P R O J E C T D O C U M E N T

Test Plan / Test Cases Design Document

Project Name	인공위성 통신 시뮬레이터의 공격 시나리오 구현

13 조

202002493 박민서
202002546 임우진
202002561 조영민

지도교수: 장진수 교수님

Table of Contents

1. INTRODUCTION 3

1.1. OBJECTIVE 3

2. TEST PLAN 4

3. TEST CASES 8

4. AI 도구 활용 정보 10

1. Introduction

1.1. Objective

본 문서는 소형 인공위성 통신 시뮬레이터 프로젝트의 테스트 계획(Test Plan)과 테스트 케이스 설계(Test Cases Design)를 서술형으로 정리한 것입니다. 해당 시뮬레이터는 NASA의 소형위성 시뮬레이션 프레임워크인 NOS3와 Core Flight System(cFS)을 기반으로 하며, Ball Aerospace의 COSMOS 소프트웨어를 지상국 GUI로 활용하고 3D 시각화 도구를 포함한 통합 환경에서 구동됩니다. 본 문서에서는 시뮬레이터의 주요 기능별로 테스트 항목을 정의하고, 각 기능 단위에 대한 구체적인 테스트 절차와 기준을 기술합니다.

테스트 계획의 목적은 인공위성 통신 시뮬레이터 시스템의 기능적 완전성과 기능 요구사항 충족 여부를 검증하는 데 있습니다. 본 문서는 테스트 전략과 범위를 명확히 하여 개발팀과 테스트 인력이 공통된 이해를 가지고 테스트를 수행할 수 있도록 돕습니다. 이를 통해 GUI, 텔레메트리 처리, 명령 처리, 위성 설정 등 각 구성 요소가 의도된 대로 작동하고, 채널 및 통신 메커니즘이 요구사항에 부합하는지 확인합니다. 더 나아가 잠재적인 결함을 사전에 발견하고 보완하여, 최종 시스템이 신뢰성이 확보된 상태로 인도될 수 있도록 하는 것이 궁극적인 목표입니다.

2.Test Plan

1. 배경과 목적
1.1 배경
저희 인공위성 통신 시뮬레이터는 실제 위성 통신 환경을 소프트웨어 시뮬레이션으로 재현하고 다양한 사이버 공격 시나리오를 실험하기 위해 개발되었습니다. 이 시스템은 저궤도 위성 통신에서 발생하는 전파 지연, 도플러 효과, 경로 손실 등 물리 현상을 모델링하고, 통신 채널에 대한 재밍(jamming), 스푸핑(spoofing), 중간자 공격(MITM) 등의 공격 시나리오를 구현하는 것이 특징입니다.
1.2 테스트 목적
테스트 계획 수립의 목적은 이러한 복잡한 시뮬레이션 시스템의 기능별 검증을 체계적으로 수행하여, 모든 구성 요소가 통합된 환경에서 제대로 작동하고 기능 요건을 충족함을 입증하는 데 있습니다. 또한 본 테스트를 통해 시스템의 약점을 식별하고 개선 사항을 도출함으로써, 향후 실제 위성 통신 환경 적용 시 위험요소를 최소화하고자 합니다.
2. 테스트 상세
2.1 테스트 항목
시스템의 **기능 단위**를 기준으로 다음과 같은 항목들에 대해 테스트를 수행합니다. 각 항목별로 상세한 테스트 케이스를 설계하여 기능의 올바른 동작 여부와 보안 측면을 검증합니다.
2.2 테스트될 요소(features)
■ GUI 기능 ■ 텔레메트리 처리 ■ 명령 처리 ■ 위성 설정 ■ 3D 시각화 ■ 채널 설정 및 모델링 ■ 공격 시나리오
2.3 테스트되지 않을 요소
■ RF 하드웨어 연동 기능 ■ 성능 및 부하 테스트 ■ 기타 미구현 기능

2.4 접근 방법

테스트는 기능별 단위 테스트(Unit Test)에서 시작하여 **통합 테스트(Integration Test)**, 그리고 전체 시스템을 대상으로 한 **시스템 테스트(System Test)** 단계로 점진적으로 진행됩니다. 각 단계에서는 **블랙박스 테스트 기법**을 주로 활용하여, 내부 구현보다는 **입력-출력 및 동작 결과**를 검증합니다. 또한 이 프로젝트의 특성상 **시나리오 중심 테스트**를 병행하여 현실적인 사용 흐름을 점검합니다.

2.5 테스트 항목의 pass/fail 기준

각 테스트 케이스별로 예상 결과(Expected Result)를 미리 정의하고, 실제 테스트 수행 시 관측된 결과와 비교하여 Pass 또는 Fail을 결정합니다. Pass/Fail 기준은 다음과 같습니다.

Pass 기준:

1. 기능이 명세대로 동작함 - 입력에 대해 명세된 출력 또는 시스템 반응이 정확히 발생함 (예: 로그 출력, ACK 수신, GUI 갱신 등)
2. 공격 로직이 정확히 반응함 - 스푸핑, 재밍, 리플레이 공격 시나리오가 정상적으로 시행되어야함.
3. 시스템 에러나 예외가 없음 - 실행 중 크래시, 미응답(hang), 비정상 출력 없음
4. 시각적 요소 일치 - 3D 모델, GUI 뷰어에서 상태 변화가 예상대로 반영됨
5. 통신/패킷 상태 정상 - 패킷 포맷, 순서, 필드 값 등이 명세와 일치함 (예: 헤더·길이·체크섬 유효)
6. 성능 조건 충족 - SNR, 패킷 손실률 등 수치적 기준이 명시된 범위 내에 존재함.

Fail 기준:

1. 출력 누락 또는 오류 출력 - 명령 실행 후 로그 미출력, NACK이 나와야 할 상황에서 ACK 발생 등
2. 공격 기능 미작동 - 공격 기능이 정상적으로 시행 안되는 경우
3. GUI/뷰어 오류 - 화면 갱신 지연, 상태 미반영, 시각화 오류 발생
4. 프로세스 비정상 종료 - 시뮬레이터 또는 COSMOS가 중단, 충돌, 무한 루프 상태에 빠짐
5. 예상 값과 수치 불일치 - SNR 변화나 패킷 손실률이 모델링 시나리오와 일치하지 않음
6. 패킷 분석 결과 불일치 - Wireshark/로그에서 캡처된 패킷의 필드 값이 사양과 다름.

2.6 테스트 산출물(deliverables)

- ■ 테스트 계획서
- ■ 테스트 케이스 명세서
- ■ 테스트 결과 보고서

3. 테스트 관리

3.1 작업

- 테스트 프로젝트를 위해 수행할 작업을 기술함

중요한 테스트 작업은 다음과 같다.

1 테스트 계획을 개발한다.

2 테스트 인원을 구성한다.

3 시스템 요구 사항과 기능 명세를 검토한다.

4 테스트 케이스를 작성하고 테스트 절차를 개발한다.

5 테스트 계획, 테스트 케이스, 절차를 검토하고 승인한다.

6 상세한 테스트 계획에 따라 시스템 기능에 대하여 테스트를 수행한다.

7 발견된 결함을 보고한다.

8 결함을 수정한다.

9 수정된 내용에 대하여 재테스트를 수행한다.

10 테스트 결과를 문서화한다.

11 테스트 종료 조건을 기준으로 시스템을 릴리스 할 시점을 결정한다.

3.2 기술 자원

하드웨어 장비(Laptop)

가상 머신(VirualBox)

NOS3, cFS, GS 등의 프레임워크

사전 작성된 테스트 시나리오 스크립트 (Python)

Wireshark, tshark

결과 저장소 (공유폴더)

3.3 책임과 권한 (인력 자원)

세 명의 테스트 담당자는 모두 해당 시스템과 도구에 대한 기본 지식을 갖추고 있습니다. 이렇게 **테스트 팀**을 이루어 상호 검토하면서 테스트를 수행합니다.

인원	주요 역할 및 권한
임우진	- 테스트 계획 수립 및 테스트 케이스 설계 - 단위/통합 테스트 시나리오 정의- AI 도구(GPT 등) 활용 가이드라인 제공
박민서	- 테스트 환경 구축 및 자동화 스크립트 구현 - 테스트 실행 및 결과 기록- 패킷 캡처·분석, 로그 관리
조영민	- 결함 보고(Bug Report) 작성 및 협의 - 테스트 마일스톤 관리 및 주간·최종 보고서 작성- 리그레션 테스트 계획 및 재시험 주도

3.4 훈련

테스트 전에 테스트 인력에 대한 도구 사용 교육을 실시합니다. NOS3/cFS 환경 및 COSMOS 사용법에 익숙하지 않은 부분을 보완하기 위해, NASA cFS 훈련 자료와 COSMOS 사용자 매뉴얼을 기반으로 사전 학습을 완료했습니다. 특히 COSMOS 스크립팅, 텔레메트리 정의 구성 방법, NOS3 가상 하드웨어 시뮬레이션 개념 등에 대해 스터디 세션을 가졌습니다. 위성 통신 공격 시나리오 (재밍, 스푸핑 등)에 대한 개념 교육도 함께 이루어져, 테스트 중 어떤 현상

이 정상이고 비정상인지 판단할 수 있도록 하였습니다.
3.5 일정
테스트 일정은 전체 프로젝트 일정에 맞추어 단계별로 계획되었습니다. 테스트 케이스 개발 단계를 프로젝트 중반부에 수행하여 개발 완료와 병행 검증이 가능하도록 하였고, 주요 기능 구현이 완료되는 시점마다 해당 기능의 통합 테스트를 진행합니다. 약 **2주**간의 테스트 기간을 두어 전체 시나리오를 시험합니다.
3.6 위험 요소와 비상 대처 상황
소프트웨어 결함으로 인한 차질: 테스트 중 치명적인 버그(예: 시뮬레이터가 자주 다운되는 오류)를 발견하면 일정 지연이 발생할 수 있습니다. 대응: 발견 즉시 결함 보고서를 작성 후 수정 인력 및 일정: 두 명의 테스트 인력으로 모든 테스트를 소화해야 하므로 업무량이 많아질 수 있습니다. 대응: 테스트 자동화를 최대한 활용하고, 우선순위를 지정하여 필수 테스트를 우선 완료합니다.

3.Test Cases

1. 서론

1.1 테스트 범위

앞서 정의한 기능별 테스트 항목(2.2절)에 명시된 바와 같이, 시뮬레이터의 모든 주요 기능을 포함합니다. 텔레메트리 수신/표시, 명령 송신/처리, 설정 변경, 3D 뷰어 갱신, 채널 모델링, 패킷 처리, 공격 시나리오 등 현재 구현된 모든 기능에 대해 테스트 케이스를 설계합니다. 다만 2.3절에서 언급한 범위 외 항목(예: 실제 RF 통신, 성능 한계 등)은 본 테스트에서 제외됩니다.

1.2 테스트 상황

각자 구현/설계한 테스트를 타 인원과 상호 검토하여 테스트의 정확성을 높인다.

1.3 문서 표기법

본 문서의 테스트 케이스 명세에서는 통일된 서식을 사용합니다. 각 테스트 케이스는 “TC NN” 형태의 식별자와 함께 제목으로 표시하고, 테스트 대상, 테스트 조건, 테스트 데이터, 예상 결과의 형식으로 기술합니다.

2. 테스트 케이스

2.1 테스트 케이스 명세

ID	테스트 대상	테스트 조건	테스트 데이터	예상 결과
TC 01	GUI 기능	COSMOS GUI 실행 후 초기 상태 진입	위성 상태 조회, 명령 버튼 클릭 등	GUI 내 상태 정보 정확히 반영됨
TC 02	위성 설정 변경	시뮬레이터 기동 후 기본 설정 상태	위성 식별자 변경 등	설정 저장 후 로그에 CONFIG UPDATE: SatID=B 표시
TC 03	명령 전송 처리	Sat-GS 링크 연결 완료	명령	cFS 로그에 출력, ACK 수신
TC 04	텔레메트리 수신 및 표시	Sat-GS 간 정상 통신 채널 활성화	주기적 TELEMETRY_PKT	COSMOS 뷰어에 1초 간격으로 텔레메트리 표시
TC 05	3D 시각화 갱신	3D 뷰어 실행 중 기본 궤도 모델 표시됨	궤도 고도 변경	궤도 모델이 고도 업데이트
TC 06	채널 모델링 효과	채널 모델링 창 활성화 모델 라이브러리(gr-leo) 로드됨	도플러 ON/경로 손실 ON 시뮬레이션 실행	SNR 그래프에 도플러 편이·감쇠 패턴 표시 패킷 손실률 증가
TC 07	재밍 공격 효과 검증	채널 모델링 창 활성화	재밍 출력전력: 20	- SNR이 정상(30

		재밍 파라미터(출력전력, 지속시간)가 설정됨	dBm 지속시간: 5 s 패턴: continuous	dB)→재밍 중(≤10 dB)로 급격히 하락 - Packet Loss Rate가 0%→≥20%로 증가 -COSMOS에 “Signal Jam Detected” 경고 표시
TC 08	스푸핑 공격 검증	정상 명령 송신 경로 활성화 스푸핑 프록시가 동작 중	정상명령: CMD_ID=0x01, 정상 페이로드 위조패킷: CMD_ID=0xFF	- 정상 패킷 또는 성공한 위조 패킷: cFS에서 처리 → ACK 송신 - 위조 패킷: decryptPacket() 시도 → 검증 실패 → NACK 송신 - cFS 로그에 “Unauthorized command: 0xFF” 기록
TC 09	리플레이 공격 검증	정상 명령 전송 후 로그에 기록됨 리플레이 모듈 대기	재전송 패킷: 이전에 정상 전송된 CMD_ID=0x01 패킷	- 재전송된 명령: 중복으로 수신 → Duplicate Command Count 증가 (1→2) ※검증 로직 활성 시: 중복 탐지 후 거부 → NACK 송신

2.2 테스트 환경

Virtual Box 상에서 구동되는 NOS3 기반 가상 위성 시스템으로, cFS flight software가 시뮬레이션 모드로 실행되고 COSMOS 지상국 소프트웨어가 동일 네트워크 상에서 위성과 통신하도록 구성됩니다.

NOS3 패키지에 포함된 3D 시뮬레이터(42 Dynamic Simulator)는 위성 궤도/자세를 계산하여 가상 센서 및 3D 모델 데이터를 제공합니다.

COSMOS는 UDP/IP를 통해 cFS와 명령/텔레메트리 패킷을 주고받으며, 사용자는 COSMOS의 Command

and Telemetry GUI를 통해 시스템과 상호작용합니다. 테스트를 위해 필요한 사전 조건(각 테스트케이스마다 기술됨)을 충족하도록 **환경 초기화 스크립트**를 실행하고, 각 케이스 실행 전 위성 소프트웨어를 초기 상태로 리셋(reset)합니다.
2.3 테스트 절차 요구사항
테스트 시작 전, NOS3 가상 엔진, cFS 비행 소프트웨어, COSMOS 지상국 GUI를 순서대로 기동하고, 모두 “정상 대기 상태”가 되었음을 확인한다 상호 통신이 가능한지 ping 또는 heartbeat packet을 통해 점검한다. COSMOS 내장 로그 뷰어와 NOS3 콘솔 로그, 시스템 리소스 모니터링 툴(예: htop, netstat)을 동시에 열어둔다. 패킷 캡처 도구(Wireshark) 또는 COSMOS 패킷 로거 기능을 사전 활성화하여, 모든 테스트 중 주고받는 패킷을 기록할 수 있도록 설정한다.

4. AI 도구 활용 정보

사용 도구	OpenAI GPT-4
사용 목적	테스트 계획서 및 케이스 설계에 있어 문서 구조 수립과 사례 조사에 AI 도구를 활용하였습니다.
프롬프트	● “위성 시뮬레이터 테스트 계획 문서 예시를 알려줘” ● “cFS와 COSMOS 연동 테스트 시 고려사항이 뭐야?”
반영 위치	1. AI 도구를 통해 얻은 인사이트와 초안은 2. TEST PLAN의 세부 전략 수립(섹션 2.4 접근 방법 등)과 3. TEST CASES의 일부 시나리오 서술(특히 채널 모델링, 보안 공격 관련 테스트케이스) 작성에 반영되었습니다.
수작업 수정	AI가 생성한 초안 문장을 기반으로 하되, 프로젝트 고유의 용어 및 설정을 반영하기 위해 직접 내용을 검토하고 수정하였습니다. 기술 검토를 거쳐 확정한 내용이며, AI 도구는 이를 정리하는 데에만 활용되었습니다.