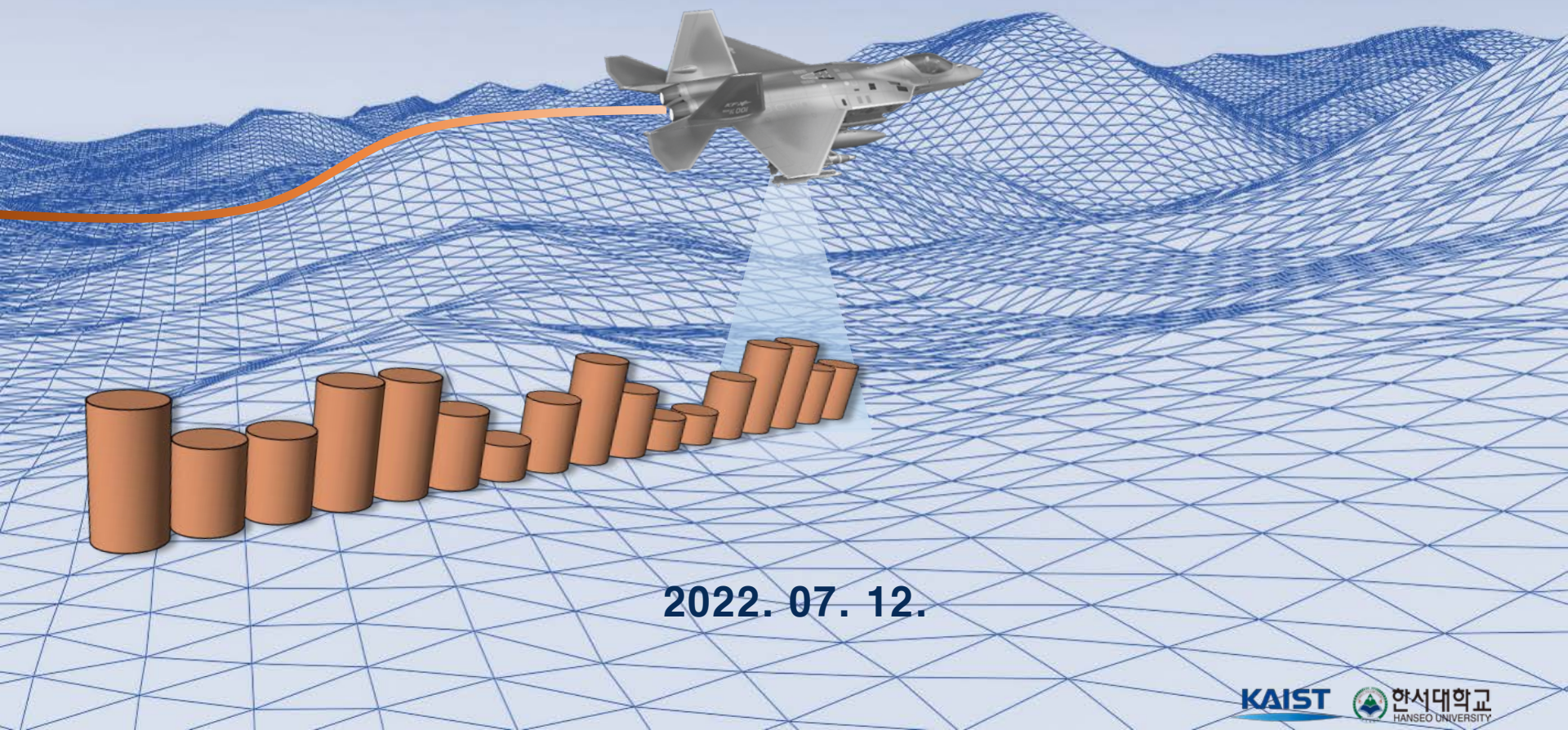


# 디지털지형항법 소프트웨어 FTG V2 실행방법 매뉴얼 (DTNS)



2022. 07. 12.

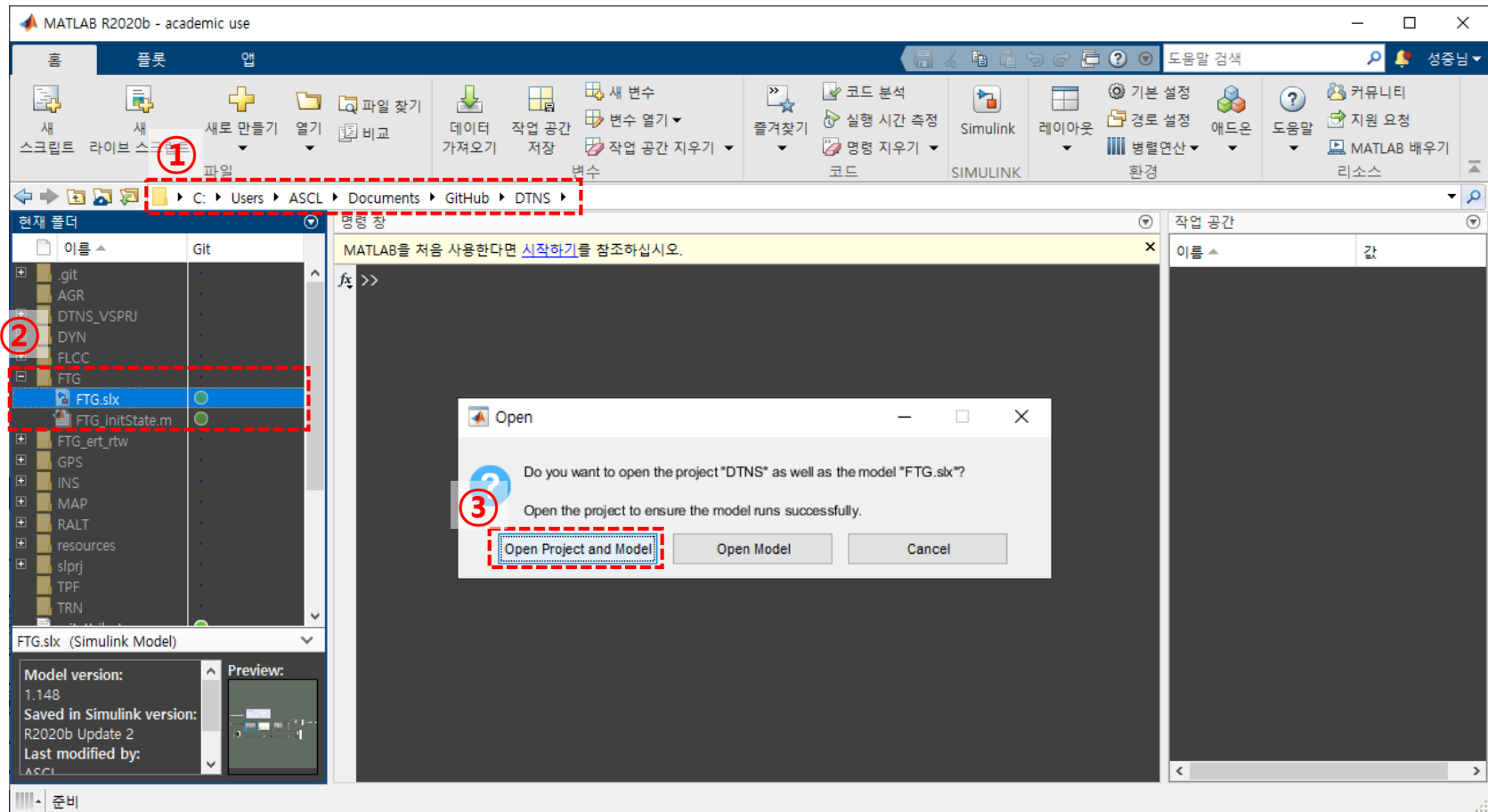
## 요약

- **오토파일럿 비행 모드 고도화**
  - 자세 유지, 고도 유지, 경로점 비행, ATF, AGCAS, PARS 등의 비행모드 분리
- **RALT 센서 모델 업데이트**
  - 출력 구조체 추가 (RALT 참값)
  - 측정치 오차 구현: 고도에 따른 백색 가우시안 잡음 추가
- **GPS 센서 모델 업데이트**
  - 입력 구조체 추가 (GPS 상태 제어를 위함)
  - 측정치 오차 구현: FOM 테이블에 근거한 간이 오차 모델링 (백색 가우시안 잡음)
- **INS 센서 모델 업데이트**
  - 입력 구조체 추가 (INS 상태 제어를 위함)
  - 측정치 오차 구현: Strap-down INS 알고리즘과 Aviation grade IMU을 이용한 오차를 가진 INS 구현

# 1. 실행방법

## • MATLAB 실행

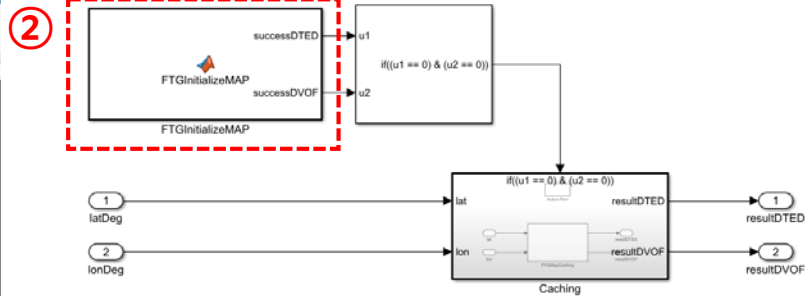
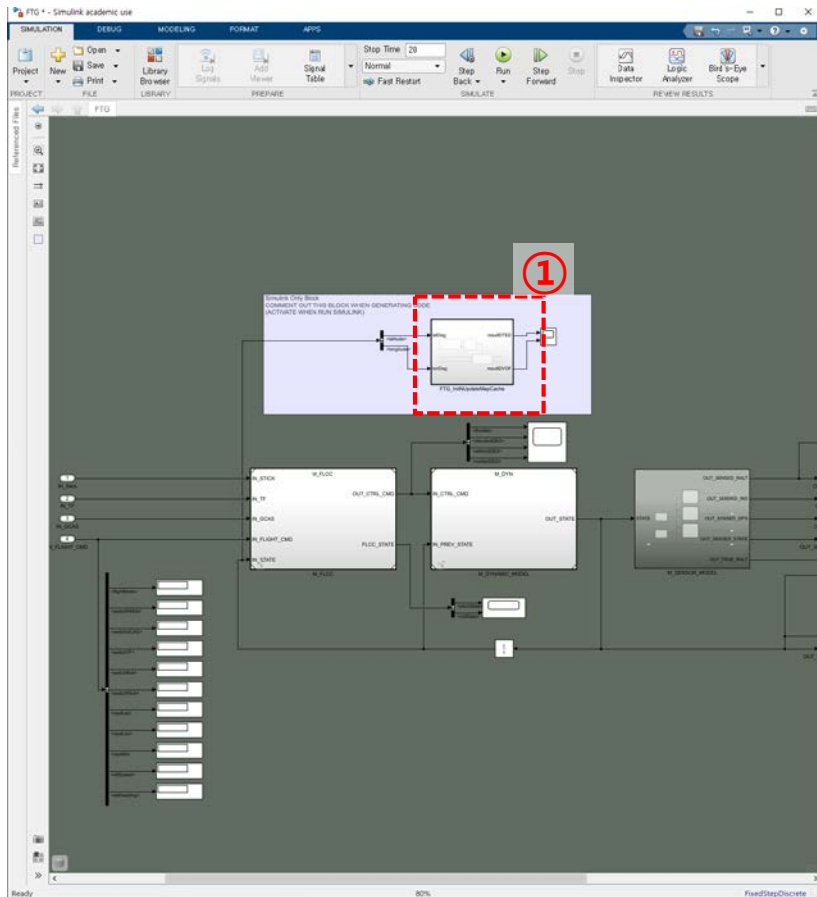
- 1) MATLAB 현재 경로를 DTNS(FTG) 프로젝트 경로로 설정
- 2) 하위폴더(\FTG) 선택하여 FTG.slx 실행
- 3) “Open Project and Model” 선택



# 1. 실행방법

## • DTNS MAP 폴더 설정

- 1) SIMULINK 프로젝트 상에서 MAP 데이터에 접근하기 위해 DTNS MAP 데이터가 저장된 폴더를 설정해주어야 함
- 2) **FTGInitializeMAP()** 함수의 pathDTED와 pathDVOF를 수정
- 3) SENSOR\_RALT 작동을 위해서는 RALT 모델 안의 **RALTInitializeMAP()**도 수정



```
function [successDTED, successDVOF] = FTGInitializeMAP()

flagDTED = int32(1);
flagDVOF = int32(1);

pathDTED = 'C:\\Users\\ASCL\\Documents\\GitHub\\DTNS\\MAP\\MAP_DB\\';
pathDVOF = 'C:\\Users\\ASCL\\Documents\\GitHub\\DTNS\\MAP\\DVOF\\';

strLenDTED = int32(strlength(pathDTED));
strLenDVOF = int32(strlength(pathDVOF));

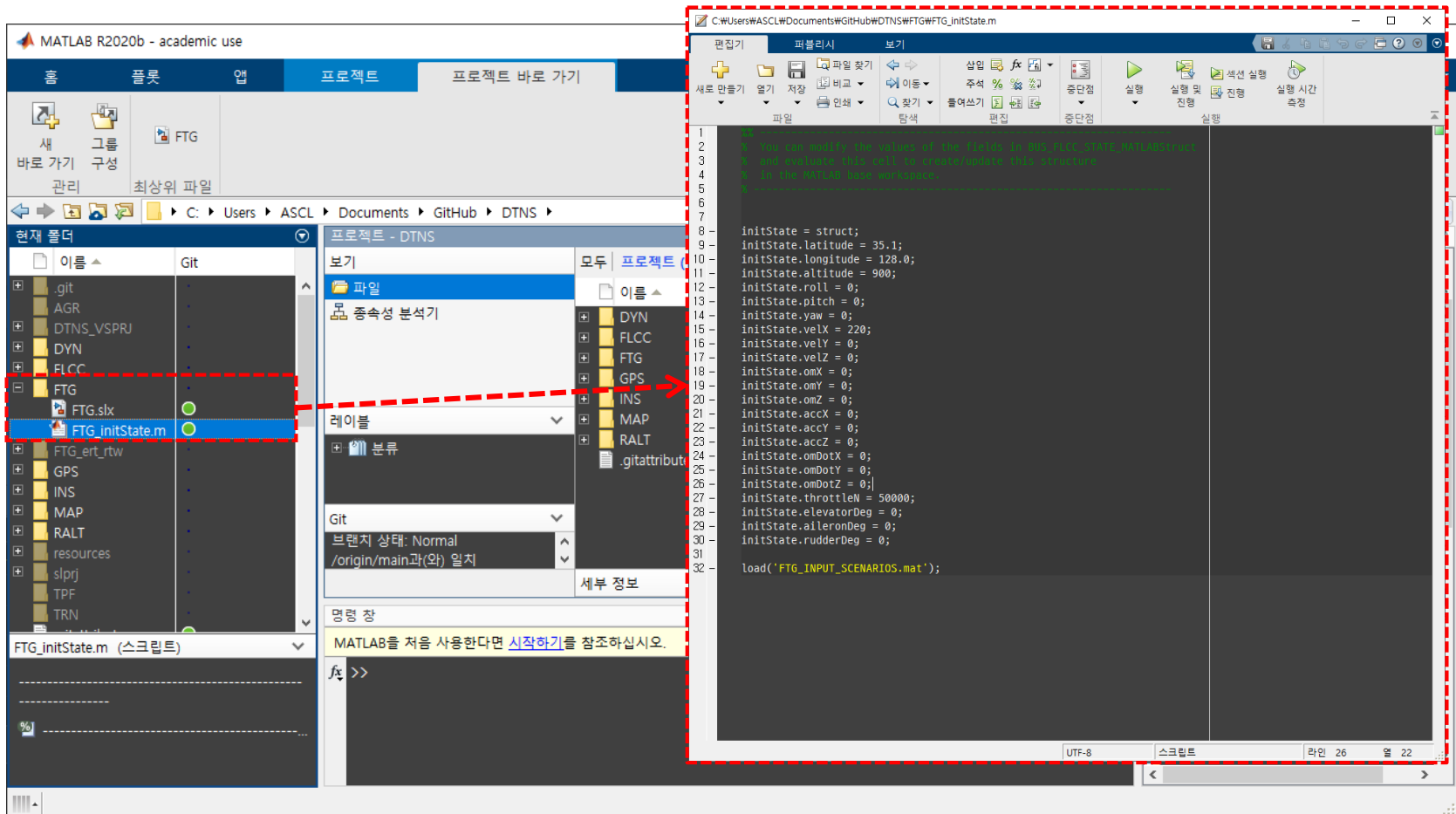
flagDTED = coder.ceval('InitialMap', pathDTED, strLenDTED);
flagDVOF = coder.ceval('InitDVOF', pathDVOF, strLenDVOF);

successDTED = flagDTED;
successDVOF = flagDVOF;
```

# 1. 실행방법

## • 초기 항공기 상태값 제어

- 1) FTG\_initState.m 스크립트 수정 후 재실행
- 2) 저장해둔 FTGScenario 파일을 이곳에서 불러올 수 있음
- 3) FTGScenario 파일은 FTG.slx의 입력변수





# 1. 실행방법

- FTG.slx 입력변수 조작
  - 1) MATLAB 작업공간 확인
  - 2) 변수 탭 확인
  - 3) 수정 원하는 입력변수 선택하여 시계열 데이터 입력

The screenshot shows the MATLAB R2020b - academic use interface. The main window displays the '변수' (Variables) tab for the 'FTGScenario' variable. A red dashed box highlights the '1x1 Dataset' section, which contains a table of input variables. A red circle with the number '2' is placed over the 'Dataset' header. Another red circle with the number '1' is placed over the 'FTGScenario' variable in the '작업 공간' (Workspace) panel on the right.

인덱스	값	이름	BlockPath	클래스
1	1x1 struct	IN_Stick		struct
2	1x1 struct	IN_TF		struct
3	1x1 struct	IN_GCAS		struct
4	1x1 struct	IN_FLIGHT_CMD		struct

Workspace (작업 공간):

이름	값
BUS_DYN_STATE	1x1 Bus
BUS_FLCC_CTRL_CMD	1x1 Bus
BUS_FLCC_FLIGHT_CMD	1x1 Bus
BUS_FLCC_FLIGHT_CM...	1x1 struct
BUS_FLCC_GCAS_CMD	1x1 Bus
BUS_FLCC_STICK	1x1 Bus
BUS_FLCC_TF_CMD	1x1 Bus
BUS_SENSOR_GPS	1x1 Bus
BUS_SENSOR_INS	1x1 Bus
BUS_SENSOR_RALT	1x1 Bus
FTGScenario	1x1 Dataset
InitState	1x1 struct
listing	27x1 struct

# 1. 실행방법

- FTG.slx 입력변수 조작
  - 1) MATLAB 작업공간 확인
  - 2) 변수 탭 확인
  - 3) 수정 원하는 입력변수 선택하여 시계열 데이터 입력

The screenshot shows the MATLAB R2020b - academic use interface. The 'Variables' tab is active, displaying the 'FTGScenario(4).targetLat' variable. The 'Time Series Name' is 'targetLat'. The 'Time' column shows values 0 and 30, and the 'Data' column shows values 35.1000 and 35.1000. A red dashed box highlights the 'Data' column, and a red circle with the number 3 is placed next to it, indicating the step to input time series data.

현재 폴더: C:\Users\ASCL\Documents\GitHub\DTNS

프로젝트 - DTNS

변수 - FTGScenario(4).targetLat

FTGScenario(4).targetLat

시계열 이름: targetLat

시간	데이터:1
0	35.1000
30	35.1000

③

☐ 이벤트 테이블 표시

특성(Attribute)... 행 추가 행 삭제

현재 시간: 등간격 0~30 초 등간격 시간 벡터...

명령 창

MATLAB를 처음 사용한다면 [시작하기](#)를 참조하십시오.

FTG.slx (Simulink Model)

Model version: 1.148  
Saved in Simulink version: R2020b Update 2  
Last modified by: ASCL

작업 공간

이름	값
BUS_DYN_STATE	1x1 Bus
BUS_FLCC_CTRL_CMD	1x1 Bus
BUS_FLCC_FLIGHT_CMD	1x1 Bus
BUS_FLCC_FLIGHT_CM...	1x1 struct
BUS_FLCC_GCAS_CMD	1x1 Bus
BUS_FLCC_STICK	1x1 Bus
BUS_FLCC_TF_CMD	1x1 Bus
BUS_SENSOR_GPS	1x1 Bus
BUS_SENSOR_INS	1x1 Bus
BUS_SENSOR_RALT	1x1 Bus
FTGScenario	1x1 Dataset
initState	1x1 struct
listing	27x1 struct

# 1. 실행방법

- FTGScenario{4}: IN\_FLIGHT\_CMD 설정
  - 1) 시험 목적에 맞게 FTGScenario 구성하여 사용
  - 2) 기본 제공 시나리오: FTG\_INPUT\_SCENARIOS\_STRG.mat
    - 북위 35.7도, 동경 128.25도, 1000m, 250m/s를 목표로 하는 경로점 비행
  - 3) 아래의 TRN 시험시 입력값 설정 추천값과 다음 쪽의 오토파일럿 상태 기계를 참고

파라미터	추천값	비고	파라미터	추천값	비고
flightMode	1	자동 비행모드	nextLat	~	경로점 입력
switchPARS	0	(=1, 즉시 회복기동)	nextLon	~	경로점 입력
switchGCAS	0, 1	OFF 혹은 수동 GCAS	nextAlt	~	경로점 고도가 아닌, 정확히는 설정 고도임
swtichTF	0, 1	OFF 혹은 수동 TF	refSpeed	100~340 m/s	
switchRoll	1	경로점 비행	setRoll	~	경로점 비행시 불필요
switchPitch	2	경로점 비행 (고도유지)	setPitch	~	경로점 비행시 불필요
			setHeading	~	경로점 비행시 불필요




TRN 시험시 입력값 설정



# 1. 실행방법

## 참고) 오토파일럿 상태기계

 피치축 오토파일럿 상태

값	상태	설명	PARS	GCAS	TF	Roll	Pitch
0	OFF	오토파일럿 미작동	0				0, (1,2)
1	ATT HOLD	입력받은 피치자세각을 유지한다	0		0, 1	0	1
2	ALT HOLD	입력받은 고도를 유지한다	0		0, 1		2
3	ATF	DBTF의 수직가속도 명령을 추종한다	0		2		
4	AGCAS	AGCAS의 회복기동 절차를 수행 중이다	0	2*			
5	PARS	PARS 절차를 수행해 롤자세 및 수평을 회복한다	1				

 롤축 오토파일럿 상태

값	상태	설명	PARS	GCAS	TF	Roll	Pitch
0	OFF	오토파일럿 미작동	0			(1, 2)	0, (1)
1	STRG SEL	입력받은 경로점으로 진행한다	0			1	2
2	ATT HOLD	입력받은 롤 자세각을 유지한다	0			0	1, 2
3	HDG SEL	입력받은 기수방위각을 추종한다	0			2	2
4	AGCAS	AGCAS의 회복기동 절차를 수행 중이다	0	2*			
5	PARS	PARS 절차를 수행해 롤자세 및 수평을 회복한다	1				

\* AGCAS is invoked when Auto-GCAS is on and PGCAS Time-to-go is less than pullUpTimeSec

### 스위치(PARS)

값	상태	설명
0	NORM	미입력 상태
1	PARS	PARS 절차 수행 트리거

### 스위치(GCAS)

값	상태	설명
0	NOT SELECTED	PGCAS 미선택
1	SELECTED	PGCAS 선택 (STBY)
2	AUTO GCAS	PGCAS 선택 및 Auto-GCAS 활성화

### 스위치(TF)

값	상태	설명
0	NOT SELECTED	DBTF 미선택
1	SELECTED	DBTF 선택 (STBY)
2	AUTO TF	DBTF 선택 및 Auto-TF 활성화

### 스위치(Roll)

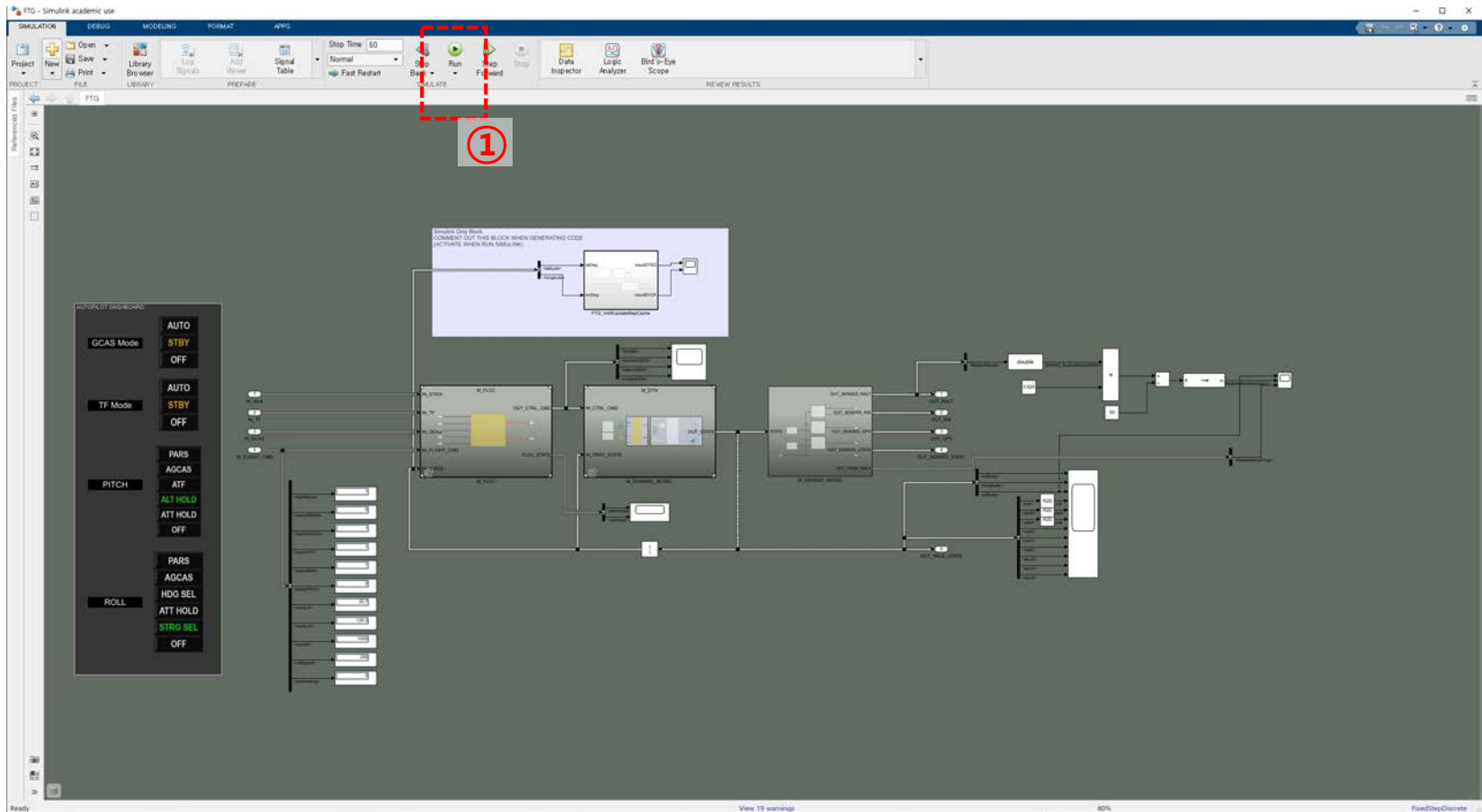
값	상태	설명
0	ATT HOLD	입력 롤 자세 유지
1	STRG SEL	경로점 비행
2	HDG SEL	입력 기수방위각 추종

### 스위치(Pitch)

값	상태	설명
0	AP OFF	오토파일럿 비활성화
1	ATT HOLD	입력 피치자세각 유지 스위치
2	ALT HOLD	입력 고도 유지 스위치

# 1. 실행방법

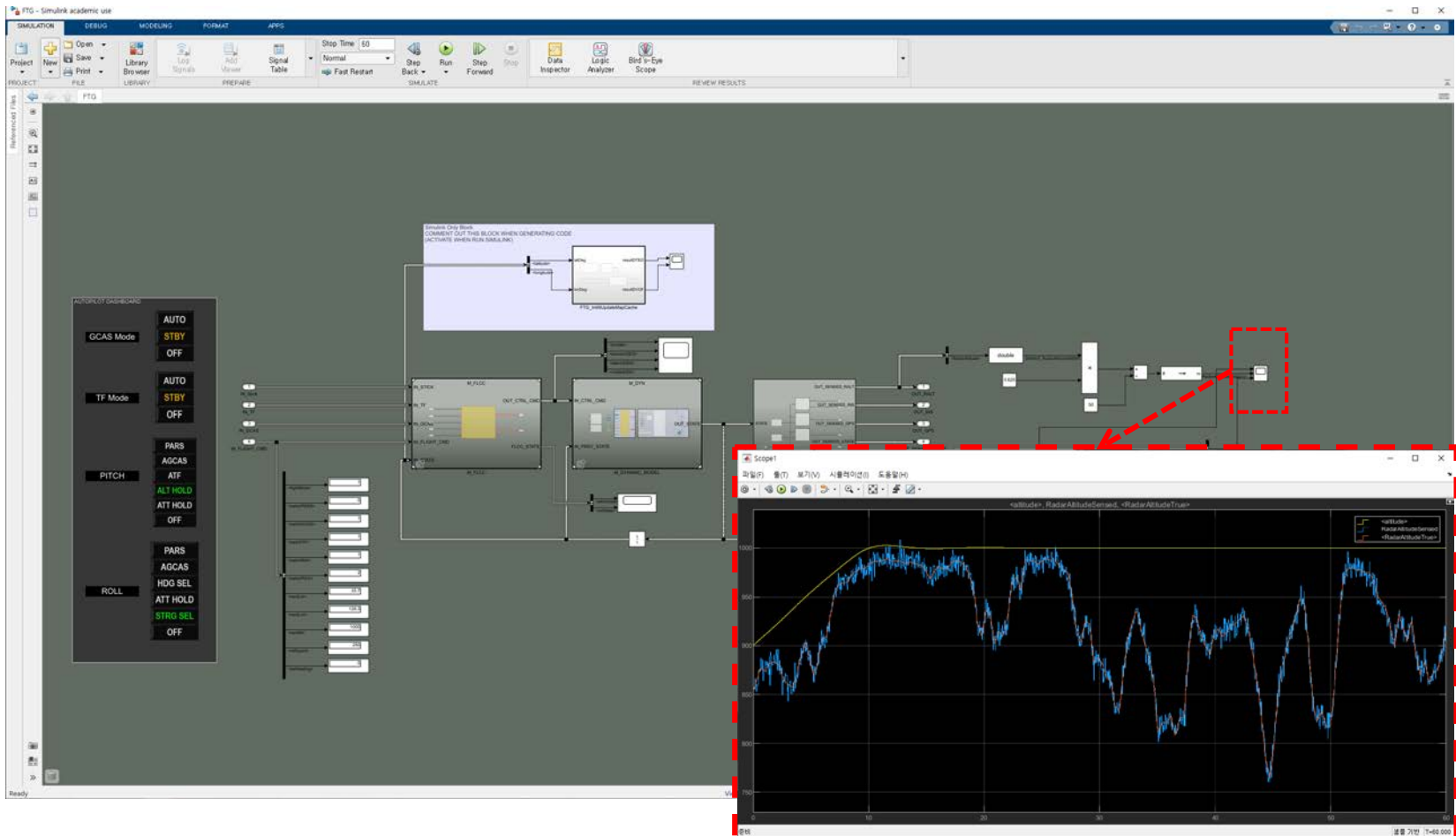
- FTG.slx 최상위 실행 화면
  - 실행 버튼 확인
  - 최초 실행시 컴파일에 1~2분 시간이 소요될 수 있음



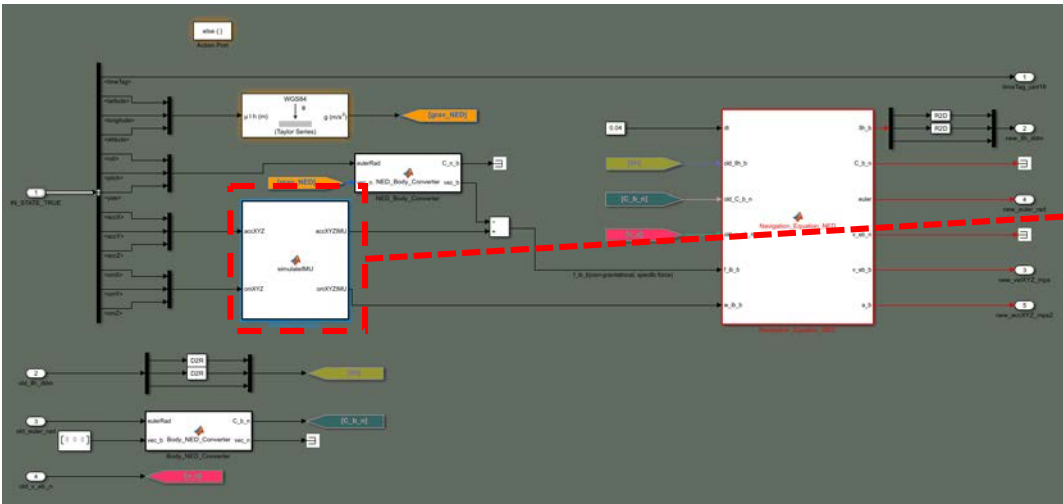
# 1. 실행방법

- 실행결과 확인

- 1) bus selector 블록으로 원하는 결과 데이터 추출
- 2) scope 블록으로 결과 데이터 그래프 도시



- INS 오차 수준은 SNU-84-1을 만족하는 Aviation-grade IMU로 모델되었음
- 시험시 필요에 따라 M\_INS 내 simulateIMU 함수를 수정



Block: M\_JNS/If Action Subsystem/MATLAB Function

편집기 보기

새로 만들기 열기 저장 파일 찾기 비교 이동 삽입 주석 % % % % 중단점 Run Model Stop Model Build 중단점 RUN

```

1 function [accXYZIMU, omXYZIMU] = simulateIMU(accXYZ, omXYZ)
2 % This function generates IMU data
3 % Inputs:
4 % accXYZ
5 % omXYZ
6 % Outputs:
7 % accXYZIMU
8 % omXYZIMU
9
10 % Accelerometer Noise (m/s^2)
11 muG = 9.8E-6; % (m/s^2)
12 accel_noise_root_PSD = 30 * muG; % (m/s^2)
13 gyro_noise_root_PSD = 0.003 * (pi/180) / 60; % (rad/s)
14 tor_i = 0.04; % (rad)
15
16 % Accelerometer Bias (m/s^2), converted to m/s^2 body axis
17 b_a = [80; -100; 70] * muG;
18
19 % Gyro Bias (rad/s), converted to rad/s body axis
20 b_g = [-0.009; 0.0013; -0.008] * (pi/180) / 3600;
21
22 % Accelerometer scale factor and cross coupling errors type, converted to m/s^2 body axis
23 M_a = [500, -300, 200;
24        -150, -600, 250;
25        -250, 100, 450] * 1E-6;
26
27 % Gyro scale factor and cross coupling errors type, converted to rad/s body axis
28 M_g = [400, -300, 250;
29        0, -300, -150;
30        0, 0, -350] * 1E-6;
31
32 % Gyro scale factor and cross coupling errors type, converted to rad/s body axis
33 G_g = [0.9, -1.1, -0.6;
34        -0.5, 1.9, -1.6;
35        0.3, 1.1, -1.3] * (pi/180) / (3600 * 9.8);
36
37 % Generate IMU data
38 accel_noise = randn(3, 1) * accel_noise_root_PSD / sqrt(tor_i);
39 gyro_noise = randn(3, 1) * gyro_noise_root_PSD / sqrt(tor_i);
40
41 % Generate IMU data
42 accXYZIMU = b_a + (eye(3) + M_a) * accXYZ + accel_noise;
43 omXYZIMU = b_g + (eye(3) + M_g) * omXYZ + G_g * accXYZIMU + gyro_noise;
44
45
46
47 end

```