MODEL RELEASE NOTE

1. 모델 버전	2. 배포 일자 (YYMMDD)
DTNS FTG V5.0	23.06.30

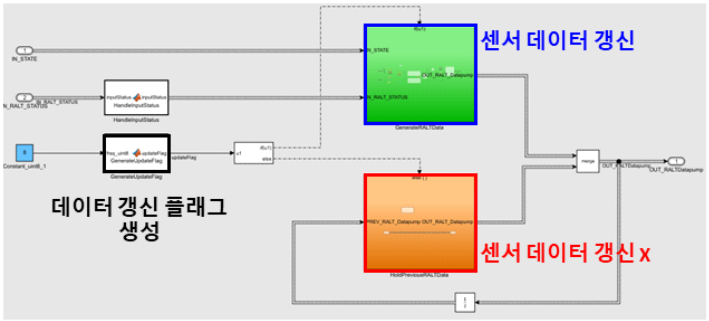
3. 변경 내용 (필요시 별도 문서 첨부)

3-A. FTG 모델 수정 (FTG.slx)

FTG 작동주기 변경

- **FTG 작동주기를 높임 (25Hz → 100Hz)**
  - 따라서, FTG 자동코드도 1초에 100회 호출되도록 스케쥴링이 필요
  - * 작동주기는 Simulink 모델 내에 하드코딩되어 있으므로 재변경시 유의하여야 함

- **센서데이터도 100Hz로 출력/로깅되나 갱신 주기는 DTNS ICD와 동일**
  - GPS(1Hz), RALT(12.5Hz), INS(25Hz)
  - 센서 데이터는 Rate Transition 블록을 이용해 샘플링하여 DTNS 알고리즘 블록에 연결



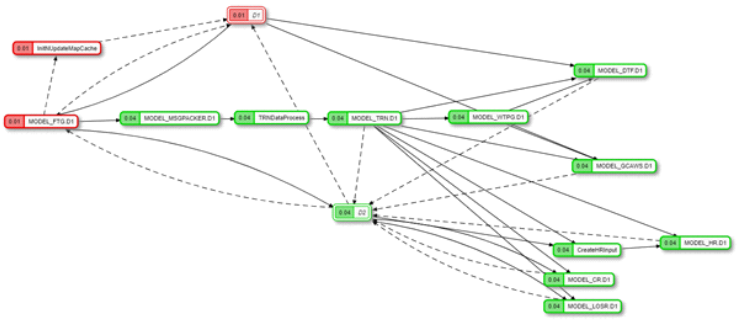
▲ 센서데이터 갱신 제어를 위한 모델 블록 구성 (M_RALT.slx의 예시)

FTG 작동주기 변경

- **센서데이터 출력 결과 확인**
  - 데이터가 ICD와 동일한 갱신주기를 가짐을 확인

시간	0	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	...	0.98	0.99	1	1.01	1.02	1.03	1.04
INS Time Tag	156	156	156	156	780	780	780	780	1404	1404	...	15132	15132	15756	15756	15756	15756	16380
RALT Time Tag	156	156	156	156	156	156	156	156	1404	1404	...	15132	15132	15132	15132	15132	15132	16380
GPS Time Tag	156	156	156	156	156	156	156	156	156	156	...	156	156	15756	15756	15756	15756	15756

- **통합 시뮬레이션 모델 (DTNS_SIM.slx) 구성**
  - Schedule Editor와 Rate Transition 블록을 이용한 실행주기 관리



▲ Schedule Editor를 이용한 모델 실행주기 관리

FTG 내부 구조체 인터페이스 수정

○ BUS_DYN_STATE 구조체(버스)에 다음 데이터를 추가

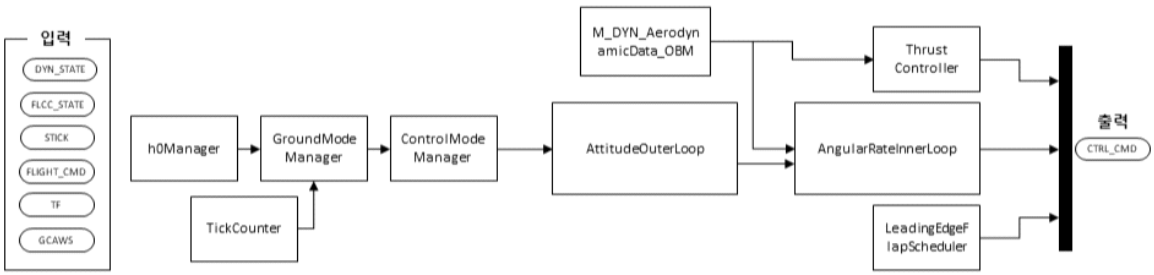
Name	DataType	Dimension	Unit	Description
alpha	double	1	radian	받음각
beta	double	1	radian	옆미끄럼각
Vt	double	1	m/s	진대기속도
C_b_n	double	[3 3]	-	방향코사인
leFlapDeg	double	1	degree	앞전플랩각
groundStage	uint8	1	-	지상운동, 이륙단계
wow	boolean	1	-	Weight on Wheels
windVelN	double	1	m/s	바람 속도(North)
windVelE	double	1	m/s	바람 속도(East)
windVelD	double	1	m/s	바람 속도(Down)
f_ib_b	double	[3 1]	m/s^2	specific force

○ BUS_FLCC_AP_STATE 구조체(버스)에 다음 데이터를 추가

Name	DataType	Dimension	Unit	Description
tick	uint32	1	-	FLCC call ticks counting from start
inFlying	Boolean	1	-	비행중 플래그
groundStage	uint8	1	-	지상운동, 이륙단계
recoveryStage	uint8	1	-	회복기동단계
h0	double	1	m	이륙지점 표고

FLCC 모델 수정

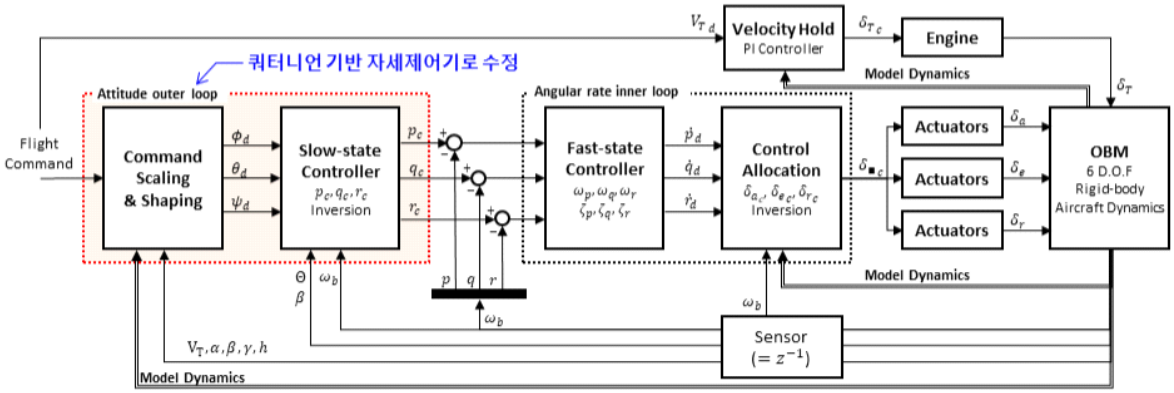
○ 내부 블록 구조 개편



- h0Manager: 지상 운동에 필요한 지상표고 정보를 현재 위치의 지형정보를 조회하여 갱신함. 100틱 마다 갱신 수행
- TickCounter: 현재 실행에서 FLCC 모델이 호출(작동)된 횟수를 갱신
- GroundModeManager: 현재 항공기가 지상운동중인지, 비행중인지를 식별하고 지상운동일 경우 이륙절차 스케쥴링을 진행
- ControlModeManager: 현재 항공기 제어모드를 결정
- AttitudeOuterLoop: 현재 항공기 제어모드에 따라 자세 명령을 설계하고, 각속도 명령을 결정
- AngularRateInnerLoop: 비선형 동적 모델역변환(NDI)을 통해 각속도를 제어하는 내측 제어루프
- ThrustController: 추력 제어기
- LeadingEdgeFlapScheduler: 앞전플랩 제어 명령을 스케쥴링한다
- M_DYN_AerodynamicData_OBM: NDI 제어를 위한 OBM

FLCC 모델 수정

- **비행제어기 외부루프에 위치한 자세제어기를 쿼터니언 기반 자세제어기로 변경**
  - 기존 오일러각 기반의 자세제어기는 특정 자세각(ex: 피치각 ±90°)에서 특이점이 발생 (같은 자세를 기술하는 자세 표현이 유일하지 않음)
  - 반복되는 Dive, Recovery 비행은 피치각 ±90°을 넘나들 수 있음
  - 특이점 대응을 위해 자세명령을 설계하는 자세제어기 외부루프를 쿼터니언 자세 제어기로 변경



▲ 다단 비선형 동적 모델역변환 비행 제어법칙 구조도

FLCC 모델 수정

- **비행제어기 외부루프에 위치한 자세제어기를 쿼터니언 기반 자세제어기로 변경**

쿼터니언을 이용한 자세 표현

- 크기가 1인 쿼터니언은 3차원 상의 회전(자세)를 표현하는데 이용할 수 있음
- 고유 회전축과 그에 대한 회전각으로 자세를 표현하는 방식

$$q = [q_0 \quad q_x \quad q_y \quad q_z]^T = [\cos(\Theta/2) \quad u_x \sin(\Theta/2) \quad u_y \sin(\Theta/2) \quad u_z \sin(\Theta/2)]^T$$

자세오차 표현 및 제어[2]

- 자세 제어기는 자세 제어오차를 되먹임하여 작동

$$q_{sp} = q \otimes q_{err}$$
$$q_{err} = q \otimes q_{sp}$$

$$\omega_{sp} = 2\bar{q} \otimes \dot{q}$$

sp : set point

$$\Theta_{err} = 2 \cdot acos(q_{0,err})$$
$$u_x = \frac{q_{x,err}}{\sqrt{1 - q_{0,err}^2}}$$
$$u_y = \frac{q_{y,err}}{\sqrt{1 - q_{0,err}^2}}$$
$$u_z = \frac{q_{z,err}}{\sqrt{1 - q_{0,err}^2}}$$

※ $\Theta_{err} > \pi$: $\Theta_{err} = |\Theta_{err} - 2\pi|$ 및 $u_x = -u_x, u_y = -u_y, u_z = -u_z$

$$\begin{bmatrix} \omega_{x,sp} \\ \omega_{y,sp} \\ \omega_{z,sp} \end{bmatrix} = 2 \cdot \begin{bmatrix} Kp_x \cdot q_{x,err} \\ Kp_y \cdot q_{y,err} \\ Kp_z \cdot q_{z,err} \end{bmatrix} + \begin{bmatrix} 0 \\ Kd_y \cdot \dot{q}_{y,err} \\ 0 \end{bmatrix} + \begin{bmatrix} p_{trim} \\ q_{trim} \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ K_\beta \cdot \beta \end{bmatrix}$$

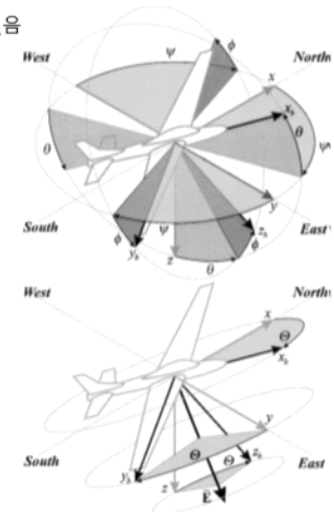


그림 오일러 자세각 표현과 오일러 축 표현(쿼터니언 자세각) 비교 [1]

[1] Phillips, W. F., C. E. Hailey, and G. A. Gebert. "Review of Attitude Representations Used for Aircraft Kinematics." *Journal of Aircraft* 38, no. 4 (2001): 718–37.
[2] Gołąbek, Michał, Michał Welcer, Cezary Szczepański, Mariusz Krawczyk, Albert Zajdel, and Krystian Borodacz. "Quaternion Attitude Control System of Highly Maneuverable Aircraft." *Electronics* 11, no. 22 (2022): 3775.

DYN 모델 수정

○ 지상운동 모사를 위한 항법방정식 수정

- DTNS FTG 모델 상에서 물리환경을 구현하는 것은 추가 툴박스 활용이나 불필요하게 많은 작업량이 요구되므로, 물리적 구속조건을 강제하는 방식으로 지상운동 모사

구속조건

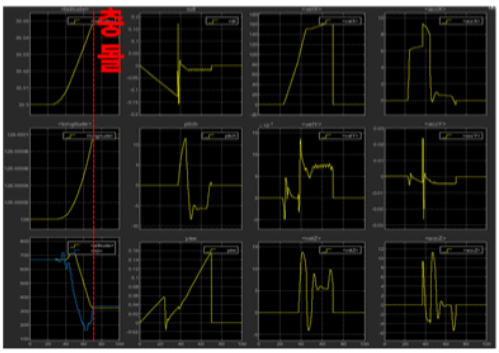
- 이륙전환 직전까지는 롤, 피칭 운동 제한: $\omega_{ib,x}^{b} = 0, \omega_{ib,y}^{b} = 0$
- 이륙전환 직전까지는 수직 운동 제한: $v_{eb,down}^{n} = 0, f_{ib,down}^{n} = -g_{down} + (w_{en}^{n} + 2w_{ie}^{n}) \times v_{eb}^{n}$
- 이륙전환 직전까지는 운동마찰력 적용

○ DYN 모델에서 wow 플래그 생성

- 지상운동 중에는 wow 플래그를 true로 출력

○ 지형 충돌상황 감지

- 비행중 항공기 고도가 (지형표고-5m) 보다 낮아지면 충돌로 간주하고 모든 운동이 0으로 수렴
- 지상운동/이륙 중에는 충돌 감지 수행하지 않음



DYN 모델 수정

○ 수치계산 안전 장치 추가

- 지상운동 모드 추가로 비행중에는 0이 되지 않는 값이 0이 되는 문제가 일부 발생
  - ➢ 진 대기속도, 동압 등이 0이 될 수 있으며, 나눗셈 활용시 0으로 나누는 문제 발생
  - ➢ 0으로 나눌 가능성이 있는 연산의 분모에 아주 작은 값을 더하여 해결

```
function CX_tot = getCXt(CX, dCX_lef, CXq, dCXq_lef, DLEF, cbar, q, V)

V = V + 1e-4;

CX_tot = CX ...
    + dCX_lef*(1-DLEF/25) ...
    + q*cbar/(2*V)*(CXq+dCXq_lef*(1-DLEF/25));

end
```

▲ 0으로 나누기를 우회하는 방법

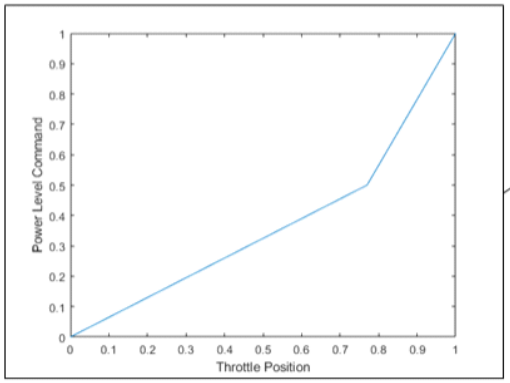
M_DYN_THRUST 모델 고도화

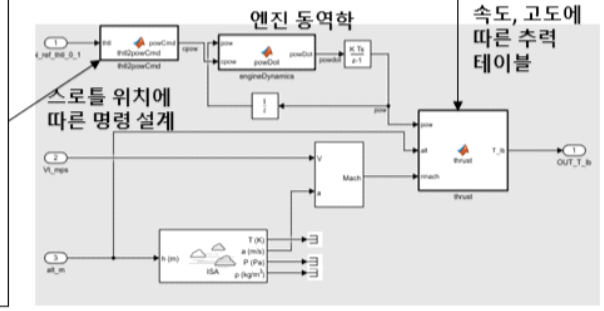
- 스로틀 레버 위치에 따른 엔진 추력 명령 수정
- 엔진 동역학(1차 시간지연 시스템) 적용
- 고도, 속도에 따른 추력 효율 적용

```
a=[1060 670 880 1140 1500 1860
    635 425 690 1010 1330 1700
    60 25 345 755 1130 1525
   -1020 -170 -300 350 910 1360
   -2700 -1900 -1300 -247 600 1100
   -3600 -1400 -595 -342 -200 700]';

b=[12680 9150 6200 3950 2450 1400
   12680 9150 6313 4040 2470 1400
   12610 9312 6610 4290 2600 1560
   12640 9839 7090 4660 2840 1660
   12390 10176 7750 5320 3250 1930
   11680 9848 8050 6100 3800 2310]';

c=[20000 15000 10800 7000 4000 2500
   21420 15700 11225 7323 4435 2600
   22700 16860 12250 8154 5000 2835
   24240 18910 13760 9285 5700 3215
   26070 21075 15975 11115 6860 3950
   28886 23319 18300 13484 8642 5057]';
```





INS 모델 수정

- IMU 모델의 오차 특성값을 변경함
  - 자이로 잡음의 PSD, 자이로 편이, 자이로 환산계수 및 교차축 오차 변경
    - SNU 84-1 관성항법장치 성능 요구사항에 준하여 INS 데이터를 모사하기 위함
      - 주요 요구사항: 첫 1시간동안 위치 오차 0.8해리 이하
    - 기존 모델은 다소 높은 오차 특성값이 적용되어 있었음
  - 자이로 g-민감도 오차 삭제
    - RLG, FOG 등 비 진동형 자이로는 항공기 가속에 따른 영향이 적음

오차특성	기호	값
micro-g	μg	$9.8 \times 10^{-6}\ m/s^2$
가속도계 잡음 스펙트럼 밀도의 제곱근 Accelerometer noise root PSD	$\sqrt{S_a}$	$20\,\mu g$
자이로 잡음 스펙트럼 밀도의 제곱근 Gyro noise root PSD	$\sqrt{S_g}$	$0.002 \times \left(\frac{\pi}{180}\right) \times \left(\frac{1}{60}\right)\ rad/s^{0.5}$
가속도계 편이 Accelerometer Bias	b_a	$[25 \quad -30 \quad -5] \times \mu g$
자이로 편이 Gyro bias	b_g	$[-0.0004 \quad 0.0005 \quad -0.0005] \times \left(\frac{\pi}{180}\right) \times \left(\frac{1}{3600}\right) rad/s$
가속도계 환산계수 및 교차축 오차 Accelerometer scalefactor and cross coupling errors	M_a	$\begin{bmatrix} 100 & -120 & 80 \\ -60 & -120 & 100 \\ -100 & 40 & 90 \end{bmatrix} ppm$
자이로 환산계수 및 교차축 오차 Gyro scale factor and cross coupling errors	M_g	$\begin{bmatrix} 2 & 20 & -25 \\ 0 & -3 & 15 \\ 0 & 0 & -1 \end{bmatrix} ppm$
자이로 g-민감도 오차 Gyro g-dependent biases	G_g	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \times \left(\frac{\pi}{180}\right) \times \left(\frac{1}{3600 \times g}\right) rad \cdot sec/m$

비활성화

▲ 수정된 INS 모델의 IMU 오차 특성 테이블

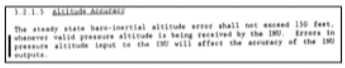
INS 모델 수정

○ IMU 측정치 참값 생성 방식 변경

- 기존 FTG는 운동학 역변환 방식으로 specific force, 각속도 참값을 생성
  - ➢ 역변환 과정에서 발생하는 수치 오차가 누적되어 INS 성능에 영향을 미침을 발견
- 동역학 모델(M_DYN)과 동일한 측정치를 참값으로 가져와 사용하여 해당 문제 해결

○ 압력고도계 잡음 수준 변경

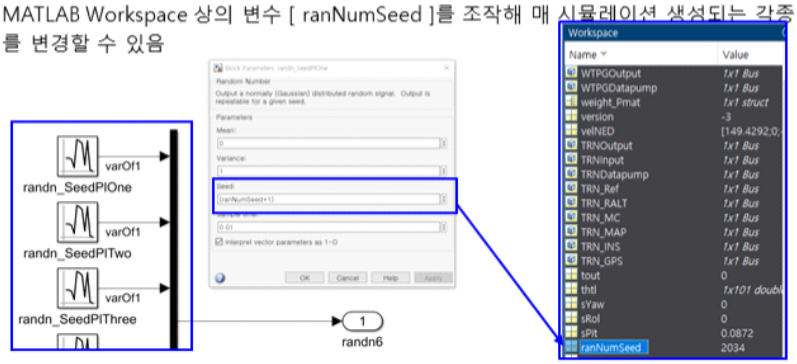
- 작동주기가 100Hz로 4배 높아짐에 따라 Random Walk를 고려해 변경 (정상상태 고도 오차가 150피트를 넘어서는 안됨)


3.2.1.5 Altitude Accuracy
The steady state baro-inertial altitude error shall not exceed 150 feet, whenever valid pressure altitude is being received by the IMU. Errors in pressure altitude input to the IMU will affect the accuracy of the IMU outputs.

▲ SNU 84-1

○ 난수생성기의 시드를 변경할 수 있도록 함

- MATLAB Workspace 상의 변수 [ranNumSeed]를 조작해 매 시뮬레이션 생성되는 각종 잡음값의 난수 시드를 변경할 수 있음

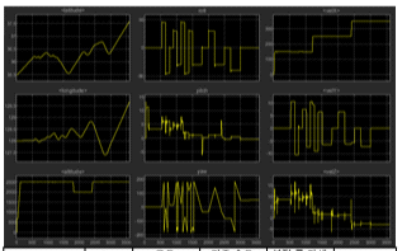


▲ 난수 시드 설정

INS 모델 수정

○ 시뮬레이션 검증

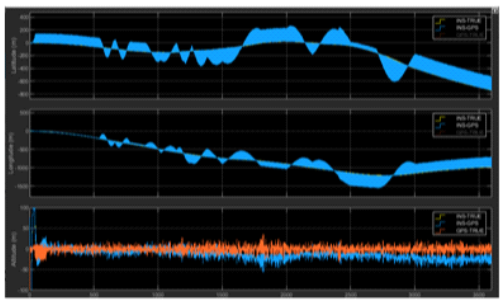
- 지상 활주/이륙하여 1시간 비행한 시나리오에서 목표 요구성능을 만족함을 확인



이벤트	Time (sec)	고도 (m)	기준 속도 (m/s)	선정 롤 자세 (deg)	비고
이륙	0	2500	0	0	
+X 롤 기동	540	2500	150	45	
-X 롤 기동	660	2500	150	45	
수평 비행	780	2500	150	0	
+X 롤 기동	900	2500	150	30	
-X 롤 기동	1020	2500	150	30	
+X 롤 기동	1140	2500	150	45	
기준 속도 변경	1200	2500	250	45	
-X 롤 기동	1260	2500	250	45	
+X 롤 기동	1380	2500	250	30	
-X 롤 기동	1500	2500	250	30	
기준 고도 변경	1800	2000	250	0	
+X 롤 기동	2000	2000	250	30	
-X 롤 기동	2200	2000	250	30	
기준 고도 변경	2400	2500	350	30	
수평 비행	2500	2500	350	0	
-X 롤 기동	2700	2500	350	40	
수평 비행	3000	2500	350	0	
비행 종료	3600	2500	350	0	

ATT HOLD | ALT HOLD

▲ 검증 시뮬레이션 기동 프로파일 및 시나리오



▲ INS 위치 오차 그래프

최대 수평 위치오차 약 1.3km

고도 정상상태 오차 50m 미만

INS 모델 수정

- 참고자료
  - [SNU 84-1] Specification for USAF Standard Form Fit and Function(F3) Medium Accuracy Inertial Navigation Unit

	FULL GC	DEGRADED GC	EIA	SH	BATH	ATT
MAXIMUM ALIGN TIME (minutes) (-40 to +71 Degrees Celsius)	8.0 - -	1.5 - -	8.0 ** TAXI 4.0 **	1.5 - -	1.5 - -	- - -
POSITION ACCURACY (nmi/hr) (CEP) Flights ≤ 1.0 hr Flights >1 and ≤ 10.0 hr (95% of all flights) Flights > 10.0 hr (95% of all flights) CROSS TRACK (nmi) ALONG TRACK (nmi)	0.8 2.0 ±20 ±25	5.0 - - -	0.5 2.0 ±20 ±25	0.8 - - -	- - - -	- - - -
VELOCITY ACCURACY Linear (X,Y) $\tau \leq$ 1 min (fps) 1 < $\tau \leq$ 2 min (fps) 2 < $\tau \leq$ 5 min (fps) $\tau \geq$ 5 min (fps) Linear (Z) (fps RMS) Linear (N,E) (fps RMS) Angular (P,Q,R) (deg/sec RMS)	2.5 RMS* 2.5 RMS* 2.5 RMS* 2.5 RMS* 2.0 * 2.5 * 0.043 *	- - - - - - -	0.15 *** 0.3 *** 0.4 *** 2.0 RMS* 2.0 * 2.5 * 0.043 *	5.0RMS* - - - 3.0 * - -	- - - - - - -	- - - - - - -
ACCELERATION ACCURACY Linear (X,Y,Z) (ft/s/s RMS) Linear (Lateral,Longitudinal, Normal) (ft/s/s RMS) Angular (deg/s/s)	0.064 * 2.0 * 10.0	- - -	0.064 * 2.0 * 10.0	- - -	- - -	- - -
STATIC ATTITUDE (ROLL, PITCH, PLATFORM AZ) Digital (deg RMS) Analog (deg RMS) True Heading (deg) Magnetic Heading (deg)	0.05 0.067 0.1 RMS 0.3 RMS	0.1 0.1 0.5 RMS 1.0 RMS	0.05 0.067 0.1 RMS 0.3 RMS	0.1 0.1 0.1 RMS 0.3 RMS	0.1 0.1 - -	- - 15/hr 15/hr

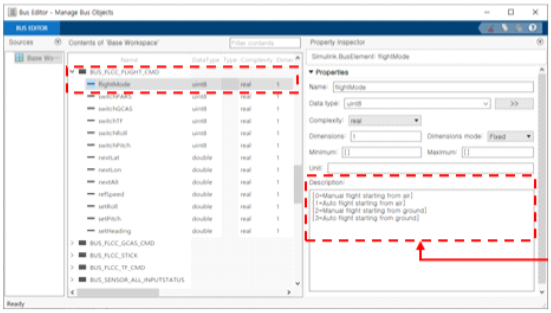
* For flights up to two hours.
** 8 Minute GC Align, Taxi to new heading ≥ 70 degrees delta, (taxi time ≤ 10 minutes), 4 Minute GC Align.
*** One Sigma Value.
τ A period starting at the selection of the NAV mode.

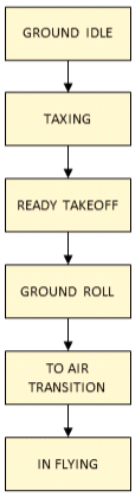
F3 OUTPUT PARAMETER	UNITS	REFRESH RATE(HZ)	LSB VALUE	ACCURACY** (RMS)	JITTER* ** (RMS)	RESPONSE TIME*	BANDWIDTH (HZ)*	OUTPUT MESSAGES
True Heading	Pirads	200 50 50	3.05176E-5	5.57042E-4 (0.1 deg)	6.66667E-5 (0.012 deg)	10 ms@200 Hz 20 ms@ 50 Hz	25	IN1-03 I01-12 I06-12
Roll	Pirads	200 200 50 50	3.05176E-5	2.76930E-4 (0.05 deg)	6.66667E-5 (0.012 deg)	10 ms@200 Hz 20 ms@ 50 Hz	25	I09-10 IN1-02 I01-10 I06-10
Pitch	Pirads	200 200 50 50	3.05176E-5	2.76930E-4 (0.05 deg)	6.66667E-5 (0.012 deg)	10 ms@200 Hz 20 ms@ 50 Hz	25	I09-11 IN1-01 I01-11 I06-11
Velocity North	fps	200	0.125	2.5	0.25	20 ms	5	IN1-04
Velocity East	fps	200	0.125	2.5	0.25	20 ms	5	IN1-05
Vert. Velocity	fps	200	0.0625	2.0	0.25	20 ms	5	IN1-06
X Velocity	fps	200 200 50 50	3.81470E-6	2.5	0.002	20 ms@200 Hz 26 ms@ 50 Hz	21	I09-03,04 IN1-08,09 I01-03,04 I06-03,04
Y Velocity	fps	200 200 50 50	3.81470E-6	2.5	0.002	20 ms@200 Hz 26 ms@ 50 Hz	21	I09-05,06 IN1-10,11 I01-05,06 I06-05,06
Z Velocity	fps	200 200 50 50	3.81470E-6	2.0	0.002	20 ms@200 Hz 26 ms@ 50 Hz	21	I09-07,08 IN1-12,13 I01-07,08 I06-07,08
Pitch Rate	Pirads/ Sec	200 50	1.22070E-4	2.38732E-4 (0.043 deg/s)	2.00000E-4 (0.036 deg/s)	20 ms@200 Hz 25 ms@ 50 Hz	21	I09-13 I01-31
Roll Rate	Pirads/ Sec	200 50	1.22070E-4	2.38732E-4 (0.043 deg/s)	2.00000E-4 (0.036 deg/s)	20 ms@200 Hz 25 ms@ 50 Hz	21	I09-12 I01-30
Yaw Rate	Pirads/ Sec	200 50	1.22070E-4	2.38732E-4 (0.043 deg/s)	2.00000E-4 (0.036 deg/s)	20 ms@200 Hz 25 ms@ 50 Hz	21	I09-14 I01-32

▲ [SNU 84-1]의 관성항법장치 성능 요구표

자동비행 지상운동 단계

- [자동 비행, 지상 시작]시 다음 절차에 따라 자동 이륙 수행 (상태 기계 구성)
  - [0 = JUST BOOTED UP]
  - [1 = GROUND IDLE]
  - [2 = TAXING]
  - [3 = READY TAKEOFF]
  - [4 = GROUND ROLL]
  - [5 = ROTATION]
  - [6 = TO AIR TRANSITION]
  - [7 = CLIMBOUT]
  - [8 = IN FLYING]
  - [9 = CRASHES]





시나리오 입력의 flightMode를 비행 목적에 맞게 입력해야 함

FTG_initState.m 수정

- 항공기 상태 메시지가 추가됨에 따라 초기 항공기 상태를 결정하는 스크립트도 수정되었음
  -

필요에 따라 수정

```
initState = struct;
initState.latitude = 35.10;
initState.longitude = 127.70;
initState.altitude = 1000;
initState.roll = 0;
initState.pitch = 3*pi/180;
initState.yaw = 0;
initState.velX = 200;
initState.velY = 0;
initState.velZ = 0;
initState.omX = 0;
initState.omY = 0;
initState.omZ = 0;
initState.accX = 0;
initState.accY = 0;
initState.accZ = 0;
initState.omDotX = 0;
initState.omDotY = 0;
initState.omDotZ = 0;
initState.throttleN = 50000;
initState.elevatorDeg = 0;
initState.aileronDeg = 0;
initState.rudderDeg = 0;
```

▲ 기존 상태 초기화 스크립트

```
% Calculate (DCM) rotation matrix
% Define trigonometric functions
cRol = cos(initState.roll);
sRol = sin(initState.roll);
cPit = cos(initState.pitch);
sPit = sin(initState.pitch);
cYaw = cos(initState.yaw);
sYaw = sin(initState.yaw);

initState.C_b_n = ...
    [ cPit*cYaw, sRol*sPit*cYaw-cRol*sYaw, cRol*sPit*cYaw+sRol*sYaw;...
      cPit*sYaw, sRol*sPit*sYaw+cRol*cYaw, cRol*sPit*sYaw-sRol*cYaw;...
          -sPit,               sRol*cPit,               cRol*cPit];

velNED = initState.C_b_n ...
    * [initState.velX; initState.velY; initState.velZ];
initState.velN = velNED(1);
initState.velE = velNED(2);
initState.velD = velNED(3);
```

▲ 추가된 스크립트 (자동 계산)

KAIST 한서대학교 HANSEO UNIVERSITY 18

4. 변경 사유 (필요시 별도 문서 첨부)

- 지상운동 모사 및 FTG 고도화 작업
- 필요한 내용은 상기 노트에 작성하였음

5. 배포자

A. 이름	B. 소속기관
김성중	KAIST

6. 첨부 (해석 결과 포함)

- FTG V5.0은 다음 비행에 대해 비행 데이터 생성이 가능함을 확인하였음

(1) 지상 시작, 활주 이륙 자동비행

(2) 자동 TF 비행

(3) 자동 GCAWS 비행

그림 16. 자동 TF 비행 시뮬레이션 결과

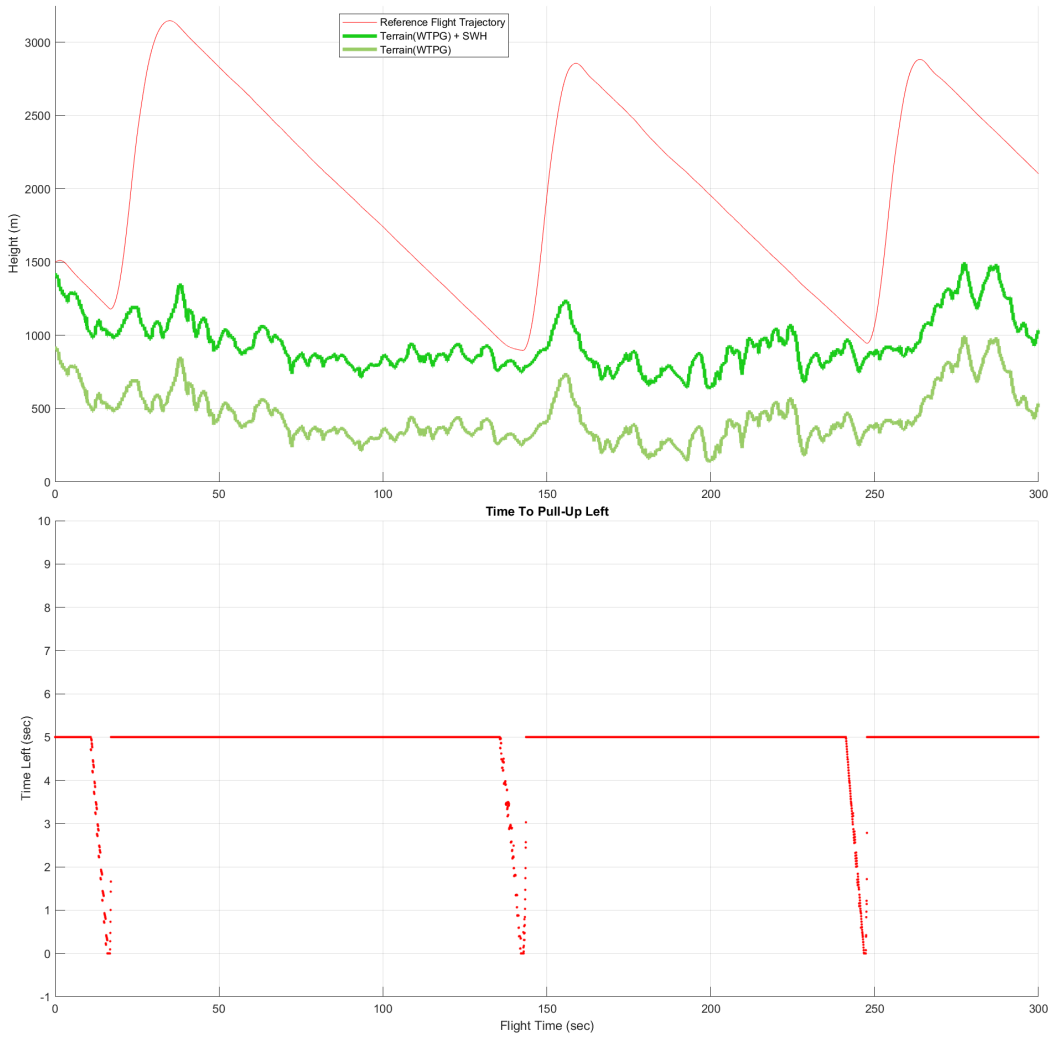


그림 17. 자동 GCAWS 비행 시뮬레이션 결과

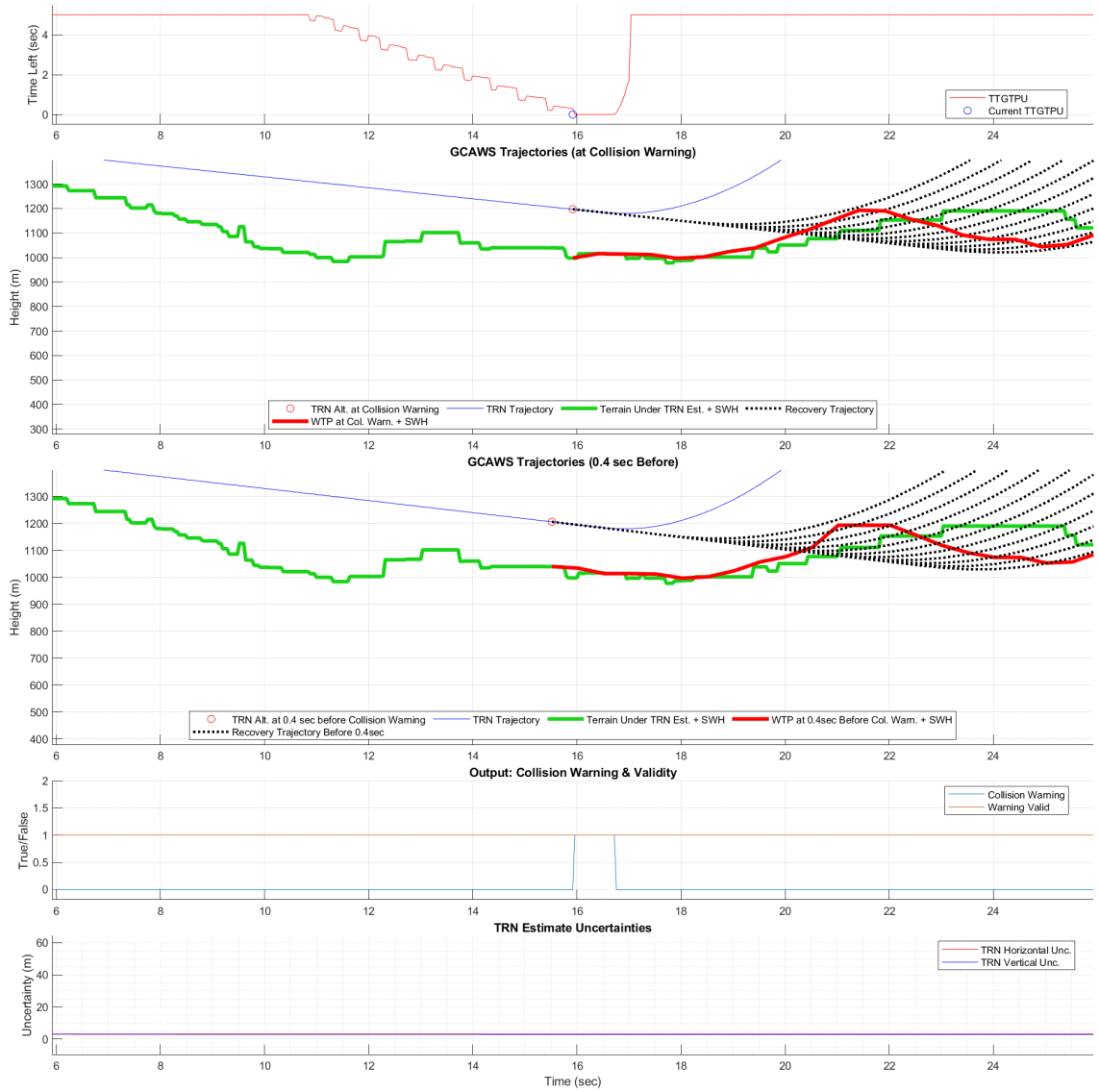


그림 18. 자동 GCAWS 비행 시뮬레이션