

비행체 자동점검장비의 유지보수성 향상 방안

Maintainability Improvement of Automatic Test Equipment for Aircraft

서민기* · 권기용 · 김성우 · 이성우
LIG 넥스원 항공연구소

Min-gi Seo* · Ki-yong Kwon · Seong-woo Kim · Seong-woo Lee
Avionics R&D Lab, LIG Nex1, Daejeon, 34115, Korea

[요 약]

공학 용어로 유지보수성(maintainability)은 제품의 유지/관리에 대한 용이성을 의미한다. 비행체 자동점검장비(ATE)는 점검대상장비(UUT)의 기능/성능 특성에 따라 점검절차가 개발되기 때문에 UUT의 설계 변경에 많은 영향을 받는다. 더욱이 통합 ATE가 다수의 UUT를 점검하는 환경이라면 UUT 설계 변경에 대한 ATE의 유지보수가 쉽지 않다. 개발자는 ATE의 유지보수성 향상을 위하여 쉽고 명확한 개발 방식을 선정하여 개발 담당자의 변경에 유연하게 대처할 수 있어야 한다. 또한 UUT 설계 변경에 영향을 받는 영역을 특정 범위로 제한하여 수정될 부분을 최소화할 수 있어야 한다. 본 논문에서는 비행체 ATE 개발 과정을 기반으로 유지보수성 향상에 초점을 둔 ATE 개발 방안을 제안한다.

[Abstract]

Maintainability in engineering terms means ease of maintenance / management of the product. Aircraft automatic test equipment (ATE) is heavily influenced by the design changes of the unit under test(UUT) since the test procedure is developed according to the function / performance characteristics of the UUT. Moreover, if the integrated ATE is an environment that checks more than one UUT, it is not easy to maintain the ATE for the UUT design changes. Developers should be able to flexibly cope with the change of development staff by selecting an easy and clear development method to improve the maintainability of ATE. It is also necessary to limit the area affected by the UUT design change to a specific range to minimize the area to be modified. In this paper, we propose ATE development method which focuses on maintenance improvement based on the aircraft ATE development process.

Key word : Automatic test equipment, Maintainability, Unit under test, Test scenario, Real-time test.

<https://doi.org/10.12673/jant.2017.21.5.508>



This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Received 18 September 2017; Revised 20 September 2017
Accepted (Publication) 26 September 2017 (30 October 2017)

*Corresponding Author; Min-gi Seo

Tel: +82-42-718-3565

E-mail: mingi.seo@lignex1.com

I. 서론

전통적인 방식의 야전급 자동점검장비는 LRU (line replaceable unit) 별로 각각 개발, 운용되었다. 하지만 이는 개발 및 유지보수에 많은 비용이 소모되는 비효율적인 구조이다. 따라서 하나의 자동점검장비가 다수의 LRU를 점검할 수 있는 통합형 자동점검장비의 필요성이 대두되었고, 국내에서는 한국형 기동헬기 개발 등에 통합형 자동점검장비로 개발된 바 있고 [1], 점검장비 공용화에 대한 제안도 있었다[2][3].

하지만 통합형 자동점검장비는 점검대상장비 (UUT; unit under test)의 설계 변화에 많은 영향을 받는다. 따라서 영향을 받는 영역을 제한하여 설계 변화 발생 시, 자동점검장비의 형상 변경을 최소화할 수 있는 유연한 설계가 필요하다. 이를테면, UUT가 운영하였던 CAN 인터페이스를 더 이상 사용하지 않는다는 결정이 발생했을 때 마우스 클릭만으로 내용을 적용할 수 있다든지, 반대로 CAN 인터페이스가 추가될 때 ATE 인터페이스 카드 여유분만 있다면 하드웨어 핀 및 와이어 추가와 소프트웨어의 간단한 수정만으로도 조치될 수 있음을 뜻한다. 여기에 더불어 설계 방식도 중요하다. 무기체계 장비 특성 상 수명주기가 긴 점을 감안하여, 개발 담당자의 변동이 발생하더라도 신규 담당자가 쉽고 명확하게 숙지할 수 있어야 한다. 위의 사항들은 모두 유지보수성 영역에 포함되는 것으로 제품의 유지/관리에 대한 용이성을 의미한다.

본 논문은 비행체 야전급 자동점검장비 (ATE; automatic test equipment) 개발 과정을 바탕으로 유지보수성 중심의 설계 방안을 제시한다. ATE의 전반적인 구성으로부터 시작하여 신호 경로 보상에 대한 방안, 실시간 동작을 위한 구조 등으로 내용을 구성하였다.

II. ATE 정의 및 구성

미국 DoD (department of defense)에서 정의하는 자동점검장비란 컴퓨터를 이용해서 복잡한 시험장비들을 제어하여 UUT의 규격조건을 시험하고 분석할 수 있는 장비를 말한다[4].

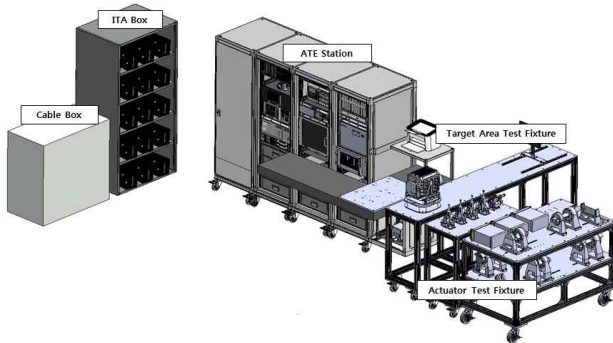


그림 1. 비행체 ATE 형상
Fig. 1. Configuration of aircraft ATE.

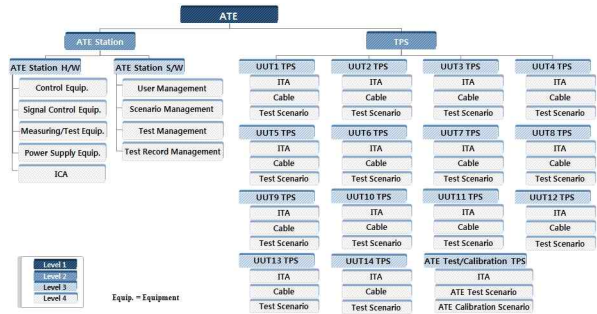


그림 2. 비행체 ATE 제품분할구조도
Fig. 2. PBS of aircraft ATE.

본 비행체 ATE 개발에서는 DoD의 자동화 시험장비 관련 용어 정의를 아래와 같이 수정하였고, ‘시험’이라는 용어의 혼신을 고려하여 ‘점검’으로 통일하였다.

$$\begin{aligned} \text{ATE} &= \text{ATE Station} + \text{TPS} \\ \text{ATE Station(ATES)} &= \text{ATES HW} + \text{ATES SW} \\ \text{TPS} &= \text{ITA} + \text{Cable} + \text{점검 시나리오} \end{aligned}$$

ATES 하드웨어는 운용 컴퓨터를 비롯하여 SBC (single board computer), 통신 인터페이스 (MIL-STD-1553, CAN 등), 계측기 (digital multimeter, arbitrary wave generator, calibrator 등), Power Supply, ICA (interface connection adapter) 등 대부분 상용품과 자체 개발 구성품인 Scanner로 구성되며, Scanner는 ATE의 입출력 신호의 경로를 제어한다.

ATES 소프트웨어는 UUT 및 ATE 자체의 점검을 수행하고 결과를 확인하는 GUI이다. 이 GUI는 점검자의 편의성을 중심으로 설계되며, 점검절차는 점검 시나리오 파일을 SBC에 적재하여 실행한다.

TPS (test program set)는 UUT 별로 구성된다. 비행체 ATE에서는 ATE 자체점검/교정 TPS까지 포함하여 총 15종의 TPS가 개발되었다. ATES에서 외부로 입출력되는 모든 신호는 ICA로 연결되는데, UUT 별 ITA (interface test adaptor)를 ICA와 결합하여 사용한다. ITA는 해당 UUT의 케이블을 연결할 수 있는 전면부와 내부 배선을 통해 ICA에 해당 인터페이스로 연결할 수 있는 후면부로 구성된다. 즉, ITA는 UUT와 ATE 간 신호 중개 역할의 장치이다.

UUT의 설계 변경은 주로 인터페이스 추가/삭제를 비롯하여 통신 ICD (interface control document)의 값/구조 변화와 기능/성능 측면의 장비 동작 수정을 들 수 있다. 따라서 ATE는 UUT의 설계 변화에 용이한 대응을 위하여 대응 계층을 ITA 및 Scanner, UUT 점검 시나리오로 제한할 수 있다. 물리적인 신호의 추가/삭제 등의 변경이 아니라면 수정 영역은 점검 시나리오로 한정된다. 점검절차와 인터페이스 정의, ATES 소프트웨어에 출력될 문구 및 출력여부 등의 기능을 집약하여 이를 손쉽게 수정할 수 있도록 구성하였다.

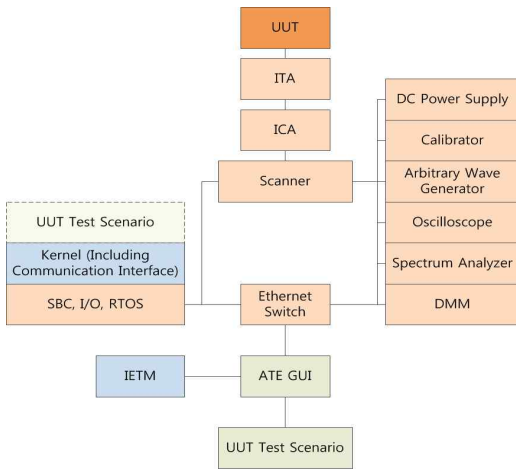


그림 3. ATE 시스템 아키텍처
Fig. 3. System Architecture of ATE.

III. 점검 시나리오 개발

점검 시나리오 개발 도구는 국내 업체인 Realtimewave社의 상용 도구 TestNgin을 선정하였다. TestNgin은 서두에 언급되었던 유지보수성 문제에 손쉽게 대응 가능하도록 다음과 같은 기능을 제공하여, 유지보수성을 향상시킨다.

- 순서도 형태의 점검 시나리오 제작 방법
- 심벌로 정의되는 인터페이스 및 데이터 입출력 기능
- XML 태그를 활용한 ATE 소프트웨어와의 연동 기능

특히 순서도 작성 방식은 담당자가 변경되더라도 쉽게 숙지할 수 있다. 추가적으로 TestNgin을 비행체 ATE 개발에 활용하기 위해 실시간성 관련 기능과 내부 데이터 제어, 자체점검 (BIT; built in test) 및 교정 (Calibration) 기능 등을 추가하였다.

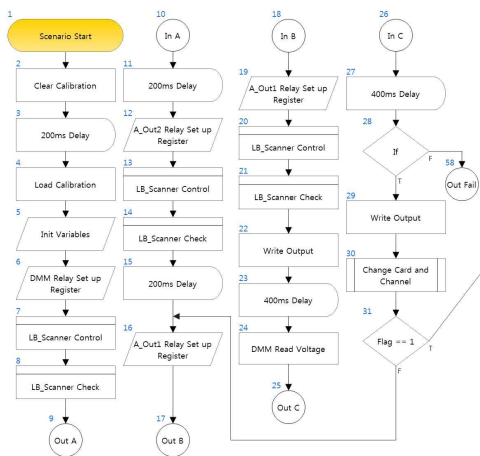


그림 4. 점검 시나리오 예시
Fig. 4. Example of test scenario.

3-1 공통 구조 설계 및 공통 모듈 개발

ATES 소프트웨어는 점검 시나리오의 모든 점검 항목 중 특정 항목을 선택하여 수행할 수 있기 때문에 절차 실행에 있어 다양한 경우의 수가 발생할 수 있어 최소의 규약을 정의하였다.

ATES 하드웨어 중에서 자체 개발 구성품인 Scanner 는 신호의 경로를 설정한다. Scanner 보드의 통신 인터페이스는 ARINC 429로 Label 구성에 따라 명령 (Command)/응답 (Response)/ARINC 429 통신 에러 발생 여부 확인 등의 부가적 기능으로 분류된다. Scanner 점검 시나리오는 두 컴포넌트로 구성하여 제어 적용 (명령), 상태 확인 (응답)으로 각기 정의하고 내부 데이터에 관련 데이터를 저장하여 모 시나리오 (parent scenario)가 확인할 수 있도록 설계 되었다. Scanner 점검 시나리오는 모든 점검 시나리오의 공통 모듈이므로 각 시나리오에 배포되어야 하는데, 해당 점검 시나리오 프로젝트에 Import 하고 관련 데이터를 정의하는 것으로 사용 가능하다.

표 1. 점검 시나리오의 실행 구조
Table 1. Execution structure of test scenario.

Category	Description	Condition Precedent
ITA Check	Confirm test scenario and ITA match	-
Initialize	Initial procedures that must be performed, regardless of the choice of test item (ex: UUT power on)	ITA Check Normal
Test	Confirm whether or not each test item is selected, and perform the detailed test procedure only for the selected test item	ITA Check Normal
Exit	A termination procedure that must be performed regardless of the choice of test item (ex: UUT power off)	ITA Check Normal

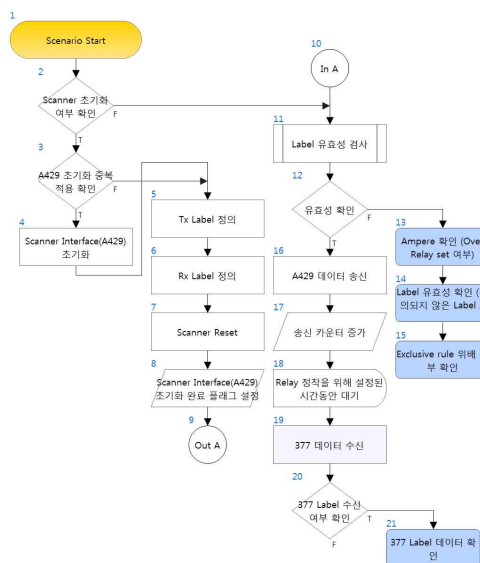


그림 5. Scanner 제어용 점검 시나리오
Fig. 5. Test scenario for scanner control.

3-2 자체점검/교정 모듈 개발

ATE의 UUT 점검은 ATE의 무결성이 전제되어야 의미가 있다. 따라서 비행체 ATE는 UUT 점검 전에 ATE의 자체점검/교정을 수행하여 무결성을 확보한다. 자체점검은 SBC와 계측기 등의 BIT 결과 확인, 자체점검 ITA를 통한 루프백 구조에서의 신호 경로 점검으로 구성하였다. 상용계측기의 자체 교정은 입고교정 판단을 위한 신호 형태 확인 기능과 입고교정 도래의 알림 기능이 포함되었다.

특히 Analog 신호 및 Calibrator의 신호 (current / TC(thermo couple) / RTD (Resistance temperature detector))는 신호 경로 상 유실되는 부분을 보상해주어야 한다. 또한 비행체 ATE는 Analog Card에서 지원하는 신호의 최대/최소치를 사용해야 하는 요구사항이 있다. 이에 대한 하드웨어 설계 방안으로 안정적인 신호 공급을 위해 Scanner 상에 신호 증폭기 및 감쇠기 (amplifier/attenuator)를 설치하였다. 즉, 점검 시나리오 상에서 설정하는 Analog 출력 값과 Scanner를 통해 ICA로 출력되는 값은 신호 유실과 증폭기 영향에 의해 상이하다. 때문에 점검 시나리오 상에서 설정하는 신호 값과 ICA로 출력되는 신호 값을 동일하게 해주는 보상 작업이 필요한데, 이 보상 과정을 신호 조정 (adjustment)이라 한다. 이 신호 보상 과정을 수행하기 위해서는, 조정 대상 신호가 ICA로의 출력되는 것 외에도 DMM으로도 유입될 수 있도록 경로를 전환 (switching) 할 수 있어야 하며 비행체 ATE에서는 Scanner에서 이를 수행한다.

Analog 신호의 경로 보상은 다음과 같이 진행하였다. 샘플링 구간 데이터가 20개 가량 (-10 V ~ 10 V, 1 V Step)이고 하드웨어적 결함이 발생된 게 아닌 이상, 큰 차이의 단일 오차는 일시적인 것으로 가정하였다. 가정 조건에 따라 데이터 정밀도는 상대적으로 낮지만 효율적으로 신호 경향을 추출해낼 수 있는 최소자승법의 선형회귀 방법으로 추정하였다. 아래는 신호 조정에 따른 결과 차이이다. 단위는 mV 단위로 ATE의 모든 Analog Out 신호 채널에 대해 각기 수집된 값을 평균으로 비교하였다.

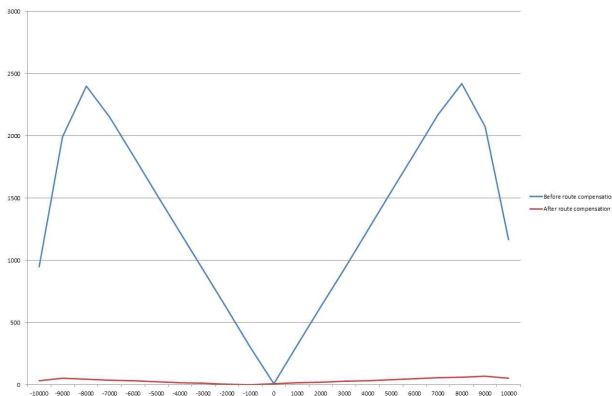


그림 6. Analog 신호의 경로 보상 전/후 비교
Fig. 6. Compare analog signal of path compensation between before and after.

청색선이 경로 보상 작업 전의 상태로 신호 유실 및 증폭기 영향을 받아 설정/실제 출력 신호가 다르다. 샘플링 구간 데이터 별 오차의 평균은 1.347 V 정도로 UUT 대상 시험에 적용하기 어려운 수준이다. 반면에 경로 보상 작업을 거친 적색선은 입/출력 신호의 수준 차이가 청색선에 비해 현저히 감소한다. 샘플링 구간 데이터 별 오차 평균은 0.033 V로 UUT의 일반 Analog 시험 수준에서의 허용 오차가 0.06 V (± 0.03 V) 이므로 목표 성능을 만족한다. 이보다 더 작은 허용 오차를 요구하는 UUT는 Calibrator를 이용하며, Calibrator 입력 신호에 대해서도 Analog 신호와 같이 신호 경로 보상 절차를 거친다.

3-3 실시간 점검 모듈 개발

비행체 ATE는 실시간으로 점검절차를 수행한다. 음성/영상 중개장치에 대한 점검은 다중 실시간 처리의 대표적인 예로, ICD 연동은 Ethernet, Serial을 각기 처리하고, 50 Hz 주기의 음성 데이터 중개 작업과 주요 절차를 수행하는 작업까지 총 4개의 Task 처리하여야 한다. 이는 점검 시나리오 주요 절차 Task에서 각 Task를 생성하고, 통신/음성 데이터 처리 Task의 결과를 주요 절차 Task 버퍼에 저장할 수 있도록 설계하였다. 구동기의 대역폭 (Bandwidth) 성능 점검은 500 Hz 주기로 사인과 신호를 구동기에 인가하여 확인한다. 로직은 상태 다이어그램 기반의 C 코드로 작성되었다. 5개의 상태로 구분하고 주요 상태에는 사인과 생성 및 위상 (Phase) 조건 확인 (zero crossing 탐지), 크기 (Magnitude) 데이터 수집 (조향각도 최대치 탐지)을 수행하고 주파수를 점차 증감하여 결과 값을 도출한다. 계단파 신호의 경우 1 KHz 주기로 생성/점검을 완료한 바 있다.

```

case MAKE_SINE_WAVE:
    // Set the Command
    // Amplitude = sin(time(sec) * 2π * SineFrequency(0.5Hz~5.1Hz))
    Steering_CMD = (unsigned short)((AMPLITUDE
        * sin((iFrameTime / CONVERT_UNIT_MS_TO_S)
        * 2u * M_PI * dFreqFrequency)) * RESOLUTION);

    // Phase check
    // Echo, Zero crossing(-) -> (+), <= 3cycle :
    // save the current time in freq time table
    if ((ZERO_CROSSING_STD > dFreqPreCommandEchoDeg) &&
        (ZERO_CROSSING_STD <= dSteeringCMDEchoFilter) &&
        (CYCLE_LIMIT_V > uiFreqCommandEchoIndex)) {
        aiFreqTimeTable[uiFreqCommandEchoIndex++] = iFrameTime;
    }

    // Status, Zero crossing(-) -> (+)
    if ((ZERO_CROSSING_STD > dFreqPreResponseDeg) &&
        (ZERO_CROSSING_STD <= dSteeringStatFilter)) {
        // duplicated table between Status index and Command Echo index.
        // Echo - Status, Phase time
        if ((TABLE_INIT_V != aiFreqTimeTable[uiFreqResponseIndex]) &&
            (uiFreqCommandEchoIndex > uiFreqResponseIndex) &&
            (CYCLE_LIMIT_V > uiFreqResponseIndex)) {
            aiFreqTimeTable[uiFreqResponseIndex++] =
                iFrameTime - aiFreqTimeTable[uiFreqResponseIndex];
        }
    }

    // Save the previous value for detect the Zero crossing
    dFreqPreCommandEchoDeg = dSteeringCMDEchoFilter;
    dFreqPreResponseDeg = dSteeringStatFilter;
    // Collect the Magnitude data, Max Echo
    if (dSteeringCMDEchoFilter > dFreqMaxCommandEchoDeg) {
        dFreqMaxCommandEchoDeg = dSteeringCMDEchoFilter;
    }

    // Collect the Magnitude data, Max Status
    if (dSteeringStatFilter > dFreqMaxResponseDeg) {
        dFreqMaxResponseDeg = dSteeringStatFilter;
    }

    // Check the timing for change to next frequency
    if (CYCLE_LIMIT_V <= uiFreqResponseIndex) {
        s_eStepFormWaveProcState = NEXT_FREQUENCY;
    }
    break;

```

그림 7. 대역폭 점검 시나리오 (사인파)
Fig. 7. Bandwidth of test scenario(sine wave).

3-4 GUI 연동 구조 개발

ATES 소프트웨어는 점검자가 ATE를 운용할 수 있도록 편의 기능을 제공하는 GUI이다. 점검자의 편의성 기반으로 구성되므로 소프트웨어 개발단계에서 점검자의 요구사항이 반영되어 공용화가 쉽지 않은 편이다. GUI는 점검자의 요구사항과 타 점검장비와의 유사성 확보를 위해 설계되었지만, 앞서 서술한 UUT의 설계 변화 영향성에 대해서는 GUI와 분리시켜야 효율적인 설계라 할 수 있다. 비행체 ATE에서는 점검 시나리오 XML 파일을 활용하여 태그로 구분, 전시/추가 기능을 수행한다. 점검 시나리오와 ATE 소프트웨어 간 프로토콜은 주로 GUI 전시 방법 정의, 에러 발생에 대한 예외 처리 정의, UUT 점검에서 필요한 Host PC에서의 파일 실행(음성 재생) 등의 부가 기능들로 구성되어 있다. ATE 소프트웨어에서는 해당 프로토콜로 정의된 데이터를 태그로부터 가져와 파싱하여 실행하는 형태이다. TestNginx 도구의 보완사항으로, 기존엔 심볼의 명령을 SBC 내 커널로 송신하고 해당 응답을 받는 구조였다. 보완된 구조는 점검 시나리오를 모두 SBC로 다운로드한 후 실행 명령을 받아 내부에서 처리하며, 상태 정보 발생 시 송신하는 형태로 변경되었다. 점검 시나리오에서는 ATE 전시 측면과 성능을 고려하여 필수적인 응답만을 받을 수 있도록 설계하여야 한다.

표 2. 점검 시나리오와 ATE 소프트웨어 간 프로토콜
Table 2. Protocol between test scenario and ATE SW.

No.	Protocol	explanation
1	MN_UUT ##	Main procedure-detailed procedure classification display
2	SRU	Display for identified fault SRUs (Including link to manual)
3	GUI_RUN	Execute the program
4	Hide	Hide the symbol
5	LB_UUT	Displayed by deleting keyword
6	TimeoutMsg	Sets a warning message when processed with timeout events
7	GUI_APPLYRE SULT	Pass even if symbol status is fail
8	GUI_TIMER	Display the progress bar (The time assigned to the property)
9	EnterProductNo	Provides a list of options for the user
10	Condition(assign)	display LV(measured)/RV(Predicted)
11	FileDialog	Provide the user with a file open dialog
12	Substitution	Modify label of user input menu
13	LINEFEED	Exhibit new line (\n)
14	AudioIN, OUT	Control the audio input/output (Host PC Sound card)
15	EndScenario	Forcibly terminates a running scenario
16	PING	Ping (Host PC)
17	NameFormat	Display internal data of the scenario
18	NameRegister	Define the display format
19	GETCHECKED	Select the test scenario
20	GUIDISPLAY	Set the display format of the test scenario internal data

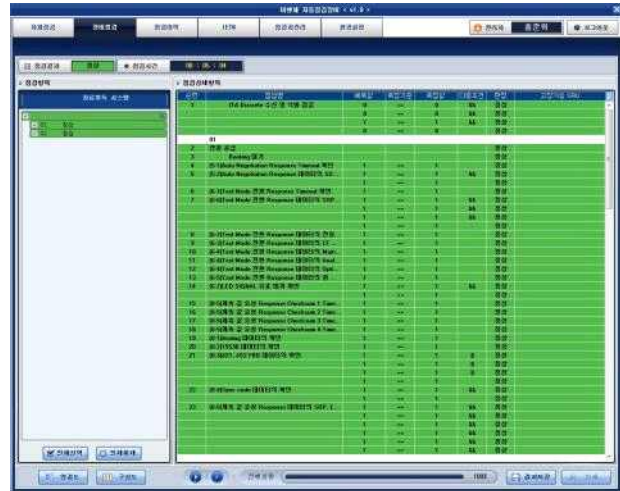


그림 8. 비행체 ATE GUI 예시
Fig. 8. Example of Aircraft ATE GUI.

IV. 결 론

본 논문에서는 비행체 ATE 개발 과정을 통해 유지보수성 향상 방안이 반영된 과정을 살펴보았다. 조금 더 고찰해봐야 할 점은 실시간 점검 사례에서 짚었던 사인파 생성이다. 스케줄링에 맞춰 동작해야 하는 로직으로 인해 순서도가 아닌 C 코드 레벨로 작성되었다. 이는 유지보수성을 위하여 개선이 필요하다. 추출할 수 있는 모듈은, 사인파/계단파 생성의 공통적인 요소(주파수, 신호 크기, 위상, 진폭)에 대해 파라미터로 식별하고 조건에 부합하는 원시 데이터를 출력해주는 모듈이다. 해당 모듈을 활용할 때는 원시 데이터에 대해 환산계수를 반영하여 ICD 프로토콜 기반으로 송신하는 형태이다. 한 가지 더 생각할 점은 보안 문제이다. TestNginx 개발 도구를 활용하거나, 직접 XML 파일을 수정할 경우 점검 시나리오 내용이 수정되기 때문이다. 이 점은 CRC (Cyclical redundancy check)를 활용해 유효성을 체크하는 방법 또는 XML 파일을 암호화하는 방안이 있겠다. 추가적으로, 비행체 ATE는 CMMI (capability maturity model integration) 인증 프로세스에 준하여 개발되었고, 기능/성능 시험을 수행하였다. 품질에 대한 평가 중 사용성에 대한 지표는 수립 되지 않았는데, 향후 이에 대한 보완예정이다[4].

Acknowledgement

본 연구는 국방과학연구소의 지원에 의하여 이루어진 연구(UC130030JD)로서, 관련기관에 감사드립니다.

References

[1] H. G. Park, "Study on design of ATE(auto test equipment for KUH(Korean utility helicopter) IDMC(integrated digital map computer)," *The Korean Institute of Electrical Engineers Information and Control Symposium*, Boryeong, pp.194-195, Oct. 2013.

[2] D. H. Cho, C. Y. S. Lim, Y. S. Yoon, and S. J. Kang, "The study on standardization and re-use for ATE design in production phase," in *The Institute of Electronics Engineers*

of KOREA Symposium, Busan, pp.1062-1065, Jun.2017.

[3] R. H. Lee, "The study on the optical test equipment for common use," in *The Institute of Electronics Engineers of KOREA Symposium*, Busan, pp.954-955, Jun.2017.

[4] Y. H. Yoon, K. Y. Ku, J. J. Keum, U. H. Hwang ,and S. Woo, "The study on improvement of ATE reliability in production phase," *The Institute of Electronics Engineers of Korea –System and Control*, Vol. 47, No. 6, pp.19-26, Nov. 2010.



서민기 (Min-gi Seo)

2012년 02월 : 한국항공대학교 컴퓨터공학과 (공학사)
 2011년 10월 ~ 현재 : LIG넥스원 항공연구소 선임연구원
 ※ 관심분야 : 항공전자, 내장형 소프트웨어



권기용 (Ki-yong Kwon)

2011년 08월 : 충남대학교 컴퓨터공학과 (공학석사)
 2011년 09월 ~ 2016년 12월 : 국방과학연구소 연구원
 2017년 03월 ~ 현재 : LIG넥스원 항공연구소 선임연구원
 ※ 관심분야 : 항공전자, 내장형 소프트웨어



김성우 (Seong-woo Kim)

2002년 08월 : 부산대학교 정보통신공학과 (공학석사)
 2002년 10월 ~ 현재 : LIG넥스원 항공연구소 수석연구원
 ※ 관심분야 : 실시간 시뮬레이션 기법 및 시험환경 응용 개발



이성우 (Seong-woo Lee)

2001년 02월 : 홍익대학교 전자공학과 (공학사)
 2001년 01월 ~ 현재 : LIG넥스원 항공연구소 수석연구원
 ※ 관심분야 : 항공전자, 임베디드 컴퓨터 하드웨어