Advanced Development of a Dynamic Test Bed for Flight Management Systems

Dongsheng Wang

A Thesis

in

The Department

of

Mechanical and Industrial Engineering

Presented in Partial Fulfillment of the Requirements
for the Degree of Master of Applied Science at
Concordia University
Montreal, Quebec, Canada

April 2003

National Library
of Canada

Acquisitions and
Bibliographic Services

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale
du Canada

Acquisitions et
services bibliographiques

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

Our file Notre référence

0-612-77704-9

ABSTRACT

Advanced Development of a Dynamic Test Bed for Flight Management Systems

Dongsheng Wang

The Flight Management System (FMS) is playing an increasingly important role during the flight task of today's flights. This importance prompts a need to test the dynamic behaviors of the FMS in order to continue its development. The Dynamic Test Bed (DTB), simulating a real time flight in a laboratory environment, is an effective and economical way to test FMS functions.

The primary structure and the basic functions of the DTB are built in a former thesis [31]. The further development of FMS also requires that DTB is further developed. In this thesis, some new important functions of DTB are developed, and are embedded into the DTB to enable the development of the FMS. First, the FMS VNAV mode that can be engaged for capturing the GPS approach glide slope is developed for the AFCS on the DTB. The DTB can be used to test the vertical guidance of the FMS through the VNAV mode. Second, an IRS navigation subsystem is developed for flight simulation; the DTB has the ability to test the IRS navigation of the FMS. Third, the DTB GUI is redesigned in such a way to increase its ergonomic functionality of the DTB. Finally, an ARINC 429 Channel Switching Software tool and its GUI are developed. This tool improves the flexibility of the DTB. All these new functions are also integrated into the DTB and tested at Concordia University, as well as at the FMS Test and Development Lab of CMC Electronics where the DTB is currently being used.

ACKNOWLEDGEMENTS

The author would like to thank Dr. K. Demirli and Dr. J.V. Svoboda for their guidance and invaluable suggestions during all stages of the project.

The author would like to thank Mr. Narendra Gollu and Nianfu Chui for their invaluable assistance and cooperation.

The author would like to thank CMC Electronics and their staff who gave me the opportunity to work on the project.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

ψ	Euler's angles of the yaw
θ	Euler's angles for the pitch
φ	Euler's angles for the roll
γ	Air constant
α	Ideal acceleration rate
M	Mach number
P_0	Sea level pressure
ρ_0	Sea level air density
d_{ref}	Reference vertical deviation
d	GPS VDEV
d_e	Vertical deviation error
θ	Actual aircraft pitch angle
θ_c	Target pitch command
V	Aircraft airspeed vector
γ	Aircraft flight path angle
γ_G	Nominal glide slope angle
a	Acceleration centripetal
Alt	Aircraft altitude
Γ	The angular deviation of the ILS glide slope beam

LIST OF ACRONYMS AND ABBREVIATIONS

ADC	Air Data Computer
AFCS	Automatic Flight Control Systems
AHRS	Attitude and Heading Reference System
ARINC	Avionics Radio Inc.
ASCB	Avionics Standard Communication Bus
BCD	Binary Coded Decimal
BNR	Binary
CDU	Control Display Unit
CRT	Cathode Ray Tube
CSDB	Commercial Standard Digital Bus
DME	Distance Measuring Equipment
DTB	Dynamic Test Bed
EFIS	Electronic Flight Instrumentation Systems
ETA	Estimated Time of Arrival
FAF	Final Approach Fix
FANS	Future Air Navigation System
FCC	Flight Control Computer
FMS	Flight Management System
FMU	Flight Management Unit
GPS	Global Positioning System

GUI	Graphical User Interface
HIL	Horizontal Integrity Limit
ICAO	International Civil Aviation Organization
IRS	Inertial Reference System
IAS	Indicated Air Speed
LANV	Lateral Navigation
MAP	Missed Approach Point
MCP	Mode Control Panel
MFC	Microsoft Fundamental Classes
MSL	Mean Sea Level
NDB	Non-Directional Beacon
NMS	Navigation Management Systems
PRAIM	GPS Predictive Receiver Autonomous Integrity Monitoring
SDI	Source/Destination Identifier
SDI	Standard Instrument Departure
SSM	Sign Status Matrix
STAR	Standard Terminal Arrival Routes
TAS	True Airspeed
TCP/IP	Transmission Control Protocol/Internet Protocol
VHF	Very High Frequency
VLF	Very Low Frequency
VANV	Vertical Navigation

VOR	VHF Omnidirectional Range
VS	Vertical Speed

CHAPTER 1

INTRODUCTION

The Dynamic Test Bed (DTB) is effectively a simplified aircraft simulator that recreates the signals for Flight Management System (FMS). The FMS acts as if it is in an aircraft. The DTB has the core software of a flight to simulate all of the interfaces of a FMS. Thus, the DTB allows for testing of the FMS in a dynamic environment. The use of the word 'dynamic' here means that all of the signals are changing simultaneously as they do in the aircraft.

In this chapter, the background knowledge and the objectives of this thesis are introduced. The FMS that acts as the basic background knowledge of this thesis is introduced first; then a detailed overview of the Dynamic Test Bed is given; finally, the objectives of this thesis are stated at the end of this chapter.

1.1 Flight Management System (FMS)

The Flight management System (FMS) is an airborne computer that manages most navigation tasks during the flight. It carries the flight plan (the route the aircraft intends to take). This route includes the vertical portion of the flight and the cruise at altitude and descent [7].

The FMS is able to control the flight from takeoff through to landing, performing all of the complex navigational calculations that otherwise the crew would have to do. Typical functions include determining waypoints. course intercepts. estimated time of arrival. holding patterns. altitude crossing restrictions. optimum holding speed and fuel-burn data [7][8][9].

Before the flight. the pilot enters the flight plan as a sequence of waypoints from the departure airport to the arrival airport. including the cruise altitude. During the flight. the FMS constantly verifies its position using as many sources as possible. These include Global Position System (GPS) receivers, the Inertial Reference System (IRS). Air Data Computers (ADC) and radio beacons. With an established position. the FMS supplies steering commands to the aircraft's autopilot. The FMS greatly reduces pilot's workload. freeing the crew to manage high-level cockpit tasks effectively during all phases of flight [1][2].

1.1.1 Components of FMS

The FMS consists of two elements: the Flight Management Unit (FMU) and the

Multifunction Control Display Unit (MCDU).

1.1.1.1 Flight Management Unit (FMU)

The FMU is an airborne computer located in the aircraft's avionics bay. A commercial airliner normally carries two, or even three FMUs. The FMU accepts data from external navigation sensors and may contain a built-in GPS sensor. It performs the signal processing and computations required generating the high performance navigational data. The FMU accepts operator commands from the MCDU. and provides navigation. steering and status data to the MCDU and primary navigation flight displays. The FMU also provides suitably formatted outputs for the flight guidance system [24].

1.1.1.2 Multifunction Control Display Unit (MCDU)

The MCDU has an interface with the crew, external sensors. external systems and the FMU. The MCDU controls and shows FMU data. The MCDU provides a color display of alphanumeric data. An alphanumeric keyboard allows data entry. data editing and system control. An internal processing capability and ARINC 429 digital interfaces provide for its integration with other avionics subsystems [24].

1.1.2 FMS Functions

The main function of the FMS is to compare a route (lateral and vertical) to the airplane's position. This data is used to produce guidance and thrust commands that control the airplane on the activated route. In addition, the FMS sends display data to the integrated display system. The FMS performs the following functions [1-3][24]:

- Produces a single focal point that allows the crew to select, activate and adjust a three dimensional route structure from the internally stored data.
- Decreases the crew workload on charts and manuals. Provides an autotune for some navaids.
- Calculates guidance and thrust commands to automatically fly the selected route.
- Shows data to monitor dynamic conditions for the route.

1.1.3 Internal Data, Inputs and Outputs of FMS

The FMU is connected to other avionics subsystems. The internal data, including the navigation database, performance database and operational program, are stored in the FMU. The FMU receives the external inputs from the MCDU and navigational sensors. And then performs the calculations and comparisons based on the internal data. The FMU then sends the outputs to the flight control systems and display systems in the aircraft to manage the flight [24]. Figure 1 is a simple chart that displays the whole system. The internal data and the inputs and outputs of the FMS are introduced next.

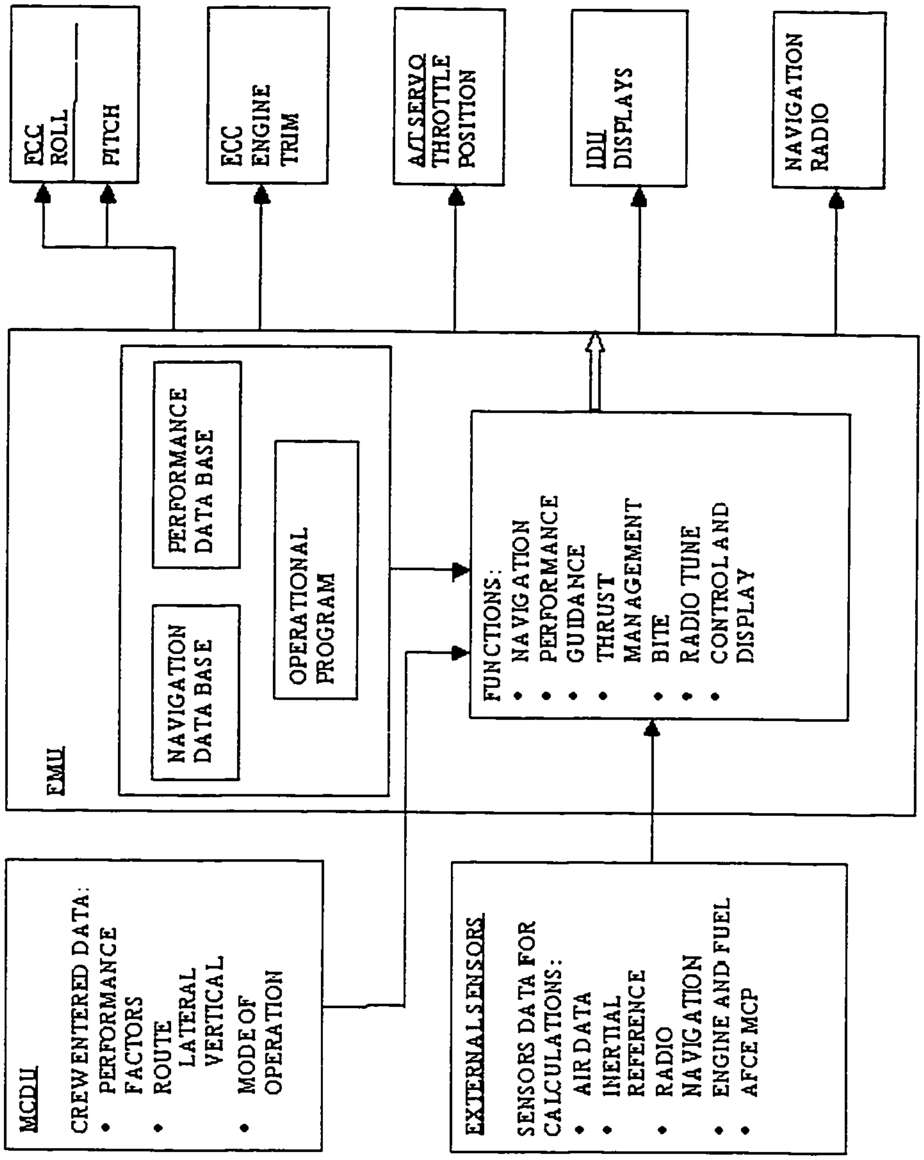


Figure 1 The Internal data, Inputs and Outputs of the FMS

1.1.3.1 Internal Data

The basis of the FMS capabilities is in the FMU program software that is stored in the

FMU as internal data. The internal data includes the navigation database, the performance database, the operational program and the thrust management function.

- The navigation database determines the route selection. It contains data on items such as airports, procedures, waypoints and navaids. The navigation data is valid for a specified period of time. At the end of the time period, a new navigation database must be loaded into the FMS. When the airplane is on the ground, a new navigation database can be loaded into the FMS by means of an external portable data loader.
- The performance database determines the dynamic operation of the airplane. The minimum and maximum limits are determined by the aircraft design.
- The operational program determines which sensors are used for calculations and how the calculations are done. It also commands the autopilot and autothrottle to the selected lateral and vertical profile.
- The thrust management function produces thrust limit commands, autothrottle servo control and thrust mode annunciation.

1.1.3.2 FMS Inputs

The inputs of the FMS are the MCDU and the external sensors:

- The MCDU allows the crew interface to interact with the FMS for preflight, inflight, data display, mode information and other requests.
- The external sensors supply data to compute the lateral and vertical aircraft position. The navigation sensors are the main external sensors connected with the FMS. Generally, the primary short-range sensors are ILS, VOR/DME and DME/DME; the

long-range sensors are GPS and IRS.

1.1.3.3 FMS Outputs

The FMS outputs the guidance commands that are changed to control surface movements by the Flight Control Computer (FCC). The FMS supplies the thrust trim commands to the Electronic Engine Control (EEC) for engine trim equalization. It supplies thrust trim commands to the Integrated Display System (IDS) for display and for the crew to monitor. The navigation radios are tuned by the FMS also.

1.1.4 Developing and Testing FMS

The FMS is an important part of the automatic flight guidance system that is used for flight planning in all modern aircrafts [3][4][5]. Today's FMS has introduced operational advantages and significant cost savings, e.g., by offering the possibility of an automatic, fuel-efficient flight from take-off to landing and reducing the pilot's workload [6][10]. A tool to test the functions of the FMS should be developed. This is the key step during the development of the FMS. Generally, before the FMS is used in a real flight, it should be tested with a flight simulator. Developing a Dynamic Test Bed (DTB) that simulates the dynamic flight with software is an effective and economical way to test the behaviors of the FMS in the lab. In the following section, a detailed overview of the DTB is given.

1.2 Dynamic Test Bed (DTB) Overview

Dynamic Test Bed (DTB) is an effective and economical way to test the dynamic functions of the FMS in real time flight simulation. The DTB is divided into two parts: the software and the hardware. In order to understand the DTB, the DTB hardware is introduced first.

1.2.1 DTB Hardware Overview

The hardware structure of the DTB is illustrated in Figure 2. The hardware includes the simulation PC. the interface PC. the breakout box and the interface cards.

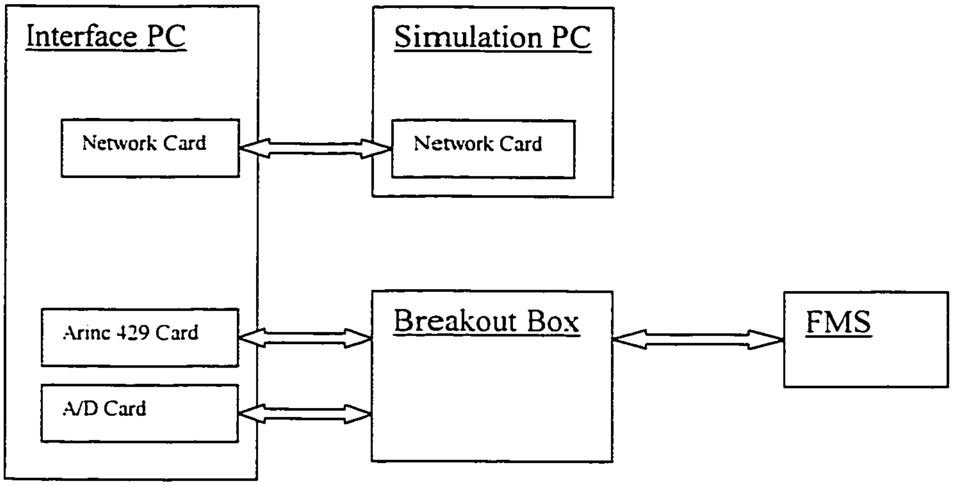


Figure 2 The Hardware Block Of the DTB

The simulation PC and the interface PC are both standard PCs with the network cards installed. They are connected together through the network. The simulation PC controls the flight simulation and display tasks during execution of the simulation. The interface cards (including the ARINC 429 card and analog/discrete card) are installed in the interface PC. Through these cards, the interface PC connects with the breakout box. The interface PC manages the interface cards and the data transfer between the flight simulator and the FMS. The breakout box is custom designed for the DTB. It acts as the interconnect box for all wiring between the FMS and the DTB.

1.2.2 DTB Software Overview

The DTB software consists of the Flight Simulation Software that includes the User Interface Software and the Flight Dynamic Simulation Software, the FMS Interface Software and the ARINC429 Channel Switch Software. The overview of the DTB software is shown in Figure 3. All DTB software is developed with Object Oriented Programming technology and developed under the Microsoft Visual C++ compiler.

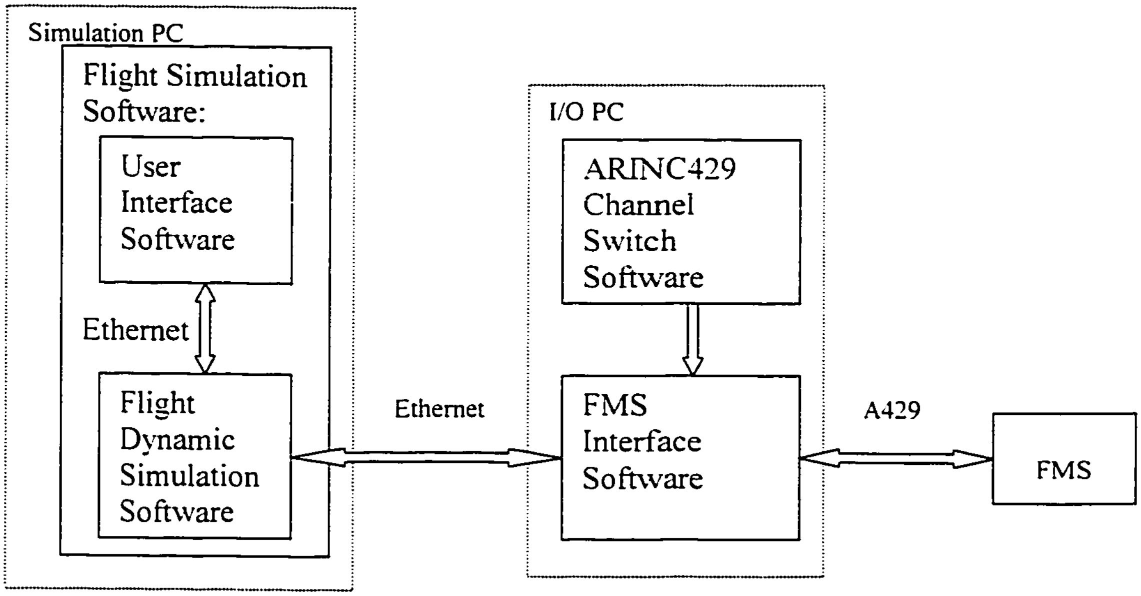


Figure 3 DTB Software Overview

1.2.2.1 The Flight Simulation Software

The Flight Simulation Software includes the Flight Dynamic Simulation Software and the User Interface Software. They communicate with each other through Ethernet. They are explained below.

- The Flight Dynamic Simulation Software is the core of the Flight Simulation Software. In fact, the Flight Dynamic Simulation Software acts as a specific aircraft type that is stored in the database of the FMS that the DTB is testing. On the basis of VPI FLSIM, a commercial flight software package, the Flight Dynamic Simulation Software is developed to simulate the dynamic aircraft performance. The simulation includes the flight model, navigation model, autopilot model, environment model and network software. The Flight Dynamic Simulation Software has such functions: receiving the control commands from the User Interface Software to control the flight simulation, transmitting the navigation data from navigation model to interface PC, receiving the FMS steering commands from the interface PC to autopilot model of the aircraft, and sending the flight data to the User Interface Software to display or record. The network software carries out all of the communication tasks listed above.
- The User Interface Software is the interface between the DTB and the user. It has a graphical user interface (GUI), which is called DTB GUI. Through DTB GUI, the user can exert a powerful control on the dynamic flight simulation. The user can select the autopilot mode, control the basic control surfaces of the aircraft, and set the atmospheric environment including the wind and temperature for the simulation. DTB GUI also includes function buttons for a flight freeze, reposition and unfreeze

for further control of the flight simulation process. The User Interface Software has the display function to display basic flight data in real time, and a recording function to record data chosen by the user as a file or as a graphic drawing of the curve of the recording data.

1.2.2.2 The FMS Interface Software

The FMS Interface Software is the interface between the flight simulator and the FMS. The FMS Interface Software initializes and configures the drivers of the interface cards at the beginning of the running of the DTB software. The FMS Interface Software controls the data exchange between the flight simulation and the FMS in real time, at the same time that it is converting data between an engineering format and the ARINC429 format.

1.2.2.3 The ARINC429 Channel Switch Software

The ARINC429 Channel Switch Software is a tool to increase the flexibility of the DTB. It has a graphical user interface for the user to reconfigure the ARINC channel setup of the DTB. It can produce the ARINC channel configuration files that are loaded to initialize the ARINC channels in DTB by the FMS Interface Software. The ARINC429 Channel Switch Software is explained in Chapter 4.

1.2.3 DTB Overall Structure

The hardware and software of the DTB have already been discussed above. Figure 4 illustrates the entire DTB system including both the hardware and software components.

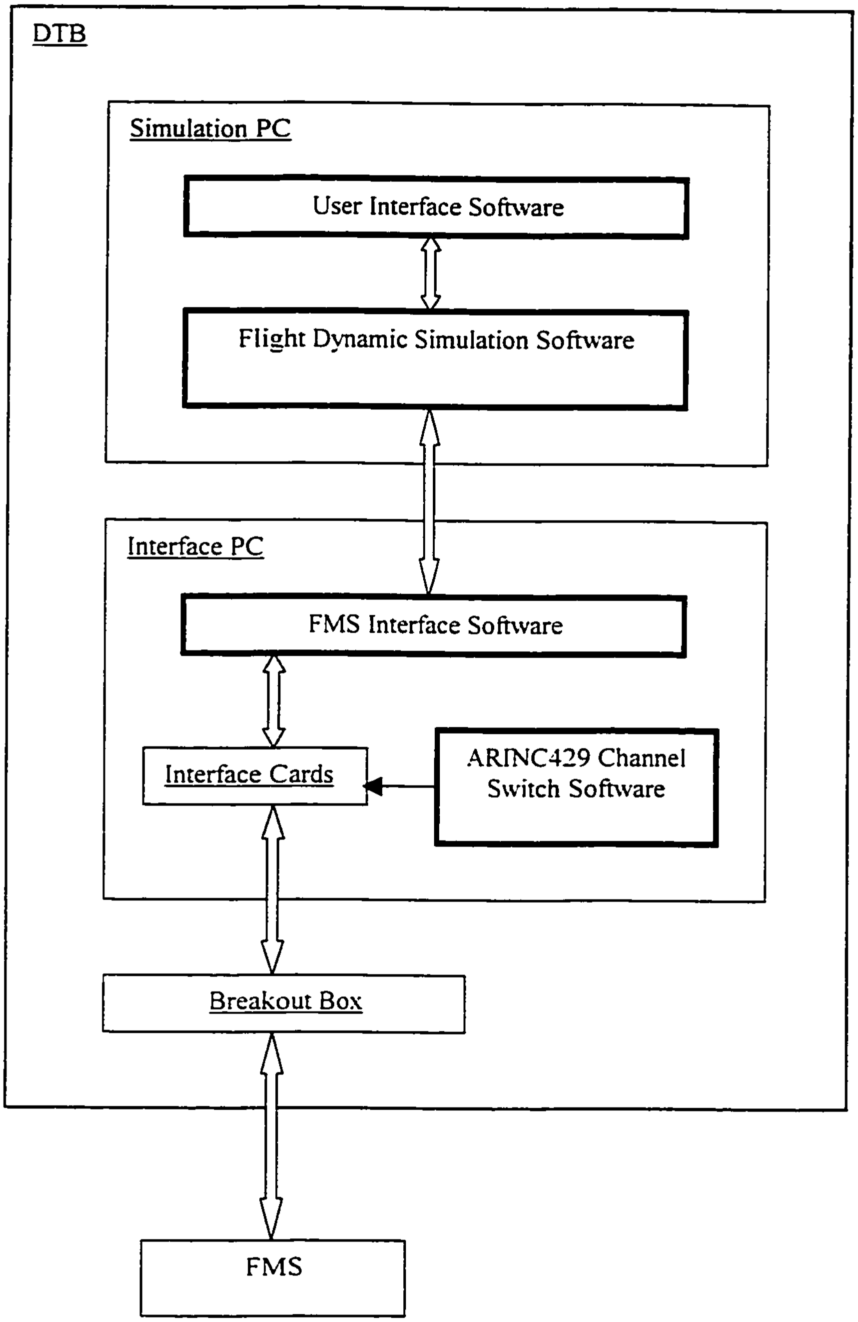


Figure 4 DTB Overall Structure

1.3 Objectives of This Thesis

In order to catch up with the development of the FMS, the DTB should also be developed simultaneously. Although the primary structure and the basic functions of the DTB are built by Chui [31], there are lots of further works to be done to guarantee that the DTB remains a viable tool. For such purpose some new important functions are developed and added into the DTB in this thesis. Specific objectives of this thesis are outlined below.

1.3.1 Developing the FMS VNAV Mode

Navigation functions are the most important functions of the FMS. The FMS navigation functions include the Lateral Navigation (FMS LNAV) and the Vertical Navigation (FMS VNAV). Therefore the DTB should have FMS LNAV mode and FMS VNAV mode to test FMS LNAV and FMS VNAV behavior of the FMS. The FMS LNAV autopilot mode of the DTB has been developed by Chui [31].

In this thesis, a FMS VNAV autopilot mode is developed, tested and embedded into the flight dynamic simulation of the DTB. A GPS glide slope control law is not only presented theoretically but also coded in C++. Development of the FMS VNAV mode is a major objective of this thesis and it is discussed in Chapter 3.

1.3.2 Developing the IRS Navigation Sensor

In order to test the navigation functions of the FMS, the DTB must provide the navigation source data in real time from the dynamic flight simulation to the FMS. Therefore it is very important to develop the navigation sensor modes in the dynamic flight simulation. There are various navigation sensors in the Boeing 747 avionics systems, such as GPS, IRS, AHR and ADC and so on, which are connected to the FMS. All such navigation sensors should be developed in the DTB to test the FMS navigation functions. The GPS, AHRS, and ADC navigation sensors have been developed by Chui [31]. Developing the IRS navigation sensor is another objective of this thesis, and it is discussed in Chapter 3.

1.3.3 Redesigning the DTB GUI

For the DTB working as a useful and applicable tool for user in developing the FMS, its ergonomic functionality is one of critical criterions that are used to evaluate the DTB. DTB GUI is the interface between user and DTB. In this thesis DTB GUI is redesigned. Comparing with the old GUI, the new one is clearer and more user friendly. Another objective of this thesis is to improve the ergonomic functionality of the DTB. and this objective is explained further in Chapter 4.

1.3.4 Developing the ARINC429 Channel Switch Software

Running the DTB to test all behaviors of the FMS is a key step during developing FMS. Each time the DTB is run to test the FMS, the user may use different ARINC429 channel setup. In order to reduce the workload for the DTB user, the ARINC429 Channel Switch Software is developed. This software should be able to produce the channel configuration files automatically according to what the DTB user desires. The flexibility and user friendliness of the DTB is enhanced by development of the ARINC429 Channel Switch Software. Developing the ARINC429 Channel Switch Software is another objective of this thesis, and this objective is discussed in Chapter 4.

CHAPTER 2

FLIGHT SIMULATION SOFTWARE

The objectives of this thesis have been introduced in Chapter 1. Three of those objectives are relative to the Flight Simulation Software in DTB. The FMS VNAV mode and IRS sensor mode are developed and embedded into the Flight Dynamic Simulation Software: DTB GUI that is redesigned in this thesis is a part of the User Interface Software. Therefore it is necessary to understand the Flight Simulation Software of DTB before those three objectives are discussed. Since the Flight Simulation Software includes the Flight Dynamic Simulation Software and the User Interface Software, both of them are introduced and explained in this chapter.

2.1 Flight Dynamic Simulation Software

The Flight Dynamic Simulation Software is developed on the basis of the commercial flight simulator software package, FLSIM, developed by Virtual Prototype Inc. What follows is a discussion of how to use FLSIM to configure an aircraft model and develop the modes of the flight simulation to meet the requirements of the DTB. First, FLSIM v7.0 is introduced, and then the flight simulation models of the Flight Dynamic Simulation Software including aircraft models, AFCS models and navigation models are introduced.

2.1.1 Introduction to FLSIM V7.0

FLSIM is a commercial software package designed by Virtual Prototypes Inc. to assist the development of flight simulation programs [22]. It transforms the aerodynamic coefficients provided by the user into a flight simulation that matches the performance of the desired aircraft. FLSIM provides a flexible and reliable real-time environment for the flight simulation. FLSIM provides the following functions that is used for the DTB:

- The main strength of FLSIM's performance is its flexibility. The flight model, meaning the parameters used to describe the performance of the simulated aircraft, can be quickly loaded from the text reconfiguration file at the beginning of use of the flight simulation software. In this way, it can build aircraft models for most fixed wing airplanes such as a Boeing 747. This allows for an effective flexibility in the DTB.

- FLSIM includes an Automatic Flight Control System (AFCS) - an autopilot system allowing the user to set up a predefined simulated flight. The AFCS includes some basic autopilot modes including the heading mode, altitude hold mode, and vertical speed mode. FLSIM also provides the users with a powerful programming interface to design autopilot modes that the users want to add the AFCS of FLSIM. The user also can modify the autopilot modes of the FLSIM. This function is used to develop the AFCS to enhance the DTB. The FLSIM also provides an uncomplicated environment mode for the flight simulation. The user can design a more complex environment mode to meet their requirements.
- FLSIM provides a runtime controller function (RTC) to control all simulation modes to be executed in a predefined sequence. All FLSIM simulation models are executed at a specified rate. The default rate is 30 Hz. In this way the DTB can realize real time flight simulation. FLSIM supports a user-dispatching execution mode. This is the mode used in this thesis, and shown in Figure 5. As it is seen in this figure. the FLSIM simulation models can be linked to a user application that deals with DTB specifications. Some models are critical to the simulation fidelity and need to be executed more frequently. while some are not that critical and can be executed less frequently to save the system resources. The execution band for each module is predefined in user program. and the user program is responsible for dispatching the FLSIM models. The simulation models are integrated at a rate specified by the user. but not more than 30Hz.
- FLSIM provides a global data structure as the base of the globe database for the

internal and external functions. This database includes all static and runtime data for the flight simulation. All of the simulation models exchange data through this database. The user can input steering commands and obtain the outputs of the flight simulation in such a data structure.

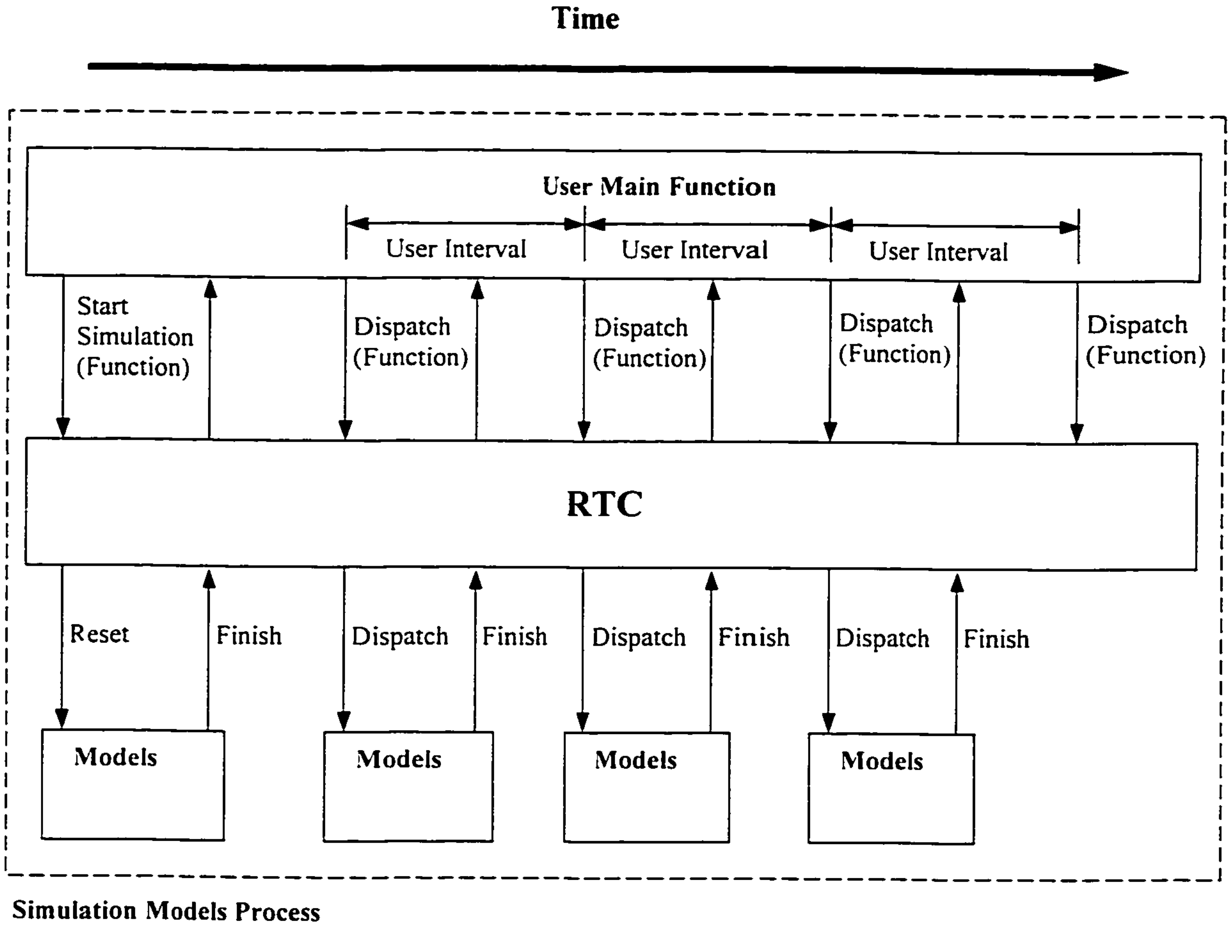


Figure 5 User-Dispatch Execution Mode

2.1.2 Development of the Aircraft Model

The aircraft model contains the configurations of the aircraft type that is to be simulated [11][16]. The configuration includes the weight, geometric and aerodynamic coefficients of the aircraft, engine and control surfaces, and so on. This aircraft type is simulated by the DTB, and this aircraft type should be stored in the FMS that is being tested by the DTB.

Since FLSIM provides an easy way to configure the aircraft model, this feature of FLSIM is used in DTB. A text file is made that contains the airplane profile information and the aircraft type name. This file is loaded at the beginning of use of the Flight Simulation Software to build the aircraft model. In DTB, a Boeing 747 aircraft model is configured.

2.1.3 Automatic Flight Control System Modes in DTB

In this section, the background knowledge for developing FMS VNAV mode in DTB is introduced. Firstly, the basic knowledge about Automatic Flight Control System (AFCS) is introduced. Then AFCS in FLSIM is discussed. The requirement of the AFCS in DTB is stated finally. How to develop the FMS VNAV mode to meet the requirement is discussed in Chapter 3.

2.1.3.1 Automatic Flight Control System

AFCS has two basic autopilot functions to perform [24][25]:

- To provide short-period attitude stability

- To provide longer period flight path control

An aircraft does not have the adequate status or dynamic stability, and the pilot has a difficult job controlling the path of the airplane in space. The pilot maintains a given attitude and power setting to execute a path, and changes the airplane's attitude stability and the power setting as required. If the pilot ignored the basic attitude stability of the airplane and did not maintain control over it, the flight path would be quite erratic and uncomfortable. The autopilot controls the airplane exactly the way in which the pilot does. It maintains a tight control on the roll and pitch attitudes of the airplane and improves the inherent stability of the airplane in all three axes. The path modes then can command attitude changes as required. Any limitations or smoothing, such as attitude command limits, rate limits, or filters, are imposed on the signals commanding the changes in attitude.

To perform two basic autopilot functions, each axis of the autopilot contains two control loops: an inner stability loop and an outer path loop. The outer loop provides path control and consists of the electronic circuitry necessary to couple the airplane's navigation systems to the input of the inner loop. The inner loop provides attitude stability and maintains the required reference attitude to maneuver the airplane in a desired path.

2.1.3.2 The AFCS in FLSIM

In FLSIM, the autopilot takes full control of the flight capabilities of the simulation. The autopilot uses the Flight Control Computer (FCC) simulation data that would normally be provided by the on-board computer of an aircraft, and the FMS to control the parameters that allow the simulation to fly through waypoints [22]. Figure 6 illustrates autopilot logic in FLSIM. In this figure, the executive sequences of the outer loop and inner loop involved in the Autopilot model in FLSIM are shown. The concept of the AFCS outer loop and inner loop are introduced in Section 2.1.3.1. In each FLSIM simulation loop, if autopilot modes are engaged, the outer loops are executed first, then the basic control commands for the control surfaces of the aircraft are calculated by outer loops, and these commands are provided as the inputs of the inner loops. After the inner loops are executed, a control loop finishes, and next control loop begins. FLSIM provides some simple control laws for the autopilot outer loops. They are: Altitude Hold and Altitude Select Control Laws, FMS Lateral Control Law, Basic and Heading Select Control Law, Roll and Yaw Align Control Laws, and so on. In this thesis, a FMS Glide Slope Control Law, embedded in the FMS VANV outer loop, is developed to meet the requirement of FMS VNAV mode in DTB. This is discussed in detail in Chapter 3

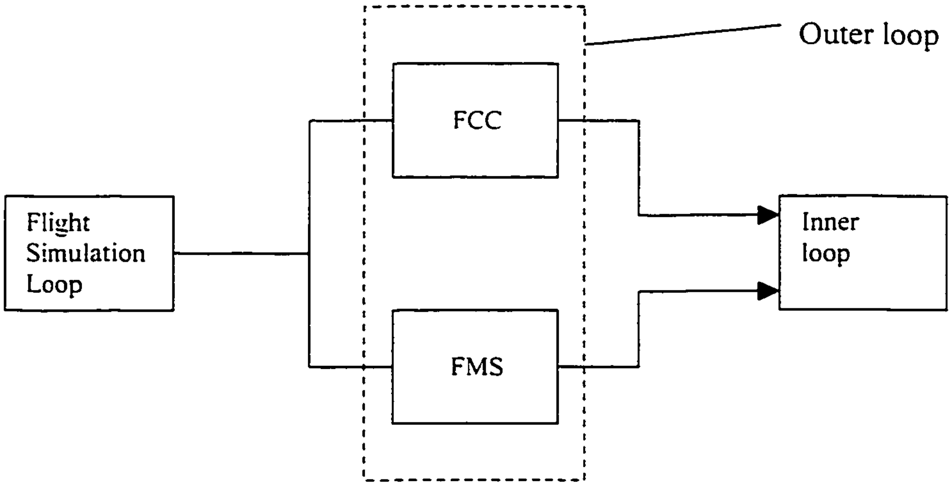


Figure 6 Autopilot Logic in FLSIM

2.1.3.3 The Requirements of the AFCS in the DTB

To meet the requirements of the DTB, the AFCS of the DTB should have the following necessary autopilot modes [26][31]:

- Speed Hold mode to maintain a selectable speed.
- Heading Hold mode to catch and maintain a selectable heading.
- Altitude Hold mode to hold a target altitude.
- Vertical Speed mode to maintain a selectable vertical speed.
- Level Change mode to catch a new target altitude with a selectable airspeed under a target thrust.
- Lateral Navigation (LNAV) mode to capture and track the lateral route under the roll command from the FMS.
- Vertical Navigation (VNAV) mode to capture and track the glide slope under the vertical deviation from the FMS in the approaching phase.

FLSIM includes some basic control laws to realize the Heading Hold mode, the Altitude Hold mode and the Vertical Speed mode. The LNAV mode, the Level Change mode and the Speed Hold mode were developed by Chui [31]. The VNAV mode is developed in this thesis, and it is explained in Chapter 3.

2.1.4 The Navigation Models in the DTB

The data from the navigation sensors is one of the main inputs to the FMS [31]. The development of the Navigation models is an important step in the development of the

DTB. In this section the navigation models that are used in the DTB are introduced. A useful method to develop the IRS Navigation Models in FLSIM is offered in Chapter 3. To meet the requirements of the DTB, the navigation models are designed as follows:

- Global Position System (GPS) model

The GPS model takes position and velocity information including latitude, longitude, altitude and the status for the GPS [24][25][31], and maps it into ARINC 429 words transmitted to the FMS through the interface computer.

- Air Data Computer (ADC) model

The ADC model takes the speed and the altitude information including True Airspeed (TAS), Pressure Altitude and Baro Corrected Altitude from the simulation [24][26][31], and maps it into ARINC 429 words transmitted to the FMS through the interface computer.

- Attitude Heading Reference System (AHRS) model

The AHRS model takes the heading information including True Heading and Magnetic Heading [25][31], and maps it into ARINC 429 words transmitted to the FMS through the interface computer.

- Inertial Reference System (IRS) model

The IRS model takes the IRS navigation information from the simulation [6][24] and maps it into ARINC 429 words in standard format transmitted to the FMS through the interface computer.

- Very High Frequency Omnidirectional Range (VOR) model

The VOR model takes the VOR navigation information including the VOR Frequency and Bearing from the simulation [6][24] and maps it into ARINC 429 words in standard format transmitted to the FMS through the interface computer.

- Distance Measuring Equipment (DME) model

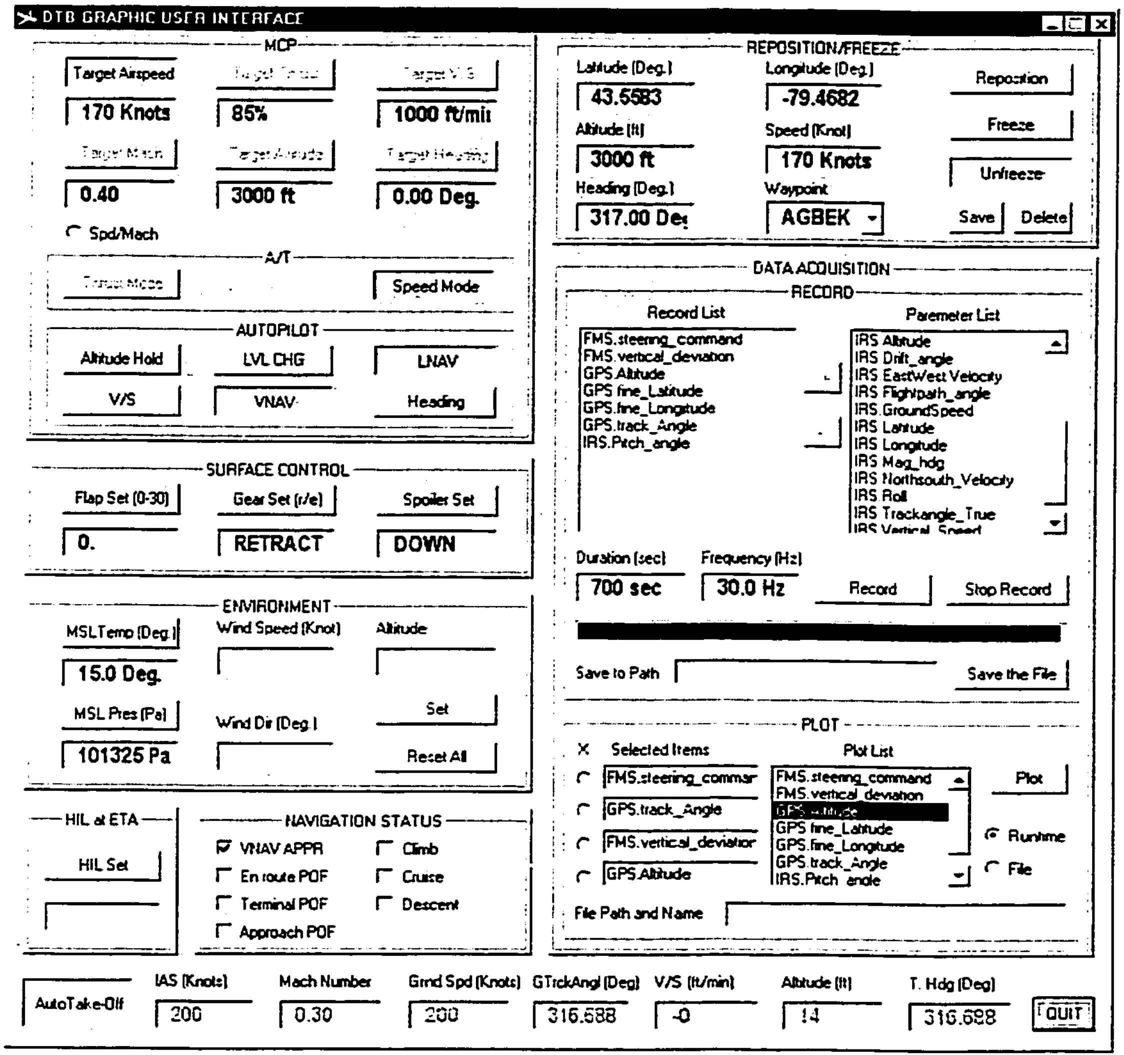
The DME model take the DME navigation information including the DME frequency and distance from the simulation [6][26] and maps it into ARINC 429 words in standard format transmitted to the FMS through the interface computer.

All of the navigation information of all navigation models in the DTB is listed in the Appendix B. GPS mode, ADC mode and AHRS mode of DTB have been developed by Chui [31]; IRS mode is developed and added into DTB in this thesis.

2.2 User Interface Software

The User Interface Software provides the necessary interface between the user and the DTB. It should have basic control and setting functions, as well as data acquisition functions. This software is developed by Chui [31]. In his thesis two dialog windows which are the main dialog window and the child dialog window are designed. The main dialog window is in charge of most of the User Interface Software jobs, and the child dialog window takes part in data acquisition functions for drawing the data curve that the user selects. In this thesis, the Mode Control Panel (MCP) in the main dialog window is redesigned. The User Interface Software is developed using Microsoft Visual C++ ActiveX controls. Only the main dialog window is introduced in this section. Redesigning the MCP of the DTB GUI is presented in Chapter 4.

The main window carries out most of the control and setting functions, and it is the primary interface. Figure 7 illustrates the Main Dialog Window. It includes the Mode Control Panel (MCP), the Surface Control model, the Environment Setting model, the Reposition and Freeze model, Data Acquisition model and the Basic Data Display Model. In this section, only the Mode Control Panel (MCP) is discussed in detail. Since the Reposition and Freeze mode is used to shorten the execution of a flight plan in this thesis, the Reposition and Freeze model is also introduced briefly.



Target Airspeed: 170 Knots
Target Thrust: 85%
Target V/S: 1000 ft/mi
Target Mach: 0.40
Target Altitude: 3000 ft
Target Heading: 0.00 Deg.

Flap Set (0-30): 0.
Gear Set (r/e): RETRACT
Spoiler Set: DOWN

MSLTemp (Deg.): 15.0 Deg.
Wind Speed (Knot):
Altitude:
MSL Pres (Pa): 101325 Pa
Wind Dir (Deg.):

☑ VNAV APPR
☐ En route POF
☐ Terminal POF
☐ Approach POF
☐ Climb
☐ Cruise
☐ Descent

Latitude (Deg.): 43.5583
Longitude (Deg.): -79.4682
Altitude (ft): 3000 ft
Speed (Knot): 170 Knots
Heading (Deg.): 317.00 De
Waypoint: AGBEK

Record List	Paremeter List
FMS.steering_command	IRS Altitude
FMS.vertical_deviation	IRS Drift_angle
GPS.Altitude	IRS EastWest Velocity
GPS fine_Latitude	IRS Flightpath_angle
GPS.fine_Longitude	IRS GroundSpeed
GPS.track_Angle	IRS Latitude
IRS.Pitch_angle	IRS Longitude
	IRS Mag_hdg
	IRS Northsouth_Velocity
	IRS Roll
	IRS Trackangle_True

Duration (sec): 700 sec
Frequency (Hz): 30.0 Hz
Save to Path:

X	Selected Items	Plot List
○	FMS.steering_commar	FMS.steering_command
○	GPS.track_Angle	FMS.vertical_deviation
○	FMS.vertical_deviatior	GPS.Altitude
○	GPS.Altitude	GPS fine_Latitude
		GPS.fine_Longitude
		GPS.track_Angle
		IRS.Pitch_angle

● Runtime
○ File

File Path and Name:

IAS (Knots)	Mach Number	Grnd Spd (Knots)	GTrckAngl (Deg)	V/S (ft/min)	Altitude (ft)	T. Hdg (Deg)
200	0.30	200	316.588	-0	14	316.628

Figure 7 The Main Dialog Window

2.2.1 The Mode Control Panel Model

In this section, the function and the requirement of the Mode Control Panel (MCP) is described and explained. All autopilot modes in MCP of DTB GUI are described in detail. How to design MCP in DTB GUI to meet these requirements is discussed in Chapter 4.

2.2.1.1 The Altitude Hold Mode

In the Altitude Hold mode, the aircraft maintains an altitude with the target airspeed, and the engines are in autothrottle.

To fly the altitude hold mode, the user must enter the target airspeed and engage the Altitude hold mode through the MCP. The Altitude Hold mode can be in any lateral mode, and is canceled by selecting any other vertical mode.

2.2.1.2 The Level Change Mode

In the Level Change mode, the aircraft climbs or descends toward the selected target altitude with the selected target airspeed. In this mode the amount of throttle position for climb is 90% and for descend is 30%.

To fly the level change mode, the user must enter the selected target altitude and the selected target airspeed, then engages the level change mode through the MCP.

The level change mode is canceled by any one of the following actions:

- Capturing the target altitude

- Engaging any other vertical mode
- Repositioning the aircraft

2.2.1.3 The Vertical Speed Mode

In the vertical speed (V/S) mode, the aircraft climbs or descends toward the selected target altitude with the selected target vertical speed. The target airspeed is maintained by the autothrottle.

To fly the V/S mode, the user must enter the selected target altitude, the selected target vertical airspeed and the selected target airspeed, and then engage the V/S mode through the MCP.

The V/S mode is canceled by any one of the following actions:

- Capturing the target altitude
- Engaging any other vertical mode
- Repositioning the aircraft

2.2.1.4 The VNAV Mode

In the VNAV mode, the aircraft flies by a vertical steering command from the FMS with the selected target airspeed. The VNAV mode is developed to fly the aircraft in the approach phase in the DTB.

To fly the VNAV mode, the user enters the target airspeed and engages VNAV mode at an approach phase through the MCP.

The VNAV mode is canceled by any one of the following actions:

- Engaging any other vertical mode
- Repositioning the aircraft

Development of the VNAV mode for DTB is discussed in detail in Chapter 3.

2.2.1.5 The LNAV Mode

In the LNAV mode, the aircraft flies by the lateral steering command and roll command from the FMS.

To fly the LNAV mode, the user engages the LNAV mode through the MCP.

The LNAV mode is canceled by any one of the following actions:

- Engaging the heading modes
- Repositioning the aircraft

2.2.1.6 The Heading Mode

In the heading mode, the aircraft captures and maintains a selected magnetic heading.

To fly the heading mode, the user must enter a selected target magnetic heading and engage the heading mode through the MCP.

The Heading mode is canceled by any one of the following actions:

- Engaging the LNAV modes
- Repositioning the aircraft

2.2.2 Reposition, Freeze and Unfreeze Modes

Through the reposition, freeze and the unfreeze Modes (REPOSITION/FREEZE section in Figure 7), the DTB user can reposition the aircraft in the simulation. The position information of the aircraft includes the latitude, longitude, altitude, airspeed, heading and waypoint. Six edit boxes were designed for entering each position information parameter. The Reposition button jumps the aircraft from the old position to a user-selected position. The Freeze button can freeze the simulation at its present position. The Unfreeze button can unfreeze a freeze action to continue the simulation from the freeze position.

CHAPTER 3

DEVELOPMENT OF FMS VNAV MODE AND IRS MODE

In this chapter, development of the VNAV mode and IRS mode is discussed.

Based on the FMS functionality, the FMS VNAV mode can be used during all phases of flight for a complete vertical flight profile. In the case of the DTB, the VNAV command provided by the FMS is only limited in the approach phase of flight, i.e., the GPS approach. Section 3.1 and Section 3.2 describe the development of the VNAV mode in the GSP approach area.

Section 3.3 discusses development of the IRS mode for DTB. First, the concept of FLSIM global data memory segment is introduced. Then how to use it to develop the IRS sensor for DTB is discussed.

3.1 ILS Approach and GPS Approach

Most modern aircrafts use the Instrument Landing System (ILS) for approaching. On an aircraft equipped with autopilot and an ILS receiver, the ILS approach mode automatically intercepts, captures, and tracks the front course localizer and glide slope signals, as shown in Figure 8. The localizer in lateral plane is captured in Figure 8-a. The glide slope capture in vertical plane occurs after the localizer has been captured [24][26][27], as shown in Figure 8-b. While the aircraft captures and tracks the glide slope, the glide slope beam is sensed by the ILS receiver, and the autopilot continuously adjusts the aircraft pitch attitude to minimize the glide slope deviation, which is an angular deviation (Γ), and presented to the pilot in dot.

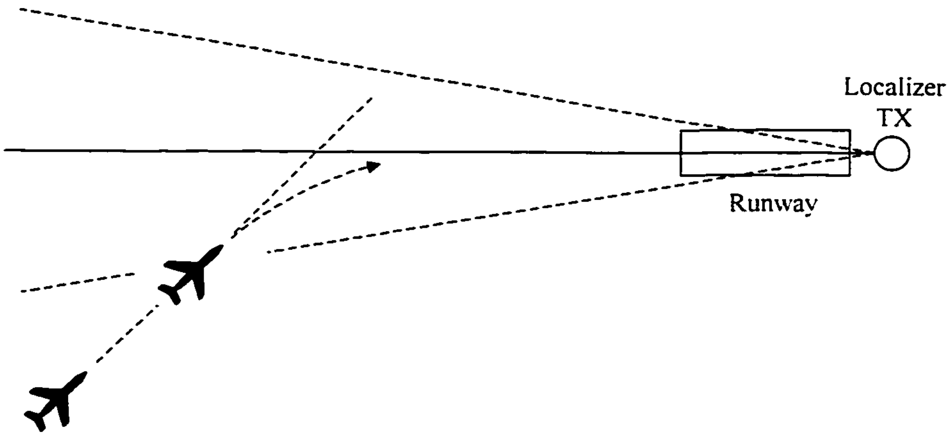
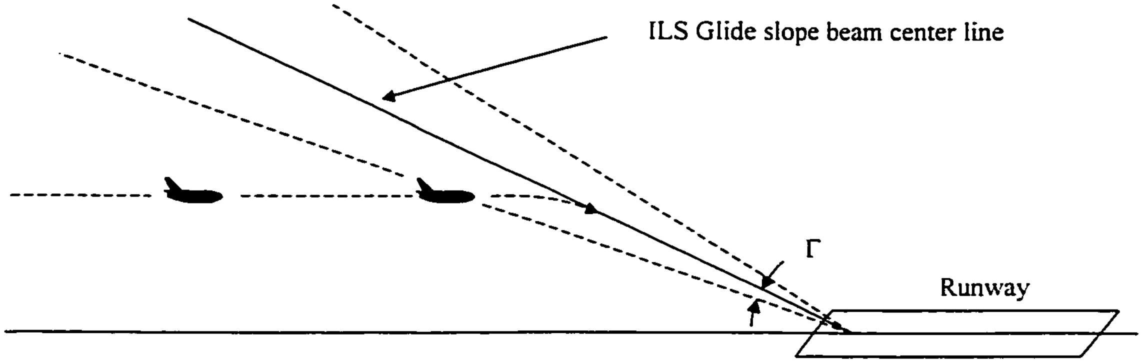


Figure 8-a ILS Approach Mode Localizer Intercept



Γ: The angular deviation of the ILS glide slope beam.

Figure 8-b ILS Approach Mode Glide slope Capture

Figure 8 ILS Approach Mode

The FMS can be coupled with the autopilot for none-precision approach. In this situation, the FMS calculates the glide slope vertical deviation (VDEV) based on its built-in navigation database and the GPS three-dimensional aircraft position. The VDEV is defined as the vertical distance from the aircraft's position to the glide slope center beam, as shown in Figure 9. This vertical deviation is then used as input by the glide slope control law to generate a desired vertical steering command and the pitch angle. By implementing the vertical steering command, the autopilot can arm, intercept, capture and track the glide slope beam.

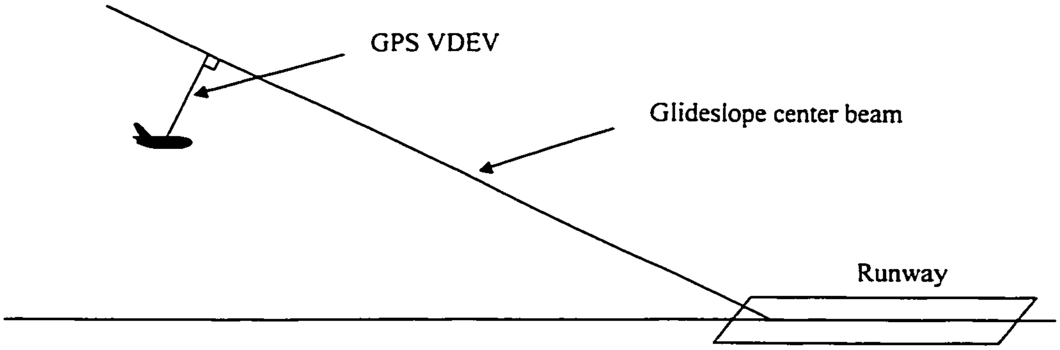


Figure 9 GSP Glide slope Vertical Deviation

Figure 10 illustrates the GPS approach. It is different from Figure 8-b that the glide slope beam is not the beam sensed by the ILS receiver, but a virtual line defined in the navigation database.

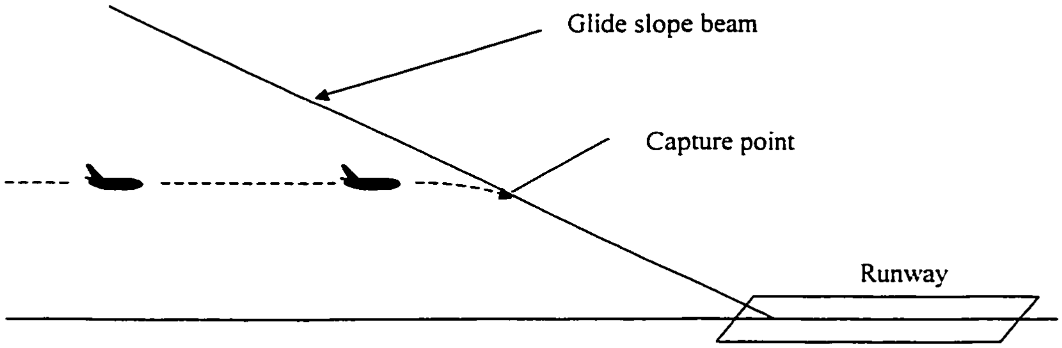
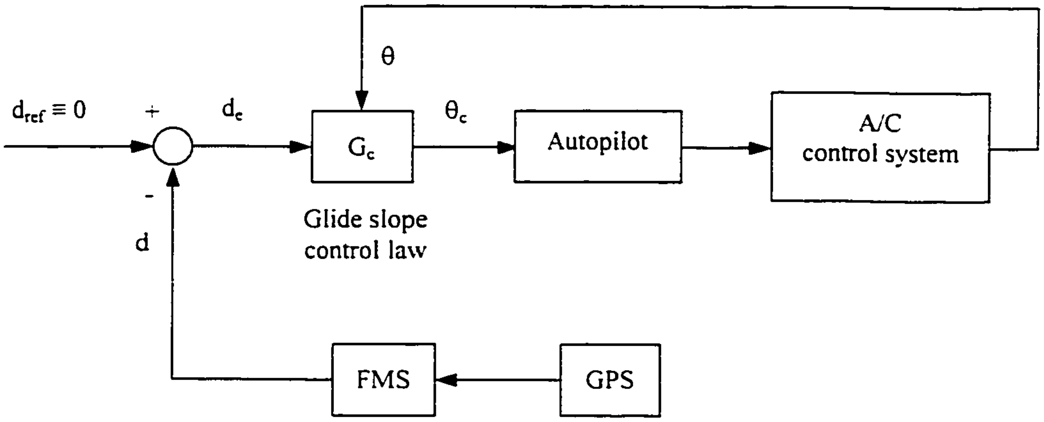


Figure 10 GPS Approach Mode Glide slope Capture

It is the glide slope control law that links the FMS with the autopilot and aircraft flight control system. Figure 11 illustrates the overall schematic of the system. At each control loop, FMS receives the vertical deviation signal from GPS sensors, then FMS calculates the GPS vertical deviation (d), the vertical deviation error (d_e) is obtained easily (the reference vertical deviation (d_{ref}) is always zero) and is input to the Glide slope control law. At the same time Glide slope control law gets actual aircraft pitch angle (θ) as another input from A/C control system. After the calculation of the Glides lope control law, the output of the glide slope control law, the target pitch angle command (θ_c) is sent to autopilot. then this command is executed by the A/C control system. Then next control loop begins. Laterally. the aircraft is guided by FMS lateral steering command to arm. intercept, capture and track the runway centerline.



d_{ref}: reference vertical deviation.
d: GPS vertical deviation calculated by FMS.
d_e: vertical deviation error.
θ: actual aircraft pitch angle.
θ_c: target pitch command.

Figure 11 System Schematic of GPS Approach Mode

3.2 Development of the Glide Slope Control Law

As described in Section 3.1, the glide slope control law is used to generate the pitch command for the automatic flight control system. The calculation is based on the GPS VDEV as well as the airspeed and vertical speed. These also are used by the control law to determine the correct glideslope capture point. By implementing this pitch command, the aircraft should be able to reduce and eliminate the glide slope vertical deviation, and therefore track the glideslope for a safe approach.

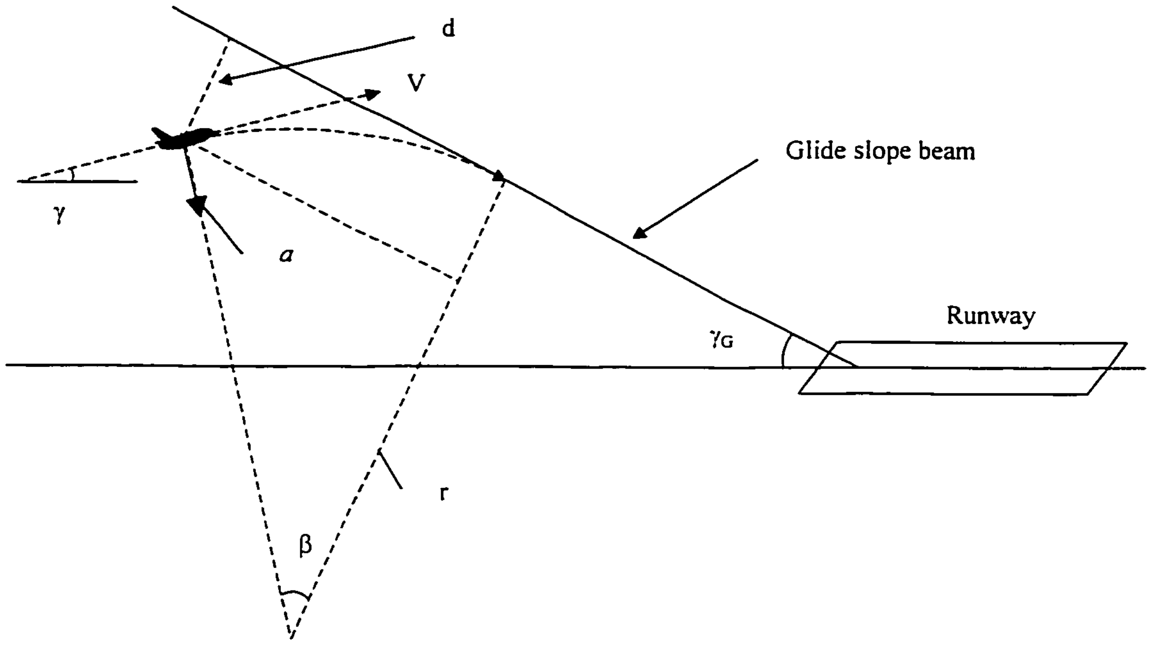
3.2.1 Criteria of Glide slope Intercept and Capture

Like the flying ILS approach, the aircraft normally descends to an appropriate altitude and holds that altitude in order to intercept the glide slope. The control law should continuously monitor the GPS VDEV, airspeed and vertical speed. At a certain point, the aircraft should be able to follow an arc of a circle by adjusting its pitch angle and then smoothly capture the glide slope. This point is called the capture point, as depicted in Figure 10. In Figure 12, the acceleration centripetal (a) should be within a limit a_{lim} that can then be defined by Equation (1)

$$a_{\text{lim}} = \frac{V^2}{r} \qquad (1)$$

where r is radius of path arc.

Equation (1) serves as the starting point of the development of glide slope control law.



d: GPS VDEV
V: aircraft airspeed vector.
γ: aircraft flight path angle
γ_G: nominal glide slope angle.
a: acceleration centripetal
r: radius of the arc

Figure 12 FMS Approach Mode Glide slope Capture

3.2.2 The Glide slope Control Law

In Figure 12, the geometric equation is given by Equation (2):

$$\beta = \gamma_G + \gamma \qquad (2)$$

where γ can be calculated by Equation (3):

$$\gamma = \sin^{-1}\left(\frac{V_\perp}{V}\right) \qquad (3)$$

where V represents the airspeed and is always positive. $V_{\perp}$ represents the vertical speed and is negative when the aircraft is descending.

Assuming the airspeed (V) is held by autothrottle under this mode. The acceleration centripetal (a) in Figure 12 is the only acceleration vector applied on the aircraft, and it can be obtain by Equation (4):

$$a = \frac{V^2}{r} \qquad (4)$$

The form of Equation (4) is changed into Equation (5):

$$r = \frac{V^2}{a} \qquad (5)$$

From the geometry in Figure 12, Equation (6) and Equation (7) are obtained:

$$d = r(1 - \cos\beta) = \frac{V^2}{a}(1 - \cos\beta) \qquad (6)$$

$$\dot{d} = V \times \sin\beta \qquad (7)$$

where $\dot{d}$ represents the rate of the vertical deviation.

When the aircraft's position and speed are satisfied such that aircraft can follow the arc trajectory shown in Figure12, the relationship between d and $\dot{d}$ can be represented by Equation (8):

$$d + \dot{d} \times f(\bullet) = 0 \qquad (8)$$

A deviation from the circular trajectory causes the unbalance of Equation (8). Therefore, a pitch angle, which is the output of the control law, is generated to bring the aircraft back to the ideal trajectory. This control law is expressed by Equation (9):

$$\theta_c = K_1 \times (\ d + \dot{d} \times f(\bullet)\ ) + \theta \qquad (9)$$

where θ is the current pitch angle and θ_c is the target pitch angle to capture and track the glide slope.

Substituting Equations (6) and (7) into (8):

$$\frac{V^2}{a} \times (1 - \cos\beta) + V \times \sin\beta \times f(\bullet) = 0 \qquad (10)$$

Solving $f(\bullet)$ from equation (10), Equation (11) is obtained:

$$f(\bullet) = -\frac{V}{a} \times \tan(\frac{\beta}{2}) \qquad (11)$$

From Figure12, when the aircraft is on track of the glide slope, β becomes zero and therefore causes $f(\bullet)$ to be zero. As a result, the damping term in Equation (9) is eliminated and causes the control to become unstable. Therefore, proportional gain and a damping term are added to Equation (11), as shown in Equation (12):

$$f(\bullet) = -K_2 \times \frac{V}{a} \times \tan(\frac{\beta}{2}) + K_3 \qquad (12)$$

The final expression of the control law can be derived by substituting Equation (12) into (9), as shown in Equation (13):

$$\theta_c = K_1 \times (\ d + \dot{d} \times (\ -K_2 \times \frac{V}{a} \times \tan(\frac{\beta}{2}) + K_3\ )) + \theta \qquad (13)$$

For the sake of simplicity, by changing the sign of K_2, Equation (13) can be changed to Equation (14):

$$\theta_c = K_1 \times (\ d + \dot{d} \times (\ K_2 \times \frac{V}{a} \times \tan(\frac{\beta}{2}) + K_3\ )) + \theta \qquad (14)$$

Figure 13 illustrates the block diagram of the glide slope control law. The aircraft command pitch attitude θ_c is restricted by a max/min pitch angle limit. Programming gains, K_1, K_2 and K_3 is tuned as described in Section 3.2.3.

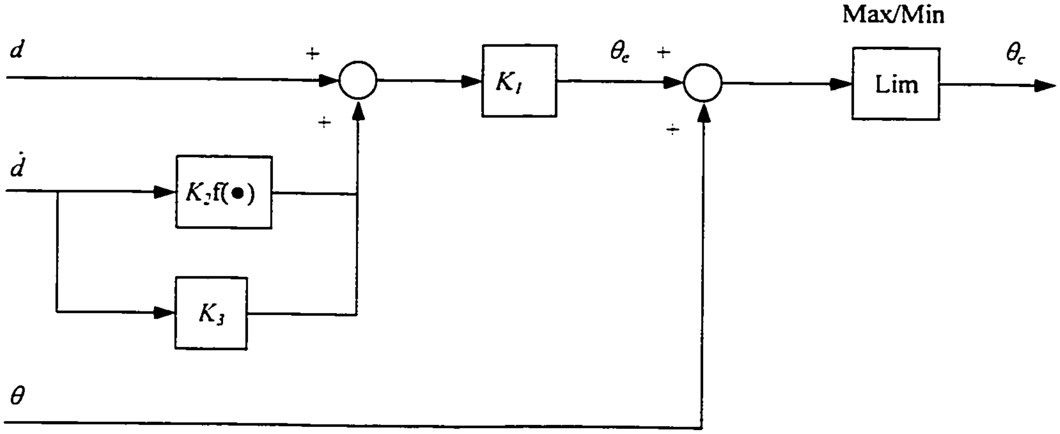


Figure 13 Block Diagram of Glide Slope Control Law

3.2.3 Test of the GPS VNAV Control Law

According to the GPS approach glide slope control law developed above, the codes of the GPS VNAV control law are programmed and embedded in DTB. The perfect approach curve line can be obtained after adjusting the value of the control parameters K_1, K_2 and K_3. Figure 14 shows how the VNAV control law works under the different values of the K_1, K_2 and K_3. The figure on the left (with K_1= -0.002, K_2=0.005, K_3=0) shows, the vertical deviation, altitude, and the pitch angle time response of the aircraft after engaging the VNAV mode in the approach area. As it can be seen in this figure, vertical deviation shows a large overshoot and pitch angle shows an oscillation. The figure on the right (with K_1=-0.0002, K_2=0.0035, K_3=0 that are optimal) in Figure 14 illustrates that the time responses do not have the overshoot because of the proper damp ratio, and the aircraft can accomplish a good approach.

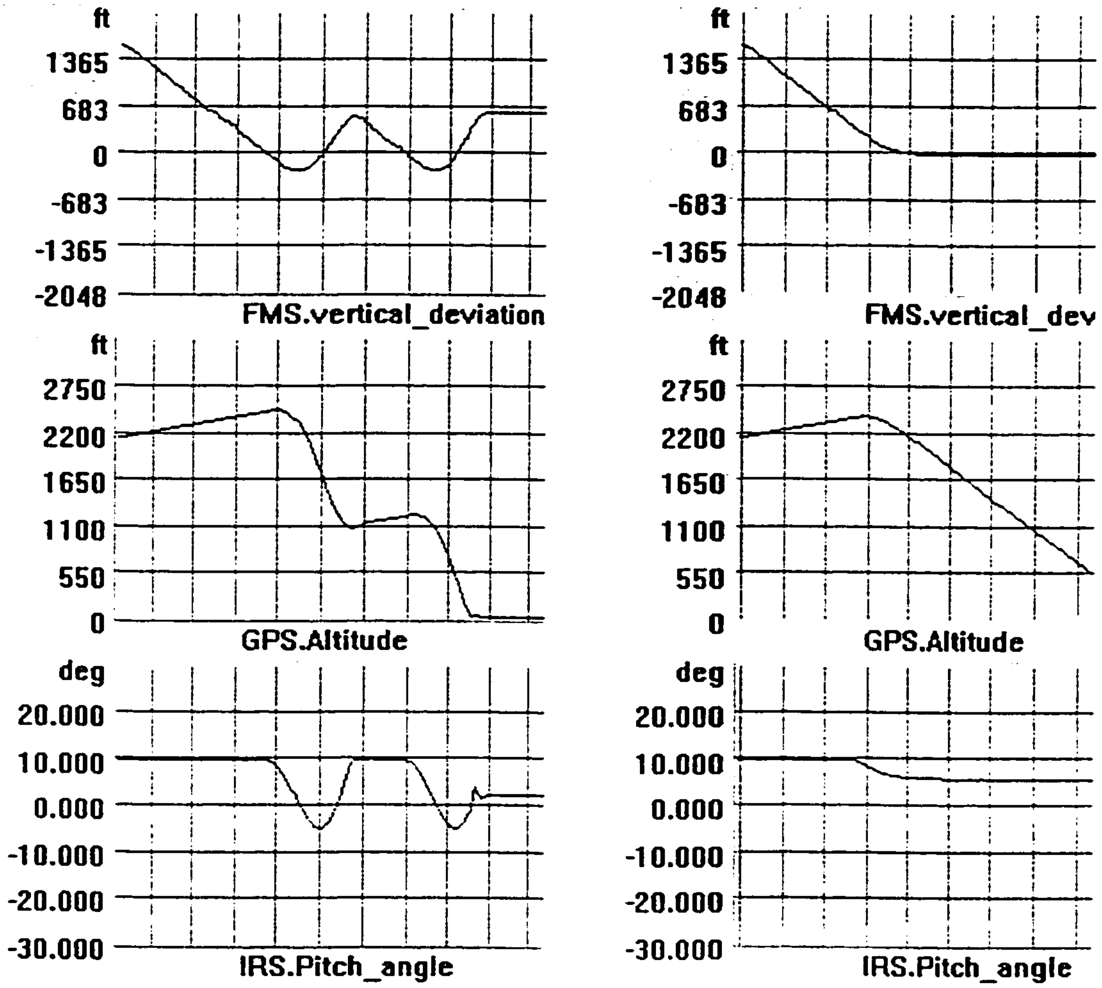


Figure 14 Adjust K_1, K_2 and K_3 to obtain the stable GPS approach (in the left figure, with K_1= -0.002, K_2=0.005, K_3=0; in the right figure, K_1=-0.0002, K_2=0.0035, K_3=0)

Figure 15 shows the vertical deviation, GPS altitude and the pitch angle time_response after repositioning the aircraft to the approach area. In Figure15, at point A the flight simulation is repositioned at an altitude 3000ft in an approach area. The VNAV mode is engaged at point B, then the aircraft begin to capture the GPS glide slope. The point C is capture point. The aircraft captures the GPS glide slope, then flies following the glide slope under the VNAV mode. When the aircraft reaches an altitude below 400 feet, the autolanding system should start, and the aircraft begins to execute the landing task. Since DTB dose not have the autolanding system yet, at point D, a missed approach is reported by FMS, then the aircraft goes around to the next waypoint.

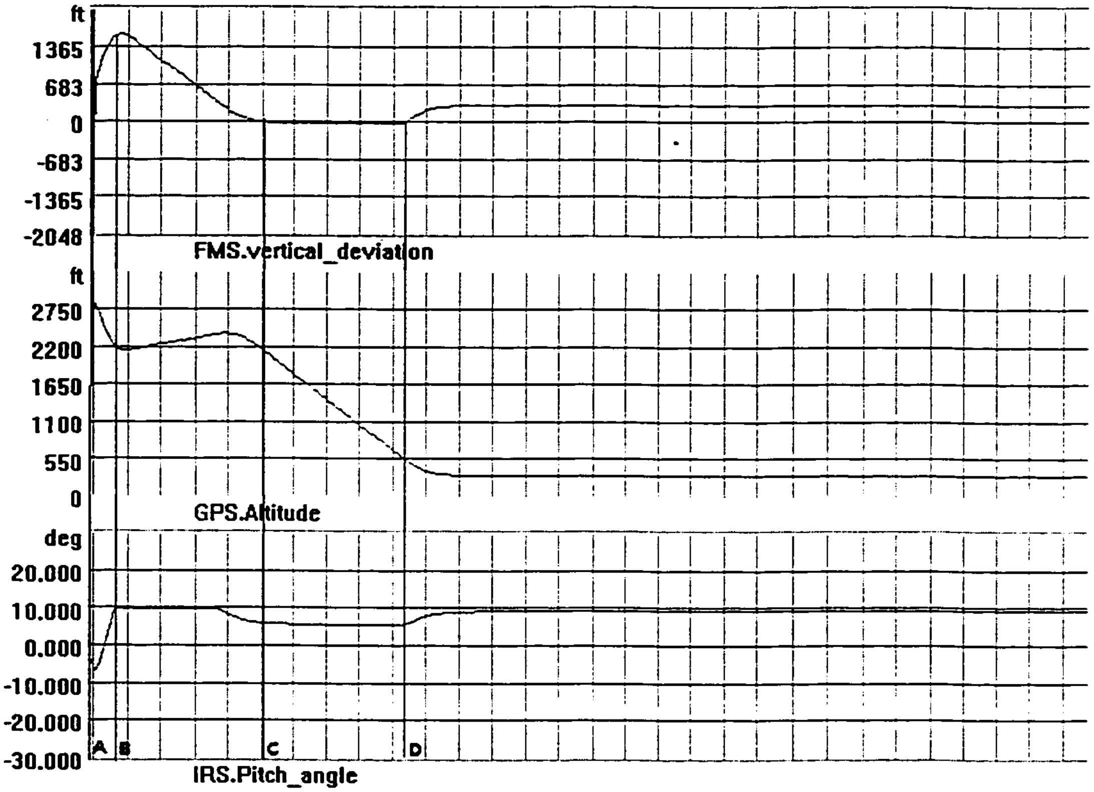


Figure 15 The vertical deviation, altitude and the pitch angle time response in the approach and the miss approach phase in the DTB.

3.3 Development of the IRS Mode

The MFS VANV mode is integrated in DTB and tested in Section 3.1 and 3.2 of this chapter. The development of IRS mode is explained in this section, this is another objective of this thesis.

3.3.1 FLSIM Global Data Memory Segment

Once FLSIM runs, a physical FLSIM global data memory segment is allocated, in which all the flight simulation data is stored. This memory segment is named as **Flsim_Global_Data** by FLSIM. **Flsim_Global_Data** is divided into two functional areas: the static data area and the runtime data area. All flight simulation data, including the navigation data, are stored in **Flsim_Global_Data**. FLSIM also provides programmers the handle pointer of **Flsim_Global_Data** named as **gdp**. Through this pointer the programmers can handle the FLSIM global data, the programmers can read the flight simulation data from output data structure in **Flsim_Global_Data**; and the programmers also can write the data into input data structure in **Flsim_Global_Data** to change and control the flight simulation. FLSIM provides descriptions of the data structure for the **Flsim_Global_Data** [23] to instruct the programmers how to call the data of **Flsim_Global_Data**. This allows for easy user access to the navigation information data from the flight simulator. In this way the navigation models are developed. For example, the ADC model can take the True Airspeed (TAS) from the flight simulator from the memory address in **Flsim_Global_Data.gdp->adc.runtime_output.speed_true**, which is provided by FLSIM.

3.3.2 Developing the IRS Mode

Unfortunately, FLSIM 7.0 does not allocate such special data structure for IRS mode as it does for the ADC mode. In order to create the IRS mode, IRS sensor must obtain IRS navigational data from the other flight data in **Flsim_Global_Data** to simulate IRS flight data. For example, IRS mode gets the altitude, latitude and longitude from the data structure for position runtime outputs in **Flsim_Global_Data**, and gets the pitch and roll angles from the data structure for instrument runtime outputs in **Flsim_Global_Data.** In this way, the IRS mode is created for the DTB, and is embedded into the Flight Dynamic Simulation Software.

CHAPTER 4

REDESIGN OF DTB GUI AND DEVELOPMENT OF ARINC429 CHANNEL SWITCH SOFTWARE

In this chapter, redesigning the DTB GUI to improve the ergonomic functionality of the DTB and the development of the ARINC429 Channel Switch Software are discussed.

Since the MCP GUI that is a part of DTB GUI is redesigned in this thesis, the design of the MCP GUI is discussed in Section 4.1. First, the layout of the MCP is introduced. Then how to engage each autopilot mode by this MCP is discussed.

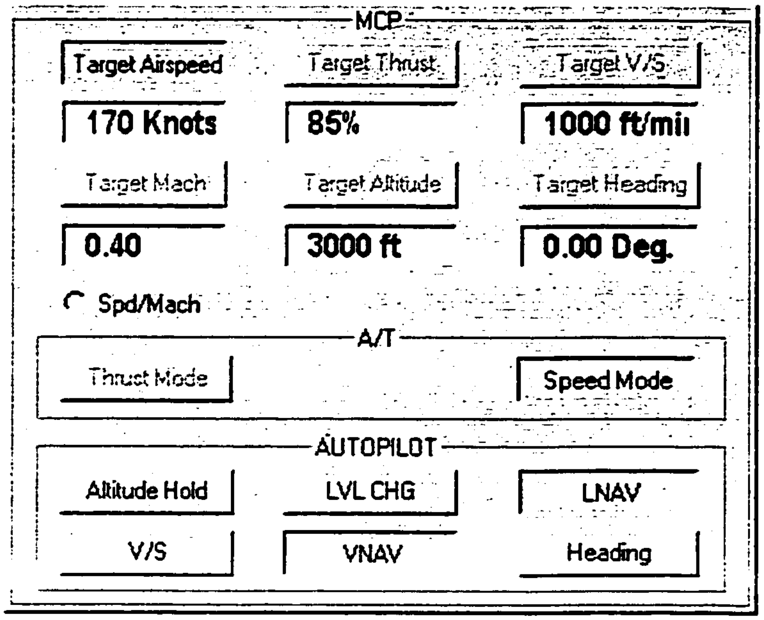
The development of the ARINC429 Channel Switch Software is also discussed in this chapter. In order to understand development of the ARINC429 Channel Switch Software, the ARINC 429 protocol in avionics and the basic concepts of the management of the ARINC 429 card are discussed in Sections 4.2 and 4.3. Finally development of the ARINC429 Channel Switch Software is explained in Section 4.4.

4.1 Redesign of the DTB GUI

As introduced in Chapter 2, MCP GUI is a main part of DTB GUI. In this section the redesigning the MCP GUI is discussed. First, the layout of the MCP is introduced. Then how to engage each autopilot mode by this MCP is discussed.

4.1.1 The Layout of The MCP

In order to realize the functions introduced above for each autopilot mode, and to provide a user friendly interface to the user, six target buttons that include Target Airspeed, Target Thrust, Target V/S, Target Mach, Target Altitude and Target Heading are designed. There are six edit boxes designed to be placed under each target button to enter and display the selected target value. These six target buttons should be invalid when they are down. They just have the display function. Once a target item is used by a selected control mode, its target button should be down automatically, and the fonts on the button should turn into black: otherwise the button should be up and the color of the fonts should be grey. The Thrust Mode button and the Speed Mode button are designed to display the autopilot mode that the user selects belongs to the speed mode or the thrust mode. They only have the display function, and are invalid when they are down. Six mode engage buttons for each autopilot mode used in the DTB are designed, they are: the Altitude Hold button, the LVL CHG button, the V/S button, the VNAV button, the LNAV button and the Heading button. The MCP is enlarged in Figure 16 from the main dialog window.



Target Airspeed 170 Knots

Target Thrust 85%

Target V/S 1000 ft/mii

Target Mach 0.40

Target Altitude 3000 ft

Target Heading 0.00 Deg.

Figure 16 The Layout of the MCP

4.1.2 Design Principles of Engaging Autopilot Mode

In order to meet the requirements submitted in Section 2.2.1. the following design principles are adopted during the design phase of the MCP.

- When the LNAV button is pushed down. the Heading button should be up. This indicates that the LNAV mode is engaged and the Heading Hold mode has been cancelled.
- When the Heading button is pushed down, the Target Heading button should be down. and the LNAV button should be up. This indicates that the selected target heading is entered and the Heading Hold mode is engaged, and that the LNAV mode has been canceled.
- When the Altitude Hold button is pushed down, the other vertical mode buttons

should be up, and the Target Airspeed button or Target Mach button should be down (depending on what kind of speed mode the user is using). The Target Thrust button, the Target Altitude button and the Target V/S button should be up. This indicates that the selected target airspeed or mach has been entered, the Altitude Hold mode is engaged, and that the other vertical autopilot modes have been canceled. The Speed Mode button is down to display that the Altitude Hold mode belongs to the speed mode.

- When the LVL CHG button is pushed down. the other vertical mode buttons should be up. The Target Altitude button and the Target Thrust button should be down to display that the selected target altitude and the thrust have been entered. The Target Airspeed or the Target Mach button should be down, depending on what kind of speed mode the user is using, to display the selected target airspeed or the mach that has been entered. This indicates that the Level Change mode is engaged and the other vertical autopilot modes have been canceled. The Thrust Mode button is down to display that the Level Change mode belongs to the thrust mode.
- When the V/S button is pushed down. the other vertical mode buttons should be up. The Target Altitude button and the Target V/S button should be down to indicate that the selected target altitude has been entered. The Target Airspeed or the Target Mach button should be down, depending on what kind of speed mode the user is using, to display the selected target airspeed or that the mach has been entered. This displays that the Vertical Speed mode is engaged and the other vertical autopilot modes have been canceled. The Speed Mode button is down to indicate that the V/S mode belongs

to the speed mode.

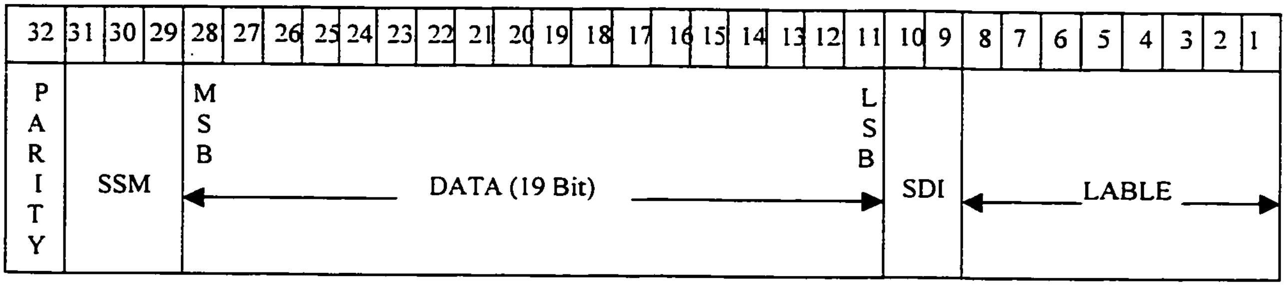
- When the VNAV button is pushed down, the other vertical mode buttons should be up. The Target Airspeed or the Target Mach button should be down, depending on what kind of speed modes the user is using, to display the selected target airspeed or that the mach has been entered. The Target Altitude button, the Target Thrust button and the Target Thrust button should be up. This indicates that the VNAV mode is engaged and that the other vertical autopilot modes have been canceled. The Speed Mode button is down to indicate that the V/S mode belongs to the speed mode.

4.2 ARINC 429

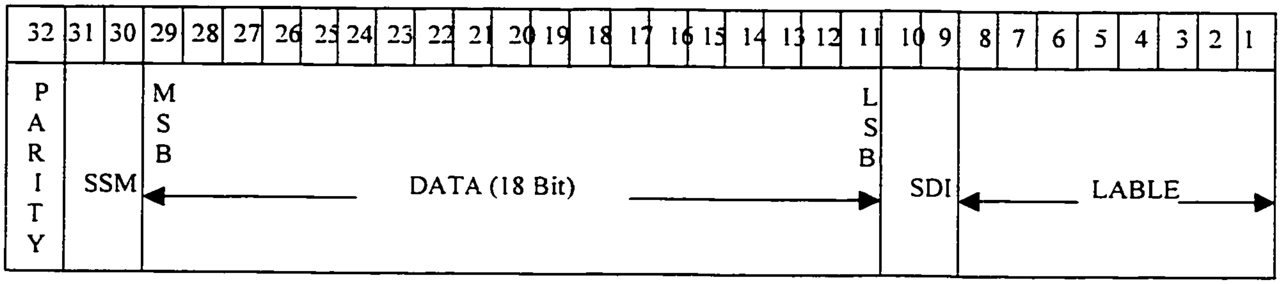
The ARINC 429 Mark 33 Digital Information System (DITS) defines the air transport industry's standard for the transfer of digital data between avionics system devices [24]. In this section, the ARINC 429 word formats are first introduced, and then the initialization and configuration of the ARINC channels for the ARINC card is explained.

4.2.1 ARINC 429 Data Words

The organization of the 32 Bit ARINC 429 data word is shown in Figure 17. A typical 32-bit word has five parts [24][28]:



ARINC 429 Format – BCD, Discrete, AIM and File Transfer Words



ARINC 429 Format – BNR Data Words

Figure 17 ARINC 429 Data Word Formats

- 8 bit label

A single ARINC bus may carry many words that each contains entirely different data. An 8-bit label code is used so that a receiving device may identify the data contained within a word. The labels contained in bit 1 through bit 8 range from 0-377 Octal. ARINC specification 429 contains the complete list of label codes known to be in use at the time of publication.

Some labels may be used to identify more than one kind of data. This may depend on the application and Equipment ID. For example, label 201 may contain:

(1) DME distance (Equipment ID 009)

(2) Fuel Tank #4 Temperature (Equipment ID 05A)

(3) DME (Equipment ID 115)

The Equipment ID code is not part of the ARINC word, although it may be contained in a data field. An ARINC word with the same label but differing Equipment ID codes is never used on the same bus.

- Data area

  Five data types may be transmitted in the ARINC word:

  1. Binary (BNR)

  2. Binary Coded Decimal (BCD): Generally used for the transfer of numeric data when there is human interaction. For example: Radio control panel frequency selections and DME distances to an indicator.

  3. Discrete (DIS): "True" or "false" condition bits within a word. For example: label 272 equipment ID 3B bit 11 Trim Wheel Enable.

     Bit 11 is 1: enabled

     Bit 11 is 0: not enabled

  4. Maintenance

  5. AIM: "Acknowledgment, ISO alphabet no. 5 and Maintenance information encoded in dedicated words"

- Odd parity bit

  Each word contains a parity bit to allow simple error checking by the receiving element. Odd parity is used.

- Source/ Destination Identifier (SDI).

  Bits 9 and 10 of BNR, BCD and DIS data words are generally reserved for the SDI. They are not used for this function when the resolution needed for BNR or data necessitates their use for that data. Bits 9 and 10 are not available for this function in alpha numeric (ISO alphabet no. 5) data words, as they are required for data.

The SDI is used in multi-device systems as follows:

1. When specific words of the same label from a source installation must be directed to specific destination installations.

2. When a number of source installations must identify themselves to a single destination.

3. The SDI is encoded as follows (Table 1)

Bit No. 10	Bit No. 9	Installation No.
0	0	1 to 3 ("All call")
0	1	1
1	0	2
1	1	3

Table 1 The SDI Encoding

Notes:

1. Code "00" is normally used for "All Call". It may be used to identify a fourth installation if required.

2. When the SDI is not used, bits 9 and 10 carry either binary zeros or valid data.

- Sign/Status Matrix (SSM)

1. The SSM is encoded in bits 30 and 31 for BCD, DIS, AIM and File Transfer words, and in bits 29 to 31 for BNR words.

2. SSM of BCD, DIS, AIM and File Transfer Words

(1) The SSM contains the following data:

- The sign of the BCD data

- The word type for AIM and File Transfer data
- The status of the transmitter for all data types

(2) The SSM is encoded as follows in Table 2.

Bit 10	Bit 9	BCD and DIS	AIM	File Transfer
0	0	Plus, north, east, right, to, above	Intermediate word	Intermediate word, plus, north, etc.
0	1	No computed data	Initial word	Initial word
1	0	Functional test	Final word	Final word
1	1	Minus, south, west, left, from, below	Control word	Intermediate word, minus, south, etc.

Table 2 The SSM Encoding for BCD, DIS, AIM and File Transfer Words

Notes:

- Bits 30 and 31 of the BCD data are set to zero when no sign is needed.
- The "No Computed Data" code is generated for BCD words when computed data is unavailable for reasons other than equipment failure. An example of this could be when a DME receiver is tuned to an action but out of range of the ground station.

1. SSM of BNR word

The SSM for the BNR words are encoded as in the following Table 3.

Bit 31	Bit 30	Bit 29	Designation BNR Data
0	0	0	Failure Warning/Plus, North, East, Right, To
0	0	1	Failure Warning/Minus, South, West, Left, From
0	1	0	No computed data
1	0	0	Functional Test/Plus, North, East, Right, To
1	0	1	Functional Test/Minus, South, West, Left, From
1	1	0	Normal Operation/Plus, North, East, Right, To
1	1	1	Normal Operation/Minus, South, West, Left, From
1	1	1	Not used

Table 3 The SSM Encoding for BNR Words

Notes:

- A source system should annunciate any detected failure that causes the unreliability of one or more of the words normally output by setting bits 30 and 31 of the affected words to the failure warning code (00). The word containing this code should to be supplied to the data bus during its failure condition.
- Bit 29 should be zero when no sign is needed.

4.2.2 Encoding and Decoding ARINC Words

Since all ARINC data words used in the DTB are standard BNR and BCD data, in this section, the encoding and decoding standards of BNR and BCD data words are introduced.

4.2.2.1 Encoding and Decoding BNR Data Words

4.2.2.1.1 ARINC Specifications of BNR Data

A reference example of Label 211, equipment ID 002, is given in Table 4 to demonstrate how to specify BNR data.

Label	Equipment ID	Parameter Name	Units	Range	SIG DIG	RESOL
211	002	Total Air Temperature	°C	512	11	0.25

Table 4 ARINC 429 Specification of Label 211, Equipment ID 002

- The Parameter Name column gives the label and the equipment ID's parameter name. In this case, the label is 211, the Equipment ID is 002, and this signifies Total Air

Temperature.

- The Units column gives the unit of the parameter. The unit of the Total Air Temperature is in degrees Celsius.
- The Range column gives the next highest binary number above the maximum parameter value. In Table 4, the range value is 512. The negative of the range value is used as the weight of the SSM sign bit.
- Significant Digits (SIG DIG) gives the numbers of bits that are used by the BNR data. A variable quantity depending on precision required. The MSB has a weight of half the range value; each successive bit has half the weight of the previous one. Unused bits are padded with zeros. In this case, the SIG DIG is 11, meaning that the data area ranges from bit 18 to bit 28, for a total of 11 bits (shown in Table 5).
- The Resolution (RESOL) column gives the exact resolution, the actual bit weight of the LSB. In this example, the resolution is 0.25 °C. The resolution of angular values are always approximate.

4.2.2.1.2 BNR Data Encoding and Decoding

Two kinds of data are generally represented by the BNR data in the DTB: Linear Data and Angular Data. First given is an example of Linear Data encoding and decoding, and then another example is given for Angular Data encoding and decoding.

4.2.2.1.2.1 Linear BNR Data Encoding and Decoding

As discussed in the example above, Label 211, Equipment ID 002 is Total Air

Temperature. It is linear BNR data. The following Table 5 shows how to encode and decode Linear BNR Data. The parameter value is represented by the sum of the weights of the set bits.

	SIGN	MSB						Data				LSB	spares
Bit No.	29	28	27	26	25	24	23	22	21	20	19	18	17......11
Bit WT(°C)	-512	256	128	64	32	16	8	4	2	1	0.5	0.25	Unused data bits padded with zeros
+25.00°C	0	0	0	0	0	1	1	0	0	1	0	0	
-25.00°C	1	1	1	1	1	0	0	1	1	1	0	0	
+25.75°C	0	0	0	0	0	1	1	0	0	1	1	1	
-25.75°C	1	1	1	1	1	0	0	1	1	0	0	1	

Table 5 ARINC 429 Linear BNR Data Encoding and Decoding Example (Label 211,EquipmentID 002: Parity, Status, SDI and Labels Omitted)

4.2.2.1.2.2 Angular BNR Data Encoding and Decoding

BNR angular data is represented as:

- Angles from 0 degree up to, but not including 180 degrees, are represented by positive binary numbers.
- Angles from 180 degrees up to, but not including 360 degrees, are represented by negative binary numbers.

An example is given below to explain how to encode and decode BNR angular data. Table 6 gives a fictional label XXX about a Heading parameter.

Label	Parameter Name	Units	Range	SIG DIG	RESOL
XXX	Heading	Degree	± 180	7	1.40625

Table 6 ARINC 429 Specification of the Label XXX

Table 6 and 7 show:

- The weight of the sign and the 7 data bits. The weight of the LSB (bit 22) is 1.40625 degrees, and defines the actual resolution of the data.

- 5 actual aircraft headings, their binary representation, and the decimal equivalent of the bit patterns. For example, the actual aircraft heading of 358.59375 degrees has a binary representation of all ones, which gives a decimal equivalent of −1.40625 degrees.

Actual Aircraft heading	SIGN	MSB			Data			LSB	Decimal equivalent of the bit pattern
	29	28	27	26	25	24	23	22	
	-180	90	45	22.5	11.25	5.625	2.8125	1.40625	
0.0°	0	0	0	0	0	0	0	0	0.0°
178.59375°	0	1	1	1	1	1	1	1	178.59375°
180.0°	1	0	0	0	0	0	0	0	-180°
181.40625°	1	0	0	0	0	0	0	1	-178.59375°
358.59375°	1	1	1	1	1	1	1	1	-1.40625°

Table 7 ARINC 429 Angular BNR Data Encoding and Decoding Example

4.2.2.2 Encoding and Decoding the BCD Data Words

4.2.2.2.1 ARINC Specification BCD Data

The reference example of Label 002, equipment ID 002, is shown in the following Table 8 to demonstrate how to specify BCD data.

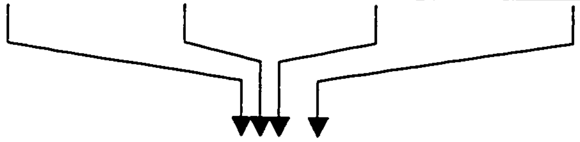
Label	Equipment ID	Parameter Name	Units	Range	SIG DIG	RESOL
002	002	Time To Go	Min	0-399.9	4	0.1

Table 8 ARINC 429 Specification of the Label 002, Equipment ID 002

4.2.2.2.2 BCD Data Encoding and Decoding

Label 002, Equipment ID 002 is Time To Go. It is linear BCD data. Table 9 shows how to encode and decode BCD Data.

32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9
P	SSM		MS Char									Data Field								LS Char		SDI	
			4	2	1	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1		
0	0	0	0	0	1	0	1	0	0	0	1	0	1	0	0	1	1	P	P	P	P	0	0



Time TO GO = 14 5 .3 Mins

Table 9 ARINC 429 Linear BCD Data Encoding and Decoding Example (Label 002.EquipmentID 002: Labels Omitted)

4.2.3 ARINC 429 Signal Characteristics

- ARINC 429 Signal Transmission Rate:
    1. Low speed (12-14.5 K Bits/Sec)
    2. High speed (100.0 K Bits/Sec ± 1%)
- Word Separation

Consecutive ARINC words must be separated by a null period of at least four bit lengths.

4.3 Management of the ARINC Card

The character of the ARINC429 word and how to encode and decode ARINC429 words have been discussed. In DTB the ARINC 429 PC 16 card produced by the SBS Inc. is used as the ARINC interface card. In the FMS Interface Software, the ARINC interface card must be managed well, and it includes proper device management, channel configuration, transmit and receive operations. In this section, ARINC card management is discussed.

4.3.1 Device Management

The interface card in the DTB used for this project was an ARINC 429 PC16 card, which has 16 ARINC 429 channels (a single ARINC 429 bus connection). A device is a logical entity that consists of eight channels. The DTB has two ARINC devices. Device management includes six steps: initialize the ARINC 429 device, create global and channel sequential monitor buffers. start ARINC 429 channel I/O processing, stop ARINC 429 channel I/O processing and close the ARINC 429 device [28]. The SBS company provides the Integrated Avionics Library with the ARINC card, which provides flexible tools for device management programming. Use of the Integrated Avionics Library functions to manage the device follows these steps [29]:

- Initialize the ARINC 429 devices

  (1) Set up the Configuration File. The device parameter values for the ARINC 429 devices are defined in a device configuration file. sbs_dev.cfg, which is an ASCII text file containing information that the FMS Interface Software uses to initialize the ARINC 429 devices. The device information should include device number. the Location of the ARINC 429 device in the physical address space. the number of the receive and transmit channels in the ARINC 429 device, and so on. At the beginning of the FMS Interface Software, a parser function is called to read the information in the configuration file.

(2) Initialize the ARINC 429 device. The ARINC 429 device must be initialized before access is permitted to other functions in the ARINC 429 programming library. The call to the function sbs_init_device( ) initializes the ARINC 429 devices according to the information in the device configuration file.

- Create the global and channel sequential monitor buffers

  Sequential monitoring stores raw ARINC 429 traffic data in linked monitor buffers for real-time data logging, recording, and analysis applications. The ARINC 429 board is capable of storing every ARINC 429 message to monitor buffers and notify the host system of buffer swaps through interrupt and polling functions. This monitoring method is useful when the host system has tight processing constraints, but must still receive every ARINC 429 message word. The key data structures for sequential monitoring are the sequential monitor buffer. There are two types of sequential monitoring: Global and Channel. The global sequential monitor stores data received from all specified channels. The channel sequential monitor is provided for each ARINC 429 receive channel. Only data received on the receive channel is stored. In the DTB, a call to the functions, a429_create_global_sm_buffers( ) and a429_create_channel_sm_buffers( ), creates the global and channel sequential monitor buffers for each ARINC 429 device.

- Start ARINC 429 channel I/O processing

  After channel configuration and monitor creation, call the function sbs_start_io() to enable the device for ARINC 429 bus processing including the ARINC 429 word transmit and receive operation.

- Stop ARINC 429 channel I/O processing. Call the function sbs_stop_io( ) to stop I/O processing on the ARINC 429 bus.
- Close the ARINC 429 device. Call the function sbs_close_device( ) to break the logical connection between the operating system and the ARINC 429 device.

4.3.2 Channel Configuration

After the ARINC 429 devices are initialized, every channel in each ARINC 429 device should be defined. There two ARINC 429 devices in DTB; according to the device configuration file each ARINC 429 device has two receive channels and 6 transmit channels. The channel configuration must be done for each ARINC 429 device. Receive and transmit operations depend on the information of which channels are used as receive channels, and which channels are used as the transmit channel, and therefore, their character must be confined.

A channel configuration file that is in a text format is designed for each ARINC 429 device. This file contains all configuration information for each channel in the ARINC 429 device. After the devices are successfully initialized, the FMS Interface Software reads the channel configuration files.

4.3.3 Transmit and Receive Operation

The channel configuration file defines which channels are to be receive channels and which are to be transmit channels, as well as the character of each receive and transmit channel. After reading the information from the channel configuration files, the FMS

Interface Software executes a transmit or receive operation according to the information it receives. In this section, the basic concept and steps of the transmit and receive operation are explained individually.

4.3.3.1 Transmit Operations

The transmit unit in each transmit channel is called the Transmitter, and it is through here that the DTB transmits the ARINC 429 words to the FMS. A transmit channel can contain one or more transmitters. Transmitter operations are governed by a single data structure consisting of one or more chains, in which each chain consists of a linked list of the transmit command blocks stored in the memory of the ARINC 429 card. Each block within a linked list contains the encoded transmitter information. These encoded transmit blocks are loaded into the ARINC 429 memory when the FMS Interface Software calls the functions supplied in the ARINC 429 programming library. During transmitter operation, the microprocessor on the ARINC 429 card decodes each block of the chain to determine ARINC word transmission sequences.

In the following part, use of the ARINC 429 library functions to run the transmitter operations in the DTB are discussed. The transmitter operations in the DTB are:

- Create a transmit channel control block. After the ARINC 429 device is initialized and the global and channel sequential monitor buffers are created, a transmit channel control block must be created for each transmit channel. The ARINC 429 Programming Library provides the function a429_create_tx_cb( ) to create the transmit channel control block for the transmit channel. The block should include the basic information of the transmitter; for example, which ARINC 429 device the transmit channel belongs to and its channel number, the frequency of the transmit channel (12.5 kHz or 100 kHz), and so on.

- Add a stand transmitter command block to the chain. The ARINC 429 Programming Library assists programmers in building the links and chains through the use of predefined structure types and a call to the function a_429_add_ximt_cmd_blk(). When called by the FMS Interface Software, the a_429_add_ximt_cmd_blk( ) function builds the encoded transmitter control link and copies this structure into the ARINC 429 memory.
- Load the Transmit command chain. Once the bus I/O processing starts, the Transmit command chain should be loaded for transmission execution. The DTB calls the function a429_load_chain( ) to load the Transmit command chain. To ensure that the loading operation is executed correctly and the correct chain has been loaded for the correct device channel, the parameters of this function should include the device number. channel number and the chain number defined in the channel configuration file.
- Transmit the ARINC 429 words on the transmitter. The designated ARINC 429 word labels to be transmitted in each transmit channel are defined in the channel configuration files. The FMS Interface Software assigns a data buffer for each transmitter to store the specific data structure containing the ARINC 429 words what to be transmitted in each transmitter. After the flight data is encoded and combined with the labels. it is stored in the data buffer. At this point, the FMS Interface

Software calls the function a429_write_tx_cb_data( ) to transmit the data in the buffer. In order to carry out the transmission correctly, the parameters of this function describe the information of the transmit, including ARINC 429 device number, the pointer to the transmit command block and the pointer to the data buffer storing the data structure that should be transmitted.

- Halt the transmit chain. Before the ARINC 429 device stops ARINC 429 bus I/O processing, the FMS Interface Software calls the function a429_halt_cb( ) to stop I/O processing on the transmitter.

4.3.3.2 Receive Operations

The receive unit in each receive channel is called the receiver; it is through here that the DTB obtains the ARINC 429 words from the FMS. A receive can contain one or more receivers. Receive operations are governed by three primary groups of data structures: Receiver Channel Control Block, Receiver Control Word and Channel Filter Table. The receive data structure is comprised of the words that describe the feature of the receiver and the receive channel. The ARINC 429 Programming Library assists the programmer in setting up these data structures and in operating the receivers.

In the following part, use of the ARINC 429 library functions to run the receiver operations in the DTB is discussed. The receiver operations in the DTB are:

- Create the receiver channel control block. After the ARINC 429 device is initialized and the global and channel sequential monitor buffers are created, a RECEIVE channel control block must be created for each receiver. The ARINC 429

Programming Library provides the function a429_create_rc_cb( ) to create the receiver channel control block for the receive channel. The block should include the basic channel information including which ARINC 429 device the receive channel belongs to and the channel number, the frequency of the receiver channel (12.5 kHz or 100 kHz), and so on.

- Set the receiver control word. The FMS Interface Software calls the function a429_set_rc_control( ) to set the receiver control word for each receiver.
- Set the Channel Filter Table. The channel configuration file defines which labels are received in each receive channel. The FMS Interface Software calls the function a429_set_filter_table( ) to set a channel filter table for each receive channel to ensure that the proper ARINC 429 words are received by each receive channel.
- Start the receive channel. Once the bus I/O processing starts, the FMS Interface Software calls the function a429_start_rc( ) to enable the receive channel.
- Read the current values for the labels from the receiver. The function a429_get_current_value( ) is called by the FMS Interface Software for each receive channel to get the values of the labels from the receivers. In order to read the values correctly, information such as the device number, channel number and the labels should be input through the parameters of this function.
- Stop the receive channel. Before the ARINC 429 bus I/O processing is stopped, the FMS Interface Software must call the function a429_stop_receiver( ) for each receive channel to stop the receive channels.

4.4 ARINC429 Channel Switch Software

In the Section 4.3.2, the ARINC 429 channel configuration and the channel configuration file of DTB are discussed. The channel configuration may be different each time because sometimes the DTB users may want to add or remove some ARINC 429 words, or they may want to switch the channels. In order to provide a user friendly interface to the DTB user, the ARINC429 Channel Switch Software is developed, which can produce the channel configuration files for each ARINC 429 device to meet the user's requirements. Details are explained below.

4.4.1 ARINC 429 Channel Configuration Files in DTB

There are two ARINC 429 devices in DTB, and each device has eight ARINC 429 channels. Once each ARINC 429 device is initialized successfully, it does the channel configuration for its eight ARINC 429 channels according its ARINC 429 channel configuration file. In the ARINC 429 configuration file, the properties of each channel are defined. These properties are: the channel is a receive channel or transmit channel; the ARINC 429 signal transmission rate of the channel is high or low; how many labels and which labels are received or transmitted in the channel.

4.4.2 Development of the ARINC429 Channel Switch Software

The ARINC429 Channel Switch Software is developed using Microsoft Visual C++

ActiveX controls. The GUI of the ARINC429 Channel Switch Software is illustrated in Figure 18. Eight slider controls are designed for each device, which represent eight channels in each device. Two group boxes are designed for each device also: the input box and output box. The input box contains the receive channel sliders; and the output box contains the transmit channel sliders. A static text control is also designed on the left of each slider control to display the name of the navigational signal that are used by the sliders. The bar on each slider can be pulled to select the channel number, and the channel number is displayed on the right of the slider. In addition, three groups of the single select buttons are designed for three output sliders in Device 1. Such three sliders can make a single choice as the signal name from the AHRS. IRS and INS. There are three buttons that are designed on the bottom of the GUI. They are Apply button, Set Default button and the Close button. If the Set Default button is pressed down, a recommended channel configuration is displayed on the GUI. After the users select their desired configuration through the controlls on the GUI, the Apply button can be pressed down to produce the new channel configuration files for two devices. The Close button can be pressed down to quit the ARINC429 Channel Switch Software.

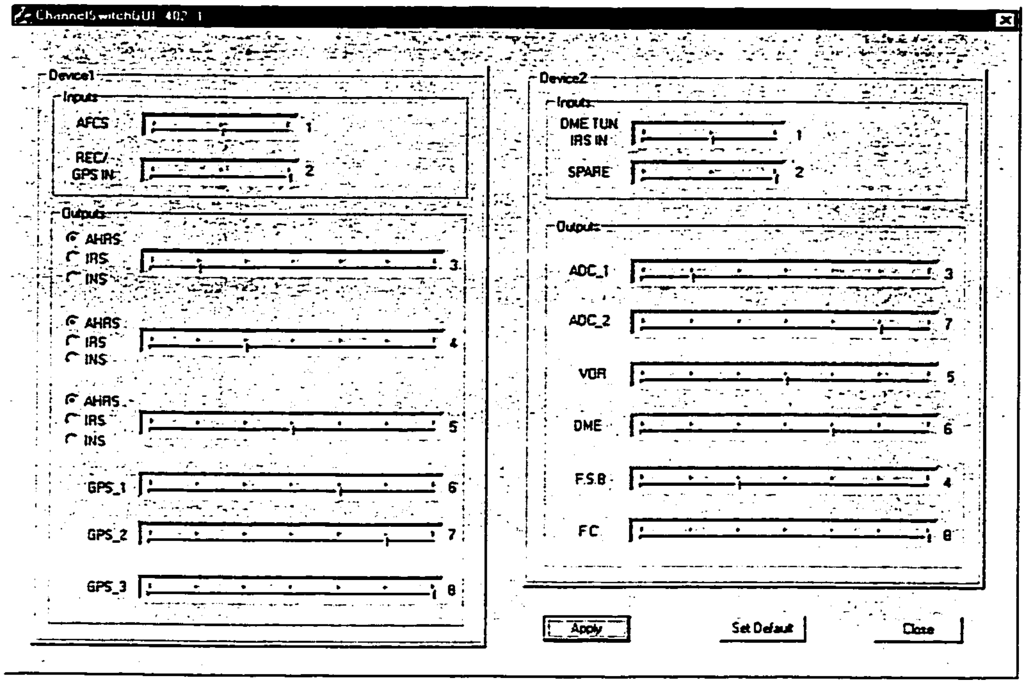


Figure 18 The ARINC429 Channel Switch Software GUI

CHAPTER 5

DTB SYSTEM INTEGRATION AND TESTING

In this chapter, the DTB system integration and testing is discussed.

First, all the software are tested and integrated into the DTB. Then the DTB is connected to the FMU 402 developed by CMC Electronics. In order to test whole DTB system, including software and hardware, a full flight plan from Montreal to Toronto (CYMX to CYYZ) is input and executed. All the test work is done at Concordia University first. Then the DTB is tested at the FMS test and development lab of CMC Electronics, where the DTB is currently being used.

5.1 DTB Software System Integration and Testing

After programming and debugging codes for the Dynamic Flight Simulation Software, the User Interface Software, the FMS Interface Software and the ARINC429 Channel Switch Software under the Microsoft Visual C++6.0 Developer, the release version of EXE files for these software are produced.

5.1.1 Execute the ARINC429 Channel Switch Software

On the I/O PC, the ARINC429 Channel Switch Software is run to produce the Channel Configuration files for the FMS Interface Software. Figure 19 shows the running of the ARINC429 Channel Switch Software. On the ARINC429 Channel Switch GUI, users can pull the slider of each type of ARINC 429 words one by one to configure the channels in the device, they can also press the default button to select the default setup configuration for the devices. Finally the user presses the Apply button to produce the configuration files. In the case of Figure 19, Channel 1 and Channel 2 of ARINC Device 1 are configured as input channels of FMS, and the channels from 3 to 8 are configured as output channels of FMS. In Device 1, the signal AFCS is set on Channel 1; REC/GPS IN is set on Channel 2; AHRS is set on Channel 3, 4 and 5; GPS is set on Channel 6, 7 and 8. In Device 2, ADC is set on Channel 3 and 7, which are the output channels of the FMS.

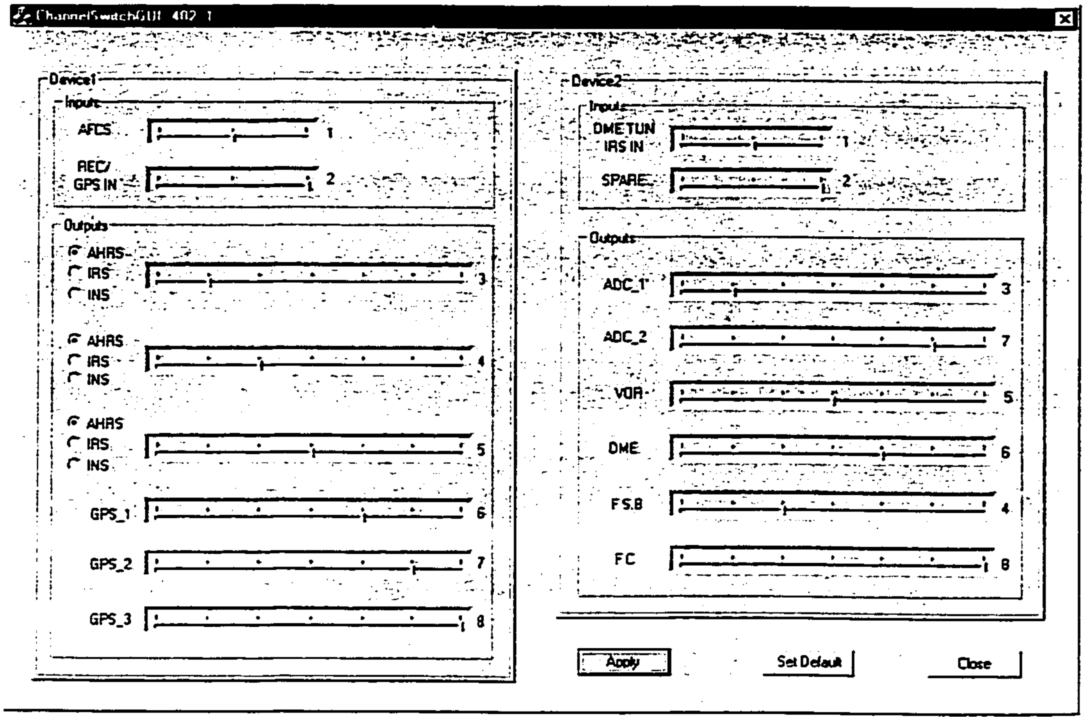
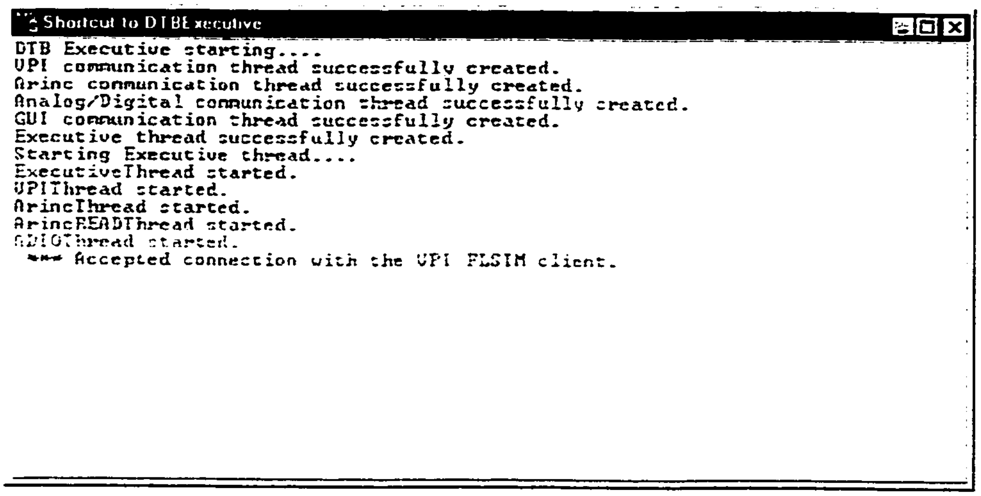


Figure 19 Running the ARINC429 Channel Switch Software on the I/O PC

5.1.2 Execute the FMS Interface Software

On the I/O PC, the FMS Interface Software is run. Figure 20 shows the running of the FMS Interface Software.

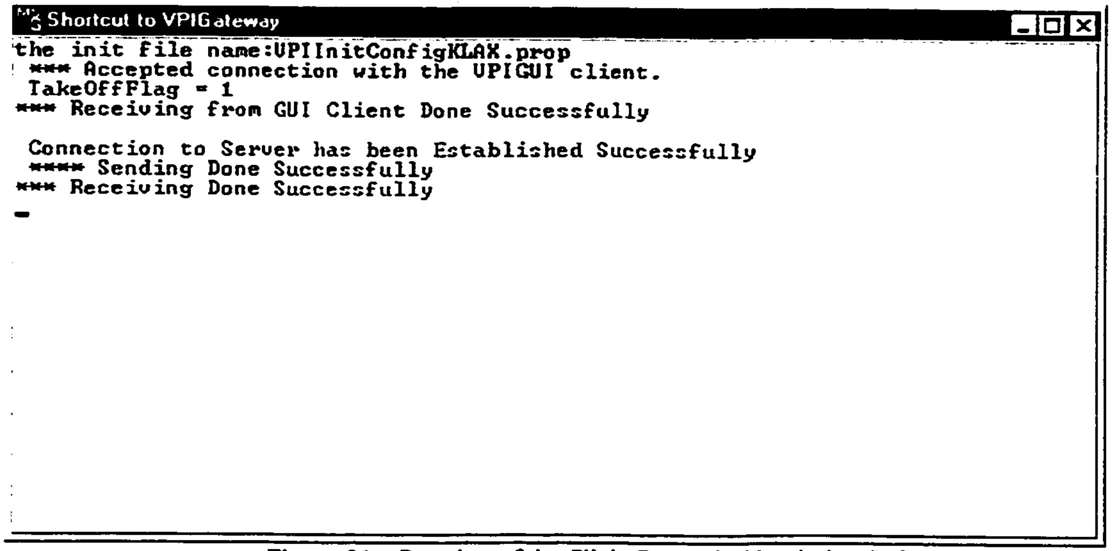


```
DTB Executive starting....
UPI communication thread successfully created.
Arinc communication thread successfully created.
Analog/Digital communication thread successfully created.
GUI communication thread successfully created.
Executive thread successfully created.
Starting Executive thread....
ExecutiveThread started.
UPIThread started.
ArincThread started.
ArincREADThread started.
ADIOThread started.
 *** Accepted connection with the UPI FLSIM client.
```

Figure 20 Running of the FMS Interface Software on the I/O PC

5.1.3 Execute the Flight Dynamic Simulation Software

On the simulation PC, the Flight Dynamic Simulation Software is run. Figure 21 shows the running of the Flight Dynamic Simulation Software.

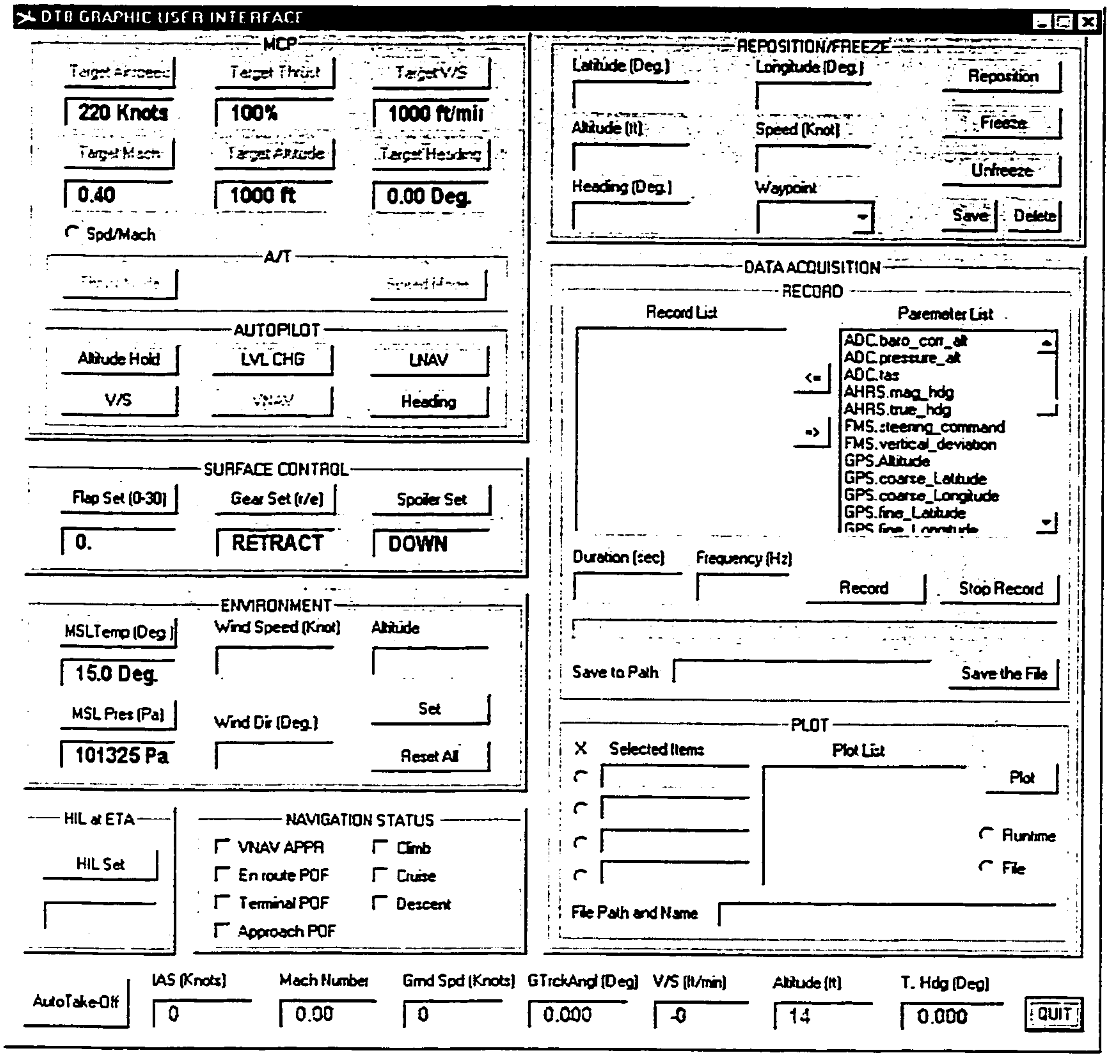


```
the init file name:UPIInitConfigKLAX.prop
 *** Accepted connection with the UPIGUI client.
 TakeOffFlag = 1
*** Receiving from GUI Client Done Successfully

 Connection to Server has been Established Successfully
  **** Sending Done Successfully
*** Receiving Done Successfully
_
```

Figure 21 Running of the Flight Dynamic Simulation Software

5.1.4 Execute the User Interface Software

On the simulation PC, the User Interface Software is run. Figure 22 shows the running of the User Interface Software.



Target Airspeed: 220 Knots
Target Thrust: 100%
Target V/S: 1000 ft/mi
Target Mach: 0.40
Target Altitude: 1000 ft
Target Heading: 0.00 Deg.
Flap Set (0-30): 0.
Gear Set (r/e): RETRACT
Spoiler Set: DOWN
MSLTemp (Deg): 15.0 Deg.
MSL Pres (Pa): 101325 Pa
IAS (Knots): 0
Mach Number: 0.00
Grnd Spd (Knots): 0
GTrckAngl (Deg): 0.000
V/S (ft/min): -0
Altitude (ft): 14
T. Hdg (Deg): 0.000

Figure 22 Running of the User Interface Software on the simulation PC

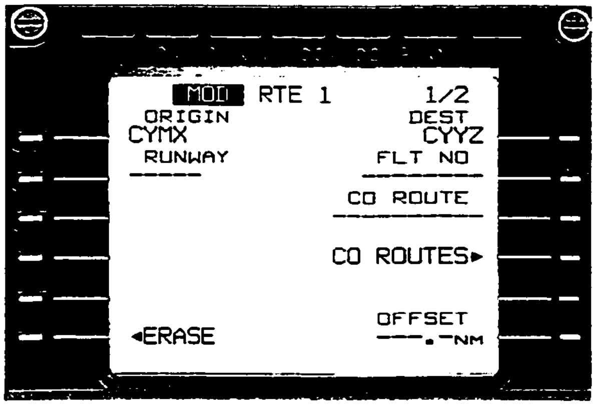
5.2 DTB System Integration and Testing

After the software system integration, the DTB is connected to the FMU through the breakout box. An entire flight plan is executed in the DTB in order to test and evaluate the DTB system. In this section, the test procedures and the evaluation of the test results are described.

5.2.1 Flight Plan Setup

In order to test the DTB, a real standard flight plan is selected from the navigation database stored in the FMU, from CYMX (Montreal International Airport) to CYYZ (Toronto international Airport). This flight plan is introduced next, and how to enter this flight plan through CDU to FMU is illustrated.

- Step1. Enter the departure airport (CYMX) and the arrival airport (CYYZ) in the CDU Route page (Figure 23).

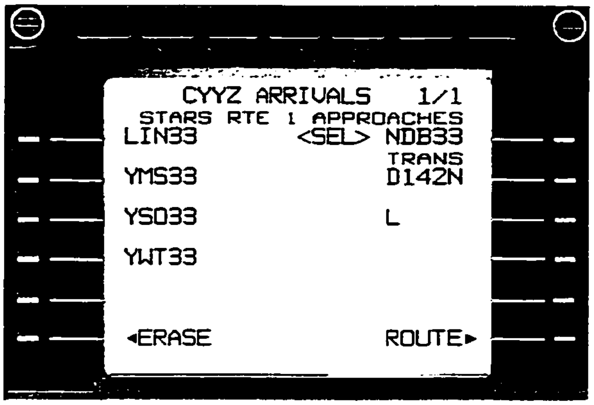


MOD RTE 1 1/2
ORIGIN DEST
CYMX CYYZ
RUNWAY FLT NO

CO ROUTE
CO ROUTES►
◄ERASE OFFSET
---.-NM

Figure 23 Step1 of the flight plan setup

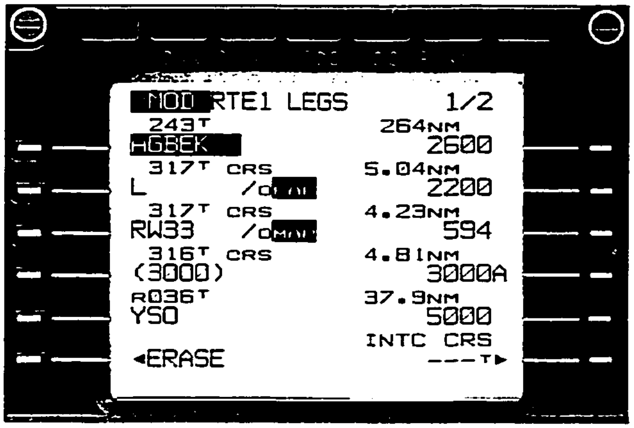
- Step2. Select the NDB33 as the APPROACH in the Arrival page on the CDU (Figure 24).



```
    CYYZ ARRIVALS    1/1
STARS RTE 1 APPROACHES
LIN33         <SEL> NDB33
                    TRANS
YMS33               D142N

YSO33               L

YWT33


◄ERASE              ROUTE►
```

Figure 24 Step2 of the flight plan setup

- Step3. After Step2 the FMU selects a standard route then display it on the CDU Route Legs page. Check the route legs and then press the EXE key to finish the flight setup.



```
MOD RTE1 LEGS        1/2
 243T              264NM
HGBEK               2600
 317T CRS         5.04NM
L      /O[illegible]           2200
 317T CRS         4.23NM
RW33   /O[illegible]            594
 316T CRS         4.81NM
(3000)             3000A
R036T             37.9NM
YSO                 5000
                INTC CRS
◄ERASE             ---T►
```

Figure 25 Route Legs page of the flight plan

Figure 25 shows all waypoints in this flight plan. This page displays the details of the flight plan selected. The airplane departs from CYMX and then flies by AGBEK. The airplane finishes the final approach fix (FAF) at L, and then flies to RW33.

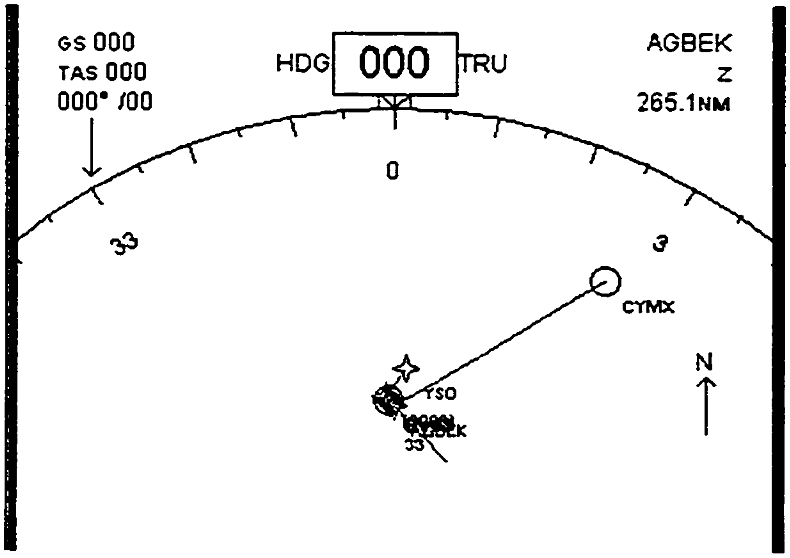


Figure 26 The flight plan from CYMX to CYYZ displayed on the EFIS

When the aircraft reaches RW33, it begins to carry out the landing on the runway of CYYZ. If the aircraft cannot execute the landing, RW33 is considered a missed approach point (MAP) and the airplane goes around at (3000) to fly to the next waypoint at YSO. Figure 26 shows the flight plan on the EFIS. In order to display the approach area, the approach part of the flight plan is enlarged in Figure 27.

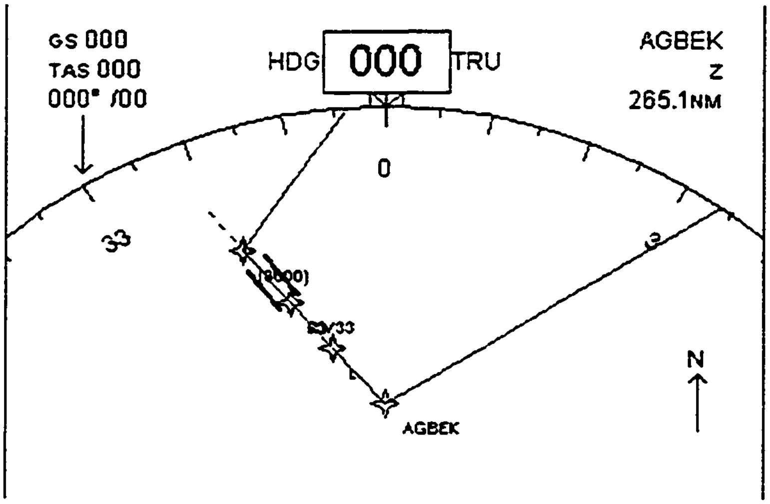


Figure27 The approach area of the flight plan on EFIS

5.2.2 Executing the Flight Plan on the DTB

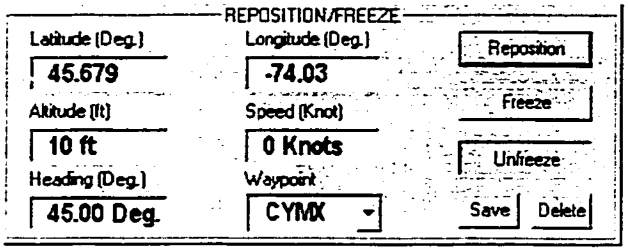
The phases of the flight are defined as follows:

- Approach. When below 15,000 feet AGL (Above Ground Level) and within 2nm of the FAF (Final Approach Fix), with all GPS instrument approach conditions satisfied.
- Terminal. For arrivals, when below 15,000 feet AGL and within 30nm radial distance of the arrival airport, but not in the approach phase of the flight. For departures, when below 16,000 feet AGL and less than 33nm radial distance from the departure airport.
- En-route. When in neither approach nor terminal phases of flight.

The flight simulation on the DTB includes all phases of the flight plan: Approach, Terminal and En-route. In the following section all flight phases on the DTB is executed, and in the simulation the flight data is recorded to test and analyze the DTB.

5.2.2.1 Preflight

Run the DTB software system. On the Reposition frame of the DTB GUI the aircraft is repositioned at CYMX; the initialized values are displayed in Figure 28. Then the simulation is unfrozen, and now the Aircraft is on the ground at CYMX heading 45 degrees from the true north.



Latitude (Deg.): 45.679
Longitude (Deg.): -74.03
Altitude (ft): 10 ft
Speed (Knot): 0 Knots
Heading (Deg.): 45.00 Deg.
Waypoint: CYMX

Figure 28 Reposition the Aircraft on the ground at CYMX.

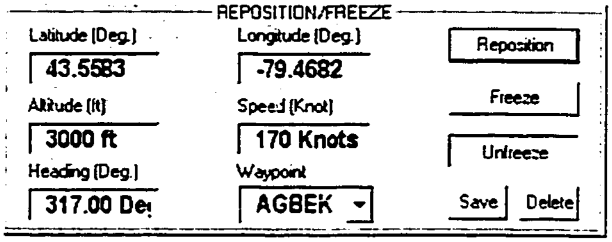
After positioning the aircraft on the ground at CYYZ, enter the flight plan from CYMX to CYYZ into the FMS (as discussed above). Check the message on the CDU; if the aircraft status is normal, get ready to takeoff from the aircraft.

5.2.2.2 Terminal for Departure

On the DTB GUI, the target airspeed is entered as 220 knots and the Auto Takeoff button on the DTB GUI is pressed; the flight simulation enters the takeoff phase. When the aircraft reaches 400 ft, Takeoff finishes, and the simulation engages the autopilot of the level change mode automatically. 1000 ft is selected as the target altitude of the level change mode. After the aircraft finishes Takeoff, the FMS LNAV mode can be engaged as the lateral flight guidance for the flight. The FMS LNAV mode is engaged by pressing the LNAV button on the DTB GUI. When the aircraft reaches 1,000 ft, the autopilot changes from the level change mode into the Altitude Hold mode. 6,000 ft is selected as the target altitude and then the level change mode is engaged. When the aircraft reaches 6,000 ft, it cruises at an altitude of 6,000ft. The Aircraft flies out of the terminal for the departure and then enters the En-route phase.

5.2.2.3 En-route

In the En-route phase, the aircraft cruises at 6,000 ft and 220 knots. The aircraft flies at heading 243 degreees to AGBEK under the FMS LNAV mode. In the flight simulation, the reposition function on the DTB GUI can be used to shorten the En-rout phase. When the aircraft catches the heading 243±2 degrees and the error of the cross track is less than 1nm, the aircraft is repositioned to AGBEK. The basic flight data of the new position is shown in Figure 29. The flight simulation then goes to the Terminal for Arrival.


REPOSITION/FREEZE

Latitude (Deg.) 43.5583
Longitude (Deg.) -79.4682
Altitude (ft) 3000 ft
Speed (Knot) 170 Knots
Heading (Deg.) 317.00 De
Waypoint AGBEK

Figure 29 Reposition the aircraft to AGBEK to shorten the simulation

5.2.2.4 Terminal for the Arrival and Approach

After repositioning to the AGBEK, the flight simulation enters the Terminal for Arrival. Now the SSM of the Vertical Deviation ARINC 429 word is available, so the VNAV button on the DTB GUI is available. The VNAV button is pressed to engage the FMS VNAV mode. The aircraft captures the glide slope under the VNAV mode. At 2nm before the waypoint L, the aircraft reaches the Approach phase. At RW33, the aircraft should execute the landing under Pilot Navigation. Since the DTB does not have the Auto Landing function yet, when the aircraft reaches RW33, the CDU displays the "Missed Approach" message; the SSM of the Vertical Deviation ARINC 429 word is unavailable. The flight quits the FMS VNAV. Then the aircraft keeps the present flight data and goes around at the waypoint (3000), and flies to the last waypoint of the flight plan at YSO. The flight plan from CYMX to CYYZ was executed on the DTB up to this point.

5.2.3 Data Acquisition and Simulation Analysis

During the flight simulation the user can select any flight data from the flight data list on the DTB GUI to record as a Microsoft Excel file or to display the data during runtime.

In Section 5.2.2, a flight plan from CYMX to CYYZ is executed. When this flight plan is executed, the FMS steering command, AHRS true heading, Vertical deviation and the GPS altitude are selected to be displayed during runtime. The curves of those data in all flight phases are shown in Figure 30. The aircraft begins the Auto Takeoff at point A in Figure 30. At point B, the LNAV mode is engaged, and the true heading of the aircraft begins to catch the heading at AGBEK according to the FMS steering command. The aircraft reaches an altitude of 6,000ft and begins to cruise under the Altitude Hold mode from point C onward. At point D, in order to shorten the simulation, the aircraft is repositioned to AGBEK at an altitude of 3,000ft, and the SSM of the ARINC 429 vertical deviation word is available. The FMS VNAV mode is engaged at point E, and then the aircraft begins to capture the Glide Slope according to the vertical deviation. At point G, the aircraft captures the glide slope and the aircraft flies following the glide slope under the FMS VNAV mode. At point F, a missed approach is reported by the FMS, as the DTB does not have the landing mode yet.

The runtime display function [31] of the DTB is used to analyze the execution of the flight plan from CYMX to CYYZ. The flight data can also be recorded as an Excel file to do the analysis. In the following section, the VNAV mode is tested in this way.

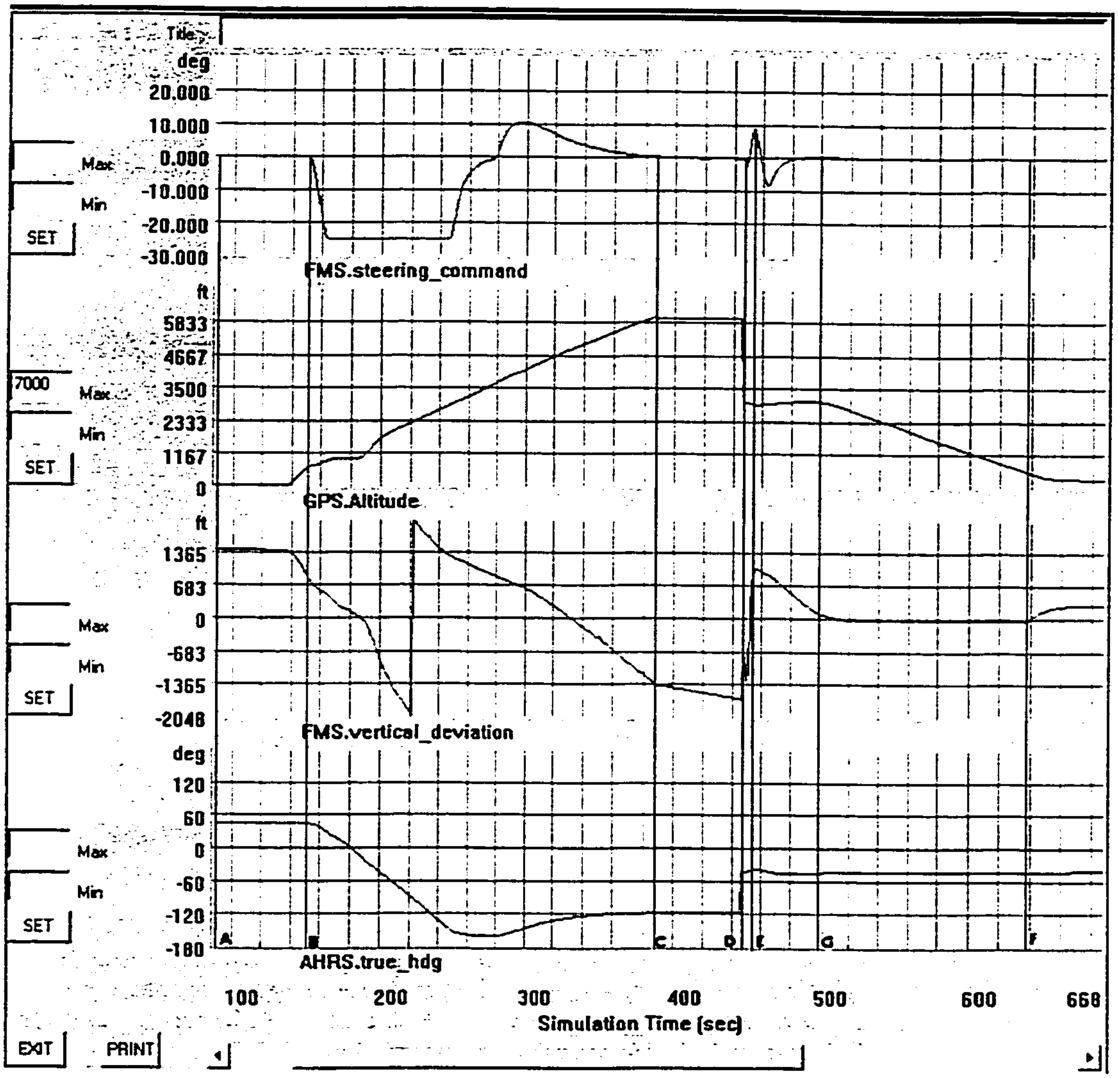


Figure 30 The curves of the FMS steering command, GPS altitude, FMS vertical deviation and the AHRS true heading in all phase from CYMX to CYYZ.

In the beginning of the execution of flight plan CYMX to CYYZ on the DTB, the pitch angle, vertical deviation and GPS altitude of the aircraft, and the simulation time are selected from the DTB GUI to be saved in an excel file at a frequency of 30 Hz. Figure 31 shows the vertical deviation, pitch angle and the GPS altitude response in simulation time obtained from Excel after the VNAV was engaged.

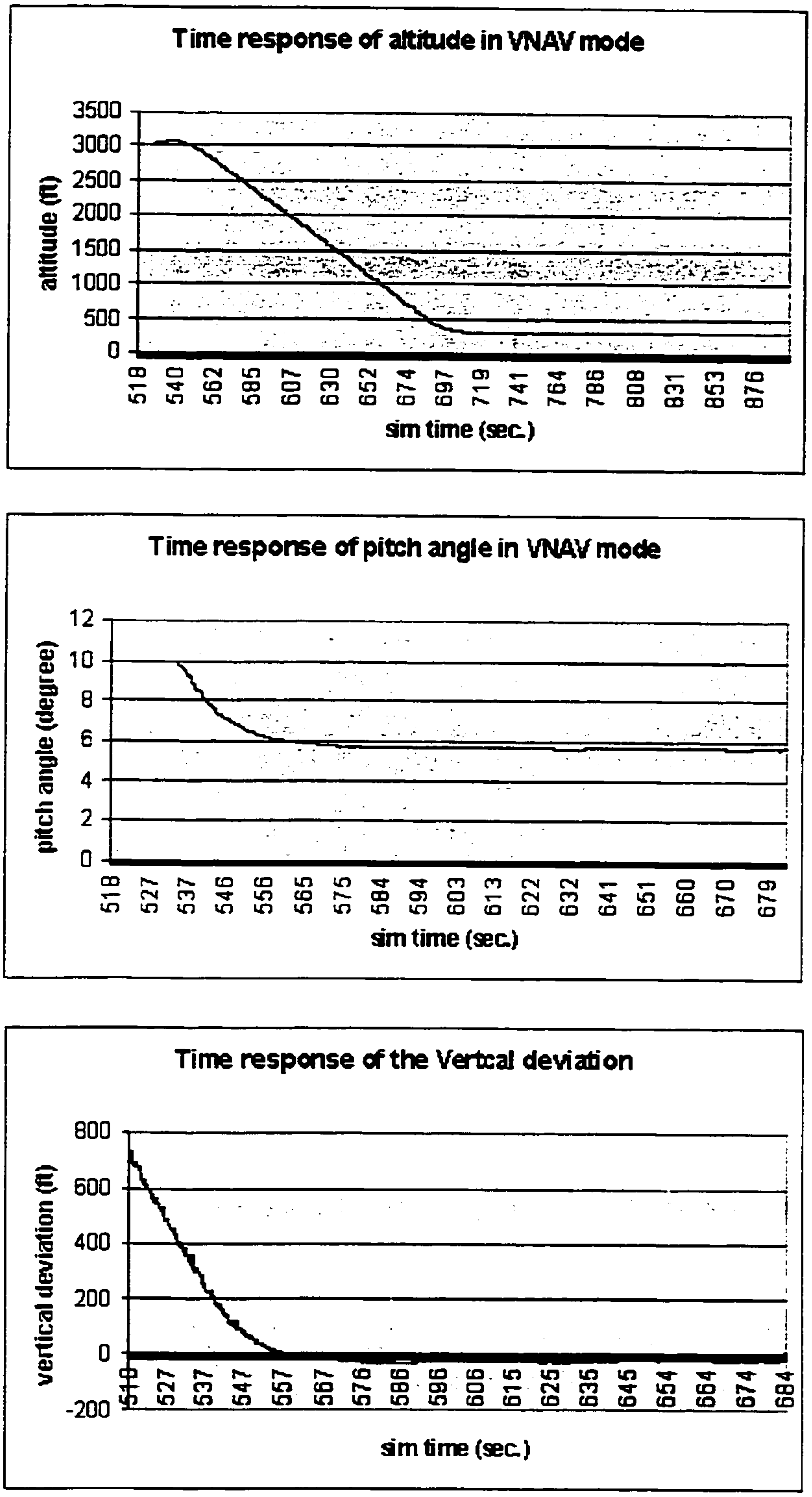


Figure 31 Time response of the vertical deviation, pitch angle and altitude in VNAV mode

From Figure 31, we can see that after repositioning the aircraft to the AGBEK, the VNAV mode was engaged, and the aircraft flies in GPS VNAV mode. At this time the VNAV control law begins to work; it controls the pitch angle according to the FMS vertical steering command, and vertical deviation. The pitch angle changes smoothly. The altitude plot shows that the aircraft flies to capture the glide slope smoothly, then the aircraft tracks the glide slope well. This test shows the FMS vertical deviation function and the performance of the VNAV flight control mode.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

In this chapter, the conclusions of this thesis are given, some ideas about the future work for DTB are also discussed.

6.1 Conclusions

In this thesis, some new important functions of DTB are developed, and are embedded into the DTB. The FMS VNAV mode is discussed in Chapter 3. The IRS navigation mode is explained in Chapter 3. The DTB GUI is redesigned to improve the ergonomic functionality of the DTB, as discussed in Chapter 4. The ARINC 429 channel switch tool is also discussed in Chapter 4. All these new functions of DTB are tested at Concordia University and at the FMS Test and Development Lab of CMC Electronics where the DTB is currently being used.

6.1.1 Development of The FMS VNAV Mode

The DTB has the FMS LNAV autopilot mode developed by Chui [31]. The aircraft can fly under the lateral steering command from FMS in LNAV mode. In this thesis, the VNAV mode is developed, so that the aircraft can fly under the vertical steering command from FMS in the VNAV mode when the aircraft reaches the approach area. The FMS VNAV mode enables the GPS approach on the DTB, and the DTB provides an effective way to test the vertical guidance behavior of the FMS.

6.1.2 Development of The IRS Navigation Mode

GPS, AHRS and ADC navigational sensors had been developed for the DTB by Chui [31]. In this thesis, the IRS navigation sensor was developed and added into the dynamic flight simulation. The DTB can be used to test the IRS navigation function of the FMS.

6.1.3 Redesign of the DTB GUI

The MCP of the DTB GUI is redesigned in this thesis. In particular, the MCP frame on the DTB GUI is redesigned according to the definition of all AFCS modes in DTB. In this way, the ergonomic functionality of the DTB is improved.

6.1.4 Development of the ARINC429 Channel Switch Software

In this thesis, the ARINC 429 Channel Switch Software is developed for the DTB. This tool reduces much of the workload for DTB users, specifically when the ARINC channel configurations need to be changed. This tool can produce the channel configuration files automatically according to what the DTB user desires. This function provides convenience to the DTB users

6.2 Future Work

After the development of two phases, the major structure and the basic functions of the DTB have been developed. Since the concept of the DTB should contain many aspects, there is much work to be done in the future development of the DTB. These are outlined below.

6.2.1 Flight Simulation

The flight simulation on the DTB was developed on the basis of FLSIM 7.0. At present, only one type of aircraft, Boeing 747, can be simulated on the DTB. The DTB should have the flexibility to simulate different aircraft types, and even could include the helicopter. So far, the DTB does not have the auto-landing mode yet, so the flight plan cannot be completed by the landing on the runway and always reports a missed approach and then carries out the flight by going around to the next waypoint. Developing an auto-landing mode is very important addition for the DTB project. In Addition, more navigation sensor simulation should be developed and added into the flight simulation, such as VOR and DME.

6.2.2 Subsystems of the Aircraft

Today's FMS was developed for the management of the subsystems of the aircraft such as the fuel system, door system, air conditioning system, and so on. The DTB should also be developed to contain the test functions for those subsystems. Simulating those subsystems in the software, and then adding them to the DTB software system is a significant job for the future development of the DTB.

6.2.3 Maintaining and Updating the DTB

The DTB should be developed to follow the development of the FMS. When the FMS is being developed, the same requirement should be submitted for development in the DTB. On the other hand, FLSIM and the ARINC 429 Programming Library will update their old versions and this also provides the chance to update the DTB. The DTB should be maintained and updated in the future to meet more complex requirements.

REFERENCES

1. RonLindenfeld, "What is an FMS?", Laminar Research http://www.ma.ultranet.com/~marzgold/FAQ-FMS.html
2. "Description and Operation of FMS", The Aeronautical Curiosities http://digilander.iol.it/andreatheone/fmcs.htm
3. Samuel B. Fishbein, "Flight Management Systems the Evolution of Avionics and Navigation Technology", Praeger Publishers, 1995.
4. C. Tomlin, G. Pappas, J. Lygeros, D. Godbole, and S. S. Sastry, "A Next Generation Architecture for Air Traffic Management Systems," *Proc. IEEE Conf. Decision and Control,* New Orleans, LA, December 1995.
5. Jim Curran. "Trends in Advanced Avionics", Iowa State University Press/Ames, 1992.
6. Myron Kayton & Walter R. Fried, "Avionics Navigation Systems", Second Edition, John Wiley & Sons, 1997.
7. "OPERATOR'S MANUAL, Flight Management System CMA-900/CMA-2014", Marconi Canada, 1999.
8. "CMA-900 Flight Management System Installation/Flight Line Manual", Marconi Canada, 1999.
9. "CMA-900 GPS/FMS", MARCONI CANADA, 1998.
10. Jörg Marrenbach, Karl-Friedrich Kraiss, "Advanced Flight Management System A new Design and Evaluation Result", HCI-Aero 2000-International Conference on

Human-Computer Interaction in Aeronautics, Sep. 27-29 2000, Toulouse France.

11. Peter Lawn, " The Enhancement of a Flight Simulator System with Teaching and Research Applications", MASc thesis, Concordia University, 1998.
12. Jean-Luc Boiffier and Onera-Cert, " the Dynamics of Flight, the Equations", John Wiley & Sons, 1998.
13. Martin E Eshelby, "Aircraft Performance: Theory and Practice", American Institute of Aeronautics and Astronauts, Inc, 2000.
14. M. V. Cook. "Flight Dynamics Principles", John Wiley & Sons, 1997.
15. Bernard Etkin and Lioyd Dufl Reid, "Dynamics of Flight Stability and Control", Third Edition, John Wiley & Sons, 1996.
16. Mark E. Connely, "Simulation of Aircraft", M.I.T. Report, Massechusetts Institute of Technology, 1958.
17. R. C. Nelson, "Flight Stability and Automatic Control". McGraw Hill Inc. New York. 1989.
18. J. Roskam, "Airplane Flight Dynamics and Automatic Flight Controls". Roskam Aviation and Engineering Corporation. Ottawa, 1994.
19. J. Roskam, "Airplane Aerodynamics and Performance", Roskam Aviation and Engineering Corporation, Ottawa, 1988.
20. "FLSIM Reference Guide", Version 7.0, Virtual Prototypes Inc. 1998.
21. "FLSIM Real-Time Data Export ICD", Version 7.0, Virtual Prototypes Inc. 1998.
22. "FLSIM Tutorial", Version 7.0, Virtual Prototypes Inc. 1998.
23. "Programmer's Guide", Version 7.0, Virtual Prototypes Inc. 1998.

24. "Avionics Systems", CAE electronics, 1996.

25. "Avionics Fundamentals", Jeppesen, Sanderson Training Products, 1991.

26. E H J Pallett, "Aircraft Instruments & Integrated Systems", Longman Scientific Technical, 1996.

27. Edward L. Safford, "Aviation Electronics Handbook", Tab Books, 1975.

28. "ARINC User's Manual", SBS Avionics Technologies, 1999.

29. "Integrated Avionics Library Reference", SBS Avionics Technologies, 1999.

30. David J. Kruglinski, Scot Wingo, and George Shepherd, "Programming Visual C++", Fifth Edition, Microsoft Press, 1998.

31. Nianfu Chui, "The Development of a Dynamic Test Bed for Flight Management Systems", MASc thesis, Concordia University, 2001

32. B. Hofmann-Wellenhof, H. Lichtenegger, and J. Collins, "Global positioning system: theory and practice", 4th, rev. ed, Wien: Springer, c1997.

33. Brian Kendal, "Manual of Avionics", Blackwell Scientific Publications, 1993.

34. Donald J. Clausing, "Aviator's Guide to Navigation", 2nd Edition, TAB Books, 1992.

35. Cary R. Spiter, "Digital Avionics Systems Principles and Practices", 2nd Edition, McGraw-Hill, Inc., 1993

36. Dr. J. Svoboda, S. Lasnier, G.M. Mckinnon, "Microprocessor-Bases Light Aircraft Simulartor", Canadian Industrual Computer society, Ottawa, 1986.

37. Ernest E. Sechler and Louis G.Dunn, "Airplane structural analysis and design", Dover Publication, New York, 1963

38. E. Pallett, "Aircraft Instruments", 2nd Edition, Pitman Publishing, Marshfield,1982

APPENDICES

Appendix A. Background of the Flight Simulation

1.0 Aircraft Dynamic Modeling

Before the mathematical modeling equations are deduced, the coordinate systems must be defined and introduced.

1.1 Coordinate System

There are six different reference frames to express the equations of motion of an aircraft. These reference frames are:

- Earth Centered Geodetic System (ECG)
- World Coordinate System (WCS)
- North-East-Down system (NED)
- Body System
- Stability System
- Wind System

1.1.1 Earth Centered Geodetic (ECG) System

The ECG system is a reference frame translating and rotating with the earth [13][20]. In this system, a point is represented by latitude, longitude and altitude. The geodetic latitude, the type commonly used on geographical maps, is the angle at which a line perpendicular to the surface of the ellipsoid at the given point intersects the equatorial plane. The altitude represents the height above the ellipsoid along a line perpendicular to the surface of the ellipsoid at the given point.

The latitude ranges between –90 and +90 degrees, where +90 is the North Pole, 0 is the equator and –90 is the South Pole. The longitude can be between –180 and +180 degrees, where 0 is the Greenwich meridian, the positive direction is eastward and the negative direction is westward.

1.1.2 World Coordinate System (WCS)

The WCS reference frame is the system defined in the Distributed Interactive Simulation (DIS) standard. Its origin is at the center of the earth and is translating and rotating with the earth. Its z-axis is defined along the earth spin axis and pointing towards the North Pole, and its x-y plane is embedded in the earth equatorial plane. The y-axis is always pointing towards the east (90 degrees of longitude) and the x-axis is always intersecting the earth surface at the Greenwich meridian [13][20][21].

1.1.3 North-East-Down System (NED)

The NED system moves with the aircraft and is vertically below the center of the gravity of the aircraft so that its x-y plane is tangent to the earth's surface [20][21].

The x-axis is aligned with the north-south direction and the positive direction is towards the north. The y-axis is aligned with the east-west direction and the positive direction is towards the east. The z-axis is the normal to the earth's surface and the positive direction is downward.

1.1.4 Body System

The body system is fixed in respect to the aircraft, with its origin at the aircraft's center of the gravity [14][20]. The x-axis is in the aircraft's plane of symmetry, parallel to the wing root chord line and with the positive direction towards the aircraft's nose. The y-axis is normal to the plane of symmetry and the positive direction is towards the right wing. The z-axis is perpendicular to the x-y plane and the positive direction is downward.

1.1.5 Stability System

The stability system consists of a rotation of the body system around its y-axis. The degree of the rotation is equal to the angle of attack of the aircraft [17].

1.1.6 Wing System

The wind system is also called the flight path frame. It consists of a rotation of the stability system around its z-axis. The degree of the rotation is equal to the sideslip of the aircraft [17][20][38].

1.2 Transformations of the Coordinate Systems

In order to transfer from one coordinate system to another, we introduce the Euler angels first; then we do the transformations of the coordinate systems.

1.2.1 Euler Angles

Euler angles allow for the conversion from one axis system to another. The Euler angles between the different systems define the orientation of one coordinate system within another [11].

The procedure of transfer from one coordinate system to another is as follows:

1. Align origin of the both systems.
2. Yaw about Z-axis by angle ψ.
3. Pitch about Y-axis by angle ϕ.

4. Roll about X-axis by angle θ.

To transfer from one coordinate system to another we yaw, pitch and roll by the Euler angles ψ, θ, ϕ respectively.

1.2.2 Transformations

The following transformations can be converted between the coordinate system used by flight simulation software:

- WCS to NED system
- NED to Body system
- Body to Stability system
- Stability to Wind system

1.2.2.1 WCS to NED system

The transformation from the WCS system to the NDE system can be expressed by performing two rotations. First performed is a rotation about its z-axis through the geodetic longitude angle followed by a rotation about the y-axis. The rotation about the y-axis can be represented by two successive rotations: a rotation through the geodetic latitude angle and then a rotation of –90 degrees. However, by the definition of latitude, the angle of the first rotation about the y-axis must be the negative value of the latitude in order to perform a right-handed rotation [15][20].

This transformation can be written as:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{NED} = \begin{bmatrix} -\sin(\mu_g)\cos(\lambda_g) & -\sin(\mu_g)\sin(\lambda_g) & \cos(\mu_g) \\ -\sin(\lambda_g) & \cos(\lambda_g) & 0 \\ -\cos(\mu_g)\cos(\lambda_g) & -\cos(\mu_g)\sin(\lambda_g) & -\sin(\mu_g) \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{WCS}$$

where:

μ_g is the geodetic latitude

λ_g is the geodetic longitude

Since the transformation matrix preserves the length of a vector, the inverse transformation from the NED system to the WCS system is given by the transpose matrix.

1.2.2.2 NED to Body system

The transformation from the NED system to the body system can be performed using three rotations. The first rotation is about the z-axis through the yaw angle. The second rotation is about the new y-axis through the pitch angle. The third rotation is about the new x-axis through the roll angle [15][17].

The result can be written as:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{body} = \begin{bmatrix} l_x & m_x & n_x \\ l_y & m_y & n_y \\ l_z & m_z & n_z \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{NED}$$

where:

$l_x = \cos(\theta)\cos(\psi)$

$m_x = \cos(\theta)\sin(\psi)$

$n_x = -\sin(\theta)$

$l_y = -\cos(\phi)\sin(\psi) + \sin(\phi)\sin(\theta)\cos(\psi)$

$m_y = \cos(\phi)\cos(\psi) + \sin(\phi)\sin(\theta)\sin(\psi)$

$n_y = \sin(\phi)\cos(\theta)$

$l_z = \sin(\phi)\sin(\psi) + \cos(\phi)\sin(\theta)\cos(\psi)$

$m_z = -\sin(\phi)\cos(\psi) + \cos(\phi)\sin(\theta)\sin(\psi)$

$n_z = \cos(\phi)\cos(\theta)$

and the ψ, θ, and ϕ represent Euler angles for the yaw, pitch and roll respectively.

Since the transformation matrix preserves the length of a vector, the inverse transformation from body system to NED system is given the transpose [20][36].

1.2.2.3 Body to Stability System

The transformation from the body system to the stability system can be performed using a rotation about the y-axis through the angle of the attack [17][18].

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{stab} = \begin{bmatrix} \cos\alpha & 0 & \sin\alpha \\ 0 & 1 & 0 \\ -\sin\alpha & 0 & \cos\alpha \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{body}$$

where: α is the angle of attack.

Since the transformation matrix preserves the length of a vector, the inverse transformation from the stability system to the body system is given by the transpose [20].

1.2.2.4 Stability to Wind System

The transformation from the stability system to the wind system can be performed using a rotation about the z-axis through the sideslip angle [17][18].

The transformation can be written as:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{wind} = \begin{bmatrix} \cos\beta & \sin\beta & 0 \\ -\sin\beta & \cos\beta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{stab}$$

where: β is the sideslip angle.

Since the transformation matrix preserves the length of a vector, the inverse transformation from the wind system to the stability system is given by the transpose [20].

1.3 Dynamic Mathematics Modeling of the Aircraft

In this section, the mathematics equations of the forces and moments of the aircraft are presented.

1.3.1 Notations

It is useful to define the notations used to describe the various velocities, acceleration, forces, and moments in respect to each axis system. Any axis system could be used for motion calculations; however, the body system is usually the most convenient since the inertia term is constant in this system [11][14][37]. The notations in the body system are shown in Table A.1.

	Roll Axis X_B	Pitch Axis Y_B	Yaw Axis Z_B
Angular Rates	P	q	r
Velocity	U	v	w
Forces	X	Y	Z
Moments	L	M	N
Moment of Inertia	I_{xx}	I_{yy}	I_{zz}

Table A.1 The notations in the body system

1.3.2 The Equations of Motion

There have already been many researchers who have completed studies on the equations of aircraft motion in relation to the Newtonian equations of motion; therefore, in the following section, just the equations is given [11][12][14]:

- Force Equations

The force equations in the body axes:

$$\dot{u} = \frac{F_{xb}}{m} + rv - qw$$

$$\dot{v} = \frac{F_{yb}}{m} - ru + pw$$

$$\dot{w} = \frac{F_{zb}}{m} + qu - pv$$

- Moment Equations

The moment equations in the body axes:

$$\dot{p} = \frac{T_{xb}}{I_{xx}} + (\frac{I_{yy} - I_{zz}}{I_{xx}})qr + \frac{I_{xz}}{I_{xx}}(pq + \dot{r})$$

$$\dot{q} = \frac{T_{yb}}{I_{yy}} + (\frac{I_{zz} - I_{xx}}{I_{yy}})pr + \frac{I_{xz}}{I_{yy}}(r^2 - p^2)$$

$$\dot{r} = \frac{T_{zb}}{I_{zz}} + (\frac{I_{xx} - I_{yy}}{I_{zz}})pq + \frac{I_{xz}}{I_{zz}}(\dot{p} - rp)$$

Both the force and the moment equations can be derived for any other axes system.

2.0 Atmospheric Modeling

Temperature and pressure can affect an aircraft's engines as well as its aerodynamic characteristics [11][15]. A reference temperature and pressure were adopted by the International Civil Aviation Organization, ICAO. The atmosphere at sea level on a standard day is shown in Table.A.2

Parameter	Symbol	English Units	Metric Units
Temperature	T_0	+59°F +518.67°R	+15°C +288.15°K
Pressure	P_0	$2116.22 lb\text{-}ft^{-2}$	$1.01325x10^5 newtons\text{-}m^{-2}$
Density	ρ_0	$2.37688x10^{-3} slug\text{-}ft^{-3}$	$1.225 kg\text{-}m^{-3}$
Tropopause Lapse Rate	A	$-0.003566°R\text{-}ft^{-1}$	$-6.5°K\text{-}km^{-1} 0.00198°C\text{-}ft^{-1}$
Gas Constant R	R	$1716.5 ft\text{-}lb\text{-}slug^{-1}\text{-}°R^{-1}$	$287.05\ N\text{-}m\text{-}kg^{-1}\text{-}°K^{-1}$
Kinematic Viscosity	ν	$1.5723x10^{-4} ft^2\text{-}s^{-1}$	$1.4607x10^{-5} m^2\text{-}s^{-1}$

Table A.2 Standard Atmosphere at sea level

The earth's atmosphere extends up beyond 100 kilometers and is defined by the layers that have different characteristics. These are the troposphere (from sea level to 11km), the stratosphere (from 11km to 20 km), the mesosphere and the thermosphere. In our project we are only interested in the troposphere and stratosphere, and especially the range from sea level to 15km. The reason for our interest in this range is that most aircrafts fly at an altitude slightly above and below 11,000 meters.

2.1 Standard Atmosphere conditions below 11km

Below 11km, the standard atmosphere temperature decreases linearly with a lapse rate of –6.5degree K/km. The temperature T, pressure P, and density ρ may be calculated by the following equations [11],

$$T = T_0 + 6.5(\frac{Alt}{1000}) = 288.15 + 6.5(\frac{Alt}{1000})(^\circ K)$$

$$P = P_0(\frac{T}{T_0})^{-\frac{1}{AR}} = 101325(\frac{T}{288.15})^{5.2563}(Pa)$$

$$\rho = \rho_0(\frac{T}{T_0})^{4.2563} = 1.225(\frac{T}{288.15})^{4.2563}(kg/m^3)$$

where: Alt is the aircraft radar altitude in meters.

2.2 Standard Atmosphere Conditions Above 11km

Above 11km, the temperature has a zero lapse rate and remains constant at 215.7 degree K. In this region the temperature T, the pressure P and the density ρ may be calculated by the following equations [11],

$$T = T_0 + 6.5(\frac{10999.92}{1000}) = 215.7(^\circ K)$$

$$P = 26600\exp(\frac{Alt + 10999.92}{6341.882})(Pa)$$

$$\rho = 0.3639\exp(\frac{Alt + 10999.92}{6341.882})(kg/m^3)$$

where: Alt is the aircraft radar altitude in meters.

Appendix B. The Navigation Information in DTB

Navigation models	ARINC 429 Label	Word Name
GPS Model	076	GPS Altitude Word
	101	GPS HDOP Word 1
	102	GPS HDOP Word 2
	103	GPS Track Angle Word
	110	Latitude Word
	111	Longitude Word
	112	Ground Speed
	120	Fine Latitude Word
	121	Fine Longitude Word
	130	GPS HIL Word
	133	GPS VIL Word
	136	GPS VFOM Word
	150	UTC Word
	165	Vertical Velocity
	247	GPS HFOM Word
	260	Date Word
	273	GPS Sensor Status

ADC Model	203	Pressure Altitude
	204	Baro Corrected Altitude
	210	True Air Speed
IRS Model	072	Inertial Latitude
	073	Inertial Longitude
	126	Time In NAV
	270	IR Discrete Status #1
	271	IR Discrete Status #2
	310	Present Position Latitude
	311	Present Position Longitude
	312	Ground Speed
	313	Track Angel True
	314	True Heading
	317	Track Angel Mag
	320	Magnetic Heading
	321	Drift Angel
	322	Flight Path Angel
	324	Pitch Angel
	325	Roll Angel
	350	IR Maintenance Word
	361	Inertial Altitude
	365	Inertial Vertical Speed

	366	North-South Velocity
	367	East-West Velocity
AHRS Model	314	True Heading
	320	Magnetic Heading
DME Model	035	DME Frequency
	202	DME Distance
VOR Model	034	VOR/ILS Frequency
	222	VOR Bearing

Table B.1 the Word Names and Labels of the Navigation Information