The Development of a Dynamic Test Bed for Flight Management Systems

Nianfu Chui

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 2002

0-612-68451-2

ABSTRACT

The Development of a Dynamic Test Bed for
Flight Management Systems

Nianfu Chui

One of the important steps in Flight Management Systems (FMS) development is its testing in the dynamic environment for a specific aircraft type. While such testing is mostly performed in the actual aircraft, a more economical approach is to carry out the bulk of the testing using a laboratory-based simulated environment or test bed, for which the basis is a re-configurable, real-time flight dynamic model.

Such a dynamic test bed, when used for the test and evaluation of the FMS, has three issues identified in order to provide a simulated, real time, dynamic working environment for the FMS. The first issue is a real time software representation of the specific aircraft type to generate aircraft performance data and navigation information for the FMS, and to implement the FMS navigational commands. An on-the-shelf commercial flight simulator software package from Virtual Prototypes Inc, which has a re-configurable fixed-wing aircraft six-degree aerodynamic model and a complete set of simulated aircraft systems, was chosen as the baseline for this simulation software development. The baseline software package was enhanced by developing new control laws, automatic flight control modes, robust environment models and basic navigation facility models to satisfy the test bed as an engineering tool. The second system is the

interface system interfacing the FMS with the test bed. This involved the multiple processes interface software configuration and development for the interface cards and the simulation software. Finally, a graphical user interface is developed for the user to operate the test bed, including controlling the simulation and performing data acquisition.

There are three issues concerned with the development of the Dynamic Test Bed, which are the system's architecture/structure, hardware, and software. This thesis is mostly devoted to the architecture/structure and software issues.

ACKNOWLEDGEMENTS

The author would like to thank Dr. K. Demirli and Dr. J. V. Svoboda for their guidance and support in completing this thesis. The author would also like to thank CMC Electronics and their staff for their continuous close collaboration and support on this project. All kinds of helps and discussions from my teammates, my friends are very much appreciated. Special thanks go to my wife, for her understanding and help throughout this work.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF VARIABLES

ψ	Euler's angles for the yaw
θ	Euler's angles for the pitch
ϕ	Euler's angles for the roll
α	Ideal acceleration rate
M	Mach number
γ	Air constant
P_o	Sea level pressure
ρ_o	Sea level air density
q_c	P_{Total}-P
θ_c	Pitch command
P	Aircraft ambient pressure in Pa
T	Aircraft ambient temperature in °K
P_{MSL}	MSL Pressure in Pa
T_{MSL}	MSL Temperature in °K
Alt	Aircraft radar altitude in meter
IAS_t	Target Indicated Air Speed equivalent to the target Mach number.

IAS_c Current Indicated Air Speed of the aircraft.

$Mach_t$ Target Mach number of the aircraft.

$Mach_c$ Current Mach number of the aircraft.

LIST OF ACRONYMS AND ABBREVIATIONS

ADC	Air Data Computer
AFCS	Automatic Flight Control Systems
AHRS	Attitude and Heading Reference System
ARINC	Aviation 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
FPS	Global Positioning System
GUI	Graphical User Interface
HIL	Horizontal Integrity Limit
ICAO	International Civil Aviation Organization
IRS	Inertial Reference System
IAS	Indicated Air Speed
LNAV	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
SID	Standard Instrument Departure

SSM	Sign Status Matrix
STAR	Standard Terminal Arrival Routes
TAS	True Airspeed
TCP/IP	Transmission Control Protocol/Internet Protocol
UML	Unified Modeling Language
VHF	Very High Frequency
VLF	Very Low Frequency
VNAV	Vertical Navigation
VOR	VHF Omnidirectional Range
VS	Vertical Speed

1 INTRODUCTION

Many short range and long range navigation systems have been developed so far, including dead reckoning, inertial reference system (IRS), Very High Frequency Omnidirectional Range (VOR), Distance Measuring Equipment (DME), VOR/DME, Non-directional Radio Beacon (NDB), and Global Positioning System (GPS) which is the latest navigation technology [1]. However, it had become clear that no single navigation system was perfect all the time and that the key to accurate long-range navigation is to be able to take advantages of the strengths of each system as appropriate to the particular situation. It had also become clear that it is possible, with proper navigation system management, to achieve an overall level of accuracy that was greater than the accuracy of the single best system at any given moment.

A super-system is needed, a system that consolidates and manages each of the individual systems into what appeared to be a single long-range system that was at least as accurate as the best individual system at any given moment. Such systems first appeared around 1982 and were called Navigation Management Systems (NMS) [2]. Later, on the Boeing 757/767 and Airbus A310, data from a flight performance computer, the digital equivalent of the charts and graphs in the aircraft flight manual, was integrated into the navigation management system, and the system that resulted from that combination was a Flight Management System (FMS) [2]. Flight management systems also combine navigational data with pre-defined flight plan and stored aircraft performance and aerodynamic data to determine an optimized flight profile in both the lateral and vertical profiles, known as the lateral navigation (LNAV) and vertical navigation (VNAV). By combining performance data with navigational data (aircraft position, indicated airspeed, ground speed, outside air temperature, and the like), the FMS can continuously recalculate the roll, pitch and power settings necessary to closely follow the flight plan and to achieve the programmed result entered by the pilot: minimum time en-route, minimum fuel, minimum cost, maximum endurance, or constant Mach number [2].

Flight management systems have to be separately developed for each aircraft type and power plant for which they are certified. One of the important steps in FMS development is its testing in the dynamic environment for a specific aircraft type. While such testing is mostly performed in the actual aircraft, a more economical and effective approach would be to carry out the bulk of the testing using a laboratory-based simulated environment or test bed, for which the basis is a re-configurable, real-time flight dynamic model. This explains the reason for this work: the development of a dynamic test bed for flight management systems.

1.1 Flight Management Systems (FMS)

It is necessary to know the general rules and functionality of the FMS in order to understand the necessity and development of the dynamic test bed. The explanation here does not refer to any specific FMS type, but as a general concept.

One of the primary purposes of the FMS is to manage navigation sensors to produce a composite position. Using the composite position, along with flight planning capabilities, the FMS can control navigation, performance, and guidance work throughout the flight. General conceptual FMS is introduced in terms of its components, data resources and functionality [3].

1.1.1 Components of the FMS

The FMS consists of a control and display unit (CDU) or a multifunction control and display unit (MCDU), and a flight management unit (FMU). A data loader is associated with a FMS to load navigation and aircraft data into the FMU. Figure 1 illustrates a picture of a typical FMS [4]. However, some FMU and CDU are integrated into a single unit. Shown in the figure also include a typical global positioning system (GPS) antenna.

Figure 1 Typical FMU, MCDU and GPS Antenna

1.1.1.1 Control Display Unit

The control display unit is the pilot interface with the FMU. It consists of a keyboard, a Cathode Ray Tube (CRT) display, and the electronics required to communicate with the FMU. The keyboard consists of line select keys (both sides of the CRT display), alphanumeric keys, function keys, annunciators, and a bright/dim control. Shown in the middle of Figure 1 is a typical multifunction control display unit.

1.1.1.2 Flight Management Unit

The Flight Management Unit contains the central processing unit, navigation sensor interface, flight guidance interface and the navigation database. It can be built as a stand-alone unit, or combined with the CDU into one box.

The FMU has two primary functions and multiple secondary functions. The primary functions are flight planning and navigation. These functions work with the associated guidance in both the lateral and vertical axes. The connected navigation sensors and stored databases, which are described later, are essential to these functions. Using the available sensors, the FMU develops an FMS position based on a blend or mixed sensor inputs. Based on the position and the flight plan, the FMU generates information for display on the CDU and the aircraft electronic flight instrumentation systems (EFIS) [5], and/or coupled to the aircraft automatic flight control systems (AFCS) [6] to guide the aircraft. The secondary functions of the FMU include fuel management, time estimates, computing optimum altitude and speed target, automatically tuning navaids, and so on.

1.1.1.3 Data Loader

The data loader is a floppy disk drive unit which is designed mainly to update the FMS navigation database. The data loader can be used to transfer any of the following information:

- Navigation data base
- Aircraft data base
- Custom data base
- Flight plans
- Maintenance data.

1.1.2 Data Sources of the FMS

The FMS performs all calculations based on all sorts of data available. The data sources include the navigation information provided by the connected navigation sensors and its data bases which include the navigation data base, custom data base, and aircraft data base. Figure 2 shows the FMS system block diagram. Some FMS may have more inputs than shown in the figure. The data bases are not shown in the figure. They are loaded into the FMS through the data loader.

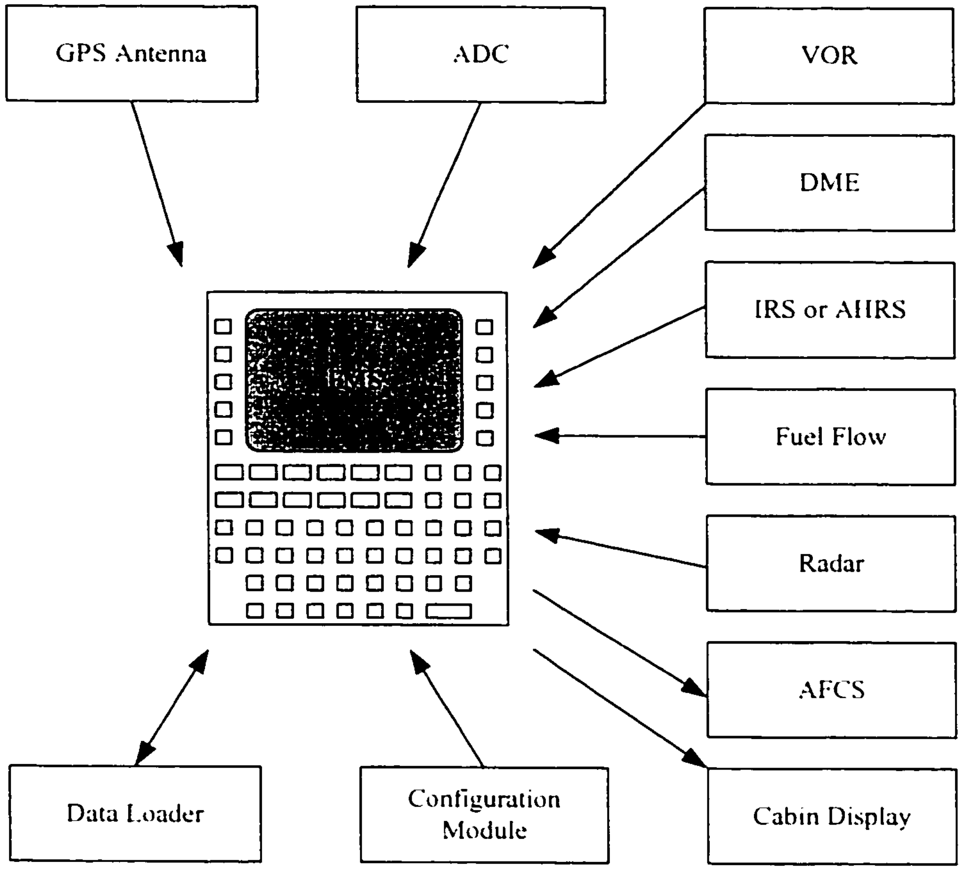

Figure 2 FMS System Block Diagram

1.1.2.1 Navigation Sensors

The FMS connects to a variety of short range and long range navigation sensors [5]. The primary short range sensors are VOR/DME [7] and DME/DME. Long range sensors include GPS [8] [9], IRS, and, for some FMS, very low frequency (VLF)/omega. Among these sensors, GPS is the highest priority sensor. Failure of one sensor results in the automatic switching to the next high priority sensor.

1.1.2.2 Navigation Data Base

The FMS retrieves information from the navigation data base [10] about waypoints and procedures used in flight planning and to tune navigation radio frequency for position calculation. The navigation data base can be tailored for different areas of coverage, and must be updated periodically.

The navigation data base contains the following information:

- Navaids
- Airports
- Runways
- Airways (high and low)
- SIDs (Standard Instrument Departure) and STARs (Standard Terminal Arrival Routes)
- Approaches
- Named Intersections
- Outer Markers.

The navigation data base is loaded into the FMU through data loader and can not be changed by the pilot.

1.1.2.3 Custom Data Base

The custom data base [10] contains pilot defined waypoints and stored flight plans. This data base is loaded into the FMU through data loader and can be modified by the pilot. Using the custom data base, the pilot can customize the FMS by defining waypoints and storing flight plans.

1.1.2.4 Aircraft Data Base

The aircraft data base contains aircraft information. This data base is specific to certain aircraft type and some information in it is even unique to certain tail number. The aircraft data base must be loaded through data loader into FMU to perform aircraft performance calculations, which could include fuel management, optimum altitude, automatic speed targets, etc.

1.1.3 Interface with Other Systems

The FMS can interface with various aircraft systems through discrete, analog [11], and different serial communication protocols [12]. The serial protocol used in this thesis will be the ARINC 429 bus [13], which is the most important avionics communication bus.

1.1.4 Functionality of the FMS

The primary functions of the flight management system are flight planning and navigation. Secondary functions include fuel management, time estimates for the flight, automatic radio tuning, etc. These are briefly described next.

1.1.4.1 Flight Planning

Flight plan [14] is a sequence of waypoints from the departure airport to the arrival airport, including the cruise altitude among waypoints. Before flight, the pilot works out and enters the flight plan into the FMS through the CDU. The pilot may also modify this predefined flight plan during flight by deleting/adding waypoints or changing cruise altitude. During flight, the flight planning function computes the active flight plan with both lateral and vertical definition.

1.1.4.2 Navigation

Navigation is another primary function of the FMS. During the flight, the FMS constantly verifies its position using navigation sources including Air Data Computers (ADC), GPS, radio beacons which include VHF Omni-directional Radio Range (VOR) and/or Distance Measurement Equipment (DME) [7] [9], and Inertial Reference System (IRS). Then, according to the established flight plan, the FMS computes the optimal vertical profile to meet all vertical and speed constraints; and computes the lateral profile including cross track errors, desired tracks, distance and time to next waypoint, etc, in order to closely follow the lateral flight plan.

The FMS navigation functions include lateral navigation (FMS LNAV) and vertical navigation (FMS VNAV) [15]. They are described next.

1.1.4.2.1 Lateral Navigation (LNAV)

The FMS LNAV is used to compute and send commands to the flight guidance computer to laterally steer the aircraft. LNAV is engaged (meaning coupled with the autopilot) on the flight guidance panel, and is available for all phases of flight. Requirements for engaging LNAV are that the FMS must be selected as the navigation source by the pilot. To use FMS LNAV, the autopilot and lateral navigation automatic flight control mode is needed.

1.1.4.2.2 Vertical Navigation (VNAV)

With GPS installed, the FMS can be used for non-precision approach. The glide slope will be tracked by using the vertical deviation calculated by the FMS from the 3-deminsional aircraft position and the physical position of the glide slope. The VNAV is not included in this thesis.

1.1.4.3 Other Functions

In addition to its primary functions, the FMS provides some other useful functions. Different FMS might include less or more than the following secondary functions:

- Navigation displays on the EFIS
- Standard instrument departure

- Automatic radio frequency tuning
- Automatic waypoint sequencing
- GPS instrument approach
- Fuel management
- Time estimate for the flight
- Automatic speed targets for each phase of flight
- Estimates optimum altitude, cruise modes, step climbs

1.1.5 Development and Testing of the FMS

As a multiple function navigation system, the advantages of the FMS are obvious. But as a sophisticated high-tech system, its development is by no means easy. One of the important steps in FMS development is its testing in the dynamic environment of the specific aircraft type. Such testing is mostly performed in the actual aircraft, which is very time-consuming and economically prohibitive. However, a more economical approach would be to carry out the bulk of the testing using a laboratory-based simulated environment or test bed, for which the basis is a re-configurable, real-time flight dynamic model [16]. This way, only a limited amount of costly, actual flight testing would be necessary and would be allocated primarily towards certification-related tasks. Such an approach would be particularly attractive for the retrofitting of FMS's, where flight testing proves to be a difficult and expensive process. This results in a need for the dynamic test bed as discussed next.

1.2 Dynamic Test Bed (DTB)

As discussed in previous section, the purpose of the test bed is to test and evaluate the dynamic performance of the FMS in order to assist its development. This has answered the purpose of the DTB. The FMS is designed for operating on a real aircraft. An identical working environment has to be provided for the FMS being tested. What functions and components should the test bed include? How are they organized in order to achieve this? These issues have to be evaluated before proceeding into development.

1.2.1 Components of the DTB

Figure 3 shows the DTB components. It includes a graphical user interface, a flight simulation software system, and an FMS interface system. These components are connected through Ethernet links. The FMS interface system is connected to the FMS hardware through ARINC 429, analog and discrete connections, as shown in the figure.

This section describe the basic functionality of each component as well as the connections.

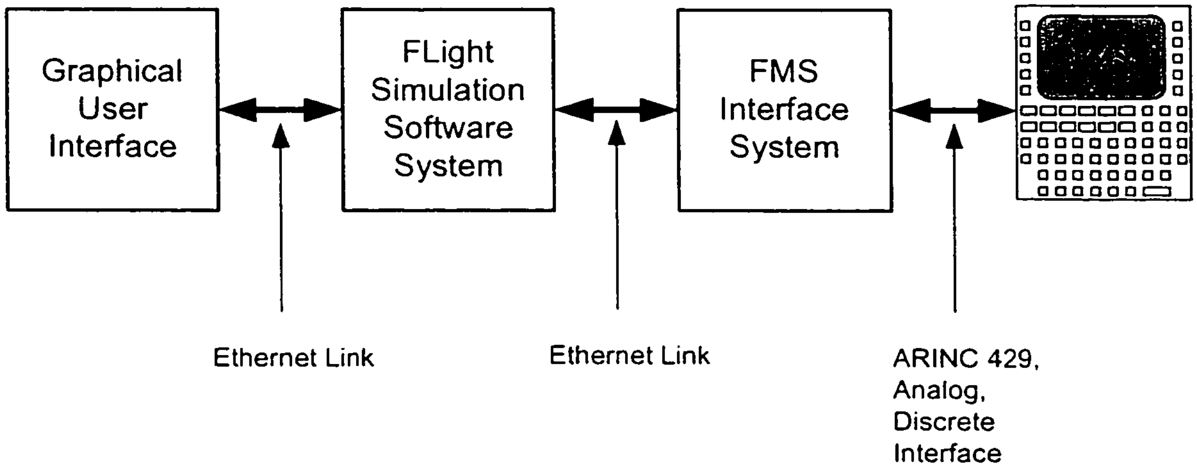

Figure 3 DTB Components

1.2.1.1 Flight Simulation Software System

The flight simulation software is used to simulate the dynamic performance of the specific aircraft type in which the FMS is working. In order to shorten the development cycle, a commercial flight simulation software package, FLSIM v7.0 from Virtual Prototypes Inc (VPI) [17], is chosen as the core for the flight simulation software development.

The flight simulation software is organized with features only necessary to fulfill the test of the FMS. Therefore, it first contains a re-configurable aircraft six-degree aerodynamic model, which comes with the FLSIM v7.0. Aircraft systems, such as aerodynamic, navigation, electric, and hydraulic systems are all simulated in the FLSIM v7.0. To stimulate the FMS, navigation models are needed to collect navigation sensors information for the FMS. Some navigation sensors are listed in Figure 2. In order for the aircraft to be driven by the FMS, autopilot and necessary automatic flight control modes are developed to control the aircraft to implement the FMS lateral and vertical steering commands. These modes do not necessarily have to be identical to the simulated aircraft type. For example, it could include heading mode, FMS LNAV mode, vertical speed mode, flight level change mode, altitude hold mode, and FMS VNAV mode. An auto take-off mode is also needed to make a smooth takeoff. Flight environment is also simulated to evaluate the dynamic performance of the FMS under different atmospheric conditions and environment. This includes a multiple level window model, an environment model with varying mean sea level (MSL) pressure and temperature [18].

1.2.1.2 FMS Interface System

The FMS interface system includes different hardware interface cards and supporting interface software. Different FMS may have different types of interfaces, and sometimes even the same FMS can be configured to use different types of interfaces. For the tested FMS, the interface cards include ARINC 429, analog and discrete types. ARINC 429 data and analog/discrete signals are transferred through these cards between the DTB the FMS. The interface software system is developed to support the I/O operations of these cards. It also transmits and receives data to and from the simulation software; stores, sorts and distributes received data to the cards; and collects data from the cards.

1.2.1.3 User Interface System

The user interface is a software graphical user interface (GUI). The user has full control of the DTB through this GUI. First, it is used by the user to control the flight of the aircraft by providing basic simulated flight control features. These include the autopilot mode selection, flight parameter settings and aircraft control surface settings. Flight freeze and reposition functions are provided. Simulated atmospheric environment can be configured and multiple level wind model can be constructed through the GUI. The GUI provides control features for data acquisition. User can monitor any labels selected from a list by plotting them on the screen in real-time, or collect the data and save in a file for post analysis.

1.2.1.4 Ethernet Links

Ethernet TCP/IP protocol [19] is selected to transmit data in real-time between software applications. The first Ethernet link is between the simulation software and the GUI. Since the simulation software and the GUI is running on the same PC, another way of communication between them could be through shared memory. But the reasons that the Ethernet link is selected are that Ethernet link has the flexibility of running the GUI on another PC by just changing the IP address, and that the user can add or remove data on the Ethernet link without worrying about the memory re-allocation, which is the case of using shared memory. The second Ethernet link is established between the flight simulation software and the interface software, which is hosted by the I/O PC. See Section 1.2.2.1 for the DTB computer arrangement.

The TCP/IP supports error-free transmission of large blocks of data. The receiving program using this protocol is able to reassemble the bytes in the exact sequence in which they are transmitted. This feature is essential to the accurate data transmission of the test bed.

1.2.2 DTB Hardware System Structure

The components of the DTB have been defined from the functional point of view in previous section. How are the hardware and software systems organized together to form a complete system and work properly? How to achieve real time data transmission between the aircraft simulation software and the FMS, which is very critical and essential to the success of the test? These questions will be answered in this section.

This section presents the overall DTB system structure. As shown in Figure 4, the DTB system is composed of a simulation PC, an I/O PC, an FMS system including one MCDU and at least one FMU. The FMS system is connected to the I/O PC through a dedicatedly designed breakout box. Their development will be described in Chapters 3 and 4.

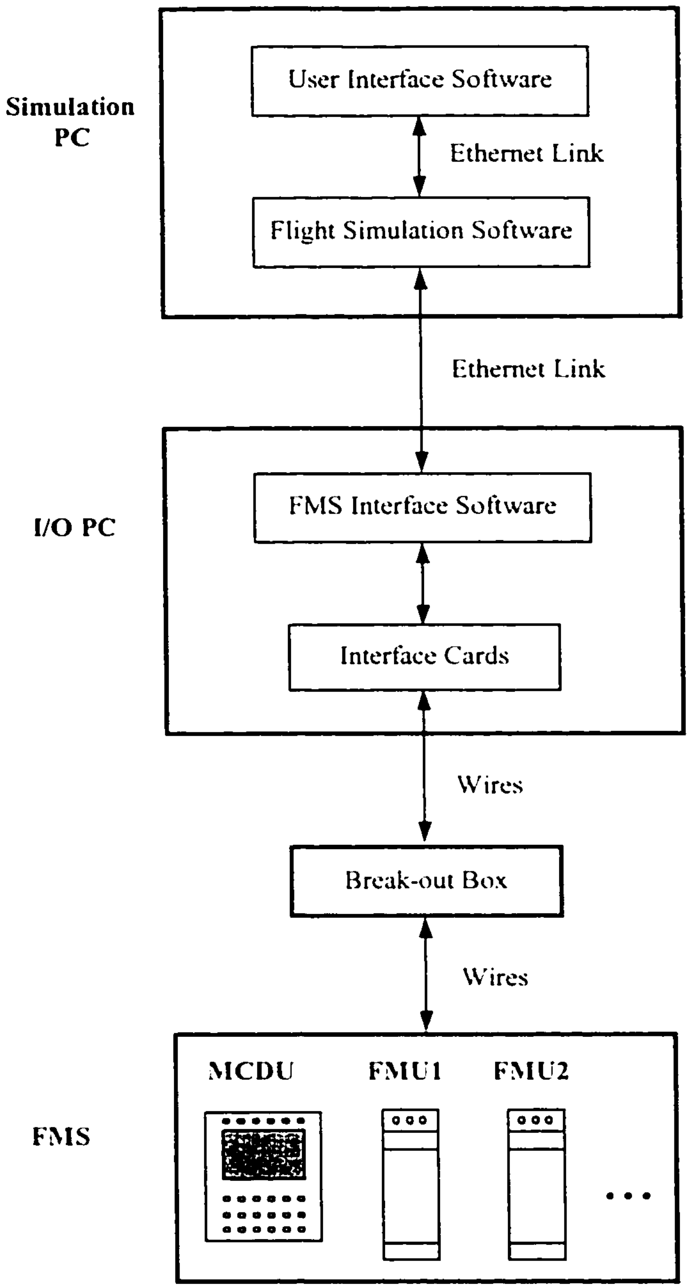

Figure 4 DTB Overall Structure

1.2.2.1 Computer Platform and Software Arrangement

Since one PC is not convenient to handle the simulation software, the FMS interface software and the GUI at the same time for real time simulation and high speed data transmission, the working platform of the test bed consists of two PCs. One PC, called the simulation PC, hosts the simulation software and the GUI. The other PC, which is called the interface PC, hosts the interface software and the interface cards [20].

1.2.2.2 Ethernet Link

Real-time simulation and real-time data transmission are critical issues for the DTB due to the working environment of the FMS. Therefore, the high speed Ethernet is selected for the communication between the two PCs, and between the GUI and the simulation software, as shown in Figure 4. Each PC has one Ethernet card installed.

1.2.2.3 Breakout Box

A breakout box is designed to provide power supply to the FMS. ARINC 429 buses and analog/digital buses are connected from the cards to the FMS through this box. However, the design and building of this box is not the work of the author, and the details of this work will not be discussed further.

2 SOFTWARE SYSTEM CONFIGURATION

The dynamic test bed consists of a hardware system and a software system. However, this thesis is mainly focused on the software system development. Therefore, the hardware system configuration is not discussed.

As described in the last chapter, the software system of the dynamic test bed consists of the flight simulation software, the FMS interface software, the graphical user interface software, and the Ethernet TCP/IP communication software. This chapter presents a closer look at the first three software. The Ethernet TCP/IP communication software serves as an assistant software and its configuration is simple, so only its development will be described in next chapter.

2.1 Flight Simulation Software

The discussion starts from the flight simulation software because it is the very basic functional component of the DTB. This software implements a six-degree aircraft aerodynamic model in order to simulate the aircraft dynamic performance. It provides a simulated working environment for the FMS to perform its functions. This software takes inputs from the user and the FMS, generates real time flight information for the GUI and the FMS [21].

2.1.1 Flight Simulator

The flight simulator contains all sub-modules involved in the simulation calculation and a dispatcher which is responsible for the execution of all the modules. Different module has different predefined execution band and is invoked by the dispatcher according to its band.

2.1.2 Aircraft Model

This model contains the configurations of the simulated aircraft type, which include the dimensions, weight, and aerodynamic coefficients for the airframe, engine, control surfaces (i.e. rudders and flaps), etc. It is re-configurable and thus different models of aircraft can be simulated on the DTB. This model is pre-configured and downloaded when running the simulation.

2.1.3 Autopilot and Automatic Flight Control Modes

An autopilot is used to fly the aircraft automatically by maintaining the current pitch angle and heading of the aircraft [22] [23]. This ensures accurate implementation of the FMS steering commands and user commands. The autopilot is available in the FLSIM v7.0.

The following automatic flight control modes are necessary to fly the aircraft in this thesis by following user commands and FMS commands:

- A heading mode to intercept and maintain a heading reference

- A LNAV mode to capture and track the predefined lateral route profile using FMS lateral steering command
- An altitude hold mode to hold a target altitude
- An altitude pre-select mode to automatically capture, level off and hold a pre-selected altitude
- A vertical speed mode to automatically maintain the pre-selected vertical speed reference
- A flight level change mode to fly the aircraft toward a new altitude while maintaining a pre-selected airspeed with target thrust

2.1.4 Aircraft Secondary Surface Control

There are three surface control models: the flaps control, the spoilers control, and the gear control. These are used to control the deflection or status of the aircraft flaps, spoilers and landing gears [24].

2.1.5 Navigation Models

The primary function of the FMS is navigation. Information from navigation sensors is the most essential inputs to the FMS. Navigation facilities are simulated by developing navigation models taking navigational information from the simulation. These navigational models include the following:

- GPS Model

GPS is the highest priority navigation sensor of the FMS. This model takes the aircraft position and velocity from the simulation results and transforms them into GPS ARINC 743a format words for transmission to the interface computer. It also takes simulated destination waypoint horizontal integrity limit (HIL) from the GUI and transforms it into GPS ARINC format for transmission to the interface PC.

- ADC Model

The ADC model takes aircraft speed and altitude from the simulator, and formats as ADC ARINC 429 format words for transmission to the interface computer.

- AHRS Model

The AHRS takes the aircraft heading from the simulator and formats as AHRS ARINC 429 format words for transmission to the interface computer.

- IRS Model

IRS is a long-range navigation sensor. This model takes simulated IRS information from the simulation and transmits in ARINC format to the interface PC.

- VOR/DME Model

The VOR/DME are primary radio navigation facilities. This model takes inputs from the simulation and transmits ARINC 429 words in standard DME format. Simulation of this model needs a navigation database which contains the information of the VOR/DME stations.

2.1.6 Environment Models

Environment changes can affect the aircraft performance. Under these conditions, the FMS should still be able to provide correct navigation to the aircraft. These changes need to be simulated to test the dynamic performance of the FMS.

Environment models include a multiple level wind model, an ambient condition model for varying the MSL pressure and temperature.

2.1.7 Discrete Mapping Model

Minor information of the FMS can be transferred in discrete format. This model is used to map the discrete signals transmitted between the simulation and the FMS.

Figure 5 illustrates the above-described components and their interconnections. The arrows represent the data flow directions. As seen from the figure, the flight simulation software is the data processing center of the system. Navigation models such as GPS and ADC, receive simulation data and send their own outputs after performing some processing. Other models shown in the figure receive information from the GUI and the I/O PC and feed into the simulation software for execution. External communications with the GUI and interface software are through Ethernet using TCP/IP, and are mapped into two separate process paths which will be explained in Chapters 3 and 4.

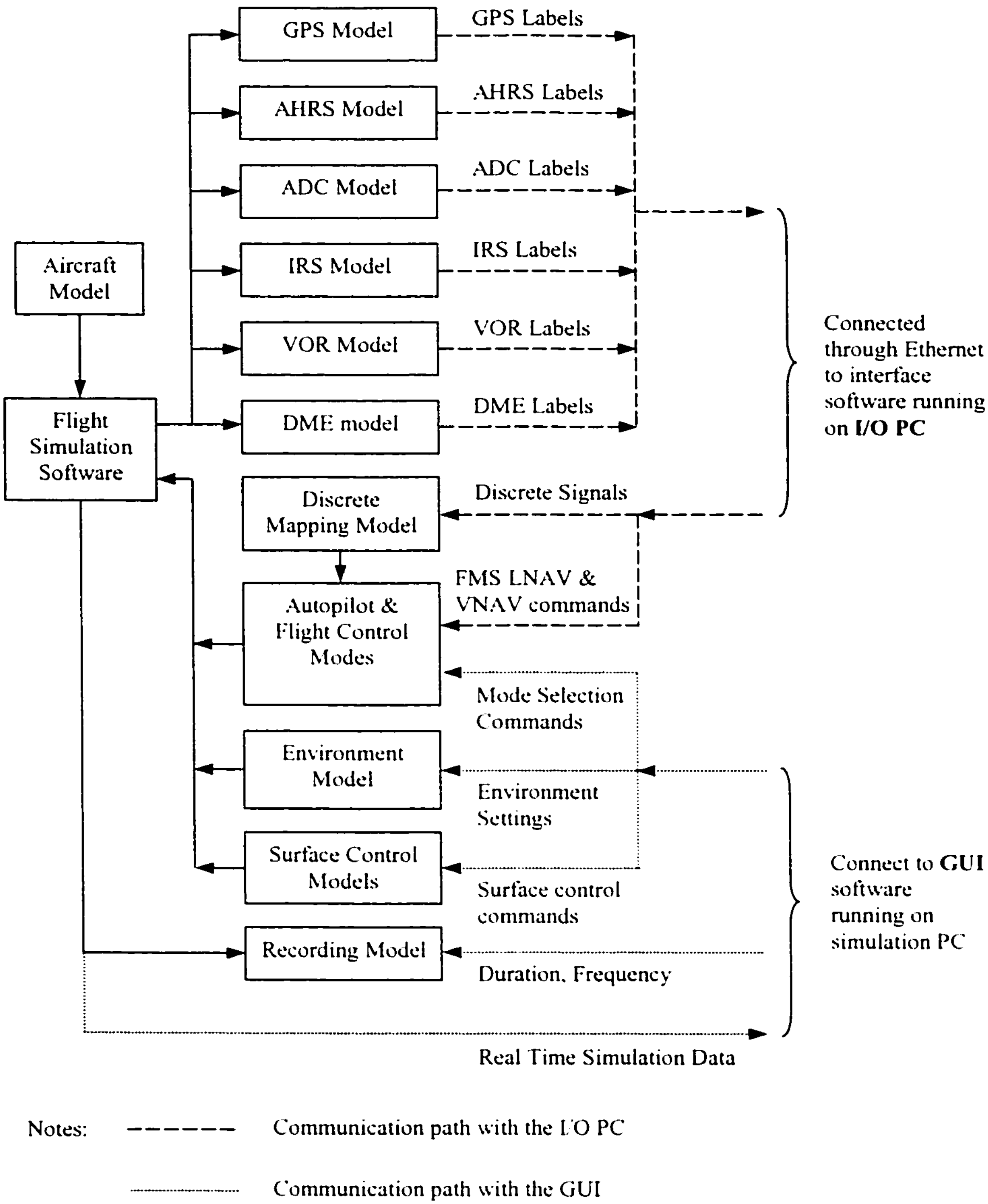

Figure 5 Flight Simulation Software Configuration

2.2 Hardware and Software Interface with FMS

The FMS interface is responsible for all data transmission between the simulation program and the FMS. This interface system mainly includes an Ethernet link from the simulation PC to the interface PC, an ARINC 429 serial interface and a discrete interface between the interface PC and the FMS. A dedicated software is developed to orchestrate and control these three interfaces for real time communication. This section briefly describes these interfaces and the control software. Their development and operation are discussed in detail in Chapter 3.

2.2.1 ARINC 429 Interface

The ARINC 429 is the primary interface with the FMS. Most of FMS inputs and outputs are transmitted in the A429 format. The ARINC 429 standard is the common protocol for data interchange between avionics systems such as the FMS. It specifies a serial system for data interchange. The FMS uses ARINC 429 as the primary data transmission media with other systems. In this thesis, a commercial ARINC 429 card consisting of 16 channels was selected as the main interface with the FMS and a software driver was developed for the card to perform I/O operations.

2.2.2 Discrete Interface

In addition to ARINC 429, the FMS can also pass minor information in discrete format to other avionics systems. Therefore, the discrete interface should also be established between the test bed and the FMS as the secondary interface. Analog interface is also provided to serve as two purposes. First, the analog signals are used to assist the development of the DTB; second, it is provided as a provision for the possible future need. Like the ARINC 429 interface, this interface also includes hardware interface card and its software driver.

2.2.3 Ethernet Communication

The Ethernet TCP/IP is a high-speed communication protocol between different computers, and different applications running on the same computer. Two computers in this thesis are linked through Ethernet for bi-directional real time data transmission. As shown in Figure 3, the simulation program and the GUI, which both are running on the same PC, also exchange data through Ethernet link. This interface includes Ethernet cards installed on the simulation PC and the interface PC, as well as Ethernet communication software.

2.2.4 Double-Buffered Container

In the interface software, there are 4 worker threads. As shown in Figure 6, the output of one thread is the inputs of other one or two threads. Because these threads are running simultaneously, therefore when one thread is writing into its buffer, other thread may be reading data from the same memory address. This will cause errors of data. To avoid the conflicts, a double-buffered container is created. Two layers in the double-buffered container are the writeable layer and the readable layer. The writeable layer is used for the threads to write data. The readable layer is used for the threads to read data from it. The executive thread dumps data from the writeable layer into the readable layer after a thread finishes the writing in the writeable layer.

In Figure 6, different threads are indicated by different type of lines. As shown in the figure, the ARINC I/O thread and the Analog/Digital Input/Output thread all pick up simulation outputs from the readable layer and send to the interface cards, which are connected to the FMS through hardware buses. The TCP/IP I/O thread picks up A429 data and analog/discrete signals, which are originally from the FMS, from the readable layer. The software implementation will be discussed in Chapter 3.

2.2.5 Executive Software

The executive software is developed to control and orchestrate the operations of the Ethernet link, the ARINC 429 interface and the discrete interface for safe data exchange and real time data transmission. The interfaces are mapped in separate process paths, called threads, (ARINC 429 input and output are in separate threads, as described in Section 3.2.1) and are running at 30 Hz concurrently. Real time information are passed between the Ethernet interface and the ARINC 429 and discrete interfaces. To avoid runtime errors such as memory corruption, these threads must be carefully synchronized [25].

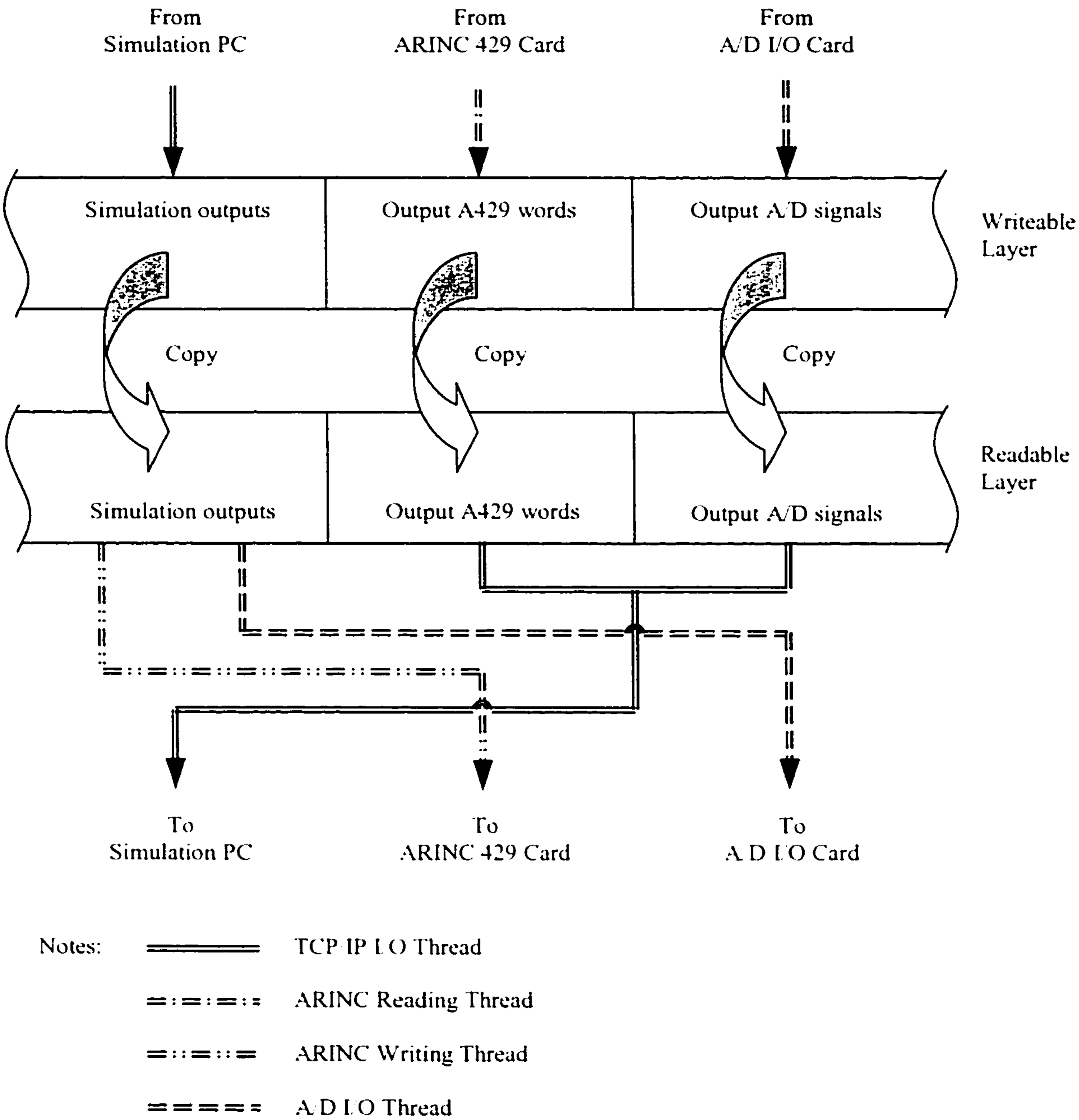

Figure 6 I/O Threads and Double-Buffered Container

2.3 Graphical User Interface Software

As proposed in Section 1.2.1, the GUI should have basic control and setting features, as well as data acquisition functions. In order for it to be user friendly, the graphical user interface is configured to have two dialog windows, which are the main dialog window and its child dialog window. A TCP/IP I/O process is incorporated into this software for the communication with the simulation software.

2.3.1 Main Dialog Window

This window is the primary interface with most of the control features and configuration functions.

2.3.1.1 Target Flight Parameter Setting

This area is used for the user to enter target flight parameter at runtime. Target parameters include target airspeed, target Mach number, target altitude, target thrust, target vertical speed and target heading. Dialog boxes should be used to enter the data.

2.3.1.2 Modes Control Panel

The modes control panel (MCP) is used for the user to select different automatic flight control modes. These modes include heading mode, LNAV mode, vertical speed hold mode, level change mode, altitude pre-select mode, altitude hold mode and VNAV mode. Press buttons are used to represent flight control modes. During the flight, the states of the buttons represent the engagement of the flight modes.

2.3.1.3 Aircraft Secondary Surface Control

This part of the GUI is used for the user to control the deflection of the aircraft flaps, spoilers, and retract or extend the landing gears. This is because sometimes these operations are necessary to control the aircraft flight. One example could be when the speed is very low, the flaps need to be deployed in some degrees to generate enough lift.

2.3.1.4 Reposition/Freeze/Unfreeze

The reposition interface allows the user to jump the aircraft from one position to another position in order to reduce the test time [26] [27] [28]. After reposition and right after the simulation is unfrozen, all systems of the aircraft are first re-calculated and refreshed by the core simulation process based on the flying situation before the reposition and the selected waypoint configuration (reposition three-dimensional aircraft position, airspeed and heading). The aircraft then continues its new flying with the same flight control as before the reposition. The freeze function is used to suspend the simulation at any time. The simulation can be resumed by using the unfreeze function. A data base containing waypoints information is created for the user to select a waypoint for reposition.

2.3.1.5 GPS PRAIM Simulation

In order to get into VNAV mode, the FMS makes GPS Predictive Receiver Autonomous Integrity Monitoring (PRAIM) request for the final approach fix (FAF) and missed approach point (MAP) [29]. To simulate the GPS system, the DTB sends the GPS HIL at Estimated Time of Arrival (ETA) of the destination waypoint to the FMS every time when receiving a PRAIM request from it. From this window the user sets a GPS HIL at ETA for the FMS.

2.3.1.6 Environment Configuration Interface

This window takes user inputs for the configuration of environment model, including the multiple level wind model, the MSL temperature model and MSL pressure model. These inputs are sent to the simulation software, and environment models will be updated there.

2.3.1.7 Data Acquisition Interface

As a useful engineering tool, the DTB should have the functionality of data acquisition. Through the data acquisition window, the user can select parameters for recording, plotting and printing. The user should be able select as many as four parameters from a name list for monitoring, plotting and printing. The source of the parameters could be from either the real-time simulation or a previously recorded file.

2.3.1.8 Runtime Flight Information Display

Runtime flight information, such as the indicated airspeed (IAS), vertical speed (VS), heading and altitude, are displayed on the screen for the user to monitoring the flight status. Dialog boxes should be used for this purpose.

2.3.2 'Child' Dialog Window

This dialog window is mainly used to plot data and to assist the user for runtime data analysis. Data are plotted with respect to simulation time. Each curve can be zoomed in or out. This window implements a dynamic scroll bar for the user to monitor data for as long as 6 hours. The printer function is associated with this window to print out the same curves on the window.

Three models are used to implement the above functions. The first is the runtime plot model. It receives commands from the plotting interface of the main window and plots real time simulation data. The second one is the plotting model for recorded data. The user can change scale for each curve. This model receives commands from the plotting interface of the main window and plots curve(s) on the screen. The data can be from the runtime simulation result or from a recorded data file. The third model is the printing model, which prints out the curves on the printer. All the three models accept zooming command.

The configuration is illustrated in Figure 7. Both windows communicate with the simulation program using TCP/IP protocol.

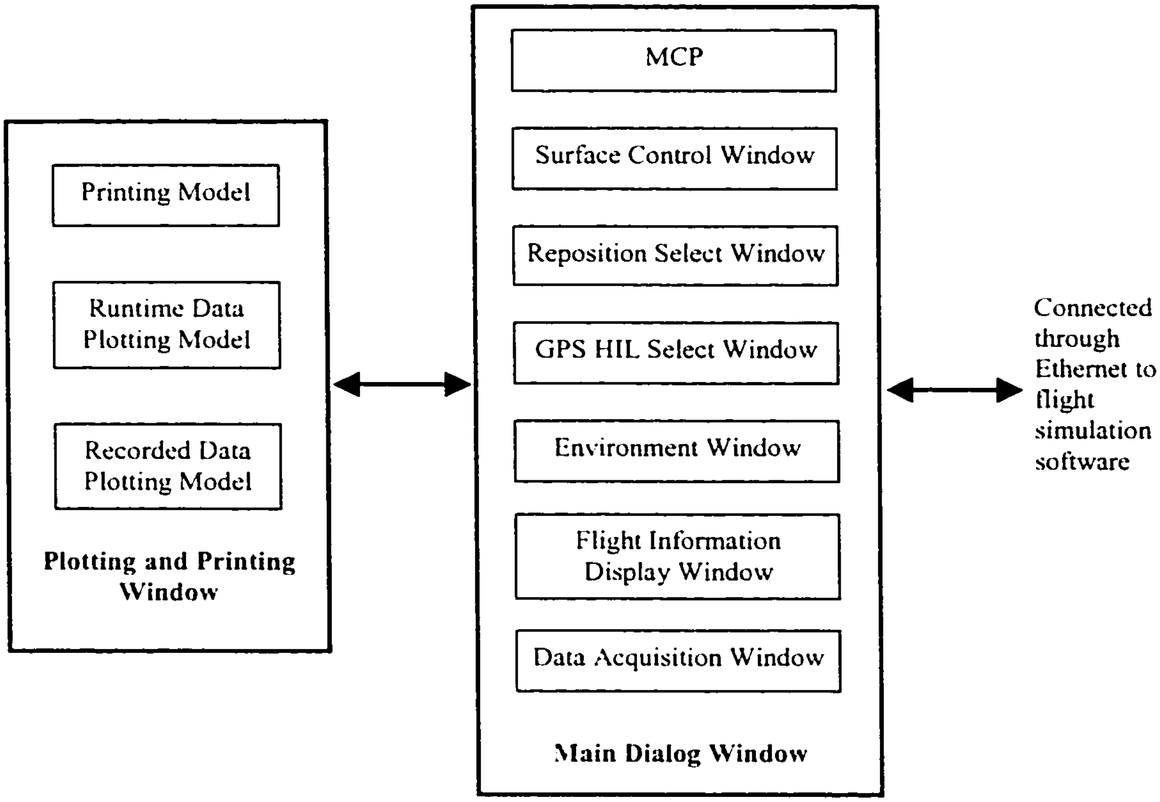

Figure 7 Graphical User Interface Software Configuration

3 DTB INTERFACE SOFTWARE DEVELOPMENT

This chapter presents the development of the DTB interface software, which includes the FMS interface software and the graphical user interface software. The DTB system software is developed using the object-oriented computer language Microsoft Visual C++, and running under Windows NT 4.0. Both interface software take full advantage of multithreaded programming technology [25] [30].

3.1 Ethernet Communication Software Development

The Ethernet communication software is developed to implement high-speed network communication between different applications of the DTB. This software has the following functionality:

- Define and create a Windows Socket

A socket is a bi-directional communication endpoint — an object through which a Windows Sockets application sends or receives packets of data across a network. It is the combination of an IP address and a port number. Streams are guaranteed to be delivered and to be correctly sequenced (delivered in the sending order) and unduplicated through the socket.

- Bind a Socket

Before it can accept connection requests, a listening server socket must select a port number and make it known to Windows Sockets by calling member function *Bind*. *Bind* establishes the local association (host address/port number) of the socket.

- Listen to the Client

The server listens for incoming connection requests. To accept connections, the socket is first created with *Create*, a backlog for incoming connections is specified with *Listen*, and then the connections are accepted with *Accept*.

- Accept the Connection

The server calls the member function *Accept* to accept a connection on a socket. If no pending connections are present, *Accept* returns zero and the original socket remains open and listening.

- Connect to the Server

The client calls the member function *Connect* to establish a connection to an unconnected socket. When this socket call completes successfully, the socket is ready to send/receive data.

- Send a Stream

The member function *Send* sends data stored in a specified buffer on a connected socket.

- Receive a Stream

The *Receive* member function receives data from a connected socket and stores the data in a buffer.

- Close the Socket

When the communication terminates, the member function *Cleanup* is called on both the server end and the client end to close the sockets.

- Communication Exception Handling

When running the DTB, one application may terminate the connection and try to connect again. In this case, a robust Ethernet connection is necessary. Process exception handling is implemented to ensure robust connection when either end of the socket terminates the connection.

3.2 FMS Interface System Development

The FMS interface system includes the ARINC 429 bus, analog/digital bus and Ethernet communication. The design of the FMS interface software uses the single-processor multithreaded programming technology to achieve real time data transmission. This section discusses the development and operation of the three interface systems.

3.2.1 ARINC 429 Interface

ARINC 429, commonly known as "ARINC Specification 429 Digital Information Transfer System, Mark 33", is the basis for digital buses in modern civil aircraft [31]. It defines the Air Transport Industry's standard for the transfer of digital data between avionics systems. The card used in this thesis is the ARINC 429 PC16 from SBS Avionics Technologies, which features sixteen ARINC 429 channels. This section starts from the general format of the ARINC 429 word and then describes the 429 words processing, channel configuration, and the I/O operations of the card. These tasks are handled by the ARINC 429 interface software, which is developed based on the SBS ARINC 429 library routines coming with the card.

3.2.1.1 General ARINC 429 Word

Communications on 429 buses use 32-bit words with odd parity. Typically, a data set is composed of one word and consists of either Binary (BNR), Binary Coded Decimal (BCD) or alphanumeric data encoded per ISO Alphabet No.5. Table 1 shows the organization of the 32 bit ARINC word.

Table 1 32 Bit ARINC Word

<table>
<tr><td>P</td><td colspan="2">SSM</td><td colspan="19">Data-19 Bits</td><td colspan="2">SDI</td><td colspan="8">8 Bit Octal Label</td></tr>
<tr><td>32</td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td></tr>
<tr><td colspan="3">MSB</td><td colspan="26">32 Bit ARINC 429 Word</td><td colspan="3">LSB</td></tr>
</table>

Bits 30 and 31 are typically assigned to the sign status matrix (SSM), which is used to report the equipment conditions or the sign (+, -, north, south, east, west, etc.). The source/destination identifier (SDI) is optional and when used, occupies bits 9 and 10 of the ARINC word. In actual use, the basic structure of the ARINC 429 word is very flexible. The only two parts of the word needing to stay intact are the Information Identifier (label) and the parity bit.

ARINC 429 words are transmitted using certain transmitting protocol. This protocol is implemented by the firmware coming with the card.

3.2.1.2 ARINC 429 Words Encoding and Decoding

Information written or read to or from the ARINC 429 card is in ARINC 429 word format. However, data are processed in the simulation program in engineering format. Therefore, data are encoded into ARINC 429 format before being written into the ARINC 429 card, and are decoded from ARINC 429 format into engineering values after being read from the card.

The encoding of an ARINC 429 word includes the conversion and assembling of the label, data bits, SDI, SSM, and parity setting. All these information are assembled into a 32 bit word as per the ARINC 429 word format. Since the 429 words are transmitted between the DTB and the FMS, which implies that the installation is single source-destination arrangement, the SDI bits are encoded as 00. The SSM bits are always encoded as normal operation for all ARINC 429 words. Engineering values are assembled into the data bits depending on the type of data used by the word (BCD, binary, or discrete). Label encoding is a fixed procedure for each label. The last step of the encoding is the parity setting to ensure odd parity of the final assembled word.

The decoding of an ARINC 429 word involves extracting data from the word and, for some labels, checking the SSM bits.

3.2.1.3 ARINC 429 Buses Configuration and Initialization

The ARINC bus is a unidirectional simplex bus on which there is only one transmitter but one or more receivers (up to a maximum of 20). In this thesis, there is only one receiver in both directions of the data transmission, i.e. the FMS or the DTB.

Depending on the priority and the amount, data can be transferred on the ARINC 429 bus at either low speed of 12.5 to 14.5kHz or high speed of 100kHz (+/-1%). A low-speed bus is used for general-purpose, low-criticality applications, and a high-speed bus is used for transmitting large quantities of data or flight critical information. The configuration of the ARINC 429 bus is shown in Appendix A. The channels are initialized automatically when running the program. The configuration data shown in the table are read from a predefined file.

3.2.1.4 ARINC 429 I/O Threads

The ARINC 429 interface involves transmission of large amount of ARINC 429 words. The transmission of data to or from the FMS has to be real time as in a real aircraft environment. To achieve this performance, two threads are allocated to the ARINC I/O operation, one for receiving and the other for transmitting. The operations of the receiving and transmitting thread are illustrated respectively in Figure 8 and 9.

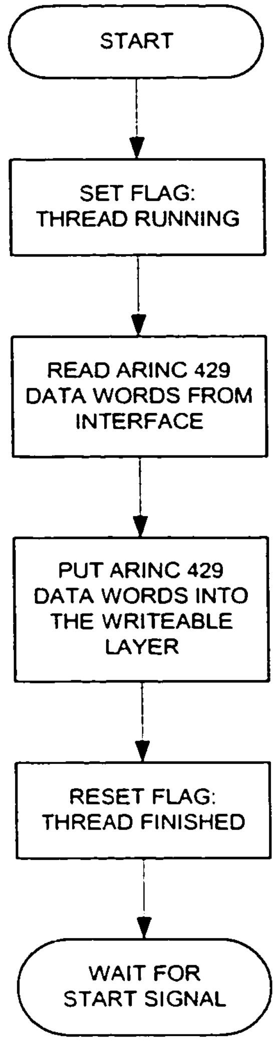

Figure 8 ARINC 429 Receier Thread

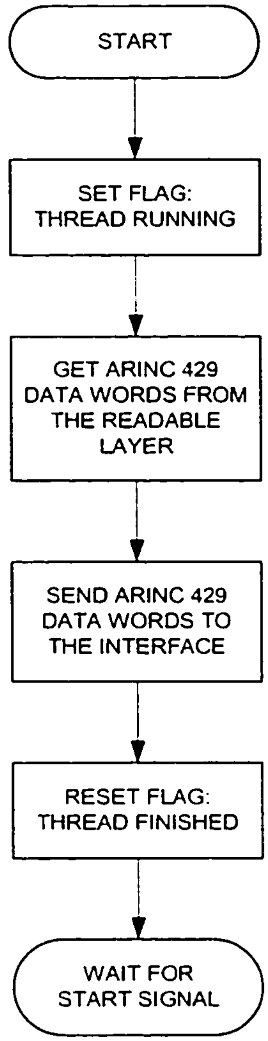

Figure 9 ARINC 429 Transmitter Thread

Both threads shown in the above two figures are started by the executive thread, which is explained in Section 3.2.5. When tasks are finished, the ARINC I/O threads wait for the start signal from the executive thread for next iteration. During the waiting time, these threads consume very little CPU usage.

Different ARINC 429 words may have different transmission repeat rate. This is achieved by mapping the transmission of the words into different bands in the transmitting and receiving threads. The first band is 30 Hz while other sub-bands can be as slow as needed.

The ARINC I/O threads have close relationship and work in parallel with the Ethernet communication thread, which is discussed in Section 3.2.3. The data received by the ARINC 429 receiving thread are transmitted by the Ethernet thread to the simulation PC; on the other hand, the data transmitted by the ARINC 429 transmitting thread are received by the Ethernet thread from the simulation PC. The ARINC I/O thread set and reset their status flags in order to establish handshakes with the double-buffered container for its proper operation, which is discussed in Section 3.2.4.

The ARINC 429 I/O threads also cooperate with each other for performing the GPS PRAIM simulation. The receiving thread reads FMS GPS PRAIM request word. As a response of the request, the transmitting thread sends to the FMS an acknowledgment word followed by simulated GPS PRAIM word in a fixed sequence. Every time when the receiving thread gets the request, the transmitting thread must give response. Because both threads are running in the same frequency (30 Hz), therefore the reading operation for the request word in the receiving thread is tuned such that the receiving thread does not read the request word unless last response transmission is finished by the transmitting thread.

3.2.2 Analog/Digital I/O Thread

The analog/digital interface includes the I/O operations of the analog and digital signals. The analog I/O is developed in this thesis for possible future expansion of the project. The digital I/O is used to transmit discrete signals such as the aircraft weight-on-wheel.

The analog/digital interface software is built based on the NI-DAQ software for PC compatibles, which includes a set of functions that control all of the National Instruments plug-in data acquisition devices for analog/digital I/O. A thread is allocated to the analog/digital I/O operation, as shown in Figure 10. As the ARINC 429 I/O threads, this thread is controlled by the executive thread and cooperates with the double-buffered container.

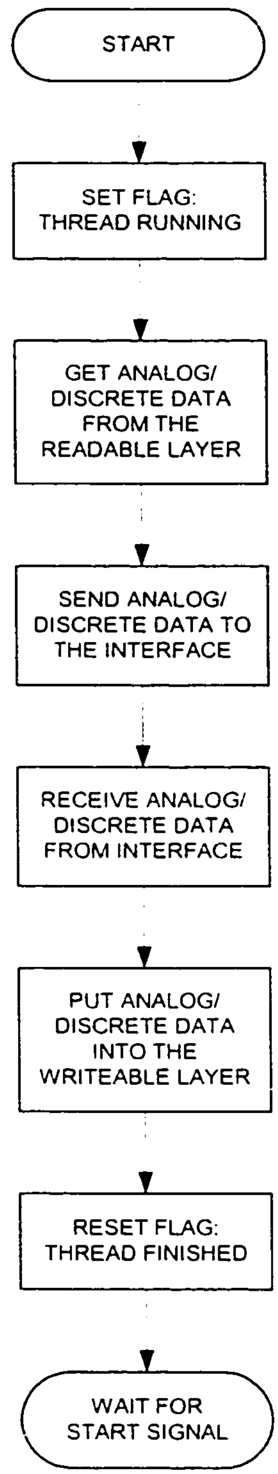

Figure 10 Analog/Digital I/O Thread

3.2.3 Ethernet Communication with the Simulation Software

Another independent thread is allocated to the Ethernet communication with the simulation software. This thread has two functions. First, it is responsible for high-speed data transmission between the two PCs through Ethernet link. Second, the pace of this thread is used by the executive thread to synchronize the operations of other threads, which is explained in Section 3.2.5. 11 illustrates the flow chart of this thread. The connection between the two PCs is first established. On the other end of the connection, i.e., the simulation PC, another communication thread is running at the same time. Both threads work together on two PCs to perform the communication tasks.

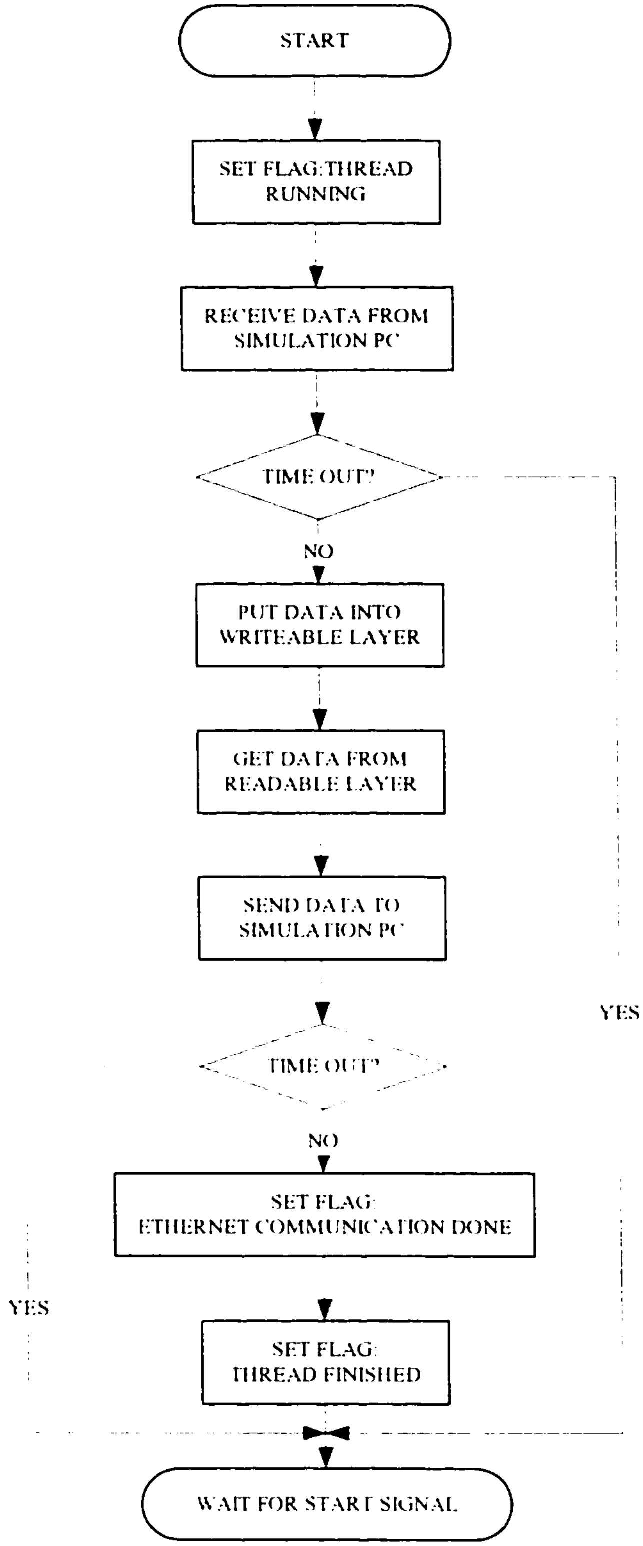

Figure 11 Etnernet Communication Thread

3.2.4 Implementation of the Double-Buffered Container

Double-buffered container is developed to ensure safe data accessing among the I/O threads described in previous sections.

In the software implementation, the three worker threads perform handshaking with the double-buffered container by resetting/setting their running status flags. Figure 12 illustrates the operation of the container, which is performed by the executive thread as explained next.

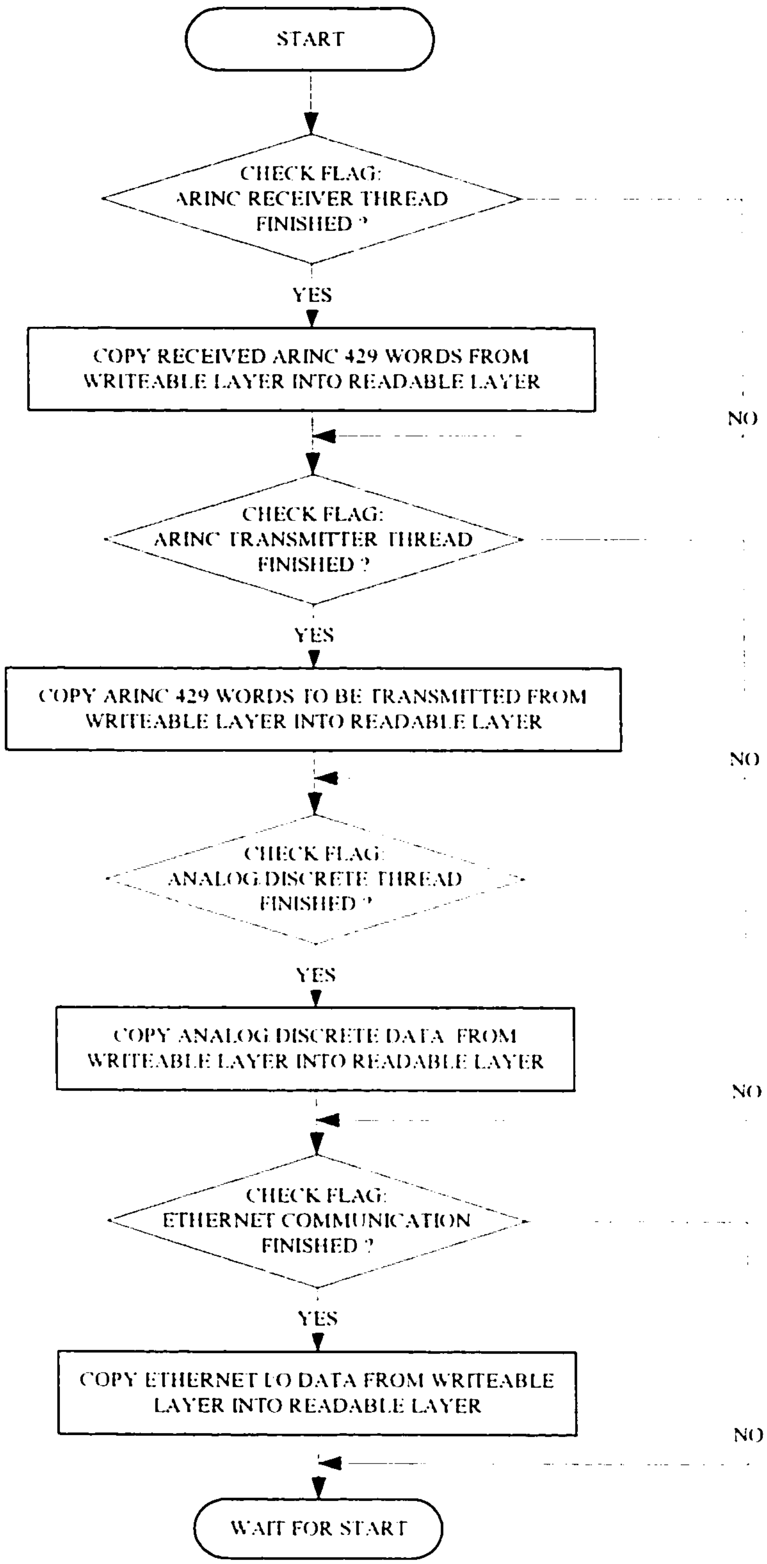

Figure 12 Double-Buffered Container Flow Chart

3.2.5 Executive Thread and Multithreaded Programming

The I/O threads developed in previous sub-sections are running in parallel and exchange information between each other. Their operations and the double-buffered container are controlled and critically synchronized by the executive thread. The executive thread is also responsible for the operation of the double-buffered container.

Figure 13 illustrates the software implementation of the executive thread and Figure 14 illustrates the relationship between the executive thread and other I/O worker threads. In Figure 13, the execution of the ARINC I/O thread and the A/D I/O thread are conditional, meaning these I/O threads are executed only when the new simulation data are received from the simulation PC. The condition is checked using the handshake between the executive thread and the Ethernet Thread.

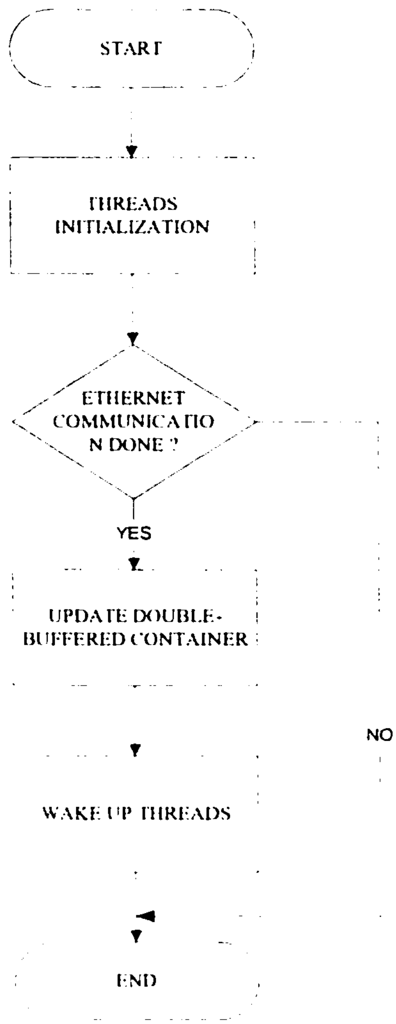

Figure 13 Multithread and Synchronization

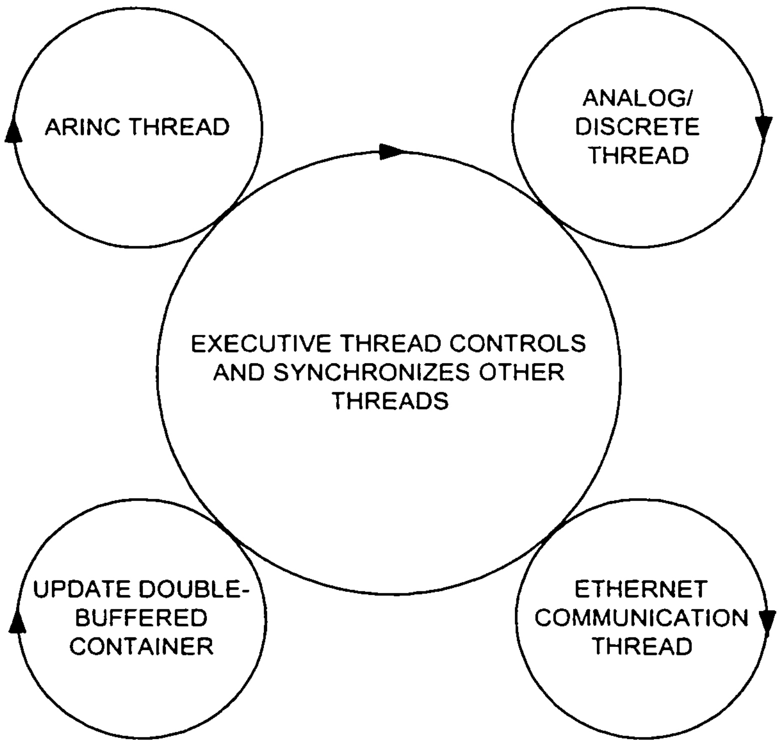

Figure 14 The Executive Thread Loop vs I/O Thread Loops

3.3 Graphical User Interface Development

The GUI software was developed using Microsoft Visual C++ ActiveX Controls [25]. The GUI provides multiple user interface functions, which mainly include operator station and data acquisition. The following sub-sections discuss the GUI developed in this work.

3.3.1 Main Dialog Window

The first window is the main window. Shown in Figure 15 is the proposed design. It consists several groups of dialogs for different interface purposes. To the upper-left corner of the GUI is the mode control panel (MCP). This panel includes boxes and buttons for setting flight control parameters and flight mode selection. Below the MCP is the panel for aircraft surface control, which include flaps, spoiler and landing gear controls. The environment panel is used to configure and multiple-level wind model and set MLS pressure and temperature. To the bottom-left of the GUI are the HIL at ETA used to simulate the GPS performance, and the navigation status window used to indicate flight status. Located on the right side of the GUI are the reposition/freeze panel and the data acquisition panels used to perform aircraft reposition/freeze and data acquisition purposes respectively.

This software also implements multithreaded programming. The foreground process of the GUI takes care of the user actions by reading in the key or mouse messages initiated by the user. The background process, which is an independently running worker thread, is responsible for the communication with the foreground process as well as the simulation software at the same time. In this way, the user commands are sent to and executed in the simulation software, the real time simulation information are read from the simulation software and displayed, plotted, printed and recorded from on the windows.

Communication with the simulation software is through TCP/IP. This task is mapped in the background thread.

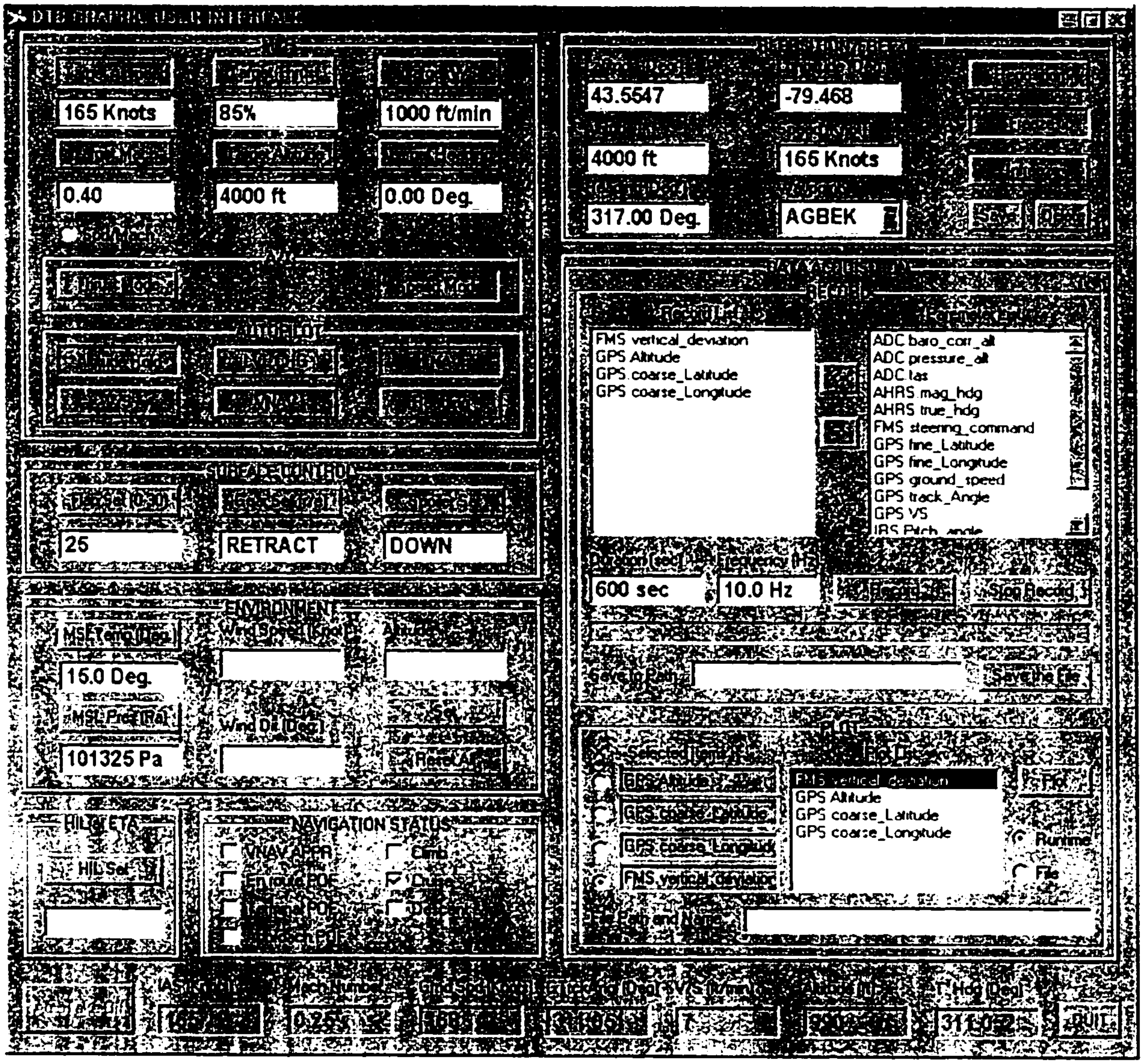

Figure 15 The Main Window of the GUI

3.3.2 Plotting Window

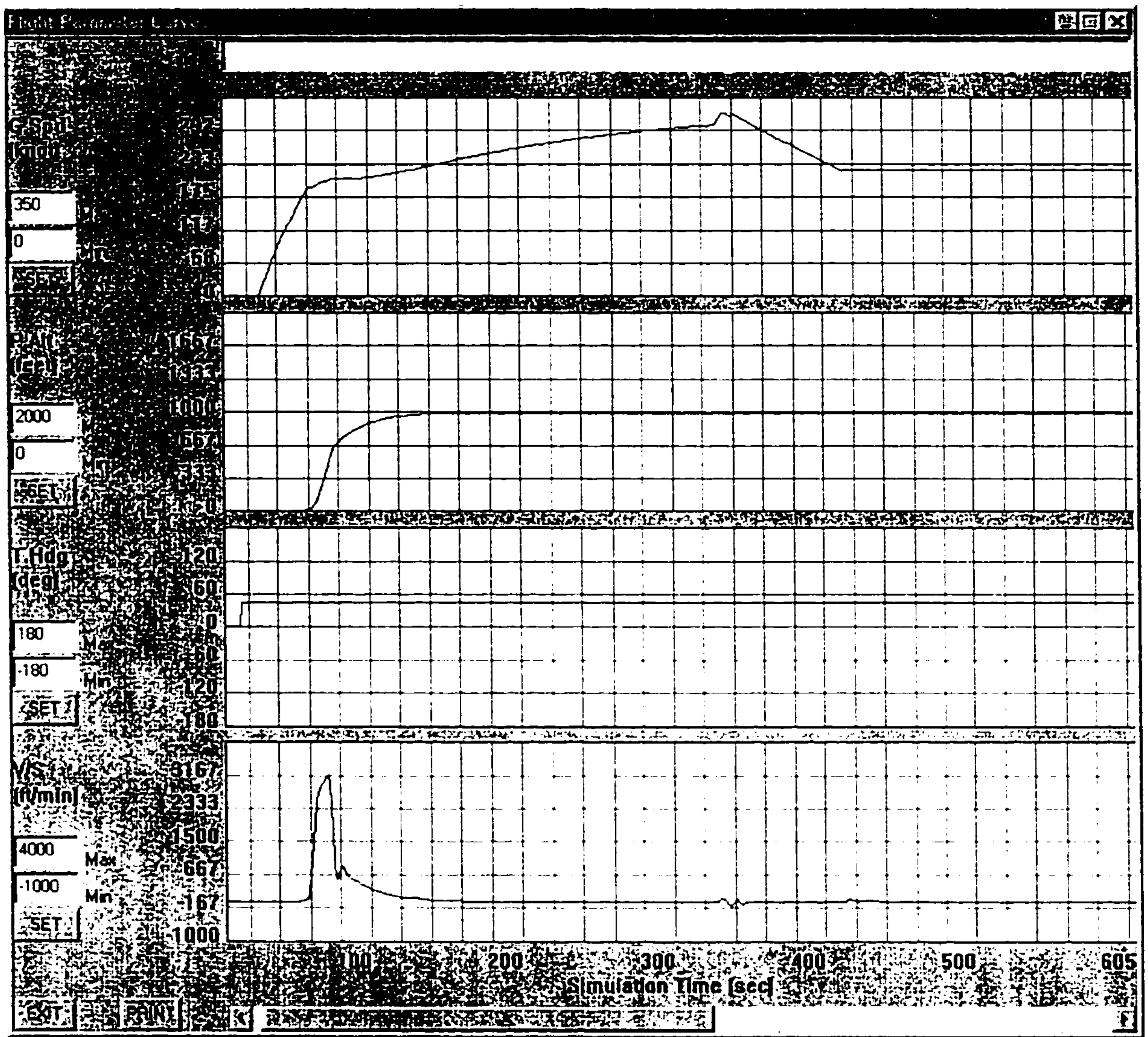

Figure 16 The Plot Window (Child Window)

This window is created as a child window of the main dialog window. The proposed layout is shown in Figure 16. Up to four parameters can be selected from the parameter list box on the main window and plotted on this window with respect to the simulation time in second. The user can zoom in or out the curves for better analysis. The plotted curves with parameter names, units and scales can be printed on the printer. Another useful feature of this window is the dynamic scroll bar. It allows the user to review back as long as 6 hours simulation data.

3.4 Summary

Through the work presented in this chapter, the interfaces of the test bed with the FMS and the user were created. The FMS interface software has multiple threads working simultaneously and synchronized by the executive thread. The GUI was developed using extensive Microsoft Visual C++, and involved extensive use of Microsoft fundamental classes (MFC). The performances of both software systems will be tested and discussed in Chapter 5.

4 FLIGHT SIMULATION SOFTWARE DEVELOPMENT

The flight simulation software is developed based on commercial available software package FLSIM V7.0. This chapter presents how this software package is modified and enhanced to meet the requirements of the DTB, and how the communications to the GUI and the executive software are implemented.

4.1 Introduction to VPI FLSIM V7.0

This software package provides an easy way for the DTB flight simulation software development. Its main features are introduced in the following subsections.

4.1.1 Aircraft Model Configuration

The VPI FLSIM V7.0 provides the flexibility of configuring any complete fixed-wing aircraft model. This is useful to the DTB since the FMS is intended to run on different aircraft types. Configurations include the six-degree of freedom aerodynamic model, weight and balance, landing gears, control surfaces, Automatic Flight Control System (AFCS), flight instruments, propulsion system, and power distribution. The configuration database can be acquired from the aircraft manufacture and is loaded during the initialization phase of the simulation.

4.1.2 Flight Control Laws

Flight control laws available in the VPI FLSIM V7.0 include the Altitude Hold and Altitude Select Control Law, Basic and Heading Select Control Laws, Roll and Yaw Align Control Laws, and Autothrottle Inner Loop [32].

4.1.3 The Global Database

VPI FLSIM V7.0 has a global data structure serving as the global database accessible for all internal and external functions. This database includes all static and runtime data of the simulation. Different modules exchange static and runtime information here.

4.1.4 Simulator Dispatcher

In the simulation software, numbers of software modules take part in the calculation. Some modules 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. The simulation dispatcher is responsible for invoking the modules based on their bands in every loop of integration. The most critical band is 33.33 milliseconds and therefore the fastest iteration rate is 30 Hz. The simulation models are integrated at a rate specified by the user, but not more than 30 Hz. This is illustrated in Figure 17.

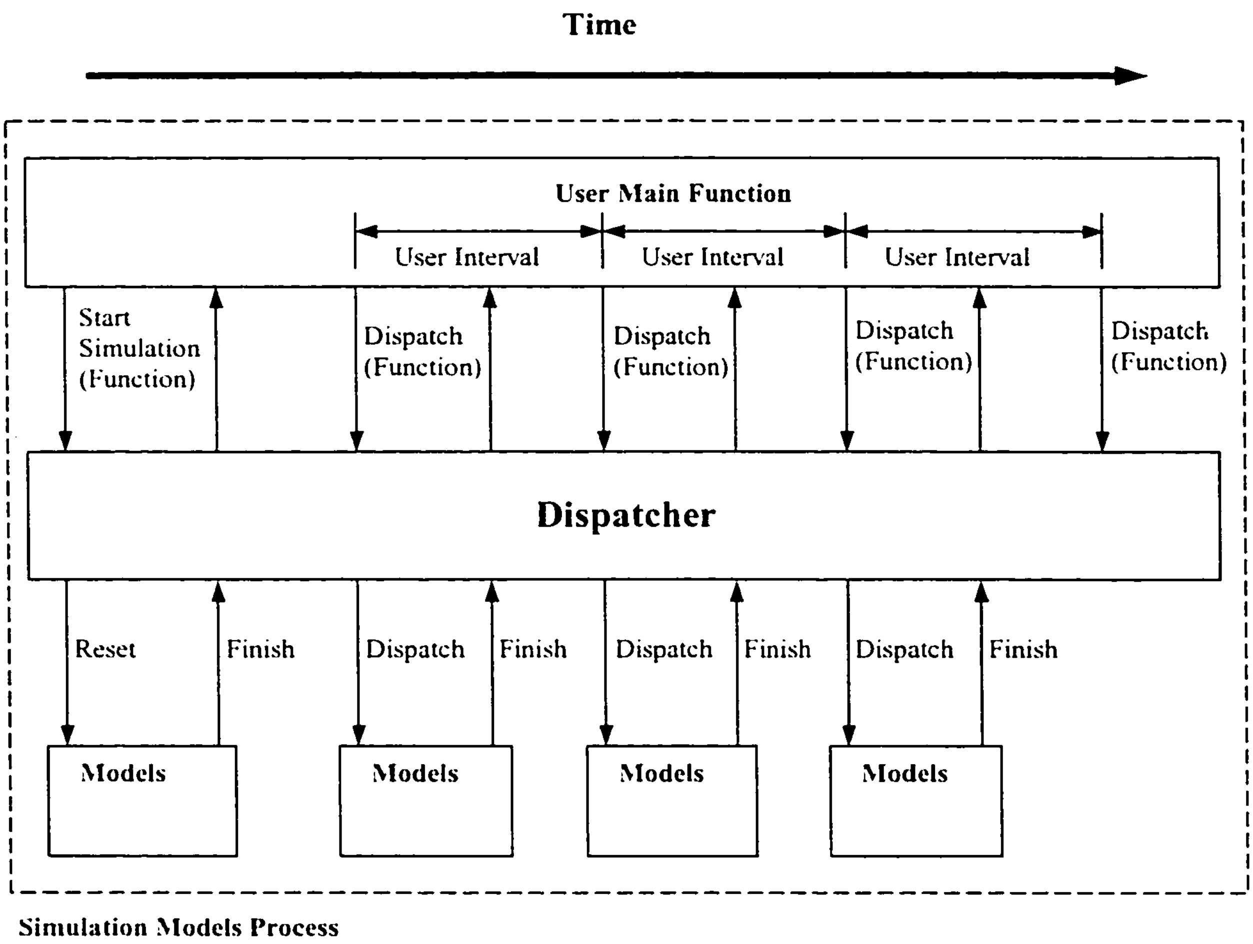

Figure 17 VPI FLSIM Dispatcher Execution Mode

4.1.5 Flexibility of Extending Simulation Functionality

The user can supply FLSIM with functionality not provided by the FLSIM simulation models. Users can also modify the behavior of FLSIM to further customize it in order to satisfy specific requirement. Next section describes the enhancement and modification to the FLSIM software package.

4.2 Development and Operation of Automatic Flight Control Modes

As indicated in Figure 15, the automatic flight control modes used are heading hold mode, LNAV mode for lateral maneuver, and altitude hold mode, vertical speed mode, level change mode, VNAV mode for vertical maneuver. Every mode implements its specific control law [32]. The control laws for the heading mode, LNAV mode, altitude hold mode and vertical speed mode already come with the FLSIM v7.0. Therefore these modes do not need be developed from scratch. However, the development of the level change mode and VNAV mode require new control laws to be developed first. Only the level change control is developed in this thesis.

4.2.1 Heading Mode

The heading mode is used to intercept and maintain a magnetic heading reference selected by the user. This mode is engaged by pushing the Heading button on the GUI shown in Figure 15. Then the Heading button is depressed automatically indicating the engagement of the mode and the Target Heading button is depressed automatically indicating the selection of the target heading. The target heading is always the heading displayed in the Target Heading box. The error signal between the current heading and the target heading is sent to the simulated automatic flight control system (AFCS). The AFCS generates the proper roll command to intercept and maintain the target heading.

- Engaging the heading mode resets any previously selected lateral mode. The heading mode is canceled by any one of the following:

- Selecting the LNAV mode
- Repositioning the aircraft.

Otherwise, this mode is not canceled when the target heading is reached and maintained.

4.2.2 LNAV Mode

The LNAV mode is used to execute the FMS roll command. This mode is engaged by pressing the LNAV button shown in Figure 15. The LNAV button keeps pressed indicating the engagement of this mode. In this mode, the AFCS uses a composite lateral steering command from the FMS. This lateral steering command is lateral gain programmed in the FMS, and is not gain programmed again in the AFCS.

- Engaging the LNAV mode resets any previously selected lateral mode. The LNAV mode is canceled by any one of the following:
- Selecting the heading mode
- Repositioning the aircraft.

Otherwise, this mode is not canceled even if the FMS steering command is zero.

4.2.3 Altitude Hold Mode

The altitude hold mode is used to maintain a barometric altitude reference. To fly altitude hold mode, the pilot must do the following:

- Be in any lateral mode
- Push the Altitude Hold button on the GUI in Figure 15.

The FCC maintains the barometric altitude that the aircraft was flying at the moment the mode was engaged. Selecting the altitude hold mode on the MCP cancels any other previously selected vertical mode.

The altitude hold mode is canceled by selecting any other vertical mode.

4.2.4 Altitude Pre-Select Mode

The altitude pre-select mode is used in conjunction with another vertical mode, which will be described next. The pilot can use altitude pre-select to automatically capture, level off, and hold a pre-selected altitude that has been entered in the target altitude box of the GUI in Figure 15. The altitude pre-select mode captures and levels off at the pre-selected altitude, while another vertical mode is used to fly to the desired altitude.

The aircraft flies toward the target altitude using one vertical mode (vertical speed mode or level change mode), while altitude pre-select mode is armed to automatically capture the pre-selected altitude. When the altitude select captures, the other active vertical mode is dropped.

The aircraft remains in the altitude pre-select mode until the altitude error is less than 25 feet, and then the vertical mode is switched to the altitude hold mode. Changing the target altitude in the Target Altitude box while in the capture phase cancels the active altitude pre-select mode, activates the previous vertical mode.

4.2.5 Vertical Speed Mode

The vertical speed mode is used to automatically maintain the aircraft at a user-selected vertical speed reference toward a target altitude. The target airspeed is maintained by the autothrottle at the same time. This mode is initiated by pushing the V/S button on the GUI. When the vertical speed hold mode is engaged, the following buttons shown in Figure 15 are pressed:

- V/S button is pressed indicating the engagement of the vertical speed hold mode
- Target V/S button is pressed indicating the selection of the target vertical speed displayed in the box
- Speed Mode button is pressed indicating the autothrottle is initiated
- Target Airspeed button is pressed indicating the target airspeed to be maintained by the autothrottle
- Target Altitude button is pressed indicating the target altitude to which the aircraft is flying.

To fly the vertical speed mode in a climb to a target altitude from any other vertical modes, the target altitude must be higher enough than the actual altitude so that the actual altitude is outside the capture margin (300 ft) of the target altitude. When the aircraft flies within 300 ft of the target altitude, the vertical speed mode is overridden by the altitude pre-select mode to capture and maintain the target altitude, which has been described in Section 4.2.4. Figure 18 illustrates a climbing profile from altitude hold mode switched to vertical speed mode.

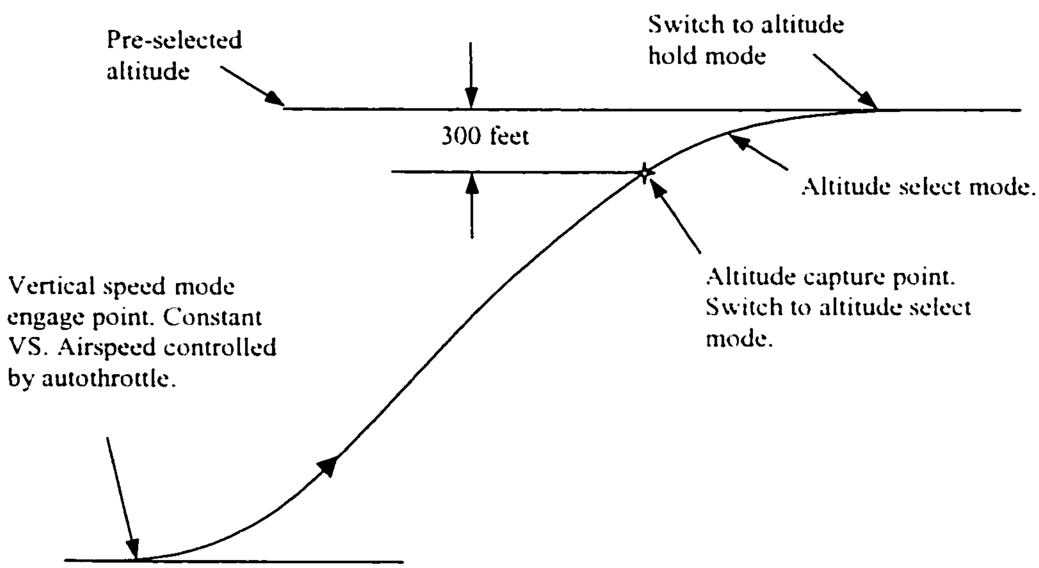

Figure 18 Vertical Speed Hold Climbing Profile

When vertical speed hold is selected, it cancels all previously selected vertical modes. The vertical speed can be canceled by any one of the following:

- Capturing of target altitude
- Selecting another vertical mode

- Repositioning the aircraft.

4.2.6 Level Change Mode

When selected, the level change mode will fly the aircraft from one altitude to a target altitude. At the same time, the engine thrust stays unchanged and the target airspeed or Mach number is tracked and maintained by varying the pitch attitude. The adjustment of the pitch attitude is governed by a control law named level change control law. This section first describes the development of the level change control law, then the development of the level change mode is followed.

4.2.6.1 Development of Level-Change Control Law

When the aircraft is flying from one altitude to another under the level change mode, and when there is a difference between the current IAS and the target IAS, the pitch attitude will be adjusted automatically in such a way that a acceleration rate will be induced to decrease the difference of IAS. The relationship between the difference of IAS and the acceleration rate is represented by the following equation:

$$\alpha_X + K_1 \times \Delta V = 0 \qquad (4\text{-}1)$$

where α_X is the acceleration rate in the aircraft X axis, ΔV is the difference of the target airspeed IAS_t and current airspeed IAS. K_1 is the gain and is assumed to be linear for simplification.

The acceleration rate in the above equation is the ideal value in order to track the target IAS. A deviation from the ideal rate will cause unbalance of the equation. A pitch angle adjustment is therefore required to adjust the acceleration rate so that the target IAS will be better followed. This pitch angle adjustment is presented by the following equation:

$$\Delta\theta = K_2 \times (\alpha_X + K_1 \times \Delta V) \qquad (4\text{-}2)$$

where $\Delta\theta$ is the pitch angle adjustment, K_2 is the gain and again is assumed to be linear for simplification.

Then the new command pitch angle for the flight control system can be represented by equation:

$$\theta_C = \Delta\theta + \theta = K_2 \times (\alpha_X + K_1 \times \Delta V) + \theta \quad (4\text{-}3)$$

where θ_C is the command pitch angle, θ is the aircraft current pitch angle.

This equation represents the level change control law. In addition to the equation, this control law is subjected to two limits. One is for the acceleration rate and the other is for the command pitch angle. Figure 19 is the schematic of the level change control law.

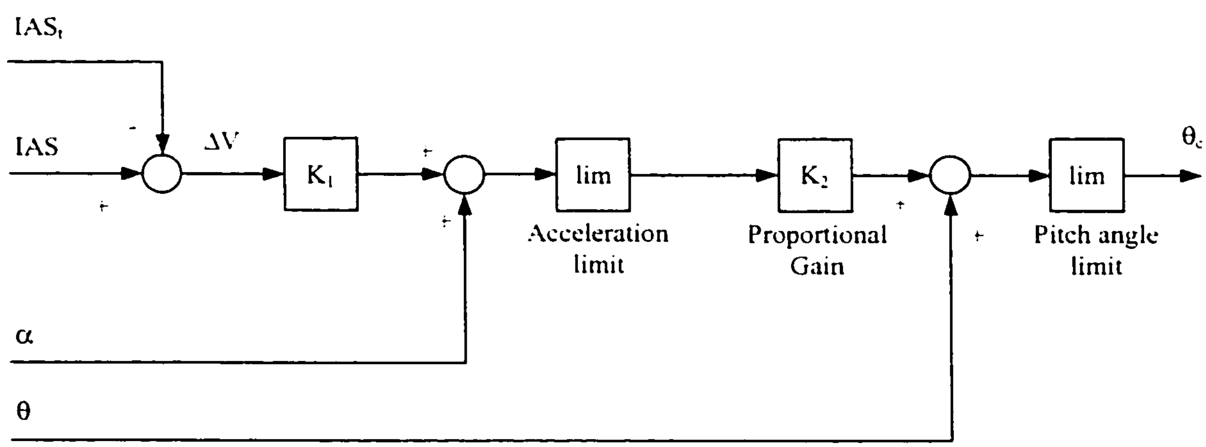

Figure 19 Level Change Control Law

4.2.6.2 Tuning of Linear Gains

As shown in Equation (4-3) and Figure 19, there are two linear gains in the control law. The K_1 is used to control how fast the target airspeed is captured. The bigger the K_1, the faster the target airspeed is captured. A too big K_1 will cause very fast capture but big overshoot. The K_2, on the other hand, directly affects the pitch angle response, and therefore indirectly affects the airspeed change rate. Bigger K2 causes sharper pitch angle change. Both gains must be tuned and complement with each other in order to achieve good static and dynamic performance.

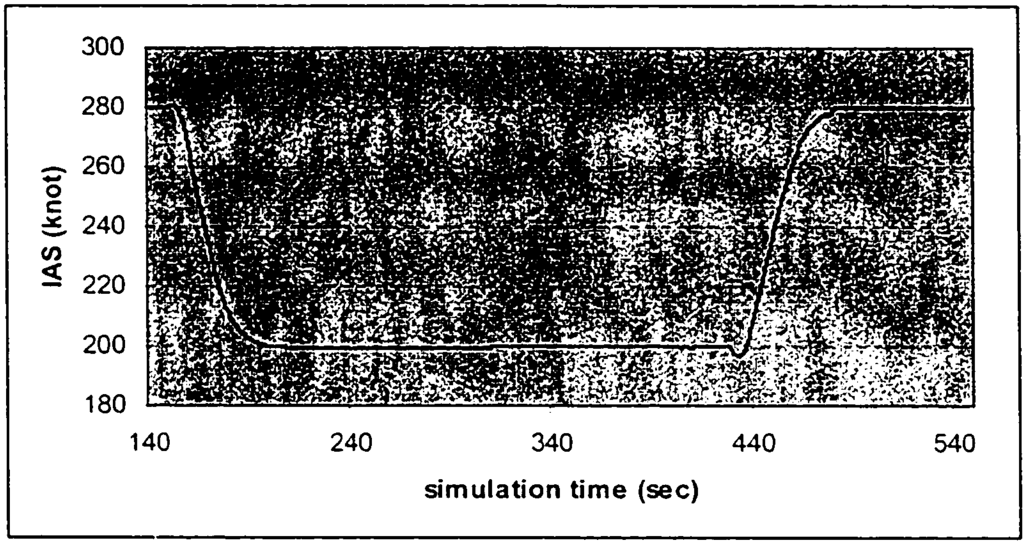

Figure 20 Target Airspeed Capturing and Tracking under Level Change Control Law (K_1=0.2, K_2=0.1)

Figure 20 to 22 illustrate the control law performance with K_1 and K_2 tuned to be 0.2 and 0.1 respectively. In these figures, the level change mode was first selected at simulation time 152 seconds, and was flying from current altitude 1000ft to target 8000ft.

When the target altitude was captured and maintained, the level change mode was again selected to fly from 8000ft to target altitude 4000ft.

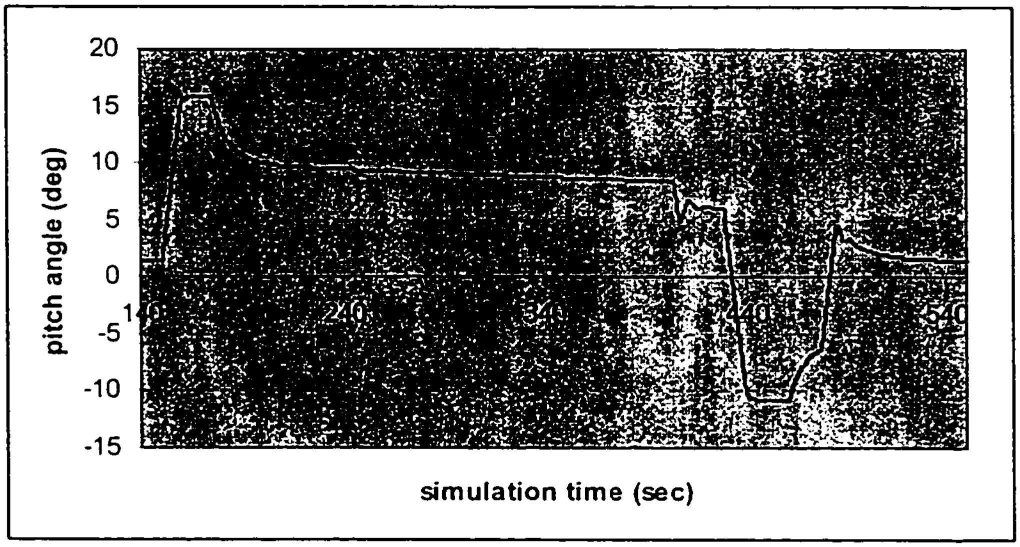

Figure 21 Pitch Angle Response under Level Change Control Law (K_1=0.2, K_2=0.1)

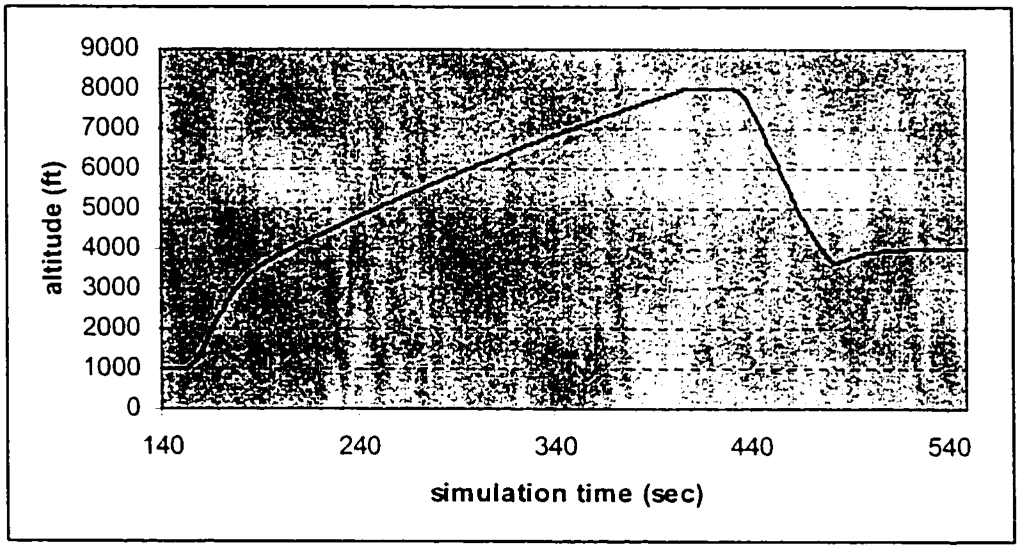

Figure 22 Target Altitude Capturing and Tracking under Level Change Control Law (K_1=0.2, K_2=0.1)

4.2.6.3 Calculation of Target Mach Number

The level change mode is designed to be able to track a target airspeed or a Mach number. When a target Mach number is tracked, it has to be translated into the corresponding target airspeed. For subsonic flight, which is the case for commercial aircraft and most business jets, the Mach number and the IAS can be calculated separately using equation (4-4) and (4-5) respectively:

$$M = \left[\left(\frac{2}{\gamma-1}\right)\left[\left(\frac{q_c}{P}+1\right)^{\frac{\gamma-1}{\gamma}}-1\right]\right]^{\frac{1}{2}} \qquad \text{(4-4) [33]}$$

$$V_{IAS} = \left[\left(\frac{2\gamma}{\gamma-1}\right)\left(\frac{P_o}{\rho_o}\right)\left[\left(\frac{q_c}{P_o}+1\right)^{\frac{\gamma-1}{\gamma}}-1\right]\right]^{\frac{1}{2}} \qquad \text{(4-5) [33]}$$

where:

γ	-	Air constant (=1.4)
P_o	-	Sea level pressure
ρ_o	-	Sea level air density
q_c	-	P_{Total}-P

Comparing the two equations gives nearly a linear relation between the Mach number and the IAS. During the simulation, the Mach number and IAS are updated every simulation iteration and stored in the FLSIM global database. Then the target IAS corresponding to the target Mach number at the aircraft's current position can be found by using equation (4-6).

$$IAS_t = (Mach_t \times IAS_c) \div Mach_c \qquad (4\text{-}6)$$

Where:

IAS_t - the target IAS equivalent to the target Mach number.

IAS_c - the current IAS of the aircraft.

$Mach_t$ - the target Mach number of the aircraft.

$Mach_c$ - the current Mach number of the aircraft.

This calculation needs to be performed every simulation iteration because for a fixed airspeed, the Mach number is different at different altitude and vice-versa. Instead of directly using the Mach number, the equivalent IAS_t will be used as the input to the level change control law in real-time.

4.2.6.4 Implementation of Level Change Mode

The level change mode takes target altitude and target airspeed or Mach number as inputs. The target altitude is used by the flight control system to determine whether to climb or descend by comparing the target altitude with the aircraft current altitude. The control system sets the throttle to 90 percent of full throttle for climbing if the target altitude is more than 300 feet above the current altitude. In the reverse case, if the target altitude is more than 300 feet below the current altitude, the control system sets the throttle to 30 percent of full throttle for decent. If the target altitude is within +/-300 feet of the current altitude, then instead of engaging the level change mode, the altitude select mode is activated in which the thrust is controlled by the autothrottle to maintain the target airspeed or Mach number.

The target airspeed or Mach number is one of the inputs of the level change control law. In case of Mach number, the equivalent target airspeed is calculated. This calculation is performed every simulation iteration because the equivalent airspeed to the target Mach number keeps changing from one altitude to another. The pitch command is calculated based on the target airspeed and the current airspeed by the control law. The pitch command and the thrust are fed into the FLSIM core simulator for integration. Above-described are illustrated in the flow chart in Figure 23.

When the target altitude is captured, the aircraft exits level change mode and switches to altitude hold mode automatically. The level change mode is canceled automatically when another vertical mode is entered.

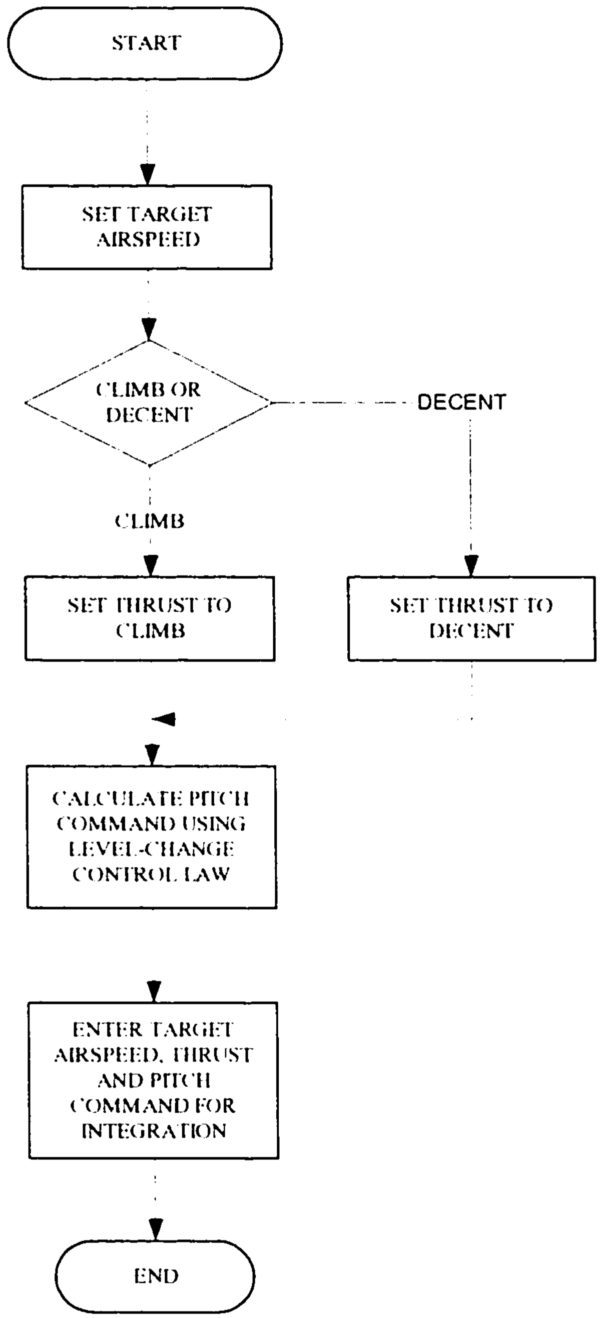

Figure 23 Level-Change Mode Flowchart

4.2.6.5 Operation of Level Change Mode

The level change mode overrides all active vertical modes. This mode is engaged by pushing the LVL CHG button on the mode control panel in Figure 15. The following buttons are pressed in the level change mode:

- LVL CHG button is pressed indicating the selection of this mode
- Thrust Mode button is pressed indicating the engagement of the thrust mode
- Target Thrust button is pressed indicating the thrust for climbing or descending
- Target Altitude button is pressed indicating the target altitude to which the aircraft is flying
- Target Airspeed button is pressed when holding an airspeed, or Target Mach button is pressed when holding a Mach number. This target airspeed or Mach number can be changed in fly.

To fly the level change mode in a climb to a target altitude from any other vertical modes, the target altitude must be higher enough than the actual altitude so that the actual altitude is outside the capture margin (300 ft) of the target altitude. When the aircraft flies within 300 ft of the target altitude, the level change mode is overridden by the altitude pre-select mode to capture and maintain the target altitude. The same conditions apply on flying in a descent. Figure 24 depicts a climbing profile under level change mode.

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

- Capturing the target altitude
- Selecting any other vertical modes on the GUI
- Repositioning the aircraft.

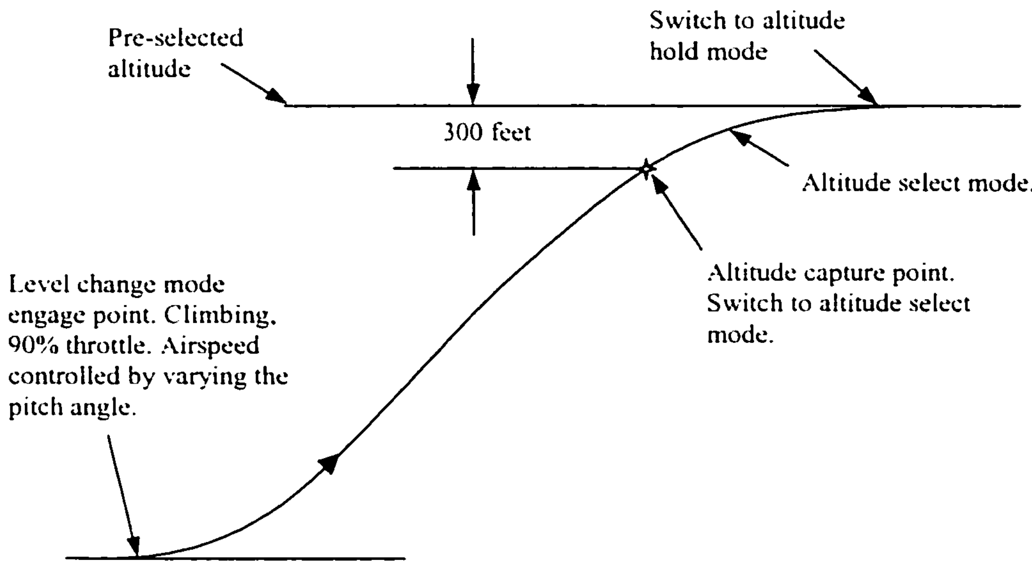

Figure 24 Level Change Mode Climbing Profile

4.3 Environment Simulation

The environment simulated include the multiple level wind, MSL temperature and MSL pressure. Development of these models are described in the following.

4.3.1 Development of Multiple Level Wind Model

This section describes the development of the multiple level wind model.

4.3.1.1 Configuration of Multiple Level Wind Model

The multiple level wind model has re-configurable wind directions and velocities. The wind direction and velocities are configured in such a way that the direction and velocity are changing linearly between adjacent two layers, as illustrated in Figures 25 and 26. The directions and velocities are configured with respect to the X, Y and Z axes of the earth's NED frame. They are transferred into the BODY frame of the aircraft for aerodynamic calculation of the simulation, as described next.

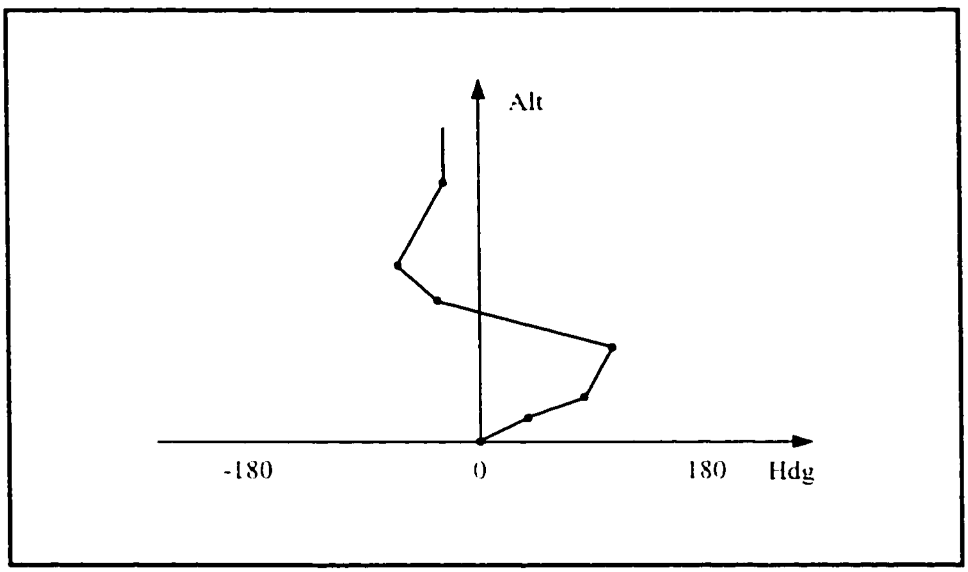

Figure 25 Illustration of Wind Direction Change between Layers

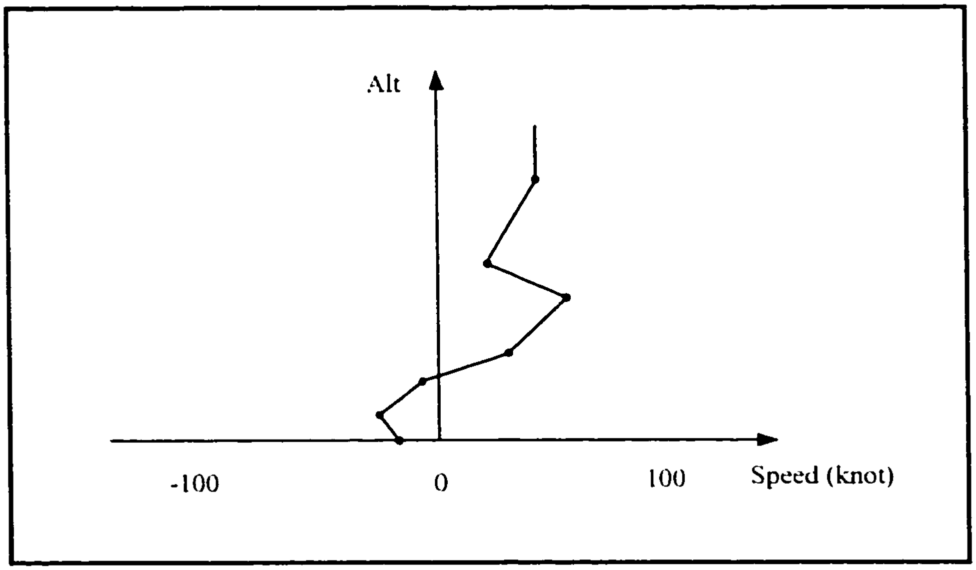

Figure 26 Illustration of Wind Velocity Change between Layers

4.3.1.2 Transformation from NED Frame to BODY Frame

The transformation from the NED frame to the BODY frame is through rotations about the x, y and z axis of the NED frame. 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.

The rotation matrix is:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{body} = \begin{bmatrix} l_x & m_x & n_x \\ l_y & m_y & n_y \\ l_z & m_z & n_z \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{NED} \qquad (4\text{-}7)\ [34]$$

Where:

$$lx = \cos(\theta)\cos(\psi) \qquad (4\text{-}8)$$

$$m_x = \cos(\theta)\sin(\psi) \qquad (4\text{-}9)$$

$$n_x = -\sin(\theta) \qquad (4\text{-}10)$$

$$l_y = -\cos(\phi)\sin(\psi) + \sin(\phi)\sin(\theta)\cos(\psi) \qquad (4\text{-}11)$$

$$m_y = \cos(\phi)\cos(\psi) + \sin(\phi\sin(\theta)\sin(\psi) \qquad (4\text{-}12)$$

$$n_y = \sin(\phi)\cos(\theta) \qquad (4\text{-}13)$$

$$l_z = \sin(\phi)\sin(\psi) + \cos(\phi)\sin(\theta)\cos(\psi) \qquad (4\text{-}14)$$

$$m_z = -\sin(\phi)\cos(\psi) + \cos(\phi)\sin(\theta)\sin(\psi) \qquad (4\text{-}15)$$

$$n_z = \cos(\phi)\cos(\theta) \qquad (4\text{-}16)$$

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

Above rotations are performed in the simulation software.

4.3.2 Development of Ambient Condition Model

The ambient condition, such as the wind, temperature and pressure, have significant influence to the flight, and therefore affect the performance of the FMS. This section describes the development of the ambient condition models used for the flight simulation.

4.3.2.1 Calculation of Ambient Temperature

The earth's atmosphere extends up beyond 100 kilometers and is defined by numerous layers or shells that have different characteristics. In ascending order of height, these layers are the troposphere (sea level to an altitude of 11 km), the stratosphere (from 11 km to 20 km), the mesosphere and the thermosphere. In studying the performance of transport jet aircraft, our interest in the atmosphere is limited to the lower 15 km, meaning that only the troposphere and stratosphere will be considered.

The International Civil Aviation Organization (ICAO) standard atmosphere is defined in order for aircraft performance reference and instrument calibration purposes. It defines a static meteorological condition that is an average representation of the atmosphere. The instrument calibrations refer to the instruments used to monitor the flight path of the aircraft and include barometric altitude, rate of climb, indicated airspeed, Mach number and total temperature. Aircraft performance data is based on that standard atmosphere. ICAO Standard Atmosphere properties are listed in Table 2 [35], and the temperature gradient for the lower three layers are shown in Figure 27 [35].

Table 2 ICAO Standard Atmosphere Properties

ICAO Standard Atmosphere 1964			
Parameter	Symbol	English Units	Metric Units
Sea Level Temperature	T_o	+59°F +518.67°R	+15°C +288.15°K
Sea Level Pressure	P_o	2116.22lb-ft^{-2}	1.01325x10^5 newtons-m^{-2}
Sea Level Density	?$_o$	2.37688x10^{-3} slug-ft^{-3}	1.225kg-m^{-3}
Sea Level Sound Speed	a_o	1116.45ft-s^{-1}	340.294m-s^{-1}
Tropopause Lapse Rate	A	-0.003566°R-ft^{-1}	-6.5°K-km^{-1} 0.00198°C-ft^{-1}

Gas Constant R for Dry Air	R	1716.5 ft-lb-slug^{-1}-°R^{-1}	287.05 N-m-kg^{-1}-°K^{-1}
Kinematic Viscosity	v	1.5723x10^{-4}ft^2-s^{-1}	1.4607x10^{-5}m^2-s^{-1}
Tropopause Pressure Altitude	h_{trop}	36,089ft	11km
Tropopause Temperature	T_{trop}	-69.7°F +389.97°R	-56.5°C +216.65°K

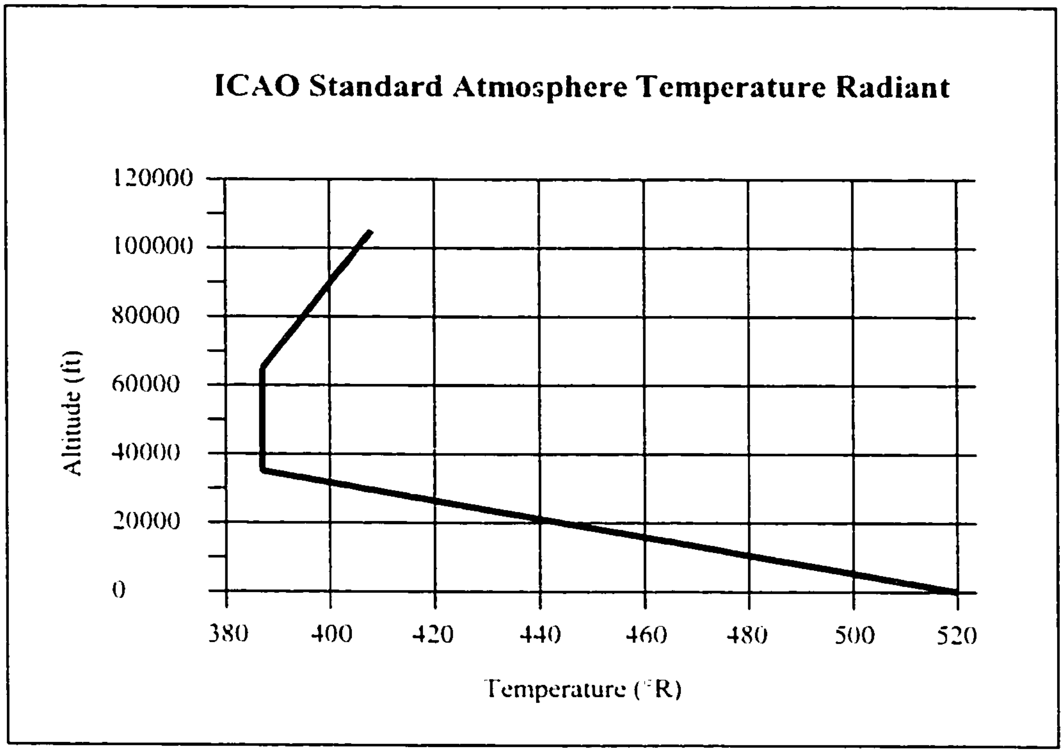

Figure 27 ICAO Standard Atmosphere Temperature Gradient

The troposphere is the lowest level and is characterized by having most of the weather (moisture), winds, pollutants, and a negative lapse rate. The average lapse rate in the ICAO Standard Atmosphere is –6.5°C per km. Thus the temperature at the aircraft's position will be calculated by equation

$$T_{TROP} = T_{MSL} + 6.5(Alt \div 1000)\ (°K) \quad (4\text{-}17)\ [36]$$

where:

T_{MSL} - MSL Temperature in °K

Alt - the aircraft radar altitude in meter

The stratosphere has a zero temperature lapse rate and therefore the temperature remains unchanged within this layer.

$$T_{STRATO} = T_{MSL} + 6.5(10999.92 \div 1000) \ (°K) \qquad (4\text{-}18)$$

4.3.2.2 Calculation of Ambient Pressure

In the Troposphere, the ambient pressure is calculated by equation:

$$P_{TROP} = (T_{TROP} \div T_{MSL})^{5.2563} \times P_{MSL} \ (Pa) \qquad (4\text{-}19) \ [36]$$

Where:

P_{MSL} - MSL Pressure in Pa

And in the Stratosphere, by equation:

$$P_{STRATO} = (T_{STRATO} \div T_{MSL})^{5.2563} \times \exp(-(Alt - 10999.92) \div 6341.882) \times P_{MSL} \ (Pa) \qquad (4\text{-}20) \ [36]$$

4.3.2.3 Calculation of Ambient Air Density

Having the ambient pressure and temperature calculated, then for both Troposphere and Stratosphere, the ambient density can be calculated as:

$$\rho = P \div (287.05 \times T) \ (kg/m^3) \qquad (4\text{-}21) \ [36]$$

where:

P	-	aircraft ambient pressure in Pa
T	-	aircraft ambient temperature in °K

4.4 Simulation of Navigation Facilities

Navigational data are fed into the FMS to perform its functionality during flight. They are simulated during flight by extracting from the flight simulation results, which are stored in a global database. The navigation facilities used by the FMS, from higher to lower priority, are GPS, VOR/DME, IRS or AHRS. Air data computer provides airspeed and attitude information to the FMS and therefore is another important sensor to the FMS.

Navigation models are created to simulate individual navigation facility connected to the FMS. Models developed currently include ADC, GPS, AHRS and IRS. To simulate the VOR/DME, a global navigation database containing radio stations information has to be loaded. The global navigation database has to be updated from time to time, as what is done to the FMS global navigation database.

The navigation information are transmitted to the FMS in ARINC 429 format as different ARINC 429 labels. Different A429 label has different transmission speed. A429 labels are grouped and transmitted on different A429 channels. Appendix B lists the transmission speeds and ARINC 429 channel assignments for all the labels.

4.5 Flight Freeze and Reposition

In the DTB, the same aircraft-identical Flight Management Systems are used in the simulation. Since these units were of course originally designed for use in aircraft, in a simulation environment they may be subjected to conditions such as freezes and repositions which they cannot handle without unwanted effects. A typical example of such a case is that of reposition, where the aircraft latitude and longitude may change drastically in value for a reposition from one place on the globe to another, perhaps thousands of miles distant. The FMS software does not recognize such a jump in position to be valid, and will only update its own position very slowly, despite that of its various input sensors such as IRS and GPS. This creates a conflict in the simulation, of course, when the FMS position does not follow during reposition. It may be updated manually via a position update by the user via the CDU, but this is both time-consuming and very inconvenient.

One solution is to build up the equivalent CDU keystroke sequences as manually updating the FMS position using the CDU key code, and feed the sequences into the CDU. This way, we need to intercept the communication between the CDU and the FMU in terms of both hardware and software. Furthermore, software utility needs to be developed to handle the key sequence construction.

A more efficient way is to do the reposition by taking advantage of the built-in FMS simulator feature. The FMS provides a built in simulator feature to handle this task. This simulator feature allows the user to acknowledge the FMS of the reposition and transmit the aircraft new position to the FMS via an ARINC 429 control bus without any conflicts between the FMS and the simulation system. When a reposition command is ordered and new position entered, a flag will be set and ARINC 429 words containing new aircraft latitude, longitude, and altitude will be built. This flag and ARINC 429 words will be sent to the FMS to update the aircraft three-dimensional position. In this way, neither extra hardware connection nor extra software is needed.

5 DTB SYSTEM INTEGRATION AND TESTING

This chapter discusses the software system integration and performance. The runtime data are collected and analyzed using Microsoft Excel.

5.1 DTB Software Systems Integration

The flight simulation software and the GUI software are running on the simulation PC. The interface software which incorporates the drivers for the national instrument card and ARINC 429 card is running on the I/O PC. These three software are connected by the Ethernet link. The connections among the three software are robust, which means that if the connection is terminated at one end, the process at the other end will wait for re-connection.

Figure 28 is a UML (Unified Modeling Language) [37] sequence diagram illustrating the overall execution sequence of the processes of the DTB after integration. As shown in the figure, the simulation software receives data from the user through the GUI, and from FMS through the interface software. After integration, the simulation software sends simulation results back to the GUI and the FMS through interface software. The user can send command (sequence 5 in Figure 28) at any moment to the GUI. The data transmission between the GUI and simulation software, and between the simulation and the interface software are real-time.

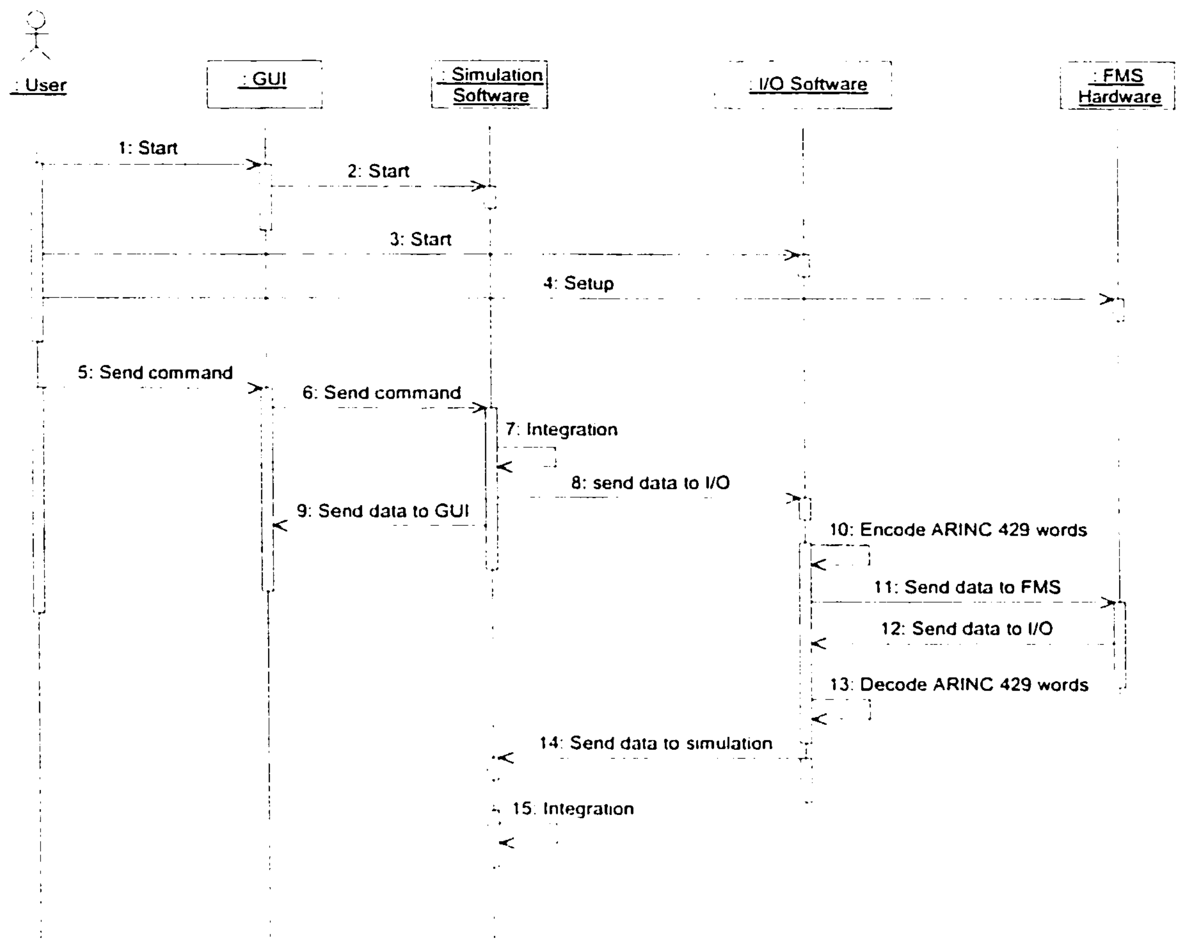

Figure 28 DTB System Overall Sequence Diagram

5.2 System Testing and Evaluation

The test of the DTB shall include individual test of the simulation software, the GUI, the interface software. It shall also include the test of the complete DTB software system that is the integration of the above-mentioned three software components. In previous chapters, test of the individual software has been described with the development. This chapter will describe the test and evaluation of the integrated DTB software system by creating a complete flight plan using the FMS, and executing the flight plan on the DTB.

5.2.1 FMS Initialization and Flight Plan Setup

Before setting up the flight plan, the FMS has to be initialized. All the DTB software systems were first started to establish the Ethernet connections and initiate the interface cards. Then the aircraft was repositioned on the ground and the required information was sent to the FMS.

A standard flight plan was selected from the CDU, which was from CYMX (Montreal Doval International Airport) to CYYZ (Toronto Airport). Waypoints contained in this flight plan was AGBEK. The aircraft was repositioned on a virtual runway heading east (90 degree).

5.2.2 Execute the Flight Plan

The flight phases simulated included take-off, en-route flight and approaching. The FMS LNAV mode was used for lateral flight guidance during all the phases of flight except the take-off. The user interface on the simulation PC, which includes the built-in head-up display (HUD) of the FLSIM v7.0 and the GUI is illustrated in Figure 29. The window located at the bottom-left corner of the figure is the HUD and the big window containing buttons and boxes is the GUI. Detailed execution of the testing is described next.

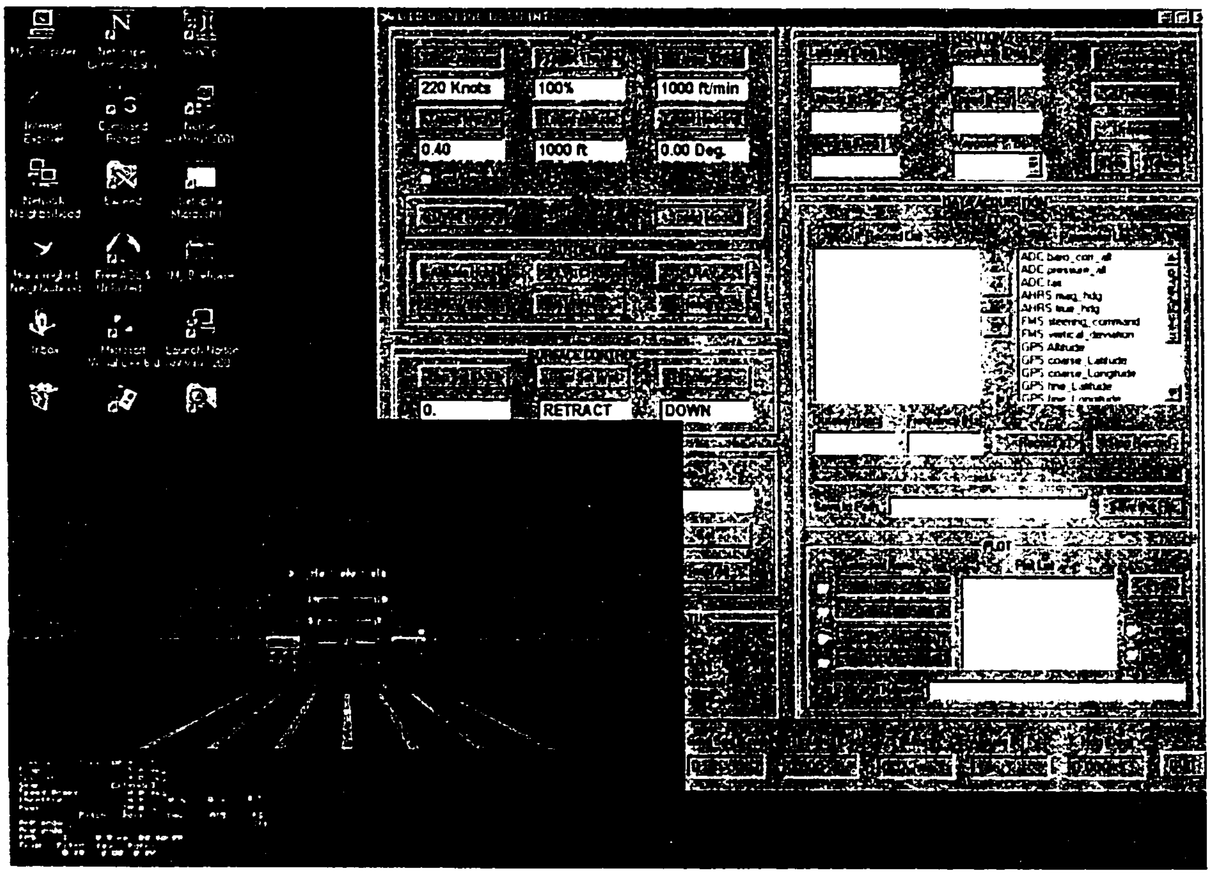

Figure 29 GUI and Head-up Display

5.2.2.1 Auto Take-off

The aircraft was positioned on a virtual runway 9, which does not exist at CYMX, the departure airport. Then by clicking on the Auto Take-off button from the GUI, the throttle was set to full and the aircraft started to accelerate. When the speed reached 160 knots, the flaps were deployed to 12 degrees and the aircraft started climbing. The take-off phase was finished when the aircraft climbed to 400 ft high. At this point, the autopilot was engaged to fly the aircraft under level change mode to the target altitude.

5.2.2.2 En-route

After the taking-off is the en-route flight. The FMS LNAV mode was always engaged for lateral navigation and flight control. The aircraft was flown by the autopilot using the AFCS LNAV mode following the FMS lateral steering. The 243 degrees heading, which is the course from CYMX to AGBEK, was captured and maintained, as shown in Figure 30 and 31.

The vertical profile was controlled by the user through the GUI during flight by entering altitude targets. The aircraft climbed to 1,000 ft and then to 4,000 ft under autopilot level change mode, which was engaged by clicking on the LVL CHG button from the GUI. The airspeed was controlled by the autothrottle to 220 knots. Then the throttle was set to 90 percent and the pitch attitude was automatically adjusted in order to capture and maintain the target airspeed. This AFCS mode controlled the aircraft flying toward the target altitude. When the aircraft reached an altitude of 3800 ft, the target altitude was considered to have been captured and the AFCS mode was switched to altitude hold mode automatically and the 4000 ft target was captured. Under the altitude hold mode, the airspeed was maintained by the autothrottle.

5.2.2.3 Reposition/Freeze/Unfreeze

When the steering command becomes less than 5 degrees, the target heading toward the next waypoint has been captured. At this point, the aircraft can be repositioned to where the test requires. In the test, the aircraft was repositioned at AGBEK waypoint with altitude 2,500 ft, IAS 170 knots, heading 317 degree, latitude N43°33.50, longitude W079°28.08. When initiating the reposition, the simulation was frozen until it got resumed by the user by clicking on the Unfreeze button.

5.2.2.4 Approach

After resuming the flight simulation from reposition, the LNAV and VNAV were all engaged to capture and track the glide slope. In this phase of flight, the FMS LNAV mode is still responsible for the lateral navigation and flight control of the aircraft. The target runway heading is 317 degrees. In the vertical direction, the FMS VNAV mode was engaged and the GPS glide slope vertical deviation was used as vertical guidance command by the aircraft VNAV flight control mode. Relative information were collected and will be analyzed next.

5.2.3 Data Acquisition and Performance Analysis

During the test, essential data were recorded by using the data acquisition functionality of the GUI. The analysis was performed online by using the GUI and offline by using the Microsoft Excel tool, as described next.

5.2.3.1 Data Acquisition

From the GUI, any data can be recorded into a file for any duration in a frequency no more than 30 Hz. Any number of parameters from the list box of the GUI (Figure 15) can be selected for recording.

By plotting the data on the screen when running the test bed, the user can monitor performance of the test bed and the FMS in run time. Some data were collected and analyzed using Microsoft Excel, as described in the following sections.

5.2.3.2 LNAV Mode

From the recorded data file, GPS latitude data can be plotted with respect to GPS longitude data in order to draw an actual flight route on the earth's surface. The route flown by the simulation is illustrated in Figure 30. The departure portion, which is the circled portion of the curve, is zoomed-in in Figure 31. As can be seen in the figure, the aircraft toke-off heading east, and then made a smooth right turn to heading 243 degrees, which is the target heading from CYMX to AGBEK.

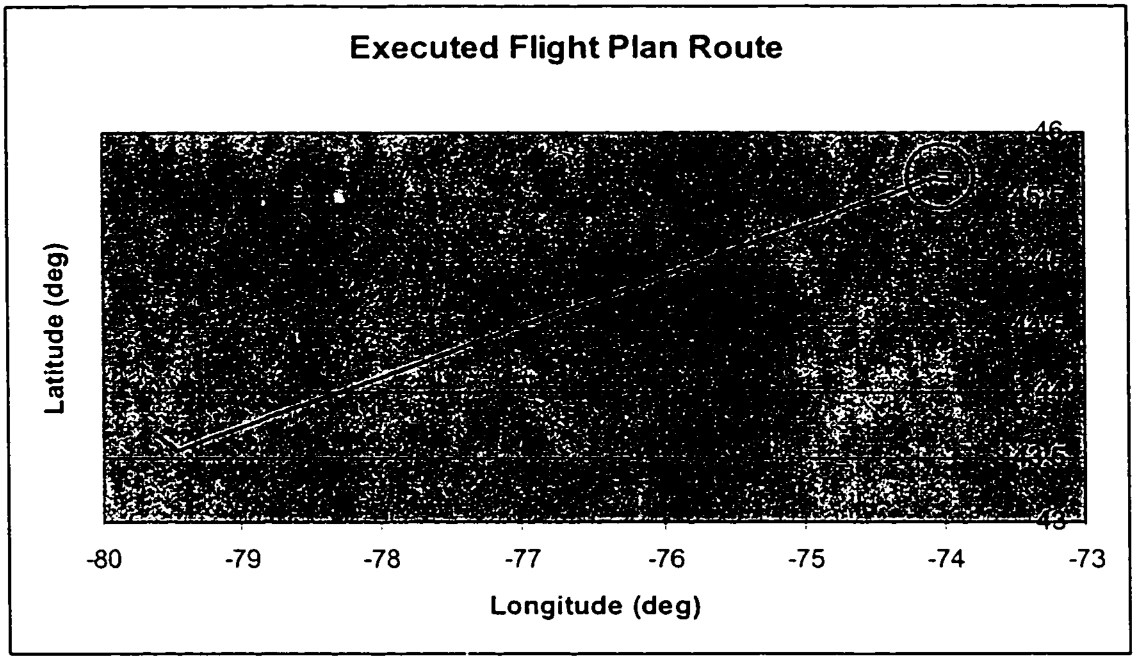

Figure 30 Lateral Profile

When the aircraft returned to wings level flight heading 243 degrees and climbed to 4,000 ft, it was repositioned to AGBEK at 3,000 ft above sea level, airspeed 170 knots, heading 317 degree that is the target runway heading at CYYZ airport. In Figure 30, the point where the aircraft was repositioned is connected by a straight line to the point where the aircraft was before the reposition happened. After reposition, the aircraft was flying again by the FMS heading the selected runway.

This test demonstrated the FMS LVAN function as well as the LNAV flight control mode developed in this work.

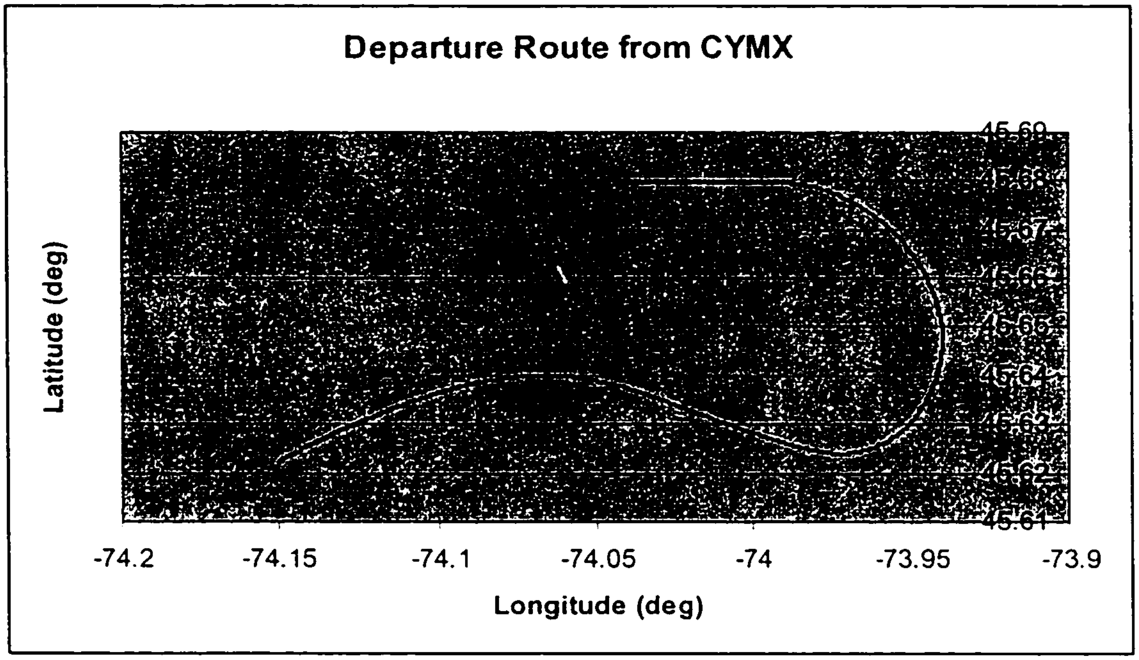

Figure 31 From Take-off to En-route

5.2.3.3 Level Change Mode

The level change mode has been tested in Chapter 4. Figure 20 through 22 have shown that the aircraft could smoothly change the altitude by flying the level change mode. Redundant analysis will not be given here.

5.2.3.4 Real-Time Performance

By definition, a system is said to be a real-time system when it carries out all its activities respecting timing constraints. The important interaction with the environment brings out several types of timing constraints, some of which are severe, and have to be respected:

- High frequency for actions incoming from the environment—event frequency, data transfer rate, etc.;
- High frequency for actions to be undertaken on the environment, for example high frequency for the control of a fast response time process;
- A maximum limit for reaction times between the moment at which an event appears and the moment at which the resulting action is completed.

To test and evaluate the FMS that is used in the real aircraft, the real-time performance is an important criteria to the system design. Some signals are very real-time critical. For example, during approaching phase of flight, the aircraft position information (including latitude, longitude and altitude) is required to be transmitted very accurately and on time because they are used by the FMS to calculate the glide slope vertical deviation. A small error of the latitude or longitude signal can result in an unacceptable error in the vertical deviation and therefore the FMS can not be tested properly.

The real-time performance was evaluated by recording simulation data from both inside the simulation program and immediately from the ARINC I/O running on the interface PC. It was shown that data were transmitted from the simulation software to the ARINC thread in exactly 30 Hz and the time delay was negligible. The vertical deviation value generated by the FMS was smooth except small fluctuations due to the resolution of the FMS.

5.2.3.5 Multiple Level Wind Model and Environment Model

As necessary functions of the dynamic test bed, the multiple level wind model and robust environment model were tested in another run on the same flight plan. During this test, wind direction and velocity, mean sea level temperature and pressure were set by the user and were recorded into a file. The DTB and FMS performance information including the track angle, true heading, steering command, barometric altitude and baro-corrected altitude were also recorded.

The wind set by the user does have a vertical division. It is presented by its X and Y division. The true heading, track angle and steering command (in degree) are plotted in Figure 32. The aircraft was holding an altitude of 3000ft under altitude hold mode and flying true north under head hold mode. This chart illustrates the system response when the wind changed from 0 knots to 30 knots true east. From this chart, the effectiveness of the wind model is obvious in testing the dynamic response of the DTB and FMS.

The MSL temperature and pressure were varied from the GUI. In a test, the aircraft was holding a baro-corrected altitude of 3000ft. The MSL temperature was changed from +15°C to -30°C. Figure 33 illustrates how this MSL temperature variation influences the barometric altitude by comparing with the baro-corrected altitude.

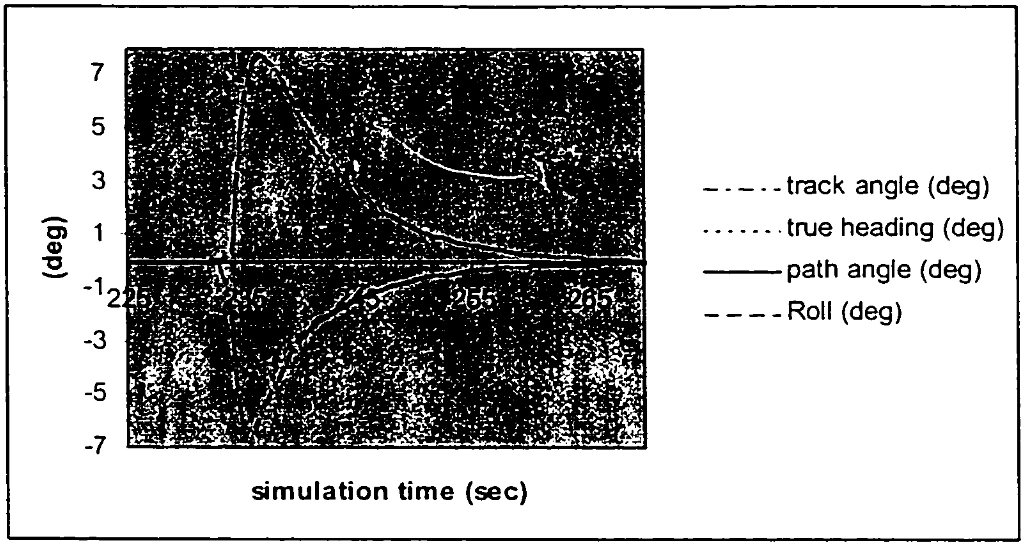

Figure 32 AFCS Response to Inserted Wind (wind: 30 knots east)

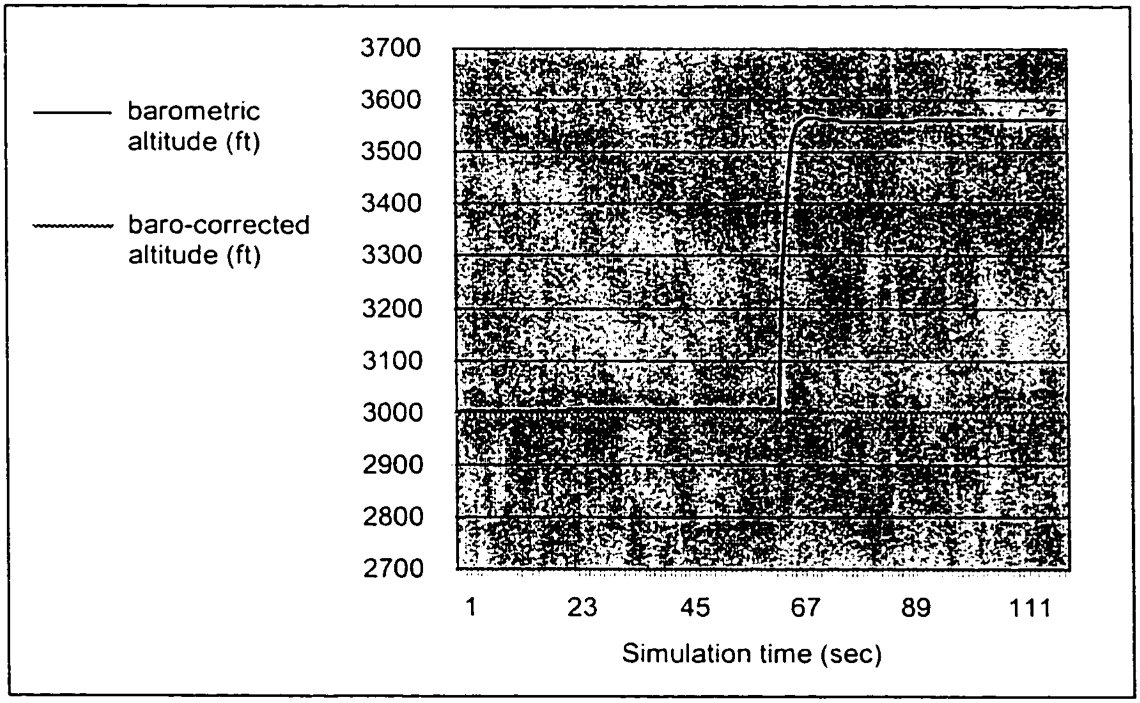

Figure 33 Barometric Altitude vs Baro-Correctdd Altitude under Altitude Hold Mode when MSL Temperature changed from +15°C to -30°C

5.3 Summary

Functions developed in this thesis have been tested in this chapter. Main functions developed in the flight simulation software include the AFCS flight control modes, reposition function, navigation models (ADC, GPS and IRS), wind and environment models, and data acquisition functions. The successful implementations of these functions do not relay only on the successful development of themselves, but also on the proper interface of the FMS. The test and analysis done in this chapter has illustrated both the successful development of the flight simulation software system, the FMS software interface system and the graphical user interface system proposed in Chapter 1 and 2.

6 SUMMARY AND CONCLUSIONS

This thesis has successfully proposed and developed a dynamic test bed for a modern flight management system. The FMS and its development are introduced in the beginning of Chapter 1. Then the dynamic test bed is proposed in the end of Chapter 1 and throughout Chapter 2. The development and integration of the dynamic test bed are described in detail in Chapters 3, 4 and 5. DTB system complete tests are performed and analyzed in Chapter5. This chapter presents review of the design and discuss necessary follow-up development work on the DTB.

6.1 Review and Discussions

The DTB proposed and developed in this thesis provides analog/discrete and ARINC 429 interface to stimulate the FMS, and provides functionality to test the FMS lateral navigation function. The DTB has been tested extensively and is effectively used in the company.

6.1.1 FMS Software Interface System

FMS interface included the ARINC 429 and analog/discrete interface. The most important interface used by the FMS is the ARINC 429. All navigation data are transmitted in ARINC 429 format. Minor information such as reposition command is transmitted as analog/discrete signals. The FMS software interface system includes the drivers for the ARINC 429 and the analog/discrete interface cards.

In addition to supporting the I/O operations of the interface cards, the interface system sends and receives data to and from the simulation PC. These operations were mapped into separate process paths by using multithreaded programming technology. This arrangement took better advantage of the computer resources and helped for better real time data transmission. These processes were properly synchronized by an executive thread, between which a handshake was established through Ethernet link. This handshake enables the flight simulation process to indirectly control the operations of the interface between the FMS and the simulation process itself. The interface processes to the FMS were idle and consumed little CPU resource if the Ethernet communication was not done yet. They were woken up every time when and only when the Ethernet communication with the simulation was done. This ensures accurate and faster data transmission between the simulation software and the FMS.

6.1.2 Graphical User Interface

The graphical user interface was developed using Microsoft Visual C++ tool. It provided interface features such as buttons and dialog boxes giving users the complete ability to run the simulation, control the aircraft, test the FMS and perform data acquisition, which is one of the most important functions of the DTB. The graphical presentation is easy to understand and user friendly.

6.1.3 Aircraft Flight Simulation System

The flight simulation software system did not need to be built from scratch since some high quality commercial flight simulation software packages are available. The DTB flight simulation software was developed based on VPI FLSIM v7.0 flight simulation software package. This software package was enhanced by developing the level change control law, glide slope control law, multiple level wind model and robust environment models. Aircraft automatic flight control modes needed to support the test of the FMS were developed. These modes include heading mode, LNAV mode, altitude hold mode, altitude pre-select mode, vertical speed mode, and level change mode. These modes were flown by the autopilot to best follow the navigation commands from the FMS and the user. FMS navigation resources, including the GPS, ADC, AHRS and IRS were simulated.

6.1.4 Ethernet Link

Ethernet link has provided fast way of communication for the test bed. The communication between two PCs and between the simulation software and the graphical user interface were all through Ethernet link. A communication software supporting TCP/IP was developed and integrated into its user software systems for this purpose. Ethernet connections between GUI and simulation software and between simulation software and interface software are robust.

6.2 Future Work

The architecture/structure, hardware and software configuration of the dynamic test bed depend on the functions and interface requirements of the FMS. As described in previous sections, a basic structure of the proposed dynamic test bed has been established and many functions have been developed through the work of this thesis. Depending on the FMS, there still remain a number of aspects to be addressed as future expansion and development.

6.2.1 VNAV Mode

As described in Section 1.1.4.2.2, VNAV is another primary function of the FMS. Development of the VNAV mode will include the development of a new control law, which is the glide slope control law. The glide slope control law is used to generate the command pitch angle from the glide slope vertical deviation calculated by the FMS.

6.2.2 Auto Landing Model

While the auto take-off model and necessary automatic flight control modes have been developed, the auto landing model is required to complete VNAV test. From glide slope to touching down requires critical FMS VNAV performance. Without the auto landing model, this can not be tested accurately. The development of the auto landing model will involve many modifications and to the FLSIM v7.0 software package as well as new development.

6.2.3 Radio Navigation Sensors Simulation

Radio navigation sensors, such as VOR/DME, are important navigation resources for the FMS. This thesis has simulated the ADC, GPS, and IRS. The radio navigation sensors are yet to be simulated. This simulation will need the navigation database installed and downloaded when running the simulation. Development of software modules might not be necessary since the FLSIM v7.0 already has these simulation.

6.2.4 Malfunctions Simulation of Navigation Sensors

The FMS has high priority and low priority navigation sensors. It automatically switches to the next highest sensor if one sensor became failure. In order to test this function of the FMS, malfunctions of the sensors should be simulated. The FLSIM v7.0 has provided the malfunctions for the ADC, VOR and DME. The IRS and GPS need to be developed.

6.2.5 Aircraft Model Library

At present the aircraft simulated is the Boeing 747-400. Other aircraft models can be collected and stored into an aircraft library. This is necessary since the FMS can be used on different aircraft types. An aircraft library containing different aircraft models will enable the DTB to download and simulate different aircraft types.

6.3 Final Words

The DTB has been successfully developed and is now successfully used in CMC. The DTB has been certified by Transport Canada as a main lab tool used for system verification for FMS software release. The DTB will be the company's FMS System Verification Tool for the FANS (Future Air Navigation System) [8] software release.

However, the current version of the DTB does not have complete functionality to test all functions of the FMS. It can only test the LNAV function of the FMS. Other functions of the FMS include VNAV, fuel performance, etc., as described in Chapter 1. These functions also need to be tested extensively during the development. Furthermore, new functions are being integrated into the FMS as the result of new development of both technology and aviation market. From the interface point of view, the DTB currently contains only ARINC 429 and analog/discrete interfaces. In addition to ARINC 429, other digital interface systems, such as ASCB [38] and CSDB [39], are also used by modern FMS. Finally, the DTB is currently limited to fixed-wing aircraft type. Modern FMS are used not only on fixed-wing aircraft, but on helicopters as well. Therefore, it will be beneficiary for both CMC and Concordia to further enhance and develop the DTB on (but not limited to) the above-discussed aspects.

REFERENCES

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

[2] Samuel B. Fishbein, "Flight management systems : the evolution of avionics and navigation technology", Westport, Conn. : Praeger, 1995

[3] E.Pallett, "Aircraft Instruments & Integrated Systems", Longman Group UK Limited, 1992

[4] "Flight Management System (FMS)" http://www.cas.honeywell.com/ats/products/fms/sysdes.html

[5] Brian Kendal, "Manual of Avionics", Blackwell Scientific Publications, 1993

[6] J. Roskam, "Airplane Flight Dynamics and Automatic Flight Controls", Roskam Aviation and Engineering Corporation, Ottawa, 1994

[7] "Airborne VOR Receiver", Arinc Characteristic 579-2, Aeronautical Radion Inc., 1989

[8] Jim Curran, "Trends in Advanced Avionics", Iowa State University Press/Ames, 1992

[9] "Avionics Fundamentals", International Standard Book Number 0-89100-293-6

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

[11] "DAQ PCI-6023E/6024E/6025E User Manual", National Instruments, October 1998 Edition

[12] Cary R. Spitzer, "Digital Avionics Systems Principles and Practices", 2nd Edition, McGraw-Hill, Inc., 1993

[13] "ARINC 429 Reference Manual", SBS Avionics Technologies

[14] "From the Ground Up", 25th ed., Aviation Publishers, Ottawa, 1987

[15] Brian L. Stevens, Frank L. Lewis, "Aircraft control and simulation", New York: Wiley, c1992

[16] Cook, M. V., "Flight dynamics principles", London : Arnold ; New York : J. Wiley, 1997

[17] "Programmer's Guide", FLSIM Version 7.0, Virtual Prototypes Inc. 1998

[18] Dr. Albert Helfrick "Principles of Avionics" 2nd Ed.

[19] Pete Loshin, "TCP/IP clearly explained", 2nd ed, Boston : AP Professional, c1997

[20] Dr. J. Svoboda, S. Lasnier, G.M. Mckinnon, "Microprocessor-Bases Light Aircraft Simulator", Canadian Industrial Computer Society, Ottawa, 1986

[21] Rolfe and Staples, "Flight Simulation", Cambridge University Press, New York, 1986

[22] Brian L. Stevens, Frank L. Lewis, "Aircraft control and simulation", New York : Wiley, c1992

[23] P. Garrison, "Autopilots, Flight Directors and Flight Control Systems", 1st ed., Tab, Blue Ridge Summit, 1985

[24] Ernest E. Sechler and Louis G. Dunn, "Airplane structural analysis and design", New York : Dover Publications, 1963

[25] David J. Kruglinski, Scot Wingo, and George Shepherd, "Microsoft Visual C++", 5th Edition, Microsoft Press, 1998

[26] "Flight Simulation/Simulators", Society of Automotive Engineers, October 1985

[27] Hassan Diab, "Design and Implementation of a Flight Simulation System", American University of Beirut

[28] Mark E. Connely, "Simulation of Aircraft", M.I.T. Report, Massechusetts Institute of Technology, 1958

[29] Global positioning system, Washington, D.C.: The Institute of Navigation, c1980-1986

[30] A. P. L. A. Marsman, "Flexible and High Quality Software on a Multi Processor Computer System Controlling a Research Flight Simulator", National Aerospace Laboratory (NLR), The Netherlands

[31] ARINC Specification 429: Mark 33 Digital Information Transfer System, Aeronautical Radio, Inc., 1977

[32] D McRuer, I Ashkenas, D Graham, "Aircraft Dynamics and Automatic Control", Princeton University Press, 1973

[33] L. J. Clancy, "Aerodynamics", New York : Wiley, 1975

[34] Louis V. Schmidt, "Introduction to Aircraft Flight Dynamics", Naval Postgraduate School, 1998.

[35] E. Pallett, "Aircraft Instruments", 2nd ed., Pitman Publishing, Marshfield, 1982

[36] Peter Lawn, "The Enhancement of a Flight Simulator System with Teaching and Research Applications", 1998.

[37] Martin Fowler with Kendall Scott, "UML distilled : applying the standard object modeling language", Addison Wesley Longman, 1997

[38] ASCB – Applications Specification – Version B Bus Users, Honeywell, Rev. P, CAE Spec EB7013682

[39] Commercial Standard Digital Bus (CSDB) , Collins General Aviation Division, 7th Edition, March 1988

APPENDIX A: DTB ARINC 429 BUS CONFIGURATION

Device No.	Channel No.	Direction	Speed
1	1	R*	FAST
	2	R	FAST
	3	T**	SLOW
	4	T	SLOW
	5	T	SLOW
	6	T	SLOW
	7	T	SLOW
	8	T	SLOW
2	1	R	FAST
	2	R	FAST
	3	T	SLOW
	4	T	SLOW
	5	T	SLOW
	6	T	SLOW
	7	T	SLOW
	8	T	SLOW

* R denotes Receiver

**T denotes Transmitter.

APPENDIX B: A429 LABELS TRANSMISSION SPEED AND CHANNEL ASSIGNMENT

	ARINC 429 Label	Word Name	Transmission interval (ms)	ARINC 429 Channel
GPS	076	GPS Altitude Word	1000	Device 1, Channel 6,7,8
	101	GPS HDOP Word 1	1000	
	102	GPS HDOP Word 2	1000	
	103	GPS Track Angle Word	1000	
	110	Latitude Word	1000	
	111	Longitude Word	1000	
	112	Ground Speed	1000	
	120	Fine Latitude Word	1000	
	121	Fine Longitude Word	1000	
	130	GPS HIL Word	1000	
	133	GPS VIL Word	1000	
	136	GPS VFOM Word	1000	
	150	UTC Word	1000	
	165	Vertical Velocity	1000	
	247	GPS HFOM Word	1000	
	260	Date Word	1000	
	273	GPS Sensor Status	1000	
ADC	203	Pressure Altitude	450	Device2, Channel 4,5
	204	Baro Corrected Altitude	450	
	210	True Air Speed	100	

	ARINC 429 Label	Word Name	Transmission interval (ms)	ARINC 429 Channel
IRS	072	Inertial Latitude	125	Device 1, Channel 3,4,5
	073	Inertial Longitude	125	
	126	Time In NAV	640	
	270	IR Discrete Status #1	320	
	271	IR Discrete Status #2	320	
	310	Present Position Latitude	125	
	311	Present Position Longitude	125	
	312	Ground Speed	31.25	
	313	Track Angel True	31.25	
	314	True Heading	31.25	
	317	Track Angel Mag	31.25	
	320	Magnetic Heading	31.25	
	321	Drift Angel	31.25	
	322	Flight Path Angel	31.25	
	324	Pitch Angel	15.625	
	325	Roll Angel	15.625	
	350	IR Maintenance Word	320	
	361	Inertial Altitude	31.25	
	365	Inertial Vertical Speed	31.25	
	366	North-South Velocity	62.5	
	367	East-West Velocity	62.5	
AHRS	314	True Heading	50	Device 1, Channel 3, 4, 5
	320	Magnetic Heading	50	

	ARINC 429 Label	Word Name	Transmission interval (ms)	ARINC 429 Channel
DME	035	DME Frenquency	135	Device 2, Channel 7
	202	DME Distance	135	
VOR	034	VOR/ILS Frequency	200	Device 2, Channel 6
	222	VOR Bearing	200	

Note: 1. The DME and VOR labels are not simulated in this thesis.

2. The maximum transmission speed in this thesis is 33 ms. All the labels that require higher transmission speed are transmitted at 33 ms.