Report Number: GAPATS/PROGRESS/972501-1

General Aviation Pilot Advisory and Training System (GAPATS)

Semiannual Technical Progress Report

Period: 1/26/97 through 7/25/97

Prepared by:
Dr. John Painter
Dr. Donald T. Ward
Dr. Wallace Kelly
Mr. John W. Crump
Mr. Ron Phillips
Major Jeff Trang
Mr. Kris Lee
Mr. Paul A. Branham
Mr. Karthik Krishnamurthy
Mr. William P. Alcorn, Jr.
Mr. Andrew P. Robbins
Mr. Ren-Jye Yu

Knowledge Based Systems, Inc.
One KBSI Place
1500 University Drive East
College Station, Texas 77840

September 16, 1997

Prepared for:
National Aeronautics and Space Administration (NASA)
NASA Langley Research Center
9A Langley Boulevard, Bldg. 1195B, Room 123
Hampton, VA 23681-0001

Contents

List of Figures

List of Tables

1. Introduction

The goal of this project is to achieve a validated General Aviation Pilot Advisor and Training System (GAPATS) engineering prototype, implemented according to commercial software standards and Federal Aviation Administration (FAA) issues of certification. Phase II builds on progress during Phase I, which exceeded proposed objectives. The basic technology has been transferred from previous NASA research (1989 to 1994). We anticipate a commercially licensable prototype, validated by pilots in a flight simulator and in a light twin-engine research aircraft for FAA certification, by January 1998.

1.1 Background

The General Aviation Pilot Advisor and Training System (GAPATS) is a computerized airborne expert system being developed by Knowledge Based Systems, Inc. (KBSI) and Texas A&M University (TAMU). This system uses fuzzy logic to infer the flight mode of an aircraft from sensed flight parameters; this inference, along with an embedded knowledge base and pilot inputs, is then used to assess the pilot's flying performance and issue recommendations for pilot actions. Such a system improves safety by enhancing the pilot's situational awareness and by reducing the cost and time required to achieve and maintain pilot proficiency. This effort is aimed specifically at the General Aviation market, in which increased safety and utility are highly desired.

KBSI and TAMU are now in Phase II of this NASA-funded project. The TAMU work effort is also jointly funded by the State of Texas's Advanced Technology Project. The goal of this phase is to develop and validate a commercially acceptable engineering prototype of the system, preparatory to FAA certification in the next phase of the project. Development of GAPATS centers on a fixed-base Engineering Flight Simulator (EFS) in the Flight Simulation Laboratory of the Aerospace Engineering Department of TAMU. Validation will be conducted initially on the EFS and later in a light twin-engine research aircraft belonging to TAMU and operated out of the Flight Mechanics Laboratory of the Aerospace Engineering Department.

1.2 Purpose

The purpose of this project is to develop and integrate the technology necessary to provide any general aviation pilot with increased situational awareness. GAPATS employs artificial intelligence to determine flight operations performed by the pilot and aircraft. Based on AI algorithms and pilot input, the system provides the pilot with a critique of present performance and procedural advice, without adding to pilot workload but increasing safety, efficiency, and operational precision. GAPATS employs the most modern software engineering techniques: object-oriented design/programming, parallel software architectures, and fuzzy logic.

1.3 Scope

This project concentrates on the following objectives:

1. Completing Flight Mode Interpreter (FMI) implementation;

2. Implementing a complete, FMI-integrated Pilot Advisor (PA);
3. Implementing FMI-switched Head-Up (HUD) and Head-Down (HDD) displays;
4. Enabling computer-communication between GAPATS PC-based software modules and the Electronic Flight Simulator (EFS) through TCP/IP communication software;
5. Installing aircraft sensors; and
6. Writing and validating display driver software in flight simulator and research aircraft.

This report documents the work performed in upgrading the simulator and the aircraft to achieve the Phase II goals during the period January 1997 through July 1997—the third six-month period of contract performance.

1.4 Acronyms and Definitions

Table 1 provides a list of acronyms and their corresponding definitions used in this report.

Table 1. Acronyms and Definitions

Acronym	Definition
CLIPS	C Language Integrated Production System
DAFS	Donated Artificial Feel System
EFS	Engineering Flight Simulator
FAA	Federal Aviation Administration
FMI	Flight Mode Interpreter (software module)
GAPATS	General Aviation Pilot Advisor and Training System
GPS	Global Positioning System
KBSI	Knowledge Based Systems, Inc.
HDD	Head-Down Display (hardware and software modules)
HUD	Head-Up Display (hardware and software modules)
NAV	Navigation and Flight Director Module (software)
PA	Pilot Advisor (software module)
STTR	Small Business Technology Transfer Pilot Program

1.5 Overview

The body of the report consists of three sections:

1. *Description of Progress for This Period*—a quantitative description of work performed during the current period;
2. *Current Challenges*—a description of current problems which may impede performance, schedule, or cost along with proposed corrective actions for each; and

3. *Planned Efforts for Next Period*—a discussion of the work to be performed during the next reporting period.

2. Description of Progress for This Period

This section contains a quantitative description of work performed during this period.

2.1 Project Management

The project management tasks are summarized in the following sections.

2.1.1 Configuration Management

As more people are required to work together on the source code, configuration management has become more critical. Team members have worked independently in the past and now have had to change that paradigm to one that supports the integration of the various software components. To this end we have been using a web-based configuration management system to allow the check-in and checkout of the source code.

2.1.2 Team Coordination

Many excellent TAMU students have participated in the project, giving the project enormous leverage with respect to talent per dollar.

The team has been very effective at integrating new members when they join the project. However, some of these students have graduated, taking their GAPATS experience and expertise into the U.S. (and global) workforce. It has been a real challenge to continue this training effort as experienced people leave and new people join the program.

Thanks to the effort of several people from TAMU, we have managed to keep everyone informed about the project. To this end, we have started a GAPATS web page (http://www.joshua.tamu.edu/astras) with project information and a central mailing list so that team members can easily communicate with others. We also meet weekly to discuss our progress and any issues that have arisen. These meetings will continue throughout this effort.

2.1.3 Critical Design Review

On March 22nd, 1997, the entire design and implementation team prepared and presented a comprehensive GAPATS Critical Design Review (CDR). The team at that time consisted of thirteen graduate engineers, including three professionals from KBSI, three MS degree candidates, and two PhD degree candidates from each of the TAMU Aerospace Engineering and Electrical Engineering Departments. The purpose of the CDR was two-fold: first, to provide visibility to each of the individual designers of the total system design and implementation, from top to bottom; and second, to expose all possible system integration problems. Both purposes were admirably satisfied. The CDR took eight hours, using a total of one hundred four computer generated and displayed Microsoft® PowerPoint™ slides.

As a result of the CDR, we set a system integration goal date of June 1st, 1997, for a "First Credible Demonstration" in the Electronic Flight Simulator (EFS) for the "all-up" GAPATS system. This demonstration would utilize all of the integrated GAPATS software modules, including FMI, PA, HUD, HDD, NAV, and Data Object. It would employ the hardware of the upgraded EFS. The word "credible" would mean that simulated flying operations in the demonstration area (Waco, Texas) would be credible to skilled pilots. In other words, each module would provide sufficient correct functionality to enable a pilot to exercise all of the GAPATS demonstration flight modes such as Taxi, Takeoff, Climbout, Cruise, Initial Approach, Final Approach, and Landing.

The goal date was missed due to computer communication problems, detailed in Section 2.8.2.2. The actual credible demo occurred on August 21st. However, in response to having set a goal date, the team was so focused in solving the system integration issues that, between March and June, so many man-hours were expended that the system was, in fact, integrated before some of the engineers graduated.

2.2 System Operations Requirements

The GAPATS system operational requirements were extensively analyzed and documented, maintaining the focus of system definition and development from a pilot's perspective. Documentation appeared as a Master of Science thesis by Major J. Trang, USA. These requirements were immediately applied to a number of areas, such as system software architecture (Section 2.3), the Flight Mode Interpreter (Section 2.4), the Pilot Advisor (Section 2.5), the Navigation Module (Section 2.6), the HDD (Section 2.7), and the HUD (Section 2.8).

The system software architecture entails passing data objects between the various subsystems within GAPATS. Because of the object-oriented approach, each module can add/change/read only those parameters it requires, ensuring that data will not be corrupted as it passes from module to module.

The Pilot Advisor system alarms previously appearing in pseudo-code were written using CLIPS for prototype display. These CLIPS rules were passed to KBSI.

Input/output requirements for the Navigation Module were finalized. These requirements, implemented in the Navigation Module by K. Lee, include utilization of an aeronautical database supplied by Jeppesen. Also, provision has been made for utilizing GPS data. Both the HDD and HUD configurations were finalized by P. Branham and R. J. Yu, respectively. These configurations included provisions for unique display sets for each flight mode using "automatic mode switching" as defined by Trang.

A potential source for procurement of a HUD was found at NASA-Ames Research Center. The HUD has been removed from a U.S. Army AH-1 Cobra helicopter. We have received tentative permission to use the HUD in the EFS. Justification for using the HUD as a research tool was sent to Ames, and formal approval is expected soon.

2.3 Software Architecture

The refined software architecture for the GAPATS prototype improves the interface between the software modules. Software modules are under development in both the electrical and aerospace engineering departments of TAMU as well as at KBSI. The distributed nature of this project's development has motivated a strong commitment to object-oriented design and modularity.

The extended design is based on a single data object that coordinates the communication between modules. Figure 1 illustrates the design. Each module has exclusive write access to the values it is responsible for updating, preserving the encapsulation benefits of an object-oriented design. All modules have read access to all the information in the data structure.

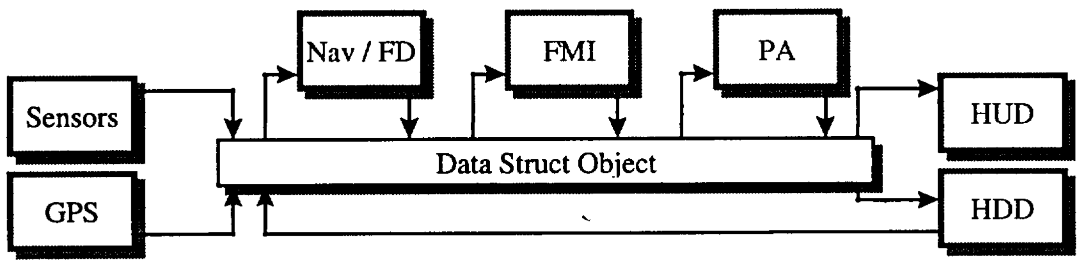

Figure 1. GAPATS Software Architecture

The traditional object-oriented approach allows the various modules to communicate directly. However, the GAPATS team feels that the extended design is appropriate for our requirements and better facilitates a distributed, parallel implementation process.

The data interface object is implemented in C++. The Flight Mode Interpreter was converted to use the new design. A C++ class was written to implement the Sensors object for recorded flight data. The other objects are currently being adapted to the new design.

2.4 Flight Mode Interpreter

The major effort regarding the Flight Mode Interpreter during this reporting period was final implementation of design changes and additions. Final performance was measured using recorded data from nearly two dozen simulated flights. The suite of MATLAB© functions for designing the Flight Mode Interpreter assisted Dr. W. Kelly and Trang in finalizing the design of the fuzzy membership functions. The final design is based on a new approach that allows for more variation in pilot flying style. It also incorporates the use of distance computations from various waypoints, such as the Initial Approach Fix (IAF), Final Approach Fix (FAF) and the Missed Approach Point (MAP).

A result of Trang's design, as documented in his MS thesis, was that Fuzzy Membership Functions could be designed, based on aircraft distance to various waypoints, such as IAF, FAF and MAP. FMI use of these Membership Functions then dramatically increased the performance of the FMI for identifying flight modes, such as Initial Approach, Final Approach, and Landing. Membership Function design for Trang's original concept was airport-dependent. However, Kelly improved Trang's scheme, making Membership Function design independent of the airport being approached. Thus, one set of Membership Functions serves for approaches to any airport, based on a one-time computation of certain distance ratios between known geographical points for each airport. This design improvement is documented in Kelly's PhD dissertation.

Final design and implementation of the Flight Mode Interpreter now employ nine rules, with corresponding Membership Functions, using variables either directly sensed or computed in the Navigation Module. These rules are Engine Power, Roll Angle, Airspeed, Altitude (AGL) (computed above terrain), Rate of Climb, and four distance ratios. The AGL altitude computation employs raw sensed altitude (MSL), GPS position, and geographical database (Jeppesen). The four distance ratios are computed from five measured and/or computed distances: aircraft distance to FAF, distance to MAP, distance between IAF and FAF, distance between FAF and MAP, and distance from MAP to airport. See Trang's thesis and Kelly's dissertation for details.

Flight mode decision filtering has been retained to further improve performance. There is an inherent trade-off between the amount of smoothing and the time delay caused by the filtering. A smoothing interval of several seconds, implemented in the final design, results in an acceptable several-second delay in FMI transition from one mode indication to another.

2.5 Pilot Advisor

A prototype expert system of Pilot Advisor (PA) is complete. The PA uses a CLIPS rule base to verify that interaction between the airplane and the pilot is seamless. The output of the PA displays various types of symbology sets and alarms on the HUD and HDD. We have been able to encode a prototype PA rule base, test rules in a CLIPS environment, and integrate the rule base with the flight simulator. We plan to conduct more thorough testing and fine-tuning later.

2.5.1 Integration with the Flight Simulator

The system had been restructured to use a global data object structure for storing and retrieving information to facilitate the use of thread and interprocess communication. After this change was made, the previous version was retired and the various modules combined by integrating the CLIPS engine into the new application. The code for CLIPS was isolated and integrated into the new application to provide rules-based pilot advisor messages. All references to the global data in the application had to be modified to read in the new global data object. In addition to incorporating the CLIPS rules into the GAPATS application, the project team determined that a priority queue for holding the pilot alerts was needed. A priority queue class was designed and written to keep track of all alerts sent from the CLIPS engine to the pilot. Accessors were added to allow the HDD to view these alert messages and display them to the pilot. There are plans for implementing an alert history into the system to keep track of all alert messages coming from CLIPS. The messages currently consist of the alert string, the level of the alert, a salience level, and a time stamp.

The introduction of the new data object mandated collection and review of all of the TCP/IP structures and classes in the earlier version of GAPATS. After all required class definitions and data structures were located and integrated into the new GAPATS executable, changes were made as needed to use data object instead of global data structures. Modifications corrected the numeric values because of the differences in architectures between the client and server.

Differences between classes and the existing TCP/IP code were resolved, resulting in a TCP/IP link that successfully transports the raw data from the simulator to the client and returns the information from the client to the simulator.

All data files were changed to text format, so the testing software which simulated the simulator was modified to read in the new data. This application is basically a server that plays data files recorded by the simulator, allowing us to fly the exact same flights over and over to fine-tune the rules. The routines that imported the data were modified, and the data structures throughout the application were changed to accept the new data for the simulator, the GAPATS client, and the data file server test application. Further modifications to the server application included writing functions to display the HUD data in an effort to emulate the data displayed by the simulator on the viewing screen and ensure that the correct data will return to the server.

2.6 Navigation Module

During this reporting period, work on the Navigation Module was continued, and the first stage of coding was completed. Coding during this period concentrated on the functions performed by the Navigation Module which do not require interfacing with the navigation database. In Figure 2**Error! Reference source not found.**, the majority of these functions are listed in the Navigation Module box. The box lists only those items required by other software modules in the system. In addition to these, the Navigation Module has the capability to provide distance, course, and time information to any location defined in the flight plan.

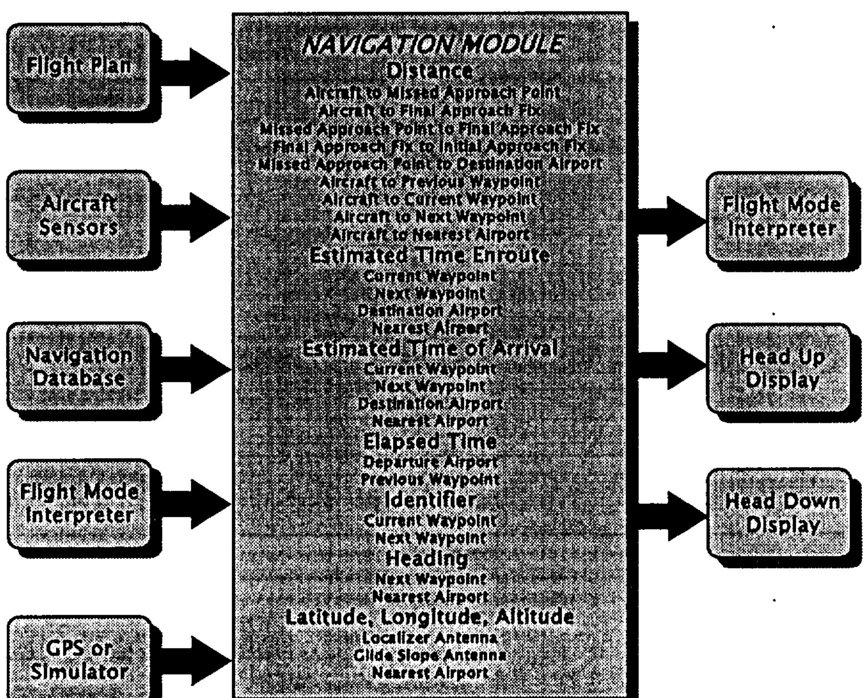

Figure 2. Navigation Module

The input/output relationship between the Navigation Module and the rest of the system is also shown in Figure 2. In general, data are received from other system components and processed through several navigation algorithms. The results of these calculations are available to the remaining software modules. Currently, the Flight Mode Interpreter, Head-Up Display, and Head-Down Display receive information directly from the Navigation Module.

Also during this period, the C++ class representing the flight plan was coded. The main element of the flight plan class is stored in memory as a linked list of objects that correspond to points the pilot entered as the flight route. During the flight, the Navigation Module continuously monitors the airplane's location. As the airplane passes the points entered in the flight plan, the Navigation Module automatically advances through the stored list of points. Thus, the navigation calculations performed relative to the "current" or "next" point in the flight plan are automatically updated as the flight progresses.

To design and test many of the functions shown in Figure 2, the flight plan class was created with a default route from the Waco Regional airport to the TSTC-Waco airport. This scenario will allow for evaluation of Navigation Module performance during the initial integration and testing of GAPATS.

Additionally, we purchased the ARINC 424 Navigation System Data Base specification document which was then used in preliminary identification of the records contained in the navigation database provided by Jeppesen Sanderson Inc.

2.7 Head-Down Display (HDD)

As pointed out in the last progress report, the HDD serves two purposes: (1) it communicates GAPATS' advisories to the pilot and (2) it provides backup instruments for use when the HUD is not available and to augment the HUD data.

The first function was implemented during this reporting period using the HDD selector switches surrounding the display (Figure 3 and Figure 4). The pilot selects an information group menu, which may include subpages. Using Borland's Object Windows Library, P. Borland has completed the basic software framework for this part of the HDD. The displays are based on Trang's HDD design. At the suggestion of another pilot, Trang's original design was modified so that the HDD will have six selector switches on each side; the rotary knobs have been eliminated. The functionality for navigating through the displays is complete. Also, having decided that text entry is unnecessary, we developed a means for numeric data entry through a keypad. Instead of text, we will rely on item selection boxes, currently under development. The HDD has been integrated with the data object. Functionality has thus been added to many of the displays. Below is the status for each of the seven displays:

- NAV—Information such as current flight plan, flight mode, airspeed, wind speed and direction, and heading can now be viewed using the NAV display mode. Also, a rudimentary moving-map is now available, showing airports, intersections, and NDSs (non-directional radio beacons). The map can be scaled using the SCALE UP and SCALE DOWN buttons. Figure 3 illustrates the NAV display mode. Addition of linear objects such as airways and runways is currently in progress.

- FLT PLN—Data entry is now possible for numeric fields. Work continues on selection boxes that will enable text entry.
- WT/BAL—This mode is now fully functional. All that remains is to include actual moment arm and weight data from the Commander 700 manual.
- CHK LSTS—The abbreviated checklists are now viewable in this mode. A somewhat more intricate design will be necessary to view Emergency and Detailed checklists because of their much greater length.
- TRNG—This mode is partially functional. The FMI and sensor values are now operational. The navigator and pilot advisor information still needs to be integrated. Also, the Record function remains to be developed.
- BIT—This mode can now be used to view the status of all GAPATS modules.
- DISPLAYS—This mode, recently added to the GAPATS design by Trang, has yet to be developed.

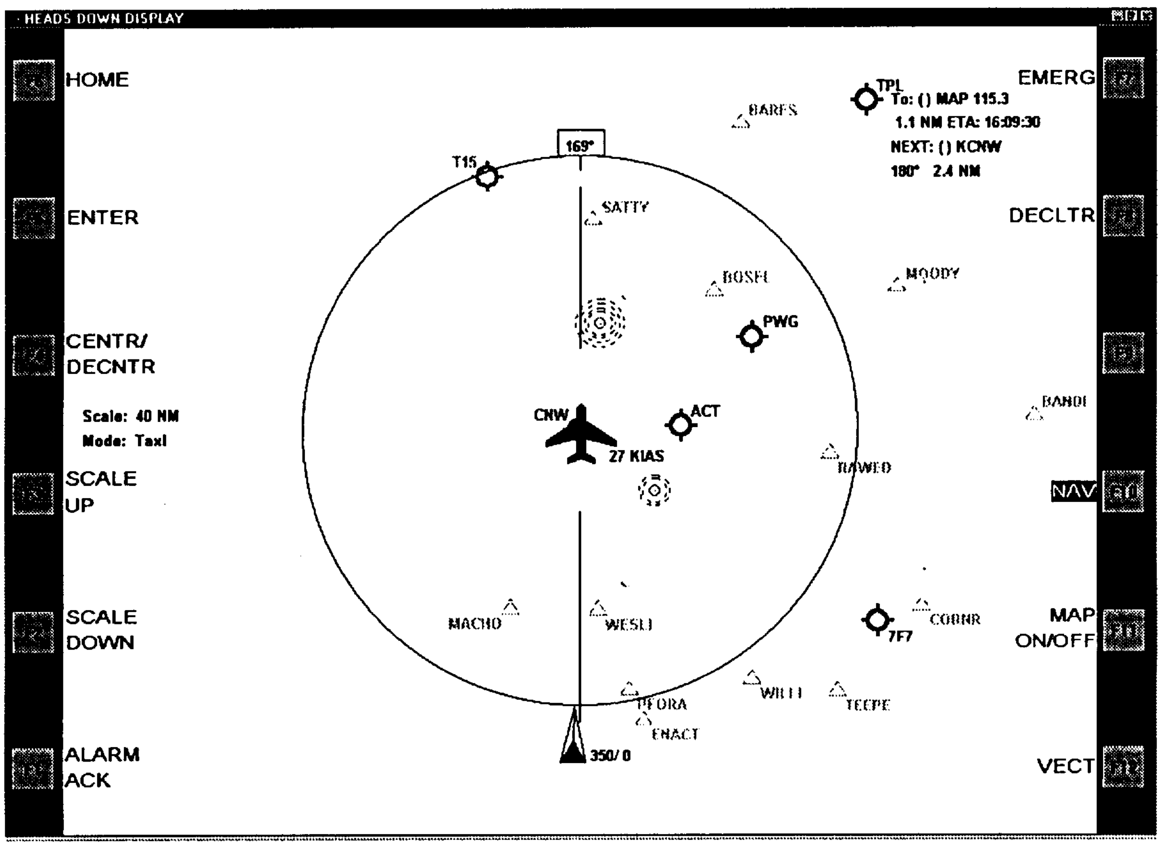

Figure 3. Example of Head-Down Display on Left Cockpit Monitor

Viewing alarm messages from the pilot advisor is also now possible. However, we still need a way to view all alarms in the PA queue. This functionality must be added to Trang's original design and then implemented.

The second function of the HDD is implemented using computer-generated instruments and gauges that emulate conventional pneumatic and electromechanical displays. These backup instruments are generated in a personal computer, driven by data from the simulator workstation through the TCP/IP processes, and displayed on a monitor on the right side of the cockpit. During this reporting period, the following basic instruments have been implemented: airspeed indicator, altimeter, vertical speed indicator, attitude indicator, turn and slip indicator, and heading indicator. A clock and a manifold pressure gauge are being debugged at this time. Development code is running with the appropriate data values driving the instruments and gauges.

Most of the effort in the simulator was directed toward completion of (1) the HDD communication link between the cockpit and the personal computer for the aircraft instruments and gauges and (2) the TCP/IP communication path between the simulator workstation and the personal computer which drives the HDD.

2.8 Simulator Modifications

This section documents modifications, both hardware and software, completed on the fixed-base simulator during this reporting period. The design and fabrication of the cockpit modifications have continued throughout this period and are now essentially complete (Figure 4).Those improvements necessary to simulate an instrument approach to any of the runways in the Waco, Texas, approach control area have been completed although the scenery for this environment is still rudimentary.

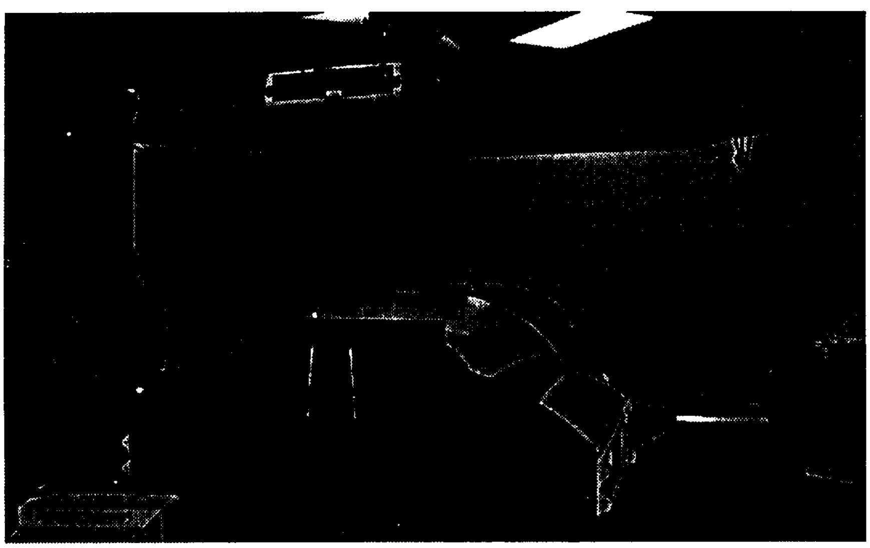

Figure 4. Overall View of Modified Simulator

2.8.1 Hardware

2.8.1.1 Cockpit Environment

The instrument panel, including Head-Down Display (HDD) selector switches, has been installed (Figure 5): the simulation, the display modes, and the GAPATS functions can be controlled through these switches. The control stick has also been installed in the left cockpit, and the throttles have been installed for both left and right seats. Additionally, brightness and contrast controls for each of the cathode ray tubes are located on the center throttle console. HDD information can now be selectively displayed on each of these monitors.

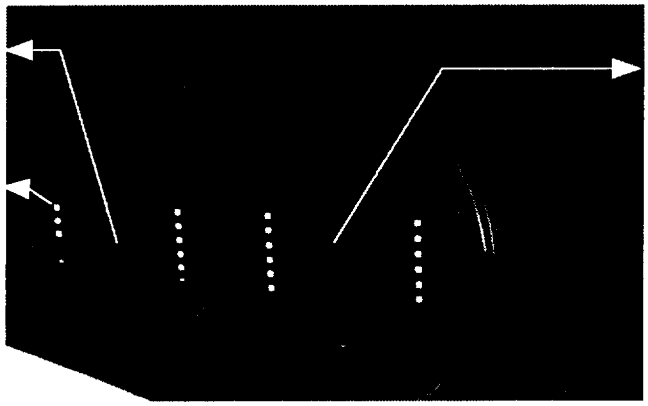

Figure 5. HDD and Selector Switches

Lighting and dimming controls for the HDD selector switches are in place, and the landing gear switch has been installed in the center of the instrument panel. Access platforms for entering and exiting the cockpit have been installed (Figure 6). With these capabilities the simulator is hardware-ready to support initial pilot evaluations of the GAPATS software even though cockpit refinements will continue to improve realism and functionality during the next reporting period.

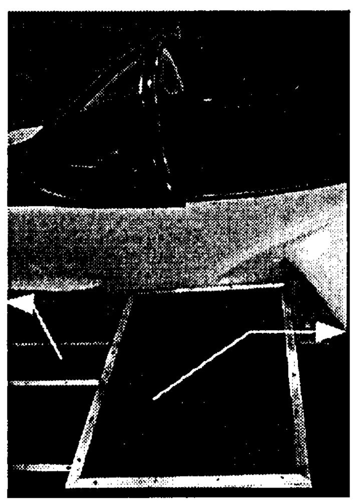

Figure 6. Access to the Left Side of the Simulator

2.8.1.2 Projection Subsystem

Installation and initial checkout of the projection apparatus was completed during the reporting period, as illustrated in Figure 4. The following list summarizes the tasks completed: (1) projectors installed in projector support structure; (2) preliminary focusing of projectors completed; (3) projector screens constructed, painted, and mounted; and (4) signal and power cables installed and tested. Design of remote power controls for the projection subsystem has been initiated.

2.8.1.3 Pilot Inputs: Data Handling and Conversion

The pilot's controls in the cockpit (stick, landing gear handle, flaps, throttles, etc.) output analog or digital information that must be converted to electrical signals representing positions and rotational angles. The sensors for this purpose all feed their information to a data handling and interface board (DHIB) installed in the right side of the cockpit nose (Figure 7). A serial port receives the signals from this interface board and transmits it to data addresses in the simulator's computer. This interface board allows quick connection of additional input devices along with a troubleshooting capability not found in the earlier simulator. Sensors connected to this interface board include (1) 24 HDD selector switches, (2) control stick switches, (3) throttle quadrant switches, (4) a landing gear switch, (5) a flap switch, and (6) a simulation reset switch.

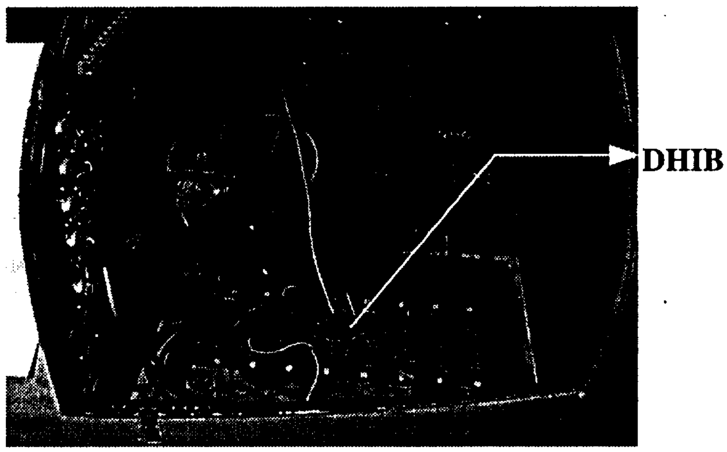

Figure 7. Data Handling and Conversion Compartment

An engineering development control stick, which permits crude control of the flight path outside the simulator cockpit (circumventing the pilot's stick and rudder pedals), was also completed early in this reporting period to allow development of simulator software and integration with GAPATS. However, installation of the cockpit controls has been the primary hardware accomplishment during this period. Optical encoders were added to the existing stick and rudder pedals. Hardware to connect the optical encoders to the control linkages was fabricated and installed. These encoders were then connected to a Dials and Buttons Box (DBB) for integration with the simulator software. The addition of the DBB reduced development time and allowed additional 32 switches for simulation control. Instrumented cockpit controls now include (1) sensors on the stick to provide pitch and roll commands, (2) sensors on the rudder pedal linkages to accept yaw commands, and (3) trim wheels to allow manual setting of trim about all three axes. The electrical trim switch on the top of the control stick was also activated although it still needs evaluation for rate of movement compared to that in the test airplane. Several such small optimizations of the trim system will be addressed during the next reporting period.

2.8.1.4 Artificial Feel Subsystem

To provide rudimentary artificial feel for the simulator, a simple spring-mass-damper linkage was designed and fabricated during this period. The design utilizes the aircraft control cables in the original trainer cockpit to position two pair of parallel springs and lever arms on the aft surface of the cockpit (Figure 8). This arrangement facilitates access to the artificial feel system for adjustment and provides ample space for a more sophisticated artificial feel system now being considered for future installation.

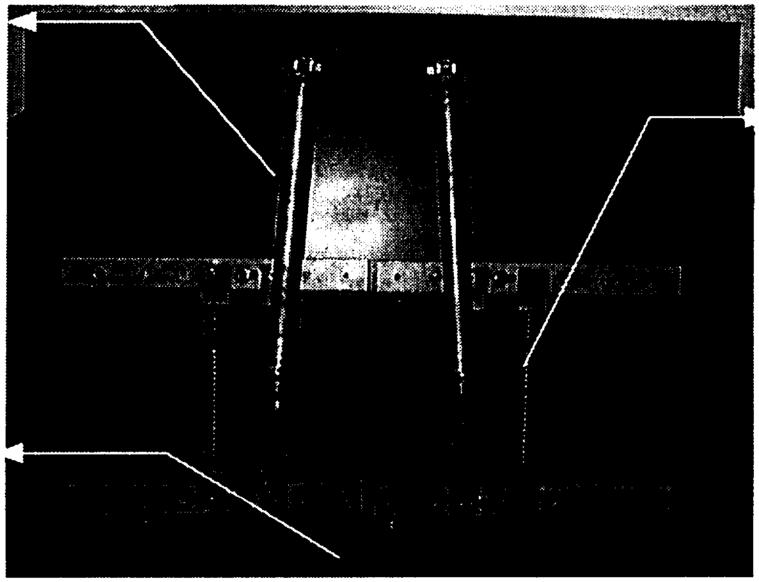

Figure 8. Artificial Feel Subsystem Components

The pitch and roll forces fed back to the pilot's stick are developed by springs that are in compression depending on the direction the control stick is deflected. Additional springs can be added to tailor these forces. A motorized stick trim could easily be incorporated to complete the artificial feel system. No artificial feel was added to the rudder since the existing spring system in the cockpit appeared to be adequate for this initial implementation. The location of the artificial feel system components does not interfere with optical encoders, and modifications to the artificial feel system can be made without damaging or relocating these sensors. The addition of the rudimentary artificial feel system gives the pilot adequate force feedback, though it does not provide forces that change with dynamic pressure that are experienced in an aircraft with a reversible control system.

2.8.2 Software

Software development for the simulator is grouped under six subtasks: executive, math model, scenery, simulator HUD (SHUD), HDD, and input/output. The computing engine is a Silicon Graphics Onyx Reality 2 workstation with 256 Mb of RAM, a serial input/output box, and a Multi-Channel Option (MCO). During this reporting period, a 9 Gb hard disk drive was added to the system to supplement the original 2 Gb one. The operating system is IRIX 6.2. As pointed out in the previous progress report, graphics frame refresh rate is a dominant performance parameter, with 30 frames/second or better on all three projection screens still mandatory. The executive module is the top-level source code that controls and calls all other routines. The math model describes the dynamic characteristics of specific aircraft, including their flight control system laws. The graphics modules of the software create a realistic scenery environment for the simulation. The SHUD, a computer-generated display superimposed on the graphics scenery, will be the primary display that the pilot uses to decipher the effects of pilot control actions and that the Pilot Advisor uses to communicate priority messages. The two monitors of the HDD are used to relay other, typically less urgent, parameters (expanded navigational information or GAPATS advisory messages, for example) to the pilot. The pilot controls the system through the HDD selector switches: for example, the pilot can choose to display formats or call up flight plan information. A numeric keypad is also needed for the pilot to enter flight plan information. The input/output module is the software code that acquires, scales, and passes control stick and rudder pedal movements, throttle settings, and other pilot inputs from the cockpit to the simulator's computers. During this reporting period, software development concentrated on three goals:

- developing the executive module,
- interfacing GAPATS software with the simulator, and
- ensuring that the cockpit control inputs were reasonable and compatible with the math model and with the TCP/IP protocol used to communicate all data to the simulator's workstation from the GAPATS personal computer.

2.8.2.1 Executive and User Interface

The executive module is the code that controls the entire simulation. Although written in C++, it must pass data between FORTRAN, C, and C++ routines while running multiple processes. The executive is derived from a code originally developed by Lockheed Martin Tactical Aircraft Systems[1]. Thus, many of the subroutines have a common ancestry, but they have been so heavily modified to suit our less complicated simulation environment that they now bear only passing resemblance to the original templates. During this reporting period, the main focus was on adapting this approach to GAPATS needs and, as detailed in other sections of this report, ensuring that GAPATS, which runs on a personal computer (PC), and the simulator workstation can communicate reliably and with low time delays. A TCP/IP connection has been chosen as the communication bridge between these machines. To utilize this TCP/IP bridge, a new software module was written for the simulator. This module spawns a connection process that links the PC and the workstation; then, the module continually sends and receives data until the session terminates. A common data structure between the TCP/IP process and the simulation process was produced during this reporting period and is currently being evaluated for robustness in preparation for the upcoming pilot evaluations.

The simulator module is responsible for updating data going to the PC, and the spawned process is responsible for updating data from the PC to the workstation. The workstation sends three types of data to the PC: trajectory data (indicated airspeed, altitude, rate of climb), navigational information (GPS latitude and longitude), and HDD selector switch selections. The PC sends HUD commands back to the workstation. Clearly, this TCP/IP data protocol must be tightly integrated with simulator software. Early results are encouraging: GAPATS appears to receive all the required data in a timely fashion, and the HUD changes format under to GAPATS mode-switching commands. Some delays in the data transfer process have been observed, and there have been a few system crashes, but generally this relatively new code has functioned satisfactorily during early checkouts.

In addition to the time devoted to the rather tedious installation and checkout of the data transfer protocol, we have developed a more straightforward way to initialize and control the simulator. As Figure 9 illustrates, the initialization (position, reference values for weight, initial altitude, and the like) now is accomplished by typing in values for each parameter on a Web-like interface designed specifically to help the simulation operator/analyst with the initial parameter set for any given simulation session.

Figure 9. Web-Based Operator Interface

Finally, considerable progress has recently been made toward adequate documentation of the simulation code. Drafts of both a system description (still lacking sections describing hardware) and a user's manual are now available. They have been reviewed once, and revisions have begun. Of course, finalizing this documentation will be heavily influenced by our experience with the pilot evaluations this fall; this intensive use of the simulation will undoubtedly tell us what additional revisions to the software are in order—as well as what is well-done.

2.8.2.2 Simulator HUD (SHUD)

The SHUD is designed to provide enough information for a pilot to fly a general aviation airplane safely in all weather conditions while still allowing the pilot to use visual cues outside the cockpit to locate the runway and to avoid conflicting air traffic. The basic display format (Figure 10) includes an aircraft reference symbol, a pitch ladder, an airspeed indicator, an altimeter, a vertical speed indicator, a magnetic heading indicator, a bank indicator, and a velocity vector or flight path marker.

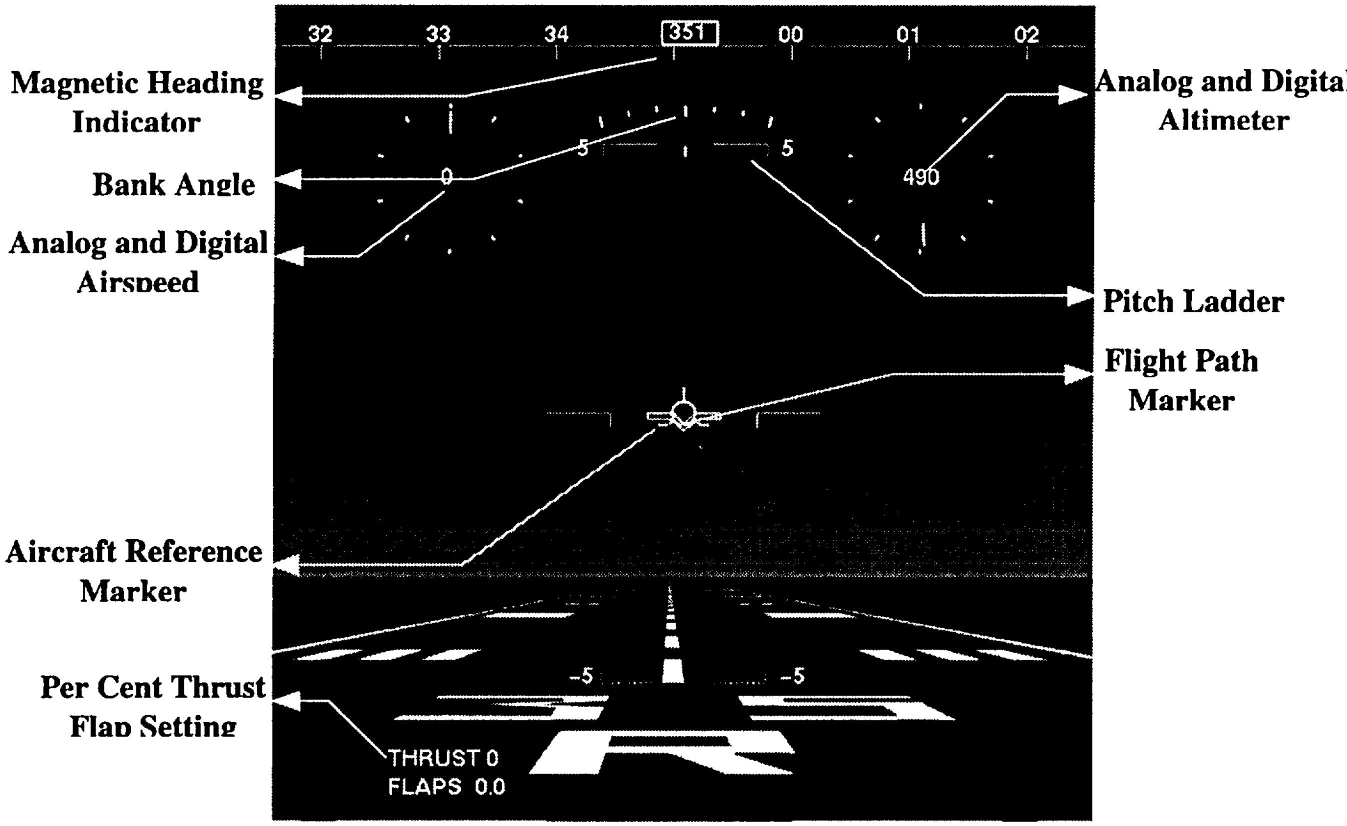

Figure 10. Basic SHUD

The airspeed indicator, the altimeter, and the magnetic heading indicator values are presented in both analog and digital form. Some symbology elements (magnetic heading indicator and bank indicator, currently) have been designed with multiple formats. Other elements will also be spelled out with a variety of formats. The best symbology combination will be sought during pilot evaluations. Such versatility and flexibility dictates that the SHUD software module be object-oriented to readily accept such variations. The module is also laid out so that altering only the head-up display configuration changes the symbology formats. It is not necessary to recompile the source code to evaluate different configurations.

Two approach formats[4] other than the basic one (Figure 10) are under active consideration, and other variations may also be examined. The ILS scale symbology is finished, and various guidance schemes utilizing the flight director are being scrutinized. A new culling and drawing algorithm is being developed so that objects like runway markers are truncated when part of the object is outside the field of view. The present algorithm shows the runway marker correct location and perspective only if the whole runway lies completely within (or completely outside) the field of view. This new algorithm will support the other two approach formats (Figure 12 and Figure 13).

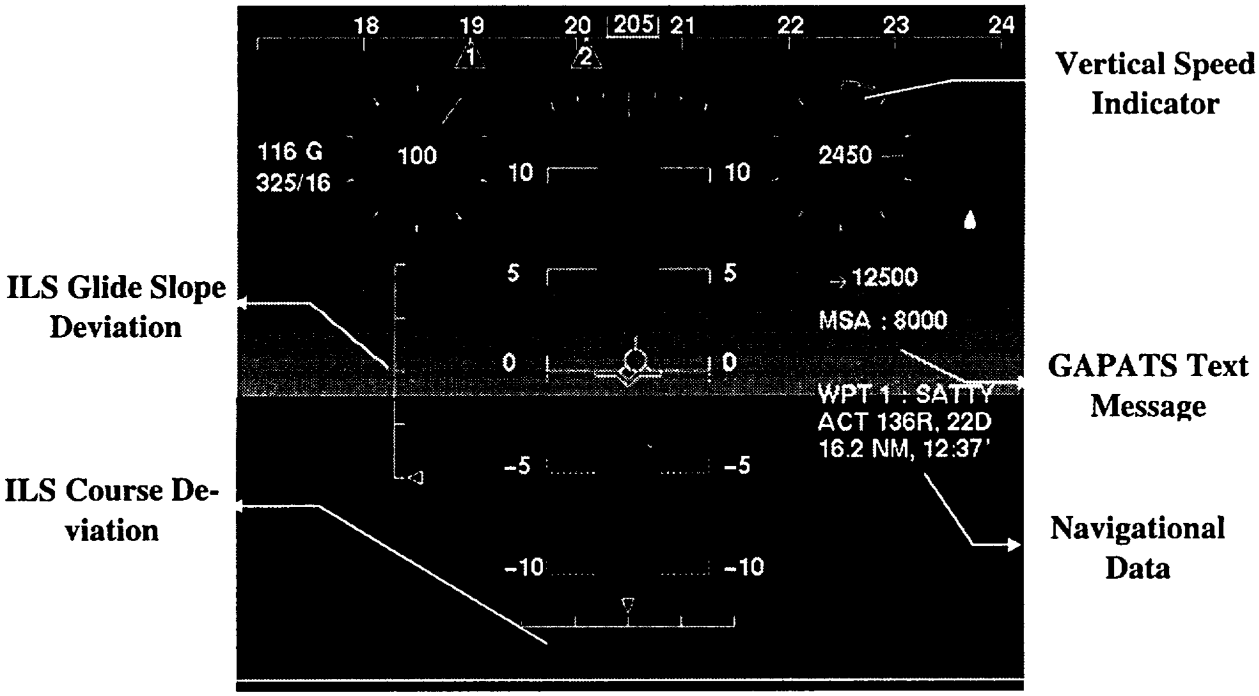

Figure 11. Alternative SHUD Configuration with ILS Glidepath and Course Deviations

The flexibility of SHUD formatting means that GAPATS can be a highly customizable program; the software can be designed to allow individual pilots to tailor their primary display to their preferences. These variations in SHUD displays must be supported by GAPATS, for development purposes at least—this characteristic may be a serious concern, however, when the software is undergoing certification with the Federal Aviation Administration. The development software handles this characteristic by controlling individual elements of the SHUD symbology with flags which set associated elements to be either displayed or not. Most of the messages generated by the Pilot Advisor currently are text messages displayed in the text message area (Figure 11).

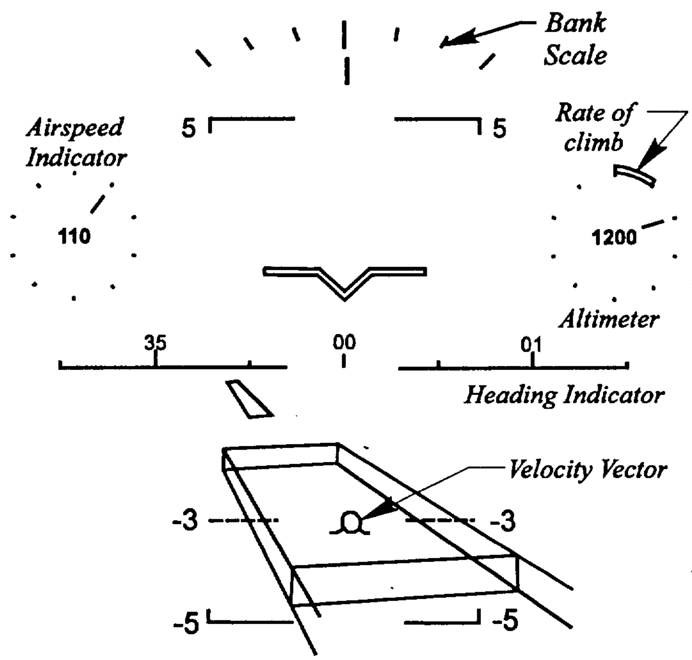

Figure 12. SHUD with ILS Depicted by a "Rectangular Box"

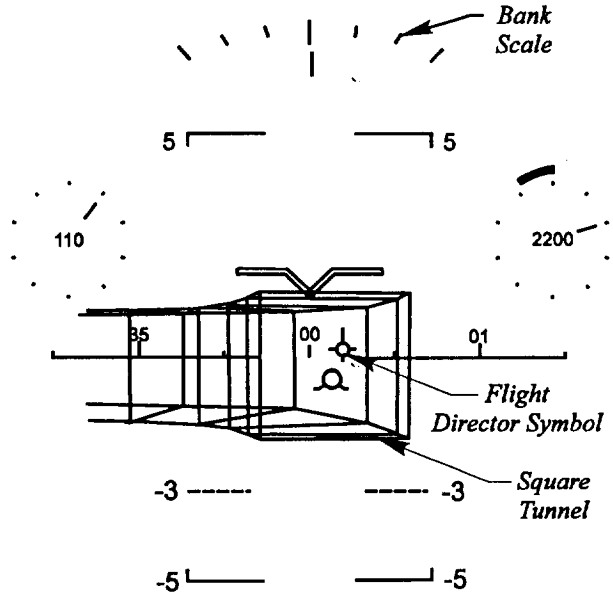

Figure 13. SHUD with ILS Depicted by a "Tunnel-in-the-Sky"

2.8.2.3 Scenery

The runway and taxiway complexes at both Waco Regional Airport and Texas State Technical College (TSTC) airports (at which we intend to conduct evaluation flights) were rendered and textured during this reporting period and are now being used for all evaluations. Both runways have the realistic markings but only the taxiway TSTC Airport is complete. All the terrain in the scenery is flat at an elevation of 470 feet MSL and has a grass texture applied. No actual elevation data have been added to the database as yet. A few buildings and roads are included in the scenery, but there are no trees. Runway lighting was temporarily removed because it dropped the frame rate drastically when first coded.

Complicated scenery with SHUD symbology drawn on top of it (using the draw callback function) suggests the need for a new scene hierarchy that can be easily maintained. Images projected on three screens give the pilot a realistic peripheral view; these multiple images also complicate the scenery generation. A common data structure for exchanging data among the subsystems is mandatory and has been coded. All the software modules (IO subsystem, math subsystem, graphics subsystem) update and transmit data through this common data structure.

An incompatibility originally existed between the TCP/IP protocol and the graphics rendering code (Performer). The aircraft simulator was originally coded to use a Pthread class. It was later discovered that the Performer rendering system was not able to coexist with Pthread architectures, causing intermittent core dumps by the simulator; thus an alternative solution was required. We investigated the use of Sproc to do the multiprocessing, and the source code for the server side application was modified. The occurrences of Pthreads were replaced with the corresponding code implementation of Sproc. Further code was written to implement the use of semaphores for data sharing.

2.9 Aircraft Modifications

This section describes progress made during this reporting period in preparing the Commander 700 light twin-engine flight research aircraft for GAPATS flight tests.

2.9.1 Sensor Suite

The sensor suite that acquires aircraft flight parameters for GAPATS has been described in previous semi-annual progress reports. During this reporting period, procurement of the sensor suite hardware and installation of most sensors have been completed. Table 2 summarizes the status of the sensor suite.

2.9.2 Computer Installation

The computer installed in the aircraft to serve GAPATS is a Pentium-based PC, driving a single flat panel HDD and a HUD and hosting data acquisition hardware. It is connected to an additional monitor and keyboard used by the Flight Test Engineer. All the hardware except the HUD is now available for mounting in the aircraft.

The HUD, the primary pilot display for GAPATS, is yet to be obtained although a promising source has been identified. This topic is discussed more completely in Section 3.1.

Preliminary design of the HDD and the HUD mountings has been completed. A flight test engineering station has also been laid out to house the computer, the monitor and keyboard. This station is designed to meet the FAA crashworthiness requirements since it will be mounted in the cabin, directly behind the evaluation pilot's seat. The station also houses the power supplies for the sensor suite.

Table 2. Status of the Sensor Suite

Sensor	Manufacturer/Model No.	Interface	Notes and Status
1. Pitot-Static System			
Airspeed pressure transducer	Omega PX162	to A/D card	existing
Altitude pressure transducer	Validyne P305D	to A/D card	existing
2. Flow Direction Vanes			
Optical encoders	U.S.Digital SP540BS43	to counter	existing
Digital counters	U.S.Digital LS7166	to D/D card	existing
3. Surface Positions			
Optical encoders	U.S.Digital E2-1024-250	to counter	existing
Digital counters	U.S.Digital LS7166	to D/D card	existing
4. Aircraft Attitude			
AHRS	Watson AHRS C-303+B52	RS232	to be installed
5. Engine Data			
MicroMonitor kit	Rocky Mountain Instrument (RMI)	Serial port	installed
Manifold pressure	Motorola		installed
RPM	RMI		installed
Fuel pressure	RMI		installed
Fuel flow	FloScan		installed
Oil temp.	RMI		installed
Oil pressure	RMI		installed
CHT			existing
EGT			existing
6. Navigation - GPS			
GPS unit	MentorPlus G100	RS232	to be installed
7.Gear/flap Data	Dedicated circuitry	to A/D card	to be installed
8. Navaid Data	Dedicated circuitry	to A/D card	to be installed
9. HDD			
Display	Lucas-Deeco Option 3032	to Controller	to be installed
Controller	Lucas-Deeco Option 2400	ISA slot	to be installed
Switches			to be installed
10. Data Acquisition			
A/D card	Computer Boards PCM-DAS16S/12	to card reader	existing
D/D card	Computer Boards PCM-D24C3	to card reader	existing
PCMCIA card reader	Envoy TMB260	ISA slot	to be installed

2.9.3 Data Acquisition

The Data Acquisition (DAQ) software and hardware in the aircraft link the sensors to the PC and process raw measurements into the form needed by GAPATS. In this period, DAQ requirements were finalized and the hardware procured; also, modifications were made to existing DAQ software to suit the needs of GAPATS. Installation of the hardware and software into the GAPATS PC is about to begin.

3. Current Challenges

The following sections describe challenges facing the team for the next reporting period.

3.1 Aircraft HUD

Obtaining viable HUD to fit into the Commander 700 cockpit instrument panel is a serious concern. At present, a verbal agreement has been reached for an indefinite loan of a HUD formerly used in the VTOL Systems Research Aircraft (VSRA) at the NASA Ames Research Center. This HUD is originally from an AV8A aircraft and, as such, is fairly complex with regard to interfaces, sheer size, and power requirements.

The dominating concern with this HUD is interfacing it with the GAPATS PC. The AV8A HUD package is composed of several components: the Pilot's Display itself, a Pilot's Control Panel, and a Display Waveform Generator, as well as several sensor boxes. For GAPATS the components required of this set are only the Pilot's Display and the Display Waveform Generator. However, the Display Waveform Generator must be driven by a Symbology Generator compatible with a PC while the video card in the GAPATS PC already drives the flat panel HDD and another conventional PC monitor. In the VSRA, a VME-bus card drives the Symbology Generator. Consequently, even if a suitable interface card is purchased or fabricated, there will still be a dearth of code to drive the HUD Symbology Generator. If multiple video outputs from the single PC are nonexistent, the only available option may be to use a dedicated PC, stripped of all non-essentials, to receive GAPATS HUD information from the GAPATS PC and send this information to the HUD's Display Waveform Generator through its dedicated video card.

The second concern is installation of the HUD's Pilot Display Unit (PDU). The PDU must be mounted with the display portion above the instrument panel glare shield and directly in front of the evaluation pilot. This unit is large, however, and this size may preclude use of most of the panel space below the PDU. This encroachment in turn necessitates shifting the aircraft HDD panel, from directly in front of the evaluation pilot to a position halfway between the two pilot's seats in the cockpit.

The third concern is the power supply for the HUD, which requires 200V/400Hz. One possible solution to this problem is the use of a dedicated inverter that runs off the aircraft 24VDC DC bus and provides 200V/400Hz power for the HUD.

3.2 Simulator Modifications

3.2.1 Hardware

3.2.1.1 Cockpit Environment

While the cockpit modifications are essentially complete, there are two challenges anticipated during the next reporting period. First, the toe brakes have not yet been instrumented and interfaced with the simulator software. This task must be completed before pilot evaluations can begin. Second, and of more far-reaching implications, the hardware in the cockpit has largely been manufactured locally or is refurbished equipment from suppliers. The reliability and maintainability of each of these components have yet to be established; periods of intense use (like the upcoming pilot evaluations) will provide a major test of this not-so-new equipment. When malfunctions do occur, there will inevitably be delays in obtaining replacement hardware and altering the configuration to accept parts that may be slightly different from the currently installed components.

3.2.1.2 Projection Subsystem

Adjustments to the projection system are extremely tedious. While the primary projector has been focused, additional work remains to improve the image generated by the left and right projectors. The output from the left screen projector suggests strongly that this projector is marginal; it will likely be replaced with a similar unit owned by the Aerospace Engineering Department. The project manager has negotiated to swap out these two projectors early in the next reporting period. Another enhancement planned for the next reporting period is the design of a centralized power switch for the projectors to make it easier to adjust them and to remotely control their operation.

3.2.1.3 Pilot Inputs: Data Handling and Conversion

Random digital signal activation leading to system crashes and core dumps has occurred, and we are trying to isolate causes by tracing wiring and troubleshooting. Lack of spares for most components in this subsystem is a real concern; failure of the DHIB or the DBB, for example, will bring GAPATS development on the simulator to a halt. If such failures occur during a critical time (like the pilot evaluations), this reliability and maintainability issue could become dominant.

3.2.1.4 Artificial Feel Subsystem

The current artificial feel subsystem is a short-term fix; it has limited capability to efficiently handle large stick force gradients. While this limitation may be of secondary importance to GAPATS development aimed at general aviation airplanes where these gradients are relatively small, eventually a more complete and complex artificial subsystem will be required. Components from a NASA Dryden simulator may be available at a later date. The rudimentary artificial feel design that is in use allows later incorporation of these or similar components, but the changes will mean another major modification. Alternatively, there is a parallel effort proposed to locally develop a low-cost, variable feel system based on "damping-on-command" magneto-rheological (MR) fluids[2,3], allowing simulator feel to be reconfigured electronically. Obviously, such an effort is a long-term solution to improving the fidelity of the artificial feel subsystem for the simulator, making it more attractive for cockpit software development.

3.2.2 Software

3.2.2.1 Executive and User Interface

This module is essentially complete though there are undoubtedly modules that will have to be added. The structure is in place, however, and the challenge in this area is to be sure that new personnel (see Section 3.3) learn the architecture as rapidly as possible so they can fix errors, modify modules for added capabilities, and add new code when it is needed. In short, this area is largely ready for maintenance tinkering, and little new code needs to be generated.

3.2.2.2 Simulator HUD

The dynamical equations that presently control the approach symbology in the SHUD (the runway outline, the flight director cue, and the "tunnel-in-the-sky", for example) are not general enough for all flight conditions and not robust enough to satisfy a large group of evaluation pilots. Developing a more flexible set of equations to control this approach symbology is the most important challenge for this aspect of the simulator code during the next reporting period.

3.2.2.3 Scenery

Building up a reasonably realistic set of objects in the Waco Approach Control area is the immediate and urgent problem to support the GAPATS pilot evaluations. This aspect is well behind schedule and must be emphasized, especially since the current set of software tools make scene rendering and texturing very time-consuming. Finally, great care will be necessary to keep the screen refresh rate at or above 30 Hz; optimization of scenery rendering is very much a trial-and-error process.

3.3 Personnel

The project is not on schedule as originally envisioned. The main reason is an unexpected loss of personnel from KBSI. We have hired new personnel and are now back on track, but we have some ground to make up during the next period. Some of the other work has slowed due to the lack of people to work on a key piece of the architecture, namely the TCP/IP connection between the simulator and GAPATS. This has been remedied, and the connection seems to be working although there may be latency problems that will need to be investigated.

Trang completed his analysis and documentation of the functional requirements for the GAPATS system and displays during this reporting period. He has since departed for his next Army assignment. His work was significant and timely in that it brought to the project the talents of a second military test pilot just when they were needed. His contribution was broad and far-reaching, ranging from display design to augmentation of the FMI functionality through incorporation of distance measurements.

Dr. W. Kelly, the Phase-2 implementer of the FMI, has completed his work and departed to industry. His work was also broad in that he not only completed the design and tuning of the FMI but conceived the need and provided the design for the Data Object, key to allowing parallel development of the several software modules. The Data Object makes the GAPATS truly object-oriented. Dr. Kelly's second key contribution was development of a MATLAB© design visualization tool for Fuzzy Membership Functions. This tool allowed the Fuzzy FMI design to be tuned from actual flight simulator recorded data.

In the Electrical Engineering Department of TAMU, K. A. Lee and P. A. Branham remain with the project, completing the implementation of the Navigation Module (NAV) and Head-Down Display (HDD), respectively. These modules are currently operable, sufficient to credible flight simulator demonstration, with their respective functionalities now being completely fleshed out.

The Aerospace Engineering Department lost a key person when D. Robbins completed his degree and left in early August to begin a graduate program at Virginia Tech. The lead software engineer developing the simulator executive and the mathematical model, he was also intimately associated with most of the hardware modifications carried out during the last two reporting periods. We also expect W. Alcorn to complete his degree and leave by the end of the next reporting period. In anticipation of these departures, three students were added to the GAPATS team during this reporting period to work on the simulator: N. Duangsungnaen, J. Hull, and S. Shandy are all assigned to the simulator development and maintenance effort. Since the modifications to the airplane will become an area of intense effort, especially as soon as the HUD unit arrives, K. Krishnamurthy's efforts have been augmented with part-time help from J. Pan and G. Ziegler. Quite clearly, a continuing major challenge is the turnover in knowledgeable personnel during the GAPATS development life—an inevitable consequence of having significant student population on the team. But the quality of their contributions generally has been quite high, and the learning curves for these new students are quite promising.

3.4 Navigation Module

Currently, the navigation database is the most challenging item. The database contains many different types of records that must be translated into C++ classes. Also, each record type contains more information than is required for the Navigation Module's duties. Thus, determining which fields to include from each record type will be necessary. This task requires complete knowledge of how the information contained in every record will be used. Such knowledge is common to experienced pilots, but not readily apparent to the software engineer.

4. Planned Efforts for Next Period

This section of the report summarizes the tasks planned for the next period.

4.1 Simulator Modifications

This section of the report summarizes the tasks planned for the simulator modifications.

4.1.1 Hardware

4.1.1.1 Cockpit Environment

The following list summarizes hardware development associated with the cockpit:

- Install and integrate of toe brake sensors and software handlers.
- Install numeric keypad for pilot data entry into GAPATS.
- Document all modifications (especially wiring) for future use and maintenance.

4.1.1.2 Projection Subsystem

The following list summarizes the planned development of the projection subsystem:

- Replace of the defective projector and final focusing of the system.
- Develop control cables and computer interface software for control of projectors.
- Document all modifications (especially wiring) and maintenance actions required.

4.1.1.3 Pilot Inputs: Data Handling and Conversion

The primary components of the data acquisition subsystem have been completed. Future work will focus on the following tasks:

- Troubleshoot the system to determine and eliminate sources of random signals.
- Document of all modifications and maintenance actions.
- Possibly replace the current optical encoders with higher resolution ones.

4.1.1.4 Artificial Feel Subsystem

The planned work on the artificial feel subsystem involves one task: documentation of the subsystem (especially wiring) and all required maintenance actions.

4.1.2 Software

4.1.2.1 Executive and User Interface

The following short list summarizes the planned work on these software modules:

- Fix bugs and add routines to accommodate changes (like toe brakes); that is, maintain the software as users ask for changes.
- Document the software; have all new personnel carefully review the documentation and make timely revisions.

4.1.2.2 Simulator HUD

Dynamical equations must be added to drive approach symbology on the SHUD:

- Drive the flight director to produce desired course interceptions with minimal overshoots and reasonably captures.
- Generate governing equations and program them to mark curved approach paths (as is needed for viable "tunnel-in-the-sky" or similar symbology formats.
- Control the position of the runway outline so it will overlay the position of the runway whenever an evaluation pilot breaks out of cloud.
- Maintain the software as users ask for changes.
- Document the current code and spell out the procedures used to govern all symbology elements on the SHUD.

4.1.2.3 Scenery

The following list identifies the needed buildings and terrain features; care must be exercised throughout to be sure the screen refresh rate does drop as complex scenery is added.

- Runway lighting to include VASI and runway threshold areas especially.
- Terminal buildings, hangars, and control towers.
- Lakes, rivers, streams, and trees.
- Terrain elevation.
- Simulated fog, clouds, and night scenes.

Documentation describing the procedures used in scenery generation must also be produced.

4.2 Flight-Mode Interpreter

The design and implementation of the Flight Mode Interpreter is complete. No further effort is contemplated, unless dictated by results of Flight Simulator or Aircraft tests.

4.3 Navigation Module

Integration of the navigation database into the Navigation Module will be accomplished. This task requires conversion of the ASCII data base records to binary instances of C++ classes. The necessary supporting coding will be performed in the upcoming period. Once completed, the navigation database will be integrated with the flight plan class, allowing the pilot to select records from the database for inclusion in a flight plan differing from the default scenario.

4.4 Head-Down Display (HDD)

During the next period, the HDD should essentially be complete. The areas of primary focus currently are integration of the Navigation database, implementation of subscreens, development of a "declutter rule base" for the moving-map, development of a customization screen (the DISPLAYS screen), and entry and integration of data from the Commander 700 manual. Data from the navigation database are necessary for the moving-map display. The moving map currently uses hard-coded locations for objects in the Waco area. Subscreens will be added to Trang's original design for such uses as selection boxes and the alarm queue viewer. A "declutter rule base" will be designed for controlling the moving-map. The rule base will be written in C++ but may be subject to implementation in Clips in the future. The pilot will use a customization screen to change such parameters as display colors, moving map scale levels, moving-map cluttering levels, and if physically possible, brightness and contrast. Finally, data must be gleaned from the Commander 700 manual and manually entered into the HDD code. These data include moment and weight data and an appreciable amount of Check List data.

4.5 Pilot Advisor

During the next period, the Pilot Advisor will be tested in the simulator with various pilots. Their feedback will drive the next iteration of rules for the advisor. This step-wise refinement will continue throughout the next period as we move the system onto the aircraft. In addition, the code for the advisor will be rigorously reviewed to identify and correct any problems. Because we expect that several other modules will be integrated during this period, the Pilot Advisor must be able to accommodate the new modules.

A thorough testing of the current PA rule base is in our plan. We have been able to test the rule base with existing flight data. However, the flight data may not contain all the extreme cases that we would like the rule base to detect. Therefore, we need to obtain more flight data for the testing.

An enhancement of the PA rule base is also in our plan. The current PA rule base considers only a portion of factors that are critical to flight. In our plan, we would consider other critical factors such as calibrated airspeed, heading, pitch, roll, thrust, and altitude if the situation allows. The enhancement of the rule base should be straightforward. We would follow the current style to add those newly considered modules into the rule base. The integration between the flight simulator and the PA does not need any modification.

4.6 Aircraft

A significant amount of work remains to be completed on the Commander 700 aircraft to prepare the aircraft for GAPATS pilot evaluation tests. The main tasks appear below:

- Interfacing the installed sensor suite hardware with the data acquisition code on the on-board computer
- Procuring and installing the Head-Up Display in the cockpit

- Installation of the Head-Down Display on the aircraft's instrument panel
- Completion of the Flight Test Engineer's station; involves installation of the computer along with associated hardware interfaces and power supplies
- Installation of the GAPATS system on the onboard computer
- Preparation of Flight Test Cards for the Evaluation Flights
- Documentation of all modifications for future use and maintenance

References

1 ATLAS User's Manual, Lockheed Martin Tactical Aircraft Systems, Fort Worth, TX, August 1996.

2 "Rheonetic™ Systems: A New Generation of Control Products," Lord Corporation, 1997, pp. 4-10.

3 Strganac, T. W. and Junkins, J. L., "Magneto-Rheological Materials Research," Proposal for the State of Texas Advanced Technology Program, Austin, Texas, July 1997.

4 Ward, D. T. and Woo, D. L. Y., "A Simulator Study of Advanced Head-Up Display Symbology for Use in General Aviation Aircraft", AIAA Paper 97-0224, 35th Aerospace Sciences Meeting and Exhibit, Reno, Nevada, Jan. 1997.

Distribution List

NASA Center for Aerospace Information (CASI)
Attn.: Accession Department
800 Elkridge Landing Road
Linthicum Heights, MD 21090-2934

NASA Langley Research Center
Acquisition Division
Attn.: Roberta Knight, Mail Stop 126
9A Langley Boulevard
Hampton, VA 23681-0001

NASA Langley Research Center
Contracting Officer Technical Representative
Attn.: Paul Schutte, Mail Stop 152
9A Langley Boulevard
Hampton, VA 23681-0001

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)

2. REPORT DATE
16 September 1997

3. REPORT TYPE AND DATES COVERED
SemiAnnual Technical Progress 1/26/97-7/25/97

4. TITLE AND SUBTITLE
General Aviation Pilot Advisory and Training System (GAPATS)

5. FUNDING NUMBERS
Contract Number:
NAS 1-20593

6. AUTHOR(S) Dr. John Painter, Dr. Donald T. Ward, Dr. Wallace Kelly, Mr. John Crump, Mr. Ron Phillips, Major Jeff Trang, Mr. Kris Lee, Mr. Paul A. Branham, Mr. Karthik Krishnamurthy, Mr. William P. Alcorn, Jr., Mr. Andrew P. Robbins, Mr. Ren-Jye Yu

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Knowledge Based Systems, Inc.
One KBSI Place
1500 University Drive East
College Station, TX 77840-2335

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
National Aeronautics and Space Administration (NASA)
Langley Research Center

Hampton, VA 23881-0001

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

This report documents the third six months (January 26, 1997 through July 25, 1997) of work performed for the Phase II NASA STTR project, a General Aviation Pilot Advisory and Training System (GAPATS), being conducted by Knowledge Based Systems, Inc. and Texas A&M University. The purpose of this project is to develop and integrate the technology necessary to provide the general aviation pilot with increased situational awareness. GAPATS employs artificial intelligence to determine flight operations being performed by the pilot and the aircraft. Based on AI Algorithms and pilot input, the system provides the pilot with a critique of present performance and procedural advice, without adding to pilot workload, thus increasing safety, efficiency, and operational precision. GAPATS employs the most modern software engineering techniques: object-oriented design/programming, parallel software architectures, and fuzzy logic. During this period, work primarily focused on enhancing PC/Simulator Communication, the Flight Planner/Navigation Module, the Situation Recognizer, and the Pilot Advisor. The GAPATS team also upgraded the Engineering Flight Simulator and the Sensor Suite.

14. SUBJECT TERMS
Aviation, pilot advisory systems, pilot training systems, flight simulation, artificial intelligence, object-oriented design/programming, and fuzzy logic

15. NUMBER OF PAGES
28

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	