 흰색은 구동명령과 구동기 작동 각도 차이를 스트립차트 형식으로 보여 주고 있다.

이를 통해 종말단계에서 유도제어기로부터 전달된 구동명령을 구동기가  $\pm 1$ 도 이내에서 잘 추종하고 있음을 확인하였다.

FlyingView를 적용하기 이전에는 비행시험 후 결과데이터를 타 부서로 송부하여 가공 및 분석을 진행한 후 그래프 형태의 분석결과를 획득하기까지 약 1주일정도의 시간이 소요되었으며 종합적으로 비행 시험 결과를 이해하는데 어려움이

있었다. 그러나 FlyingView를 비행시험 분석업무에 적용한 결과 비행시험 후 1시간 이내에 비행시험에 대한 가시적인 분석이 가능해졌으며, 연구자가 반복적으로 비행시험 결과를 눈으로 확인함으로써 비행시험에 대한 이해도가 크게 향상되었다.

### 2.5 향후 계획

현재 버전의 FlyingView는 상용 또는 오픈소스로서 공개하여 활용하기에는 가시화 기능의 개발이 미흡하다. 따라서 모든 유도무기 기반의 비행시험 프로세스를 지원할 수 있도록 기능개선을 진행할 예정이다. 또한 유도무기를 개발하기 위해서 데이터를 분석해야 하는 일은 비행시험 결과데이터에 국한되지 않는다. 오히려 개발초기형상 및 성능을 결정하기 위해 진행되는 체계 6 자유도 모델 기반의 체계성능모의 결과의 분석이 더 중요하다고 판단되므로 이를 위한 요구도 분석 및 기능개발을 계획 중이다.

## III. 결 론

항공무기체계 개발 과정에서 필수적으로 수행되는 비행시험은 방대한 양의 측정 자료가 생성되어 자료 분석에 많은 노력이 소요된다. FlyingView는 비행시험 자료를 입체적으로 재현할 수 있을 뿐 아니라 다양한 분석 기능을 제공함으로써 비행시험 시 발생한 물리적 현상을 보다 쉽게 이해할 수 있도록 한다. 실제 FlyingView는 국내 최초로 개발된 공대지 유도무기인 KGGB 사업에 적용되어 비행시험분석에 활용됨으로써 효용성이 입증되었다. 향후 KGGB 사업 경험을 통해 도출된 개선 사항을 바탕으로 지속적으로 성능 개량이 진행 중이며, 향후 타 무기체계 개발에도 쉽게 적용 가능하도록 모듈화할 예정이다.

## References

- 1) Richard J. Schwartz, Gary A. Fleming, "Advanced Visualization of Experimental Data in Real Time Using LiveView3D" 25th AIAA Aerodynamic Measurement Technology and Ground Testing Conference, June. 2006
- 2) Joseph A. Kaplan, Ronnie Chen, "Development of a Flight Simulation Data Visualization Workstation," NASA Technical Memorandum 110308, Dec. 1996

- 3) Jae-Moon Yoon, Sim-Yong Lee, An Introduction of SIMDIS and its applications to analyze and visualize the results of simulations, Agency for Defense Development, 2008, 9
- 4) Jae-Suk Gi, "Study on Developing a Data Monitoring System for Live Fire Testing", Journal of the Korea Industrial and Systems Engineering, 2004. 10.
- 5) Ki-Bum Lee, Yung-Sin Kang, Im-Joo Kang, Su-Young Kim, Ho-Sung Kim, "Flight Test Improvement with Flight Verification by Running Simulation in the Control Room", Journal of the Korean society for aeronautical & space, 2003. 4.
- 6) Se-Ryong Oh, Joong-Sun Park, Chan-Jo Kim, "The Design of Flight Test Data Processing System using Linux Cluster Method," Journal of the Korean society for aeronautical & space, 2007. 11.
- 7) Byung-Soo Kim, Gi-Young Kim, "Parallel Multi-Monitor Display of Flight Simulation Graphics Using Linux Cluster", Journal of the Korea Institute of Military Science and Technology, 2006.
- 8) Woong-Rae Roh, Sang-Bum Cho, Jeong-Hwan Ko, Byung-Chan Sun, Jeong-Yong Kim, Jeong-Joo Park, Gwang-Rae Cho, "Trajectory and Attitude Analysis for the 1st Flight Test of KSLV-I Launch Vehicle", Journal of the Korean society for aeronautical & space sciences, 2010. 3
- 9) Hoon-Hee Lee, Dong-Young Ryu, Chul-Heoi Gu, Kwang-Hyuk Ju, "Introduction of 3-Dimensional Visualization for Planet Exploration: Moon 3D" Journal of the Korean society for aeronautical & space sciences, 2012. 11.