

MODEL RELEASE NOTE

1. 모델 버전

DTNS FTG V4.0 / V4.1

2. 배포 일자 (YYMMDD)

23.04.12

3. 변경 내용 (필요시 별도 문서 첨부)

3-A. FTG 모델 수정 (FTG.slx)

① FTG 입출력 인터페이스 변경

- (V4.1) 입력 포트 [IN_FTG_TF_CMD(BUS_FLCC_TF_CMD)]를 [IN_FTG_DTFOutput(DTFOutput)]으로 변경

→ AUTO TF 모드로 비행시 DTF 출력값을 입력받기 위함

- (V4.1) 입력 포트 [IN_FTG_GCAS_CMD(BUS_FLCC_GCAS_CMD)]를 [IN_FTG_GCAWSOutput(GCAWSOutput)]으로 변경

→ AUTO GCAWS 모드로 비행시 GCAWS 출력값을 입력받기 위함

- (V4.0) 입력 포트 [IN_FTG_SENSOR] 추가

→ 출력값 변조가 아닌 입력단에서 센서 동작을 제어하기 위함

→ 센서별 ModeWord(유효성 플래그 등), 상태, 주입 오차 등을 요소로 포함하는 구조체

*** 입력 포트 변경으로 인해 기존 FTG_INPUT_SCENARIOS_*가 호환되지 않음.**

새로 입력시나리오를 생성하거나 그림 2와 같이 추가된 입력 포트에 빈값 []을 할당해주어야 함.

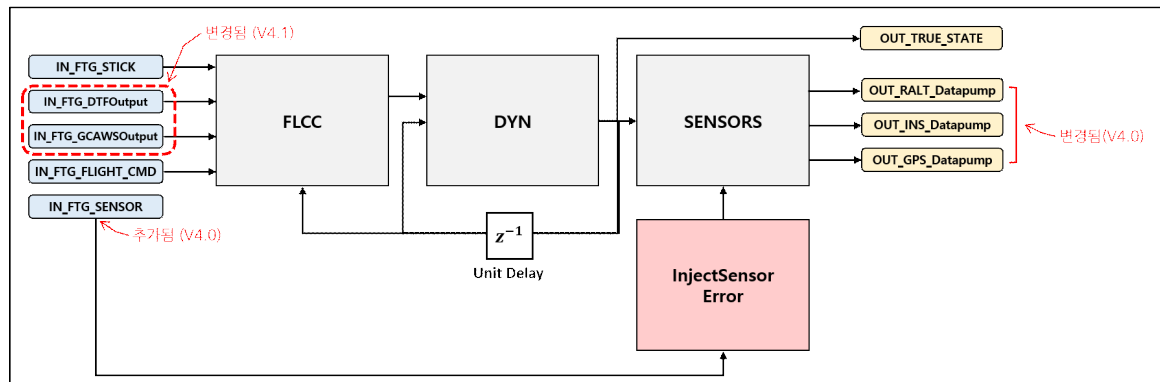


그림 1. FTG V4.1 구조



그림 2. 기존 FTGScenario 호환 방법

- (V4.0) 출력 포트 수정 [OUT_SENSOR_*_DP]

→ 모든 수치 데이터를 double로 출력

→ ModeWord, TimeTag 등은 원 자료형 유지

```

/* Customized model step function */
extern void FTG_Step(BUS_FLCC_STICK *p_IN_FTG_STICK,
..... DTFOutput *p_IN_FTG_DTFOutput,
..... GCAWSOutput *p_IN_FTG_GCWASOutput,
..... BUS_FLCC_FLIGHT_CMD *p_IN_FTG_FLIGHT_CMD,
..... BUS_SENSOR_ALL_INPUTSTATUS *p_IN_FTG_SENSOR,
..... BUS_SENSOR_RALT_DP *p_OUT_RALT_Datapump,
..... BUS_SENSOR_INS_DP *p_OUT_INS_Datapump,
..... BUS_SENSOR_GPS_DP *p_OUT_GPS_Datapump,
..... BUS_DYN_STATE *p_OUT_TRUE_STATE);

```

그림 3. 수정된 FTG 모델 C코드의 호출함수 원형

② (V4.0) 최신 MAP API (Post방식 포함) 적용

- Post방식이 구현된 최신 MAP API가 적용되었음
참조 Mail#:A2K2023032203, Mail#:A2K2023032402

3-B. RALT 센서모델 수정 (M_RALT.slx)

① (V4.0) RALT 센서모델 인터페이스 변경

- (표 1 참조) 모든 수치 데이터는 double로 출력
- ModeWord, TimeTag, Valid 플래그 등은 원 자료형 유지

[입력] BUS_SENSOR_RALT_INPUTSTATUS			
자료명	자료형	단위	비고
InputActivated	boolean		INPUTSTATUS 주입 활성화 플래그 (기본값: false)
radarAltitudeValid	boolean		레이더 고도 유효
coneFOV	double	deg	(0~30도) 범위에서 사용 (기본값: 30도)
InjectedError	double	m	레이더 고도 주입 오차
[출력] BUS_SENSOR_RALT_DP			
자료명	자료형	단위	비고
RALTModeWord	uint16		
TimeTag	uint16		
RadarAltitude	double	m	직하방, 오차O
RadarAltitudeTrue	double	m	직하방, 오차X
RadarAltitudeCone	double	m	원뿔형, 오차O
RadarAltitudeValid	boolean		RALTModeWord에서 RadarAltitudeValid 값 추출

표 1. RALT 센서모델 인터페이스

```

extern void M_RALT_Init(void);
extern void M_RALT_Step(const BUS_DYN_STATE *p_IN_STATE, const
BUS_SENSOR_RALT_INPUTSTATUS *p_IN_RALT_STATUS, BUS_SENSOR_RALT_DP
*p_OUT_RALTDatapump);

```

그림 4. 변경된 M_RALT 모델의 호출함수와 전달인자

② (V4.0) 3종의 레이더 고도계 측정치 출력 제공

- 1) 직하방 AGL

- 2) 직하방 AGL + 고도 비례 가우시안 오차
- 3) 원뿔형 최단 경사거리 + 고도 비례 가우시안 오차

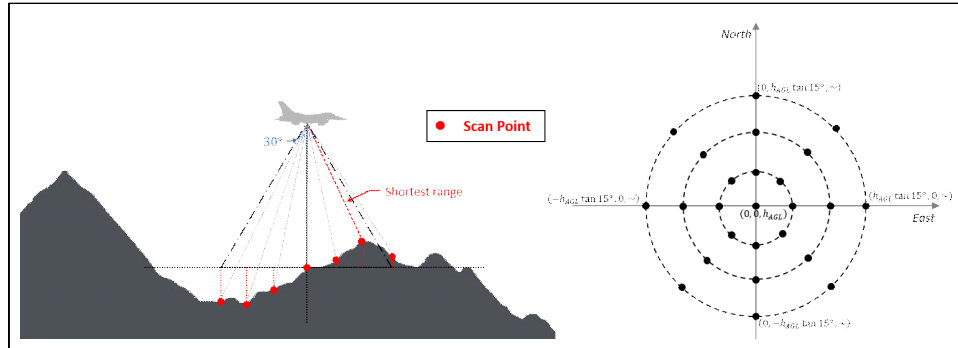


그림 5. 원뿔형 레이더 고도계 측정치 작동 방식

③ (V4.0) 최신 MAP API (Post방식 포함) 적용

- Post방식이 구현된 최신 MAP API가 적용되었음
- 참조 Mail#: A2K2023032203

3-C. INS 센서모델 수정 (M_INS.slx)

① (V4.0) INS 센서모델 인터페이스 변경

- (표 2 참조) 모든 수치 데이터는 double로 출력
- ModeWord, TimeTag, Valid 플래그 등은 원 자료형 유지

[입력] BUS_SENSOR_INS_INPUTSTATUS			
자료명	자료형	단위	비고
InputActivated	boolean		INPUTSTATUS 주입 활성화 플래그 (기본값: false)
INSMoDeWord	uint16		(기본값: 0)
AlignmentTrigger	boolean		true로 trigger시 INS 항법해가 참값과 일치됨 (기본값: false)
InjectedError	double[12]	m	INS 주입 오차 (위치(3), 속도(3), 자세(3), 가속도(3))
[출력] BUS_SENSOR_INS_DP			
자료명	자료형	단위	비고
INSMoDeWord	uint16		
INSTimeTag	uint16		
VelocityX	double	m/s	
VelocityY	double	m/s	
VelocityZ	double	m/s	
PlatformAzimuth	double	deg	
Roll	double	deg	
Pitch	double	deg	
TrueHeading	double	deg	
AccelerationX	double	m/ss	
AccelerationY	double	m/ss	
AccelerationZ	double	m/ss	
Latitude	double	deg	
Longitude	double	deg	
InertialAltitude	double	m	

표 2. INS 센서모델 인터페이스

② (V4.0) 최신 MAP API (Post방식 포함) 적용

- FTG V4.0부터는 Post방식이 구현된 최신 MAP API가 적용되었음
참조 Mail#: A2K2023032203

3-D. GPS 센서 모델 수정 (M_GPS.slx)

① (V4.1) GPS 센서모델 인터페이스 변경

- (표 3 참조) 모든 수치 데이터는 double로 출력
- ModeWord, TimeTag, Valid 플래그 등은 원 자료형 유지
- GPSStatusFOM은 13~16번째 비트에 FOM 값이 할당되었으나(V4.0), 수치입력을 직접 받도록 수정하였음(V4.1). (ex: FOM=3 => GPSStatusFOM=3)

[입력] BUS_SENSOR_GPS_INPUTSTATUS			
자료명	자료형	단위	비고
InputActivated	boolean		INPUTSTATUS 주입 활성화 플래그 (기본값: false)
GPSPModeWord	uint16		1st bit (=1): Navigation Data Valid / (=0): Not Valid
SAASStatusWord	uint16		1st Bit (=1): Daily Key In Use (P code)
GPSStatusFOM	uint16		Figure of Merit (1~9)
GPSSatelliteState	uint16		
InjectedError	double[6]	m	GPS주입 오차 (위치(3), 속도(3))
[출력] BUS_SENSOR_GPS_DP			
자료명	자료형	단위	비고
GPSPModeWord	uint16		
GPSTimeOfValidity	uint16		
SAASStatusWord	uint16		
Latitude	double	deg	
Longitude	double	deg	
Altitude	double	m	
VelocityEast	double	m/s	
VelocityNorth	double	m/s	
VelocityUp	double	m/s	
GPSStatusFOM	uint16		
EstimatedHorizontalPositionError	double	m	
EstimatedVerticalPositionError	double	m	
GPSSatelliteState	uint16		

표 3. GPS 센서모델 인터페이스

② (V4.0) P코드 동작 모드 구현

- 입력된 SA/AS Status Word가 “Daily Key In Use”인 경우 P코드 사용 수준의 GPS 오차 사용하도록 구현
- 10초 주기로 GPS 오차가 범위 내에서 변하도록 $u \sim U(0,1)$ 설계

P, FOM	σ	
P Available	6.25 + 4.3u	PCodeAvailable = logical(bitget(statusWord, 1));
FOM = 1	12.5 + 12.5u	if (PCodeAvailable)
FOM = 2	25 + 25u	EPE = 6.25 + 4.3 * variableNoise;
FOM = 3	50 + 25u	else
FOM = 4	75 + 25u	switch FOM
FOM = 5	100 + 100u	case uint16(1)
FOM = 6	200 + 300u	EPE = 12.5 + 12.5 * variableNoise;
FOM = 7	500 + 500u	case uint16(2)
FOM = 8	1000 + 4000u	EPE = 25 + 25 * variableNoise;
FOM = 9	5000 + 4000u	case uint16(3)
		EPE = 50 + 25 * variableNoise;
		case uint16(4)
		EPE = 75 + 25 * variableNoise;
		case uint16(5)
		EPE = 100 + 100 * variableNoise;

3-E. 센서 메시지 생성 모델 분리 (M_MSGPACKER.slx)

- 기존 FTG는 센서 출력값을 ICD와 유사한 구조체로 인코딩하여 출력하였으나, 인터페이스를 수정하면서 수치 데이터는 모두 double형 데이터로 출력하고 있음.
- 센서 출력값을 ICD 유사 메시지로 변환하는 모델을 별도로 생성해 해당 기능을 분리하였음

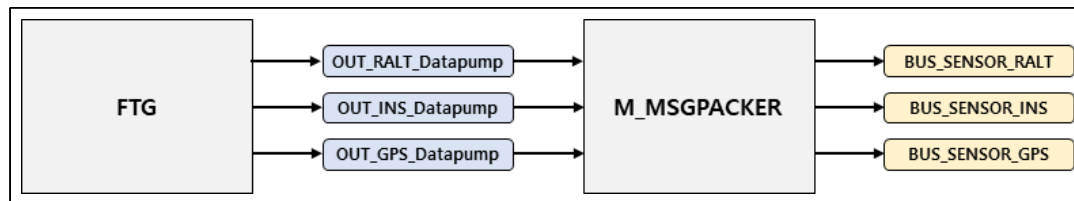


그림 7. M_MSGPACKER 입출력 구조

3-F. 오토 파일럿 개선 및 수정 (M_FLCC.slx)

① (V4.0) 고도 제어기 개선 [M_FLCC/AutoPilot/PITCH_ALT HOLD/holdAltitudePitchCommand]

- 선회비행시 고도가 진동하며 유지되지 못하는 문제 발생
- 고도 제어를 위한 피치 명령 생성기의 PI 제어이득 값을 조정
(Kh_p = 0.4 -> 0.2, Kh_i = 0.1->0.01)
- 피치 명령 생성기의 고도 변화율 출력 제한을 (hDotlim 삭제)
(hLim = 40 -> 250, hDotlim 삭제)

11	% PI control gains for altitude controller	11	% PI control gains for altitude controller
12		12	
13	Kh_p = 0.4;	13	Kh_p = 0.2;
14	Kh_i = 0.1;	14	Kh_i = 0.01;
15		15	
16	alp = alphaDeg*d2r;	16	alp = alphaDeg*d2r;
17		17	
18	% Altitude controller	18	% Altitude controller
19	hLim = 40;	19	hLim = 100;
20	hErrLim = 10;	20	hErrLim = 10;
21		21	
22	% Altitude for command shaping	22	% Altitude for command shaping
23	hErr = fcnSaturation(hCmd - h, hLim);	23	hErr = fcnSaturation(hCmd - h, hLim);
24	hErrInt = fcnSaturation(hErrInt + hErr*dt, hErrLim);	24	hErrInt = fcnSaturation(hErrInt + hErr*dt, hErrLim);
25		25	
26	hDotlim = 10;	26	
27	hDot = Kh_p*hErr + Kh_i*hErrInt;	27	hDot = Kh_p*hErr + Kh_i*hErrInt;
28	hDot = fcnSaturation(hDot, hDotlim);	28	
29		29	
30	% Flight path angle command	30	% Flight path angle command
31	gamCmd = hDot / Vt;	31	gamCmd = hDot / Vt;

그림 8. 고도제어기 게인 값 및 출력 제한값 해제(좌 수정 전, 우 수정 후)

② (V4.0) 경로점 비행을 위한 선회비행시 롤 자세 명령 한계값 변경 [M_FLCC/AutoPilot/If Action Subsystem/getDesiredRollAngle]

- 현재 경로점 비행시 최대 뱅크각(롤 자세각)은 60도로 설정되어 있으며, 이는 레이더 고도계가 정상 작동할 수 없는 값임
- 경로점 비행 중 선회비행시에도 레이더 고도계가 정상 작동 범위에 있도록 최대 28도 이내로 최대 뱅크각 제한

③ (V4.1) Auto-GCAWS를 위한 회복기동 제어기 개선

- 입력되는 최대 비행경로각, 최대 풀업가속도에 따라 회복기동 수행
- Auto-GCAWS 혹은 PARS 트리거시 회복기동 절차에 돌입
 - 1) 롤 자세 회복
 - 2) 최대 풀업가속도 명령 인가 (최대 비행경로각 도달 후 10초까지)
 - 3) 수평 회복 (5초간)
 - 4) 회복기동 종료

4. 변경 사유 (필요시 별도 문서 첨부)

- FTG SW 개선을 위한 탑재 모델 수정

5. 배포자

A. 이름
김성중, 이호형

B. 소속기관
KAIST

6. 첨부 (해석 결과 포함)

레이더 고도계의 직하방 측정치와 원뿔형 측정치 비교

- 원뿔형 방사패턴으로 획득한 레이더 고도계 측정치와 직하방 측정치를 비교한 그래프는 그림 9과 같다.
- 원뿔형 측정치는 방사된 좌표점과의 경사거리 가운데 최단 거리를 채택하기 때문에, 항상 직하방 측정치보다 작거나 같은 AGL 고도가 측정됨 (파란 선이 노란 선 보다 아래에 위치)

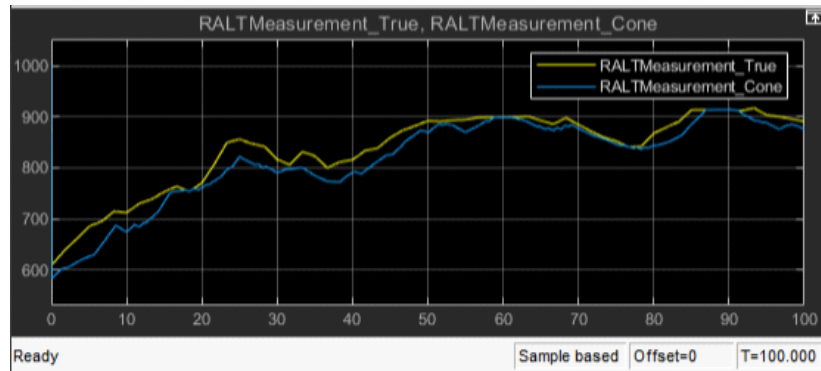


그림 9. 직하방 측정치와 원뿔형 측정치 비교

고도 제어기 개선 및 수정

- 연속되는 선회 기동 및 상승 기동 시뮬레이션을 수행한 결과, 고도 결과는 그림 10-11과 같음.
- 동일한 기동 수행시 기존 버전 대비 비슷한 상승률을 유지하면서 고도제어기의 반응에서 진동 현상이 줄어들음

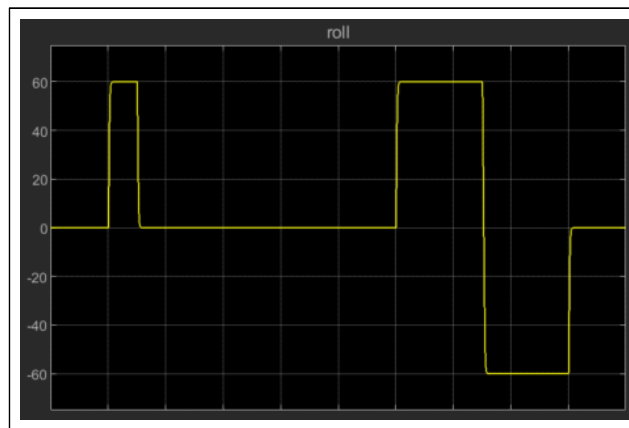


그림 10. 선회 기동 시 시간에 따른 롤각 반응

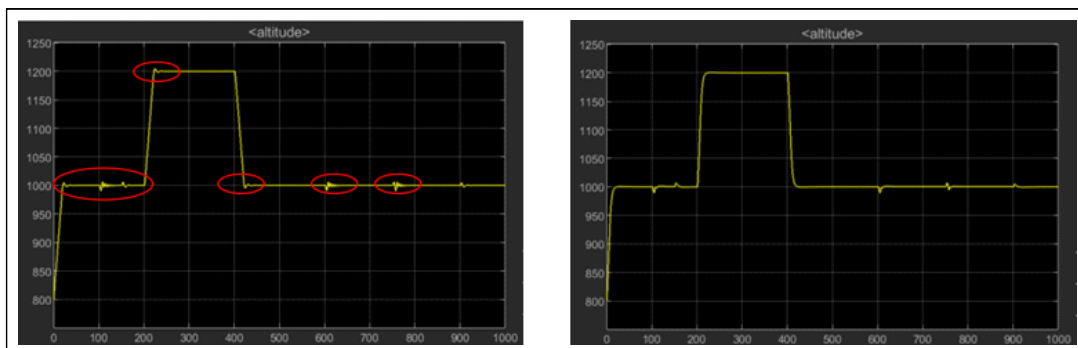
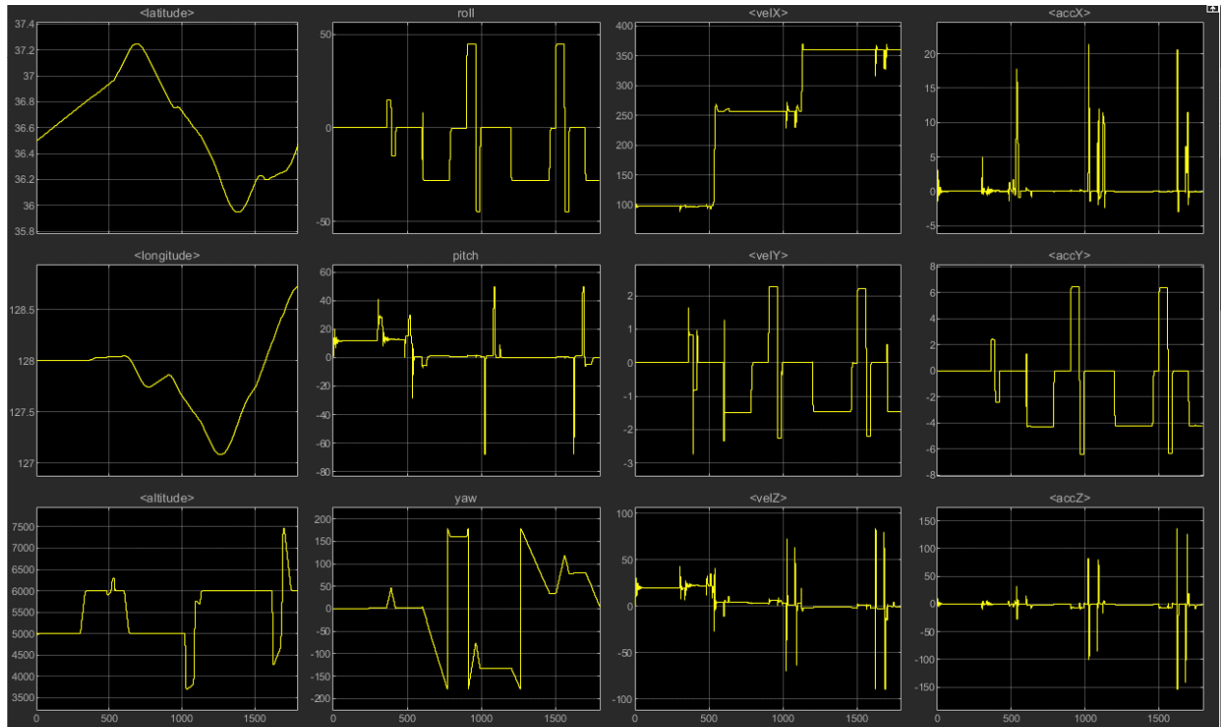


그림 11. 이전 버전 고도제어기와 수정된 고도 제어기 비교

FTG 오토파일럿 모드 작동 검증

- 입력 롤/피치 자세 유지 기능 정상
- 입력 기준고도 유지/변경 기능 정상
- 입력 기준속도 유지/변경 기능 정상



Ready

Sample based Offset=0 T=1800.000

Time	위도 (deg)	경도 (deg)	고도 (m)	기준 속도 (m/s)	설정 롤 자세 (deg)	설정 피치 자세 (deg)	설정 방위각 (deg)	switchRoll	switchPitch	switchTF	switchGCAS	비고
00:00	36.5	128	5000	100	0	0	0	0	2	1	1	ATT HOLD ALT HOLD TF SELECTED GCAWS SELECTED
05:00	36.5	128	6000	100	0	0	0	0	2	1	1	ATT HOLD ALT HOLD TF SELECTED GCAWS SELECTED
06:00	36.5	128	6000	100	45	0	0	0	2	1	1	ATT HOLD ALT HOLD TF SELECTED GCAWS SELECTED
06:30	36.5	128	6000	100	-45	0	0	0	2	1	1	ATT HOLD ALT HOLD TF SELECTED GCAWS SELECTED
07:00	36.5	128	6000	100	0	0	0	0	2	1	1	ATT HOLD ALT HOLD TF SELECTED GCAWS SELECTED
08:00	36.5	128	6000	100	0	0	0	0	1	1	1	ATT HOLD ATT HOLD TF SELECTED GCAWS SELECTED
08:05	36.5	128	6000	100	0	15	0	0	1	1	1	ATT HOLD ATT HOLD TF SELECTED GCAWS SELECTED
08:30	36.5	128	6000	100	0	40	0	0	1	1	1	ATT HOLD ATT HOLD TF SELECTED GCAWS SELECTED
08:40	36.5	128	6000	100	0	15	0	0	1	1	1	ATT HOLD ATT HOLD TF SELECTED GCAWS SELECTED
08:50	36.5	128	6000	257	0	0	0	0	2	1	1	ATT HOLD ALT HOLD TF SELECTED GCAWS SELECTED
10:00	36.5	128	5000	257	0	0	0	1	2	1	1	STRG SEL ALT HOLD TF SELECTED GCAWS SELECTED
15:00	36.5	128	5000	257	45	0	0	0	2	1	1	ATT HOLD ALT HOLD TF SELECTED GCAWS SELECTED
16:00	36.5	128	5000	257	-45	0	0	0	2	1	1	ATT HOLD ALT HOLD TF SELECTED GCAWS SELECTED
16:30	36.5	128	5000	257	0	0	0	0	2	1	1	ATT HOLD ALT HOLD TF SELECTED GCAWS SELECTED
17:00	36.5	128	5000	257	0	-70	0	0	1	1	1	ATT HOLD ALT HOLD TF SELECTED GCAWS SELECTED
17:05	36.5	128	5000	257	0	1	0	0	1	1	1	ATT HOLD ATT HOLD TF SELECTED GCAWS SELECTED
18:00	36.5	128	5000	257	0	50	0	0	1	1	1	ATT HOLD ATT HOLD TF SELECTED GCAWS SELECTED
18:10	36.5	128	5000	257	0	0	0	0	1	1	1	ATT HOLD ATT HOLD TF SELECTED GCAWS SELECTED
18:40	36.5	128	6000	360	0	0	0	0	2	1	1	ATT HOLD ALT HOLD TF SELECTED GCAWS SELECTED
20:00	36.5	128	6000	360	0	0	0	1	2	1	1	STRG SEL ALT HOLD TF SELECTED GCAWS SELECTED
25:00	36.5	128	6000	360	45	0	0	0	2	1	1	ATT HOLD ALT HOLD TF SELECTED GCAWS SELECTED
26:00	36.5	128	6000	360	-45	0	0	0	2	1	1	ATT HOLD ALT HOLD TF SELECTED GCAWS SELECTED
26:30	36.5	128	6000	360	0	0	0	0	2	1	1	ATT HOLD ALT HOLD TF SELECTED GCAWS SELECTED
27:00	36.5	128	6000	360	0	-70	0	0	1	1	1	ATT HOLD ATT HOLD TF SELECTED GCAWS SELECTED
27:05	36.5	128	6000	360	0	1	0	0	1	1	1	ATT HOLD ATT HOLD TF SELECTED GCAWS SELECTED
28:00	36.5	128	6000	360	0	50	0	0	1	1	1	ATT HOLD ATT HOLD TF SELECTED GCAWS SELECTED
28:10	36.5	128	6000	360	0	0	0	1	2	1	1	STRG SEL ALT HOLD TF SELECTED GCAWS SELECTED

그림 12. FTG 기능 시험 결과 및 입력 테이블

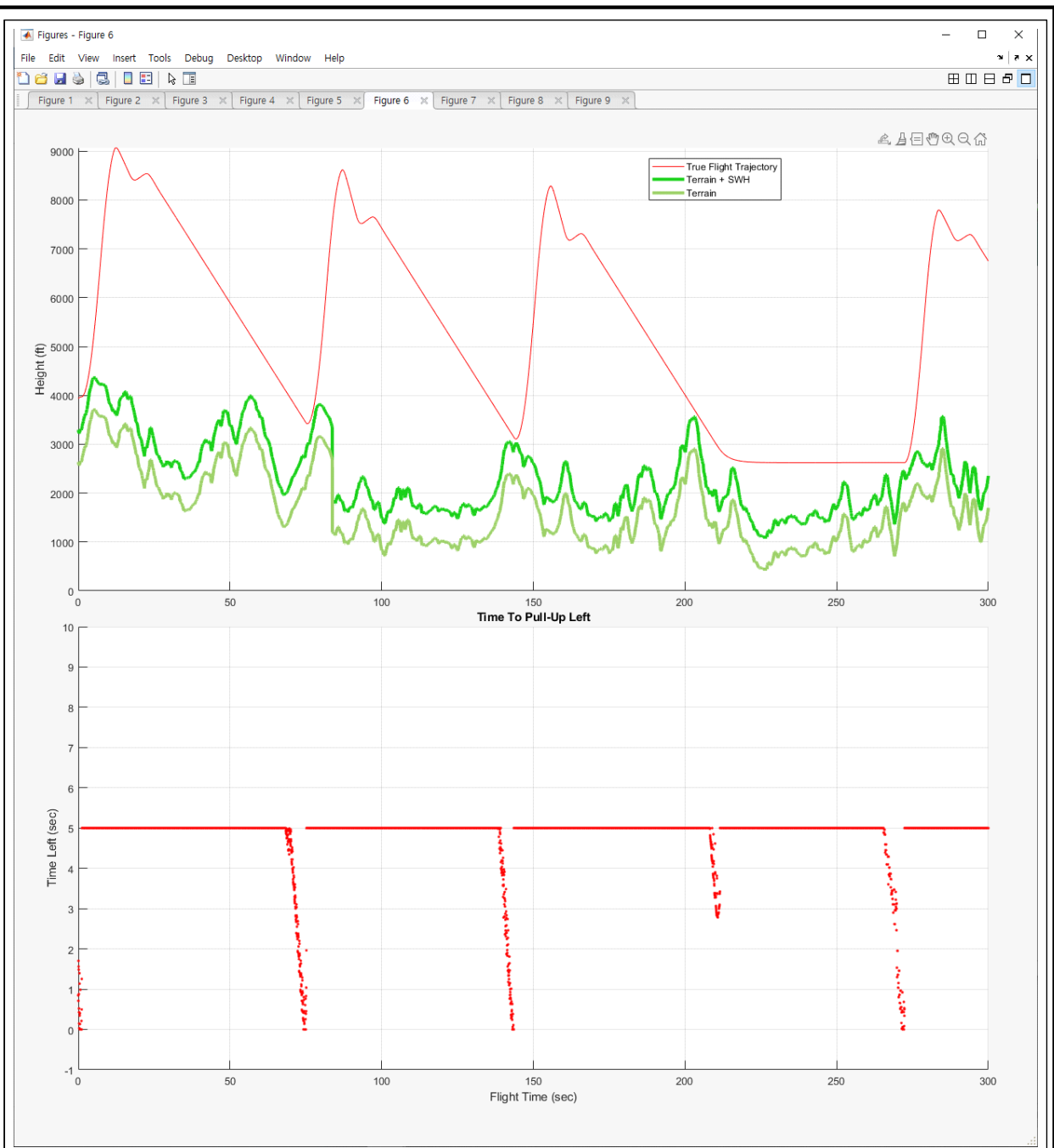


그림 13. Auto-GCAWS 시험 결과