

論文

J. of The Korean Society for Aeronautical and Space Sciences 44(6), 530-536(2016)

DOI: <http://dx.doi.org/10.5139/JKSAS.2016.44.6.530>

ISSN 1225-1348(print), 2287-6871(online)

레거시 시뮬레이터를 활용한 FLIGHTLAB[®] 모델 기반의 조종성 평가 환경 설계 연구

양창덕*, 이승덕**, 조환희**, 정동우**

A Design of Handling Quality Assessment Environment Based on FLIGHTLAB Model Using Legacy Simulator

Chang Deok Yang*, Seung Deok Lee**, Hwan Heui Cho** and Dong Woo Jung**

Korea Aerospace Industries, LTD.

ABSTRACT

The handling quality simulator including high fidelity flight mechanics model is indispensable component to design and verify the flight control system. Korea Aerospace Industries, LTD. (KAI) has been performing LCH (Light Civil Helicopter) core technology development program regarding automatic flight control system (AFCS) software development. And KAI has been developing flight mechanics model using FLIGHTLAB to design and evaluate the AFCS flight control law. This paper presents the handling quality assessment environment development results through the combining FLIGHTLAB with a legacy simulator. And this paper details the FLIGHTLAB model, application development process and FLIGHTLAB interface design. The developed handling quality assessment environment has been demonstrated with the ADS-33E hover and pirouette MTE (Mission Task Element) maneuver simulation.

초 록

고-신뢰도 모델을 이용한 조종성 평가 시뮬레이션 환경은 비행제어시스템의 설계/평가에 필수적으로 요구된다. 한국항공우주산업(주)에서는 소형민수헬기 핵심기술 개발과 관련하여 자동비행조종장치 소프트웨어 개발과제를 수행 중에 있으며 제어법칙 설계를 위한 비행동역학 모델 및 해석을 위해 상용 도구인 FLIGHTLAB을 이용하고 있다. 본 연구에서는 기존에 개발된 레거시 시뮬레이터를 FLIGHTLAB 모델과 연동하고 이를 조종성 평가에 활용한 내용을 다루었다. 본 논문에서는 외부 연동을 위한 FLIGHTLAB 모델의 설정, 연동 프로그램 개발 및 연동 방안에 대한 내용을 수록하였다. 또한 본 논문에서는 레거시 시뮬레이터와의 연동을 통해 ADS-33E-PRF의 호버 및 pirouette MTE 기동비행을 수행하고 평가 결과를 수록하였다.

Key Words : FLIGHTLAB, Legacy Simulator, Handling Quality Simulator(조종성 평가 시뮬레이터), Handling Quality Assessment(조종성 평가)

I. 서 론

비행 시뮬레이터는 그의 활용도에 따라 연구 개발과 조종사의 훈련 등의 분야에서 광범위하게 사용되고 있다. 훈련용 시뮬레이터는 항공기가 제작된 후 조종사가 지상에서 항공기의 동특성, 조종기술 및 운용절차 등의 관속 훈련을 위한 것으로 비행제어시스템을 포함한 대부분의 기능들을 소프트웨어적으로 구현하여 적용한다. 반면 연구개발용 시뮬레이터는 비행제어시스템을 실제 탑재 하드웨어로 적용하여 비행체의 운동을 제어하며, 액추에이터 및 센서 등의 하드웨어 또한 실제 하드웨어 또는 실제 하드웨어에 준하는 장비로 대체하여 실시간 시뮬레이션을 수행한다. 항공기 연구 개발용 시뮬레이터는 항공기의 초기 설계 과정에서부터 개발 및 검증 단계에까지 비행제어시스템의 성능 및 안전성 등의 평가에 필수적으로 사용되는 핵심 요소이다.

한국항공우주산업(주)에서는 조종성 국제규격(ADS-33E-PRF)을 만족하는 헬리콥터 전자식 비행제어(FBW: Fly-By-Wire) 시스템 기술 개발 과제를 수행 하였다[1,2]. 해당 프로그램을 통하여 주관기관 및 참여기관들은 연구개발용 조종성 평가 시뮬레이터와 함께 헬리콥터 조종성 국제규격 기반의 FBW 비행제어법칙, 비행역학모델(HETLAS: Helicopter Trim, Linearization and Simulation), 비행운용 소프트웨어 그리고 비행제어 컴퓨터 하드웨어를 개발하였다. 개발된 시뮬레이터는 HILS (Hardware-In-The-Loop Simulation) 시험 환경에서 FBW 시스템의 성능을 입증 하는데 성공적으로 사용되었다. 해당 프로그램을 통해 개발된 연구개발용 시뮬레이터는 설계한 비행제어법칙에 대한 조종성 평가, 비행역학 모델 평가 그리고 조종석 설계 등의 연구에 활용할 수 있다.

참고문헌[3]에서는 비행동역학 모델링 소프트웨어인 FLIGHTLAB을 이용하여 6자유도 모션시스템을 갖는 시뮬레이터를 개발하였다. 시뮬레이터의 충실도를 검증하기 위해 ADS-33E-PRF[4] 기반의 기동비행을 실 항공기와 시뮬레이터에서 각각 수행하고 조종성 평가와 관련한 변수들에 대해 시뮬레이터의 충실도를 평가하는 방법을 제안하였다. 참고문헌[5]에서는 KA-32T 훈련용 시뮬레이터의 개발을 위해 FLIGHTLAB을 이용하여 시뮬레이션 모델을 개발하였다. KA-32T 헬기의 비행시험을 수행하고 획득 데이터를 이용하여 FLIGHTLAB 모델을 튜닝하여 이를 훈련용 시뮬레이터에 적용하였다.

한국항공우주산업에서는 2015년 소형민수헬기(LCH: Light Civil Helicopter) 개발과 관련한 소형무장헬기 연계 민수헬기 국외업체 이전 기피 핵심기술 개발 과제를 착수하였다. 해당 과제에서는 자동비행조종장치(AFCs: Automatic Flight Control System) 소프트웨어를 개발할 계획이며 이를 위해 FLIGHTLAB을 이용하여 비행동역학 모델을 개발 중에 있다.

본 논문에서는 한국항공우주산업(주)에서 보유한 레거시 시뮬레이터와 FLIGHTLAB과의 연동 인터페이스의 개발내용을 다루었다. 또한 개발 결과를 이용하여 조종성 평가에 활용할 수 있는 가능성을 보이기 위해 ADS-33E-PRF에서 제시한 호버 MTE 기동 및 pirouette MTE 기동을 수행하고 그 결과를 수록하였다. 본 논문에서는 레거시 시뮬레이터가 FLIGHTLAB으로 개발한 다양한 모델링 환경과 통합하여 조종성 평가 환경으로 사용 가능하도록 하였다.

II. 본 론

2.1 FLIGHTLAB Modeling

FLIGHTLAB 소프트웨어는 다물체 동역학(Multi-Body Dynamics)을 기반으로 한 항공기 모델링 및 시뮬레이션 해석 도구로서 항공기 구성 요소 별로 물성치를 데이터베이스화 하여 모델링을 수행할 수 있다. 또한 FLIGHTLAB은 모델을 생성하거나 해석 수행을 위한 FLME와 Xanalysis와 같은 GUI 환경을 제공하며 다양한 수준의 모델링 기법을 제공하여 선택적인 사용이 가능하다. 한국항공우주산업(주)에서는 수리온 헬기에 대한 FLIGHTLAB 모델을 개발하였으며 비

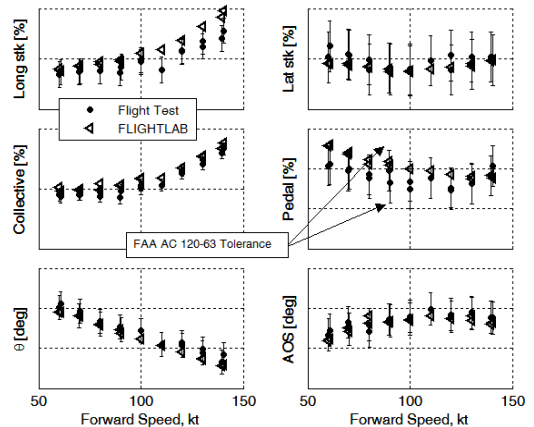


Fig. 1. FLIGHTLAB Analysis Results

행시험 결과를 바탕으로 모델의 충실도를 해석적으로 검토하였다. 수리온 모델의 주 로터는 플랩 및 리드-래그 자유도를 포함하는 articulated 로터 형상을 갖 요소이론을 이용하여 모델링 하였으며, 유입류 모델은 Petter-He finite state 모델을 사용하였다. 꼬리로터는 Bailey 로터로 모델링 하였다. 기체에 대한 공력모델은 동체 모델과 동체에 수평 및 수직 안정판을 장착한 scale-model 에 대한 풍동시험 데이터베이스를 활용하였다. Fig. 1은 FLIGHTLAB 모델의 전진비행 트림해석 결과와 비행시험 결과를 혼련용 시뮬레이터 검증 요구사항인 FAA AC120-63[6]에서 제시한 오차를 기준으로 비교한 결과이다.

FLIGHTLAB을 통해 해석한 트림비행의 조종 입력 및 자세 각 등의 결과가 비행시험 결과와 비교하여 FAA AC120-63 기준오차 이내로 예측되는 것을 알 수 있다.

2.2 FLIGHTLAB Interface Design

FLIGHTLAB 소프트웨어에서 제공하는 외부 인터페이스 기능은 standalone model을 이용하는 FCM (FLIGHTLAB Code-Gen Model) 그리고 FLIGHTLAB run-time 환경인 'scope'을 이용하는 FLCOMMS 및 NetFLC가 있다. FLCOMMS는

Table 1. FLIGHTLAB External Interface Function

	FCM	FLCOMMS	NetFLC
실행 환경	<ul style="list-style-type: none"> - 사용자가 개발한 application에서 모델의 실행이 가능함 - Standalone model license 필요함 	<ul style="list-style-type: none"> - Scope의 run-time 환경으로 모델의 시뮬레이션 실행을 제어함 - 사용자 application에서의 데이터 제공 시점과 모델 실행 시점의 동기화 어려움 	
주요 기능	<ul style="list-style-type: none"> <i>fcm_open</i> - Obtain handle for fcm file, varlist, field <i>fcm_close</i> - Free fcm resources <i>fcm_create</i> - Create an instance of a fcm model <i>fcm_invoke</i> - Call a model operation <i>fcm_read</i> - Read model data <i>fcm_write</i> - Write model data 	<ul style="list-style-type: none"> <i>flc_attach</i> - Attach the flcomms shared memory block <i>flc_open</i> - Open a connection to a varlist <i>flc_read</i> - Read varlist values into C variables <i>flc_write</i> - Write C variables values into varlist <i>flc_detach</i> - Detach the flcomms shared memory block 	

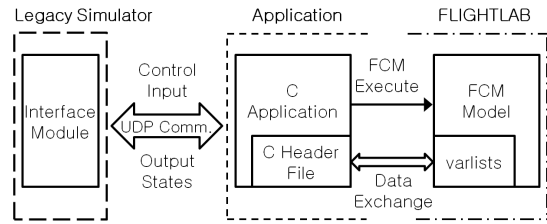


Fig. 2. FLIGHTLAB Interface Architecture

FLIGHTLAB의 외부 인터페이스를 위한 기능 중의 하나로 FLIGHTLAB run-time 환경 내에서 실행이 가능하다. NetFLC는 FLCOMMS기능을 통해 네트워크 내의 타 컴퓨터와 연동하여 시뮬레이션을 실행 할 수 있는 기능을 제공한다[7]. Table 1은 FCM, FLCOMMS 및 NetFLC의 실행 환경과 주요 기능에 대한 내용을 보여준다.

레거시 시뮬레이터의 환경을 이용하는 점을 고려하여 본 연구의 FLIGHTLAB 외부 인터페이스 설계는 시뮬레이션 수행의 제어(실행, 정지, 중지 등) 및 시뮬레이션 time step의 실행을 레거시 시뮬레이터의 호스트 컴퓨터가 제어할 수 있도록 설계되어야 한다. 따라서 본 연구에서는 FLIGHTLAB의 run-time 환경을 사용하지 않는 FCM library를 이용하여 연동 환경을 개발하였다. Fig. 2는 본 논문에서 개발한 FLIGHTLAB과 레거시 시뮬레이터의 전체적인 연동 인터페이스 구조를 보여준다.

FLIGHTLAB의 FCM standalone model은 object code로서 FLIGHTLAB run-time 시스템인 "Scope"을 사용하지 않고 사용자가 'C'언어로 개발한 연동 프로그램(application)을 이용하여 모델을 구동할 수 있는 환경을 제공한다. FLIGHTLAB standalone model을 이용한 모델 실행환경을 개발하기 위해서는 1) FCM (FLIGHTLAB Code-Gen) standalone model의 생성, 2) 사용자가 개발한 연동프로그램(application)과 연결하기 위한 "C" header file 생성 그리고 3) 연동프로그램(application) 개발의 절차가 필요하다.

2.2.1 FCM standalone model 생성

FCM (FLIGHTLAB Code-Gen) standalone model은 FLIGHTLAB의 development system을 통하여 생성한다. FCM standalone model은 'Xanalysis'에서 모델의 초기조건을 설정하고 트림 해석을 수행하여 모델을 트림상태로 설정한 후 'fcmdump' 명령을 이용하여 생성할 수 있다. FCM 모델이 외부 C 연동프로그램과 주고받는 데이터는 변수리스트인 'varlist'로 정의하여야 하며, 이는 FCM standalone 모델이 생성되기 전에

Table 2. FLIGHTLAB Pseudo-script for FCM Generation

FLIGHTLAB Script (*.exc)	Explanation
pushg(world_model_airframe_cpg_xaout) varlist @BODYSTV "Airframe states" = .. posxi, posyi, poszi, ... vxb, vyb, vzb, ... phi, theta, psi, ... popg;	- varlist: 외부 인터페이스 varlist 정의 - BODYSTV: 사용자가 정의한 varlist 이름 - "Airframe states" : 사용자가 정의한 varlist 설명
fcmdump(@BODYSTV);	- 외부 인터페이스 varlist 생성
exec("xatestcond.exc",1);	- 트림 초기조건 설정
exec("xamodeltrim.exc",1);	- 트림해석 수행
fcmdump("modelname.fcm");	- standalone model 생성 - Xanalysis에서 로딩한 모델을 FCM 모델로 변환함

정의되어 있어야 한다. Table 2는 FLIGHTLAB의 'varlist' 정의 및 FCM 모델 생성과 관련한 FLIGHTLAB pseudo-script를 보여준다. 즉, FCM 모델 생성을 위해서는 Xanalysis에서 FLIGHTLAB 모델을 로딩한 후 varlist를 이용하여 인터페이스에 사용할 변수를 정의/생성 하여야 한다. 그 후 설정한 트림조건으로 모델을 세팅하고 'fcmdump' 명령어를 이용하여 FCM 모델로 변환한다.

FCM 모델은 fcm_invoke 명령으로 모델이 호출될 때 모델이 갖고 있는 적분시간 주기에 따라 1-step 시뮬레이션을 수행한다. 따라서 FCM 모델의 생성 시 레거시 시뮬레이터 환경에서 모델을 호출하는 시간주기에 따라서 FCM 모델의 적분시간 주기를 설정해 주어야 정밀한 시뮬레이션을 수행할 수 있다. 본 연구에서는 수리온 헬기 주 로터의 분당 회전수가 272rpm 이며 레거시 시뮬레이터에서는 FCM 모델을 128Hz로 구동하는 점을 고려하여 FLIGHTLAB 모델 solution parameter의 'Number of rotor azimuth step/rev'를 28.239로 설정하였다.

2.2.2 C 헤더파일 생성

FCM 모델의 데이터 인터페이스를 위해서는 FLIGHTLAB varlist와 연동프로그램과의 데이터 관계를 정의한 C 언어 header 파일이 필요하다.

C header 파일은 FCM 모델 생성 시 정의한 varlist 데이터의 데이터 이름 및 데이터 변수 형과 일대일 정합되도록 작성하여야 한다. Table 3은 C header 파일과 varlist의 데이터 구조의 예

Table 3. C Header File Structure example

FCM varlist	'C' header 데이터 구조
pushg(world_model_airframe_cpg_xaout) varlist @BODYSTV "Airframe states" = .. posxi, posyi, poszi, ... vxb, vyb, vzb, ... phi, theta, psi, ... popg;	struct BODYSTV { double posxi, posyi, poszi; double vxb, vyb, vzb; double phi, theta, psi; };

Table 4. Pseudo-code for Application

Application script	Explanation
#include "fcm.h" #include "application.h"	FCM 헤더 및 사용자 헤더 포함
FCM_FILE fcm;	fcm 변수 선언
FCM_MODEL model;	모델 instance handle 선언
FCM_VARLIST vl_controls, vl_states;	varlist 변수 선언
FCM_OPERATION op_step;	operation handle 변수 선언
const char *fcm_filename ="/fbw/%TARGET/%TARGET.fcm"	fcm 모델 이름 변수 선언
fcm =fcm_open(fcm_filename)	fcm 모델을 open 하고 FCM_FILE handle 'fcm'을 가져옴
vl_controls= fcm_lookup_varlist(fcm, "@FLINPUT")	fcm 모델의 varlist 'FLINPUT'의 handle을 가져옴
vl_states=fcm_lookup_varlist (fcm, "@FLOUTPUT")	fcm 모델의 varlist 'FLOUTPUT'의 handle을 가져옴
op_step=fcm_lookup_operation(fcm, "STEP")	fcm 모델의 integration step에 대한 operation handle을 가져옴
model=fcm_create(fcm)	FCM_MODEL instance를 생성함
fcm_write(model, vl_controls, &FLIN, sizeof(FLIN))	fcm 모델의 varlist 'vl_controls'에 사용자 application에서 정의한 구조체 FLIN을 기록함
fcm_invoke(model, op_step)	fcm 모델을 integration step인 op_step만큼 실행함
fcm_read(model, vl_states, &FLOUT, sizeof(FLOUT))	fcm 모델의 varlist 'vl_states'를 사용자 application에서 정의한 구조체 FOUT으로 읽어들이
fcm_close(fcm)	fcm handle의 모델을 close함

를 보여준다.

2.2.3 연동 application 개발

FLIGHTLAB은 외부 application이 FCM 모델을 제어할 수 있는 수단으로 API utility를 제공한다[7]. FLIGHTLAB API utility의 사용은 FLIGHTLAB 설치 경로인 '\$FL_DIR/include/'에 있는 FCM 모델의 시스템 헤더 파일인 fcm.h를 사용자가 작성한 application에 include하고 빌드/링크 시 fcm 라이브러리인 '\$FL_DIR/lib/\$FL_MACHTYPE/libfcm.a' 을 포함하는 것으로 가능하다. 이를 통해 사용자가 작성한 application은 FLIGHTLAB API utility의 fcm_open, fcm_invoke, fcm_write 및 fcm_read 등의 함수를 이용하여 FCM 모델의 로딩, 실행 또는 varlist 데이터 값의 읽기/쓰기 등의 수행이 가능하다. FLIGHTLAB API utility 자료 형과 FCM 모델의 실행명령 함수를 이용하여 작성한 FLIGHTLAB 연동 application의 pseudo-code를 Table 4에 나타내었다.

본 연구에서는 작성한 application의 실행 파일의 구동을 위해 리눅스 셸 스크립트를 사용하였다. 레거시 시뮬레이터의 인터페이스 모듈이 ssh 터널을 통해 FLIGHTLAB standalone 모델 서버에 접속하여 셸 스크립트를 실행하면 application 실행 파일이 구동하도록 프로그램을 작성하였다.

2.3 Legacy 시뮬레이터를 이용한 조종성 평가

본 연구에서는 FLIGHTLAB과 레거시 시뮬레이터의 연동을 통해 개발한 조종성 평가환경의 활용성을 확인하기 위해 ADS-33E-PRF에 명시된 MTE (Mission Test Element) 기동을 수행하였다. 본 논문에서는 호버 기동비행과 pirouette 기동비행의 수행 결과를 수록하였다. Fig. 3 및 Fig. 4는 각각 ADS-33E-PRF에서 제시한 호버기동과 pirouette MTE 기동의 코스를 보여준다. 호버 기동은 속도 6~10kts와 방향각 45도로 목표지점으로 접근하다 감속하여 지면 및 전방의 참조점에 대해 호버상태를 유지하는 기동이다. Pirouette 기동은 반경 100ft의 지면 참조점을 따라 기수를 원 중심으로 유지하며 비행하는 기동이다.

Figure 5는 레거시 시뮬레이터와 호버 MTE 수행을 위한 영상 데이터베이스를 보여준다. 본 연구에서 사용하는 레거시 시뮬레이터 및 FLIGHTLAB 연동 환경은 조종입력에서 영상이 시현되기까지의 데이터 transport delay가 훈련용 시뮬레이터 인증 기준 FAA AC120-63[6]에서 요구하는 100ms 이하가 되도록 설계하였다.

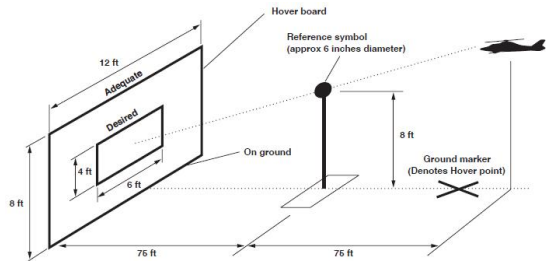


Fig. 3. Hover MTE Course Side View

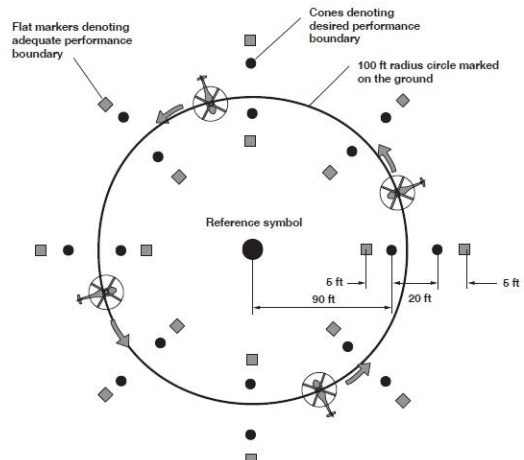


Fig. 4. Pirouette MTE Course



Fig. 5. Handling Quality Simulator and Visual Database for Hover MTE

시뮬레이션 수행을 위해 헬리콥터 전자식 비행제어(FBW: Fly-By-Wire) 시스템 기술 개발 과제를 통해 개발한 attitude command/attitude hold(ACAH) 제어기[7]를 사용하였다. 기동비행 시뮬레이션은 연동 환경 자체의 성능검토를 위해 대기 외란이 없는 조건으로 시험을 수행하였다.

Table 5. Performance Requirement for the Hover MTE and Simulation Results

Requirement	Desired	Adequate	Simulation Results
Attain stabilized hover within X seconds of initiation of deceleration	5	8	8+α
Maintain a stabilized hover for at least X seconds	30	30	30
Maintain the longitudinal and lateral position within ±X ft on the ground	3	6	3
Maintain altitude ±X ft	2	4	1
Maintain heading within ±X°	5	10	1
There shall be no objectionable oscillations during the transition to hover or during the stabilized hover	Applies	N/A	Satisfied

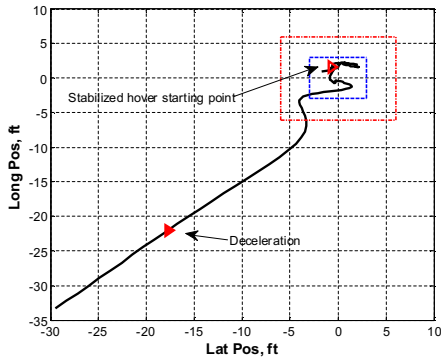


Fig. 6. Plan Position in Hover MTE

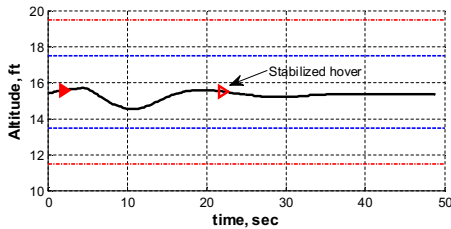


Fig. 7. Altitude in Hover MTE

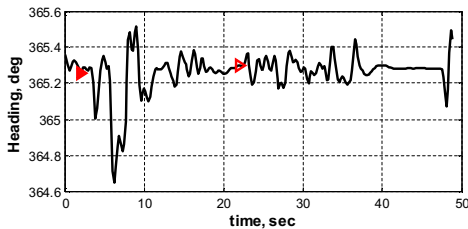


Fig. 8. Heading in Hover MTE

Table 6. Performance Requirement for the Pirouette MTE and Simulation Results

Requirement	Desired	Adequate	Simulation Results
Maintain a selected reference point on the rotorcraft within ±X ft of circumference of the circle	10	15	10
Maintain altitude ±X ft	3	10	3
Maintain heading so that the nose of the rotorcraft points at the center of the circle within ±X°	10	15	15
Complete the circle and arrive back over the starting point within:	45	60	45
Achieve a stabilized hover within X seconds after returning to the starting point	5	10	5
Maintain the stabilized hover for X second	5	5	5

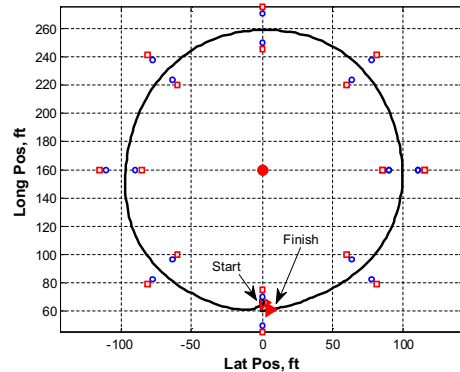


Fig. 9. Plan Position in Pirouette MTE

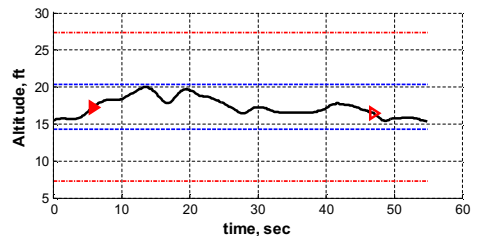


Fig. 10. Altitude in Pirouette MTE

Table 5는 ADS-33E-PRF에서 제시하는 utility 항공기에 대한 hover 기동의 성능 요구사항과 FLIGHTLAB 연동을 통해 수행한 기동비행 시뮬레이션 결과를 보여준다. Fig. 6~8은 시뮬레이터

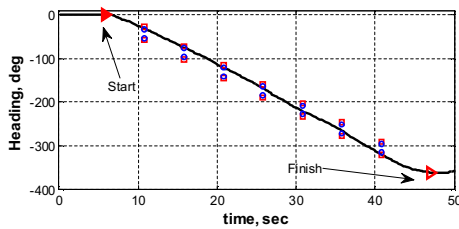


Fig. 11. Heading in Pirouette MTE

상에서 기동비행 시뮬레이션 수행의 결과인 중/횡 방향 항공기 위치, 고도 및 기수 방향각을 보여준다. 감속기동 후 hover 상태로 안정화 하는데 걸리는 시간은 ADS-33 요구사항에 불만족 하였지만 중 및 횡 방향 위치 유지, 고도 및 방향각에 대한 결과는 Desired 요구사항을 만족하는 결과를 보였다.

Table 6은 pirouette 기동의 성능 요구사항과 기동비행 시뮬레이션 결과를 보여준다. Fig. 9~11은 중/횡 방향 항공기 위치, 고도 및 방향각에 대한 기동 결과를 보여준다. 기동비행 중 기수 방향각이 일부 10도를 벗어나는 구간이 발생하였으나 그 외의 항목들은 모두 ADS-33의 Desired 요구사항을 만족하는 결과를 보였다.

III. 결 론

한국항공우주산업(주)에서는 소형 민수헬기 개발과 관련한 자동비행조종장치 소프트웨어 개발 과제를 수행중이다. 본 연구에서는 해당 과제를 통해 개발하는 FLIGHTLAB 모델을 이용한 조종성 평가환경 개발을 위해 기존 보유 시뮬레이터와 FLIGHTLAB 모델과의 연동 환경 개발에 관한 내용을 다루었다. 본 논문에서는 외부 연동을 위한 FLIGHTLAB의 모델 설정, application 개발 및 연동 방안에 대한 내용을 수록하였다. 또한 레거시 시뮬레이터와의 연동을 통해 ADS-33E-PRF의 호버 및 pirouette MTE 기동을 수행하여 평가 결과를 수록하였다. 본 논문에서는 FLIGHTLAB의 연동 인터페이스 개발을 통하여 레거시 시뮬레이터의 활용성을 보였다.

후 기

본 논문은 산업통상자원부 산업기술혁신사업으로 지원된 연구결과입니다. [10053157, 소형무장헬기 연계 민수헬기 핵심기술 개발사업]

References

- 1) Chang Deok Yang, Ho-che Jung, Chang-Joo Kim, et.al, "A Design of Helicopter Control Law Rapid Prototyping Process Using HETLAS," *J. of The Korean Society for Aeronautical and Space Sciences*, Vol. 43, No. 8, 2015, pp.731~738.
- 2) In-Je Cho, "Helicopter Digital Flight Control(FBW) System Development Based on Flying Quality Standard Specification(ADS-33)," *KSAS Conference*, NOV, 2011.
- 3) Mark, W., Gareth, D. Padfield., et.al, "Acceptance Testing of a Rotorcraft Flight Simulator for Research and Teaching," *35th European Rotorcraft Forum*, 2009.
- 4) anon., "ADS-33E-PRF, Handling Qualities Requirements for Military Rotorcraft," *US Army, Mach*, 2000.
- 5) Jasper, van der Vorst., et. al., "Flight Mechanics Model Development for a KA32 Training Simulator," *35th European Rotorcraft Forum*, 2009.
- 6) anon., "FAA Advisory Circular AC120-63, Helicopter Simulator Qualification," *Federal Aviation Authority*, Nov. 1994.
- 7) anon., "FLIGHTLAB Real-time Application Tutorial," *Advanced Rotorcraft Technology, Inc.*, Jun. 2006.
- 8) Jong-Sub Kim, et. al., "A Study on Design and Validation of Model Following Control Law for Fly By Wire Helicopter," *The Journal of Aerospace Industry*, Vol. 75, 2012, pp.138~156.